Unsupervised Machine Learning for Pathological Radar Clutter Clustering: the P-Mean-Shift Algorithm

Yann Cabanes, Frédéric Barbaresco, Marc Arnaudon, Jérémie Bigot

Unsupervised Machine Learning for Pathological Radar Clutter Clustering: the P-Mean-Shift Algorithm

Yann Cabanes[1,2], Frédéric Barbaresco[1], Marc Arnaudon[2], and Jérémie Bigot[2]

[1] Thales LAS, Advanced Radar Concepts, Limours, FRANCE
yann.cabanes@gmail.com; frederic.barbaresco@thalesgroup.com
[2] Institut de Mathématiques de Bordeaux, Bordeaux, FRANCE
marc.arnaudon@math.u-bordeaux.fr;
jeremie.bigot@math.u-bordeaux.fr

Abstract. This paper deals with unsupervised radar clutter clustering to characterize pathological clutter based on their Doppler fluctuations. Operationally, being able to recognize pathological clutter environments may help to tune radar parameters to regulate the false alarm rate. This request will be more important for new generation radars that will be more mobile and should process data on the move. We first introduce the radar data structure and explain how it can be coded by Toeplitz covariance matrices. We then introduce the manifold of Toeplitz covariance matrices and the associated metric coming from information geometry. We have adapted the classical k-means algorithm to the Riemaniann manifold of Toeplitz covariance matrices in [1], [2]; the mean-shift algorithm is presented in [3], [4]. We present here a new clustering algorithm based on the p-mean definition in a Riemannian manifold and the mean-shift algorithm.

Keywords: radar clutter · machine learning · unsupervised classification · p-mean-shift · autocorrelation matrix · Burg algorithm · reflection coefficients · Kähler metric · Tangent Principal Components Analysis · Capon spectra.

1 Introduction

Radar installation on a new geographical site is long and costly. We would like to shorten the time of deployment by recognizing automatically pathological clutters with past known diagnosed cases. This requirement will become more important because new generation radar sensors will be mobile and should work on the move and self-adapt to the environment. The objective is therefore to develop machine learning algorithms to recognize specific clutter characteristics from their Doppler Spectrum statistical fluctuations. Future work will study the spatial variations of the Doppler information to characterize the clutter.

To recognize pathological radar environments using a pulse-Doppler radar, we need to extract very precise Doppler information from a very small series of pulses (around 10). In this context, the classical FFT (Fast Fourier Transform) is not satisfactory due to its low resolution output for such small time series. To overcome this difficulty, we propose a clustering method based on the signals Toeplitz covariance matrices.

To begin with, we briefly introduce the radar data which we intend to analyze. For simplicity, we first consider one fixed direction in which a radar sends radio waves and

we subdivide this direction into cells. The radar sends a burst of radio waves in a direction and then receives the echoes. For each echo we measure its amplitude r and phase ϕ, so that it can be represented by a complex number $u = re^{i\phi}$. As a result, the original radar observation value of each cell is a complex vector $\mathbf{u} = [u(0), u(1), ..., u(n-1)]^T$, where n is the number of radio waves emitted in each burst and $\cdot^T$ denotes the matrix transposition. We now try to extract sharp Doppler information from this short time series. Instead of using directly the original observation vector $\mathbf{u}$ of each cell, we assume it to be a realization of a centered stationary complex Gaussian process and identify it with its covariance matrix $\mathbf{R} = \mathbb{E}[\mathbf{u}\,\mathbf{u}^H]$, where $\cdot^H$ denotes the complex matrix conjugate transpose. In other words, the new observation value for each cell is a covariance matrix estimation, which is Toeplitz due to the assumption of stationarity of the process. Then our clustering problem can be summarized as follows: regroup in a same cluster the cells having close Toeplitz covariances matrices.

In order to achieve this clustering problem, three fundamental issues should be addressed. The first one is how to estimate a Toeplitz covariance matrix from each original radar observation vector $\mathbf{u}_i$. The second one is to define a metric on the set of Toeplitz covariance matrices. The third one is to adapt our clustering algorithms to the newly defined metric space of Toeplitz covariance matrices. The first question will be answered in section 4.1 through autoregressive models, the second one will be answered in section 4.2 by giving to the set of Toeplitz covariance matrices a Riemannian structure and the last issue will be answered in section 5 by studying a new clustering algorithm called the p-mean-shift. Finally, clustering experiments on simulated data are presented in section 6.1 and clustering results on real data are discussed in section 6.2 in which we tackle the issue of clustering signals of different lengths.

2 Introduction to signal processing theory

2.1 From radar data to complex matrices

In this study, the input data will be taken on a single burst for a single elevation corresponding to the horizontal beam.

Therefore, the radar provides us a 2D complex matrix of size $(\#impulses) \times (\#cells)$:

$$U = \begin{bmatrix} \boxed{\begin{matrix} u_{0,0} \\ u_{1,0} \\ \vdots \\ u_{n-1,0} \end{matrix}} & \boxed{\begin{matrix} u_{0,1} \\ u_{1,1} \\ \vdots \\ u_{n-1,1} \end{matrix}} & \begin{matrix} u_{0,2} & \cdots & u_{0,p-1} \\ u_{1,2} & \cdots & u_{1,p-1} \\ \vdots & \ddots & \vdots \\ u_{n-1,2} & \cdots & u_{n-1,p-1} \end{matrix} \end{bmatrix} \tag{1}$$

where n denotes the number of pulses of the burst, p the number of cells.

The complex coefficient u_{ij} represents the amplitude and phase after pulse compression of the echo beam at time index i (ith impulse) at distance index j from the radar.

The data to classify are the cells, each cell being represented by a column of the matrix U.

2.2 Model and hypotheses

We now focus on a single column of the matrix U defined in equation (1) and define its autocorrelation matrix.

We denote by $\cdot^T$ the matrix transposition, $\cdot^H$ the complex matrix conjugate transpose and $\cdot^*$ the complex scalar conjugate.

We denote:

$$\mathbf{u} = [u(0), u(1), ..., u(n-1)]^T \tag{2}$$

the one dimensional complex signal registered in a cell.

We assume this signal to be stationary with zero mean:

$$\mathbb{E}[u(k)] = 0 \quad \forall k \tag{3}$$

We also assume that this signal can be modeled as an autoregressive Gaussian process of order $n-1$:

$$u(k) + \sum_{i=1}^{n-1} a_i^{n-1}\, u(k-i) = w(k) \tag{4}$$

where a_i^{n-1} are the prediction coefficients and w is the prediction error.

Interested readers may refer to [6] for a comprehensive course on complex signal processing theory.

2.3 From the input vector to the autocorrelation matrix

We define the autocorrelation matrix:

$$\mathbf{R} = \mathbb{E}[\mathbf{u}\,\mathbf{u}^H] \tag{5}$$

$$r_{i,j} = \mathbb{E}[u(k+i)u(k+j)^*] \tag{6}$$

We define the time lag: $t = i - j$.

Proposition 1 (autocorrelation and stationarity). *The signal is supposed to be stationary, so $r_{i,j}$ depends only of the lag t.*

$$\begin{aligned} r_{i,j} &= \mathbb{E}[u(k+i)u(k+j)^*] \\ &= \mathbb{E}[u(k+i-j)u(k)^*] \\ &= \mathbb{E}[u(k+t)u(k)^*] \\ &= r_t \end{aligned} \tag{7}$$

Proposition 2 (autocorrelation and conjugation).

$$\begin{aligned} r_{-t} &= \mathbb{E}[u(k-t)u(k)^*] \\ &= \mathbb{E}[u(k)u(k+t)^*] \\ &= \mathbb{E}[u(k+t)u(k)^*]^* \\ &= r_t^* \end{aligned} \tag{8}$$

Consequence **R** is a Toeplitz Hermitian Positive Definite matrix.

$$\mathbf{R} = \begin{bmatrix} r_0 & r_1^* & r_2^* & \dots & r_{n-1}^* \\ r_1 & r_0 & r_1^* & \dots & r_{n-2}^* \\ r_2 & r_1 & r_0 & \dots & r_{n-3}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{n-1} & r_{n-2} & r_{n-3} & \dots & r_0 \end{bmatrix} \tag{9}$$

Note that the assumptions made in section 2.2 that the signal can be modeled as a complex stationary autoregessive Gaussian process with zero mean has the following equivalent vectorial formulation: $\mathbf{u} = \mathbf{R}^{1/2}x$ with **R** a Toeplitz Hermitian Positive Definite matrix and x a standard complex Gaussian random vector which dimension is equal to the number of pulses. In the section 3, we present a simulation model strongly related to this vectorial formulation of our hypotheses.

3 The simulation model

Each cell is simulated independently. For each cell, we simulate a complex vector using a SIRV (Spherically Invariant Random Vectors) model [12], [13]:

$$Z = \underbrace{\sqrt{\tau}R_s^{1/2}x}_{\text{information coming from the environment}} + \underbrace{b_{radar}}_{\text{thermal noise coming from the radar itself}} \tag{10}$$

with:

τR_s**:** the environment autocorrelation matrix.

τ**:** clutter texture; it is a positive real random variable independent from x and b_{radar}.

R_s**:** scaled environment autocorrelation matrix (Toeplitz Hermitian Positive Definite).

x**,** b_{radar}**:** independent standard complex Gaussian random vectors whose dimension is equal to the number of pulses.

To construct the scaled autocorrelation matrix R_s, we learn experimentally from a great number of radar measures the power spectral density S_f of the clutter we want to simulate. The scaled autocorrelation coefficients $r_s(k)$ of the Toeplitz matrix R_s can then be computed from the power spectral density using the inverse Fourier transform.

For example, if we want to simulate a clutter which spectrum has the shape of a Gaussian distribution of mean m and variance σ^2 with a power coefficient P, *i.e.*:

$$S_f(\xi) = P\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(\xi-m)^2}{2\sigma^2}} \tag{11}$$

We compute the corresponding autocorrelation function R_f:

$$\begin{aligned} R_f(\tau) &= \widehat{S_f}(-\tau) \quad \Big(\text{since } S_f(\tau) = \widehat{R_f}(\tau)\Big) \\ &= P\frac{1}{\sqrt{2\pi\sigma^2}}\left(x \mapsto \widehat{e^{-\frac{(x-m)^2}{2\sigma^2}}}\right)(-\tau) \\ &= P\frac{1}{\sqrt{2\pi\sigma^2}}e^{-i2\pi m(-\tau)}\left(x \mapsto \widehat{e^{-\frac{x^2}{2\sigma^2}}}\right)(-\tau) \\ &= P\frac{1}{\sqrt{2\pi\sigma^2}}e^{i2\pi m\tau}\sqrt{\frac{\pi}{\frac{1}{2\sigma^2}}}e^{-2\sigma^2\pi^2(-\tau)^2} \\ &= Pe^{i2\pi m\tau}e^{-2\pi^2\sigma^2\tau^2} \end{aligned} \tag{12}$$

Finally, if we want to simulate a signal f which sampling period is T, the scaled autocorrelation matrix R_s (which is Toeplitz) will be defined by its coefficients:

$$r_s(k) = R_f(kT) \tag{13}$$

In the rest of the study, we will assume that the environment autocorrelation matrix τR_s has diagonal coefficients sufficiently larger than one to consider that the whole signal autocorrelation matrix $\mathbf{R}$ will be close enough to the environment autocorrelation matrix τR_s to characterize the clutter. For the simulated datasets used in section 6.1, the clutter texture τ is a Weibull distribution of scale parameter $\lambda = 0.7418$ and shape parameter $k = 0.658$; the mean of this probability law is $\mu = 1$. The scaled environment autocorrelation matrix R_s is computed using Gaussian distributions of power coefficient $P = 10^4$.

We now explain how to estimate the autocorrelation matrix $\mathbf{R}$ from the data vector $\mathbf{u}$.

4 The data representation space

4.1 Autocorrelation matrix estimation

In our clustering problem, the autocorrelation matrix $\mathbf{R}_i$ of the whole signal (environment and radar noise) will be estimated independently for each cell $\mathbf{u}_i$:

$$U = \begin{bmatrix} \boxed{\begin{matrix} u_{0,0} \\ u_{1,0} \\ \vdots \\ u_{n-1,0} \end{matrix}} & \boxed{\begin{matrix} u_{0,1} \\ u_{1,1} \\ \vdots \\ u_{n-1,1} \end{matrix}} & \boxed{\begin{matrix} u_{0,2} \\ u_{1,2} \\ \vdots \\ u_{n-1,2} \end{matrix}} & \cdots & \boxed{\begin{matrix} u_{0,p-1} \\ u_{1,p-1} \\ \vdots \\ u_{n-1,p-1} \end{matrix}} \\ \downarrow & \downarrow & \downarrow & & \downarrow \\ \widehat{\mathbf{R}_0} & \widehat{\mathbf{R}_1} & \widehat{\mathbf{R}_2} & & \widehat{\mathbf{R}_{p-1}} \end{bmatrix} \tag{14}$$

Empirical covariance matrix To estimate the Toeplitz autocorrelation matrix $\mathbf{R}$ from the data vector $\mathbf{u}$, we can estimate each coefficient r_t by the following empirical mean:

$$\widehat{r_t} = \frac{1}{n-t} \sum_{k=0}^{n-1-t} u(k+t)u(k)^* \quad t = 0, ..., n-1 \tag{15}$$

Note that this method is unprecise when the vector length n is small, especially when the lag *t* is close to $n-1$. We now propose a more robust method to estimate the autocorrelation matrix with few data based on the estimation of autoregessive coefficients and the relation between the autoregressive model and the autocorrelation matrix.

According to [5] the Levinson algorithm gives us the following bijection:

$$\begin{aligned} &\mathcal{T}_n^+ \rightarrow \mathbb{R}_+^* \times \mathbb{D}^{n-1} \\ &R_n \mapsto (p_0, \mu_1, ..., \mu_{n-1}) \end{aligned} \tag{16}$$

where $\mathcal{T}_n^+$ denotes the set of Positive Definite Hermitian Toeplitz matrices of size n; $p_0 = r_0$ is the averaged quadratic power and $\mu_j = a_j^j$ $(1 \leq j \leq n-1)$ are the reflection coefficients, where a_i^j denotes the i^{th} coefficient of the Gaussian autoregressive model of order j.

It is therefore equivalent to estimate the coefficients $(p_0, \mu_1, ..., \mu_{n-1})$ and R_n.

Burg algorithm The regularized Burg algorithm is based on the optimization of the prediction coefficients a_i^j of the autoregressive model for $1 \leq i \leq j \leq n-1$ to lower the predictions errors. It allows us to transform the original complex vector $\mathbf{u}$ into a power factor p_0 in $\mathbb{R}_+^*$ and reflection coefficients $\mu_j = a_j^j$ $(1 \leq j \leq n-1)$ lying in $\mathbb{D}^{n-1}$, where $\mathbb{D}$ represents the complex unit disk. The regularized Burg algorithm of order M and regularization coefficient γ is detailed in [1], [2], [7], [8], [14], [17]. One of the interests of representing the data in $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ rather than $\mathcal{T}_n^+$ is the nice expression of the metric in $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ presented in section 4.2; it also offers the possibility to compare signals of different lengths (see section 6.2).

4.2 The Kähler metric

Each data vector $\mathbf{u}_i$ is now represented by an estimation of its autocorrelation matrix $\widehat{\mathbf{R}_i}$ which is a Toeplitz Hermitian Positive Definite matrix. We define the metric on the set $\mathcal{T}_n^+$ of Toeplitz Hermitian Positive Definite matrices as coming from the Fisher metric on the manifold of complex Gaussian distributions with zero means, Toeplitz Hermitian Positive Definite covariance matrices and null relation matrices.

According to the previous bijection, we can represent a Toeplitz Hermitian Positive Definite matrix T_i by the corresponding coefficients $(p_{0,i}, \mu_{1,i}, ..., \mu_{n-1,i})$. The following distance has been introduced by F. Barbaresco in [9] on the set $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ to make this bijection an isometry. In the Encyclopedia of Distance by Deza [10], this distance is called Barbaresco distance:

$$\begin{aligned} d^2_{\mathcal{T}_n^+}(T_1,T_2) = & \quad d^2_{\mathcal{T}_n^+}\left((p_{0,1},\mu_{1,1},...,\mu_{n-1,1}),(p_{0,2},\mu_{1,2},...,\mu_{n-1,2})\right) \\ = & \quad n\log^2\left(\frac{p_{0,2}}{p_{0,1}}\right)+\sum_{l=1}^{n-1}\frac{n-l}{4}\log^2\left(\frac{1+\left|\frac{\mu_{l,1}-\mu_{l,2}}{1-\mu_{l,1}\mu^*_{l,2}}\right|}{1-\left|\frac{\mu_{l,1}-\mu_{l,2}}{1-\mu_{l,1}\mu^*_{l,2}}\right|}\right) \end{aligned} \tag{17}$$

Note that the metric on the product space $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ is a product metric, which greatly simplifies the computations. The equations of the geodesics of the set $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ endowed with the Kähler metric are described in [7].

To work on the metric space $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ endowed with the Kähler metric described previously, we now introduce the notion of mean of a set of points lying in a Riemannian manifold.

4.3 The Riemannian p-mean

The p-mean of a dataset $(x_i)_{i=1,...,N}$ is defined in [7] as the set of minimizers of the function:

$$g : x \mapsto \frac{1}{N}\sum_{i=1}^{N} d(x,x_i)^p \tag{18}$$

We can see in equation (17) that the squared distance between two matrices T_1 and T_2 is a linear combination of squared distances between the coordinates $(p_{0,1},\mu_{1,1},...,\mu_{n-1,1})$ and $(p_{0,2},\mu_{1,2},...,\mu_{n-1,2})$. Hence the coordinates can be averaged independently:

$$\begin{array}{rcl} T_0 & \mapsto & (\boxed{p_{0,0},} \boxed{\mu_{1,0},} \cdots, \boxed{\mu_{n-1,0}}) \\ \vdots & & \boxed{\vdots} \boxed{\vdots} \quad \boxed{\vdots} \\ T_{m-1} & \mapsto & (\boxed{p_{0,m-1},} \boxed{\mu_{1,m-1},} \cdots, \boxed{\mu_{n-1,m-1}}) \\ & & \boxed{\downarrow} \quad \boxed{\downarrow} \quad \boxed{\downarrow} \\ T & \leftarrow & (\quad p_0, \qquad \mu_1, \quad \cdots, \quad \mu_{n-1} \quad) \end{array} \tag{19}$$

The p-mean can be approximated performing a gradient descent on the function g, which is equivalent to a gradient descent on each coordinate. This decomposition is one of the reason we choose $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ rather than $\mathcal{T}_n^+$ for the computations.

4.4 The p-mean for $p \in [1, \infty[$

For $p \in [1,+\infty[$, the function g has a unique global minimizer e_p in $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ endowed with the Kähler metric (see [7] for a proof). e_1 is called the median, e_2 is called the mean.

4.5 The p-mean for $p \in]0, 1[$

For $p \in]0, 1[$, there are several local minimizers of the function g. The derivative of the function $x \mapsto x^p$ is the function $x \mapsto px^{p-1}$ whose value tends towards infinity when x tends towards zero, x being positive. Therefore there is a local minimizer located at each point of the dataset. These minima are rather sharp and thin, like the function $x \mapsto x^p$ for x close to zero. There might be additional smooth local minimizers in areas gathering many points. The large scale shape of the function g defined in equation (18) is a smooth function depending on the shape of the dataset with in addition sharp local minimizers related to each point of the dataset. If each point can step out of its local minimum, we can think of a clustering algorithm performing a gradient descent on the function g, and then clustering the points according to which local minimizer they arrived at. We now present this algorithm that we called p-mean-shift.

5 The p-mean-shift clustering algorithm

5.1 The generalized mean-shift algorithm

In the usual mean-shift algorithm, we use a kernel function k (of size s) to define a function f estimating the density of the dataset $(x_i)_{i=,\dots,N}$:

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} k_s(d(x, x_i)) \tag{20}$$

where $d(x, x_i)$ represents the distance between x and x_i. A kernel function taking into account the curvature of the manifold is described in [15] and adapted to the Kähler metric in [3]. The function f is now generalized in the sense that we do not impose k_s to be positive and the integral of f to be equal to one over the Riemaniann manifold; $k_s : \mathbb{R}_+ \mapsto \mathbb{R}$ is only assumed to be decreasing and smooth almost everywhere.

To reach a local maximum of the estimated density function, we shift each point according to a gradient ascent on the function f:

$$\begin{aligned} \nabla f(x) &= \frac{1}{N} \sum_{i=1}^{N} \nabla \left(k_s(d(x, x_i)) \right) \\ &= \frac{1}{N} \sum_{i=1}^{N} \nabla d(x, x_i)\, k_s'(d(x, x_i)) \\ &= \frac{1}{N} \sum_{i=1}^{N} \frac{-\overrightarrow{x\, x_i}}{d(x, x_i)}\, k_s^{'}(d(x, x_i)) \end{aligned} \tag{21}$$

where $\overrightarrow{x\, x_i}$, also written $exp_x^{-1}(x_i)$, denotes the element of the tangent space of the manifold $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ at x such that the geodesic starting at x at time 0 with inital tangent vector $\overrightarrow{x\, x_i}$ arrives at x_i at time 1.

5.2 Comparison of the generalized mean-shift algorithm and the p-mean-shift algorithm

Comparison of the generalized mean-shift algorithm and the p-mean-shift algorithm for $p \in]0,1[$. Note that the following problems are equivalents:

$$\begin{aligned} & Minimize \quad x \mapsto \frac{1}{N}\sum_{i=1}^{N} d(x,x_i)^p \\ \iff & Maximize \quad x \mapsto \frac{1}{N}\sum_{i=1}^{N} -d(x,x_i)^p \\ \iff & Maximize \quad x \mapsto \frac{1}{N}\sum_{i=1}^{N} k_p(d(x,x_i)) \quad with\ k_p : x \mapsto -|x|^p \end{aligned} \tag{22}$$

From this point of view, the p-mean-shift algorithm is a particular case of the genralized mean-shift algorithm in which we allow the function k to be negative and f to be non-integrable over the Riemaniann manifold. Note that the extrema of the functions defined in equation (22) are scale independent: if all the distances are multiplied by the same positive constant c, the whole functions are multiplied by c^p, hence the extrema does not change.

Comparison of the generalized mean-shift algorithm and the p-mean-shift algorithm for $p \xrightarrow[p>0]{} 0$. For $p = 0$, the functions of equation (22) are constant. To extend the p-mean-shift algorithm for $p = 0$, we notice that the minima of the function $x \mapsto \frac{1}{N}\sum_{i=1}^{N} d(x,x_i)^p$ are the same than those of the function $x \mapsto \left(\frac{1}{N}\sum_{i=1}^{N} d(x,x_i)^p\right)^{\frac{1}{p}}$. We then study the behaviour of this last function when $p \xrightarrow[p>0]{} 0$.

Proposition 3. *Let μ be a probability measure on a measurable space M; and g be a function defined on M such that there exists two positive constants a and b for which $0 < a < g < b$ and such that the function $x \mapsto \ln(g(x))$ is integrable on M for the measure μ. Then we have the following result:*

$$\|g\|_{L^p,\mu} \xrightarrow[\substack{p>0}]{\substack{p\to 0}} exp\left(\int_M ln(g(x))d\mu(x)\right) \tag{23}$$

Demonstration 1

$$\begin{aligned}
\left(\int_M g(x)^p d\mu(x)\right)^{\frac{1}{p}} &= exp\left(\frac{1}{p}\ ln\left(\int_M g(x)^p d\mu(x)\right)\right) \\
&= exp\left(\frac{1}{p}\ ln\left(\int_M exp(p\ ln(g(x)))d\mu(x)\right)\right) \\
&= exp\left(\frac{1}{p}\ ln\left(\int_M (1+p\ ln(g(x)) + o(p\ ln(g(x))))d\mu(x)\right)\right) \\
&= exp\left(\frac{1}{p}\ ln\left(\int_M (1+p\ ln(g(x)) + o(p)\ ln(g(x)))d\mu(x)\right)\right) \\
&= exp\left(\frac{1}{p}\ ln\left(1+p\ \int_M ln(g(x))d\mu(x) + o(p)\int_M ln(g(x))d\mu(x)\right)\right) \\
&= exp\left(\frac{1}{p}\ ln\left(1+p\ \int_M ln(g(x))d\mu(x) + o(p)\right)\right) \\
&= exp\left(\frac{1}{p}\ \left(p\ \int_M ln(g(x))d\mu(x) + o(p)\right)\right) \\
&= exp\left(\int_M ln(g(x))d\mu(x) + o(1)\right)
\end{aligned} \tag{24}$$

Proposition 4. *Using the previous result for the discrete measure $f = \frac{1}{N}\sum_{i=1}^{N}\delta_{x_i}$ on our Riemannian manifold, we obtain for all x different from any point x_i of the dataset:*

$$\left(\frac{1}{N}\sum_{i=1}^{N} d(x,x_i)^p\right)^{\frac{1}{p}} \underset{\substack{p\to 0\\ p>0}}{\longrightarrow} exp\left(\frac{1}{N}\sum_{i=1}^{N} ln(d(x,x_i))\right) \tag{25}$$

The following problems are then equivalents:

$$\begin{aligned}
&Minimize \quad x \mapsto \left(\frac{1}{N}\sum_{i=1}^{N} d(x,x_i)^p\right)^{\frac{1}{p}} \quad for\ p \underset{p>0}{\longrightarrow} 0 \\
\iff &Minimize \quad x \mapsto exp\left(\frac{1}{N}\sum_{i=1}^{N} ln(d(x,x_i))\right) \\
\iff &Minimize \quad x \mapsto \frac{1}{N}\sum_{i=1}^{N} ln(d(x,x_i)) \\
\iff &Maximize \quad x \mapsto \frac{1}{N}\sum_{i=1}^{N} -ln(d(x,x_i)) \\
\iff &Maximize \quad x \mapsto \frac{1}{N}\sum_{i=1}^{N} k(d(x,x_i)) \quad with\ k : x \to -ln(x)
\end{aligned} \tag{26}$$

As for $p \in]0,1[$, the case $p \underset{p>0}{\longrightarrow} 0$ is a particular case of the generalized mean-shift algorithm in which we allow the function k to be negative and f to be non-integrable over the Riemaniann manifold. As for $p \in]0,1[$, the extrema of the functions defined in equation (26) are scale independent.

Conclusion: The p-mean-shift algorithm is a particular case of the generalized mean-shift algorithm for specific functions k. The corresponding function f defined in equation (20) does not approximate the density of the dataset anymore. However, the maxima of f are still located in areas regrouping many points.

One of the advantages of the p-mean-shift algorithm over the mean-shift algorithm is the simplicity of the tuning parameter $p \in [0,1[$ which is bounded. This algorithm is scale independent, which means that the parameter p will be tuned according the shape of the dataset and not its scale. Due to the rather important influence of the function k over distant points compared with the usual mean-shift kernel functions k, there might be less outliers left alone in their own cluster. Its main drawback is the presence of a local minimum (or maximum if you consider the opposite function) at each point of the dataset. To help the point starting at x_i to step out the local minima associated with x_i, we do not take into account the point x_i in the first step of the gradient descent.

6 Results

6.1 Results on simulated data

Using the model described in section 3, we have simulated 30 vectors with the model parameters (τ_1, R_1) and 30 vectors with the model parameters (τ_2, R_2), τ_i being a random process and R_i a constant matrix. To be more precise, the matrices R_1 and R_2 have been computed using the model detailed in section 3 to simulate clutter with a spectrum of Gaussian shape of parameters $m_1 = 0.1$, $\sigma_1^2 = 0.0018$ and $m_2 = -0.2$, $\sigma_2^2 = 0.0018$ respectively. Operationally, we often try to center the average Doppler of each signal to zero to cluster the signals using their width and shape rather than their average Doppler; this scenario would correspond to the simulation parameters $m_1 = m_2 = 0$, and $\sigma_1^2 \neq \sigma_2^2$ which has been dealt with in [1], [2].

Then for each vector $\mathbf{u}_i$ we try to recover the parameters (τ_i, R_i) used to simulate it. In practice we use the Burg algorithm to recover the equivalent parameters $(p_{0,i}, \mu_{1,i}, ..., \mu_{n-1,i})$. In this paper, we classify the data only on the scaled autocorrelation matrix R, represented by the reflection coefficients $(\mu_1, ..., \mu_{n-1})$. Future work might also use the texture parameter τ, influencing the power coefficient p_0, to classify the data.

Each vector $\mathbf{u}_i$ is now represented by its reflection coefficients $(\mu_{i,1}, ..., \mu_{i,n-1})$ in the metric space $\mathbb{D}^{n-1}$ endowed with the Kähler metric.

We show in figure 1 the Tangent Principal Components Analysis (TPCA) of the simulated dataset. The TPCA is an extension of the classical Principal Components Analysis (PCA) to a Riemannian manifold. The first step is to compute the mean x of the dataset and then represent each point x_i of the dataset by the tangent vector $\overrightarrow{x\,x_i}$ which we renormalize according to the scalar product at the mean. Hence, we

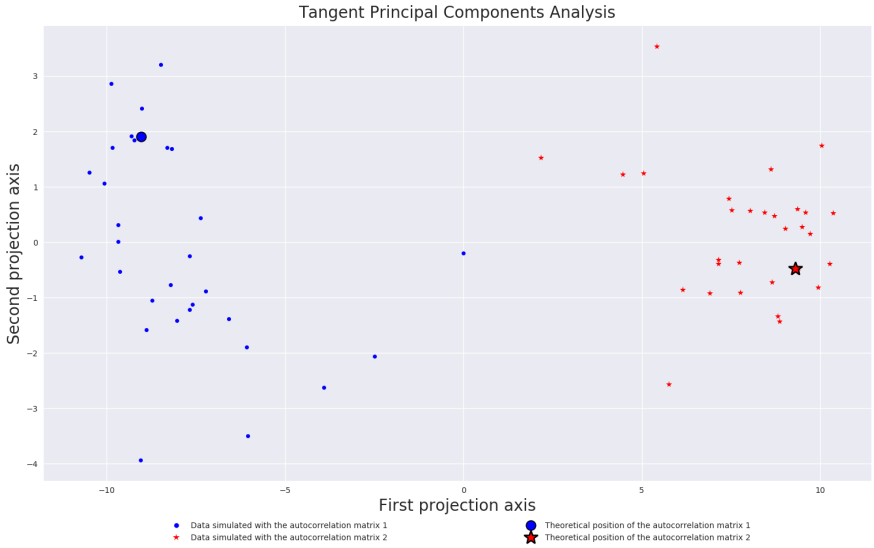

Fig. 1. The Tangent Principal Components Analysis on $\mathbb{D}^{n-1}$ using the Kähler metric

represent each point x_i of the Riemannian manifold by a vector in the tangent space at the mean which is a vector space and therefore suitable for Principal Components Analysis. Interested readers may refer to [16] for more details on the TPCA. In figure 1, the 30 points simulated with the model parameters (τ_1, R_1) are represented by blue disks and the 30 points simulated with the model parameters (τ_2, R_2) are represented by red stars. The large blue disk and the large red star represent the positions computed with the Levinson algorithm of the scaled environment autocorrelation matrices R_1 and R_2. The TPCA can be useful to visualize the shape of the dataset and choose the clustering algorithm accordingly; it can also be used to visualize the clustering result.

We now cluster these points using the p-mean-shift algorithm introduced in section 5. In figure 2, we plot the number of clusters obtained on our dataset for different values of $p \in [0, 1]$. This number should decrease when p increases and will be equal to 1 for $p = 1$ (uniqueness of the median on $\mathbb{D}^{n-1}$ endowed with the Kähler metric [7]). We then choose a value of p in an area in which the number of clusters looks stable: this means that this number of clusters in well adapted to the shape of the dataset. To get a meaningful clustering, do not choose the stable areas for which each point is in a different cluster ($p \leq 0.1$ on figure 2) or for which every points are in the same cluster ($p \geq 0.5$ on figure 2). For the rest of the study, we choose $p = 0.4$ which correspond to 3 clusters on the graphic.

We then compare the p-mean-shift algorithm to the well-known mean-shift [3] and the k-means [1] [2] algorithms, all adapted to the space $\mathbb{D}^{n-1}$ endowed with the Kähler metric. In figure 3 we plot the FFT and the Capon spectra [17] of the simulated data

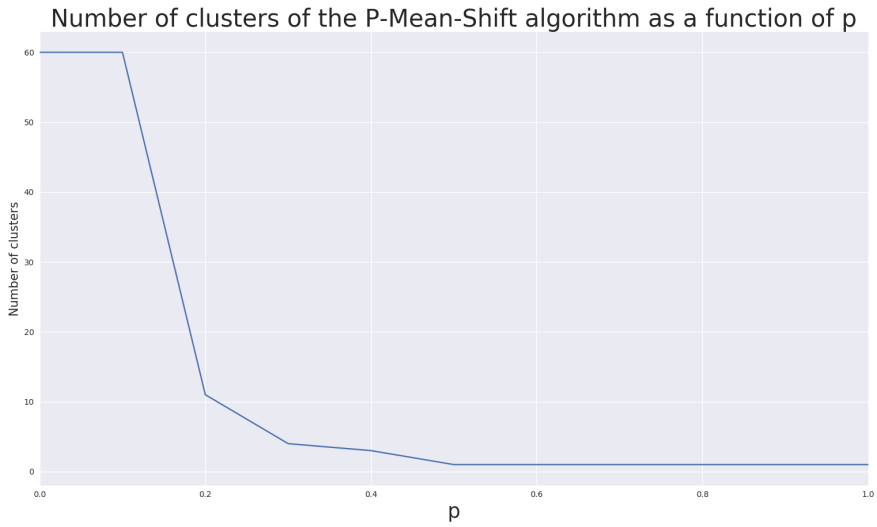

Fig. 2. The number of clusters as a function of p for the p-mean-shift algorithm on $\mathbb{D}^{n-1}$ using the Kähler metric

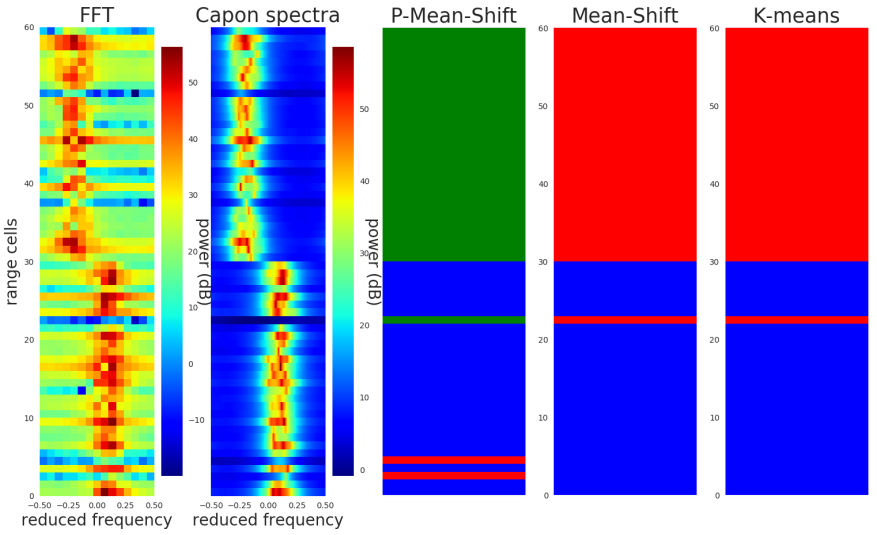

Fig. 3. FFT, Capon spectra and clustering results of the p-mean-shift for $p = 0.4$, the mean-shift and the k-means algorithms on the space $\mathbb{D}^{n-1}$ endowed with the Kähler metric

on the left graphics, and the corresponding clusterings obtained with the p-mean-shift for $p = 0.4$, the mean-shift and the k-means algorithms on the right graphics. Note that each of these clustering algorithms has parameters to tune which might influence greatly the result of the clustering: the parameter p for the p-mean-shift algorithm, the kernel function and the kernel size for the mean-shift algorithm and the number of clusters k for the k-means algorithm.

To visualize the p-mean-shift clustering for $p = 0.4$ on the Capon spectra, we plot in figure 4 the Capon spectrum of each point of the dataset and color it according to the clustering obtained on figure 3.

In figure 5 we color each point of the dataset in $\mathbb{D}^{n-1}$ according the p-mean-shift clustering for $p = 0.4$. For convenience, only the four first coefficients of reflection are represented. Note that the expression of the first coefficient of reflection μ_1 is given by the Levinson algorithm [5] in a simple manner with respect to the autocorrelation coefficients: $\mu_1 = -\frac{r_1}{r_0}$. Using equation 13, we can then express explicitly the theoretical position of the first coefficient of reflection μ_1 in the complex unit disk $\mathbb{D}$ of the spectrum of Gaussian shape of parameters m, σ^2 and sampling period $T = 1$ as a simple function of m and σ^2:

$$\mu_1 = -\frac{r_1}{r_0} = -\frac{Pe^{i2\pi m}e^{-2\sigma^2\pi^2}}{P} = -e^{i2\pi m}e^{-2\sigma^2\pi^2}. \tag{27}$$

Hence the parameter m influences the argument of μ_1 and the parameter σ^2 influences its modulus. The reflection coefficients $\mu_i, i \geq 2$ can also be expressed explicitly as functions of m of σ^2 using the Levinson algorithm. Note that the reflection coefficients estimation is less precise as the order of the coefficient grows; see in figure 5 the position of the estimated reflection coefficients compared with the position of the autocorrelation matrices used to simulate the data. This decrease in precision comes from the difficulty of estimating the coefficients of an autoregressive model of high order with few data. In the Burg algorithm the coefficients are estimated recursively on the order of the model; the approximation error of a coefficient thus propagates to the next coefficient. To avoid very irregular spectra, the Burg algorithm has a regularization parameter that tends to reduce the modulus of the reflection coefficients.

When working on simulated data, once the clustering results are obtained for each clustering algorithm, we compute the corresponding F1 score to evaluate and compare the clustering algorithms performances. The F1 score is a way of measuring the performance of a supervised classification algorithm. We adapted it to our unsupervised classification algorithms by doing all possible permutations in the clustering labels in order to find the best matching with the expected result knowing the simulation parameters of each point of the dataset; the best matching being defined as the best F1 score. Finally we plot in figure 6 the normalized confusion matrix of each clustering algorithm using the labels corresponding to the best F1 score.

6.2 Results on real data

We now present some results on real radar data. We display on figure 7 a ground map of Saint-Mandrier in which the position of the radar is represented by a red dot. The

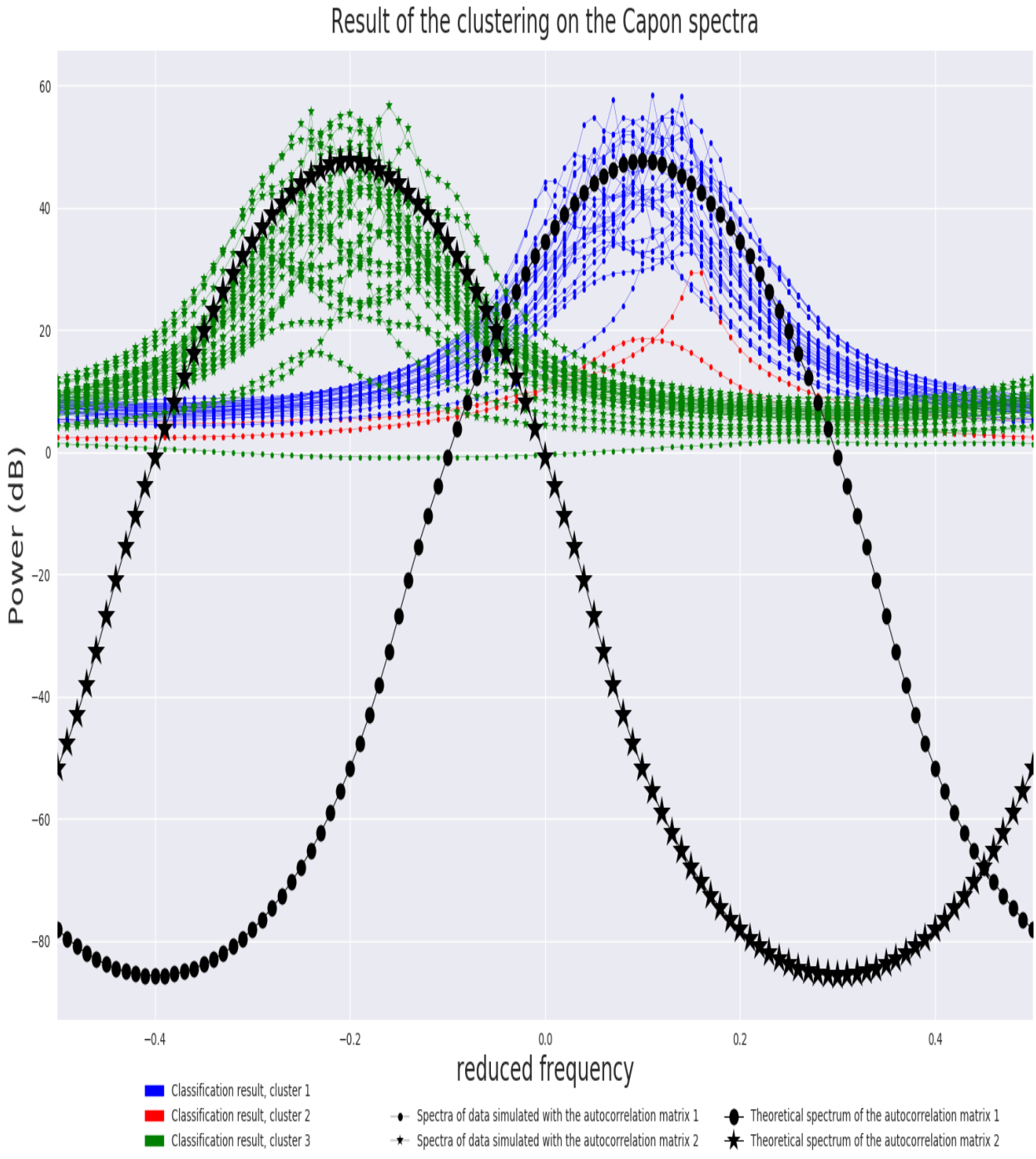

Fig. 4. Results of the p-mean-shift clustering for $p = 0.4$ on the Capon spectra

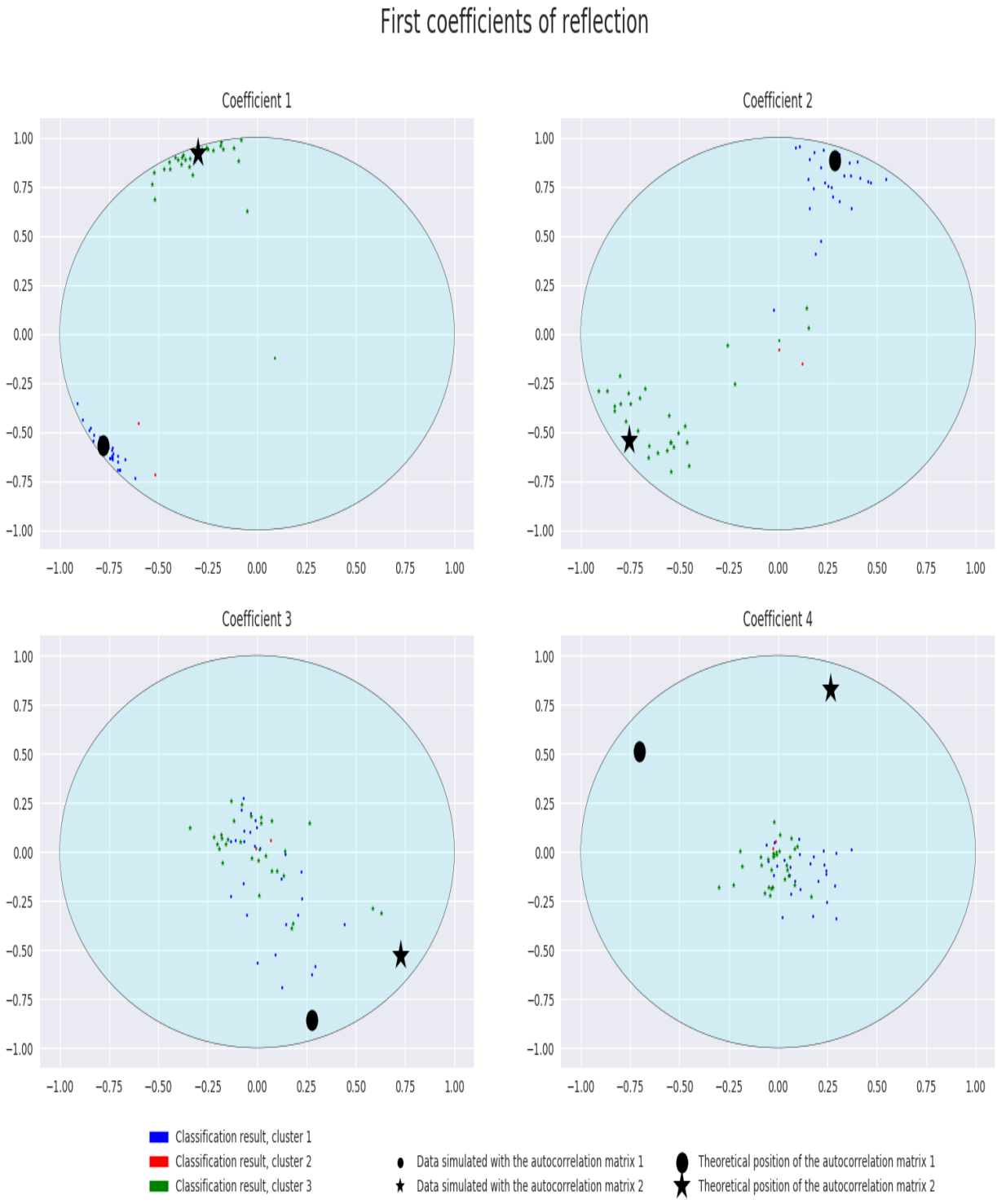

Fig. 5. Results of the p-mean-shift clustering for $p = 0.4$ on the four first coefficients of reflection

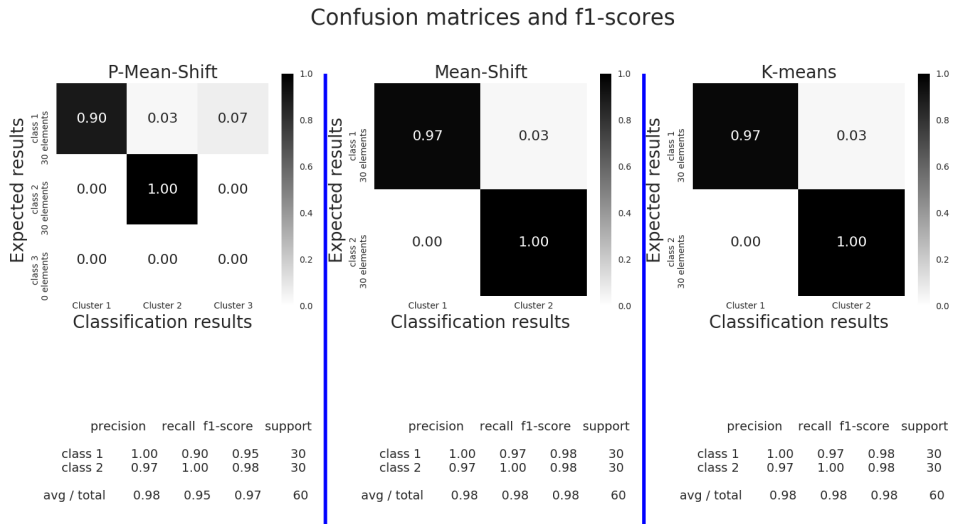

Fig. 6. Confusion matrices and F1 scores of the p-mean-shift for $p = 0.4$, the mean-shift and the k-means algorithms on the space $\mathbb{D}^{n-1}$ endowed with the Kähler metric

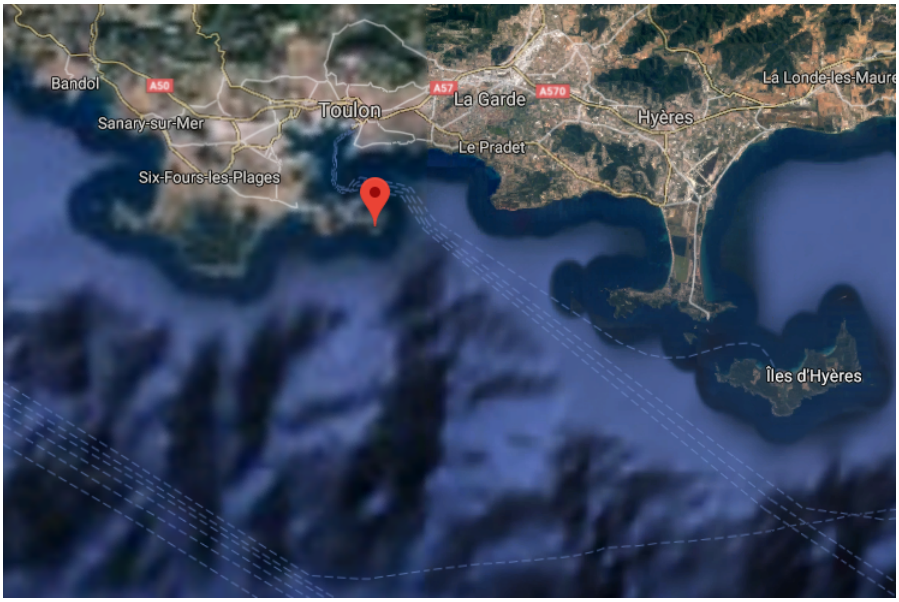

Fig. 7. Ground map Saint-Mandrier

dataset contains 150 bursts of different shapes: both the number of pulses n and the number of cells p (see the matrix U defined in (1)), vary from one burst to another.

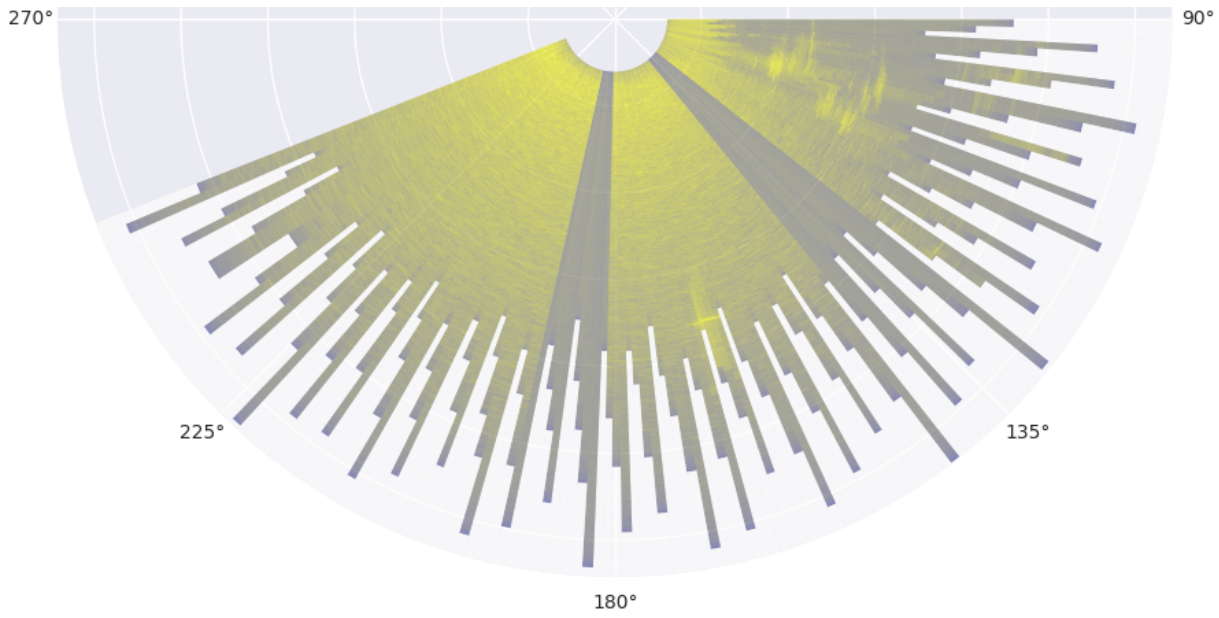

Fig. 8. Power map Saint-Mandrier

We plot on figure 8 a map of the averaged power registered by the radar cell per cell. The power is represented from blue to yellow, yellow meaning that a lot of power has been registered. The two well delimited grey zones in the south may correspond to areas where the radar was in listen-only mode and did not emit any signals.

We then would like to cluster the data cell by cell. As the length of the signals varies from one burst to another, we decided to represent every signal in $\mathbb{D}^{n-1}$ where n denotes the length of shortest signals, and to ignore the last coefficients of reflection of the signals longer than n. As the main Doppler information is contained in the first coefficients of reflection, the loss of information is minor.

The figure 9 represents the result of the k-means clustering for four classes on $\mathbb{D}^{n-1}$ endowed with the Kähler metric. In figure 10, we visualize the clustering result on the four first coefficients of reflection; the colors correspond to those of figure 9. We can see that the clustering mainly depends on the argument of the first reflection coefficient which is related to the average Doppler of the signal as shown in equation(27). In future work we will center the average Doppler of the signals before the clustering to cluster the signals on their shape rather than on their average Doppler.

We chose the k-means algorithm here rather than the p-mean-shift for complexity reasons: the complexity of the k-means algorithm is linear with respect to the number of

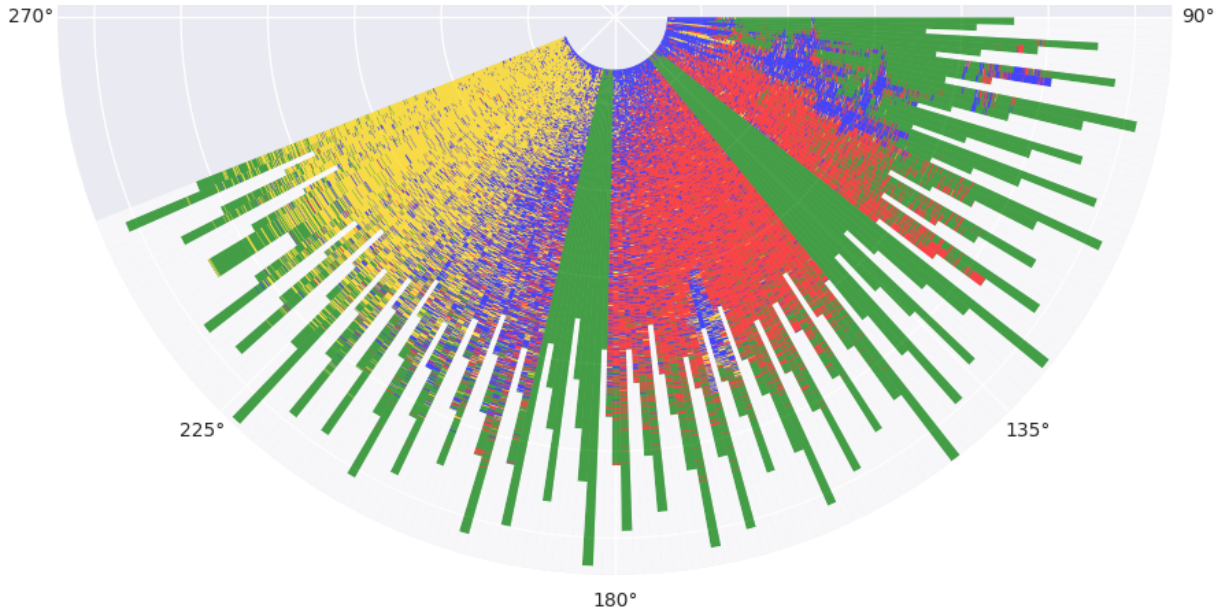

Fig. 9. Clustering map obtained with the k-means algorithm for 4 classes on the reflection coefficients

points of the dataset whereas the complexity of the p-mean-shift algorithm is quadratic, as the mean-shift algorithm. However the p-mean-shift and the mean-shift algorithms are better suited to datasets with nested shapes than the k-means algorithm. The choice of the clustering algorithm is therefore influenced by the dataset. The Tangent Principal Components Analysis can help to visualize the shape of the dataset to choose an appropriate clustering algorithm.

7 Conclusion

We presented a new clustering algorithm: the p-mean-shift. The idea of this algorithm comes from the non-uniqueness of the p-mean for $p \in]0, 1[$ and the classical mean-shift algorithm. This algorithm has been adapted to the Riemannian manifold $\mathbb{D}^{n-1}$ endowed with the Kähler metric. The value of the parameter p is of great influence on the number of clusters obtained; the graphic of the number of clusters as a function of p drawn for our dataset on figure 2 helped us to choose a meaningful value of p. Proposing another clustering algorithm, this article completes [1], [2] in which the k-means algorithm is adapted to the space $\mathbb{D}^{n-1}$ and the articles [3], [4] for the mean-shift algorithm.

These clustering algorithms can help delimiting areas of homogeneous Doppler characteristics. Further work will study the spatial Doppler fluctuations inside these areas of homogeneous Doppler characteristics for finer clutter characterizations.

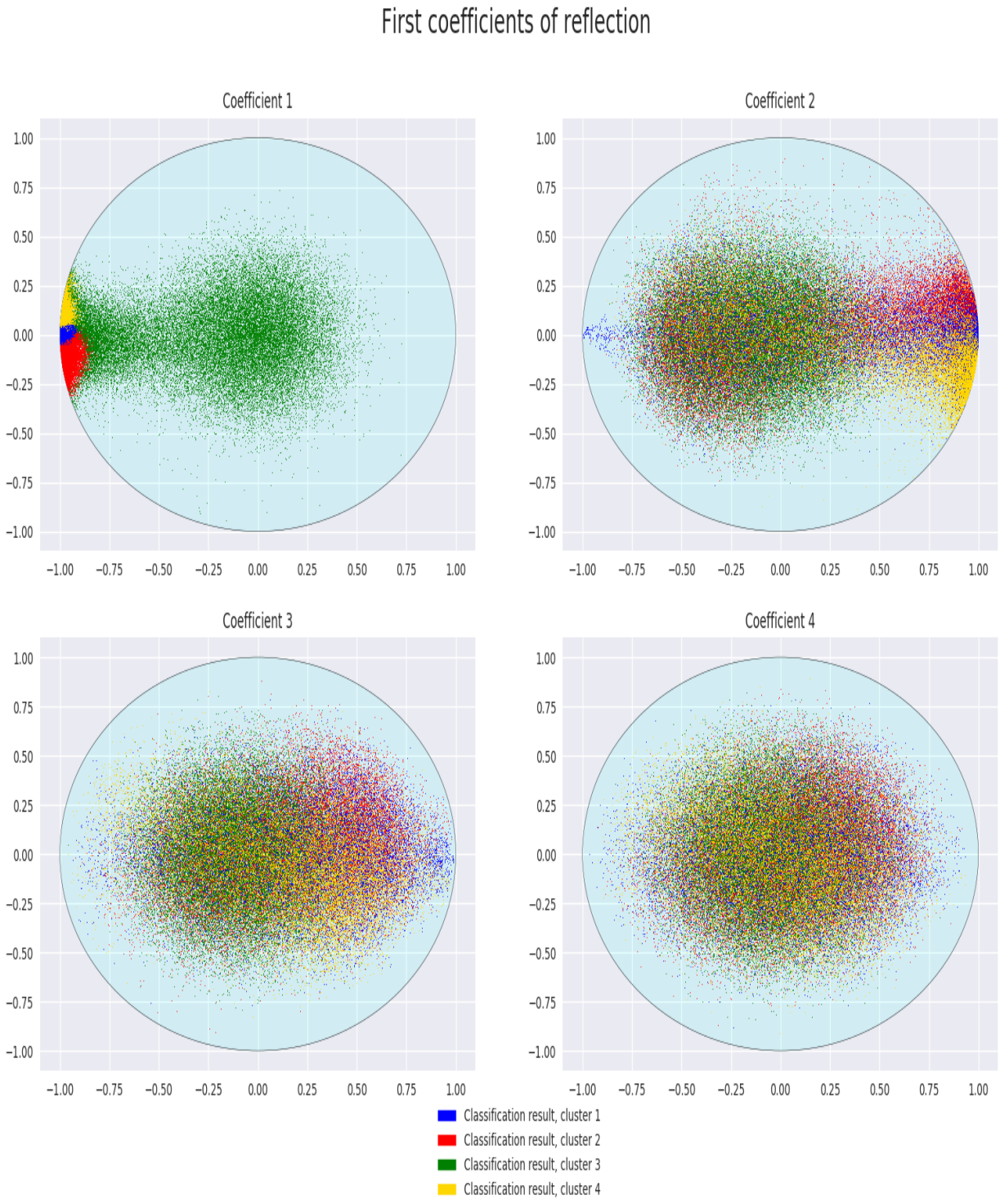

Fig. 10. Visualization of the clustering result obtained with the k-means algorithm for 4 classes on the four first coefficients of reflection

Acknowledgments

We thank the French MoD, DGA / AID for funding (convention CIFRE $N°2017.0008$).

References

1. Yann Cabanes, Frédéric Barbaresco, Marc Arnaudon, Jérémie Bigot, "Toeplitz Hermitian Positive Definite Matrix Machine Learning based on Fisher Metric", IEEE, 2019
2. Yann Cabanes, Frédéric Barbaresco, Marc Arnaudon, Jérémie Bigot, "Non-Supervised High Resolution Doppler Machine Learning for Pathological Radar Clutter", IEEE, 2019
3. Emmanuel Chevallier, Thibault Forget, Frédéric Barbaresco, Jesus Angulo, "Kernel Density Estimation on the Siegel Space with an Application to Radar Processing", Entropy, 2016
4. F. Barbaresco, T. Forget, E. Chevallier, Jesus Angulo, "Doppler spectrum segmentation of radar sea clutter by mean-shift and information geometry metric", 2017.
5. B. Jeuris and R. Vandrebril, "The Kähler mean of Block-Toeplitz matrices with Toeplitz structured blocks", 2016
6. S. Haykin, "Adaptive Filter Theory", Pearson, 2014.
7. Marc Arnaudon, Frédéric Barbaresco, Le Yang, "Riemannian Medians and Means With Applications to Radar Signal Processing", IEEE journal, August 2013.
8. F. Barbaresco, "Super resolution spectrum analysis regularization: Burg, Capon and AGO-antagonistic algorithms", in EUSIPCO-96, Trieste, Italy, 1996, pp. 2005-2008
9. F. Barbaresco, "Information Geometry of Covariance Matrix: Cartan-Siegel Homogeneous Bounded Domains, Mostow/Berger Fibration and Fréchet Median". In Matrix Information Geometry; Bhatia, R., Nielsen, F., Eds.; Springer, 2012; pp. 199256.
10. M. M. Deza, E. Deza, "Encyclopedia of Distances", Springer, 2016, ISBN 978-3-662-52844-0, (http://www.springer.com/us/book/9783662528433)
11. D. Bini, B. Iannazzo, B. Jeuris and R. Vandebril, "Geometric means of structured matrices", BIT, volume 54, issue 1, pp. 55-83, 2014.
12. J. Barrie Billingsley, "Low-Angle Radar Land Clutter, Measurements and Empirical Models", William Andrew Publishing, 2002.
13. Maria S. Greco and Fulvion Gini, "Radar Clutter Modeling".
14. A. Decurninge, F. Barbaresco, "Robust Burg Estimation of Radar Scatter Matrix for Mixtures of Gaussian Stationary Autoregressive Vectors", IET Radar, Sonar & Navigation, Volume 11, Issue 1, January 2017, p. 78 89, 2016
15. Jean-Michel Loubes, Bruno Pelletier, "A kernel-based classifier on a Riemannian manifold", Statistics & Decisions 26, 2008
16. P. Thomas Fletcher, Conglin Lu, Stephen M. Pizer, and Sarang Joshi, "Principal Geodesic Analysis for the Study of Nonlinear Statistics of Shape", IEEE Transactions on Medical Imaging, August 2004.
17. Alice Le Brigant, "Probability on the spaces of curves and the associated metric spaces using information geometry; radar applications", PhD thesis, 2017.