MŰEGYETEM 1782

Pro Progressio/HSNLab Research Report

Gergely Öllös, Ph.D. student
Research supervisor: Dr. Rolland Vida, Ph.D.

Traffic Model for Event Forecasting Protocol Validation

Supported by the Pro-Progressio Fundation.

July 22, 2013

Contents

Abstract

In this report we introduce a novel traffic model for event forecasting research that we validated against real world traffic. For the collection of reference events we used a small WSN deployed nearby various types of roads and crossroads, where the sensors transformed the sampled waveforms generated by vehicles into event descriptions that included the sensor's location information, a time-stamp, and the intensity of the signal as a degree of confidence in its detection.

We analyzed the reference sequences of events and we proposed a probabilistic, non i.i.d. (independent and identically distributed) aperiodic traffic model, that can be used to demonstrate the effectiveness of various event forecasting protocols.

When we compared the statistical properties of the generated events to the reference, we found that the proposed model can well approximate the various descriptive statistics as well as correlation patterns of real world measurements.

Introduction

The traffic model proposed here is used to validate the instance based anytime rare event forecasting solution that we proposed in an earlier work[1]. In spite of the fact that the model was proposed for a specific task or protocol, the formulation is general, so it can be used to validate other forecasting models even if they are based on statistical[2], fuzzy logic[3] or other forecasting methods.

We define the binary event $e = (ID, \mu, t)$ as something that happened with the probability μ, at a given sensor identified by ID, at time t. We assume that a localized or distributed signal processing algorithm can detect an event (for instance a truck passing a sensor node can be detected based on the audio spectrum signature that it produces) and provide a detection confidence $\mu \in [0..1]$ for each predefined event $e \in E$, where E is the set of all possible events. If the μ detection confidence is significant, the node broadcasts a message about the event where the $\mu \in [0..1]$ parameter is proportional to the norm between the reference and the measurement.

As the cars move through the network, they generate different event sequences. A moving car crossing the monitored field (or part of the road) generates a *Global Sequence* (GS) which is a sequence of events in the network. There can be multiple GSs in the network, which can overlap with each other or even with themselves (multiple cars crossing the field randomly on a similar path). Each node is aware of only a subset of the global sequence (GS), either due to its own observations, or by overhearing nodes in its vicinity. These subsets are called *Local Sequences* (LS).

For illustration, let us suppose that a sensor field partially covers an uphill road, as in Fig. 1 (left). Cars are moving on the road, generating

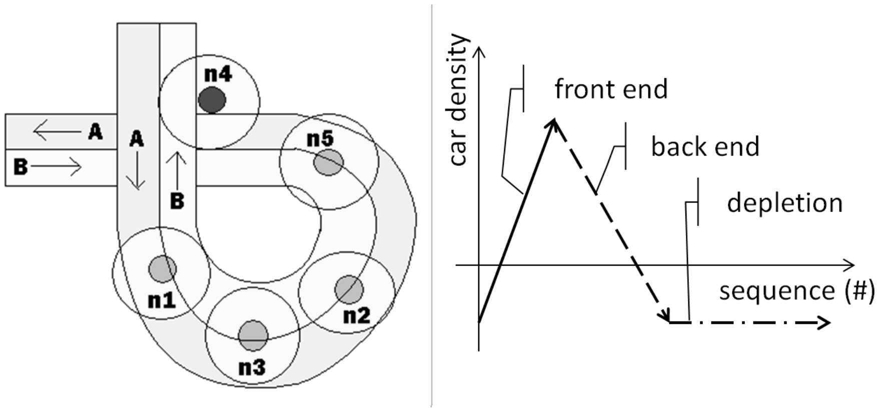

Figure 1: Example of a sensor field that monitors an uphill road (left) car density profile function (right)

events on the nearby nodes (the larger circles depict the sensing ranges). If a car passes on the road on side A, it will be seen in order by nodes n1, n3, n2, n5, n4. If we suppose that all these five nodes are in each other's radio range, all of them will be informed by the passage of the car next to any of the other nodes. Thus, when the car arrives next to node n4, this node will see a corresponding normalized local sequence similar to LS^1={(4,0.92,0), (5,0.91,-3), (2,0.95,-5), (3,0.97,-9), (1,0.9,-11)}, which means that at relative time $t = -11$ node n1 detected the car with a probability $\mu = 0.9$, then node n3 at time $t = -9$ detected the event with probability $\mu = 0.97$ and so on.

The probability values and the detection times here are just some examples. However, the differences in the successive detection times are an indication of the vehicle's speed, and can vary from car to car. Similarly, if another car running on side B arrives next to node n4, having passed next to nodes n5, n2, n3, and n1 in this order, the corresponding local sequence on node n4 will be similar to LS^2={(4,0.9,0), (1,0.99,-5), (3,0.92,-7), (2,0.91,-10), (5,0.91,-13)}.

Please note that the first sensor that detects a car on side B is not n4 but n5, since the B side of the road at the bottom of the hill is not covered by 4n's sensing range. Since n4 is the latest that detected the car at the top of the hill, it is the first in the LS description (as in LS^1). The time between car arrivals is exponentially distributed, but once a car arrives, it continues its way on the road in similar manner as the previous cars. Node n4 covers parts of both road sides, it will thus receive events originating from cars running in both directions. The events are not labeled, so the node cannot distinguish between events generated by cars passing on side A or side B of the road. Also, since more than one car can be on the same road side in the same time, the received sequences can be overlapped even with themselves.

The proposed traffic model

Based on the retrieved data we devised a model that can generate similar traffic to that of the observed.

Algorithm 1 $S = F(LS, sT, Jt_\lambda, Jm_\theta, Jm_k, \lambda)$

1: $S \leftarrow \{\oslash\}$; $i \leftarrow 0$;
2: **for** $n \leftarrow 1$ **to** $\#LS$
3: $\quad Cs \leftarrow 0$;
4: $\quad$ **while** $Cs \leq sT$
5: $\quad\quad L \leftarrow LS^n$; $D \leftarrow 0$; $i \leftarrow i + 1$;
6: $\quad\quad$ **for** $m \leftarrow 1$ **to** $|L|$
7: $\quad\quad\quad D \leftarrow D - 1/Jt_\lambda + \sim Exp(Jt_\lambda)$;
8: $\quad\quad\quad L_t[m] \leftarrow L_t[m] + D$;
9: $\quad\quad Cs \leftarrow Cs - \text{densProfile(i)} + \sim Exp(LS^n_\lambda)$;
10: $\quad\quad S \leftarrow S \cup (L^n_t[.] + Cs)$;
11: $S \leftarrow S \cup \text{PoissonPointProcess}(\lambda)$;
12: $S_\mu[.] \leftarrow\sim \text{Gamma}(Jm_\theta, Jm_k)$;

Algorithm 1 defines the proposed traffic generation model. Let's take the Fig. 1 (left) as an example throughout this description and let's say that we want to model this scenario. In this instance there will be two local sequences enforced by the topology, namely LS^1 and LS^2, as discussed earlier. Further let's say that we want to generate traffic that covers sT [sec] time period. So there is a monitored uphill road and the question is what events will be generated by the traffic. First we discuss how each sequence is successively merged into a collection of interlaced events, and then we discuss the added noise.

The generated events (line 1) are collected in the set S, which will be the result. The variable i is used to iterate the density profile. For each LS (line

2-10) we generate the event sequences (line 4-10) by starting the n-th local sequence multiple times (on the same road multiple cars are passing through), until it reaches the simulation end time sT. By starting we mean to copy the events of the n-th LS (L^n) to the S set where the event's timestamps are shifted (line 10). Between each start there is an exponentially distributed random delay $Exp(LS^n_\lambda)$ (line 9) (in our example $LS^1_\lambda = 1/80$ and $LS^2_\lambda = 1/100$), which represents the time between car arrivals (in this case there is more traffic on road A, represented by LS^1). The notation $L^n_t[.] + Cs$ means (line 10) that we select the time stamp (t) of each ([.]) event from the L^n n-th LS and add to it Cs (this is the shifting), which stores the offset when the L local sequence is started and merged with the rest of the events in (S).

We also add a time jitter to each event (line 6-8), which follows exponential distribution (shifted as to have zero expected value) of parameters $\sim Exp(Jt_\lambda)$ (line 7). This jitter models the time deviations between events in a particular sequence (each car has a different speed profile). Then, we also add noise to the events, using a λ rate Poisson point process (line 11), i.e., the time between noise events is of λ parameter exponential distribution (mean $1/\lambda$). The noisy events are uniformly distributed among the nodes. Finally, we generate the μ record for each event in S which is of $\sim Gamma(Jm_\theta, Jm_k)$ distribution (line 12). This models the inaccuracy of the detection.

As it can be seen (line 9) the inter arrival time (between sequence starts) is not i.i.d., since we introduced a density profile function (line 9, Fig. 1 (right)) which modulates the mean time between sequences starts (car arrivals). The car density profile in our case is a simple zero mean repetitive triangular function as it can be seen in Fig. 1 (right). The front end linear function defines the beginnings of a traffic batch (which is usually steeper than the back end), the back end linear function defines the end profile of a traffic batch, while the depletion constant balances the overall delay to zero, so it only changes the relative positions of cars and does not create therefore new cars in order to temporally increase the traffic.

Traffic model evaluation

In this section we present six major comparisons. Fig. 2, illustrates the events generated by the above algorithm and detected by the five nodes, over a 1500sec simulation period. The burstiness of the traffic can be well seen (modulated according to the density profile, line 9, Fig. 1 (right)). In Fig.

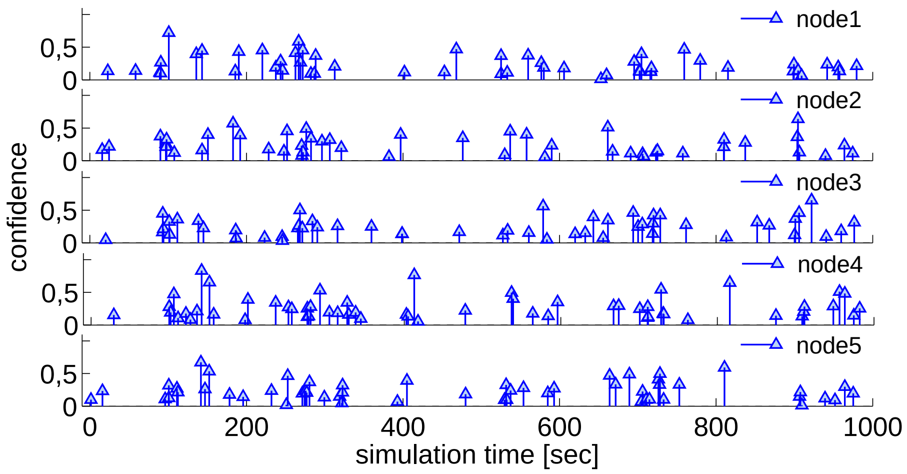

Figure 2: Mixed events obtained after multiple executions of LS^1 and LS^2

3, (top) we compare the run sequence plots of the data. The run sequence plot displays the consecutive time gaps between events that the network as a whole received, indexed by their order of occurrence. As it can be seen there are no significant shifts or trends, nor outliers in respect to the reference.

In Fig. 3, (bottom) we compare the lag plots of the data, where both axes list the run sequence (consecutive time gaps between events) as in Fig. 3, (top). The lag is chosen equal to 1 so the vertical axis is the same as the horizontal, except that it is shifted forward by one, i.e., the x coordinate

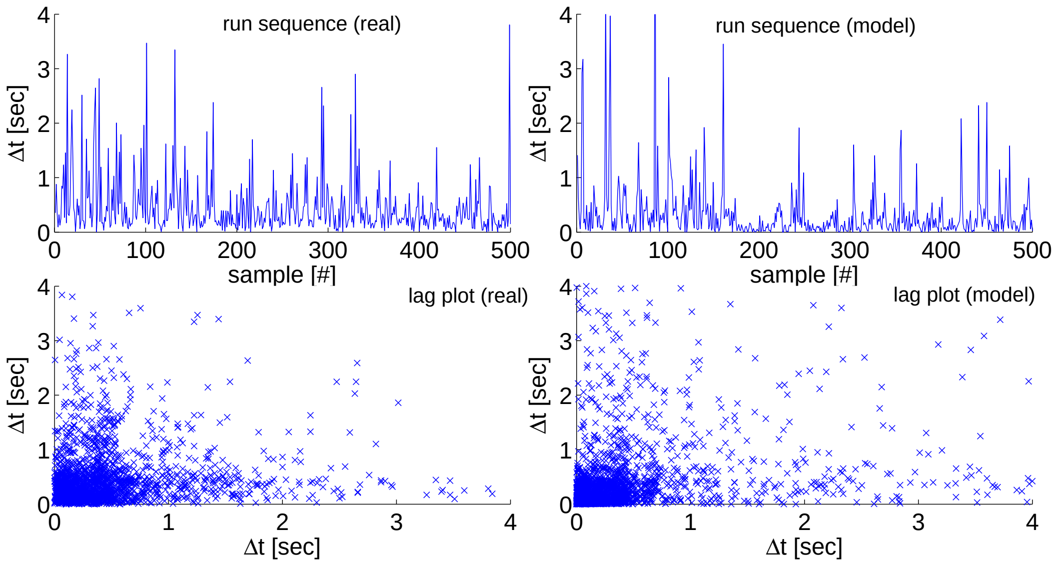

Figure 3: Run sequence and lag plot comparisons

of a particular P_k point on the lap plot represents the time gap before the $k-th$ event, and the y coordinate represents the time gap after the $k-th$ event. It can be seen that the structure is similar, and there is neither significant structure nor trend. Fig. 4, compares different probability density function (pdf) profiles (in the figures we depicted the absolute frequency instead of the relative one so that the comparison can be stricter) and the auto-correlation of the inter-event time sequences. Each curve is depicted with a 95% confidence interval (the empirical probability density function is assumed to be approximately normal, since the central limit theorem applies). Fig. 4, (top, left) depicts the p.d.f. profile of the inter-event time originating from the network as a whole. The spikes in the profile can be traced back to the average inter-event time in the local sequences. If we study the p.d.f. profile of the inter event time on node-by-node basis - Fig. 4, (top, right) (just rearranged events) - the spikes disappear, since then it is a close estimation of the p.d.f. profile of the time between car arrivals (assuming here the topology from Fig. 1 (left)). Both profiles closely match. In Fig. 4, (bottom, left) we can see the p.d.f. density profile of the event occurrence probability, which can be closely approximated by a Gamma distribution. Finally, Fig. 4,

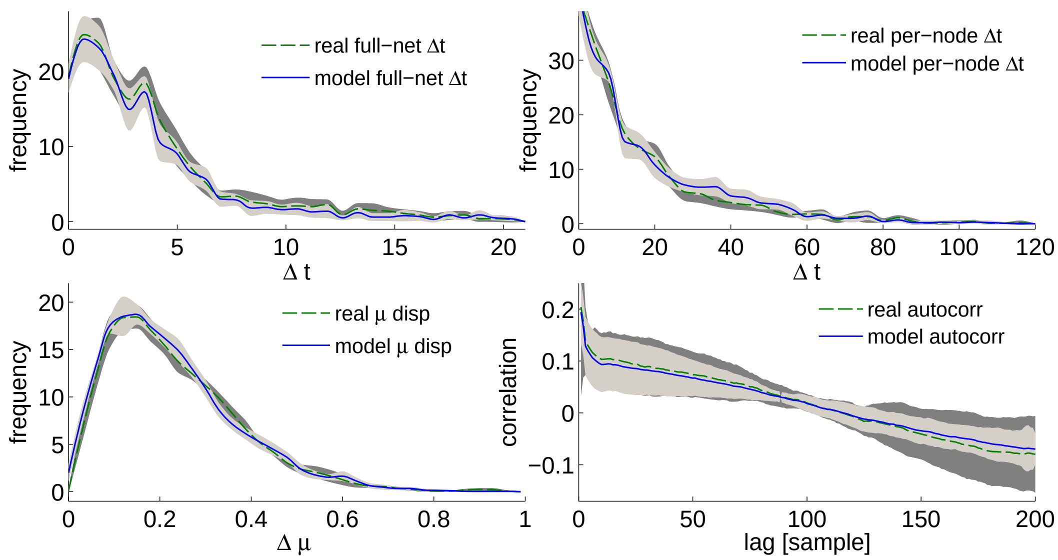

Figure 4: Autocorrelation and various empirical density function profile comparisons

(bottom, right) shows the auto-correlation of the inter-event time sequences, which is caused by the burstiness of the traffic. It is well extrapolated by the discussed density profile.

Conclusion

In this report we have outlined a probabilistic approach to model non i.i.d. (independent and identically distributed) aperiodic traffic, which is then used to demonstrate the effectiveness of an earlier proposed event forecasting method.

The traffic model was validated against well documented real world measurements of various types. We presented numerous comparisons, and the main features were found to match the features of real measurements.

Bibliography

[1] G. Ollos and R. Vida, "Adaptive event forecasting in wireless sensor networks," in *IEEE Vehicular Technology Conference (IEEE VTC 2011)*, May 2011.

[2] Y. Chakhchoukh, P. Panciatici, P. Bondon, and L. Mili, "Robust short-term load forecasting using projection statistics," in *2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, dec. 2009, pp. 45 –48.

[3] S.-M. Chen and C.-D. Chen, "Taiex forecasting based on fuzzy time series and fuzzy variation groups," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 1–12, feb. 2011.