Non-Supervised High Resolution Doppler Machine Learning for Pathological Radar Clutter

Yann Cabanes, Frédéric Barbaresco, Marc Arnaudon, Jérémie Bigot

Non-Supervised High Resolution Doppler Machine Learning for Pathological Radar Clutter

Yann Cabanes, Frédéric Barbaresco, Marc Arnaudon, Jérémie Bigot

Non-Supervised High Resolution Doppler Machine Learning for Pathological Radar Clutter

Yann Cabanes
Advanced Radar Concepts
Thales Surface Radar
Limours, France
yann.cabanes@gmail.com

Frédéric Barbaresco
Advanced Radar Concepts
Thales Surface Radar
Limours, France
frederic.barbaresco@thalesgroup.com

Marc Arnaudon
Institut de Mathématiques de Bordeaux
Bordeaux, France
marc.arnaudon@math.u-bordeaux.fr

Jérémie Bigot
Institut de Mathématiques de Bordeaux
Bordeaux, France
jeremie.bigot@math.u-bordeaux.fr

***Abstract*—In this paper we propose a method to classify radar clutter from radar data using a non-supervised classification algorithm. As a final objective, new radars will therefore be able to use the experience of other radars to improve their performances: 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.**

***Index Terms*—radar clutter, machine learning, non-supervised classification, k-means, autocorrelation matrix, Burg algorithm, reflection coefficients, Kähler metric**

I. Introduction

Our aim is to classify the radar clutter cell by cell, a cell being an area of small lenght in a direction in which the radar sends a series of pulses. A cell is represented by a temporal sequence of complex numbers corresponding to the amplitude a phase of the echoes coming back from this area to the radar. The idea is to classify each cell according to the autocorrelation matrix of the complex temporal signal. In [1], this autocorrelation matrix is said to be equivalent to coefficients of an autoregressive model, called reflection coefficients, which will be estimated thanks to Burg algorithms. We will then classify the cells according to these reflection coefficients. The non-supervised classification of radar data is dealt in [2] with a mean-shift algorithm. Here we will present another classification algorithm called k-means, and test it on simulated data, showing promising results.

II. Introduction to signal processing theory

A. 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 $(\#impulses) \times (\#cells)$:

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

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 distance index i from the radar, at time index j (jth impulse).

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

B. Model and hypotheses

We will now 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:

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

the one dimensional complex signal registered in a cell.

We assume this signal to be stationary with zero mean:

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

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

Interested readers may refer to [3] for a comprehensive course on complex signal processing theory.

C. 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 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.

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

Proposition 2 (autocorrelation and conjugation):

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

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

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

Note that the assumptions made in section II-B that the signal can be modeled as a complex stationary autoregessive Gaussian process with zero mean has the following equivalent formulation : $\mathbf{u} = \mathbf{R}^{1/2}x$ with $\mathbf{R}$ a Toeplitz Hermitian Positive Definite matrix and x a standard complex Gaussian random vector which dimension is equal to the number of pulses.

D. Autocorrelation matrix estimation

In our classification problem, the autocorrelation matrix $\mathbf{R}_i$ will be estimated independently for each cell $\mathbf{u}_i$:

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

a) 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 now propose a more robust method to estimate the autocorrelation matrix with few data, based on an autoregessive model.

b) 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 described in algorithm 1 and detailed in [4], [5].

Algorithm 1 regularised Burg algorithm

Initialization:

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

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

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

for $i = 1, ..., M$: **do**

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

where:

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

$$\left\{ \begin{array}{rcll} a_{k,i} & = & a_{k,i-1} + \mu_i \bar{a}_{i-k,i-1} & k = 1, ..., i-1 \\ a_{i,i} & = & \mu_i \end{array} \right. \tag{16}$$

and

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

end for

return $(p_0, \mu_1, ..., \mu_{n-1})$

The regularized Burg algorithm allows us to transform the original complex vector $\mathbf{u}$ 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:

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

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

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

E. The Kähler metric

According to the previous bijection, we will represent a Positive Definite Hermitian Toeplitz matrix T_i by the corresponding coefficients $(p_{0,i}, \mu_{1,i}, ..., \mu_{n-1,i})$.

The following distance on $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ has been introduced by F. Barbaresco in [6]:

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

In the Encyclopedia of Distance by Deza [7], this distance is called Barbaresco distance.

F. The Kähler mean

The computation of the mean is an important step of the k-means algorithm described below. Our first task is to give a meaningful definition the mean in a non-Euclidean metric space.

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:

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

We can see in equation 19 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 [8] as a gradient descent on the function f, which is equivalent to a gradient descent on each coordinate. Remember that the space $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ is not endowed with the Euclidean metric, so this is not a classical gradient descent in a Euclidean space but a gradient descent in the Riemannian manifold $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ endowed with the Kähler metric. At each step of the algorithm, once the gradient is computed (it is an element of the tangent space), we move on $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ following its geodesics.

III. 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 [9], [10] for more details about clutter modeling.

IV. Classification problem

A. Methodology

Using the previous model, we simulate 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. 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 classify these vectors using a k-means algorithm described in the next section. The k-means algorithm is a classical clustering algorithm in Euclidean spaces, the main difficulty was to adapt it to the Riemannian manifold $\mathbb{D}^{n-1}$ endowed with the Kähler metric. In figure 1 , we plot the FFT of each simulated vector on the left graphic, each FFT being drawn horizontally; the vertical axis represents the different cells along the distance axis. On the graphic in the middle of figure 1, we plot the result of the corresponding k-means clustering. We present in figure 2 the result of the clustering on the first coefficients of reflection.

Once the classification result is obtained, we compute the F1 score of the classification. The F1 score is a way to measure the performance of a supervised classification algorithm. We adapted it to our non-supervised classification algorithm by doing all possible permutations in the classification results labels in order to find the best matching with the expected results; the best matching being defined as the best F1 score.

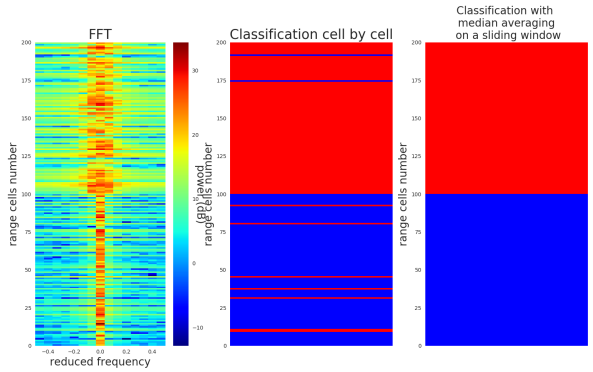

Figure 1. FFT and classification results, k-means on $\mathbb{D}^{n-1}$, Kähler metric

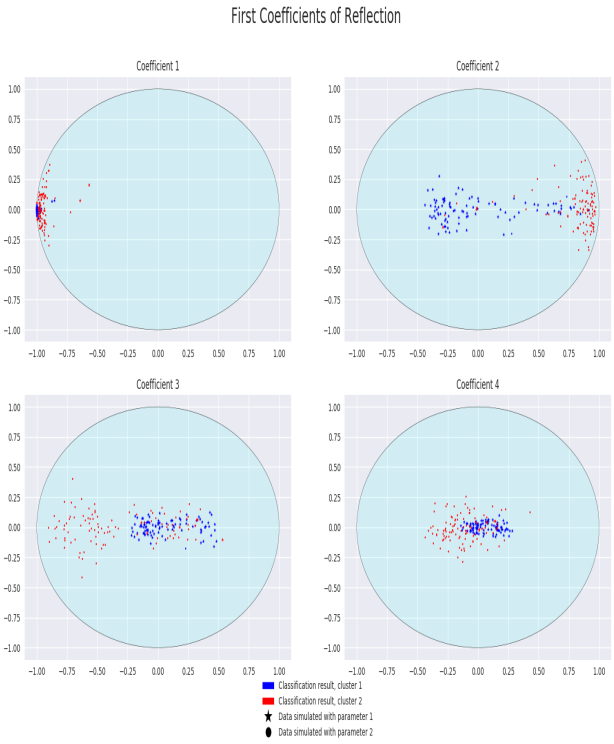

Figure 2. First coefficients of reflection, k-means on $\mathbb{D}^{n-1}$, Kähler metric

Finally we plot the normalized confusion matrix using the labels corresponding to this best F1 score, see figure 3.

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

a) The algorithm: The k-means algorithm is described in algorithm 2.

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. This can affect the limit value of the barycenters. To solve this problem, we run the algorithm several times and keep the classification result with the most compact clusters.
- The number of loops of the algorithm.
- The number of loops and the step size of the Kähler mean approximation.

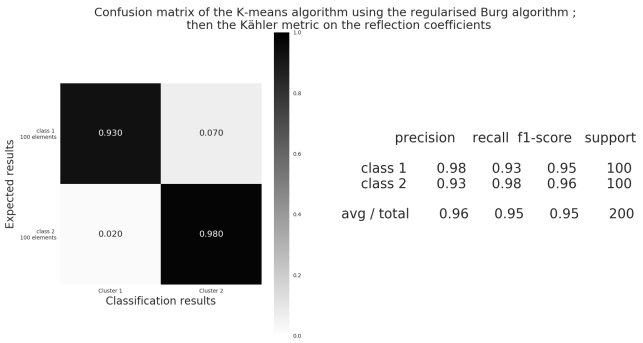

Figure 3. Confusion matrix and F1 score, k-means on $\mathbb{D}^{n-1}$, Kähler metric

Algorithm 2 k-means algorithm for N clusters

Initialization:
Pick randomly N points in the dataset. They now represent the barycenters of each class.
for i = 1 to m **do**
Assign each point of the dataset to the closest barycenter.

Compute the new barycenter of each class.
end for
return Each point is labeled according to the closest barycenter.

b) 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.

C. Median averaging

During all this study, we classified the data cell by cell, regardless of the spatial positioning of the data, each cell being considered independently from its neighbours. If we assume that each cell is correlated to the neighbouring cells, we can avoid missclassification due to outliers by associating to each data an average of its neigbouring cells, and performing the classification on the averaged data.

In figure 1, the graphic on the right represents the classification result given by a sliding window of size 9 (the classification result was perfect). In each window, we compute a median of the data in $\mathbb{D}^{n-1}$. The median of a set of points $(x_1, x_2, ..., x_n)$ in a metric space (E, d) is defined as follows:

$$median(x_1, x_2, ..., x_n) = argmin_{x \in E} \sum_{i=1}^{n} d(x, x_i) \qquad (23)$$

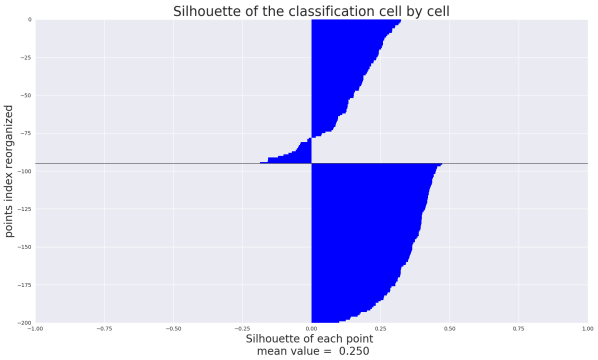

Figure 4. Silhouette of the classification cell by cell using k-means on $\mathbb{D}^{n-1}$ endowed with the Kähler metric

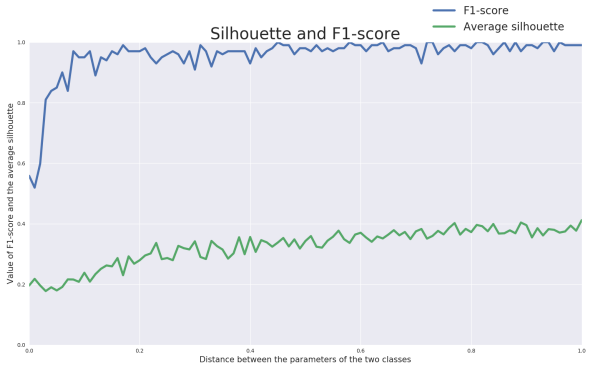

Figure 5. Evolution of the F1 score and the silhouette mean with the distance between the parameters of the model used to simulate the two classes

The median is more robust to outliers than the mean, the mean being the point minimizing the sum of squared distances. We then select the closest points of the barycenter to get rid of outliers, keeping half of the points, and compute the new median of these selected points. The center cell of the sliding window is now represented by this last median. Interested readers will find in [11] an algorithm to compute the median of several points in $\mathbb{D}^{n-1}$ endowed with the Kähler metric.

D. Silhouette

On real radar data, one of the challenges is to decide the number of clusters k we would like from the k-means clustering. We now introduce a tool to guide us in the choice of the number of clusters k.

Figure 4 show a graphical display called silhouette. The silhouette gives an idea of shape of the clusters.

We now define the silhouette $s(i)$ of a point i. Take any point i in the data set, and denote by A the cluster to which it has been assigned. When cluster A contains other points apart from i, then we can compute:

$a(i) =$ average dissimilarity of i to all other objects of A.

If A contains only one point i, we set $s(i) = 0$ as a neutral choice.

Let us now consider a cluster C different from A, and compute:

$d(i, C) =$ average dissimilarity of i to all objects of C.

After computing $d(i, C)$ for all clusters C different from A, we select the smallest of those numbers, and denote it by:

$b(i) = min_{C \neq A} d(i, C)$

$b(i)$ represents how far i is from the cluster that would have been the second best choice for i. The silhouette number $s(i)$ is then computed as follows:

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{24}$$

The silhouette measures how much i is closer from the cluster A than the cluster that would have been the second best choice for i.

We always have $-1 \leqslant s(i) \leqslant 1$.

The silhouette gives indications about the shape of the clusters given by the classification algorithm. Clusters with small radius far from the other clusters will give silhouette values close to 1.

However, the silhouette does not give any information about the performance of the classification agorithm. Most of the time, if the clusters overlap, a perfect classification will give a worst silhouette than the k-means clustering. Moreover, well separated clusters with non-spherical interlaced shapes can also have a low mean silhouette value.

Interested readers can read [12] for a reference about the silhouette construction, and [2] for another radar application.

When the clusters do not overlap, the average silhouette value can by used as a criterion to determine the number of clusters: we classify the data with different number of clusters and keep the classification giving the best mean silhouette value.

In figure 4, the points have been ordered in each cluster according to their silhouette value, in decreasing order.

We plot in figure 5 the evolution of both the F1 score and the silhouette mean according to the distance between the parameters of the model used to simulate the two classes. As expected, they both increase as the simulated clusters are spliting.

V. 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. Future work may also take into account the texture coefficient τ; the normalized Burg algorithm presented in [13] might help to take this texture coefficient τ into consideration. More clustering algorithms will be adapted to the Kähler metric to deal with clusters of unusual interlaced shapes, like the mean-shift algorithm presented in [2], [14].

Acknowledgments

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

REFERENCES

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

[2] Emmanuel Chevallier, Thibault Forget, Frédéric Barbaresco, Jesus Angulo, “Kernel Density Estimation on the Siegel Space with an Application to Radar Processing”, Entropy, 2016

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

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

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

[6] 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.

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

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

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

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

[11] Marc Arnaudon, Frédéric Barbaresco, Le Yang, “Riemannian medians and means with applications to radar signal processing”, IEEE transactions on signal processing

[12] Peter J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis”, Journal of Computational and Applied Mathematics 20, 1987, North-Holland, pp. 53-65

[13] 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

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