Exploring Complex Time-series Representations for Riemannian Machine Learning of Radar Data

Daniel A Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, Matthieu Cord

EXPLORING COMPLEX TIME-SERIES REPRESENTATIONS FOR RIEMANNIAN MACHINE LEARNING OF RADAR DATA

Daniel A. Brooks[*†], *Olivier Schwander*[†], *Frédéric Barbaresco*[*], *Jean-Yves Schneider*[*], *Matthieu Cord*[†]

[*]Thales Air Systems, GBU LAS, Advanced Radar Concepts
Limours, FRANCE
email: daniel.brooks@lip6.fr

[†]Sorbonne Université, CNRS, LIP6 - Laboratoire d'Informatique de Paris 6, F-75005
Paris, FRANCE

ABSTRACT

Classification of radar observations with machine learning tools is of primary importance for the identification of non-cooperative radar targets such as drones. These observations are made of complex-valued time series which possess a strong underlying structure. These signals can be processed through a time-frequency analysis, through their self-correlation (or covariance) matrices or directly as the raw signal. All representations are linked but distinct and it is known that the input representation is critical for the success of any machine learning method. In this article, we explore these three possible input representation spaces with the help of two kinds of neural networks: a temporal fully convolutional network and a Riemannian network working direcly on the manifold of covariances matrices. We show that all the considered input representations are a particular case of a generic machine learning pipeline which goes from the raw complex data to the final classification stage through convolutional layers and Riemannian layers. This pipeline can be learnt end-to-end and is shown experimentally to give the best classification accuracy together with the best robustness to lack of data.

Index Terms— Neural networks, Radar, Information geometry, Riemannian machine learning

1. INTRODUCTION AND RELATED WORKS

The classification of non-cooperative targets with machine learning tools is a task of interest for radar applications, in particular with the numerous drones which are more and more widespread in the airspace. As the success or failure of machine learning often depends on the input representation of the data, the study and evaluation of the various input space is critical. In their raw and most general representation, the signals we consider take the form of a complex-valued time series, representing a temporal evolution of phase and amplitude associated to a given physical phenomenon. In this paper, we focus on radar waves, but the developed insights remain valid for any signal following the above definition. The analysis of any such signal is not usually done on its raw temporal form but rather on a spectral or mixed temporal-spectral representation, ie a Fourier transform. The Fourier transform, whether it be purely frequencial or varying in time, exhibits a plethora of properties allowing it to reveal even to the naked eye many interesting characteristics of the signal at hand [1]: as such, perhaps the most popular representation for radar data analysis is the windowed Fourier transform, or spectrogram, also called micro-Doppler signature [2], which takes the form of an image where spatial locality is replaced by temporal and frequencial locality. Recent radar classification schemes thus apply modern Computer Vision algorithms on micro-Doppler signatures, such as convolutional neural networks [3] (CNNs) or recurrent networks [4] (RNNs).

Furthermore, within the theoretical developments in the time-frequency literature lies the central concept of auto-correlation, or that of covariance [5] [6]: intuitively, much of a signal's characteristics is underpinned by how the different points in time correlate with each other. As such, more recent research has focussed on studying the correlations themselves, more specifically the covariance matrices sampled from an underlying process, which is naturally modeled as Gaussian [7] [8] [9]. Covariance matrices possess a particular geometric structure, that of being symmetric positive definite (SPD), ie they belong to the SPD matrix Riemannian manifold. The statistical analysis of manifold-valued data requires to generalize usually straightforward concepts in machine learning such as convolution, separating hyperplanes or even subtraction to a differential geometric, Riemannian theoretical framework. Differential analysis on statistical manifolds constitutes the field of Information Geometry; interested readers may refer to [10], [11] or [12] for further information. Initial developments in classification on $\mathcal{S}_*^+$ focussed on minimum distance to mean (MDM) approaches [13] [14] [15]. More recent proposals introduced a neural network, called SPDnet, respecting the manifold ge-

ometry [16]. As an SPDnet directly deals with the covariance of data, we also call it a second-order model.

We have thus two different classification methods, first and second order neural networks, which can work on different representations of the input data: the convolutional network can use the raw complex signal or a time-frequency representation and the SPDNet works on a covariance matrix which can be computed either from the raw complex signal, on the frequency features of a spectrogram or on the feature map built by convolutional layers.

In this paper, we propose a machine learning pipeline which encompasses all mentioned natural representations of a time series. More specifically, we describe a neural network architecture which combines a first order part, and a second order part and we show that the aforementioned representations can be seen as bifurcations inside this architecture, obtained by computing a covariance matrix at different stages of the architecture. The earlier bifurcation being the use of a SPDNet directly on the auto-correlation matrix and the latest being the end-to-end learning of the time-frequency representation through convolution layers before the use of a SPDNet. We show experimentally that the complete end-to-end architecture leads to the best classification accuracy and to the best robustness to lack of data (by reducing the size of the training set to only 5% of its original size).

Our contributions are the following:

- an analysis of the possible input representations of a complex-valued time series,
- a neural network architecture which encompass these representations,
- an experimental analysis on a radar dataset, with experiments focusing on the size of the training set.

The following sections are organized as follows. First we recall the machine learning tools used in our pipeline (complex convolution, SPDNet and fully convolutional networks). Secondly we describe our pipeline and the possible bifurcations leading to different input representations. Finally, we show experimental results on a radar dataset.

2. LEARNING ON RADAR REPRESENTATIONS

Radar data can be seen from various viewpoints, all rooted in sound theoretical background, and different deep learning models may operate independently on the different representations. The first one is the raw complex time series, for which complex filter banks may be learned in an end-to-end fashion. The second is the spectrogram, or windowed Fourier transform, upon which modern CNNs can be applied. The third is the auto-correlation, or covariance, of the time series, which requires a particular kind of networks, called SPDnets.

2.1. Learning complex filter banks

First we recall the discrete Fourier transform (DFT) of a discrete complex signal x of duration N:

$$\forall k \leq N,\ DFT_x(k) = \sum_{l \leq N} x(l) e^{-2i\pi k \frac{l}{N}} \tag{1}$$

Thus, a windowed Fourier transform of duration n is exactly equivalent to the convolution of the signal by the n Fourier atoms, which are the n base vectors of n-th-roots of the unit, $(e^{-2i\pi k \frac{\cdot}{n}})_{k \leq n}$. From this point of view, the DFT can be seen as an initial convolutional layer in a CNN dealing directly with spectrograms. Thus, instead of fixing the DFT, we prepend to a micro-Doppler CNN a $1D$ complex convolutional layer initialized with the Fourier atoms, and allow the weights to be learnable, ie we allow Fourier atoms to vary to best accomodate the optimization crtiterion. By doing so, we allow additional flexibility in the learning process, all the while utilizing the expert knowledge of a proper time-frequency representation. On a technical note, the optimization of a complex convolution layer in itself constitutes a digression to traditional Machine Learning, and is described in [17].

2.2. Learning on SPD matrices

The particularity of SPD matrices lies in their inherent belonging to the curved Riemannian manifold $\mathcal{S}_*^+$: as such, not taking the underlying geometric information into account leads to poor results in practice. Here we briefly describe the recently introduced SPDnets [16]: similarly to a classical neural network, an SPDnet aims at building a hierarchical sequence of more compact and discriminative manifolds. More specifically, an input matrix $X^{(l-1)}$ at layer $(l-1)$ yields $P^{(l)}$ through a bilinear mapping (BiMap layer), which in turn yields $X^{(l)}$ at layer (l) through rectified eigenvalues (ReEig layer) activations according to the following formulas:

$$\begin{aligned} &P^{(l)} = {W^{(l)}}^T X^{(l-1)} W^{(l)} \text{ with } W^{(l)} \in \mathcal{O}(n_{l-1}, n_l) \\ &X^{(l)} = U^{(l)} \max(\Sigma^{(l)}, \epsilon I_n) {U^{(l)}}^T \text{ with } P^{(l)} = U^{(l)} \Sigma^{(l)} {U^{(l)}}^T \end{aligned} \tag{2}$$

In the equations above, $\mathcal{O}(n_{l-1}, n_l)$ is the manifold of semi-orthogonal rectangular matrices, $P^{(l)} = U^{(l)} \Sigma^{(l)} {U^{(l)}}^T$ designates the eigenvalue decomposition of $P^{(l)}$, and ϵ is a fixed threshold for the eigenvalues. The final feature manifold is then transformed via a logarithmic mapping to a Euclidean space (LogEig layer) to perform the actual classification. We illustrate such an SPDnet architecture in Figure 1.

The main difficulties of learning an SPDnet lie both in the backpropagation through structured Riemannian functions, and in the manifold-constrained optimization. Interested readers may refer to [18] and [19] for the detailed underlying mechanism of an SPDnet.

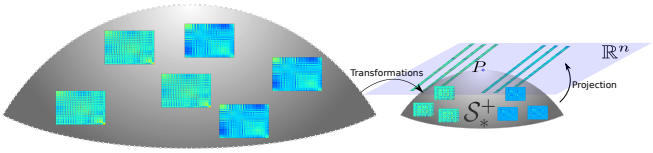

Fig. 1. Illustration of a generic SPD neural network. Successive bilinear layers followed by activations build a feature SPD manifold, which is then transformed to a Euclidean space to allow for classification.

2.3. Fully-temporal neural network

In this paper, we propose to build a neural architecture which conserves a temporal structure throughout the layers, the aim being to be able to study the covariance of the features at any level in the network. Although convolutions do preserve temporal structure, the final layers of a typical CNN being fully-connected, that structure is lost, precisely when we reach the most representative feature spaces in the network. The conservation of the temporal structure requires the usage of fully-convolutional networks (FCN), first introduced by [20] in the context of Computer Vision, and first adapted by [3] in the context of radar classification. The consequence of being fully convolutional is that, having transformed the raw complex time series to a learnt spectral representation, then transformed this representation in a traditional deep learning fashion, we can then model and analyse the covariance of the learnt features themselves as a final temporal representation of the signal.

3. FULL PIPELINE

In this section we show, and illustrate in figure 2, how branching through the pipeline leads to different models on the signal representations.

3.1. Possible bifurcations

As a first note, never bifurcating to covariance analysis amounts to a fully-temporal convolutional network taking as input the raw signal, sequentially building a spectral and hierarchical feature representations. If the DFT is allowed to be fine-tuned, we call such an architecture a filter-bank learning network (FilterNet); if not, we simply refer to it as an FCN. For classification, Global Average Pooling (GAP) is used to pool the feature maps' temporal evolution to a single dimension. However, as mentioned before it is possible to branch out at any stage, precisely thanks to the fully-temporal property of the network. So far, we have identified three meaningful temporal representations of the signal, ie three reasonable entry points for covariance analysis: the raw complex time-series, the spectrogram and the learnt feature representation. Either can be modeled by their covariances

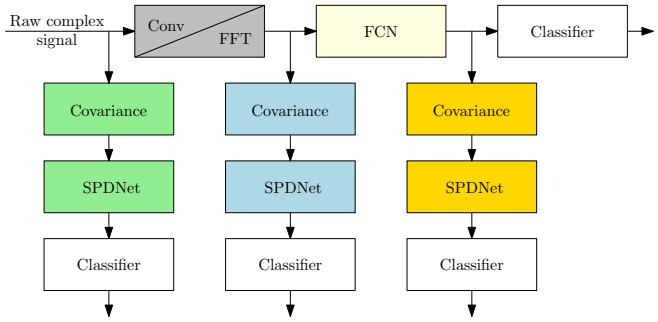

Fig. 2. Illustration of the global pipeline proposed for the classification of time-frequency signals. Possible bifurcations to covariance analysis stem either from the raw data, the spectral representation or the learnt temporal features. The first convolutional layer can be either learnt fom the data or used as a Fourier transform.

and handled by an independent SPDnet. We respectively call the corresponding second-order architectures, SPDnet, SpectroSPD and fully-temporal covariance network (FTCovNet).

3.2. Covariance pooling

A key module linking the first- and second-order models is the covariance pooling (CovPool): given an n-dimensional signal sampled during N timesteps $s = (s_i)_{i \leq N}$, where $s_i \in \mathbb{R}^n$, its covariance matrix $X \in \mathcal{S}_*^+(n)$ is estimated as:

$$X = \sum_{i \leq N} \bar{s}_i \bar{s}_i^T \tag{3}$$

In the equation above, often referred to as maximum likelihood estimation (MLE), $\bar{s}_i$ is the centered version of s_i.

One problem to be addressed is that the covariance of the raw complex signal as is, is scalar. We overcome this problem by considering the raw signal as a series of possibly overlapping windowed elementary sub-signals, introducing effective multi-variability through the length L of the window. Thus, one s_i in equation 3 becomes $[x(l) \cdots x(l+L)]^T$. In essence, the windowing is similar to performing a windowed Fourier transform, and can also be represented as a convolution, for which the weights are no longer the Fourier atoms but adequately placed zeros and ones.

Another problem may arise from the MLE estimation: theoretically, for X to be regular, ie positive definite, it is required to have $n \geq N$, ie to observe more samples than their dimension. This can be problematic in estimating the covariance of the learnt features, ie when the s_i in equation 3 are the FCN's final feature maps f_i. In this scenario, the temporal sampling has been reduced through the FCN's convolutions and possible poolings, while feature size could have been set to arbitrarily large dimensions. Readers may refer

Table 1. Performance comparison of SPDnets on radar data.

Train size	100%	20%	5%
SPDnet	92.6 ± 0.54	91.5 ± 0.74	88.4 ± 3.06
FCN	98.9 ± 0.44	93.4 ± 1.21	84.3 ± 2.51
FilterNet	99.4 ± 0.17	96.2 ± 1.12	87.4 ± 1.94
SpectroSPD	95.1 ± 0.49	91.9 ± 0.82	84.6 ± 3.49
FTCovNet	99.5 ± 0.16	97.2 ± 0.90	93.9 ± 0.74

to [21], [22] and [23] for details on how to best extrapolate covariance from CNN features: in summary, it is possible to use robust estimators such as described in [24], to split the feature maps in smaller groups, or even to introduce an transitional convolutional layer to learn a dimension reduction suited for the covariance pooling.

4. EXPERIMENTS

In this section we test the proposed pipeline on a simulated radar dataset developed in [3], which consists in 3 classes of drones, each class totaling 4 minutes of radar recordings dispatched in 1000 signals of 250ms long sampled at 4kHz, thus of 1000 signals of 1000 rax complex samples long, per class. Throughout the experiments, we study the data through a sliding 5ms window of 20 samples with 50%-overlap. For instance, this means the covariance of one signal is estimated over 99 windows of length 20. The same windowing is used for covariance and spectral representations to keep comparisons fair. We bifurcate the introduced pipeline at various stages in various configurations, which amounts to different learning models, specifically the following:

1. SPDnet: a SPDnet on the raw complex data's covariance, estimated over 99 windows of 20 samples;
2. SpectroSPD: a SPDnet on the spectrograms;
3. FCN: a FCN on spectrograms as proposed in [3];
4. FilterNet: a FCN on raw complex data as described above, where Fourier filter banks are fine-tuned;
5. FTCovNet: a fully-temporal covariance network, where an SPDnet is appended to the the final feature representation of the FCN.

Furthermore, we repeat the experiments with decreasing amount of training data in the hope that injecting geometric information in the learning would compensate for lack in data volume. All networks are trained with stochastic gradient descent with a 0.9 momentum and learning rate varying from $5e^{-3}$ to $5e^{-2}$, over a 5-fold cross-validation using a 20% validation split. Results are displayed in table 1.

The first remark is that the SPDnet yields the worst accuracy when all training data is available. Intuitively, it makes sense that it score lower than SpectroSPD and FTCovNet as the covariance is then sampled from a more adapted or discriminative model, ie respectively a spectrogram and FCN learnt features. As for the FCN and FilterNet, these are deep learning models with a total of 13647 parameters or more, whereas the SPDnet is the Riemannian equivalent a rather shallow network, with only 704 parameters.

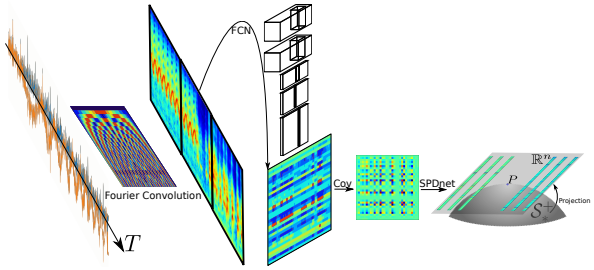

Fig. 3. Illustration of the fully temporal covariance network, the best model derived from the proposed pipeline.

However, as training data decreases, we see the SPD methods become better than the traditional deep models: they seem to exhibit much higher robustness to lack of data, which validates the usefulness of exploiting the geometric information of the data.

Comparing the FCN and FilterNet's performances across the different amounts of training data, FilterNet slightly yet consistently scores higher, validating the usage of complex filter bank learning. Future works could include a fully complex FCN.

The final observation concerns the FTCovNet, which performs best across all configurations. In a sense, this model benefits both from deep temporal feature learning, and from covariance geometry modeling. We illustrate the full FTCovNet in figure 3

5. CONCLUSION

We have introduced a neural pipeline consisting of a first-order fully convolutional network, onto which a second-order SPD neural network can be appended at any stage of the first-order model. We find that the deeper down the first-order model, the better the performance of the global second-order model. Furthermore, models exploiting covariance structure seem to be more robust to lack of data than the first-order models, even outperforming them in critically sparse scenarios (with only a few seconds of training data per class). Finally, the end-to-end second-order model outperforms all other models in any of the experimented configurations, which points to the definite possibility of getting the best from the two worlds: a set of discriminative features learnt by convolutional layers, and a Riemannian processing on the resulting temporal covariance matrices.

6. REFERENCES

[1] Shannon C. E., "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, July 2013.

[2] Victor C. Chen, Fayin Li, S.-S. Ho, and Harry Wechsler, "Micro-Doppler effect in radar: phenomenon, model, and simulation study," *IEEE Transactions on Aerospace and electronic systems*, vol. 42, no. 1, pp. 2–21, 2006.

[3] Daniel A. Brooks , Olivier Schwander , Frédéric Barbaresco , Jean-Yves Schneider , and Matthieu Cord , "Temporal Deep Learning for Drone Micro-Doppler Classification," .

[4] Pavlo Molchanov, Ronny I.A. Harmanny, Jaco J.M. de Wit, Karen Egiazarian, and Jaakko Astola, "Classification of small UAVs and birds by micro-Doppler signatures," *International Journal of Microwave and Wireless Technologies*, vol. 6, no. 3-4, pp. 435–444, June 2014.

[5] Patrick Flandrin, *Time-Frequency/Time-Scale Analysis, Volume 10*, Academic Press, Inc., Orlando, FL, USA, 1st edition, 1998.

[6] Franz Hlawatsch and François Auger, *Time-Frequency Analysis*, John Wiley & Sons, Mar. 2013, Google-Books-ID: tOeeJyP95IQC.

[7] Le Yang, Marc Arnaudon, and Frédéric Barbaresco, "Riemannian median, geometry of covariance matrices and radar target detection," Nov. 2010, pp. 415–418.

[8] Frédéric Barbaresco, "Information Geometry of Covariance Matrix: Cartan-Siegel Homogeneous Bounded Domains, Mostow/Berger Fibration and Fréchet Median," in *Matrix Information Geometry*, pp. 199–255. Springer, Berlin, Heidelberg, 2013.

[9] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multiclass BrainComputer Interface Classification by Riemannian Geometry," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 4, pp. 920–928, Apr. 2012.

[10] Shun-ichi Amari, *Information Geometry and Its Applications*, Applied Mathematical Sciences. Springer Japan, 2016.

[11] Colin Atkinson and Ann F. S. Mitchell, "Rao's Distance Measure," *Sankhy: The Indian Journal of Statistics, Series A (1961-2002)*, vol. 43, no. 3, pp. 345–365, 1981.

[12] Frédéric Barbaresco, "Jean-Louis Koszul and the elementary structures of Information Geometry," in *Geometric Structures of Information Geometry*. SPRINGER, 2018.

[13] A. Le Brigant, F. Barbaresco, and M. Arnaudon, "Geometric barycenters of time/Doppler spectra for the recognition of non-stationary targets," in *2016 17th International Radar Symposium (IRS)*, May 2016, pp. 1–6.

[14] Marc Arnaudon, Frédéric Barbaresco, and Le Yang, *Medians and Means in Riemannian Geometry: Existence, Uniqueness and Computation*, Nov. 2011.

[15] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten, "Classification of covariance matrices using a Riemannian-based kernel for BCI applications," *Neurocomputing*, vol. 112, pp. 172–178, July 2013.

[16] Zhiwu Huang and Luc J. Van Gool, "A Riemannian Network for SPD Matrix Learning.," in *AAAI*, 2017, vol. 1, p. 3.

[17] Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, Joao Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal, "Deep complex networks," in *International Conference on Learning Representations*, 2018.

[18] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu, "Matrix Backpropagation for Deep Networks with Structured Layers," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, Dec. 2015, pp. 2965–2973, IEEE.

[19] A. Edelman, T. Arias, and S. Smith, "The Geometry of Algorithms with Orthogonality Constraints," *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, Jan. 1998.

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[21] "DeepKSPD: Learning Kernel-matrix-based SPD Representation for Fine-grained Image Recognition," .

[22] Dinesh Acharya, Zhiwu Huang, Danda Pani Paudel, and Luc Van Gool, "Covariance Pooling for Facial Expression Recognition," 2018, pp. 367–374.

[23] Kaicheng Yu and Mathieu Salzmann, "Second-order Convolutional Neural Networks," *arXiv:1703.06817 [cs]*, Mar. 2017, arXiv: 1703.06817.

[24] Olivier Ledoit and Michael Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *Journal of Multivariate Analysis*, vol. 88, no. 2, pp. 365–411, Feb. 2004.