Change point detection in mobile advertising

Nina Golyandina, Kliment Merzlyakov

Saint Petersburg State University
Mathematical faculty
Applied statistics department

Saint Petersburg
2018

Content

- Change point detection
  - What is change point detection?
  - Real world examples of change point detection
  - Reasons to detect change points
- Change point detection techniques
- Airpush cases
  - Fraud elimination
  - Trend extraction
  - Smart alerts

Change point detection

What is change point detection?

- Change point — point in time series where some significant change occurred
- Change point detection — group of methods to find change points in time series
- Change points can be two types:
  - Local — one time change
  - Global — change of time series structure

What is change point detection?

Let's formalize change point detection problem:

- Suppose $\mathrm{X} = (x_1, ..., x_N)$ — time series of length N
- $\mathbf{t} = \{t_1, t_2, ...\} \subset \{1, ..., K\}$ — set of change points indexes
- $K - 1$ — amount of changes in time series, which can be either known or unknown
- Change point detection problem — is a problem of searching such set of indexes, which would be the closest to real set of change point indexes: $\widehat{\mathbf{t}} = \{\widehat{t_1}, \widehat{t_2}, ...\}$
- To estimate change point detection algorithm quality we need tagged time series with known indexes $\widehat{\mathbf{t}} = \{\widehat{t_1}, \widehat{t_2}, ...\}$
- Quality can be estimated as sum of distance between real indexes and estimated indexes normalized by length of time series:

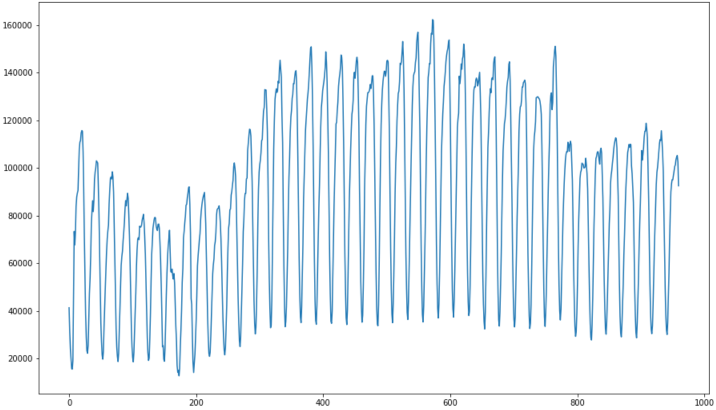
$$\sum_{i=1}^{K} \frac{|\widehat{t_i} - t_i|}{N}$$

Types of change points

- Trend change
- Mean change
- Variance change
- Single point change
- Period change
- Structure change

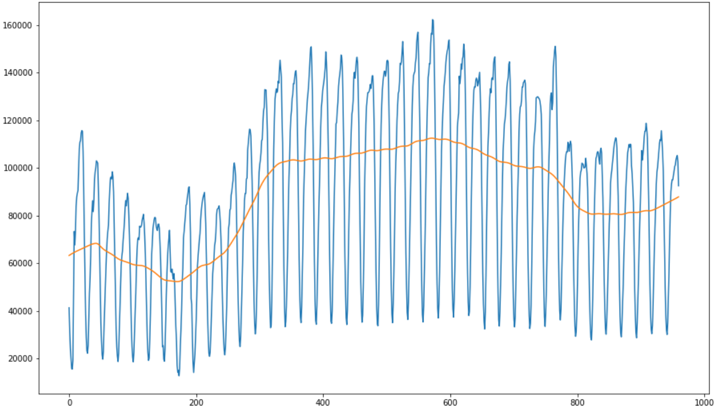
Change point examples

Common graph



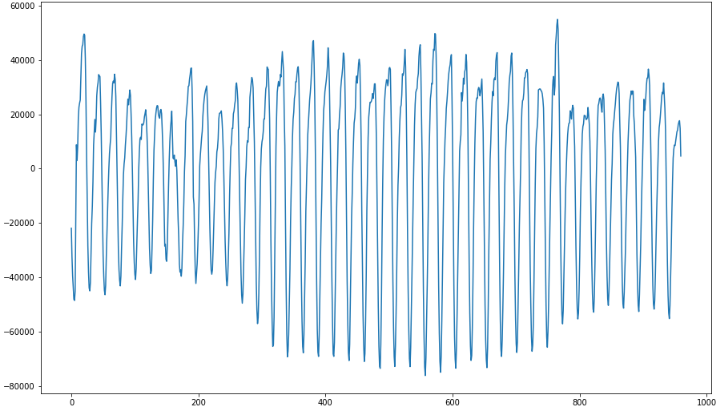
Change point examples

Common graph with trend



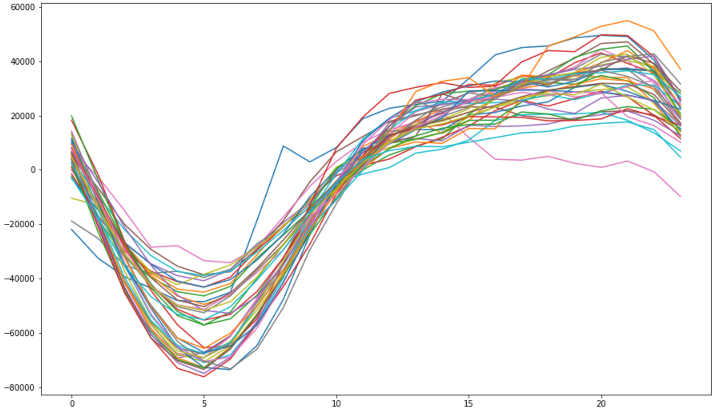
Change point examples

Common graph without trend



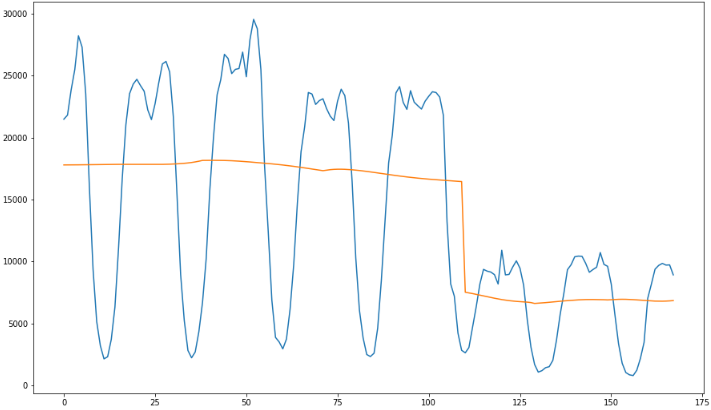
Change point examples

Common graph periodic frequency



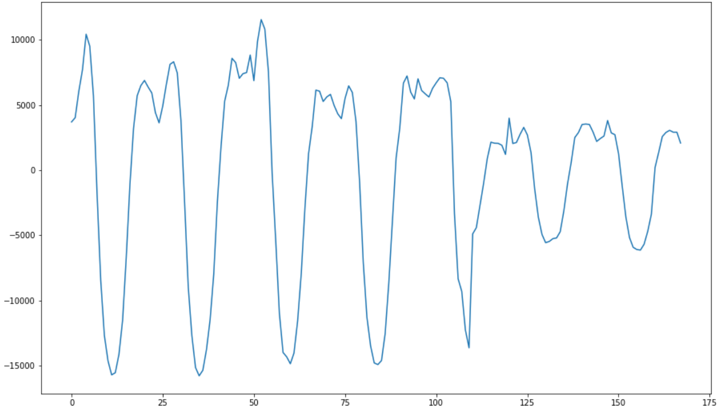
Change point examples

Mean change



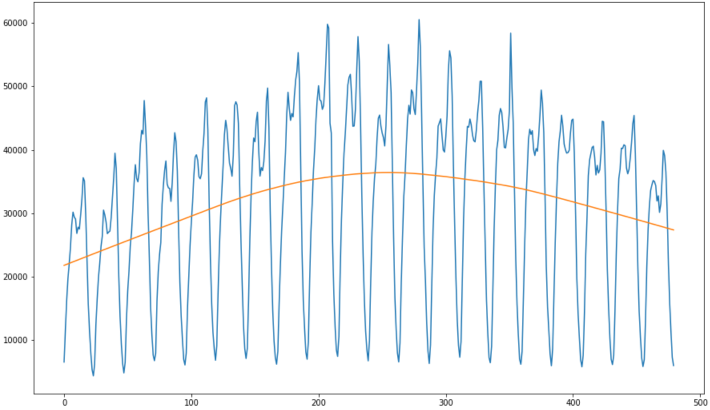
Change point examples

Variance change



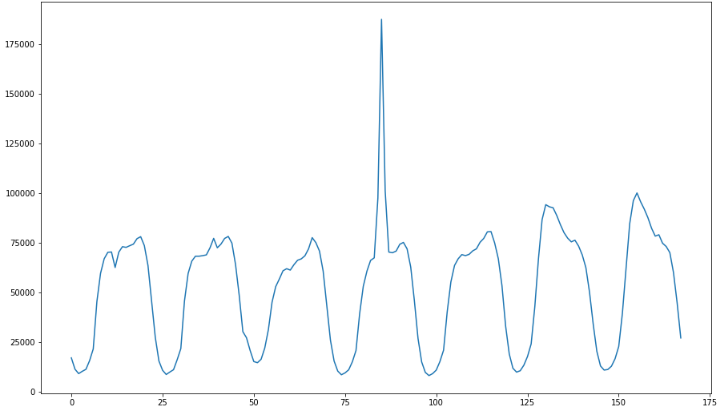
Change point examples

Trend change



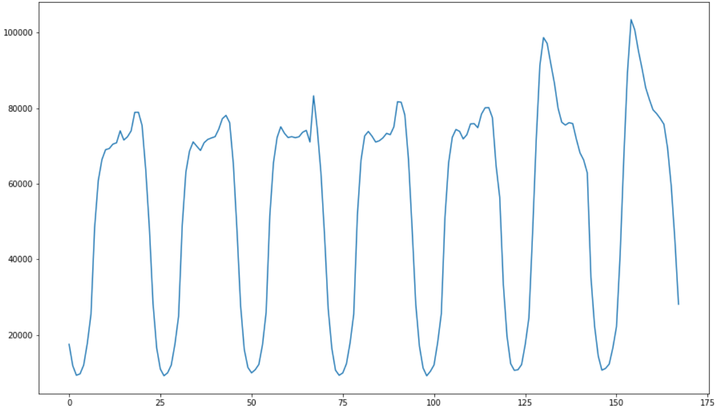
Change point examples

Point change



Change point examples

Structure change



Reasons to detect change points

- Searching issues in historical data
- Reacting on changes quickly
- Extracting trend more accurately

Change point detection techniques

SSA for change point detection

Singular spectrum analysis (SSA) method can be used to allocate change points. Let $x_1, x_2, \ldots$ be a time series and N, L, l, p, q be fixed integers so that $0 \le l < L \le N/2$ and $0 \le p < q$. The SSA change point detection algorithm is as follows. For each $n = 0, 1, \ldots$ we compute:

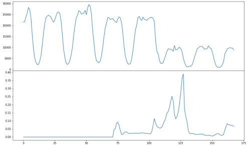
- the base matrix $\mathbb{X}^{(n)}$,
- the lag-covariance matrix $\mathbb{R}_n = \mathbb{X}^{(n)}(\mathbb{X}^{(n)})^\top$,
- the SVD of $\mathbb{R}_n$,
- $\nu_{n.I,p,q}$, the sum of the squared Euclidean distances between the vectors $\mathbb{X}_j^{(n)}(j = p+1, \ldots, q$ and the l-dimensional subspace $L_{n,I}$, and
- S_n, the normalized squared distance.

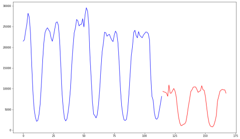
Large values of S_n indicate a change in the time series.
This method has the following parameters:

- N — window size for base matrix
- L, I — SSA window and number of eigenvectors
- p, q — window size for test matrix
- threshold

SSA in action. Mean change

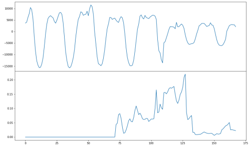
Let's try to apply SSA algorithm to real time series mentioned above. With manually settled parameters.

Mean change  **Detection** N = 48, L = 24, l = 5, p = 49, q = L, threshold = 0.15

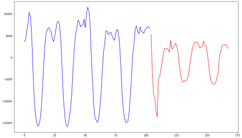


Works good for mean change

SSA in action. Variance change

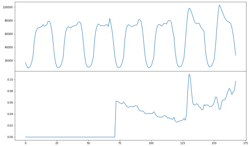


N = 48, L = 24, I = 5, p = 49, q = L, threshold = 0.15

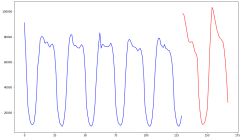


Works good for variance change

SSA in action. Period change

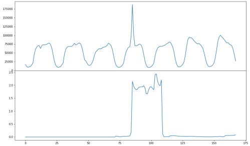


$N = 48$, $L = 24$, $l = 5$, $p = 49$, $q = L$, threshold $= 0.1$

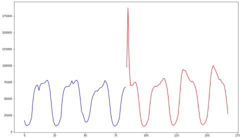


Works well for period change

SSA in action. Point change



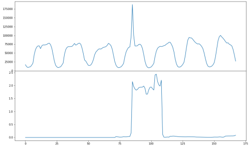
N = 48, L = 24, l = 5, p = 49, q = L, threshold = 0.1



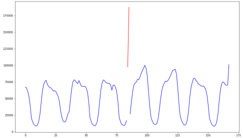
Doesn't work for point change. The reason is that we always move window from left to the right.

SSA in action. Point change

Let's try to reverse the time series each time we find change point.



$N = 48$, $L = 24$, $l = 5$, $p = 49$, $q = L$, threshold $= 0.1$

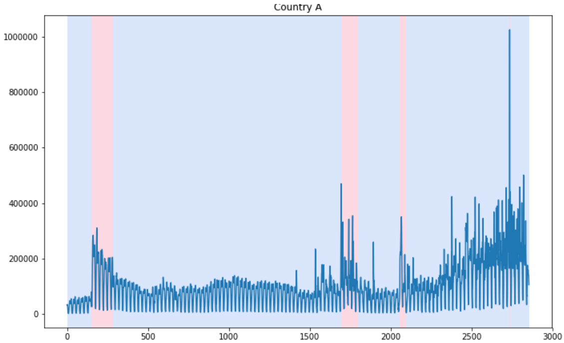


Using this upgrade we make SSA algorithm working for point change as well

Long time series

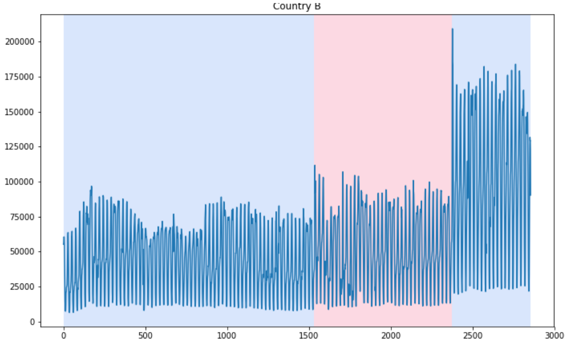
- We have 5 time series 2856 points each. It is impressions time series for different countries hourly for 17 weeks.
- We have tagged change points manually in all these time series

Country A



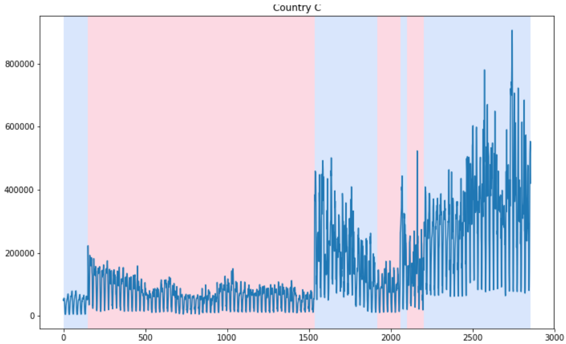
In time series for Country A we manually detected 6 change points. First one - mean change along with variance change. Second one - variance change. Third one - point change. Fourth one - variance change. Fifth and sixth - points change.

Country B



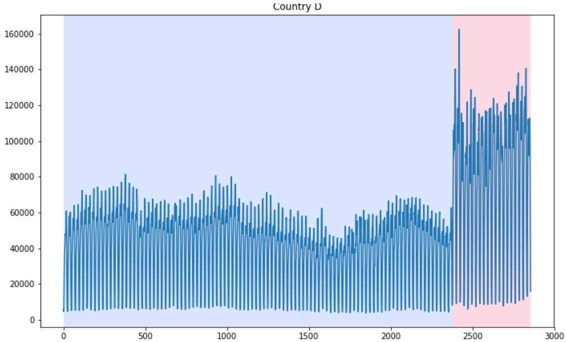
In time series for Country B we manually detected 2 change points. First one - mean change along with variance change. Second one - mean change along with variance change.

Country C



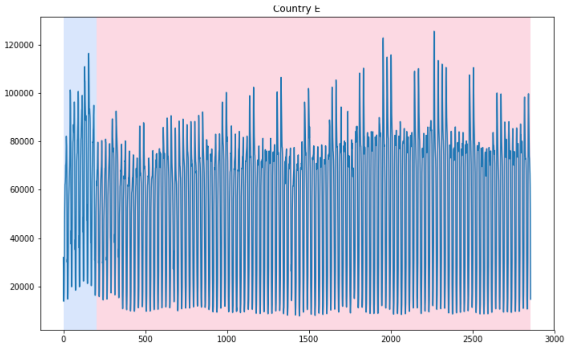
In time series for Country C we manually detected 5 change points. First, second and fifth - mean change along with variance change (similar to country A). Third - variance change. Fourth - point change

Country D



In time series for Country D only one change point was manually detected - mean change along with variance change.

Country E



In time series for Country E there is only one change point - mean change.

Another change point detection approach

- Apart from SSA there is another approach to find change points
- We can define change point detection problem as minimizing contrast function:

$$\mathbb{V}(\mathbf{t}, \mathrm{X}) := \sum_{i=1}^{K} c(x_{t_i} \ldots x_{t+1})$$

, where $c()$ — some cost function, and $x_{t_i} \ldots x_{t+1}$ — sub series of initial time series

- When quantity of change points K is known the problem is:

$$\min_{t} \mathbb{V}(\mathbf{t}, \mathrm{X})$$

- When quantity of change points unknown, then additional parameter penalty is added:

$$\min_{t} \mathbb{V}(\mathbf{t}, \mathrm{X}) + pen(\mathbf{t}, \mathrm{X})$$

Thus, such approach can be described as follows steps:

- Choosing cost function
- Minimizing contrast function
- Searching for change points quantity (when unknown beforehand)

Cost functions

There could be plenty of cost functions. Let's describe the most popular

- Mean absolute deviation. It finds changes in medians.

$$c(\mathbf{X}_I) = \sum_{t \in I} ||\mathbf{X}_t - \bar{\mathbf{X}}||_1$$

- Mean squared deviation. It finds changes in means.

$$c(\mathbf{X}_I) = \sum_{t \in I} ||\mathbf{X}_t - \bar{\mathbf{X}}||_2^2$$

- Kernel based method

$$c(\mathbf{X}_I) = \sum_{t \in I} ||\Phi(\mathbf{X}_t) - \bar{\mu}||_{\mathcal{H}}^2$$

- Autoregression

$$c(\mathbf{X}_I) = \min_{\delta \in \mathbb{R}_p} \sum ||\mathbf{X}_t - \delta' z_t||_2^2$$

Cost functions. Continue

We can also create our own cost function. Fore example, according to understanding of our time series we can assume that it can be described with such model:

$$\mathrm{X} = a + C\cos(2\pi\frac{n}{24}) + S\sin(2\pi\frac{n}{24}) + \epsilon$$

Then cost function can be:

$$c(\mathbf{X}_I) = \sum_{t \in I} ||\mathbf{X}_t - (a + C\cos(2\pi\frac{n}{24}) + S\sin(2\pi\frac{n}{24}))||_2^2$$

Search method

Apart from cost function we need to choose method of search change points. There are several well researched methods, we will use one of them — window method. Idea is straightforward:

- Choosing wide of the window w
- Then algorithm slides across the time series with this window
- On each iteration it splits time series into two sub series
- Then for each sub series special metric is calculated:

$$d(x_{u..v}, x_{v..w}) = c(x_{u..w}) - c(x_{u..v}) - c(x_{v..w})$$

Applying approach to the real data

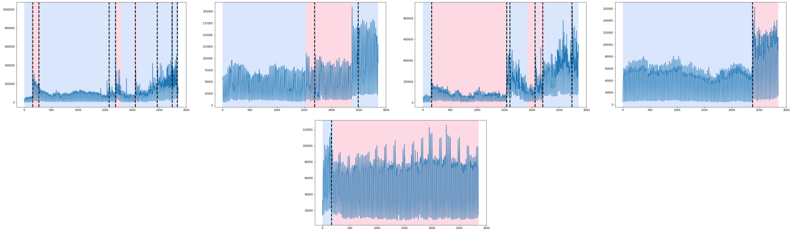
Let's try to apply all of the mentioned cost functions to the time series described above

	custom function	L1	L2	RBF	Autoregression
Country A	1.3582	0.2507	1.305	0.6877	2.3939
Country B	0.3312	0.0942	0.3312	0.8267	0.4111
Country C	1.2686	0.3407	1.1478	0.4265	1.6971
Country D	0.0007	0.0007	0.0007	0.0011	0.0025
Country E	0.4982	0.013	0.7521	0.0042	0.3022

The best result shows L_1 cost function (mean absolute deviation).

Applying approach to the real data. Contine

Let's take a look on the data



Most of the time looks pretty good.

Applying approach to the real data. Contine

Above case was when we defined number of change points in advance. But in real life problems most of the time do not know it. Thus let's try to calculate quantity of change points, using following penalty function:

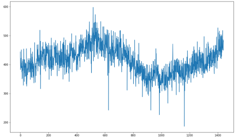
$$pen_{BIC,L_2}(t) := \sigma \log T,$$

Here is the results on number of change points:
Country A — Actual amount: 8; Predicted amount: 5
Country B — Actual amount: 2; Predicted amount: 2
Country C — Actual amount: 6; Predicted amount: 6
Country D — Actual amount: 1; Predicted amount: 2
Country E — Actual amount: 1; Predicted amount: 1

Airpush cases

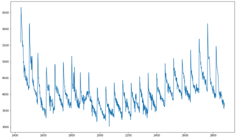
Airpush cases. Fraud elimination

- Apps minutely requests data
- Red flag: strong pattern.
- Can be a automated bots behind pattern



Clean application

Fraud application



Airpush cases. Fraud elimination

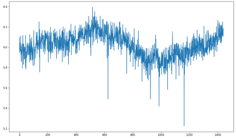
- Goal: to be able to find such apps automatically
- We can reach this goal using frequency analysis

The framework can be described as follows:

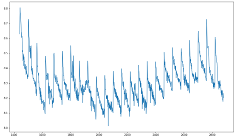
1. Apply logarithm to time series to stabilyze amplitude
2. Remove trend (low frequent part) from time series
3. Apply Fourier transform to time series
4. Estimate the distribution of periodogram values
5. Compare distributions of each application with a distribution of white noise (which is exponential) using Kullback–Leibler divergence
6. If divergence $>$ threshold, then application is marked as suspicious

Airpush cases. Fraud elimination
1. Apply logarithm

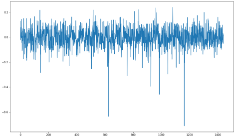
Clean application



Fraud application

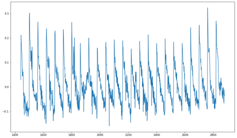


Airpush cases. Fraud elimination
2. Remove trend



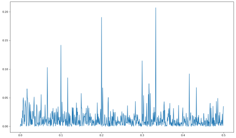
Clean application

Fraud application

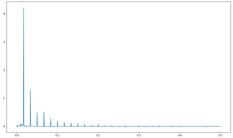


Airpush cases. Fraud elimination
3. Apply Fourier transform

Clean application



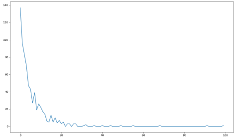
Fraud application



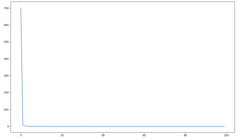
Airpush cases. Fraud elimination

4. Estimate the distribution of periodogram values

Clean application



Fraud application



Airpush cases. Fraud elimination
5. Compare distributions

- Clean app score: 0.09
- Fraud app score: 1.87