1-27-2020

A change-point detection and clustering method in the recurrent-event context

Qing Li
Iowa State University, qlijane@iastate.edu

Kehui Yao
University of Wisconsin - Madison

Xinyu Zhang
North Carolina State University

A change-point detection and clustering method in the recurrent-event context

Abstract

Change-point detection in the context of recurrent-event is a valuable analysis tool for the identification of the intensity rate changes. It has been an interesting topic in many fields, such as medical studies, travel safety analysis, etc. If subgroups exist, clustering can be incorporated into the change-point detection to improve the quality of the results. This paper develops a new algorithm named Recurrent-K-means to detect the change-points of the intensity rates and identify clusters of objects with recurrent events. It also proposes a test-based method to perform a heuristic search in determining the number of underlying clusters. In this study, the objects are assumed to fall in several clusters while the objects in the same cluster share identical change-points. The event count for an object is assumed to be a non-homogeneous Poisson process with a piecewise-constant intensity function. The methodology estimates the change-point as well as the intensity rates before and after the change-point for each cluster. The methodology establishes a clustering analysis based on K-means algorithm but enhances the procedure to be model based. The simulation study shows that the methodology performs well in parameter estimation and determination of the number of clusters in different scenarios. The methodology is applied to the UK coal mining disaster data to show its possible role in shaping government regulations and improving coal industry safety.

Keywords

K-means, maximum likelihood estimate, non-homogeneous Poisson process, parametric bootstrap, piecewise-constant intensity

Disciplines

Applied Statistics | Systems Engineering

Comments

This is an Accepted Manuscript of an article published by Taylor & Francis in Journal of *Statistical Computation and Simulation* on January 27, 2020, available online: DOI: 10.1080/00949655.2020.1718149. Posted with permission.

A change-point detection and clustering method in the recurrent-event context

Qing Li[1], Kehui Yao[2], and Xinyu Zhang[3]

[1]Department of Industrial & Manufacturing Systems Engineering, Iowa State University, Ames, IA, USA
[2]Department of Statistics, University of Wisconsin-Madison, Madison, WI, USA
[3]Department of Statistics, North Carolina State University, Raleigh, NC, USA

Published in Journal of Statistical Computation and Simulation in 2020

Abstract

Change-point detection in the context of recurrent-event is a valuable analysis tool for identification of the intensity rate changes. It has been an interesting topic in many fields, such as medical studies, travel safety analysis, etc. If subgroups exist, clustering can be incorporated into the change-point detection to improve the quality of the results. This paper develops a new algorithm named Recurrent-K-means to detect the change-points of the intensity rates, and identify clusters of objects with recurrent events. It also proposes a test-based method to perform heuristic search in determining the number of underlying clusters. In this study, the objects are assumed to fall in several clusters while the objects in the same cluster share identical change-points. The event count for an object is assumed to be a non-homogeneous Poisson process with a piecewise constant intensity function. The methodology estimates the change-point as well as the intensity rates before and after the change-point for each cluster. The methodology establishes a clustering analysis based on K-means algorithm but enhances the procedure to be model based. The simulation study shows that the methodology performs well in parameter estimation and determination of the number of clusters in different scenarios. The methodology is applied to the UK coal mining disaster data to show its possible role in shaping government regulations and improving coal industry safety.

KEY WORDS: K-means, maximum likelihood estimate, non-homogeneous Poisson process, parametric bootstrap, piecewise constant intensity

[1]Address correspondence to Qing Li: Department of Industrial & Manufacturing Systems Engineering, Iowa State University, Ames, IA, USA; E-mail: qlijane@iastate.edu

1. Introduction

Recurrent-event data analysis is widely used in various fields such as reliability, medical studies, social science and criminology when an object has multiple events. An interesting problem in recurrent-event data analysis is change-point detection that aims to identify the times, when the intensity rates change. For example, the intensity rates of drivers might change over time as they have more experience and learn from driving education programs [19]; the rate of recurrent disease episodes may change because of a treatment or the effects of treatment wearing out [9]; the rate of coal accidents may change because of government regulations or the effects of the market need for coal [23]; the rate of machine malfunction changes because of aging [26]. The change-point reveals critical information on the recurrence patterns, and provides reference for research such as similarities and heterogeneity among objects.

Most of the literature on change-point detection in the recurrent-event context assumed that the event counts follow a non-homogeneous Poisson process (NHPP). The NHPP is a Poisson process whose intensity function is not a constant over time [24, p. 32]. Examples include detecting the change-points in the ozone level by a Bayesian method [5], and proposing non-parametric estimators for the change-point when there were multiple subjects [10].

NHPPs with piecewise-constant intensity functions are widely used for change-point detection in the recurrent-event context because of the simplicity and robustness [17]. For example, Achcar et al. [1], Gupta and Baker [12], Montoya-Noguera and Wang [21], Raftery and Akman [23], West and Odgen [32] developed Bayesian methods to detect the change-points and conducted model selection on the number of change-points for one object; Frobish and Ebrahimi [9], Li et al. [19] proposed maximum likelihood estimators (MLEs) of change-points for multiple subjects.

In recurrent-event change-point analysis, clustering the objects is necessary when the recurrence patterns of the objects vary. Research in this area is limited. Most of the existing clustering algorithms cannot be used in the recurrent-event context directly and the adjustments needed are usually not straightforward. Li et al. [18] detected the change-points and clustered teenage drivers with recurrent events by their risk profiles using a Bayesian finite mixture model, which was distribution-based clustering. Such type of methods are relatively complex, computationally expensive, and relies heavily on parametric assumptions as well as the convergence of Monte Carlo Markov chains.

Clustering objects with recurrent events can also be thought as clustering object-specific curves, where the curve for each object can be the cumulative number of events versus time. Clustering curves is relatively a new study area. Perets [22] clustered the lines in two dimensional Euclidean space by centroid-based clustering. Dass et al. [6] proposed a non-parametric Bayesian approach to cluster curves with change-points in trends.

The centroid-based clustering is another commonly used clustering technology. This family of clustering algorithm needs a distance or similarity measure to measure the dissimilarity or similarity among objects. One of the most popular centroid-based clustering methods is K-means clustering, which iteratively finds a fixed num-

ber of centroids and assigns objects to the nearest centroid in order to find a partition that minimizes the within-cluster sum of squares [20]. Original K-means algorithm can not be applied directly to the recurrent-event clustering problem because it is hard to define centroids and a distance measure. Another challenge for the K-means algorithm is to determine the number of clusters.

Existing methods on detecting the number of clusters include direct methods which optimize a predefined criterion, statistical testing methods which compare evidence against null hypothesis, Bayes factor [8], and information-based methods such as **the Akaike information criterion (AIC) [2] and the Bayesian information criterion (BIC) [27]**. Notice that these information criteria consist of a log-likelihood item and a penalty on the number of parameters. The log-likelihood term is often much larger than the penalty term in the recurrent-event context, meaning that the information value is dominated by the log-likelihood. Thus using information-based method to choose the number of clusters in the recurrent-event context is not proper.

The optimal number of clusters relies on the similarity measures and the parameters used for clustering, and is subjective in some sense [15]. None of the existing methods for detecting the number of clusters work universally well [29].

In this paper, we advance the original K-means algorithm to Recurrent-K-means to incorporate clustering in recurrent-event change-point analysis by defining the centroids with three key parameters of the recurrence pattern and using a likelihood-based measure as the similarity measure. We also propose a heuristic searching method based on parametric bootstrap and a hypothesis test to determine the number of clusters. We assume that there are multiple objects and the recurrent event counts follow NHPPs with piecewise-constant intensity rates. We further assume that clusters exist among the objects and objects in the same cluster share identical change-points.

Our algorithm clusters the objects and provides the estimates of change-point and intensity rates before and after the change-point for each subgroup simultaneously. It can determine the number of clusters automatically as well. The proposed methodology is novel and straightforward to implement, can be easily applied to other problems, and is demonstrated to work well and outperform AIC and BIC in the recurrent-event context by simulation in Section 3.

The rest of the article is organized as follows. In Section 2, we develop a variation of K-means clustering to detect the change-points and cluster the objects in the recurrent-event context, and we propose a test-based method to detect the number of clusters. The Simulation study is in Section 3. A real data analysis is provided in Section 4. Section 5 is the conclusion and discussion.

2. A Recurrent-k-means method for change-point detection and clustering in the recurrent-event context

This section presents an novel algorithm which combines the K-means clustering algorithm and the likelihood-based change-point detection method in the recurrent-event context in Frobish and Ebrahimi [9], Li et al. [18]. The original K-means cannot be directly used in the recurrent-event context because the number of events vary

across objects and the similarity measure or distance measure cannot be calculated in this situation. We also propose a test-based method to detect the number of clusters.

Assume that there are m objects from K groups with recurrent events, and each cluster has an underlying change-point. The objects in the same cluster share identical change-point and intensity rates. We first show how to estimate the change-point and intensity rates for a cluster by maximizing the likelihood. Then we propose how to cluster the objects when the number of clusters K is given. Lastly we show how to automatically detect the number of clusters using a test-based method. Notice that the objects with no events are excluded from the analysis, because no information is provided for the change-point in such case. In addition, the proposed approach does not distinguish different types of events in the analysis. If it is not reasonable to combine different types of events, the proposed approached need to be adjusted.

Denote $n_j \geq 1$ to be the total number of events and c_j be the total follow up time for the j^{th} object, $j = 1, 2, \cdots, m$. Notice that c_j's can be different as the objects might have varying ends of histories. c_j will be used as the censoring time in the analysis. These events occurred at ordered times $0 < t_{j1} < t_{j2} < \cdots < t_{jn_j}$. We assume these m objects fall in K groups, and the group index is k, $k = 1, \cdots, K$. If the j^{th} object is from group k, we denote it as $j \in G_k$.

Denote the number of events for object j till time t to be $\{N_j(t), t \geq 0\}$, which is a counting process. One of the commonly used counting processes is the Poisson process, which can be described by a non-negative integrable intensity function over time t. When the intensity function of the Poisson process is not a constant across time, it is an NHPP.

We assume that the event counts of the objects in group k follow an NHPP with piecewise-constant intensity function $\lambda_k(t) = \lambda_{kb} I(0 \leq t < \mu_k) + \lambda_{ka} I(t \geq \mu_k)$, where μ_k is the unknown change-point for the k^{th} group and $\mu_k \leq \min_{j \in G_k}\{c_j\}$. $I(t)$ is the indicator function. λ_{kb} is the intensity rate before μ_k, and λ_{ka} is the intensity rate after μ_k. Integrating it yields the cumulative intensity function of group k:

$$\Lambda_k(t) = \lambda_{kb} t I(0 \leq t < \mu_k) + \left[\lambda_{kb}\mu_k + \lambda_{ka}(t - \mu_k)\right] I(t \geq \mu_k). \tag{1}$$

Let $n_j^{(b)}$ be the number of events for the j^{th} object before the change-point, and $n_j^{(a)}$ be the number of events after the change-point. Then $n_j^{(b)} + n_j^{(a)} = n_j$. **Notice that for a Poisson process, $N_j(t)$ follows a Poisson distribution: $N_j(t) \sim Poisson(\Lambda_k(t))$.** Table 1 gives a summary of the notations.

2.1. Change-point detection by maximizing the likelihood

We assume that all the objects in the same cluster share identical intensity rates and change-point. Here we summarize the maximum likelihood estimators (MLEs) for μ_k, λ_{kb} and λ_{ka} proposed by Frobish and Ebrahimi [9], Li et al. [19].

Table 1. Notations in this paper.

Symbol	Meaning
m	The total number of objects
K	The total number of groups
j	The object index, $j = 1, 2, \cdots, m$
n_j	The total number of events for the j^{th} object, $n_j \geq 1$
c_j	The follow up time for the j^{th} object
i	The event index, $i = 0, 1, 2, \cdots, n_j$, where $i = 0$ indicates the starting point
t_{ji}	The i^{th} event time for the j^{th} object, assuming $t_{ji_1} \neq t_{ji_2}$ for $\forall i_1 \neq i_2$
x_{ji}	The inter-event time: $x_{ji} = t_{ji} - t_{j,(i-1)}$
k	The group index, $k = 1, 2, \cdots, K$
N_k	The number of objects in the k^{th} group
$\Lambda_k(t)$	The cumulative intensity function for the k^{th} group over time t
$n_j^{(b)}$	The total number of events for the j^{th} object before the change-point
$n_j^{(a)}$	The total number of events for the j^{th} object after the change-point
μ_k	The change-point for the k^{th} group
τ_j	The change-point for the j^{th} object
λ_{kb}	The intensity rate before the change-point for the k^{th} cluster
λ_{ka}	The intensity rate after the change-point for the k^{th} cluster
λ_{bj}	The intensity rate before the change-point for the j^{th} object
λ_{aj}	The intensity rate after the change-point for the j^{th} object

The likelihood of object j given that it is in group k [30] is:

$$L_j(\mu_k, \lambda_{kb}, \lambda_{ka}|\boldsymbol{X}_j) = exp[-\Lambda_k(c_j)] \prod_{i=1}^{n_j} \lambda_k(t_{ji}) = exp[-\Lambda_k(c_j)] \lambda_{kb}^{n_j^{(b)}} \lambda_{ka}^{n_j^{(a)}},$$

where $\boldsymbol{X}_j = (t_{j1}, \cdots, t_{jn_j}, c_j)$. Denoting $\boldsymbol{X}_{(k)}$ to be the event times and censoring times in group k, the total log-likelihood of the N_k objects in this group **assuming conditional independence among objects** is

$$\begin{aligned} logL_{(k)}(\mu_k, \lambda_{kb}, \lambda_{ka}|\boldsymbol{X}_{(k)}) = &- (\lambda_{kb} - \lambda_{ka}) N_k \mu_k - \lambda_{ka} \sum_{j \in G_k} c_j \\ &+ \left(\sum_{j \in G_k} n_j^{(b)} \right) log\lambda_{kb} + \left(\sum_{j \in G_k} n_j^{(a)} \right) log\lambda_{ka}. \end{aligned} \tag{2}$$

Notice that $\left(\sum_{j \in G_k} n_j^{(b)}\right) + \left(\sum_{j \in G_k} n_j^{(a)}\right) = \sum_{j \in G_k} n_j$. Taking partial derivatives of $logL_{(k)}$ over the intensity rates and setting them to zero, we obtain the MLEs for intensity rates:

$$\hat{\lambda}_{kb} = \frac{\sum_{j \in G_k} n_j^{(b)}}{\mu_k N_k}, \hat{\lambda}_{ka} = \frac{\sum_{j \in G_k} n_j^{(a)}}{\sum_{j \in G_k} c_j - \mu_k N_k}. \tag{3}$$

The MLEs of the intensity rates are the average number of events per object per

unit time. The profile log-likelihood $logL_{(k)}(\mu_k, \hat{\lambda}_{kb}, \hat{\lambda}_{ka}|\boldsymbol{X}_{(k)})$ can be obtained by plugging the MLEs of intensity rates in Eq. (3) into Eq. (2):

$$\begin{aligned} logL_{(k)}(\mu_k, \hat{\lambda}_{kb}, \hat{\lambda}_{ka}|\boldsymbol{X}_{(k)}) = &- \left(\frac{\sum_{j\in G_k} n_j^{(b)}}{\mu_k N_k} - \frac{\sum_{j\in G_k} n_j^{(a)}}{\sum_{j\in G_k} c_j - \mu_k N_k} \right) N_k \mu_k \\ &- \frac{\sum_{j\in G_k} n_j^{(a)}}{\sum_{j\in G_k} c_j - \mu_k N_k} \sum_{j\in G_k} c_j + \left(\sum_{j\in G_k} n_j^{(b)} \right) log(\frac{\sum_{j\in G_k} n_j^{(b)}}{\mu_k N_k}) \\ &+ \left(\sum_{j\in G_k} n_j^{(a)} \right) log \left(\frac{\sum_{j\in G_k} n_j^{(a)}}{\sum_{j\in G_k} c_j - \mu_k N_k} \right). \end{aligned}$$

According to Frobish and Ebrahimi [9], the value of μ_k that maximizes the profile log-likelihood $logL_{(k)}(\mu_k, \hat{\lambda}_{kb}, \hat{\lambda}_{ka}|\boldsymbol{X}_{(k)})$ locates at one of the event times, and the MLE of μ_k is consistent **on a predefined interval** $[\mu_l, \mu_u]$:

$$\hat{\mu}_k = argmax_{\mu_k = t_{ji}, \mu_l \le \mu \le \mu_u | j\in G_k, 1\le i\le n_j} logL_{(k)}(\mu_k, \hat{\lambda}_{kb}, \hat{\lambda}_{ka}|\boldsymbol{X}_{(k)}). \tag{4}$$

So $\hat{\mu}_k$ can be found by plugging all the event times in $[\mu_l, \mu_u]$ from this group into $logL_{(k)}(\mu_k, \hat{\lambda}_{kb}, \hat{\lambda}_{ka}|\boldsymbol{X}_{(k)})$ and choosing the event time with the maximum profile log-likelihood. Because the change-points are positive, zero can be a natural lower bound. An upper bound μ_u is required for the consistency of the MLE of μ_k. This upper bound can be chosen based on the experience or intuition such as the minimum censoring time. If an upper bound less than the minimum censoring time is known before data collection, that would improve the accuracy of the estimation. We use the minimum censoring time of group k as the upper bound for μ_k. Plugging $\hat{\mu}_k$ back into Eq. (3), we get the numerical values for the intensity rates $\lambda_{kb}, \lambda_{ka}$:

$$\hat{\lambda}_{kb} = \frac{\sum_{j\in G_k} n_j^{(b)}}{\hat{\mu}_k N_k}, \hat{\lambda}_{ka} = \frac{\sum_{j\in G_k} n_j^{(a)}}{\sum_{j\in G_k} c_j - \hat{\mu}_k N_k}. \tag{5}$$

If further flexibility is needed, the object-specific intensity rates can be estimated by:

$$\hat{\lambda}_{bj} = \frac{n_j^{(b)}}{\hat{\mu}_k}, \hat{\lambda}_{aj} = \frac{n_j^{(a)}}{c_j - \hat{\mu}_k}, j \in G_k. \tag{6}$$

2.2. Recurrent-K-means clustering

The K-means algorithm cannot be directly used in the recurrent-event context because the observational unit j, $j = 1, \cdots, m$, has a sequence of event times, and the number of events varies among objects. We advance the K-means clustering algorithm to cluster m objects into K clusters in the recurrent-event context where K is given. The proposed algorithm is named Recurrent-K-means.

Our method is using a likelihood-based similarity measure. Suppose there are K underlying groups, and the centroid of group k is

defined by the change-point and intensity rates $(\mu_k, \lambda_{kb}, \lambda_{ka})$. Denote $P_{jk} = f(\boldsymbol{X}_j|\mu_k, \lambda_{kb}, \lambda_{ka})$, which is the likelihood of $(\mu_k, \lambda_{kb}, \lambda_{ka})$ giving data $\boldsymbol{X}_j$ and that object j falls in group k:

$$P_{jk} = exp[-\Lambda(c_j)]\lambda_{kb}^{n_j^{(b)}} \lambda_{ka}^{n_j^{(a)}}. \tag{7}$$

$log(P_{jk})$ is used as the similarity measure between object j and centroid k. Larger values of $log(P_{jk})$ indicate more similarity while smaller values of it indicate less similarity.

The iterative optimization procedure for K centroids follow the same scheme as in Hartigan and Wong [13]. Starting from K initial centroids, we assign each object to the centroid with the largest similarity, and then update the centroids using Eqs. (4)–(5). We repeat the process until the partition does not change. The details of the Recurrent-K-means is as follows:

2.2.1. Initialization

Denote the initialized K centroids as $(\mu_k^{(0)}, \lambda_{kb}^{(0)}, \lambda_{ka}^{(0)}), k = 1, \cdots, K$. Like the original K-means problem, it does not guarantee a global maximum solution. The K-means algorithm yields better results when the initial partition is dispersed and close to the final result [14]. The intuition behind initialization is to spread out the initial centroids but to avoid outliers. We propose a procedure as follows to obtain $(\mu_k^{(0)}, \lambda_{kb}^{(0)}, \lambda_{ka}^{(0)}), k = 1, \cdots, K$.

Step 1: Calculate the MLE of the change-point $\hat{\tau}_j$ of each object as a special case that each cluster only has one object using Eq. (4), $j = 1, \cdots, m$.

Step 2: Given the number of cluster K, group the objects based on the estimated change-points $\hat{\tau}_1, \cdots, \hat{\tau}_m$ from Step 1 using the K-means algorithm [20].

Step 3: Calculate the centroids using Eqs. (4)–(5) as the initial centroids $(\mu_k^{(0)}, \lambda_{kb}^{(0)}, \lambda_{ka}^{(0)})$.

2.2.2. Updating

Starting from $t = 1$,

Step 1: Given $(\mu_k^{(t-1)}, \lambda_{kb}^{(t-1)}, \lambda_{ka}^{(t-1)}), k = 1, ...K$, assign object j to the cluster with the largest log-likelihood $log(P_{jk}), j = 1, \cdots, m$.

Step 2: Update the centroids using Eqs. (4)–(5). $t = t + 1$. Notice that the objects with no events are excluded from the analysis, because the MLE of the change-point cannot be calculated in this case.

Repeat the two steps for updating until the partition does not change. The resulted partition will be the final clustering. The K centroids output in the last iteration will be the estimates of our cluster centroids. The proposed Recurrent-K-means has similar computational complexity as the original K-means. In our simulation and real data analysis, it usually takes two to three iterations for the Recurrent-K-means to converge. In practice, the user can define the maximum number of iterations to make the algorithm have linear complexity.

2.3. Optimal number of clusters

In Section 2.2, we cluster m observations into K groups when K is given. However, K is usually unknown in practice. We propose a heuristic searching method in this section, combining the idea of parametric bootstrapping and hypothesis testing. We fit models with different numbers of clusters and then conduct hypothesis testing to choose the best number of clusters. For a given positive integer K, the null hypothesis is that the number of clusters is K, and the alternative is that the number of clusters is larger than K.

Define $P_{(K)}(\boldsymbol{X}) = \prod_{k=1}^{K} \prod_{j \in G_k} P_{jk}$ to be the total likelihood to cluster $\boldsymbol{X}$ into K groups by the Recurrent-K-means method in Section 2.2, where $\boldsymbol{X} = (\boldsymbol{X}_1, \ \cdots, \ \boldsymbol{X}_m)$. The natural log of the likelihood ratio $Y_{(K)}(\boldsymbol{X}) = log\frac{P_{(K+1)}(\boldsymbol{X})}{P_{(K)}(\boldsymbol{X})}$ that $\boldsymbol{X}$ has $K+1$ clusters against K clusters can be used as the test statistic.

The asymptotic properties of $Y_{(K)}(\boldsymbol{X})$ is complex. We can obtain a random sample from the distribution of the test statistic by parametric bootstrapping. For parametric bootstrapping, a parametric model is fitted to the data and new samples are generated from the fitted model [7]. Denote $\boldsymbol{C}_{(K)}(\boldsymbol{X}) = (\boldsymbol{C}_1, \ \cdots, \ \boldsymbol{C}_K)$, where $\boldsymbol{C}_k = (\mu_k, \ \lambda_{kb}, \ \lambda_{ka})$ is the centroid of the k^{th} cluster in data set $\boldsymbol{X}$. Then define $\boldsymbol{X}_{\boldsymbol{C}_{(K)}} = (\boldsymbol{X}_1, \ \cdots, \ \boldsymbol{X}_m)$ as the random data set of m objects generated from $\boldsymbol{C}_{(K)}$. Appendix A provides the details on how to generate a data set given $\boldsymbol{C}_{(K)}$, the size of each cluster, and the censoring times of the objects. The following steps show how to obtain a random sample from the distribution of the test statistic $Y_{(K)}(\boldsymbol{X})$ by parametric bootstrapping. Firstly, calculate the observed test statistic $Y_{(K)}(\boldsymbol{X})$ based on the original data $\boldsymbol{X}$. Secondly, estimate the centroids $\hat{\boldsymbol{C}}_{(K)}(\boldsymbol{X}) = (\hat{\boldsymbol{C}}_1, \ \cdots, \ \hat{\boldsymbol{C}}_K)$. Thirdly, generate B data sets with the same number of objects, cluster sizes and censoring times as $\boldsymbol{X}$ based on $\hat{\boldsymbol{C}}_{(K)}$ denoted as $\boldsymbol{X}^{(1)}_{\hat{\boldsymbol{C}}_{(K)}}, \ \cdots, \ \boldsymbol{X}^{(B)}_{\hat{\boldsymbol{C}}_{(K)}}$, which is parametric bootstrapping. Lastly, calculate $Y_{(K)}(\boldsymbol{X}^{(l)}_{\hat{\boldsymbol{C}}_{(K)}}), \ l = 1, \ \cdots, \ B$ on each random data set, which forms a random sample from the distribution of $Y_{(K)}(\boldsymbol{X})$.

We propose the following method to test whether the number of clusters is K based on the random sample from the distribution of $Y_{(K)}(\boldsymbol{X})$, which is quite intuitive and computationally efficient. Using the random sample of the test statistic by parametric bootstrapping, we calculate the average of B samples as an estimate of the test statistic expectation: $\bar{Y}_{(K)} = \frac{1}{B}\sum_{l=1}^{B} Y_{(K)}(\boldsymbol{X}^{(l)}_{\hat{\boldsymbol{C}}_{(K)}})$. Let $sd(Y_{(K)}(\boldsymbol{X}))$ be the standard deviation of the random sample. Taking account of the simulation error, the adjusted standard error is proposed to be $s_K = \sqrt{1 + \frac{1}{B}sd(Y_{(K)}(\boldsymbol{X}))}$. We reject the null hypothesis that there are K clusters when observing $Y_{(K)}(\boldsymbol{X}) \geq \bar{Y}_{(K)}(\boldsymbol{X}) + s_K$. The '1-standard-error' type of rule was used elsewhere such as Breiman [3]. In simulation studies in Section 3, the '1-standard-error' method works well. **Notice that a probability of more extreme test statistic can also be estimated from the sample of the test statistic**.

Assuming the lower bound of the number of clusters by expert knowledge or user preference is K_0, the procedure to determine the number of clusters is as follows. Starting from $K = K_0$, we test whether the number of clusters is K using the '1-

standard-error' method above. If the test rejects the null hypothesis, set $K = K+1$ and test again. We stop updating K until the test fails to reject the null hypothesis. The latest value of K will be used as the final number of clusters. We do not test all the possible values for K but stop when the test fails to reject the null hypothesis to be consistent with Occam's razor principle that the simpler model is better. If it is believed that the number of clusters is relatively small, $K_0 = 1$ is recommended.

For the purpose of comparison, we also used the silhouette analysis [25], AIC, and BICto determine the number of clusters. The major difficulty of using silhouette analysis or other methods like Gap statistics [31] to find the number of clusters is how to define the distance in the recurrent-event context. We propose a distance metric similarly as Euclidean distance and the details of silhouette analysis in the recurrent-event context is in Appendix B.The AIC value of a model can be calculated by $AIC = 2 * p - 2ln(\hat{L})$ [2], while BIC is $BIC = k * ln(n) - 2ln(\hat{L})$[27], where p is the number of estimated parameters in the model, $\hat{L}$ is the maximum likelihood function of the model, and n is the sample size. Smaller AIC or BIC values indicate better model. To find the optimal number of clusters, the proposed method uses parametric bootstrapping and is relatively more computationally expensive than AIC and BIC.

3. Simulations

We ran simulations to check the performance of the methodology proposed in Section 2 in different scenarios. Data were generated from NHPPs with piecewise-constant intensity functions according to the distribution of the inter-event times [16]. The details for data generation are in Appendix A.

3.1. Simulation settings

Table 2 shows the 12 settings for data generation with different change-points, intensity rates, cluster sizes, censoring time, and the number of clusters. For ease of presentation, all the intensity rates are multiplied by 1,000.

Setting 1 is the reference setting and we checked how assumptions in the data would affect the results. The total number of objects m are 40 except for Setting 2. The censoring time c_j is uniformly distributed from 450 to 500 except having a larger variation in Setting 6. The objects are equally likely to be from different clusters except for Setting 5. The objects from the same cluster share identical change-point and intensity rates except for Settings 7–8. There are two clusters except for Settings 11–12. The change-points are different while the intensity rates are the same among clusters except for Settings 9-10.

For each setting, we generated $T = 200$ data sets, and conducted model selection and estimation separately. For model selection, each data set was resampled 200 times by parametric bootstrapping to determine the number of clusters using the method in Section 2.3. The initial number of clusters is $K_0 = 1$. The percentages that the number of clusters is correctly detected (P_1) was calculated to evaluate the model selection performance. The 95% confidence interval was also obtained from these resampled 200 data sets as explained in Appendix C. For parameter estimation,

Table 2. Twelve settings for data generation.

Setting	Description
1	There are two clusters ($K = 2$). The number of objects is $m = 40$. The objects are equally likely to be from two clusters with centroids $(\mu_1, \lambda_{1b}, \lambda_{1a}) = (150, 250, 100)$ and $(\mu_2, \lambda_{2b}, \lambda_{2a}) = (300, 250, 100)$ respectively. The censoring time is uniformly generated from 450 to 500: $c_j \sim Unif[450, 500]$. For the purpose of presentation, all the intensity rates are multiplied by 1,000.
2	The same as Setting 1 except that the number of objects is $m = 80$.
3	The same as Setting 1 except that the change-point of the second cluster is different: $\mu_2 = 200$. The change-points are closer.
4	The same as Setting 1 except that the centroids of two clusters become $(150, 250, 200)$ and $(300, 250, 200)$ respectively. The change in the intensity rates is smaller.
5	The same as Setting 1 except for unbalanced cluster sizes: $(N_1 = 30,\ N_2 = 10.)$ There are much more objects in the first cluster.
6	The same as Setting 1 except that the censoring time $c_j \sim Unif[\tau_j + 10, 500]$. τ_j is the change-point for object j. The variation in the censoring time is larger.
7	The same as Setting 1 except that the change-points of objects in the same cluster are slightly different. τ_j is sampled from $Unif[145, 155]$ if the object is from the first cluster, and from $Unif[295, 305]$ otherwise.
8	The same as Setting 1 except that the intensity rates are slightly different among objects: $\lambda_{bj} \sim Gamma(25, 100)$, $\lambda_{aj} \sim Gamma(10, 100)$, where $Gamma(a, b)$ is a Gamma distribution with an expectation of a/b.
9	The same as Setting 1 except that the centroids of the two clusters are $(150, 250, 100)$ and $(300, 100, 250)$. The intensity rates of the two clusters are different.
10	The same as Setting 1 except that the centroids of the two clusters are $(150, 250, 100)$ and $(150, 100, 250)$. The change-points of the two clusters are the same.
11	The same as Setting 1 except that the objects are equally from three clusters with change-points 100, 200, and 300 respectively. The intensity rates before and after the change-point are 250 and 100.
12	The same as Setting 1 except that the objects are equally from four clusters with change-points 100, 150, 200, and 250 respectively. The intensity rates before and after the change-point are 250 and 100.

the true number of clusters was given and the framework in Section 2.2 was used on each data set to estimate the centroids. The root-mean-square error (RMSE) for a parameter θ was calculated by $\sqrt{(1/T)\sum_{t=1}^{T}(\widehat{\theta}-\theta)^2}$. The absolute percentage bias |bias (%)| was calculated by $(1/T)\sum_{t=1}^{T}|\widehat{\theta}-\theta|/\theta \times 100\%$. The average percentages of correctly grouped objects (P_2) given the correct number of clusters was calculated. The coverage probability by the 95% confidence interval was also provided.

3.2. Simulation results

For model selection, the percentages that the number of clusters is correctly detected (P_1) by the method in Section 2.3, the silhouette analysis, AIC, and BIC are shown in Table 3. The details of the silhouette analysis are in Appendix B. For estimation given the correct number of clusters, the mean percentages of correctly grouped objects (P_2) are shown in Table 3, and the other estimation results by the algorithm in Section 2.2 are shown in Tables 4–6.

Table 3. Two percentages multiplied by 100 for all the simulation settings. For model selection, P_1(New) is the percentage that the number of clusters is correctly detected by the method proposed in Section 2.3, P_1(Silhouette) by silhouette analysis, P_1(AIC) by AIC and P_1(BIC) by BIC. For parameter estimation given the correct number of clusters, P_2 is the average percentage of correctly grouped objects.

Setting	1	2	3	4	5	6
P_1(New)	97.5	97.5	80.0	67.5	92.5	100.0
P_1(Silhouette)	80.0	85.0	57.5	62.5	77.5	77.5
$P_1(AIC)$	5.0	17.5	30.0	0.0	15.0	17.5
$P_1(BIC)$	52.5	37.5	22.5	10.0	15.0	17.5
P_2	97.71	98.65	88.05	77.73	97.77	99.50
Setting	7	8	9	10	11	12
P_1(New)	97.5	85.0	97.5	100.0	77.5	80.0
P_1(Silhouette)	85.0	77.5	75.0	80.0	40.0	25.0
$P_1(AIC)$	27.5	45.0	35.0	2.5	40.0	25.0
$P_1(BIC)$	27.5	45.0	42.5	5.0	25.0	22.5
P_2	99.10	92.35	99.10	100.00	75.00	77.31

Considering $P_1(New)$ and P_2 in Table 3, both of the two percentages decrease slightly in Setting 3 when the change-points of the two clusters are closer, in Setting 4 when the intensity rates before and after the change-point are closer, in Setting 8 when there are variations in the intensity rates among objects, and in Settings 11–12 when there are more than two clusters. Only P_1(New) decreases slightly in Setting 5 when the cluster sizes are quite different. Both of the two percentages are high

Table 4. Estimation results of Settings 1–5 giving the correct number of clusters. The number of objects is $m = 80$ for Setting 2, and $m = 40$ for other settings. The censoring time is uniformly generated from 450 to 500: $c_j \sim Unif[450, 500]$. There are two clusters. The change-point is μ_k for cluster $k, k = 1, 2$. The intensity rates before and after the change-point are λ_{kb} and λ_{ka}, respectively. The coverage probability was estimated by the 95% credible interval.

Setting	Parameter	True value	Average of estimates	RMSE	|Bias (%)|	Coverage probability (%)
1	μ_1	150	149.01	2.23	0.66	95.0
	μ_2	300	300.08	1.31	0.03	90.0
	λ_{1b}	250	249.90	1.27	0.04	97.5
	λ_{2b}	250	252.87	0.71	1.15	95.0
	λ_{1a}	100	100.18	0.37	0.18	92.5
	λ_{2a}	100	99.59	0.65	0.41	97.5
2	μ_1	150	149.78	0.81	0.14	90.0
	μ_2	300	299.98	1.00	0.01	92.5
	λ_{1b}	250	253.01	0.69	1.20	97.5
	λ_{2b}	250	249.56	0.50	0.17	90.0
	λ_{1a}	100	99.78	0.29	0.22	92.5
	λ_{2a}	100	99.45	0.36	0.55	92.5
3	μ_1	150	149.10	2.39	0.60	97.5
	μ_2	200	199.45	2.81	0.28	97.5
	λ_{1b}	250	250.92	1.03	0.37	95.0
	λ_{2b}	250	253.51	1.09	1.40	95.0
	λ_{1a}	100	97.56	0.63	2.44	97.5
	λ_{2a}	100	100.97	0.65	0.97	100.0
4	μ_1	150	138.49	16.57	7.67	100.0
	μ_2	300	291.98	24.28	2.67	100.0
	λ_{1b}	250	237.58	2.34	4.97	100.0
	λ_{2b}	250	259.63	1.37	3.85	100.0
	λ_{1a}	200	183.26	1.68	8.37	100.0
	λ_{2a}	200	199.39	0.26	0.30	100.0
5	μ_1	150	150.17	3.73	0.11	100.0
	μ_2	300	300.51	1.25	0.17	95.0
	λ_{1b}	250	247.16	1.24	1.13	100.0
	λ_{2b}	250	250.40	0.52	0.16	100.0
	λ_{1a}	100	99.74	0.61	0.26	97.5
	λ_{2a}	100	100.64	0.43	0.64	87.5

Table 5. Estimation results of Settings 6–10 giving the correct number of clusters. The number of objects is $m = 40$. The censoring time is uniformly generated from 450 to 500 except for Setting 6. There are two clusters. The change-point is μ_k for cluster $k, k = 1, 2$. The intensity rates before and after the change-point are λ_{kb} and λ_{ka}, respectively. For Setting 7, $\bar{\mu}_1$ indicates the average change-point value for the first cluster. For Setting 8, $\bar{\lambda}_{1b}$ indicates the average intensity rate before the change-point for the first cluster. The coverage probability was estimated by the 95% credible interval.

Setting	Parameter	True value	Average of estimates	RMSE	|Bias (%)|	Coverage probability (%)
6	μ_1	150	149.90	2.88	0.06	87.5
	μ_2	300	299.55	1.05	0.15	77.5
	λ_{1b}	250	250.46	0.88	0.18	92.5
	λ_{2b}	250	251.12	0.63	0.45	97.5
	λ_{1a}	100	98.29	0.43	1.71	100.0
	λ_{2a}	100	101.80	0.54	1.80	95.0
7	$\bar{\mu}_1$	150	151.05	2.07	0.70	97.5
	$\bar{\mu}_2$	300	300.64	2.72	0.21	87.5
	λ_{1b}	250	246.90	0.85	1.24	90.0
	λ_{2b}	250	247.99	0.56	0.80	90.0
	λ_{1a}	100	100.79	0.42	0.79	87.5
	λ_{2a}	100	102.21	0.46	2.21	97.5
8	μ_1	150	148.81	2.39	0.79	100.0
	μ_2	300	298.76	2.32	0.41	92.5
	$\bar{\lambda}_{1b}$	250	249.61	1.02	0.16	92.5
	$\bar{\lambda}_{2b}$	250	248.46	1.09	0.62	95.0
	$\bar{\lambda}_{1a}$	100	93.89	1.03	6.11	97.5
	$\bar{\lambda}_{2a}$	100	101.99	0.86	1.99	95.0
9	μ_1	150	150.10	2.27	0.07	97.5
	μ_2	300	300.16	1.34	0.05	90.0
	λ_{1b}	250	253.56	0.98	1.42	77.5
	λ_{2b}	100	101.59	14.85	1.59	85.0
	λ_{1a}	100	101.74	0.52	1.74	87.5
	λ_{2a}	250	248.31	14.85	0.67	92.5
10	μ_1	150	150.80	1.89	0.53	90.0
	μ_2	150	150.22	0.95	0.15	85.0
	λ_{1b}	250	246.96	0.01	1.22	95.0
	λ_{2b}	100	99.68	0.00	0.32	95.0
	λ_{1a}	100	99.89	0.00	0.11	95.0
	λ_{2a}	250	248.38	0.01	0.65	95.0

Table 6. Estimation results of Settings 11–12 giving the correct number of clusters. The number of objects is $m = 40$. The censoring time is uniformly generated from 450 to 500. There are three clusters for Setting 11 and four clusters for Setting 12. The change-point is μ_k for cluster k. The intensity rates before and after the change-point are λ_{kb} and λ_{ka}, respectively. The coverage probability was estimated by the 95% credible interval.

Setting	Parameter	True value	Average of estimates	RMSE	|Bias (%)|	Coverage probability (%)
11	μ_1	100	99.92	0.73	0.08	100.0
	μ_2	200	200.86	2.57	0.43	100.0
	μ_3	300	299.58	2.06	0.14	97.5
	λ_{1b}	250	255.30	1.43	2.12	97.5
	λ_{2b}	250	246.81	0.41	1.27	100.0
	λ_{3b}	250	248.87	0.48	0.45	100.0
	λ_{1a}	100	100.08	0.31	0.08	100.0
	λ_{2a}	100	97.80	0.46	2.20	100.0
	λ_{3a}	100	102.51	0.88	2.51	97.5
12	μ_1	100	95.24	13.76	4.76	100.0
	μ_2	150	144.35	25.14	3.77	100.0
	μ_3	200	204.03	32.46	2.01	100.0
	μ_4	250	247.05	18.61	1.18	100.0
	λ_{1b}	250	257.89	2.92	3.16	100.0
	λ_{2b}	250	260.05	2.61	4.02	100.0
	λ_{3b}	250	250.16	2.37	0.06	100.0
	λ_{4b}	250	243.33	1.63	2.67	100.0
	λ_{1a}	100	99.27	0.90	0.73	100.0
	λ_{2a}	100	97.98	0.72	2.02	100.0
	λ_{3a}	100	94.67	1.17	5.33	100.0
	λ_{4a}	100	102.41	1.32	2.41	100.0

in other settings. Both percentages will decrease when there are less heterogeneity among clusters, because it is more difficult to cluster with less heterogeneity.

Comparing P_1(New) with P_1(Silhouette), P_1(AIC), and P_1(BIC) in Table 3, P_1(New) is consistently much higher than P_1 by other methods. The proposed method to determine the optimal number of clusters performs well because discrepancy from the true number of clusters violates the parametric assumptions and will result in more extreme test statistic. The proposed test statistic is sensitive to the violation of assumptions. Notice that the AIC and BIC methods perform relatively bad on determining the number of clusters, mainly because the penalty terms of AIC and BIC are very small comparing with the log-likelihood in the recurrent-event context.

For the estimation results giving the correct number of clusters in Tables 4–6, the RMSE and $|Bias(\%)|$ of the change-points are higher in Setting 4 when the intensity rates before and after the change-point are closer, and in Setting 12 when the change-points are closer. The coverage probabilities by the 95% credible interval for the change-points are much lower in Setting 6 when there are larger variations in the censoring time. The RMSE and $|Bias(\%)|$ of the other settings are relatively small and the coverage probabilities are high.

Overall, our methodology performs well in detecting the number of clusters (P_1) and grouping the objects correctly under various situations giving the correct number of clusters (P_2) under our assumptions. It outperforms the silhouette analysis, AIC and BIC in detecting the number of clusters significantly. The larger the heterogeneity among clusters, the easier to cluster. The estimation is accurate given the correct number of clusters under most of the simulation settings, indicating that our method is robust and accurate in estimation.

Note: For a Poisson process, the number of events per object follows a Poisson distribution. Appendix D provides the means and standard deviations (SD) of number of events per subject in three simulation settings as a reference. All the simulations are implemented in MatLab R2016a. The code is available in https://github.com/KEHUIYAO/code.

4. Application

We applied the method in Section 2 to the coal-mining accidents data in UK from 1 January 1850 to 31 December 1901 obtained from the coal mining history resource center (https://archive.is/VM86o) as an illustration. The UK coal-mining disaster data was used for change-point detection in literature such as Raftery and Akman [23], Carlin et al. [4], Green [11]. Existing analysis of the problem didn't consider the heterogeneity among locations and detected the change-points in the disaster rate of UK as one unit.

This data set records the mining disasters of 24 location units, either a traditional shire or a geological prefecture, and there are 342 events in total. The minimum and maximum number of events per location are 2 and 40 correspondingly. The average number of events per location is 14.25 with the standard deviation to be 11.43. The censoring time is 31 December 1901.

The mining industry in UK experienced a peak development during this study

period and collapsed after 1970 [28]. The Second Industrial Revolution affected the coal mining industry, thus the rate of coal disasters. Therefore, it is reasonable to assume a change-point in the rate of coal disasters.

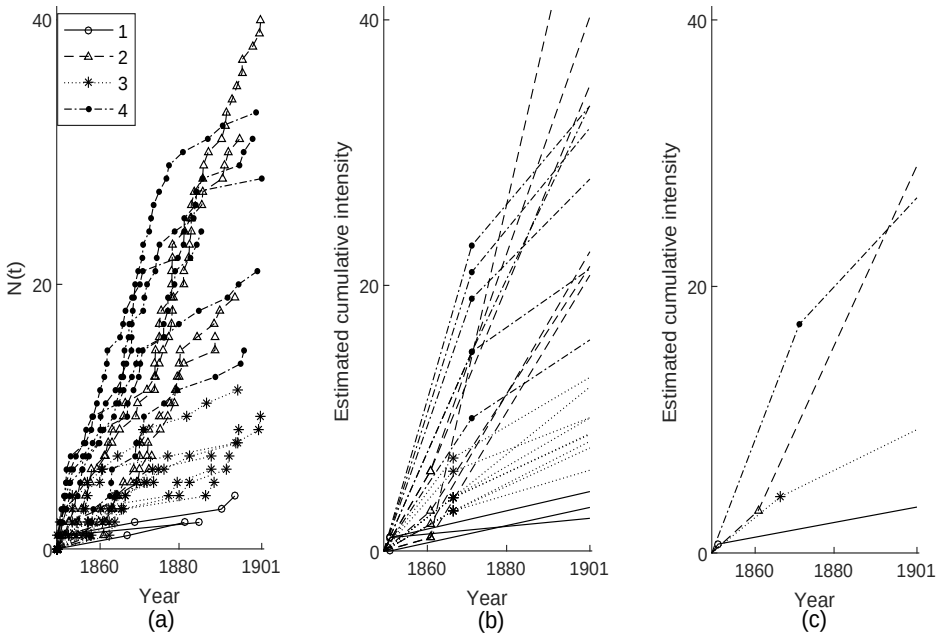

Figure 1. The results of coal-mining data with four clusters: (a) The cumulative event plot. The circle, triangle, star, and dot mark the disaster times by cluster. (b) The estimated cumulative intensity functions of the 24 locations. (c) The estimated cumulative intensity functions of the 4 clusters.

Table 7. The summaries of the coal mining data change-points. SE is the standard error. CI lower and CI upper are the bounds of the 95% confidence interval obtained by parametric bootstrap.

Cluster	Change-points	Mean	SE (Weeks)	CI lower	CI upper	Size
1	μ_1	08/12/1851	170.68	01/15/1850	02/12/1861	3
2	μ_2	10/01/1861	228.53	02/17/1852	08/27/1867	6
3	μ_3	06/04/1867	240.33	09/25/1860	06/10/1873	9
4	μ_4	05/12/1872	174.41	10/01/1867	01/07/1879	6

Three factors may affect the disaster rate. Firstly, whether on a coastal line or not is a crucial factor for the local industry. Those on coastal lines are likely to have more coal disasters since their coal industry provides coal not only for local use but also for global use by export. Secondly, the size of the coalfield, which implicates the maximum capacity of collieries, influences the disaster rate. Lastly, the geological proximity usually reflects the proximity of the economic and political background

Table 8. The means of the coal mining intensity rates. The intensity rate is the average number of events per location per 12 weeks scaled by 1,000.

Cluster	Parameter	Mean
1	λ_{1b}	95.24
	λ_{1a}	12.69
2	λ_{2b}	61.99
	λ_{2a}	148.01
3	λ_{3b}	55.74
	λ_{3a}	33.50
4	λ_{4b}	177.89
	λ_{4a}	73.27

of the region which influences the overall development of the coal mining industry. These 24 units vary in size, geological location, and coal industry development. Clusters might exist among locations.

Using the method in Section 2.3, the best number of clusters is four. That is, 24 locations could be divided into four groups. Figure 1(a) shows the cumulative event plot. The circle, triangle, star, and dot mark the disaster times by cluster. Figure 1(b) shows the estimated cumulative intensity functions of the 24 locations, which is consistent with Figure 1(a). The change-points were estimated first by the method in Section 2.2. Then the location-specific intensity rates were calculated by Eq. (6). Figure 1(c) is the estimated cumulative intensity functions of the 4 clusters. Table 7 shows the summary of change-points estimation, including the mean, standard error (SE), 95% confidence interval and the size of each cluster. The SE and 95% confidence interval were obtained by parametric bootstrapping (200 times) as shown in Appendix C. Table 8 shows the means of the estimated intensity rates. The average shows the average number of events per location per 12 weeks scaled by 1,000.

As Figure 1(c) shows, the intensity rates of the coalfields in Group 2 increase after the change-point while others decrease. The change-point of Group 2 is slightly ahead of the Second Industrial Revolution. The increase in the intensity rates of Group 2 is possibly caused by the drastic increase in the coal production. The change-point for Group 1 is small for its limited number of events. The change-points for Groups 3 and 4 are around 1870 when the Second Industrial Revolution started. Such a decreasing pattern reasonably corresponds to the technology development which usually helps avert dangers and thus decreases the risk.

Figure 2 visualizes the clustering result in the map of UK. Coalfields in Group 1 contains the least number of accidents, and as Figure 2 shows they are conterminous in the southern part of UK with geological proximity. Coalfields in groups 2 and 4 contain the most number of events, which is consistent with the fact that most of them locate along the coastal line. Group 3 mainly locates in the inland according to Figure 2.

Figure 3 (a) and (b) show the histograms of the location-specific intensity rates before and after the change-point with kernel fitting correspondingly. Figure 3 (c)

Figure 2. Clustering result visualization on the map of UK. The circle, triangle, star, and dot mark the four clusters correspondingly.

shows the scatter plot of the intensity rate before the change-point versus after the change-point. The intensity rates increase after the change-point for some locations and decrease for other locations.

In conclusion, the clustering results are reasonable and consistent with the historical and geological aspects.

Note: For the coal-mining data, the censoring times are the same among locations. Our methodology in Section 2 works for varying censoring times among objects which is a typical scenario in practice.

5. Discussion and conclusions

Recurrent-event change-point detection is an interesting topic in many fields such as transportation, reliability, and medical studies. Clustering can be incorporated into recurrent-event change-point detection to model the heterogeneity among objects.

We propose a Recurrent-K-means algorithm using the MLEs of key parameters for recurrent-event change-point analysis and a likelihood-based similarity to cluster the objects. We also propose a test-based method to determine the number of clusters, which outperforms AIC, BIC and silhouette analysis based on the simulation studies. The proposed methodology is straightforward to implement and is a computationally efficient alternative to the Bayesian finite mixture model (BFMM) in Li et al. [18]. The simulation studies show that the method is accurate in detecting the change-point and in clustering the objects under various scenarios. The results of the real data analysis are reasonable. Our proposed methods perform well in both of the simulation studies and the application. Notice that the definition of the

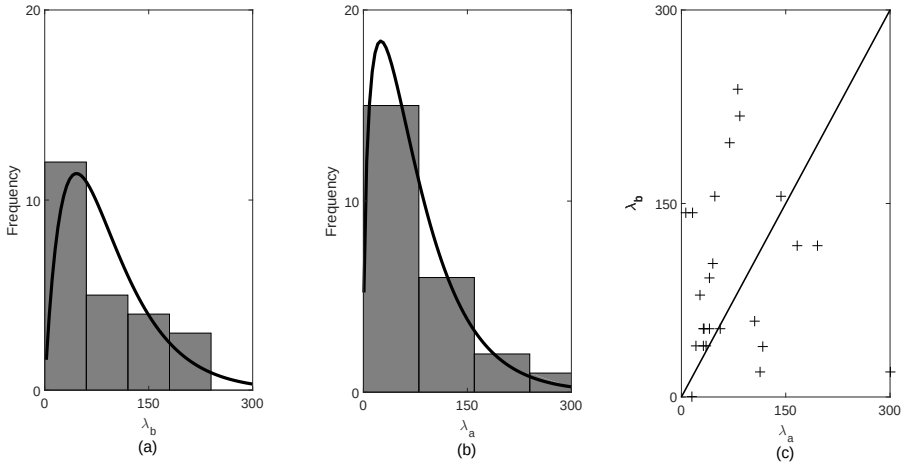

Figure 3. The location-specific intensity rates: (a) histogram of the intensity rates before the change-point ($\hat{\lambda}_{bj}$); (b) histogram of the intensity rates after the change-point ($\hat{\lambda}_{aj}$); (c) the scatter plot of $\hat{\lambda}_{bj}$ vs. $\hat{\lambda}_{aj}$. The straight line is the line where $\hat{\lambda}_{bj} = \hat{\lambda}_{aj}$.

clusters are restricted to what we have defined in this article. How it works in a broader context needs further exploration. In addition, our proposed method works well when the number of objects per cluster is not too small (larger than four, for example) based on our simulation. The intensity rates should neither be too small because only these event times provide information for the change-point. For the number of clusters, our simulation and application example have a relatively small number of clusters. We do not see a problem when there are much more clusters. But further research is needed to explore the method performance when there are much more clusters.

The methodology can be easily extended to other applications. The framework is relatively sensitive to outliers, so a future focus will be making the algorithm more robust to outliers. We also want to consider other forms of intensity functions to allow more flexibility of our method. Lawless and Zhan [17] suggested using piecewise constant intensity functions for the NHPP when true form is unknown because of the simplicity and robustness. If different intensity functions such as linear piecewise intensity are used, the formulators for the likelihood and the MLE of change-points will change slightly. The centroids of the Recurrent-K-means will have more than three parameters. The proposed algorithm would still be able to cluster the objects with recurrent events, however, the performance will need to be evaluated.

References

[1] Achcar, J., Loibel, S., and Andrade, M. (2007). Interfailure data with constant hazard function in the presence of change-points. *REVSTAT*, 5:209–226.

[2] Akaike, H. (1998). Information theory and an extension of the maximum likeli-

hood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. Springer, New York, NY.

[3] Breiman, L. (2017). *Classification and regression trees*. Chapman and Hall/CRC, Boca Raton, FL.

[4] Carlin, B. P., Gelfand, A. E., and Smith, A. F. (1992). Hierarchical bayesian analysis of changepoint problems. *Appl Stat*, 41:389–405.

[5] Cruz-Juárez, J. A., Reyes-Cervantes, H., and Rodrigues, E. R. (2016). Analysis of ozone behaviour in the city of puebla-mexico using non-homogeneous poisson models with multiple change-points. *J Environ Prot*, 7(12):1886.

[6] Dass, S. C., Lim, C. Y., Maiti, T., and Zhang, Z. (2015). Clustering curves based on change point analysis: A nonparametric bayesian approach. *Stat Sin*, 25:677–708.

[7] Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap methods and their application*, volume 1. Cambridge university press, Cambridge, United Kingdom.

[8] Fraley, C. and Raftery, A. E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *Comput J*, 41(8):578–588.

[9] Frobish, D. and Ebrahimi, N. (2009). Parametric estimation of change-points for actual event data in recurrent events models. *Comput Stat Data Anal*, 53:671–682.

[10] Frobish, D., Ebrahimi, N., and Pham, D. (2016). Semiparametric estimation of a change-point for recurrent events data. *Commun Stat, Simul Comput*, 45(9):3339–3349.

[11] Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732.

[12] Gupta, A. and Baker, J. W. (2015). A bayesian change point model to detect changes in event occurrence rates, with application to induced seismicity. In *12th International Conference on Applications of Statistics and Probability in Civil Engineering*, ICASP12, Vancouver, Canada.

[13] Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *J Royal Stat Soc, Ser C (Appl Stat)*, 28(1):100–108.

[14] Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc., Englewood Cliffs, NJ.

[15] Kassambara, A. (2017). *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*, volume 1. STHDA.

[16] Klein, R. W. and Roberts, S. D. (1984). A time-varying Poisson arrival process generator. *Simulation*, 43:193–5.

[17] Lawless, J. and Zhan, M. (1998). Analysis of interval-grouped recurrent-event data using piecewise constant rate functions. *Can J Stat*, 26:549–565.

[18] Li, Q., Guo, F., Kim, I., Klauer, S., and Simons-Monton, B. (2018). A Bayesian finite mixture change-point model for assessing the risk of novice teenage drivers. *J Appl Stat*, 45:604–625.

[19] Li, Q., Guo, F., Klauer, S., and Simons-Monton, B. (2017). Evaluation of risk change-point for novice teenage drivers. *Accid Anal Prev*, 108:139–146.

[20] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Trans Inf Theory*, 28(2):129–137.

[21] Montoya-Noguera, S. and Wang, Y. (2017). Bayesian identification of multiple seismic change points and varying seismic rates caused by induced seismicity. *Geophys Res Lett*, 44(8):3509–3516.

[22] Perets, T. (2011). *Clustering of lines*. Open University of Israel.

[23] Raftery, A. and Akman, V. (1986). Bayesian analysis of a poisson process with a change-point. *Biometrika*, pages 85–89.

[24] Ross, S. M. (2006). *Simulation (4th ed.)*. Academic Press, Cambridge, MA.

[25] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math*, 20:53–65.

[26] Ruggeri, F. and Sivaganesan, S. (2005). On modeling change points in non-homogeneous poisson processes. *Stat Inference Stoch Process*, 8(3):311–329.

[27] Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

[28] Seddon, M. (2013). The long, slow death of the uk coal industry. *Guardian*.

[29] Sugar, C. A. and James, G. M. (2003). Finding the number of clusters in a dataset: An information-theoretic approach. *J Am Stat Assoc*, 98(463):750–763.

[30] Thompson, W. (2012). *Point process models with applications to safety and reliability*. New York, NY.

[31] Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *J R Stat Soc, Ser B (Stat Methodol)*, 63(2):411–423.

[32] West, R. and Odgen, R. (1997). Continuous-time estimation of a change-point in a Poisson process. *J Stat Comput Simul*, 56:293–302.

Appendices

Appendix A. Data generation

This section shows how to generate a data set given the centroids (including the change-point, the intensity rates before and after the change-point), the size of each cluster, and the censoring times of the objects.

Given the ordered time to events $T_0 = t_0 = 0,\ T_1 = t_1,\ \cdots,\ T_i = t_i$, the cumulative density function (CDF) of the $(i+1)^{th}$ inter-event time $X_{i+1} = T_{i+1} - T_i$ for each object is:

$$\begin{aligned} F_{i+1}(x) &= Pr\left[X_{i+1} \leq x | T_p = t_p,\ p = 1, 2, \cdots, i\right] \\ &= 1 - exp\left[\Lambda(t_i) - \Lambda(t_i + x)\right], \end{aligned} \tag{8}$$

where $\Lambda(\cdot)$ is the cumulative intensity function of the NHPP in Eq. (1). Notice that $\Lambda(\cdot)$ is fully determined by the centroid of the corresponding cluster. Then $t_{i+1} = x_{i+1} + t_i$.

Starting from $i = 0$, the detailed algorithm to simulate the event times of one object is:

Step 1: Sample x_{i+1} from a distribution with CDF F_{i+1};

Step 2: Set $t_{i+1} = t_i + x_{i+1}$;

Step 3: Set $i = i + 1$, return to Step 1.

The above process is run until t_{i+1} is larger than the censoring time c_j. $t_1,\ t_2,\ \cdots,\ t_i$ are the ordered times to event for the j^{th} object. We follow the same procedure to generate the event times of all the objects.

Appendix B. The optimal number of clusters via silhouette analysis

This section illustrates how to determine the optimal number of clusters in the recurrent-event context by silhouette analysis.

Suppose the data is clustered into K groups based on the Recurrent-K-means method in Section 2. The major challenge to determine the optimal number of clusters in the recurrent-event context by silhouette analysis is how to define the distance metric between two objects. We propose to transform the data into the same dimension and use Euclidean distance as the distance metric.

Define $N_j(t)$ to be the total number of events till time t for object j and assume that $\{N_j(t),\ t \geq 0\}$ is a counting process. For any two objects j and k with censoring time c_j and c_k, randomly generate $t_1,\ \cdots,\ t_M$ from a uniform distribution $Unif(0,\ minimum(c_j, c_k))$. Then $(t_1,\ N_j(t_1)),\ \cdots,\ (t_M,\ N_j(t_M))$ can be used to represent object j, and $(t_1,\ N_k(t_1)),\ \cdots,\ (t_M,\ N_k(t_M))$ for object k. Here, M is a positive integer which is suggested to be greater than 100 to keep sufficient information from the original counting process. Define $D_{jk} = \sqrt{\Sigma_{i=1}^{M}[N_j(t_i) - N_k(t_i)]^2}$ to be the distance between object j and k. Then the average dissimilarity of an object j to a cluster c can be defined as the average of the distance from object j to all objects in c.

The typical procedure of silhouette analysis to find the optimal value of K is as follows [25]. The silhouette coefficient for object j is calculated by $s(j) = \frac{b(j)-a(j)}{maximum\{a(j),\ b(j)\}}$, where $a(j)$ is the average distance between object j and all other objects in the same cluster, and $b(j)$ is the smallest average dissimilarity of object j to any other clusters of which j does not belong to. The silhouette value ranges from -1 to 1 and measures how similar an object is to its own cluster compared with other clusters. A high silhouette value indicates that the object is better matched to its own cluster than other clusters. The average silhouette value of all objects can be used as a criterion to determine the number of clusters. For a given data set, we can try different values of K in a range and select the value with the highest average silhouette value as the optimal number of clusters. For the simulation study in Section 3, we tried K from 1 to 5.

Appendix C. The confidence interval and standard error estimation by parametric bootstrap

The following procedure shows how to obtain the confidence interval and standard errors of the parameters by parametric bootstrap in the recurrent-event context with piecewise constant intensity functions.

Firstly, estimate the centroids $\hat{\boldsymbol{C}}_{(K)}(\boldsymbol{X})$ of a data set $\boldsymbol{X}$ using Recurrent-K-means in Section 2.2, where $\boldsymbol{C}_{(K)}(\boldsymbol{X}) = (\boldsymbol{C}_1, \cdots, \boldsymbol{C}_K)$, and $\boldsymbol{C}_k = (\mu_k, \lambda_{kb}, \lambda_{ka})$ is the centroid of the k^{th} cluster. Secondly, generate B data sets with the same number of objects, cluster sizes and censoring times as $\boldsymbol{X}$ based on $\hat{\boldsymbol{C}}_{(K)}(\boldsymbol{X})$ denoted as $\boldsymbol{X}^{(1)}_{\hat{\boldsymbol{C}}_{(K)}}, \cdots, \boldsymbol{X}^{(B)}_{\hat{\boldsymbol{C}}_{(K)}}$, which is parametric bootstrapping. See Appendix A for the details about how to generate a data set given $\boldsymbol{C}_{(K)}$, the size of each cluster, and the censoring times of the objects. Thirdly, estimate the centroids of each new data set. Lastly, use the percentiles of the estimates as the bounds for the confidence intervals and the standard deviations as the standard error for the parameter estimates.

Appendix D. The distribution of number of events per object in the simulations

As shown in Appendix A, we generate a censoring time first for each object, then the event times are generated until the next event time will be larger than the censoring time. To give examples of the distribution of number of events per subject in simulation, we consider three situations with the censoring time $c_j \sim [450, 500]$, the number of objects to be 40 per situation, and the centroids $(\mu, \lambda_b, \lambda_a)$ to be $(150, 250, 100)$, $(300, 250, 100)$, and $(150, 250, 200)$ correspondingly. Table 9 shows the means and standard deviations (SD) of the number of events per subject in each setting.

Table 9. The means and SDs of the number of events per subject in three settings.

Setting	Centroids $(\mu, \lambda_b, \lambda_a)$	Mean	SD
1	(150, 250, 100)	70	8.5
2	(300, 250, 100)	93	9.9
3	(150, 250, 200)	102	10.4