CHANGEPOINT MODEL FOR BAYESIAN ONLINE FRAUD DETECTION IN CALL DATA

by

Hilal Tüysüz

B.S., Mathematics, Boğaziçi University, 2015

M.S., Industrial Engineering, Sabancı University, 2018

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Sabancı University
2018

CHANGEPOINT MODEL FOR BAYESIAN ONLINE FRAUD DETECTION IN CALL DATA

APPROVED BY:

Asst. Prof. Dr. Sinan Yıldırım (Thesis Supervisor) ....

Assoc. Prof. Dr. Semih Onur Sezer ....

Prof. Dr. Ali Taylan Cemgil ....

DATE OF APPROVAL: 31/07/2018

ACKNOWLEDGEMENTS

I would like to thank my family who always supported me with their unconditional love.

I would like to express my deepest gratitude to my thesis advisor, Sinan Yıldırım for his guidance and support through last year. I will always feel honored for having the opportunity of working with him.

I am grateful for my friends in Sabancı University especially Nozir, Veciye, Sinan, Sinem, Başak, Hazal and Umut. They always helped and encouraged me during hard times.

Lastly, I would like to thank NETAŞ for their support and ideas they shared which helped our research to improve.

I thank all the jury members for reading, listening and evaluating my thesis.

ABSTRACT

CHANGEPOINT MODEL FOR BAYESIAN ONLINE FRAUD DETECTION IN CALL DATA

Keywords: forward-backward recursions, Hidden Markov Model, online event-based fraud detection

Illegal use in the phone network is a massive problem for both telecommunication companies and their users. By gaining criminal access to customers' telephone, fraudsters make an illicit profit and cause heavy traffic in the call network. After rising trend in mobile phone fraud, telecommunication companies' security departments mainly focused on increasing the efficiency of fraud detection algorithms and decreasing the number of false alarms. In this thesis, we represent an online event-based fraud detection algorithm based on Hidden Markov Models (HMM). Detection problem is formulated as a changepoint model on caller's behavior. To capture call behavior more specifically, we split it into three parts; call frequency, call duration and call features. We prefer to adapt changepoint model for call data because of its memoryless property; the data before the changepoint does not depend on the data after the change point. To investigate the performance of our algorithm, we conducted an extensive computational study on our generated data. Our results indicate that the algorithm is practical and resampling methods can control the difficulty of linearly increasing computational cost.

ÖZET

DEĞİŞİM NOKTASI MODELİ KULLANARAK ARAMA VERİSİNDE GERÇEK ZAMANLI, BAYESÇİ TELEFON DOLANDIRICILIĞI TESPİTİ

Anahtar Kelimeler: ileri-geri yayılım algoritması, Saklı Markov Modelleri, gerçek zamanlı, olay esaslı dolandırıcılık tespiti

Telekomünikasyon ağlarındaki usulsüz kullanım hem arama şirketleri hem de kullanıcıları için büyük bir sorun. Müşterilerin telefonlarına yasadışı erişim sağlayarak, dolandırıcılar haksız bir gelir elde etmekte ve arama ağlarında yoğun trafiğe sebep olmaktadır. Cep telefonu dolandırıcılığında artan trendten sonra, telekomünikasyon şirketlerinin güvenlik departmanları dolandırıcılık yakalama algoritmalarının etkinliğini arttırmaya ve yanlış alarm sayısını azaltmaya odaklanmıştır. Bu tezde, gerçek zamanlı, olay esaslı ve saklı markov modellerine dayanan dolandırıcılık tespiti algoritması anlatıyoruz. Bu hata tespit problemi arayıcının davranışına odaklanan bir değişim noktası modeli olarak formüle edildi. Arayıcının davranışını daha iyi yansıtabilmek için, bu arama sıklığı, arama süresi ve arama özellikleri olarak üçe bölündü. Değişim noktası modelini tercih etmemizin sebebi de bunun belleksizlik olmasıydı; değişim noktasından önceki veri , değişim noktasından sonraki veriye bağlı değil. Algoritmamızın performansını test etmek için, kendi ürettiğimiz veride kapsamlı bir çalışma yapılmıştır. Sonuçlarımız algoritmamzın etkili olduğunu ve linear olarak artan hesaplama süresi budama metodlarıyla kontrol edilebilir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

CDR	Call Detail Record
CE	Call End
CP	Call Progress
CS	Call Start
DC	Day Change
HMM	Hidden Markov Model
LDA	Latent Dirichlet Allocation
MCM	Multiple Changepoint Model
NC	No Call
TF	Time Frame
WC	Week Change
WF	Week Frame

1. INTRODUCTION

In the Report to the Nations on Occupational Fraud and Abuse [ACFE, 2016], it is stated that a typical organization suffers loss up to 5% of its revenues in a given year as a consequence of fraud. The financial and telecommunication networks, government and public administrations, and credit card companies are the ones that suffer from criminal activity mostly. Phone fraud is unauthorized use of telecommunication services with the intent of gaining money from, or neglecting to pay, a telecommunication company or its users. Fraudsters with hacking skills can easily access phone accounts and cause considerable losses to both service providers and their customers. According to Global Fraud Loss Survey, telecommunication companies lose $38.1 billion in a year [CFCE, 2015]. There are many types of phone fraud, ranging from mobile phone theft to hacking to the communication network. [Becker et al., 2010] introduced a fraud type called *intrusion fraud* which is the case of victimization of a legitimate phone account by a fraudster who makes or sells calls to gain illegal money. The focus on protecting customer's privacy and finding ways to reduce revenue loss without sacrificing on service quality has made fraud detection a highly critical problem for communication companies. According to [H. Cahill et al., 2002], a suitable detection system must be event driven not time driven and can detect fraud for every account. In our work, we have focused on intrusion fraud and its real-time, event-based Bayesian detection using Multiple Changepoint Model (MCM).

Fraudsters usually move rapidly and cleverly in the network, which makes identifying fraud a tough task. One solution to this problem is constructing a neural network as a model based on the past behavior of a customer [Davey et al., 1996] [Moreau et al., 1997]. Whenever a phone call is completed, their algorithm creates a structure called call detail record (CDR) which includes call time, duration and receiving area, etc. They recorded CDR's for both creating a user profile and also comparing recent behavior with historical behavior to identify fraud. [Xing and Girolami, 2007] formulated a signature-based detection method called Latent Dirichlet Allocation (LDA). They used CDR to create call features, for example, the time of the call initiation, call duration, class of destination number, number of calls per day. Customers' behavior interpreted as a probability distribution over these call features. To detect whether a user making a call versus intruder making the same call, LDA compares the likelihood of a call being fraud to the likelihood of a call being standard.

In addition to neural networks, [Moreau et al., 1998] presented a rule-based approach. According to this approach, a call is considered illegal if it follows pre-determined rules created for the detection algorithm. A study by [Rosset et al., 1999] indicated these pre-determined rules could be established by past examples of normal and unauthorized usage in the network. Call details, total number or duration of calls over a specified period, and customer's price plan are essential in setting rules for a fraud case. We can deduce that both neural network and rule-based approach require training for customers' past data. Also, in the case of an unprecedented intrusion, the new call will not fit the set of rules based on the historical data and rule-based approach will fail to detect fraud.

Another way to identify the anomaly is presented in [Taniguchi et al., 1998] as a Bayesian networks method. Bayesian networks are a proper framework for handling uncertainty in fraud detection problem. Initially, Bayesian network model constructs an intuitive stochastic model for the behavior of the customer. Once the model is established, they estimated the probability of a phone account being victimized. [Taniguchi et al., 1998] says there is no deterministic approach to classify a call as a fraud. However, they formulated probability of fraud given the user's transactions in the phone network. The data they used was based on toll tickets which are created after a call is completed. Unlike in [Taniguchi et al., 1998], we desire to capture fraudsters at the time of the action, not after it. [Scott, 2004] stated that an intrusion detection system depending on stochastic models could be applied to many networks. Customers' accounts must be monitored in real time to catch intrusion quickly; an important challenge is to set a proper model that describes customer and intruder behavior. [Hollmén and Tresp, 1998] proposed a real-time detection system for phone fraud which is based on a stochastic model like in [Scott, 2004]. They introduced a Bayesian hierarchical regime-switching model which is a type of Hidden Markov Model (HMM) where hidden states show whether an account currently under attack or not. Although it gives an insight about real-time Bayesian fraud detection, [Hollmén and Tresp, 1998] does not inform readers about which stochastic model should be used to describe customers' behavior and how to collect 's callers' data in real-time.

After an intrusion, a caller's behavior suddenly starts to deviate from the usual pattern, and new observations do not resemble observations before the intrusion. In this case, observations of a customer would look like two disjoint sets. Our primary goal is to detect the breakpoint which separate observations into fragments. [Barry and Hartigan, 1992] and [Barry and Hartigan, 1993] proposed a product partition model that accepts observations in different segments of the data are independent and [Yang and Kuo, 2001] suggested a Bayesian approach to locate the changepoints in the Poisson process. They commented that as the number of observations increases, the computations becomes infeasible for the large number of changepoints. [Fearnhead, 2006], [Fearnhead and Liu, 2007] utilized filtering recursions to find the probability of time being a changepoint. The computational cost of recursions is quadratic in the number observations. To overcome the complexity of the algorithm, they proposed re-sampling algorithms. A survey by [Kurt et al., 2018] have developed a Bayesian changepoint model for the intrusion, and they use filtering recursions in [Fearnhead and Liu, 2007] to calculate the probability of change at each time point recursively.

In this thesis, we focus on a real-time application of Bayesian changepoint detection problem introduced by [Kurt et al., 2018]. Unlike their work, our observations are collected as discrete events which helps keeping track of callers' transactions continuously and therefore distinguish anomaly as early as possible. One assumption in this work is every caller has a distinct call behavior that describes his/her actions in the network. An algorithm is developed to detect change in users' behavior with assuming fraud is one of the most important cause of deviation in callers' usual patterns. We split callers' behavior into three parts: call frequency, call duration and call features and try to to detect fraud caused by a change in one of the call behaviors we mentioned. For catching fraud, we use Bayesian networks method which finds probability of fraud by calculating forward-backward recursions separately for each behavior type. To bound the computational cost of these recursions, we utilize a resampling algorithm. Lastly, service providers are cautious when it comes to sharing customers' personal data,hence, we implemented a call simulator which generates discrete events for one user.

The remainder of the thesis is organized as follows. Chapter 2 describes continuous-time call fraud detection and elements of call behavior. Our solution methodology is provided in 3, followed by the results of computational experiments in 4. Finally, Chapter 5 summarizes the thesis with remarks and points to some potential research directions.

2. MODEL DESCRIPTION AND FORMULATION

In this chapter, we will first introduce continuous time call fraud detection process Section 2.1 and later describe call behavior elements 2.2.

2.1. Continuous-Time Call Fraud Detection

By acquiring illegal access to the telecommunication network, criminals cause substantial loss to service providers and users. The goal in call fraud detection algorithm is to distinguish fraudulent calls from the normal ones. The main challenges of this problem is the following: Call fraud is very rare, and fraudsters do not occupy the system for a long time. Telecommunication companies monitor their customers' transactions with desire to detect anomaly instantly.

In many previous works like [Kurt et al., 2018], fraud detection algorithms discretize time and collect data in intervals with length of some time Δt for easier calculation. However, since criminals rush into the network, companies recognize the need to adopt a continuous-time model where they update their data at of every successive event and therefore detect frauds as early as possible.

Fraud detection in call data is generally a challenging task. A caller has an established behavior that describes his/her patterns in mobile networks, and callers' behavior does not need to follow a uniform process. Caller behavior parameters such as calling rate can vary during the day and the week. In this case, the change in the caller's patterns should not be considered as an intrusion. On the other hand, when criminals have access to a caller's account, they sometimes increase the calling rate, change the location of the customer or make calls to specific phone accounts. Therefore, it would be useful to create a detector which finds not only a change in the user's behaviour but also identify the reason for the change in order to determine whether the change is within the user's normal behavior or an intrusion.

The approach adopted in this thesis for the call fraud detection problem is based on the assumption that callers' behaviour changepoint model we introduce in Section 3.1. In many previous works, observation vector y_t in (Fig: 3.1) usually stored the information for the time interval $[t-1, t]$. However, we model our observations as events that we collect at time of the action.

We call the time interval between two successive changepoints a regime. In our model, we assume a non-homogeneous Poisson process for call arrivals in a single regime. Duration of successful calls are also modelled as the time for the first arrival of a non-homogeneous Poisson process. The non-homogeneities introduced are to reflect the different behaviour types of the user across different intervals of the day (or week). In that way, we hope to get rid of the false alarms due to the change in behavior by an hour and day. For example, once can divide a day into seven periods, as in Table 2.1, such as morning, lunch, afternoon, evening, night, overnight and dawn and separate week as weekday and weekend. In addition to arrivals and call durations, a call has a feature vector, whose each component is modelled as a random variable from the same Multinomial distribution within a regime. In order to represent caller's behavior better and understand the reason of anomaly, we split call behavior into three parts: frequency, duration, and features.

Table 2.1: periods of the day

08:00-12:00	12:00-14:00	14:00-18:00	18:00-21:00	21:00-00:00	00:00-04:00	04:00-08:00
morning	lunch	afternoon	evening	night	overnight	dawn

In the following Sections 2.2, 2.3 and 2.4, parts of the call behavior defined in this thesis are presented.

2.2. Call Frequency

In this section, we introduce the first part of the call behaviour,namely call frequency. Call frequency can roughly be defined as number of call starts per time. We assume that every caller has a specific calling rate for a particular hour and day of the week as we mentioned in Section 2.1. The notations for call frequency are summarized in Table 2.2.

Table 2.2: notation 1

Notation	
A_n	Start time of the n'th call.
E_n	End time of the n'th call.
$N^a(t)$	number of calls arriving from time 0 until time t.
$I_{i,j}$	Set of times for the (i, j)-type intervals.
$N^a(t_1, t_2)$	number of calls arriving between time t_1 and time t_2.
$N^a_{i,j}(t_1, t_2)$	number of calls arriving in an (i, j) interval between time t_1 and time t_2.
$\lambda^a_{i,j}$	call arrival rate for the (i, j)'th interval.
$\lambda^a(t)$	call arrival rate at time t.
$\tau_{i,j}(t_1, t_2)$	time spent in the (i, j) type intervals between t_1 and t_2.
n_d	number of time frame.
n_w	number of week frame.

To build a Bayesian model, we need to choose a stochastic process to describe call generation from a phone. As we mentioned in Section 2.1, We model customers' call traffic as an non-homogeneous Poisson process having a piecewise constant rate function that changes over n_w "week frame (WF)" and n_d "time frame (TF)"periods.

To be concrete in our desciption and to further build up our notation, let us continue with our example where a week is divided into $n_w = 2$ week periods (weekday, weekend), and a day is into $n_d = 7$ day periods. For each i, j, define the union of (i, j)-type periods as $I_{i,j}$. Here i denotes whether it is weekday ($i = 1$) or weekend ($i = 2$), and j denotes the j'th period of a day, starting from "morning" and ending at "dawn". For example, in words, $I_{1,1}$ $(1, 1$ stands for "weekday" and "morning") will be

$$I_{1,1} = \sum_{w=1}^{\infty} \sum_{d=1}^{5} \{\text{morning of } d\text{'th day of week } w\}. \tag{2.1}$$

For each (i, j) period, our model has a separate $\lambda^a_{i,j}$ which denotes the call arrival rate of the Poisson process during the (i, j)'th period for $i = 1, \ldots, n_w$ and $j = 1, \ldots, n_d$. Table 2.3 shows an example of call arrival rates when $n_w = 2$ and $n_d = 7$

Table 2.3: call arrival rate for each period

	morning	lunch	afternoon	evening	night	overnight	dawn
weekday	$\lambda^a_{1,1}$	$\lambda^a_{1,2}$	$\lambda^a_{1,3}$	$\lambda^a_{1,4}$	$\lambda^a_{1,5}$	$\lambda^a_{1,6}$	$\lambda^a_{1,7}$
weekend	$\lambda^a_{2,1}$	$\lambda^a_{2,2}$	$\lambda^a_{2,3}$	$\lambda^a_{2,4}$	$\lambda^a_{2,5}$	$\lambda^a_{2,6}$	$\lambda^a_{2,7}$

Specially, we define a non-homogeneous Poisson process $N^a(t)$ with intensity function $\lambda^a(t)$ defined as

$$\lambda^a(t) = \sum_{i=1}^{n_w} \sum_{j=1}^{n_d} \lambda^a_{i,j} \mathbb{I}(t \in I_{i,j}). \tag{2.2}$$

$$\mathbb{I}(x \in A) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

Let T_n refer the elapsed time between $(n-1)$'st and n'th call event

Cumulative distribution function for A_{n+1} given A_n:

$$\begin{aligned} \mathbb{P}(A_{n+1} \leq t_1 | A_n = t_0) &= 1 - \mathbb{P}(A_{n+1} > t_1 | A_n = t_0) & (2.3) \\ &= 1 - \mathbb{P}(T_{n+1} > t_1 - t_0 | A_n = t_0) & (2.4) \\ &= 1 - \exp\left\{ - \int_{t_0}^{t_1} \lambda^a(t) dt \right\}, & (2.5) \end{aligned}$$

hence,

$$p_{A_{n+1}|A_n}(t_1|t_0) = \lambda^a(t_1)\exp\left\{-\int_{t_0}^{t_1}\lambda^a(t)dt\right\}.$$

- $\tau_{i,j}(t_1, t_2)$: amount of time spent in the (i, j)'th period in the interval $(t_1, t_2]$.

$$\tau_{i,j}(t_1, t_2) = \int_{(t_1,t_2]\cap I_{i,j}} 1dt \tag{2.6}$$

- $N^a_{i,j}(t_1, t_2)$: number of calls that fall in the (i, j)'th period in the interval $(t_1, t_2]$.

$$N^a_{i,j}(t_1, t_2) \sim \mathcal{PO}\left(\lambda^a_{i,j}\tau_{i,j}(t_1, t_2)\right)$$

In particular, the probability distribution of the vector of counts during $(t_1, t_2]$ is given by,

$$N^a(t_2) - N^a(t_1) = \sum_{i=1}^{n_w}\sum_{j=1}^{n_d} N^a_{i,j}(t_1, t_2). \tag{2.7}$$

2.3. Call Duration

In this section, we will introduce the second part of the call behaviour: call duration. Call duration can be defined as average call length in a particular time interval and we assume that every caller has a specific call length for a particular hour and day of the week as we mentioned in Section 2.1. The notations for call duration are summarized in Table 2.4.

Table 2.4: notation 2

Notation	
E_n	End time of the n'th call.
$N^u(t_1, t_2)$	number of unsuccessful calls ending between time t_1 and time t_2.
$\lambda^d_{i,j}$	call duration rate for the (i, j)'th interval.
$\lambda^d(t)$	call arrival rate at time t.
γ	probability of a call being unsuccessful
α^c, β^c	Hyperparameters for the probability of a call being unsuccessful.

In our model, a call is successful with a certain probability $\gamma \in (0, 1)$ and the duration of a successful call is modelled as if it is an interarrival time for a non-homogeneous Poisson process with a piece-wise constant rate function $\lambda^d_{i,j}$ that is constructed in a similar fashion to that of the arrival process. Specifically, given intervals $I_{i,j}$ for $i = 1, \dots, n_w$, $j = 1, \dots, n_d$, we define the rate function as

$$\lambda^d(t) = \sum_{i=1}^{n_w} \sum_{j=1}^{n_d} \lambda^d_{i,j} \mathbb{I}(t \in I_{i,j}). \tag{2.8}$$

Table 2.5: call duration rate for each period

	morning	lunch	afternoon	evening	night	overnight	dawn
weekday	$\lambda^d_{1,1}$	$\lambda^d_{1,2}$	$\lambda^d_{1,3}$	$\lambda^d_{1,4}$	$\lambda^d_{1,5}$	$\lambda^d_{1,6}$	$\lambda^d_{1,7}$
weekend	$\lambda^d_{2,1}$	$\lambda^d_{2,2}$	$\lambda^d_{2,3}$	$\lambda^d_{2,4}$	$\lambda^d_{2,5}$	$\lambda^d_{2,6}$	$\lambda^d_{2,7}$

Table 2.5 shows an example of call duration rates when $n_w = 2$ and $n_d = 7$

For end times of the calls,we can write cumulative distribution function for E_n

given A_n similar to equation (2.3).

$$\mathbb{P}(E_n \leq t_1 | A_n = t_0) = \begin{cases} \gamma, & t_1 = t_0 \\ (1-\gamma)\left[1 - \exp\left\{-\int_{t_0}^{t_1} \lambda^d(t)dt\right\}\right], & t_1 > t_0 \\ 0. & \text{else.} \end{cases} \tag{2.9}$$

where γ is the probability of a call being unsuccessful. Hence, the probability density function of a positive duration is

$$p_{E|A}(t_1|t_0) = (1-\gamma)\lambda^d(t_1)\exp\left\{-\int_{t_0}^{t_1} \lambda^d(t)dt\right\}, \quad t_1 > t_0.$$

2.3.1. Call Success Probability

In our model, we define unsuccessful calls as the calls which their duration equal to 0 seconds. The change in the success rate of the call process is an indicator for fraud. Let $N^u(t_1, t_2)$ be the number of unsuccessful calls ending between time t_1 and time t_2.

The probability of a call being unsuccessful is γ and

$$\mathbb{P}(N^u(t_1, t_2) = x | N^a(t_1, t_2) = n) = \binom{n}{x}\gamma^x(1-\gamma)^{n-x} \tag{2.10}$$

Table 2.6: notation 3

Notation	
$C_j^{(i)}(t_1, t_2)$	number of calls in j'th category of feature i in $(t_1, t_2]$.
M	number of call features.
m_i	number of categories for feature i.
$\pi^{(i)}$	The distribution of categories belonging to the i'th feature.

The notations for call features are summarized in Table 2.6.

2.4. Features

In this section, we will introduce the last part of the call behavior: call features which can be determined at the start of the call. We model call features as a multinomial distribution with total sum during $(t_1, t_2]$ being $N(t_2) - N(t_1)$. Specifically, let us have M features with than time period of calls, and i'th feature has m_i categories.We define

$C_j^{(i)}(t_1, t_2)$ = the number of calls in j'th category of feature i in $(t_1, t_2]$,

$i = 1, \ldots, M$; $j = 1, \ldots, m_i$. Then, given the probability vector for feature i

$$\pi^{(i)} = (\pi_1^{(i)}, \ldots, \pi_{m_i}^{(i)})$$

and $N(t_1, t_2) = N(t_2) - N(t_1)$, number of calls between t_1 and t_2, the count vector

$$C^{(i)}(t_1, t_2) = (C_1^{(i)}(t_1, t_2), \ldots, C_{m_i}^{(i)}(t_1, t_2))$$

is multinomially distributed:

$$\mathbb{P}(C^{(i)}(t_1, t_2) = c_{1:m_i} | N(t_1, t_2) = n, \pi^{(i)}) = \text{Mult}(c_{1:m_i}; n, \pi^{(i)}). \tag{2.11}$$

Note that, in implementation one can include the call success information in call's features and, in the duration process, focus on detection of a changepoint only on successful calls. This is indeed what we do in our implementation. In Section 2.2, 2.3 and 2.4, we introduce the elements of the call behavior and associate each one with a parameter. We give more information about the variables of the call behavior in Section 2.5.

2.5. Priors for the Parameters

The behaviour of a caller during a regime can be characterised by the parameters of the model, which are

$$\Phi = \{\lambda^a_{i,j}, \lambda^d_{i,j}, i = 1, \ldots, n_w; j = 1, \ldots, n_d\}, \quad \gamma, \quad \{\pi^{(i)}, i = 1, \ldots, M\},$$

Since in these variables are not observed directly, we will call them the latent variables of the model. We treat those variables as random and assign them some distributions. In our problem, we choose a Bayesian approach to distinguish fraudulent behavior from the normal ones and the first step of this approach is to choose meaningful prior and posterior distribution. We assume that interarrival times and call durations are exponantially distributed with parameters $\lambda^a_{i,j}$ and $\lambda^d_{i,j}$, we set the gamma distribution as a prior for $\lambda^a_{i,j}$ and $\lambda^d_{i,j}$

$$\lambda^a_{i,j} \sim \text{Gamma}_1(\kappa^a_{i,j}, \theta^a_{i,j})$$
$$\lambda^d_{i,j} \sim \text{Gamma}_1(\kappa^d_{i,j}, \theta^d_{i,j})$$

$$X \sim \text{Gamma}_1(\kappa, \theta) \to p(x) = \frac{1}{\Gamma(\kappa)\theta^\kappa} x^{\kappa-1} e^{-x/\theta}$$

For γ, we assign a Beta prior for it.

$$\gamma \sim \text{Beta}(\alpha^c, \beta^c).$$

Finally, for the call features, we model them as a multinomial distribution in Section 2.4 with probabilities $\pi^{(i)} = (\pi_1^{(i)}, \ldots, \pi_{m_i}^{(i)}), \quad i = 1, \ldots, M$ and we choose $\pi^{(i)}$ from dirichlet distribution.

$$\pi^{(i)} \sim \text{Dir}(\rho_1^{(i)}, \ldots, \rho_{m_i}^{(i)})$$

Conjugacy is an important tool for Bayesian problems in terms of tractability of the posterior distributions.

Note that the latent variables are constant during a regime. When the regime changes following a changepoint, the latent variables are reinitiated from their distributions. What stays constant across the regimes is the set of hyperparameters

$$\mu = \{\kappa_{i,j}^a, \theta_{i,j}^a, \kappa_{i,j}^d, \theta_{i,j}^d, i = 1, \ldots, n_w; j = 1, \ldots, n_d\}, \quad \alpha^c, \beta^c, \quad \{\rho^{(i)}, i = 1, \ldots, M\},$$

2.6. Changepoints and Detecting Fraud

As we have mentioned before, our work mainly focuses on detecting illegal intrusion in the communication system. In this section, we present the idea behind changepoints and approach for locating them.

A fraud detection system should be quick to respond to intrusion. On the other hand, it must abstain from giving false alarms which can affect customers' satisfaction severely. A multiple changepoint model breaks data into disjoint sets, so that the data after the changepoint will become independent of the data before it [Fearnhead and Liu, 2007]. In our problem, we presume that after customer's telephone account is victimized, it is likely that his behavior patterns will change and new observations will be unrelated to normal behavior. As we mention in Chapter 1, we choose to adapt MCM because its memoryless property is suitable with our problem.

According to a multiple changepoint model, a set of observations $\{y_1, y_2, \ldots, y_n\}$ is divided into some unknown and random number, c > 0, of *segments*,

$$[y_{i_0}, y_{i_1-1}], [y_{i_1}, y_{i_2-1}], \ldots, [y_{i_{c-1}}, y_{i_c-1}] \qquad 1 = i_0 < i_1 < i_2 < \ldots < i_c - 1 = n$$

where each segment is independent from the other. The indices $i_0, i_1, i_2, \ldots i_c - 1$ constitutes the set of changepoints and we presume for our detection algorithm that time of the initial event, i_0 is always a changepoint.

Note that in this thesis we are interested in performing Bayesian filtering given the continuous-time process. In a discrete-time model, $\{y_1, \ldots, y_n\}$ is a realization of a sequence of vectors of random variables. In a continuous-time model, however, y_t can be taken to be the t'th portion of the continuous-time observation process, or the t'th event. Throughout the rest of the thesis, we will stick to this abuse of notation, sometimes without giving explicit reference to y_t, for sake of simplicity.

In this thesis, we are interested in finding three different partitions $\{y_1, y_2, \ldots, y_n\}$ of the same observation set, with respect to changes in the call arrival process, call duration distributions, and features. We consider, for example, a change in the behavior related to increasing call rate different from the change in call features. We assume that in the case of varying call frequency, duration and features, the observations can be split into u, v, and z segments which need not be identical.

$$[y_{i_0^a}, y_{i_1^a-1}], [y_{i_1^a}, y_{i_2^a-1}], \ldots, [y_{i_{u-1}^a}, y_{i_u^a-1}], \qquad 1 = i_0^a < i_1^a < i_2^a < \ldots < i_u^a - 1 = n$$

$$[y_{i_0^d}, y_{i_1^d-1}], [y_{i_1^d}, y_{i_2^d-1}], \ldots, [y_{i_{v-1}^d}, y_{i_v^d-1}], \qquad 1 = i_0^d < i_1^d < i_2^d < \ldots < i_v^d - 1 = n$$

$$[y_{i_0^f}, y_{i_1^f-1}], [y_{i_1^f}, y_{i_2^f-1}], \ldots, [y_{i_{z-1}^f}, y_{i_z^f-1}], \qquad 1 = i_0^f < i_1^f < i_2^f < \ldots < i_z^f - 1 = n$$

[Taniguchi et al., 1998] suggested that there is no deterministic approach to label a call as a fraud. However, one can calculate probability of intrusion given the caller's transactions in the phone network by keeping track of the caller's account in real time.

We have developed three separate detection algorithms working simultaneously to distinguish call frauds and their reasons from data calculating various probabilities, such as filtering, smoothing and fixed L-lag smoothing which will describe in Chapter 3

3. SOLUTION METHODOLOGY

In this chapter we first describe our model and present forward-backward recursions. We later describe how we adapt our problem to this model.

3.1. Multiple Changepoint Model Description

For our problem, we consider that given the position of a changepoint (the moment of victimization), the call data before the changepoint are independent of the data after the changepoint. Multiple changepoint models (MCM) are a form of hidden Markov models,HMM, where the observed states $\{y_1, y_2, \ldots\}$ conditionally depend on hidden states, and the hidden states either follow the previous regime or jump to a different one.
So given the properties of MCM, we choose to model our problem as a multiple changepoint model which breaks data into fractions and presume after the criminal access, a new regime starts. Main problem is finding the position of the changepoints where the user's behavior has changed.

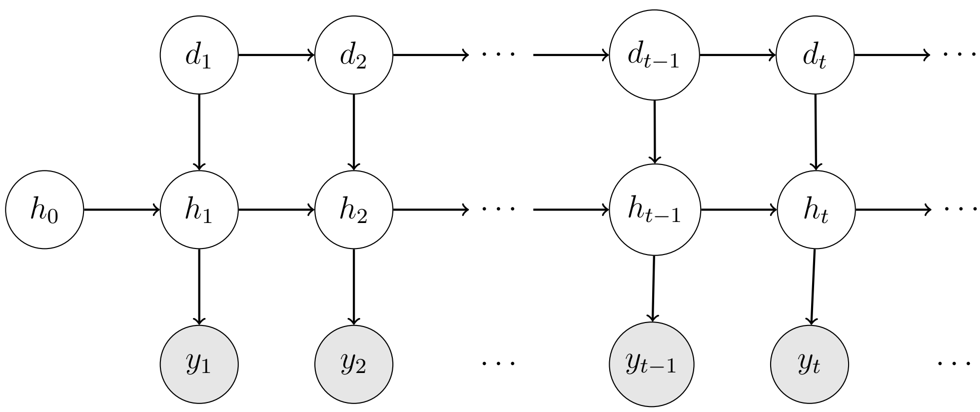


Figure 3.1: Hidden Markov Model

$$h_0 \sim \phi(h_0; \mu)$$
$$d_t \sim p(d_t|d_{t-1})$$
$$h_t \sim p(h_t|d_t, h_{t-1})$$
$$y_t \sim p(y_t|h_t)$$

where δ is Dirac delta function [Kurt et al., 2018].

In this model, we have three variables which changes over time according to Markov process. At the lowest hierarchical level, y_t represents observation at time t and it is a random variable sampled from $p(y_t|h_t)$. At the next level, h_t appears as the unknown parameters for the observed data. At the beginning, h_0 , initial parameters, are drawn from $\phi(h_0; \mu)$ - μ represents hyperparameters of the distribution. At each time point t, if data jumps to new regime ($d_t = 1$), then the parameters re-drawn from $\phi(h_t, \mu)$ distribution. In other case, they are equal to the previous value ($h_t = h_{t-1}$).

$$p(h_t|d_t, h_{t-1}) = \begin{cases} \phi(h_t; \mu) & \text{if} \quad d_t = 1 \\ h_{t-1} & \text{if} \quad d_t = d_{t-1} + 1 \end{cases}$$

At the highest level, we define d_t as time spent in the current regime (segment) at time t.

$$p(d_t|d_{t-1}) = \begin{cases} 1, & \text{if the regime changes at time } t \quad \text{with probability } \xi \\ d_{t-1} + 1, & \text{if the old regime continues at time } t \quad \text{with probability } 1 - \xi \end{cases}$$

where ξ is probability of regime change and $d_1 = 1$

For any $t \geq 1$, the joint probability density of $(d_{1:t}, h_{1:t}, y_{1:t})$ is given by

$$p(d_{1:t}, h_{1:t}, y_{1:t}) = p(h_0) \prod_{k=1}^{t} p(d_k|d_{k-1})p(h_k|h_{k-1}, d_k)p(y_k|h_k) \tag{3.1}$$

In the case of conjugacy, h_t integrals out and we have tractable density

$$p(d_{1:t}, y_{1:t}) = \prod_{k=1}^{t} p(d_k|d_{k-1})p(y_k|d_{k-1}, d_k, y_{1:k-1}) \tag{3.2}$$

3.2. Filtering and Smoothing

Our main aim is to capture the moment of change from normal behaviour to fraud based on our observations. At each time point t, we calculate the posterior probability of regime changing based on observations up to time t (filtering probabilities) $p(d_t = 1|y_{1:t})$. From Bayes Rule,

$$p(d_t|y_{1:t}) = \frac{p(d_t, y_{1:t})}{p(y_{1:t})} \tag{3.3}$$

Observations up to time t can be derived as

$$p(y_{1:t}) = \sum_{d_t} p(d_t, y_{1:t}) \tag{3.4}$$

For calculating $p(d_t, y_{1:t})$, we need to update the probability from the previous step, $p(d_{t-1}, y_{1:t-1})$, by taking consideration of the new observation ,y_t.

3.2.1. Forward Filtering and Conjugate Priors

Our objective is to calculate joint probability density $p(d_t, y_{1:t})$ recursively.

We start with $t = 1$, and set $p(d_1 = 1, y_1) = 1$ as we mentioned in Section 2.6.

When $t > 1$,

$$\alpha_t(k) = p(d_t = k, y_{1:t}) = p(d_t = k, y_{1:t-1}, y_t) \tag{3.5}$$

$$= \sum_{l=1}^{t-1} p(d_t = k, d_{t-1} = l, y_{1:t-1}, y_t) \tag{3.6}$$

$$= \sum_{l=1}^{t-1} \underbrace{p(d_{t-1} = l, y_{1:t-1})}_{\alpha_{t-1}(l)} \overbrace{p(d_t = k|d_{t-1} = l, y_{1:t-1})}^{p(d_t=k|d_{t-1}=l)} p(y_t|d_t = k, d_{t-1} = l, y_{1:t-1}) \tag{3.7}$$

$$= \sum_{l=1}^{t-1} \alpha_{t-1}(l) p(d_t = k|d_{t-1} = l) p(y_t|d_t = k, d_{t-1} = l, y_{1:t-1}) \tag{3.8}$$

For finding a repetitive relation between $p(d_t, y_{1:t})$ and $p(d_{t-1}, y_{1:t-1})$, we marginalize over all possible d_{t-1} values to calculate $p(d_t, y_{1:t})$ at equation (3.6) . d_{t-1}, time spent in the current regime at time $t-1$, can vary between 1 and $t-1$. In equation (3.7), which we obtained from Bayes Rule, the first part of the equation shows what we need for building a recursive relation: $p(d_{t-1}, y_{1:t-1})$. In addition, the second part can be obtained from the conditional independence property which we showed in Figure 3.1. And the last part indicates probability of the new observation given past observations and the length of the current and previous regime segments.

$$p(d_t = k|d_{t-1} = l) = \begin{cases} \xi, & \text{if} \quad k = 1 \\ 1 - \xi, & \text{if } k = l + 1 \\ 0 & \text{otherwise} \end{cases}$$

$$p(y_t|d_t = k, d_{t-1} = l, y_{1:t-1}) = p(y_t|y_{t-k+1:t-1})$$

Since observations don't depend on the previous regime.

- if $k = 1$,

  This means new regime has started, and y_t is independent of past observations. Then, $p(y_t|y_{t-k+1:t-1}) = p(y_t)$

- if $k > 1$,

$$p(y_t|y_{t-k+1:t-1}) = \frac{p(y_{t-k+1:t})}{p(y_{t-k+1:t-1})} \tag{3.9}$$

$$p(y_{\tau:t}) = \int\limits_h p(y_{\tau:t}, h)dh = \int\limits_h p(y_{\tau:t}|h)p(h)dh \tag{3.10}$$

In Bayesian statistics, primary step to build a statistical model is to decide on the likelihood, i.e. the conditional distribution of the data given the unknown parameter. The likelihood represents the model choice for the data and it should represents the real stochastic dynamics/phenomena of the data generation process as accurately as possible.

$$\text{posterior} \sim \text{prior x likelihood}$$

It is useful to consider a certain family of distributions for the prior distribution so that the posterior distribution has the same form as the prior distribution but with different parameters. For making our calculations in equation (3.10) tractable, we choose h as conjugate prior for the likelihood $p(y_t|h)$.

Lastly, ın reference to equation (3.4)

$$\alpha_t(k) = p(d_t = k, y_{1:t}), \quad p(d_t = k|y_{1:t}) = \frac{\alpha_t(k)}{\sum\limits_{l=1}^{t} \alpha_t(l)} \tag{3.11}$$

3.2.2. Backward Smoothing

For real time fraud detection, our algorithm uses posterior probabilities to capture moment of change in the behaviour. This online algorithm only needs the observations until the current time, $y_{1:t}$. In an offline setting where we have all observations for time $t = 1, \ldots, n$, estimate for the change point becomes more accurate after calculating $p(d_t = 1|y_{1:n})$ (smoothing probability).

We start our recursion with $p(d_n = 1|y_{1:n})$ which can be calculated from forward recursions in Section 3.2.1. The goal is to derive $p(d_t = 1|y_{1:n})$ from $p(d_{t+1} = 1|y_{1:n})$. There are two ways to calculate smoothing probabilities.

- Forward filtering-backward smoothing

$$p(d_t = k|y_{1:n}) = \sum_{d_{t+1}} p(d_t = k, d_{t+1}|y_{1:n}) \tag{3.12}$$

$$= \sum_{l=1}^{t+1} p(d_{t+1} = l|y_{1:n}) p(d_t = k|d_{t+1} = l, y_{1:n}) \tag{3.13}$$

$$= \sum_{l=1}^{t+1} p(d_{t+1} = l|y_{1:n}) p(d_t = k|d_{t+1} = l, y_{1:t}) \tag{3.14}$$

$$= \sum_{l=1}^{t+1} p(d_{t+1} = l|y_{1:n}) \frac{p(d_{t+1} = k|d_t = l) p(d_t = l|y_{1:t})}{\sum_{z=1}^{t} p(d_{t+1} = l|d_t = z) p(d_t = z|y_{1:t})} \tag{3.15}$$

The reason for the change from $y_{1:n}$ to $y_{1:t}$ in equation (3.14) is the conditional independence property of our model.

- Two-filter smoothing

$$p(d_t = k|y_{1:n}) = \frac{p(d_t = k, y_{1:n})}{\sum_{l=1}^{t} p(d_t = l, y_{1:n})} \tag{3.16}$$

$$p(d_t = k, y_{1:n}) = p(d_t = k, y_{1:t}, y_{t+1:n})$$

$$= p(d_t = k, y_{1:t}) p(y_{t+1:n}|d_t = k, y_{1:t}) \tag{3.17}$$

$$\beta_t(k) = p(y_{t+1:n}|d_t = k, y_{1:t})$$

For convention, we choose $\beta_n(k) = 1, \quad \forall k = 1, \ldots, n$.

The backward recursion can be shown as

$$\beta_t(k) = p(y_{t+1:n}|d_t = k, y_{1:t}) \tag{3.18}$$

$$= \sum_{d_{t+1}} p(y_{t+1:n}, d_{t+1} = l|d_t = k, y_{1:t}) \tag{3.19}$$

$$= \sum_{l=1}^{t+1} p(y_{t+1}, y_{t+2:n}, d_{t+1} = l|d_t = k, y_{1:t}) \tag{3.20}$$

$$= \sum_{l=1}^{t+1} \overbrace{p(d_{t+1} = l|d_t = k, y_{1:t})}^{p(d_{t+1}=l|d_t=k)} p(y_{t+1}|d_{t+1} = l, y_{1:t}) \underbrace{p(y_{t+2:n}|y_{1:t+1}, d_{t+1} = l, d_t = k)}_{\beta_{t+1}(l)} \tag{3.21}$$

$$= \sum_{l=1}^{t+1} p(d_{t+1} = l|d_t = k)p(y_{t+1}|y_{t-y+2:t})\beta_{t+1}(l) \tag{3.22}$$

The smoothed density is calculated as

$$p(d_t = k|y_{1:n}) = \frac{p(d_t = k, y_{1:n})}{\sum_{l=1}^{t} p(d_t = l, y_{1:n})} = \frac{\alpha_t(k)\beta_t(k)}{\sum_{l=1}^{t} \alpha_t(l)\beta_t(l)} \tag{3.23}$$

3.2.3. Tracking Latent Variables

The behaviour of a user during a regime can be characterized by the hidden variables of the model, which are

$$\phi = (\{\lambda^a_{i,j}, \lambda^d_{i,j}, i = 1, \ldots, n_w; j = 1, \ldots, n_d\}, \quad \gamma, \quad \{\pi^{(i)}, i = 1, \ldots, M\}),$$

Since these variables are not observed directly, we will call them the latent variables of the model. In our problem, unknown parameters, h_t in 3.1 do not follow the previous regime when $d_t = 1$.

To visualize the change in unknown parameters clearly, we calculate the expectations of h_t at each event t.

Let h_t be the variable of interest so that we want to find

$$E(H_t|y_{1:t}) = \int_{h_t} p(h_t|y_{1:t})h_t dh_t = \int_{h_t} \sum_{d_t=1}^{N_t} p(d_t, h_t|y_{1:t})h_t dh_t$$

$$\sum_{d_t=1}^{N_t} \int_{h_t} p(d_t, h_t|y_{1:t})h_t dh_t = \sum_{d_t=1}^{N_t} \int_{h_t} p(d_t|y_{1:t})p(h_t|d_t, y_{1:t})h_t dh_t$$

$$= \sum_{d_t=1}^{N_t} p(d_t|y_{1:t}) \int_{h_t} p(h_t|d_t, y_{1:t})h_t dh_t = \sum_{d_t=1}^{N_t} p(d_t|y_{1:t})E(H_t|y_{1:t}, d_t)$$

$$E(H_t|y_{1:n}) = \sum_{d_t=1}^{N_t} \int_{h_t} p(d_t, h_t|y_{1:n})h_t dh_t = \sum_{d_t=1}^{N_t} \int_{h_t} p(d_t|y_{1:n})p(h_t|d_t, y_{1:n})h_t dh_t$$

Let

$$\eta_t^s(k) = E(H_t|y_{1:n}, d_t = k) = \int_{h_t} p(h_t|d_t = k, y_{1:n})h_t dh_t$$

and

$$\eta_t^f(k) = E(H_t|y_{1:t}, d_t = k) = \int_{h_t} p(h_t|d_t = k, y_{1:t})h_t dh_t$$

Also, use short-hand notations $\gamma_t^f(k) = p(d_t = k|y_{1:t})$ and $\gamma_t^s(k) = p(d_t = k|y_{1:n})$. The smoothed expectation can be rewritten as

$$E(H_t|y_{1:n}) = \sum_{k=1}^{N_t} \eta_t^s(k)\gamma_t^s(k) \tag{3.24}$$

We want to derive a backward recursion for $\eta_t^s(k)$, hence for $E(H_t|y_{1:n})$ indirectly.

$$\begin{aligned}
\eta_t^s(k) &= \int_{h_t} \sum_{k'} \int_{h_{t+1}} p(h_t, d_{t+1} = k', h_{t+1}|d_t = k, y_{1:n}) h_t dh_t dh_{t+1} \\
&= \int p(h_t, d_{t+1} = k+1, h_{t+1}|d_t = k, y_{1:n}) h_t dh_t dh_{t+1} \quad + \\
&\qquad \int p(h_t, d_{t+1} = 1, h_{t+1}|d_t = k, y_{1:n}) h_t dh_t dh_{t+1} \\
&= \int \frac{p(h_t, d_{t+1} = k+1, h_{t+1}, d_t = k|y_{1:n})}{p(d_t = k|y_{1:n})} h_t dh_t dh_{t+1} \quad + \\
&\qquad \int \frac{p(h_t, d_{t+1} = 1, h_{t+1}, d_t = k|y_{1:n})}{p(d_t = k|y_{1:n})} h_t dh_t dh_{t+1}
\end{aligned}$$

We investigate the two terms in the last line. The first term is

$$\begin{aligned}
&\int \frac{p(h_t, d_{t+1} = k+1, h_{t+1}, d_t = k|y_{1:n})}{p(d_t = k|y_{1:n})} h_t dh_t dh_{t+1} \\
&= \int \frac{p(h_{t+1}|d_{t+1} = k+1, y_{1:n}) \gamma_{t+1}^s(k+1) p(d_t = k, h_t|d_{t+1} = k+1, h_{t+1}, y_{1:n})}{\gamma_t^s(k)} h_t dh_t dh_{t+1} \\
&= \int \frac{p(h_{t+1}|d_{t+1} = k+1, y_{1:n}) \gamma_{t+1}^s(k+1)}{\gamma_t^s(k)} \left[\int p(d_t = k, h_t|d_{t+1} = k+1, h_{t+1}, y_{1:n}) h_t dh_t \right] dh_{t+1} \\
&= \int \frac{p(h_{t+1}|d_{t+1} = k+1, y_{1:n}) \gamma_{t+1}^s(k+1)}{\gamma_t^s(k)} h_{t+1} dh_{t+1} \\
&= \eta_{t+1}^s(k+1) \frac{\gamma_{t+1}^s(k+1)}{\gamma_t^s(k)}
\end{aligned}$$

The second term is

$$\begin{aligned}
&\int \frac{p(h_t, d_{t+1} = 1, h_{t+1}, d_t = k|y_{1:n})}{p(d_t = k|y_{1:n})} h_t dh_t dh_{t+1} \\
&= \int \frac{p(h_{t+1}|d_{t+1} = 1, y_{1:n}) \gamma_{t+1}^s(1) p(d_t = k, h_t|d_{t+1} = 1, h_{t+1})}{\gamma_t^s(k)} h_t dh_t dh_{t+1} \\
&= \int \frac{p(h_{t+1}|d_{t+1} = 1, y_{1:n}) \gamma_{t+1}^s(1)}{\gamma_t^s(k)} \left[\int p(d_t = k, h_t|d_{t+1} = 1, h_{t+1}, y_{1:n}) h_t dh_t \right] dh_{t+1} \\
&= \int \frac{p(h_{t+1}|d_{t+1} = 1, y_{1:n}) \gamma_{t+1}^s(1)}{\gamma_t^s(k)} dh_{t+1} \int p(d_t = k, h_t|y_{1:t}) h_t dh_t \\
&= \frac{\gamma_{t+1}^s(1)}{\gamma_t^s(k)} \eta_t^f(k) \gamma_t^f(k)
\end{aligned}$$

Summing up, we have

$$\eta_t^s(k) = \eta_{t+1}^s(k+1)\frac{\gamma_{t+1}^s(k+1)}{\gamma_t^s(k)} + \frac{\gamma_{t+1}^s(1)}{\gamma_t^s(k)}\eta_t^f(k)\gamma_t^f(k) \tag{3.25}$$

3.2.4. Fixed L-lag Smoothing

In an ideal world, filtering probabilities should be enough to give accurate estimations for the probability of fraud. However, sometimes, depending on the type of fraud, the intrusion detection might require additional information about the data. Since smoothing density $p(d_t = 1, y_{1:n})$ possesses more information about the data, it always gives better evaluation for anomaly in the observations than the filtering density, $p(d_t = 1, y_{1:t})$. However, the goal of our model is finding fraud as soon as the customer is victimized and since we are working with real-time data, n is unbounded. In our case, using smoothing density for fraud detection causes a high delay in the performance of the algorithm. As it is suggested in [Kurt et al., 2018] , to solve this problem, we allow ourselves a reasonable delay time and find $p(d_t = 1|y_{1:t+L})$ -posterior probability of fraud at time t, given data up to time $t + L$ where L is fixed and called the *lag*.

3.2.5. Resampling Methods

As we have demonstrated in section 3.2.1, to calculate the filtering densities at event t exactly, we require to store all filtering densities $p(d_{t-1} = i|y_{1:t})$ where $i = 1, \ldots, t-1$. We basically assume every time point until t can be a changepoint candidate. Hence, the computational cost of our algorithm increases linearly over time and for t time steps it becomes $\mathcal{O}(t^2)$. In the interest of bounding the computational time, we can use a resampling algorithm which the time points with lower filtering densities can be omitted.

We predetermine a lower bound for filtering density, ω and upper bound for number of changepoint candidates N. At time t, Our algorithm choose maximum N changepoint candidates that their filtering density is higher than ω and approximate their filtering densities.

Algorithm 1: Resampling Algorithm

Input: Filtering probabilities at time t : $p(d_t = i|y_{1:t}) \quad i = 1, \dots, t$,
maximum number of particles: N, threshold : ω

Output: Resampled filtering probabilities $\widetilde{p}(d_t = i|y_{1:t}) \quad i \in S \subseteq \{1, \dots, t\}$

$\widetilde{P} = [\quad], \widetilde{r} = [\quad]$
for $i = 1, \dots, t$ **do**
  $x_t^i = p(d_t = i|y_{1:t})$
  if $x_t^i \geq \omega$ **then**
    add x_t^i to $\widetilde{P}$
Calculate the length of the vector $\widetilde{P}$ as $N_{\widetilde{P}}$ and set $N_s = min(N, N_{\widetilde{P}})$
Sort $\widetilde{P}$ in a descending order and set $\widetilde{P} = \widetilde{P}[1 : N_s]$
Set $\widetilde{r}$ as $\{i\}_{x_t^i \in \widetilde{P}}$
Calculate total weight , $b = SUM\left[\left\{x_t^i\right\}_{x_t^i \in \widetilde{P}}\right]$
for $i = 1, \dots, t$ **do**
  if $i \in \widetilde{r}$ **then**
    Normalize x_t^i, $\quad x_t^i = x_t^i / b$
    set $\widetilde{p}(d_t = i|y_{1:t}) = x_t^i$
  else
    $x_t^i = 0$

3.3. Adaptation of Real-Time and Event-Based Fraud Detection to MCM

In this section, we describe how we adapted our real-time and event-based fraud detection problem to multiple changepoint model we described above

In the model we introduced in Section 3.1, we show that the variable y_t represents the observation at time t. However in our problem, t in y_t stands for the event number and we refer $\bar{t}$ as the occurrence time of the event t. In Section 3.3.1, we will describe our events and their difference from previous works.

3.3.1. Event Types

A study by [Kurt et al., 2018], they also adapted their problem multiple change-point models and stored their observations in multidimensional vectors. However, in their problem, they collected y_t for time a time interval $[t-1,t]$ which is not compatible to our goals. Since we are monitoring the call traffic of a customer in real-time, our observations are mainly when a call starts and ends, where the callee resides or whether a call is successful or not. We define our observations, y_t as vectors which hold the information we need. We consider the following events generated by a single user, ordered in time, as observations. We have 6 event types:

Table 3.1: event types

Event type	**Description**	**Event form**
Call Start	Start of a call	`[`$\bar{t}$`, 'CS', call ID, features]`
Call End	End of a call	`[`$\bar{t}$`,'CE', call ID]`
Day Change	Change from one interval of the day to the next	`[`$\bar{t}$`,'DC']`
Week Change	Change from one interval of the week to the next	`[`$\bar{t}$`, 'WC']`
Call in Progress	The call is in progress	`[`$\bar{t}$`,'CP', call ID]`
No Call	This is a 'dummy' event, simply indicating no event	`[`$\bar{t}$`, 'NC']`

In all the event types, the first entry $\bar{t}$ is the time of the event and the second entry is the abbreviation for the type of the event. `Call ID` is the call ID, and `features` is an array whose elements correspond to several features of the call, including whether a call is successful or not.

The event types `'DC'`, `'WC'`, `'CP'`, and `'NC'` are not user-generated. Rather, they are artificially generated by the detection system. The detection system has to be noticed the interval changes since we assume that even within a single regime the user behavior might vary over different intervals of the week. The events `'CP'`, and `'NC'` are optional. We set a fix call process checking time t_{CP}. If a call starting at time t continues at time $t + t_{CP}$, we add Call Process as a new observation. `'CP'` may be useful for detection of a changepoint in call duration behavior even during a call since thanks to this event the system does not need to wait until the call ends. Similarly, `'NC'` can be used to detect an adverse change in call frequency earlier than the start of a next call which may happen after a very long time. They are useful if the changepoint detection system is desired to process the calls in the middle of a call or when there are no arrivals for a while. In the next section, we explain the use of discrete events for tracking the caller's transactions.

3.3.2. Equivalance of events and the process

The continuous-time changepoint process we described is fully observable as a discrete-event system. In other words, all the states and cumulative statistics that are necessary to perform changepoint detection in the data can exactly be extracted from a sequence of events constructed appropriately.
Specifically, let the sequence of call-related events generated by a user, up to the n'th event, be denoted by

$$y_1, y_2, \ldots, y_n$$

Also, let the occurrence times of those events be

$$\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_n$$

Finally, let the continuous time observation process be

$$\{Y(t)\}_{t\geq 0}$$

Note that $\{Y(t)\}_{t\geq 0}$ contains all the necessary states and statistics that are necessary to perform inference, such as the arrival process, the process of the set of all of active calls, feature counts for the calls, etc.
The k'th event y_k can be designed carefully so that, for any $k \geq 1$, given $Y([0, \bar{t}_{k-1}])$, the event y_k tells us the rest of the process until time $\bar{t}_k$, i.e. the portion $Y((\bar{t}_{k-1}, \bar{t}_k])$, where for a time interval A we define $Y(A) = \{Y(t) : t \in A\}$.

3.3.3. Changepoints

As we have mentioned in Section 2.6, we choose the time of the initial event as a changepoint candidate for our real-time fraud detection problem. A significant adaptation of the initial model we described to our problem is our understanding of the changepoints. According to the model we described at section 3.1, at observation y_t, the time spent in the current regime can vary between 1 to t which implies every event time up to t is a candidate for a changepoint.

In our problem, the perception about the changepoints has been changed after considering that treating every event as a changepoint candidate is not useful. Our main aim is to detect the moment of victimization and intrusion can only be seen when the fraudster starts making phone calls.

To clarify, we assume only the times of events which are call start (CS) can be seen as a changepoint candidate. For every event y_k, we define a set

$$[CS]_k = \{\text{initial time} + \text{All call start times up to event k}\}$$

. $[CS]_k$ stores all changepoint candidate times up to event k or equivalently time $\overline{t_k}$.

3.4. Filtering and Smoothing

We define a new random variable X_k to be the occurrence time of the last changepoint at event k.

$$\{X_k = i\} = \{\text{last changepoint at event } k \text{ occured at event } i \text{ where } \overline{t_i} \in [CS]_k\}$$

- **Filtering**

  Now, we calculate sequentially the probabilities

  $$P(X_k = i|y_{1:k}), \qquad \overline{t_i} \in [CS]_k$$

  In particular, we are interested whether there is a changepoint at a given event k, i.e.

  $$P(X_k = k|y_{1:k}),$$

- Smoothing

  $$P(X_k = i|y_{1:n}), \qquad \overline{t_i} \in [CS]_k$$

  In particular, we are interested whether there is a changepoint at a given event k, i.e.

  $$P(X_k = k|y_{1:n}),$$

We used particle filters that represents the changepoint candidates and every particle stores the information that we need for finding probability of fraud. In our problem, whenever a new call arrives, we first update the information in the particles and filtering densities then we add a new particle that represents the latest changepoint.

3.4.1. Conjugate Priors

As we have mentioned, for the purpose of conjugacy, we choose $\lambda^a_{i,j}$ and $\lambda^d_{i,j}$ from the gamma distribution:

$$\lambda^a_{i,j} \sim \text{Gamma}_1(\kappa^a_{i,j}, \theta^a_{i,j})$$
$$\lambda^d_{i,j} \sim \text{Gamma}_1(\kappa^d_{i,j}, \theta^d_{i,j})$$

$X \sim \text{Gamma}_1(\kappa, \theta) \rightarrow p(x) = \frac{1}{\Gamma(\kappa)\theta^\kappa} x^{\kappa-1} e^{-x/\theta}$

Proposition 1. *Let* $\lambda \sim \text{Gamma}_1(\kappa, \theta)$ *with* $p(\lambda) = \frac{\theta^{-\kappa}}{\Gamma(\kappa)} \lambda^{\kappa-1} e^{-\lambda/\theta}$ *and* $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} \text{Exp}(\lambda)$. *Then, for any* $m \leq n$,

$$\lambda | X_{1:m} = x_{1:m}, X_{m+1:n} > x_{m+1:n} \sim \text{Gamma}_1 \left(m + \kappa, \left[\frac{1}{\theta} + \sum_{i=1}^{n} x_i \right]^{-1} \right) \tag{3.26}$$

and

$$p(X_{1:m} = x_{1:m}, X_{m+1:n} > x_{m+1:n}) = \frac{\theta^{-\kappa}}{(1/\theta + \sum_{i=1}^{n} x_i)^{m+\kappa}} \frac{\Gamma(m+\kappa)}{\Gamma(\kappa)} \tag{3.27}$$

Interarrival times and are exponentially distributed, hence we can apply the equation (3.26) for interarrival times to calculate the likelihood. Let T_n the elapsed time between $(n-1)$'st and n'th call start and T_n is calculated as soon as n'th call arrives.

Let the changepoint for the duration process be t_c ,the current time be t, The following are needed to compute the probability of the events in the last regime:

- $\tau_{i,j}(t_c,t)$ for each (i,j); $i=1,\ldots,n_w$; $j=1,\ldots,n_d$,
- $N^a_{i,j}(t_c,t)$ for each i,j; $i=1,\ldots,n_w$; $j=1,\ldots,n_d$.

Let m be the number interarrival times in period (i,j) between t_c and t. Also, the number interarrival times in period (i,j) equals to number of calls visiting period (i,j), $N^a_{i,j}(t_c,t)=m$.

$$p(T_{1:m}=t_{1:m},T_{m+1:n}>t_{m+1:n})=\frac{(\theta^a_{i,j})^{-\kappa^a_{i,j}}}{\left[1/\theta^a_{i,j}+\sum_{i=1}^{N^a_{i,j}(t_c,t)}t_i\right]^{m+\kappa^a_{i,j}}}\frac{\Gamma(m+\kappa^a_{i,j})}{\Gamma(\kappa^a_{i,j})} \tag{3.28}$$

can be calculated for each (i,j) period, where $\sum_{i=1}^{N^a_{i,j}(t_c,t)} t_i$ equals to total time spent in (i,j) interval between t_c and t, $\tau_{i,j}(t_c,t)$

By Proposition 1, given $\tau_{i,j}(t_c,t)=\tau_{i,j}$, the joint probability of $\{N^a_{i,j}(t_c,t)=n_{i,j};i,j\}$ is

$$p^a(n_{1:n_w,1:n_d};\tau_{1:n_w,1:n_d})=\prod_{i=1}^{n_w}\prod_{j=1}^{n_d}\frac{(\theta^a_{i,j})^{-\kappa^a_{i,j}}}{(1/\theta^a_{i,j}+\tau_{i,j})^{n_{i,j}+\kappa^a_{i,j}}}\frac{\Gamma(n_{i,j}+\kappa^a_{i,j})}{\Gamma(\kappa^a_{i,j})}.$$

Let the changepoint for the duration process be t_c ,the current time be t, For each changepoint for call durations, the following are needed to compute the probabilities between the two regimes:

- $N^p_{i,j}(t_c,t)$: number of calls progressed in interval (i,j) between t_c and t for each (i,j); $i=1,\ldots,n_w$; $j=1,\ldots,n_d$,
- $N^e_{i,j}(t_c,t)$: number of calls ended in interval (i,j) between t_c and t for each (i,j); $i=1,\ldots,n_w$; $j=1,\ldots,n_d$,
- $N^a_{i,j}(t_c,t)$,
- $D^p_{i,j}(t_c,t)$: duration of calls progressed in interval (i,j) between t_c and t for each (i,j); $i=1,\ldots,n_w$; $j=1,\ldots,n_d$,

- $D^e_{i,j}(t_c,t)$: duration of calls ended in interval (i,j) between t_c and t for each (i,j); $i=1,\ldots,n_w$; $j=1,\ldots,n_d$.

For call duration, Let D_n denote the duration of the nth call and m be the number of calls that ended in period (i,j) between t_c and t, $N^e_{i,j}(t_c,t)$. It has been noted that D_n's are exponentially distributed with rate $\lambda^d_{i,j}$

$$p(D_{1:m}=d_{1:m}, D_{m+1:N^a_{i,j}(t_c,t)} > d_{m+1:N^a_{i,j}(t_c,t)}) = \frac{(\theta^d_{i,j})^{-\kappa^d_{i,j}}}{(1/\theta^d_{i,j}+\sum_{i=1}^{N^a_{i,j}(t_1,t_2)} d_i)^{m+\kappa^d_{i,j}}} \frac{\Gamma(m+\kappa^d_{i,j})}{\Gamma(\kappa^d_{i,j})} \tag{3.29}$$

By Proposition 1, given $N^a_{i,j}(t_c,t)=n$, the joint probability of $\{D^p_{i,j}(t_c,t) = d^p_{i,j}, D^e_{i,j}(t_c,t)=d^e_{i,j}; i,j\}$ with $\{N^p_{i,j}(t_c,t)=n^p_{i,j}, N^e_{i,j}(t_c,t)=n^e_{i,j}; i,j\}$ is

$$p^d(d^p_{1:n_w,1:n_d}, d^e_{1:n_w,1:n_d}) = \tag{3.30}$$

$$\prod_{i=1}^{n_w}\prod_{j=1}^{n_d} \frac{(\theta^d_{i,j})^{-\kappa^d_{i,j}}}{(1/\theta^d_{i,j}+\sum_{l=1}^{n^e} d^e_{i,j}(l)+\sum_{l=1}^{n^p} d^p_{i,j}(l))^{n^e_{i,j}+\kappa^d_{i,j}}} \frac{\Gamma(n^e_{i,j}+\kappa^d_{i,j})}{\Gamma(\kappa^d_{i,j})}. \tag{3.31}$$

where $\sum_{l=1}^{n^e} d^e_{i,j}(l)+\sum_{l=1}^{n^p} d^p_{i,j}(l)$ is the sum of call durations of the calls which processed or ended at period (i,j)

Let the changepoint for the duration process be t_c ,the current time be t, For each changepoint for call features, the following are needed to compute the probabilities between the two regimes: For the call features, we model them as a multinomial distribution in Section 2.4 with probabilities $\pi^{(i)}=(\pi^{(i)}_1,\ldots,\pi^{(i)}_{m_i})$ and for the purpose of conjugacy, we choose $\pi^{(i)}$ from dirichlet distribution

$$\pi^{(i)} \sim \mathrm{Dir}(\rho^{(i)}_1,\ldots,\rho^{(i)}_{m_i})$$

As a result, we have tractable marginal distribution for the counts in Section 2.11:

$$\mathbb{P}(C^{(i)}(t_c,t) = c^{(i)} | N^a(t_c,t) = n) = \text{Dir-Mult}\left(c^{(i)}; \rho^{(i)}, n\right) \tag{3.32}$$

where Dir-Mult denotes the Dirichlet-Multinomial distribution, whose probability distribution is given by

$$\text{Dir-Mult}\left(x_1,\ldots,x_M;\rho,n\right) = \frac{\Gamma(\sum_{k=1}^{M}\rho_k)n!}{\Gamma(n+\sum_k \rho_k)} \prod_{k=1}^{M} \frac{\Gamma(x_k+\rho_k)}{\Gamma(\rho_k)x_k!}.$$

Therefore, for all the counts,

$$\mathbb{P}(C^{(1)}(t_c,t) = c^{(1)},\ldots,C^{M}(t_1,t_2) = c^M | N^a(t_c,t) = n) = \prod_{i=1}^{M} \text{Dir-Mult}\left(c^{(i)}; \rho^{(i)}, n\right) \tag{3.33}$$

In Section 2.3.1, we defined the the probability of a call being unsuccessful as γ and we assign a Beta prior for it.

$$\gamma \sim \text{Beta}(\alpha^c, \beta^c).$$

The marginal probability for the number of unsuccessful calls is a Beta-Binomial probability with parameters α^c and β^c:

$$\mathbb{P}(N^u(t_c,t) = x | N^a(t_c,t) = n) = \binom{n}{x} \frac{\text{B}(\alpha^c + x, \beta^c + n - x)}{\text{B}(\alpha,\beta)}$$

With every event y_t, we update the elements we used for calculating the likelihood and our filtering densities. See Appendix B.1 for keeping sufficient statistics

3.4.2. Hyperparameters

Hyperparameters are the variables that stay constant across all the regimes. Unlike real world, we presumed that the values of hyperparameters are given. Let μ is the set of hyperparameters

$$\mu = \{\kappa^a_{i,j}, \theta^a_{i,j}, \kappa^d_{i,j}, \theta^d_{i,j}, i = 1, \ldots, n_w; j = 1, \ldots, n_d\}, \quad \alpha^c, \beta^c, \quad \{\rho^{(i)}, i = 1, \ldots, M\},$$

For parsimony, we reduce the number of hyperparameters by setting $\kappa^a_{i,j} = \kappa^a$, $\theta^a_{i,j} = \theta^a$, $\kappa^d_{i,j} = \kappa^d$, $\theta^d_{i,j} = \theta^d$, $\rho^{(i)}_j = \rho$.

4. EXPERIMENTS AND RESULTS

In this chapter, we perform computational studies to show the performance of our proposed algorithm in Chapter 3.

4.1. Experimental Setup

4.1.1. Data Generation

In Chapter 3, we discussed the important features of our data types and commented that in order to detect fraud we must collect our observations as events we have shown in Section 3.3.1. Because of the privacy reasons, we create our own data generator which simulates events as we discussed. See Appendix B.

4.1.2. Decision Making

There are several possible ways to make an ultimate decision whether a changepoint occurs or not:

- **Changepoint probabilities:** Since we are using Bayesian method, our filtering probabilities tells us how likely a time point is a last changepoint at time $\bar{t}$

  We refer to $p(X_k = i|y_{1:k})$ as probability of last changepoint time is $\overline{t_i}$ at time $\overline{t_k}$ given events up to time $\overline{t_k}$.
  In case of filtering (respectively smoothing, fixed-lag smoothing), a decision for a changepoint can be made based on tracking the probabilities of

$$p(X_k = k|y_{1:k}) \quad \text{(resp. } p(X_k = k|y_{1:n}),\, p(X_k = k|y_{1:k+L})\text{) for each } k > 0.$$

  Our algorithm creates an alarm if probability of being a changepoint exceeds a predetermined threshold ς,

$$p(X_k = k|y_{1:k}) > \varsigma \quad (\text{resp. } p(X_k = k|y_{1:n}) > \varsigma, \quad p(X_k = k|y_{1:k+L}) > \varsigma)$$

- **Latent variables:** By tracking the changes in the latent variables, a decision making procedure can be made based on the common behavioural characteristics of fraud.

4.2. Results

In this section, we present the results of our fraud detection algorithm that we implemented in Python 3.6 and all tests are conducted on a 64-bit server with and Intel Core i7 7700HQ , 2.8 GHz processor and 16 GB RAM running Windows 10 Professional.

We simulated a call data of 15 days for one customer using the values from Table 4.2 with three changepoints. We run three different fraud detection algorithm simultaneously to detect the moment of intrusion.

Table 4.1: inputs for the call generation

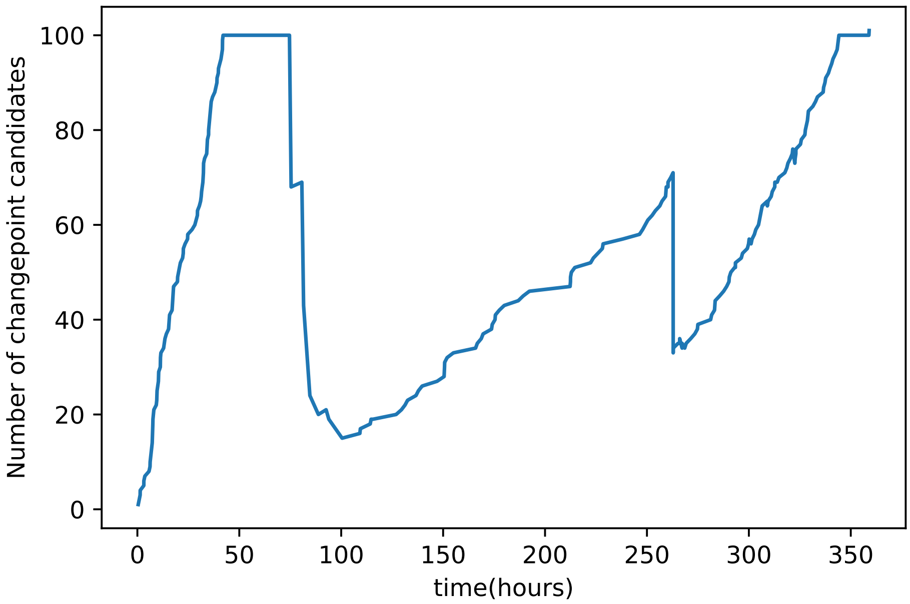
total time, T	$1,296 \times 10^3$ seconds
initial time, t_0	0.00 seconds
changepoint probability, ξ	0.008
n_w	1
n_d	1
M	2
κ^a	2.225
κ^d	2.10
θ^a	1.51×10^{-4}
θ^d	2.5×10^{-4}

α^c	0.1
β^c	0.9
ρ	0.1

Since we are trying to test the performance of our algorithm, we know the locations of the changepoints in advance. According to data test we use, the change point times are at $c_1 = 74.61949727$, $c_2 = 257.39426515$, $c_3 = 259.13771393$ hours.

As we have mentioned in Section 3.3.3, every 'CS' event is a candidate for a changepoint. Hence, the computational cost of calculating filtering densities increases linearly over time. In Section 3.2.5, we represented a resampling method that considers up to N number of 'CS' events with filtering probabilities higher than the threshold value ω as a set of changepoint candidates.

Figure 4.1: Resampling Algorithm



The number of changepoint candidates fluctuating over time when $N = 100$ and $\omega = 10^{-4}$ is shown in Figure 4.1. In the next sections, we compare the performances of filtering, fixed L-lag smoothing and smoothing for the simulated data and explain our method for producing an alert for unusual behavior.

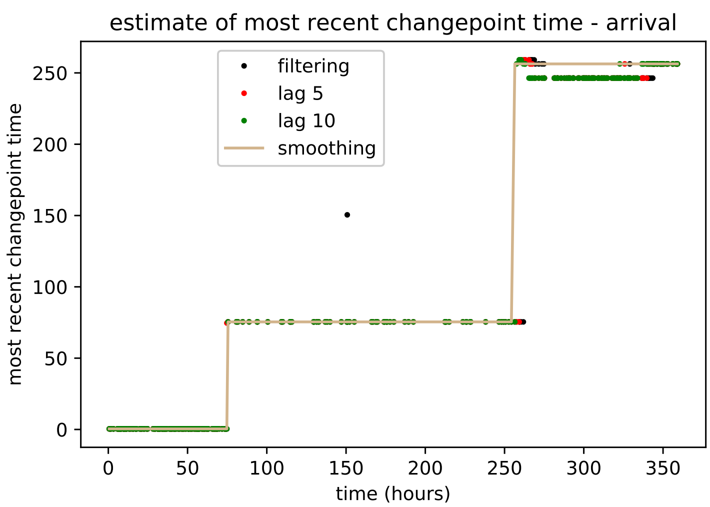
Table 4.2: changepoints for the simulated data

changepoint	time (hours)
c_1	74.62
c_2	257.40
c_3	259.14

4.2.1. Detecting ChangePoints

As one can say intuitively, the more information we possess about the data, the better evaluations we can carry out. For this reason, smoothing densities give best estimations for fraudulent behavior in comparison to filtering and fixed L-lag smoothing. In Figure 4.2, 4.3, and 4.4, event times and the corresponding time our method considers to be the most probable, recent changepoint with respect to $p(X_k = i|y_{1:k})$, $p(X_k = i|y_{1:k+5})$, $p(X_k = i|y_{1:k+10})$ and $p(X_k = i|y_{1:n})$ for call frequency, call duration and call features are given.

Figure 4.2: Estimating last changepoint time for call frequency using filtering, fixed L-lag smoothing and smoothing densities.



An important part of fraud detection algorithm is limiting the number of false alarms which can damage callers' satisfaction considerably. For example, given c_1, c_2,and c_3 in Table 4.2, filtering finds a false changepoint for call frequency around 150'th hour, Figure 4.2, where fixed L-lag smoothing and smoothing perform considerably better. In our algorithm, we aim to minimize the number of false alarms while being quick to detect the deviations from normal behavior. Although smoothing probability gives the best estimation, the delay is high considering the need to collect all the data up to y_n.

Figure 4.3: Estimating last changepoint time for call duration using filtering, fixed L-lag smoothing and smoothing densities.

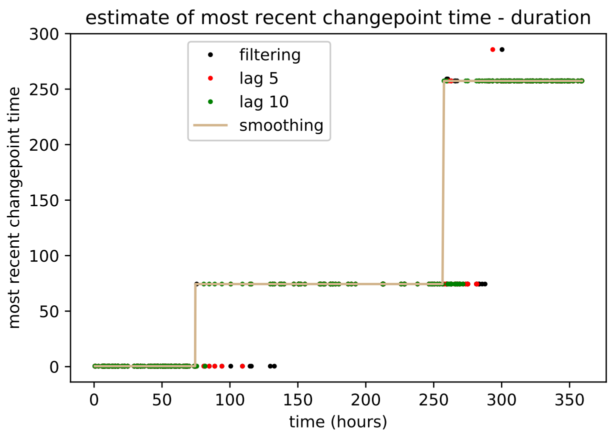
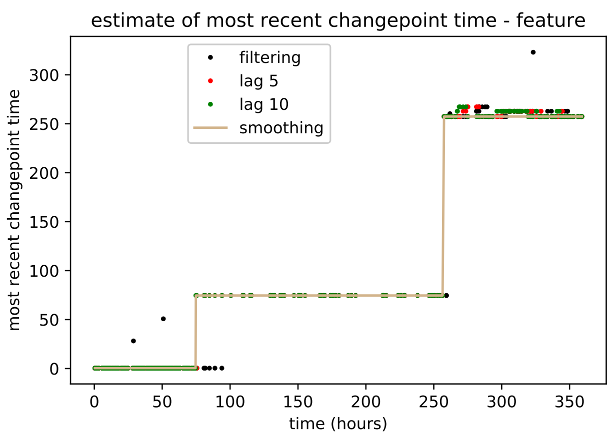


Figure 4.4: Estimating last changepoint time for call features using filtering, fixed L-lag smoothing and smoothing densities.



Fixed L-lag smoothing, compared to filtering, gives improved results for fraud detection while coming L events behind which can be considered as an acceptable delay for our algorithm. In the next section, we introduce another method for catching anomaly in the caller's account.

4.2.2. Tracking Latent Variables

Call behavior in the communication network is defined by the unobserved parameters $\lambda^a_{i,j}$, $\lambda^d_{i,j}$, and π^i which are constant during a regime. After a changepoint, the latent variables are resampled from their respective prior distributions. Estimations for the latent variables when $n_w = 1$, $n_d = 1$ and $M = 2$, $m_i = 2$ for $i = 1, 2$ are represented in Figures 4.5, 4.6, and 4.7.

Figure 4.5: Estimation of call frequency parameters

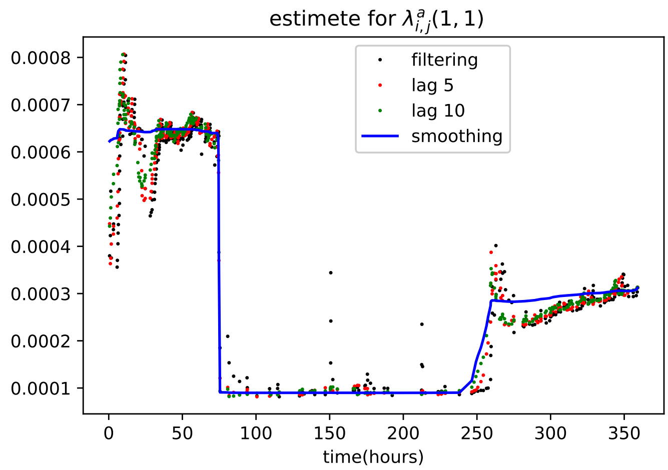


Figure 4.6: Estimation of call duration parameters

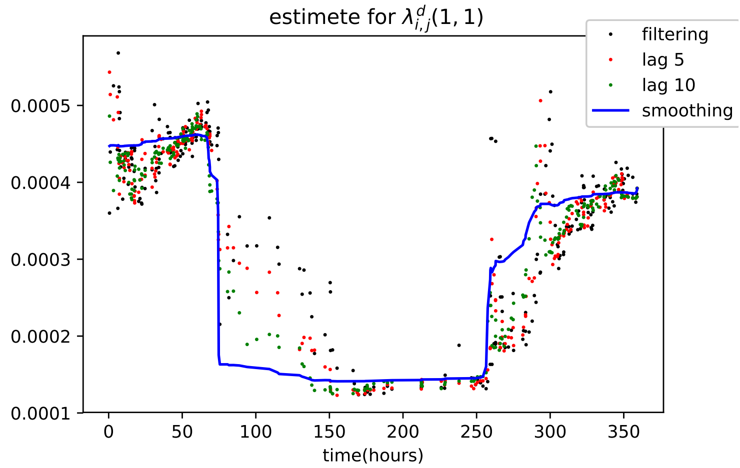
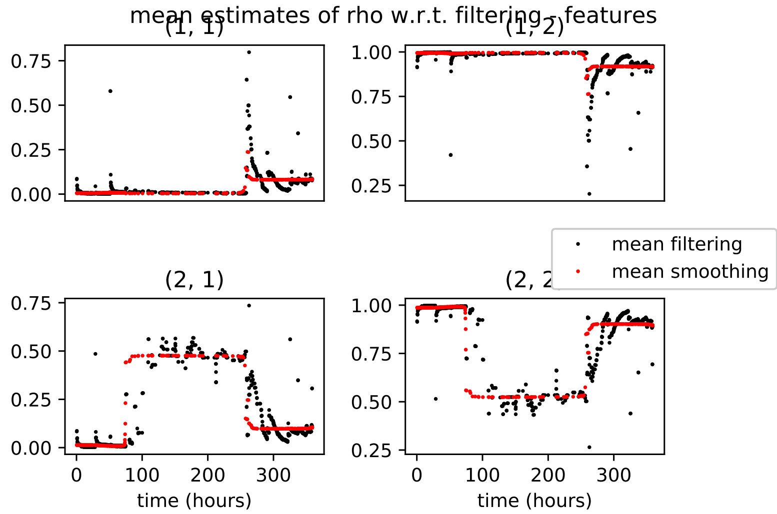


Figure 4.7: Estimation of call features parameters



4.2.3. F-score

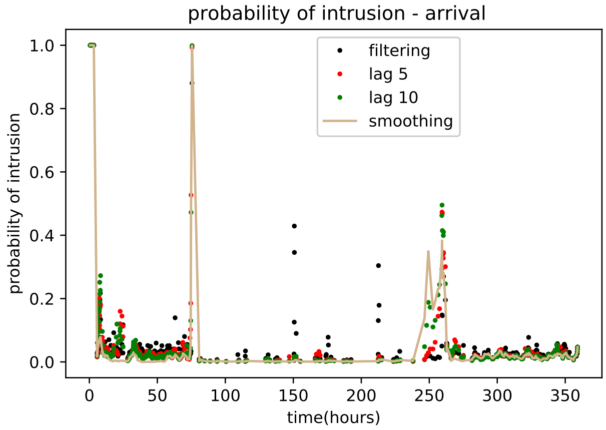
In this section, we introduce F-score which is a measurement we used to test the accuracy of our algorithm. F-score uses both the precision value and recall value of the test.

$$\text{F-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

where precision $= \dfrac{\text{number of true alarms}}{\text{number of alarms}}$ and recall value $= \dfrac{\text{number of true alarms}}{\text{number of changepoints}}$
High precision indicates that the algorithm returned substantially more relevant results than irrelevant ones, while high recall means that the algorithm returned most of the relevant results. F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

When $p(X_k = k|y_{1:k}) > \varsigma \quad or \quad \big(p(X_k = k|y_{1:n}) > \varsigma, \quad p(X_k = k|y_{1:k+L}\big) > \varsigma)$, our algorithm initiates an alarm at time $\overline{t_k}$. If there is a changepoint time c such that $\overline{t_k} - \epsilon \leq c \leq \overline{t_k}$ for some tolerance window ϵ, then we classify it as a true alarm. Alarms which do not fall in a tolerance window of a any changepoint time are considered as false alarms. In Figures 4.8, 4.9 and 4.10, the probabilities of filtering, smoothing and fixed L-lag smoothing are given.

Figure 4.8: Probability of a changepoint in call frequency in the last three hours



For example, when $\varsigma = 0.15$, detection algorithm constructed using filtering probabilities for call arrival finds all changepoints, however it gives false alarms around 7'th, 150'th and 212'th hour. If we set $\varsigma = 0.50$, we would decrease the number of false alarm but no longer detect the changepoints at c_2 and c_3

Figure 4.9: Probability of a changepoint in call duration in the last three hours

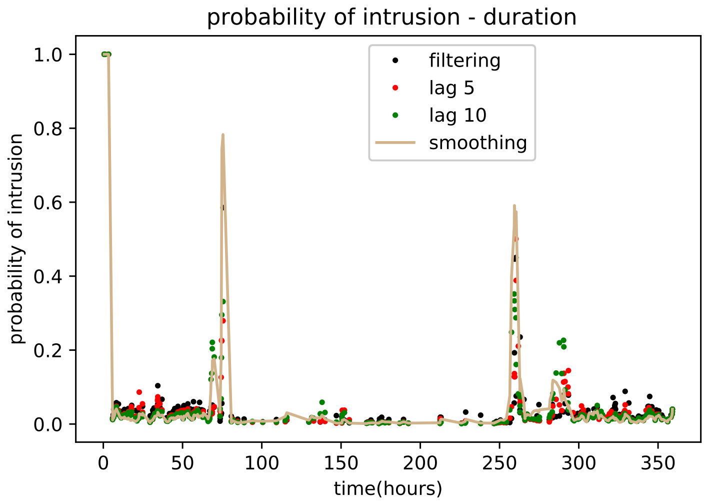
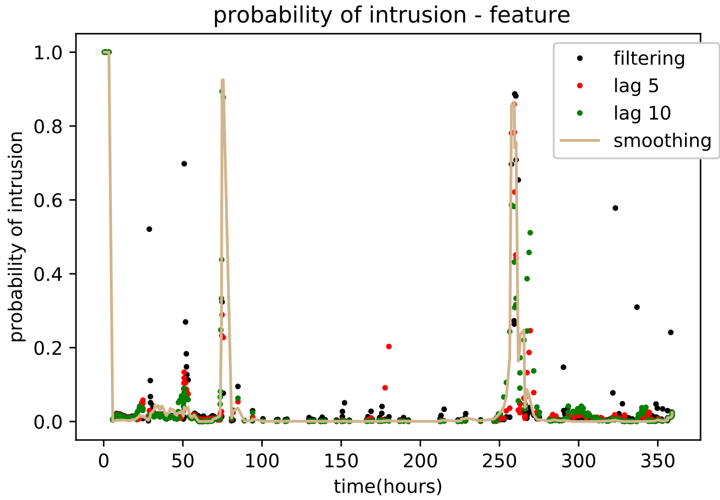


Figure 4.10: Probability of a changepoint in call features in the last three hours



As expected, smoothing probabilities do not fluctuate at non-fraudulent event times and depending on the threshold value number of false and true alarms can vary. In Table 4.3, averages of F-scores of 30 runs that generated from Table 4.2 with respect to different call behavior elements and thresholds are given. Interested readers can find the extensive F-score values in Appendix C

Table 4.3: average F-score sample mean ± sample variance

behavior type	threshold	filtering	fixed lag smoothing		smoothing
			L=5	L=10	
arrival	0.15	0.23 ± 0.063	0.28 ± 0.054	0.29 ± 0.054	0.32 ± 0.041
	0.30	0.17 ± 0.065	0.25 ± 0.084	0.28 ± 0.076	0.31 ± 0.075
	0.50	0.15 ± 0.047	0.23 ± 0.084	0.23 ± 0.0821	0.26 ± 0.071
duration	0.15	0.18 ± 0.045	0.24 ± 0.045	0.27 ± 0.039	0.31 ± 0.036
	0.30	0.16 ± 0.072	0.27 ± 0.065	0.28 ± 0.078	0.29 ± 0.075
	0.50	0.11 ± 0.044	0.21 ± 0.078	0.25 ± 0.077	0.27 ± 0.083
feature	0.15	0.55 ± 0.035	0.62 ± 0.030	0.63 ± 0.032	0.64 ± 0.027
	0.30	0.58 ± 0.031	0.65 ± 0.032	0.67 ± 0.030	0.69 ± 0.027
	0.50	0.60 ± 0.035	0.66 ± 0.032	0.67 ± 0.034	0.70 ± 0.027

From Table 4.3, one can easily see that the mean F-score value for smoothing is always better than the filtering. Also, our algorithm gives significantly better F-score values for call features compared to call frequency and call duration. One interpretation we can make for this result is unlike call frequency and duration; call features are collected at the beginning of each 'CS' event. Hence, a change in the call features is observable at the time of the changepoint in contrast to the other call behaviors.

5. CONCLUSION

Finding fraud in communication network more efficiently would lead to higher revenues and customer satisfaction for telecommunication companies. In this study, we addressed phone fraud detection as a Multiple Changepoint model and build an event-based, real-time Bayesian call fraud detection algorithm which requires collecting the high volume of data continuously. We used forward-backward recursions to find the probability of change [Fearnhead and Liu, 2007]. We deduce that while filtering probabilities calculate the probability of change in real-time, smoothing probabilities give a much consistent estimation giving that they have more information about the caller's observations.

We designed an algorithm that detects change in callers' behavior with the assumption that illegal use in the network is one of the main reasons of deviation from the usual behavior. Our work is extended to three detection algorithms which work separately to catch fraudulent activity in the call network. By doing that, we also consider the fact that call behavior of a customer is not homogeneous. Apart from that, we adapted our problem to an event-based detection fraud detection model where our data is updated as soon as an event occurs. We have created our call data generator which simulates call data considering all the characteristics of a caller.

As mentioned in [Yang and Kuo, 2001], [Fearnhead, 2006], [Fearnhead and Liu, 2007], to overcome the quadratically increasing computational costs of estimating filtering and smoothing probabilities, we implemented a resampling method which basically omits the changepoint candidates with low filtering probability and approximates the filtering and smoothing probabilities while decreasing the computational cost.

For the experimental tests, we randomly generated datasets in Python 3.6 which call traffic followed a non-homogeneous process and user's behavior jump to a new regime with probability ξ. We tested our detection algorithm on simulated data by keeping track of the true and false alarms.. We use F-score as a measurement for our test results and conclude that our detection algorithm catches changepoints resulting from deviations in the call features better.

For future research, an online expectation maximization algorithm can be adapted to our model to approximate hyperparameters and analyze real call data. Also, algorithm can be changed in a way that it can be applicable for multi-user case.

REFERENCES

ACFE, 2016. ACFE (2016). Report to the nations on occupational fraud and abuse.

Barry and Hartigan, 1992. Barry, D. and Hartigan, J. A. (1992). Product partition models for change point problems. *The Annals of Statistics*, 20(1):260–279.

Barry and Hartigan, 1993. Barry, D. and Hartigan, J. A. (1993). A bayesian analysis for change point problems. *Journal of the American Statistical Association*, 88(421):309–319.

Becker et al., 2010. Becker, R. A., Volinsky, C., and Wilks, A. R. (2010). Fraud detection in telecommunications: History and lessons learned. *Technometrics*, 52:20–33.

CFCE, 2015. CFCE (2015). 2015 global fraud loss survey by communications fraud control association.

Davey et al., 1996. Davey, N., Field, S., Frank, R., Barson, P., and McAskey, G. (1996). The detection of fraud in mobile phone networks. *Neural Network World*, 6(4):477–484. Full text of this article is not available in the UHRA.

Fearnhead, 2006. Fearnhead, P. (2006). Exact and efficient bayesian inference for multiple changepoint problems. *Statistics and Computing*, 16(2):203–213.

Fearnhead and Liu, 2007. Fearnhead, P. and Liu, Z. (2007). On-line inference for multiple changepoint problems. 69:589–605.

H. Cahill et al., 2002. H. Cahill, M., Lambert, D., C. Pinheiro, J., and X. Sun, D. (2002). Detecting fraud in the real world. pages 911–929.

Hollmén and Tresp, 1998. Hollmén, J. and Tresp, V. (1998). Call-based fraud detection in mobile communication networks using a hierarchical regime-switching model.

Kurt et al., 2018. Kurt, B., Çağatay Yıldız, Ceritli, T. Y., Sankur, B., and Cemgil, A. T. (2018). A bayesian change point model for detecting sip-based ddos attacks. *Digital Signal Processing*, 77:48 – 62. Digital Signal Processing SoftwareX - Joint Special Issue on Reproducible Research in Signal Processing.

Moreau et al., 1998. Moreau, Y., Preneel, B., Burge, P., Shawe-Taylor, J., Stoermann, C., Ag, S., Cooke, C., and , V. (1998). Novel techniques for fraud detection in mobile telecommunication networks.

Moreau et al., 1997. Moreau, Y., Verrelst, H., and Vandewalle, J. (1997). Detection of mobile phone fraud using supervised neural networks: A first prototype. pages 1065–1070.

Rosset et al., 1999. Rosset, S., Murad, U., Neumann, E., Idan, Y., and Pinkas, G. (1999). Discovery of fraud rules for telecommunications - challenges and solutions.

Scott, 2004. Scott, S. L. (2004). A bayesian paradigm for designing intrusion detection systems. *Computational Statistics Data Analysis*, 45:69–83.

Taniguchi et al., 1998. Taniguchi, M., Haft, M., Hollmén, J., and Tresp, V. (1998). Fraud detection in communication networks using neural and probabilistic methods.

Xing and Girolami, 2007. Xing, D. and Girolami, M. (2007). Employing latent dirichlet allocation for fraud detection in telecommunications. *Pattern Recognition Letters*, 28(13):1727 – 1734.

Yang and Kuo, 2001. Yang, T. Y. and Kuo, L. (2001). Bayesian binary segmentation procedure for a poisson process with multiple changepoints. *Journal of Computational and Graphical Statistics*, 10(4):772–785.

APPENDIX A: DERIVATIONS

A.1. Conjugacy

Throughout this document, we have used $\text{Gamma}_1(\kappa,\theta)$ and $\text{Gamma}_2(a,b)$ to mean two different parametrisations of the Gamma distribution.

(1) The probability density functions of these distributions are given by

$$X \sim \text{Gamma}_1(\kappa,\theta) \rightarrow p(x) = \frac{1}{\Gamma(\kappa)\theta^{\kappa}} x^{\kappa-1} e^{-x/\theta} \quad \text{(A.1)}$$

$$X \sim \text{Gamma}_2(a,b) \rightarrow p(x) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}. \quad \text{(A.2)}$$

Therefore, $\text{Gamma}_1(\kappa,\theta) = \text{Gamma}_2(\kappa,\theta^{-1})$.

(2) The relation between gamma and inverse gamma distributions are given by

$$X \sim \text{Gamma}_1(\kappa,\theta) \rightarrow X \sim \text{Gamma}_2(\kappa,\theta^{-1}) \rightarrow 1/X \sim \mathcal{IG}(\kappa,\theta^{-1})$$

where the pdf of inverse gamma distribution is given by

$$Y \sim \mathcal{IG}(\alpha,\beta) \rightarrow p(y) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} y^{-\alpha-1} e^{-\beta/y}$$

(3) If $X \sim \text{Gamma}_1(\kappa,\theta)$, $\mathbb{E}(X) = \kappa\theta$ and $\text{Var}(X) = \kappa\theta^2$. Furthermore,

$$\mathbb{E}(1/X) = \frac{1}{\theta(\kappa-1)}, \quad \text{Var}(1/X) = \frac{1}{\theta^2(\kappa-1)^2(\kappa-2)} = \mathbb{E}(1/X)^2 \frac{1}{\kappa-2} \quad \text{(A.3)}$$

(4) We are interested in the moments of $1/X$ when X is gamma distributed since in the model interarrival times as well as call durations are exponentially distributed with rates that are gamma distributed. and the mean of $\text{Exp}_1(\lambda)$ is $1/\lambda$.

(5) We can adjust the variability of the expected values of those durations across regimes between changepoints by playing with the shape and scale parameters of

the Gamma distribution for rate, the hyperparameters of the model. If desired expected value and standard deviation of those durations are μ and σ, then, using (A.3), we would have

$$\kappa = \frac{\mu^2}{\sigma^2} + 2, \quad \theta = \frac{1}{\mu(\kappa - 1)}$$

(6) If $X \sim \text{Gamma}_1(\kappa, \theta)$ and $N \sim \mathcal{PO}(X)$, then

$$\mathbb{P}(N = n) = \frac{\Gamma(\kappa + n)}{\Gamma(\kappa)n!} \left(\frac{\theta}{\theta + 1}\right)^n \left(\frac{1}{\theta + 1}\right)^\kappa \tag{A.4}$$

$$= \frac{\Gamma(\kappa + n)}{\Gamma(\kappa)n!} \frac{\theta^n}{(1 + \theta)^{n+\kappa}} \tag{A.5}$$

This is because $X \sim \text{Gamma}_2(\kappa, \theta^{-1})$ and we defer the reader to the wikipedia page noting that when we solve for $(1 - p)/p = \theta^{-1}$ we get $p = 1/(\theta^{-1} + 1)$.

(7) `python` and `MATLAB` use the parametrisation $\Gamma_1(a, b)$ and $\text{Exp}_2(\mu) = \text{Exp}_1(1/\mu)$

We will use the proposition below.

If $p_1, \ldots, p_k \sim \text{Dir}(\alpha_1, \ldots, \alpha_k)$ and $x_1, \ldots, x_k \sim \text{Mult}(n; p_1, \ldots, p_k)$. .... where $\mathcal{DM}$ denotes the Dirichlet-Multinomial distribution, whose probability distribution is given by

$$\mathcal{DM}\left((x_1, \ldots, x_K); \rho, n\right) = \frac{n!\Gamma(\sum_{k=1}^K \rho_k)}{\Gamma(n + \sum_k \rho_k)} \prod_{k=1}^K \frac{\Gamma(x_k + \rho_k)}{(x_k!)\Gamma(\rho_k)}$$

APPENDIX B: Data Generator

In order to represent our model fully, we simulate our events for a single user according to our depiction of call events in Section 3.3.1.

Inputs:

- hyperparameters: $\kappa^a_{i,j} = \kappa^a$, $\theta^a_{i,j} = \theta^a$, $\kappa^d_{i,j} = \kappa^d$, $\theta^d_{i,j} = \theta^d$, $\rho^{(i)}_j = \rho$, α^c, β^c
- initial time: t_0
- total time: T
- intervals of the day: vector of $1 \times n_d$ that shows the beginning times of the every interval in the day.
- changepoint probability ξ
- class sizes: vector of $1 \times M$ that shows number of categories for corresponding feature.
- checking time for call progress: t_{CP}
- checking time for no call: t_{NC}

Outputs: past events

At the beginning of the simulation, we set an empty vector called active calls and set number of active calls to 0. We start our algorithm with adding events [$t_0 + t_{NC}$, 'NC'], 'WC' and 'DC' events with their times showing the change in the interval.

- **Initialization**
  - number of active calls = 0,
  - active calls = [ ],
  - call number = 1
  - future events = [[T,"END]],
  - find the period (w, c), t_0 belongs to
  - $t = t_0$
  - schedule no call event [$t + t_{NC}$, 'NC'] and add to future events

- t_{WC}: time of the next weekday/weekend change event
- t_{DC}: time of the next day interval change event
- add [t_{DC}, 'DC'] and [t_{WC}, 'WC'] to future events.
- get parameters $\lambda^a_{i,j} \sim \text{Gamma}_1(\kappa^a_{i,j}, \theta^a_{i,j}), \lambda^d_{i,j} \sim \text{Gamma}_1(\kappa^d_{i,j}, \theta^d_{i,j})$ for all $i = 1, \ldots, n_w$, $j = 1, \ldots, n_d$, $\pi^f \sim \text{Dir}(\rho^f)$, $f = 1, \ldots, M$ and $\gamma \sim \text{Beta}(\alpha^c, \beta^c)$
- sample interarrival time $a_t \sim Exp(\lambda^a_{i,j})$
- Schedule a call start by adding event [$t + a_t$, 'CS', call number]

• **while** $t < T$

  - pick the earliest event from future events call it current event
  - event type = current event [1] (type of current event)
  - t = current event[0] (time of current event)
  - process the event

Process of Events

• if event type = 'CS'

  - sample feature vector cat from π^i
  - current event ← current event + cat
  - future events ← future events + [$t + t_{CP}$, 'CP', call number]
  - $u \sim U(0, 1)$.
  - if $u \leq \xi$
    * update parameters $\lambda^a_{i,j} \sim \text{Gamma}_1(\kappa^a_{i,j}, \theta^a_{i,j}), \lambda^d_{i,j} \sim \text{Gamma}_1(\kappa^d_{i,j}, \theta^d_{i,j})$ for all $i = 1, \ldots, n_w$, $j = 1, \ldots, n_d$, $\pi^f \sim \text{Dir}(\rho^f)$, $f = 1, \ldots, M$ and $\gamma \sim \text{Beta}(\alpha^c, \beta^c)$
  - $u \sim U(0, 1)$.
  - if $u \leq \gamma$
  - future events ← [future events + [t, 'CE', call number]]
  - else
  - active calls ← [active calls , call number]
  - number of active calls ← number of active calls + 1

  - sample duration time $d_t \sim Exp(\lambda^d_{i,j})$
  - future events ← [future events + [$t + d_t$, 'CE', call number]]
- add current event to past event

- if event type = 'WC'
  - change i, move to weekday or weekend
  - add next week change event to future events meaning future events ← [future events,[t_{WC}, 'WC']]
  - add current event to past events

- if event type = 'DC'
  - change j, move to next day interval
  - add day interval change event to future events meaning future events ← [future events,[t_{DC}, 'DC']]
  - add current event to past events

- if event type = 'CE'
  - call number = current event [3]
  - remove call number from active calls vector
  - number of active calls ← number of active calls -1
  - add current event to past events

- if event type = 'NC'
  - if number of active calls = 0
    - add current event to past events

- if event type = 'CP'
  - if current event in active calls
    - add current event to past events

Notation

- N: number of calls arriving from time 0 until time t.
- $N^a_{i,j}$: number of calls arriving in an (i, j) interval
- $N^p_{i,j}$: number of successful calls in progress that have visited (i, j)-type interval
- $N^e_{i,j}(t_1, t_2)$: number of successful calls ending in an (i, j) interval between time t_1 and time t_2.
- N^u: number of unsuccessful calls.
- $\tau_{i,j}$: time spent in the (i, j) type intervals.
- $D^p_{i,j}$: Vector (of size $N^p_{i,j} \times 1$) of times spent in (i, j) intervals for ongoing calls in the (i, j) intervals.
- $\mathrm{ID}^p_{i,j}$: The vector of call ID's of the calls in $D^p_{i,j}$.
- $D^e_{i,j}$: Vector (of size $N^e_{i,j} \times 1$) of times spent in (i, j) intervals for ending calls in the (i, j) intervals.
- $I_{i,j}$: Set of times for the (i, j)-type intervals.

B.1. Bookkeeping for the sufficient statistics

For each event type, the content of the event and the list of actions in order to update the observation density are given below:

- `Call Start (CS)`: the event is of the form

```
[CS, t, Call ID, feature vector, Call success]
```

The actions to be taken in a `CS` event at the time t_k of the k'th event:

- For every `index` in $\{1, \ldots, N^p_{i,j}\}$, if $\mathrm{ID}^p_{i,j}[\texttt{index}] \in \mathrm{ID}^p$

$$D^p_{i,j}[\texttt{index}] \leftarrow D^p_{i,j}[\texttt{index}] + t_k - t_{k-1}$$

- $N \leftarrow N + 1$
- $\tau_{i,j} \leftarrow \tau_{i,j} + t_k - t_{k-1}$
- $N^a_{i,j} \leftarrow N^a_{i,j} + 1$

– Check the call status:

  * If call is unsuccessful, then

    · $N^u \leftarrow N^u + 1$;

  * If call is successful, then $\mathrm{ID}^p \leftarrow [\mathrm{ID}^p \quad \texttt{Call ID}]$. Furthermore, for all i, j

    · $\mathrm{ID}^p_{i,j} \leftarrow [\mathrm{ID}^p_{i,j} \quad \texttt{Call ID}]$

    · $D^p_{i,j} \leftarrow [D^p_{i,j} \quad 0]$

    · $N^p_{i,j} \leftarrow N^p_{i,j} + 1$

– Update the counts in the feature vectors:

$$c^{(i)}_{\texttt{feature vector[i]}} \leftarrow c^{(i)}_{\texttt{feature vector[i]}} + 1, \quad i = 1, \ldots, n_f$$

• `Call End (CE)`: the event is of the form

`[CE, t, Call ID]`

The actions to be taken in a `CE` event at the time t_k of the k'th event:

– Find the element valued `Call ID` in ID^p and $\mathrm{ID}^p_{i,j}$, set the indices to `temp index` and `temp index local`

– Remove the `temp index`'th element from ID^p

– Remove the `temp index local`'th element from $\mathrm{ID}^p_{i,j}$

– Remove `temp index local`'th element from $D^p_{i,j}$, keep the value of the element as `temp duration`

– $N^p_{i,j} \leftarrow N^p_{i,j} - 1$

– $D^e_{i,j} \leftarrow [D^e_{i,j} \quad \texttt{temp duration} + t_k - t_{k-1}]$

– $N^e_{i,j} \leftarrow N^e_{i,j} + 1$

– $\tau_{i,j} \leftarrow \tau_{i,j} + t_k - t_{k-1}$

– For every `index` in $\{1, \ldots, N^p_{i,j}\}$, if $\mathrm{ID}^p_{i,j}[\texttt{index}] \in \mathrm{ID}^p$

$$D^p_{i,j}[\texttt{index}] \leftarrow D^p_{i,j}[\texttt{index}] + t_k - t_{k-1}$$

- `Call in progress (CP)`: the event is of the form

  `[CP, t, Call ID]`

  The actions to be taken in a `CP` event at the time t_k of the k'th event:

  - For every `index` in $\{1, \ldots, N^p_{i,j}\}$, if $\mathrm{ID}^p_{i,j}[\texttt{index}] \in \mathrm{ID}^p$

  $$D^p_{i,j}[\texttt{index}] \leftarrow D^p_{i,j}[\texttt{index}] + t_k - t_{k-1}$$

  - $\tau_{i,j} \leftarrow \tau_{i,j} + t_k - t_{k-1}$

- `No Call (NC)`: the event is of the form

  `[NC, t]`

  The actions to be taken are

  - For every `index` in $\{1, \ldots, N^p_{i,j}\}$, if $\mathrm{ID}^p_{i,j}[\texttt{index}] \in \mathrm{ID}^p$

  $$D^p_{i,j}[\texttt{index}] \leftarrow D^p_{i,j}[\texttt{index}] + t_k - t_{k-1}$$

  - $\tau_{i,j} \leftarrow \tau_{i,j} + t_k - t_{k-1}$

- `weekday/weekend change (WC)`: the event is of the form

  `[NC, t]`

  The actions to be taken are

  - For every `index` in $\{1, \ldots, N^p_{i,j}\}$, if $\mathrm{ID}^p_{i,j}[\texttt{index}] \in \mathrm{ID}^p$

  $$D^p_{i,j}[\texttt{index}] \leftarrow D^p_{i,j}[\texttt{index}] + t_k - t_{k-1}$$

  - $\tau_{i,j} \leftarrow \tau_{i,j} + t_k - t_{k-1}$

- We update the current interval:

$$(i, j) \leftarrow (i \bmod n_w + 1, j)$$

- `change in period of the day (DC)`: the event is of the form

```
[DC, t]
```

  The actions to be taken are

  - For every `index` in $\{1, \ldots, N^p_{i,j}\}$, if $\text{ID}^p_{i,j}[\texttt{index}] \in \text{ID}^p$

$$D^p_{i,j}[\texttt{index}] \leftarrow D^p_{i,j}[\texttt{index}] + t_k - t_{k-1}$$

  - $\tau_{i,j} \leftarrow \tau_{i,j} + t_k - t_{k-1}$
  - We update the current interval:

$$(i, j) \leftarrow (i, j \bmod n_d + 1)$$

APPENDIX C: Detailed Results for Algorithm

Table C.1: F-score for arrival, threshold = 0.15

run no.	number of events	number of calls	number of changepoints	F-Scores: filtering	fixed lag smoothing L=5	fixed lag smoothing L=10	smoothing
1	473	184	2	0.67	0.44	0.57	0.57
2	1641	586	3	0.50	0.40	0.44	0.33
3	880	298	3	0.67	0.50	0.57	0.60
4	947	378	3	0.00	0.00	0.00	0.36
5	1433	504	1	0.33	0.40	0.40	0.50
6	969	341	4	0.00	0.29	0.25	0.18
7	562	200	5	0.50	0.40	0.33	0.40
8	902	333	5	0.44	0.00	0.00	0.25
9	856	367	2	0.00	0.00	0.00	0.00
10	1045	399	4	0.00	0.33	0.29	0.40
11	1907	689	7	0.25	0.53	0.40	0.44
12	1470	542	2	0.25	0.22	0.18	0.18
13	673	230	2	0.00	0.00	0.00	0.00
14	860	344	3	0.60	0.60	0.60	0.60
15	536	184	2	0.29	0.67	0.57	0.44
16	1037	410	0	0.00	0.00	0.00	0.00
17	1278	479	2	0.00	0.00	0.00	0.00
18	846	367	7	0.17	0.00	0.15	0.00
19	727	239	1	0.00	0.40	0.40	0.40
20	1067	383	4	0.29	0.29	0.40	0.46
21	322	99	2	0.00	0.50	0.40	0.40
22	1199	458	2	0.40	0.29	0.00	0.29
23	1077	401	6	0.18	0.18	0.33	0.35
24	878	461	2	0.00	0.00	0.29	0.33
25	1122	413	1	0.00	0.00	0.00	0.00
26	709	228	1	0.00	0.50	0.67	0.40
27	461	152	1	0.67	0.50	0.50	0.50
28	947	363	6	0.62	0.71	0.71	0.71
29	1413	490	2	0.00	0.25	0.29	0.25
30	675	323	2	0.00	0.00	0.00	0.33

Table C.2: F-score for arrival, threshold = 0.30

run no.	number of events	number of calls	number of changepoints	F-Scores filtering	fixed lag smoothing L=5	fixed lag smoothing L=10	smoothing
1	473	184	2	0.67	0.44	0.57	0.57
2	1641	586	3	0.29	0.50	0.50	0.44
3	880	298	3	0.40	0.67	0.40	0.67
4	947	378	3	0.00	0.00	0.00	0.00
5	1433	504	1	0.50	0.50	0.50	0.67
6	969	341	4	0.00	0.33	0.33	0.25
7	562	200	5	0.00	0.44	0.50	0.55
8	902	333	5	0.00	0.00	0.00	0.00
9	856	367	2	0.00	0.00	0.00	0.00
10	1045	399	4	0.00	0.00	0.00	0.22
11	1907	689	7	0.00	0.36	0.33	0.43
12	1470	542	2	0.33	0.29	0.29	0.33
13	673	230	2	0.00	0.00	0.00	0.00
14	860	344	3	0.67	0.86	0.86	0.75
15	536	184	2	0.40	0.80	0.80	0.80
16	1037	410	0	0.00	0.00	0.00	0.00
17	1278	479	2	0.00	0.00	0.00	0.00
18	846	367	7	0.00	0.00	0.00	0.00
19	727	239	1	0.00	0.50	0.50	0.50
20	1067	383	4	0.33	0.44	0.44	0.40
21	322	99	2	0.00	0.00	0.50	0.40
22	1199	458	2	0.00	0.00	0.00	0.00
23	1077	401	6	0.25	0.25	0.40	0.40
24	878	461	2	0.00	0.00	0.40	0.40
25	1122	413	1	0.00	0.00	0.00	0.00
26	709	228	1	0.00	0.00	0.00	0.00
27	461	152	1	0.67	0.50	0.50	0.50
28	947	363	6	0.72	0.72	0.67	0.72
29	1413	490	2	0.00	0.00	0.00	0.00
30	675	323	2	0.00	0.00	0.00	0.33

Table C.3: F-score for arrival, threshold = 0.5

run no.	number of events	number of calls	number of changepoints	F-Scores: filtering	F-Scores: fixed lag smoothing L=5	F-Scores: fixed lag smoothing L=10	F-Scores: smoothing
1	473	184	2	0.40	0.57	0.57	0.57
2	1641	586	3	0.40	0.67	0.57	0.57
3	880	298	3	0.40	0.40	0.40	0.40
4	947	378	3	0.00	0.00	0.00	0.00
5	1433	504	1	0.50	0.67	0.67	0.67
6	969	341	4	0.00	0.00	0.00	0.29
7	562	200	5	0.00	0.29	0.50	0.50
8	902	333	5	0.00	0.00	0.00	0.00
9	856	367	2	0.00	0.00	0.00	0.00
10	1045	399	4	0.00	0.00	0.00	0.00
11	1907	689	7	0.00	0.00	0.33	0.33
12	1470	542	2	0.50	0.40	0.33	0.40
13	673	230	2	0.00	0.00	0.00	0.00
14	860	344	3	0.40	0.40	0.40	0.40
15	536	184	2	0.50	0.80	0.80	0.50
16	1037	410	0	0.00	0.00	0.00	0.00
17	1278	479	2	0.00	0.00	0.00	0.00
18	846	367	7	0.00	0.00	0.00	0.00
19	727	239	1	0.00	0.50	0.50	0.50
20	1067	383	4	0.33	0.57	0.50	0.50
21	322	99	2	0.00	0.00	0.00	0.50
22	1199	458	2	0.00	0.00	0.00	0.00
23	1077	401	6	0.25	0.25	0.00	0.00
24	878	461	2	0.00	0.00	0.00	0.00
25	1122	413	1	0.00	0.00	0.00	0.00
26	709	228	1	0.00	0.00	0.00	0.00
27	461	152	1	0.67	0.67	0.67	0.67
28	947	363	6	0.25	0.72	0.72	0.72
29	1413	490	2	0.00	0.00	0.00	0.00
30	675	323	2	0.00	0.00	0.00	0.00

Table C.4: F-score for duration, threshold = 0.15

run no.	number of events	number of calls	number of changepoints	F-Scores			
				filtering	fixed lag smoothing		smoothing
					L=5	L=10	
1	473	184	2	0.00	0.00	0.29	0.33
2	1641	586	3	0.33	0.40	0.33	0.33
3	880	298	3	0.00	0.29	0.29	0.31
4	947	378	3	0.00	0.00	0.25	0.25
5	1433	504	1	0.29	0.40	0.40	0.40
6	969	341	4	0.60	0.44	0.44	0.44
7	562	200	5	0.29	0.25	0.22	0.25
8	902	333	5	0.33	0.57	0.57	0.57
9	856	367	2	0.00	0.33	0.25	0.33
10	1045	399	4	0.25	0.36	0.36	0.31
11	1907	689	7	0.33	0.50	0.33	0.50
12	1470	542	2	0.14	0.18	0.44	0.50
13	673	230	2	0.33	0.25	0.17	0.25
14	860	344	3	0.75	0.75	0.60	0.67
15	536	184	2	0.00	0.00	0.00	0.00
16	1037	410	0	0.00	0.00	0.00	0.00
17	1278	479	2	0.00	0.00	0.00	0.00
18	846	367	7	0.22	0.00	0.50	0.20
19	727	239	1	0.00	0.40	0.40	0.40
20	1067	383	4	0.00	0.25	0.00	0.20
21	322	99	2	0.00	0.00	0.00	0.33
22	1199	458	2	0.44	0.44	0.50	0.57
23	1077	401	6	0.00	0.31	0.33	0.40
24	878	461	2	0.00	0.00	0.00	0.00
25	1122	413	1	0.00	0.00	0.50	0.40
26	709	228	1	0.50	0.29	0.50	0.67
27	461	152	1	0.00	0.00	0.00	0.00
28	947	363	6	0.33	0.43	0.14	0.17
29	1413	490	2	0.25	0.29	0.33	0.29
30	675	323	2	0.00	0.00	0.00	0.33

Table C.5: F-score for duration, threshold = 0.30

run no.	number of events	number of calls	number of changepoints	F-Scores: filtering	fixed lag smoothing L=5	fixed lag smoothing L=10	smoothing
1	473	184	2	0.00	0.00	0.00	0.40
2	1641	586	3	0.44	0.44	0.50	0.50
3	880	298	3	0.00	0.00	0.00	0.29
4	947	378	3	0.00	0.00	0.29	0.40
5	1433	504	1	0.67	0.50	0.50	0.67
6	969	341	4	0.50	0.50	0.50	0.50
7	562	200	5	0.00	0.25	0.25	0.25
8	902	333	5	0.44	0.44	0.67	0.62
9	856	367	2	0.00	0.50	0.50	0.50
10	1045	399	4	0.33	0.50	0.50	0.36
11	1907	689	7	0.43	0.43	0.43	0.53
12	1470	542	2	0.00	0.33	0.25	0.33
13	673	230	2	0.00	0.50	0.33	0.00
14	860	344	3	0.86	0.67	0.86	0.75
15	536	184	2	0.00	0.00	0.00	0.00
16	1037	410	0	0.00	0.00	0.00	0.00
17	1278	479	2	0.00	0.00	0.00	0.00
18	846	367	7	0.00	0.00	0.00	0.00
19	727	239	1	0.00	0.50	0.50	0.50
20	1067	383	4	0.00	0.00	0.00	0.00
21	322	99	2	0.00	0.00	0.00	0.00
22	1199	458	2	0.57	0.80	0.80	0.80
23	1077	401	6	0.00	0.40	0.40	0.44
24	878	461	2	0.00	0.00	0.00	0.00
25	1122	413	1	0.00	0.00	0.00	0.00
26	709	228	1	0.67	0.50	0.67	0.67
27	461	152	1	0.00	0.00	0.00	0.00
28	947	363	6	0.00	0.36	0.00	0.00
29	1413	490	2	0.00	0.40	0.40	0.33
30	675	323	2	0.00	0.00	0.00	0.00

Table C.6: F-score for duration, threshold = 0.50

run no.	number of events	number of calls	number of changepoints	F-Scores: filtering	F-Scores: fixed lag smoothing L=5	F-Scores: fixed lag smoothing L=10	F-Scores: smoothing
1	473	184	2	0.00	0.00	0.00	0.00
2	1641	586	3	0.29	0.67	0.67	0.40
3	880	298	3	0.00	0.00	0.00	0.33
4	947	378	3	0.00	0.00	0.00	0.00
5	1433	504	1	0.67	0.50	0.50	0.67
6	969	341	4	0.33	0.57	0.57	0.57
7	562	200	5	0.00	0.29	0.25	0.25
8	902	333	5	0.29	0.50	0.67	0.67
9	856	367	2	0.00	0.50	0.50	0.50
10	1045	399	4	0.00	0.00	0.29	0.33
11	1907	689	7	0.55	0.50	0.46	0.36
12	1470	542	2	0.00	0.40	0.40	0.40
13	673	230	2	0.00	0.00	0.50	0.00
14	860	344	3	0.40	0.67	0.00	0.86
15	536	184	2	0.00	0.00	0.00	0.00
16	1037	410	0	0.00	0.00	0.00	0.00
17	1278	479	2	0.00	0.00	0.00	0.00
18	846	367	7	0.00	0.00	0.00	0.00
19	727	239	1	0.00	0.00	0.50	0.50
20	1067	383	4	0.00	0.00	0.00	0.00
21	322	99	2	0.00	0.00	0.00	0.00
22	1199	458	2	0.67	0.80	0.80	0.80
23	1077	401	6	0.00	0.00	0.44	0.44
24	878	461	2	0.00	0.00	0.00	0.00
25	1122	413	1	0.00	0.00	0.00	0.00
26	709	228	1	0.00	0.67	0.67	0.67
27	461	152	1	0.00	0.00	0.00	0.00
28	947	363	6	0.00	0.25	0.00	0.00
29	1413	490	2	0.00	0.00	0.40	0.40
30	675	323	2	0.00	0.00	0.00	0.00

Table C.7: F-score for feature, threshold = 0.15

run no.	number of events	number of calls	number of changepoints	F-Scores: filtering	fixed lag smoothing L=5	fixed lag smoothing L=10	smoothing
1	473	184	2	0.40	0.67	0.57	0.57
2	1641	586	3	0.33	0.55	0.50	0.46
3	880	298	3	0.44	0.57	0.57	0.67
4	947	378	3	0.60	0.60	0.67	0.67
5	1433	504	1	0.33	0.40	0.40	0.67
6	969	341	4	0.80	0.80	0.89	0.80
7	562	200	5	0.83	0.83	0.83	0.83
8	902	333	5	0.57	0.77	0.77	0.77
9	856	367	2	0.80	0.80	0.80	0.80
10	1045	399	4	0.55	0.67	0.67	0.67
11	1907	689	7	0.56	0.71	0.67	0.71
12	1470	542	2	0.67	0.80	0.80	0.80
13	673	230	2	0.40	0.40	0.33	0.40
14	860	344	3	0.50	0.60	0.55	0.60
15	536	184	2	0.50	0.40	0.67	0.80
16	1037	410	0	0.00	0.00	0.00	0.00
17	1278	479	2	0.80	0.80	0.80	0.57
18	846	367	7	0.67	0.71	0.67	0.75
19	727	239	1	0.67	0.67	0.67	0.67
20	1067	383	4	0.57	0.67	0.67	0.73
21	322	99	2	0.80	0.50	0.50	0.50
22	1199	458	2	0.33	0.50	0.50	0.50
23	1077	401	6	0.67	0.71	0.80	0.71
24	878	461	2	0.67	0.80	0.80	0.80
25	1122	413	1	0.29	0.50	0.67	0.67
26	709	228	1	0.40	0.67	0.67	0.67
27	461	152	1	0.67	0.67	0.67	0.67
28	947	363	6	0.53	0.73	0.73	0.73
29	1413	490	2	0.50	0.57	0.57	0.57
30	675	323	2	0.57	0.50	0.44	0.67

Table C.8: F-score for feature, threshold = 0.30

run no.	number of events	number of calls	number of changepoints	F-Scores filtering	fixed lag smoothing L=5	fixed lag smoothing L=10	smoothing
1	473	184	2	0.40	0.40	0.67	0.67
2	1641	586	3	0.50	0.67	0.75	0.86
3	880	298	3	0.25	0.57	0.57	0.67
4	947	378	3	0.75	0.75	0.75	0.75
5	1433	504	1	0.50	0.50	0.67	0.67
6	969	341	4	0.89	0.89	0.89	0.89
7	562	200	5	0.83	0.83	0.83	0.83
8	902	333	5	0.67	0.67	0.77	0.77
9	856	367	2	0.80	0.80	0.80	0.80
10	1045	399	4	0.60	0.67	0.67	0.67
11	1907	689	7	0.59	0.71	0.71	0.71
12	1470	542	2	0.67	0.80	0.80	0.80
13	673	230	2	0.40	0.40	0.40	0.50
14	860	344	3	0.55	0.67	0.67	0.75
15	536	184	2	0.50	0.40	0.40	0.80
16	1037	410	0	0.00	0.00	0.00	0.00
17	1278	479	2	0.80	0.80	0.80	0.80
18	846	367	7	0.67	0.77	0.77	0.86
19	727	239	1	0.67	0.67	0.67	0.67
20	1067	383	4	0.73	0.73	0.73	0.73
21	322	99	2	0.50	0.50	0.50	0.50
22	1199	458	2	0.50	0.50	0.50	0.50
23	1077	401	6	0.67	0.77	0.80	0.62
24	878	461	2	0.67	0.80	0.80	0.80
25	1122	413	1	0.50	0.67	0.67	0.67
26	709	228	1	0.50	0.67	0.67	0.67
27	461	152	1	0.67	0.67	0.67	0.67
28	947	363	6	0.57	0.73	0.73	0.73
29	1413	490	2	0.57	0.67	0.57	0.67
30	675	323	2	0.57	0.80	0.80	0.80

Table C.9: F-score for feature, threshold = 0.50

run no.	number of events	number of calls	number of changepoints	F-Scores			
				filtering	fixed lag smoothing L=5	fixed lag smoothing L=10	smoothing
1	473	184	2	0.40	0.40	0.67	0.67
2	1641	586	3	0.67	0.57	0.33	0.67
3	880	298	3	0.29	0.57	0.67	0.67
4	947	378	3	0.57	0.75	0.75	0.75
5	1433	504	1	0.50	0.50	0.67	0.67
6	969	341	4	0.89	0.89	0.89	0.89
7	562	200	5	0.83	0.83	0.83	0.83
8	902	333	5	0.80	0.80	0.91	0.83
9	856	367	2	0.80	0.80	0.80	0.80
10	1045	399	4	0.67	0.67	0.67	0.67
11	1907	689	7	0.71	0.71	0.71	071
12	1470	542	2	0.67	0.80	0.80	0.80
13	673	230	2	0.40	0.40	0.40	0.50
14	860	344	3	0.44	0.67	0.75	0.86
15	536	184	2	0.50	0.50	0.50	0.80
16	1037	410	0	0.00	0.00	0.00	0.00
17	1278	479	2	0.80	0.80	0.80	0.80
18	846	367	7	0.67	0.77	0.77	0.77
19	727	239	1	0.67	0.67	0.67	0.67
20	1067	383	4	0.80	0.73	0.73	0.73
21	322	99	2	0.50	0.50	0.50	0.50
22	1199	458	2	0.50	0.50	0.50	0.50
23	1077	401	6	0.83	0.77	0.77	0.67
24	878	461	2	0.50	0.80	0.80	0.80
25	1122	413	1	0.50	0.67	0.67	0.67
26	709	228	1	0.50	0.67	0.67	0.67
27	461	152	1	0.67	0.67	0.67	0.67
28	947	363	6	0.67	0.73	0.73	0.73
29	1413	490	2	0.57	0.80	0.67	0.80
30	675	323	2	0.57	0.80	0.80	0.80