CLUTTER IDENTIFICATION BASED ON KERNEL DENSITY ESTIMATION AND SPARSE RECOVERY

by

Haokun Wang

B.E in Electrical Engineering and Automation, Tongji University,

2016

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2018

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Haokun Wang

It was defended on

April 5 2018

and approved by

Murat Akcakaya, Ph D., Assistant Professor, Electrical and Computer Engineering

Zhi-Hong Mao, Ph D., Associate Professor, Electrical and Computer Engineering

Natasa Miskov-Zivanov, Ph D., Assistant Professor, Electrical and Computer Engineering

Thesis Advisor: Murat Akcakaya, Ph D., Assistant Professor, Electrical and Computer Engineering

CLUTTER IDENTIFICATION BASED ON KERNEL DENSITY ESTIMATION AND SPARSE RECOVERY

Haokun Wang, M.S.

University of Pittsburgh, 2018

Many existing radar algorithms are developed under the hypothesis that the environment (clutter) is stationary. However, in real applications, the statistical characteristics of the clutter might change immensely in space, time, or both, depending on the radar-operational scenarios. If unaccounted for, these non-stationarities may extremely hamper the radar performance. Therefore, to overcome such performance degradations, we have developed a cognitive radar framework to dynamically detect changes in the clutter characteristics, and to adapt to these changes by identifying the new clutter distribution. In this work, we present a sparse-recovery based clutter identification technique. In this technique, we build a dictionary matrix of well-known clutter statistics such that each column of the matrix is a kernel density estimation of a specific clutter distribution. When radar measurements arrive, sparse recovery, more specifically, orthogonal matching pursuit (OMP) algorithm is used to identify the distribution of the radar measurements by matching the kernel density estimation of the measurements to one of the columns of the dictionary matrix. We analyze the effect of different kernels and distance measures between the kernel density estimations on the clutter identification accuracy. With numerical examples, we demonstrate that the sparse-recovery based method provides high accuracy in clutter identification and this technique is robust to changes in the training and test sample sizes.

Keywords:Clutter identification, Sparse recovery, KDE, OMP, Ozturk .

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

First of all, I would like to thank my advisor Dr. Murat Akcakaya for all the support he has given me, both academically and personally. I am gratefully for all of the advice and suggestions he has made throughout this entire process. I would also like to thank Dr. Zhi-Hong Mao and Dr.Natasa Miskov-Zivanov for being part of my committee to review my thesis work. I started the radar research with fundamental knowledge of signal processing, a lot of ideas and skills are obtained during our continuously weekly meeting and discussion, I always get helpful perspective from these group members. And I'm proud to work with them. I would also like to thank all of our collaborators from Washington University in St. Louis who have worked closely with us through our research. Additionally, all our collaborators have given endless advice and suggestions and have helped to make this project successful. Without their help it would not be possible to finish this project. I am also grateful for all the time they have spent discussing and revising our works. Thanks to Satyabrata Sen, Malia Kelsey and Yijian Xiang for all of the support. Finally, I would like to thank everyone I have met and worked with since arriving in Pittsburgh, it's a great pleasure to make them acquaintance.

1.0 INTRODUCTION

In the late 90's the idea about knowledge-based systems and agile waveform design was first introduced, which becomes the foundation of cognitive radar[9, 6], a modern concept introduced by Haykin in 2006[24]. The definition of cognitive radar quoted by Haykin "A cognitive radar continuously learns about the environment through experience gained from interaction with the environment, the transmitter adjust its illumination of the environment in an intelligent manner, the whole radar system constitutes a dynamic closed feedback loop encompassing the transmitter, environment, and receiver"[24].

As a distinctive features of the human brain, cognition distinguishes human being from all other mammalian species. It is not absurd that when mention cognitive control, we will naturally consider cognitive control in the brain. More importantly, cognitive control works on the executive part in brain, conversely coupled to its perception through the working memory[31]. With this three-fold combination, which resulted to the perception-action cycle that assimilate the environment, thereby formed as a closed-loop feedback system[31]. In engineering realization, the cognition is two way approach, one is inside-out and another is outside-in. Based on the source information inside or outside the receiver, the two approach could be different from the commonly understanding. If the prior knowledge of the environment is applied, as part of the receiver, then it's the inside-out approach, where the prior knowledge is depended on the application. On the contrary, the outside-in approach could be seen as short-term memory, which is updated by the receiver. The radar-scene analyzer initiate the cognition responding to the information obtained from the environment either by radar itself or other sensors[31, 24].

First feature of cognitive radar is perception, the system would sense the environment and got trained from significant information of the target and background. Second feature

is surveillance and tracking, after obtaining the information about the environment, it will adapt the transmitted signal for optimally matching. In application design, the receiver is the learning component, it iterate the information learned from experience via the interaction with the environment. The transmitter is the adaption part, which transform the feature of the environment in a way of optical in agreement with the information transited via receiver. And the feed back loop, like many control system, coordinate the performance of receiver and transmitter in a synchronous way[25].

Detecting and tracking targets in the presence of nonstationary clutter, noise, and interference have been the most pertinent and challenging problems in radar systems. In the practical scenarios, various issues, such as the terrain and weather conditions, dynamics of the targets, and hostile electronic environments, may fluctuate and alter the statistical characteristics of the environmental background (clutter) during the radar operation period [5]. These nonstationarities of the clutter, if not adaptively coped with, can significantly hinder the performance of the classical detection and tracking techniques [27, 49]. For example, in the target detection problem, the fluctuation of the clutter distribution parameters may require to readjust the threshold of the detector, and even worse, a change of the family of the clutter distributions may require to redesign the detector altogether, in order to maintain the optimal or nearly optimal detection performance.

Traditionally, however, in most of the radar applications, the target detection and tracking techniques have been developed with a specific clutter distribution, which is assumed to be known a priori and be stationary throughout the entire processing period. Although the Gaussian distributions are used extensively to represent the clutter characteristics, it has been shown in the literature that the Gaussian representations suffer from performance deterioration when the the measured clutter data are heavy-tailed [30, 36]. Instead, some compound-Gaussian distributions, such as K distribution and Student-t distribution, are proposed to accurately model the received clutter, for example, sea- or foliage-clutter when radars operate in high-resolution and/or low-grazing-angle modes [4, 42, 7, 51, 21]. Weibull and lognormal distributions are other two popular clutter distributions that achieve great fitness to the real data, and are capable of modeling the spiky nature of the clutter [44, 12]. In nonstationary operating scenarios, however, a prior knowledge of the clutter characteristics

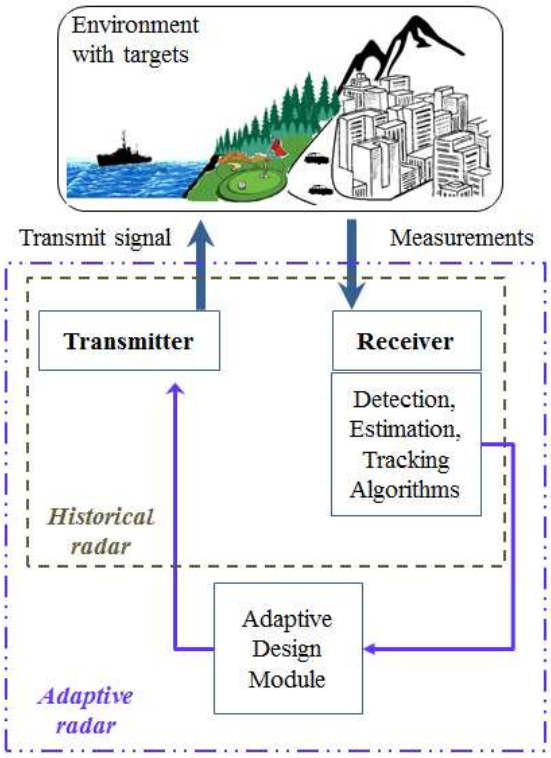


Figure 1: Block diagram of cognitive radar

represented using a fixed, parametric distribution does not hold true anymore as the clutter statistics may drastically alter during analysis. Therefore, having a capability of determining the clutter distribution on-the-fly would be crucial to maintain, or even to improve, the radar detection/tracking performance in nonstationary environments.

In order to create such an adaptive framework, recent advances in computational capabilities have allowed radar system designers to consider the design of more complex and intelligent systems, termed as cognitive radar systems [8, 24]. Cognitive radars aim to early detect the changes of the environments (clutter), precisely learn the new distribution of clutter, and adaptively update the detection/tracking algorithms for maintaining or bettering the performances achieved by current (nonadaptive) state-of-the-art systems. In our previous work[5], we proposed a data-driven method and used the (extended) CUSUM algorithm to address the first issue in the cognitive radar framework, i.e., finding out whether the modeled/assumed clutter distribution has changed or not. For the second issue, we previously developed a sparse recovery based clutter identification method [28, 53], that applies the kernel density estimation (KDE) and a batch orthogonal matching pursuit (BOMP) method to identify the distribution of the received clutter data based on a pre-learned dictionary of distributions. Earlier, another clutter identification method, namely the Ozturk algorithm [35, 39], was proposed to identify the received clutter distribution as the nearest neighbor to a dictionary of distributions after transforming each distribution to a point on two-dimensional plane. The Ozturk algorithm is used to transform different distributions into a point in two- dimensional space. These points in the two-dimensional space are used to built a library, and subsequently to identify the clutter distributions. While this method is able to identify the K-distribution with low shape values with an accuracy of about 70%, it does not perform well as the shape parameter increases. However, comparing the performances of the sparse-recovery based clutter identification approach (i.e., BOMP method) with that of the Ozturk algorithm, we have shown [53] that the BOMP method has (i) improved accuracy in identifying clutter distributions that have different parameters, but are from the same family; and (ii) robustness in terms of measurements used for dictionary generation and test distribution identification.

To further explore the potentials of the sparse-recovery based clutter identification, in the paper, we investigate the effects of different kernels types and kernel parameters used in KDE on the clutter identification accuracy, while only the normal kernel with a default parameter was considered in our previous work. We observe that the BOMP method is robust to the kernel bandwidth selection, the Epanechnikov kernel is found to be the most suited kernel for the BOMP algorithm and canberra metric presented to be the best atom selection measurement.

The rest of the paper is organized as follows. In Section 2, the basic idea about the dictionary learning and sparse recovery are introduced. In Section 3, we describe the sparse recovery based clutter identification method and provide the details of the kernel density estimation approach. In Section 4, we demonstrate various simulation results to objectively compare the effects of different types of kernels and kernel parameters. Section 4 provides a discussion of the observed results, and Section 5 concludes the paper.

2.0 DICTIONARY LEARNING AND SPARSE RECOVERY

2.1 DICTIONARY LEARNING

For radar signals, mixed with many environment noise, making it largely redundant in two main parts:1) The multiple correlated versions of the same scenario are contained, 2) Each version of the scenario is densely sampled by sensors. Compared with the recorded data, the relevant information about the underlying process which form the observation is much reduced. The dictionary learning method determines the proper features of the data through subspace with reduced dimension. It could be applied to both features of the signal and processing task. This method is based on the idea that observation could be viewed as a sparse subset chosen from a redundant dictionary[45, 33, 1]. The dictionary learning method focuses on the dictionary of atoms building algorithm which give effective representation of groups of signals. For most of the dictionary learning problems, the degree of sparsity is the key constraints to that.

Sparse approximation is made to present a signal y with a specific dimension n as a linear combination of a few number of atoms from the dictionary. Typically, the element of the dictionary is unit norm function, the atom. The dictionary could be denote as D and the atom ϕ_k, $k = 1, ...N$, where N denotes the dictionary size. It should be no surprise that the dictionary is over complete and its atoms are all linear dependent, which means every signal could be viewed as a linear combination of atoms in dictionary[45, 33, 1, 48, 18]. Since the dictionary is redundant, it's impossible to set a unique combination of the dictionary. To make the problem applicable in math, the approximation error is introduced to find the sparse linear combination.

Now the problem becomes finding the sparse vector which makes the significant coefficients minimum while the others are nearly zero. In this case, the atoms used for signal formation are minimized, represented as:

$$\min_{a} ||a||_0, y = \Phi_a + \eta, ||\eta||_2^2 < \epsilon \tag{1}$$

Where η denotes the approximation error and ϵ denotes the energy. The polynomial time approximation algorithms are applied to this problem finding a sub-optimal solution for the sparse vector since it's NP-hard problem[45, 33, 37, 54]. Normally, these approximation algorithms could be assigned to two groups. The first group mainly include the greedy process, like orthogonal matching pursuit, which select the local optimal vectors every iteration. The other group of algorithms are based on convex relaxation method like the east absolute shrinkage and selection operator (LASSO)[33, 54, 22, 38]. For dictionary learning, there are three widely used algorithm groups, first is the probabilistic learning, second is the clustering based or vector quantization based, and third is the particular construct learning, which normally driven by prior knowledge of the data structure or the target usage[38]. The probabilistic learning is usually established as the two-step optimization structure, the sparse approximation step and the dictionary update step. One of the commonly used probabilistic learning algorithm is the maximum likelihood dictionary learning, developed by Olshausen and Field[32]. Since their work focused on image representation, it's reasonable that the signal they processed is high dimensional and complex, making the represent and code of the image a hard problem. The maximum likelihood learning method is called sparse coding, which aims to find the evidence that the coding process in human cortex, specifically the visual area V1 could probably match a model created with sparse coding. Based on their hypothesis, the visual cortex would reduce the high dimensional signal to reduced space which defined by active neurons[32].

For cluster based methods, which keeps the core two-step optimization structure but based on vector quantization(VQ) applied through K-means clustering[32, 33, 22]. The VQ approach was first applied to video-coding[43], which optimiza a dictionary with patches of images by first allocate the pattern which has minimum distance to the given atom, then update the atom to ensure minimize the total distance. An implicit assumption, every single

patch could be made by a atom which has coefficient equal to one, was made to reduce the learning progress to a K-means clustering[43, 32]. Known that each patch was represented by a single atom, the sparse approximation becomes trivial to the whole algorithm. The K-means algorithm based dictionary learning is K-SVD algorithm, which use OMP to create the sparse approximation step and apply the singular value decomposition to make the error minimum[2]. The dictionary update step is therefore a generalized K-means algorithm because every single patch could be composed by multiple atoms accomplished with different weights. Although not globally convergence, it still has relative high performance in real application.

For the specific structure case, the dictionary is obtained from the generation function. Since these functions are parametric, they could present a short description of the atoms, making it useful in the case where a restriction of memory or complexity is needed[32]. To build such a parametric dictionary, we could use the prior knowledge of the formation of signal. For example, applying perceptual criteria could alter the selection of generation function while building the atoms. Once the goal is to rebuild the data which already perceived by censor system, the learning pattern could become learn parameters for generating function, which largely reduced the complexity[32, 55]. This dictionary design algorithm, like the others, reveals some defects. The main defect lays on the parametric class, which is not explicitly a given one. For instance, if the real data performs more likely locates in a subspace among signal space, the created optimal dictionary would result in more atoms to suit that case of the subspace. Which, however, might violate the minimum coherence constraint so it's not suitable for dictionary with a large size.

2.2 SPARSE RECOVERY

The sparse recovery is normally applied to the acquisition of compressible signals which calculate an approximate representation through a few non-adaptive linear measurement of the signal[20]. In practical, a given signal $x \in R^N$ is reconstructed with the linear measurement y = Ax, where A is a $M \times N$ matrix, which usually referred as the measurement vector of x. While the representation size M is much smaller than the signal size N, it still has many useful information of the signal x which could be applied to acquire a compression or sparsely approximation[20, 3].

The key assumption lays on that the signal x is a $K-$sparse or compressible one with a feature: rapidly decaying entry magnitude when sorted. Based on this situation, one can find the solution of this under-determined linear system through the sparse recovery theory and also recover the signal x from measurement y. To exact reconstruct the signal x, one sufficient condition is needed, the so called restricted isometry property (RIP)[10]. Practically, it means a coefficient δ_K is needed to ensure:

$$(1-\delta_K)\|\mathrm{x}\|_2^2 \leq \|Ax\|_2^2 \leq (1+\delta_K)\|\mathrm{x}\|_2^2 \tag{2}$$

works for all

$$\|x\|_0 \leq K$$

.

The measurement vector A is usually chosen as a random matrix whose entries are created as independent Gaussian distribution. If the sizes meet the relation $M \geq O(Klog(N/K))$, then the random matrix would most probably satisfy the RIP[10, 29]. Then it ensures the convex optimization

$$\arg\min_{x \in \mathcal{R}^N} \|x\|_1 \quad \text{such that} \quad y = Ax \tag{2}$$

could reconstruct the sparse signal x.

It's quite common in real applications, for example in remote sensing the process of disposing direct measurement is impossible while the indirect measurement is the solution. It's also the same case that these measurements are much fewer compared with the amount to

necessary complete the describe of object due to the sensor limitations of data processing or transmitting, costs of detecting. Based on these limitations, the problem we processed become ill-posed, which no longer has unique solution and not totally depend on data received. So when recovery the signal, the extra information denoted as prior in the statistical literature corresponding to the conditional probability is required[20].

3.0 CLUTTER IDENTIFICATION METHOD

In this section, we introduce the sparse-recovery based clutter identification method, including the impact of kernel density estimation, specifically the effects of the kernel-type and kernel-bandwidth on the estimation procedure.

3.1 BOMP METHOD

Sparse recovery algorithms aim to estimate a signal by linearly adding columns from a dictionary of predefined waveforms. Typically, representing the dictionary as a matrix $\boldsymbol{D} = \{\boldsymbol{\phi}_\omega : \omega \in \Omega\}$, whose each member ϕ_ω is called as atom and coefficient ω is obtained from a index set Ω, sparse recovery techniques solve for $\boldsymbol{\gamma}$ using $\boldsymbol{D\gamma} = \boldsymbol{s}$, where $\boldsymbol{s}$ is the original signal and $\boldsymbol{\gamma}$ is a coefficient vector. In general, the objective is to estimate the signal with m atoms, where m is much smaller than the size of dictionary N,[47, 46, 15] and hence it is referred to as a sparsity-based estimation technique.

In general, dictionaries $\boldsymbol{D}$ are designed to be fat matrices, meaning a single exact solution of $\boldsymbol{\gamma}$ does not exist. Instead, greedy approaches are used to solve for the signal as an approximation. The most popular greedy approaches fall under the category of Matching Pursuit (MP) algorithms, one of which is the orthogonal matching pursuit (OMP) algorithm that reconstructs the input signal with the least number of atoms, implying the sparsest recovery. For exact-sparse problem, the exact recovery condition (ERC) for OMP method is given as $\max_{\boldsymbol{\phi}} ||\boldsymbol{a} \in \text{span}\{\boldsymbol{\phi}_\lambda : \lambda \in \Lambda_j\}|| < 1$, where the maximum reaches over the atoms that are not be part in the optimal representation of the signal, meaning that the sparest signal reconstruction is unique.[46] Starting with the initial approximation $\boldsymbol{a}_0 = \boldsymbol{0}$

and initial residual $\boldsymbol{r}_0 = \boldsymbol{s}$, OMP method at each step tries to find an atom which correlates most perfectly with the residual;for example, at step j, the atom index λ_j is calculated by solving the optimization problem:

$$\lambda_j \in \arg\max_{\omega \in \Omega} |\langle \mathbf{r_j}, \phi_\omega \rangle| \tag{3}$$

Subsequently, the jth approximation is computed as

$$\mathbf{a_j} = \arg\min_{\mathbf{a}} \|\mathbf{s} - \mathbf{a}\|_\mathbf{2}, \quad \text{subject to} \quad \mathbf{a} \in \text{span}\{\phi_\lambda : \lambda \in \mathbf{\Lambda_j}\}, \tag{4}$$

where $\Lambda_j = \{\lambda_1, ...\lambda_j\}$ denotes the atom-indexes selected till the jth step. Because the residuals are orthogonal to the atoms which have already been chosen, OMP never chooses the same atom twice, resulting to a zero residual after d steps[23, 50]. It's clear that OMP is a greed method which at each step the atom is chosen as the most correlated to the present residual, by calculating the highest inner product. Once the atom is chosen, the signal is ensured to project to the sapn of the chosen atoms orthogonally,as 1 shows. The step 4 is the greed approach and indeed the similarity calculation step. The accuracy at this step is crucial since the atom chosen is the key to the loop, which strongly correlated to the accuracy of the whole algorithm. The later section will study the improvement in detail. And take the time complexity into consideration, step 7 is definitely the high cost step.

Algorithm 1 OMP algorithm

1: **Input**: Dictionary D, signal $\boldsymbol{y}$, target error ϵ
2: **Initialize**: Set $\boldsymbol{I} := ()$, $\boldsymbol{r} := \boldsymbol{y}$, $\boldsymbol{\gamma} := \mathbf{0}$,$n := 1$
3: **while** $r - \gamma \leq \epsilon$ **do**
4: $\quad \hat{k} := \text{argmax}_k |\boldsymbol{d}_k^T \boldsymbol{r}|$
5: $\quad I := (I, \hat{k})$
6: $\quad \boldsymbol{\gamma_I} = (\boldsymbol{D_I})^+ x$
7: $\quad \boldsymbol{r} = \boldsymbol{x} - D_{\boldsymbol{I}} \boldsymbol{\gamma_I}$
8: $\quad n = n + 1$
9: **end while**

This time cost could be reduced by implying a matric trick, the Cholesky process[14]. The Cholesky factorization is a decomposition process where a Hermitian, positive-definite matrix becomes the product of lower triangular and its conjugate transpose. It can be presented as:$A = LL^*$, where L is a lower triangular matrix with real and positive diagonal entries.In OMP1, the step 6 could be written as: $\boldsymbol{\gamma_I} = (\boldsymbol{D_I})^+ x = (\boldsymbol{D_I^T D_I})^{-1} \boldsymbol{D_I^T} x$. Since it's an orthogonal process, the inner-matrix $\boldsymbol{D_I^T D_I}$ is non-singular and symmetric positive definite. For each iteration, the inner-matrix only append a single column/row to it, so the Cholesky factorization only needs to computation the last row. For such a inner matrix $\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{L}}^T \in \mathbb{R}^{(n-1)\times(n-1)}$, the Cholesky factorization of $\boldsymbol{A} = \begin{pmatrix} \tilde{\boldsymbol{A}} & v \\ v^T & c \end{pmatrix} \in \mathbb{R}^{n\times n}$ is given as $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^T$, where $L = \begin{pmatrix} \tilde{\boldsymbol{L}} & 0 \\ w^T & \sqrt{c - w^T w} \end{pmatrix}$,$w = \tilde{\boldsymbol{L}}^{-1} v$.

Algorithm 2 Cholesky-OMP algorithm

1: **Input**: Dictionary D, signal $\boldsymbol{y}$, target error ϵ
2: **Initialize**: Set $\boldsymbol{I} := ()$, $L := [1]$, $\boldsymbol{r} := \boldsymbol{y}$, $\boldsymbol{\gamma} := \boldsymbol{0}$, $\boldsymbol{\alpha} := D^T \boldsymbol{x}, n := 1$
3: **while** $r - \gamma \leq \epsilon$ **do**
4: $\quad \hat{k} := \text{argmax}_k |\boldsymbol{d}_k^T \boldsymbol{r}|$
5: $\quad$ **if** $n > 1$ **then**
6: $\quad\quad \boldsymbol{w} :=$ Solve for $\boldsymbol{w}$ $L\boldsymbol{w} = D_{\boldsymbol{I}}^T \boldsymbol{d}_{\hat{k}}$
7: $\quad\quad L := \begin{bmatrix} L & 0 \\ \boldsymbol{w}^T & \sqrt{1 - \boldsymbol{w}^T \boldsymbol{w}} \end{bmatrix}$
8: $\quad$ **end if**
9: $\quad I := (I, \hat{k})$
10: $\quad \boldsymbol{\gamma_I} :=$ Solve for $\boldsymbol{c}$ $\{LL^T \boldsymbol{c} = \boldsymbol{\alpha_I}\}$
11: $\quad \boldsymbol{r} = \boldsymbol{x} - D_{\boldsymbol{I}} \boldsymbol{\gamma_I}$
12: $\quad n = n + 1$
13: **end while**

The OMP algorithm is updated by applying Cholesky factorization, however in real application the signal and dictionary size are huge, pre-computing process could reduce the calculation amount drastically. By further implementing the BOMP(batch-OMP), the time

complexity of the algorithm is significantly reduced[41]. As a variant of OMP algorithm, BOMP still aims to solve the reconstruction problem by finding the local minimum solution of an undetermined linear system, while reducing the computational complexity by introducing the Cholesky factorization for residual calculation.[41, 15] Denoting $\boldsymbol{\alpha} = \boldsymbol{D}^T\boldsymbol{r}$, $\boldsymbol{\alpha}^0 = \boldsymbol{D}^T\boldsymbol{s}$, $\boldsymbol{G} = \boldsymbol{D}^T\boldsymbol{D}$, and the sub-matrix $\boldsymbol{D}_\Lambda$ containing the columns indexed by Λ, we can write a new equation involving the pseudo inverse as

$$\boldsymbol{\alpha} = \boldsymbol{D}^T\left(\boldsymbol{s} - \boldsymbol{D}_\Lambda(\boldsymbol{D}_\Lambda)^+\boldsymbol{s}\right) = \boldsymbol{\alpha}^0 - \boldsymbol{G}_\Lambda\left({\boldsymbol{G}_{\Lambda,\Lambda}}^{-1}\boldsymbol{\alpha}^0_\Lambda\right) \ . \tag{5}$$

Therefore, with the pre-calculated $\boldsymbol{G}$ and $\boldsymbol{\alpha}^0$, it only needs to compute $\boldsymbol{\alpha}$ instead of $\boldsymbol{r}$ at each iteration. Also, the new multiplier $\boldsymbol{G}_{\Lambda,\Lambda}$ replaces the dictionary $\boldsymbol{D}$, where $\boldsymbol{G}_{\Lambda,\Lambda}$ denotes the progressive Cholesky factorization result.[47, 41, 15] The key point is that for the atom selection step, it's not necessary to know the r and γ exactly, but only the D^Tr. Based on the feature that the residual is not explicitly computed, the error-based stopping criterion could be applied to the algorithm. After n-th iteration, denoting the r^n as residual and γ^n as the approximation, we could get

$$r^n = x - D\gamma^n = x - D\gamma^{n-1} + D\gamma^{n-1} - D\gamma^n = r^{n-1} + D(\gamma^{n-1} - \gamma^n) \tag{6}$$

With the knowledge that the residual is orthogonal to the approximation, $r^{nT}Dr^n = 0$, the squared approximation e^n could be expressed as:

$$||r^n||^2 = ||r^{n-1}||^2 - (\gamma^n)^T G\gamma^n + \gamma^{n-1}G\gamma^{n-1} \tag{7}$$

To simplify the equation, we denote $\sigma = (\gamma^n)^T G\gamma^n$, so the approximation equation could be written as: $e^n = e^{n-1} - \sigma^n + \sigma^{n-1}$. The basic idea of BOMP is to reduce the calculation amount when recovering a great number of signals from the same measurement matrix [41]. By using BOMP, the stopping criterion C could be used to further enforce the sparsity of the output. It allows the user to specify the number of columns from dictionary applying for the description of the original signal. The algorithm could be summarized as in Algorithm 3. Note that $\boldsymbol{I}$ is an index vector, $D_{\boldsymbol{I}}$ is a matrix formed by indicated columns of D, $\boldsymbol{\alpha}_{\boldsymbol{I}}$ and $\boldsymbol{\gamma}_{\boldsymbol{I}}$ are vectors formed by indicated elements of vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, respectively, and $\boldsymbol{d}_k$ is k_{th} column of D. After this transformation, the batch-OMP algorithm becomes:

Algorithm 3 Batch-OMP algorithm

Input: Dictionary D, signal $\boldsymbol{y}$, target error e, $G = DD^T$

Initialize: Set $\boldsymbol{I} := ()$, $L := [1]$, $\boldsymbol{\gamma} := \mathbf{0}$, $\boldsymbol{\alpha} := \alpha^0 = D^T\boldsymbol{x}, n := 1$

while $e^{n-1} \geq e$ **do**

  $\hat{k} := \text{argmax}_k |\boldsymbol{d}_k^T \boldsymbol{r}|$

  if $n > 1$ **then**

    $\boldsymbol{w} :=$ Solve for $\boldsymbol{w}$ $L\boldsymbol{w} = D_{\boldsymbol{I}}^T \boldsymbol{d}_{\hat{k}}$

    $L := \begin{bmatrix} L & 0 \\ \boldsymbol{w}^T & \sqrt{1 - \boldsymbol{w}^T\boldsymbol{w}} \end{bmatrix}$

  end if

  $I := (I, \hat{k})$

  $\boldsymbol{\gamma_I} :=$ Solve for $\boldsymbol{c}$ $\{LL^T\boldsymbol{c} = \boldsymbol{\alpha_I}\}$

  $\beta = \boldsymbol{G}_I \gamma_I$

  $\alpha := \alpha^0 - \beta$

  $\sigma^n = \gamma_I^T \beta_I$

  $e = e^{n-1} - \sigma^n + \sigma^{n-1}$

  $n = n + 1$

end while

3.2 OZTURK ALGORITHM

The Ozturk algorithm provides a graphical distance measurement between the sampled data and distributions in the dictionary [35, 39, 34]. This algorithm could be applied for univariate and multivariate cases, by normalizing the ordered samples and then converting the ordered samples into points in the two-dimensional plane. The distance between the endpoint of the sampled data and that of a distribution in the dictionary presents the fitness of the sampled data with the specific distribution [35].

The Ozturk algorithm is originally designed for the fitness test. Assume that $X_1, \ldots, X_n$ are randomly sampled from a distribution function $F(x)$. Then the ordered samples are written as $X_{1:n} \leq X_{2:n} \leq \cdots \leq X_{n:n}$. Assume that a location-scale distribution $F_0((x-\mu)/\sigma)$ is the null distribution (reference distribution), where μ and σ are the location and scale parameters, respectively. Then we denote the expected order statistics from the standard null distribution as $m_{1:n}, m_{2:n}, \ldots, m_{n:n}$. The standardized i_{th} sample order statistic could be written as $Y_{i:n} = \left|X_{i:n} - \bar{X}\right| / S$, where $\bar{X}$ and S are the sample mean and standard deviation, respectively. Then the two-dimensional location of i_{th} point corresponding to the i_{th} sample order statistic could be defined by $Q_{i:n} = (U_{i:n}, V_{i:n}), i = 1, 2, \ldots, n$, where $U_{i:n} = \frac{1}{n} \sum_{j=1}^{i} \cos\left(\pi F_0(m_{j:n})\right) Y_{i:n}$, and $V_{i:n} = \frac{1}{n} \sum_{j=1}^{i} \sin\left(\pi F_0(m_{j:n})\right) Y_{i:n}$.

The graph starts from the origin in the two-dimensional system, and each point $(U_{i:n}, V_{i:n})$ is plotted to form linked vectors. These vectors could reveal a certain pattern under the null hypothesis [34]. Further, if samples are drawn from the hypothesized distribution then it should create a pattern uniformly close to the expected linked vector pattern. In this way, a $(100\,(1-\alpha))$ confidence contour for the expected endpoint $(E(U_{n:n}), E(V_{n:n}))$ can be generated, which is able to test whether samples are obtained from the hypothesized distribution [34]. This fitness test method is extended to be a distribution identification algorithm in [35] by selecting the nearest neighbour of the sample endpoint $(U_{n:n}, V_{n:n})$ from the graphical dictionary generated by the expected endpoints $(E(U_{n:n}), E(V_{n:n}))$ of various predefined distributions. For the given samples $X_1, X_2, \cdots, X_n$, the Ozturk algorithm for the distribution identification could be summarized as in Algorithm 4.

As pointed out in [35], the statistic $Q_{n:n}$ is location and scale invariant. If the expected endpoints $(E(U_{n:n}), E(V_{n:n}))$ are plotted for different distributions, then any location-scale family of distributions could be represented as a single point, while distributions having shape parameters form a curve.

Algorithm 4 Ozturk algorithm

1: Obtain the ordered sample observations $X_{i:n}$
2: Calculate the standard order statistics $Y_{i:n}$
3: Calculate the statistics $U_{n:n}$ and $V_{n:n}$ and plot the endpoint $Q_{n:n} = (U_{n:n}, V_{n:n})$.
4: Compare the sample endpoint $Q_{n:n} = (U_{n:n}, V_{n:n})$ with the expected endpoints $(E(U_{n:n}), E(V_{n:n}))$ generated by the existing distributions in the graphical dictionary, and find the nearest neighbouring distribution.

3.3 KERNEL DENSITY ESTIMATION

Kernel density estimation (KDE) is commonly used to estimate the pdf of a random variable, which could be viewed as an update of histogram, where weight function becomes the kernel function with bandwidth. For example, given a set of random samples $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ from an unknown distribution $f_X(x)$, the kernel density estimator is represented as

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) , \tag{8}$$

where $K(\cdot)$ is a kernel function determining the shape of weight function, and h is the kernel bandwidth determining the amount of smoothing applied in the estimation process.[16, 13] The kernel function could be any symmetric pdf since it meets the following properties: $\int K(t)dt = 1$ and $K(t) \geq 1$. Furthermore, given sufficient number of samples, the kernel density estimator $\hat{f}(x)$ would asymptotically converge to any density function $f_X(x)$, and therefore KDE is applicable for almost every distribution.[16]

Now, it is obvious that the choice of kernel type and bandwidth would critically affect the estimation performance. To evaluate the estimation accuracy, let us define the mean squared error (MSE) as $\mathrm{MSE}(\hat{f}(x)) = \mathrm{E}(\hat{f}(x) - f_X(x))^2 = \mathrm{bias}^2(\hat{f}(x)) + \mathrm{var}(\hat{f}(x))$. Then, by transforming and expanding with Talyor series, we get

$$\mathrm{MSE}(\hat{f}(x)) \approx \frac{1}{4} h^4 k_2^2 f''(x)^2 + \frac{1}{nh} j_2 , \tag{9}$$

where $k_2 = \int z^2 K(z)dz$ and $j_2 = \int K(z)^2 dz$. Therefore, the global estimation accuracy can be expressed in terms of the mean integrated square error (MISE) as

$$\mathrm{MISE} \approx \frac{1}{4} h^4 k_2^2 \beta(f) + \frac{1}{nh} j_2 , \quad \text{where} \quad \beta(f) = \int f''(x)^2 dx . \tag{10}$$

As MISE is a function of bandwidth h, a simple way to obtain the optimal bandwidth is to take gradient of MISE and set it to zero, which results in

$$h_{\mathrm{opt}} = \left[\frac{1}{n} \frac{\gamma(K)}{\beta(f)}\right]^{\frac{1}{5}} , \quad \text{where} \quad \gamma(K) = j_2 k_2^{-2} . \tag{11}$$

However, as h_{opt} depends on the pdf which is unknown, its practical computation is not possible.

To select the kernel bandwidth h, one commonly applied method is to assume a reference distribution, mostly the Gaussian, and then to compute the optimal bandwidth as $h_{\text{opt}} = (\frac{4\sigma_{\text{s}}^5}{3n})^{\frac{1}{5}}$, where σ_{s} is the sample standard deviation $\sigma_{\text{s}} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})}$. Based on the Gaussian assumption, a better expression of bandwidth is given as $h_{\text{opt}} = \frac{0.9\tilde{\sigma}}{n^5}$, where $\tilde{\sigma} = \min\left(\sigma_{\text{s}}, \frac{\text{IQR}}{1.34}\right)$ and IQR is the inter-quartile range, i.e., the difference between the 75th and 25th percentile points.

Another way to estimate the bandwidth is to test a set of bandwidth values, and select the one with highest accuracy. It's reasonable to assume that a smaller bandwidth would make the histogram more accurate since it keeps more points. However, numerically the small bandwidth would increase the time complexity of the algorithm, especially for the large dictionary and in some scenario the data needs to be processed online. So it's important to find a suitable bandwidth fit both the accuracy rate and the algorithm complexity. Usually the default bandwidth is created by Gaussian assumption, although different from the real distribution, the result is acceptable. A fast and efficient way is to chose the default bandwidth as base, and then create a test bandwidth set consisted with a variety percentage of the default bandwidth. Notice here the percentage set is usually less than 100%, while in real application the signal is non-stationary and the distribution parameters are unknown, the default bandwidth might be smaller for histogram. So in the numerical test, the bandwidth set includes elements larger than the default one. And by comparing the identification rate of different bandwidth, a suggested selection range could be obtained.

A more data-driven method is the 'plug-in' estimation, by using a separate smooth trick for $f''(x)$ estimation and calculating the gradient based bandwidth[17, 26, 19].

3.4 METRIC STUDY

For the clutter identification purpose, we formulate the dictionary based on a data-driven approach using the KDE paradigm as

$$\boldsymbol{D} = [\, f_1(\boldsymbol{s}) \;\; f_2(\boldsymbol{s}) \;\; \cdots \;\; f_N(\boldsymbol{s}) \,] \;\; , \tag{12}$$

where each column is created for a pre-defined clutter distribution estimated via KDE. Specifically, to create each dictionary column, S samples are used to calculate an estimated clutter pdf, $f_n(\boldsymbol{s})$, which is then normalized on a set support of W points. Thus, the final dictionary has dimension $W \times N$. In a similar manner, we create the test signal as an estimated pdf $g(\boldsymbol{s})$ by first collecting N_t target-free radar measurements and then applying KDE with the same support W which is used to build the dictionary. Once $g(\boldsymbol{s})$ has been estimated, the BOMP method is applied to select the column(s) from the dictionary $\boldsymbol{D}$ that is(are) the best match to the estimated pdf $g(\boldsymbol{s})$ of the measured clutter data[28]. Sparse recovery aims to find a solution to the following equation $D\boldsymbol{\gamma} = \boldsymbol{y}$ to obtain an estimate of $\boldsymbol{\gamma}$, where $\boldsymbol{y}$ is a vector of observations, D is a fat matrix (dictionary matrix,i.e the atom) such that it has more columns than rows, and $\boldsymbol{\gamma}$ is the unknown vector to be estimated [40]. Since D is a fat matrix, there are less number of observations in $\boldsymbol{y}$ than the unknowns in $\boldsymbol{\gamma}$. It was shown that if $\boldsymbol{\gamma}$ is sparse, then there are greedy approaches to optimize the solution of this problem. Examples of such greedy approaches include matching pursuit algorithms, and in this paper, we apply batch orthogonal matching pursuit (BOMP) to find a solution to this optimization problem [40]. Next, we describe how we formulate the clutter identification as a sparse recovery problem.

The atom index selection is made by inner product3, a commonly used measurement to identify the similarity between two distributions. Since the atom is sparse, select the index with inner product would be less precise.

To update the algorithm, the metric method for a more precise distance/similarity measurement is needed. Many geometrical distances could be applied to the BOMP algorithm. Like the inner product method, city block is a simple and high efficient method, similar

Table 1: Table of metrics

Metric Name	Function
Inner Product	$S = \sum_{i=1}^{n} P_i Q_i$
Canberra	$S = \sum_{i=1}^{n} \frac{\lvert P_i - Q_i \rvert}{P_i + Q_i}$
Intersection	$S = \sum_{i=1}^{n} \min(P_i, Q_i)$
Fidelity	$S = \sum_{i=1}^{n} \sqrt{P_i - Q_i}$

to compare two pdfs and the discrete versions of various divergences in probability and information theory fields are considered.[11] In this paper, four metrics are applied for the Euclidean distance while the effect of a large difference in a single dimension is dampened because since the distances are not squared. The four methods listed here are all typical because each method represents a group of similarity algorithm. For these commonly used algorithms, the implementation effect is unknown for this scenario, but different groups have specific features. Then the numeric test could be designed as first choose the representative algorithm of each group, comparing the identification accuracy rate. Then find the highest accuracy rate one, then do more test based on the algorithm group.

Canberra distance is a weighted Manhattan distance, similar to Srensen distance while normalizes the absolute difference of the individual level. It's suitable for sparse application since the sensitivity to small changes near zero.[11] The intersection between two pdfs is a widely used form of similarity, so as the Fidelity, the sum of geometric means.

4.0 NUMERICAL RESULTS AND DISCUSSION

This section shows the numerical examples of comparing different kernel types as well as bandwidth among specific test sample size and dictionary size applying BOMP. By randomly choosing the test parameters from dictionary and creating test samples with KDE, the change of accuracy rate in different scenarios reveals the impact of kernel types and bandwidth. The dictionary is predefined with these four distributions[52]:

1. K-distribution: $s_{\mathrm{K}} = |\sqrt{\tau} n|$, where s_{K} follows a K-distribution when $\tau \sim \mathbf{Gamma}(k, \theta)$ [k is the shape parameter and θ is the scale parameter] and $n \sim \mathcal{CN}(0, \sigma_n^2)$.
2. Weibull distribution: $s_{\mathrm{Wbl}} \sim \mathbf{Wbl}(\alpha, \beta)$, where s_{Wbl} follows a Weibull distribution with the shape parameter α and the scale parameter β.
3. Log-normal distribution: $s_{\mathrm{LN}} \sim \mathbf{LogN}(\mu_{\mathrm{LN}}, \sigma_{\mathrm{LN}}^2)$, where s_{LN} follows a log-normal distribution, implying that $\frac{(\ln y_{\mathrm{LN}} - \mu_{\mathrm{LN}})}{\sigma_{\mathrm{LN}}} \sim \mathcal{N}(0, 1)$.
4. Student-t distribution: $s_{\mathrm{St}} = \sqrt{\tau} w$, where s_{St} follows a non-standardized Student-t distribution when $1/\tau \sim \mathbf{Gamma}(v, 1/v)$ and $w \sim \mathcal{N}(0, \sigma_w^2)$.

Firstly, based on these distributions, we first construct the dictionary D with l elements, where each element is pre-learned using N i.i.d. samples from a specific clutter distribution. For example, an element of dictionary in the Ozturk algorithm based method is an expected endpoint generated by a clutter pdf. Then, considering the standard Gaussian distribution as the reference distribution, i.e., considering F_0 as a standard Gaussian cdf, the nearest neighbouring distribution in the graphical dictionary is identified as the underlying clutter distribution, as suggested by [35]. Note that the expectations of the endpoints are computed by $10,000$ Monte Carlo trials in the dictionary generation. In addition, usually the expected order statistics $m_{i:n}$ do not have closed-form expressions; therefore, we use $20,000$ Monte Carlo runs to approximate them. For the BOMP based method, an element of dictionary D

is a pre-learned discretized pdf (estimated by KDE method), which is then normalized. The underlying clutter pdf is then identified by BOMP based method with sparsity level $C = 1$.

For simplicity, we define a notation $\{l : \Delta : u\}$ as a set that collects real numbers starting from l to u with increment Δ. For instance, $\{1 : 0.5 : 3\} = \{1.0, 1.5, 2.0, 2.5, 3.0\}$.

Secondly, we comprehensively compare the BOMP based clutter identification technique with the Ozturk algorithm based method using a dictionary that includes the K, Weibull, log-normal, and Student-t distributions. In addition, we consider three dictionary sample sizes, $N = 500$, 1000, and 2500. For each of them, the test sample sizes N_{t} vary from 300 to 2800. Note that here we fix the dictionary sample size regardless of the test sample size while even applying the Ozturk method, because in many radar applications only one dictionary is preferred. To test the identification performance, we randomly select test pdfs from the dictionary, and apply the BOMP and Ozturk based methods to identify them.

Table 2: Table of kernels

Kernel Name	Function
Normal	$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}$
Triangle	$K(u) = 1 - \lvert u \rvert, \lvert u \rvert \leq 1$
Rectangle	$K(u) = \frac{1}{2u}, \lvert u \rvert \leq 1$
Epanechnikov	$K(u) = \frac{3}{4}(1 - u^2), \lvert u \rvert \leq 1$

4.1 KERNEL RESULTS

To study the impact of different kernels, we compared the accuracy of identification under different kernel types and different sample sizes with a fixed bandwidth calculated by rule-of-thumb (while the bandwidth calculated by rule-of-thumb). In practical, the sample sizes are chosen as 500,1000 and 2500. For each sample size, the accuracy test would be computed by 10,000 Monte Carlo trials. In this paper,four different types of kernel are applied as 2 And the distribution parameters are chosen as in the previous work to reveal the kernel feature, given as:

1. K-distributions with fixed $\sigma_n = 1$, fixed $\theta = 1$, and $k \in \{0.1 : 0.2 : 3.9\} \cup \{4 : 2 : 24\} \cup \{50 : 25 : 200\}$
2. Weibull distributions with fixed scale $\beta = 1$, and shape parameters $\alpha \in \{0.1 : 0.1 : 3.9\} \cup \{4 : 2 : 20\}$
3. Log-normal distributions with $\mu_{\mathrm{LN}} = 0$, and $\sigma_{\mathrm{LN}} \in \{0.05 : 0.05 : 1\} \cup \{1.1 : 0.1 : 3\}$
4. Student-t distribution with $\sigma_w = 1$, and $v \in \{0.1 : 0.2 : 4.9\} \cup \{5 : 5 : 25\} \cup \{50 : 25 : 200\}$

This part shows the average identification accuracy when using the default bandwidth calculated by rule-of-thumb, with different test dictionary sample sizes (500,1000 and 2500) and test sample sizes (from 300 to 2800). The result is supposed to show the performance of kernels in different scenarios. The results are presented in Figure 2(a) to Figure 2(d), every figure shows a specific kernel accuracy rate.

4.2 BANDWIDTH RESULTS

To study the impact of bandwidth, we compared the accuracy of identification under different bandwidth selection methods. In practical, the sample sizes are chosen as 500,1000 and 2500 and the kernel types are the same as in the previous simulation. For each sample size, the accuracy test would be computed by 10,000 Monte Carlo trials. And the bandwidth selection methods are given as:

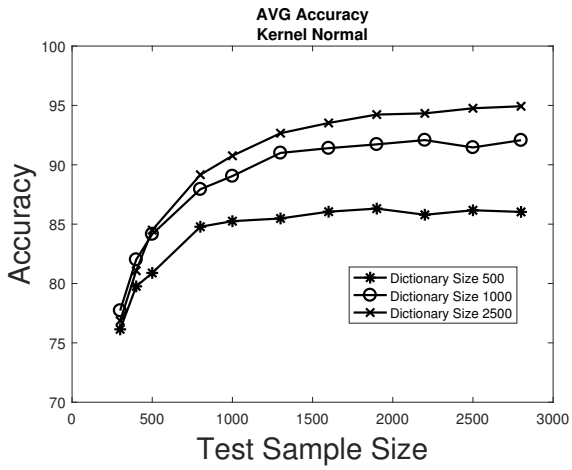
1. Rule-of-thumb : $\hat{h}_{opt} = \frac{0.9\sigma}{n^5}$.

2. Subjective bandwidth set: Chosen the rule-of-thumb bandwidth as reference, creating a set with [0.1:0.1:1.1] $\hat{h}_{opt}$.

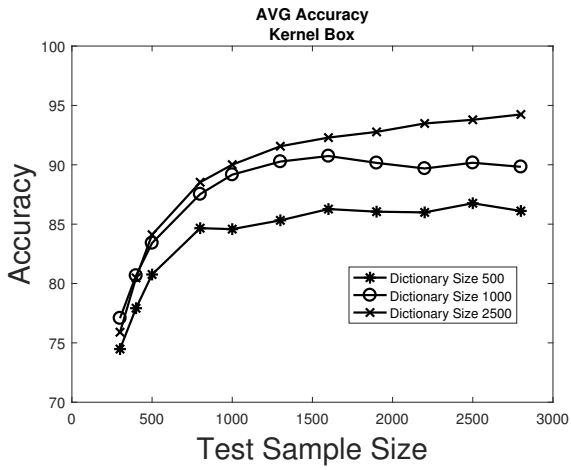
This part shows the average identification accuracy when using the set of bandwidth. The result is supposed to show the performance of bandwidth in different scenarios. The results presented in Figure 3(a) to Figure 3(d) show different accuracy rate, computed with an unique kernel function and depending only on the bandwidth set3.

Table 3: Table of bandwidth selection

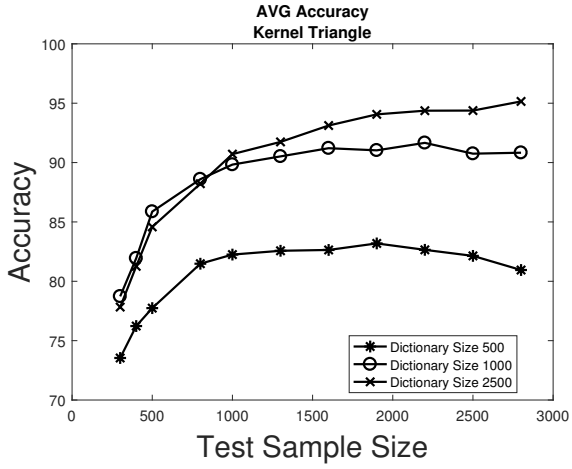
Bandwidth selection Name	Function
Maximum likelihood cross-validation	The pseudo-likelihood $\prod_{i=1}^{n} f(X_i)$ is maximized
Biased cross-validation	Estimate $R(f^{r+2})$
Complete cross-validation	Estimation of derivative of the density
Unbiased cross-validation	estimate h the minimizer of $ISE(h)$



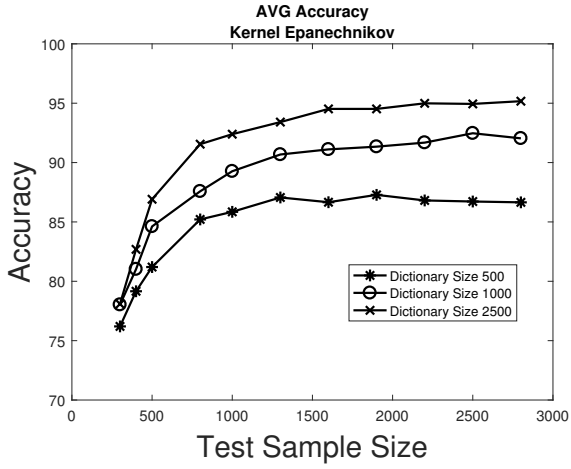
(a) Accuracy normal kernel.



(b) Accuracy box kernel.



(c) Accuracy triangle kernel.



(d) Accuracy epanechnikov kernel.

Figure 2: Effects of kernel types.

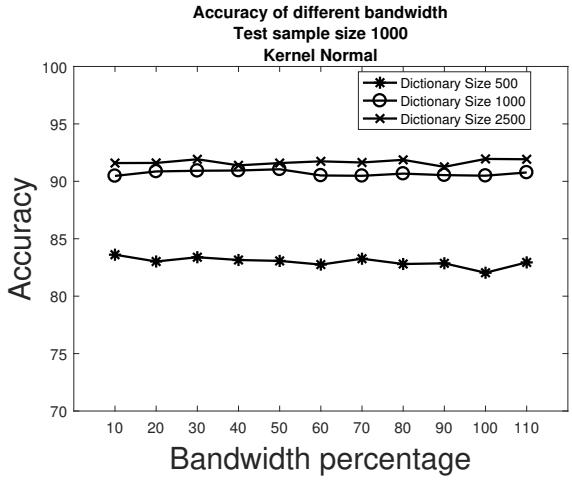
4.3 METRIC RESULTS

To study the impact of metric methods, we first implement the comparing test among four representative similarity algorithms. Since they are the most common used method in their group, the result could be viewed as an average performance. The kernel is chosen as epanechnikov, which is the best in previous numerical test and with different test dictionary sample sizes (500,1000 and 2500) and test sample sizes (from 300 to 2800). The results presented in Figure 4(a) to Figure 4(d) show different accuracy rate, computed with an unique metric.

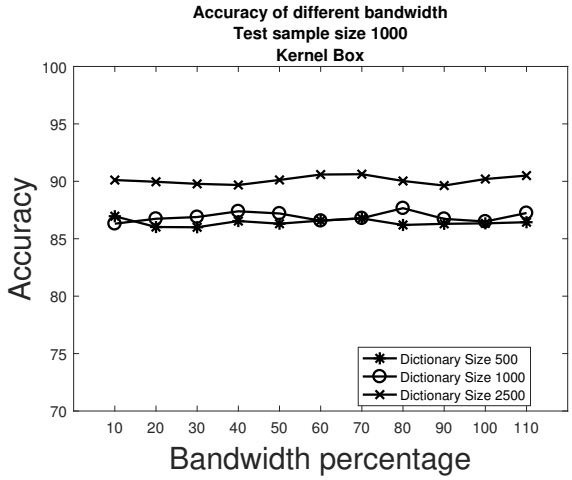
Obtained the results above, it's clear that the canberra algorithm performs the best. Based on this result, the further discussion is about the optimal metric method in this group. Thereby three other metric methods are included in the further test, they all have similar calculation part, however, still keeps features which could make a significance change in simulation. The three other metric methods are Soergel, Kulczynski and Srensen. So the following numerical test are based on the L_1 groups. The results presented in Figure 5(a) to Figure 5(d) show different accuracy rate, computed with an unique metric function. The metric families shown below:

Table 4: Table of canberra metric

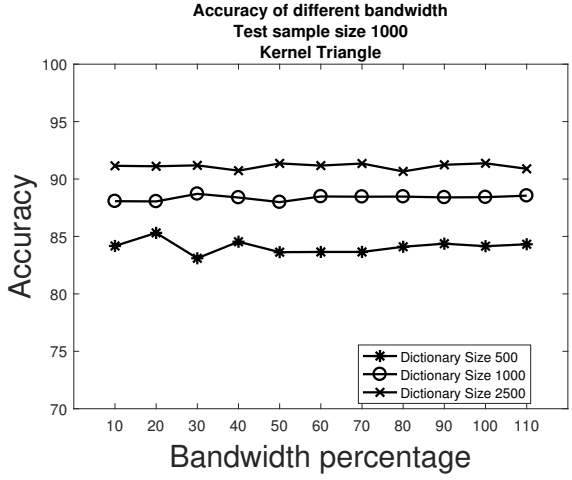
Metric name	Function
Canberra	$S = \frac{\sum_{i=1}^{n} \lvert P_i - Q_i \rvert}{\sum_{i=1}^{n} \max P_i, Q_i}$
Soergel	Estimate $R(f^{r+2})$
Kulczynski	$S = \frac{\sum_{i=1}^{n} \lvert P_i - Q_i \rvert}{\min P_i, Q_i \sum_{i=1}^{n}}$
Srensen	$S = \frac{\sum_{i=1}^{n} \lvert P_i - Q_i \rvert}{\sum_{i=1}^{n} P_i + Q_i}$



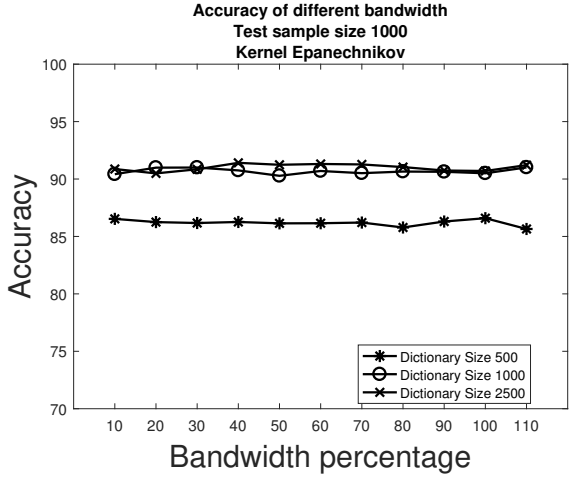
(a) Accuracy normal kernel.



(b) Accuracy box kernel.

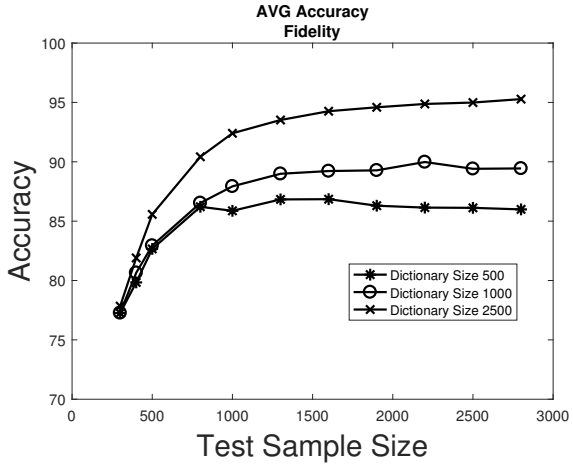


(c) Accuracy triangle kernel.

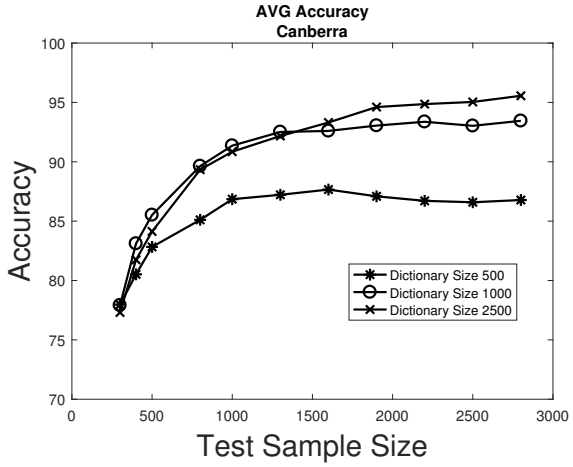


(d) Accuracy epanechnikov kernel.

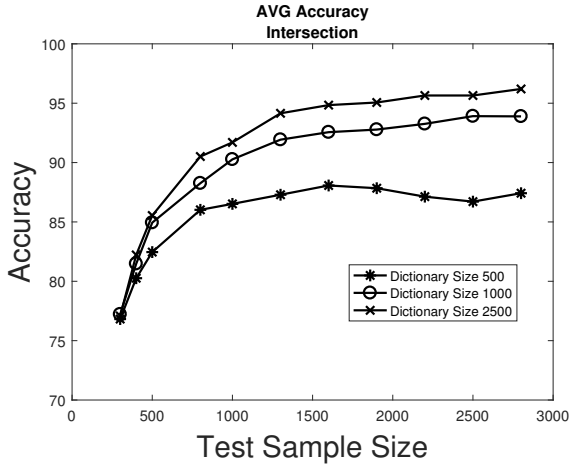
Figure 3: Effects of bandwidth.



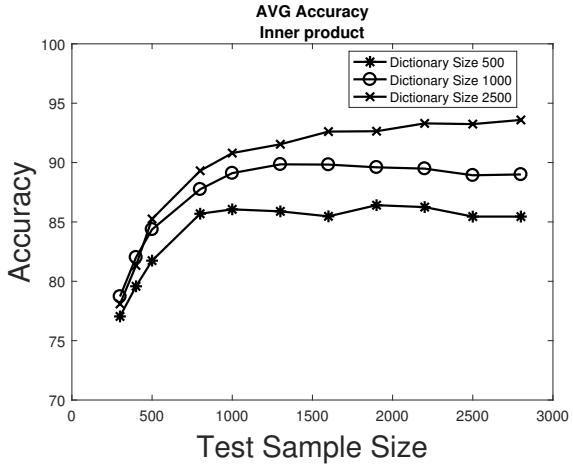
(a) Fidelity accuracy.



(b) Canberra accuracy.

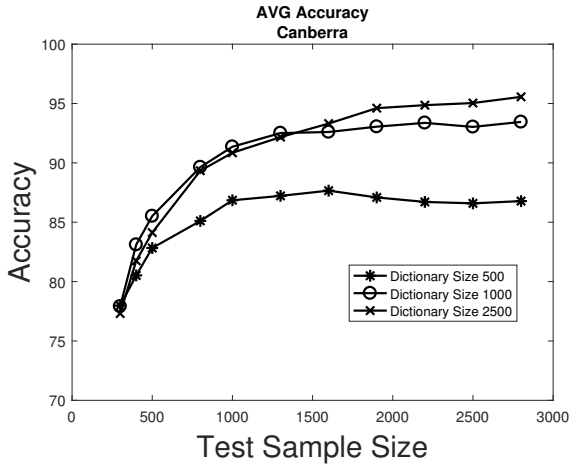


(c) Intersection accuracy.

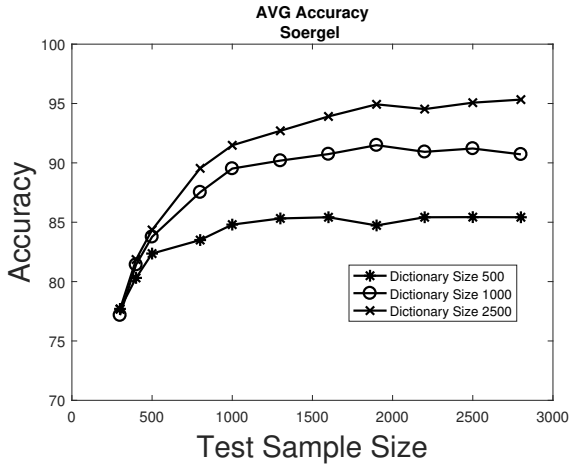


(d) Inner product accuracy.

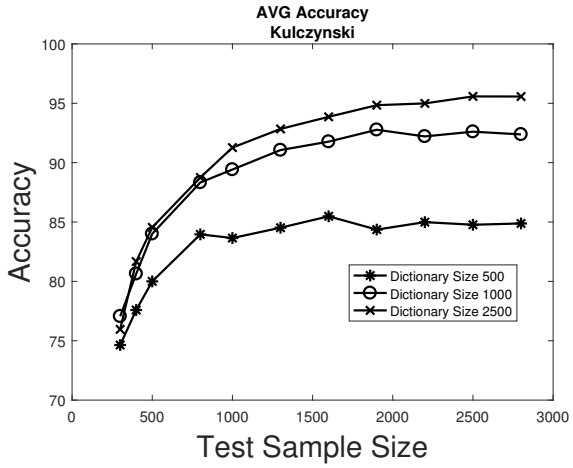
Figure 4: Four typical metrics accuracy.



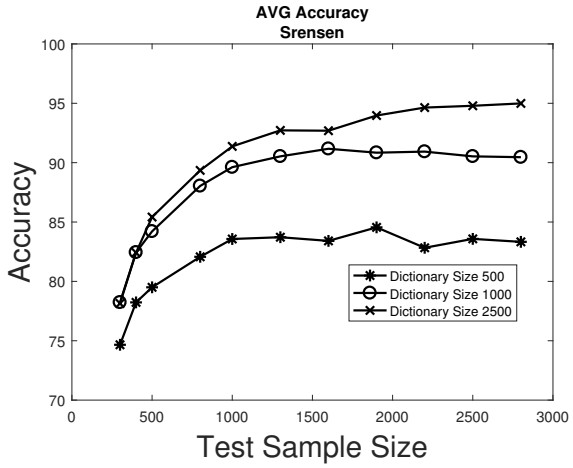
(a) Canberra accuracy.



(b) Soergel accuracy.



(c) Kulczynski accuracy.



(d) Srensen accuracy.

Figure 5: L1 group accuracy.

4.4 COMPARISON OF OZTURK ALGORITHM AND BOMP

The Graphical dictionaries(K,Weibull,Log-normal,Student's-t distributions) generated by Ozturk method using different sample sizes(500,1000,2500) is presented below. The graphical dictionary changes when the sample sizes are varying, especially for the Weibull and log-normal distributions. More precisely, some distributions in the dictionary are relative stable with respect to the sample size, while the other distributions are not. Now, in the Ozturk algorithm based clutter identification method, if we plan to identify a test distribution, in principle the test sample size N_t needs to match the dictionary sample size N, unless the graphical dictionary is not sensitive to dictionary sample size. In 4.4, as an empirical observation result, those endpoint-sensitive cases generally correspond to the distributions with relative large variances. However, to the best of our knowledge, there is no theoretical analysis which thoroughly discussed the sensitivity of the dictionary of the Ozturk method to the sample size. In modern cognitive radar framework, it would be an advantage that radar can adaptively control/change the number of (test) samples acquired from the environment and the target. In such applications, radar systems might need to store various graphical dictionaries with different N values, rather than one dictionary, to adapt to different test sample sizes while using the Ozturk method.

And the Comparison of BOMP and Ozturk methods for clutter identification is shown below. we comprehensively compare the BOMP based clutter identification technique with the Ozturk algorithm based method using a dictionary that includes the K, Weibull, log-normal, and Student-t distributions. Specifically, we consider

1. K-distributions with fixed $\sigma_n = 1$, $\theta \in \{1, 10\}$, and $k \in \{0.1 : 0.5 : 3.6\} \cup \{4 : 5 : 24\}$,
2. Weibull distributions with shape parameters $\alpha \in \{0.5 : 0.5 : 3\} \cup \{4 : 4 : 20\}$, and scale parameters $\beta \in \{1, 2\}$,
3. Log-normal distributions with $\mu_{\text{LN}} = 0$, and $\sigma_{\text{LN}} \in \{0.05 : 0.2 : 0.85\} \cup \{1 : 0.5 : 2\}$,
4. Student-t distributions with $\sigma_w = 1$, and $v \in \{0.5 : 0.5 : 1.5\} \cup \{2 : 3 : 8\}$.

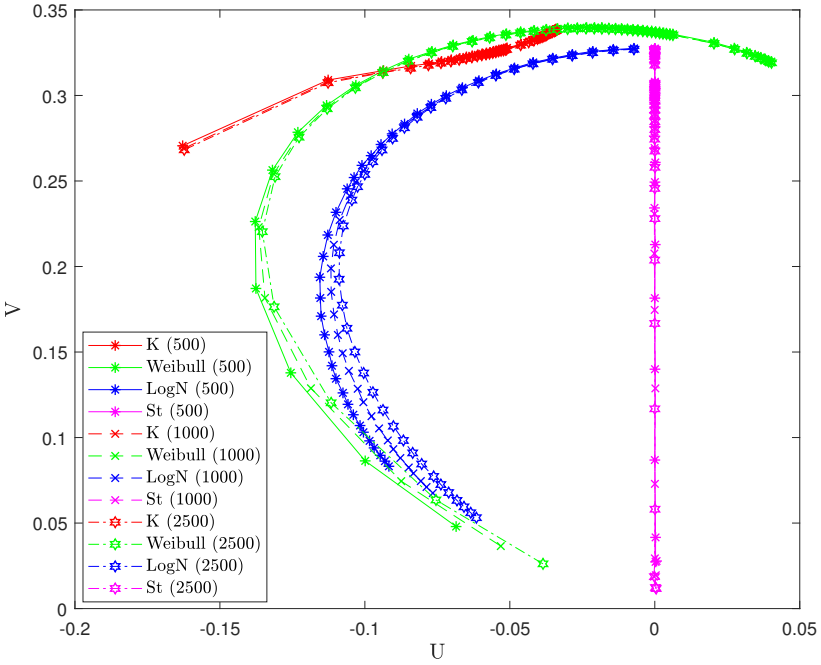


Figure 6: Ozturk dictionary

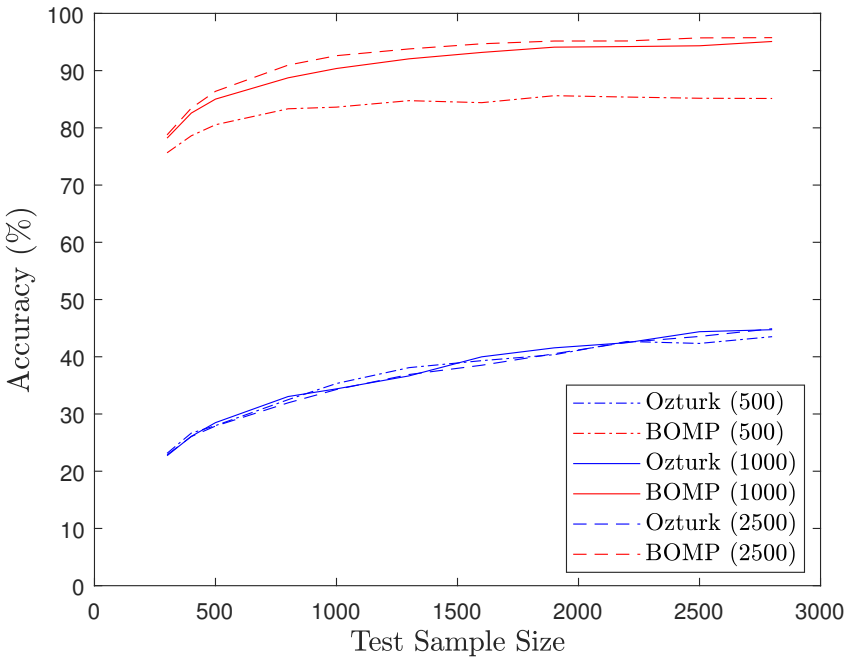


Figure 7: Bomp and Ozturk comparison

In addition, we consider three dictionary sample sizes, $N = 500$, 1000, and 2500. For each of them, the test sample sizes N_t vary from 300 to 2800. Note that here we fix the dictionary sample size regardless of the test sample size while even applying the Ozturk method, because in many radar applications only one dictionary is preferred. To test the identification performance, we randomly select test pdfs from the dictionary, and apply the BOMP and Ozturk based methods to identify them. The resulting performance comparison is shown in Fig. 4.4. In general, we notice from Fig. 4.4 that the proposed BOMP based technique for clutter identification significantly outperforms the Ozturk algorithm based method. It is due to the fact that the deficiencies of the Ozturk algorithm, such as the location-scale invariant property, hinder its identification performance. In addition, we observe that both methods have improved performances with more test samples. Also, for BOMP based identification technique, the larger the dictionary sample size, the better is the overall identification accuracy. However, for the Ozturk algorithm based method, increasing the sample size in the dictionary generation does not substantially change the overall identification performance. Compared to the simulation for Fig. 4.4, we do not include Log-normal distributions of large variances in the dictionary in the simulation for Fig. 4.4, and thus the performance of Ozturk algorithm seems robust. In the second group of figures, Figure 2(a) to Figure 2(d) we notice that the average clutter identification accuracy is higher when dictionary size is larger. With larger dictionary size, each column in the dictionary has more support points, which helps solving the BOMP optimization problem more accurately. Also, the test sample size is roughly correlated with the accuracy, while a fall back is detected when test sample are larger than 1500. This situation is obviously shown in low dictionary size, for which the normal kernel and Epanechnikov kernel are more robust than the others. When applying larger dictionary size, all kernels perform well and with the increase of the test sample size, the accuracy rate roughly rises from 78% to 95% at same pace. When the dictionary has a size of 1000, the accuracy of box kernel is inferior to the others. When dictionary size is doubled from 500 to 1000, the accuracy rate of kernel box increases much slower than the others. Comparing Figure 2(a) and Figure 2(d), the kernel normal seems to be more robust to discrepancies between the dictionary and test sample sizes than the Epanechnikov kernel for large dictionary sizes.

In the second group of figures, Figure 3(a) to Figure 3(d), the accuracy rate of clutter distribution identification changes slightly when bandwidth varies from 10% to 110% of the bandwidth computed through the rule-of-thumb method. This implies that the BOMP method is robust for the application scenarios with different bandwidth. We do not observe a direct correlation between the changes in the bandwidth and accuracy of clutter distribution identification. The rule-of-thumb method, even though it is established for calculation of bandwidth for normal kernels, performs well for other kernels as well. Therefore, we suggest using the rule-of-thumb method for the calculation of the kernel bandwidths. Among the four chosen kernels, the triangle kernel still performs inferior to the others. The normal and Epanechnikov kernels are preferable, as both of them are robust in terms of bandwidth change. On the other hand, the box kernel requires a large dictionary size to achieve similar clutter distribution identification accuracies. Also, we note that the Epanechnikov kernel shows good results with small dictionary size, which is significant for scenarios in which not enough data are available to build the dictionary, especially when updating dictionary online. In such cases, we suggest to apply the Epanechnikov kernel function.

In the third group of figures, Figure 4(a) to Figure 4(d), the four typical metric methods are applied. Comparing with the accuracy rate, the atom selection algorithm using canberra and intersection obviously have higher performance than others. Also, taken the real application into consideration, the canberra metric behaves well in smaller dictionary size, which makes it suitable for the online processing. Since canberra is a typical metric of norm group, it's much helpful to apply other metric measurements of the same group to find an optimal choice. Then Figure 5(a) to Figure 5(d) are introduced, where each graph represents a common metric measurement in norm group. It can be viewed that all their performance is better than the other typical metrics previously, and the canberra method still performs the best among them.

5.0 CONCLUSIONS AND FUTURE WORK

In this work, we presented a sparse-recovery based clutter identification method and analyzed its performance with respect to two different kernel bandwidth selection methods and for different kernel functions. Based on the numerical examples, we demonstrated that the sparse-recovery based technique provided (i) robustness in terms of different kernel types and bandwidths; and (ii) high accuracy in identifying clutter measurements originating from different families of distributions. Our results further demonstrated that the Epanechnikov kernel with canberra metrics performs the best compared to the other kernels. We observed that compared to the sparse recovery based method, Ozturk algorithm does not have sufficient accuracy in identifying the distributions originating from the same family but with different parameters, especially it suffers from identifying the scale parameters correctly due to its location-invariant property and this in result decreases its overall efficiency/accuracy in identifying clutter distributions.

In our future work, we plan to adaptively increase and decrease dictionary size of the sparse-recovery based method; such adaptive change in dictionary size will be crucial in order to characterize measured data that may not be well-represented by any specific distribution in the dictionary and to control the computational load. Furthermore, with real measured data, we will incorporate the sparse-recovery based clutter identification method into the design of a fully cognitive radar system, which will include the statistical tests for estimating change points in the clutter distribution, methods for identifying the new clutter distribution and adaptation techniques for detection/tracking algorithms to the newly learned clutter distribution.

BIBLIOGRAPHY

[1] M. Aharon, M. Elad, and A. Bruckstein. *rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, Nov 2006. ISSN 1053-587X.

[2] M. Aharon, M. Elad, and A. Bruckstein. *rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, Nov 2006. ISSN 1053-587X.

[3] A. Akbari, M. Trocan, and B. Granado. Sparse recovery-based error concealment. *IEEE Transactions on Multimedia*, 19(6):1339–1350, June 2017. ISSN 1520-9210.

[4] M. Akcakaya and A. Nehorai. Adaptive MIMO radar design and detection in compound-Gaussian clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 47(3):2200–2207, 2011.

[5] M. Akcakaya, S. Sen, and A. Nehorai. A novel data-driven learning method for radar target detection in nonstationary environments. *IEEE Signal Processing Letters*, 23(5): 762–766, May 2016. ISSN 1070-9908.

[6] A. Aubry, A. D. Maio, B. Jiang, and S. Zhang. Ambiguity function shaping for cognitive radar via complex quartic optimization. *IEEE Transactions on Signal Processing*, 61 (22):5603–5619, Nov 2013. ISSN 1053-587X.

[7] A. Balleri, A. Nehorai, and J. Wang. Maximum likelihood estimation for compound-Gaussian clutter with inverse gamma texture. *IEEE Transactions on Aerospace and Electronic Systems*, 43(2):775–779, April 2007. ISSN 0018-9251.

[8] K. L. Bell, C. J. Baker, G. E. Smith, J. T. Johnson, and M. Rangaswamy. Cognitive radar framework for target detection and tracking. *IEEE Journal of Selected Topics in Signal Processing*, 9(8):1427–1439, Dec. 2015. ISSN 1932-4553.

[9] S. D. Blunt and E. L. Mokole. Overview of radar waveform diversity. *IEEE Aerospace and Electronic Systems Magazine*, 31(11):2–42, November 2016. ISSN 0885-8985.

[10] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, Dec 2005. ISSN 0018-9448.

[11] S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. 2007.

[12] H. C. Chan. Radar sea-clutter at low grazing angles. *IEE Proceedings F - Radar and Signal Processing*, 137(2):102–112, Apr 1990. ISSN 0956-375X.

[13] D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):281–288, Feb 2003. ISSN 0162-8828. doi: 10.1109/TPAMI.2003.1177159.

[14] S. F. Cotter, R. Adler, R. D. Rao, and K. Kreutz-Delgado. Forward sequential algorithms for best basis selection. *IEE Proceedings - Vision, Image and Signal Processing*, 146(5):235–244, Oct 1999. ISSN 1350-245X.

[15] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constructive Approximation*, 13(1):57–98, Mar 1997. ISSN 1432-0940.

[16] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proc. of the IEEE*, 90(7):1151–1163, Jul 2002. ISSN 0018-9219. doi: 10.1109/JPROC.2002.801448.

[17] A. Elgammal, R. Duraiswami, and L. S. Davis. Efficient kernel density estimation using the fast Gauss transform with applications to color modeling and tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1499–1504, Nov 2003. ISSN 0162-8828. doi: 10.1109/TPAMI.2003.1240123.

[18] K. Engan, S. O. Aase, and J. H. Husoy. Method of optimal directions for frame design. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258)*, volume 5, pages 2443–2446 vol.5, 1999.

[19] J. Fan, N. E. Heckman, and M. P. Wand. Local polynomial kernel regression for generalized linear models and quasi-likelihood functions. *Journal of the American Statistical Association*, 90(429):141–150, 1995. ISSN 01621459.

[20] A. Gilbert and P. Indyk. Sparse recovery using sparse matrices. *Proceedings of the IEEE*, 98(6):937–947, June 2010. ISSN 0018-9219.

[21] F. Gini, A. Farina, and F. Lombardini. Effects of foliage on the formation of K-distributed SAR imagery. *Signal Processing*, 75(2):161–171, 1999.

[22] R. Gribonval and M. Nielsen. Sparse representations in unions of bases. *IEEE Transactions on Information Theory*, 49(12):3320–3325, Dec 2003. ISSN 0018-9448.

[23] J. Haupt, R. Castro, R. Nowak, G. Fudge, and A. Yeh. Compressive sampling for signal classification. In *Proc. 40th Asilomar Conference on Signals, Systems and Computers*, pages 1430–1434, Oct 2006. doi: 10.1109/ACSSC.2006.354994.

[24] S. Haykin. Cognitive radar: A way of the future. *IEEE Signal Processing Magazine*, 23 (1):30–40, Jan. 2006.

[25] S. Haykin, Y. Xue, and P. Setoodeh. Cognitive radar: Step toward bridging the gap between neuroscience and engineering. *Proceedings of the IEEE*, 100(11):3102–3130, Nov 2012. ISSN 0018-9219.

[26] M. C. Jones, J. S. Marron, and S. J. Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91(433):401–407, 1996. ISSN 01621459.

[27] S. Kay. *Fundamentals of Statistical Signal Processing: Detection theory*. Prentice-Hall PTR, 1998. ISBN 9780135041352.

[28] M. Kelsey, S. Sen, Y. Xiang, A. Nehorai, and M. Akcakaya. Sparse recovery for clutter identification in radar measurements. In *Proc. SPIE*, volume 1021106, pages 1–10, May 2017.

[29] J. N. Laska, P. T. Boufounos, M. A. Davenport, and R. G. Baraniuk. Democracy in action: Quantization, saturation, and compressive sensing. *Applied and Computational Harmonic Analysis*, 31(3):429 – 443, 2011. ISSN 1063-5203.

[30] L. J. Marier. Correlated K-distributed clutter generation for radar detection and track. *IEEE Transactions on Aerospace and Electronic Systems*, 31(2):568–580, April 1995. ISSN 0018-9251.

[31] E. K. Miller and J. D. Cohen. An integrative theory of prefrontal cortex function. *Annual Review of Neuroscience*, 24:167–202, 2001. Copyright - Copyright Annual Reviews, Inc. 2001; Last updated - 2014-05-18.

[32] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311 – 3325, 1997. ISSN 0042-6989.

[33] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37(23):3311 – 3325, 1997. ISSN 0042-6989.

[34] A. Ozturk. A general algorithm for univariate and multivariate goodness of fit tests based on graphical representation. *Communications in Statistics - Theory and Methods*, 20(10):3111–3137, 1991.

[35] A. Ozturk. An application of a distribution identification algorithm to signal detection problems. In *Proc. 27th Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 248–252, 1993.

[36] R. Palama, M. S. Greco, P. Stinco, and F. Gini. Statistical analysis of bistatic and monostatic sea clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 51(4): 3036–3054, Oct. 2015. ISSN 0018-9251.

[37] M. D. Plumbley. Dictionary learning for l1-exact sparse coding. In M. E. Davies, C. J. James, S. A. Abdallah, and M. D. Plumbley, editors, *Independent Component Analysis and Signal Separation*, pages 406–413, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-74494-8.

[38] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies. Sparse representations in audio and music: From coding to source separation. *Proceedings of the IEEE*, 98(6):995–1005, June 2010. ISSN 0018-9219.

[39] M. Rangaswamy, D. D. Weiner, and A. Ozturk. Non-Gaussian random vector identification using spherically invariant random processes. *IEEE Transactions on Aerospace and Electronic Systems*, 29(1):111–124, Jan. 1993. ISSN 0018-9251.

[40] G. Rath and A. Sahoo. A comparative study of some greedy pursuit algorithms for sparse approximation. In *Proc. 17th European Signal Processing Conference*, pages 398–402, Aug 2009.

[41] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. Technical report, Computer Science Department, Technion Israel Institute of Technology, Aug. 2008.

[42] P. F. Sammartino, C. J. Baker, and H. D. Griffiths. Adaptive MIMO radar system in clutter. In *Proc. IEEE Radar Conference*, pages 276–281, April 2007.

[43] P. Schmid-Saugeon and A. Zakhor. Dictionary design for matching pursuit and application to motion-compensated video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(6):880–886, June 2004. ISSN 1051-8215.

[44] M. Sekine, T. Musha, Y. Tomita, T. Hagisawa, T. Irabu, and E. Kiuchi. Weibull-distributed sea clutter. *IEE Proceedings F - Communications, Radar and Signal Processing*, 130(5):476–, August 1983. ISSN 0143-7070. doi: 10.1049/ip-f-1.1983.0076.

[45] I. Tosic and P. Frossard. Dictionary learning. *IEEE Signal Processing Magazine*, 28(2): 27–38, March 2011. ISSN 1053-5888.

[46] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, Oct 2004. ISSN 0018-9448. doi: 10.1109/TIT.2004.834793.

[47] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, Dec 2007. ISSN 0018-9448. doi: 10.1109/TIT.2007.909108.

[48] J. A. Tropp and S. J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, June 2010. ISSN 0018-9219.

[49] H. L. Van Trees. *Detection, Estimation, and Modulation Theory*. John Wiley & Sons, 2004.

[50] H. Wang, J. Vieira, P. Ferreira, B. Jesus, and I. Duarte. Batch algorithms of matching pursuit and orthogonal matching pursuit with applications to compressed sensing. In *Proc. International Conference on Information and Automation*, pages 824–829, June 2009. doi: 10.1109/ICINFA.2009.5205034.

[51] J. Wang, A. Dogandzic, and A. Nehorai. Maximum likelihood estimation of compound-Gaussian clutter and target parameters. *IEEE Transactions on Signal Processing*, 54 (10):3884–3898, Oct 2006. ISSN 1053-587X.

[52] K. D. Ward, S. Watts, and R. J. Tough. *Sea Clutter: Scattering, the K Distribution and Radar Performance*. IET, 2006.

[53] Y. Xiang, M. Kelsey, H. Wang, S. Sen, M. Akcakaya, and A. Nehorai. A comparison of cognitive approaches for clutter-distribution identification in nonstationary environments. In *IEEE Radar Conference*, 2018.

[54] M. Yaghoobi, L. Daudet, and M. E. Davies. Parametric dictionary design for sparse coding. *IEEE Transactions on Signal Processing*, 57(12):4800–4810, Dec 2009. ISSN 1053-587X.

[55] M. Yaghoobi, L. Daudet, and M. E. Davies. Parametric dictionary design for sparse coding. *IEEE Transactions on Signal Processing*, 57(12):4800–4810, Dec 2009. ISSN 1053-587X.