

Analysis of Change Points With Bayes Factor, Thresholds, and CUSUM

Lucila Muñiz Merino¹, Bulmaro Juárez Hernández¹ & Hugo A. Cruz Suárez¹

¹ Facultad de Ciencias Físico Matemáticas, Universidad Autónoma de Puebla, Puebla, México

Correspondence: Facultad de Ciencias Físico Matemáticas, Universidad Autónoma de Puebla, Puebla, México, Postal address San Claudio, Cd. Universitaria, Jardines de San Manuel, 72572, Puebla, Pue

Received: January 21, 2021 Accepted: March 3, 2021 Online Published: March 24, 2021

doi:10.5539/ijsp.v10n3p31 URL: <https://doi.org/10.5539/ijsp.v10n3p31>

Abstract

In this work, an analysis of change points is made with the Bayes factor, thresholds, and cumulative sum (CUSUM) statistics methods. For the analysis of change points with the Bayes factor, Poisson data were simulated; the threshold method was worked with a regression and data of the National Institute of Statistics, Geography and Informatics (INEGI) of Mexico and coronavirus were used for the CUSUM.

Keywords: Bayes factor, CUSUM, thresholds

1. Introduction

When a change point is mentioned, the first question that comes to mind is: what is a change point? Chen and Gupta (2012) defined it as the site, or point in time t , in a succession of data $\{x_i\}$ $i = 1, \dots, n$ observed and ordered with respect to time, in such a way that these observations follow a distribution F_1 , before a point, and in another point after it, the distribution is F_2 . That is, from the statistical point of view, the succession of observations shows an inhomogeneous behaviour.

Under the classical and Bayesian approaches, the change point problem is considered one of the central problems of statistical inference, since it interrelates the statistical control theory, the hypothesis tests (when detecting if there is any change in the sequence of observed random variables), and the estimation theory (when estimating the number of changes and their corresponding locations). This under the classical and Bayesian approaches.

The change point problems originally appear in the quality control and generally can be found in the mathematical modeling of various disciplines such as Environmental Science, Epidemiology, Seismic Signal Processes, Economics, Finance, Geology, Medicine, Biology, Physics, etc. According to Chen and Gupta (2012), the change point problem is, in general, visualized as follows:

Let X_1, X_2, \dots, X_n be a succession of independent random vectors (or variables) with probability distribution functions F_1, F_2, \dots, F_n , respectively. The change point problem is to test the null hypothesis, so the problem of the point of change consists of testing the null hypothesis H_0 about the non-existence of change against the alternative, H_a that there is at least one change point:

$$H_0 : F_1 = F_2 = \dots = F_n$$

vs

$$H_a : F_1 = \dots = F_{(k_1)} \neq F_{(k_1+1)} = \dots = F_{(k_2)} \neq F_{(k_2+1)} = \dots = F_{(k_q)} \neq F_{(k_q+1)} = \dots = F_n.$$

Where $1 < k_1 < k_2 < \dots < k_q < n$, q is the unknown number of change points and k_1, k_2, \dots, k_q are the respective unknown positions that have to be estimated. If the distributions F_1, F_2, \dots, F_n become a common parametric family $F(\theta)$, where $\theta \in \mathbb{R}^p$, then the problem of change points is to test the null hypothesis H_0 about the non-existence of change in the parameters $\theta_i, i = 1, \dots, n$ of the population against the alternative H_a that there is at least one change point:

$$H_0 : \theta_1 = \theta_2 = \dots = \theta_n = \theta \text{ (unknown)}$$

vs

$$H_a : \theta_1 = \dots = \theta_{(k_1)} \neq \theta_{(k_1+1)} = \dots = \theta_{(k_2)} \neq \theta_{(k_2+1)} = \dots = \theta_{(k_q)} \neq \theta_{(k_q+1)} = \dots = \theta_n.$$

where q and k_1, k_2, \dots, k_q must be estimated. Together, these hypotheses reveal the inference aspects of the change points in order to determine if there is any change point in the process, as well as to estimate their number and their respective positions.

The objective of this work is to develop algorithms and programs to apply some procedures to detect change points, particularly the Bayes factor, the threshold and the cumulative sum statistics (CUSUM) methods.

In this work, some procedures were programmed in order to obtain change points, for this the Bayes factor was used to detect temporal and spatiotemporal change points. The homogeneous Poisson process was also used since data was simulated with this process. Multiple temporal-change points and spatiotemporal change points were detected with the Bayes factor. Changes were also detected with the threshold method and with the cumulative sum (CUSUM) method. Finally, the Buishand range, the Pettitt and the standard normal homogeneity tests were applied, which confirmed the change points detected by the cumulative sum method. For the application of the Bayes factor and the threshold methods, the concepts presented in Altieri (2015) were taken as the basis, Taylor (2000)’s article was used for the cumulative sum method.

In this work, the programs were elaborated with some instructions from the INLA package in Gómez (2020), Blangiardo and Cameletti (2015), except for the standard normal homogeneity, Pettitt and Buishand ranges tests, for which the Trend package (from R) of Pohlert (2020) was used.

Simulated data from a Poisson process as well as cement database, INEGI, and coronavirus data were used. The INEGI and cement data were taken from an example of regression in Gómez (2020), and coronavirus data was taken from the page *coronavirus.gob.mx*.

2. Materials and Methods

2.1 Bayes Factor

To detect the change points, the Bayes factor was used. The definition of the Bayes factor will be given below. The change points are determined according to the values in Table 1.

If there is a problem in which you must choose between two possible models, based on an observed data set D, the plausibility of the two different models M_1 and M_2 , parameterized by parameter vectors θ_1 and θ_2 , can be measured using the Bayes factor, which is defined as:

$$B = \frac{P(D|M_1)}{P(D|M_2)} = \frac{\int_{\theta_1} P(D|\theta_1, M_1)\Pi(\theta_1|M_1)d\theta_1}{\int_{\theta_2} P(D|\theta_2, M_2)\Pi(\theta_2|M_2)d\theta_1}$$

where $P(D|M_1)$ is called the marginal likelihood or integrated likelihood. This is similar to what is done in the likelihood ratio tests but now, instead of maximizing the likelihood, the Bayes factor performs a weighted average on the distribution of the parameters.

A value of $B > 1$ means that the data supports M_1 more than M_2 .

In the case of the Bayes factor, Jeffreys established an interpretation scale of B, this is shown in Table 1.

Table 1. Interpretation scale of B, according to Jeffreys

B	Strength of the evidence in favor of M_1
$B \leq 1$	Negative supports M_2
$1 < B \leq 3$	Very scarce
$3 < B \leq 10$	Substantial
$10 < B \leq 30$	Strong
$30 < B \leq 100$	Very strong
> 100	Decisive

Another way to consider the Bayes factor is as follows:

Suppose two hypotheses H_0 and H_1 such that the a priori probabilities are $f_0 = P(H_0)$ and $f_1 = P(H_1)$.

After observing a sample, the a posteriori probabilities of both hypotheses are $\alpha_0 = P(H_0|x)$ y $\alpha_1 = P(H_1|x)$. The Bayes factor in favor of H_0 is defined as:

$$B = \frac{\frac{\alpha_0}{\alpha_1}}{\frac{f_0}{f_1}} = \frac{\alpha_0 f_1}{\alpha_1 f_0}$$

Thus, the Bayes factor represents the a posteriori plausibility divided by the a priori plausibility and reports the changes in our beliefs introduced by data. This has the property of being almost objective and partially eliminates the influence of the a priori distribution.

As an example suppose the simple contrast:

$$H_0 : \theta = \theta_0 \quad vs \quad H_a : \theta = \theta_1.$$

We have that the a posteriori distributions are:

$$\alpha_0 = P(H_0|\mathbf{x}) = \frac{f_0 L(\theta_0|\mathbf{x})}{f_0 L(\theta_0|\mathbf{x}) + f_1 L(\theta_1|\mathbf{x})},$$

$$\alpha_1 = P(H_1|\mathbf{x}) = \frac{f_1 L(\theta_1|\mathbf{x})}{f_0 L(\theta_0|\mathbf{x}) + f_1 L(\theta_1|\mathbf{x})}.$$

So, the Bayes factor is:

$$B = \frac{\alpha_0 f_1}{\alpha_1 f_0} = \frac{f_0 L(\theta_0|\mathbf{x}) f_1}{f_1 L(\theta_1|\mathbf{x}) f_0} = \frac{L(\theta_0|\mathbf{x})}{L(\theta_1|\mathbf{x})},$$

which coincides with the likelihood ratio, so that the a priori distribution would not influence the Bayes factor, in this case.

Thus, the Bayes factor for the change point, when it is divided into two segments, is given by the likelihood ratio:

$$\frac{L_0}{L_1} = \frac{Q_1 Q_2}{L_1},$$

where Q_1 is the likelihood of segment 1 and Q_2 is the likelihood of segment 2, under the alternative hypothesis; and L_1 is the likelihood under the null hypothesis.

Applying the logarithm, we have

$$B = \log\{Q_1\} + \log\{Q_2\} - \log\{L_1\}.$$

2.1.1 Homogeneous Poisson Process

To find the change points, first is simulated a homogeneous Poisson process and after of obtained data is worked on. The homogeneous Poisson process is defined as follows:

Definition 1: A collection of random variables $\{N(t) : t \geq 0\}$, defined in a probability space (ω, F, P) is said to be a Poisson process (homogeneous) with intensity $\lambda > 0$ if it satisfies the following properties:

- i) $P(N(0) = 0) = 1$, this is, $N(0)$ it is always 0.
- ii) For all $0 < s < t$, $N(t) - N(s)$ has Poisson distribution of parameter $\lambda(t - s)$.
- iii) For all $0 \leq t_1 < \dots < t_n, n \geq 1$ (that is, to say for all finite set of times), the random variables $N(t_n) - N(t_{n-1}), \dots, N(t_2) - N(t_1), N(t_1) - N(0), N(0)$, are independent. This is known as the property of independent increments.

2.1.2 Bisection Method

The bisection method was used to determine the change points with the Bayes factor. This method consists of dividing the data in half and looking for a change point with the Bayes factor as well as with the likelihoods corresponding to the null and alternative hypotheses. Then we go to the left side of the data and it is also divided in half; subsequently, the right side is divided, and so on. We continue going successively to the left and to the right, dividing each side in half and looking for change points with the Bayes factor.

2.2 Threshold Method

The following is an example of change points determined by the threshold method, explained in Altieri (2015), which consists of associating a posteriori probability to the data and defining a threshold, in such a way that, if the probability is less than that threshold, it is said that there is a change point.

An example of how to associate probabilities is given below, using de following multiple linear regression:

$$Y_i = \beta_0 + \sum_{j=1}^4 \beta_j X_{j,i} + \epsilon_i.$$

For this example, we took the cement database in Gómez (2020). In the model, Y_i represents the evolved heat of the observation i and $X_{j,i}$ is the proportion of the component j in the observation i . The parameter β_0 is an intercept and β_j , $j = 1, \dots, 4$ are coefficients associated with the covariates. Finally, ϵ_i , $i = 1, 2, \dots, n$ is an error term with a Gaussian distribution having zero mean and precision τ .

2.2.1 Cement Data

The cement database used is shown in Table 2.

Table 2. Regression data

X_1	X_2	X_3	X_4	Y	X_1	X_2	X_3	X_4	Y
7	26	6	60	78.5	1	31	22	44	72.5
1	29	15	52	74.3	2	54	18	22	93.1
11	56	8	20	104.3	21	47	4	26	115.9
11	31	8	47	87.6	1	40	23	34	83.8
7	52	6	33	95.9	11	66	9	12	113.3
11	55	9	22	109.2	10	68	8	12	109.4
3	71	17	6	102.7					

2.3 Cumulative Sum Method

The Cumulative Sum method consists of the following:

- 1.- The average is calculated $\frac{x_1+x_2+\dots+x_{32}}{32}$.
2. The cumulative sum starts at zero $S_0 = 0$.
3. Other cumulative sums of the form $S_i = S_{i-1} + (X_i - \bar{X})$ are calculated for $i = 2, \dots, 32$

$$S_0 = 0,$$

$$S_1 = S_0 + (X_1 - \bar{X}),$$

$$S_2 = S_1 + (X_2 - \bar{X}),$$

⋮

$$S_{32} = S_{31} + (X_{32} - \bar{X}).$$

Bootstrapping is also performed, but an estimator of the magnitude of the change is required before that. An option that works well regardless of distribution and despite multiple changes is *Sdiff*, which is the maximum difference of S_i and the minimum of S_i , as can be seen below:

$$Sdiff = Smax - Smin \text{ where } Smax = \max_{i=0,\dots,33} S_i, Smin = \min_{i=0,\dots,33} S_i.$$

The magnitude of the change is *Sdiff*. A positive value of *Sdiff* indicates that there was a change from low to high, which means that traffic accidents changed. The latter is the topic of the problem addressed. Next, a bootstrap analysis of a single routine is performed. The procedure is the next:

1. A bootstrap sample of 32 units, denoted by $X_1^o, X_2^o, \dots, X_{32}^o$, was generated, randomly rearranging the 32 original values. This is called sampling without replacement.
2. Based on the bootstrap samples, the CUSUM bootstrap is calculated. This is denoted by $S_0^o, S_1^o, \dots, S_{32}^o$.
3. The maximum, the minimum and the CUSUM bootstrap difference, denoted by $S^o max, S^o min$, and $S^o diff$ are calculated.
4. It is determined if the bootstrap difference $S^o diff$ is less than the difference *Sdiff*.

A bootstrap analysis consists of performing a large number of bootstraps and counting the number of bootstraps for which

$S^{\circ}diff$ is less than $Sdiff$. Let N be the number of bootstrap samples performed and let X be the bootstrap for which $S^{\circ}diff < Sdiff$. Then the confidence of the change occurred is calculated as a percentage as follows:

$$\text{Confidence level} = 100 \frac{X}{N} \%$$

2.3.1 Traffic Accident Data

The cumulative sum method was applied to traffic accident data in 32 cities, which are the states of the Mexican Republic. The data, which correspond to the number of traffic accidents, were obtained from INEGI page and are presented in Table 3:

Table 3. Number of traffic accidents

City	Accidents	City	Accidents	City	Accidents
1	74	11	188	22	146
2	60	12	56	23	69
3	26	13	36	24	113
4	24	14	351	25	278
5	94	15	49	26	208
6	22	16	171	27	41
7	78	17	68	28	136
8	270	18	42	29	80
9	223	19	229	30	140
10	104	20	36	31	37
11	188	21	206	32	83

2.3.2 Coronavirus Data

Coronavirus data corresponding to May 2020, obtained from the *coronavirus.gob.mx* page, were used to detect the change points, applying the cumulative sum method. Three groups of data were formed, corresponding to eight, seven, and five days, respectively. In each database, it was detected where the change from minor to major occurred, that is, where a higher number of contagion cases occurred in the country. In the database they are divided into infections of men and women; however, for the analysis of changes, the total contagion cases (men plus women) were considered. Table 4 shows the data used.

Table 4. Coronavirus infection data

Date	Data	Date	Data	Date	Data
1	1225	9	1270	16	1483
2	1241	10	1161	17	1145
3	1121	11	2512	18	2747
4	2298	12	2428	19	2261
5	1840	13	2516	20	2083
6	2024	14	2353		
7	2085	15	2564		
8	2071				

3. Results and Discussion

3.1 Analysis of Multiple Change Points With the Bayes Factor

In order to detect, analyze, and compare points of change, a program was developed to detect 5 points of change. A set of 60 data with 6 different values for the parameter λ were used, which were obtained from a Poisson simulation process. This exercise is intended to detect the changes generated for the different values of λ . In addition, a comparison was made with the results obtained when applying an a priori uniform, a log-gamma, and a Gaussian.

A program was generated in R to detect multiple points of change in time. In total, for five change points, 60 data were used; the smallest segment was four points. A homogeneous Poisson process with different λ was simulated and the a posterior distribution was approached with the R INLA package, which performs the approximation through Taylor series. Bayes factor and the bisection method were used to detect the change points.

A uniform a priori, log gamma and a Gaussian were used. Five change points are shown in Figure 1.

For the following problem, 60 data are used and the smallest division was reached, which was four data. So, there are fifteen divisions as shown in Table 5.

The change points for a uniform a priori with $\lambda = 2, 1, 4, 7, 6, 1$ and in the data divisions or segments 8, 7, 15, 15, 8, 7, were as shown in Table 5, the change points are the first four and the seventh.

With the same change points and the a priori loggamma with parameters 0.01 and 0.01, the result is that shown in Table 5. It can be seen that compared to the uniform, the log-likelihoods of the loggamma are smaller, however they also detect the change points.

With the a priori Gaussian with zero mean and precision parameter 0.001, for the same number of change points as the uniform and the log gamma, the five change points were detected; these are the first four and the seventh, in Table 5. It can be seen that the value of the log-likelihood of the seventh change point is less than those of the following positions, however.

The five change points in Table 5 were detected with the Bayes factor B, mentioned above. The values for B indicate the strength of the evidence in favor of the change point hypothesis. Values less than 1 indicate that there is no change point. The data in bold, which correspond to the change points, were detected in the following way: value 1 in the middle of the data, value 2 in data 15, value 3 in data 45, value 4 in data 8, and value 7 in data 53. As can be seen, the values of the change points are not equal to each other or to the rest. On the other hand, where a change point did not occur, there were values equal to each other, for example, values 5 and 6, as well as 8, 10, and 14, and in the same way as 9, 11, 13, and 15; therefore, all these values indicate the non-existence of change points. The strength of the evidence in favor of the alternative hypothesis, indicated by applying the bisection method for gradual detection of the change points, is strong for the first point of change and decreases for the following ones; notice this in Table 5. The change points are shown in

Table 5. Change points

Number	uniform	loggamma	Gaussian
1	49.565011	43.644575	42.972975
2	24.392282	25.829232	26.592785
3	13.940830	26.041694	28.329433
4	16.000633	9.990938	11.641769
5	12.478724	2.387813	4.477951
6	12.478724	2.387813	4.477951
7	12.572475	4.498905	5.523642
8	5.701123	1.835601	5.801758
9	4.732525	1.384310	4.912561
10	5.701123	1.835601	5.801758
11	4.732525	1.384310	4.912561
12	5.701123	1.835601	5.801758
13	4.732525	1.384310	4.912561
14	5.701123	1.835601	5.801758
15	4.732525	1.384310	4.912561

Figure 1 and are at 8, 15, 30, 45, and 53 divisions.

3.1.1 Six Change Points With the Bayes Factor

With the same uniform model, six change points were detected for $\lambda = 2, 1, 4, 7, 6, 1, 2$, in the data divisions or segments of 8, 7, 15, 15, 8, 4, 3. The change points detected are the first four, the seventh, and the last. These are shown in Tables 6 and 7.

With the a priori logarithm of gamma, six change points were also detected, which are the first four, the seventh and the last in Table 6. The difference with the uniform is that the log likelihood is smaller, as can be seen.

For the a priori Gaussian, six change points were also detected. These are shown below in log likelihood values. The results are somewhat similar to those from the uniform. The change points correspond to the first four, the seventh, and the last in Table 6. The difference between Gaussian and uniform is that the last change point is less than the previous log likelihood values, where there is no change point.

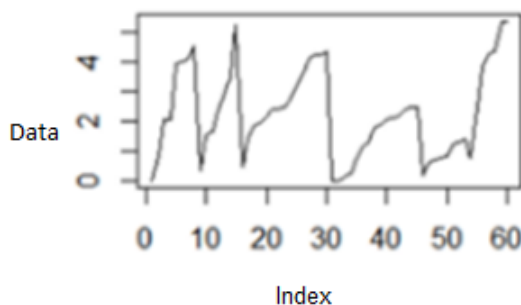


Figure 1. Five change points graph

With the beta logit (with parameters 2 and 2), six change points were well detected, being the same as for the uniform and the gamma logit: the first four, the seventh, and the last. No big differences are visualized with respect to the uniform.

For the a priori truncated Normal with mean 0 and precision 0.001, the likelihood is smaller compared to the uniform. This is very similar to the loggamma, since six change points were also detected: the first four, the seventh, and the last in Tables 6 and 7.

In a similar way to the previous problem, the observed change points are shown in tables 6 and 7. These values are not repeated among themselves or among the rest. Instead, the rest of the values are repeated, for example, 5 and 6, as well as 8, 10, 12 and 14, and also 9, 11, and 13; therefore, all these values do not indicate change points, even when they are greater than 1.

Table 6. Six change points

Num	uniform	loggamma	Gaussian
1	37.382628	44.717137	38.258157
2	23.107912	28.913661	26.506732
3	22.987741	27.851420	26.529322
4	16.466681	5.045601	11.798824
5	12.478724	2.387813	4.477951
6	12.478724	2.387813	4.477951
7	16.555598	3.062915	12.572375
8	5.701123	1.835601	5.801758
9	4.732525	1.384310	4.912561
10	5.701123	1.835601	5.801758
11	4.732525	1.384310	4.912561
12	5.701123	1.835601	5.801758
13	4.732525	1.384310	4.912561
14	5.701123	1.835601	5.801758
15	4.733385	1.720949	4.280097

The change points are shown in Figure 2. These are given in 8, 15, 30, 45, 53, and 57 divisions.

Programming was done in R. The program is presented in appendix A as Program A1.

An algorithm was created for the problem of detecting six change points with the Bayes factor. This algorithm is presented below:

Algorithm 1: it detects six change points.

Input the INLA function returns the variables **mp1**, **mp2**, and **mp** which contain data simulated with an AR1 autoregressive model and data from a Poisson family for **datos A**, **datos B** and **datos**.

Output the vector **respuesta** keeps the likelihoods of 15 divisions.

Initialize vectors are initialized **mp1**, **mp2**, and **mp** which contain the simulated data with an AR1 autoregressive model

Table 7. Six change points

Num	logit beta	normal truncated
1	49.664317	51.841263
2	30.567246	16.016904
3	18.174484	33.658036
4	9.944739	7.969972
5	4.633706	2.186421
6	4.633706	2.186421
7	11.248933	1.984581
8	4.076750	1.881876
9	3.794214	1.783849
10	4.076750	1.881876
11	3.794214	1.783849
12	4.076750	1.881876
13	3.794214	1.783849
14	4.076750	1.881876
15	4.405551	2.260447

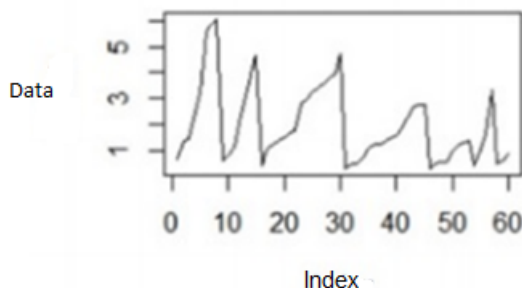


Figure 2. Six change points graph

and a Poisson family for **datos A**, **datos A**, and **datos**.

Return {respuesta} vector that keeps the verosimilitudes of the divisions.

The **datos** vector is divided into 1 to 30 and 31 to 60, and it is assigned to **datos A** and **datos B**.

- i) The likelihood for **datos A** and **datos B** is calculated and stored in the **respuesta** vector,
- ii) Vectors **datos A** and **datos B** are halved and the data are assigned to **datos C** and **datos D**,
- iii) The likelihood fo **datos C** and **datos D** is calculated.
- iv Steps i) and ii) are repeated for the new divisions of the data until the division of only two data.
- v) The results obtained are recorded.

3.1.2 Space-Time Analysis of Multiple Change Points With the Bayes Factor

For multiple change points in space-time, what was done was assigning positions. The first one was assigned in the center and the rest around it in ascending and counterclockwise order. The a posteriori was approximated with the R-INLA package. The a priori uniform, the Bayes factor, and the bisection method were also used. The simulation of values was done for a homogeneous Poisson process; four random values of λ (between 1 and 15) were used for each position, so three change points were estimated. The values of λ are given in each row of Table 8.

And the resulting change points are the first three of Tables 9, 10, 11, and 12. These are in 15, 30, and 45 divisions.

The behavior of the values is similar to that of the first problem presented in this work. The change points indicated in bold in tables 9, 10, 11, and 12 are not repeated, unlike the rest. According to the strength of the evidence for Bayes factor B, values less than 1 indicate that there is no point of change. These values are observed in table 10, position 6, value 3, and in table 11, position 9, value 2.

Table 8. Lambda values

	[1]	[2]	[3]	[4]
[1]	10	3	5	13
[2]	8	12	6	5
[3]	14	8	13	5
[4]	5	4	7	12
[5]	12	7	6	8
[6]	12	9	1	6
[7]	1	13	5	3
[8]	11	14	1	3
[9]	1	13	5	15
[10]	13	1	7	4
[11]	5	11	8	9
[12]	10	3	7	15
[13]	1	13	12	6
[14]	7	1	11	12
[15]	13	12	4	1
[16]	5	6	12	7

Table 9. Change points space-time

	[1]	[2]	[3]	[4]
1	33.693643	44.955163	35.315124	41.732270
2	27.795301	25.226235	28.387088	19.482977
3	28.727677	21.122882	27.775696	28.697886
4	12.478724	12.478724	12.478724	12.478724
5	12.478724	12.478724	12.478724	12.478724
6	12.478724	12.478724	12.478724	12.478724
7	12.478724	12.478724	12.478724	12.478724
8	5.701123	5.701123	5.701123	5.701123
9	4.732525	4.732525	4.732525	4.732525
10	5.701123	5.701123	5.701123	5.701123
11	4.732525	4.732525	4.732525	4.732525
12	5.701123	5.701123	5.701123	5.701123
13	4.732525	4.732525	4.732525	4.732525
14	5.701123	5.701123	5.701123	5.701123
15	4.732525	4.732525	4.732525	4.732525

The Bayes factor method detects the change points well. Its weakness is that it only detects changes up to a division of four, but not when the amount of data is lower.

Programming was performed in R. It is shown in appendix A as program A2.

An algorithm was built to detect 3 space-time points of change in 16 positions. This algorithm is shown below:

Algorithm 2: it detects 3 space-time points of change

Input the INLA function returns the variables **mp1**, **mp2**, and **mp** which contain data simulated with an AR1 autoregressive model and data from a Poisson family for **datos A**, **datos B** and **datos**.

Output the **matrizrespuesta** keeps the likelihoods of 5 divisions in 16 positions.

Initialize vector **mp1**, **mp2**, and **mp** are initialized. These contain data simulated with an AR1 autoregressive model and data from a Poisson family for **datos A**, **datos B** and **datos**

n=16 means 16 positions.

for i in 1:n

Table 10. Change points space-time

	[5]	[6]	[7]	[8]
1	33.574630	61.47789451	52.546434	44.358894
2	27.549217	27.37020690	18.324691	21.142981
3	28.777261	-0.08807506	20.678559	1.851282
4	12.478724	12.47872437	12.478724	12.478724
5	12.478724	12.47872437	12.478724	12.478724
6	12.478724	12.47872437	12.478724	12.478724
7	12.478724	12.47872437	12.478724	12.478724
8	5.701123	5.70112304	5.701123	5.701123
9	4.732525	4.73252533	4.732525	4.732525
10	5.701123	5.70112304	5.701123	5.701123
11	4.732525	4.73252533	4.732525	4.732525
12	5.701123	5.70112304	5.701123	5.701123
13	4.732525	4.73252533	4.732525	4.732525
14	5.701123	5.70112304	5.701123	5.701123
15	4.732525	4.73252533	4.732525	4.732525

Table 11. Change points space-time

	[9]	[10]	[11]	[12]
1	71.06948660	35.332035	34.663739	28.888859
2	-0.08807506	28.183967	26.321000	25.185406
3	19.83004426	27.173106	28.558468	28.850767
4	12.47872437	12.478724	12.478724	12.478724
5	12.47872437	12.478724	12.478724	12.478724
6	12.47872437	12.478724	12.478724	12.478724
7	12.47872437	12.478724	12.478724	12.478724
8	5.70112304	5.701123	5.701123	5.701123
9	4.73252533	4.732525	4.732525	4.732525
10	5.70112304	5.701123	5.701123	5.701123
11	4.73252533	4.732525	4.732525	4.732525
12	5.70112304	5.701123	5.701123	5.701123
13	4.73252533	4.732525	4.732525	4.732525
14	5.70112304	5.701123	5.701123	5.701123
15	4.73252533	4.732525	4.732525	4.732525

4 **lambdas** from a 1-15 Poisson process are randomly calculated and they are included in the vector **vlambda**.

The four vectors simulated by change points, corresponding to the four lambdas, are included in the vector **datos** which has space for 60 data

end for

for i in 1:n

The vector **datos** is divided into 1 to 30 and 31 to 60 and assigned to **datos A** and **datos B**.

i) The likelihood for each vector in **datos A** and **datos B** is calculated and stored in the **matrizrespuesta**,

ii) The vector **datos A** and **datos B** are halved and data are assigned to **datos C** and **datos D**,

iii) The likelihoods for **datos C** and **datos D** are calculated.

iv) Steps **i)** and **ii)** are repeated for the new divisions of the data until the division of only two data.

v) The results obtained are recorded.

end for

return returns the **matrizrespuesta**. This matrix contains the likelihoods.

Table 12. Change points space-time

	[13]	[14]	[15]	[16]
1	55.319548	36.493439	53.350957	37.905475
2	7.263903	27.399136	14.610914	25.226621
3	28.683089	26.696299	17.792607	28.209964
4	12.478724	12.478724	12.478724	12.478724
5	12.478724	12.478724	12.478724	12.478724
6	12.478724	12.478724	12.478724	12.478724
7	12.478724	12.478724	12.478724	12.478724
8	5.701123	5.701123	5.701123	5.701123
9	4.732525	4.732525	4.732525	4.732525
10	5.701123	5.701123	5.701123	5.701123
11	4.732525	4.732525	4.732525	4.732525
12	5.701123	5.701123	5.701123	5.701123
13	4.732525	4.732525	4.732525	4.732525
14	5.701123	5.701123	5.701123	5.701123
15	4.732525	4.732525	4.732525	4.732525

3.2 Threshold Method

For the example of the change point, the variable X_1 in Table 2 of Section 2.2.1 was used. Probabilities of the a posteriori distribution were assigned to data of X_1 . A threshold of $3.041631e - 07$ was set, so data having a probability less than this value were considered as change points.

The program A3 in Appendix A was developed in R. The results were given in zeros and ones:

0 0 1 1 0 1 0 0 0 1 0 1 1

The change occurs where the number 1 begins to be more constant, that is, where the a posteriori probability assigned to the data is less than the defined threshold, indicating that a point of change exists. The latter happens in position ten and three, with a threshold of $3.041631e-07$.

An algorithm was created which detects multiple points of change by the threshold method. This algorithm is presented below:

Algorithm 3: it detects multiple change points by the threshold method

Input cement matrix, datos matrix, treshold, regression in ml.

Output Vector **menores** include zeros and ones; 1 means that the a posteriori probability is less than threshold y 0 means that the a posteriori probability is greater than threshold.

Inizialite n=13 is the amount of data.

For h in 1:n

The a posteriori probability is calculated for each data.

End For

For h in 1:n

if the a posteriori probability is less than threshold, then a 1 is stored in the vector **menores** **else** if it is greater a zero is stored.

End For

Print menores

3.3 Cumulative Sum Method

3.3.1 With Traffic Accident Data

The algorithm of Section 2.3 was applied to the traffic accident data of Section 2.3.1, Table 3. According to the magnitude of the change, the minimum sum was -439.6875 , and it was found in the data located in the seventh position. The maximum sum, the minimum sum and the difference were:

Smax=183.875
 Smin=-439.6875
 Sdif=623.5625

Confidence Level for a hundred sample bootstrap= $100 \frac{X}{N} \% = 83\%$.

It can be seen in Figure 3 that the change point is at value 7, where it changes from the smallest number of traffic accidents to a higher number.

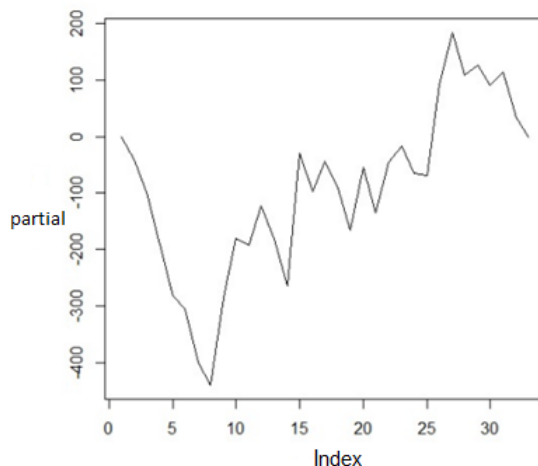


Figure 3. Graph with change point at 7

What the algorithm establishes was applied and the program A4 in appendix A was developed.

An algorithm was created with the cumulative sum method, which detects a single point of change. This algorithm is shown below:

Algorithm 4: it detects a single change point

Input the matrix **datos** contain data for the analysis

Output **parciales** are cumulative data sums, **rparciales** are cumulative bootstrap sums, **rSdif** are the resampling differences.

Inizialite n=33 they are 33 data

m=100 is resampling of size 100 with 33 data

For i in 2:n

The cumulative sum is calculated and stored in **parciales**

End For

The maximum **Smax=max(parciales)**, ,minimum **Smin=min(parciales)** and difference **Sdif=Smax-Smin** are calculated

For k in 2:n

The resampling cumulative sum is calculated and stored in **rparciales**

End For

It is calculated the maximum of **rparciales rSmax=max(rparciales)**

mimimum **rSmin=min(rparciales)**, and difference **rSdif=rSmax-rSmin**

return rSdif returns the resampling difference vector.

For j in 1:m a bootstrap resampling of size 100 with 33 data is performed

The differences are calculated for each resampling **diferencias[m] = remuestreo(sample(datos, 32))**

End for

For j in 1:m

If the resampling differences are less than **Sdif** then the normal differences are stored in the vector **respuesta**.

End for

The confidence Interval is calculated:

$$Nconfianza=100*\text{sum}(\text{respuestas})/\text{length}(\text{respuestas})$$

Other tests applied with traffic accident data

The Buishand, Pettitt and standard normal homogeneity tests were performed on the same traffic accident data. For all tests, the result was the change point at position 7. It was used in R the trend library in Pohlert (2020), using the following code:

```
library(trend)
data <- c(74, 60, 26, 24, 94, 22, 78, 270, 223, 104, 188, 56, 36,
, 351, 49, 171, 68, 42, 229, 36, 206, 146, 69,
113, 278, 208, 41, 136, 80, 140, 37, 83)
y <- -ts(data, start = c(1900), freq = 1)
x <- -ts(data, start = c(1900), freq = 1)
buishand_ranges test
br.test(y, m = 20000)
pettit test
pettitt.test(x)
standard normal homogeneity test
snh.test(x, m = 20000)
```

3.3.2 With Coronavirus Data

The coronavirus data from Section 2.3.2 in Table 4 were used. The infection data are shown in a bar graph, in Figure 4. The bar graph shows how the number of infections decreases and increases again. Increases or changes occurred at May



Figure 4. Graph of contagion

4, May 11 an May 18, for the first, second and last group, respectively. This is observed in the CUSUM cumulative sums graph shown in Figure 5. For each cumulative sum, the statistics and the percentage are those shown in Table 13.

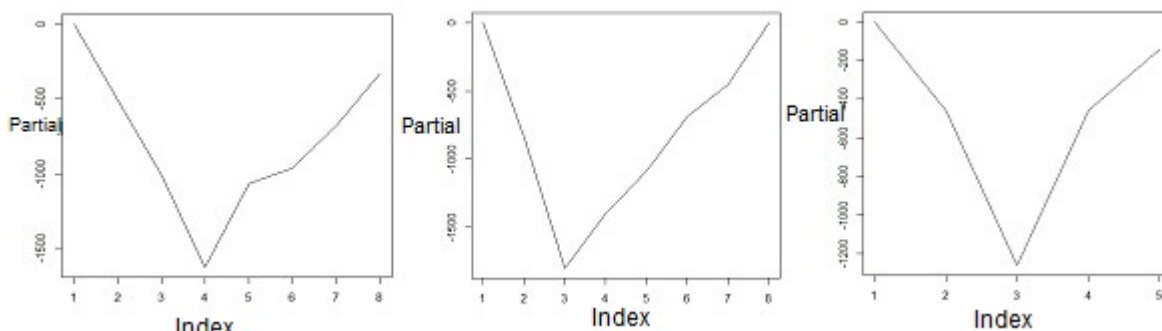


Figure 5. Data groups 1, 2 and 3

The R program applied for the analysis of coronavirus data, was the same as that used for the traffic accidents data.

With the CUSUM method only one change point is detected, however changes are detected even when the database is small, which is the case with the coronavirus data.

Table 13. Statistics and percentage

Group 1	Group 2	Group 3
Sdiff=1627.375	Sdiff=1798.714	Sdiff=1259.6
Smax=0	Smax=0	Smax=0
Smin=-1627.375	Smin=-1798.714	Smin=-1259.6
ConfidenceL=89%	83%	61%

4. Conclusion

Programs were developed for the Bayes factor, thresholds, and CUSUM methods. In the case of the Bayes factor method, the purpose was to numerically detect the points of change and compare the results with those of the a priori distributions used. For the threshold and CUSUM methods, the objective was to detect the change points. Coronavirus, car traffic, and regression (cement) databases were used, as well as a database simulated with a Poisson process.

Pettitt, standard normal homogeneity, and Buishand ranges methods were used. The R Trend package, which contains these methods, was applied to detect the change points.

It can be concluded that the threshold and CUSUM methods detect change points well. The difference is that the threshold method allows to detect multiple points of change in a regression.

The CUSUM method only detects a point of change, as do Pettitt, standard normal homogeneity, and Buishand ranges methods. Using the CUSUM method, changes are detected even when the database is small, as happened with the coronavirus data.

The Bayes factor method detected the change points well. Its weakness is that it can only detect changes up to a division of four, but when the amount of data in the division is lower, the change points are no longer detected.

References

Altieri, L. (2015). A bayesian changepoint analysis on spatio-temporal point processes (Doctoral dissertation). University of Bologna. Italy. <https://doi.org/10.1016/j.spasta.2015.05.005>

Blangiardo, M., & Cameletti, M. (2015). *Spatial and Spatio-temporal Bayesian Models with R-INLA*. John Wiley & Sons.

Chen, J., & Gupta, A. K. (2012). *Parametric Statistical Changepoint Analysis: With Applications to Genetics, Medicine, and Finance* (2nd ed.). Birkhauser. <https://doi.org/10.1007/978-0-8176-4801-5>

Gómez, R. V. (2020). *Bayesian Inference with INLA* (1st ed.). Chapman & Hall.

Pohlert, T. (2020). *Non-parametric trend tests and change-point detection*. R Package version 3.0.

Taylor, W. A. (2000). *Change-point analysis: A powerful new tool for detecting changes*. Baxter Healthcare Corporation, Round Lake, IL.

Appendix A

Program A1

```
rm(list = ls())
library(poisson)
library(INLA)
vector <- hpp.event.times(2, 15, num.sims = 1, t0 = 0)
plot(vector)
vector1 <- hpp.event.times(1, 15, num.sims = 1, t0 = 0)
vector2 <- hpp.event.times(4, 15, num.sims = 1, t0 = 0)
vector4 <- hpp.event.times(6, 15, num.sims = 1, t0 = 0)
datos <- c(vector, vector1, vector2, vector4)
plot(datos, type = "l")
datosA = datos[1 : 30]
datosB = datos[31 : 60]
datos
datosA
datosB
```

```
vec_res = rep(0, 20)
factor_b <- function(datosA, datosB, datos){
  p1 <- data.frame("num" = seq(1, length(datosA), 1), "datosA" = datosA)
  p2 <- data.frame("num" = seq(1, length(datosB), 1), "datosB" = datosB)
  mp1 <- inla(num f(datosA, model = "ar1"), data = p1, family = "poisson")
  mp2 <- inla(num f(datosB, model = "ar1"), data = p2, family = "poisson")
  midf <- data.frame("num" = seq(1, length(datos), 1), "datos" = datos)
  mp <- inla(num f(datos, model = "ar1"), data = midf, family = "poisson")
  respuesta <- as.numeric(mp1$mlik[1, 1]) + as.numeric(mp2$mlik[1, 1])
  -as.numeric(mp$mlik[1, 1])
  return(respuesta)
}
vec_res[1] = factor_b(datosA, datosB, datos)
vec_res
datosC = datosA[1 : 15]
datosD = datosA[16 : 30]
vec_res[2] = factor_b(datosC, datosD, datosA)
vec_res
datosE = datosB[1 : 15]
datosF = datosB[16 : 30]
vec_res[3] = factor_b(datosE, datosF, datosB)
vec_res
datosG = datosC[1 : 8]
datosH = datosC[9 : 15]
vec_res[4] = factor_b(datosG, datosH, datosC)
vec_res
datosI = datosD[1 : 8]
datosJ = datosD[9 : 15]
vec_res[5] = factor_b(datosI, datosJ, datosD)
vec_res
datosK = datosE[1 : 8]
datosL = datosE[9 : 15]
vec_res[6] = factor_b(datosK, datosL, datosE)
vec_res
datosM = datosF[1 : 8]
datosN = datosF[9 : 15]
vec_res[7] = factor_b(datosM, datosN, datosF)
vec_res
datosO = datosG[1 : 4]
datosP = datosG[5 : 8]
vec_res[8] = factor_b(datosO, datosP, datosG)
vec_res
datosQ = datosH[1 : 4]
datosR = datosH[5 : 7]
vec_res[9] = factor_b(datosQ, datosR, datosH)
vec_res
datosS = datosI[1 : 4]
datosT = datosI[5 : 8]
vec_res[10] = factor_b(datosS, datosT, datosI)
vec_res
datosU = datosJ[1 : 4]
datosV = datosJ[5 : 7]
vec_res[11] = factor_b(datosU, datosV, datosJ)
vec_res
datosX = datosK[1 : 4]
datosY = datosK[5 : 8]
vec_res[12] = factor_b(datosX, datosY, datosK)
```

```

vec_res
datosW = datosL[1 : 4]
datosZ = datosL[5 : 7]
vec_res[13] = factor_b(datosW, datosZ, datosL)
vec_res
datosAA = datosM[1 : 4]
datosBB = datosM[5 : 8]
vec_res[14] = factor_b(datosAA, datosBB, datosM)
vec_res
datosCC = datosN[1 : 4]
datosDD = datosN[5 : 7]
vec_res[15] = factor_b(datosCC, datosDD, datosN)
vec_res

```

To use another a priori function, it is changed in the program by the following instruction, where the a priori distribution to use and its parameters are included, since the INLA package has the a priori distributions to use.

```

prec.prior <- list(prec = list(prior = "logtgaussian", param = c(0, 0.001)))
mp1 <- inla(num f(datosA, model = "ar1", hyper = prec.prior), data = p1, family = "poisson")

```

Program A2

```

rm(list = ls())
library(poisson)
library(INLA)
esp_t = matrix(0, 60, 16)
matriz_resp = matrix(0, 15, 16)
mlambdas = matrix(0, 16, 4)
factor_b <- function(datosA, datosB, datos){
p1 <- data.frame("num" = seq(1, length(datosA), 1), "datosA" = datosA)
p2 <- data.frame("num" = seq(1, length(datosB), 1), "datosB" = datosB)
mp1 <- inla(num f(datosA, model = "ar1"), data = p1, family = "poisson")
mp2 <- inla(num f(datosB, model = "ar1"), data = p2, family = "poisson")
midf <- data.frame("num" = seq(1, length(datos), 1), "datos" = datos)
mp <- inla(num f(datos, model = "ar1"), data = midf, family = "poisson")
respuesta <- as.numeric(mp1$mlik[1, 1]) + as.numeric(mp2$mlik[1, 1])
-as.numeric(mp$mlik[1, 1])
return(respuesta)
}
for(i in 1 : 16){
vlambda = sample(1 : 15, 4)
vector <- hpp.event.times(vlambda[4], 15, num.sims = 1, r0 = 0)
plot(vector)
mlambdas[i, ] = vlambda
vector1 <- hpp.event.times(vlambda[1], 15, num.sims = 1, r0 = 0)
vector2 <- hpp.event.times(vlambda[2], 15, num.sims = 1, r0 = 0)
vector4 <- hpp.event.times(vlambda[3], 15, num.sims = 1, r0 = 0)
datos <- c(vector, vector1, vector2, vector4)
esp_t[, i] = datos
}
for(indice in 1 : 16){
datosA = esp_t[1 : 30, indice]
datosB = esp_t[31 : 60, indice]
datos = esp_t[, indice]
datosA
datosB
matriz_resp[1, indice] = factor_b(datosA, datosB, datos)
datosC = datosA[1 : 15]
datosD = datosA[16 : 30]

```



```

matriz_resp[2, indice] = factor_b(datosC, datosD, datosA)
datosE = datosB[1 : 15]
datosF = datosB[16 : 30]
matriz_resp[3, indice] = factor_b(datosE, datosF, datosB)
datosG = datosC[1 : 8]
datosH = datosC[9 : 15]
matriz_resp[4, indice] = factor_b(datosG, datosH, datosC)
datosI = datosD[1 : 8]
datosJ = datosD[9 : 15]
matriz_resp[5, indice] = factor_b(datosI, datosJ, datosD)
datosK = datosE[1 : 8]
datosL = datosE[9 : 15]
matriz_resp[6, indice] = factor_b(datosK, datosL, datosE)
datosM = datosF[1 : 8]
datosN = datosF[9 : 15]
matriz_resp[7, indice] = factor_b(datosM, datosN, datosF)
datosO = datosG[1 : 4]
datosP = datosG[5 : 8]
matriz_resp[8, indice] = factor_b(datosO, datosP, datosG)
datosQ = datosH[1 : 4]
datosR = datosH[5 : 7]
matriz_resp[9, indice] = factor_b(datosQ, datosR, datosH)
datosS = datosI[1 : 4]
datosT = datosI[5 : 8]
matriz_resp[10, indice] = factor_b(datosS, datosT, datosI)
datosU = datosJ[1 : 4]
datosV = datosJ[5 : 7]
matriz_resp[11, indice] = factor_b(datosU, datosV, datosJ)
datosX = datosK[1 : 4]
datosY = datosK[5 : 8]
matriz_resp[12, indice] = factor_b(datosX, datosY, datosK)
datosW = datosL[1 : 4]
datosZ = datosL[5 : 7]
matriz_resp[13, indice] = factor_b(datosW, datosZ, datosL)
datosAA = datosM[1 : 4]
datosBB = datosM[5 : 8]
matriz_resp[14, indice] = factor_b(datosAA, datosBB, datosM)
datosCC = datosN[1 : 4]
datosDD = datosN[5 : 7]
matriz_resp[15, indice] = factor_b(datosCC, datosDD, datosN)
indice = indice + 1
}
matriz_resp
mlambdas

```

Program A3

```

library(INLA)
cement
m1 <- inla(y ~ x1 + x2 + x3 + x4, data = cement)
dato <- c(7, 1, 11, 11, 7, 11, 3, 1, 2, 21, 1, 11, 10)
m1$marginals.fixed$x1
1 - inla.pmarginal(0, m1$marginals.fixed$x1)
probas = seq(1, 13)
for(h in 1 : 13){
  probas[h] = inla.pmarginal(dato[h]+0.2, m1$marginals.fixed$x1) - inla.pmarginal(dato[h]-0.2, m1$marginals.fixed$x1)
}
menores = seq(1, 13)

```

```

umbral = 3.041631e - 07
for(h in 1 : 13){
menores[h] = probas[h] < umbral
}
menores

```

Program A4

```

datos <- c(74, 60, 26, 24, 94, 22, 78, 270, 223, 104, 188, 56, 36, 351, 49, 171, 68, 42, 229, 36, 206,
146, 69, 113, 278, 208, 41, 136, 80, 140, 37, 83)
promd <- -mean(datos)
promd
parciales <- -seq(1, 33)
parciales[1] = 0
for(iin2 : 32){
parciales[i] = parciales[i - 1] + datos[i - 1] - promd
}
parciales
Smax = max(parciales)
Smin = min(parciales)
Sdif = Smax - Smin
Sdif
Smax
Smin
plot(parciales, type = "T")
remuestreo <- -function(rdatos){
rpromd <- -mean(rdatos)
rparciales <- -seq(1, 33)
rparciales[1] = 0
for(k in 2 : 33){
rparciales[k] = rparciales[k - 1] + rdatos[k - 1] - rpromd
}
rSmax = max(rparciales)
rSmin = min(rparciales)
rSdif = rSmax - rSmin
return(rSdif)
}
diferencias = seq(1, 100)
for(m in 1 : 100){
diferencias[m] = remuestreo(sample(datos, 32))
}
diferencias
respuestas <- -seq(1, 100)
for(m in 1 : 100){
respuestas[m] = diferencias[m] < Sdif
}
respuestas
diferencias
Sdif
NConfianza = 100 * sum(respuestas)/length(respuestas)
NConfianza

```

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).