Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia
COPPE UFRJ

CHANGE POINT DETECTION IN END-TO-END MEASUREMENTS TIME SERIES

Diego Ximenes Mendes

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Edmundo Albuquerque de Souza e Silva

Rio de Janeiro
Janeiro de 2017

CHANGE POINT DETECTION IN END-TO-END MEASUREMENTS TIME SERIES

Diego Ximenes Mendes

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

RIO DE JANEIRO, RJ – BRASIL
JANEIRO DE 2017

Ximenes Mendes, Diego

Change Point Detection in End-to-End Measurements Time Series/Diego Ximenes Mendes. – Rio de Janeiro: UFRJ/COPPE, 2017.

IX, 12 p.: il.; $29,7$cm.

Orientador: Edmundo Albuquerque de Souza e Silva

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2017.

Bibliography: p. 12 – 12.

1. Change Point Detection. 2. Time Series. 3. Machine Learning. I. Albuquerque de Souza e Silva, Edmundo. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

CHANGE POINT DETECTION IN END-TO-END MEASUREMENTS TIME SERIES

Diego Ximenes Mendes

Janeiro/2017

Orientador: Edmundo Albuquerque de Souza e Silva

Programa: Engenharia de Sistemas e Computação

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

CHANGE POINT DETECTION IN END-TO-END MEASUREMENTS TIME SERIES

Diego Ximenes Mendes

January/2017

Advisor: Edmundo Albuquerque de Souza e Silva

Department: Systems Engineering and Computer Science

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Contributions

1.2 Dissertation Outline

Chapter 2

Literature Review of Change Point Detection

- Here the change point problem will be "defined", including offline and online versions. - we deal with univariate unevenly time series. Explain that some methods can be expanded no multivariate - unknow number of change points - we disconsider changes in periodicity

2.1 Notation

In this work an univariate time series composed of n points is defined by two vectors: $\mathbf{x} = (x_1, ..., x_n)$ and $\mathbf{y} = (y_1, ..., y_n)$. The value y_i indicates the $i-$th sampled value and x_i indicates the associated sample time. It is assumed that the points are sorted by time, that is, $x_{i-1} < x_i$ for $i = 1, ..., n$. Since we consider unevenly time series $x_i - x_{i-1}$ can be different for different i values. For $s \geq t$ the following convention is adopted $\mathbf{y}_{s:t} = (y_s, ..., y_t)$.

The presence of k change points indicates that the data is splitted into $k + 1$ segments, also called windows. Let τ_i indicate the $i-$th change point for $i = 1, ..., k$. Let $\tau_0 = 0$ and $\tau_{k+1} = n$. Then, the $i-$th segment is defined by $\mathbf{y}_{\tau_{i-1}+1:\tau_i}$, assuming that $\tau_{i-1} < \tau_i$ for $i = 0, ..., k + 1$. Therefore $\boldsymbol{\tau} = (\tau_0, ..., \tau_{k+1})$.

2.2 Sliding Window Techniques

Mainly a change point detection algorithm aim to find k and $\boldsymbol{\tau}$.

Describe sliding window techniques in change point detection. Describe how two windows can be compared. Probably this technique will not be part of my solution, but I will write about it since is the simplest and most intuitive solution.

2.3 Combinatorial Optimization Model

2.3.1 Constrained Case

Given a fixed value of k, one approach is to define a cost function that measure the homogeneity of a segment and therefore choose the change points that globally optimize this homogeneity. Let the cost of the i-th segment be defined as $C(y_{\tau_{i-1}+1:\tau_i})$. The cost of a segmentation is then $\sum\limits_{i=1}^{k+1} C(y_{\tau_{i-1}+1:\tau_i})$.

A commom choice for function C is the MSE (Mean Squared Error) which can capture changes in the mean. Another usual approach is to consider a distributin model and use the negative maximum log-likelihood. The latter can capture changes in mean and variance, also considers that data within a segment is iid. Therefore, given a fixed k, the optimal segmentation is obtained through the following optimization problem:

$$F_{k,n} = \min_{\boldsymbol{\tau}_{1:k}} \sum_{i=1}^{k+1} C(y_{\tau_{i-1}+1:\tau_i}) \tag{2.1}$$

This problem can be solved using dynamic programming:

$$F_{k,t} = \begin{cases} 0, & \text{if } k = 0 \text{ and } t = 0 \\ \infty, & \text{if } k = 0 \text{ and } t > 0 \\ \min\limits_{s \in \{0,\ldots,t-1\}} \left[F_{k-1,s} + C(y_{s+1:t})\right], & \text{otherwise} \end{cases} \tag{2.2}$$

The overal time complexity of this algorithm is $O(kn^2 f(n))$, where $f(n)$ is the time complexity to evaluate C.

2.3.2 Segment Cost Function

Several segment cost functions can be evaluated in $O(1)$ after a preprocessing phase, implying in an overall $O(kn^2)$ time complexity. Next is provided the procedures to achieve this efficiency using MSE, negative maximum log-likelihood of normal and exponential distributions.

MSE

Let $\mu_{s,t}$ the mean value of the segment $\mathbf{y}_{s:t}$:

$$\mu_{s,t} = \frac{\sum\limits_{i=s}^{t} y_i}{t - s + 1} \tag{2.3}$$

Then, the $MSE(\mathbf{y}_{s:t})$ is defined as:

$$
\begin{aligned}
MSE(\mathbf{y}_{s:t}) &= \frac{\sum_{i=s}^{t}(y_i - \mu_{s,t})^2}{t-s+1} \\
&= \frac{\sum_{i=s}^{t}(y_i^2 - 2y_i\mu_{s,t} + \mu_{s,t}^2)}{t-s+1} \\
&= \frac{\sum_{i=s}^{t} y_i^2 - 2\mu_{s,t}\sum_{i=s}^{t} y_i + (t-s+1)\mu_{s,t}^2}{t-s+1}
\end{aligned}
\tag{2.4}
$$

Let $S_i = \sum_{j=0}^{i} y_j$, that is, $\mathbf{S}$ represents the prefix sum of $\mathbf{y}$ for different indexes. $\mathbf{S}$ can be computed in $O(n)$ with the following dynamic programming procedure:

$$
S_i = \begin{cases} 0, & \text{if } i = 0 \\ S_{i-1} + y_i, & \text{otherwise} \end{cases} \tag{2.5}
$$

Let $Q_i = \sum_{j=0}^{i} y_j^2$. As with $\mathbf{S}$, $\mathbf{Q}$ can be computed in $O(n)$:

$$
Q_i = \begin{cases} 0, & \text{if } i = 0 \\ Q_{i-1} + y_i^2, & \text{otherwise} \end{cases} \tag{2.6}
$$

Given that $\mathbf{S}$ and $\mathbf{Q}$ are previously computed, it is possible to evaluate the following equations in $O(1)$:

$$
\mu_{s,t} = \frac{S_t - S_{s-1}}{t-s+1} \tag{2.7}
$$

$$
\sum_{i=s}^{t} y_i = S_t - S_{s-1} \tag{2.8}
$$

$$
\sum_{i=s}^{t} y_i^2 = Q_t - Q_{s-1} \tag{2.9}
$$

With equations 2.7, 2.8, 2.9 it is possible to observe that 2.4 can be computed in $O(1)$ given an $O(n)$ precomputation.

Negative Log-Likelihood of Normal Distribution

The pdf of a Gaussian distribution is given by:

$$
N(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{2.10}
$$

The maximum likelihood estimators are:

$$\widehat{\mu} = \frac{\sum_{i=1}^{n}}{n} \tag{2.11}$$

$$\widehat{\sigma}^2 = \frac{\sum_{i=1}^{n}}{n} \tag{2.12}$$

Through these equations is possible to observe that negative of the maximum log likelihood can be expressed using MSE:

$$\begin{aligned} NLL &= \frac{n\ln(2\pi)}{2} + \frac{n\ln(\sigma^2)}{2} + \frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2\sigma^2} \\ &= \frac{n\ln(2\pi)}{2} + \frac{n\ln(MSE)}{2} + \frac{nMSE}{2\sigma^2} \end{aligned} \tag{2.13}$$

Negative Log-Likelihood of Exponential Distribution

- degenerate distribution

2.3.3 Penalized Case

When the number of change points is unknown.

2.3.4 Pruning

- heuristics

2.4 Bayesian Inference

I will erase this section if I don't use bayesian inference. Describe Fearnheard (offline) and MacKay (online) solutions. Say that there are other versions.

2.5 HMM

I will not describe HMM algorithms (viterbi, baum welch, etc), I will only describe how HMM have been used in change point detection. Describe Left-Right HMM, full HMM, and Regularized HMM in this problem.

2.6 Other Algorithms

Only cite other used algorithms and say why I chose the previous one to analyse.

2.7 Performance Evaluation

Describe how datasets are constructed in literature. Describe how an algorithm output is evaluated.

Chapter 3

Dataset

3.1 Description of End-to-End Packet Loss Measurements Time Series

Here will be presented the TGR dataset. Small description on how data are collected, including client informations (geographic position, routes, etc) Plots: distribution between two consecutive measures, autocorrelation after time binarization, loss distribution, hour of day x loss, day of week x loss. Maybe: clusterize clients by distribution or time series.

3.2 Change Points Classification Survey

Describe web system used to get ”true” change points. Describe majority voting. Describe how data were divided in train/test dataset.

Chapter 4

Applying Change Point Detection

In each algorithm section I will: Describe adaptations and aproaches. Describe dificulties of this algorithms in real data and in the current dataset that lead to adaptation.

4.1 Preprocessing

Filters applied to time series before presenting to algorithms.

4.2 Tuning Hyperparameters

4.2.1 Grid Search and Randomized Grid Search

4.2.2 Bayesian Optimization

I don't know if I am going to use this method.

4.2.3 Particle Swarm Optimization

I don't know if I am going to use this method.

4.3 Sliding Window

4.4 Dynamic Programming

4.5 HMM

4.6 Bayesian Inference

I don't know I will use this: poor performance.

4.7 LSTM

I don't know I will use this: maybe I will not have enough data.

4.8 Ensembles

If there are enough models to be tested describe how to use ensembles.

Chapter 5

Results

5.1 Classification Accuracy

Present false positive/false negative/...

5.2 Unsupervised Analysis

Clusterize clients according with change points detected and check if latent information of clusters are also clusterized.

5.3 Algorithms Comparison

Compare algorithms results and computational performance.

Chapter 6

Conclusions

Bibliography