Laboratoire
Paul Painlevé

Mémoire de Master

# Low Rank Approximation for Kernel Change-Point Detection


*Encadrant :*
Alain CELISSE

*Rapport rédigé par :*
Ali MOURTADA


Promotion 2020

# Acknowledgements

I would like to thank my supervisor Mr. Alain Célisse, first for proposing this Master thesis, the subject was new for me, I learned a lot from it and it allowed me to apply a lot of the things that I have learned this year in the Master and in DAD.

I want to thank him especially for the consistant supervising during this period of nearly seven months. He kept pushing me towards things I was not sure I am capable of. Thank you Sir for giving from your precious time to meet me every week.
This work would not have been done without, first your previous brilliant papers, and second, without the ideas that you have helped me with.

I hope this work will be a subject of satisfaction for you even though we did not reach some of the goals we set.

I want to thank our professors Mr. Dermoune, Mr. Matheron, Mr. Heinrich and Mr. Hardy. What you taught us in this Master helped me to be confortable while reading the main papers from the literature and understanding some of the difficult proofs and results and it certainly helped me to establish the main theoretical result of this work.

I want to thank also Mr. Delaire and all the members of the jury.

# Contents

# Introduction

Change-point detection is a statistics' problem that aims to study the abrubt change in the the distribution of a flow of data (Times series for example). It was a subject of mathematical studies since the 30's.

In time series, stationarity is usually assumed, we would think naturally that the distribution of the data is invariant over a specific period in time. But this assumption no longer holds in practice. Changes occur in the distributions' parameters which makes any study of times series difficult and compromised.
Here comes another reasonable assumption which is that the distribution is stationnary only on a number of segments instead of the whole interval of study. Hence, the goal of change-point detection is to detect these segments where the distribution is the same and study the signal separately in each segment.

An example of the use of change-point techniques in practice is in Finance.
The time series models proposed by quantitative analyst tend to work well in the short term, but they noticed that in the long term, the models fail. They have come to decide that the models should be updated or reparametrized several times. So, change-point detection is used to find the best segmetation of a time series in an interval, and then in each segment, a specific model is applied. This process tend to yield better results for time series forecasting.
we can also use change-point detection to detect changes in the scene in videos for example or to detect abrubt changes in DNA sequences. There are several applications of change-point detection in multiple fields of knowledge

There are two types of change-point detection, **On-line** and **Off-line**.
In the on-line change-point detection, data points are received sequentially in real time, and in the off-line change-point detection, we work with the entire signal at once.
We will be interested in this work in the Off-line change-point detection problem.

The subject of this master thesis is the study of Kernel change-point detection (or **KCP**) with low rank approximation.

Kernel change-point detection is proven to be a state of the art method for multiple change-point detection. But is has one drawback, it can be time and memory consuming because of the use the the kernel or its associated Gram matrix.
The goal of our work is to investigate the kernel change-point detection and explore two main low rank approximation methods to see how we can accelerate the KCP, and eventually give a mathematical analysis of these approximation methods in the framework of KCP.

# Chapter 1

# Kernel Change Point Detection

In Statistical Learning, Kernel Methods or Kernel Machines fall into a class of very robust methods. kernels are used in Support Vector Machine classifiers, in Regression problems, etc.

Kernels are the main brick of the kernel change-point detection problem. They were introduced in the context of change-point detection by Harchaoui and Cappé [2007] and Celisse et al. [2012] who developped a new exact Kkernel change-point detection algorithm.

## 1.1 Kernel Methods

### 1.1.1 Definitions

Let $\mathcal{X}$ be a vector space. A kernel $k$ defined on $\mathcal{X}$ is basically a measurable fucntion from $\mathcal{X} \times \mathcal{X}$ to $\mathbb{R}$.
If $x_1, x_2, ..., x_n \in \mathcal{X}$ are points in $\mathcal{X}$, then we can define the **Gram Matrix** associated to the kernel $k$ as :

$$\boldsymbol{G} := [k(x_i, x_j)]_{1 \leq i,j \leq n}$$

A kernel is called positive semi-definite if the associated Gram matrix is positive semi-definite for every set of points $x_1, x_2, ..., x_n \in \mathcal{X}$. That means that for every $n \in \mathbb{N}^*$, for every $x_1, x_2, ..., x_n \in \mathcal{X}$ and for every $\lambda_1, \lambda_2, ..., \lambda_n \in \mathbb{R}$ we have :

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j k\left(x_i, x_j\right) \geq 0$$

The majority of kernels used in Machine Learning are chosen to be positive semi-definite because of their ability to measure distances in some features spaces. Therefore, in what follows, we will use simply “kernel” instead of positive semi-definite kernel.

### 1.1.2 Examples

Here are examples of the mainly used kernels in Statistical Learning :

- **The Linear Kernel** : The vector space in this case is $\mathcal{X} = \mathbb{R}^d$, and the kernel is defined as the scalar product in $\mathbb{R}^d$, $k(x, y) := \langle x, y \rangle_{\mathbb{R}^d}$
  It is simple to verify that

$$\sum_{i,j=1}^{m} \lambda_i \lambda_j k_\ell (x_i, x_j) = \left\| \sum_{i=1}^{m} \lambda_i x_i \right\|^2 \geq 0$$

- **The Polynomial Kernel** : It generalizes the first one. The Polynomial kernel of degree $\alpha \in \mathbb{N}^*$ is defined as $k(x, y) := (\langle x, y \rangle_{\mathbb{R}^d} + c)^\alpha$ for $x, y \in \mathcal{X} = \mathbb{R}^d$ with $c \geq 0$.

- **The Gaussian Kernel** : It is the most used kernel in Machine Learning and is defined as :

$$k(x, y) = \exp\left( \frac{-\|x - y\|^2}{2\gamma^2} \right)$$

  The Gaussian Kernel is positive semi-definite because

$$\sum_{i,j=1}^{n} \lambda_i \lambda_j k (x_i, x_j) = \sum_{i,j=1}^{n} \lambda_i \lambda_j \mathbb{E}\left[ \mathrm{e}^{\mathrm{i}\gamma^{-1}(x_i - x_j)^{\mathrm{T}} Z} \right] = \mathbb{E}\left[ \left| \sum_{i=1}^{n} \lambda_i \mathrm{e}^{\mathrm{i}\gamma^{-1} x_i^\top Z} \right|^2 \right] \geq 0$$

  where $Z$ is a standard Gaussian random variable $\mathcal{N}(0, I_d)$.

### 1.1.3 The Kernel Trick

It is clear that any scalar product is a kernel. That raises the question whether the other sense is true or not.
Let's get back to the polynomial kernel and let $k$ be a polynomial kernel of degree $\alpha = 2$. Let $x = (x_1, ..., x_d)$ and $y = (y_1, ..., y_d)$ be two vectors from $\mathbb{R}^d$. Then

$$\begin{aligned}
k(x, y) &= \left( \sum_{i=1}^{d} x_i y_i + c \right)^2 \\
&= \left( \sum_{i=1}^{d} x_i y_i \right)^2 + 2c \left( \sum_{i=1}^{d} x_i y_i \right) + c^2 \\
&= \sum_{i=1}^{d} \left( x_i^2 \right) \left( y_i^2 \right) + \sum_{i=2}^{d} \sum_{j=1}^{i-1} \left( \sqrt{2} x_i x_j \right) \left( \sqrt{2} y_i y_j \right) + \sum_{i=1}^{d} \left( \sqrt{2c} x_i \right) \left( \sqrt{2c} y_i \right) + c^2 \\
&= \langle \Phi(x), \Phi(y) \rangle
\end{aligned}$$

where

$$\Phi(x) = \left( x_1^2, \ldots, x_d^2, \sqrt{2} x_d x_{d-1}, \ldots, \sqrt{2} x_d x_1, \sqrt{2} x_{d-1} x_{d-2}, \ldots, \sqrt{2c} x_d \ldots, c \right)^\top \in \mathbb{R}^{\frac{(d+1)(d+2)}{2}}$$

So, the polynomial kernel $k$ is a scalar product in the new larger feature space $\mathbb{R}^{\frac{(d+1)(d+2)}{2}}$. $\Phi$ is called the feature map, which is the function that maps each point $x \in \mathcal{X}$ to a new point $\Phi(x)$.
The question is whether we can generalize this on all the kernels or not.

Fortunately this has been proven by **Moore-Aronszajn** in the general case.

**Theorem 1** ***(The Kernel Trick)*** *A Kernel $k$ is a positive semi-definite kernel on $\mathcal{X}$ if, and only if, there exists a Hilbert space $\mathcal{H}$ and a mapping $\Phi : \mathcal{X} \to \mathcal{H}$ such that*

$$\forall x, y \in \mathcal{X}, k(x, y) = \langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}}$$

.

The feature map $\Phi$ sends the points of $\mathcal{X}$ to a much more interesting Hilbert space $\mathcal{H}$ usually bigger than $\mathcal{X}$ (In terms of dimension).
For exemple, the Gaussian kernel is associated to a Hilbert space of infinite dimension. The polynomial kernel of degree 2 is associated with a Hilbert space of dimension $\frac{(d+1)(d+2)}{2}$ as seen previously.
In Statistical Learning, the kernel trick enables us to work in a more complex space were the data is well represented in order to perform Machine Learning methods without having the need to acces the feature map $\Phi$ and performing explicite computations in $\mathcal{H}$. It is due to the fact that the distance between points in $\mathcal{H}$ is simply computed using the kernel (Or the Gram matrix) :

$$\begin{aligned} d(\Phi(x), \Phi(y))^2 &= \|\Phi(x) - \Phi(y)\|_{\mathcal{H}}^2 \\ &= \langle \Phi(x) - \Phi(y), \Phi(x) - \Phi(y) \rangle_{\mathcal{H}} \\ &= \langle \Phi(x), \Phi(x) \rangle_{\mathcal{H}} + \langle \Phi(y), \Phi(y) \rangle_{\mathcal{H}} - 2\langle \Phi(x), \Phi(y) \rangle_{\mathcal{H}} \\ &= k(x, x) + k(y, y) - 2k(x, y) \end{aligned}$$

We will now define a **R**eproducing **K**ernel **H**ilbert **S**pace which is a very important notion in the kernel change-point detection framework. Its properties will be exploited in the Theoretical analysis section.

**Definition 1** ***(RKHS)*** *Let $\mathcal{X}$ be a non-empty set and $\mathcal{H}$ be a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$. Then $\mathcal{H}$ is called a RKHS if there exists a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that satisfies the reproducing property*

$$\forall f \in \mathcal{H}, \quad \forall x \in \mathcal{X}, \quad \langle f, k(x, \cdot) \rangle = f(x)$$

*In particular, $\langle k(x, \cdot), k(y, \cdot) \rangle = k(x, y)$*
*We then say that $k$ is a Reproducing kernel.*

It has been proven that every positive semi-definite kernel is a reproducing kernel.
This property implies that the canonical feature map is of the form $\boldsymbol{\Phi(x) = k(x, \cdot)}$.

This was a short introduction on Kernel methods. It will be very useful in the next section where we will elaborate on the main problem which is kernel change-point detection.
For further details on Kernel methods, we refer to Garreau [2017].

## 1.2 Kernel change-point detection

In this section, we will introduce the problem of change-point detection and mainly the kernel change-point detection. We will also present and discuss two kernel change-point detection algorithm, the first one is the **KCP** algorithm introduced by Celisse et al. [2012] which exploits dynamic programming techniques to find the exact segmentation of a signal. The second one is the Kernelized Binary Segmentation algorithm introduced in Celisse et al. [2017]. This one will be of intrest to us because it will allow us to work with the two main approximation methods to scale up the solution of the change-point detection problem.

### 1.2.1 Change-point problem

Let's consider $n$ random variables $X_1, ..., X_n$ in a measurable space $\mathcal{X}$. The goal of the change-point detection is to detect abrupt changes in the distribution of the variables, that means the exact points where the distribution changes. This implies that between two consecutive change-points, the distribution is constant in that segment.
For a fixed number of change-points $D \in \{1, ..., n\}$, and a set of points $0 = \tau_0 < \tau_1 < ... < \tau_D = n$, we define the segmentation $\tau := [\tau_0, ..., \tau_D]$ of $\{1, ..., n\}$ which is the set of segments of the form $\tau_{\ell-1} + 1, \tau_{\ell-1} + 2, ..., \tau_\ell$ where $l \in \{1, ..., D\}$. Usually, $D$ is denoted $D_\tau$ which is the number of change-points in the segmentation $\tau$
The change-points are the points $\tau_\ell$. We note $\mathcal{T}_n^D$ the set of segmentation of $D$ segments

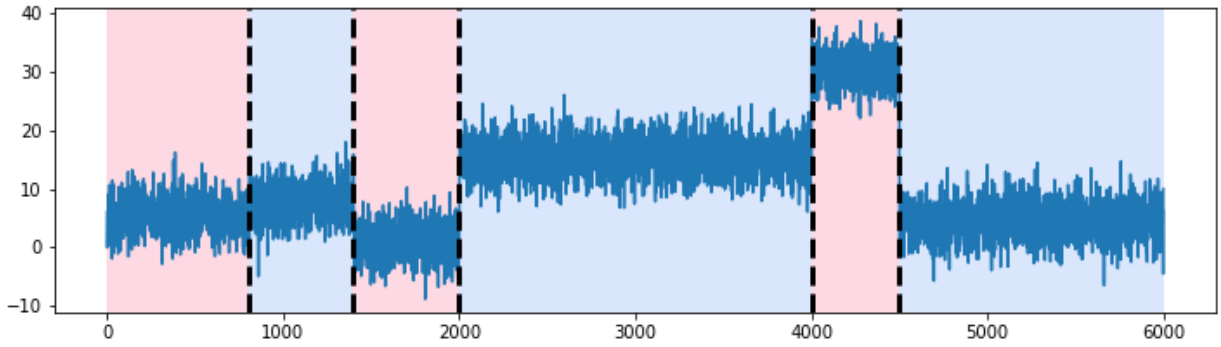
$$\mathcal{T}_n^D := \left\{(\tau_0, \ldots, \tau_D) \in \mathbb{N}^{D+1} / 0 = \tau_0 < \tau_1 < \tau_2 < \cdots < \tau_D = n\right\}$$

and $\mathcal{T}_n := \cup_{D=1}^n \mathcal{T}_n^D$ which is the set of all the segmentations of $\{1, ..., n\}$.

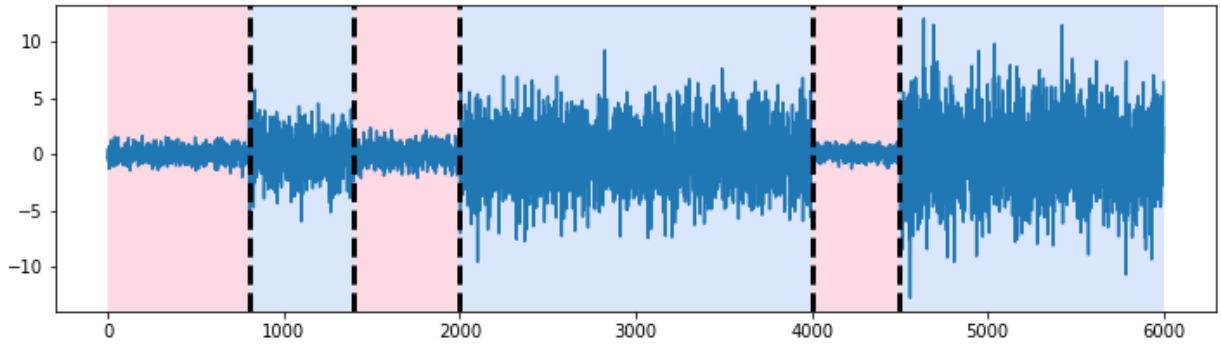
What do we mean by a change in the distribution?

Well, there are several types of change in the distribution of $n$ random variables, the simplest one is the change in the mean as we see in figure 1.1(a).
Another type of change is the change in the variance of the signal, or in any other moment of the variables. This type of change is usually harder to detect with the naked eye.
A change can also occur in the frequency of a signal, etc.

In the two figures, we show a change in the distribution in six points. The changes occured in the mean in 1.1(a) and occured in the variance in 1.1(b).




(a) Change in the mean of the distribution




(b) Change in the variance of the distribution

If we denote $P_{X_i}$ the distribution of $X_i$. The true segmentation $\tau^\star := [\tau_0, ..., \tau_D]$ should verify :

$$\forall \ell \in \{1, ..., D\} : \quad P_{X_{\tau_{\ell-1}+1}} = P_{X_{\tau_{\ell-1}+2}} = ... = P_{X_{\tau_\ell}}$$
$$P_{X_{\tau_1}} \neq P_{X_{\tau_\ell}} \neq ... \neq P_{X_{\tau_D}}$$

In the change-point detection problem, we either work under a known number of change point detection $D$, so, the goal is to find the exact positions of the changes in the distribution, or, we do not know the number of change-point which is usually the case, so, the goal is to find the number of change-points and the exact positions of the changes.

### 1.2.2 Detecting changes with kernels

In order to find the best segmentation of a signal, we should define a sort of cost function over the set of segmentations.
We will start by defining a cost function that detects the change in the mean of a distribution which is the simplest change-point problem.
A suitable cost function should be able to measure the difference between each point and the mean of

the signal within each segment.
For a particular segmentation $\tau = [\tau_0, ..., \tau_{D_\tau}]$, the Least Square Criterion in $\mathcal{X} = \mathbb{R}^d$ is defined as :

$$\widehat{\mathcal{R}}_n(\tau) = \frac{1}{n}\sum_{\ell=1}^{D}\sum_{i=\tau_{\ell-1}+1}^{\tau_\ell}\left\|X_i - \bar{X}_{[\tau_{\ell-1}+1,\tau_\ell]}\right\|^2 \quad \text{where} \quad \bar{X}_{[\tau_{\ell-1}+1,\tau_\ell]} := \frac{1}{\tau_\ell - \tau_{\ell-1}}\sum_{j=\tau_{\ell-1}+1}^{\tau_\ell} X_j$$

If we develop the Least Square Criterion we get :

$$\begin{aligned}
\widehat{R}_n(\tau) &= \frac{1}{n}\sum_{\ell=1}^{D}\sum_{i=\tau_{\ell-1}+1}^{\tau_\ell}\left\|X_i - \frac{1}{\tau_\ell - \tau_{\ell-1}}\sum_{j=\tau_{\ell-1}+1}^{\tau_\ell} X_j\right\|^2 \\
&= \frac{1}{n}\sum_{\ell=1}^{D}\sum_{i=\tau_{\ell-1}+1}^{\tau_\ell}\left(\|X_i\|^2 - \frac{2}{\tau_\ell - \tau_{\ell-1}}\sum_{j=\tau_{\ell-1}+1}^{\tau_\ell}\langle X_i, X_j\rangle_{\mathbb{R}^d}\right. \\
&\left. + \frac{1}{(\tau_\ell - \tau_{\ell-1})^2}\sum_{j,j'=\tau_{\ell-1}+1}^{\tau_\ell}\langle X_j, X_{j'}\rangle_{\mathbb{R}^d}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\|X_i\|^2 - \frac{1}{n}\sum_{\ell=1}^{d}\frac{1}{\tau_\ell - \tau_{\ell-1}}\sum_{i,j=\tau_{\ell-1}+1}^{\tau_\ell}\langle X_i, X_j\rangle_{\mathbb{R}^d}
\end{aligned}$$

This means that

$$\widehat{R}_n(\tau) = \frac{1}{n}\sum_{i=1}^{n}k(X_i, X_i) - \frac{1}{n}\sum_{\ell=1}^{D}\frac{1}{\tau_\ell - \tau_{\ell-1}}\sum_{i,j=\tau_{\ell-1}+1}^{\tau_\ell}k(X_i, X_j)$$

where $k$ is the linear kernel in $\mathbb{R}^d$.

This criterion measures the adequacy of a segmetation $\tau$ in the case of changes in the mean.
To detect more types of changes, Harchaoui and Cappé [2007] were inspired by this criterion in $\mathbb{R}^d$ to define the **Kernel Least Square Criterion** for any kernel $k$ over any space $\mathcal{X}$ as long as the kernel is well defined in $\mathcal{X}$. It is defined exactely as above :

$$\widehat{R}_n(\tau) = \frac{1}{n}\sum_{i=1}^{n}k(X_i, X_i) - \frac{1}{n}\sum_{\ell=1}^{D}\frac{1}{\tau_\ell - \tau_{\ell-1}}\sum_{i,j=\tau_{\ell-1}+1}^{\tau_\ell}k(X_i, X_j) \tag{1.1}$$

Kernel change-point detection is the detection of change-points (i.e the best segmentation) using the Kernel Least Square Criterion.
The best segmentation then is the segmentation $\widehat{\tau}$ that minimizes $\widehat{R}_n(\tau)$.

$$\widehat{\tau} \in \text{argmin}_{\tau\in\mathcal{T}_n}\,\widehat{\mathcal{R}}_n(\tau)$$

We should highlight the fact that the minimization of this criterion over all possible segmentations will yield the trivial segmentation which is the segmentation of $n$ points, each segment then is a point

which can be seen as overfitting. $\widehat{R}_n(\tau) = 0$ in this case.
A general way to deal with overfitting is by introducing a regularization term or a penalty term.
Several penalty terms have been studied, among them is the linear penalty $\mathbf{pen_\ell(\tau) := \frac{CM^2 D_\tau}{n}}$ where $C$ and $M$ are constants.
Another penalty term was introduced by Celisse et al. [2012] which is based on model selection and theoretically built to produce risk bounds with high probabilty

$$\mathbf{pen(\tau) := \frac{1}{n}\left(c_1 \log\left(\begin{array}{c} n-1 \\ D_\tau - 1 \end{array}\right) + c_2 D_\tau\right)}$$

The main problem then becomes:

$$\widehat{\tau} \in \text{argmin}_{\tau \in \mathcal{T}_n} \left\{ \widehat{\mathcal{R}}_n(\tau) + \text{pen}(\tau) \right\}$$

There are several advantanges of the definition of the kernel least square criterion. The first one is the flexibility of the choice of the space which does not have to be a vector space. For example, kernel change-point detection can be applied to detect changes is the DNA sequence, a proper kernel can be defined in the space of DNA sequences $\mathcal{X} = \{A, T, C, G\}^{\mathbb{N}}$.
We can use this criterion to detect changes of the scene in videos for example.
Another example of the use of kernel change-point detection is Natural Language Processing. Several change-point detection techniques have been exploited to detect change in the meaning of words in time, **KCP** would work fine if a proper kernel was provided.

But the main advantage of kernel change-point detection is the ability of this method to detect changes in the distribution of the random variables no matter how complex and how imperceptible the changes are. It is due to the fact that some kernels are **Characteristic**, which we will get back to later.
For example, a change in the third moment of a distribution is not visible to the eye, a linear kernel is only able to detect changes in the mean of the distribution, so it will not work here.
It has been proven that a LS crierion with a Polynomial kernel of degree $p$ is able to detect changes in the first $p$ moments of a distribution.
Therefore, if somehow we know the nature of the change, we would choose a Polynomial kernel with a guarentee that it will work on this problem.
Forunately, several classic kernels used in Machine Learning fall into the class of Characteristic kernels (e.g. Gaussian, Laplace, etc), that means that even if we do not know the nature of the change, these types of kernels would detect the change-point. That is why the new kernel change-point detection method is considered a State of the Art method.

### 1.2.3 The reproducing Kernel Hilbert space

As stated before, using the least square criterion with a linear kernel can only detect changes in the mean of the distribution. That is why a new criterion was defined based on a kernel $k$.
In fact, using the kernel least square criterion in the input space $\mathcal{X}$ is equivalent to using the least square criterion with a linear kernel in a new feature space. So, instead of detection some changes of unknown nature in $\mathcal{X}$, we will try to detect changes in the mean of some new random variables in a new feature space. This feature space is the reproducing Kernel Hilbert space associated to $k$.

Let $\mathcal{H}$ be the RKHS associated to $k$ and $\Phi$ the feature map : $\Phi(x) = k(x, \cdot)$.
Let us denote $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ (resp. $\| \cdot \|_{\mathcal{H}}$ ) the inner product (resp. the norm) on $\mathcal{H}$.
$X_1, \cdots, X_n$ are the input variables which live in $\mathcal{X}$. Let us define the new random variables $Y_i = \Phi(X_i) = k(X_i, \cdot) \quad \forall i \in \{1, \cdots, n\}$.
$Y_i$ live in the RKHS $\mathcal{H}$, and by definition we have :

$$\langle Y_i, Y_j \rangle_{\mathcal{H}} = k(X_i, X_j) \quad \forall i, j \in \{1, \cdots, n\}$$

This non-linear transformation of random variables allow us to rewrite the kernel least square criterion in $\mathcal{H}$ as :

$$\widehat{\mathcal{R}}_n(\tau) = \frac{1}{n} \sum_{\ell=1}^{D} \sum_{i=\tau_{\ell-1}+1}^{\tau_\ell} \left\| Y_i - \bar{Y}_{[\tau_{\ell-1}+1, \tau_\ell]} \right\|_{\mathcal{H}}^2 \quad \text{where} \quad \bar{Y}_{[\tau_{\ell-1}+1, \tau_\ell]} := \frac{1}{\tau_\ell - \tau_{\ell-1}} \sum_{j=\tau_{\ell-1}+1}^{\tau_\ell} Y_j$$

So, as we said earlier, the true segmentation of the signal in $\mathcal{X}$ is the segmentation where the mean of the new variables $Y_i$ is constant in each segment. We should now define the mean of an element in the RKHS $\mathcal{H}$.

Under the assumption that $\mathcal{H}$ is separable (contains a dense countable subset) and that $\forall i \in \{1, \ldots, n\}, \quad \mathbb{E}[\sqrt{k(X_i, X_i)}] < +\infty$, we can define the mean of the variable $Y_i$ as :

$$\mu_i^\star := \mathbb{E}[Y_i] = \mathbb{E}[k(X_i, \cdot)] \in \mathcal{H}$$

.
So, if $\tau^\star := [\tau_0^\star, ..., \tau_D^\star]$ is the true segmentation, then :

$$\mu_1^\star = \cdots = \mu_{\tau_1^*}^\star, \quad \mu_{\tau_1^*+1}^\star = \cdots = \mu_{\tau_2^*}^\star, \quad \cdots \quad \mu_{\tau_{D^*-1}^*+1}^\star = \cdots = \mu_n^\star$$

and $\quad \forall i \in \{1, \ldots, D^\star - 1\}, \quad \mu_{\tau_i}^\star \neq \mu_{\tau_{i+1}}^\star$.
But, what guarentees the fact that a change in the mean in the feature space translates the change in the distribution in the input space?

This totally depends on the kernel. There is a class of kernels that can guarentee that, it's **Characteristic kernels**.
A characteristic kernel is a kernel that guarentees the injectivity of the application $P \mapsto \mathbb{E}_{X \sim P}[\Phi(X)]$, that means that two different means $\mathbb{E}_{X \sim P}[\Phi(X)]$ and $\mathbb{E}_{X \sim P'}[\Phi(X)]$ define two different distributions $P$ and $P'$. The Gaussian kernel for example is a characterisic kernel.
This definition is similar to the definition of the characteristic function where the characteristic function also defines the law or the distribution.

All of this being said, $\mathcal{H}$ is a quite complex space of functions, so, we never work in the feature space thanks to the kernel trick. In practice, we work directly with the kernel $k$ and the input variables $X_1, \cdots, X_n$ through the expression of the kernel least square criterion. But all the theoretical results either in the literature or in this work are based on the properties of the variables in the RKHS feature space.

### 1.2.4 Two KCP algorithms

In this section, we will present two kernel-change detection algorithms. The first one is an exact algorithm that solves the minimization problem of KCP, it is based on dynamic programming.
The second algorithm is an approximative algorithm, it is the Binary Segmentation algorithm, it will be very useful in the context of scaling up the KCP method with the two main approximation methods.

#### 1.2.4.1 The exact KCP algorithm

We defined the best segmentation previously as

$$\widehat{\tau} \in \text{argmin}_{\tau \in \mathcal{T}_n} \left\{ \widehat{\mathcal{R}}_n(\tau) + \text{pen}(\tau) \right\}$$

$\widehat{\mathcal{R}}_n(\tau)$ is defined using the evaluation of the kernel in each pair on points $X_i, X_j$. So, the calculation of the least square criterion requires the storage of the Gram matrix associated to the kernel $G = [k(X_i, X_j)]_{i,j \leq n}$.

In order to find the best exact segmentation, we devide the problem into two sub-problems. The fist problem is to find the best segmentation of $D$ segments where $D$ is a fixed predefined number of segments. In this sub-problem, we will solve the minimization problem $\widehat{\tau}_D \in \text{argmin}_{\tau \in \mathcal{T}_n^D} \left\{ \widehat{\mathcal{R}}_n(\tau) \right\}$ with dynamic programming. Because we fixed the number of segments, there is no need for a penalty term.
The second sub-problem is to find the best segmentation, for this we will fixe a maximum number of segments $D_{max}$ and we will store all the $\widehat{\tau}_D$ segmentations found by solving the first sub-problem for $D \in \{1, \cdots, D_{max}\}$.
The exact segmentation then is the solution of the minimization problem

$$D^\star = \text{argmin}_{D \leq D_{max}} \left\{ \widehat{\mathcal{R}}_n(\widehat{\tau}_D) + \text{pen}(\widehat{\tau}_D) \right\} \quad \textit{and} \quad \tau^\star = \widehat{\tau}_{D^\star}$$

Let us solve the first sub-problem.
First of all, we should define the cost of a segment $[a, b[$.

$$C(a, b) = \sum_{i=a+1}^{b} k(X_i, X_i) - \frac{1}{b-a} \sum_{i=a+1}^{b} \sum_{j=a+1}^{b} k(X_i, X_j)$$

Therefore, for a segmentation $\tau_D$ of $D$ segments we have :

$$\widehat{\mathcal{R}}_n(\tau_D) = \frac{1}{n} \sum_{l=1}^{D} C(\tau_{\ell-1}, \tau_\ell)$$

We also define the optimal cost of segmenting the signal up to time $b$ in $D$ segments as :

$$\text{Cost}(D, b) = \min_{\tau \in \mathcal{T}_b^D} \left\{ n \widehat{\mathcal{R}}_n(\tau) \right\} = \min_{\tau \in \mathcal{T}_b^D} \left\{ \sum_{l=1}^{D} C(\tau_{\ell-1}, \tau_\ell) \right\}$$

The goal of KCP is to find the quantity $\frac{1}{n}\operatorname{Cost}(D,n)$ (or $\operatorname{Cost}(D,n)$).
The motivation behind using dynamic programming to solve this problem is the fact that the optimal cost, the way it is defined is additive, that means that :

$$\begin{aligned}\operatorname{Cost}(D,b) &= \min_{\tau\in\mathcal{T}_b^D}\left\{\sum_{l=1}^{D-1} C(\tau_{\ell-1},\tau_\ell)+C(\tau_{D-1},b)\right\}\\ &= \min_{a\leq b}\left\{\operatorname{Cost}(D-1,a)+C(a,b)\right\}\end{aligned}$$

which is a dynamic programming scheme.

In order to solve this problem, we only need the values $C(a,b)$ for $a,b\in\{1,\cdots,n\}$, so, we will store these values of the cost in a matrix $S$ of size $n\times n$.
Let us define $\Gamma$ the cumulative sum and $T$ the cumulative trace of the Gram matrix $G$, i.e.

$$\forall 1\leq a<b\leq n,\quad \Gamma_{a,b}:=\sum_{i=1}^{a}\sum_{j=1}^{b}k\left(X_i,X_j\right)\qquad\text{and}\qquad T_a:=\sum_{i=1}^{a}k\left(X_i,X_i\right)$$

We then have :

$$C(a,b)=T_b-T_a-\frac{1}{b-a}\left(\Gamma_{b,b}-2\Gamma_{a,b}+\Gamma_{a,a}\right)\tag{1.2}$$

In order to optimize the computation of the matrix $\Gamma$ and the column $T$, it will be done at the same time as the computation of the Gram matrix $G$. Which mean, the computation of these three tables is going to be in $\mathcal{O}\left(n^2\right)$ instead of $\mathcal{O}\left(n^4\right)$ (which is the time of the computation of $\Gamma$ if done naively for each element).

Based on Equation 1.2, we can program a simle function that takes in argument the Gram matrix $G$ and returns the matrix of costs $S$, let's call it ***SegmentCosts***.

This is a pseudo-code describing the dynamic programming procedure to find the best segmentation without the penalty term

**Algorithm 1:** KCP with dynamic programming

**Result:** The best segmentation $\widehat{\boldsymbol{\tau}}$
**Input** : The Gram matrix $\boldsymbol{G}$, The number of segments $\boldsymbol{D}$
// Block1 : Initializations
$\boldsymbol{S} = \boldsymbol{SegmentCosts}(G)$
$\boldsymbol{Costs} = \boldsymbol{zeros(D, n)}$
$\boldsymbol{Indices} = \boldsymbol{zeros(D, n)}$
$\widehat{\boldsymbol{\tau}} = \boldsymbol{zeros(D)}$
// Block2 : Computing the optimal costs
**for** $\boldsymbol{b \leftarrow 1 : n}$ **do**
  $Costs(1, b) \leftarrow S(0, b)$
**end**
**for** $\boldsymbol{d \leftarrow 2 : D}$ **do**
  **for** $\boldsymbol{b \leftarrow d : n}$ **do**
    $Costs(d, b) \leftarrow \min_{a \leq b} \{Costs(d-1, a) + S(a, b)\}$
    $Indices(d, b) \leftarrow \text{argmin}_{a \leq b} \{Costs(d-1, a) + S(a, b)\}$
  **end**
**end**
// Block3 : Backtracking
$\boldsymbol{p \leftarrow n}$
**for** $\boldsymbol{d \leftarrow 1 : D}$ **do**
  $\widehat{\tau}[d] \leftarrow Indices(D - d + 1, p)$
  $p \leftarrow \widehat{\tau}[d]$
**end**

If we add the penalty term, the algorithm does not change that much. Instead of working with $D$ we will work with $D_{max}$. The other major change is that in the last columnof the matrix $\boldsymbol{Costs}$ we should add the penalty terms to each corresponding element : $Costs(d, n) \leftarrow Costs(d, n) + \text{Penalty}(d) \quad \forall d \in \{1, \cdots, D_{max}\}$.
The $D$ in the Backtracking block would be in this case $\widehat{D} = \text{argmin}_{d \leq D_{max}} \{Costs(d, n)\}$

**The Main problem :**

As we can see from the presented pseudo-code, finding the best segmentation take at least $\mathcal{O}(D_{max}n^2)$ in time and $\mathcal{O}(n^2)$ is space.
In fact, we need $\mathcal{O}(n^2)$ to store the Gram matrix and $\mathcal{O}(n^2)$ to store the matrix of costs $S$. These are the main quantities that consume storage.

For the time complexity, the computation of Gram matrix and the matrix of costs is also done $\mathcal{O}(n^2)$, but it is not the main concern here. The computation of the $D \times n$ matrix of optimal costs in the second block algorithm takes $\mathcal{O}(n^2)$ for each value of $d$, that means a total time complexity of $\mathcal{O}(D_{max}n^2)$.

In fact, the two for loops in the second block cost

$$\sum_{d=2}^{D_{max}} \sum_{b=d}^{n} b = \mathcal{O}(D_{max} n^2)$$

A dynamic programmig based algorithm is generally solved in quadratic time complexity which is the case here.

This was a general presentation of the exact KCP algorithm.
Celisse et al. [2017] tried to improve the storage and time complexity of this procedure. They manage to reduce the space complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(D_{max} n)$ by working with dynamic columns instead of $n \times n$ matrices. We refer to their paper for more details.
The time complexity is always $\mathcal{O}(D_{max} n^2)$ because, as we said, it is due to dynamic programming which appears to be the best and optimal way to find the best segmentation.

To sum up, the best implementation of the exact KCP algorithm costs $\mathcal{O}(D_{max} n^2)$ in time and $\mathcal{O}(D_{max} n)$ in space.

With these kind of performances, we should be able to process signals up to $n = 10^5$. But once we want to go higher, the quadratic time complexity becomes a severe limitation for Kernel change-point detection.
This is the motivation behind our work. Using low rank approximations, we should be able to work with signals where $n \geq 10^5$, but unfortunately, even with these approximations, we can not scale up the exact KCP algorithm because the main source of the quadratic complexity is the dynamic programming and it is not related to the kernel or the Gram matrix. Since we can not proceed differently to recover the exact segmentation, we will work instead with an approximative algorithm called **Binary Segmentation**.
In the following section, we will present the algorithm and analyse its time complexity.
Then we will present the two main low rank approximation methods and how they can be applied to scale up the Binary segmentation algorithm.

#### 1.2.4.2 The Binary Segmentation algorithm

Binary segmentation is a standard heuristic used to solve the problem of multiple change-point detection. It consist on iteratively computing a new $\tau_{D+1}$ segmentation from an old $\tau_D$ by spliting one segment of $\tau_D$ into two new segments. This segment is found and splitted using a certain criterion.
For example, a famous Binary segmentation algorithm was introduced in 2014 by Fryzlewicz et al. [2014]. It is called the **Wild Binary Segmentation algorithm** and instead of using a least square criterion for example as in KCP, they use another criterion which is the **CUSUM Statistic** to find the segment to be splitted and the change-point in that segment.
This algorithm is not a subject of intrest in this work, we will work with the standard binary segmentation algorithm along with the kernel least square criterion.

The principle is simple, in each iteration from 1 to $D_{max}$, we look for the best segment to be splitted and within that segment we look for the best change-point and we split the segment into two

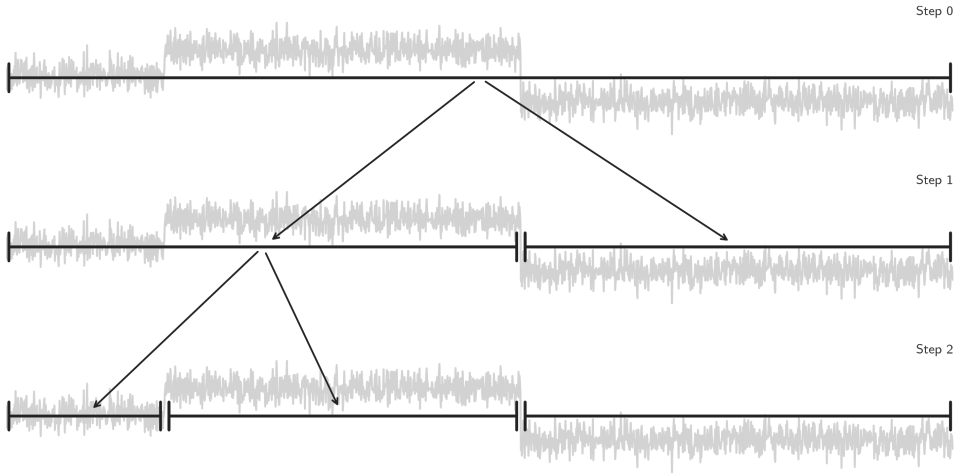



Figure 1.1: Binary segmentation schema.
**Source** : Truong et al. [2019]

new segments.
We provide the pseudo-code for this algorithm below.

**Algorithm 2:** The Binary segmentation

**Result:** The best segmentation of $[\mathbf{1}, \boldsymbol{n}]$
**Input** : The Gram matrix $\boldsymbol{G}$, $\boldsymbol{D_{max}}$
// Block1 : Initializations
$\boldsymbol{S} = \boldsymbol{SegmentCosts}(G)$
$\boldsymbol{Segments} \leftarrow \{[\mathbf{1}, \boldsymbol{n}]\}$
$\boldsymbol{Change\_points} \leftarrow \{\}$
$\boldsymbol{CandidateSplit} \leftarrow \{\}$
// Block2 : Binary segmentation
**for** $\boldsymbol{d} \leftarrow \mathbf{1} : \boldsymbol{D_{max}}$ **do**
  **for** $\boldsymbol{seg} \in \boldsymbol{Segments}$ **do**
    // Recovering the best segmentation of the current segment
    $[a, b[ = seg$
    $m = \min_{a \leq c \leq b} \{S(a, c) + S(c, b)\}$
    $\hat{c} = \text{argmin}_{a \leq c \leq b} \{S(a, c) + S(c, b)\}$
    $r = S(a, b) - m$ // The reduction in cost
    **insert** $\{r, \hat{c}, [a, \hat{c}[, [\hat{c}, b[\}$ **in** $CandidateSplit$
  **end**
  Extract the best split of $CandidateSplit$ and recover $\hat{c}$, $[a, \hat{c}[$ and $[\hat{c}, b[$
  **insert** $\hat{c}$ **in** $Change\_points$
  $Segments \leftarrow \{[a, \hat{c}[, [\hat{c}, b[\}$
**end**

**CandidateSplit** is a set of tuples, each tuple contains four informations : The change-point, the reduction in cost and the two segments.
So, in each iteration over a segment, two tuples are inserted in **CandidateSplit**. And the best *CandidateSplit* of the two splits is extracted according to the reduction in cost of each segmentation. The best candidate is the one that guarentees the bigger reduction in cost.
**Segments** is then updated with the new best two segments. The same procedure is iterated until we reach $D_{max}$ change-points.

The set **Segments** has always two segments in memory at best, so the for loop over the segments does not add much to the total cost of the algorithm.
The main contributor to the total time complexity of the Binary segmentation algorithm is the computation of $m = \min_{a \leq c \leq b} \{S(a,c) + S(c,b)\}$.
If we want to program a fuction that computes this minimum naively, we would end up computing all the segment costs $C(a,c)$ and $C(c,b)$, $b-a$ times, which will yield to a time complexity of $\mathcal{O}((b-a)^3)$. This means that the Binrary segmentation algorithm will have a time complexity of $\mathcal{O}(D_{max}n^3)$ which is way worse than the dynamic programming time complexity.

Fortunately for us, there is a way to reduce the computation cost of the minumum $m = \min_{a \leq c \leq b} \{C(a,c) + C(c,b)\}$ to $\mathcal{O}((b-a)^2)$.

As a quick reminder, $C(a,b)$ is the segmentation cost of the segment $[a,b[$ and $S(a,b)$ is the element of indices $a,b$ in the matrix $S$. So they are exactely the same quantities.

The trick for this reduction in time complexity is presented in Celisse et al. [2017].
We observed that the cost of a segment $[a,c[$ can be written as follows

$$C(a,c) = \sum_{i=a}^{c-1} \left( k\left(X_i, X_i\right) - \frac{A_{i,c}}{c-a} \right) = D_{a,c} - \frac{1}{c-a} \sum_{i=a}^{c-1} A_{i,c}$$

where $D_{a,c} = \sum_{i=a}^{c-1} k\left(X_i, X_i\right)$ and $A_{i,c} = -k\left(X_i, X_i\right) + 2\sum_{j=i}^{c-1} k\left(X_i, X_j\right)$
Both $D_{a,c}$ and $\{A_{i,c}\}_{i \leq c}$ can be iteratively computed from $c$ to $c+1$ by use of the two following equations:

$$D_{a,c+1} = D_{a,c} + k\left(X_c, X_c\right), \quad \text{and} \quad A_{i,c+1} = A_{i,c} + 2k\left(X_c, X_c\right), \forall i \leq c$$

with $A_{c+1,c+1} = -k\left(X_{c+1}, X_{c+1}\right)$.
Therefore, as long as computing $k\left(X_i, X_j\right)$ requires $\mathcal{O}(1)$ operations, updating from $c$ to $c+1$ requires $\mathcal{O}\left(c\right)$ operations.

This update rule will give rise to the algorithm below :

**Algorithm 3:** BestSplit

**Result:** The best split of a segment $[\boldsymbol{a}, \boldsymbol{b}[$
**Input** : The Gram matrix $\boldsymbol{K}$, The segment $[\boldsymbol{a}, \boldsymbol{b}[$
$\boldsymbol{m} = +\boldsymbol{\infty}$
$\boldsymbol{change\_point} = \boldsymbol{a}$
**for** $\boldsymbol{c} \leftarrow \boldsymbol{a} : \boldsymbol{b}$ **do**
  Update $C(a, c)$ // The update is done in $\mathcal{O}(c)$
  Update $C(c, b)$ // We can apply a similar update rule here
  $\hat{m} = C(a, c) + C(c, b)$
  **if** $\hat{m} \leq m$ **then**
    $m = m$
    $change\_point = c$
  **end**
**end**

This function return the best split of a segment $[a, b[$ in $\mathcal{O}\left((b-a)^2\right)$.

$$\sum_{c=a}^{b} \mathcal{O}(c) = \mathcal{O}\left((b-a)^2\right)$$

In conclusion, using this procedure in the Binary segmentation algorithm, we will have a total time complexity of $\boldsymbol{\mathcal{O}\left(D_{max} n^2\right)}$ in the worst case scenario.
This is exactely the same complexity of the exact KCP algorithm, so there is no intrest in working with this algorithm in this form.
The particularity of this algorithm is in its ability to be scaled up using a kernel approximation method such as Nystrom approximation or Random features.

We can see clearly that if we managed to reduce the cost of the update from $c$ to $c+1$ in Algorithm 3 to a constant cost $\mathcal{O}(1)$, the algorithm will cost $\mathcal{O}(b-a)$ which yield to $\mathcal{O}(D_{max} n)$ for the Binary segmentation algorithm.
In the next chapter, we will see how can we enhance the Binary segmentation algorithm from a quadratic cost to a linear cost at the expense of an approximation.

# Chapter 2

# Low Rank Approximation methods

## 2.1 Nystrom Approximation

### 2.1.1 General Presentation

The first approximation method we will see is Nystrom approximation.
The goal is to approximate the Gram matrix with a low rank matrix that is computed in significantely less time.
In this section we will present the method, analyse the different approximation errors, apply this method to **KCP** and evaluate the performance.

Given a kernel $k$, the goal is to build the Gram matrix $\mathbf{G} = [k(X_i, X_j)]_{1 \leq i,j \leq n}$ in less time and store it in less memory space.
The targeted approximation is of the form $\tilde{G}_k = CW_k^{\dagger}C^{\top}$, where $C$ is a matrix consisting of a small number $p$ of columns of $G$ sampled randomly and $W_k$ is the best rank-k approximation to $W$, the matrix formed by the intersection between those $p$ columns of $G$ and the corresponding $p$ rows of $G$.

Given a Gram matrix $\mathbf{G}$, the first thing we should fix is $p$ the number of columns to be sampled from $\mathbf{G}$, so, in order to gain in time and space we should choose $p \ll n$.
The second parameter to fix is the rank $k \leq p$.

We then sample randomly $p$ index from $\{1, ..., n\}$ let's call $\mathcal{I}$ the subset of indices. Different samplimg techniques have been studied in order to guarentee optimal results, but generally the indices are sampled uniformely.
$W = \mathbf{G}_{\mathcal{I},\mathcal{I}} \in \mathbb{R}^{p \times p}$ is the submatrix of elements which indices are in $\mathcal{I}$ and $C \in \mathbb{R}^{n \times p}$ is the submatrix of columns which indices are in $\mathcal{I}$.

The approximation of $\mathbf{G}$ is then

$$\tilde{G}_k = CW_k^{\dagger}C^T \in \mathbb{R}^{n \times n}$$

where $W_k^{\dagger}$ is the pseudo inverse of the rank-k approximation of $W$. This approximation is usually computed using Singular Values Decomposition (SVD).

The way we performed the approximation is not of intrest because we started already from a Gram matrix $\in \mathbb{R}^{n\times n}$ which is what we want to avoid.
A more effecient way is to :

1. Sample a set $\mathcal{I}$ of $p$ indices randomly in $\{1, ..., n\}$
2. Compute $C \in \mathbb{R}^{n\times p}$ with $C_{ij} = k(X_i, X_j)$ for $i \in \{1, \dots, n\}$ and $j \in \mathcal{I}$
3. Form the matrix $W \in \mathbb{R}^{p\times p}$ with $W_{ij} = k(X_i, X_j)$ for $i, j \in \mathcal{I}$
4. Compute $W_k \in \mathbb{R}^{p\times p}$, the best rank-k approximation of $W (k \leq p)$
5. Compute rank $k$ approximation of $G$ :

$$\tilde{G}_k = CW_k^{\dagger}C^T$$

So, up to Step 4, we only store $np + p^2$ number which is way smaller than $n^2$.

The approximation of the matrix itself is not very interesting for us in the kernel change-point detection context, we should be able to approximate the feature map $\Phi$. A Nystrom approximation of $\Phi$ is of the form :

$$\hat{\Phi}(x) = K_x^{\mathcal{I}} U_k \Sigma_k^{-1/2} \tag{2.1}$$

where $K_x^{\mathcal{I}} = [k(x, X_i)]_{i\in\mathcal{I}} \in \mathbb{R}^{1\times p}$ and $W_k = U_k\Sigma_k U_k^T$ is the SVD of $W_k$.

So, to compute our new features $\hat{\Phi}(X_1), ..., \hat{\Phi}(X_n)$ it will cost $\mathcal{O}(p^3)$ to compute the SVD and $\mathcal{O}(p^2)$ for the matrix multiplication for each $\hat{\Phi}(X_i)$. That yields in total $\mathcal{O}(p^3 + np^2)$.

### 2.1.2 Theoretical aspects of Nystrom Approximation

We will now present some theoretical guarentees of the Nystrom approximation.

In Drineas and Mahoney [2005], the autors claim that sampling columns according to the uniform distribution does not guarentee optimal results. They propose another way of sampling along with a theorem to back their claim.

Let $\mathbf{G}$ be our Gram matrix. According to Drineas and Mahoney [2005], a data dependent probability distribution would perform better in Nystrom approximation. Instead of sampling w.r.t $(p_1, ..., p_n) = (\frac{1}{n}, ..., \frac{1}{n})$ which is our uniform distribution, they propose a data dependent distribution where

$$\forall 1 \leq i \leq n : \quad p_i = \frac{\mathbf{G}_{ii}^2}{\sum_{j=1}^n \mathbf{G}_{jj}^2}$$

Because the probabilities now are different, we need to rescale the sampled columns by some rescaling factors detailed Drineas and Mahoney [2005] main algorithm.

The thoerem that goes along with this new method controls the norms of the difference between the input matrix $\mathbf{G}$ and the approximated low rank matrix $\tilde{G}_k$ w.r.t the difference betweeen $\mathbf{G}$ and its best rank $k$ approximation $G_k$.

**Theorem 2** *Suppose* $\mathbf{G}$ *is an* $n \times n$ *SPSD matrix, let* $k \leq p$ *be a rank parameter, and let* $\tilde{G}_k = CW_k^{\dagger}C^T$ *be constructed by sampling columns of* $\mathbf{G}$ *with probabilities* $\{p_i\}_{i=1}^n$ *such that*

$$p_i = \frac{G_{ii}^2}{\sum_{i=1}^n G_{ii}^2}$$

*Let* $r = rank(W)$ *and let* $G_k$ *be the best rank-k approximation* $\mathbf{G}$*. In addition, let* $\varepsilon > 0$ *and* $\eta = 1 + \sqrt{8\log(\frac{1}{\delta})}\cdot$ *If* $p \geq \frac{64k\eta^2}{\varepsilon^4}$ *then with probability at least* $1 - \delta$

$$\left\|G - \tilde{G}_k\right\|_F \leq \|G - G_k\|_F + \varepsilon \sum_{i=1}^n G_{ii}^2$$

*In addition, if* $p \geq \frac{4\eta^2}{\varepsilon^2}$ *then with probability at least* $1 - \delta$

$$\left\|G - \tilde{G}_k\right\|_2 \leq \|G - G_k\|_2 + \varepsilon \sum_{i=1}^n G_{ii}^2$$

$\|\|_2$, $\|\|_F$ stand for the spectral norm and Frobenius norm of a matrix, respectively.

According to the theorem , if we fix $p$ the number of columns to be sampled, then for the Frobenius norm, the smallest $\varepsilon$ that satisfies the theorem's conditions is $\varepsilon = (\frac{64k\eta^2}{p})^{\frac{1}{4}}$, so with high probability:

$$\left\|G - \tilde{G}_k\right\|_F \leq \|G - G_k\|_F + \mathcal{O}\left(\frac{n}{\sqrt[4]{p}}\right)$$

and with the same reasoning :

$$\left\|G - \tilde{G}_k\right\|_2 \leq \|G - G_k\|_2 + \mathcal{O}\left(\frac{n}{\sqrt{p}}\right)$$

Clearly, the bound measured in Frobenius norm is significantly worse in terms of $p$, with a convergence rate of $\mathcal{O}\left(\frac{n}{\sqrt[4]{p}}\right)$.
Mahdavi et al. [2012] raised the question weather there is a way to improve this bound to at least $\mathcal{O}\left(\frac{n}{\sqrt{p}}\right)$. The answer is yes but under some assumptions on the eigengaps in the spectrum of Gram matrix.

So, in the best case scenario, with high probability we have

$$\left\|G - \tilde{G}_k\right\|_F \leq \|G - G_k\|_F + \mathcal{O}\left(\frac{n}{\sqrt{p}}\right)$$

A similar theorem to that of Drineas and Mahoney [2005] is presented by Kumar et al. [2009] in the case of uniform sampling without replacement.

**Theorem 3** *Let $\mathbf{G} \in \mathbb{R}^{n\times n}$ be an SPSD Gram matrix. Assume that p columns of $\mathbf{G}$ are sampled uniformly at random without replacement, let $\tilde{G}_k$ be the rank-k Nyström approximation to $\mathbf{G}$ as described previously and let $G_k$ be the best rank-k approximation to $\mathbf{G}$.*
*For $\varepsilon > 0$, if $p \geq \frac{64k}{\varepsilon^4}$, then*

$$\mathrm{E}\left[\left\|G - \tilde{G}_k\right\|_F\right] \leq \|G - G_k\|_F + \varepsilon \left[\left(\frac{n}{l}\sum_{i\in D(p)} G_{ii}\right)\sqrt{n\sum_{i=1}^{n} G_{ii}^2}\right]^{\frac{1}{2}}$$

*where $\sum_{i\in D(p)} G_{ii}$ is the sum of the largest p diagonal entries of $\mathbf{G}$.*
*Further, if $\eta = \sqrt{\frac{\log(\frac{2}{\delta})\alpha(p,n-p)}{p}}$, with $\alpha(p, n-p) := \frac{p(n-p)}{n-1/2}\cdot\frac{1}{1-1/(2\max\{p,n-p\})}$ and if $p \geq \frac{64k}{\varepsilon^4}$ then with probability at least $1-\delta$*

$$\left\|G - \tilde{G}_k\right\|_F \leq \|G - G_k\|_F + \varepsilon \left[\left(\frac{n}{l}\sum_{i\in D(p)} G_{ii}\right)\left(\sqrt{n\sum_{i=1}^{n} G_{ii}^2} + \eta\max\left(nG_{ii}\right)\right)\right]^{\frac{1}{2}}$$

With high probability, this theorem also yields :

$$\left\|G - \tilde{G}_k\right\|_F \leq \|G - G_k\|_F + \mathcal{O}\left(\frac{n}{\sqrt[4]{p}}\right)$$

The fact that this bound is so similar to that of Drineas and Mahoney [2005] raises quastions weather their sampling method provides really better results or not.
Kumar et al. [2009] have proven empirically that the uniform sampling technique is the best technique that guarentees the best relative accuracy $\frac{\|G-G_k\|_F}{\left\|G-\tilde{G}_k\right\|_F}$.
Another drawback of the data dependent sampling method is that the probability weights are computed in $\mathcal{O}(n)$ which can be restricting when working with very large datasets.
In the literature in general, Nystrom approximations are associated with uniform sampling.

The Nystrom approximation has two sources of errors, the approximation of $\mathbf{G}$ with a rank-k matrix and the approximation via the column sampling.
If we are lucky to choose $k \geq rank(G)$, then $\|G - G_k\|_F = 0$. So, in the best case scenario (Under the eigengaps assumptions), $\left\|G - \tilde{G}_k\right\|_F \leq \mathcal{O}\left(\frac{n}{\sqrt{p}}\right)$ and in the worse case scenarion : $\left\|G - \tilde{G}_k\right\|_F \leq \mathcal{O}\left(\frac{n}{\sqrt[4]{p}}\right)$.

## 2.2 Random Fourier Features

### 2.2.1 General Presentation

The second approximation method we will investigate is the **Random Fourier Features**. It is a very popular method used to accelerate the training of almost every Kernel method in Machine Learning. The goal is to propose a map function that maps the input data to a randomized low-dimensional feature space and then apply existing fast linear methods.

Rahimi and Recht [2008] studied Random Features and provided theoretical guarentees of the approximation method, they claim that in large-scale classification and regression tasks, linear machine learning algorithms applied to these features outperform state-of-the-art large-scale kernel machines.

Rudi and Rosasco [2017] also studied this approximation method in the context of Kernel Ridge Regression. They proved that we can guarentee optimal statistical performance using only few number of features ($\mathcal{O}\left(\sqrt{n}\log n\right)$ instead of $\mathcal{O}\left(n^2\right)$) which can significantely accelerate the Kernel Ridge Regression.

We will now present the Random Fourier Features approximation.

Given a positive definite and shift-invariante kernel $k : \mathcal{X} \to \mathbb{R}$, we can prove that the kernel has an integral representation in the form of :

$$k\left(x, x'\right) = K\left(x - x'\right) = \int_{\Omega} \psi(x, \omega)\psi\left(x', \omega\right) d\pi(\omega), \quad \forall x, x' \in \mathcal{X} \tag{2.2}$$

where $(\Omega, \pi)$ is probability space and $\psi : X \times \Omega \to \mathbb{R}$.

This is a generalization of Bochner's theorem :

**Theorem 4** ***(Bochner)*** *A continuous shift-invariant kernel $k\left(x, x'\right) = K(\Delta)$ on $\mathbb{R}^d$ is positive definite if and only if $K(\Delta)$ is the Fourier transform of a nonnegative probability measure. In particular, if $k$ is properly scaled, we have:*

$$K\left(x - x'\right) = \int_{\mathbf{R}^d} P(\omega) e^{i\omega^T x} e^{-i\omega^\top x'} d\omega$$

*where $P(\omega)$ is a real-valued probability density function over $\mathbb{R}^d$.*

By developing the result of Bochner's theorem we have:

$$\begin{aligned}
K\left(x-x'\right) &= \int_{\mathbb{R}^p} P(\omega) e^{i\omega^\top X} e^{-i\omega^\top x'} d\omega \\
&= \int_{\mathbb{R}^p} P(\omega) \cos\left(\omega^\top x - \omega^\top x'\right) d\omega \\
&= \int_{\mathbb{R}^p} P(\omega) \left(\cos\left(\omega^\top x\right)\cos\left(\omega^\top x'\right) + \sin\left(\omega^\top x\right)\sin\left(\omega^\top x'\right)\right) d\omega \\
&= \int_{\mathbb{R}^p}\int_{b=0}^{2\pi} \frac{P(\omega)}{2\pi} 2\cos\left(\omega^\top x + b\right)\cos\left(\omega^\top x' + b\right) d\omega db \\
&= \mathbb{E}_{\omega\sim P, b\sim\mathcal{U}(0,2\pi)}\left[\sqrt{2}\cos\left(\omega^\top x + b\right)\sqrt{2}\cos\left(\omega^\top x' + b\right)\right]
\end{aligned}$$

$K\left(x-x'\right), P(\omega) \in \mathbb{R}$ So we can ignore imaginary part in line 2.

That means that $\psi\left(x,\tilde{\omega}\right) = \sqrt{2}\cos\left(\omega^T x + b\right)$ where $\tilde{\omega} = (\omega, b) \sim P, \mathcal{U}(0, 2\pi)$

If $\mathcal{X} = \mathbb{R}^d$ then P is given by the scaled Fourier transform of $K(\Delta)$

$$P(\omega) = \frac{1}{(2\pi)^d}\int_{\mathbb{R}^d} K(\Delta) e^{-i\omega^\top \Delta} d\Delta$$

For reasons of generality, we will work under the abstract and general formulation of the approximation presented in 2.2.
This representation provides an approximation of the kernel $k$ with $p$ features in the form of :

$$k\left(x,x'\right) = \mathbb{E}_{\omega\sim\pi}\left[\psi(x,\omega)\psi(x',\omega)\right] \approx \Phi_p(x)^\top \Phi_p\left(x'\right) \tag{2.3}$$

where $\Phi_p(x) = \frac{1}{\sqrt{p}}(\psi(x,\omega_1), ..., \psi(x,\omega_p))$ with $\omega_1, ..., \omega_p$ sampled independetly with respect to $\pi$. $\boldsymbol{\Phi_p : \mathcal{X} \to \mathbb{R}^p}$ is the feature map that maps the input data $x$ to new points in $\mathbb{R}^p$.

### 2.2.2 Theoretical aspects of Random Fourier Features

The first theoretical guarentee is presented in Rahimi and Recht [2008]. It is a direct application of Hoeffding's concentration inequality.
Since

$$\Phi_p(x)^\top \Phi_p\left(x'\right) = \frac{1}{p}\sum_{i=1}^{p} \psi(x,\omega_i)\psi(x',\omega_i)$$

and $\mathbb{E}_{\omega\sim\pi}\left[\psi(x,\omega)\psi(x',\omega)\right] = k\left(x,x'\right) \quad \forall i \in \{1, ..., p\}$, we have :

$$\mathbb{P}\left[\left|\Phi_p(x)^\top \Phi_p\left(x'\right) - k\left(x,x'\right)\right| \geq \epsilon\right] \leq 2\exp\left(\frac{-p\epsilon^2}{2}\right) \tag{2.4}$$

That means that with probability higher than $1-\delta$ :

$$\left|\Phi_p(x)^T \Phi_p\left(x'\right) - k\left(x,x'\right)\right| \leq \left(\frac{2\log(\frac{2}{\delta})}{p}\right)^{\frac{1}{2}}$$

Rahimi and Recht [2008] tried to give a better result which they did but will not be exploited in this work :

**Theorem 5** ***(Uniform convergence of Fourier features)***. *Let $\mathcal{M}$ be a compact subset of $\mathbb{R}^d$ with diameter diam($\mathcal{M}$). Then, for the mapping $\Phi_p$ defined in Equation 2.3, we have*

$$\mathbb{P}\left[\sup_{x,y\in\mathcal{M}}\left|\Phi_p(\mathbf{x})^T\Phi_p(\mathbf{y}) - k(\mathbf{x},\mathbf{y})\right| \geq \epsilon\right] \leq 2^8\left(\frac{\sigma\,\mathrm{diam}(\mathcal{M})}{\epsilon}\right)^2 \exp\left(-\frac{D\epsilon^2}{4(d+2)}\right)$$

*where $\sigma^2 \equiv \mathbb{E}_{\omega\sim\pi}\left[\omega^T\omega\right]$ is the second moment of the Fourier transform of $K$.*

Overall, we can say that the results of Rahimi and Recht [2008] guarentee an approximation error decreasing with $\mathcal{O}\left(\frac{1}{\sqrt{p}}\right)$.

A particular paper of Sutherland and Schneider [2015] studies different error bounds of Random features in several Machine learning and Statstics problems. We have been inspired from these results to prove a theorem that guarentees the convergence of the kernel change-point detection problem with Random features. This theorem will be presented in the next chapter.

## 2.3 Applications to KCP

### 2.3.1 Low rank approximations in KCP

In the previous section, we saw that the Nystrom approximation provides a low rank approximation of the Gram matrix, whereas, the Random Fourier features approximate the kernel itself. But we also saw that both these methods approximate the canonical feature map $\Phi$.
For Nystrom method, $\Phi_p(x) = K_x^{\mathcal{I}} U_k \Sigma_k^{-1/2}$ where $K_x^{\mathcal{I}} = [k(x, X_i)]_{i\in\mathcal{I}} \in \mathbb{R}^{1\times p}$ and $W_k = U_k \Sigma_k U_k^T$ is the SVD of $W_k$, and for the Random features, $\Phi_p(x) = \frac{1}{\sqrt{p}}(\psi(x, \omega_1), ..., \psi(x, \omega_p))$ where $\omega_1, ..., \omega_p$ sampled independetly with respect to $\pi$.
The commun parameter here is the number of features $p$.

Each data point $X_i$ is mapped with either methods to a new point $Z_i = \Phi_p(X_i) \in \mathbb{R}^p$.
From now on, we only handle these new points in $\mathbb{R}^p$.
We have seen previously that the detection of a change in the distribution in $\mathcal{X}$ is equivalent to the detection of a change in the mean of the mapped variables in the RKHS $\mathcal{H}$.
The low rank approximation allow us to work in a very simple Hilbert space which is $\mathbb{R}^p$. The associated inner product is the usual scalar product in $\mathbb{R}^p$.

The cost of a segment $[a, b[$ is :

$$\begin{aligned} C(a,b) &= \sum_{i=a+1}^{b} k(X_i, X_i) - \frac{1}{b-a} \sum_{i=a+1}^{b} \sum_{j=a+1}^{b} k(X_i, X_j) \\ &\approx \sum_{i=a+1}^{b} Z_i Z_i^\top - \frac{1}{b-a} \sum_{i=a+1}^{b} \sum_{j=a+1}^{b} Z_i Z_j^\top \\ &\approx \sum_{i=a+1}^{b} \|Z_i\|^2 - \frac{1}{b-a} \left\| \sum_{i=a+1}^{b} Z_i \right\|^2 \end{aligned} \tag{2.5}$$

In the Binary segmentation algorithm, we work with an approximated segment cost instead of the exact segment cost :

$$\widetilde{C}(a,b) := \sum_{i=a+1}^{b} \|Z_i\|^2 - \frac{1}{b-a} \left\| \sum_{i=a+1}^{b} Z_i \right\|^2$$

Let's now review the computation complexity of the Binary segmentation KCP algorithm using the low rank approximated segment cost.

we said previously that if we managed to reduce the complexity of the update of $\widetilde{C}(a,b)$ to $\widetilde{C}(a,b+1)$ to a constant cost, we can guarentee a linear time complexity for the entire Binary segmentation KCP algorithm.

In fact, we can write $\widetilde{C}(a,b+1)$ as :

$$\widetilde{C}(a, b+1) = \sum_{i=a+1}^{b+1} \|Z_i\|^2 - \frac{1}{b+1-a} \left\| \sum_{i=a+1}^{b+1} Z_i \right\|^2$$
$$= \sum_{i=a+1}^{b} \|Z_i\|^2 + \|Z_{b+1}\|^2 - \frac{1}{b+1-a} \left\| \sum_{i=a+1}^{b} Z_i + Z_{b+1} \right\|^2$$

So, if we stock in memory, in each iteration in Algorithm 3, the two sums $\sum \|Z_i\|^2$ and $\sum Z_i$, we only have to perform three major operations to update $\widetilde{C}(a, b)$ to $\widetilde{C}(a, b+1)$ which are an addition of $\|Z_{b+1}\|^2$ to the first sum, and addition of $Z_{b+1}$ to the second sum and the computation of the norms in $\mathbb{R}^p$.
All these operation come with a cost of $\mathcal{O}(p)$ which is a constant cost (w.r.t $n$).

Therefore, in total the Binary segmentation KCP algorithm has a time complexity of $\boldsymbol{\mathcal{O}(D_{max} pn)}$ which is linear in the number of data and linear in the number of features.

Here, we assumed that we already have the new mapped data points $Z_i$. But the computation of these new point has also a cost depending on the method used in the approximation.
Nystrom method costs $\boldsymbol{\mathcal{O}(p^3 + p^2 n)}$ as seen previously.
Random Fourier features approximation cost $\boldsymbol{\mathcal{O}(p)}$ because there is no matrix decomposition or vector multiplication.

Overall, the low rank KCP with Binary segmentation requires a computational time in $\boldsymbol{\mathcal{O}(p^3 + p^2 n + D_{max} pn)}$ if the approximation is performed with the Nystrom approximation and $\boldsymbol{\mathcal{O}(D_{max} pn)}$ if we use Random features.

We can see that as long as $p = \mathcal{O}(\sqrt{n})$, we have an important gain in the computational cost compared to $\boldsymbol{\mathcal{O}(D_{max} n^2)}$ of the exact KCP algorithm.

### 2.3.2 Experiments

In order to verify the gain in computational cost with our low rank approximation methods, we experimented on a synthetic signal of size $n = 1000$. We tried to detect $D^\star = 11$ and $D^\star = 20$ change-points in a signal.
The signal is generated according to two scenarios. For each scenario, we generate $N = 500$ independent samples, from which we estimate all quantities that are reported in this section.

**Scenario 1 : Real-valued data with changing (mean, variance)**
The distribution of $X_i \in \mathbb{R}$ is randomly picked out from: $\mathcal{B}(10, 0.2)$ (Binomial), $\mathcal{NB}(3, 0.7)$ (Negative-Binomial), $\mathcal{H}(10, 5, 2)$ (Hypergeometric), $\mathcal{N}(2.5, 0.25)$ (Gaussian), $\gamma(0.5, 5)$ (Gamma), $\mathcal{W}(5, 2)$ (Weibull) and Par$(1.5, 3)$ (Pareto).
Note that the pair (mean, variance) in each segment changes from that of its neighbors.

The distribution within segment $\ell \in \{1, \dots, D^\star\}$ is given by the realization of a random variable $S_\ell \in \{1, \dots, 7\}$, each integer representing one of the 7 possible distributions. The variables $S_\ell$ are generated as follows: $S_1$ is uniformly chosen among $\{1, \dots, 7\}$, and for every $\ell \in \{1, \dots, D^\star - 1\}$, given $S_\ell, S_{\ell+1}$ is uniformly chosen among $\{1, \dots, 7\} \setminus \{S_\ell\}$. Figure 1a shows one sample generated according to this scenario.

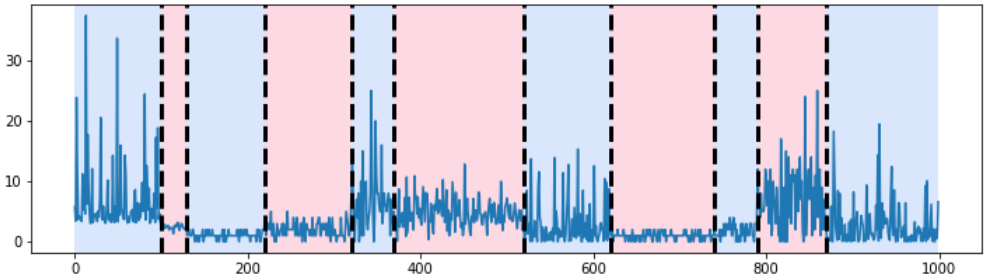



Figure 2.1: The input signal with the change-points generated by scenario 1

We tested the binary segmentation algorithm on the input signal with a Gaussian kernel of parameter $\gamma = 0.1$, then we mapped the input data using the two low rank approximation methods and tested with the same algorithm.
The choice of the Gaussian kernel comes from the fact that it is a characteristic kernel, so, it should detect any kind of change in the distribution.

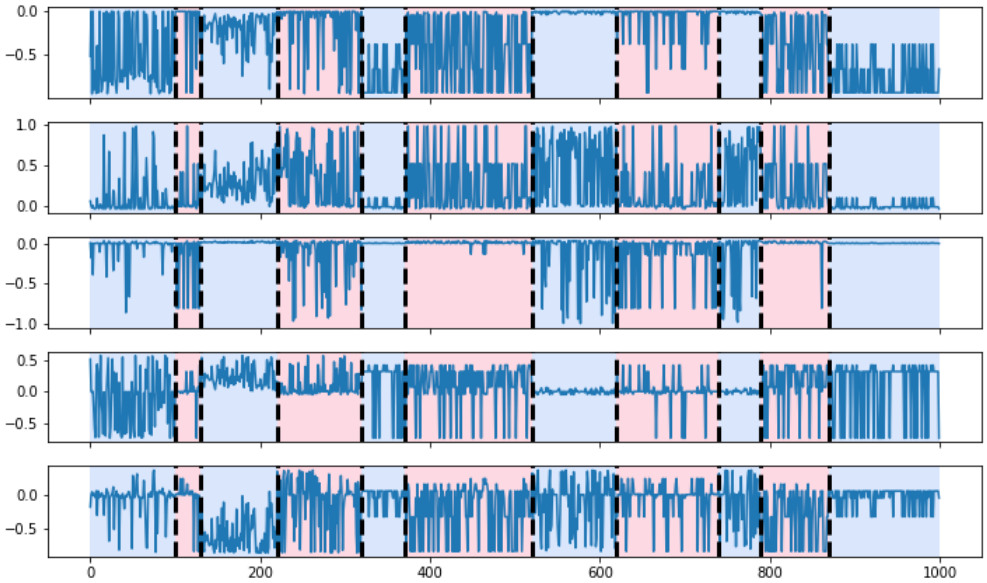



Figure 2.2: The mapped signal with the change-points (5 features)

**Scenario 2: Real-valued data with constant mean and variance**
The distribution of $X_i \in \mathbb{R}$ is randomly chosen among (1) $\mathcal{B}(0.5)$ (Bernoulli), (2) $\mathcal{N}(0.5, 0.25)$

(Gaussian) and (3) $\mathcal{E}(0.5)$ (Exponential). These three distributions have a mean 0.5 and a variance 0.25.
The distribution within segment $\ell \in \{1, \dots, D^\star\}$ is given by the realization of a random variable $S_\ell \in \{1, 2, 3\}$, similarly to what is done in Scenario 1 (replacing 7 by 3).

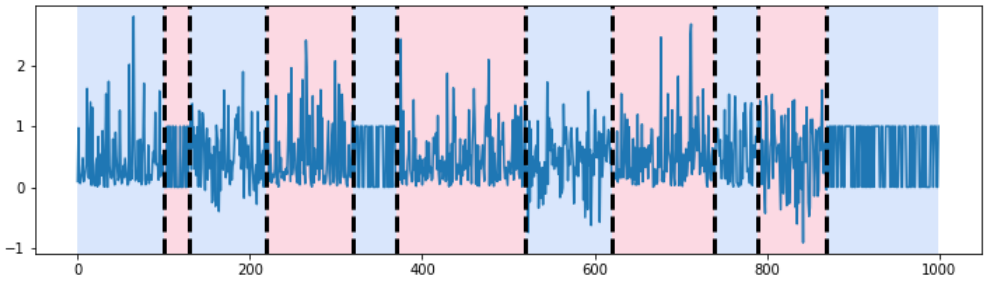



Figure 2.3: The input signal with the change-points generated by scenario 2

For this experiment, we programmed the two low rank approximation methods in **Python**, and for the KCP algorithm, we used the **ruptures** Python library developped by Truong et al. [2019].

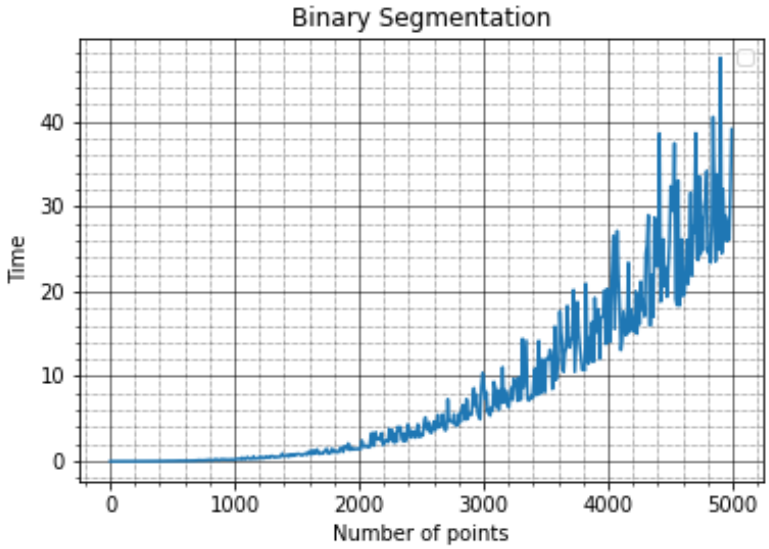



Figure 2.4: Running time of a binary segmentation algorithm

As we said previously, the cost of a Binary segmentation algorithm is quadratic in the number of data. We checked for this in our experiments, and the result is in figure 2.4.

In figure 2.5, we show the evolution of the computational time with respect to the number of features used to approximate the kernel (or the Gram matrix).

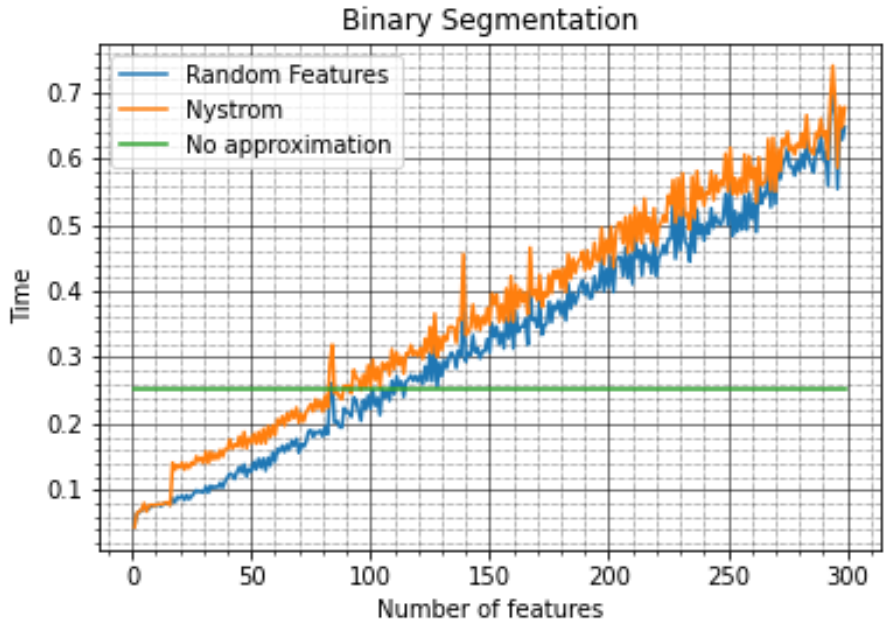



Figure 2.5: Running time for the KCP ($n = 1000$)

We can see clearly that there is a huge gain in time complexity when using either the Nystrom approximation of the Random features with a small number of features.
There is a difference of 60 % if we use 10 features, which is quite a lot considering that our signal is not very large (1000 points).
We notice that there is no interest of using the approximation methods with a number of features greater than 100 which is the square root of $n = 1000$. This confirms what we stated earlier. As long as $p = \mathcal{O}(\sqrt{n})$, we have an important gain in the computational cost.

The evolution w.r.t the number of features is linear as expected for the Random features approximation ($\boldsymbol{\mathcal{O}(D_{max}pn)}$), and it is also linear for the Nystrom approximation, which was odd at first because normally we should expect a time complexity of $\boldsymbol{\mathcal{O}(p^3 + p^2n + D_{max}pn)}$ because of the additional cost of the SVD decomposition.
But we have done some experimentation and we noticed that the ratio between the cost of the SVD and binary segmentation KCP is very low i.e.

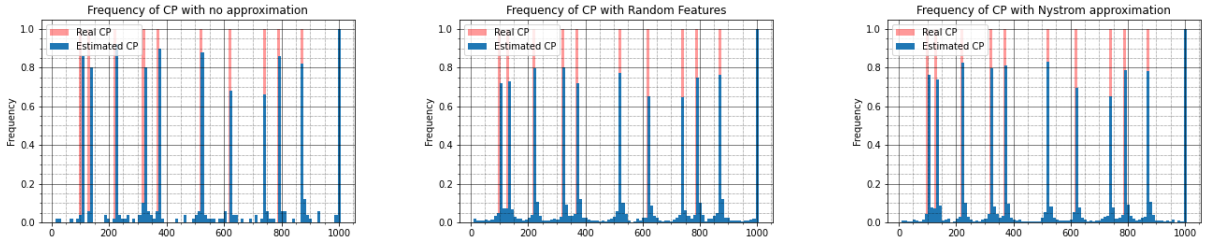
$$\frac{Cost(SVD)}{Cost(KCP)} = \frac{1}{1000}\frac{p^2n + p^3}{D_{max}pn} = \frac{1}{1000}\frac{pn + p^2}{D_{max}n}$$

Since $p \leq 300$, the SVD hardly contributes in the total computational time, but once the number of features is high enough, the quadratic term will appear.

**Statistical results : Scenario 1**

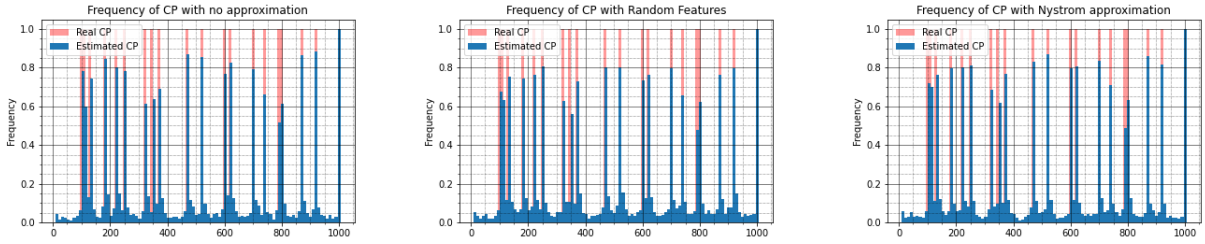
As we said previously, we generated a signal according to two scenarios, with 11 and 20 change-points, we tried to detect the change-points using binary segmentation with a Gaussian kernel (No approximation), with the random features method and with the Nystrom approximation. We repeated the experience for $N = 500$ times and ploted a histogram of the number of times a change-point was detected.

The resultes are in figures 2.6 and 2.7.




(a) (b) (c)

Figure 2.6: Probability, for each instant $i \in \{1, \cdots, n\}$, that $\widehat{\tau}$ puts a change-point at $i$ using $p = 10$ number of features and $D^\star = 11$




(a) (b) (c)

Figure 2.7: Probability, for each instant $i \in \{1, \cdots, n\}$, that $\widehat{\tau}$ puts a change-point at $i$ using $p = 10$ number of features and $D^\star = 20$

We see clearly that the performance of KCP with the two main methods of approximation is very close to the performance of the same algorithm with no approximation. For the two settings ($D^\star = 11$ and $D^\star = 20$) and with only $p = 10$ features, which is very small compared to $n = 1000$, we notice that all the real change-points are detected with probability higher than 50 %. We barely see any effect of the approximations in the results of scenario 1.
In this scenario, we can say that the low rank approximations are equally effective for recovering the true change-points. The number of change-points does not seem to have any effect on the results here.

To evaluate a segmentation, we compute a sort of distance between the estimated segmentation $\widehat{\tau}$ and the true segmenrartion $\tau^\star$.
We consider two measures of distance between segmentations. For any $\tau, \tau' \in \mathcal{T}_n$, we define the Hausdorff distance between $\tau$ and $\tau'$ by

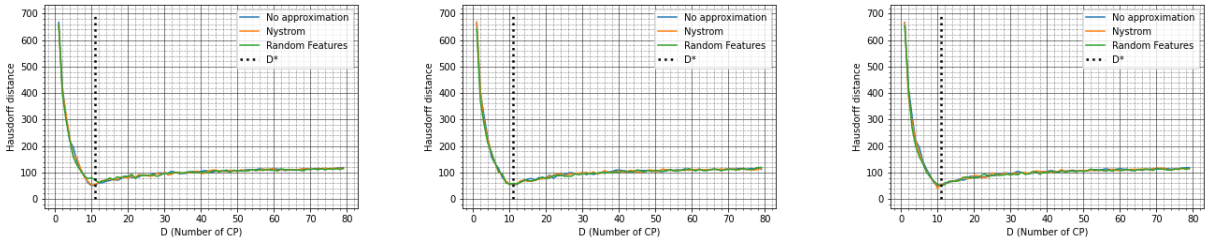
$$d_H\left(\tau, \tau'\right) := \max\left\{\max_{1\leq i\leq D_\tau-1}\min_{1\leq j\leq D_{\tau'}-1}\left|\tau_i - \tau'_j\right|, \max_{1\leq j\leq D_{\tau'}-1}\min_{1\leq i\leq D_\tau-1}\left|\tau_i - \tau'_j\right|\right\}$$

and the Frobenius distance between $\tau$ and $\tau'$

$$d_F\left(\tau, \tau'\right) := \left\|M^\tau - M^{\tau'}\right\|_F = \sqrt{\sum_{1\leq i,j\leq n}\left(M^\tau_{i,j} - M^{\tau'}_{i,j}\right)^2}$$

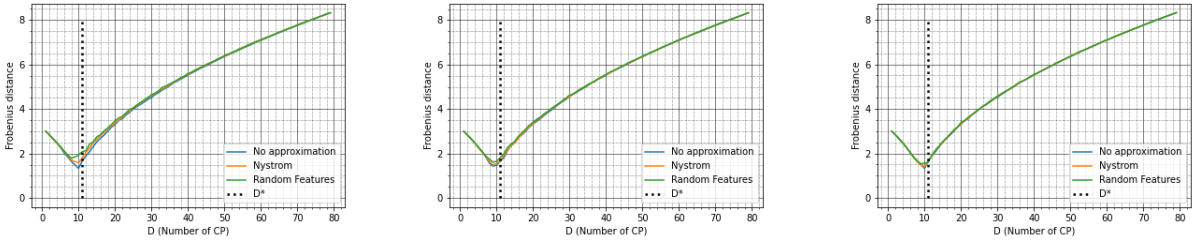
where

$$M^\tau_{i,j} = \frac{1_{\{i \text{ and } j \text{ belong to the same segment of } \tau\}}}{\text{Card(segment of } \tau \text{ containing } i \text{ and } j)}$$




(a) p = 5 (b) p = 10 (c) p = 50

Figure 2.8: Hausdorff distance between $\widehat{\tau}$ and $\tau^\star$ with $p = 5, 10, 50$ number of features and $D^\star = 11$



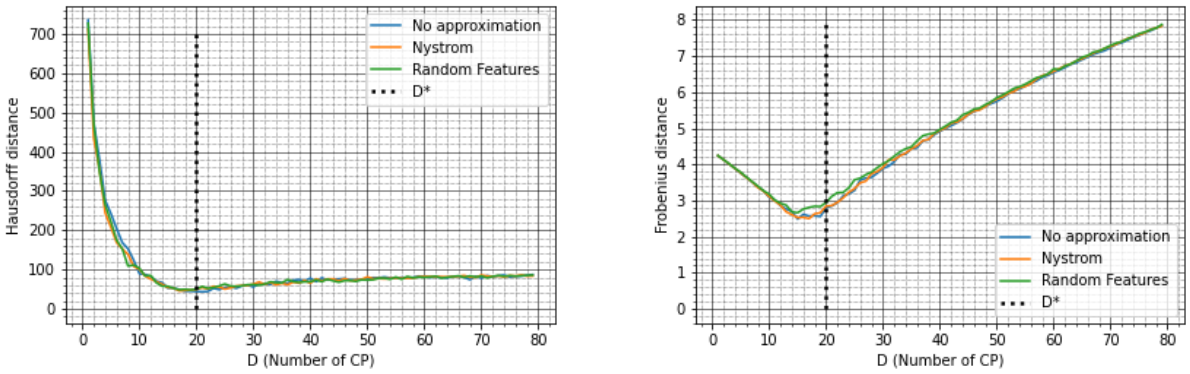

(a) p = 5 (b) p = 10 (c) p = 50

Figure 2.9: Frobenius distance between $\widehat{\tau}$ and $\tau^\star$ with $p = 5, 10, 50$ number of features and $D^\star = 11$

To study further the effect of the approximations, we measure the distances between the true segmentation and the segmentation recovered using KCP with the approximations. Here, we change the number of features ($p = 5, 10, 50$). Also, the experiments are repeated 500 times and the distances here are averaged in order to avoid stochasticity in the results.
The results are summarized in figures 2.8 and 2.9.

We can see that the Hausdorff distance and the Frobenius distance reach a minimal point in $\widehat{D} = 10$ which is relatively close to the real number of change-points $D^\star = 11$.
The curves of the distances recovered with the approximations methods, using the three number of features, are identical to the curves recovered with no approximation, which confirms that the approximations do not affect much the performances of KCP even with a small number of features (e.g. $p = 5$).

We get similar results woth $D^\star = 20$. Results are in figure 2.10.



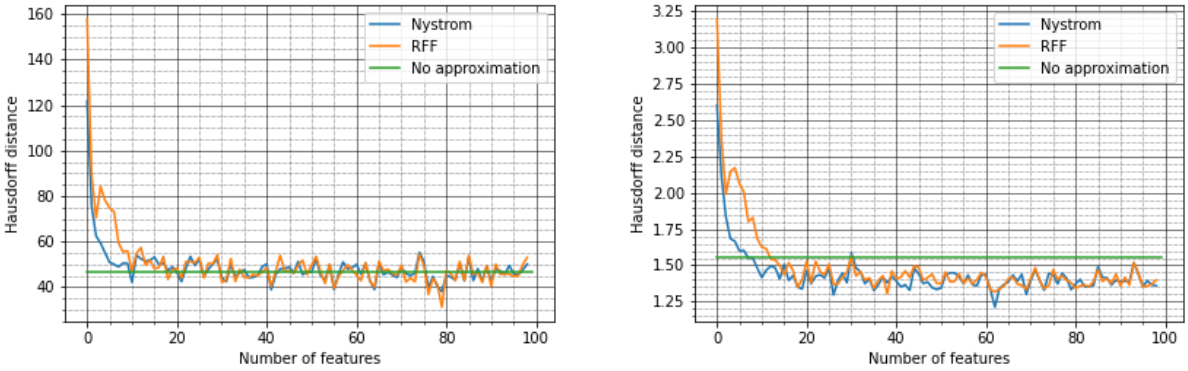

(a) Hausdorff distance

(b) Frobenius distance

Figure 2.10: Distances between $\widehat{\tau}$ and $\tau^\star$ with $p = 10$ number of features and $D^\star = 20$

Figure 2.11 shows the variation of the Hausdorff and The Frobenius distances as function of the number of features.
The distances decrease quickly until they stabilize around the distance recovered with no approximation.
The Frobenius distance reach even lower values. This difference can be explained by the high sensitivity of the Frobenius distance for false positives compared to the Hausdorff distance.

Overall, for this scenario where the mean and the variance of the distributions change, the random features approximation and the Nystrom approximation work well for KCP, the performances are relatively the same as the KCP with a Gaussian kernel. This means that these low rank approximation methods are quite robuste for this kind of scenario.

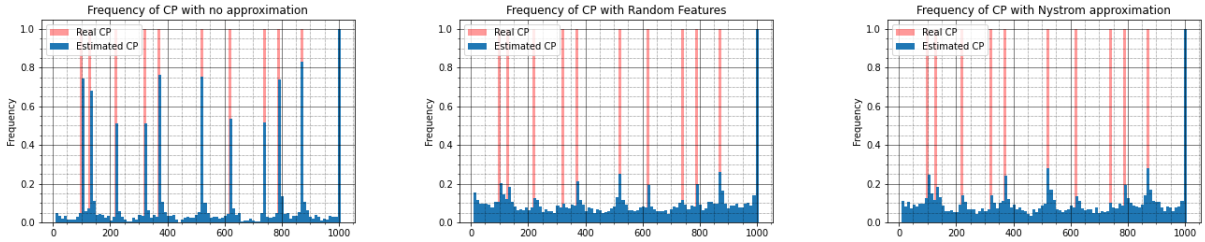



(a) Hausdorff distance (b) Frobenius distance

Figure 2.11: Distances as function of the number of features

Next, we summarize the results of the same experiments, done on a signal generated with scenario 2 where the mean and the variance are constant.

**Statistical results : Scenario 2**

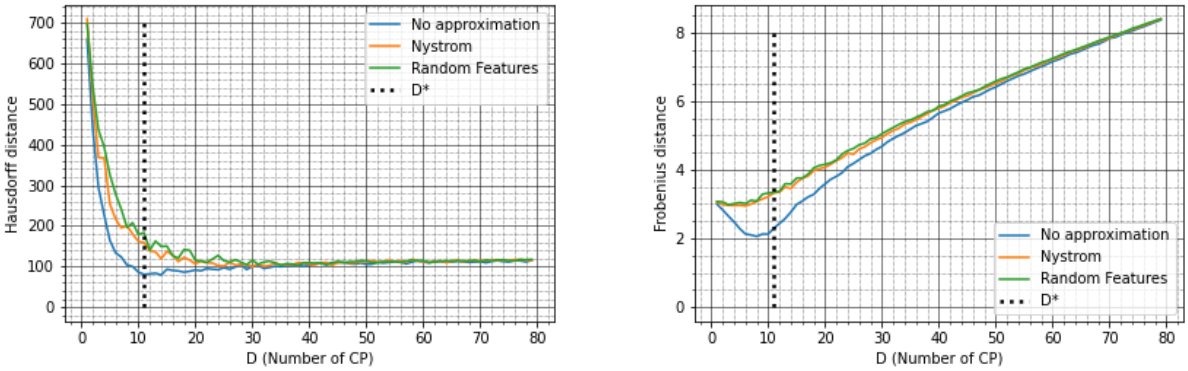



(a) (b) (c)

Figure 2.12: Probability, for each instant $i \in \{1, \cdots, n\}$, that $\widehat{\tau}$ puts a change-point at $i$ using $p = 10$ number of features and $D = 11$

All three figures show that there is a huge difference between the performances of KCP with and with no approximation.

In figure 2.12, with 10 features, the change-points are barely detected. All the probabilities of recovery are below 30 %. And the results do not change when we increase the number of features, even with $p = n = 1000$, the probabilities are still very low. Figure 2.14 emphasizes on the difference between the distances. It is clearer that the number of features has no effect on the performance for this scenario. The Hausdorf and Frobenius distances fluctuate around constant values very far from the distances recovered with no approximation.

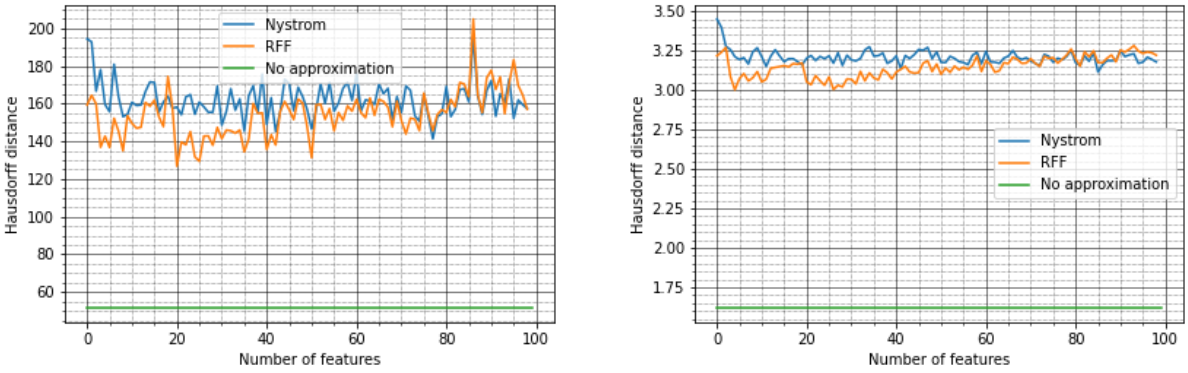



(a) Hausdorff distance

(b) Frobenius distance

Figure 2.13: Distances between $\widehat{\tau}$ and $\tau^\star$ with $p = 10$ number of features and $D = 11$

This huge difference in the performance between the results of scenario 1 and scenario 2, given that we tried to study the effect of the parameter $D^\star$ and $p$, can only mean that some other parameters control how effective the approximation can be.




(a) Hausdorff distance

(b) Frobenius distance

Figure 2.14: Distances as function of the number of features

# Chapter 3

# Theoretical Analysis

## 3.1 Abstract Formulation of KCP and Assumptions

Let $\mathcal{H} = \mathcal{H}_k$ denote the reproducing kernel Hilbert space (RKHS) associated to the positive semi-definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The canonical feature map $\Phi : \mathcal{X} \mapsto \mathcal{H}$ is then defined by $\Phi(x) = k(x, \cdot) \in \mathcal{H}$ for every $x \in \mathcal{X}$.
Let us define $Y_i = \Phi(X_i) \in \mathcal{H}$ for every $i \in \{1, \dots, n\}, Y = (Y_i)_{1 \leq i \leq n} \in \mathcal{H}^n, \mathcal{T}_n := \bigcup_{D=1}^{n} \mathcal{T}_n^D$ the set of segmentations of $D$ change-points as defined in Chapter 2 and for every $\tau \in \mathcal{T}_n$

$$F_\tau := \left\{ f = (f_1, \dots, f_n) \in \mathcal{H}^n \text{ s.t. } f_{\tau_{t-1}+1} = \cdots = f_{\tau_e} \quad \forall 1 \leq \ell \leq D_\tau \right\}$$

which is a linear subspace of $\mathcal{H}^n$. We also define on $\mathcal{H}^n$ the canonical scalar product by $\langle f, g \rangle := \sum_{i=1}^{n} \langle f_i, g_i \rangle_{\mathcal{H}}$ for $f, g \in \mathcal{H}^n$, and we denote by $\| \cdot \|$ the corresponding norm. Then, for any $g \in \mathcal{H}^n$

$$\Pi_\tau g := \operatorname{argmin}_{f \in F_\tau} \left\{ \|f - g\|^2 \right\} \tag{3.1}$$

is the orthogonal projection of $g \in \mathcal{H}^n$ onto $F_\tau$, and satisfies

$$\forall g \in \mathcal{H}^n, \forall 1 \leq \ell \leq D_\tau, \forall i \in [\tau_{\ell-1} + 1, \tau_\ell], \quad (\Pi_\tau g)_i = \frac{1}{\tau_\ell - \tau_{\ell-1}} \sum_{j=\tau_{\ell-1}+1}^{\tau_\ell} g_j \tag{3.2}$$

The proof of this assertion is given in Celisse et al. [2012].

If the kernel $k$ is shift invariant, then it has an integral representation :

$$k\left(x, x'\right) = \int_\Omega \psi(x, \omega) \psi\left(x', \omega\right) d\pi(\omega), \quad \forall x, x' \in \mathcal{X} \tag{3.3}$$

where $(\Omega, \pi)$ is probability space and $\psi : X \times \Omega \to \mathbb{R}$.

The Least Square Criterion associated to the kernel $k$ is :

$$\widehat{\mathcal{R}}_n(\tau) := \frac{1}{n} \sum_{i=1}^{n} k\left(X_i, X_i\right) - \frac{1}{n} \sum_{\ell=1}^{D_\tau} \left[ \frac{1}{\tau_\ell - \tau_{\ell-1}} \sum_{i=\tau_{\ell-1}+1}^{\tau_\ell} \sum_{j=\tau_{\ell-1}+1}^{\tau_\ell} k\left(X_i, X_j\right) \right] \tag{3.4}$$

Using the kernel approximation via the Random Fourier Features we have :

$$k(X_i, X_j) \approx \frac{1}{p} \sum_{k=1}^{p} \psi(\omega_k, X_i)\psi(\omega_k, X_j) \tag{3.5}$$

where $\omega_1, ..., \omega_p$ are sampled independently with respect to $\pi$ and $p$ is the number of features used to approximate the kernel.

By replacing $k(X_i, X_j)$ in 3.4 with its approximation we define the RFF Least Square Criterion by :

$$\widetilde{\mathcal{R}}_n(\tau) := \frac{1}{p} \sum_{k=1}^{p} Z_k^\tau \tag{3.6}$$

where

$$Z_k^\tau = \frac{1}{n} \sum_{i=1}^{n} \psi^2(\omega_k, X_i) - \frac{1}{n} \sum_{\ell=1}^{D_\tau} \left[ \frac{1}{\tau_\ell - \tau_{\ell-1}} \sum_{i=\tau_{\ell-1}+1}^{\tau_\ell} \sum_{j=\tau_{\ell-1}+1}^{\tau_\ell} \psi(\omega_k, X_i)\psi(\omega_k, X_j) \right] \tag{3.7}$$

Note that each random variable $Z_k^\tau$ depends on $\omega_k$ which is a random variable, on the data points $(X_1, ..., X_n)$ which are also stochastic and on the segmentation $\tau$.

We can verify that the least square criterion can be written as $\widehat{\mathcal{R}}_n(\tau) = \frac{1}{n} \|Y - \widehat{\mu}_\tau\|^2$ where $\widehat{\mu}_\tau = \Pi_\tau Y$

KCP is a change-point method that aims to detect changes in the mean of the distribution $P_X$ of the input signal mapped into an RKHS, $Y_i = \Phi(X_i) \in \mathcal{H}$.
If $\mathcal{H}$ is separable and $\mathbb{E}[k(X_i, X_i)] < +\infty$, we can define the (Bochner) mean $\mu_i^\star \in \mathcal{H}$ of $\Phi(X_i)$ by

$$\forall g \in \mathcal{H}, \quad \langle \mu_i^\star, g \rangle_{\mathcal{H}} = \mathbb{E}[g(X_i)] = \mathbb{E}[\langle Y_i, g \rangle_{\mathcal{H}}]$$

So, we can write

$$\forall 1 \leq i \leq n, \quad Y_i = \mu_i^\star + \varepsilon_i \in \mathcal{H} \quad \text{where} \quad \varepsilon_i := Y_i - \mu_i^\star$$

where $(\varepsilon_i)_{1 \leq i \leq n}$ are independent and zero mean variables. So, $\widehat{\mu}_\tau$ can be seen as the least-squares estimator over $F_\tau$ of $\mu^\star = (\mu_1^\star, \ldots, \mu_n^\star) \in \mathcal{H}^n$

As stated earlier, in order to define the Bochner mean element $\mu_i^\star$ the RKHS should be separable. The theoretical study we made is on Random Fourier Feature approximation which is valid only when the kernel $k$ is shift invariant.

These conditions are to be assumed in order to carry out our theoretical study.

**Assumption 1** *The kernel $\boldsymbol{k}$ is shift invariant. That means that it has an integral representation*

$$k(x, x') = \bar{k}(x - x') = \int_\Omega \psi(x, \omega)\psi(x', \omega)\, d\pi(\omega), \quad \forall x, x' \in \mathcal{X}$$

**Assumption 2** *$\mathcal{H}$ is a separable Hilbert space.*

**Assumption 3** *There exists $M > 0$ such that $|\psi(x, \omega)| \leq M$ almost surely.*

The separability of $\mathcal{H}$ is assumed since it is satisfied by most of the resonably used kernels. Note that assumption 1 is only needed in proposition 3.

We can deduce from Assumption 3 that the kernel itself is bounded by $M^2$.

For every $1 \leq i \leq n$, we also define the "variance" of $Y_i$ by

$$v_i := \mathbb{E}\left[\|\Phi(X_i) - \mu_i^\star\|_{\mathcal{H}}^2\right] = \mathbb{E}\left[k(X_i, X_i)\right] - \|\mu_i^\star\|_{\mathcal{H}}^2$$

If Assumption 3 holds true, then $\forall i, j \in \{1, \ldots, n\}, \quad |k(X_i, X_j)| \leq M^2$ and

$$\forall i \in \{1, \ldots, n\}, \quad \mathbb{E}\left[\|Y_i\|_{\mathcal{H}}\right] = \mathbb{E}\left[\|\Phi(X_i)\|_{\mathcal{H}}\right] = \mathbb{E}\left[\sqrt{k(X_i, X_i)}\right] \leq M < \infty \quad a.s.$$

So, the mean element $\mu^\star = (\mu_1^\star, \ldots, \mu_n^\star) \in \mathcal{H}^n$ can be defined and $\forall i \in \{1, \ldots, n\} \quad v_i \leq M^2$.
We denote $v_{max} = \max_{i \leq n} v_i$.

## 3.2 Concentration inequalities

The main theoretical result, stated in Section 3.3, relies on three concentration inequalities, two for some linear and quadratic functionals of Hilbert-valued vectors and one for the approximated RFF Least Square Criterion.

**Proposition 1** ***(Concentration of the quadratic term)*** *Let $\tau \in \mathcal{T}_n$ and recall that $\Pi_\tau$ is the orthogonal projection onto $F_\tau$ in $\mathcal{H}^n$ defined by 3.1. Let $X_1, \ldots, X_n$ be independent $\mathcal{X}$ -valued random variables and assume that Assumptions 2 and 3 hold true, so that we can define $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n) \in \mathcal{H}^n$ as in the previous section. Then for every $x > 0$, with probability at least $1 - \mathrm{e}^{-x}$*

$$\|\Pi_\tau \varepsilon\|^2 - \mathbb{E}\left[\|\Pi_\tau \varepsilon\|^2\right] \leq \frac{14M^2}{3}(x + 2\sqrt{2xD_\tau})$$

The proof of this proposition is based on a combination of Bernstein's and Pinelis-Sakhanenko's inequalities.
We can also prove with the same reasoning that for every $x > 0$, with probability at least $1 - \mathrm{e}^{-x}$

$$\|\Pi_\tau \varepsilon\|^2 - \mathbb{E}\left[\|\Pi_\tau \varepsilon\|^2\right] \geq -\frac{14M^2}{3}(x + 2\sqrt{2xD_\tau})$$

**Proposition 2** ***(Concentration of the linear term)***
*For every $x > 0$, with probability at least $1 - 2\mathrm{e}^{-x}$ :*

$$\forall \theta > 0, \quad |\langle (I - \Pi_\tau)\mu^\star, \varepsilon \rangle| \leq \theta \|\Pi_\tau \mu^\star - \mu^\star\|^2 + \left(\frac{v_{\max}}{2\theta} + \frac{4M^2}{3}\right) x$$

Propositions 1 and 2 are introduced and proved in Celisse et al. [2012] and will not be proven here.

Our main contribution in this work is the concentration inequality presented in proposition 3 and that will serve along with the previous proposition to prove the main theorem. It is based on Bernstein inequality applied to the random variables $Z_k^\tau$.

**Proposition 3** ***(Concentration of the RFF least square criterion)*** *Let $\tau \in \mathcal{T}_n$, let $X_1, \ldots, X_n$ be independent $\mathcal{X}$ -valued random variables and let $p \in \mathbb{N}^\star$ and Assume that Assumptions 1, 2 and 3 hold true. $\widehat{\mathcal{R}}_n(\tau)$ and $\widetilde{\mathcal{R}}_n(\tau)$ are respectively the exact least square criterion and the RFF least square criterion approximated with p Random Features. Then for every $x > 0$, with probability at least $1 - \mathrm{e}^{-x}$ :*

$$\forall \theta > 0, \quad \widetilde{\mathcal{R}}_n(\tau) - (1+\theta)\widehat{\mathcal{R}}_n(\tau) \leq \frac{4M^2 x}{p}\left(\frac{1}{3} + \theta^{-1}\right)$$

*Similarely, for every $x > 0$, with probability at least $1 - \mathrm{e}^{-x}$ :*

$$\forall \theta > 0, \quad \widetilde{\mathcal{R}}_n(\tau) - (1-\theta)\widehat{\mathcal{R}}_n(\tau) \geq -\frac{4M^2 x}{p}\left(\frac{1}{3} + \theta^{-1}\right)$$

## 3.3 Oracle Inequality for KCP with RFF

Similarly to the results of Celisse et al. [2012], we state below a non-asymptotic oracle inequality for KCP with Random Fourier Features Approximation. First we define the quadratic risk of any $\mu \in \mathcal{H}^n$ as an estimator of $\mu^\star$ by

$$\mathcal{R}(\mu) = \frac{1}{n}\left\|\mu - \mu^\star\right\|^2 = \frac{1}{n}\sum_{i=1}^{n}\left\|\mu_i - \mu_i^\star\right\|_{\mathcal{H}}^2$$

**Theorem 6** *We consider the framework and notation introduced in previous sections. Let $C \geq 0$ be a positive constant. Assume that Assumptions 1, 2 and 3 hold true and that **pen** : $\mathcal{T}_n \to \mathbb{R}$ is some penalty function satisfying*

$$\forall \tau \in \mathcal{T}_n, \quad \mathrm{pen}(\tau) \geq CM^2\left[\frac{1}{n} + \frac{1}{\sqrt{p}}\right]\left[\log\begin{pmatrix} n-1 \\ D_\tau - 1\end{pmatrix} + D_\tau\right]$$

*Then, some numerical constants $L > 0$ and $B > 0$ exist such that the following holds: if $C \geq L$, for every $y \geq 0$, an event of probability at least $1 - \mathrm{e}^{-y}$ exists on which, for every*

$$\widetilde{\tau} \in \mathrm{argmin}_{\tau \in \mathcal{T}_n}\left\{\widetilde{\mathcal{R}}_n(\tau) + \mathrm{pen}(\tau)\right\}$$

*we have*

$$\mathcal{R}\left(\widehat{\mu}_{\widetilde{\tau}}\right) \leq 2\inf_{\tau \in \mathcal{T}_n}\left\{\mathcal{R}\left(\widehat{\mu}_\tau\right) + \mathrm{pen}(\tau)\right\} + ByM^2\left[\frac{1}{n} + \frac{1}{\sqrt{p}}\right]$$

**Theorem 7** *We consider the framework and notation introduced in previous sections. Let $C \geq 0$ be a positive constant and $\theta > 0$. Assume that Assumptions 1, 2 and 3 hold true and that **pen** : $\mathcal{T}_n \to \mathbb{R}$ is some penalty function satisfying*

$$\forall \tau \in \mathcal{T}_n, \quad \mathrm{pen}(\tau) \geq CM^2\left[\frac{1}{n} + \frac{1}{p}\right]\left[\log\begin{pmatrix} n-1 \\ D_\tau - 1\end{pmatrix} + D_\tau\right]$$

*Then, some numerical constants $L_\theta > 0$ and $B_\theta > 0$ exist such that the following holds: if $C \geq L_\theta$, for every $y \geq 0$, an event of probability at least $1 - \mathrm{e}^{-y}$ exists on which, for every*

$$\widetilde{\tau} \in \mathrm{argmin}_{\tau \in \mathcal{T}_n} \left\{ \widetilde{\mathcal{R}}_n(\tau) + \mathrm{pen}(\tau) \right\}$$

*we have*

$$\mathcal{R}\left(\widehat{\mu}_{\widetilde{\tau}}\right) \leq 2 \inf_{\tau \in \mathcal{T}_n} \left\{ \mathcal{R}\left(\widehat{\mu}_\tau\right) + \mathrm{pen}(\tau) \right\} + B_\theta y M^2 \left[ \frac{1}{n} + \frac{1}{p} \right] + 6\theta M^2$$

**Discussion :**

The main result of our work are Theorem 7 (and Theorem 6) which are inspired from an Oracle inequality stated for KCP in Celisse et al. [2012].
If we do not work under any type of approximation of the kernel, then the least square criterion is $\widehat{\mathcal{R}}_n(\tau)$.
The theorem of Celisse et al. [2012] is the following :

**Theorem 8** ***Celisse et al. [2012]*** *We consider the framework and notation introduced in previous sections. Let $C \geq 0$ be a positive constant. Assume that Assumptions 2 and 3 hold true and that* ***pen*** *: $\mathcal{T}_n \to \mathbb{R}$ is some penalty function satisfying*

$$\forall \tau \in \mathcal{T}_n, \quad \mathrm{pen}(\tau) \geq \frac{CM^2}{n} \left[ \log \begin{pmatrix} n-1 \\ D_\tau - 1 \end{pmatrix} + D_\tau \right]$$

*Then, some numerical constant $L > 0$ exists such that the following holds: if $C \geq L$, for every $y \geq 0$, an event of probability at least $1 - \mathrm{e}^{-y}$ exists on which, for every*

$$\widetilde{\tau} \in \mathrm{argmin}_{\tau \in \mathcal{T}_n} \left\{ \widehat{\mathcal{R}}_n(\tau) + \mathrm{pen}(\tau) \right\}$$

*we have*

$$\mathcal{R}\left(\widehat{\mu}_{\widetilde{\tau}}\right) \leq 2 \inf_{\tau \in \mathcal{T}_n} \left\{ \mathcal{R}\left(\widehat{\mu}_\tau\right) + \mathrm{pen}(\tau) \right\} + \frac{83 y M^2}{n}$$

This thoerem means that the quadratic risk of $\widehat{\mu}_{\widehat{\tau}} = \Pi_{\widehat{\tau}} Y$ is very close to the smallest quadratic risk over all the segmentations (with a certain penalty term) plus a quantity that decreases with the number of data points.
We deduce that the empirical criterion $\widehat{\mathcal{R}}_n(\tau) + \mathrm{pen}(\tau)$ mimics the theoritical quadratic risk $\mathcal{R}\left(\widehat{\mu}_\tau\right)$ which is the best criterion to detect the best segmentation $\tau \in \mathcal{T}_n$.
In other words, $\widehat{\mu}_{\widetilde{\tau}}$ is almost as good as the best estimator $\widehat{\mu}_{\tau^*}$ in a Mean-squared error sense.

Oracle inequalities are in fact very powerful tools in Statistics. This is in general the idea behind them : we try to measure the performance of an estimator by comparing it with the ideal or best estimator over the set of parameters (which are $\tau$ in our case). We refer to Candes [2006] for more information about the Oracle inequalities.

In the framework of low rank approximation, we add another element of error to the equation which is the approximation of the kernel or the Gram matrix via Nystrom approximation or Random

features.
In this case, we are no longer allowed to use the exact least square criterion $\widehat{\mathcal{R}}_n(\tau)$ because it requires the use of the $n \times n$ Gram matrix. So, instead, we use practically an approximated criterion denoted $\widetilde{\mathcal{R}}_n(\tau)$ which has the same formula but with the approximated values of $k(X_i, X_j)$ instead.
The most important parameter in the approximation is the number of features $p$. Intuitively, we can see that the closest $p$ to $n$ the better is the approximation and the better is the KCP performance.

This is the goal of Theorem 7.
in Theorem 7, we provide a theoretical guarentee of KCP with Random Features low rank approximation.
Given a segmentation $\widetilde{\tau}$ that satisfies the new approximative RFF criterion
$\widetilde{\tau} \in \text{argmin}_{\tau \in \mathcal{T}_n} \left\{ \widetilde{\mathcal{R}}_n(\tau) + \text{pen}(\tau) \right\}$, we have a similar bound of $\mathcal{R}\left(\widehat{\mu}_{\widetilde{\tau}}\right)$ but with an additional term decreasing with the number of data point and the **number of features** plus an bias term that we can choose arbirtrarily small (Theorem 7).
If we discard the bias term $6\theta M^2$, we would have proven a guarentee of the convergence of the approximation with the random features w.r.t the number of features $p$, and this is the robustness of our theorem. We notice in the literature that all the error bounds are in $\mathcal{O}(\frac{1}{\sqrt{p}})$. Well, we managed to do better with our $\mathcal{O}(\frac{1}{p})$ in Theorem 7 with a Bernstein inequality plus a little trick that will be shown in the proof section. The additionnal bias term $6\theta M^2$ can be arbitrarely small because we are free to choose any value of $\theta$.

If we want to get ride of the bias term, we would have an error decreasing with $\frac{1}{\sqrt{p}}$, which is the result of Theorem 6.

What is the weekness of our main theorem ?

We notice that if we want to guarentee an Oracle inequality similar to the one with no approximation, we should have in the best case scenario $p \sim n$ or $p \sim n^2$. For the first one it is useless and for the second one it is even worse than $n$ which is inacceptable in practice for detecting changes. We would have a time complexity in $\mathcal{O}(n^3)$ or $\mathcal{O}(n^5)$.

The ultimate goal would be to prove a similar theorem with a power of $p$ instead of $p$ or $\sqrt{p}$.
For example a bound of the form :

$$\mathcal{R}\left(\widehat{\mu}_{\widetilde{\tau}}\right) \leq 2 \inf_{\tau \in \mathcal{T}_n} \left\{ \mathcal{R}\left(\widehat{\mu}_{\tau}\right) + \text{pen}(\tau) \right\} + ByM^2 \left[ \frac{1}{n} + \frac{1}{p^{\alpha}} \right]$$

where $\alpha > 1$.

## 3.4 Main Proofs

We will now prove the results of this work, **Proposition 3** and **Theorem 7**.

### 3.4.1 Proof of Proposition 3

Under Assumption 1, the kernel $k$ has an integral representation 3.3.
If we choose a number $p$ of random features to approximate the kernel $k$ we have then :

$$\widetilde{\mathcal{R}}_n(\tau) := \frac{1}{p}\sum_{k=1}^{p} Z_k^\tau$$

where

$$Z_k^\tau = \frac{1}{n}\sum_{i=1}^{n}\psi^2(\omega_k, X_i) - \frac{1}{n}\sum_{\ell=1}^{D_\tau}\left[\frac{1}{\tau_\ell - \tau_{\ell-1}}\sum_{i=\tau_{\ell-1}+1}^{\tau_\ell}\sum_{j=\tau_{\ell-1}+1}^{\tau_\ell}\psi(\omega_k, X_i)\psi(\omega_k, X_j)\right]$$

and $\omega_1, ..., \omega_p$ are sampled independently with respect to $\pi$.

Since $\mathbb{E}_\pi\left[\psi(\omega_k, X_i)\psi(\omega_k, X_j)\right] = k(X_i, X_j)$, via the expectation linearity property we have :

$$\forall 1 \leq k \leq p, \quad \mathbb{E}_\pi\left[Z_k^\tau\right] = \widehat{\mathcal{R}}_n(\tau)$$

The random variables $(\omega_k)_k$ are i.i.d and each random variable $Z_k^\tau$ is a measurable function of $\omega_k$. So $(Z_k^\tau)_k$ are also i.i.d.
And since $|\psi| \leq M$ almost surely, with a simple triangle inequality we guarentee that $\forall 1 \leq k \leq p, \quad |Z_k^\tau| \leq 2M^2$

With these properties, we are tempted to use a concentration inequality.
The first one we though of is Hoeffding's inequality, which is used to prove Teorem 6, but the result of Theorem 6 only guarentees a $\mathcal{O}(\frac{1}{\sqrt{p}})$ upper bound for the concentration which was not satisfying as we showed previously.

In Theorem 7, we try to do better using a Berstein inequality, but it comes with an additional cost.

**Proposition 4** ***(Hoeffding's inequality for sum of random variables)***. *Let $X_1, \ldots, X_p$ be a sequence of independent and identically distributed random variables on $\mathbb{R}$ with zero mean. If there exists an $T \in \mathbb{R}$ such that $X_i \leq T$ almost everywhere for $i \in \{1, \ldots, p\}$. For any $x > 0$ the following holds with probability at least $1 - \mathrm{e}^{-x}$*

$$\frac{1}{p}\sum_{i=1}^{p} X_i \leq T\sqrt{\frac{x}{2p}}$$

*If there exists $T' \geq \max_i |X_i|$ almost everywhere, then the same bound, with $T'$ instead of $T$, holds for the for the absolute value of the left hand side, with probability at least $1 - 2\mathrm{e}^{-x}$*

**Proposition 5** ***(Bernstein's inequality for sum of random variables)****. Let $X_1, \dots, X_p$ be a sequence of independent and identically distributed random variables on $\mathbb{R}$ with zero mean. If there exists an $T, S \in \mathbb{R}$ such that $X_i \leq T$ almost everywhere and $\mathbb{E}X_i^2 \leq S$, for $i \in \{1, \dots, p\}$. For any $x > 0$ the following holds with probability at least $1 - \mathrm{e}^{-x}$*

$$\frac{1}{p}\sum_{i=1}^{p} X_i \leq \frac{2Tx}{3p} + \sqrt{\frac{2Sx}{p}}$$

*If there exists $T' \geq \max_i |X_i|$ almost everywhere, then the same bound, with $T'$ instead of $T$, holds for the for the absolute value of the left hand side, with probability at least $1 - 2\mathrm{e}^{-x}$*

We will try to apply the Bernstein inequlity with a little twist to discard the $\mathcal{O}(\frac{1}{\sqrt{p}})$ term.
We should make sure that our variables $Z_k^\tau$ fall into the class of variables that satisfy the Bernstein properties.

- The variables $\widetilde{Z}_k^\tau := Z_k^\tau - \widehat{\mathcal{R}}_n(\tau)$ are zero mean variables, independent and identically distributed because $\omega_k$ are.
- $\widetilde{Z}_k^\tau \leq Z_k^\tau \leq 2M^2$ almost everywhere. ($\widehat{\mathcal{R}}_n(\tau) \geq 0$)
- $\mathbb{E}_\pi\left(\left|\widetilde{Z}_k^\tau\right|^2\right) = \mathbb{V}(Z_k^\tau) = \mathbb{E}_\pi\left(|Z_k^\tau|^2\right) - \widehat{\mathcal{R}}_n(\tau)^2 \leq 2M^2\mathbb{E}_\pi\left(|Z_k^\tau|\right)$

  The variable $Z_k^\tau$ is positive. In fact,

$$Z_k^\tau = \frac{1}{n}\sum_{\ell=1}^{D_\tau}\sum_{i=\tau_{\ell-1}+1}^{\tau_\ell}\left|\psi(\omega_k, X_i) - \frac{1}{\tau_\ell - \tau_{\ell-1}}\sum_{j=\tau_{\ell-1}+1}^{\tau_\ell}\psi(\omega_k, X_j)\right|^2 \geq 0$$

  So, $\mathbb{E}_\pi\left(\left|\widetilde{Z}_k^\tau\right|^2\right) =\leq 2M^2\mathbb{E}_\pi\left(Z_k^\tau\right) = 2M^2\widehat{\mathcal{R}}_n(\tau)$.

  So, the variable $\widetilde{Z}_k^\tau$ satisfies the Bernstein conditions with $T = 2M^2$ and $S = 2M^2\widehat{\mathcal{R}}_n(\tau)$.

  We can now apply the Bernstein's inequality : For every $x > 0$,

$$\mathbb{P}_\omega\left(\frac{1}{p}\sum_{i=1}^{p} Z_k^\tau - \widehat{\mathcal{R}}_n(\tau) \leq \frac{4M^2x}{3p} + \sqrt{\frac{4M^2\widehat{\mathcal{R}}_n(\tau)x}{p}}\right) \geq 1 - \mathrm{e}^{-x}$$

  We know that for every $\theta > 0$, $\quad\sqrt{\frac{4M^2\widehat{\mathcal{R}}_n(\tau)x}{p}} \leq \widehat{\mathcal{R}}_n(\tau)\theta + \frac{4M^2x}{p}\theta^{-1}$, so,

$$\mathbb{P}_\omega\left(\frac{1}{p}\sum_{i=1}^{p} Z_k^\tau - \widehat{\mathcal{R}}_n(\tau) \leq \frac{4M^2x}{3p} + \widehat{\mathcal{R}}_n(\tau)\theta + \frac{4M^2x}{p}\theta^{-1}\right)$$

is greater than

$$\mathbb{P}_\omega\left(\frac{1}{p}\sum_{i=1}^{p} Z_k^\tau - \widehat{\mathcal{R}}_n(\tau) \leq \frac{4M^2x}{3p} + \sqrt{\frac{4M^2\widehat{\mathcal{R}}_n(\tau)x}{p}}\right)$$

So, for every $x > 0$ and for every $\theta > 0$,

$$\mathbb{P}_\omega \left( \widetilde{\mathcal{R}}_n(\tau) - (1+\theta)\widehat{\mathcal{R}}_n(\tau) \leq \frac{4M^2 x}{p} \left( \frac{1}{3} + \theta^{-1} \right) \right) \geq 1 - \mathrm{e}^{-x}$$

We are very close to proving proposition 3, here we only took into account the stochasticity according to $\omega_1, ..., \omega_p$, so in fact what we proved so far is : For every $x > 0$ and for every $\theta > 0$,

$$\mathbb{P}_{\omega,X} \left( \widetilde{\mathcal{R}}_n(\tau) - (1+\theta)\widehat{\mathcal{R}}_n(\tau) \leq \frac{4M^2 x}{p} \left( \frac{1}{3} + \theta^{-1} \right) | X_1, ..., X_n \right) \geq 1 - \mathrm{e}^{-x}$$

Let $A(\omega, X, \tau)$ be the random variable $\boldsymbol{\widetilde{\mathcal{R}}_n(\tau) - (1+\theta)\widehat{\mathcal{R}}_n(\tau)}$.

$$\begin{aligned}
\mathbb{P}_{\omega,X} \left( A(\omega, X, \tau) \leq \frac{4M^2 x}{p} \left( \frac{1}{3} + \theta^{-1} \right) \right) &= \mathbb{E} \left( \mathbb{1}_{A(\omega,X,\tau) \leq \frac{4M^2 x}{p} \left( \frac{1}{3} + \theta^{-1} \right)} \right) \\
&= \mathbb{E} \left( \mathbb{E} \left( \mathbb{1}_{A(\omega,X,\tau) \leq \frac{4M^2 x}{p} \left( \frac{1}{3} + \theta^{-1} \right)} | X_1, ..., X_n \right) \right) \\
&= \mathbb{E} \left( \mathbb{P}_\omega \left( A(\omega, X, \tau) \leq \frac{4M^2 x}{p} \left( \frac{1}{3} + \theta^{-1} \right) \right) \right) \\
&\geq \mathbb{E} \left( 1 - \mathrm{e}^{-x} \right) = 1 - \mathrm{e}^{-x}
\end{aligned}$$

So, $\mathbb{P} \left( \boldsymbol{\widetilde{\mathcal{R}}_n(\tau) - (1+\theta)\widehat{\mathcal{R}}_n(\tau)} \leq \frac{4M^2 x}{p} \left( \frac{1}{3} + \theta^{-1} \right) \right) \geq 1 - \mathrm{e}^{-x}$
Finally, for every $x > 0$ and with probability higher than $1 - \mathrm{e}^{-x}$ we have :

$$\forall \theta > 0, \quad \widetilde{\mathcal{R}}_n(\tau) - (1+\theta)\widehat{\mathcal{R}}_n(\tau) \leq \frac{4M^2 x}{p} \left( \frac{1}{3} + \theta^{-1} \right)$$

Using exactely the same reasoning we prove the other part of the proposition.
For every $x > 0$ and with probability higher than $1 - \mathrm{e}^{-x}$ we have :

$$\forall \theta > 0, \quad \widetilde{\mathcal{R}}_n(\tau) - (1-\theta)\widehat{\mathcal{R}}_n(\tau) \geq -\frac{4M^2 x}{p} \left( \frac{1}{3} + \theta^{-1} \right)$$

### 3.4.2 Proof of Theorem 7

Let $\widetilde{\tau} \in \operatorname{argmin}_{\tau \in \mathcal{T}_n} \left\{ \widetilde{\mathcal{R}}_n(\tau) + \operatorname{pen}(\tau) \right\}$ that means that

$$\forall \tau \in \mathcal{T}_n : \quad \widetilde{\mathcal{R}}_n(\widetilde{\tau}) + \operatorname{pen}(\widetilde{\tau}) \leq \widetilde{\mathcal{R}}_n(\tau) + \operatorname{pen}(\tau)$$

$$\forall \tau \in \mathcal{T}_n : \quad \frac{1}{1+\theta} \widetilde{\mathcal{R}}_n(\widetilde{\tau}) + \frac{1}{1+\theta} \operatorname{pen}(\widetilde{\tau}) \leq \frac{1}{1+\theta} \widetilde{\mathcal{R}}_n(\tau) + \frac{1}{1+\theta} \operatorname{pen}(\tau) \tag{3.8}$$

Therfore,

$$\mathcal{R}\left(\widehat{\mu}_{\widetilde{\tau}}\right) + \frac{1}{1+\theta} \operatorname{pen}(\widetilde{\tau}) - \operatorname{pen}_{\mathrm{MC}}(\widetilde{\tau}) \leq \mathcal{R}\left(\widehat{\mu}_\tau\right) + \frac{1}{1+\theta} \operatorname{pen}(\tau) - \operatorname{pen}_{\mathrm{MC}}(\tau) \tag{3.9}$$

where

$$\forall \tau \in \mathcal{T}_n, \quad \text{pen}_{\text{MC}}(\tau) := \mathcal{R}\left(\widehat{\mu}_\tau\right) - \frac{1}{1+\theta}\widetilde{\mathcal{R}}_n(\tau) + \frac{1}{n}\|\varepsilon\|^2 \tag{3.10}$$

The new MC penalty can be written as : $\text{pen}_{\text{MC}}(\tau) = \mathcal{R}\left(\widehat{\mu}_\tau\right) - \widehat{\mathcal{R}}_n(\tau) - \frac{1}{1+\theta}\Delta_\tau + \frac{1}{n}\|\varepsilon\|^2$ where

$$\Delta_\tau := \widetilde{\mathcal{R}}_n(\tau) - (1+\theta)\widehat{\mathcal{R}}_n(\tau)$$

The goal here is to find a concentration inequality for $\text{pen}_{\text{MC}}(\tau)$ for every $\forall \tau \in \mathcal{T}_n$.

$$\begin{aligned}
\text{pen}_{\text{MC}}(\tau) &= \frac{1}{n}\left[\|\widehat{\mu}_\tau - \mu^\star\|^2 - \|\widehat{\mu}_\tau - Y\|^2 + \|\varepsilon\|^2\right] - \frac{1}{1+\theta}\Delta_\tau \\
&= \frac{1}{n}\left[\|\widehat{\mu}_\tau - \mu^\star\|^2 - \|\widehat{\mu}_\tau - \mu^\star - \varepsilon\|^2 + \|\varepsilon\|^2\right] - \frac{1}{1+\theta}\Delta_\tau \\
&= \frac{2}{n}\left\langle \widehat{\mu}_\tau - \mu^\star, \varepsilon\right\rangle - \frac{1}{1+\theta}\Delta_\tau \\
&= \frac{2}{n}\left\langle \Pi_\tau\left(\mu^\star + \varepsilon\right) - \mu^\star, \varepsilon\right\rangle - \frac{1}{1+\theta}\Delta_\tau \\
&= \frac{2}{n}\left\langle \Pi_\tau \mu^\star - \mu^\star, \varepsilon\right\rangle + \frac{2}{n}\left\langle \Pi_\tau \varepsilon, \varepsilon\right\rangle - \frac{1}{1+\theta}\Delta_\tau \\
&= \frac{2}{n}\left\langle \Pi_\tau \mu^\star - \mu^\star, \varepsilon\right\rangle + \frac{2}{n}\left\|\Pi_\tau \varepsilon\right\|^2 - \frac{1}{1+\theta}\Delta_\tau
\end{aligned}$$

Therefore,

$$\text{pen}_{\text{MC}}(\tau) = -\frac{2}{n}\left\langle (I - \Pi_\tau)\mu^\star, \varepsilon\right\rangle + \frac{2}{n}\left\|\Pi_\tau \varepsilon\right\|^2 - \frac{1}{1+\theta}\Delta_\tau \tag{3.11}$$

We can also put the $\text{pen}_{\text{MC}}(\tau)$ in another form :

$$\text{pen}_{\text{MC}}(\tau) = -\frac{2}{n}\left\langle (I - \Pi_\tau)\mu^\star, \varepsilon\right\rangle + \frac{2}{n}\left\|\Pi_\tau \varepsilon\right\|^2 - \frac{1}{1+\theta}\Delta'_\tau + \frac{2\theta}{1+\theta}\widehat{\mathcal{R}}_n(\tau) \tag{3.12}$$

where $\Delta'_\tau =:= \widetilde{\mathcal{R}}_n(\tau) - (1-\theta)\widehat{\mathcal{R}}_n(\tau)$.

Each one of the three propositions provides a concentration inequality for each term of the MC penalty.

Under Assumptions 2 and 3, a direct application of Proposition 1 gives with probability $\geq 1 - \mathrm{e}^{-x}$ for every $x > 0$ and $\forall \tau \in \mathcal{T}_n$ :

$$\frac{2}{n}\left\|\Pi_\tau \varepsilon\right\|^2 \leq \frac{2}{n}\left(\mathbb{E}\left[\left\|\Pi_\tau \varepsilon\right\|^2\right] + \frac{14M^2}{3}(x + 2\sqrt{2xD_\tau})\right) \tag{3.13}$$

**Computation of $\mathbb{E}\left[\left\|\Pi_\tau \varepsilon\right\|^2\right]$ :**

The computation of $\mathbb{E}\left[\left\|\Pi_\tau \varepsilon\right\|^2\right]$ is detailed in Celisse et al. [2012]. We start by providing a formula for $\left\|\Pi_\tau \varepsilon\right\|^2$.

$$\|\Pi_\tau \varepsilon\|^2 = \sum_{\ell=1}^{D_\tau} \left[ \frac{1}{\tau_\ell - \tau_{\ell-1}} \left\| \sum_{i=\tau_{\ell-1}+1} \varepsilon_i \right\|_{\mathcal{H}}^2 \right]$$
$$= \sum_{\ell=1}^{D_\tau} \left[ \frac{1}{\tau_\ell - \tau_{\ell-1}} \sum_{\tau_{\ell-1}+1 \le i,j \le \tau_\ell} \langle \varepsilon_i, \varepsilon_j \rangle_{\mathcal{H}} \right]$$

And, $\forall i, j \in \{1, ..., n\}$ we have :

$$\begin{aligned}
\mathbb{E}\left[\langle \varepsilon_i, \varepsilon_j \rangle_{\mathcal{H}}\right] &= \mathbb{E}\left[\left\langle \Phi(X_i) - \mu_i^\star, \Phi(X_j) - \mu_j^\star \right\rangle_{\mathcal{H}}\right] \\
&= \mathbb{E}\left[\langle \Phi(X_i), \Phi(X_j) \rangle_{\mathcal{H}}\right] - \mathbb{E}\left[\langle \mu_i^\star, \Phi(X_j) \rangle_{\mathcal{H}}\right] - \mathbb{E}\left[\left\langle \Phi(X_i), \mu_j^\star \right\rangle_{\mathcal{H}}\right] + \left\langle \mu_i^\star, \mu_j^\star \right\rangle_{\mathcal{H}} \\
&= \mathbb{E}\left[\langle \Phi(X_i), \Phi(X_j) \rangle_{\mathcal{H}}\right] - \langle \mu_i^\star, \mathbb{E}[Y_j] \rangle_{\mathcal{H}} - \left\langle \mathbb{E}[Y_i], \mu_j^\star \right\rangle_{\mathcal{H}} + \left\langle \mu_i^\star, \mu_j^\star \right\rangle_{\mathcal{H}} \\
&= \mathbb{E}\left[\langle \Phi(X_i), \Phi(X_j) \rangle_{\mathcal{H}}\right] - \left\langle \mu_i^\star, \mu_j^\star \right\rangle_{\mathcal{H}} \\
&= \mathbf{1}_{i=j} \left( \mathbb{E}[k(X_i, X_i)] - \|\mu_i^\star\|_{\mathcal{H}}^2 \right) = \mathbf{1}_{i=j} v_i
\end{aligned}$$

So, by combining the two results we get :

$$\mathbb{E}\left[\|\Pi_\tau \varepsilon\|^2\right] = \sum_{\ell=1}^{D_\tau} \left[ \frac{1}{\tau_\ell - \tau_{\ell-1}} \sum_{i=\tau_{\ell-1}+1}^{\tau_\ell} v_i \right]$$

This means that

$$\mathbb{E}\left[\|\Pi_\tau \varepsilon\|^2\right] \le \sum_{\ell=1}^{D_\tau} \left[ \frac{1}{\tau_\ell - \tau_{\ell-1}} \sum_{i=\tau_{\ell-1}+1}^{\tau_\ell} v_{max} \right] \le D_\tau M^2$$

By pluging this inequality in 3.13, we then have with probability $\ge 1 - \mathrm{e}^{-x}$ for every $x > 0$ and $\forall \tau \in \mathcal{T}_n$ :

$$\frac{2}{n} \|\Pi_\tau \varepsilon\|^2 \le \frac{2M^2}{n} \left( D_\tau + \frac{14}{3} x + \frac{28}{3} \sqrt{2 x D_\tau}) \right) \tag{3.14}$$

Under Assumptions 2 and 3 and by Proposition 2, for every $x > 0$ and for every $\tau \in \mathcal{T}_n$, with probability at least $1 - 2\mathrm{e}^{-x}$ we have :

$$\forall \eta > 0, \quad \frac{2}{n} |\langle (I - \Pi_\tau) \mu^\star, \varepsilon \rangle| \le \frac{2\eta}{n} \|\Pi_\tau \mu^\star - \mu^\star\|^2 + \frac{2}{n} \left( \frac{v_{\max}}{2\eta} + \frac{4M^2}{3} \right) x$$

Therfore,

$$\forall \eta > 0, \quad \frac{2}{n} |\langle (I - \Pi_\tau) \mu^\star, \varepsilon \rangle| \le \frac{2\eta}{n} \|\Pi_\tau \mu^\star - \mu^\star\|^2 + \frac{x M^2}{n} \left( \eta^{-1} + \frac{8}{3} \right) \tag{3.15}$$

According to Proposition 3, for every $x > 0$ and for every $\tau \in \mathcal{T}_n$, with probability at least $1 - \mathrm{e}^{-x}$ we have :

$$-\Delta'_\tau \leq \frac{4M^2 x}{p}\left(\frac{1}{3} + \theta^{-1}\right) \tag{3.16}$$

and for every $x > 0$ and for every $\tau \in \mathcal{T}_n$, with probability at least $1 - \mathrm{e}^{-x}$ we have :

$$-\Delta_\tau \geq -\frac{4M^2 x}{p}\left(\frac{1}{3} + \theta^{-1}\right) \tag{3.17}$$

For $\tau \in \mathcal{T}_n$ and for $x > 0$ let $\Omega_x^\tau$ be the event on which Equations 3.14,3.15, 3.16 and 3.17 hold true. Using $\mathbb{P}\left(A \cap B \cap C \cap D\right) \geq 1 - \mathbb{P}\left(\bar{A}\right) - \mathbb{P}\left(\bar{B}\right) - \mathbb{P}\left(\bar{C}\right) - \mathbb{P}\left(\bar{D}\right)$ we get that $\mathbb{P}\left(\Omega_x^\tau\right) \geq 1 - 5\mathrm{e}^{-x}$. So, combining Equations 3.11, 3.14, 3.15 and 3.16 shows that on $\Omega_x^\tau$ and for every $\eta > 0$ we have :

$$\begin{aligned}\mathrm{pen}_{\mathrm{MC}}(\tau) \leq &\left[\frac{2M^2}{n}\left(D_\tau + \frac{14x}{3} + \frac{28}{3}\sqrt{2xD_\tau}\right) + \frac{2\eta}{n}\left\|\Pi_\tau \mu^\star - \mu^\star\right\|^2 + \frac{xM^2}{n}\left(\eta^{-1} + \frac{8}{3}\right)\right] \\ &+ \frac{4M^2 x}{p}\left(\frac{\theta+3}{3\theta(\theta+1)}\right) + \frac{2\theta}{1+\theta}\widehat{\mathcal{R}}_n\left(\tau\right)\end{aligned}$$

By definition of the orthogonal projection $\Pi_\tau$ we have

$$\frac{1}{n}\left\|\Pi_\tau \mu^\star - \mu^\star\right\|^2 = \mathcal{R}(\Pi_\tau \mu^\star) \leq \mathcal{R}(\widehat{\mu}_\tau)$$

So the previous equation becomes :

$$\begin{aligned}\mathbf{pen_{MC}(\tau)} \leq 2\eta\mathcal{R}\left(\widehat{\mu}_\tau\right) &+ \left[\frac{2M^2}{n}\left(D_\tau + \frac{14x}{3} + \frac{28}{3}\sqrt{2xD_\tau}\right) + \frac{xM^2}{n}\left(\eta^{-1} + \frac{8}{3}\right)\right] \\ &+ \frac{4M^2 x}{p}\left(\frac{\theta+3}{3\theta(\theta+1)}\right) + \frac{2\theta}{1+\theta}\widehat{\mathcal{R}}_n\left(\tau\right)\end{aligned} \tag{3.18}$$

On the other hand, we have also on $\Omega_x^\tau$ :

$$\begin{aligned}\mathrm{pen}_{\mathrm{MC}}(\tau) &\geq -\frac{2}{n}\left\langle\left(I - \Pi_\tau\right)\mu^\star, \varepsilon\right\rangle - \frac{1}{1+\theta}\Delta_\tau \\ &\geq -\frac{2\eta}{n}\left\|\Pi_\tau \mu^\star - \mu^\star\right\|^2 - \frac{xM^2}{n}\left(\eta^{-1} + \frac{8}{3}\right) - \frac{4M^2 x}{p}\left(\frac{\theta+3}{3\theta(\theta+1)}\right) \\ &\geq -2\eta\mathcal{R}\left(\widehat{\mu}_\tau\right) - \frac{xM^2}{n}\left(\eta^{-1} + \frac{8}{3}\right) - \frac{4M^2 x}{p}\left(\frac{\theta+3}{3\theta(\theta+1)}\right)\end{aligned}$$

So, on $\Omega_x^\tau$:

$$\mathbf{pen_{MC}(\tau)} \geq -2\eta\mathcal{R}\left(\widehat{\mu}_\tau\right) - \frac{xM^2}{n}\left(\eta^{-1} + \frac{8}{3}\right) - \frac{4M^2 x}{p}\left(\frac{\theta+3}{3\theta(\theta+1)}\right) \tag{3.19}$$

**Union bound and conclusion :**

Let $y > 0$ and let $\Omega_y$ be the event $\Omega_y = \cap_{\tau \in \mathcal{T}_n} \Omega^\tau_{x(y,\tau)}$ where $x(y,\tau) = y + \log(\frac{5}{e-1}) + D_\tau + \log\binom{n-1}{D_\tau - 1}$.

$$\begin{aligned}
\mathbb{P}(\Omega_y) &\geq 1 - \sum_{\tau \in \mathcal{T}_n} \mathbb{P}(\bar{\Omega}^\tau_{x(y,\tau)}) \\
&= 1 - 5 \sum_{\tau \in \mathcal{T}_n} \mathrm{e}^{x(y,\tau)} \\
&= 1 - 5 \sum_{D=1}^{n} \mathrm{Card}\left\{\tau \in \mathcal{T}_n | D_\tau = D\right\} \exp\left[-y - \log(\frac{5}{e-1}) - D - \log\binom{n-1}{D-1}\right]
\end{aligned}$$

We know that : $\mathrm{Card}\left\{\tau \in \mathcal{T}_n | D_\tau = D\right\} = \binom{n-1}{D_\tau - 1}$
So,

$$\mathbb{P}(\Omega_y) \geq 1 - (e-1)\mathrm{e}^{-y} \sum_{D=1}^{n} \mathrm{e}^{-D} \geq 1 - \mathrm{e}^{-y}$$

In addition, on $\Omega_y$ and for every $\tau \in \mathcal{T}_n$, since 3.18 and 3.19 hold true with $x = x(y,\tau) \geq D_\tau$, taking $\eta = 1/6$, we get :

$$-\frac{1}{3}\mathcal{R}(\widehat{\mu}_\tau) - x(y,\tau)M^2\left[\frac{26}{3n} + \frac{4}{p}\left(\frac{\theta+3}{3\theta(\theta+1)}\right)\right] \leq \mathbf{pen_{MC}(\tau)}$$

and

$$\mathbf{pen_{MC}(\tau)} \leq \frac{1}{3}\mathcal{R}(\widehat{\mu}_\tau) + x(y,\tau)M^2\left[\frac{k}{n} + \frac{4}{p}\left(\frac{\theta+3}{3\theta(\theta+1)}\right)\right] + \frac{2\theta}{1+\theta}\widehat{\mathcal{R}}_n(\tau)$$

where $\kappa = \frac{60+56\sqrt{2}}{3}$

Let $\kappa_\theta = \max\left[\kappa, 4\left(\frac{\theta+3}{3\theta(\theta+1)}\right)\right]$ and $\kappa'_\theta = \max\left[\frac{26}{3}, 4\left(\frac{\theta+3}{3\theta(\theta+1)}\right)\right]$. Then,

$$-\frac{1}{3}\mathcal{R}(\widehat{\mu}_\tau) - \kappa'_\theta x(y,\tau)M^2\left[\frac{1}{n} + \frac{1}{p}\right] \leq \mathbf{pen_{MC}(\tau)}$$

and

$$\mathbf{pen_{MC}(\tau)} \leq \frac{1}{3}\mathcal{R}(\widehat{\mu}_\tau) + \kappa_\theta x(y,\tau)M^2\left[\frac{1}{n} + \frac{1}{p}\right] + \frac{2\theta}{1+\theta}\widehat{\mathcal{R}}_n(\tau)$$

Let's try to bound $\frac{2\theta}{1+\theta}\widehat{\mathcal{R}}_n(\tau)$.

$$\begin{aligned}
\widehat{\mathcal{R}}_n(\tau)^2 &= \left(\frac{1}{n}\sum_{i=1}^{n} k(X_i,X_i) - \frac{1}{n}\sum_{\ell=1}^{D}\frac{1}{\tau_\ell-\tau_{\ell-1}}\sum_{i,j=\tau_{\ell-1}+1}^{\tau_\ell} k(X_i,X_j)\right)^2 \\
&\leq 2\left(\frac{1}{n}\sum_{i=1}^{n} k(X_i,X_i)\right)^2 + 2\left(\frac{1}{n}\sum_{\ell=1}^{D}\sum_{i,j=\tau_{\ell-1}+1}^{\tau_\ell}\frac{1}{\sqrt{\tau_\ell-\tau_{\ell-1}}}\frac{k(X_i,X_j)}{\sqrt{\tau_\ell-\tau_{\ell-1}}}\right)^2 \\
&\leq 2\frac{1}{n}\sum_{i=1}^{n} k(X_i,X_i)^2 + 2\frac{1}{n^2}\left(\sum_{\ell=1}^{D}\sum_{i,j=\tau_{\ell-1}+1}^{\tau_\ell}\frac{1}{\tau_\ell-\tau_{\ell-1}}\right) \\
&\times\left(\sum_{\ell=1}^{D}\sum_{i,j=\tau_{\ell-1}+1}^{\tau_\ell}\frac{1}{\tau_\ell-\tau_{\ell-1}}k(X_i,X_j)^2\right) \\
&\leq 4M^4
\end{aligned}$$

Therefore, we get :

$$-\frac{1}{3}\mathcal{R}(\widehat{\mu}_\tau) - \kappa'_\theta x(y,\tau)M^2\left[\frac{1}{n}+\frac{1}{p}\right] \leq \mathbf{pen_{MC}(\boldsymbol{\tau})} \tag{3.20}$$

and

$$\mathbf{pen_{MC}(\boldsymbol{\tau})} \leq \frac{1}{3}\mathcal{R}(\widehat{\mu}_\tau) + \kappa_\theta x(y,\tau)M^2\left[\frac{1}{n}+\frac{1}{p}\right] + 4\theta M^2 \tag{3.21}$$

Let's assume that $C \geq (1+\theta)k_\theta$, since $\mathrm{pen}(\tau) \geq CM^2\left[\frac{1}{n}+\frac{1}{p}\right]\left[\log\begin{pmatrix} n-1 \\ D_\tau-1\end{pmatrix} + D_\tau\right]$ we have :

$$\mathbf{pen_{MC}(\boldsymbol{\tau})} \leq \frac{1}{3}\mathcal{R}\left(\widehat{\mu}_\tau\right) + \frac{1}{1+\theta}\,\mathrm{pen}(\tau) + \kappa_\theta M^2[y+\log(\frac{5}{\mathrm{e}-1})]\left[\frac{1}{n}+\frac{1}{p}\right] + 4\theta M^2 \tag{3.22}$$

$$\boldsymbol{\mathcal{R}\left(\widehat{\mu}_\tau\right) - \mathbf{pen_{MC}(\boldsymbol{\tau})}} \leq \frac{4}{3}\mathcal{R}\left(\widehat{\mu}_\tau\right) + \frac{\kappa'_\theta}{C}\,\mathrm{pen}(\tau) + \kappa'_\theta M^2[y+\log(\frac{5}{\mathrm{e}-1})]\left[\frac{1}{n}+\frac{1}{p}\right]$$

According to Equation 3.9,

$$\begin{aligned}
\mathcal{R}\left(\widetilde{\mu}_{\widetilde{\tau}}\right) + \frac{1}{1+\theta}\,\mathrm{pen}(\widetilde{\tau}) - \mathrm{pen}_{\mathrm{MC}}(\widetilde{\tau}) - \frac{1}{1+\theta}\,\mathrm{pen}(\tau) &\leq \frac{4}{3}\mathcal{R}\left(\widehat{\mu}_\tau\right) + \frac{\kappa'_\theta}{C}\,\mathrm{pen}(\tau) \\
&+\kappa'_\theta M^2[y+\log(\frac{5}{\mathrm{e}-1})]\left[\frac{1}{n}+\frac{1}{p}\right]
\end{aligned}$$

If we apply 3.22 to $\widetilde{\tau}$ we then have on $\Omega_y$ and for every $\tau \in \mathcal{T}_n$:

$$\begin{aligned}
\frac{2}{3}\mathcal{R}\left(\widehat{\mu}_{\widetilde{\tau}}\right) - \kappa_\theta M^2[y+\log(\frac{5}{\mathrm{e}-1})]\left[\frac{1}{n}+\frac{1}{p}\right] - 4\theta M^2 &\leq \frac{4}{3}\mathcal{R}\left(\widehat{\mu}_\tau\right) + \left(\frac{1}{1+\theta}+\frac{\kappa'_\theta}{C}\right)\mathrm{pen}(\tau) \\
&+\kappa'_\theta M^2[y+\log(\frac{5}{\mathrm{e}-1})]\left[\frac{1}{n}+\frac{1}{p}\right]
\end{aligned}$$

Therefore,

$$\mathcal{R}\left(\widehat{\mu}_{\widetilde{\tau}}\right) \leq 2\mathcal{R}\left(\widehat{\mu}_\tau\right) + \frac{3}{2}\left(1+\frac{\kappa'_\theta}{C}\right)\mathrm{pen}(\tau) + \frac{3}{2}(\kappa_\theta+\kappa'_\theta)M^2[y+\log(\frac{5}{\mathrm{e}-1})]\left[\frac{1}{n}+\frac{1}{p}\right] + 6\theta M^2$$

$$\mathcal{R}\left(\widehat{\mu}_{\widetilde{\tau}}\right) \leq 2\mathcal{R}\left(\widehat{\mu}_{\tau}\right) + \frac{3}{2}\left(1 + \frac{\kappa'_\theta + (\kappa_\theta + \kappa'_\theta)\log(\frac{5}{\mathrm{e}-1})}{C}\right)\mathrm{pen}(\tau) + \frac{3}{2}(\kappa_\theta + \kappa'_\theta)M^2 y\left[\frac{1}{n} + \frac{1}{p}\right] + 6\theta M^2$$

because $\mathrm{pen}(\tau) \geq CM^2\left[\frac{1}{n} + \frac{1}{p}\right]$ for every $\tau \in \mathcal{T}_n$.

Let's define

$$A_\theta := 3\left[\kappa'_\theta + (\kappa_\theta + \kappa'_\theta)\log(\frac{5}{\mathrm{e}-1})\right]$$

so that

$$\frac{3}{2}\left(1 + \frac{\kappa'_\theta + (\kappa_\theta + \kappa'_\theta)\log(\frac{5}{\mathrm{e}-1})}{A_\theta}\right) = 2$$

Then if $C \geq L_\theta := \max(A_\theta, (1+\theta)\kappa_\theta)$, we have for every $\tau \in \mathcal{T}_n$ on $\Omega_y$ :

$$\mathcal{R}\left(\widehat{\mu}_{\widetilde{\tau}}\right) \leq 2\mathcal{R}\left(\widehat{\mu}_{\tau}\right) + 2\,\mathrm{pen}(\tau) + \frac{3}{2}(\kappa_\theta + \kappa'_\theta)M^2 y\left[\frac{1}{n} + \frac{1}{p}\right] + 6\theta M^2$$

Finally, we get on $\Omega_y$

$$\mathcal{R}\left(\widehat{\mu}_{\widetilde{\tau}}\right) \leq 2\inf_{\tau\in\mathcal{T}_n}\left\{\mathcal{R}\left(\widehat{\mu}_{\tau}\right) + \mathrm{pen}(\tau)\right\} + \frac{3}{2}(\kappa_\theta + \kappa'_\theta)M^2 y\left[\frac{1}{n} + \frac{1}{p}\right] + 6\theta M^2 \tag{3.23}$$

with $\mathbb{P}(\Omega_y) \geq 1 - \mathrm{e}^{-y}$ which is the desired result of Theorem 7 with
$L_\theta = \max(A_\theta, (1+\theta)\kappa_\theta)$ and $B_\theta = \frac{3}{2}(\kappa_\theta + \kappa'_\theta)$.

This achieves the proof of Theorem 7.
The proof of Theorem 6 follows the same logic. It is simpler actually. Since we have a concentration inequality of type Hoeffding, by defining the $\mathrm{pen}_{MC}$ as :

$$\forall\tau \in \mathcal{T}_n, \quad \mathrm{pen}_{\mathrm{MC}}(\tau) := \mathcal{R}\left(\widehat{\mu}_{\tau}\right) - \widetilde{\mathcal{R}}_n(\tau) + \frac{1}{n}\|\varepsilon\|^2 \tag{3.24}$$

we can follow exactely the same logic of the previous proof, exploiting the Hoeffding inequality of $\widetilde{\mathcal{R}}_n(\tau)$. We should expect a term of $\frac{1}{\sqrt{p}}$ instead of $\frac{1}{p}$ in the final Oracle Inequality, which is a weaker result in terms of number of features $p$, but we will get ride of the bias term $6\theta M^2$ which will enable us the control the error independently of the constant $\theta$.

## 3.5 What about Nystrom Approximation ?

In order to obtain a similar result with the Nystrom approximation, the goal is to look for a concentration inequality of the approximated least square criterion $\widetilde{\mathcal{R}}_n(\tau)$, then the same proof would apply to get an Oracle inequality. So, how can we bound the difference $\boldsymbol{\widetilde{\mathcal{R}}_n(\tau) - \widehat{\mathcal{R}}_n(\tau)}$ ?

Let $\tilde{G}_k$ be the rank $k$ approximation of the Gram matrix $G$ according to the Nystrom procedure. Let's denote the elements of $\tilde{G}_k$ by $\widetilde{K}_{ij}$ i.e $\tilde{G}_k = \left[\widetilde{K}_{ij}\right]_{i,j\leq n}$. We remind the Frobenius norm of a marix M : $\|M\|_F^2 = \sum_{i,j=1}^n M_{ij}^2$

$$\begin{aligned}
\left|\widetilde{\mathcal{R}}_n(\tau) - \widehat{\mathcal{R}}_n(\tau)\right|^2 &= \left(\frac{1}{n}\sum_{i=1}^{n}\left[\widetilde{K}_{ii} - k(X_i, X_i)\right] - \frac{1}{n}\sum_{\ell=1}^{D_\tau}\frac{1}{\tau_\ell - \tau_{\ell-1}}\sum_{i,j=\tau_{\ell-1}+1}^{\tau_\ell}\left[\widetilde{K}_{ij} - k(X_i, X_j)\right]\right)^2 \\
&\leq 2\left(\frac{1}{n}\sum_{i=1}^{n}\left[\widetilde{K}_{ii} - k(X_i, X_i)\right]\right)^2 + 2\left(\frac{1}{n}\sum_{\ell=1}^{D_\tau}\sum_{i,j}^{\tau_\ell}\frac{1}{\tau_\ell - \tau_{\ell-1}}\left[\widetilde{K}_{ij} - k(X_i, X_j)\right]\right)^2 \\
&\leq 2\frac{1}{n}\sum_{i=1}^{n}\left[\widetilde{K}_{ii} - k(X_i, X_i)\right]^2 + 2\frac{1}{n^2}\left(\sum_{\ell=1}^{D_\tau}\sum_{i,j=\tau_{\ell-1}+1}^{\tau_\ell}\frac{1}{(\tau_\ell - \tau_{\ell-1})^2}\right) \\
&\times \left(\sum_{\ell=1}^{D_\tau}\sum_{i,j=\tau_{\ell-1}+1}^{\tau_\ell}\left[\widetilde{K}_{ij} - k(X_i, X_j)\right]^2\right) \\
&\leq \frac{2}{n}\left\|G - \tilde{G}_k\right\|_F^2 + \frac{2D_\tau}{n^2}\left\|G - \tilde{G}_k\right\|_F^2 = 2\left\|G - \tilde{G}_k\right\|_F^2\left(\frac{1}{n} + \frac{D_\tau}{n^2}\right)
\end{aligned}$$

According to Theorem 2, with probability higher than $1 - \mathrm{e}^{-x}$ we have :

$$\left\|G - \tilde{G}_k\right\|_F \leq \|G - G_k\|_F + \sqrt[4]{\frac{64k(1+\sqrt{8x})^2}{p}}\sum_{i=1}^{n} G_{ii}^2$$

where $p$ is the number of sampled columns (or the number of features).

By plugging the previous result in this inequality, we get, with a probability higher than $1 - \mathrm{e}^{-x}$ :

$$\left|\widetilde{\mathcal{R}}_n(\tau) - \widehat{\mathcal{R}}_n(\tau)\right| \leq \sqrt{2\left(\frac{1}{n} + \frac{D_\tau}{n^2}\right)}\left(\|G - G_k\|_F + \sqrt[4]{\frac{64k(1+\sqrt{8x})^2}{p}}\sum_{i=1}^{n} G_{ii}^2\right) \tag{3.25}$$

We have not been so tight is our inequalities, especially with the use of Cauchy-Schwarz inequality to bound the least square criterion. We believe we can do better because if we analyze this result, we can see that in the best case scenario,

$$\left|\widetilde{\mathcal{R}}_n(\tau) - \widehat{\mathcal{R}}_n(\tau)\right| \leq \mathcal{O}\left(\sqrt[4]{\frac{n^2 k}{p}}\right)$$

which is not optimal.
We can also work under the eigengap assumtion to exploit the improved result of Mahdavi et al. [2012], we would have then,

$$\left|\widetilde{\mathcal{R}}_n(\tau) - \widehat{\mathcal{R}}_n(\tau)\right| \leq \mathcal{O}\left(\sqrt{\frac{n}{p}}\right)$$

## 3.6 Perspectives

### 3.6.1 Guarentees on the distance between segmentations

In Theorem 7, we proved an Oracle inequality on the quadratic risk of the best least square estimator of $\mu^\star$, $\mathcal{R}(\widehat{\mu})$. This inequality holds several abstract quantites, we could have tried to prove with similar techniques, a direct inequality on a mesure of distance between the estimated semgentation $\widehat{\tau}$ and the true segmentation $\tau^\star$.
Such inequality would provide direct information on the error that the low rank approximations impose.

The possibility of the existence of such an inequality comes from the fact that Garreau et al. [2018] proved a similar theorem for KCP with penalty outside the low rank approximations framework.

To present Garreau's theorem, we need to define new quantities.
Let $\Delta$ be the size of the smallest jump of $\mu^\star$ in $\mathcal{H}$.

$$\Delta := \min_{i/\mu_i^* \neq \mu_{i+1}^\star} \left\|\mu_i^\star - \mu_{i+1}^\star\right\|_{\mathcal{H}}$$

For any $\tau \in \mathcal{T}_n$, we denote the (normalized) size of its smallest segment by

$$\underline{\Lambda}_\tau := \frac{1}{n} \min_{1 \leq \ell \leq D_\tau} |\tau_\ell - \tau_{\ell-1}|$$

**Theorem 9** *Suppose that Assumption 3 holds true. For any $y > 0$, an event $\Omega$ of probability at least $1 - \mathrm{e}^{-y}$ exists on which the following holds true. For any $C > 0$, let $\widehat{\tau}$ be defined by*

$$\widehat{\tau} \in \min_{\tau \in \mathcal{T}_n} \{\widehat{\mathcal{R}}_n(\tau) + \mathrm{pen}(\tau)\}$$

*with $pen = \mathrm{pen}_\ell$ defined by*

$$\mathrm{pen}_\ell(\tau) := \frac{CM^2 D_\tau}{n}$$

*Set*

$$C_{\min} := \frac{74}{3}\left(D^\star + 1\right)\left(y + \log(n) + 1\right) \quad \textit{and} \quad C_{\max} := \frac{\Delta^2}{M^2}\frac{\underline{\Lambda}_{\tau^*}}{6D^\star} n$$

*Then, if*

$$C_{\min} < C < C_{\max}$$

*on $\Omega$, we have*

$$\widehat{D} = D^\star \quad \textit{and} \quad \mathrm{d}_{\mathrm{F}}^2\left(\tau^\star, \widehat{\tau}\right) \leq \frac{1849 D^{\star 2}}{\underline{\Lambda}_{\tau^*}} \cdot \frac{M^2}{\Delta^2} \cdot \frac{y + \log(n) + 1}{n}$$

$\mathrm{d}_{\mathrm{F}}$ is the Frobenius distance defined in Chapter 2.

The techniques used in the proof of this theorem are very similar to the one we used to prove our main theorem. The low rank approximations still present some new quantities that make the adaptation of his proof harder than it was expected.
For example, the fact that $\widehat{D} = D^\star$ plays a big role in Garreau's proof. This result can't be guarenteed

in the presence of a low rank approximation method, hence we could not go through with the proof when we were trying.

We think that with more time, it could be possible to surpass this obstacle and prove a similar result in the low rank approximation framework, a result that involves the parameter of the number of features and from which we can draw some insights on how the approximations affects the error between the estimated and the true segmentation.

### 3.6.2 Optimal Statistical performances with Random Features

In the second chapter we presented the KCP algorithms and explained how we can scale up the binary segmentation KCP algorithm in order to be able to handle datasets with $n \geq 10^5$. We saw that as long as the number of features $p = \mathcal{O}(\sqrt{n})$, we have a fast approximative algorithm.
Clearly, there is no free launch, this gain in computation comes with a loss in statistical performances. In our theoretical analysis, we managed to give a convergence guarentee with Theorems 7 and 6, but is order to guaarentee optimal statistical performances (Similar bounds as if there is no approximation), the two theorems propose to work with $p = \mathcal{O}(n^2)$ or $p = \mathcal{O}(n)$ in the best case scenario, which is far from the recommended $p = \mathcal{O}(\sqrt{n})$.

We think that it is possible to theoretically guarentee optimal statistical performances of KCP with low rank approximation, especially if using Random feaures.
The reason it is possible is beacause Rudi and Rosasco [2017] have succeeded to prove that for the Kernel Ridge Regression problem.

#### Ridge Regression

The simplest model for a regression problem is

$$f(x) = \sum_{i=1}^{d} \omega_i x_i, \text{ where } x = (x_1, ..., x_d) \in \mathbb{R}^d \tag{3.26}$$

Another class of models in Ridge Regression uses a feature map $\Phi(x) = \begin{pmatrix} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_M(x) \end{pmatrix}$ which is a map defined from $\mathbb{R}^d$ to $\mathbb{R}^m$. The problem becomes:

$$f(x) = \sum_{i=1}^{m} \omega_i \phi_i(x) = \mathbf{w}^\top . \Phi(x) \text{ where } x = (x_1, ..., x_d) \in \mathbb{R}^d \tag{3.27}$$

Besides having to find the parameters $\omega_i$ we know have to find the right feature map $\Phi$ to our problem.

The most common cost function that needs to be minimized is :

$$C(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^\top . \Phi(x_i))^2 + \lambda \|w\|_2^2 \tag{3.28}$$

where $n$ is the number of training data, $x_i$ are the data points and $\lambda$ is the regularization parameter.

The $x$ points live in a $d$ dimensional space whereas $\Phi(x)$ live in an $m$ dimensional space.
Let $\Phi$ be the matrix :

$$\Phi = \begin{pmatrix} \phi_1(x_1) & \phi_1(x_2) & \dots & \phi_1(x_n) \\ \phi_2(x_1) & \phi_2(x_2) & \dots & \phi_2(x_n) \\ \vdots & & \ddots & \vdots \\ \phi_m(x_1) & \phi_m(x_2) & \dots & \phi_m(x_n) \end{pmatrix} \in \mathbb{R}^{m \times n} \tag{3.29}$$

and $y$ be the vector $\boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$.
By calculating the differential of the cost function, we find a solution to the problem that can be written as :

$$\mathbf{w} = (\lambda I_m + \Phi\Phi^\top)^{-1}\Phi.y \tag{3.30}$$

**Kernel Ridge Regression**

In the previous equation, we have to perform an inverse on a $m \times m$ Matrix.
There is a trick that allows us to perform the inverse in the smallest space, either $\mathcal{M}_m(\mathbb{R})$ or $\mathcal{M}_n(\mathbb{R})$.
The trick is given by the following identity :

$$(P^{-1} + B^\top R^{-1} B)^{-1} B^\top R^{-1} = PB^\top (BPB^\top + R)^{-1} \tag{3.31}$$

We apply the formula to $R = I_n$, $P = \frac{1}{\lambda} I_m$ and $B = \Phi^\top$. We get :

$$\mathbf{w} = \Phi(\lambda I_n + \Phi^\top \Phi)^{-1}.y \tag{3.32}$$

So, the estimation of the function $f$ is written as :

$$\hat{f}_{\lambda,m}(x) = \mathbf{w}^\top \phi(x) = y^\top.(\lambda I_n + \Phi^\top \Phi)^{-1}.\Phi^T.\Phi(x) \tag{3.33}$$

$\Phi^T\Phi$ is a $n \times n$ matrix. Let's name it $G$.

$$G = \left[\Phi(x_i)^\top \Phi(x_j)\right]_{0 \le i,j \le n} \tag{3.34}$$

We also have $\Phi^T\Phi(x) = \begin{pmatrix} \Phi(x_1)^\top \Phi(x) \\ \Phi(x_2)^\top \Phi(x) \\ \vdots \\ \Phi(x_n)^\top \Phi(x) \end{pmatrix}$
So, using these notations, we never actually need access to the feature vectors, which could be infinitely long (and impractical). We only need to have access to a function $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$.
In this case,

$$k(x, y) = \Phi(x)^\top \Phi(y) = \langle \Phi(x) | \Phi(y) \rangle. \tag{3.35}$$

which is a dot product in $\mathbb{R}^m$.

Finally,

$$\hat{f}_{\lambda,m}(x) = y^\top.(\lambda I_n + G)^{-1}. \begin{pmatrix} k(x_1, x) \\ k(x_2, x) \\ \vdots \\ k(x_n, x) \end{pmatrix} \tag{3.36}$$

$G$ the **Gram Matrix** associated to the data points and the kernel $k$.

So, the Ridge Regression problem implies that we work with a Kernel $k$ defined from $\mathbb{R}^n \times \mathbb{R}^n$ to $\mathbb{R}$. The form of the Kernel is defined above with a dot product and a family of functions $\phi_i$.
This form makes the kernel positive semi-definite
Until know we proved that **RR with a Feature Map** $\Rightarrow$ **RR with a PD Kernel**.

Fortunately, the other implication is guaranteed by the **Kernel trick**.

Therefore, **RR with a Feature Map** $\Leftrightarrow$ **RR with a PSD Kernel**.
So, kernel Ridge Regression is a regression problem to which the solution is of the form of Equation 3.36.

### Learning with Random Features

From Equation 3.30, we can deduce that solving the Ridge Regression problem requires, if $m \leq n$, $\mathcal{O}(m^2 + mn) = \mathcal{O}(mn)$ in space (to store $\Phi\Phi^\top$ and $\Phi$) and $\mathcal{O}(m^2 n + m^3)$ in time (to calculate the inverse and the product).
It is much better than $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$ which is what is required to achieve a $\mathcal{O}(\frac{1}{\sqrt{n}})$ learning bound.
So, as long as $m << n$ we can gain a lot in time and memory.

For kernels that can be written as $\langle \Phi(\cdot) | \Phi(\cdot) \rangle$, with $\Phi(\cdot) \in \mathbb{R}^m$ and $m << n$, we're OK.
But, only considering kernels in this form with a feature Map that lives in a finite dimensional space can be restrictive.
An example of kernels that do not satisfy this form is the Gaussian kernel, which is on of the most used kernel in Machine Learning.

So, how can we use this Kernel in Ridge Regression and at the same time reduce the computational complexity?

The solution proposed by Rudi and Rosasco [2017] is to relax the equation 3.35 assuming that it holds approximately

$$k(x, y) \approx \Phi_p(x)^\top \Phi_p(y) \tag{3.37}$$

where $\Phi_p(x) \in \mathbb{R}^p$ and $p \ll n$.

We have already seen this kind of approximation with the Random Fourier Features. Now, we need to know how to choose the number of features $p$ to guarentee optimal statistical performances.

We should first define what is a **Learning bound**.
The aim of every Machine Learning algorithm is to minimize the empirical risk (or the training error) which is not in itself a solution to the learning problem, it could only be considered a solution if we can guarantee that the difference between the training error and the generalization error is small enough with a certain probability.
This is formalized by :

$$\mathbb{P}\left[\mathcal{E}(\hat{f}_{\lambda,m}) - \mathcal{E}(f_{\mathcal{H}}) < \epsilon\right] \approx 1 \tag{3.38}$$

Where $\epsilon$ is a strictly positive number, $\mathcal{E}(\hat{f}_{\lambda,m})$ is the expected risk of the estimated function $\hat{f}_{\lambda,m}$ and $\mathcal{E}(f_{\mathcal{H}})$ is the expected risk of the real function $f_{\mathcal{H}}$ assuming it exists.
In general, the learning bound depends on the number of features selected to solve the problem.

A $\mathcal{O}(n)$ features give a $\mathcal{O}(\frac{1}{\sqrt{n}})$ learning bound in Ridge Regression.
The following theorem shows that with only $\mathcal{O}(\sqrt{n}\log n)$ features instead of $\mathcal{O}(n)$, we can obtain $\mathcal{O}(\frac{1}{\sqrt{n}})$ learning bound.

**Theorem 10** *Assume that $k$ is a kernel with an integral representation 2.2. Assume that $\psi$ is continuous such that $|\psi(x,\omega)| \leq k$ almost surely with $k \in [1,\infty)$ and $|y| \leq b$ almost surely, with $b > 0$. Let $\delta \in (0,1]$. if $n > n_0$ and $\lambda_n = n^{-1/2}$, then a number of random features $M_n$ equal to*

$$M_n = c_0\sqrt{n}\log\frac{108k^2\sqrt{n}}{\delta}$$

*is enough to guarantee. with probability at least $\delta$ that*

$$\mathcal{E}(\hat{f}_{\lambda_n,M_n}) - \mathcal{E}(f_{\mathcal{H}}) \leq \frac{c_1 \log\frac{18}{\delta}^2}{\sqrt{n}}$$

.
*In particular, the constants $c_0, c_1$ do not depend on $n, \lambda, \delta$ and $n_0$ does not depend on $n, \lambda, f_{\mathcal{H}}$ and $P_{X,y}$.*
*($P_{X,y}$ is the probability law according to which the data is sampled)*

This means that we go from $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$ in time and space to $\mathcal{O}(n^2)$ and $\mathcal{O}(n\sqrt{n})$ using Random Features without trading-off statistical accuracy.

This was short summary of the work that has been done to scale up kernel ridge regression without having to trade-off statistical accuracy.
We believe that it is possible to prove simlar results for KCP.

In their proof, Rudi and Rosasco [2017] exploited the regularity of the solutions in the RKHS $\mathcal{H}$ to draw their results.
In kernel change-point detection, the solution is always piecewise constant, so we know more about its regularity and we should be able to prove similar results.

This conjecture, in term of greatness and in term of difficulty to prove, would make a perfect objectif for a PhD thesis.

# Conclusion

In this work, we studied the impact of low rank approximation methods, mainly Nystrom approximation and Random features, on the computational time and statistical performances of kernel change-point detection algorithms.
We presented several guarentees for these approximation methods independently from the KCP framework, then we proved two more or less similar Oracle inequalities for KCP with low rank approximation, inspired from previous work of Celisse et al. [2012].

We also showed that the low rank approximations allow a linear time complexity compared to the quadratic original complexity of KCP algorithms, which will enable us to work with very large signal ($n \geq 10^5$).
Empirical studies have also been done on synthetic signals, they show that the impact of the approximation is not very perceptible especially for simple changes in the distribution (mean or variance). We hope in the future to experiment on real signals.

Our main contributions in this work are, first of all, a thorough dissection of the different KCP algorithms, either outside the approximation framework or in the presence of low rank approximation, second, the empirical study of the approximations using an existing **Python** module and a module of our own, and finally and most importantly, the theoretical analysis of low rank approximations in the KCP framework. We believe that the theoretical results are not yet optimal. This can be the subject of further research.

# Bibliography


Zaid Harchaoui and Olivier Cappé. Retrospective mutiple change-point estimation with kernels. In *2007 IEEE/SP 14th Workshop on Statistical Signal Processing*, pages 768–772. IEEE, 2007.

Alain Celisse, Sylvain Arlot, and Zaid Harchaoui. A kernel multiple change-point algorithm via model selection. *arXiv preprint arXiv:1202.3878*, 2012.

Damien Garreau. *Change-point detection and kernel methods*. PhD thesis, 2017.

Alain Celisse, Guillemette Marot, Morgane Pierre-Jean, and Guillem Rigaill. New efficient algorithms for multiple change-point detection with kernels. *arXiv preprint arXiv:1710.04556*, 2017.

Piotr Fryzlewicz et al. Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281, 2014.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, page 107299, 2019.

Petros Drineas and Michael W Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(Dec):2153–2175, 2005.

Mehrdad Mahdavi, Tianbao Yang, and Rong Jin. An improved bound for the nystrom method for large eigengap. *arXiv preprint arXiv:1209.0001*, 2012.

Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling techniques for the nystrom method. In *Artificial Intelligence and Statistics*, pages 304–311, 2009.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3215–3225, 2017.

Dougal J Sutherland and Jeff Schneider. On the error of random fourier features. *arXiv preprint arXiv:1506.02785*, 2015.

Emmanuel J Candes. Modern statistical estimation via oracle inequalities. *Acta numerica*, 15:257–325, 2006.

Damien Garreau, Sylvain Arlot, et al. Consistent change-point detection with kernels. *Electronic Journal of Statistics*, 12(2):4440–4486, 2018.