Approximating incident occurrence time with a change-point latent variable framework

Francisco Câmara Pereira
Singapore-MIT Alliance for Research and Technology, Future Urban Mobility
1 CREATE Way, #09-02 CREATE Tower, Singapore 138602
Tel: 65-6601 1547, Fax: 65-6778 5654
Email address: camara@smart.mit.edu

Oren Lederman
Singapore-MIT Alliance for Research and Technology, Future Urban Mobility
1 CREATE Way, #09-02 CREATE Tower, Singapore 138602
Tel: 65-6601 1547, Fax: 65-6778 5654
Email address: orenled@mit.edu

Moshe Ben-Akiva
Massachusetts Institute of Technology
Room 1-181, 77 Massachusetts Avenue, Cambridge, MA, 02139
Telephone: 617.253.5324
Email address: mba@mit.edu

4944 words + 7 figures + 2 tables= 7194 (limit=7500 words)
October 28, 2013

ABSTRACT

We propose a methodology to approximate actual incident occurrence time by analyzing downstream volume sensor data. We model the time difference between actual occurrence time and reported time (or *delay*) as a latent variable that becomes a parameter in a change-point time series model. We then apply a *maximum a posteriori* (MAP) framework to infer the most probable delay. This MAP framework uses the time series model as the likelihood function and a bayesian prior based on field knowledge.

We applied our model on 5 months of traffic sensor data and accident reports from 3 Singapore expressways and corrected the accident start times for 1086 accidents in total. We compared the results with a manually constructed baseline and obtained a mean absolute error (MAE) between 5.7 and 7.4 minutes and a root mean squared error (RMSE) between 10 and 12.

INTRODUCTION

By their nature, accurate traffic incident occurrence is difficult to detect both spatially and temporally. Plenty of intelligent transportation systems (ITS) research exist that focus on incident detection (e.g. Weil et al. (*1*), Karim and Adeli (*2*) and Tang and Gao (*3*)) but the focus is mostly on detecting that an incident *has* occurred, rather than precisely *when* it happened. The problem is particularly complex whenever the incident occurs away from sensors or when the overall scenario is highly prone to sensor noise (e.g. due to harsh weather, sensor quality, traffic conditions).

The detection and consequent reporting of incidents to the traffic managers can happen in different ways, namely via the above mentioned ITS incident detection systems as well as driver reporting (e.g. people involved in the accident), personnel reporting (e.g. traffic police) or through camera monitoring in the traffic management center. The actual incident occurrence is very often not observed at all.

Accurate information of incident occurrence time can be relevant in a few situations. Traffic prediction systems can improve their performance by incorporating incident details at the right time and place. Without such information, they may rely on wrong assumptions thus generating faulty results. Even with an efficient self-adaptive mechanism, they need time to correct the parameters, particularly under complex scenarios. For example, dynamic traffic assignment (DTA) models need to update capacity parameters accordingly for affected areas. Data-driven algorithms (e.g. Neural Networks, ARIMA) need to adapt to different *regimes* (e.g. Antoniou et al. (*4*)). Of course, this incident information will arrive itself with some delay, and this needs to be considered by the traffic prediction algorithm itself (e.g. by a “roll-back" mechanism).

The ability of traffic prediction engines to roll-back and re-generate their calculations is fundamental because they rely on spatial an temporal correlations, i.e, the error due to lack of incident information will be propagated unless the system integrates it properly in the right time.

The second general motivation regards to post-hoc incident analysis. From the point of view of traffic emergency management, it is important to assess the performance of incident response systems, and more accurate information will lead to better informed decisions in a context where timing is crucial.

A third reason, yet very specific to our case, is to correct incident duration information in our automated incident analysis framework where we estimate capacity reduction and incident clearance duration sequentially in time Pereira et al. (*5*). These two variables alone, capacity reduction and incident duration, sufficiently specify the role of an incident for traffic prediction systems, particularly when based on simulation models (e.g. DynaMIT, from Ben-Akiva et al. (*6*)).

In this paper, we will focus on estimating actual incident occurrence time, t_{occ}, at some reporting time t_0 or later, where obviously $t_0 > t_{occ}$. Our incident start detection model will rely on a signal feed that consists of volumes aggregated by 5 minutes intervals, as observed by the closest sensor downstream to the incident. This specific setting is determined by our case study, the Singapore expressways, from which we have volumes for a period of 5 months.

Our task can be described as follows. At time t_0, a report is received about an incident that occurred at a certain location. We have a feed of volume information from the sensor downstream to that location, aggregated by 5 minute intervals. We want to determine the most likely period when this incident has occurred. For the purposes of this paper, we assume complete availability of data (before and after time t_0), leaving for future work a real-time sequential version, where this process is run at time t_0 and subsequently as new information arrives.

More often than not, the incident will happen far from the downstream sensor, so its impact on the volume signal will be itself delayed. In our case, due to the short distance between sensors in the Singapore expressways, it has been observed that this delay is negligible, particularly taking into account the 5 minutes aggregation Mak (*7*).

Another note relates to the reporting times. Even though traffic entities give their best to efficiently streamline the process, the heterogeneous characteristics of the incidents and their response sometimes lead to considerable delay in the reporting itself. Besides the detection delay, there may be other operational sources of delay that are determined by the incident characteristics (e.g. the traffic police may detect it soon but first try to unblock the road or provide local assistance; in case of multiple incidents, some reports may be keyed in the system with more delay). As a result, we may observe significant differences between reported time and actual incident occurrence.

We propose to approach this problem in the following way:

- each accident has its time series of volume data from the downstream sensor, we translated this into flow. The actual accident occurrence should originate a change in the flow time series parameters. This corresponds to the concept of “change point" in time series analysis literature;
- the true accident occurrence time is, in general, unobserved or *latent*;
- we will determine t_{occ} through a latent variable whose best approximation should support the maximum likelihood of the observed time series of flow for the downstream signal;
- since we do not have the ground truth values for t_{occ}, we will build a manual (visual) baseline of the start times from the time series signals.

The next section will be dedicated to the literature review, followed by the description of the methodology (Section 4). In Section 5, we describe the experiments. The paper will end with a discussion (Section 6) and the conclusions (Section 7).

LITERATURE REVIEW

Literature exclusively about inference of incident start times is not abundant, but, on the other hand, there exist many works on the related topic of incident detection. In this review, we will approach these two major topics.

Within an analysis of incident detection methods in a Singapore expressway, Mak (*7*) verified that report times do not reflect the actual start time of accidents. And, due to the delay/time lag between reported and actual start time, simply using the reported start time would lead to misclassified traffic patterns and affect the accuracy for incident detection models. On the other hand, he concluded that, based on the distances between the accident and the adjacent detector, the time it takes for the disturbance to reach the sensor falls between 0.4 and 1.39 minutes.

Finding the incident occurrence time in the sensor signal is however a difficult challenge. Dia (*8*) proposes that a 20% disturbance in traffic parameters (speed, occupancy and volume at the upstream station) could be used to indicate either the start or end of an incident. According to (7), compared to a visual inspection, this method is more efficient and requires less experience from the user. However, in his experiments the two methods gave different start and end time values.

Besides, a strict threshold value may be too strong a constraint if we also consider other aspects, such as sensor quality, general traffic conditions, weather status or time of day.

Rather than taken by itself, the task of incident start time inference has often been integrated within automatic incident detection and incident analysis works (e.g. (9)). In these cases, the manual analysis is chosen for practical reasons: incident start times are only needed to help calibrate a classification algorithm (of incident detection), not being a necessary input of the model. In fact, it is arguable that, in some applications, the exact start time is of secondary importance. On an operational setting, the need is to detect the occurrence and characterization of incidents as soon as possible, regardless of its precise on-set time. Although the start times are not secondary in incident analysis, the traditional practice is still to manually analyze the signal (e.g. using 5D stacked bar charts (Lee et al. (*10*))) or use simple heuristics such as mentioned earlier (Dia (*8*)).

However, there are situations where approximating the start time is important. For example, Jeong (11) explains that recent studies to validate AID algorithms had to rely on simulated data since reported start time normally maintained by freeway patrols and incident management systems is not precise. The difference might be a couple of minutes of even more, and creates an undesirable shift in the incident data.

In traffic prediction too, an accurate characterization of incidents is important. For example, DynaMIT (6) uses such information to roll-back the network state estimation process with revised assumptions about capacity reduction on the affected links. By doing that, the system will better understand the flow changes and generate more accurate predictions. Notice that, in such a context, one needs a fully-automated system for incident start time inference rather than one base on manual/visual analysis.

A final motivating argument for incident start time inference relates to incident analysis, particularly related to clearance duration and response times. Traffic agencies regularly need to assess and revise their incident management procedures and wrong incident start times will obviously affect accuracy of such evaluations.

It is thus remarkable that little more has been done on this specific topic despite its relevance. This may be explained by the almost total lack of observability of incident occurrence times and for the higher focus on incident detection systems (which do not necessarily need to know *when* it happen, rather they are focused on that it *had* happened) rather than for traffic prediction. Current growing emphasis on traffic prediction services may possibly reverse this trend.

METHODOLOGY

Our goal is to estimate the start time of an incident, given a time series signal with traffic flow data. It is known that an accident *has* occurred as well as its reported location. We are also given the sensor data from the vicinities of the accident.

From the point of view of the signal analysis, we can think of two general methodologies: heuristics application; anomaly detection. The heuristics solution follows the line proposed by Dia (8), that assumes a 20% disturbance in traffic parameters. As with others (e.g. *largest flow drop*), this rule is obviously too rigid, and this threshold would probably vary from dataset to dataset. Even with the best choice of threshold, one would face problems in applying the rule. For example, what would the time window be? Incident disturbance may be gradual through time, or be abrupt and recover quickly.

The anomaly detection approach defines the model of *expected* behavior and captures deviations as being anomalies. It lends itself to a more flexible approach than using heuristics in the sense that it does not expect a specific threshold or a deterministic rule.

In our case, we know that an accident occurred and expect it to somehow impact on the traffic signal. We prefer not to make strong heuristic assumptions but still need to assume that there was in fact an observable change in the signal. A concept that fits nicely with this description is that of *time series change-point*: when different subsequences of a data series follow different statistical distributions, commonly of the same functional form but having different parameters (12). In other words, an incident should originate a change in the traffic time series signal parameters that lasts during a period of time. It is also plausible that this change ranges from a simple shift in the mean to general parameter redefinition.

Since the actual incident occurrence time is not observed, we will determine it through a latent variable. Our task is thus to obtain the most likely value for this variable given the change-point model assumption that the signal before the incident occurrence time will have a certain set of parameters, and after it will have another set of parameters.

Formally, our goal is to find the incident occurrence time t_{occ}, which is determined by $t_{occ} = t_0 - z$, where t_0 is the reported time and z is the latent *delay* observed from the signal. This delay in practice corresponds to the reporting delay, and it is in fact our latent variable. We want to maximize the probability $p(z|\mathbf{y},\theta)$, such that

$$p(z|\mathbf{y},\theta) \propto \mathrm{p}(\mathbf{y}|z,\theta) * \mathrm{p}(z)$$

with $\mathbf{y}$ being the vector of time-series signal and θ the vector of parameters. $p(z)$ is the bayesian prior for the latent variable z, which will be discussed later. Notice that we can obtain the exact posterior probability for certain z_i by normalization:

$$p(z_i|\mathbf{y},\theta) = \frac{\mathrm{p}(\mathbf{y}|z_i,\theta) * \mathrm{p}(z_i)}{\sum_{\mathrm{j}} \mathrm{p}(\mathbf{y}|z_j,\theta) * \mathrm{p}(z_j)}$$

In general, we are looking for maximizing the probability of z (a process also known as *maximum a posteriori*, or MAP)[1]. Thus, we will vary choices for θ parameters as well as for the value of z. Formally, we want:

$$\underset{\mathrm{z},\theta}{\mathrm{argmax}}\, \mathrm{p}(\mathbf{y},z|\theta) = \underset{z}{\mathrm{argmax}} \left[\mathrm{p}(\mathbf{y}|z, \underset{\theta}{\mathrm{argmax}}\, \mathrm{p}(\mathbf{y}|z,\theta)) * \mathrm{p}(z) \right]$$

The likelihood, $p(\mathbf{y}|z,\theta)$, is expanded as

$$p(\mathbf{y}|z,\theta) = p(\mathrm{y}_{t_0-l}, \mathrm{y}_{t_0-l+1}, \ldots, \mathrm{y}_{t_0}, \ldots, \mathrm{y}_{t_0+r-1}, \mathrm{y}_{\mathrm{t_0+r}}|z,\theta) \tag{1}$$

where l and r correspond respectively to the left and right time window boundaries to inspect. Notice that, if the algorithm is intended to run in real-time, r will be constrained to the most recent data. The likelihood function is determined by the typical multivariate gaussian, such that

$$ln(p(\mathbf{y}|z,\theta)) = -\frac{1}{2}\ln((2\pi)^{\mathrm{n}}|\Sigma|) - \frac{1}{2}(\mathbf{y}-\hat{\mathbf{y}})^{\mathrm{T}}\Sigma(\mathbf{y}-\hat{\mathbf{y}}) \tag{2}$$

[1] In fact, the posterior distributions of z should not be ignored since they carrier potentially relevant information about uncertainty of the start time approximation. However, for simplicity in this paper, we will only focus on the MAP.

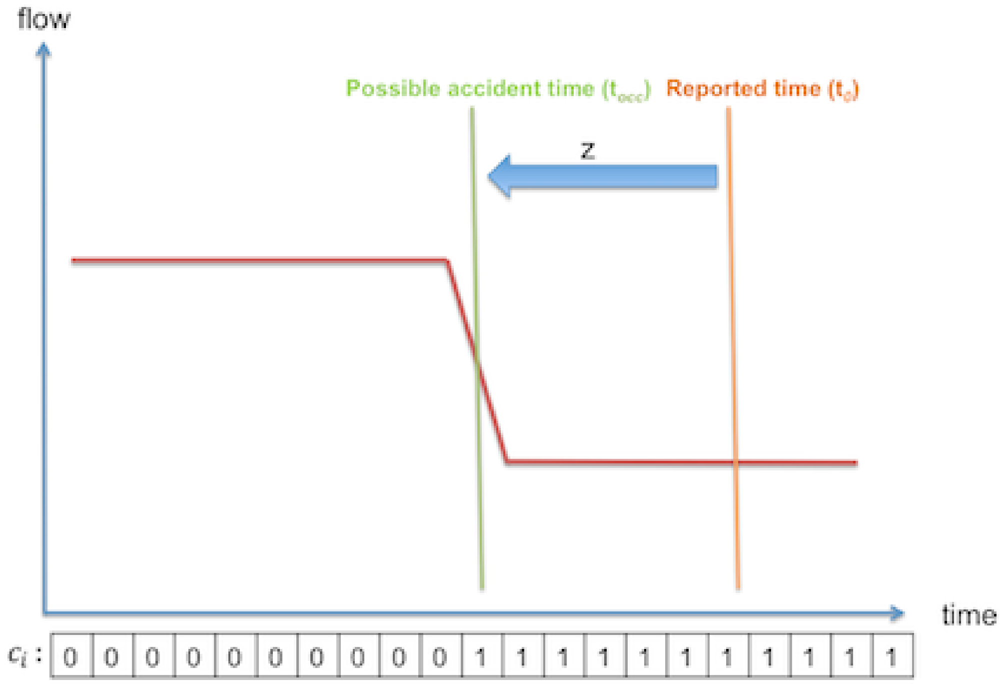

FIGURE 1 The role of z in determining the overlay and the occurrence time.

for $n = r + l + 1$ (all points in the time window). Σ is the covariance matrix created with $\Sigma_{i,j}$ containing the auto covariance for lag $|i - j|$. $\hat{\mathbf{y}}$ corresponds to the prediction for $\mathbf{y}$ according to a time series model that considers the change point specified in z. In practice, we use the time series library from the Weka package (Hall et al. (*13*)). It formulates the time series prediction task as a (potentially non-linear) regression problem and transforms each time series point as a single input vector that carries temporal relations (e.g. lags) as well as other features, called “overlays". In this way, we can run any Weka regression method over our data, such as support vector machines (SVM), neural networks and so on. The overlays correspond to the indicator functions in time series literature and are crucial to our context.

The change-point defined by z is turned into the following overlay:

$$c_i = \begin{cases} 1 & t_i + z > t_0 \\ 0 & otherwise \end{cases}$$

where c_i is the value of the overlay at time t_i. The visual intuition for this effect is in Figure 1.

In this way, the regression algorithm will be able to distinguish between the two regimes while still keeping the whole time series in a single model. With adaptable regression models, this should both allow for superficial, mean-shift changes, as well as to entire regime changing situations, maximizing the global coherence, as opposed to explicitly breaking the series into two separate parts and re-estimating the models separately.

Regarding the bayesian prior model for z, its role is to introduce local knowledge about the delays. Although the precise occurrence time is generally impossible to capture, for many accidents traffic operators are able to provide rough estimates. In our case, traffic operators shared the intuition that reporting delays should average around 5 minutes or less. This obviously provides only the mean for the prior, so its form is left to the modeler. Care must be taken to avoid negative delays (i.e. an accident cannot occur after reported time), therefore symmetric forms such as the normal distribution are not an option. Moreover, it is reasonable to expect high mass in lower delays (e.g. 5 minutes and below) and a long tail for the high delays. A log-normal distribution can provide such a behavior.

After determining the most likely occurrence time in the signal, there is one last step to consider. This analysis only gave us the moment in time where the incident disturbance reached the sensor. We also need to consider the time it takes such disturbance to go from the actual incident location to that sensor's location. We can consider two situations, depending on the sensor position relative to the accident, namely whether it is upstream or downstream.

In the upstream case, the disturbance propagation should be dependent on the queue formation rate. For such case, one needs to model the queue shockwave (also using information on speed and capacity) to determine the time it takes for the disturbance to reach the sensor. A few solutions exist for this model (e.g. cell transmission model (Daganzo (*14*)), shockwave speed model (Kuhne and Michalopoulos (*15*))).

When the sensor is downstream to incident, the time difference of the disturbance should be much smaller. Mak (*7*) showed that, for the case of Singapore expressways, the time it takes for the disturbance to reach the sensor is between 0.4 and 1.39 minutes. If we're working with larger time intervals (e.g. 5 minutes), this time difference becomes negligible. Our own observations on the same dataset reinforce this conclusion.

As mentioned before, this paper deals with the downstream case, leaving a solution that simultaneously considers upstream and downstream sensors to further work.

Summarizing, our method approximates the incident occurrence time by maximizing the likelihood of a change-point model. This model is parameterized by the latent variable, z, that represents the delay and directly defines the change-point. The other parameters are determined by a general training procedure. This method does not demand any particular functional form for the change-point time series model and uses a bayesian prior for the variable z that builds on operational field knowledge.

As in earlier works, since we don't have a set of ground-truth observations, we will manually define a baseline for comparison. Of course, such baseline will be affected by our own perception biases. For example, one intuition is that a "very large drop" in flow should indicate the incident occurrence. While this may be a reasonable assumption, this drop may itself follow a trend (e.g. peak hour effects) or be a second consequence of the incident (e.g. police arrival). To allow a clearer analysis, we also assign a confidence value (between 1-low and 5-high) to each case. We have high confidence when the signal is stable enough and the disturbance is clear enough (as in Figure 1) or supported by the text report. We will evaluate our model with the root mean squared error (RMSE) as well as the Mean Absolute Error (MAE).

EXPERIMENTS

Data

The dataset comprises 5 months of traffic flow data from 268 sensors located in 3 Singapore expressways. During this period a total of 1086 traffic incidents were recorded. The distance between incidents and downstream sensors is distributed according to Figure 2.

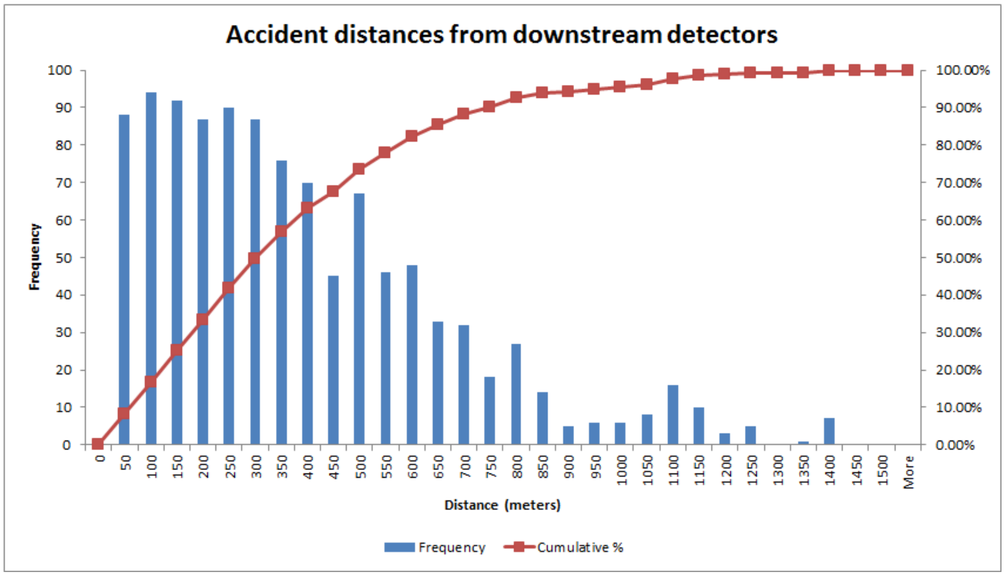

FIGURE 2 Distance between incidents and downstream sensors.

For each incident, we have the report start time as well as the traffic volumes for the downstream sensor, aggregated on 5 minute windows, the observed incident duration and a small text report. In Figure 3, we show two incident cases where we consider that the disturbance is clear (top) and unclear (bottom). We also show the results according to our algorithm ("t_{occ}").

For each incident, we obtained the downstream sensor data from 180 minutes before and 60 minutes after the reported time (i.e. $l = 180$ and $r = 60$ in equation 1). We standardized the signal having as reference the same time of day and type of day (weekend/weekday) throughout the entire dataset. The total number of cases is 1086. From this set, we manually inspected 401 cases for later comparison and validation. This manual inspection was essentially visual, very occasionally using the report text for further verification.

Experimental design

For each incident, we independently ran our model. After a trial period, where we tested with the complete regression portfolio from Weka (Hall et al. (*13*)), we decided to use a support vector machine algorithm. We also defined the number of lags to be 12 and kept the remaining parameters at default values after some exploratory experiments.

We defined the bayesian prior for z to be the lognormal with mean 1.87 and shape (σ^2) of 0.26.

Since our dataset is aggregated on 5 minute intervals, our values for z will also vary discretely in the same fashion. For each incident and each possible value of z within the range $[-l, r]$ (with z being a multiple of 5), we estimate a time series model as above described. We calculate the likelihood using equation 2 and multiply it with the prior probability. The highest value will indicate the maximum a posteriori, i.e. the best estimate for the incident occurrence time.

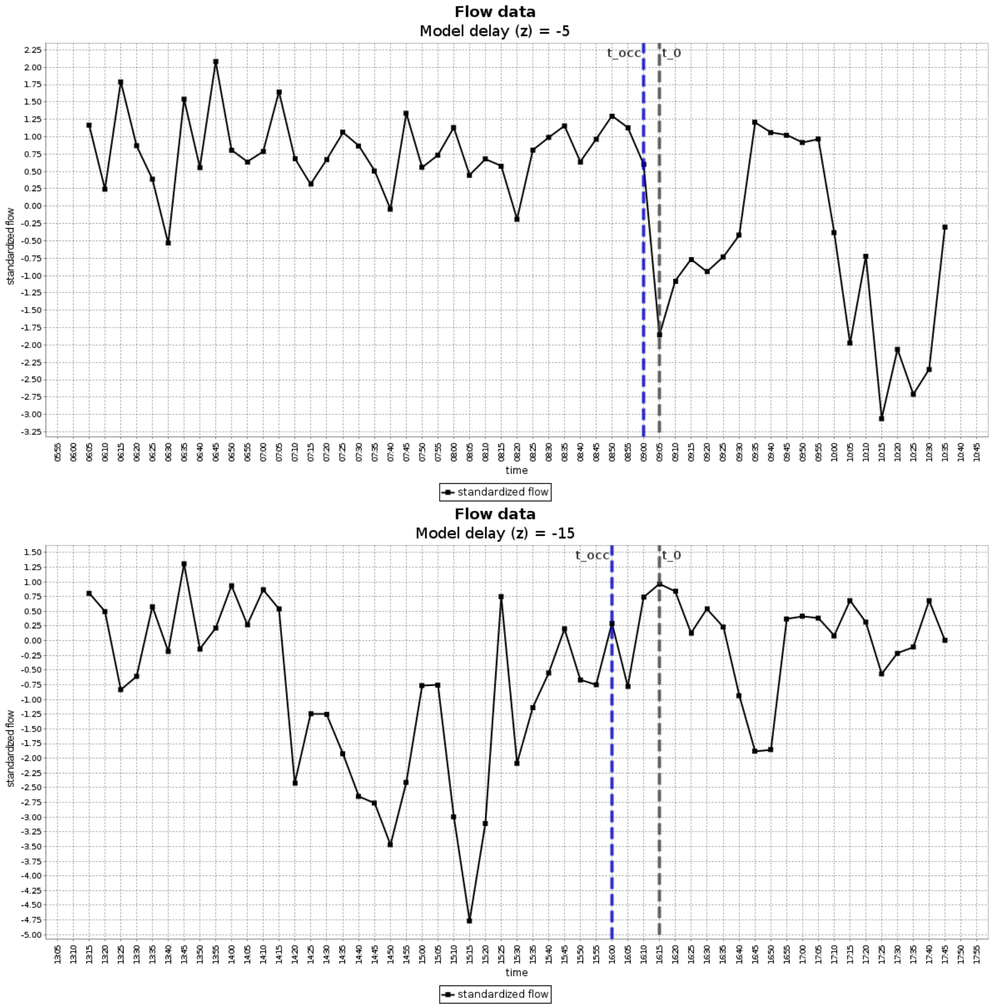

FIGURE 3 Two incident signals.

Results

The left plot in Figure 4 shows the distribution of the general results through the dataset. The values are negative with respect to reported time (i.e. -5 corresponds to "5 minutes before the reported time"). Expectably, we see a high frequency of 0 and 5 minute delays, but also that there is a relatively long tail up to 45 minutes. One can argue that such concentration in short delays is artificially induced by the prior. To understand its influence, we depict the results without the prior effect on the right.

It turns out that the bayesian prior does have a strong influence. Its role is essentially to bias the maximum a posteriori towards the 5 minutes delay other things being approximately equal. Notice that, for each incident, the dataset has a reasonably large number of points (50 points, from -180 to 60 minutes) and therefore the prior is only relevant when multiple MAP candidates exist with competitive probabilities. This behavior is expectable for a prior. Whether it is desirable or correct is subject to the context. In our case, we will compare these results with the manual baseline mentioned above.

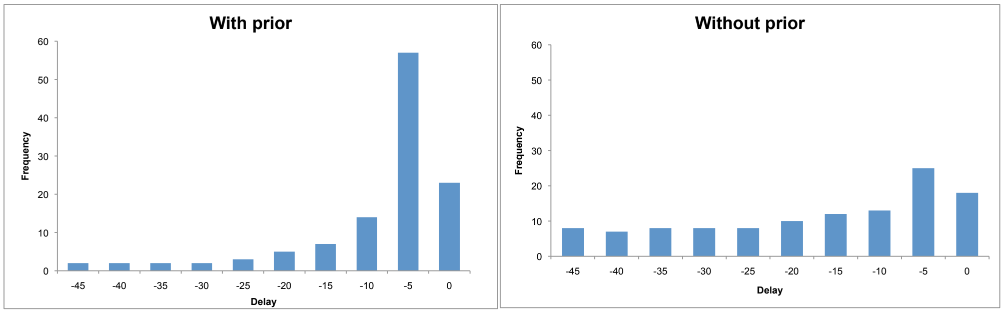

FIGURE 4 Distribution of results through the dataset.

Tables tables 1 and 2 show the MAE and RMSE results, respectively, in comparison with our manual baseline, with and without using the bayesian prior for z. We show the results for the cases with higher confidence (ranked either 4 or 5) and for all cases evaluated.

TABLE 1 Comparison with manual baseline (MAE)

Prior	No Prior	
5.7692	9.9573	High confidence
7.3852	12.615	all cases

TABLE 2 Comparison with manual baseline

Prior	No Prior	
10.096	15.832	High confidence
11.874	17.739	all cases

Regarding the bayesian prior, it has a non-negligible role in both MAE and RMSE. We also tested with different scale parameters and the chosen mean and scale yielded the best results. It is arguable that our own baseline is itself biased, but the intuition that the prior helps avoid over-fitting is relevant. Since we estimate an individual change-point model for each incident, without the prior, the only knowledge considered would be the flow time series for that specific window, which would make the model too sensitive to local context (e.g. sensor noise, secondary accidents). Of course, the parameters of the prior themselves should be realistic as much as possible.

On a less positive note, the MAE and RMSE errors in comparison with the baseline are considerably high, even when using the bayesian prior. There may be several general explanations for this fact. Upon visual inspection, there are incidents with more than one plausible occurrence time

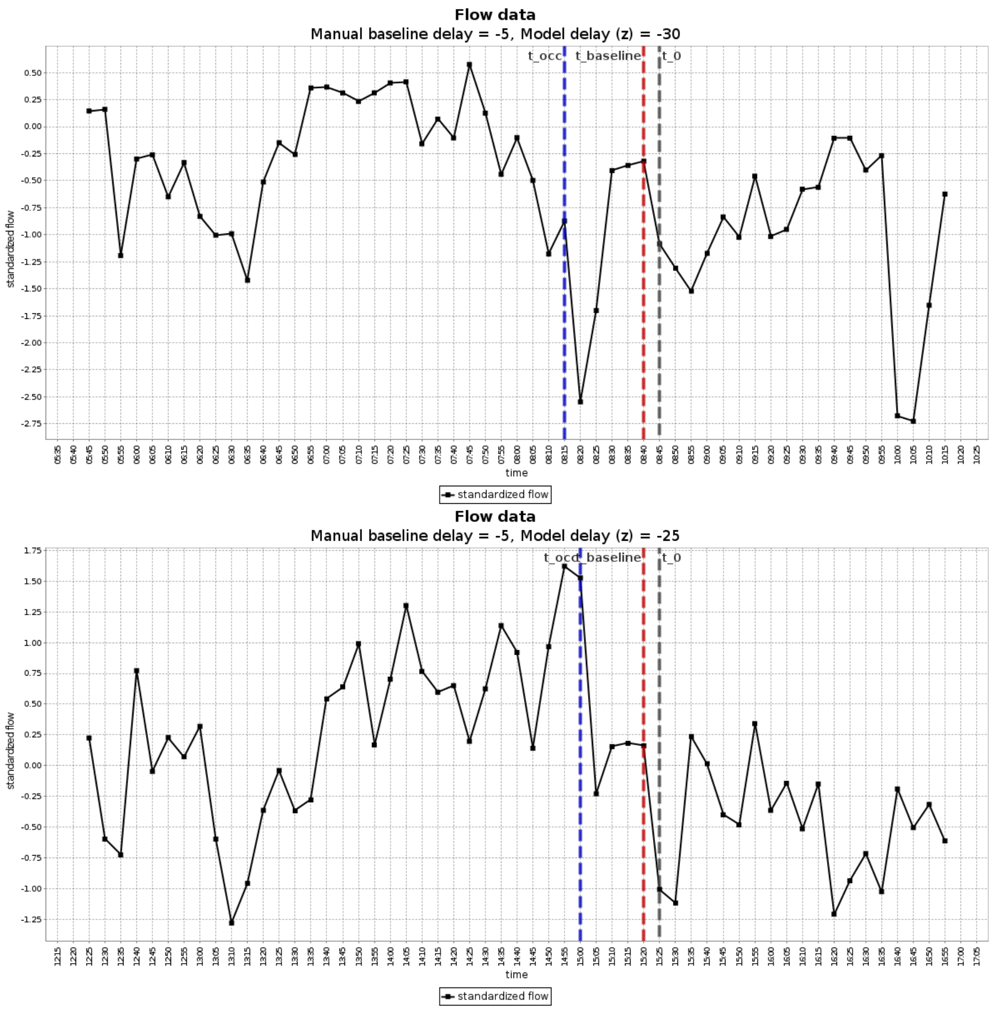

FIGURE 5 Both baseline and adjusted are plausible incident occurrence times.

and, in these cases, intuition provides contradictory answers: delay time shouldn't be unreasonably high (i.e. it should be as close as possible to report time, t_0); the incident may generate several flow disturbances (i.e. occurrence should be the earliest possible, followed by other episodes, maybe more intense). Figure 5 gives two examples.

Another explanation for the differences to the baseline is that our algorithm captures time series model changes rather than single signal drops, therefore it may "see" patterns that the human eye cannot. In Figure 6, we show an example where our algorithm apparently caught the beginning of a new time series pattern rather than the earlier big flow drop.

Another problem is ambiguity in the change-point determination due to the discrete nature of the signal. In some baseline cases, we chose the beginning of a large drop to correspond to the incident occurrence time, while the algorithm may also choose the end or middle of this drop (see Figure 7 for an example) depending on the maximum likelihood of the change point model.

Finally, an important limitation relates to having only one change point. The sensor signal may have more than two regime changes due to the incident. Indeed, if we want to include the

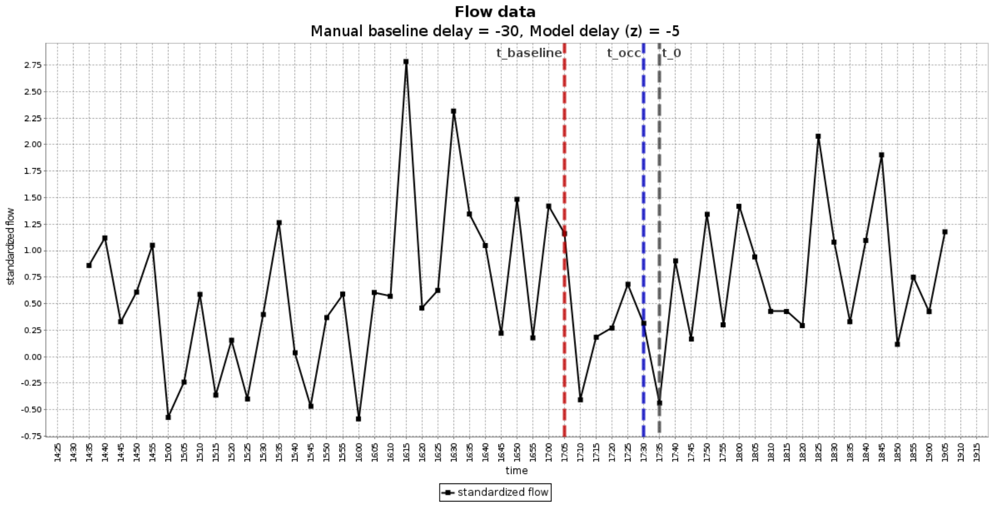

FIGURE 6 Chosen time is the beginning of a new time series.

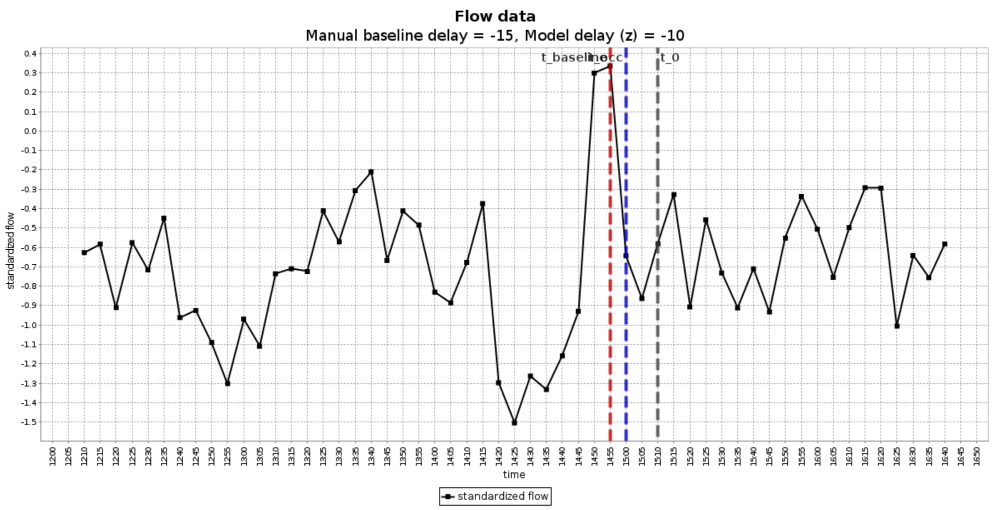

FIGURE 7 Ambiguity due to signal discretization.

recovery phase, we'd need at least two change points.

DISCUSSION

As with any other model designed to solve a concrete real-world problem, our methodological decisions were sensitive to the available dataset, which had 5 min aggregations of flow data and its own characteristics in terms of sensor quality and spatial distribution. With speed information for the same sensor locations, we could extend the model in a number of ways. Ideally we would integrate the speeds into the joint distribution of equation 1, thus Σ in equation 2 would also have to consider speed-flow signal correlations, for example using the fundamental traffic flow diagram. Alternatively we could use a quasi-likelihood solution, by assuming independence between speed and flow.

Another extension would be to include upstream sensor data, in a dual-sensor model. Following the same principles from this paper, we could obtain two independent approximations, but we should take advantage of their spatial/temporal relationships. Depending on distance to both sensors as well as upstream queue buildup, the disturbance should arrive at potentially distinct, and physically plausible, times to both locations. Furthermore, unless there is an intersection between the sensors, their time series models should be somehow correlated, particularly before the incident occurs. These considerations imply a joint distribution model with both signals, in equation 1, and again particular care with covariance matrix Σ to reflect the cross-correlations between the two signals.

An extension of this model that considers speed data as well as both upstream and downstream sensors will be presented in a subsequent article.

The empirical evaluation of the results shows that, in general, our model proposes plausible incident occurrence times. However, the lack of observability makes this evaluation essentially subjective. An alternative validation methodology is to use a microsimulation traffic model that is able to simulate incidents (e.g. MITSIMLab (Ben-Akiva et al. (*16*))). Having noise and spatial models for sensors we can understand the sensitivity of our proposal with respect to these aspects.

From the point of view of the modeler, the role of the bayesian prior needs particular attention. While one should not “tweak" it to influence the results towards some subjective goal, it may be a mistake to ignore field knowledge and intuition. The results showed that the knowledge introduced by the field operators was relevant to the quality of the model, as compared to the baseline. More objective solutions could have been explored, such as conditioning the prior on some general heuristic, such as the *largest drop* or the 20% rule suggested by Dia (*8*).

Finally, this model intends ultimately to be applied on a real-time basis. This implies that the data available will be limited, particularly the right bound, r, of the time window will be increasing sequentially in time. An obvious next step is thus to simulate such a sequential model and observe how its predictions evolve accordingly.

CONCLUSION

We proposed a methodology that fully automates the approximation of incident occurrence time by analysis of downstream sensor flow data. We apply a latent variable framework that uses a change-point time series model as the likelihood function. The actual incident occurrence time has been generally neglected in literature, either being reduced to a calibration parameter during training of incident detection models; or in post-hoc incident analysis works. In both cases, the typical approach is to manually analyze the signal (e.g. (Mak (*7*))) or apply simple heuristics such as disturbance thresholds in traffic parameters (e.g. (Dia (*8*))).

We built the model over a few principled statements: unless in exceptional cases, the reported time should have a delay with respect to actual incident occurrence time; unless under very low flow/high capacity, incidents should affect the traffic flow signal during a period of time right after its occurrence; unless there is an intersection or the distance to the sensor is too high, such disturbance should reach the downstream signal shortly after the incident.

We tested our model on a dataset with flows from Singapore expressways for a period of 5 months, together with an incident records database. In order to evaluate our model, we manually built a baseline on a sub-set of this dataset. Results show that our model generally proposes plausible incident occurrence time approximations, even when disagreeing with our baseline.

One of the biggest challenges of such a framework is effectively the lack of ground-truth and consequently an objective validation. A next step is to use traffic microsimulator model to generate incidents as well as sensor data, and study the quality of our model with respect to sensor data quality and distance to the incidents.

We will also extend the model to consider other types of data (e.g. speed information) as well as both the downstream and upstream sensors in a single formulation.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge Land Transport Authority of Singapore for providing the dataset and helping with local expertise. This research was supported by the National Research Foundation Singapore through the Singapore MIT Alliance for Research and Technology's FM IRG research programme.

REFERENCES

[1] Weil, R., J. Wootton, and A. GarcŠa-Ortiz, Traffic incident detection: Sensors and algorithms. *Mathematical and Computer Modelling*, Vol. 27, No. 9Ð11, 1998, pp. 257 – 291.

[2] Karim, A. and H. Adeli, Incident Detection Algorithm using Wavelet Energy Representation of Traffic Patterns. *J. Transp. Eng.*, Vol. 128, No. 3, 2002, p. 232Ð242.

[3] Tang, S. and H. Gao, Traffic-incident detection-algorithm based on nonparametric regression. *Intelligent Transportation Systems, IEEE Transactions on*, Vol. 6, No. 1, 2005, pp. 38–42.

[4] Antoniou, C., H. N. Koutsopoulos, and G. Yannis, Dynamic Data-Driven Local Traffic State Estimation and Prediction. *Transport Research - Part C*, forthcoming.

[5] Pereira, F., F. Rodrigues, and M. Ben-Akiva, Text analysis in incident duration prediction. *submitted, in review*, 2013.

[6] Ben-Akiva, M., H. N. Koutsopoulos, C. Antoniou, and R. Balakrishna, Traffic Simulation with DynaMIT. In *Fundamentals of Traffic Simulation* (J. Barceló, ed.), Springer, New York, NY, USA, 2010, pp. 363–398.

[7] Mak, C. L., *New dual-variable algorithms for detecting lane-blocking incidents on expressways*. Ph.D. thesis, Nanyang Technological University, Singapore, 2002.

[8] Dia, H., *Artificial Neural Network Models for Automated Freeway Incident Detection*. Ph.D. thesis, Monash University, Australia, 1996.

[9] Mak, C. and H. Fan, Heavy SSow-based incident detection algorithm using information from two adjacent detector stations. *Journal of Intelligent Transportation Systems*, Vol. 10, No. 1, 2006, pp. 23–31.

[10] Lee, D.-H., S.-T. Jeng, and P. Chandrasekar, Applying data mining techniques for traffic incident analysis. *Journal of The Institution of Engineers, Singapore*, Vol. 44, No. 2, 2004.

[11] Jeong, Y., M. Castro-Neto, M. Jeong, and L. Han, A Wavelet-Based Freeway Incident Detection Algorithm with Adapting Threshold Parameters. *Transportation Research Part C Ð Emerging Technologies*, Vol. 19C, No. 1, 2011, pp. 1–19.

[12] Hawkins, D. M., Fitting multiple change-point models to data. *Computational Statistics & Data Analysis*, Vol. 37, No. 3, 2001, pp. 323 – 341.

[13] Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, Vol. 11, No. 1, 2009.

[14] Daganzo, C. F., The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological*, Vol. 28, No. 4, 1994, pp. 269–287.

[15] Kuhne, R. and P. Michalopoulos, Continuum flow models. In *Traffic flow theory: A state of the art report - revised monograph on traffic flow theory* (N. H. Gartner, C. Messer, and A. K. Rathi, eds.), Oak Ridge National Laboratory, 1997.

[16] Ben-Akiva, M., H. Koutsopoulos, T. Toledo, Q. Yang, C. Choudhury, C. Antoniou, and R. Balakrishna, Traffic Simulation with MITSIMLab. In *Fundamentals of Traffic Simulation* (J. BarcelŮ, ed.), Springer New York, Vol. 145 of *International Series in Operations Research & Management Science*, 2010, pp. 233–268.