

Toeplitz Hermitian Positive Definite Matrix Machine Learning based on Fisher Metric

Yann Cabanes^{1,2}, Frédéric Barbaresco¹, Marc Arnaudon², and Jérémie Bigot²

¹ Thales Surface Radar, 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. Here we propose a method to classify radar clutter from radar data using an unsupervised classification algorithm. The data will be represented by Positive Definite Hermitian Toeplitz matrices and clustered using the Fisher metric. Once the clustering algorithm dispose of a large radar database, new radars will be able to use the experience of other radars, which will improve their performances: learning 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.

Keywords: radar clutter, machine learning, unsupervised classification, k-means, autocorrelation matrix, Burg algorithm, reflection coefficients, Kähler metric

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. 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 unsupervised classification of radar data is dealt in [2] with a mean-shift algorithm. Here we will present another clustering algorithm called k-means, and test it on simulated data, showing 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.

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

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

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 (j th impulse).

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:

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

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.

2.3 From input vector to autocorrelation matrix

We define the autocorrelation matrix:

$$\mathbf{R} = \mathbb{E}[\mathbf{u} \mathbf{u}^H] \quad (4)$$

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

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 .*

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

Proposition 2 (autocorrelation and conjugation).

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

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

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

Note that the assumptions made in section 2.2 that the signal can be modeled as a complex stationary autoregressive 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.

2.4 Autocorrelation matrix estimation

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

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

$$\begin{array}{cccc}
 \downarrow & \downarrow & \downarrow & \downarrow \\
 \widehat{\mathbf{R}}_0 & \widehat{\mathbf{R}}_1 & \widehat{\mathbf{R}}_2 & \widehat{\mathbf{R}}_{p-1}
 \end{array} \tag{9}$$

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:

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

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

Algorithm 1 regularised Burg algorithm

Initialization:

$$f_{0,k} = b_{0,k} = u_k \quad k = 0, \dots, n - 1 \quad (11)$$

$$a_{0,k} = 1 \quad k = 0, \dots, n - 1 \quad (12)$$

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

for $i = 1, \dots, M$: **do**

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

where:

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

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

and

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

end for

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

The regularized Burg 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:

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

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

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

2.5 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 will represent a Toeplitz Hermitian Positive Definite matrix T_i by the corresponding coefficients $(p_{0,i}, \mu_{1,i}, \dots, \mu_{n-1,i})$. The following distance has been introduced by F. Barbaresco in [6] on the set $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ to make this bijection an isometry. In the Encyclopedia of Distance by Deza [7], this distance is called Barbaresco distance:

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

The equations of the geodesics of the set $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ endowed with the Kähler metric are described in [4].

2.6 The Kähler mean

The Kähler mean of (T_0, \dots, T_{m-1}) is defined as the point T_{mean} such that the following function $f(T) = \sum_{i=0}^{m-1} d^2(T, T_i)$, sum of the squared distances from T to T_i , reaches its unique minimum.

The Kähler mean algorithm is performed in [4], [8] as a gradient descent on the function f . The gradient expression of f is:

$$\vec{\nabla} f(T) = \sum_{i=0}^{m-1} 2 \vec{\nabla} d(T, T_i) d(T, T_i) = 2 \sum_{i=0}^{m-1} -\frac{\overrightarrow{T T_i}}{d(T, T_i)} d(T, T_i) = -2 \sum_{i=0}^{m-1} \overrightarrow{T T_i} \quad (20)$$

where $\vec{\nabla}$ denotes the gradient operator and $\overrightarrow{T T_i}$, also written $exp_T^{-1}(T_i)$, denotes the element of the tangent space of the manifold $\mathbb{R}_+^* \times \mathbb{D}^{n-1}$ at T such that the geodesic starting at T at time 0 with initial tangent vector $\overrightarrow{T T_i}$ arrives at T_i at time 1.

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

$$\begin{array}{ccc} T_0 & \mapsto & \left(\begin{array}{c|c|c} p_{0,0} & \mu_{1,0} & \dots & \mu_{n-1,0} \\ \vdots & \vdots & & \vdots \end{array} \right) \\ \vdots & & \\ T_{m-1} & \mapsto & \left(\begin{array}{c|c|c} p_{0,m-1} & \mu_{1,m-1} & \dots & \mu_{n-1,m-1} \\ \vdots & \vdots & & \vdots \end{array} \right) \\ & & \begin{array}{c} \downarrow \quad \downarrow \quad \downarrow \\ T \leftarrow (\quad p_0, \quad \mu_1, \quad \dots, \quad \mu_{n-1} \quad) \end{array} \end{array} \quad (21)$$

The gradient descent on the function f is therefore 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:

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

with:

τ : clutter texture coefficient (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.

The radar noise b_{radar} is assumed to be small enough in comparison with the information coming from the environment $\sqrt{\tau} R^{1/2} x$ for estimating the autocorrelation matrix τR using the methods described in section 2.4.

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 the clutter modeling.

4 Classification problem

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

Each vector is now represented by its reflection coefficients 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 a visualization of the clustering on the first coefficients of reflection.

Once the clustering is done, 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 unsupervised classification algorithm by doing all possible permutations in the classification results labels in order to find the best matching with the expected results. Finally we plot on figure 3 the normalized confusion matrix using the labels corresponding to this best matching.

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

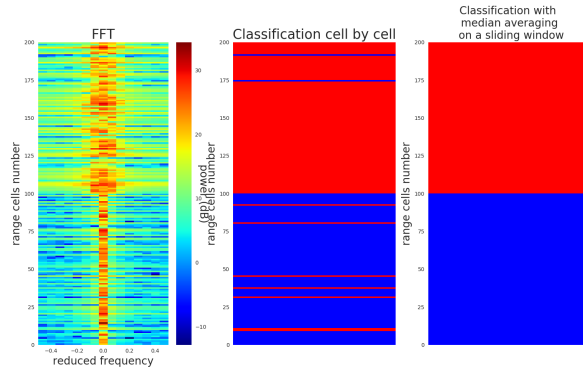


Fig. 1. FFT and classification results, k-means on \mathbb{D}^{n-1} , Kähler metric

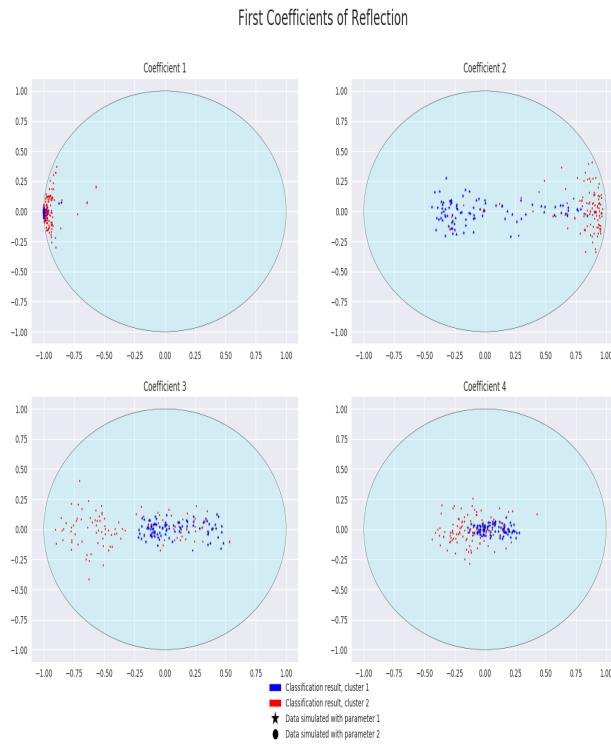


Fig. 2. First coefficients of reflection, k-means on \mathbb{D}^{2n-1} , Kähler metric

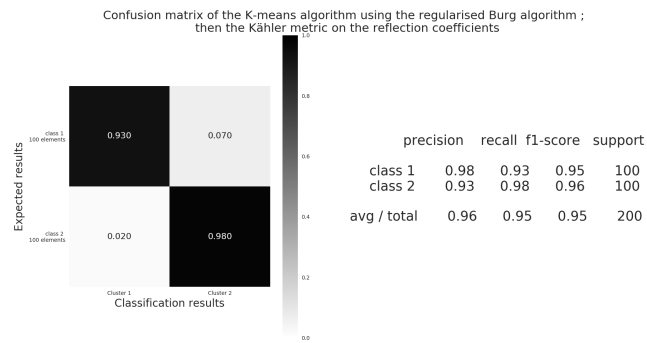


Fig. 3. Confusion matrix and F1 score, k-means on \mathbb{D}^{2n-1} , Kähler metric

The algorithm The k-means algorithm is described in algorithm 2.

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 loop number **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.

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.

4.3 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 neighbouring 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, \dots, x_n) in a metric space (E, d) is defined as follows:

$$\text{median}(x_1, x_2, \dots, x_n) = \underset{x \in E}{\operatorname{argmin}} \sum_{i=1}^n d(x, x_i) \quad (23)$$

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 reader will find in [11] an algorithm to compute the median of several points in \mathbb{D}^{n-1} .

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. Future work may also take into account the texture coefficient τ ; the normalized Burg algorithm presented in [12]

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], [13]. These clustering algorithms will also be used to cluster groups of neighbouring cells: we will use a multidimensional spatial autoregressive model to represent the data (the autocorrelation matrices will be Positive Definite Block-Toeplitz matrices) and adapt our clustering algorithms to this higher dimensional space [1], [14].

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. 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. 199–256.
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. 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
13. F. Barbaresco, T. Forget, E. Chevallier, Jesus Angulo, “Doppler spectrum segmentation of radar sea clutter by mean-shift and information geometry metric, 2017.
14. F. Barbaresco, “Radar Micro-Doppler Signal Encoding in Siegel Unit Poly-Disk for Machine Learning in Fisher Metric Space”, *IRS’18*, Bonn, June 2018