UNIVERSITY COLLEGE LONDON

DOCTORAL THESIS

Modeling Temporal point processes using Recurrent Neural Nets

Author:
Badrul ALOM

Supervisor:
Dr. Mark HERBSTER

Advised by:
Pedro Mediano (Emotech Ltd.)

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Science

July 29, 2017

Declaration of Authorship

I, Badrul ALOM, declare that this thesis titled, "Modeling Temporal point processes using Recurrent Neural Nets" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

University College London

Abstract

Faculty Name
Department of Computer Science

Master of Science

Modeling Temporal point processes using Recurrent Neural Nets

by Badrul Alom

tbc

Chapter 1

Introduction

Event sequences are common in areas such as retail, finance, and utilities. For example, the times at which a customer makes a purchase from an online retailer, the time between financial transactions in the stock market, and the times at which utility service customers make high use of gas and electricity. These are a particular category of event sequences in which the temporal factor could be said to be a key dimension in predicting the next occurence. In such cases, understanding and predicting user behaviors, based on purely the temporal aspect of their occurence, are of great practical, economic, and societal interest.

1.1 Motivation

The impetus for this research was to estimate the liklihood of a person being interested in listening to music in the current time-period based only on their listening history. The raw data is a series of timestamps denoting when songs were played. The implicit assumption is that a persons playlist history contains a temporal pattern such as a combination of daily and weekly schedule, that can be modelled. This concept can also be applied to other areas where a temporal pattern is thought to exist at an individual user level, such as the repeat purchase of household products, or the sleeping and eating habits of a person.

The objective of the research is to evaluate the effectiveness of several different techniques for determining the probability of a user listening to music in a given period. One such application of this would be home audio devices which could anticipate when a user would like to listen to music and play without user intervention.

The research was guided by Emotech Ltd. the creators of Olly [7].

1.2 Point Processes

One way of modeling the problem is as series of events and non-events known as a temporal point process. This has a rich history of methods as outlined in the literature review.

1.3 The dataset

The dataset being used in this analysis is the LastFM1k dataset containing the listening history of a thousand LastFM listeners.

The dataset contains the timestamp, userId, and trackId of users listening habits over a number of years (2005-2009).

1.4 Structure of the report

tbc

Chapter 2

Literature Review

Within literature the most common way of tackling the problem is through Point Processes. More recently both Gaussian Processes and Deep Learning have been applied.

2.1 Point Processes

A temporal point process [1] is a way of modeling events data with t being a sequence of a fixed period interval with $t_i \epsilon R + and i \epsilon Z+$.

It can be modelled as a series of inter-event times (time until next event) or the number of events occurring in the interval. Examples of point processes are the times between financial transactions [3], and the times at which a customer makes a purchase from an online retailer *** insert ref***.

At its simplest a point process can be represented as:

$$\xi = \sum_{i=1}^{n} \delta_{X_i},$$

where δ denotes the Dirac measure, a probability measure of whether a set contains point x or not.

Point processe models seek to estimate the probability of an event happening at time t, based on an event history upto, but not including, time t.

There are different ways of representing point process data as shown in figure 2.1, with the inter-event time being the most common.

2.1.1 Conditional Intensity Function

A conditional intensity function is the most popular method for modeling point processes [2]. In this method the probability of an event $\lambda(t)$ is derived from a stochastic model such as the Poisson process.

Conditional intensity functions can be inhomogenous such as with a Gaussian Kernel $\lambda(t) = \sum_{i=1}^{k} \alpha(2\pi\sigma_i^2)^{-1/2} exp(-(t-c_i)^2/\sigma_i^2)$, for $t \epsilon [0, T]$ where c_i and σ are fixed center and standard deviations, respectively, and α_i is the weight for kernel i.

Or they can vary in intensity such as with the self-exciting (Hawkes) process where the intensity is determined by previous events through the parametric form $\lambda(t) = \mu + \beta \sum_{t_i<t} g(t-t_i) where g is some non-negative kernel function.$

However, as noted by Wass et. al [10], conventional point process models often make unrealistic assumptions about the generative processes of the event sequences. The conditional intensity function make various parametric assumptions about the

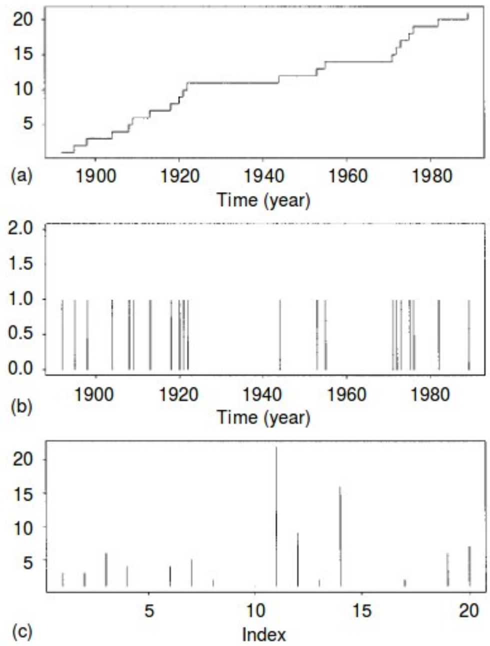

FIGURE 2.1: Three different representations of the same point-process a) cumulative count b) date of occurence c) interval time between floods

latent dynamics governing the generation of the observed point patterns. As a consequence, model misspecification can cause significantly degraded performance using point process models.

2.2 Deep RNN Point Process Models

In recent years deep learning has demonstrated the power to learn hierarchical nonlinear patterns on large-scale datasets [5] through multiple layers of abstraction (e.g. multi-layer feedforward neural networks). It has achieved state-of-the-art performances on a wide range of applications, such as computer vision [4], natural language processing [9], and protein structure prediction [6].

However it has not been applied to temporal point processes until recently with Xiao et. al [10] applying Generative Adversarial Networks (GANS) to the problem. GANs consist of two neural network models - a generator tasked with generating (i.e. predicting) a future sequence of events based on the history, and a discriminator tasked with detecting the true (fround truth) sequence amongst the generated ones.

For measuring the loss between a generated and true sequence, the authors found the Wassertein-Distance [8] performed better than Maximum Likihood Estimate (MLE) which they remarked "may suffer from mode dropping or get stuck in an inferior local minimum".

Their findings showed that where as parametric point process models work better with problems where a parametetric form exists, with real world data a GAN model with Wasserterin-Distance outperformed all other models (including an RNN model using MLE). This signals a promising new direction for temporal point process research.

Chapter 3

Methodology

The research set out to evaluate different methods for estimating when a listening event is likely to occur. In this section we describe the theory behind each one. The list of methods that were evaluated are:

- Bayesian Frequency analysis (BFrq)
- Support Vector Regression (SVR)
- RBF Regression (RBF)
- Recurrent Neural Networks (RNN)

We start with discussing the data preparation that was performed in order to perform the analysis, and the evaluation criteria, before describing each of the methods in turn. In the next chapter we show the results of the preliminary analysis followed by an evaluate of each method.

3.0.1 Data Preparation

The analysis was carried out in Python (via Jupyter notebooks) running on Ubuntu. The raw data consisted of timestamps and userIDs. These were loaded as-is into a SqlLite3 database.

UserIDs were then converted to integer (e.g. 'User0005' became '5') and a period table was defined of n minute intervals to which all timestamps could be mapped to. n was chosen to be 30 although it is possible to re-run the analysis for other levels of granularity.

More siginificantly the data, which contained entried for the times at which each user listened to music, was supplemented with all the times they did *not* listen to music, between their date of their first and last play. As can be imagiend this increased the size of the dataset signifciantly from <> rows to <> rows.

This was necessary in order to evaluate the success of the models in predicting when users would like to listen to music vs. when they would not. From here the data was modelled in two different ways.

For the BFrq model the counts of plays and non-plays were aggregated to the userID, timeslot level, where timeslot was a period within a week. Specifically timeslot consisted of weekday-hourOfDay-start minute of period. Notice here that the temporal dimension is lost - counts from the same period in two separate weeks are aggergated together.

THe second method of modeling the data retains the temporal aspect and is inline with times-series approaches. The data was structure into the following features: PeriodID, UserID, HrsFrom6pm, isSun, isMon, isTue, isWed, isThu, isFri, isSat, t, t1, t2, t3, t4, t5, t10, t12hrs, t24hrs, t1wk, t2wks, t3wks, t4wks

3.0.2 Evaluation critera

3.0.3 Bayesian Inference Frequency analysis

We start with a simple inituitive method in which we set up a time period

3.0.4 Logistic Regression

3.0.5 Suppert Vector Regression

3.0.6 RBF Regression

3.0.7 Gaussian Processes

3.0.8 Recurrent Neural Networks

Python, via Jupyter notebooks was the primary source of analysis with some SQL as the data was manipualated and stored in a SqlLite3 database first.

Note: In the interest of time, analysis was carried out on 381 of the full 1000 user dataset.

3.1 Preliminary analysis

3.1.1 Context

Let us first consider the real-world aspect of the data we have - the timestamps on which users played a song. This does not necessarily mean they played the song in its entirety. Indeed initial analysis shows plenty of cases where a song was started, seconds after the previous one, suggesting that the dataset contains both tracks that were played and tracks that were skipped. For our purposes we can consider both these to be the same as they both indicate that the user was interested in playing music at time t.

We can also assume that the song plays are not i.i.d, in that the probability of a play event at time t+1 is significantly higher if there was an event at time t.

3.1.2 Basic statistics

We start with some basic information about the raw data file that was recieved. Num. of rows: 7,500,000 Num. of users: 381 Num. of unique timestamps: 7,226,934 earliest timestamp: 2005-02-14 00:02:10 latest timestamp: 2009-06-19 17:37:23

What does this tell us? Well we can deduce there are approximately 19.6k timestamps per user on average. This feels like a high number. Fig 2.1

3.1.3 Outlier analysis

3.2 Bayesian Inference

Here we apply a counting process

3.3 How to cite references in Latex

[Reference1]

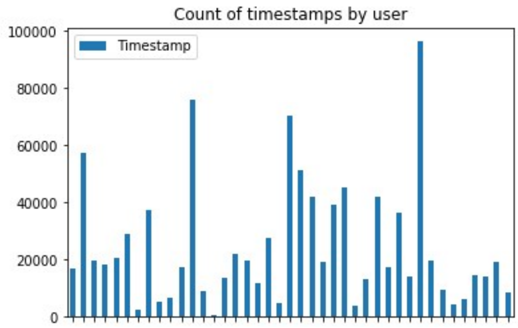

FIGURE 3.1

This document is an example of `thebibliography` environment using in bibliography management. Three items are cited: *The LaTeX Companion* book [**latexcompanion**], the Einstein journal paper [**einstein**], and the Donald Knuth's website [**knuthwebsite**]. The LaTeX related items are [**latexcompanion**, **knuthwebsite**].

Chapter 4

Chapter Title Here

4.1 Main Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam ultricies lacinia euismod. Nam tempus risus in dolor rhoncus in interdum enim tincidunt. Donec vel nunc neque. In condimentum ullamcorper quam non consequat. Fusce sagittis tempor feugiat. Fusce magna erat, molestie eu convallis ut, tempus sed arcu. Quisque molestie, ante a tincidunt ullamcorper, sapien enim dignissim lacus, in semper nibh erat lobortis purus. Integer dapibus ligula ac risus convallis pellentesque.

4.1.1 Subsection 1

Nunc posuere quam at lectus tristique eu ultrices augue venenatis. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam erat volutpat. Vivamus sodales tortor eget quam adipiscing in vulputate ante ullamcorper. Sed eros ante, lacinia et sollicitudin et, aliquam sit amet augue. In hac habitasse platea dictumst.

4.1.2 Subsection 2

Morbi rutrum odio eget arcu adipiscing sodales. Aenean et purus a est pulvinar pellentesque. Cras in elit neque, quis varius elit. Phasellus fringilla, nibh eu tempus venenatis, dolor elit posuere quam, quis adipiscing urna leo nec orci. Sed nec nulla auctor odio aliquet consequat. Ut nec nulla in ante ullamcorper aliquam at sed dolor. Phasellus fermentum magna in augue gravida cursus. Cras sed pretium lorem. Pellentesque eget ornare odio. Proin accumsan, massa viverra cursus pharetra, ipsum nisi lobortis velit, a malesuada dolor lorem eu neque.

4.2 Main Section 2

Sed ullamcorper quam eu nisl interdum at interdum enim egestas. Aliquam placerat justo sed lectus lobortis ut porta nisl porttitor. Vestibulum mi dolor, lacinia molestie gravida at, tempus vitae ligula. Donec eget quam sapien, in viverra eros. Donec pellentesque justo a massa fringilla non vestibulum metus vestibulum. Vestibulum in orci quis felis tempor lacinia. Vivamus ornare ultrices facilisis. Ut hendrerit volutpat vulputate. Morbi condimentum venenatis augue, id porta ipsum vulputate in. Curabitur luctus tempus justo. Vestibulum risus lectus, adipiscing nec condimentum quis, condimentum nec nisl. Aliquam dictum sagittis velit sed iaculis. Morbi tristique augue sit amet nulla pulvinar id facilisis ligula mollis. Nam elit libero, tincidunt ut aliquam at, molestie in quam. Aenean rhoncus vehicula hendrerit.

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

`\hypersetup{urlcolor=red}`, or

`\hypersetup{citecolor=green}`, or

`\hypersetup{allcolor=blue}`.

If you want to completely hide the links, you can use:

`\hypersetup{allcolors=.}`, or even better:

`\hypersetup{hidelinks}`.

If you want to have obvious links in the PDF but not the printed text, use:

`\hypersetup{colorlinks=false}`.

Bibliography

[1] DJ Daley and D Vere-Jones. *An introduction to the theory of point processes*. 2003.

[2] Nan Du et al. "Time-sensitive recommendations from recurrent user activities". In: *Advances in Neural Information Processing*. NIPS. 2015.

[3] Robert Engle and Jeffrey R. Russell. "Autoregressive Conditional Duration: A New Model for Irregularly Spaced Transaction Data". PhD thesis. University of California, 1996.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing*. NIPS. 2012.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521 (May 2015).

[6] Pietro Di Lena, Ken Nagata, and Pierre Baldi. "Deep architectures for protein contact map prediction". In: *Bioinformatics* 28 (19).

[7] Emotech Ltd. *Your robot with personality*. URL: https://www.heyolly.com/.

[8] Léon Bottou Martin Arjovsky Soumith Chintala. "Wasserstein GAN". PhD thesis. Facebook AI Research, 2017.

[9] Richard Socher et al. "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, WA: Association for Computational Linguistics, 2013, pp. 1631–1642.

[10] Shuai Xiao, Mehrdad Farajtabar, and Xiaojing Ye. "Wasserstein Learning of Deep Generative Point Process Models". PhD thesis. Georgia Institute of Technology, 2017.