Non-supervised Machine Learning Algorithms for Radar Clutter High-Resolution Doppler Segmentation and Pathological Clutter Analysis

Yann Cabanes*, Frédéric Barbaresco*, Marc Arnaudon, Jérémie Bigot****

*Thales Surface Radar, Advanced Radar Concepts, Limours, FRANCE
emails: yann.cabanes@gmail.com; frederic.barbaresco@thalesgroup.com

**Institut de Mathématiques de Bordeaux, Bordeaux, FRANCE
emails: marc.arnaudon@math.u-bordeaux.fr; jeremie.bigot@math.u-bordeaux.fr

Abstract: *Here we propose a method to classify radar clutter from radar data using a non-supervised classification algorithm. Thus new radars will be able to use the experience of other radars, which will improve their performance: learning pathological radar clutter can be used to fix some false alarm rate created by strong echoes coming from hail, rain, waves, mountains, cities; it will also improve the detectability of slow moving targets, like drones, which can be hidden in the clutter, flying close to the landform.*

1. Introduction

Our aim is to classify the radar clutter cell by cell. The idea is to classify each cell according to its autocorrelation matrix. This autocorrelation matrix is equivalent to coefficients of an autoregressive model, called reflection coefficients, which will be estimated thanks to Burg algorithms. These notions are well explained in [1]. We will then classify the cells according to these reflection coefficients. Finally we will present a classification algorithm called k-means, and test it on simulated data. The non-supervised classification of radar data is a new problem; we present here some promising results.

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.

During a burst, a radar sends a series of pulses. After each pulse, the radar records the echoes. The first echoes coming back represent objects close to the radar, the last echoes represent objects far from the radar. The information coming back after each pulse is therefore divided into cells corresponding to different distances from the radar.

978-3-7369-9860-5

Hence the radar provides us a 2D complex matrix of size $(\#pulses) \times (\#cells)$:

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

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 of the beam reflected by the cell i after the jth pulse.

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

2. 2. Model and hypotheses

In this section, we will focus on a single column of the matrix U defined in equation 1. We will define its autocorrelation matrix and explain how to estimate an equivalent formulation of this 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:

$$\mathbb{E}[u(n)] = 0 \quad \text{for all } n \tag{3}$$

We also assume that this signal can be modeled as an autoregressive Gaussian process.

Interested readers may refer to [2] for a comprehensive course on complex signal processing theory.

2. 3. From input vector to autocorrelation matrix

We define the autocorrelation matrix:

$$\mathbf{R} = \mathbb{E}[\mathbf{u}\ \mathbf{u}^H] \tag{4}$$

$$r_{i,j} = \mathbb{E}[u(k+i)u(k+j)^*] \tag{5}$$

We define the 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.*

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

Proposition 2 (autocorrelation and conjugation)**.**

$$r_{-t} = \mathbb{E}[u(k-t)u(k)^*] = \mathbb{E}[u(k)u(k+t)^*] = \mathbb{E}[u(k+t)u(k)^*]^* = r_t^* \tag{7}$$

Consequence $\mathbf{R}$ is a Toeplitz Hermitian Positive Definite matrix.

$$\mathbf{R} = \begin{bmatrix} r_0 & r_1^* & r_2^* & \dots & r_{n-1}^* \\ r_1 & r_0 & r_1^* & \dots & r_{n-2}^* \\ r_2 & r_1 & r_0 & \dots & r_{n-3}^* \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{n-1} & r_{n-2} & r_{n-3} & \dots & r_0 \end{bmatrix} \tag{8}$$

2. 4. Autocorrelation matrix estimation

$$U = \begin{bmatrix} \boxed{\begin{matrix} u_{0,0} \\ u_{1,0} \\ \vdots \\ u_{n-1,0} \end{matrix}} & \boxed{\begin{matrix} u_{0,1} \\ u_{1,1} \\ \vdots \\ u_{n-1,1} \end{matrix}} & \boxed{\begin{matrix} u_{0,2} \\ u_{1,2} \\ \vdots \\ u_{n-1,2} \end{matrix}} & \begin{matrix} \dots \\ \dots \\ \ddots \\ \dots \end{matrix} & \boxed{\begin{matrix} u_{0,p-1} \\ u_{1,p-1} \\ \vdots \\ u_{n-1,p-1} \end{matrix}} \end{bmatrix} \\ \begin{matrix} \downarrow \\ \widehat{\mathbf{R}_0} \end{matrix} \quad \begin{matrix} \downarrow \\ \widehat{\mathbf{R}_1} \end{matrix} \quad \begin{matrix} \downarrow \\ \widehat{\mathbf{R}_2} \end{matrix} \quad \begin{matrix} \downarrow \\ \widehat{\mathbf{R}_{p-1}} \end{matrix} \tag{9}$$

In our classification problem, the autocorrelation matrix will be estimated cell by cell.

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:

$$\widehat{r_t} = \frac{1}{n-t} \sum_{k=0}^{n-1-t} u(k+t)u(k)^* \quad t = 0, ..., n-1 \tag{10}$$

Note that this method is unprecise when the vector length n is small, especially when the lag t is close to $n-1$. We will propose a more robust method to estimate the autocorrelation matrix with few data, based on an autoregessive model.

Burg algorithm The Burg algorithm principle is to minimize the forward and the backward prediction errors. The regularised Burg algorithm of order M and regularization coefficient γ is detailed in [3], [4] as follows:

Initialization:

$$f_{0,k} = b_{0,k} = u_k \quad k = 0, ..., n-1 \tag{11}$$

$$a_{0,k} = 1 \quad k = 0, ..., n-1 \tag{12}$$

$$p_0 = \frac{1}{n} \sum_{k=0}^{n-1} |u_k|^2 \tag{13}$$

Iteration: for $i = 1, ..., M$:

$$\mu_i = -\frac{\left(\frac{2}{n-i} \sum\limits_{k=i}^{n-1} f_{i-1,k} \bar{b}_{i-1,k-1} + 2 \sum\limits_{k=1}^{i-1} \beta_{k,i} a_{k,i-1} a_{i-k,i-1}\right)}{\left(\frac{1}{n-i} \sum\limits_{k=i}^{n-1} |f_{i-1,k}|^2 + |b_{i-1,k-1}|^2 + 2 \sum\limits_{k=0}^{i-1} \beta_{k,i} |a_{k,i-1}|^2\right)} \tag{14}$$

where:

$$\beta_{k,i} = \gamma (2\pi)^2 (k-i)^2 \tag{15}$$

$$\begin{cases} a_{k,i} &= a_{k,i-1} + \mu_i \bar{a}_{i-k,i-1} \quad k = 1, ..., i-1 \\ a_{i,i} &= \mu_i \end{cases} \tag{16}$$

and

$$\begin{cases} f_{i,k} &= f_{i-1,k} + \mu_i b_{i-1,k-1} \quad k = i, ..., n-1 \\ b_{i,k} &= b_{i-1,k-1} + \bar{\mu}_i f_{i-1,k} \quad k = i, ..., n-1 \end{cases} \tag{17}$$

The above Burg regularized algorithm allows us to transform the original data into a power factor in $\mathbb{R}_+^*$ and reflection coefficients in $\mathbb{D}^{n-1}$, where $\mathbb{D}$ represents the complex unit disk.

According to [1], the following transformation is a bijection:

$$\begin{aligned} &\mathcal{T}_n^+ \rightarrow \mathbb{R}_+^* \times \mathbb{D}^{n-1} \\ &R_n \mapsto (p_0, \mu_1, ..., \mu_{n-1}) \end{aligned} \tag{18}$$

where $\mathcal{T}_n^+$ denotes the set of Toeplitz Hermitian Positive Definite matrices.

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

2. 5. The Kähler metric

According to the previous bijection, we will represent a Toeplitz Hermitian definite positive 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 [5]. In the Encyclopedia of Distance by Deza [6], this distance is called Barbaresco distance:

$$\begin{aligned} d^2_{\mathcal{T}_n^+}(T_1, T_2) &= d^2_{\mathcal{T}_n^+}\left((p_{0,1}, \mu_{1,1}, ..., \mu_{n-1,1}), (p_{0,2}, \mu_{1,2}, ..., \mu_{n-1,2})\right) \\ &= n \log^2\left(\frac{p_{0,2}}{p_{0,1}}\right) + \sum_{l=1}^{n-1} \frac{n-l}{4} \log^2\left(\frac{1 + \left|\frac{\mu_{l,1} - \mu_{l,2}}{1 - \mu_{l,1}\mu_{l,2}^*}\right|}{1 - \left|\frac{\mu_{l,1} - \mu_{l,2}}{1 - \mu_{l,1}\mu_{l,2}^*}\right|}\right) \end{aligned} \tag{19}$$

2. 6. The Kähler mean

The Kähler mean of $(T_0, ..., T_{m-1})$ is defined as the point X such that the following function f, sum of the squared distances from X to T_i, reaches its minimum:

$$f(X) = \sum_{i=0}^{m-1} d^2(X, T_i) \tag{20}$$

Note 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:

$$\begin{array}{ccccccccc} T_0 & \mapsto & (& \boxed{p_{0,0},} & & \boxed{\mu_{1,0},} & \cdots, & \boxed{\mu_{n-1,0}} &) \\ \vdots & & & \boxed{\vdots} & & \boxed{\vdots} & & \boxed{\vdots} & \\ T_{m-1} & \mapsto & (& \boxed{p_{0,m-1},} & & \boxed{\mu_{1,m-1},} & \cdots, & \boxed{\mu_{n-1,m-1}} &) \\ & & & \downarrow & & \downarrow & & \downarrow & \\ T & \leftarrow & (& p_0, & & \mu_1, & \cdots, & \mu_{n-1} &) \end{array} \tag{21}$$

The Kähler mean algorithm is performed in [7] as a gradient descent on the function f, which is equivalent to a gradient descent on each coordinate. At each step of the algorithm, once the gradient is computed, we move on $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ following its geodesics.

3. Simulation model

Each cell is simulated independently from the others. For each cell, we simulate a complex vector using a SIRV (Spherically Invariant Random Vectors) model:

$$Z = \underbrace{\sqrt{\tau} R^{1/2} x}_{\text{information coming from the environment}} + \underbrace{b_{radar}}_{\text{noise coming from the radar itself}} \tag{22}$$

with:

τ**:** clutter texture (positive real random variable).

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

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

To choose the matrix R, we learn experimentally from radar measures the spectrum shape of the clutter we want to simulate. The scaled autocorrelation coefficients of the matrix R can then be computed from the spectrum using the inverse Fourier transform.

See [8], [9], for more details about clutter modeling.

4. Classification problem

4. 1. Methodology

Using the previous model, we simulated 100 vectors with the model parameters (τ_1, R_1) and 100 vectors with the model parameters (τ_2, R_2), τ_i being a random process and R_i a constant matrix. On the left graphic in figure 1, we plot the FFT of the simulated

vectors, the 100 vectors simulated with the model parameters (τ_1, R_1) being represented by the first 100 rows, the 100 vectors simulated with the model parameters (τ_2, R_2) being represented by the last 100 rows. The classification result on the right graphic is obtained using the k-means algorithm presented below.

Then we tried to recover the simulation model parameters from the model outputs thanks to Burg algorithm. In this paper, we classify the data only on the scaled autocorrelation matrix R, represented by the reflection coefficients $(\mu_1, ..., \mu_{n-1})$. In figure 2 we plot the second coefficient of reflection of each cell. Future work might also use the texture parameter τ represented by the power coefficient p_0 to classify the data.

Finally we use a clustering algorithm and plot in figure 3 the normalized confusion matrix C of the classification result. The classification being non-supervised, a perfect classification would give either the diagonal matrix I_2 or the antidiagonal matrix J_2. From this matrix, we compute the distance between the perfect result $R_{perfect}$ and the classification result $R_{classification}$ as follows:

$$d(R_{perfect}, R_{classification}) = \frac{1}{2}\, min(\|I_2 - C\|_1, \|J_2 - C\|_1) \tag{23}$$

$$0 \leqslant d(R_{perfect}, R_{classification}) \leqslant 1 \tag{24}$$

The performance of the classification algorithm on the samples is computed as follows:

$$P = 1 - d(R_{perfect}, R_{classification}) = 1 - \frac{1}{2}\, min(\|I_2 - C\|_1, \|J_2 - C\|_1) \tag{25}$$

$$0 \leqslant P \leqslant 1 \tag{26}$$

Note that the performance is computed on a randomly generated sample using the model described previously.

4. 2. k-means on $\mathbb{D}^{n-1}$ with the Kähler metric

The algorithm k-means for N clusters:

Initialization: Pick randomly N points in the dataset. They now represent the barycenters of each class.

During each loop the barycenter of each class moves in two steps:

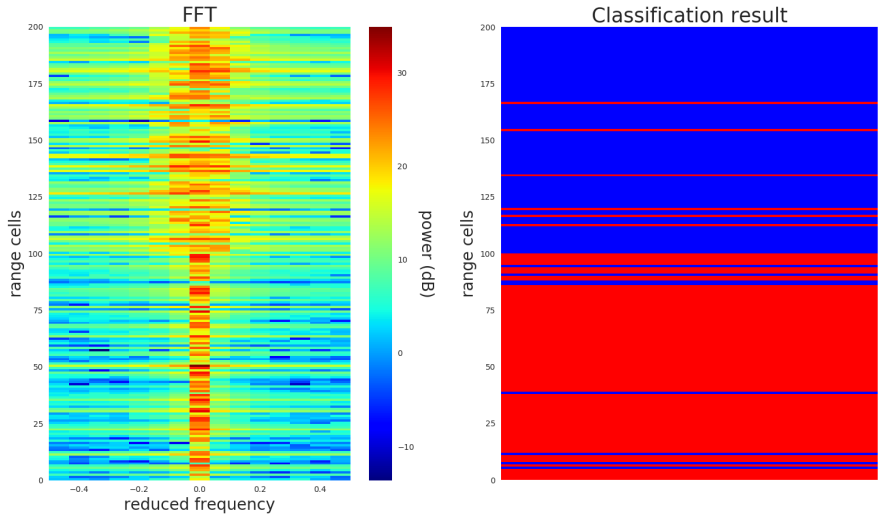

Figure 1: FFT and classification results, k-means on $\mathbb{D}^{n-1}$, Kähler metric

- Assign each point of the dataset to the closest barycenter.
- Compute the new barycenter of each class.

Note that the k-means algorithm complexity is linear with respect to the amount of data.

Note also that the result of this algorithm depends on arbitrary factors:

* The initialization of the algorithm is random.
* The number of loops of the algorithm.
* The number of loops and the step size of the Kähler mean approximation.

Predictions Once an effective k-means algorithm is developed, we can easily predict the class of the new radar data: they will be assigned to the cluster having the closest barycenter.

5. Conclusion

We developed a k-means algorithm to classify the radar clutter. This algorithm has been adapted to the Kähler metric and has given promising results. Here we clustered the data using only the scaled autocorrelation matrix R; the normalized Burg algorithm presented in [10] might help to take into account the texture coefficient τ for future work. More

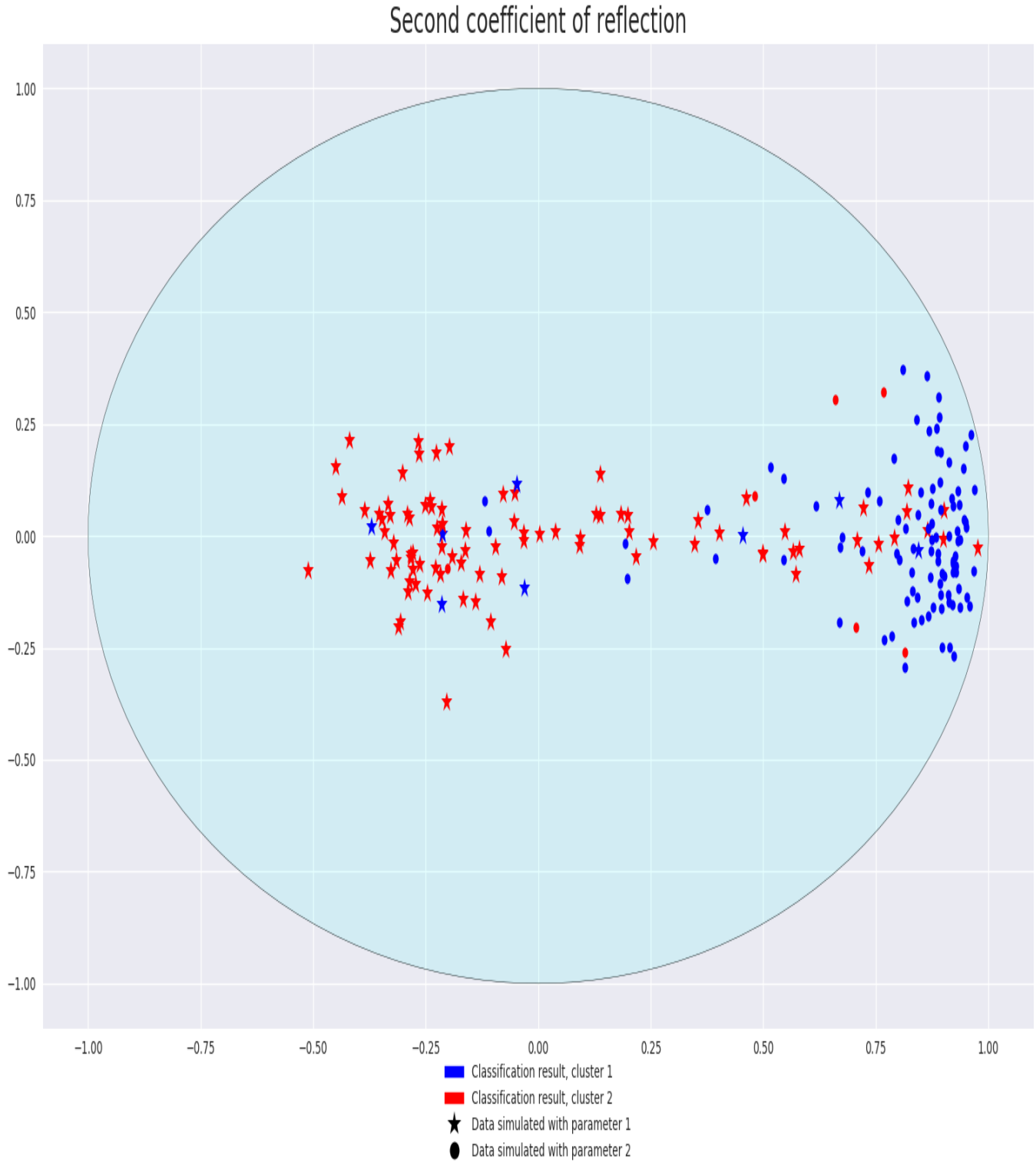

Figure 2: Second coefficient of reflection, k-means on $\mathbb{D}^{n-1}$, Kähler metric

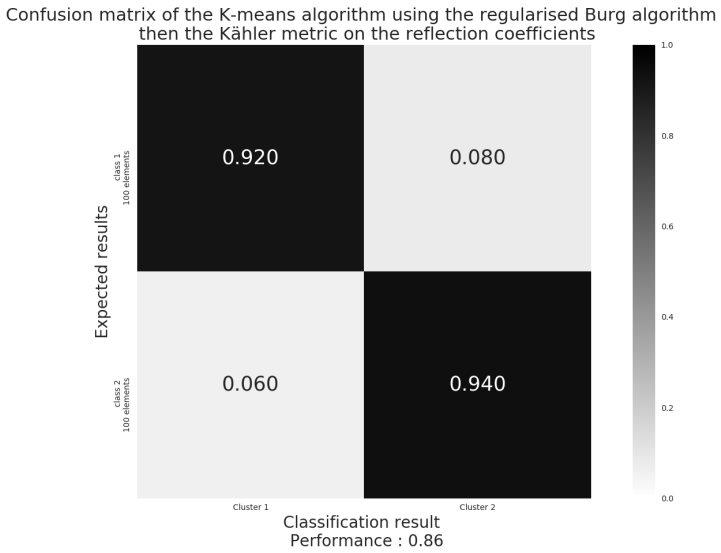

Figure 3: Confusion matrix, k-means on $\mathbb{D}^{n-1}$, Kähler metric

classification algorithms will be adapted to the Kähler metric to deal with clusters of unusual shapes, like the mean-shift algorithm presented in [11].

Acknowledgments

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

References

[1] B. Jeuris and R. Vandebril, “The Kähler mean of Block-Toeplitz matrices with Toeplitz structured blocks”, 2016

[2] S. Haykin, “Adaptive Filter Theory”, Pearson, 2014.

[3] M. Arnaudon, F. Barbaresco and L. Yang, “Riemannian Medians and Means With Applications to Radar Signal Processing”, IEEE journal, August 2013.

[4] F. Barbaresco, “Super resolution spectrum analysis regularization: Burg, Capon and AGO-antagonistic algorithms”, in EUSIPCO-96, Trieste, Italy, 1996, pp. 2005-2008

[5] 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. 199–256.

[6] M. M. Deza, E. Deza, “Encyclopedia of Distances”, Springer, 2016, ISBN 978-3-662-52844-0, (http://www.springer.com/us/book/9783662528433)

[7] D. Bini, B. Iannazzo, B. Jeuris and R. Vandebril, ”Geometric means of structured matrices”, BIT, volume 54, issue 1, pp. 55-83, 2014.

[8] J. Barrie Billingsley, “Low-Angle Radar Land Clutter, Measurements and Empirical Models”, William Andrew Publishing, 2002.

[9] Maria S. Greco and Fulvion Gini, “Radar Clutter Modeling”.

[10] 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

[11] F. Barbaresco, T. Forget, E. Chevallier, Jesus Angulo, “Doppler spectrum segmentation of radar sea clutter by mean-shift and information geometry metric, 2017.