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.

VIII, 18 p.: il.; 29, 7cm.

Orientador: Edmundo Albuquerque de Souza e Silva

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

Bibliography: p. 17 – 18.

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

Change Point Detection

A change point detection algorithm is concerned to identify points in time where the statistical properties of a time series have changed. This problem have a broad application in different knowledge fields, and in general, the algorithms performance are closely related with the input characteristics. Also, if the latent information of the procedures that generated a time series is missing, the target statistical properties can be considered subjective, bringing difficulties not only in the detection phase but also in the problem formalization.

In this context this chapter specifies the problem and briefly discusses several change point detection algorithms. The literature of this area is extensive, and it is common to find methods that presents a poor performance due to a variety of reasons, such as they are too specific or because the mechanisms were only analyzed through theoretical aspects. Therefore, it was chosen a set of methods that can provide a good practical and theoretical perspective, and also flexibility to insert adaptations that can better handle some input peculiarities. Furthermore, through this chapter is exposed several obstacles when dealing with real data, and some adopted solutions which are not described in literature.

2.1 Problem Definition

The problem can be categorized in offline or online. In the offline version, to decide if a specific point at time t is a change point the solver has available the whole time series, including past and future information on t. In the other hand, in the online version the information is available up to time t. The choice between these options is defined by the application domain, in some cases data are processed in real time and the change points should be detected as soon as possible, but in others the changes are identified by historical purposes and offline algorithms can be used.

It is intuitive that the offline case is more robust, since there are more information to make a classification. In practice, to increase the statistical confidence of a decision the online definition is relaxed, and to decide if a point at time t is a change point it is possible to use data up to a small window in the future of t, which in real time processing means that the application should wait until more data are available. This trick plays a trade-off between minimizing the time to detect a change and correctly classify a point. Therefore, in some cases, the online version can be transformed in the offline case by only modifying the input availability.

In this work it is considered the following input and change points characteristics, which were defined considering the final application scenario:

- Univariate time series. However, it is possible to extend several methods presented here to deal with multivariate data.
- Unevenly time series, that is, data is not regularly sampled in time.
- Time series with different lengths.
- Unknown number of change points.
- Different number of points between change points.
- Focus on changes in the underlying mean and distribution, disregarding other kinds of changes such as in periodicity.
- Outliers are not considered statistical changes.
- There is no latent information of the time series.
- It is considered the online and offline options.

2.2 Notation

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 $i = 2, ..., n$. Since unevenly time series is considered, $x_i - x_{i-1}$ 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 implies that data is split into $k + 1$ segments, also called windows. Let τ_i indicates the i-th change point for $i = 1, ..., k$. Also let $\tau_0 = 0$, $\tau_{k+1} = n$ and $\boldsymbol{\tau} = (\tau_0, ..., \tau_{k+1})$. Then, the i-th segment is defined by $\mathbf{y}_{\tau_{i-1}+1:\tau_i}$, assuming that $\tau_{i-1} < \tau_i$ for $i = 1, ..., k + 1$.

Through the previous definitions, change point detection algorithms mainly aim to find both k and $\boldsymbol{\tau}$.

2.3 Sliding Windows

Sliding windows techniques use two sliding windows over the time series, and reduce the problem of detecting change points to the problem of testing whether data from the segments were generated by different distributions. One approach is to consider a distance metric between two empirical distributions as the base to infer the change points. Letting $d(\mathbf{a}, \mathbf{b})$ be the distance between two empirical distributions defined by the windows $\mathbf{a}$ and $\mathbf{b}$, and considering windows of length m, the Algorithm 1 presents a simple sliding windows method.

Algorithm 1 Sliding Windows

$i \leftarrow 1$
while $i + 2m - 1 \leq n$ **do**
  if $d(\mathbf{y}_{i:i+m-1}, \mathbf{y}_{i+m:i+2m-1}) > \alpha$ **then**
    Report $i + m - 1$ as a change point
    $i \leftarrow i + m$
  else
    $i \leftarrow i + 1$
  end if
end while

In this method, when the distance between the distributions is above some threshold α a change point is reported. This is a common approach for an online application, however, it is possible to increase the classification accuracy in offline cases. As an example, the top plot of figure 2.1 presents a simulated time series, the segment $\mathbf{y}_{1:1000}$ was generated sampling a $N(1, 0.2)$ distribution, and $\mathbf{y}_{1001:2000}$ was sampled through $N(5, 0.2)$. The distribution of a window was constructed binning the data with bins of size 0.02. The bottom plot of the same figure presents the associated Hellinger distance [1] between two sliding windows, where the point (i, H_i) represents the distance between the windows $\mathbf{y}_{i-100:i-1}$ and $\mathbf{y}_{i:i+99}$.

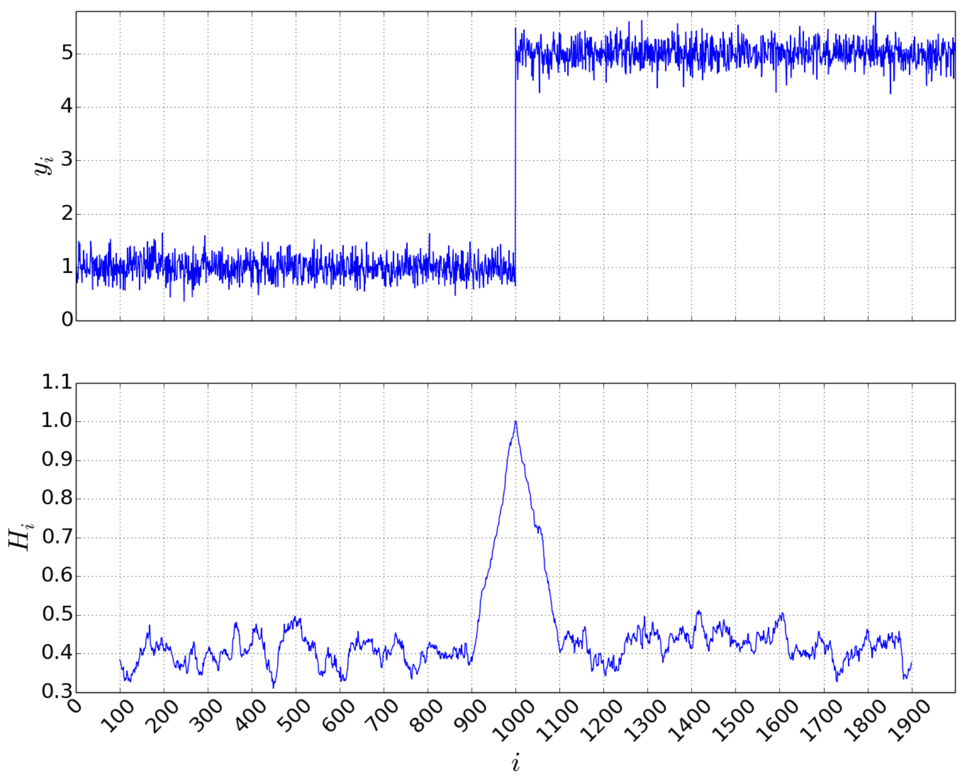


Figure 2.1: Toy example of a sliding windows method.

It can be observed that there is a peak on the distance in the exact location where the distribution changed. However, using only the threshold method it is possible to prematurely infer the position of the change point, therefore, an alternative is to also use a peak detection algorithm. Besides, the distance function choice has a direct impact on the classification accuracy.

As stated in [2], a performance improvement can be achieved concurrently executing the same sliding windows algorithm, however, with different windows lengths, which facilitates the detection of segments with distinct number of points.

2.4 Optimization Model

Given a fixed value of k, one approach is to define a cost function that measures the homogeneity of a window, and therefore, choose the change points that globally optimize this homogeneity. Let the cost of the i-th segment be defined as $C(\mathbf{y}_{\tau_{i-1}+1:\tau_i})$, then the cost of a segmentation is the sum of all segments costs.

A common choice for the function C is the MSE (Mean Squared Error), which can capture changes in the mean. Another usual approach is to consider distribution changes through negative maximum log-likelihood functions, considering that data within a window is iid.

Therefore, given a fixed k, the optimal segmentation is obtained through the following optimization problem, which is called the constrained case [3]:

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

This problem can be solved using dynamic programming with $O(kn^2 f(n))$ time complexity, where $f(n)$ is related with C evaluation. Several segment cost functions can be evaluated in $O(1)$ after a $O(n)$ preprocessing phase, implying in an overall $O(kn^2)$ complexity. It is possible to proof that MSE, negative maximum log-likelihood functions of normal, exponential, poisson and binomial distributions have this characteristic. Also, the formulation can consider a minimum value of a window length.

Modeling segments with continuous distributions can lead to practical difficulties. One of them is the fact that segments can form degenerate distributions, that is, the data of a window can have zero variance, which is always the case of unitary length segments. In these scenarios the negative maximum log-likelihood is undefined. Two approaches can be used to overcome this situation. The first one tries to avoid degenerate segments adding a white noise with small variance to the time series. The second one considers that the cost of any degenerate distribution is equal to a constant.

When the number of change points is unknown an usual way is to introduce a non decreasing penalty function $g(k)$. Then, the new optimization problem, called penalized case [3], is:

$$\min_{k, \boldsymbol{\tau}_{1:k}} \sum_{i=1}^{k+1} C(\mathbf{y}_{\tau_{i-1}+1:\tau_i}) + g(k) \tag{2.2}$$

This problem can be solved in $O(Kn^2 f(n))$. However, if the penalty function is linear in k, the problem can be formulated more efficiently and solved in $O(n^2 f(n))$.

Also, there are several pruning algorithms to speedup the computation [3–5], in general trying to reduce the $\boldsymbol{\tau}$ search space but maintaining optimality.

2.5 HMM (Hidden Markov Model)

The idea that each segment is associated with a specific latent configuration has a direct interpretation to a HMM model [6–8]. In this context, each window is related to a hidden state of a HMM, and the observation distribution of this state represents the distribution of that segment. Therefore, the mechanism models the time series using a HMM, and through the hidden state path assesses the times when a transition between different hidden states occur.

There are several approaches in the detection and training phases. For example, given a trained HMM, is possible to analyze the most probable hidden state path that a time series can follow through the viterbi algorithm. Also, it is possible to evaluate the probability of a transition between different hidden states at time t, and then apply a threshold and peak detection methods, as well as in sliding windows techniques. For the training step, it is possible to use several time series to train a single HMM, and then use this model to detect change points in all time series. Another way is to, for each time series, train a single model using only the target time series.

It is important to note that the structure of the hidden state graph has a large impact on the performance. Using a fully connected graph, the number of states defines the maximum number of distribution configurations. Employing a left to right structure, the number of hidden states will induce the maximum number of segments.

In [8] is stated that when using a fully connected structure, the time interval that a time series stays in the same hidden state is low, which can not reflect real data. To overcome this problem, [8] suggests to increase the time that a time series stands in the same hidden state, using a dirichlet prior regularization.

2.6 Bayesian Inference

There are several Bayesian methods with the objective to assess the probability that a point is a change point. Following an offline fashion, the work of [9] recursively calculates, for each i, the probability of $\mathbf{y}_{i:n}$ given a change point at i. With these probabilities is possible to simulate the time of the first change point, and then, compute the conditional distribution of the time of the second change given the first, and so on. To achieve this, the mechanism assumes that observations are independents, and that each segment is modeled by conjugate priors. Also, the procedure considers priors to model the number of changes and the time between two consecutive change points. The overall complexity of this method is $O(n^2)$, considering that the likelihood of a segment can be evaluated in $O(1)$.

In [10] it is also considered that parameters of different segments are independents, and that data within a window is iid. However, through an online mode, the procedure is concerned with the estimation of the distribution of the length of the current time since the last change point, called run length, given the data so far observed. To achieve this, the method assumes the probability of current run length given the last run length as a prior. Assuming exponential-family likelihoods to model a segment, the time complexity to process a point is linear in the number of points already observed.

Chapter 3

Dataset

This chapter describes the used datasets, presenting their construction methodology and a brief descriptive analysis.

3.1 End-to-End Packet Loss Fraction Time Series Dataset

3.1.1 Methodology

The time series presented in this work represent network end-to-end measures of a cable-television infrastructure running DOCSIS, with asymmetric download and upload bandwidths. Home routers connected to the cable modem communicates with one or more servers strategically located by the ISP. Measurement results from each home router are consolidated every half hour and, by the end of every day, are transferred to a database for analysis. The software responsible for these procedures was developed by TGR in partneship with UFRJ.

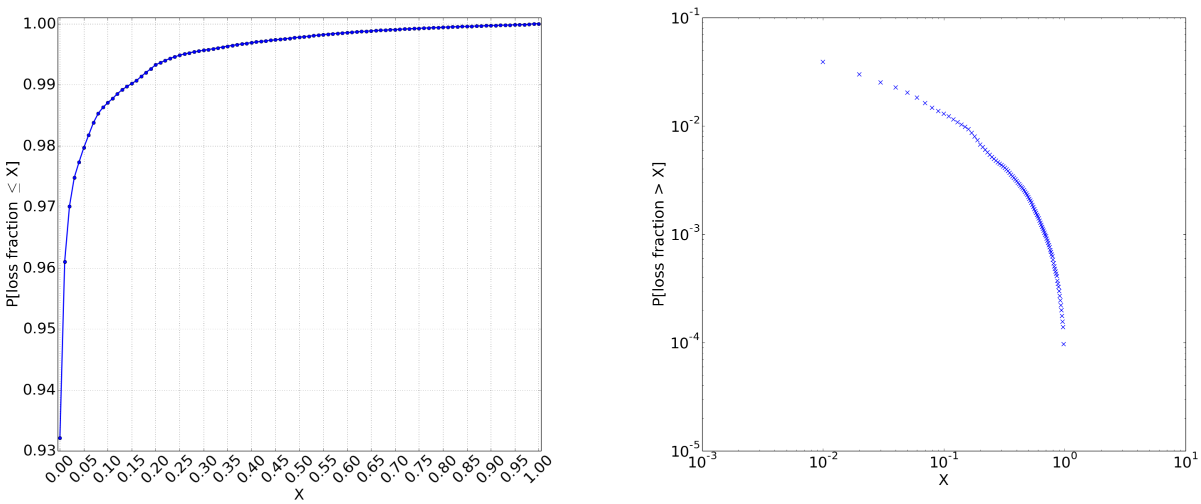
The focus of this work is to analyze the loss fraction, however, [11] presents a further analysis on other metrics, such as link throughput, round trip latency and loss bursts. To measure the round trip packet loss fraction between the home router and the associated server, the home router sends a train of short UDP packets, and then the server bounces back them. The data presented here considers a train of 100 UDP packets of 32 bytes and 1 milliseconds apart.

The resulted time series are unevenly due to a range of reasons. One of them is that measurements are initiated only if the residential link is not under use by the ISP customer. Also, it is possible to the home router be without Internet access to start a measurement, or even without electrical energy.

3.1.2 Descriptive Analysis

The TGR software run in a subset of the ISP customers in several brazilian states. In the analysis of this section, it was considered the period from 01/may/2016 to 20/may/2016. It was selected every client in which all measures of this period occurred against the same server. Another imposed constraint was that the home router and the server must belong to the same state. This filter resulted in 1870 clients time series and the total of 1537272 measures.

In figure 3.1 is presented the packet loss fraction CDF and CCDF of all time series. It is possible to note that 93.2% of the measures have zero losses.

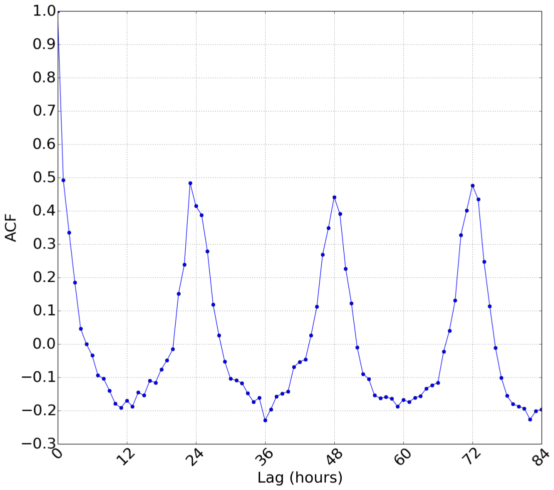


(a) CDF

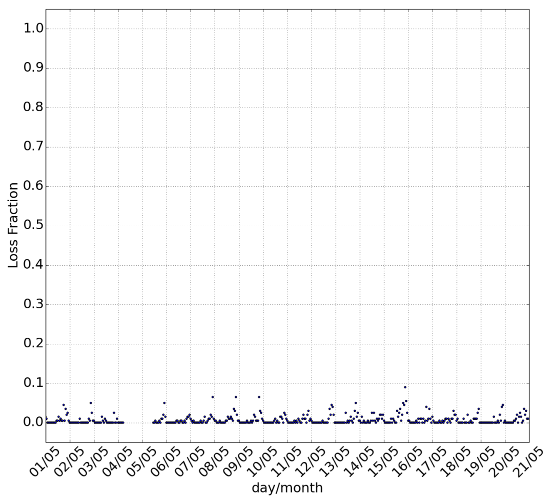
(b) CCDF in log-log scale

Figure 3.1: Packet Loss Fraction

Figures 3.2, 3.3, 3.4 are three examples of common autocorrelation patterns in this dataset. Since the time series are unenvely, it was considered only the date and hour components of a measure time to compute the autocorrelation function, ignoring then the minutes and seconds information. Therefore, if more than one measure occurred at the same hour and date, it was taken the average of these samples.

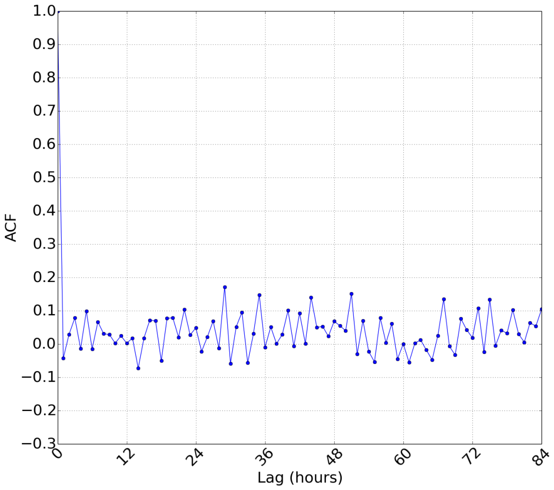


(a) Autocorrelation

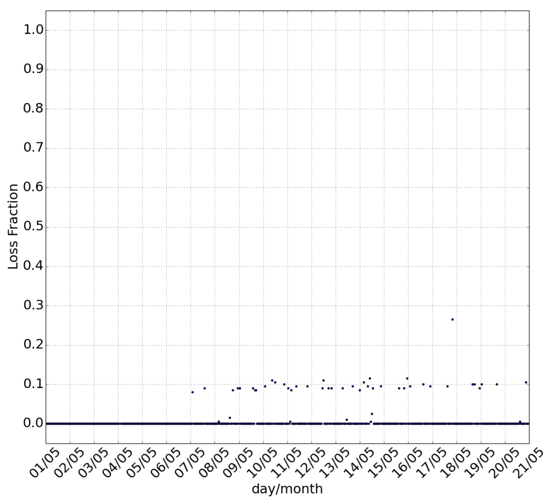


(b) Time Series

Figure 3.2: Client 1

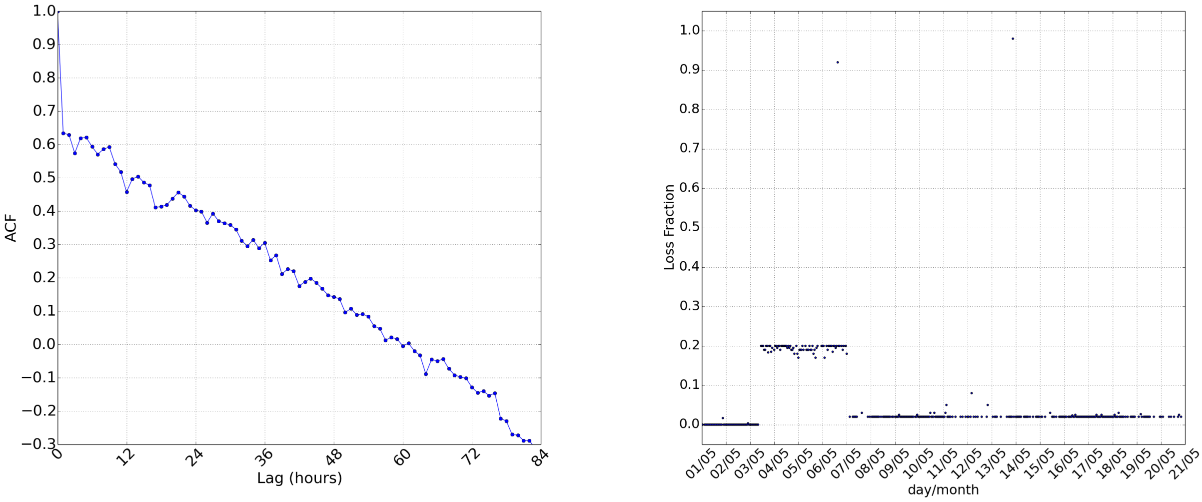


(a) Autocorrelation



(b) Time Series

Figure 3.3: Client 2



(a) Autocorrelation

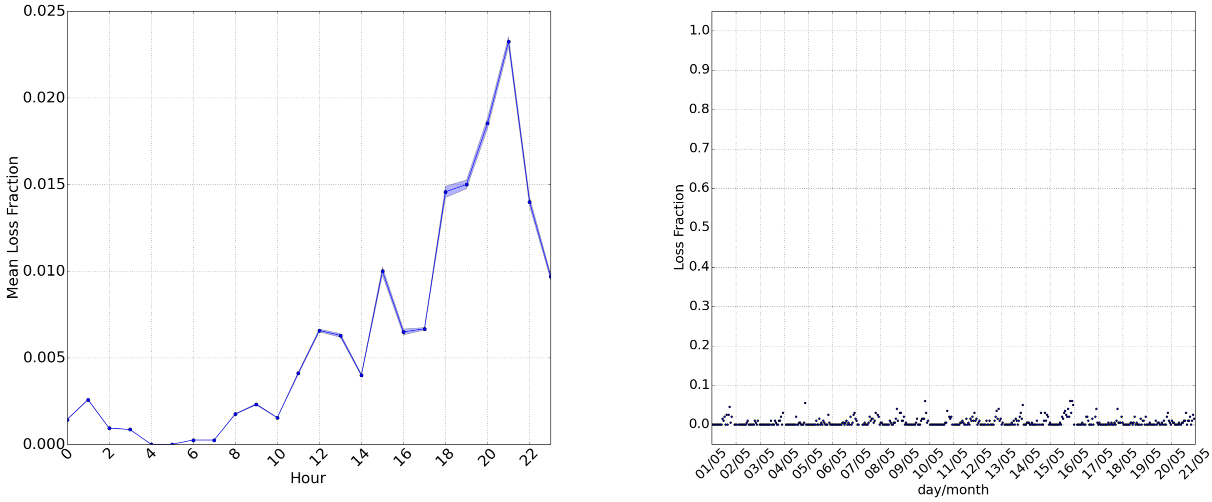
(b) Time Series

Figure 3.4: Client 3

The autocorrelation of figure 3.2 have a periodic pattern, with peaks around multiples of 24 hours. In this client, it is possible to observe that losses a more usual at night. However, in figure 3.3, the autocorrelation quickly decreases and fluctuates around zero. The corresponding time series have a common characteristic, from 06/may on measures with losses alternated with zero losses measures. Figure 3.4 shows a linear decreasing autocorrelation, and the associated time series presents abrupts changes in the mean.

Also, through a visual analysis, as in the previous figures, it is possible to observe different time series and change points patterns.

As in figure **??** same clients have a daily pattern in which losses occur more frequently at night, in which the Internet usage is known to be bigger. Figure 3.5 corroborates that observation, which presents the mean and variance of the measures that occurred in a specific hour during the 20 days period. This can be a indication of congestion during peak of usage hours.



(a) Mean and variance per hour

(b) Time Series

Figure 3.5: Client 1

3.2 Change Points Dataset

There are several approaches to construct a change points dataset to test a change point detection method. Some works in the literature simply create a simulated time series thorugh generative models, however different segments are generated by the same model but with different parameters. In general, this types of time series are more easily handled in change point detection algorithms, since some methods assumes the same models that generated the data or because real data can have more complex characteristics. Another way is to create a time series appending different segments provenient from different real time series data. In this cases is easy to simplify the dataset since some characteristics as the segments lengths are defined by the dataset creator and another ones are fakely introduced. Another works assumes that the latent information of the time series are available, and with specific knowledge of the application field, it is possible to assume what kinds of configuration changes could induce a change in the time series. In the case of the application field of the present work this would be difficult, since the latent information would be connected with the network situation such as network topology, characteristics of routers congestions, physical equipments problems, which is a too complex, or even impossible information to collect and also to assume which variables could impact the time series. Other apprach also uses the latent information but instead creates a controlled environment in which is possible to change the configuration over time, but as in the previous case, this would be too complex. The approach followed by this work was to use a visual annotation of the time series. It were conducted, with an application domain specialist, visuallyu indicating the relevant changes. It is known that human visual inspection methods can bring erroneous conclusions, but with the data and application scenarion it was the best fit, as network engineers visually identify, after simple automatic filters, the change points.

This work is interested in work directly with real data and satisfy a real application problem.

As in other tasks, it is difficult to translate a human visual perception in a sistematically method.

In general, when the exact types of target changes are previously know the problem is easier.

3.3 Methodology

It was created an annotation system to a specialist visually indicates the change points of the time series dataset. The user indicated with the mouse the points in time where he thinks that where the cahnge points occurred. To avoid results misunderstanding X axis of the time series represented onlu the time order of the measures, since the time series are unevenly this fact could visually wrong infer change points. Also, since is known that variances in the losses fractions when the losses are low have more impact in the user QoE than when the losses are big, the system also provides two other y scales than linear, the log scale and piecewise linear with bigger length in [0, 0.1] than (0.1, 1.0]. The user can click in any scale.

A single person classified all time series. This person has experiences with academic and industry network measurements and statistical modelling, however without background in change point detection analysis. The user could take the time he wants to make a classification and it was able to classify in different days. Before the user could start the classifications, it was indicated a series of instructions:

- In the case of packet loss fraction, mean changes between 0 and 0.1 are more sensible to the end users.
- The "time" axis only represents the temporal order of the measurements. However, in general, consecutive points in "time" axis are separated by 30 minutes.
- Outlier is not a statistical change. An outlier is an observation that lies outside the overall pattern of a distribution.

Since several time series of the previously described time series have almost all measures with zero losses, these time series was filtered to reduce the number of time series and keep only the ones which provide change points, increasing the entropy. Also, to better the visualization, it was selected time with 10 days of data, therefore the dataset consists of time series from 1 may 2016 to 10 may 2016, and from 11 may 2016 to 20 may 2016. The specific filter was: it was selected only the time series that has 85% of the maximum possible data in the specified time period, considering that each home router executes the measurement procedure at most two times in a hour. Reducing to 522 time series. Also it were only selected the time series that have at least one window of length 48 with at least 6 measures with loss fraction bigger than 0.01.

3.4 Descriptive Analysis

- explain possible ways to get the ground truth
- description of the volunteer
- user instructions
- snapshots of system
- how time series were selected to be in the survey

3.5 Description of Change Points Dataset

- number of time series, number of change points: it is a high dimensional problem
- distribution number of changes per time series
- distribution time between change points
- distribution time for the first change point
- distribution time from last change point to time series end
- distribution of classification time?
- measure the difference between consecutive segments?

3.6 Performance Evaluation

- how the performance is asserted in literature
- ROC curve
- confusion matrix and accuracy metrics

Chapter 4

Conclusions

Bibliography

[1] WIKIPEDIA. "Hellinger distance". 2016. https://en.wikipedia.org/wiki/Hellinger_distance/.

[2] KIFER, D., BEN-DAVID, S., GEHRKE, J. "Detecting Change in Data Streams". In: *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pp. 180–191. VLDB Endowment, 2004. ISBN: 0-12-088469-0. Disponível em: <http://dl.acm.org/citation.cfm?id=1316689.1316707>.

[3] MAIDSTONE, R., HOCKING, T., RIGAILL, G., et al. "On optimal multiple changepoint algorithms for large data", *Statistics and Computing*, pp. 1–15, 2016.

[4] KILLICK, R., FEARNHEAD, P., ECKLEY, I. "Optimal detection of changepoints with a linear computational cost", *Journal of the American Statistical Association*, v. 107, n. 500, pp. 1590–1598, 2012.

[5] HAYNES, K., ECKLEY, I. A., FEARNHEAD, P. "Computationally Efficient Changepoint Detection for a Range of Penalties", *Journal of Computational and Graphical Statistics*, v. 0, n. ja, pp. 1–28, 0. doi: 10.1080/10618600.2015.1116445. Disponível em: <http://dx.doi.org/10.1080/10618600.2015.1116445>.

[6] KEHAGIAS, A. "A hidden Markov model segmentation procedure for hydrological and environmental time series", *Stochastic Environmental Research and Risk Assessment*, v. 18, n. 2, pp. 117–130, 2004. ISSN: 1436-3259. doi: 10.1007/s00477-003-0145-5. Disponível em: <http://dx.doi.org/10.1007/s00477-003-0145-5>.

[7] LUONG, T. M., ROZENHOLC, Y., NUEL, G. "Fast estimation of posterior probabilities in change-point analysis through a constrained hidden Markov model", *Computational Statistics and Data Analysis*, v. 68, pp. 129 – 140, 2013. ISSN: 0167-9473. doi: http://dx.doi.org/10.1016/

j.csda.2013.06.020. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167947313002326>.

[8] MONTAÑEZ, G. D., AMIZADEH, S., LAPTEV, N. "Inertial Hidden Markov Models: Modeling Change in Multivariate Time Series". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 1819–1825. AAAI Press, 2015. ISBN: 0-262-51129-0. Disponível em: <http://dl.acm.org/citation.cfm?id=2886521.2886573>.

[9] FEARNHEAD, P. "Exact and efficient Bayesian inference for multiple changepoint problems", *Statistics and Computing*, v. 16, n. 2, pp. 203–213, 2006. doi: 10.1007/s11222-006-8450-8. Disponível em: <http://dx.doi.org/10.1007/s11222-006-8450-8>.

[10] ADAMS, R. P., MACKAY, D. J. "Bayesian online changepoint detection", *arXiv preprint arXiv:0710.3742*, 2007.

[11] MENDES, D. X., SENGES, G. D. S., SANTOS, G. H. A. D., et al. "A Preliminary Performance Measurement Study of Residential Broadband Services in Brazil". In: *Proceedings of the 2016 Workshop on Fostering Latin-American Research in Data Communication Networks*, LANCOMM '16, pp. 16–18, New York, NY, USA, 2016. ACM. ISBN: 978-1-4503-4426-5. doi: 2940116.2940135. Disponível em: <http://doi.acm.org/2940116.2940135>.

[12] AUGER, I. E., LAWRENCE, C. E. "Algorithms for the optimal identification of segment neighborhoods", *Bulletin of Mathematical Biology*, v. 51, n. 1, pp. 39–54, 1989. ISSN: 1522-9602. doi: 10.1007/BF02458835. Disponível em: <http://dx.doi.org/10.1007/BF02458835>.