Statistica Sinica Preprint No: SS-2024-0336	
Title	On Robust Clustering of Event Stream Data
Manuscript ID	SS-2024-0336
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202024.0336
Complete List of Authors	Yuecheng Zhang, Guanhua Fang and Wen Yu
Corresponding Authors	Guanhua Fang
E-mails	fanggh@fudan.edu.cn
Notice: Accepted author version.	

ON ROBUST CLUSTERING OF EVENT STREAM DATA

Yuecheng Zhang Guanhua Fang Wen Yu

Fudan University

Abstract: Clustering of event stream data is of great importance in many application scenarios, including but not limited to, e-commerce, electronic health, online testing, mobile music service, etc. Existing clustering algorithms fail to take outlier data into consideration and are implemented without theoretical guarantees. In this paper, we propose a *robust* temporal point processes clustering framework that works under mild assumptions and meanwhile addresses several important issues in the event stream clustering problem. Specifically, we introduce a computationally efficient model-free distance function to quantify the dissimilarity between different event streams so that the outliers can be detected and the good initial clusters could be obtained. We further propose a classification algorithm incorporated with a Catoni's influence function for robust estimation and fine-tuning of clusters. We also establish the theoretical results including algorithmic convergence, estimation error bound, outlier detection, etc. Simulation results corroborate our theoretical findings and real data applications show the effectiveness of our proposed methodology.

Key words and phrases: Catoni estimator, Event stream, Initialization, Outliers.

1. Introduction

In recent applications, many real-world data can be characterized by time-stamped event sequences/streams. For example, in e-commerce (Xu et al., 2014), the actions taken by a customer in purchasing and viewing the items on the website can form an event sequence. In electronic health (Enguehard et al., 2020), the messages sent by a patient through an AI medical assistant can be viewed as a sequence of events. In online testing (Xu et al., 2018), the students take steps to complete the complex problem-solving questions on the computer, and their response history can be treated as an event stream. In mobile music service (Carneiro et al., 2011), the users can search and play different song tracks and their listening history will be recorded and hence be treated as an event sequence. Such event data is complicated and entails a lot of individual-level information, which is particularly useful for personalized treatment and recommendation (Hosseini et al., 2017; Wang et al., 2021; Cao et al., 2021).

To explore the underlying patterns and structures of event stream data, one of the primary tasks is user/individual clustering (Yan, 2019). That is, given a collection of event sequences, we aim to identify groups displaying similar user/individual behaviors. In recent years, quite a few studies have been investigating this topic. The existing methods on event stream clustering can be mainly summarized into two categories, namely distance-based clustering and model-based clustering. The methods in the former category measured the similarity among distinct event sequences based on extracted features or pre-specified metrics. For example, Berndt

and Clifford (1994) introduced a dynamic time warping approach to detect similar patterns. Pei et al. (2013) used the discrete Frechet distance to construct the similarity matrix. The methods in the second one adopted a temporal point process (TPP) framework, where the event sequences are assumed to follow a mixture of point process models. Most popular algorithms fall into this category. Xu and Zha (2017) proposed a Dirichlet mixture of Hawkes processes, which is the first attempt in TPP clustering. Yin et al. (2021) considered a mixture of multi-level log-Gaussian Cox processes and developed an efficient semi-parametric estimation algorithm. Zhang et al. (2022) introduced a mixture of neural TPP framework, which first incorporates the TPP clustering with neural network techniques.

Despite the recent progress in TPP clustering mentioned above, there are still some fundamental practical issues remaining. In real-world applications, there could exist quite many noisy data. That is, a collection of observed event sequences can not be assumed to exactly follow a mixture of temporal point processes. Instead, a small proportion of event sequences should be treated as *outliers*. Ignoring this could lead to biased or unreliable classification results. Consequently, it comes with another issue that how to properly determine whether an observed event sequence is an outlier or not. Unlike the case in panel data where we could use Eculidean distance, Manhattan distance, or other well-specified metrics to quantitatively detect the outliers, there is no consensus on the metric to be used for event stream data. Last but not least, in the current literature, there is no theoretical study on the performance of TPP clustering or the convergence property of proposed algorithms

even in the setting without outlier event streams. With the existence of outliers, developing the new TPP clustering methodology and the related theoretical guarantees are non-trivial tasks.

In this work, we make an attempt to address the above issues. In particular, we propose a robust TPP clustering framework that is less sensitive to outliers and provides reasonable classification results with theoretical guarantees. Our method works under very mild assumptions that (i) the "inlier" event stream follows a mixture of non-homogeneous Poisson (NHP) processes while the "outlier" event stream can be any arbitrary sequence and (ii) we do not assume the specific temporal point process formula for modeling the "inlier" event stream. The clustering algorithm consists of two components, initialization and robust estimation. In the first component, we construct a distance function induced by the cubic spline (De Boor, 1972) to quantify the dissimilarity between different event sequences and use the new distance for outlier screening to get a subset which presumably contains the "inlier" event streams only. We then apply the K-means++ (Arthur and Vassilvitskii, 2007) method to such a subset to determine the initial center of each group and compute the initial probability of how likely each sample belongs to each group based on the distance from the center. In the second component, in order to fine-tune the clusters, we iteratively maximize a pseudo likelihood function over a working model space. (Since we neither specify the formula of "inlier" event sequences nor assume the distribution of "outlier" event streams, then it is impossible to write down the exact likelihood function. Therefore we use a pseudo likelihood as the alternative

objective. The working model space is of great flexibility. A leading example considered here is the span of linear combinations of cubic spline functions. Other choices are also provided.) Moreover, the estimation equation is incorporated with a Catoni-type (Catoni, 2012) influence function which is known to be robust and enjoys many computational and theoretical advantages. The gradient descent is used for updating the parameters.

The technical contributions of this work are summarized as follows. (a) We introduce a new model-free metric to quantitatively characterize the distance between distinct event sequences. The proposed metric is computationally efficient compared with the existing one (e.g. discrete Frechet distance). Moreover, it can be generalized to a shift-invariant version. (b) We propose a robust estimation procedure that utilizes the Catoni's influence function. We explicitly give out the gradient formula to update the working model parameters. In terms of computational complexity, it only requires an additional step to compute the adjusted weight (which re-weights the possibility of being in a particular group and reduces the impact of outliers) for each sample. (c) A complete theoretical analysis is provided. Under mild conditions, we show the effectiveness of the proposed algorithm. For the initialization component, it can return a set of high-quality centers. For the robust estimation component, it enjoys a linear convergence rate. With the help of Catoni's influence function, the method is robust and has a relatively high break-down point. When the model is correctly specified and the tuning parameter is carefully chosen, the error bound of the estimated parameter is nearly optimal

and the algorithm can detect all outliers with high probability. (d) Moreover, we illustrate that the estimation routine can be incorporated with many optimization approaches including variational inference, (stochastic) EM methods. This suggests that our robust estimation idea can be easily integrated into different algorithms for the estimation of latent TPPs. To the best of our knowledge, this is the first theoretical work in studying the convergence of TPP clustering.

The rest of the paper is organized as follows. A preliminary of event stream data, temporal point process model, Catoni's influence function, and related existing work are provided in Section 2. The main methodology of robust clustering is described in Section 3. We provide the corresponding theoretical analyses in Section 4. In section 5, simulation studies are carried out to show the effectiveness of the new method. Two real data applications are given in Section 6 to show the superior performance of our proposed algorithm. Finally, a concluding remark is given in Section 7.

2. Preliminary

In this section, we provide a brief introduction to the event stream data, Catoni's estimator, clustering techniques, and the metrics to quantify the dissimilarities between two event sequences. These are core ingredients of our later proposed methodology.

2.1 Data Format

We consider the following event stream data, $\Big\{(t_{n1}, ..., t_{ni}, ..., t_{nM_n}); n = 1, ..., N\Big\}$, where t_{ni} is the i-th event time stamp of the n-th individual, M_n is the number of

events observed for individual n, and N is the total number of individuals. For the notional simplicity, we may use S_n to denote observation sequence of individual n, i.e., $S_n = (t_{n1}, ..., t_{ni}, ..., t_{nM_n})$. To help readers gain more intuitions, we provide two real data examples in Table I and Table J in Supplementary F, which show the event stream sequence of a randomly selected user from the internet protocol television (IPTV) data and music listening (Last.FM 1K) data, respectively.

To mathematically characterize the event stream data, it is appropriate to adopt the framework of TPP (Daley et al., 2003), also known as the counting process. For any increasing event time sequence $0 < t_1 < t_2 < ... < t_M$, we let $N(t) := \sharp\{i : t_i \leq t\}$ be the number of events observed up to time t. Then we can define the conditional intensity function, $\lambda^*(t) := \lim_{dt \to 0} \mathbb{E}[N[t, t+dt)|\mathcal{H}_t]/dt$, where $N[t, t+dt) := N(t+dt) - N(t)$ and $\mathcal{H}_t := \sigma(\{N(s); s < t\})$ is the history filtration before time t. Intensity $\lambda^*(t)$ describes the dynamic of the event process and is of great importance and interest for statistical modelling.

2.2 Robustness

In event stream analysis, one could always observe that a few individuals may behave very differently from the majority of the users (Gupta et al., 2013; Sani et al., 2019). Therefore, we need to take into account the potential existence of outliers and develop robust methods to alleviate estimation bias. In the literature of robust M-estimation, there exist different types of methods to estimate population mean, including but not limited to, median of mean (Bubeck et al., 2013), geometric median (Hsu and Sabato, 2016), Huber's estimator (Huber, 1992), trimmed mean

(Lugosi and Mendelson, 2021), robust empirical mean (Prasad et al., 2020), and Catoni's estimator (Catoni, 2012). As discussed in the seminal work (Catoni, 2012), Catoni's estimator is shown to have sub-Gaussian non-asymptotic error bound with optimal multiplicative constant. Furthermore, as shown in the recent work (Bhatt et al., 2022), Catoni's estimator has the highest break-down point compared with other computational friendly methods, i.e., trimmed mean and robust empirical mean. Moreover, according to the numerical results in Fang et al. (2023), Catoni's estimator could achieve the best empirical performance among all methods mentioned above. Hence, we will focus on Catoni's estimator in the remaining sections.

To be mathematically formal, given a set of observations $\{X_i\}_{i=1}^n$, a Catoni's estimator is defined as the solution to the following non-linear equation, $\sum_{i=1}^n \phi(\alpha(X_i - \mu)) = 0$, with respect to μ, where the influence function ϕ is non-decreasing and satisfies

$$-\log\left(1 - x + x^2/2\right) \leq \phi(x) \leq \log\left(1 + x + x^2/2\right), \tag{2.1}$$

and α is a tuning parameter. Throughout the paper, we choose $\phi(x)$ to have the second continuous derivative and its formula is given in Supplementary A.

2.3 Clustering

In many real applications, we could observe strong clustering effects, that is, individuals can be classified into groups according to whether their behaviors are similar or not. For the classical panel data, the clustering problem has been investigated thoroughly. However, for analyzing event stream data, there is no unanimous method yet. Existing methods can be divided into two categories, distance-based

clustering (Berndt and Clifford, 1994; Bradley and Fayyad, 1998; Peng and Müller, 2008) and model-based clustering (Luo et al., 2015; Xu and Zha, 2017; Yin et al., 2021). The former one quantifies the similarities between event streams based on some extracted features and then applies classical clustering algorithms such as K-means, spectral clustering, etc. The latter one assumes that event streams are generated from some underlying parametric mixture models of point processes so that the likelihood function can be derived and EM algorithm could be applied.

However, none of above mentioned methods is robust to outliers or provides any theoretical guarantee to ensure the correct clustering results. In this work, we attempt to propose a new algorithm enjoying the merits of both metric-based and model-based methods. We use a metric-based component for screening outliers and obtaining good initializations of group centers. We use a model-based component for fine-tuning the model parameters and final clustering results.

2.4 How to define a suitable distance

Most existing distances for TPPs are based on the random time change theorem (Brown et al., 2002). Such metrics suffer severe non-identifiability issues. Two very different event streams can be very close under such metrics. More failure modes can be found in Pillow (2009), a detailed explanation is in Supplementary B.

In the literature, there also exists an intensity-free metric called discrete Frechet distance (Eiter and Mannila, 1994; Pei et al., 2013). It can be used to measure the difference between any two polygonal curves in the metric space. However, in terms of computation, it requires a dynamic programming technique, which

leads to quadratic computational complexity. That is, the computational time is proportional to the square of the number of observed event numbers. Hence, it is not a desired method when the data size becomes larger. Therefore, we need to seek a different type of distance which will be described in later sections.

3. Robust Clustering Algorithm

Our new methodology is given in this section. In Section 3.1, we propose a new metric to characterize the distance between two event streams. In Section 3.2, we provide a robust clustering method to handle outliers. In Section 3.3, we give an initialization method to further improve the clustering performance.

3.1 Distance Induced via Cubic Spline

For any two event streams, $S_A = (t_{A,1}, ..., t_{A,M_A})$ and $S_B = (t_{B,1}, ..., t_{B,M_B})$, we consider quantifying the distance between them by adopting the cubic splines. We suppose that event streams are observed within time interval $[0, T]$ or they are periodic with the same period T. Then we define the following distance,

$$d(S_A, S_B) := \int_0^T \left| \hat{\lambda}_{S_A}(t) / \sqrt{M_A} - \hat{\lambda}_{S_B}(t) / \sqrt{M_B} \right| dt, \tag{3.2}$$

where M_A and M_B is the number of events of the sequence S_A and S_B and $\hat{\lambda}_S(\cdot)$ is the estimated intensity function by fitting cubic splines to event stream S. Moreover, if we want to make the distance to be shift invariant, we can adopt the following generalized definition,

$$\tilde{d}(S_A, S_B) := \min_{s \in [0,T]} \int_0^T \left| \hat{\lambda}_{S_A}(t+s) / \sqrt{M_A} - \hat{\lambda}_{S_B}(t) / \sqrt{M_B} \right| dt, \tag{3.3}$$

where $\hat{\lambda}_S(t+s) = \hat{\lambda}_S(t+s-T)$ when $t+s > T$. (3.3) becomes useful when event data are collected from users of different countries in distinct time zones.

In order to compute $\hat{\lambda}_S(\cdot)$ for a fixed event stream S, we need to construct basis functions in the form of cubic splines. Note that the event streams are assumed to be periodic. Therefore, we also enforce the basis to be periodic as well, that is, its value, the first- and second-order derivatives are all continuous at the boundaries. The detailed construction procedure of the basis is given in Supplementary C. We then estimate $\hat{\lambda}_S(t)$ by $\sum_{h=1}^{H} b_{h,S}\kappa_h(t)$, where H is the number of basis, $\kappa_h(t)$ is the h-th basis, and $\{b_{h,S}\}$'s satisfy

$$(b_{1,S}, ..., b_{H,S}) = \arg\max_{(b_1,...,b_H)} \left\{ \sum_{i=1}^{N_S} \log \lambda(t_i) - \int_0^T \lambda(t)dt \right\} \tag{3.4}$$

with $\lambda(t) = \sum_{h=1}^{H} b_h \kappa_h(t)$. Note that (3.4) is essentially a convex optimization problem that can be efficiently solved. Computation of (3.2) or (3.3) scales linearly with the lengths of event sequences. Therefore, the proposed metric is more computationally friendly than the discrete Frechet distance.

Note that we divide the estimated intensity by the square root of the number of events in (3.2). This is due to the following observation.

Proposition 1. *Suppose $S = (t_1, t_2, \cdots)$ follows a homogeneous Poisson process with intensity λ and $f(\cdot)$ is a bounded function in $[0, T]$. The variance of $\sum_i f(t_i)/\sqrt{N(T)}$ is $(\int_0^T f^2(t)dt/T) \cdot (1/4 + O(1/\lambda))$.*

According to Proposition 1, we rescale the intensity function to make the distance function be insensitive to the magnitude of intensity. Thus we can classify

different individuals based on their intrinsic patterns instead of the absolute value of event number.

To end this subsection, we show that $d(S_A, S_B)$ ($\tilde{d}(S_A, S_B)$) given in (3.2) ((3.3)) is a proper distance function. Here $d(S_A, S_B)$ is called as a distance function if it satisfies three properties: (i) the distance between an event sequence and itself is always zero, (ii) the distance between distinct event sequences is always positive and symmetric, and (iii) the distance satisfies the triangle inequality.

Theorem 1. *The function defined in* (3.2) *or* (3.3) *is a distance function.*

Theorem 1 is proved in Supplementary I. Without validating these, directly applying existing clustering algorithms may fail without theoretical guarantees.

3.2 Clustering with Robust Estimation

In this section, we propose a clustering algorithm based on a mixture model (Fraley and Raftery, 2002; McLachlan et al., 2019). In particular, we assume the observed event sequences $\mathbf{S} = \{S_n\}_{n=1}^N$ are generated from mixture non-homogeneous Poisson processes with K classes and possible outlier sequences. All of them have the same period T. If an event sequence belongs to class $k \in [K]$, then its corresponding population-level intensity, or rate, is $\lambda_k^*(t)$. To be more mathematically formal, the event sequences (without outliers) $\tilde{S}_1, ..., \tilde{S}_N$ are i.i.d. with mixture intensity $\sum_{k=1}^K \pi_k \lambda_k^*(t)$. There are at most η percent outlier sequences, that is,

$$\sum_{i=1}^{N} \mathbf{1}\{\tilde{S}_i \neq S_i\} \leq \eta N. \tag{3.5}$$

(3.5) allows arbitrary outliers, which is known as the strong contamination model (**?**) and is bit more general than η-Huber contamination model (Huber, 2004).

At the moment, we do not put any structural assumption on $\lambda_k^*(t)$'s. Instead, throughout the current paper, we mainly focus on the following working model, that is, $\lambda_k^*(t)$ can be approximated by

$$\lambda_k(t) := \sum_{h=1}^{H} b_{k,h}\kappa_h(t), \tag{3.6}$$

where $\kappa_h(t)$ is the h-th basis function defined in the last section. We write $\boldsymbol{B}_k := [b_{k,h}] \in \mathbb{R}_{0+}^H$ as the coefficient parameter in non-homogeneous Poisson process of class k, $\mathbf{B} := \{\boldsymbol{B}_k\}_{k=1}^K$ as the whole parameter for simplicity.

Remark 1. The non-homogeneous Poisson assumption and the working model (3.6) can be replaced by any type of model assumption for TPPs in the existing literature. Some detailed discussions on other commonly used working models can be found in Supplementary D.

According to the classical mixture models (Xu and Zha, 2017; Zhang et al., 2022) with no outliers, we let Z_n denote the latent label for the n-th event stream. In other words, $Z_n = k$ represents that the n-th event sequence belongs to the k-th class. If there is **no** outlier, we can write down the probability of an event stream S as $p(S;\mathbf{B}) = \sum_k \pi_k \cdot \mathrm{NHP}\left(S|\boldsymbol{B}_k\right)$ with

$$\mathrm{NHP}\left(S \mid \boldsymbol{B}_k\right) := p(S|Z=k) = \prod_i \lambda_k\left(t_i\right) \exp\left(-\int_0^{L(S)\cdot T} \lambda_k(t)dt\right),$$

where π_k 's are class probabilities, $\mathrm{NHP}\left(S \mid \boldsymbol{B}_k\right)$ is the conditional probability of the event sequence S if it belongs to class k, and $L(S)$ is the number of periods in event sequence S. We write $\mathbf{Z} = \{Z_n\}_{n=1}^N$. Then the (pseudo) likelihood of $\mathbf{S}$, $\mathbf{Z}$ is

$$p(\mathbf{S}, \mathbf{Z}; \mathbf{B}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left[\pi_k \, \mathrm{NHP}\left(S_n \mid \boldsymbol{B}_k\right)\right]^{\mathbf{1}\{Z_n = k\}}$$

and the (pseudo) marginal likelihood of $\mathbf{S}$ is

$$p(\mathbf{S}; \mathbf{B}) = \prod_{n=1}^{N} \left\{ \sum_{k=1}^{K} \pi_k \, \mathrm{NHP}\left(S_n \mid \boldsymbol{B}_k\right) \right\}. \tag{3.7}$$

Then the goal becomes to compute the maximizer, $\mathbf{B}_{opt} := \arg\max_{\mathbf{B}} p(\mathbf{S}; \mathbf{B})$.

Remark 2. Here we call (3.7) as the pseudo likelihood since it is not the exact likelihood function. This is because we treat all N observed event sequences as inliers even if it is not. In other words, we try to estimate the group centers and model parameters under the mis-specified setting.

In order to solve $\mathbf{B}_{opt}$, the standard and most popular computational approach is the expectation-maximization (EM) algorithm (Dempster et al., 1977) in the literature. However, due to the existence of outliers, we cannot directly apply the EM algorithm. We make the modification to it by using the Catoni influence function to reweight each observed event sequence. At time step t, E-step and M-step are given as follows.

E-step. We first compute the posterior $p(\mathbf{Z}|\mathbf{S}; \mathbf{B}^{(t-1)})$, where $\mathbf{B}^{(t-1)}$ is the parameter estimate in the previous step. It is not hard to find that

$$p(\mathbf{Z}|\mathbf{S}; \mathbf{B}^{(t-1)}) = \prod_{n=1}^{N} p(Z_n|S_n; \mathbf{B}^{(t-1)}) = \prod_{n=1}^{N} \prod_{k=1}^{K} (r_{nk}^{(t)})^{\mathbf{1}\{Z_n = k\}} \tag{3.8}$$

$$\text{with} \qquad r_{nk}^{(t)} = \frac{\rho_{nk}^{(t)}}{\sum_{k'} \rho_{nk'}^{(t)}}, \tag{3.9}$$

where $\rho_{nk}^{(t)} := \pi_k^{(t-1)} \cdot \mathrm{NHP}(S_n|\boldsymbol{B}_k^{(t-1)})$. For simplicity, we write $p(\mathbf{Z}|\mathbf{S}; \mathbf{B}^{(t-1)})$ as $q^{(t)}(\mathbf{Z})$. Thus the Q-function, the expectation of the complete log-likelihood over $q^{(t)}(\mathbf{Z})$, $\mathcal{Q}(\mathbf{B}|\mathbf{B}^{(t-1)})$ is

$$\mathbb{E}_{q^{(t)}(\mathbf{Z})}[\log p(\mathbf{S} \mid \mathbf{Z}, \mathbf{B})] + C = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}^{(t)} \log \mathrm{NHP}\left(S_n \mid \boldsymbol{B}_k\right) + C \tag{3.10}$$

M-step. The classical routine is to find the estimate $\mathbf{B}^{(t)} = \arg\max_{\mathbf{B}} \mathcal{Q}(\mathbf{B}|\mathbf{B}^{(t-1)})$. In our setting, we have the following observation that $\mathbf{B}^{(t)} \equiv (\boldsymbol{B}_k^{(t)})_{k=1}^K$ with

$$\boldsymbol{B}_k^{(t)} := \arg\max_{\boldsymbol{B}_k} \sum_{n=1}^{N} r_{nk}^{(t)} \log \mathrm{NHP}(S_n|\boldsymbol{B}_k),$$

which can be equivalently written as $\boldsymbol{B}_k^{(t)} := \arg\max_{\boldsymbol{B}_k} \mu_{avg}^{(t)}(\boldsymbol{B}_k)$ with $\mu_{avg}^{(t)}(\boldsymbol{B}_k)$ being the solution to

$$\sum_{n=1}^{N} r_{nk}^{(t)} \left(\log \mathrm{NHP}\left(S_n \mid \boldsymbol{B}_k\right) - \mu\right) = 0 \tag{3.11}$$

with respect to μ. Given the existence of outliers, we instead consider the following robust estimator

$$\boldsymbol{B}_k^{(t)} := \arg\max_{\boldsymbol{B}_k} \hat{\mu}_{\phi}^{(t)}(\boldsymbol{B}_k), \tag{3.12}$$

where $\hat{\mu}_{\phi}^{(t)}(\boldsymbol{B}_k)$ is the solution to

$$\sum_{n=1}^{N} r_{nk}^{(t)} \cdot L(S_n) \cdot \phi_\rho \left(\log \mathrm{NHP}\left(S_n \mid \boldsymbol{B}_k\right)/L(S_n) - \mu\right) = 0 \tag{3.13}$$

with respect to μ, where $\phi_\rho(x) := \rho^{-1} \cdot \phi(\rho \cdot x)$ with $\phi(x)$ defined in (1) and ρ being a tuning parameter. (The following results will not be affected, if we also allow ρ depend on class index k.) Especially, when $\phi(x)$ is an identity function, (3.13) reduces to (3.11) up to a multiplicative constant (free of $\boldsymbol{B}_k$). To solve (3.12), we consider to use the gradient descent-type method. In particular, we can compute the gradient with the explicit formula which is given in the following proposition.

Proposition 2. *The gradient* $\varrho_k^{(t)}$ *of* $\hat{\mu}_\phi^{(t)}(\boldsymbol{B}_k)$ *with respect to parameter* $\boldsymbol{B}_k$ *at* $\boldsymbol{B}_k^{(t-1)}$ *(i.e.* $\varrho_k^{(t)} := \frac{\partial \hat{\mu}_\phi^{(t)}(\boldsymbol{B}_k)}{\partial \boldsymbol{B}_k}|_{\boldsymbol{B}_k^{(t-1)}}$*) is*

$$\sum_{n=1}^{N} \frac{r_{nk}^{(t)} w_{nk}^{(t)}}{\sum_{n=1}^{N} r_{nk}^{(t)} w_{nk}^{(t)} L(S_n)} \cdot \frac{\partial \log \mathrm{NHP}\left(S_n \mid \boldsymbol{B}_k\right)}{\partial \boldsymbol{B}_k} \, |_{\boldsymbol{B}_k^{(t-1)}}, \tag{3.14}$$

where $w_{nk}^{(t)} = \phi_\rho' \left(\log \mathrm{NHP}\left(S_n \mid \boldsymbol{B}_k^{(t-1)}\right) / L(S_n) - \hat{\mu}_\phi(\boldsymbol{B}_k^{(t-1)}) \right)$.

According to Proposition 2, we actually adjust $\boldsymbol{B}_k$'s gradient via influence function ϕ_ρ. Here $w_{nk}^{(t)}$ can be viewed as the adjusted weight of n-th event stream. By the construction of ϕ_ρ, it can be checked that $w_{nk}^{(t)} \in [0, 1]$. When $w_{nk}^{(t)}$ is close to one, it indicates the strong confidence that event stream n is more likely to belong to class k. On the other hand, if $w_{nk}^{(t)}$ is close to zero, it implies the corresponding event stream could be an outlier or is at least far away from class k. If an event sequence n is truly an outlier, then its weights w_{nk}'s are uniformly small for all $k \in [K]$. Then it has negligible influence to the gradient according to (3.14), which in turn implies the robustness of our proposed method. The parameter update is

$$\boldsymbol{B}_k^{(t)} = \boldsymbol{B}_k^{(t-1)} - \mathrm{lr} \cdot \varrho_k^{(t)} \quad \text{for } k \in [K], \tag{3.15}$$

where lr is the learning rate/step size. When $\|\boldsymbol{B}_k^{(t)} - \boldsymbol{B}_k^{(t-1)}\| \leq \epsilon$ (ϵ is a small tolerance parameter), we stop the E- and M-steps. Lastly, for class probabilities, we can update $\{\pi_k\}_{k=1}^K$ by $\pi_k^{(t)} = \sum_{n=1}^N r_{nk}^{(t)}/N$.

In the case of a time shift, we need to assign a shift parameter to each event sequence. We let shift_n be the time zone of n-th event stream. In addition to update $\boldsymbol{B}$ at time step t, we also update

$$\text{shift}_n^{(t)} = \underset{\text{shift}_n \in \{\frac{T}{H_{\text{shift}}}, \frac{2\cdot T}{H_{\text{shift}}}, \cdots, T\}}{\text{argmin}} \int_0^T \left| \hat{\lambda}_{S_n}(u + \text{shift}_n) - \hat{\lambda}_{Z_n^{(t)}}(u) \right| du, \tag{3.16}$$

where H_{shift} represents the number of possible time shifts (e.g. H_{shift} can be seen as the 24 time zones), $\hat{\lambda}_{S_n}(\cdot)$ is obtained from (3.4) and $\hat{\lambda}_{Z_n^{(t)}}(\cdot)$ is the estimated intensity function of class $Z_n^{(t)}$ with $Z_n^{(t)} = \arg\max_k r_{nk}^{(t)}$. Again, when $u+\text{shift}_n > T$, we define $\hat{\lambda}_{S_n}(u + \text{shift}_n) := \hat{\lambda}_{S_n}(u + \text{shift}_n - T)$.

The algorithm of robust clustering is summarized in Algorithm 1.

Remark 3. Here EM algorithm is not a necessary component in our methodology. We can also use other optimization approaches (for latent variable models), e.g., variational inference (VI, Blei et al. (2017)), stochastic EM (Diebolt and Ip, 1995), etc. Therefore, our framework is very flexible and can be integrated with any widely-used estimation algorithm. More explanations can be found in Supplementary D.

3.3 Initialization

A major weakness of EM-type algorithm is that it can only return local optimal solutions. With bad initialization, the algorithm may give the erroneous classification results which could be very different from the true underlying clusters. As we find in the numerical study, this issue becomes even worse under the temporal point process settings.

Arthur and Vassilvitskii (2007) introduced the K-means++ algorithm, an extended K-means method, to alleviate local convergence issues. K-means++ has since gained popularity for its ability to produce high-quality initial centers, leading to faster convergence and better clustering performance. Following the main

ideas of K-means++, we propose a robust K-means++ initialization algorithm. It mainly consists of two steps, (i) outlier screening and (ii) inlier weighting.

Outlier screening. We first introduce several tuning parameters M, N', β, and α. M is an integer which is much smaller than N, N' is the pre-determined number of inliers, β is the screening speed ($\beta \in (0, \frac{N'}{N})$), and $\alpha \in (0, 1)$ is the quality parameter. Outlier screening iteratively repeats the following procedures until it finds N' inliers.

At round 0, we set $\mathcal{S}_{in}$ to be the empty set. For n-th event sequence, we calculate its corresponding distance set $\mathcal{D}_n^{(0)}$, where $\mathcal{D}_n^{(0)} := \{d(S_n, S_{n,m}^{(0)})\}_{m=1}^{M}$ with $S_{n,m}^{(0)}$ being a uniformly randomly selected sample from the whole dataset $\mathbf{S}$ and metric function d being defined according to (3.2) (or (3.3) when shift is considered). We then compute the lower α-quantile $q_{n,\alpha}^{(0)}$ of $\mathcal{D}_n^{(0)}$. We rank $\{q_{n,\alpha}^{(0)}\}_{n=1}^{N}$ from the smallest to the largest and add the first $\lfloor \beta \cdot N \rfloor$ samples into $\mathcal{S}_{in}$.

At round $t \geq 1$, for event sequence n not in $\mathcal{S}_{in}$, we similarly calculate its corresponding distance set $\mathcal{D}_n^{(t)}$, where

$$\mathcal{D}_n^{(t)} := \{d(S_n, S_{n,m}^{(t)})\}_{m=1}^{M} \tag{3.17}$$

with $S_{n,m}^{(t)}$ being a uniformly randomly selected sample from $\mathcal{S}_{in}$. We similarly compute its lower α-quantile $q_{n,\alpha}^{(t)}$ of $\mathcal{D}_n^{(t)}$. We then rank $\{q_{n,\alpha}^{(t)}\}_{n \notin \mathcal{S}_{in}}$ from the smallest to the largest and add the first $\min\{\lfloor \beta \cdot |\mathbf{S} \backslash \mathcal{S}_{in}| \rfloor, N' - |\mathcal{S}_{in}|\}$ samples into $\mathcal{S}_{in}$. We repeat this procedure until $\mathcal{S}_{in}$ reaches N'. (We let $M \ll N$ since the computation of distance sets could be time consuming.)

In summary, the above procedure recursively detects inliers. If an event se-

quence is closer to the center of inliers, then it is more likely to be detected in very early rounds. If an event sequence is far from other samples, then it is hard to be included in set $\mathcal{S}_{in}$.

Inlier weighting. After obtaining $\mathcal{S}_{in}$, a set tentatively consisting of inliers only, we then perform K-means++ algorithm (Arthur and Vassilvitskii, 2007; Georgogiannis, 2016; Deshpande et al., 2020) on it. The details are given as follows.

(a) Select the first center c_1: Choose one event stream from $\mathcal{S}_{in}$.

(b) Select subsequent centers c_k's: For the next center, randomly select the event stream with the probability proportional to the square of the distance from it to the nearest existing center. That is, $p(S_n) = \frac{D(S_n)^2}{\sum_{S\in\mathcal{S}_{in}} D(S)^2}$, where $D(S) = \min_{k'\in[k-1]} d(S, c_{k'})$.

(c) Repeat step (b) until K centers are chosen.

We denote K selected centers by $\mathcal{C}^{ini} = \{c_k\}_{k=1}^K$. To make the subsequent classification more robust, we also design the initial weight for sequence S_n in $\mathcal{S}_{in}$ of being in class k as

$$r_{nk} = \frac{\psi_{\alpha_k}(d(S_n, c_k))}{\sum_{n\in\mathcal{S}_{in}} \psi_{\alpha_k}(d(S_n, c_k))}, \tag{3.18}$$

where α_k is the the median of the set $\{d(S_n, c_k)\}_{n\in\mathcal{S}_{in}}$ and $\psi_\alpha(x) = \psi(x/\alpha)$ with $\psi(x) := \phi'(x) \equiv x/(1 + x + 0.5 \cdot x^2)$. The reason of doing this inlier weighting is to reduce the weights of a few outliers that may still remain in $\mathcal{S}_{in}$. For $n \notin \mathcal{S}_{in}$, we let $r_{nk} \equiv 0$ for any $k \in [K]$.

In the case of data shift, we also return the initial shift parameter. For event stream S_n, we set

$$\text{shift}_n = \underset{\text{shift}\in\{\frac{T}{H_{\text{shift}}},\frac{2\cdot T}{H_{\text{shift}}},\cdots,T\}}{\text{argmin}} \int_0^T \left|\hat{\lambda}_{S_n}(t+\text{shift}) - \hat{\lambda}_{c_{k_n}}(t)\right| dt, \tag{3.19}$$

where $c_{k_n} = \text{argmin}_{c_k\in\mathcal{C}}\, d(S_n, c_k)$.

The algorithm of initialization is summarized in Algorithm 2.

Remark 4. Our current framework allows the following two flexibilities: (1) temporal shift among different event streams, (2) different numbers of periods but with the same length of time period. However, it could happen that different streams may have varying time periods in many practical scenarios. If this happens, one simple modification to our methodology is to apply the time transformation to each of the event streams so that they have the same time period after the time re-scaling.

4. Theoretical results

In this section, we provide a theoretical analysis of our proposed method. In particular, we show that Algorithm 2 can return high-quality initial points (Theorem 3) and Algorithm 1 can ensure the local convergence under mild conditions (Theorem 4).

Previously, we have not put any requirements on the observed event sequences yet. To start with, we introduce several technical assumptions.

Assumption 1. Suppose the dataset has the following decomposition, $\mathbf{S} = \mathcal{S}_{inlier} \cup \mathcal{S}_{outlier} = \mathcal{S}_1 \cup ... \cup \mathcal{S}_K \cup \mathcal{S}_{outlier}$. Here $\mathcal{S}_{outlier}$ is the set of outlier event sequences, $\mathcal{S}_k$ is the set of inlier event streams that belong to class k, and $\mathcal{S}_{inlier}$ is the union of all interior samples. $\mathcal{S}_1, ..., \mathcal{S}_K, \mathcal{S}_{outlier}$ are non-overlapping. Assume that

for any $k \in [K]$, $\max_{S_{n_1}, S_{n_2} \in \mathcal{S}_k} d(S_{n_1}, S_{n_2}) < \min\{\min_{S_{n_1} \in \mathcal{S}_k, S_{n_2} \in \mathcal{S}_{outlier}} d(S_{n_1}, S_{n_2}), \min_{S_{n_1}, S_{n_2} \in \mathcal{S}_{outlier}} d(S_{n_1}, S_{n_2})\}$.

Here Assumption 1 requires that, for any $k \in [K]$, the upper bound of the distance between two different sequences in $\mathcal{S}_k$ is smaller than the distance between any two sequences in $\mathcal{S}_{inlier}$ and $\mathcal{S}_{outlier}$, and it is also smaller than the distance between any two outliers. With the help of Assumption 1, it guarantees that outliers can be identified. In fact, this assumption can be relaxed. The requirement that $\min_{S_{n_1}, S_{n_2} \in \mathcal{S}_{outlier}} d(S_{n_1}, S_{n_2})$ is larger than the maximum distance between inliers is not necessary. We can allow the distance between a small number of outliers to be close, which will not affect our results.

Assumption 2. There is a lower bound $\pi_{low} > 0$ for the proportion of each inlier cluster, that is, $\pi_k \geq \pi_{low}$ for $k \in [K]$.

Assumption 2 ensures "inlier" identifiability, i.e., every inlier cluster is not drained and inliers will not be treated as outliers. On the other hand, if some outliers, whose number is much less than $\pi_{low} \cdot N$, are close together, they will not be recognized as a new cluster.

Assumption 3. The space of model parameters $\boldsymbol{B}_k$'s defined in (3.6) is bounded. That is, there exists $\Omega_B > 0$ such that $\|\boldsymbol{B}_k\|_1 < \Omega_B$ for all $k = 1, 2, \ldots, K$.

Assumption 3 is a standard technical condition (Lehmann and Casella, 2006; Casella and Berger, 2021) that parameters are in the compact and bounded space.

Assumption 4. There exist τ and Ω such that $0 < \tau \leq \lambda_k^*(t) \leq \Omega$ for all $t \in [0, T]$ and $k = 1, 2, \ldots, K$.

Assumption 4 is also a classical technical requirement (Cai et al., 2022; Fang et al., 2023) to ensure that the intensity function is bounded away from zero and from above. We define the true working model parameter,

$$\boldsymbol{B}_k^* = \arg\max_{[b_{k,h}]} \left\{ \int_0^T (\log \lambda_k(t)) \cdot \lambda_k^*(t) dt - \int_0^T \lambda_k(t) dt \right\} \quad \forall k \in [K] \tag{4.20}$$

with $\lambda_k(t)$ being defined in (3.6). We write $\lambda_{\boldsymbol{B}_k^*}(t) = \sum_{h=1}^H b_{k,h}^* \kappa_h(t)$, then $\lambda_{\boldsymbol{B}_k^*}(t)$ is the intensity function closest to $\lambda_k^*(t)$ in the working model space.

Assumption 5. For any two different classes k and k', there exists a constant $C_{gap} > 0$ such that, if event stream S belongs to Class k, then it holds $\mathbb{E}[\log \mathrm{NHP}(S|\boldsymbol{B}_{k'}^*)] < \mathbb{E}[\log \mathrm{NHP}(S|\boldsymbol{B}_k^*)] - C_{gap} \cdot L$, $\forall\ k' \neq k$.

Assumption 5 ensures "class" identifiability that $\boldsymbol{B}_k^* \neq \boldsymbol{B}_{k'}^*$ when $k \neq k'$. In other words, event streams from different classes can be distinguished by our working model, the non-homogeneous Poisson process. Here we assume that all the event streams have the same number of periods L for simplicity. When the number of periods are different, Assumption 5 still holds if L is replaced by $\min_n L(S_n)$.

Next we show that our initialization algorithm can return a set of high-quality centers. To see this, we need to introduce the following quantities. Define $\Upsilon_{\mathcal{C}}(\mathcal{S}_{in}) := \sum_{S \in \mathcal{S}_{in}} \min_{c \in \mathcal{C}} d(S, c)^2$. We also define $\mathcal{C}_{\mathrm{OPT}}$ is the set that minimizes $\Upsilon_{\mathcal{C}}(\mathcal{S}_{in})$ over all possible $\mathcal{C}$. Therefore, $\Upsilon_{\mathcal{C}_{\mathrm{OPT}}}(\mathcal{S}_{in}) = \min_{\mathcal{C}} \Upsilon_{\mathcal{C}}(\mathcal{S}_{in})$. $\Upsilon_{\mathcal{C}}(\mathcal{S}_{in})$ evaluates the quality of $\mathcal{C}$, i.e., smaller $\Upsilon_{\mathcal{C}}(\mathcal{S}_{in})$ is, better $\mathcal{C}$ is.

Theorem 2. *Apply Algorithm 2 and get $\mathcal{C}^{ini}$. It holds that $E[\Upsilon_{\mathcal{C}^{ini}}(\mathcal{S}_{in})|\mathcal{S}_{in}] \leq 16(\ln K + 2)\Upsilon_{\mathcal{C}_{\mathrm{OPT}}}(\mathcal{S}_{in})$, where K is the number of clusters.*

The above theorem indicates that, given the screening set $\mathcal{S}_{in}$, the set $\mathcal{C}^{ini}$ is nearly optimal up to a multiplicative constant in the average sense. Furthermore,

when L becomes large, Theorem 2 implies that the algorithm can well identify centers from K different classes. See the following theorem.

Theorem 3. *Let $\mathcal{C}_{lack}$ be any set such that it consists of K event streams, but at least two of them are from the same true underlying class. When $L \to \infty$, we have $\Upsilon_{\mathcal{C}_{lack}}(\mathcal{S}_{in}) > 16(\ln K + 2)\Upsilon_{\mathcal{C}_{\text{OPT}}}(\mathcal{S}_{in})$ with high probability under Assumptions 1, 2 and 5.*

Then we illustrate that the gradient descent step in Algorithm 1 leads to the local convergence property with high probability when L is large enough. For $k \in [K]$, we define function $\mu(\boldsymbol{B}_k \mid \boldsymbol{B}_k^*)$ which satisfies

$$\mathbb{E}_S \left[w_k \left(S; \boldsymbol{B}_k^* \right) \phi_\rho \left(\log \text{NHP}(S \mid \boldsymbol{B}_k) \right) / L - \mu(\boldsymbol{B}_k \mid \boldsymbol{B}_k^*) \right] = 0,$$

where $w_k(S; \boldsymbol{B}) := \pi_k \,\text{NHP}(S \mid \boldsymbol{B}_k) / \sum_j \pi_j \,\text{NHP}(S \mid \boldsymbol{B}_k)$.

Theorem 4. *Suppose Assumption 3, 4, 5, and $\eta = |\mathcal{S}_{outlier}|/N < (4 \sup_x |\phi(x)|)^{-1}$ hold. There exists a constant $a > 0$ such that $C_{gap} - 2a - 3\bar{m}_c \log\left((\tau + a/T)/\tau\right) > 0$; if $\left\| \boldsymbol{B}_k^{(t)} - \boldsymbol{B}_k^* \right\| < a$ for $k \in [K]$ and learning rate $lr = 2/(\lambda_{\max} + \lambda_{\min})$, then update (3.15) satisfies*

$$\left\| \boldsymbol{B}_k^{(t+1)} - \boldsymbol{B}_k^* \right\| \leq \frac{\lambda_{\max} - \lambda_{\min} + 2\gamma}{\lambda_{\max} + \lambda_{\min}} \left\| \boldsymbol{B}_k^{(t)} - \boldsymbol{B}_k^* \right\| + \epsilon^{unif}, \tag{4.21}$$

where $\lambda_{\max}$ and λ_{min} are the largest and smallest eigenvalue of $-\Delta\mu(\boldsymbol{B}_k \mid \boldsymbol{B}_k^*)$ (the second derivative matrix of $-\mu(\boldsymbol{B}_k \mid \boldsymbol{B}_k^*)$), $\bar{m}_c := \sup_k \int_0^T \lambda_k^*(t)dt$, γ is a parameter satisfying $\gamma \leq O(\sqrt{H}L\exp(-GL)) \to 0$ for sufficiently large L, and $\epsilon^{unif} = O_p(\sqrt{H}L\exp(-GL)/\sqrt{N} + (\rho + \sqrt{H})(1/\sqrt{NL} + \rho/L + \log N/(\rho N) + \eta/\rho))$.

Theorem 4 implies that $\|\boldsymbol{B}_k^{(t)} - \boldsymbol{B}_k^*\|$ decreases geometrically until it has the same order of ϵ^{unif}. Moreover, the consequence of Theorem 3 and Theorem 4 is that

$\boldsymbol{B}^{(0)}$ obtained in Algorithm 1 will eventually satisfy $\left\|\boldsymbol{B}_k^{(0)} - \boldsymbol{B}_k^*\right\| < a$ as $L \to \infty$. Hence our robust clustering algorithm enjoys linear convergence speed.

As we know, the *break-down point* is the maximum proportion of outliers the estimator can handle before giving arbitrarily incorrect results. Theorem 4 gives us the following immediate result.

Corollary 1. *Under the same assumptions of Theorem 4, the break-down point of the proposed algorithm is* $1/(4\sup_x |\phi(x)|)$. *In particular, we choose* $\phi(x) = \phi_{sharp}(x)$ *given in (3) of Supplementary A, and the break-down point is 36%.*

Remark 5. We would like to point out that the local convergent result, Theorem 4, holds for a wide range of working model families beyond (3.6). In other words, the update satisfies (4.21), once we can verify the smallest eigenvalue of $-\Delta\mu(\boldsymbol{B}_k \mid \boldsymbol{B}_k^*)$ is positive.

Corollary 2. *Under the same conditions specified in Theorem 4, we choose* $\rho = \sqrt{L \cdot (\log N/N + \eta)}$. *Then* $\|\hat{\boldsymbol{B}}_k - \boldsymbol{B}_k^*\| = O_p\left(\sqrt{\log N/(NL) + \eta/L} + \epsilon\right)$, *where* ϵ *is the tolerance parameter in Algorithm 1.*

As we can see, the estimation error consists of two parts, $\sqrt{\log N/(NL)}$ and $\sqrt{\eta/L}$. The former one corresponds to the stochastic variability caused by the inlier event streams and the latter one is the price we need to pay when there exist $100 \cdot \eta$ percent outlier event streams. Note that in the robust statistical literature (Lugosi and Mendelson, 2021; Bhatt et al., 2022), the minimax M-estimator enjoys the

rate of $1/\sqrt{\text{sample size}} + \sqrt{\text{proportion of outliers}}$. Hence our proposed estimator is (nearly) statistically optimal.

In addition to the convergence of working model parameter, we also show that Algorithm 1 can identify almost all outliers under certain additional assumptions. We say an outlier event stream S is indistinguishable by the working NHP model if $\int_0^T (\lambda_o(t) - \lambda_k^*(t)) \log \lambda_{\boldsymbol{B}_k^*}(t) dt = 0$ for some $k \in [K]$, where S is generated according to intensity $\lambda_o(t)$. We then define $\mathcal{S}_{indis} := \{S \in \mathcal{S}_{outlier} | S \text{ is indistinguishable}\}$ to be the set of indistinguishable event streams. On the other hand, the outliers detected by our proposed method can be constructed as

$$\hat{\mathcal{S}}_{outlier} := \{S_n | \phi'_\rho(\log \text{NHP}\left(S_n \mid \hat{\boldsymbol{B}}_k\right)/L(S_n) - \hat{\mu}_\phi(\hat{\boldsymbol{B}}_k)) < \epsilon_{bound}; \forall k \in [K]\},$$

where we can set $\epsilon_{bound} = 0.1$. In other words, an event stream is treated as the outlier if its adjusted weight for any class is less than the cutoff 0.1.

Theorem 5. *Under Assumptions 1 - 5, it holds* $\mathbb{P}\left(\hat{\mathcal{S}}_{outlier} = \mathcal{S}_{outlier} \backslash \mathcal{S}_{indis}\right) \to 1$ *as* $L \to \infty$, *if we choose* $\rho = L^\beta$ *(with* $0 < \beta < \frac{1}{2}$*).*

Note that set $\mathcal{S}_{indis}$ is of measure zero if $\lambda_o(t)$ is uniformly randomly selected from a continuous function space. Therefore, generically speaking, all outliers can be identified out as suggested by Theorem 5.

5. Simulation Study

In this section, a series of simulation studies is conducted that includes an ablation study to demonstrate the feasibility and efficiency of our robust clustering method.

We compare our proposed method with the other two baseline methods. One method is a standard clustering algorithm with random initialization of $\boldsymbol{B}^{(0)}$, $\pi_k^{(0)}$'s

and identity influence function, and the other one is almost the same as the proposed algorithm but with random initialization. Three working models, the non-homogeneous Poisson process (3.6), the Hakwes process, and the frailty model, are considered. To save the space, we only present the results for the non-homogeneous Poisson process (NHP). Results of the other two can be found in Supplementary E.

We generate inlier event sequences according to the following intensity functions with a total of $K = 4$ classes,

$$\lambda_1^*(t) = 5/3\exp(-(t+4.8)^2/10) + 5/3\exp(-(t-2.4)^2/50),$$

$$\lambda_2^*(t) = 5/3\exp(-(t-6)^2/4) + 15/4\exp(-(t-21.6)^2/4),$$

$$\lambda_3^*(t) = 15/4\exp(-(t-4.8)^2/1.5) + 35/12\exp(-(t-12)^2) + 15/4\exp(-(t-19.2)^2/1.5),$$

$$\lambda_4^*(t) = 10/3\exp(-(t-21.6)^2/40) + 5/3\exp(-(t-26.4)^2/10),$$

where $t \in [0, T]$ with $T = 24$ (corresponding to 24 hours). At the same time, we consider the three types of outlier event sequences according to the following intensity functions:

$$\lambda_{out1}(t) = 125/6 \cdot (U + 0.1) \text{ , where } U \sim U(0,1) + 0.1,$$

$$\lambda_{out2}(t) = 125/18(U + 0.1) + 125/3\exp(-(t - 24B_1)^2/0.5), \text{where } U \sim U(0,1), B_1 \sim U(0,1),$$

$$\lambda_{out3}(t) = 25/2 \cdot \exp(-(t - 24B_1)^2/0.02) + 25/3 \cdot \exp(-(t - 24 \cdot B_2)^2/0.02)$$
$$+ 25/6 \cdot \exp(-(t - 24 \cdot B_3)^2/0.02) \text{ , where } B_i \sim U(0,1) \quad \forall i \in \{1,2,3\}.$$

Based on the formula, we can find that outlier event sequence of the first type follows a homogeneous Poisson process, the outlier intensity function of the second type has a unimodel shape, and the third one has three modes. Based on the intensity value, we can observe that the number of events in the first two type

outliers are generally larger than those of inliers, while the number of events in the third type outliers are slightly smaller than those of inliers. For each setting, we generate 60 event sequences for each inlier class and 60 event sequences according to one of the three outlier intensities. In total, there are $N = 60 \times 4 + 60 = 300$ samples. We let the number of periods $L \in \{1, 2, 4\}$. In addition, we also consider to shift the n-th sample by shift_n which is an integer uniformly sampled between 0 and 23. We apply our proposed method and two baselines by setting number of classes equal to 4, 5, or 6. All the above settings are repeated for 100 times. We set tuning parameter ρ for class k to be $0.6 \cdot \sqrt{\int_0^T \log^2 \lambda_k^{(0)}(t) \cdot \lambda_k^{(0)}(t) dt}$, $\epsilon = 0.1$, $\alpha = 0.2$, $\beta = 0.3$, $M = 50$, and $N' = 0.75 \cdot N$.

We use the clustering purity (Schütze et al., 2008) to evaluate the performances of three methods. Specifically, the purity index is defined as

$$\text{purity}(\hat{\mathcal{S}}, \mathcal{S}^*) = \frac{1}{N} \sum_k \max_{k'} \left| \hat{\mathcal{S}}_k \cap \mathcal{S}_{k'}^* \right|, \tag{5.22}$$

where $\hat{\mathcal{S}} = \{\hat{\mathcal{S}}_1, ..., \hat{\mathcal{S}}_{\hat{K}}\}$ and $\mathcal{S}^* = \{\mathcal{S}_1^*, ..., \mathcal{S}_{K^*}^*\}$ are two partitions of of the data set according to the estimated labels and true underlying labels. It is easy to see that the range of purity value is between 0 and 1. The higher the purity value is, the better the clustering result is. Moreover, the purity is non-decreasing as $\hat{K}$ increases. In other words, for a fixed algorithm, the purity will get larger if we wish to cluster the data into more classes. The results are summarized in Table 1.

As seen from the tables, the proposed method outperforms the other two baselines by a big margin under almost all settings. As K varies from 4 to 6, the purity returned by the two baseline methods is always smaller than that of the proposed

method. This suggests our method is truly robust even with mis-specified number of classes. As number of periods L increases, the purity increases and converges to 1, which confirms our theoretical results. When time shift is considered, the two baselines can only give very low purity values while the result given by our proposed method is still quite descent. According to the construction of outliers, our method seems to be more effective when the outliers tend to consist of more events (i.e., outlier type 1 and type 2 have larger intensity values). Moreover, our methodology is universally effective under all the three TPP working models. This suggests that our robust estimation procedure is flexible and can be integrated into a very wide range of parametric or nonparametric models. In the supplementary E.4, we further provide ablation studies to illustrate the effectiveness of each component in the proposed algorithm.

6. Real Data Application

To illustrate the usefulness of the proposed methodology, we apply our method to two real data sets, IPTV and Last.fm.

IPTV dataset The IPTV log-data set (Luo et al., 2014) used in our study are collected from a large-scale Internet Protocol television (IPTV) provider, China Telecom, in Shanghai, China. As a privacy protection, anonymous data is used in this study. The log-data records viewing behaviors of users, which is composed of anonymous user logs, time stamps (which are at the precision of one second) of the beginnings and the endings of viewing sessions. The log-data is family-based and each family has only one user ID. For the family with more than one

Television, all viewing behaviors are also recorded under the same user account. The data collector randomly selected 302 users from the data set and collected their household structures and their watching history from 2012 January 1st to 2012 November 30th through phone surveys with the help of China Telecom. On average, each household has $10 - 15$ events per day.

We do some preprocessing on the IPTV data. By exploratory analysis, we can see strong evidence that households' watching behavior is periodic with a period equal to 24 hours (i.e. $T = 24$). For each household, we construct an event sequence with the number of periods $L = 7$ based on the raw data as follows. Let period $l \in \{1, 2, ...7\}$ correspond to Monday, Tuesday, ..., Sunday. We choose the working model to be the non-homogeneous Poisson process, which enjoys the independent increment property. Thus superposition of sub event sequences in different periods will not affect the estimation results. We then superpose data from 5 randomly selected days (Mondays, ..., Sundays) into each period. Those households with insufficient data are excluded. In the end, we construct $N = 297$ clean event sequences with $T = 24$ and $L = 7$. The choices of tuning parameters are specified the same as those in the simulation studies.

Since we do not know the true underlying class labels for each household, the purity index cannot be computed. Instead, we use two other criteria to compare the performance between the proposed algorithm and baseline methods. For the first one, we define

$$\mathrm{L1}_n = \int_0^T \left|\hat{\lambda}_n(t) - \hat{\lambda}^*_{k(n)}(t)\right| dt / \sqrt{\int_0^T \hat{\lambda}^*_{k(n)}(t)dt}, \tag{6.23}$$

where $k(n)$ is the estimated label of sample n, the $\hat{\lambda}_n(t)$ is the estimated intensity function of sample i via cubic spline approximation, and $\hat{\lambda}^*_{k(n)}(t)$ is the estimated intensity function of class $k(n)$. In (6.23), the normalizer $\sqrt{\int_0^T \hat{\lambda}^*_{k(n)}(t)dt}$ is the estimated standard deviation of the number of events for class $k(n)$. This helps to eliminate the influence of intensity magnitudes of different classes. Then the L1 error criterion is given by

$$\text{L1-error} = \frac{1}{N_{in}} \sum_{n \notin \hat{\mathcal{S}}_{outlier}} \text{L1}_n, \tag{6.24}$$

where $\hat{\mathcal{S}}_{outlier}$ is the index set of outlier returned by the proposed method (i.e. the sample with w_{nk}'s smaller than 0.1 is treated as the outlier) and $N_{in} = N - |\hat{\mathcal{S}}_{outlier}|$.

For the second one, we define the MLE index of n-th event stream as $\text{MLE}_n(alg) := \log \text{NHP}(S_n | \boldsymbol{B}^{alg}_{k(n)^{alg}})$, where the superscript "alg" indicates one of the three algorithms. We compute the MLE comparison ratio as

$$\text{MLE}_{out}(alg_1, alg_2) = \frac{1}{N_{in}} \sum_{n \notin \hat{\mathcal{S}}_{outlier}} \mathbf{1}\{\text{MLE}_n(alg_1) > \text{MLE}_n(alg_2)\}, \tag{6.25}$$

$$\text{MLE}_{all}(alg_1, alg_2) = \frac{1}{N} \sum_{n \in [N]} \mathbf{1}\{\text{MLE}_n(alg_1) > \text{MLE}_n(alg_2)\}. \tag{6.26}$$

If the index $\text{MLE}_{out}(alg_1, alg_2)$ (or $\text{MLE}_{all}(alg_1, alg_2)$) is larger than 0.5, then it indicates that "alg_1" performs better than "alg_2".

From Table 2, we can see that the proposed algorithm achieves the smallest L1-error among all the three algorithms under any choice of $K \in \{3, ..., 8\}$. This suggests the clusters returned by our method are more compact. From Table 2, we also see that the MLE comparison ratios of the proposed method against others are uniformly greater than 0.5. This indicates that the inclusion of influence function

ϕ and K-means++ type initialization indeed makes an improvement on majority of the samples.

Last.FM 1K User Dataset Last.fm 1K is a public data set released by lastfm (Òscar Celma, 2010). It collects all listening history records (about 20 million records) of 992 users of different countries from July 2005 to May 2009. The data contains two tables. The record table includes information such as userID, event timestamp, artistID, artist_name, songID, and song_name, while the user feature table includes information such as gender, age, country, registration time, etc. On average, each user has about 40 events per day.

Similar to IPTV data, we also do the preprocessing on the Last.fm data. From Figure 4, we again see the evidence that users' song track playing frequency is periodic with $T = 24$ hours. The size of raw data is huge so that we down-sample the data and construct the event sequence for each user with $L = 10$. That is, we extract event streams from 10 randomly selected days for each user. After discarding those users with insufficient data, we have 966 users left. In other words, we construct $N = 966$ clean event sequences with $T = 24$ and $L = 10$. Since users may come from different countries, we consider the time shift in this data set. Again, the choice of ρ and ϵ is the same as before.

From Table 3, we can also see that the proposed algorithm performs the best among all the three methods in terms of both L1-error and MLE comparison ratio. This confirms the generality of the proposed method. Both influence function and initialization procedure contribute to the improvement of performance .

7. Conclusion

In the current literature, there is no work studying the clustering of event stream data under the outlier setting. In this work, we make an effort to solve this task and propose a robust TPP clustering framework. Our algorithm can be viewed as a non-parametric method that builds on the cubic spline regression. There are two key ingredients in the new approach. One is the construction of a TPP-specific distance function which can be efficiently implemented. The other is the incorporation of Catoni's influence function which allows us to have robust parameter training. Under mild assumptions, the proposed method is shown to have decent performance. Theories on convergence property, (non) asymptotic error bound, and outlier detection have been established. Three different types of outliers are considered in the simulations and the results validate the effectiveness of the proposed method. Two real data applications are provided. Our algorithm achieves the superior performance over the other two baseline methods.

Lastly, we discuss a few potential extensions in the future work. (i) In the current work, we introduce a new distance function based on cubic spline regression. It is possible to design other types of metric which can also be computed efficiently. (ii) In the "fine-tuning" step, we construct the pseudo likelihood function based on nonparametric working models. It can further be replaced by neural network-based models. (iii) The current definition of outliers is individual/user-level. However, in practice, it could happen that a user behaves normally for almost all time but except for a very short period. Therefore, it may be improper to treat the whole

event sequence as the outlier. Instead, we should consider the problem on the event-level. (iv) The proposed method empirically works well under any choice of K. The selection of the number of clusters is not within the scope of the present work. In the future, it may still be desired to design an optimal guideline for choosing the best number of clusters for practitioners. (v) Current literature on TPP generally assumes fixed intervals or fixed time periods, with limited exploration of heterogeneous temporal periods observed in real-world scenarios like customer activity logs or medical monitoring data. It is desired to design better clustering methods when users have different lengths of activity periods.

Supplementary Material The online material contains technical proofs, more simulation results, explanations and discussions.

Acknowledgment. The authors would like to thank the Associate Editor and the two anonymous referees for their constructive suggestions and comments, which helped improve the quality of the paper. Guanhua Fang is partly supported by the National Natural Science Foundation of China (Grant No. 12301376) and Shanghai Educational Development Foundation (Grant No. 23CGA02). Wen Yu is partially supported by the National Natural Science Foundation of China (Grant No. 12071088).

References

Arthur, D. and S. Vassilvitskii (2007). K-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035.

Berndt, D. J. and J. Clifford (1994). Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining*, pp. 359–370.

REFERENCES

Bhatt, S., G. Fang, P. Li, and G. Samorodnitsky (2022). Minimax m-estimation under adversarial corruption. In *Proceedings of the 39th International Conference on Machine Learning (ICML), Bartimore, MD*.

Blei, D. M., A. Kucukelbir, and J. D. McAuliffe (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association 112*(518), 859–877.

Bradley, P. S. and U. M. Fayyad (1998). Refining initial points for k-means clustering. In *ICML*, Volume 98, pp. 91–99. Citeseer.

Brown, E. N., R. Barbieri, V. Ventura, R. E. Kass, and L. M. Frank (2002). The time-rescaling theorem and its application to neural spike train data analysis. *Neural computation 14*(2), 325–346.

Bubeck, S., N. Cesa-Bianchi, and G. Lugosi (2013). Bandits with heavy tail. *IEEE Transactions on Information Theory 59*(11), 7711–7717.

Cai, B., J. Zhang, and Y. Guan (2022). Latent network structure learning from high-dimensional multivariate point processes. *Journal of the American Statistical Association*, 1–14.

Cao, J., X. Lin, X. Cong, S. Guo, H. Tang, T. Liu, and B. Wang (2021). Deep structural point process for learning temporal interaction networks. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part I 21*, pp. 305–320. Springer.

Carneiro, M. J. T. et al. (2011). Towards the discovery of temporal patterns in music listening using last. fm profiles.

Casella, G. and R. L. Berger (2021). *Statistical inference*. Cengage Learning.

Catoni, O. (2012). Challenging the empirical mean and empirical variance: a deviation study. In *Annales de l'IHP Probabilités et statistiques*, Volume 48, pp. 1148–1185.

Daley, D. J., D. Vere-Jones, et al. (2003). *An introduction to the theory of point processes: volume I: elementary theory and methods*. Springer.

De Boor, C. (1972). On calculating with b-splines. *Journal of Approximation theory 6*(1), 50–62.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological) 39*(1), 1–22.

Deshpande, A., P. Kacham, and R. Pratap (2020). Robust k-means++. In *Conference on Uncertainty in Artificial Intelligence*, pp. 799–808. PMLR.

Diebolt, J. and E. H. Ip (1995). Stochastic em: method and. *Markov chain Monte Carlo in practice*, 259.

Eiter, T. and H. Mannila (1994). Computing discrete fréchet distance.

REFERENCES

Enguehard, J., D. Busbridge, A. Bozson, C. Woodcock, and N. Hammerla (2020). Neural temporal point processes for modelling electronic health records. In *Machine Learning for Health*, pp. 85–113. PMLR.

Fang, G., P. Li, and G. Samorodnitsky (2023). Empirical risk minimization for losses without variance. *arXiv preprint arXiv:2309.03818*.

Fang, G., G. Xu, H. Xu, X. Zhu, and Y. Guan (2023). Group network hawkes process. *Journal of the American Statistical Association* (just-accepted), 1–78.

Fraley, C. and A. E. Raftery (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association 97*(458), 611–631.

Georgogiannis, A. (2016). Robust k-means: a theoretical revisit. *Advances in Neural Information Processing Systems 29*.

Gupta, M., J. Gao, C. C. Aggarwal, and J. Han (2013). Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and data Engineering 26*(9), 2250–2267.

Hosseini, S. A., K. Alizadeh, A. Khodadadi, A. Arabzadeh, M. Farajtabar, H. Zha, and H. R. Rabiee (2017). Recurrent poisson factorization for temporal recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 847–855.

Hsu, D. and S. Sabato (2016). Loss minimization and parameter estimation with heavy tails. *The Journal of Machine Learning Research 17*(1), 543–582.

Huber, P. J. (1992). Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer.

Huber, P. J. (2004). *Robust statistics*, Volume 523. John Wiley & Sons.

Lehmann, E. L. and G. Casella (2006). *Theory of point estimation*. Springer Science & Business Media.

Lugosi, G. and S. Mendelson (2021). Robust multivariate mean estimation: the optimality of trimmed mean. *The Annals of Statistics 49*, 393–410.

Luo, D., H. Xu, H. Zha, J. Du, R. Xie, X. Yang, and W. Zhang (2014). You are what you watch and when you watch: Inferring household structures from iptv viewing data. *IEEE Transactions on Broadcasting 60*(1), 61–72.

Luo, D., H. Xu, Y. Zhen, X. Ning, H. Zha, X. Yang, and W. Zhang (2015). Multi-task multi-dimensional hawkes processes for modeling event sequences.

McLachlan, G. J., S. X. Lee, and S. I. Rathnayake (2019). Finite mixture models. *Annual review of statistics and its application 6*, 355–378.

Pei, T., X. Gong, S.-L. Shaw, T. Ma, and C. Zhou (2013). Clustering of temporal event processes. *Inter-*

national Journal of Geographical Information Science 27(3), 484–510.

Peng, J. and H.-G. Müller (2008). Distance-based clustering of sparsely observed stochastic processes, with applications to online auctions.

Pillow, J. (2009). Time-rescaling methods for the estimation and assessment of non-poisson neural encoding models. *Advances in neural information processing systems 22*.

Prasad, A., S. Balakrishnan, and P. Ravikumar (2020). A robust univariate mean estimator is all you need. In *International Conference on Artificial Intelligence and Statistics*, pp. 4034–4044. PMLR.

Sani, M. F., S. J. van Zelst, and W. M. van der Aalst (2019). Repairing outlier behaviour in event logs using contextual behaviour. *Enterprise Modelling and Information Systems Architectures (EMISAJ) 14*, 5–1.

Schütze, H., C. D. Manning, and P. Raghavan (2008). *Introduction to information retrieval*, Volume 39. Cambridge University Press Cambridge.

Wang, D., X. Zhang, Y. Wan, D. Yu, G. Xu, and S. Deng (2021). Modeling sequential listening behaviors with attentive temporal point process for next and next new music recommendation. *IEEE Transactions on Multimedia 24*, 4170–4182.

Xu, H., G. Fang, Y. Chen, J. Liu, and Z. Ying (2018). Latent class analysis of recurrent events in problem-solving items. *Applied Psychological Measurement 42*(6), 478–498.

Xu, H. and H. Zha (2017). A dirichlet mixture model of hawkes processes for event sequence clustering. *Advances in neural information processing systems 30*.

Xu, L., J. A. Duan, and A. Whinston (2014). Path to purchase: A mutually exciting point process model for online advertising and conversion. *Management Science 60*(6), 1392–1412.

Yan, J. (2019). Recent advance in temporal point process: from machine learning perspective. *SJTU Technical Report*.

Yin, L., G. Xu, H. Sang, and Y. Guan (2021). Row-clustering of a point process-valued matrix. *Advances in Neural Information Processing Systems 34*, 20028–20039.

Zhang, Y., J. Yan, X. Zhang, J. Zhou, and X. Yang (2022). Learning mixture of neural temporal point processes for multi-dimensional event sequence clustering. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, Vienna, Austria*, pp. 23–29.

Òscar Celma (2010, March). lastfm music recommendation dataset.

Tables and Figures

Outlier type	Time	Algorithm	No shift			shift		
			$K=4$	$K=5$	$K=6$	$K=4$	$K=5$	$K=6$
type 1	$L=1$	Standard	0.5056	0.6111	0.7480	0.3868	0.4506	0.5046
		Robust	0.5225	0.6438	0.7590	0.4198	0.4896	0.5172
		Robust & Initialization	**0.9026**	**0.9758**	**0.9797**	**0.6420**	**0.6678**	**0.6910**
	$L=2$	Standard	0.4648	0.5495	0.6688	0.3857	0.4740	0.5351
		Robust	0.4849	0.6090	0.7205	0.4046	0.5023	0.5739
		Robust & Initialization	**0.9240**	**0.9916**	**0.9988**	**0.7313**	**0.7728**	**0.7910**
	$L=4$	Standard	0.3950	0.4725	0.5650	0.3703	0.4581	0.5368
		Robust	0.4051	0.5153	0.6550	0.3958	0.4900	0.5921
		Robust& Initialization	**0.9150**	**0.9925**	**1**	**0.7610**	**0.8130**	**0.8147**
type 2	$L=1$	Standard	0.3996	0.5283	0.6302	0.3132	0.3481	0.3856
		Robust	0.5835	0.6859	0.8017	0.3901	0.4442	0.4636
		Robust & Initialization	**0.9520**	**0.9796**	**0.9791**	**0.6544**	**0.6838**	**0.6939**
	$L=2$	Standard	0.4239	0.5246	0.6019	0.3057	0.3573	0.4180
		Robust	0.5445	0.6440	0.7115	0.3784	0.4548	0.5095
		Robust & Initialization	**0.9264**	**0.9838**	**0.9988**	**0.7431**	**0.7681**	**0.8141**
	$L=4$	Standard	0.4025	0.4950	0.5798	0.3197	0.3784	0.4169
		Robust	0.4975	0.5725	0.6625	0.4235	0.4963	0.5374
		Robust & Initialization	**0.9225**	**0.9850**	**1**	**0.7969**	**0.8026**	**0.8233**
type 3	$L=1$	Standard	0.8975	0.9733	0.9764	0.4520	0.4978	0.5152
		Robust	0.8623	0.9613	0.9774	0.4560	0.5006	0.5198
		Robust & Initialization	**0.9161**	**0.9753**	**0.9783**	**0.6420**	**0.6418**	**0.6853**
	$L=2$	Standard	0.9069	0.9882	0.9907	0.4810	0.5169	0.5467
		Robust	0.8811	0.9656	0.9887	0.4874	0.5240	0.5568
		Robust & Initialization	**0.9592**	**0.9928**	**0.9984**	**0.6366**	**0.7167**	**0.7695**
	$L=4$	Standard	0.8873	0.9624	0.9875	0.5042	0.5348	0.5611
		Robust	0.8750	0.9525	0.9900	0.5151	0.5450	0.5818
		Robust & Initialization	**0.9574**	**0.9900**	**1**	**0.6735**	**0.7356**	**0.8083**

Table 1: Purity indices returned by three algorithms under the setting of outlier type 1, 2, and 3 with non-homogeneous Poisson working model.

Clusters	MLE comparison ratio						L1-error		
	Ours vs. Standard		Ours vs. Robust		Robust vs. Standard		Ours	Robust	Standard
	Out	All	Out	All	Out	All			
$K = 3$	67.70	61.61	55.25	56.57	66.15	60.61	**2.678**	2.682	2.989
$K = 4$	66.54	59.25	55.25	51.85	60.70	56.90	**2.505**	2.633	2.689
$K = 5$	64.09	58.25	57.92	55.56	62.93	58.26	**2.439**	2.506	2.570
$K = 6$	64.63	59.60	58.17	55.89	61.60	56.90	**2.389**	2.462	2.557
$K = 7$	64.71	60.27	53.31	53.20	66.54	61.95	**2.386**	2.415	2.503
$K = 8$	67.16	61.61	56.34	55.55	61.94	57.91	**2.232**	2.364	2.454

Table 2: MLE comparison ratios and L1-error indices given by all three methods for IPTV data.

Clusters	MLE comparison ratio						L1-error		
	Ours vs. Standard		Ours vs. Robust		Robust vs. Standard		Ours	Robust	Standard
	Out	All	Out	All	Out	All			
$K = 3$	64.57	62.11	58.94	59.32	59.26	56.83	**2.246**	2.482	2.585
$K = 4$	69.41	66.56	66.81	64.29	54.59	52.28	**2.146**	2.373	2.409
$K = 5$	65.10	62.63	62.70	62.11	57.14	54.55	**2.123**	2.308	2.338
$K = 6$	62.15	60.25	57.74	56.63	53.23	51.55	**1.984**	2.127	2.180
$K = 7$	61.19	59.42	59.57	57.97	55.78	53.73	**1.934**	2.119	2.158
$K = 8$	61.17	59.32	54.34	52.59	57.38	56.11	**1.925**	2.103	2.175

Table 3: MLE comparison ratios and L1-error indices given by all three methods for Last.FM data.

Algorithm 1 Robust clustering

Input Sequences $S = \{s_n\}_{n=1}^N$, tolerance parameter ϵ.

—— **Initialize clusters** ——

Run Algorithm 2 to get r_{nk}^{ini} (and $\{\text{shift}_n^{ini}\}$ if necessary).

Compute initial $\mathbf{B}^{(0)}$ by maximizing $\mathcal{L}(\mathbf{B})$ specified in (3.10) with $r_{nk}^{(0)}$ replaced by r_{nk}^{ini}.

Compute initial $\pi_k^{(0)} = \sum_n r_{nk}^{ini} / \sum_{n,k} r_{nk}^{ini}$ and set $t = 0$.

—— **Fine-tune clusters** ——

repeat

  Compute $r_{nk}^{(t)}$ according to Eq.(3.9).

  Compute $\pi_k^{(t)} = \sum_{n=1}^N r_{nk}^{(t)}/N$.

  Compute $w_{nk}^{(t)} = \phi'_\rho \left(\log \text{NHP}\left(s_n \mid \boldsymbol{B}_k^{(t-1)}\right) / L(S_n) - \hat{\mu}_\phi(\boldsymbol{B}_k^{(t-1)}) \right)$.

  Update $\boldsymbol{B}_k^{(t)}$ according to Eq.3.15.

  Update the shift parameter according to Eq. (3.16), if necessary.

  Increase t by one.

until $\|\boldsymbol{B}_k^{(t)} - \boldsymbol{B}_k^{(t-1)}\| \leq \epsilon, \ \forall k \in [K]$.

Output: $\hat{\mathbf{B}}$, $\{\hat{r}_{nk}\}$.

Algorithm 2 Robust Initialization

Input: Data $\mathbf{S} = \{s_n\}_{n=1}^N$ and tuning parameters α, β, $N'(< N)$

Outlier Screening: set $\mathcal{S}_{in} = \emptyset$.

repeat

For event stream n not in $\mathcal{S}_{in}$, compute $\mathcal{D}_n^{(t)}$ and $q_{n,\alpha}^{(t)}$ according to (3.17). Rank the quantiles $q_{n,\alpha}$'s in the increasing order and add the first $\min\{\lfloor \beta \cdot |\mathbf{S} \backslash \mathcal{S}_{in}| \rfloor, N' - |\mathcal{S}_{in}|\}$ samples into $\mathcal{S}_{in}$.

until $|\mathcal{S}_{in}| \geq N'$.

Inlier weighting: follow steps (a)-(c) to get K centers $\{c_1, ..., c_K\}$.

Compute the weight matrix $\{r_{nk}\}$'s according to (3.18).

Compute the initial shift parameter shift_n of S_n according to (3.19), if necessary.

Output: Weight matrix $\{r_{nk}\}$, shift parameters $\{\text{shift}_n\}$, inlier set $\mathcal{S}_{in}$; centers $\mathcal{C}^{ini}$.