Master thesis on Information and Communication Technologies
Universitat Pompeu Fabra

A generative model of user activity in the Integrated Learning Design Environment

Joan Bas Serrano

Supervisor: Vicenç Gómez and Davinia Hernández-Leo
Co-Supervisor:

July 2017

Universitat
Pompeu Fabra
Barcelona

Contents

Acknowledgement

The work has been carried out in the context of an internship funded by the DTIC-UPF Maria de Maetzu strategic program (MDM-2015-0502), linked to projects Educational Data Science project (EDS) and Understanding and Improving Social Interactions in Online Participation Platforms project (UISIOPP).

Abstract

The objective of this project has been to build a generative model of the user activity in the Integrative Learning Design Environment (ILDE) able to describe the data of different communities and which can be used for both to gain understanding about the data and to test hypothetical situations.

The model that we present is called Hierarchical Multivariate Hawkes Model and works with a two layer procedure that first draws the beginning of working sessions and then fills these sessions with events of different kinds using a Multivariate Hawkes Model.

In the project we first make an statistical temporal analysis of the data, to understand it and see the important features to be modeled, then we introduce the model, validate it, and show some of its applications.

Through these steps it has been shown that the model is able to reproduce satisfactorily the sequences of events produced by the ILDE users and it can be easily used to tackle real problems that would be difficult to face with typical statistical tools.

Keywords: Generative Model; Hawkes; Poisson; Educators Community

Chapter 1

INTRODUCTION

The Integrated Learning Design Environment (ILDE), which is presented in [1], is a platform that supports the development of learning design communities in which their members share and co-create multiple types of learning designs, and explore and comment the designs of other users.

The environment integrates different existing free and open-source tools that include co-design support for teacher communities, learning design editors from different pedagogical approaches, and interface for designs on mainstream Virtual Learning Environments. The integration of these tools is done to cover the complete life-cycle of learning design, which comprises:

- **Conceptualizing**: Work done before starting with the actual creation of learning designs with the aim of explaining the context in which the designs will be applied, sketch ideas, abstractions and drafts around design elements and their interconnections.
- **Authoring**: Producing a detailed definition of a learning design that can be deployed and executed with a specific group of learners.
- **Implementing**: Applying an authored learning design with a particular group of students using a specific virtual learning environment and a set of tools.

The ILDE have been developed as a part of the *Meeting teachers' co-design needs by integrated learning environments* (METIS) project [2]. The objective of this project is to contribute to the improvement of national in-service teacher training curricula, which is one of the EU Lisbon strategic goals. This is done by offering high quality material for teachers' professional development on the use of modern learning techniques and tools.

On the other hand, the project *Understanding and Improving Social Interaction in Online Participation Platforms*, e.g. [3] [1]. This is a project aimed to develop machine learning methods to be applied in computational social science. The goal of these tools is to analyze, model, and influence social behavior in order to optimize performance of a given online platform.

The research project that we are going to present is born as a collaboration between these two projects, with the objective of applying techniques used and developed in the latter to the ILDE platform. In concrete, the aim is to create a generative model able to capture the essence of the ILDE user activity, to gain understanding of the way how users interact with the platform, and be able to analyze real and hypothetical situations through the analysis and usage of this synthetic model.

1.1 MOTIVATION

As we have said, the ILDE is a very promising platform where activity with high value for our society is produced. This is why we think that it is a great opportunity to use machine learning and data science tools and techniques to contribute to a better understanding and improvement of the platform.

On the other hand, making models able to make inference in online communities and platforms is a big challenge due to the huge quantity of data produced every day in this kind of online platforms (Facebook, Twitter, ILDE, etc.). There are many approaches to that problem and we want to tackle the problem building a model with few parameters that could be interpretable to make it easier to extract

[1] https://www.upf.edu/web/mdm-dtic/projects/-/asset_publisher/Ef1was9TxNY4/content/id/4072448#.WV6wf58xA8o

conclusions from the parameters and also to allow us to easily tune the parameters to analyze hypothetical scenarios.

We refer to a model as a mathematical description of the user generated activity. Such a description is necessarily constrained by the phenomena that it tries to explain or capture. For example, a model can be defined at the detailed level of a learning design of a particular user or it can model the aggregate activity on the entire platform. The type of model that we pursue in this thesis should be able to generate temporal traces of activity events, such as the ones shown in figure 1. A generative model is able to predict and to generate those instances.

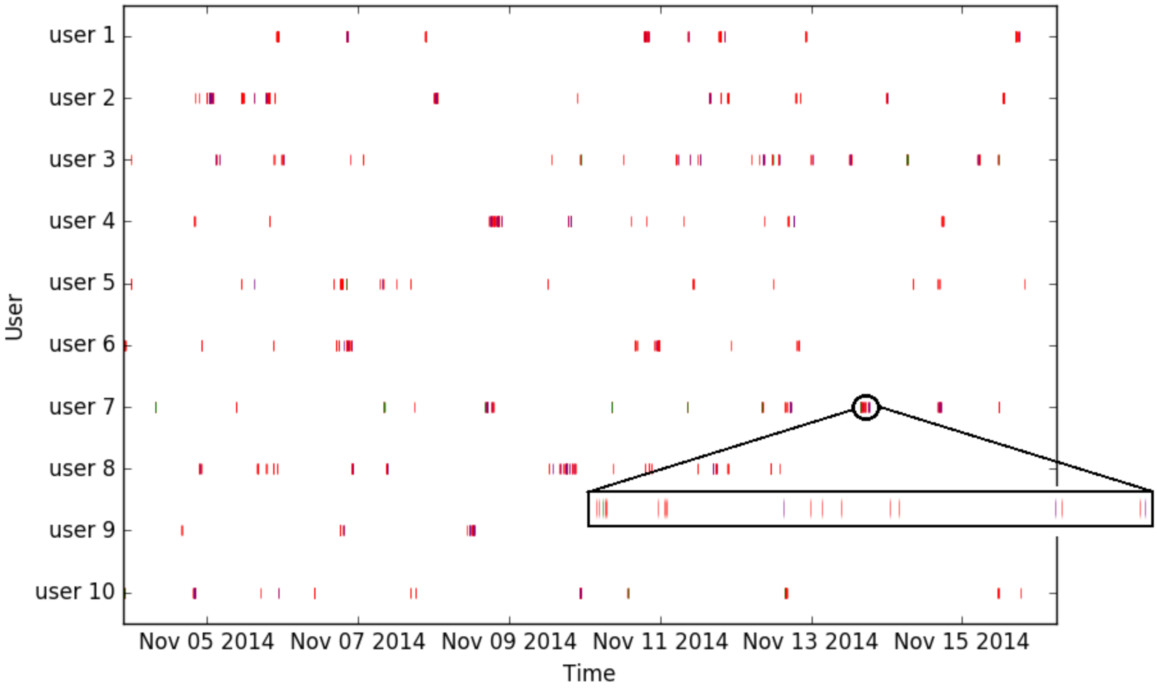


Figure 1: Activity time-line of some random users. Each vertical line corresponds to one different event.

Models can be parametric. In this case, the behavior of the model depends on the values of those parameters, that can be adjusted to better fit the data. Those parameters can explicitly relate to features. In this thesis, we focus on parsimonious models, that is, models which try to explain phenomena with as few parameters as possible. Models of this type, in contrast to fully data-driven (black-box) approaches that fit a large-number of parameters (e.g. Deep Neural Networks), are more interpretable and thus provide a better understanding of the mechanisms that determine the activity in the platform.

For these reasons, we have worked on a generative model that captures the activity of the ILDE users. This model allows a deeper and better understanding of the users and their interaction with the platform and between them, enabling us to study and analyze features of the data that would be impossible or very difficult to tackle with a basic statistical analysis. We can also make comparisons between different ILDE implementations by analyzing differences in the parameters of the models trained for each implementation.

Such model also allows us to make inference in hypothetical situations, enabling us to answer questions of the type "What would happen if...?". It can be done just by tuning of some parameters that we think that some action or change in the ILDE would modify, and observe the simulated results after this change.

Apart from all of these points, it is also important that a model such as the one that we want to build is not only exploitable in the context of the ILDE. In fact, as we will see in this project this model is applicable in lots of environments and situations.

1.2 OBJECTIVES

As we have said, the objective of this project is to build a generative model of user activity in the ILDE. We want this model to be accurate and represent well the activity in the ILDE, in order to be reliable enough to allow us to extract conclusions about the data, gain knowledge about the platform, and make predictions and test hypothetical situations with some guarantees. To have these guarantees, we also need to validate statistically that the model is able to capture well the phenomena that is trying to describe.

1.3 STRUCTURE OF THE REPORT

This report is divided in five chapters (plus the introduction) that will cover all the different parts of the project necessary to understand the final model, from the statistical analysis behind the model to the model itself and its applications. Here we present the chapters and briefly explain their content:

- STATISTICAL ANALYSIS OF THE ILDE: In this chapter we first describe the data that we have used in this project and some pre-processing steps. We then analyze statistically the temporal characteristics of this data.
- THEORETICAL BACKGROUND: This chapter presents the theory necessary to understand the final model, introducing key concepts as Point Process and a particular Point Process called Multivariate Hawkes process, which is a very important part of the final model. We also present some techniques to learn the optimal parameters of this model and generate sample data.
- HIERARCHICAL MULTIVARIATE HAWKES MODEL: This chapter describes the developed Hierarchical Multivariate Hawkes Model, its implementation, its validation, and finally we show a couple of applications to see how can we use the model to solve real problems.
- CONCLUSIONS: In this chapter we revise some results seen during the project and extract some conclusions

Chapter 2

STATISTICAL ANALYSIS OF THE ILDE DATA

In order to understand the motivations behind the generative model presented in further sessions, it is important to understand the data we are working with and its characteristics. It is why this chapter is dedicated to understand the data extracted from the ILDE. A non-temporal statistical analysis of the ILDE data can be found in [4].

In this chapter, we are first going to present the data used in this project and explain its extraction and pre-processing. After that, we are going to analyze this data from a temporal point of view, in order to detect patterns and characteristics that are relevant for the development of the model.

2.1 THE DATA

We are interested in making a generative model for the user activity. To do so, we have chosen a set of events that we think that are specially interesting because they reflect very well the different kind of actions that can be performed in the ILDE. Despite having chosen these events, our methodology is general and applicable to other types of events.

We have selected the following types of events:

- Views: When a user views the design of another user.
- Comments: When a user comments the design of another user.
- Edits: When a user edit one design or, in other words, saves the changes made in one design.

In all of this kind of events, we will focus on the *kind of action* performed (view, comment or edit) and the *user* who performed the action, and we will discard the *design* viewed, commented or edited and its *owner*.

To record these events there are some URLs that are linked to these events, so every time that one user goes to these URLs it counts as if he had performed the corresponding action. Furthermore, there might be different URLs that map to a same event. For example, in figure 2, we have the visualization of one design and we see that inside this visualization we can go to different tabs to go to different information of that design ("Untitled document", "Support Document"), and the corresponding URLs associated to these tabs are mapped to the same event (user "A" views design of user "B"), so every time that one user press one of these tabs, it counts as a new view event with the same characteristics as the previous one. This is a very serious issue, given that it creates the illusion that a user is looking many designs when in fact he is just looking at different features of the same design, and it is in principle impossible to distinguish between this case and the one when the user visits one design and some time later visits the same design again.

We address this issue by discarding events that are performed from the same user on the same design during less than five minutes after the first event, i.e. if an event A is produced at time t, any other event of the same kind performed between t and $t + 5$ minutes is removed. This criteria is quite arbitrary and of course is not perfect but it prevents us to have this sequences of many identical events compressed in a small time interval. It is important to realize that with this procedure we are not decreasing the resolution of the analysis, since this step only involves events of the same user with the same design, i.e. the user could do other things involving other designs in a smaller time-scale than 5 minutes.

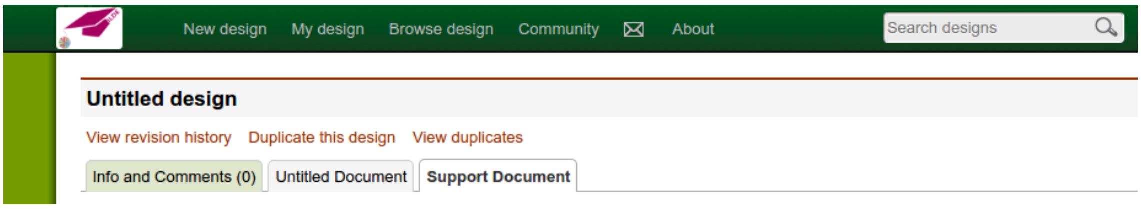


Figure 2: Part of a capture of the ILDE interface when performing a view.

In this project, we have used three different datasets corresponding to three different communities which are **Demo**, **MOOC1** and **MOOC2**. The first community (demo) has been used freely from anyone interested in design processes for learning and it is open for free exploration and contribution. There are no formal facilitators handling this community but it has been used some times for doing workshops. On the other hand, the MOOC communities were designed as a course guide to teachers through the complete cycle of designing activities ready to use in their classroom and there was the guidance of some facilitators. Participants were using ILDE for the tasks required by the MOOC and were posting messages for evaluating and reflecting on designs of other users. The main difference between these two MOOCs is the language used. While the first one only supported English, the second one allowed several languages (the learning material and the announcements were still in English). In table 1 we can see synthesized the main features of the different communities. A more detailed description and analysis of these communities and their characteristics can be found in [4]. In [5] the design of the MOOCs and its evaluation is explained.

	Demo	MOOC1	MOOC2
# Users	526	323	396
# Facilitators	-	4	~30
Languages	English, German, Spanish, Catalan, Italian, Greek, Hebrew.	English	Bulgarian, Greek, Catalan, Spanish, English.
Duration	from 2013 unil now	5 weeks	5 weeks

Table 1: Characteristics of the different ILDE communities.

As we can see, the datasets are very different and due to this heterogeneity, defining a model that is able to generate their data well is challenging.

2.2 STATISTICAL ANALYSIS

In this section we present a statistical temporal analysis in order to find temporal patterns, routines and periodicity in the data. In concrete, we are going to analyze the inter event times and the daily and weekly periodicity for the three communities.

2.2.1 INTER-EVENT TIMES

We first focus on characterizing the inter-event time distribution, i.e. what is the probability of an event (a view of a design, for example) to occur after the last event is distributed. This is a fundamental quantity to understand online communication, because it shows how the activity is spread over the time. This distribution typically is heavy-tailed (spans many orders of magnitude) and it is governed by circadian cycles [6, 7, 8, 9]. A key point to be able to define our model is to find out which are the different time-scales of our data. For example, when a user is editing one design, it produces many events concentrated in the duration of this working session (1 hour for example), and after that it might not do anything until some hours later, the next day or some days after, when he starts a new working session. This kind of behaviour produces many inter-event times in the scale of minutes and much less inter-event times in the scale of hours to days. So, both the time scales and their associated number of events will vary a lot. Figure 3 shows the empirical distributions of inter event times for the different communities, in a log-log scale in order to appreciate the different time scales. For all of the communities we can see a bimodal distribution with one peak centred in the scale of minutes (0-1 minutes to few hours) and another peak centred in the scale of days. Actually this second peak is composed of some other sub peaks that correspond to one day (1440 minutes, which is the biggest), two days (2880 minutes), three days, etc. We can also see that the first peak is much higher than the second one (almost one order of magnitude bigger), and this make sense with the behaviour that we have just explained.

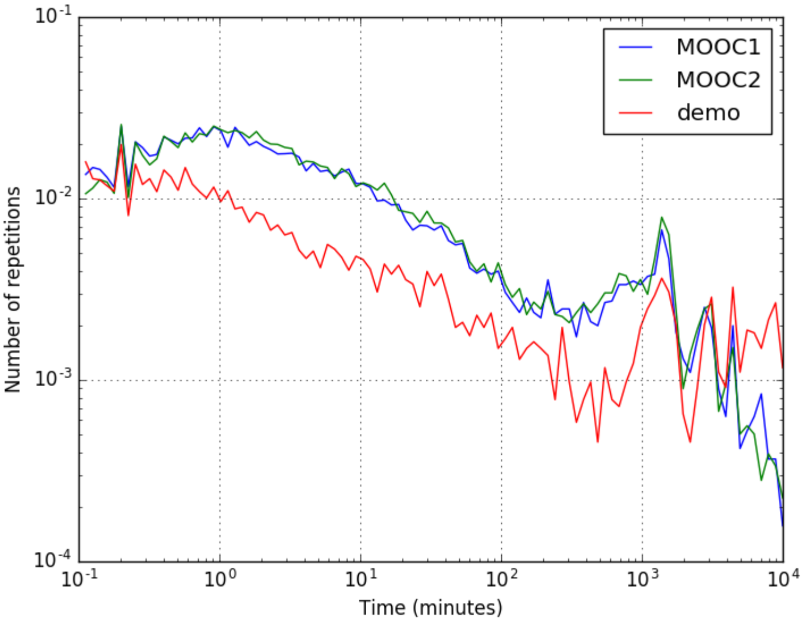


Figure 3: Empirical distributions of inter-event times for the different communities in a log-log scale.

2.2.2 WEEKLY AND DAILY PATTERNS

Because of the circadian cycles of humans and the fixed structure of days (morning, afternoon, evening...) and weeks (Monday, Tuesday, Weekend...) and their associated routines, another kind of patterns that seem reasonable in our data are those associated with these structures.

Figure 4 shows the aggregate average activity as a function of the hour of the day for the three communities. We can see that in both MOOCs the activity have a peak in the evening and for the demos the peak of activity is in the morning. So we see that in all of the cases there are clear daily patterns and that those patterns are different depending on the platform.

We can see in this figure and in the following figures in this chapter that we only report average. It is because the variance is very large, more than the dimensions of the figure. It is normal because the working routines of the users are not that strict for them to work every day at the same hour, so the fluctuations are very high.

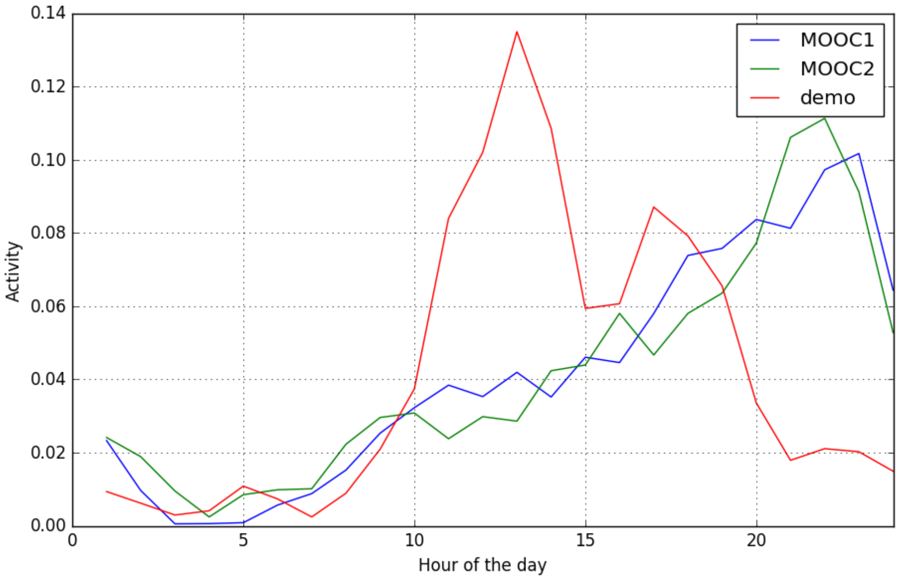


Figure 4: Empirical distributions of the total daily amount of activity in each platform.

In figure 5 we have the relative amount of activity during the days of the week and we see that for the MOOCs the activity is quite regular during the week; the main differences between two MOOCs is that the MOOC1 have a maximum of activity at Thursday and the MOOC2 have its maximum during the weekend. For the demo community, the activity is more or less constant from Monday to Friday (with a peak on Tuesday) and it decreases a lot during the weekend.

The results seen until now are an average over all of the users but there is a large variability between users. To illustrate this, we have plot activity distributions for different users selected in order to show different profiles of user. In figures 6 and 7 we have histograms of the daily and weekly amount of activity for different users respectively and we can easily see their very different profiles.

From these results we can conclude that despite there are clear global tendencies in the different communities, the users from each community there have different profiles. So, our generative model should be able to reproduce these patterns of activity. It is important because these patterns are very representative of how the users use the platform. For instance, we have seen in figures 4 and 5 that the users of the demo community use the ILDE mainly during their working hours, this is, during the morning (and also some activity during the afternoon) of the days from Monday to Friday, which is a behaviour very different to the one found in the MOOCs. We will talk more about it in chapter 4, where we will present the model and how it takes into account these patterns.

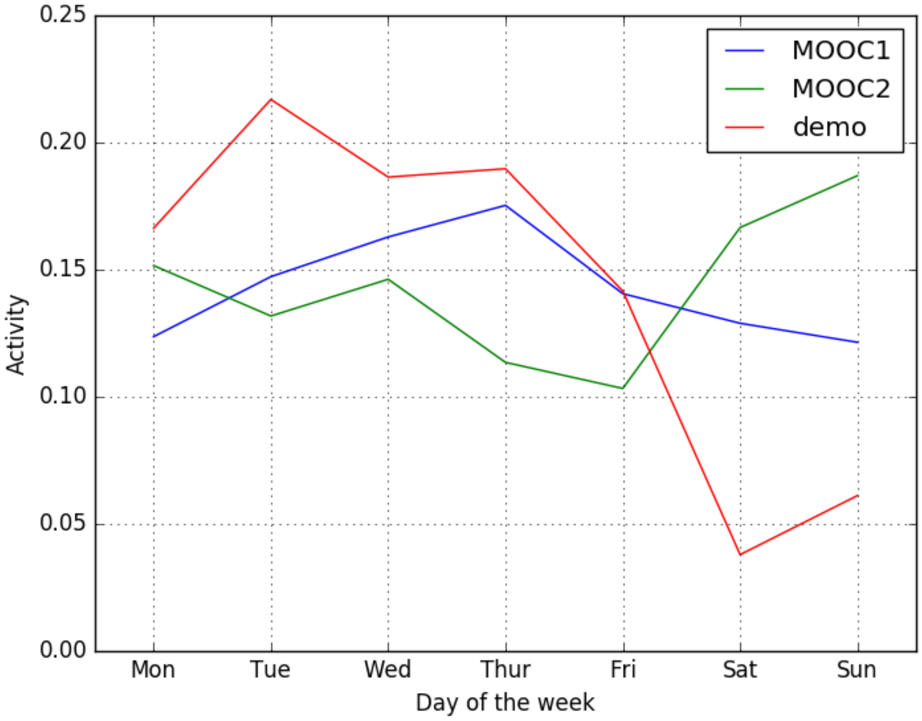
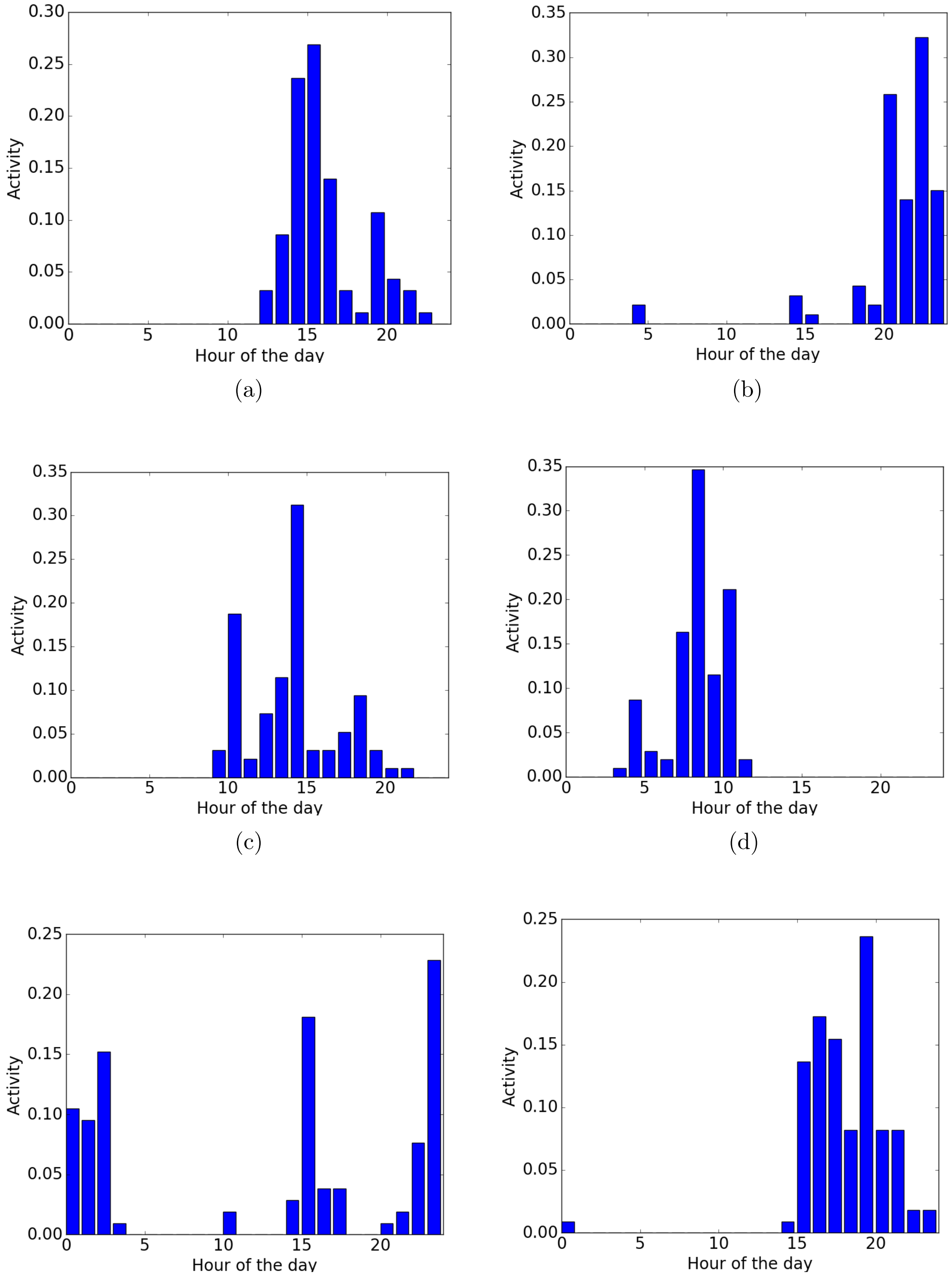


Figure 5: Empirical distributions of the total weekly amount of activity in each platform.



(a) (b) (c) (d) (e) (f)

Figure 6: Histograms of the total daily amount of activity for different users.

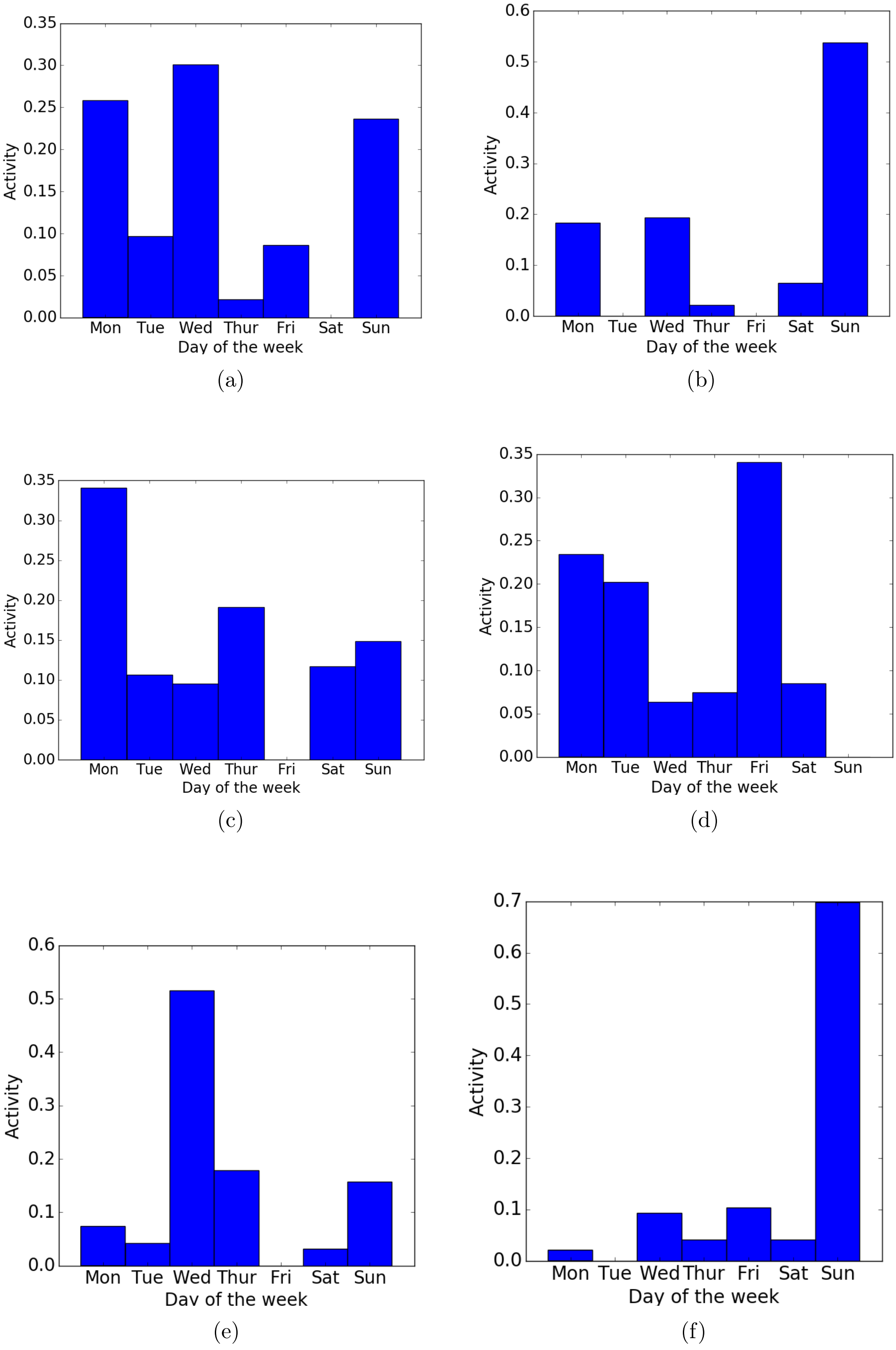


Figure 7: Histograms of the total weekly amount of activity for different users.

Chapter 3

THEORETICAL BACKGROUND

The objective of this chapter is to introduce some key theoretical concepts and definitions that we have used to build our model. In particular, the final aim is to understand the Multivariate Hawkes process, which is one of the main parts of our model. To do so, we are first going to introduce the concept of point process, then we are going to see the most simple example of point process, after which we will present a univariate Hawkes process to finally explain the multivariate Hawkes process. After that we will also dedicate two sections to explain how the parameters of the multivariate Hawkes process are estimated using its log-likelihood function, and how can we simulate sequences of events using an algorithm based on rejection sampling.

3.1 POINT PROCESS

In statistics and probability theory, a point process is a collection of mathematical points (in our case, events) randomly located on some underlying mathematical space (in our case, a one dimensional space; the time). We will write this sequences of points as $(t_i)_{i \in \mathcal{N}}$ where $\forall i \in \mathcal{N}$, $t_i < t_{i+1}$.

Associated to each point process, there is a counting process defined as

$$N(t) = \sum_{i \in N} 1_{t_i < t} \tag{3.1}$$

So, the counting process $N(t)$ counts how many points or events have been produced until time t.

The simplest example of a point process is the Poisson point process. It is completely defined in the following equation, that given a set of regions, or in our case, time intervals $(B_1, B_2 ... B_k)$, tells us the probability of each of these regions having a given number of events $(n_1, n_2 ... n_k)$:

$$P_r\{N(B_i) = n_i, i = 1, 2, 3, 4...k\} = \prod_{i=1}^{k} \frac{[\lambda |B_i|]^{n_i}}{n_i!} e^{-\lambda |B_i|} \tag{3.2}$$

Where the counting process $N(B_i)$ is the number of evens in the region (interval of time) B_i with $B_i \cap B_j = \emptyset$ for $i \neq j$, k_i is the number of intervals that we consider, and λ is the intensity of the process, a variable that determines the density of points and is related with the mean $M(B_i)$ and the variance $V(B_i)$ with the the following expression:

$$M(B_i) = \lambda |B_i| = V(B_i) \tag{3.3}$$

From equation (3.2) we can extract three very important features of the Poisson process:

- The number of points in each interval (B_i) has a Poisson distribution.
- The number of points in different (disjoint) intervals are independent.
- The distributions are stationary, meaning that they depend only on the length of the intervals.

A much more complete explanation and analysis of the Poisson process can be found in [10].

3.2 HAWKES PROCESS

Due to its homogeneity, the Poisson process can be too simple to characterize our data, where the events seem to have correlations between them as we have seen in the previous chapter. It is why we are more interested in nonhomogeneous point processes. This other kind of processes can be described with the following expression which is very similar to the expression (3.2) but with some differences that enables the inhomogeneity:

$$P_r\{N(B_i) = n_i, i = 1, 2, 3, 4...k\} = \prod_{i=1}^{k} \frac{[\Lambda(B_i)]^{n_i}}{n_i!} e^{-\Lambda(B_i)} \tag{3.4}$$

where

$$\Lambda(B_i) = \int_B \lambda(t)dt \tag{3.5}$$

and $\lambda(t)$ is an intensity function that depends on the events occurred in the past. Similarly to the homogeneous Poisson process (equation (3.3)), $\Lambda(B)$ is the expected value of events in B, and so the intensity $\lambda(t)$ can be understood as a density of events. We can see an example of counting process and conditional intensity function for two point processes in figure 8.

Now we are going to present the Hawkes process, a nonhomogeneous Poisson process that have been widely used [11],[12], [13] to model sequences of events in different areas.

The idea behind the Hawkes process is that there is an interaction between events; the occurrence of an event increases the probability of having new events in the near future. In the Hawkes process, this increase of probability decays exponentially with the time, as it can be seen in figure 8 (b). More precisely, the Hawkes process is a self-exciting process with

$$\lambda(t) = \lambda_0 + \int_{-\infty}^{t} \nu(t-s)dN(s) \tag{3.6}$$

where $N(s)$ is the number of events at time s and the function $\nu(t) = \sum_{j=0}^{P} \alpha_j e^{-\beta_j t}$, being P the number of different kinds of influence that an event can produce. So in a Hawkes process the intensity is composed for a constant base intensity (λ_0) another set of terms corresponding to increases of intensity produced by passed events. If we take into account that $dN(s) = \sum_i \delta_{st_i}$ where t_i is the time when the event i was produced, we can rewrite (3.6) as

$$\lambda(t) = \lambda_0 + \sum_{t_i<t} \sum_{j=0}^{P} \alpha_j e^{-\beta_j (t_i - s)} \tag{3.7}$$

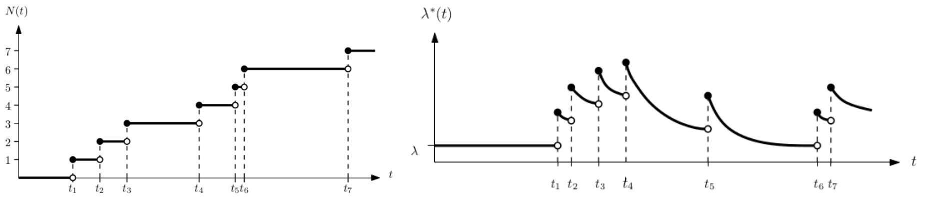
In the simplest case when $P = 1$, we have

$$\lambda(t) = \lambda_0 + \sum_{t_i<t} \alpha e^{-\beta (t_i - s)} \tag{3.8}$$

A much more deep explanation and analysis of this concepts about the Hawkes process with a more exhaustive study of its properties can be found in [12]. There we also can find a review of some applications of the Hawkes process and some procedures for estimate the parameters of the model and determine whether a series of points form a Poisson process. To simulate a Hawkes process, we can find a procedure based on the maximum likelihood in [14], which is based on the previous work of T. Ozaki [14].

3.3 MULTIVARIATE HAWKES PROCESS

Since now, the point processes that we have presented have been unidimensional point processes, thought to model single kind of event, but usually we have different kinds of events that have correlations between them. Imagine for example that we



(a) An example point process realization {t1 , t2 , ... } and corresponding counting process N(t).

(b) An example conditional intensity function for a Hawkes process.

Figure 8: N(t) and λ from of a different examples of point process from [12]

have two users A and B and we have one kind of event that is that the user A sends a message to the user B and another kind of event corresponding to the user B sending a message to the user A. In this case, we have two different kind of events and each one incentives the other. As we are going to see, it is relatively easy to modify the Hawkes process seen before to obtain a more general version able to take into account different kind of events.

In the multivariate Hawkes process, we will have an M-dimensional point process $\{(t_i^m)_{i\in\mathcal{N}}\}_{m=1,2,...,M}$, with its corresponding M-dimensional counting process $\boldsymbol{N}(t) = \big(N^1(t), N^2(t), ..., N^M(t)\big)$.

For the multivariate Hawkes process, the expressions are analogous to the ones seen for its univariate version but each kind of event m have its own intensity, determined by

$$\lambda^m(t) = \lambda_0^m + \sum_{n=0}^{M} \int_0^t \sum_{j=1}^{P} \alpha_j^{mn} e^{-\beta_j^{mn}(t-s)} dN^n(s) \tag{3.9}$$

and again for $P = 1$ it can be written as

$$\lambda^m(t) = \lambda_0^m + \sum_{n=0}^{M} \sum_{t_i^n < t} \alpha^{mn} e^{-\beta^{mn}(t-t_i^n)} \tag{3.10}$$

so as we can see the intensity have a first term corresponding to the constant rate of the kind of event m, and another group of terms that are increases of intensity caused by past events of all kinds. So, each kind of event m have its own base intensity μ_0^m and the influence of other events of kind $n = 1, 2...M$ that is determined for the parameters α^{mn} and β^{mn}. Since there are M different kinds, the number of parameters needed to generate an M-dimensional Hawkes process is $M + 2M^2$, that does not scale very well with M so as we will see, since we do not have a lot of data, we will work with two different event kinds.

As we have seen before, the average number of events in a given interval of time T is the integral of the intensity function in this interval of time. So, because of the increases of intensity of different events are independent, we can easily compute the expected value of events of kind m that are produced as a consequence of the increase of intensity caused by an event of kind n just by integrating the corresponding term:

$$\mathbf{E}(N_T^{mn}) = \int_0^T \alpha^{mn} e^{-\beta^{mn}t} dt = \frac{\alpha^{mn}}{\beta^{mn}} \left(1 - e^{-\beta^{mn}T}\right) \tag{3.11}$$

And if we want to know the total impact, from the creation of the action to the infinite, then the second term becomes zero and we have $\mathbf{E}(N^{mn}) = \frac{\alpha^{mn}}{\beta^{mn}}$.

For more information about the Multivariate Hawkes process, the thesis [15] offers a very complete and deep mathematical analysis.h

Because this is the technique that we are going to use in our model, we are going to show how to calculate the likelihood of a sequence of events given a set of parameters and after that we are going to see a technique to simulate multivariate Hawkes processes.

3.3.1 MAXIMUM LIKELIHOOD FUNCTION ESTIMATION

The likelihood is a function of the parameters of a model given data. It tells how likely is that our model generates such data. For this reason, maximizing this function (i.e. finding the parameters for which the likelihood takes its maximum value), gives us the optimal parameters.

Usually the log-likelihood (logarithm of the likelihood) is used instead of the likelihood. It is done because the logarithm have some properties that are very useful when computing the function, such as that the multiplicative terms in the likelihood become summed terms after the application of the logarithm. A part from that, the logarithm is a monotonically increasing function, which means that it achieves its maximum value at the same points as its argument, and so maximizing the log-likelihood is the same as doing it for the likelihood.

LOG-LIKELIHOOD

Given a particular sequence of events composed by a sub sequence for each dimension ($\{t_k^m\}_{k=1,2,...}$ for $m = 1, 2...M$) in the interval $[0, T]$, the log-likelihood function for a multivariate Hawkes process with parameters θ is the sum of the log-likelihoods for each dimension:

$$\ln L(\theta|\{t_k^m\}_{m=1,2,...M}) = \sum_{m=1}^{M} \ln L^m(\theta|\{t_k^n\}_{n=1,2,...M}) \tag{3.12}$$

As it is deduced in [16], each of the log-likelihoods in the summation can be written as

$$\ln L^m(\theta|\{t_k^n\}_{n=1,2,...M}) = -\int_0 \lambda_\theta^m(t|\omega)dt + \int_0^T \ln \lambda_\theta^m(t|\omega)dN^m(t) \tag{3.13}$$

$$= -\mu_m T - \sum_{n=1}^{M} \frac{\alpha_{mn}}{\beta_{mn}} \sum_{\{k:t_k^n<T\}} \left[1 - e^{-\beta_{mn}(T-t_k^n)}\right] + \sum_{k:t_k^m<T} \ln \left[\mu_m + \sum_{n=1}^{M} \alpha_{mn} R_{mn}(k)\right] \tag{3.14}$$

with $R_{mn}(k)$ defined recursively as

$$R_{mn}(k) = e^{-\beta_{mn}(t_k^m - t_{k-1}^m)} R_{mn}(k-1) + \sum_{\{i:t_{k-1}^m \leq t_i^n < t_k^m\}} e^{-\beta_{mn}(t_k^m - t_i^n)} \tag{3.15}$$

with the initial condition

$$R_{mn}(0) = 0 \tag{3.16}$$

3.3.2 SIMULATION OF THE MULTIVARIATE HAWKES PROCESS

To simulate the multivariate Hawkes process we are going to use a procedure proposed in [17]. Before going to the algorithm we are going to present the rejection sampling technique, given that the algorithm is based on this technique. We will explain the rejection sampling in one dimension, given that it is what we are going to use, but it can be easily extrapolated to many dimensions.

Rejection sampling

Assume that we have a one dimensional density function $P(x) = \frac{P^*(x)}{Z}$ that we do not know how to sample from it directly. Now imagine that we have another density function $Q(x)$ from which we do know how to sample and such that

$$cQ(x) > P^*(x) \forall x \tag{3.17}$$

with c being a constant.

Given these two functions, the rejection sampling works as follows:

1. Generate x from the distribution $Q(x)$.
2. Evaluate $cQ^*(x)$ and generate a uniformly distributed random variable u from the interval $[0, cQ^*(x)]$.
3. Evaluate $P^*(x)$ and compare it with the value of u. If $u \leq P(x)$ then x is accepted and you have finished. Otherwise, it is rejected and you have to start again.

This procedure can be understood very well from a geometrical point of view looking at figure 9. The proposed x in the first step and the u generated in the second one can be interpreted as a point (x, u) and because of how we have generated this point we can easily realize that it comes with uniform probability from the lightly shaded area underneath the curve $cQ^*(x)$. After that, in the third step we reject the point if it above the curve $P^*(x)$ so the accepted points (x, u) will be uniformly distributed in the heavily shaded area $P^*(x)$. This implies that the samples will be independent samples from $P(x)$, given that the distribution of x is proportional to $P^*(x)$ as we have shown.

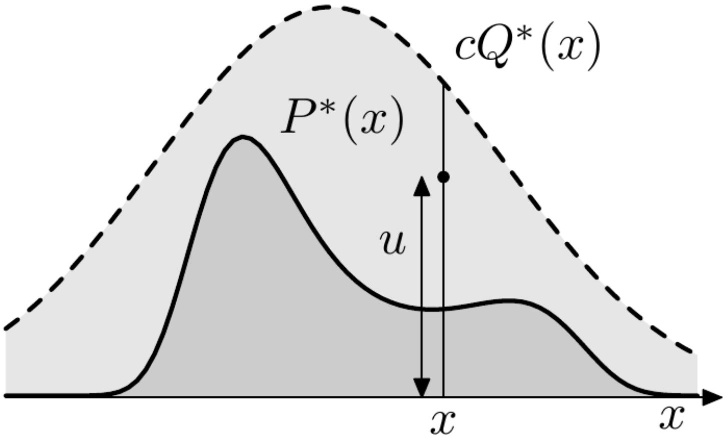


Figure 9: Illustration of a graphical interpretation of rejection sampling from the book [18].

The algorithm

The algorithm extracted from [17] works sampling inter-event times and it have two parts; in the first one, it is decided when the next event is going to be produced (an inter-event time is sampled) using the combined distribution of inter-event times for all of the event kinds. In the second part of the algorithm, it is decided the kind of this event.

For the first part of the algorithm, we know that if the intensity λ was constant, the inter-event time would follow an exponential distribution $P(x) = e^{-\mu t}$ (being μ the sum of all of the intensities μ^m). This is why the algorithm takes as a proposal density an exponential distribution $e^{-\bar{\mu}t}$, where $\bar{\mu}$ is this combined intensity just after the last event occurred, which is the moment when the intensities have its maximum value (until a new event is produced), so it is guaranteed that the real distribution will be always smaller.

After the rejection sampling have been applied, the next step is to decide the event kind and this decision is proportional to the intensity of each event kind at the decided time t.

These ideas are illustrated in figure 10, where we can see an example of simulating a bivariate Hawkes process with exponential decays.

Once we have understood the ideas behind the algorithm, we present the detailed performance:

1. Set $\mathcal{T}^1 = \mathcal{T}^2 = \cdots = \mathcal{T}^M = \emptyset$, $s = 0$ and $n^1 = n^2 = \cdots = n^M = 0$.
2. Repeat the following until s > T:
   (a) Set $\bar{\lambda} = \sum_{m=1}^{M} \lambda^m(s^+) = \sum_{m=1}^{M} \left(\mu_m + \sum_{n=1}^{M} \sum_{\tau \in \mathcal{T}^n} \alpha_{mn} e^{-\beta_{mn}(s-\tau)}\right)$
   (b) Generate u from a uniform distribution on $[0, 1]$.
   (c) Generate $\omega = -\ln u/\bar{\lambda}$ as the interarrival to the next candidate point.
   (d) Set the new candidate point $s = s + \omega$.
   (e) Generate D from a uniform distribution on $[0, 1]$.
      - If $D \leq \sum_{m=1}^{M} \lambda^m(s^+)/\bar{\lambda}$:
         - Find $k \in \{1, 2, ..., M\}$ such that $\sum_{m=1}^{k-1} \lambda^m(s) < D\bar{\lambda} \leq \sum_{m=1}^{k} \lambda^m(s)$.
         - Assign candidate point s to dimension k by setting $n^k = n^{k+1}$, $t^l_{n^k} = s$ and $\mathcal{T}^k = \mathcal{T}^k \cup \{t^k_{n^k}\}$.
      - Else, do nothing
3. If $t^k_{n^k} > T$, then subtract the last point $(\mathcal{T}^k_{n^k})$
4. $\mathcal{T}^m$ for $m = 1, 2, ..., M$ contain the simulated points.

We can see that in the algorithm, the criteria used to determine the duration of the algorithm is simply to stop when we have exceeded the maximum time T. Another option would be to fix the total number of events instead of the maximum time, so we would stop producing events when a given amount of events N was reached.

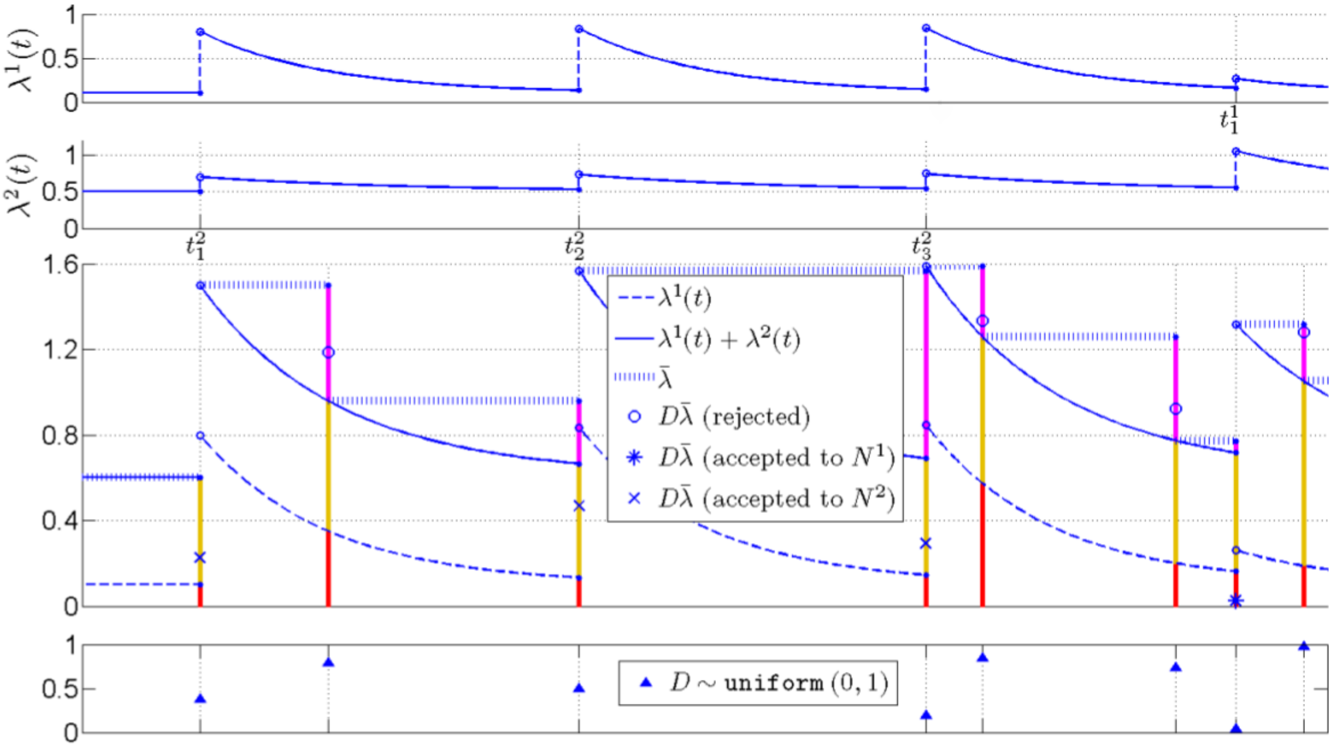


Figure 10: An illustrative example of simulating a bivariate Hawkes process with exponential decays where we can see the evolution of the intensities.

Chapter 4

HIERARCHICAL MULTIVARIATE HAWKES MODEL

Once we have seen the characteristics of our data and an overview of point processes that can be used to generate sequences of events, we now present a generative model for our data.

The first aspect of our data that we have to take into account seen from chapter 2 is that the patterns of activity (see figure 3) correspond to very different time scales, going from patterns associated to the different days of the week, to patterns that can be observed at much smaller length scales, more associated to the working sessions of the ILDE users.

Based on these results, our hypothesis is that the user activity can be basically decomposed in two processes, each of which acts on a different time scale. The first one, corresponding to the time scale of days, is the responsible of the beginning of different working sessions, i.e. when the user decides to start working during their everyday live. The second one corresponds to the behaviour of the user inside these working sessions.

To face this problem, we take a similar approach as in [9] where an hierarchical model is used to generate sequences of sent e-mails. In this project the authors use the already mentioned concept of **sessions**, which correspond to periods of time in which the user is using e-mail and so it is very likely that this user sends one or more e-mails. This simple model is composed of a first layer that decides when a session starts, which is governed by the following non-homogeneous Poisson process with periodic intensity:

$$\lambda(t) = N_w p_d(t) p_w(t), \tag{4.1}$$

where N_w is the average number of events (sent e-mails) during the week, and $p_d(t)$ and $p_w(t)$ are the daily and weekly distributions of session initiation.

The second layer in this model is governed by an homogeneous Poisson process and starts once a session starts. So, the sessions are filled with an homogeneous Poisson process with intensity λ_s with N_s events, with N_s drawn from some distribution $p(N_s)$. In figure 11, we can see an example of how this periodic distributions ($p_d(t)$ and $p_w(t)$) are combined to give us the resulting intensity, and a generated sample with this method under the obtained intensity.

To validate the accuracy of a model they use the area test statistic which is the difference between the cumulative distributions of both the simulated data and the real one.

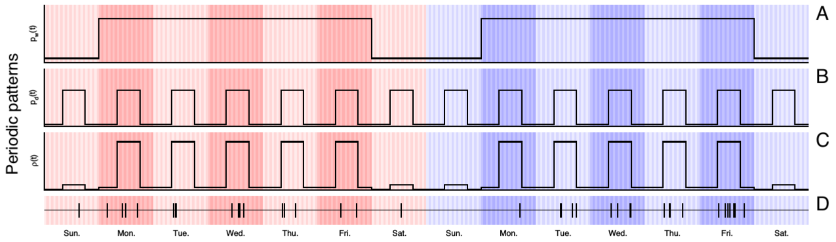


Figure 11: Image from [9]. Periodic weekly (A) and daily (B) session initiation distributions. In (C) we can see the resultant intensity λ_s (in the picture is called $\rho(t)$). In (D) we can see a sample time series.

Inspired in [9], our model is also a two-layer hierarchical model, with the main difference that, instead of using a homogeneous Poisson process at the second layer, we use a multivariate Hawkes process so it takes into account different kinds of event and interactions between them. Apart from that, our first layer is not exactly a point process as we will see in the next section.

The reason why we have used an structure similar to the one used in [9] in our model is that the idea of "sessions" fits very well our data, given that it is the normal way of using a platform such as the ILDE. However, the model of Malmgren is too limited for us, since in our case a simple Poisson model is not sufficient to capture the temporal structure of events within a session, as we will show later. For instance, we have observed that it is very usual that, inside a session, when the user makes one visit, it tends to do more visits in the following minutes. This behaviour is logic if we imagine a user working in his project that makes a break and starts looking at the designs of other users. This kind of self-exciting behaviour is reproducible with a Hawkes process but not with a Poisson one.

In the following sections we are going to explain in more detail this hierarchical Hawkes model, from its structure, implementation and training to its validation and applications.

4.1 STRUCTURE OF THE MODEL

As we have said, our generative model is a hierarchical model with two layers. In figure 12 we have a very simple flow chart of the model. In the first layer, it is decided the beginning of the different sessions and its duration. In the second layer, the sessions are filled with the events. With this two mechanisms the sequences of events during the desired period of time (e.g. one month) are drawn. This procedure can be performed for each user independently, what we call **individualized model**, or for the whole platform, what we call **combined model**. We have seen that both approaches have good performances and can be useful in different situations but in the validation and applications sections we are going to focus on the second one for simplicity.

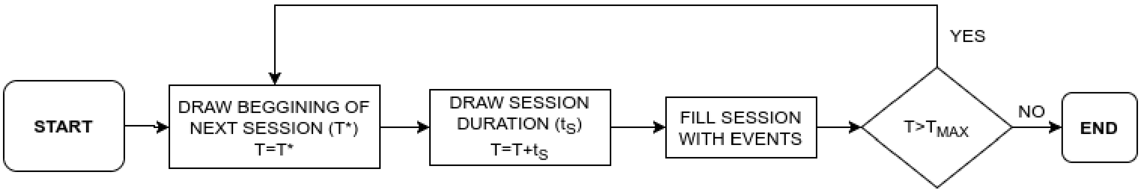

START
DRAW BEGGINING OF NEXT SESSION (T*) T=T*
DRAW SESSION DURATION (t_S) T=T+t_S
FILL SESSION WITH EVENTS
T>T_{MAX}
YES
NO
END

Figure 12: Flow chart of the model where T is the current time, and T_{MAX} is the period of time that we want to simulate.

4.1.1 SESSION DISTRIBUTION

First of all, it is important to remark that despite we are using this "session" framework, there is no easy and direct way to split our data into sessions. In [9], this split into sessions is treated as another feature to be optimized, which makes a lot of sense because the e-mail data is very spread and with few events per session, which makes difficult to identify sessions. However, given that in our data the working sessions contain a lot of events, it is relatively simple (in most of the cases) to detect sessions by searching conglomerations of events in the time-line. This is why we have opted for a simpler approach; we simply consider that two consecutive events belong to the same session if the time between them is smaller than one hour. This amount of time is based in the figure 3, where we saw that the peak corresponding to inter-session events starts losing a lot of importance in the time from 1-2 hours on. A part from that, we think that it is reasonable to think that after one hour of not viewing, not creating, not editing and not commenting, this user have finished working, and in the worst case (that will happen few times) we will be simply splitting a session with a long period without events into two, so our multivariate Hawkes process will not be taking into account the influence that the events of the first half of session have in the second half of session. The other thing to take into account is that the concept of session in real life is not always what we could think in theory; one can stop working to do any other thing such as watch the TV or a YouTube video, do the laundry, do some other work, etc. to continue working after that in the "same" working session, which introduce an unavoidable noise in our model.

Once we have the data of each user split into sessions, for the individualized model we perform the procedure that we are going to explain for each user independently and for the combined model, we put all of the sessions together and consider it as if it was a single user.

To perform the session distribution, in contrast with [9], where a non-homogeneous Poisson Process is used, we have decided to use a binomial process, where at each hour of the week we have a given probability for a session to be produced. These two processes are not very different and in fact a binomial distribution tends to the Poisson one when the number of time windows tend to ∞ (i.e. the length of the time intervals tend to 0) and of course the probability of an event being produced in each of these intervals tend to zero. So, the Poisson process can be understood as a binomial process made continuous.

The main difference between the Poisson process and the binomial one is that in the second one, in each of the time windows only one event can occur, while for the Poisson process there is no limit in number of events per interval of time. So, in our case, the difference between the two processes is that for the binomial one, there will be the restriction that there can not be more than one session per hour, but it make sense with our data. Furthermore, as we have seen, when we want to divide the real data into sessions, what we do is to say that two consecutive events belong to the same session if the time between them is smaller than one hour, so there is implicitly a similar constraint saying that the time between sessions is always larger than one hour.

To determine the probability of a session being produced inside an hour during the week (P_h), we use a similar procedure to the one reflected in the expression 4.1, which is:

1. Compute the empirical activity distribution during the hours of the day P_D, as we have seen in figure 6 (array of 24 values, one per hour).

2. Compute P_W the empirical activity distribution during the days of the week, as we have seen in figure 7 (array of 7 values, one per day).

3. Combine both distribution to create a new distribution P_h that is an array of 24*7=168 values of the form
$[P_D^1 P_W^1, P_D^2 P_W^1, P_D^3 P_W^1 ..., P_D^1 P_W^2, P_D^2 P_W^2, P_D^3 P_W^2, ...P_D^{22} P_W^7, P_D^{23} P_W^7, P_D^{24} P_W^7]$
After that, we normalize this array to sum one and multiply it for the average number of sessions per week. At figure 13 we can see an example of distribution P_h.

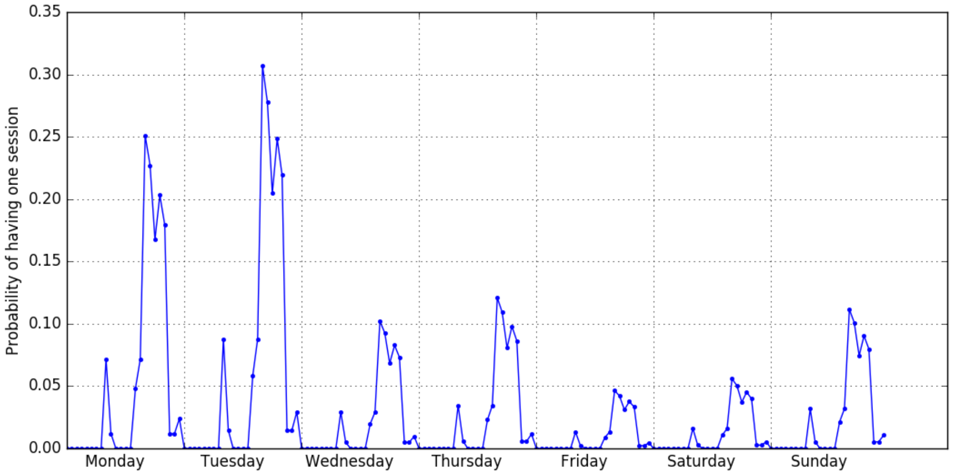


Figure 13: Example of normalized weekly activity distribution used in the Hierarchical Hawkes Model for a user.

So, we decide if one hour have a session or not according to P_h. When we have to put one session into an hour, we homogeneously choose a random time inside the hour to start the session. After that, we decide the duration of that session and fill it with events (this step is explained in the next part of the project). It can happen that one session starts in one hour and ends in another. If this happens, the next step is to go directly to this other hour h_i and repeat the procedure but now the part of the session susceptible to have a new session beginning is just the part that is not in the last session. Furthermore, the probability of having a session in this portion of hour (Δh) is modified according to the length of this portion $\left(P_h^*(h_i) = P_h(h_i)\frac{|\Delta h|}{|h|}\right)$

Another important feature of this procedure is that we are implicitly considering that the session probability distribution during the week is the same every week, so the amount of weekly activity is homogeneous. Apart from that, our model also assumes that the shape of the daily distribution is always the same and the only difference from one day to another is a multiplicative factor, which of course is generally not true. For instance, one user could work always the Monday mornings and the Friday afternoons, and our model would model it as if he worked both Mondays and Fridays during the morning and the afternoon. Despite this, we think that our procedure offers a good trade-off between model complexity and well representation of the data.

4.1.2 EVENT DISTRIBUTION INSIDE THE SESSIONS

In order to make the model able to reproduce different kind of events and interactions between them, the interior of the sessions are filled with a multivariate Hawkes process.

First of all, the duration of the session is chosen. To do so, we can choose if we want to fix the number of events inside the session or the total time of duration of that session. In both options, we draw the used value from their empirical distribution.

Among these two options to determine the duration of the session, using the distribution of number of events is much more precise, given that for the empirical distribution of times, it is impossible to know the exact duration of the sessions from our data. The only thing that we can know are the times when the first and the last events were produced. What we do is to take this time between the first and the last event as the session duration but of course it have a bias making these recorded times shorter than what they really are. Despite this, fixing the times is more natural and for most proposes it is more convenient. It is because if we want to test hypothetical situations by modifying some parameters, if we fix the number of events it will be impossible to determine the changes in terms of number of events, which is one of the most interesting features. For these reasons, despite using the number of events to determine the session length is more accurate and give better fit to the real data, we will generally use the time, that as we will see in section 4.3, give also quite good results.

Once we have determined the session duration, we do the simulation using the algorithm for simulating a multivariate Hawkes process that have been described in the section 3.3.2. Figure 14 shows a simulated session of 30 minutes for the couple of events views-edits and the corresponding intensities during the session.

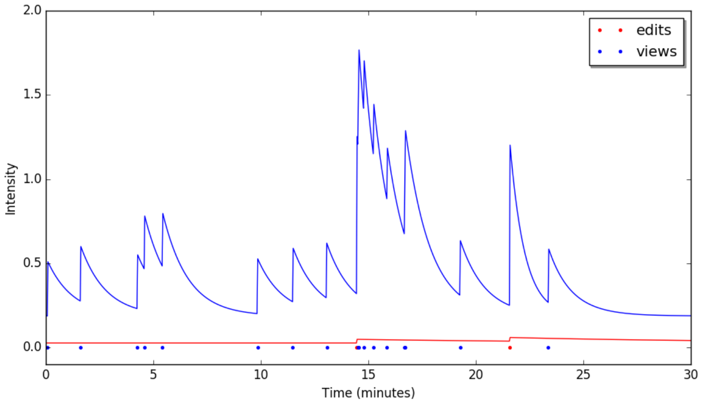


Figure 14: Example of 30 minutes session for the pair of events views-edits simulated using our model.

4.2 TRAINING OF THE PARAMETERS

The parameters that we have to train are the ones from the multivariate Hawkes process. For two kind of events, this represent a total of 10 parameters which are μ^1, μ^2, α^{11}, α^{21}, α^{21}, α^{22}, β^{11}, β^{12}, β^{21}, and β^{22}.

To train the parameters of the Hawkes process, we use the Nelder-Mead optimizer to minimize the negative log-likelihood (section 3.3.1) given the real sessions.

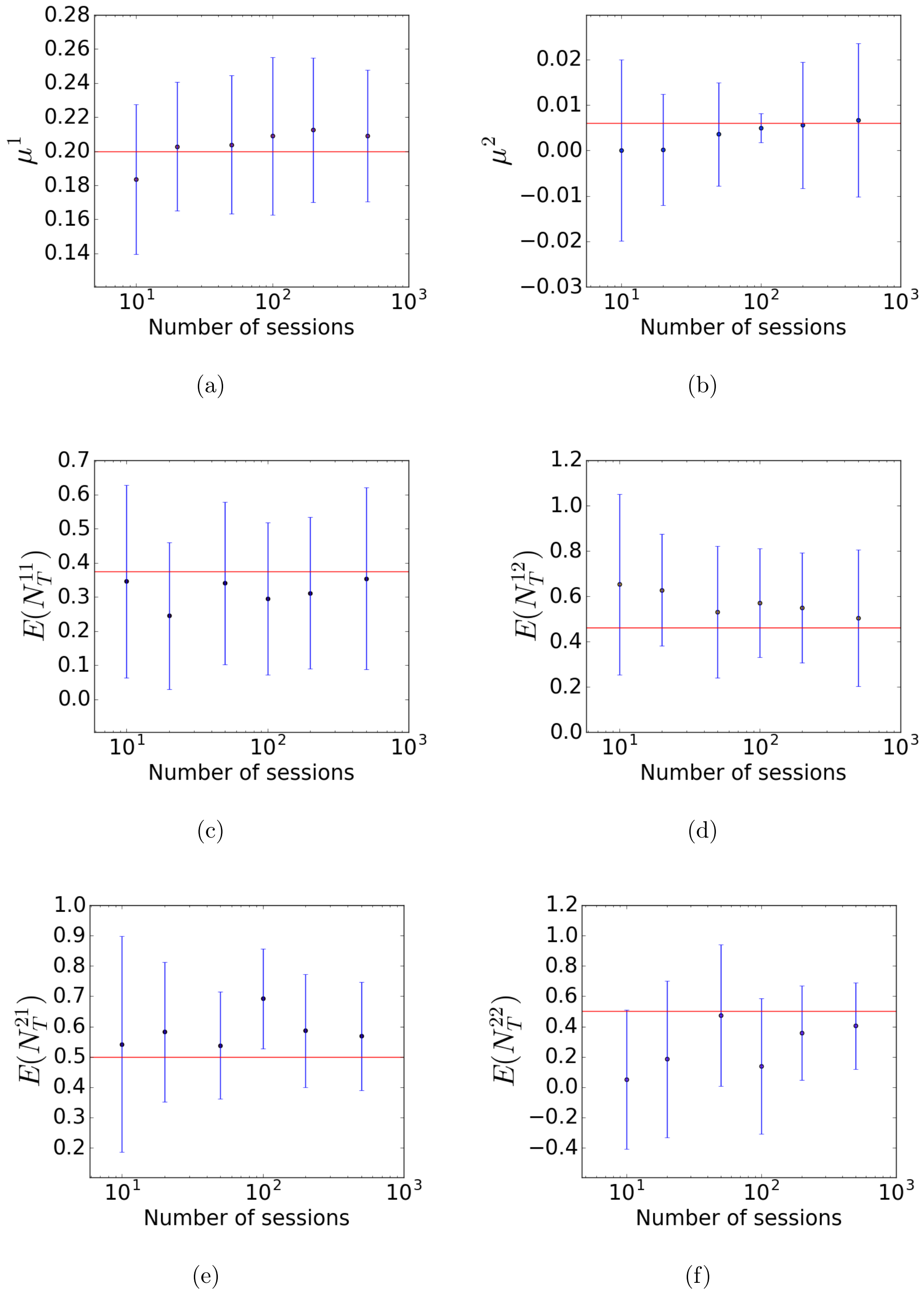
Because of our data is limited, it is important to study the amount of data needed to have guarantees that we will find a good approximation of the parameters. To do so, we have generated groups of 10, 20, 50, 100, 200, and 500 synthetic sessions of

10 minutes of duration each one (it is more or less the average session length) with known parameters similar to the ones found with the real data, and we have tried to train the parameters to see the dependence between accuracy number of sessions.

We have realized that the accuracy for the α^{mn}s and β^{mn}s is quite bad, which is reasonable because the pairs of parameters (α^{mn},β^{mn}) are very dependent and it is easy to obtain similar performances with very different combinations of these two parameters. E.g. ($\alpha^{mn} = 0, \beta^{mn} = 1$) would produce almost the same result as ($\alpha^{mn} = 2, \beta^{mn} = 10^4$), given that in practice, it is very similar a null impulse with a long extinguish time than a high impulse with a very small extinguish time. To solve this problem, instead of analyzing these parameters we have analyzed the $\mathbf{E}(N_T^{mn})$ (see equation 3.11), which relates the previous parameters and represents the average amount of events of kind m produced in a session as a result of an event of kind n.

In figure 15 we can see the average results of this experiment and its deviation after repeating the experience 50 times for each session length. The horizontal red lines are the values of the real parameters. We can see that, unexpectedly, the standard deviation is quite constant among the different results and the accuracy does not improve very much neither. Despite this, we can see that the results are quite good. For the μ^1 and μ^2 parameters the results are very close, with a standard deviation that never exceeds 0.06 for the μ^1 and 0.03 for the μ^2. For the $\mathbf{E}(N_T^{mn})$, the noise is much larger but the average results are still quite good, even with few sessions.

Having seen how the parameters are trained using ideal sessions from real Hawkes processes, we can go to the real data. For the real data we repeat the training 20 times and take the execution with the minimum negative log-likelihood. It is done because the likelihood function is expected to have local minimums so we want to start the search randomly at different initial configurations to improve our search. Apart from that, doing this repeated search we can also see whether there are many local minimums or not and analyze if there is a clear optimal combination of parameters with a much better negative log-likelihood than the others or if there are different combinations of parameters that represents well the data.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 15: Results obtained for different parameters in the log-likelihood minimization procedure as a function of the number of synthetic sessions used. For each selected number of synthetic sessions the experiment have been repeated 50 times. The solid points correspond to the mean values after all of the executions, the bars are their standard deviation and the red lines correspond to the real value of the parameters i.e. the parameters used to draw the synthetic sessions.

In the table 2 we have the results obtained for the combined model for the couples of events views-edits and edits-comments. We can realize that as we have explained for the ideal scenario, the standard deviations for the α^{mn}s and β^{mn}s are very high, which indicates that there are different combinations of these parameters that represents well the data. Despite this, for the μ^{m}s and the $\mathbf{E}(N_T^{mn})$s the standard deviation is very small, meaning that our model actually converges to a clear minimum. It is also supported for the log-likelihood values, where we see that the standard deviation is a 0.5% and a 1.3% of the total log-likelihood for the views-edits and the edits-comments combinations respectively.

	Views-Edits			Edits-Comments		
	Mean	Median	Std	Mean	Median	Std
α^{00}	0.694	0.011	2.041	0.015	0.02	0.008
α^{01}	0.001	0.0	0.002	0.603	0.0	2.022
α^{10}	2.227	2.684	1.155	0.005	0.0	0.013
α^{11}	0.311	0.31	0.009	144.087	0.038	627.892
β^{00}	35.106	6.794	67.223	11.704	0.113	41.998
β^{01}	10.207	0.84	17.229	97.429	11.36	250.734
β^{10}	4.826	5.137	1.852	410.982	2.418	1731.647
β^{11}	0.813	0.812	0.024	4544.302	1.071	19680.757
μ^{1}	0.03	0.032	0.003	0.042	0.041	0.002
μ^{2}	0.191	0.189	0.005	0.012	0.012	0.002
$\mathbf{E}(N_T^{00})$	0.084	0.001	0.149	0.142	0.173	0.062
$\mathbf{E}(N_T^{01})$	0.001	0.0	0.002	0.016	0.0	0.061
$\mathbf{E}(N_T^{10})$	0.44	0.522	0.195	0.0	0.0	0.001
$\mathbf{E}(N^{11})_T$	0.383	0.382	0.003	0.24	0.213	0.237
Log-Likelihood	54979.876	54997.811	290.939	14994.565	15036.235	190.123

Table 2: Results obtained in the combined model for the couples of events Views-Edits and Edits-Comments.

4.3 VALIDATION

In this section we will test our model and see how well it captures the data. We will focus on testing the combined model because it offers and overview of the behaviour of the users and the parameters are easily interpretable.

A part from that, for the analysis done in this section we have chosen the couple of events views-edits and the community MOOC2, because the results for the other events and other communities are similar (with an exception for the demo community that we will tackle later) and analyzing all of the possible combinations of events and communities would be unnecessary extensive.

In order to test and evaluate our model, we will compare the distribution of inter-event times obtained with our model with the inter-event time distributions from:

- The real data as it is done in [9] to see how close is our model to the reality.
- The results obtained using a Hierarchical Poisson Model (HPM), that is a version of our model where the events inside the sessions are drawn using a Poisson process. The structure of this model is very similar to the one used in [9].
- The results obtained using a Hierarchical Hawkes Model (HHM), which is a version of our model where inside the sessions we use a simple Hawkes Process instead of a multivariate one. I.e., there is no interaction between events of different kind.

We are going to analyze both the PDF (Probability Distribution Function) and the CDF (Cumulative Distribution Function) with both linear and logarithmic y-axis each one, in order to appreciate better the differences.

Figure 4.3 shows the results for the views. In the picture 4.16(a), we can see the PDFs. We see that that the two peaks of the real distribution are well represented in our model, specially the one associated to the time-scale of days, where we can see the sub-peaks corresponding to one day, two days, etc. The HHM fits the true distribution as good as the multivariate version, which is remarcable given that it uses much less parameters. Despite this, the motivation for using a multivariate Hawkes process instead of its single version is not just the increase in accuracy but also the possibility of interpreting the parameters that tell us how does the events of different kinds interact between them. The HPM also encodes the two peaks quite well but it is clearly less precise, with a high overestimation in the first peak.

We can also observe that the region where the fit is worst for the three models is between this two peaks, where our simulation underestimates the reality. We can also observe that the data is quite noisy, and nor the real neither the simulated data have a smooth distribution. It causes that different executions of the distribution give slightly different results, but always with a similar shape.

In the picture 4.16(b) we can see the corresponding CDFs, that have less noise and the shapes can be better appreciated. We see again that the fits of the Hawkes models are quite good with the little deviation between 10 and 100 minutes. Here we can also appreciate that using a Hawkes process gives better results than a Poisson process, specially for small times.

Figure 17 shows the results for the edits, we see that for the edits the shape of both the PDFs and the CDFs are very different to the views ones, but conserving the bimodal distribution. We see again that the distribution is very well fit by our model and almost as well fit for its single version, and again the worst fit is at 100 minutes. We see that this kind of event is more noisy than the views which we think that is simply because there are much more views than edits. Despite this, the shape is well reflected for our model. The HPM also have a very good performance for the edits. The linear-log graph in picture 4.17(b) shows that our model is quite better than the HPM but this second one is also quite good. It might be because, as we can see in table 4, the edits are much less vulnerable to interactions than the views, making them more similar to a Poisson Process.

Another important feature about these simulations that does not appear in the graphs is that despite we do not explicitly fix the number of events, the amount of simulated events is very close to the amount of real events. For the views, the number of real events is 15544 and the number of simulated ones is 16102, a difference smaller than a 4%. For the edits, the number of real events is 1660 and the number of simulated ones is 1618, which is less than a 1%.

Despite we have seen that in general our model fits the data better than the HPM, as we have said before the motivation for using our model is not just this improvement but also that taking into account the interactions makes our model much more interesting. It gives us the possibility to study and interpret the interaction parameters and also allow us to make predictions and inferences about hypothetical situations where these interactions play a key role such as how the increase in the frequency of one particular event affect other events. An example of how this characteristic can be exploited can be sen in the example of application of the section 4.4.

While the results for MOOC1 and MOOC2 are quite similar and represent well the data, the demo present much worst fits. We think that it is mainly due to the fact that the demo have been used for different purposes and also that in the demo the assumption that the amount of weekly activity during the week is constant is not true, given that the duration of the demo is very long and the activity of many users covers small portions of time.

4.4 APPLICATIONS

In this section we will apply our model to some specific problems to see how can it be used for different proposes. In concrete, we present two different problems and show how our model can tackle them.

4.4.1 Encouraging the usage of new tools

The objective of the MOOCs is to teach the users how to create learning designs using different tools from the ILDE. To do so, there are directed activities where the users are guided through the usage of a specific tool. Of course, these activities do not cover the totality of the tools but only a small fraction of them, and it is expected to be the users the ones that investigate to discover and try new tools that suit their necessities.

In this first application example we want to find out possibilities to encourage the users to try new tools by themselves, because it would make the MOOCs much more profitable for them. To make this analysis we will use the data from the MOOC2.

We have identified two possible strategies that determine the usage of new tools:

- Make the users view the profile of other users (as an activity of the MOOC for example) because seeing the designs of other users could encourage them to try the tools used for these other users.
- Make the users use more tools. Because making them use new tools will make them curious and comfortable to try more different tools. Realize that there is a difference between the case when we make the users to use more designs (throw directed activities) and the case when the users try new designs by themselves, and that the second case indicates that the users are using the ILDE properly and autonomously.

So, the question that we have to answer is: What will encourage more the users to try new tools by themselves, making them view more designs of other people or making them do more activities using new tools?

To answer this question we first have to see which events we need to analyze. The first event are of course the views, and the second event are the new tools, which are a subset of the edits that correspond to every time that a user creates a design using a tool that he have never used before.

The second step is to translate the question that we want to answer to our model. To do so we have to realize that the increases of events that are imposed by us (through directed activities) can be understood as an increase in the base rate (this relation is not bidirectional). So, to analyze the effect of an increase in the number of views on the usage of new tools, we can check how the model responds to an increase of the base rate.

For the case when we impose the usage of more new tools, the analysis is more delicate given that increasing the base rate of the new tools also increase the number of new used tools. Despite this, because of the relation between the base rate and the number of events is linear, we can just subtract to the resultant number of simulated events, the expected increase of events caused by the increase in the base rate, which is simply $\Delta\mu_0^{NewTools} * N^*$, where N^* is the number of simulated events with the base rates not modified.

	AVERAGE	STANDARD DEVIATION	AVG. INCREASE
Original Model	143.8	2.3	0 (0%)
$\Delta\mu_0^{Views}$ increased a 30%	146.9	1.2	3.1 (2.2%)
$\Delta\mu_0^{NewTools}$ increased a 30%	160.7 (203.9)	3.2	16,9 (9.4%)

Table 3: Results of different simulations for the pair of events Views-New Tools under different base rate modifications. The new tools average in parenthesis is the result before subtracting the expected number of new tool events caused by the increase in the new tools base rate.

In the experiment performed to make the analysis we have done several simulations in each of three scenarios which are the original model, the original model with the Views intensity increased a 30%, and the original model with the New Tools intensity increased a 30%. For each scenario we have performed 100 simulations and we have studied the average results to reduce the noise.

In table 3 we have a summary of the results of the experiment. We see that in this case the best strategy is to make activities that makes the users use new tools given that it produce the greater impact. On the other hand, increasing the views seems to have very few impact on the usage of new tools, given that the very small impact observed (2%) is covered for the standard deviation.

In this analysis, we have also considered both interventions independently, but we could also explore scenarios where both events are stimulated. It is also important to say that this analysis is advisory given that there are other factors that our model do not take into account and can have impact.

4.4.2 Comparison between platforms

In chapter 2 we have seen many differences between the different platforms. One way to understand what is behind these differences is to analyze the trained parameters of our model for each platform. In this application example we are going to analyze the parameters to extract some conclusions.

The first feature that we have analyzed are the base rates in the different platforms. We have realized that the results for both MOOCs are in general quite close and they differ a little more to the ones from the demo. To show this we have calculated the average difference between the base rates obtained in the three platforms and we have found that for the MOOCs the average base rate difference is 0.007 (i.e. the average difference in expected number of events is 0.007 events per minute, which is the same as 0.007*60=0.42 events per hour), that is very small if we compare it with the magnitude of the base rates. We analyze it using the total difference instead of the percentage of variation due to the fact that there are base rates with values very close to zero where it is easy to find very large percentage differences that in fact are very similar (e.g. 10^{-4} is a 900% larger than 10^{-5}). The differences between the MOOCs and the demo are more different but still not much. In concrete between the MOOC1 and the demo there is an average difference of 0.077 and between the MOOC2 and the demo we have an average difference of 0.076.

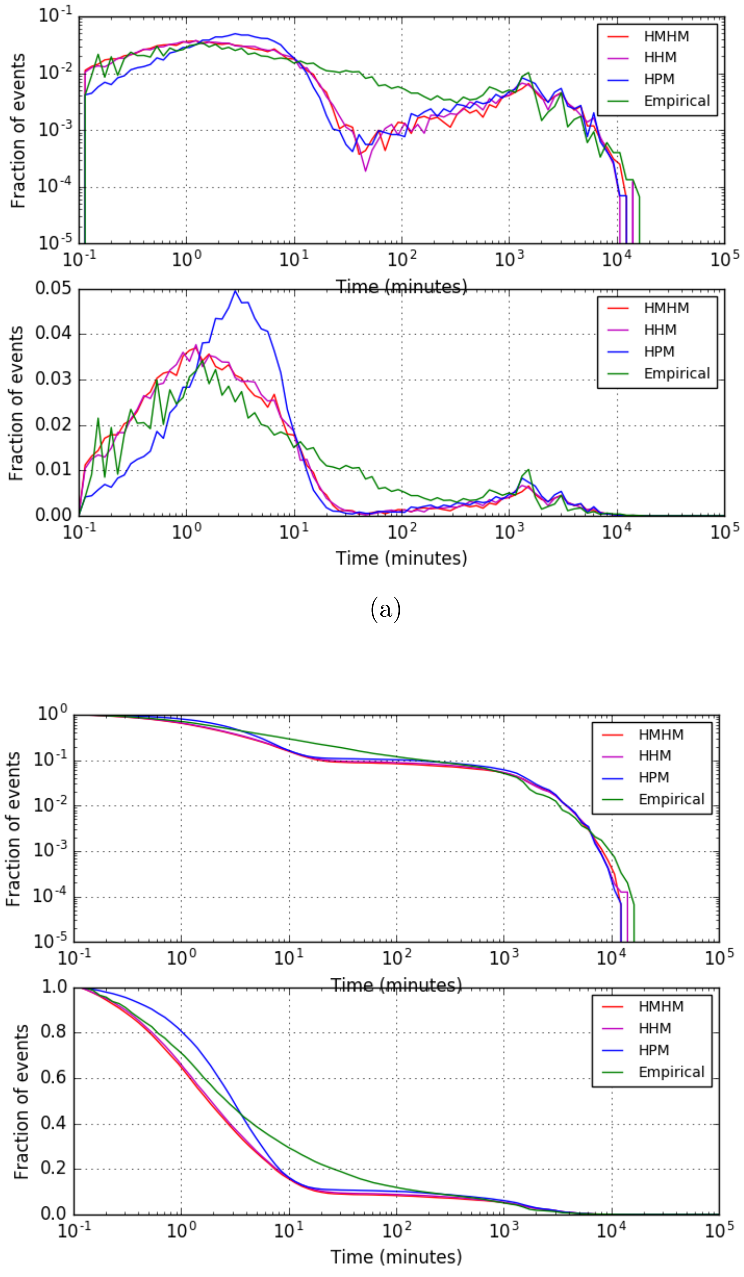
Table 2 also shows that the largest base rate differences are the ones for the comments, where the base rates are one order of magnitude larger in the demo than in the MOOCs.

If we look at the values for the average number of events of kind m produced due to the appearance of an event of kind n , $\mathbf{E}N_T^{mn}$, we also see that the results found in the MOOCs are generally closer between them than to the demo's ones. Despite this the differences for these quantities are larger than the ones seen for the base rates. For the MOOCs, we have an average difference of 0.072, which is one order of magnitude larger than the base rate difference but still not a very high difference and the difference between the demo and the MOOCs 1 and 2 are 0.198 and 0.214 respectively.

We can observe that one of the largest values of $\mathbf{E}N_T^{mn}$ are for $m = n =$views, where we find values around $\mathbf{E}N_T^{Views-Views} = 0.4$. Despite this, we can see that the $\beta^{Views-Views}$ is close to 0.8, which means that the time in which the increase of intensity decreases a factor $1/e$ is $1/\beta^{Views-Views} = 1/0.8 = 1.2$ minutes so the interactions between views is quite short.

Another very high value of $\mathbf{E}N_T^{mn}$ is for $m =$views and n=edits (and n=new tools), specially in the MOOCs which indicates that after editing the users tend to visit designs of other user which make sense as a "break" after working. In this case, the value for $\beta^{Views-Edits}$ is much larger than for the views (near to 5), indicating that view other designs just after the edition $(1/5\beta^{Views-Views} = 0.25$ minutes $= 15$ seconds).

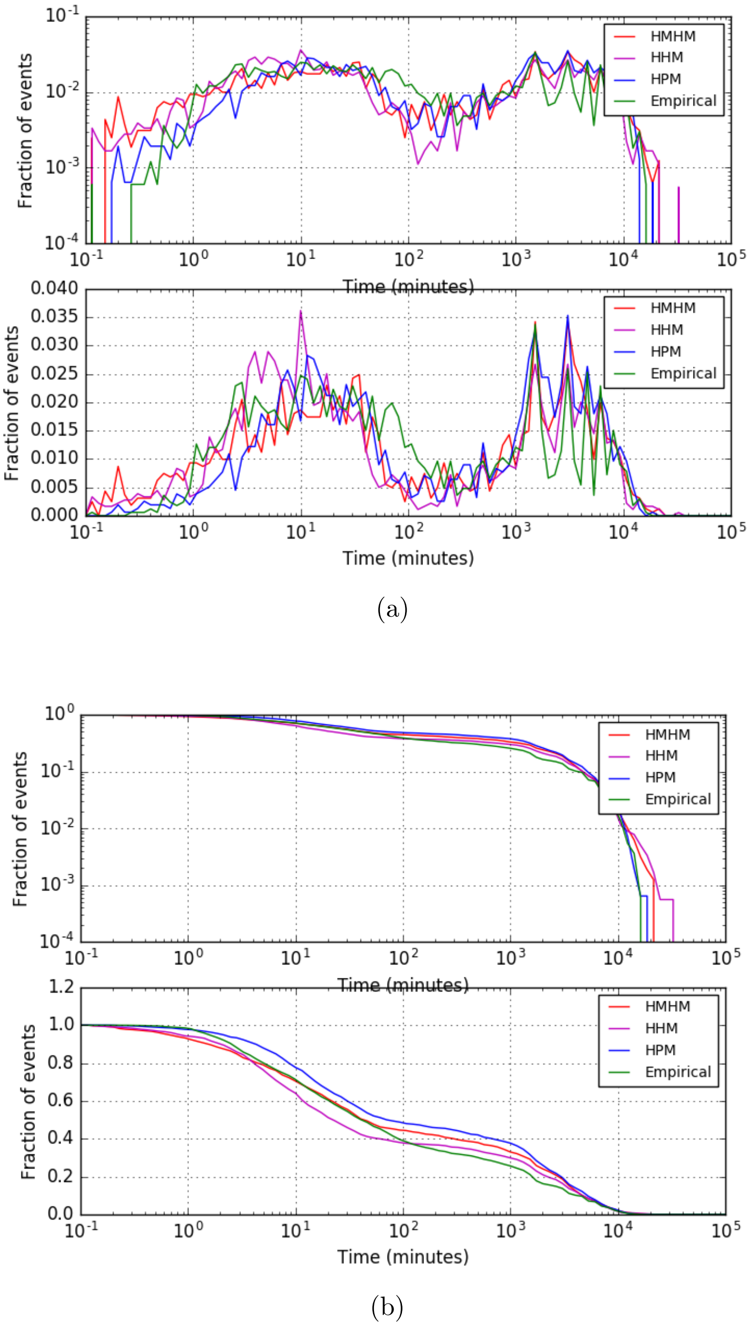
Finally, we can observe some anomalies or unexpected values that differ a lot from one platform to another. For example, we can observe that $\mathbf{E}N_T^{Views-NewTool} \simeq 0.6$ for the MOOCs and 0.06 in the demo, which indicates that in the MOOCs the users tend to view designs after using a new tool (maybe to get ideas) much more than in the demo. We can also see many anomalies related with the kind of event "comments" that it is difficult to give them an explanation and maybe there are just due to the fact that there generally are few comments so it might be a lot of noise associated to them. For instance $\mathbf{E}N_T^{Comments-Views} = 0.911$ in the MOOC1 and it is equal to 0 in the demo and in the MOOC2, which seems to indicate that in the MOOC1 the views are very inactivated by comments but it does not happen in MOOC2 and demo.



(a)

(b)

Figure 16: PDFs (a) and CDFs (b) of the of the inter-event times obtained from the Hierarchical Multivariate Hawkes Model (HMHM), the Hierarchical Poisson Process (HPP) and the real empirical distribution for the views in a Views-Edits simulation. The data is presented in log-log and log-linear scale.



(a)

(b)

Figure 17: PDFs (a) and CDFs (b) of the of the inter-event times obtained from the Hierarchical Multivariate Hawkes Model (HMHM), the Hierarchical Poisson Process (HPP) and the real empirical distribution for the edits in a Views-Edits simulation. The data is presented in log-log and log-linear scale.

		μ_0^1	μ_0^2	$\mathbf{E}N_T^{11}$	$\mathbf{E}N_T^{12}$	$\mathbf{E}N_T^{21}$	$\mathbf{E}N_T^{22}$	α^{11}	$\alpha12$	α^{21}	α^{22}	β^{11}	β^{12}	β^{21}	β^{22}
Views/ Edits	MOOC1	0,183	0,045	0,401	0,511	0,002	0,030	0,365	3,355	0,002	0,028	0,911	6,564	0,999	0,154
	MOOC2	0,188	0,029	0,388	0,522	0,000	0,084	0,299	2,652	0,000	0,065	0,772	5,079	27,136	0,579
	Demo	0,209	0,055	0,484	0,383	0,001	0,151	0,404	1,455	0,000	0,126	0,835	3,797	0,396	0,305
Views/ Comm.	MOOC1	0,200	0,003	0,397	0,911	0,000	0,046	0,353	1,395	0,000	0,046	0,891	1,531	11,603	0,079
	MOOC2	0,203	0,007	0,382	0,000	0,000	0,037	0,305	0,000	0,000	0,036	0,799	0,651	0,011	0,056
	Demo	0,251	0,251	0,462	0,000	0,000	0,462	0,438	0,000	0,000	0,438	0,948	0,006	0,099	0,948
Edits/ Comm.	MOOC1	0,064	0,004	0,000	0,001	0,000	0,001	0,000	0,013	0,000	0,047	34,557	12,352	1,237	0,086
	MOOC2	0,046	0,010	0,000	0,003	0,000	0,006	0,000	0,670	0,000	0,042	6,924	195,121	0,120	0,084
	Demo	0,058	0,225	0,363	0,002	0,305	1,384	0,105	0,013	1,318	0,398	0,288	7,785	4,316	0,881
Views/ N. Tools	MOOC1	0,200	0,006	0,402	0,621	0,000	0,001	0,354	0,594	0,000	0,001	0,881	0,957	0,036	52,007
	MOOC2	0,201	0,003	0,380	0,665	0,000	0,049	0,307	2,161	0,000	0,039	0,808	3,250	10,809	0,248
	Demo	0,241	0,014	0,481	0,063	0,000	0,044	0,419	0,132	0,000	0,038	0,872	2,083	0,994	0,701
Edits/ N. Tools	MOOC1	0,066	0,008	0,000	0,098	0,000	0,000	0,000	0,288	0,000	0,000	9,212	2,934	6,410	2,551
	MOOC2	0,053	0,006	0,000	0,139	0,000	0,006	0,000	0,477	0,000	0,018	2,831	3,440	0,350	0,237
	Demo	0,116	0,024	0,000	0,250	0,000	0,000	0,000	11,908	0,000	0,000	0,131	47,575	0,021	0,006
Comm./ N. Tools	MOOC1	0,012	0,018	0,415	0,000	0,001	0,001	0,059	0,000	0,003	0,000	0,140	0,766	3,835	0,958
	MOOC2	0,024	0,010	0,237	0,000	0,013	0,357	0,036	0,000	0,169	0,053	0,149	0,040	12,989	1,031
	Demo	0,239	0,014	0,478	0,467	0,000	0,000	0,426	0,487	0,000	0,002	0,891	1,043	0,040	0,000

Table 4: Trained parameters for different event pairs and different ILDE communities.

Chapter 5

CONCLUSIONS

In this project we have presented a Hierarchical Multivariate Hawkes Model, a generative model for the ILDE that works with a two layers procedure that first draws the beginning of working session and then fills this sessions with events. In section 4.3 we have seen that the model describes quite well the data, specially the data from MOOC1 and MOOC2, and that in the first layer, using a multivariate Hawkes process (or even the univariate one) instead of a simple Poisson Model actually improves the fit and also give us more information about the data. Between the Hierarchical Multivariate Hawkes Model and its univariate version we have seen that the differences are almost negligible so taking into account interactions between different kind of events does not increase the accuracy significantly, at least at our level of study.

In the project, we have focused on training and using the model with couples of two event kinds,because with two kinds we can show how can the model encode the interactions and simulate different event kinds but at the same time the number of parameters (N) of the model is not very large (grows as $N + 2N^2$) so the training procedure is feasible as we have seen in section 4.2. Despite this, the model can be used for as many event kinds as we want and if we have enough data to train the parameters, doing so should increase the accuracy of our model, given that the not used parameters act as hidden variables that our model not use but still interact with the used parameters.

Another way that we think that could significantly increase the accuracy of the model is allowing more kinds of interactions between couples of events, i.e., adding new interaction terms in equation 3.10 for each couple of events. If this terms are also exponential decays, the expression for the intensity is 3.9, where P are each of the different kind of interactions. Adding more interaction terms allow the model to encode more different kind of interactions that work at different time-scales. For instance, the events of kind m could influence the events of kind n in a very short time scale, increasing a lot the intensity in the following seconds after the event occurred but extinguishing that increase very fast, and also in a lager length scale, increasing the intensity a little amount but for a much longer period of time (e.g. 30 minutes). Of course we could also add other kind of interaction terms with more complex intensity increases but it would not be a Hawkes Model any more. Anyway, it is important to take into account that any increase in the number of interactions also increases the number of parameters. In the Multivariate Hawkes Model, increasing in one the number of interactions P, increases the number of parameters $2N^2$, so using 2 event kinds, using $P = 2$ instead of $P = 1$ would increase the number of parameters from 10 to 18, which is almost the double.

So, the main conclusion is that there are ways to improve the accuracy of the second layer of our model but they also have drawbacks related to the need to train more parameters, which with the amount of data that we have in this model is still difficult.

Another feature of our model that could be improved is the part when the length of the session is decided. As we have seen it is done using the empirical distribution but we think that in a future work it would be interesting to substitute the empirical distribution for some analytic approximation, to make it more interpretable and treatable. It would be an interesting improvement given that as we have seen in the applications section, this model is very useful to test hypothetical situations by modifying the parameters, and using a distribution function would allow us to modify the parameters of the length distribution function to study the effects of the modification.

About the fist layer, where we use a Poisson process with variable intensity (t) to draw the beginning of the sessions, we think that it would be interesting to try using a Hawkes process with variable base intensity $\lambda_0(t)$. It would be specially interesting for the demo community, where the amount of activity per week of the users is very variable. There, it makes a lot of sense using a Hawkes procedure given that the users have periods of time with high density of sessions and others with null activity. It does not happen in the MOOCs, where the activity is much more constant because they have a short duration ($\sim$ 1 month) full of directed activities so the users have to be regularly using the ILDE.

In conclusion, the model built works well and we have seen that can be used to understand better the data and test hypothetical situations very easily. It is remarkable the simplicity of the model, that allow very good representation of the data and at the same time is interpretable and easy to tune. Apart from that, the model is quite general and despite we have only used it in the ILDE context, it could be extrapolated to other domains with few modifications. The only condition for a domain to allow us to model it with our model is that it have to be possible to describe it throw discrete events, and this is not something unusual.

List of Figures

List of Tables

Bibliography

[1] Hernández-Leo, D., Asensio-Pérez, J. I., Derntl, M., Prieto, L. P. & Chacón, J. ILDE: Community environment for conceptualizing, authoring and deploying learning activities. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2014).

[2] METIS: Meeting teachers' co-design needs by means of integrated learning environments (2017). URL `http://www.metis-project.org/`.

[3] Gómez, V., Kaltenbrunner, A. & Laniado, D. The online conversation threads repository (2016). URL `http://repositori.upf.edu/handle/10230/26270`.

[4] Michos, K. & Hernández-Leo, D. *Understanding Collective Behavior of Learning Design Communities*, 614–617 (Springer International Publishing, Cham, 2016). URL `https://doi.org/10.1007/978-3-319-45153-4_75`.

[5] Garreta-Domingo, M., Sloep, P. B., Hérnandez-Leo, D. & Mor, Y. Design for collective intelligence: pop-up communities in MOOCs. *AI & SOCIETY* (2017). URL `https://doi.org/10.1007/s00146-017-0745-0`.

[6] Kaltenbrunner, A., Gómez, V. & López, V. Description and prediction of slashdot activity. In *Proceedings of the 5th Latin American Web Congress (LA-WEB 2007)* (IEEE Computer Society, Santiago de Chile, 2007).

[7] Kaltenbrunner, A., Gómez, V., Moghnieh, A., Meza, R., Blat, J. & López, V. Homogeneous temporal activity patterns in a large online communication

space. *International Journal on WWW/INTERNET* **6**, 61–76 (2008). 10th International Conference on Business Information Systems. Workshop on Social Aspects of the Web.

[8] Rybski, D., Buldyrev, S. V., Havlin, S. & Liljeros, F. Communication activity in a social network: relation between long-term correlations and inter-event clustering. *Nature* **2**, 1–8 (2012).

[9] Malmgren, R. D., Stouffer, D. B., Motter, A. E. & Amaral, L. A. A poissonian explanation for heavy tails in e-mail communication. *Proceedings of the National Academy of Sciences* **105**, 18153–18158 (2008).

[10] Daley, D. J. & Vere-Jones, D. *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods* (Springer, New York, 2003), second edn.

[11] Embrechts, P., Liniger, T. & Lin, L. Multivariate Hawkes processes: an application to financial data. *Journal of Applied Probability* **48**, 367–378 (2011).

[12] Laub, P. J., Taimre, T. & Pollett, P. K. Hawkes Processes. *arXiv:1507.02822,* (2015).

[13] Bacry, E. & Muzy, J.-F. Hawkes model for price and trades high-frequency dynamics. *Quantitative Finance* **14**, 1147–1166 (2014). URL `http://dx.doi.org/10.1080/14697688.2014.897000`.

[14] Ozaki, T. Maximum likelihood estimation of Hawkes' self-exciting point processes. *Annals of the Institute of Statistical Mathematics* **31**, 145–155 (1979). URL `https://doi.org/10.1007/BF02480272`.

[15] Liniger, T. *Multivariate Hawkes Processes.* Ph.D. thesis (2009).

[16] Chen, Y. Likelihood Function for Multivariate Hawkes Processes (2016). URL `http://www.math.fsu.edu/~ychen/research/HawkesLikelihood.pdf`.

[17] Chen, Y. Multivariate Hawkes Processes and Their Simulations (2016). URL `http://www.math.fsu.edu/~ychen/research/multiHawkes.pdf`.

[18] MacKay, D. J. C. *Information Theory, Inference, and Learning Algorithms* (Cambridge University Press, Cambridge, 2003), 4 edn.