AN-TP 0388

Traffic Flow Modelling with Point Processes

K. W. Lim*, **W. Wang**, **H. Nguyen**, **Y. Lee**, **C. Cai**, **F. Chen**

Data61/CSIRO, Australia

Abstract

The main objective of this article is to predict the traffic flow for the purpose of urban traffic management. Long term incidents, such as floods, road maintenance or major traffic incidents can cause significant disruptions on the traffic flows in large metropolitan areas such as Sydney. There is a need to comprehend how traffic will be impacted in a city during such an event. One main reason for this is to manage the flow of this traffic so that it does not obstruct with emergency protocols. Such modelling requires the capturing of fast-evolving and ever changing conditions, taking into account road closures, *Twitter* or *Facebook* information as well as other circumstances. One important characteristic in traffic flow modelling is the ability to describe traffic *burstiness*, i.e. vehicles tend to cluster together, thus creating traffic congestion. This paper provides some promising results in regards to traffic flow modelling using point processes framework. We are able to simulate traffic flow that mimics real life conditions, and also perform efficient inference thereof.

Keywords:

Traffic flow modelling, machine learning, point processes with stochastic intensity

1. Introduction

The use of social media information to provide insight into traffic management is not new (see Zhang et al., 2015). In this paper, we propose to use readily available news from social media such as Twitter and Facebook as defining features to model traffic flow in major cities. We propose a variation of the shot noise Cox processes (Cox and Isham, 1980) to model traffic flow. This parameterisation allows us to incorporate the availability of these social media data to accurately capture the correlations of incident times and traffic burstiness, which is inherent in traffic.

Since the number of vehicles for a given road is a counting process, we feel that the framework of modelling traffic flow should be discrete in nature. This is in contrast to the recent work by Tahmasbi and Hashemi (2014), where the authors proposed a continuous variable to model the rate of flow of traffic through a mean reverting dynamics. We also note that although our work is motivated by the flow modelling of Menon et al. (2015), we employ a different approach to tackling the problem, that is, using point processes.

2. Problem Statement

The aim of this work is to model the traffic flow for urban traffic management. Major incidents such as floods, road maintenance or major traffic can cause significant disruptions on the traffic flow. One main reason for this is to manage the flow of this traffic so that it does not obstruct with emergency protocols. One of the aims is to create rich simulated traffic counts. One of the important characteristics in traffic flow is to incorporate traffic burstiness, see for example Naja (2012). Vehicles tend to cluster together, creating traffic congestion. The aim of this work is to construct a model to simulate traffic flow that mimics real life conditions and subsequently perform basic descriptive statistics.

3. Model Description

This section introduces some pieces of counting process theory needed in what follows. The *general nonhomogeneous Poisson process* $\hat{N} = \{\hat{N}(t) : t \geq 0\}$ with intensity function $\lambda(t)$ has the following properties:

(a) The process starts at 0 with $\hat{N} = 0$.

(b) The process has independent increments, i.e. for any t_i, $i = 0, ..., n$, the increments $\hat{N}_{t_i} - \hat{N}_{t_{i-1}}$ are mutually independent.

(c) There exists a non-decreasing right continuous function $m : [0, \infty) \to [0, \infty)$ with $m(0) = 0$ such that the increments $\hat{N}_t - \hat{N}_s$ for $0 < s < t < \infty$ have a Poisson distribution with mean value function $m(t) - m(s)$.

The relationship between m and λ is given by

$$m(t) = \int_0^t \lambda(s)\, ds \tag{1}$$

In the following, our proposed model has the intensity function $\lambda(t)$ that incorporates external excitations. In particular, we use the set up with differing degrees of generality, adapted from Cox and Isham (1980). To be precise, we give a full description of our model.

The Cox Model. We are interested in a counting process $N(t)$ whose behaviour is affected by past events. These past events contains *externally* excitation elements that trigger the observations in N. Our point process $N(t)$ has a non-negative $\mathcal{F}_t-$ stochastic intensity function $\lambda(t)$ of the form:

$$\lambda(t) = \mathcal{B}_0(t) + \sum_{j:\, t > S_j} \mathcal{C}(X_j, t - s_j). \tag{2}$$

External events can occur at times s_j and stimulate with a portion of $\mathcal{C}(X_j, t - s_j)$: this is known as the externally-excited part. The quantities X_j are positive random elements describing the

amplitudes by which λ increases during event times. The quantity $\mathcal{B}_0 : \mathbb{R}_+ \mapsto \mathbb{R}_+$ denotes the *deterministic* base intensity. We write $N_t := N(t)$ and $\lambda_t := \lambda(t)$ to ease notation and $\{\mathcal{F}_t\}$ being the history of the process and contains the list of times of events up to and including t, i.e. $\{s_1, s_2, ..., s_{N_t}\}$. We define formally a point process N on $\mathbb{R}^+$ with the non-negative $\mathcal{F}_t$ conditional random intensity

$$\lambda_t = a + \sum_{j=1}^{J_t} X_j \, e^{-\delta(t-s_j)}, \tag{3}$$

for $t \geq 0$, where we have the following features:

- **Deterministic background.** The parameter $a \geq 0$ is the constant mean-reverting level. This means that $\mathcal{B}_0(t) = a$, which is a simplifying assumption in this paper.
- **External-excitations.** The random variables X_j are levels of excitation from an external factor. They form a sequence of independent and identically distributed positive elements with distribution function $H(c), c > 0$. The sequence s_i denotes the times at which external events happen and it follows a homogenous Poisson process J_t of constant rate $\mu > 0$. Note that $\mathcal{C}(X_j, t - s_j) := X_j \, e^{-\delta(t-s_j)}$. The parameter $\delta > 0$ is the constant rate of exponential decay that dilutes the external excitation through time.

The intuition here is that a is the background intensity that generates observations in N irrespective of the external events, which is assumed to be a constant. In addition to the background rate, each event in N is affected by the external events for which their event times are captured by s_j. These event times are assumed to be following a homogenous Poisson process J_t with rate μ. An illustrative example of these events include news and Twitter posts. The dynamics for which the external events affect is modelled by the function $\mathcal{C}(X_j, t - s_j)$. One commonly used function is the exponentially-decaying function first introduced by Hawkes (1971), which has seen numerous applications such as finance (Dassios and Zhao, 2013). The strength of which the external excitation decays is determined by the parameter δ.

4. Model Learning

We employ a fully Bayesian inference framework to learn the model parameters. In particular, we introduce a hybrid Markov chain Monte Carlo (MCMC) algorithm combining the features of the Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970) and the Gibbs sampler (Geman and Geman, 1984). In particular, the background intensity a and the decay rate d are learned with MH algorithms, while the external factor intensity μ is inferred by a Gibbs sampler.

4.1. Hybrid MCMC Learning Algorithm

We present a hybrid of MCMC algorithms that updates the parameters one at a time, either by direct draws using Gibbs sampling or through the Metropolis Hastings (MH) algorithm. A

hybrid algorithm combines the features of the Gibbs sampler and the MH algorithm (Robert and Casella, 2005), thereby providing significant flexibility in designing the inference thereof for different parameters within our model.

To see how it works, consider a two-dimensional setting as an illustration. Let θ_Q and θ_W be parameters of interest. Assume that the posterior $\mathbf{P}(\theta_Q \,|\, \theta_W)$ is of a known distribution, we can perform inference directly utilizing the Gibbs sampler. On the other hand, suppose $\mathbf{P}(\theta_Q \,|\, \theta_W)$ can only be evaluated but not directly sampled; then, we resort to the use of an MH algorithm to update θ_Q given θ_W. For the MH step, the candidate θ'_Q is drawn from $\mathbf{R}(\theta'_Q \,|\, \theta_W^{(k)}, \theta_W^{(k)})$, which indicates that the current step can depend on the past draw of θ_Q. The Metropolis step samples from $\mathbf{R}(\theta'_Q \,|\, \theta_Q^{(k)}, \theta_W^{(k)})$ which implies that we draw $\theta_Q^{(k+1)} \sim \mathbf{R}(\theta'_Q \,|\, \theta_Q^{(k)}, \theta_W^{(k)})$ and that the criteria to accept or reject the proposal candidate is based on the acceptance probability, denoted by $A(\theta_Q^{k+1})$:

$$A(\theta_Q^{k+1}) = \min\left(1, \frac{\mathbf{P}(\theta'_Q \,|\, \theta_W)\, \mathbf{R}(\theta_Q^{(k)} \,|\, \theta'_Q, \theta_W^{(k)})}{\mathbf{P}(\theta_Q^{(k)} \,|\, \theta_W)\, \mathbf{R}(\theta'_Q \,|\, \theta_Q^{(k)}, \theta_W^{(k)})}\right). \tag{4}$$

The hybrid algorithm in this setting is as follows. Given $\left(\theta_Q^{(0)}, \theta_W^{(0)}\right)$, for $k = 0, 1, ..., K$,

1. Sample $\theta_Q^{(k+1)} \sim \mathbf{R}\left(\theta'_Q \,|\, \theta_Q^{(k)}, \theta_W^{(k)}\right)$ and *accept* or *reject* $\theta_Q^{(k+1)}$ based on equation (4).
2. Sample $\theta_W^{(k+1)} \sim \mathbf{P}\left(\theta_W \,|\, \theta_Q^{(k+1)}\right)$ with Gibbs sampling.

4.2. Likelihood

Likelihood, by contrast, attaches to our parameter estimates and to our framework. Take for example, given that we have observed 9 heads in 10 flips of a coin, the likelihood that the probability of flipping a head is 50% (i.e., that *prob* = 0.5) is very low. The likelihood that *prob* = 0.9 is greater by a factor of almost 40. The likelihood function tells us the relative likelihoods of the different possible values for *prob*. In the case of point process, with Proposition 7.3.III in Daley and Vere-Jones (2003), the *likelihood function* for the events is given by

$$\mathbf{P}(\mathbf{t}, \mathbf{s} \,|\, a, \mu, \delta, \mathbf{X}) = \prod_{i=1}^{N_T} \lambda(t_i)\, e^{-m(T)} \prod_{j=1}^{J_T} \mu\, e^{-\mu T} \tag{5}$$

$$= \prod_{i=1}^{N_T} \left(a + \sum_{j:\, s_j < t_i} X_j\, e^{-\delta(t_i - s_j)}\right) e^{-m(T)} \mu^{J_T}\, e^{-\mu T} \tag{6}$$

where the boldface symbols $\mathbf{t}$, $\mathbf{s}$ and $\mathbf{X}$ represent a collection of similar variables. Here, T is represents the maturity time, of which we stop observing the point process N. The function m (also known as the compensator) is defined as

$$m(t) = \int_0^t \lambda_s\, ds \tag{7}$$

4.3. Priors

The likelihood function is, however, only one of two major blocks needed for a Bayesian calculation. The other is known as the *prior distribution* or just *prior* which is necessary for estimating parameters and for drawing probability conclusions. A prior distribution could and should take account of what one already knows. However, when one knows very little, one can use the Jeffreys priors, which is a non-informative belief for a parameter space and it is proportional to the square root of the determinant of the Fisher information. This class of priors are some of the most interesting and useful prior distributions, and they are derived from the mathematical implications of knowing usually nothing about the parameters one wants to infer other than their possible ranges, for example, whether if they are positive, negative, or both on the real space $\mathbb{R}$.

In our model, we assign the following prior. We first assume the priors of a, μ and δ to be exponential with rate 1:

$$a \sim \text{Exp}(1) \qquad \mu \sim \text{Exp}(1) \qquad \delta \sim \text{Exp}(1) \tag{8}$$

The levels of excitation, on the other hand, follows a positive random distribution. For instance, we could assume that they are identically and independently characterised by a Gamma distribution with parameters α and β. In this case then we will write

$$X_j \sim \text{Gamma}(\alpha, \beta) \qquad \mathbf{P}(X_j = x \,|\, \alpha, \beta) \propto x^{\alpha-1}\, e^{-\beta x} \tag{9}$$

and since α and β are free parameters, we can assign *hyperpriors* to them, say, exponential priors as follows:

$$\alpha \sim \text{Exp}(1) \qquad \beta \sim \text{Exp}(1) \tag{10}$$

In this case, the joint prior likelihood is formulated as

$$\mathbf{P}(a, \mu, \delta, \alpha, \beta) = \mathbf{P}(a)\, \mathbf{P}(\mu)\, \mathbf{P}(\delta)\, \mathbf{P}(\alpha)\, \mathbf{P}(\beta) \tag{11}$$

$$\propto e^{-a} \cdot e^{-\mu} \cdot e^{-\delta} \cdot e^{-\alpha} \cdot e^{-\beta} \tag{12}$$

4.4. Posterior

We now need to calculate the posterior distribution in which this quantity summarizes the current state of knowledge about all the uncertain quantities in a Bayesian analysis setup. Analytically, the posterior density is the product of the likelihood and prior density. This is the central quantity to be computed. Once the posterior has been computed, one can evaluate point and interval estimates of parameters, prediction inference for future data, and probabilistic evaluation of our current point process framework. With the above, the posterior (or joint posterior) is

given by

$$\mathbf{P}(a, \mu, \delta, \alpha, \beta, \mathbf{X} \,|\, \mathbf{t}, \mathbf{s}) \propto \mathbf{P}(\mathbf{t}, \mathbf{s} \,|\, a, \delta, \mathbf{X})\, \mathbf{P}(a, \mu, \delta, \alpha, \beta) \tag{13}$$

which is then used in the inference algorithm.

5. Data Collection

Before moving to discussing our experimental results, we briefly discuss the acquisition of the data used in this paper. In particular, we perform the proposed learning algorithm on both synthetic data and real traffic flow data observed in Sydney. Running the experiments on synthetic data serves to validate the correctness of our proposed inference algorithm and provides some insight to the underlying modelling problem. While experiments on the real world traffic data enable us to interpret the model and connect to it.

5.1. Synthetic Data

We simulate the traffic flow for a fictitious city, assuming that its traffic flow follows the model described in Section 3. The model parameters, chosen for ease of illustration, are: background intensity, $a = 0.5$; external factor intensity, $\mu = 1.0$; decay rate, $\delta = 1.0$; and the level of excitation is fixed to a constant $X_j = k = 5.0$. We set the maturity time to $T = 24$ hours for simplicity. These parameters are called the ground truth parameters in the following sections. The simulation algorithm, following the principle of Dassios and Zhao (2013) but adapted to our case, is exact in the sense that no approximation is made during the data generation.

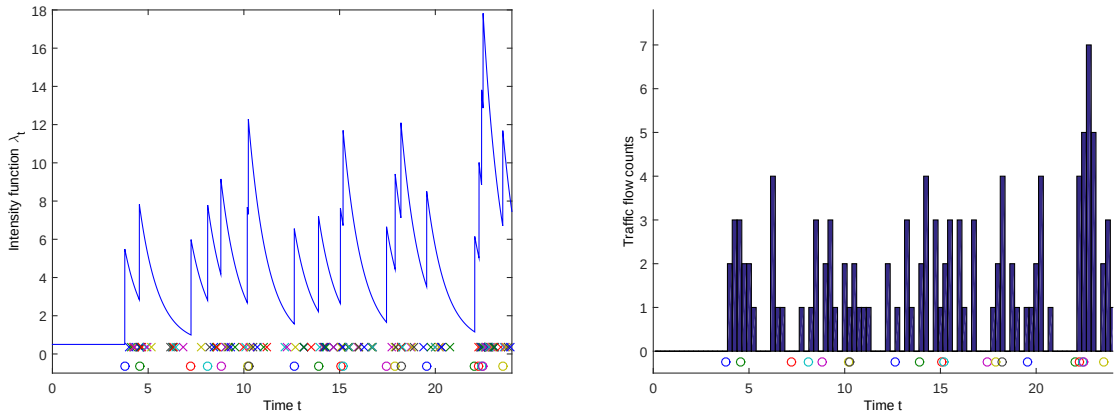

Figure 1: (a) Intensity function and (b) histogram of the simulated traffic flow

We present a realisation of the intensity process and the event times of the traffic flow in Figure 1(a). Here, the circles at the bottom represent the times for the external events, while the crosses are the (simulated) observed times for the traffic flows in the city. We can see that clusters of traffic flows are observed right after the occurrence of an external event, due to the

increase intensity of $X_j = k = 5$ at the time of the external events. Recall that the intensity is positively related to the rate of arrival of the traffic flow events. The corresponding vehicle counts are presented in Figure 1(b), which better illustrates the clustering property.

5.2. Sydney Traffic Data

We apply our method to model the traffic flow in Sydney. In particular, as a motivating example, we look at the vehicle flow in the Sydney on 30th March 2015. This dataset is obtained from SCATS,[1] which stands for *Sydney Coordinated Adaptive Traffic System*. We then process the dataset into a time series of vehicle flow events, for which we observe a total of 12,426 event times. We note that this time series is denominated in seconds.

For illustration, we construct a histogram of the event times in Figure 2, binned half-hourly. From this histogram, we can see that the number of events are not uniform through time, which is expected. High volume of vehicles are observed during peak hours, while the opposite is seen in the period after late night and before dawn.[2]

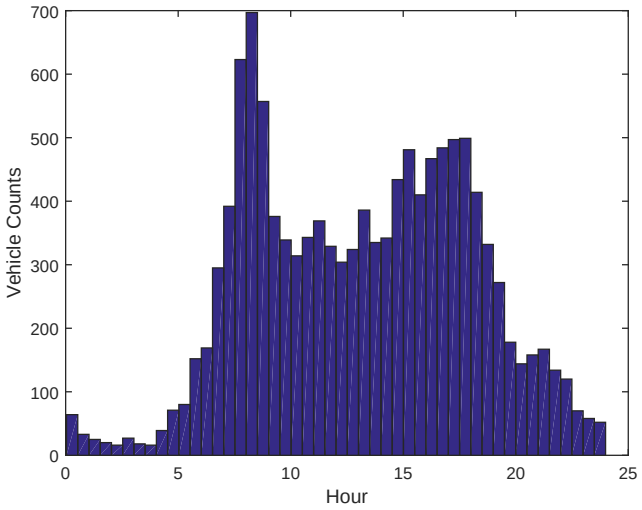

Figure 2: Histogram of the vehicle flow in Sydney in one day

On the other hand, as for the external events, we tap into the Twitter feed. We employ the *Traffic Watch*[3] framework (Nguyen et al., 2016) to query for tweets that are related to Sydney traffic condition. The workflow of the Traffic Watch is summarised by Figure 3, which details tweets crawling and processing. The architecture admits two phases: machine learning (ML) training process and real-time incident detection.

With large volume of tweets, our system extracts useful information by aggregating tweets into meaningful clusters, which provides a summary of current incident types as the tweets

[1] http://www.scats.com.au/
[2] Between 11pm and 5am.
[3] http://adait.io/project-twatch.html

emerge over time. To give an example, we present a snapshot of our system output in Figure 4. Here, crash events are detected from tweets and clustered based on the incident types, they are accompanied by an extract of the tweets for a detailed investigation or merely for perusal. For simplicity, however, we will use only the tweets' event time in the following modelling exercise.

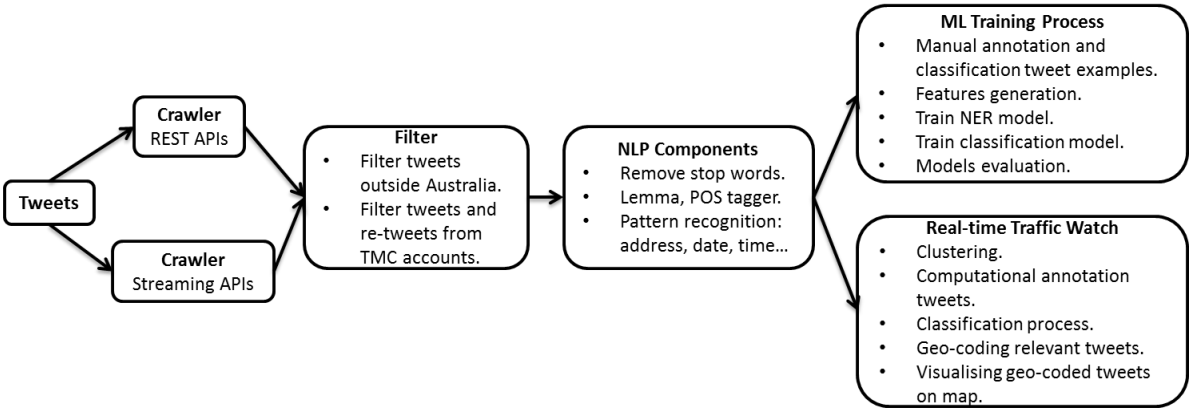

Figure 3: Traffic Watch workflow on Twitter data

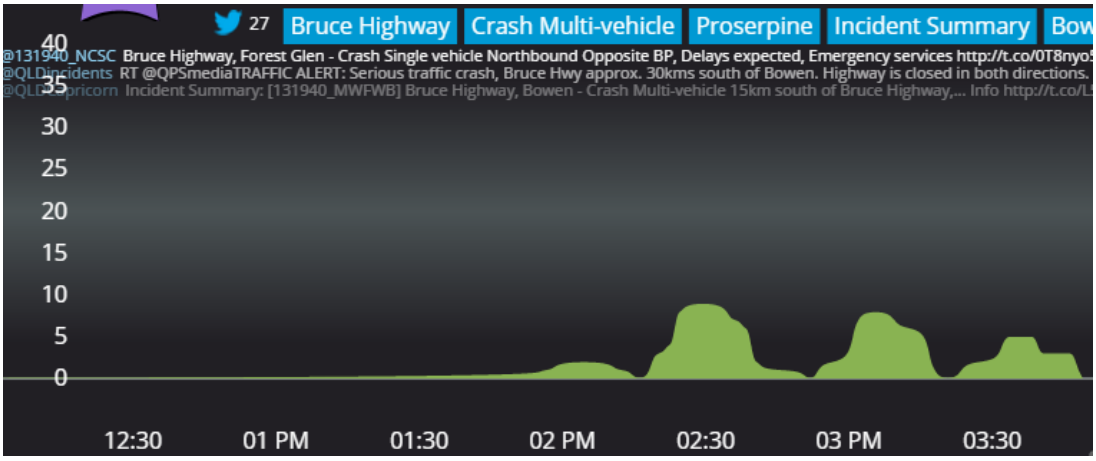

Figure 4: A snapshot of the Traffic Watch analysis

6. Experiment Results

Here, we discuss the experiments on both the synthetic data and the Sydney traffic flow data.

6.1. Assessing Modelling Accuracy on Synthetic Data

By running the proposed inference algorithm on the synthetic data described in Section 5.1, we are able to test the validity of the proposed MCMC method. Since the levels of excitation are fixed to $X_j = k = 5$, we assume it is known and learn the other parameters a, μ, and δ. These parameters are initialised to 0.5 and are updated with the inference algorithm described in Section 4. We perform the inference algorithm for 500 iterations and with a burn-in of 100.

The learned parameters are found to be

$$\hat{a} = 0.47 \qquad \hat{\mu} = 0.86 \qquad \hat{\delta} = 0.97 \tag{14}$$

which are very closed to the ground truth parameters of $a = 0.5$, $\mu = 1.0$, and $\delta = 1.0$. The calibrated parameters agree to within 0.15 units from the true parameters. This deviation can be explained by the randomness (of the random variables) in simulation.

To assess the convergence of the inference algorithm, we inspect the plot of model log-likelihood[4] against the number of iterations, which is presented in Figure 5. Since the initial parameters are quite close to the ground truth, we can see that the loglikelihood converges quickly and fluctuates around -60. This indicates that the inference mechanism performs well and as intended.

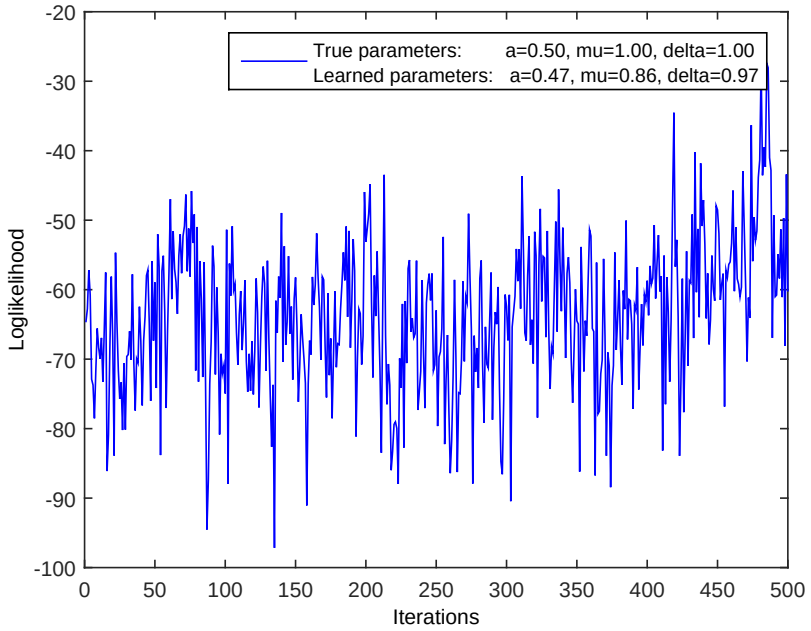

Figure 5: Loglikelihood trace during the learning of the model parameters

6.2. Modelling Sydney Traffic Flow

After testing our proposed inference method, we move on to the modelling of Sydney traffic data described in Section 5.2. In this dataset, since the levels of excitation X_j are not known, for simplicity, we will assume that they follow an exponential distribution with rate parameter β. This is equivalent to setting the shape parameter $\alpha = 1$ above:

$$X_j \sim \text{Exp}(\beta) \equiv \text{Gamma}(1, \beta) \tag{15}$$

[4]Note that the loglikelihood shown here is unnormalised and is subject to a constant shift in value.

Following the experiment setting from the synthetic experiment, we initialise all parameters to 0.5 and perform the inference algorithm for 500 iterations with burn-in of 100. The learned parameters are

$$\hat{a} = 0.1440 \qquad \hat{\mu} = 0.0005 \qquad \hat{\delta} = 3.0334 \qquad \hat{\beta} = 3.4104 \tag{16}$$

Here, we present an interpretation of the learned model. Without any external influence, we expect that, on average, there will be 0.144 vehicle per second passing through a given point in Sydney, this translates to about 9 vehicles per minute. Additionally, we find that each external event generates an expected number of

$$\frac{\mathbf{E}[X]}{\hat{\delta}} = \frac{1}{\hat{\beta} \cdot \hat{\delta}} = 0.09666 \tag{17}$$

vehicles in its life time.

Finally, we present a diagnostic test to assess the convergence of our learning algorithm on this dataset. Since the final learned parameters are close to their initialisation, we expect the model loglikelihood to improve during learning. We present the full loglikelihood plot vs. training iterations in Figure 6. As we can see, the loglikelihood quickly improves during training and has converged within the burn-in period. To inspect further, we plot only the loglikelihood after the burn-in period in Figure 7, which fluctuates around -3.688×10^4. We note that the loglikelihood here is much smaller than that of the synthetic experiment, which is due to using a much larger dataset.

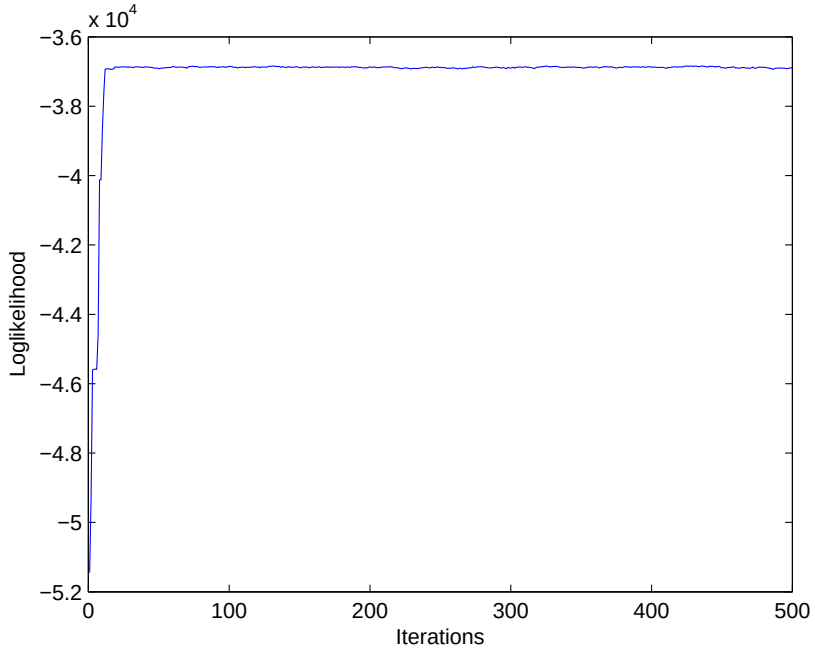

Figure 6: Full loglikelihood trace during the learning

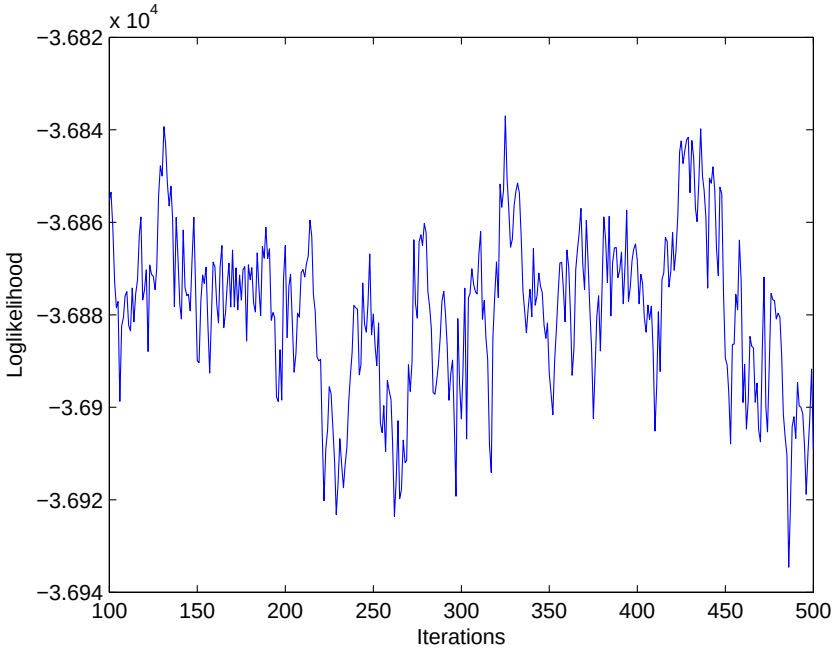

Figure 7: After burn-in loglikelihood trace during learning

7. Conclusion and Future Research

We have introduced a flexible point process framework in the modelling of traffic flow. This model can capture clustering, thereby mimicking traffic congestion. In addition, the rate at which vehicles passes through a particular stretch of road is influenced by the information of external events. We have also successfully developed a combined algorithm which uses both the Gibbs and MH algorithms that provides significant flexibility during inference. It is worthwhile investigating the potential usefulness of combining our model with the ongoing research in traffic demand modelling, which is a main course of current research initiated by Ben-Akiva and Lerman (1985).

As avenue for future investigation, we are interested in employing more kinds of data from Traffic Watch to improve traffic flow modelling. One particular interesting piece of information is the textual data accompanying tweets. Perhaps topic models from the machine learning community can be applied (for example, see Lim et al., 2016). Additionally, to improve modelling, it would be interesting to extend the proposed model to induce correlation and stochasticity in the levels of excitation, such as using the stochastic processes in Lee et al. (2016).

References

Ben-Akiva, M. E. and Lerman, S. R. (1985). *Discrete Choice Analysis: Theory and Application to Travel Demand*, volume 9. MIT press.

Cox, D. R. and Isham, V. (1980). *Point Processes*. Chapman & Hall.

Daley, D. J. and Vere-Jones, D. (2003). *An Introduction to the Theory of Point Processes*. Springer-Verlag New York, 2nd edition.

Dassios, A. and Zhao, H. (2013). Exact simulation of Hawkes process with exponentially decaying intensity. *Electronic Communications in Probability*, 18:1–13.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109.

Hawkes, A. G. (1971). Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, pages 89–90.

Lee, Y., Lim, K. W., and Ong, C. S. (2016). Hawkes processes with stochastic excitations. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, pages 79–88. Microtome Publishing.

Lim, K. W., Buntine, W., Chen, C., and Du, L. (2016). Nonparametric Bayesian topic modelling with the hierarchical PitmanYor processes. *International Journal of Approximate Reasoning*, 78:172–191.

Menon, A. K., Cai, C., Wang, W., Wen, T., and Chen, F. (2015). Fine-grained od estimation with automated zoning and sparsity regularisation. *Transportation Research Part B: Methodological*, 80:150 – 172.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092.

Naja, R. (2012). Highway preventive congestion control through input regulation in vehicular networks. In *Advanced Infocomm Technology - 5th IEEE International Conference, ICAIT*, pages 340–356.

Nguyen, H., Liu, W., Rivera, P., and Chen, F. (2016). *TrafficWatch: Real-Time Traffic Incident Detection and Monitoring Using Social Media*, pages 540–551. Springer International Publishing.

Robert, C. P. and Casella, G. (2005). *Monte Carlo Statistical Methods*. Springer-Verlag New York.

Tahmasbi, R. and Hashemi, S. M. (2014). Modeling and forecasting the urban volume using stochastic differential equations. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):250–259.

Zhang, R., Shu, Y., Yang, Z., Cheng, P., and Chen, J. (2015). Hybrid traffic speed modeling and prediction using real-world data. In *IEEE International Congress on Big Data*, pages 230–237. IEEE.