Imperial College London

Statistical Inference of Hawkes Processes with Deep Learning

Tom Keane

Supervised by: Dr. Seth Flaxman

17 September 2020

Problem Overview

Imperial College London

Aim

Perform statistical inference on Hawkes processes which have been observed in an aggregated form using deep learning.

Stochastic Point Processes are used to describe random phenomena through time.

The Hawkes process (Hawkes, 1971) is a **self-exciting** stochastic point process.

The Hawkes Process

Imperial College London

Notation

The realisations of a stochastic point process are denoted by $\{t_i\}_{i=1,\dots,n}$. Alternatively, these can be represented by the corresponding count process; $\{N(t)\}_{t\geq 0}$.

Conditional Intensity of a Hawkes Process

$$\lambda^*(t) = \mu + \int_0^t k(t-u)\,\mathrm{d}N(u) \tag{1}$$

where $\mu \geq 0$ and $k : (0,\infty) \mapsto [0,\infty)$.

Kernel Function

The form of the kernel function defines the self-exciting property. Hawkes (1971) defined this as:

$$k(t-u) = \alpha\exp(-\beta(t-u)) \tag{2}$$

Aggregated Hawkes Process

The ability to precisely observe event times is reduced in certain applications.

This censoring of precise event times leads to stochastic processes being represented by the count of events over disjoint intervals of time.

Precise Event Times

$$\{1.32610, 1.65561, 1.68044, 1.69923, 1.75455, 2.07640, 2.88915, \dots\}$$

$$\downarrow$$

Observed Event Times

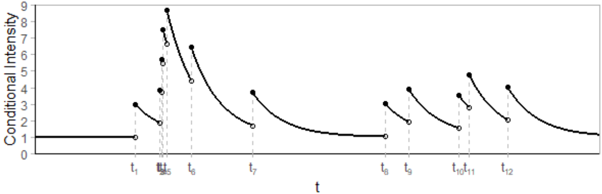
$$\{1.3, 1.7, 1.7, 1.7, 1.8, 2.1, 2.9, \dots\}$$

$$\downarrow$$

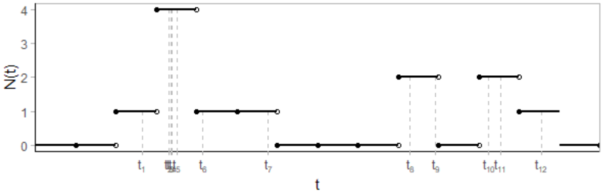
Aggregated Data (discretised in steps of $\Delta = 0.1$)

$$\left\{N_j^{(\Delta)}\right\}_{j=1,\dots,\frac{T}{\Delta}} = \{0, \dots, 1, 0, 0, 0, 3, 1, 0, 0, 1, \dots\}$$

The Hawkes Process



(a) Plot of Conditional Intensity Function



(b) Plot of Aggregated Hawkes Process (discretised in steps of $\Delta = 0.5$)

Figure 1: Hawkes process with parameter $\{\alpha, \beta, \mu\} = \{2, 2.5, 1\}$

Data Simulation

Simulation of an Aggregated Hawkes process requires the specification of:

- Two of the following three:
  - self-exciting rate, α
  - intensity decay rate, β
  - branching ratio, η
- baseline intensity, μ
- time horizon, T
- discretisation step-size, Δ

Branching Ratio

$$\eta = \int_0^\infty \alpha \exp(-\beta(s)) \, \mathrm{d}s = \frac{\alpha}{\beta}$$

Data Simulation

Imperial College London

Realisations of Hawkes processes were simulated using the 'hawkes' package in R (Zaatour, 2014).

Simulated data was aggregated using a user defined function.

Processes were simulated with an approximately constant expected activity.

Expected Activity

$$\mathbb{E}(\lambda^*(t)) = \frac{\mu}{1-\eta}$$

Estimating η and μ

The estimation of $\{\alpha, \beta, \mu\}$ was approached as a regression problem using the aggregated Hawkes process as the independent variables.

Estimation of the branching ratio, η, and baseline intensity, μ, was achieved using a Neural Network.

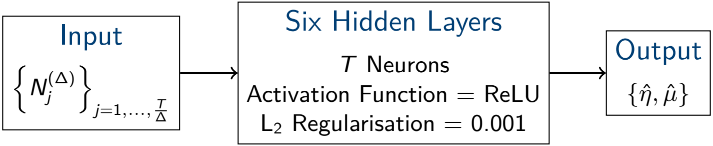


Figure 2: Neural Network Structure Used for Estimation

Estimating α and β

Estimation of the self-exciting rate, α, was achieved using a normal linear model.

This was achieved using the mean maximum event count as the independent variable.

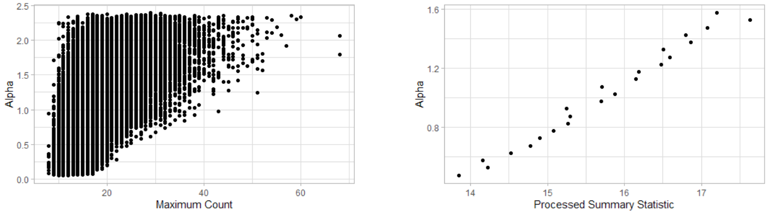
Link between α and maximum count

Inter-arrival times of a Hawkes process are distributed according to an exponential distribution with intensity $\lambda^*(t)$.

Due to the jump in conditional intensity, the expected waiting time, $1/\lambda^*(t)$, decreases after an event.

This decrease is proportional to the value of α.

Estimating α and β



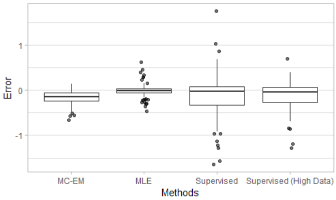
(a) Relation between maximum count and α

(b) Relation between pre-processed summary statistic and α

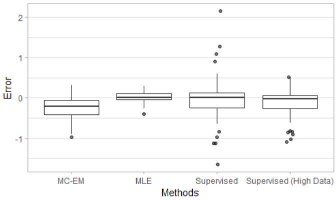
Figure 3

The estimated values of α and η were then used to calculate $\hat{\beta} = \frac{\hat{\alpha}}{\hat{\eta}}$.

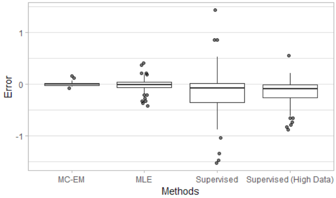
Supervised Learning Results



(a) Comparison of α estimation



(b) Comparison of β estimation



(c) Comparison of μ estimation

Figure 4: Comparison of Parameter Estimation Methods

Supervised Learning Discussion

Imperial College London

The MC-EM algorithm (Shlomovich et al., 2020) outperforms the Supervised Learning method in accuracy.

This was an unsupervised test using supervised learning algorithms so performance may be improved.

Supervised learning method has significantly lower computational time in current implementations.

Supervised learning performance was robust across parameter ranges, most importantly the level of discretisation which ranged from 0.25 to 6.

Variational Auto-Encoders (VAEs)

Imperial College London

Aim: To perform inference on $\theta = \{\eta, \mu\}$ given an observation of an Aggregated Hawkes process, x.

Variational Auto-Encoders combine deep learning and variational inference to approximate $p(z|x)$, where z is an unobserved latent variable.

Variational Auto-Encoders provide a generative model which can allow for Bayesian inference on $p(\theta|x)$ using MCMC sampling (Mishra et al., 2020).

Variational Auto-Encoders (VAEs)

A VAE uses a neural network to 'encode' data, x, into a continuous lower dimensional latent variable, z.

The input is reconstructed or 'decoded' from the latent variable using a separate neural network

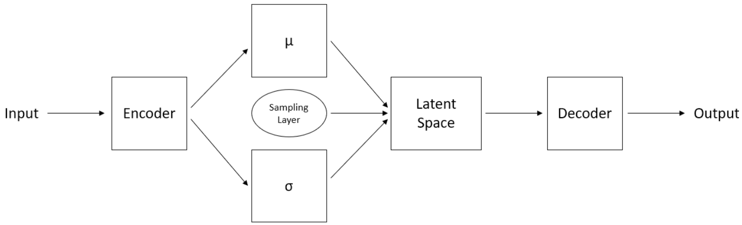


Figure 5: Basic Structure of a VAE with a MVN latent variable

Variational Auto-Encoders (VAEs)

Imperial College London

Evidence Lower Bound (ELBO)

The Evidence Lower Bound forms a lower bound for the log marginal likelihood of the data, x.

$$\log(p_\theta(x)) = D_{KL}(q_\phi(z|x) \,||\, p_\theta(z|x)) + \text{ELBO}$$

$$\log(p_\theta(x)) \geq \underbrace{\mathbb{E}_{q_\phi(z|x)}\Big[\, \log p_\theta(x|z) \,\Big] - D_{KL}(q_\phi(z|x) \,||\, p_\theta(z))}_{\text{ELBO}}$$

The form of the likelihood $p(x|z)$ is required to be specified for the training of the VAE.

Poisson VAE

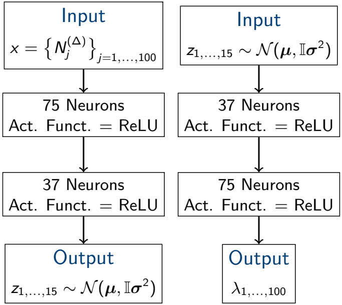
Poisson Log Likelihood

$$\log(p(x|z)) \propto \sum_{j=1}^{100} N_j^{(\Delta)} \log(\lambda_j) - \log(N_j^{(\Delta)}!)$$

Interpreting $\lambda_{1,\dots,100}$

Through the link with the Poisson Process:

$$N_j^{(\Delta)} \sim \mathrm{Poi}\left(\int_{(j-1)\Delta}^{j\Delta} \lambda^*(s)\,\mathrm{d}s \right)$$

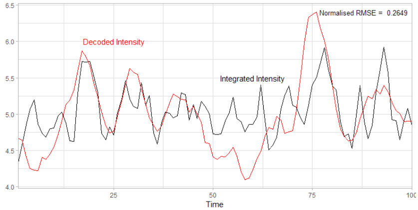


(a) Encoder (b) Decoder

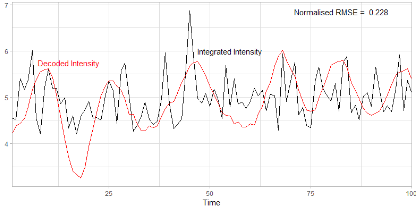
Figure 6: Poisson VAE structure

Poisson VAE Results

Imperial College London



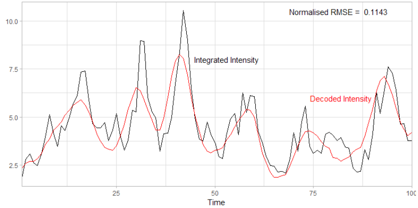
(a) Test 1: $\{\alpha, \beta, \mu\} = \{0.2, 1.0, 4\}$



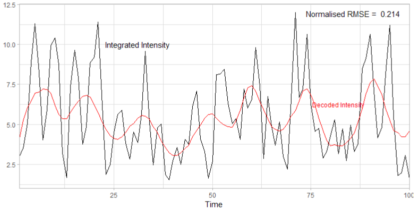
(b) Test 2: $\{\alpha, \beta, \mu\} = \{0.6, 3.0, 4\}$

Figure 7: Reconstruction of Integrated Intensity using Poisson VAE

Poisson VAE Results



(a) Test 3: $\{\alpha, \beta, \mu\} = \{0.7, 1.0, 1.5\}$



(b) Test 4: $\{\alpha, \beta, \mu\} = \{2.1, 3.0, 1.5\}$

Figure 8: Reconstruction of Integrated Intensity using Poisson VAE

Bayesian Inference

Aim: Use MCMC sampling to generate a sample from $p(z|x)$

The unnormalised posterior is:

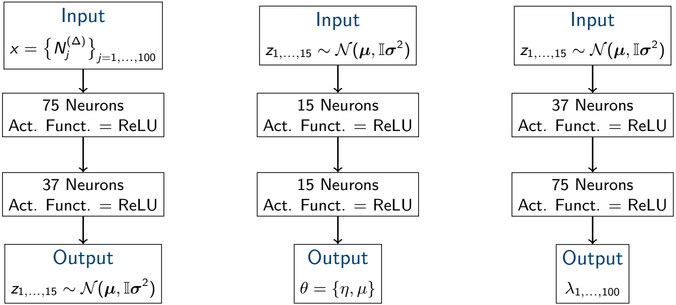
$$
\begin{aligned}
p(z|x) &\propto p(x|z)p(z) \\
&\propto \prod_{j=1}^{100} {\lambda_j}^{N_j^{(\Delta)}} \exp(-\lambda_j) \prod_{i=1}^{15} \exp(-0.5(z_i^2))
\end{aligned}
$$

where $\lambda_j = \Big(\mathrm{Decoder}(z)\Big)_j$.

Therefore, a sample from $p(z|x)$ can be drawn using the Poisson Decoder.

Dueling Decoder

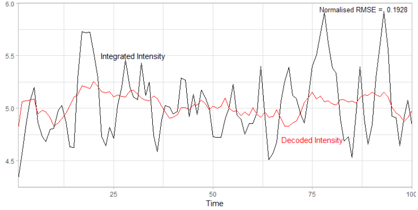
The Dueling decoder framework was proposed in Seybold et al. (2019).



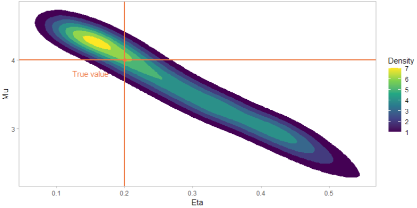
(a) Encoder (b) Primary Decoder (c) Secondary Decoder

Figure 9: Dueling Decoder structure

Dueling Decoder Results



(a) Reconstruction

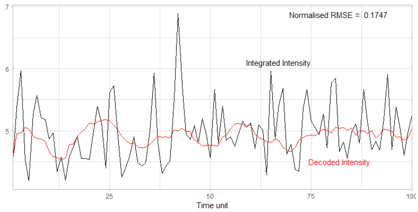


(b) Density plot of Predictive posterior for θ

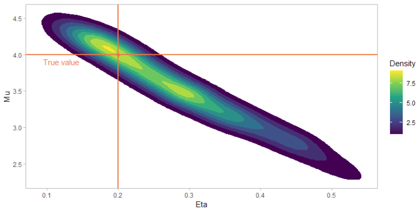
Figure 10: Performance of Dueling Decoders on Test 1: $\{\alpha, \beta, \mu\} = \{0.2, 1.0, 4\}$

Dueling Decoder Results

Imperial College London



(a) Reconstruction

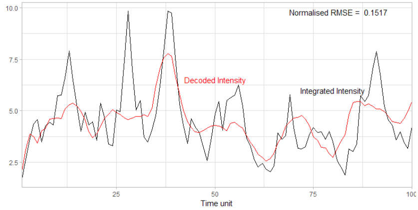


(b) Density plot of Predictive posterior for θ

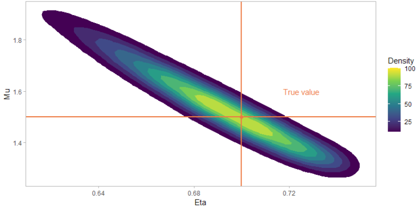
Figure 11: Performance of Dueling Decoders on Test 2: $\{\alpha, \beta, \mu\} = \{0.6, 3.0, 4\}$

Dueling Decoder Results

Imperial College London



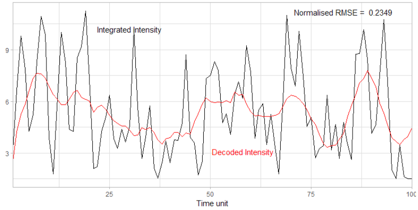
(a) Reconstruction



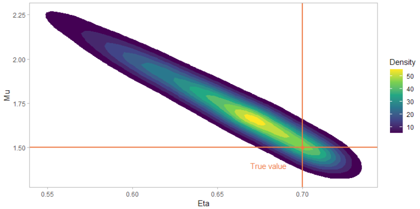
(b) Density plot of Predictive posterior for θ

Figure 12: Performance of Dueling Decoders on Test 3: $\{\alpha, \beta, \mu\} = \{0.7, 1.0, 1.5\}$

Dueling Decoder Results



(a) Reconstruction



(b) Density plot of Predictive posterior for θ

Figure 13: Performance of Dueling Decoders on Test 4: $\{\alpha, \beta, \mu\} = \{2.1, 3.0, 1.5\}$

VAE Discussion

Imperial College London

Inference on the distribution of $\theta|z$ was succesfully performed using VAEs.

The Poisson VAE reconstructed the intensities well.

The reconstruction performance of the Dueling decoder was reduced, but the density estimation performed well.

The reconstruction performance could be improved using an alternative weighting of the loss function.

Further work on the training of these VAEs is necessary.

Conclusion

Aim

Perform statistical inference on Hawkes processes which have been observed in an aggregated form using deep learning.

Inference on $\{\alpha, \beta, \mu\}$ was achieved using a blend of supervised learning techniques.

Inference on the joint distribution of $\theta = \{\eta, \mu\}$ was performed using Variational Auto-Encoders.

Citations I

Imperial College London

A. G. Hawkes. Spectra of Some Self-exciting and Mutually Exciting Point Processes. *Biometrika*, 58(1):83–90, 1971.

S. Mishra, S. Flaxman, and S. Bhatt. π VAE: Encoding Stochastic Process Priors with Variational Autoencoders. *arXiv preprint arXiv:2002.06873*, 2020.

B. Seybold, E. Fertig, A. Alemi, and I. Fischer. Dueling Decoders: Regularizing Variational Autoencoder Latent Spaces. *arXiv preprint arXiv:1905.07478*, 2019.

L. Shlomovich, E. Cohen, N. Adams, and L. Patel. A Monte Carlo EM Algorithm for the Parameter Estimation of Aggregated Hawkes Processes, 2020.

R. Zaatour. *Hawkes: Hawkes Process Simulation and Calibration Toolkit*, 2014. URL `https://CRAN.R-project.org/package=hawkes`. R package version 0.0-4.