Universite Paris-Sud
Annee 2016-2017
BRICHLER Nicolas

RAPPORT DE STAGE DE FIN D'ETUDE

Ground Instability risk profiling using satellite image data

.....

Mathematiques de l'aleatoire

Enseignant référent : Sylvain Arlot
Maître de stage : Stephane Chretien
Dates du stage : 01 mai 2017 - 01 septembre 2017
National Physical Laboratory
TW11 0LW Teddington
United Kingdom

Sommaire

Abstract

The aim of this internship is to implement a method for joint modelling of satellite image data and insurance claims processed in a British city using machine learning technique. Thus we want to be able to predict which areas of the city are at risk of suffering material damage due to ground movement (for example due to maintenance works on the underground lines). The satellite dataset can be viewed as a sequence of time series, each coresponding to the evolution of the ground movement in a particular pixel of the images. The insurance claims will firstly be modelled as the result of a Poisson-Cox point process using reproducing kernels [1,2,4]. The satellite image data set will then be studied using efficient time series analysis techniques and plugged into the point process as parameters of its intensity. Using lasso regression[7,8], we will finally be able to detect the most influential time series, thus allowing better assessment of the locations at risk.

Résumé

Le but de ce stage est d'établir un modèle reliant les déclarations de sinistre dans une ville britannique avec les déplacements de terrain mesurés par satellite et de l'implémenter en utilisant des méthodes récentes de machine learning pour obtenir des informations prédictives. Ainsi, nous voulons découvrir si nous sommes capables de prévoir quelles zones de la ville risquent de subir des dégâts matériels suite a des mouvements de terrain (par exemple lors de travaux sur les lignes de métro). Nous disposons d une succession d'images satellites de la ville, prises avec une fréquence régulière, et pour chaque pixel de l'image, de mesures du gradient de sa position. Ainsi nous considérons pour chaque pixel l'évolution de ce gradient comme une série temporelle.
Les plaintes d'assurance sont tout d'abord modelisees comme le resultat d'un processus ponctuel de Poisson Cox[1,2,4] . Puis les series temporelles sont inserees comme parametres de l'intensite de ce processus ponctuel. Enfin, nous utiliserons une regression lasso pour trouver la relation entre les deplacements de terrain et les plaintes d'assurance ainsi que pour detecter les series les series les plus influentes[7,8].

1 Introduction

1.1 National Physical Laboratory

The National Physical Laboratory (NPL) is the national measurement standards laboratory for the United Kingdom, based at Bushy Park in Teddington, London, England. It is the largest applied physics organisation in the UK. Today it provides the scientific resources for the National Measurement System financed by the Department for Business, Innovation and Skills.
NPL also offers a range of commercial services, applying scientific skills to industrial measurement problems and cooperates with professional networks to support scientists and engineers concerned with areas of work in which it has expertise.
NPL is at the forefront of new developments in metrology, such as researching metrology for, and standardising, nanotechnology.

1.2 Context

Big data is a very extensively studied multifaceted field. Big data often comes as multivariate time series obtained through sensor networks (satellite sensors in the case of this internship). Decomposition of high dimensional time series into meaningful components is a new field and only a few methodologies have been devised worldwide. Producing a new method and a fast algorithm for this task will enhance NPL's capability tremendously in the field of time series analytics. The ability to handle extremely large dataset and explore its correlation structure allows to better understand the data and helps the decision making process.

Moreover, I have studied during this internship various machine learning areas of interest such as matrix completion, multiplicative update and Poisson Process estimation and have tried to enhance NPL's analysis tools. For example, the ability to estimate efficiently the intensity of non-homogeneous Poisson Processes would be a great addition to NPL's tools that would find many uses in a lot of different areas.

2 Presentation of the model and statistical tools

2.1 Point Process

A **Point process** [9] on $\mathbb{R}^d$ is a random variable on $(\mathbb{T}, \mathscr{F})$ where $\mathbb{T}$ is the collection of all sequences ϕ of points of $\mathbb{R}^d$ satisfying the following condition :

1. the sequence ϕ is locally finite (i.e. every compact set of $\mathbb{R}^d$ possesses only a finite number of points of ϕ) ;
2. $\forall i \neq j$, $x_i \neq x_j$ (the process is said to be simple,but this condition is not necessary).

Another possible notation is $\Phi = (X_i)_{i \in \mathbb{N}}$ where X_i are random variables on $\mathbb{R}^d$.

A point process Φ thus possesses a probability distribution on $(\mathbb{T}, \mathscr{F})$, the distribution P of Φ.
If B is a Borel set, let us note $\Phi(\omega, B)$ the number of points of the realization of $\Phi(\omega)$ in the Borel set B. The random variable $\Phi(B) : \omega \to \Phi(\omega, B)$ is called a point count.

From **Kolmogorov extension theorem** , the probability distribution of Φ is completely determined by its finite-dimensional distributions $(\Phi(B_1), ..., \Phi(B_n))$ where $(B_1, ..., B_n)$ are bounded Borel sets.

Let Λ be a measure on $\mathbb{R}^d$ so that for every bounded measurable set A $\Lambda(A)$ is finite (i.e. Λ is locally finite). Then Λ is a Radon measure.
A Poisson point process with intensity Λ on $\mathbb{R}^d$ is a point process Φ satisfying the following condition : For every bounded disjoint subsets $(B_1, ..., B_k)$ and non-negative integers $(n_1, ..., n_k) \in \mathbb{N}^k$ we have that

$$\mathbb{P}(\Phi(B_1) = n_1, ..., \Phi(B_k) = n_k) = \prod_{i=1}^{k} e^{-\Lambda(B_i)} \frac{(\Lambda(B_i))^{n_i}}{n_i!}$$

Φ is said to be homogeneous if $\Lambda(B) = \lambda |B|$ with $\lambda \in \mathbb{R}^+$

A point process Φ is said to be **stationary** if for all $y \in \mathbb{R}^d$, $\Phi = (X_i)$ has the same distribution as $\Phi_y = (X_i + y)$.
A point process is said to be **isotropic** if its characteristics are invariant by rotation, that is to say that for every rotation **r** around the origin Φ and $\mathbf{r}\Phi$ have the same distributions.

2.2 Cox Point Process

A **Cox point process** is a generalisation of the Poisson point process where the intensity λ of the process itself is a random measure. For a Cox point process, we have the follwing properties :

1. Given Λ, $\Phi(B)$ is Poisson distributed with parameter $\Lambda(B)$ for any bounded Borel subset B ;
2. For any finite collection of disjoint subsets B_1,.., B_n and conditioned on $\Lambda(B_1)$,. . . , $\Lambda(B_n)$, we have that $\Phi(B_1), ..., \Phi(B_n)$ are independent.

When we can write $\Lambda(B) = \int_B \lambda(x)dx$, λ is called the intensity field or intensity function.

This will be very useful to model the insurance claims in this study. Indeed, these claims can be seen as the realizations of a Cox point process on $\mathbb{R}^3$ (a time dimension and two spatial ones) whose intensity λ depends on multiple parameters, including the one we are focused on : ground movement.

In the case of our study, it seems logical to consider that buildings and human constructions are at a greater to risk to suffer damages when there is a lot of ground movement. Thus we consider that the intensity of the Point process generating the insurance claims is random variable dependant on ground movement.

2.3 Reproducing Kernel Hilbert Space

Let $\mathbb{H}$ be a Hilbert space of functions mapping from a non-empty set $\mathbb{X}$ to $\mathbb{R}$ with scalar product $(f|g)$ and corresponding norm $\|f\|^2 = (f|f)$.

For $x \in \mathbb{H}$ the map $\delta_x : \mathbb{H} \to \mathbb{R}, \delta_x : f \to f(x)$ is called the Dirac evaluation functional at x.

A Hilbert space $\mathbb{H}$ of functions $f : \mathbb{X} \to \mathbb{R}$ is said to be a Reproducing Kernel Hilbert Space (RKHS) if δ_x is continuous $\forall x \in X$. Thus norm convergence implies pointwise convergence.

A function k : $\mathbb{X} \times \mathbb{X} \to \mathbb{R}$ is called a reproducing kernel of $\mathbb{H}$ if the following conditions are met :

1. $\forall x \in \mathbb{X}, y :\to k(y, x) \in \mathbb{H}$
2. $\forall x \in \mathbb{X}, \forall f \in \mathbb{H}, (f|k(\cdot, x)) = f(x)$. This is the so-called reproducing property.

In a RKHS, thanks to the Riesz representation theorem, this kernel exists and is unique [4,5].
Remark : the reproducing property gives $(k(x_j, \cdot)|k(x_i, \cdot)) = k(x_j, x_i)$.

What is very interesting with a RKHS is that, even if its dimension is infinite, optimization problems of the type :

$$\min_{f \in \mathbb{H}} J(f(x_1), ..., f(x_n)) + \Omega(\|f\|^2) \tag{2.1}$$

are actually finite-dimensional problems :

If $\mathbb{X}$ is compact and k is continuous, let us define the integral kernel operator T_k : $L_2(\mathbb{X}) :\to L_2(\mathbb{X})$ given by $T_k g = \int_{\mathbb{X}} k(x, \cdot)g(x)dx$. T_k is positive, self-adjoint and compact, thus its eigenfunctions $\{e_j\}_{j \in \mathbb{N}^*}$ are orthornormal and its eigenvalues $\{\eta_j\}_{j \in \mathbb{N}^*}$ positive. **Mercer's theorem** holds that :

$$\forall x, y, \in \mathbb{X}, k(x, y) = \sum_{j \in \mathbb{N}^*} \eta_j e_j(x) e_j(y) \tag{2.2}$$

Any function $f \in \mathbb{H}$ can be written as $f = \sum \beta_j e_j$ with $\|f\|^2 = \sum \frac{\beta_j^2}{\eta_j} < \infty$ [6]. If we write f as the sum $f(\cdot) = \sum_{i=1}^n \alpha_i k(x_i, \cdot) + v$ where v is orthogonal to the span of $\{k(x_i, \cdot)\}_{1 \leq i \leq n}$, using the reproducing property we have that

$$f(x_i) = (f|k(\cdot, x_i)) = \sum_{j=1}^{n} (\alpha_j k(x_j, \cdot)|k(x_i, \cdot)) + (v|k(\cdot, x_i)) = \sum_{j=1} \alpha_j k(x_j, x_i) \tag{2.3}$$

Furthermore, we see that :

$$\|\sum_{i=1}^{n} \alpha_i k(x_i, \cdot) + v\|^2 = \|\sum_{i=1}^{n} \alpha_i k(x_i, \cdot)\|^2 + \|v\|^2 \geq \|\sum_{i=1}^{n} \alpha_i k(x_i, \cdot)\|^2 \tag{2.4}$$

Thus choosing $v \neq 0$ can only increase the objective function in (3.1) and the minimizer can be written simply as $f^*(\cdot) = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot)$, which reduces the problem to finding $\alpha \in \mathbb{R}^n$.

2.4 Lasso regularization

Many statistical problems are modelled using the classical regression equation

$$Y = X\beta + \epsilon \tag{2.5}$$

where X is a $n \times p$ matrix of explanatory variables, $\beta \in \mathbb{R}^p$ is the unkmown parameter we want to estimate, $Y \in \mathbb{R}^n$ is our response vector and $\epsilon \in \mathbb{R}^n$ is the error vector often assumed to be Gaussian-centered.
The LASSO estimator of β is the solution of the following minimization problem [7,8] :

$$\hat{\beta}_\lambda = \underset{\beta \in \mathbb{R}^p}{\arg\min} \|Y - X\beta\|_2^2 + \lambda \|\beta\|_1 \tag{2.6}$$

The LASSO estimator shrinks the components of the regular least square estimator and even sets some of them to zeros, thus automatically doing model selection based on the value of parameter λ [10]. So it is a very helpful tool for statisticians working in a high dimensional environment as it can significantly reduce the number of relevant variables in a model.

2.5 Model

Our satellite data set is made of a number T of satellite images, each with N pixels. At each pixel, the ground movement can be measured accurately by the satellite, thus giving us knowledge of a N-dimensional time series matrix $X \in \mathbb{R}^{T \times N}$.

From our insurance claims data, we want to be able to estimate the intensity function λ of the Cox point process. We will write $\lambda_i(t), 1 \leq i \leq N, 1 \leq t \leq T$ the value of the intensity function λ at the location corresponding to pixel i and image t. We thus propose to model the dependency between the intensity function and the time series values :

$$\lambda_i(t) = \lambda_{0i} exp(\sum_{k=0}^{K} \sum_{j=1}^{N} \beta_{j,k}^i X_{t-k,j}) + \epsilon_t^i \tag{2.7}$$

$\beta^i \in \mathbb{R}^{N\times T}$ and $\lambda_{0i} \in \mathbb{R}, 1 \leq i \leq N$ are unknown parameters. $(\epsilon_t^i)_{1\leq i\leq N, 1\leq t\leq T}$ is the remaining error supposed to follow a Gaussian $N(0,\sigma^2)$ distribution.
So each pixel has its owns regression parameters on all the values of the time series on the last K+1 images. We can expect β to be sparse since it is very likely that the ground movement in one pixel is only correlated to a few other (nearby pixels).

To try and retrieve this sparsity, we will use LASSO regression on the slightly modified vectors and matrices :

$$\beta'^i = \begin{bmatrix} \beta^i_{.,1} \\ \beta^i_{.,2} \\ \vdots \\ \beta^i_{.,K} \end{bmatrix} \tag{2.8}$$

$$X' = \begin{bmatrix} X_{1,.}X_{0,.}\cdots X_{1-K,.} \\ X_{2,.}X_{1,.}\cdots X_{2-K,.} \\ \vdots \\ X_{T,.}X_{T-1,.}\cdots X_{T-K,.} \end{bmatrix} \tag{2.9}$$

Where $\beta^i_{.,1}$ is the first column of β^i and $X_{1,.}$ the first line of X. If $i \leq 0, X_{i,.}$ is equal to a null vector of corresponding size.
These are basically matrices β^i and X conveniently re-scaled so that we can write

$$\Lambda_i = \lambda_{0i} exp(X'\beta'^i) + \epsilon^i \tag{2.10}$$

We can then obtain a classical LASSO minimization problem :

$$\hat{\beta}'_c = \underset{\beta\in\mathbb{R}^p}{\arg\min} \|log(\Lambda_i) - log(\lambda_{0i}) - X'\beta\|_2^2 + c|\|\beta\|_1 \tag{2.11}$$

3 Cox point process intensity estimation with reproducing kernels

3.1 General idea

As previously stated, we assume that the insurance claims in our data set are the result of a Cox point process on a compact domain $S = [0, a1] \times [0, b1] \times [0, t1]$ representing the city of London during a certain time frame. We use the nonparametric method developed by Seth Flaxman, Yee Whye Teh and Dino Sejdinovic in [1] to estimate the intensity function of the process using reproducing kernels. The intensity function λ of the point process is modelled by :

$$\forall x \in S, \lambda(x) = af^2(x) \tag{3.1}$$

where function f belongs to an RKHS $\mathbb{H}$ with kernel $k : S \times S \to \mathbb{R}$ and $a > 0$ is a scale parameter.

We then write the log-likelihood corresponding to some observations $\{x_i, 1 \leq i \leq n\}$:

$$l(x_1, ..., x_n|\lambda) = \sum_{i=1}^{n} log(\lambda(x_i)) - \int_S \lambda(x)dx \tag{3.2}$$

And we add a term corresponding to the square Hilbert space norm $\|\|^2_{\mathbb{H}}$ of f to obtain the following penalized minimization problem :

$$\min_{f \in \mathbb{H}} - \sum_{i=1}^{n} log(af^2(x_i)) + a \int_S f^2(x)dx + \gamma \|f\|^2_{\mathbb{H}} \tag{3.3}$$

Flaxman et al. showed that a solution of (4.3) can be found in another particular Kernel space $\tilde{\mathbb{H}}$ with kernel function $\tilde{k} = \sum_j \frac{\eta_j}{a\eta_j + \gamma} e_j(x)e_j(x')$ by minimizing the following objective function :

$$J(f) = - \sum_{i=1}^{n} log(af^2(x_i)) + \|f\|^2_{\tilde{\mathbb{H}}} \tag{3.4}$$

Moreover, the solution takes the form $f(\cdot) = \sum_{i=1}^{n} \alpha_i \tilde{k}(xi, \cdot)$

3.2 Computation

We will try to approximate Mercer decomposition of k with eigenvalues $\{\eta_j\}_{j\in\mathbb{N}^*}$ and eigenvectors $\{e_j\}_{j\in\mathbb{N}^*}$ from 2.2 :

$$\forall x, y, \in \mathbb{X}, k(x,y) = \sum_{j\in\mathbb{N}^*} \eta_j e_j(x) e_j(y)$$

We will sample m points $u_1, ..., u_m$ uniformly from the domain S and we create the Gram matrix K_{uu} defined by $[K_{uu}]_{i,j} = k(u_i, u_j)$. Let R and Q be respectively the matrices of eigenvalues and corresponding eigenvectors so that

$$K_{uu} = QRQ^t \tag{3.5}$$

Further works from [1,11, 12] shows that the Gram matrix $[\tilde{K}]_{i,j} = \tilde{k}(x_i, x_j)$ can be estimated by matrix $\hat{\tilde{K}}_{xx}$:

$$\hat{\tilde{K}}_{xx} = K_{xu} Q (\frac{a}{m} R^2 + \gamma R)^{-1} Q^t K_{ux} \tag{3.6}$$

where $[K_{xu}]_{i,j} = k(x_i, u_j)$. We can then write, with $\alpha = (\alpha_1, ..., \alpha_n)^t$

$$\begin{aligned} J(f) &= -\sum_{i=1}^{n} log(af^2(x_i)) + \|f\|^2_{\mathbb{H}} \\ J(\alpha) &= -\sum_{i=1}^{n} log(a(\sum_{j=1}^{n} \alpha_j \tilde{k}(x_i, x_j))^2) + \alpha^t \tilde{K} \alpha \end{aligned} \tag{3.7}$$

Hence

$$\nabla J(\alpha) = -2 \sum_{i=1}^{n} \frac{2\tilde{K}_{.i}}{\sum_{j=1}^{n} \alpha_j \tilde{K}_{ij}} + 2\tilde{K}\alpha \tag{3.8}$$

Thus we can minimize the penalized likelihood with gradient descent.

3.3 Simulation

We simulated non homogeneous Poisson point processes on $[0,1]\times[0,1]$ to test and calibrate the method. The intensity of the process is a randomized sum of Gaussian functions.

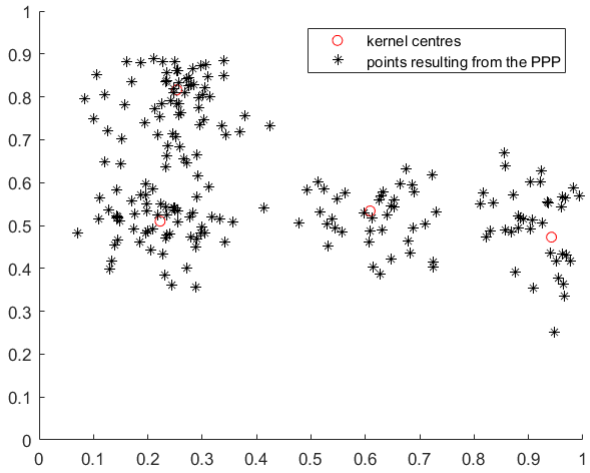

FIGURE 3.1 – Example of a Cox point process

We simulate the point process by thinning[2], i.e. we firstly simulate a homogeneous Poisson point process whose intensity is an upper bound M of the desired non constant intensity $\lambda(\cdot)$. Then, for each point i of the point process with coordinates (x_i, y_i), we draw a uniform variable u_i on [0,1]. If $u_i \leq \lambda(x_i, y_i)/M$, we keep the point in our desired process, else we delete it.

The hyperparameters γ and the lengthscale of the Gaussian kernel are chosen through cross-validation. To estimate parameter a, we proceed differently from [1] and compute it during the gradient descent so that :

$$a = \frac{N}{\int_S f^2(x)dx} = \frac{N}{\int_S (\sum_j \alpha_j k(xj, \cdot))^2(x)dx} \tag{3.9}$$

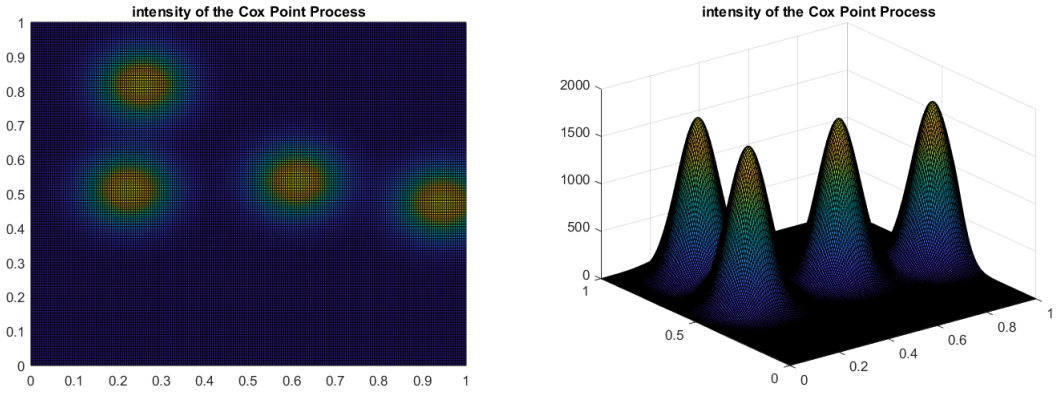

FIGURE 3.2 – Intensity of the Cox process

3.4 Results

Our real insurance claims dataset consisted of 817 claims made in an English city from April 2016 to May 2017(fig 3.3). The coordinates of the claimants have been modified for anonymity's sake.

3.5 Clustering approach

Another idea we had to find the intensity function was to cluster our insurance claims data set ans then set a Gaussian function at the centre of each cluster with a scaling parameter corresponding to the number of points in the cluster. The benefit of this method was that it was really fast, however, every error in the clustering would lead to a very biased intensity estimation, which is the reason we didn't chose this method in the end. One other challenge of this approach is to automatically find an adequate number of cluster for a given data set. So here is a description of the clustering methods considered for the projects.

3.5.1 K-means

Firstly, we tried the well known K-means method to cluster our data set. To find a good parameter k for the data set, we used Bayesian information criterion (BIC)[14] and the silhouette method[13], decribed below :

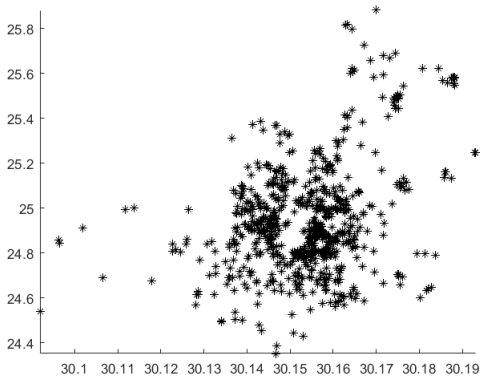

FIGURE 3.3 – Spatial representation of our insurance claims dataset

We cluster our data set with K-means, with k varying from 1 to 15. For every clustering done and for every data point i in a given clustering, two values are computed :

1. $a(i)$ measuring the average dissimilarity with the other data points within the cluster.
2. $b(i)$ measuring the lowest dissimilarity of i to another cluster.

The silhouette is defined by

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{3.10}$$

Functions a and b are usually based on the distance used to do the K-means clustering. When $s(i)$ is close to 1 it means that $a(i) << b(i)$ so the data point is well clustered. To the contrary, if $s(i)$ is to close to -1, then it means that the cluster of i is badly chosen. Thus, using the average silhouette value for a given k gives a method for selecting an appropriate k.

Despite its efficiency, K-means clustering has drawbacks. It can fail to distinguish clusters that are close to each other and sometimes it creates new clusters just for a few outliers, thus distorting the overall intensity.

3.5.2 Minimum Spanning Tree based clustering

The minimum spanning tree of a data set is the subset of edges that connects all the data points together with the minimum total edge length or edge weight (figure 3.4). There

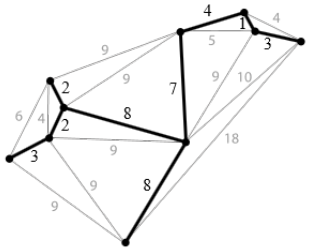

FIGURE 3.4 – Minimum spanning tree of a weighted graph

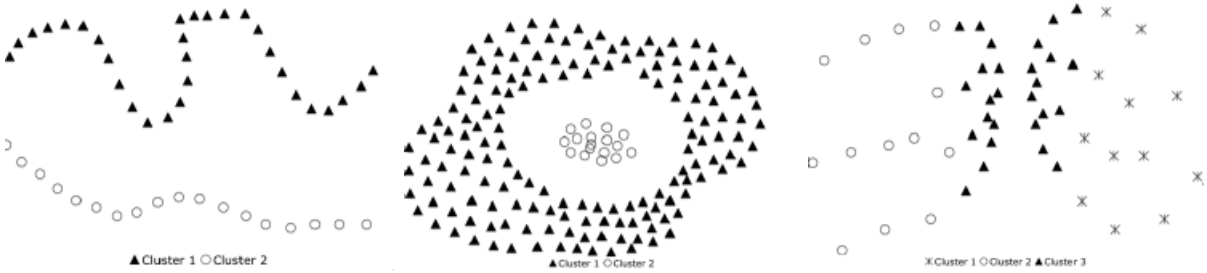

FIGURE 3.5 – Examples of clustering obtained from the MSDR algorithm [3]. We can see that if the density is non homogeneous, the algorithm may fail to detect the right clusters

are multiple clustering algorithms based on the minimum spanning tree. The interesting property of a minimum spanning tree is that it is without cycles. Thus, every edge that is removed from the tree will create two separate trees. Thus removing edges is a natural way of clustering our data. The algorithms from Grygorash, Oleksandr, Yan Zhou, and Zach Jorgensen[3] aim to minimize the within cluster edge length standard deviation. This way of clustering has the added benefit of detecting clusters of any shape (figure 3.5), however it requires the clusters to be well separated and once again, the presence of possible outliers may greatly bias the results.

4 Time Series Analysis

Our ground movement dataset consists of millions of ground movement measurements spread across Great Britain in space and several years in time. However, if we note Nts the number of time series and T the number of time steps, we have that $Nts >> T$, that is to say we have measurements for hundreds of thousands of positions in the country but only for hundreds of dates. Thus, we will use sampling to reduce the number of time series used for our analysis.

In addition to being useful for evaluating and predicting insurance claims, this dataset also contains interesting information in itself. We thus wanted to better understand our dataset, to see for example the potential correlations between the various time series, if there were any clusters, or whether neighbouring time series behaved similarly or not. To this end, used Prony's method to separate the trend and seasonality from the noise of the time series[15,16].

4.1 Prony's method

Prony's method is a useful method for modelling signals using a finite sum of exponential terms. Let us note $(Y_t)_{t=1,...,T}$ one of the time series that we observe. It can be split into different signals : the trend, the seasonality and the noise. Our goal is to separate the noise ϵ_t from the rest of the signal Y_t. To this end, we will approximate the original signal with a sum of exponential of the type :

$$s_t = \sum_{k=-K}^{K} c_k z_k^t \tag{4.1}$$

where the $z_k, k = -K, ..., K$ are complex numbers. If they have modulus 1, this gives sinusoidal functions. Otherwise it corresponds to a damping factor. So this is a more flexible model than the well-known Fourier Transform. The frequencies obtained need not be multiples of the same base frequency. We will then assume that $Y_t = s_t + \epsilon_t$.

The Prony method is a way of computing the complex numbers z_k. First, we consider the singular matrix

$$S_r = \begin{pmatrix} s_1 & s_2 & ... & s_r \\ s_2 & s_3 & ... & s_{r+1} \\ \vdots & \vdots & ... & \vdots \\ s_{T-r+1} & s_{T-r+2} & ... & s_T \end{pmatrix} \tag{4.2}$$

Let $v \in Ker(S_r)$, then the z_k belong to the set of roots of the polynomial

$$p(z) = \sum_{i=1}^{r} v_i z^{r-i} \tag{4.3}$$

So the elegance of Prony's method is that we recover the sum of exponentials that reconstruct a given signal simply by solving a linear algebra problem. Now that we have the numbers z_k, we want to recover the values c_k. In a setting without noise, ie $s_t = Y_t$, we would only need to solve the following Vandermonde linear system :

$$\begin{pmatrix} z_{-K}^{-K} & z_{-K+1}^{-K} & ... & z_K^{-K} \\ \vdots & \vdots & ... & \vdots \\ z_{-K}^{K} & z_{-K+1}^{K} & ... & z_K^{K} \end{pmatrix} \begin{pmatrix} c_{-K} \\ \vdots \\ c_K \end{pmatrix} = \begin{pmatrix} Y_0 \\ \vdots \\ Y_{2K+1} \end{pmatrix} \tag{4.4}$$

It has been shown in [15] that using a nuclear norm penalization improves the efficiency of Prony's method, particularly with respect to noise and missing data. So this is the method that we used in practice. The first thing that we do is approximating the original matrix Hank(y) by a matrix $\hat{Y}$ of lower rank.

$$Hank(y) = \begin{pmatrix} y_1 & y_2 & ... & y_r \\ y_2 & y_3 & ... & y_{r+1} \\ \vdots & \vdots & ... & \vdots \\ y_{T-r+1} & y_{T-r+2} & ... & y_T \end{pmatrix} \tag{4.5}$$

The idea is that the noise will add new frequencies in the signal and that by reducing the rank, we reduce its effect. We can do that by using a least-squares minimization with the nuclear norm penalization :

$$\min_{Y \in \mathbb{H}_{T,K}} \frac{1}{2} \|Hank(y) - Y\| + \lambda \|Y\|_* \tag{4.6}$$

where $\mathbb{H}_{T,K}$ is the set of matrices with the same size as $Hank(y)$. The coefficients z_k can then be retrieved in the same way as in the original method. As for the coefficients c_k, we run a simple least-squares minimization procedure.

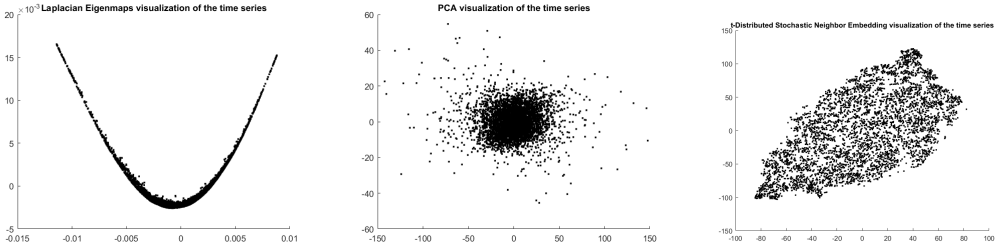

FIGURE 4.1 – 2D Visualization of 5000 time series uniformly sampled in London with 54 time steps over 2 years.

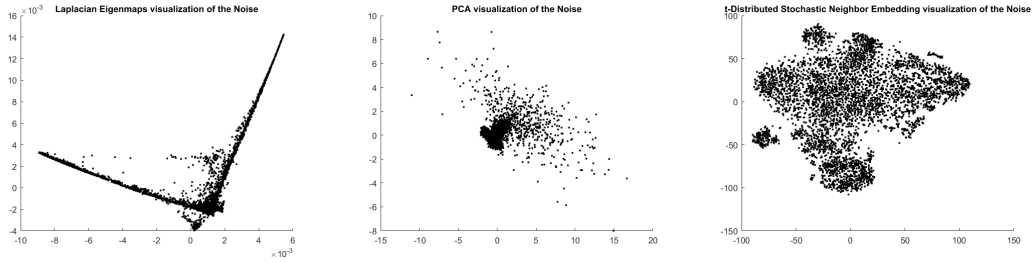

FIGURE 4.2 – 2D Visualization of the noise resulting from the same 5000 time series.

4.2 Dimensionality Reduction

In this section, we present representations in two dimensions of some time series sampled from our dataset using classical dimensionality reduction techniques like Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE)[17], Laplacian Eigenmaps or Maximum Variance Unfolding (MVU) from the very useful toolbox of Laurens van der Maaten (https ://lvdmaaten.github.io/drtoolbox/).
It appears difficult to notice clusters, especially from the PCA representation of the data, which resembles a Gaussian distribution. Nevertheless, the t-SNE representation of the noise shows some little clusters around the main one, so we tried anyway to cluster the time series by fitting a Gaussian mixture distribution [18].

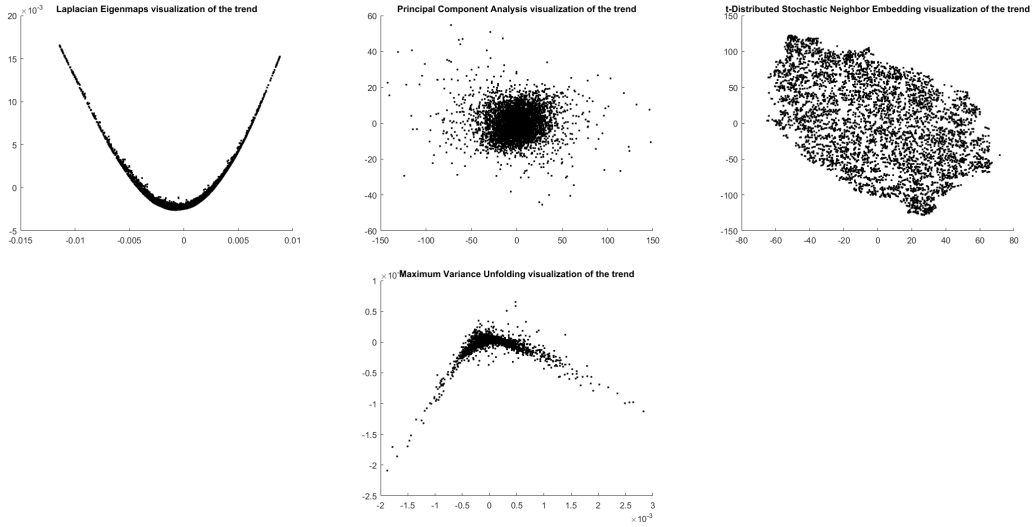

FIGURE 4.3 – 2D Visualization of the trend and seasonality resulting from the same 5000 time series.

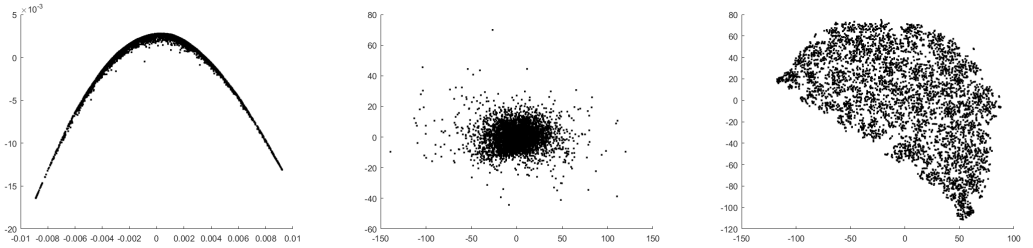

FIGURE 4.4 – 2D Visualization of 5000 time series sampled within 50meters of insurance claims locations

4.3 Visualization of the time series

Using Akaike information criterion and Bayes Information Criterion, we tried to cluster the "smoothed" signal obtained from Prony's method. This revealed some very interesting clusters (shown in fig 4.5). The fact that some of these clusters correspond to geographic areas seems to show that this clustering is meaningful. Furthermore, when we compare the clustering with the geology map of the city, it looks like the separation between the two ground types (shown in grey and green) roughly corresponds to the separation between the blue and cyan clusters. This is encouraging and does seem to prove there is indeed a link between the values of the time series and the type of ground they are located in. As can be seen in figure 4.6, using the 95th quantile of the Gaussian distribution seem to capture most of the time series within a cluster, so modelling our data with Gaussian distributions appears to be a valid assumption.

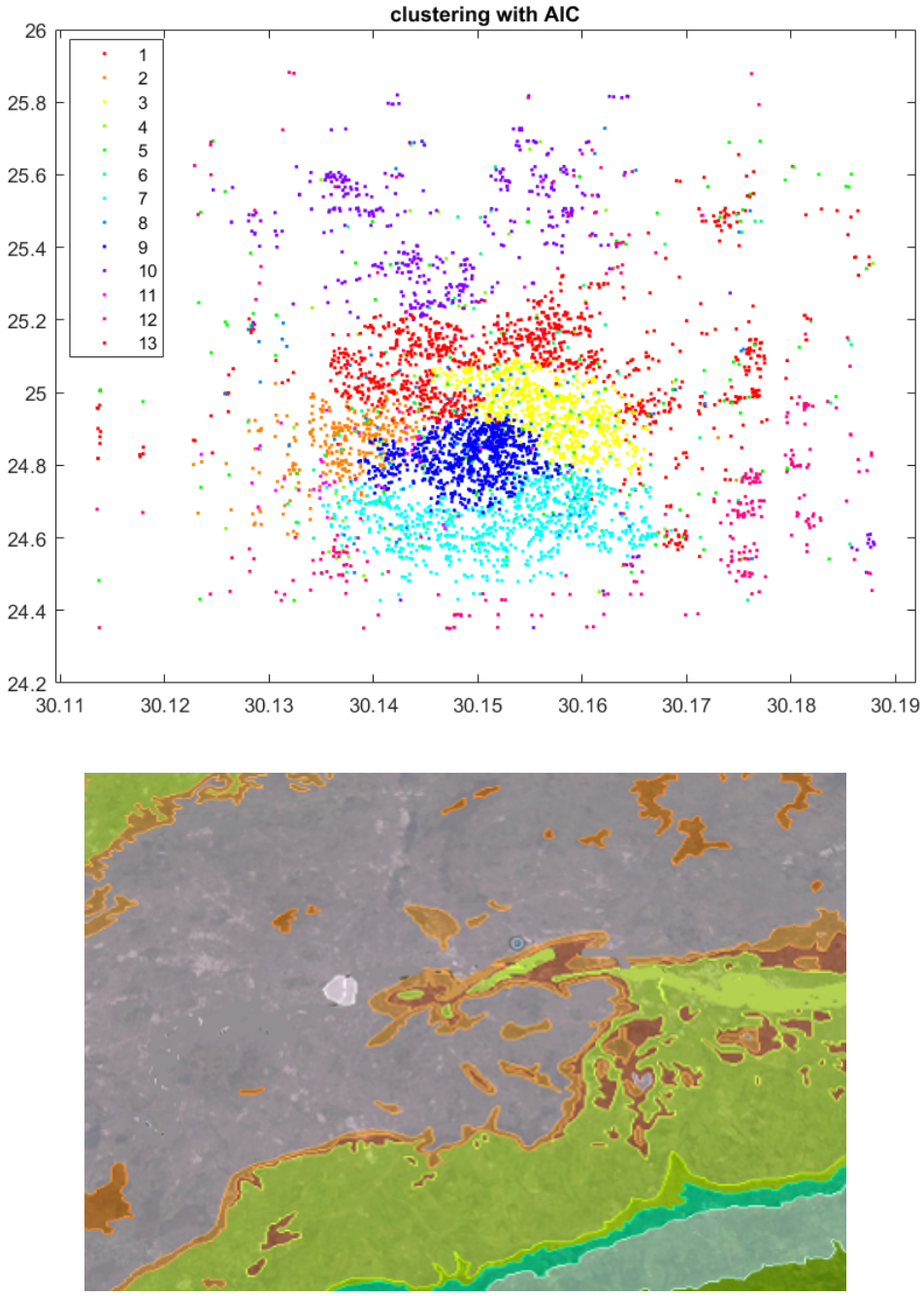

FIGURE 4.5 – Comparison of the clustering of the TS dataset with a Gaussian mixture model using AIC with a geology map of the same city

FIGURE 4.6 – Representation of the time series within one cluster

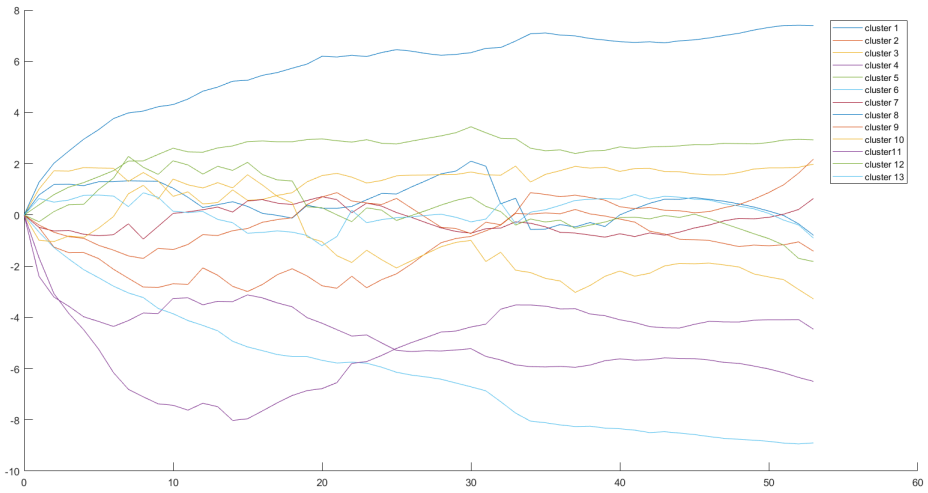

FIGURE 4.7 – Representation of the time series average per cluster

5 Matrix Completion

The problem of missing values in our dataset led us to study the question of matrix completion [19,21]. It consists of recovering a complete matrix using only some of its observed entries. This is a recurring problem in collaborative filtering and recommendation system. For example it was the subject of the famous "Netflix Prize" that challenged hundreds of teams across the world to predict user ratings for films. In our case, our goal would be to recover the missing values of our time series.

In this type of problems, the matrix is assumed to be a low-rank one. In the case of the netflix prize, one could see it as the equivalent of saying that there are trends among users and thus the ratings of may users will be similar because of their common preferences. In our case, the clusters that we could identify lead us to believe that the same hypothesis should be true, i.e. that many time series behave very similarly and that we can assume that the $Nts \times T$ matrix containing all the values of the time series is of low rank. In this chapter, we will present two online matrix completion methods.

5.1 Nuclear norm minimization

This first method uses a basic observation from the Singular Value Decomposition of a matrix. Any $n \times p$ matrix A of rank r can be decomposed as :

$$A = \sum_{j=1}^{r} \sigma_j u_j v_j^\intercal \tag{5.1}$$

where r is the rank of A, $\sigma_1^2, ..., \sigma_r^2$ are the nonzero eigenvalues of $A^\intercal A$ and $u_1, ..., u_r$ and $v_1, ..., v_r$ are two orthornormal families of $\mathbb{R}^n$ and $\mathbb{R}^p$ such that $AA^\intercal u_j = \sigma_j^2 u_j$ and $A^\intercal A v_j = \sigma_j^2 v_j$. As the rank of matrix A is equal to its amount of nonzero singular values, one may be tempt to solve the matrix completion problem by penalizing the nuclear norm :$\|A\|_* = trace\left(\sqrt{A^\intercal A}\right) = \sum_{i=1}^{r} \sigma_i(A)$. As in the case of the LASSO penalization, using the L1 norm results in a sparse vector, penalizing with the nuclear norm will give us some null singular values and thus a low rank solution.

The idea presented by Lafond, Wei and Moulines in [22] is to use a stochastic version of the Frank-Wolfe algorithm to solve the following problem (which is equivalent by duality) :

$$\min_{\theta \in C_R} f(\theta) = \|M - \theta\|_F^2, C_r = \{\theta \| \theta \|_* < R\} \tag{5.2}$$

The benefit of the Frank Wolfe method is that there is no need to project on the set of constraints. We simply need to solve a linear problem at each step t :

$$\alpha_t = \arg\min_{\alpha \in C}(\alpha, \nabla f(\theta_t)) \tag{5.3}$$

Now, $\nabla f(\theta) = 2(U - M)$, so minimizing the scalar product of α with $2(U - M)$ with respect to $\|\theta\|_* < R$ comes down to finding the top singular vectors of $U - M$, u_1 and v_1, and then setting

$$\alpha_t = -Ru_1 v_1^\intercal$$

where $\gamma_t = \frac{K}{t+K-1}$ and $K \in \mathbb{N}^*$ is a step size parameter. After each iteration, we update

$$\theta_{t+1} = (1 - \gamma_t)\theta_t + \gamma_t \alpha_t$$

With this method, the error decreases as $O\left(\sqrt{\frac{log(t)}{t}}\right)$.

5.2 Alternating Minimization

Alternating minimization represents a widely applicable and empirically successful approach for finding low-rank matrices that best fit the given data. For example, for the problem of low-rank matrix completion, this method is believed to be one of the most accurate and efficient, and formed a major component of the winning entry in the Netflix problem. In the alternating minimization approach, the low-rank target matrix M of size $d1 \times d2$ is written in a bilinear form :

$$M = UV^\intercal \tag{5.4}$$

where U is of size $d1 \times k$, V of size $d2 \times k$ and k is the rank of M. The algorithm then alternates between optimizing U and V. While the overall problem is generally non-convex, each sub-problem where one of the two matrices U or V is supposed to be constant is convex.

For example, the algorithm devised by Chi, Kakade and Netrapalli [19] starts by taking the top k SVD of $\frac{d1d2}{|\Omega|}M_{init}$ to initialize U_0 and V_0. We thus make sure that we are working with a k-rank matrix.
Then at each step $t = 1, .., T$, we recover $W_U D W_V^\intercal$ the SVD of $U_{t-1}V_{t-1}^t$ and renormalize

$$\tilde{U}_{t-1} = W_U D^{\frac{1}{2}}$$

$$\tilde{V}_{t-1} = W_V D^{\frac{1}{2}}$$

After we get a new observation $M_{i,j}$, we update :

$$U_t = \tilde{U}_{t-1} - 2\eta d1d2(\tilde{U}_{t-1}\tilde{V}_{t-1}^\intercal - M)_{i,j}e_i e_j^\intercal \tilde{V}_{t-1}$$

$$V_t = \tilde{V}_{t-1} - 2\eta d1d2(\tilde{U}_{t-1}\tilde{V}_{t-1}^\intercal - M)_{i,j}e_j e_i^\intercal \tilde{U}_{t-1}$$

which can be seen as a gradient descent with step parameter η. To obtain Frobenius norm error ϵ, this method requires $O\left(log(1/\epsilon)\right)$ time steps. While, alternating minimization methods are not as well understood as the other optimization methods for matrix completion, they show very good empirical results and are very successful in practice.

Moreover, in the case where M is symmetric and can thus be written as $M = UU^t$, it has been shown[21] that the objective function

$$f(U) = \sum_{i,j\in\Omega} [M_{i,j} - (UU^\intercal)_{i,j}]^2 \tag{5.5}$$

where Ω is the set of observed entries of M, has no spurious local minima. So any local minimum U of f is actually a global minimum with $f(U) = 0$ and thus recovers a correct low-rank factorization of M. However, given that for any orthonormal matrix R, we have $UU^\intercal = URR^\intercal U^\intercal = UR(UR)^\intercal$, this factorization is not unique. But all these solutions are equivalent.

6 Conclusion

Analyzing ground movement time series has been a very enlightening experience. We have seen thus far that there is indeed a relation between the values of a time series and its geographical position, as shown by the existence of separated clusters. However, it is hard at this point to know if our LASSO regression has good predictability value since we only have two years of observation data and thus it is hard to correctly take into account the yearly seasonality. A greater time frame would allow us to build a better test to gauge the validity of this regression.

Moreover, we have created several useful tools which will enhance NPL's ability to handle and analyze time series and point processes. These tools can be readily applied without much tinkering, which should help NPL's future projects. Finally, this internship has been a great learning experience for me. I have become familiar with many statistical and mathematical tools such as LASSO regression, Prony's method, various clustering techniques and Poisson point processes.

7 References

1. Flaxman, Seth, Yee Whye Teh, and Dino Sejdinovic. "Poisson intensity estimation with reproducing kernels." arXiv preprint arXiv :1610.08623 (2016).
2. Lewis, Peter A., and Gerald S. Shedler. "Simulation of nonhomogeneous Poisson processes by thinning." Naval research logistics quarterly 26.3 (1979) : 403-413.
3. Grygorash, Oleksandr, Yan Zhou, and Zach Jorgensen. "Minimum spanning tree based clustering algorithms." Tools with Artificial Intelligence, 2006. ICTAI'06. 18th IEEE International Conference on. IEEE, 2006.
4. Berlinet, A. and Thomas-Agnan, C. Reproducing Kernel Hilbert Spaces in Probability and Statistics. Kluwer, 2004.
5. Kreyszig, E. Introductory Functional Analysis with Applications. Wiley, 1989.
6. Schölkopf, B. and Smola, A. J. Learning with kernels : support vector machines, regularization, optimization and beyond. MIT Press, 2002.
7. Tibshirani, Robert. "The lasso method for variable selection in the Cox model." Statistics in medicine 16.4 (1997) : 385-395.
8. Tibshirani, Robert. "Regression shrinkage and selection via the lasso." Journal of the Royal Statistical Society. Series B (Methodological) (1996) : 267-288.
9. Chiu, Sung Nok, et al. Stochastic geometry and its applications. John Wiley and Sons, 2013.
10. Giraud, Christophe. Introduction to high-dimensional statistics. Vol. 138. CRC Press, 2014.
11. Williams, C. K., and C. E. Rasmussen. "Gaussian processes for machine learning, vol. 2." (2006) : 4.
12. Baker, Christopher TH. "The numerical treatment of integral equations." (1977).
13. Rousseeuw, Peter J. "Silhouettes : a graphical aid to the interpretation and validation of cluster analysis." Journal of computational and applied mathematics 20 (1987) : 53-65.
14. Wit, Ernst, Edwin van den Heuvel, and Jan-Willem Romeijn. "'All models are wrong...' : an introduction to model uncertainty." Statistica Neerlandica 66.3 (2012) : 217-236.

15. Al Sarray, Basad, et al. "Enhancing Prony's method by nuclear norm penalization and extension to missing data." Signal, Image and Video Processing (2017) : 1-8.
16. Peter, Thomas. "generalized Prony method".Georg-August-Universitat Gottingen. Dissertation zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades.
17. L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik. Dimensionality Reduction : A Comparative Review. Tilburg University Technical Report, TiCC-TR 2009-005, 2009.
18. McLachlan, Geoffrey, and David Peel. Finite mixture models. John Wiley Sons, 2004.
19. Jin, Chi, Sham M. Kakade, and Praneeth Netrapalli. "Provable efficient online matrix completion via non-convex stochastic gradient descent." Advances in Neural Information Processing Systems. 2016.
20. Li, Tianyang, et al. "Statistical inference using SGD." arXiv preprint arXiv :1705.07477 (2017).
21. Ge, Rong, Jason D. Lee, and Tengyu Ma. "Matrix completion has no spurious local minimum." Advances in Neural Information Processing Systems. 2016.
22. Lafond, Jean, Hoi-To Wai, and Eric Moulines. "Convergence analysis of a stochastic projection-free algorithm." stat 1050 (2015) : 5.