CS-AdaBoost: One-Bit Compressed Sensing for Direction-of-Arrival Estimation

Majdoddin Esfandiari*, Petteri Pulkkinen*†, Sergiy A. Vorobyov*, and Visa Koivunen*
*Department of Information and Communications Engineering, Aalto University, Espoo, Finland
†Saab Finland Oy, Helsinki, Finland
Email: majdoddin.esfandiari@aalto.fi, petteri.pulkkinen@aalto.fi, sergiy.vorobyov@aalto.fi, visa.koivunen@aalto.fi

***Abstract*—Using One-bit analog-to-digital converters (ADCs) instead of high-precision counterparts for direction-of-arrival (DOA) estimation is a promising alternative to considerably reduce power consumption and manufacturing transceiver costs. However, these benefits come at the cost of information loss, as one-bit ADCs retain only the sign of the signals and discard the amplitude information. Consequently, developing customized one-bit DOA estimation methods is necessary. In this work, we propose a learning-based one-bit DOA estimator referred to as compressed sensing adaptive boosting (CS-AdaBoost). The proposed method employs a two-stage weak classifier within AdaBoost framework. It first discretizes the DOA angular interval and builds an over-complete dictionary for the array steering matrix and then estimates the corresponding source signal matrix in an iterative manner. In each AdaBoost iteration, an approximate weighted least ℓ_2-norm estimation is used as the first stage, followed by hard-thresholding for imposing sparsity at the second stage. Numerical simulations demonstrate the superiority of the CS-AdaBoost method over other existing methods, especially in the face of closely spaced and correlated sources.**

***Index Terms*—One-bit ADC, DOA estimation, AdaBoost, least norm estimation, compressed sensing**

I. Introduction

One-bit direction-of-arrival (DOA) estimation has gained considerable attention in the context of array signal processing due to substantial reduction in power consumption and manufacturing costs. The use of one-bit analog-to-digital converters (ADCs) provides substantial savings especially in systems using large-scale arrays [1], [2]. In [4], the authors applied the arcsine law to reconstruct the covariance matrix, which allowed the use of conventional beamforming technique for one-bit DOA estimation. The authors of [8] proposed the one-bit multiple signal classification (MUSIC) DOA estimator, which first uses the arcsine law to restore the array covariance matrix and then identifies the noise subspace. The DOAs are estimated by selecting the highest peaks in the well-known MUSIC pseudo-spectrum. In [9], MUSIC was also applied to reconstructed array measurement matrix. A maximum-likelihood (ML) one-bit DOA estimation algorithm was developed in [10].

Imposing signal sparsity has been considered in many approaches, such as joint sparse representation [11], deep fixed-point continuation (DeepFPC) [12], and complex-valued binary iterative hard threshold (CBIHT) [13], [14]. One-bit extension of the generalized sparse Bayesian learning (Gr-SBL) approach [16] was developed in [15]. Moreover, a one-bit compressed sensing (CS) method named robust one-bit CS (ROCS) was proposed in [17] and applied to the one-bit DOA estimation problem. One-bit DOA estimation for sparse linear arrays (SLAs) has been explored in numerous works such as [3], [6], [18]–[20].

Gridless one-bit DOA estimators have been developed in the literature to overcome the challenges such as grid mismatch and difficulty in resolving closely spaced sources that some of the grid-based methods face. For example, gridless compressed sensing [21], atomic norm denoising (AND) [22], and accelerated proximal gradient (APG) [23] are gridless methods that rely on promoting the Toeplitz structure of the array covariance matrix in one-bit DOA estimation. The authors in [24] proposed an off-grid iterative reweighted approach (OGIR) which jointly estimates DOAs as well as row sparse source signal matrix. The OGIR method iteratively solves a maximum a posteriori (MAP) optimization problem using the block successive upper-bound minimization (BSUM) [25].

Formulating one-bit parameter estimation as the problem of finding the separating hyperplane in a binary classification problem has been promoted in several works [26], [29]. In this context, a support vector machine (SVM)-based one-bit DOA estimation technique was introduced in [26]. This iterative method is initialized by the discrete Fourier transform (DFT) matrix as the grid-based dictionary. Each iteration consists of two steps: first, estimating source signals using SVM, followed by refining the DOA estimates in a gridless manner.

Despite significant research efforts, accurate one-bit DOA estimation in cases with closely spaced and/or highly correlated sources remains a challenge. Therefore, developing more reliable one-bit DOA estimators is crucial.

In this paper, we propose a CS-based one-bit DOA estimator, where the over-complete array steering matrix is first constructed. A two-stage weak classifier with the capability of enforcing sparsity is then utilized within adaptive boosting (AdaBoost) framework [27], [28]. In the first stage of each iteration, an approximate weighted least ℓ_2-norm estimation of the separating hyperplane is employed to estimate the source signal matrix, while a hard-thresholding operator is applied to the estimated source signal matrix to impose sparsity in the second stage of each iteration. After a predefined number of AdaBoost iterations, the DOAs are determined as grid angles corresponding to the largest peaks of the spatial spectrum obtained by computing the ℓ_2-norm of each row of the final source signal matrix.

Notation: Matrices and vectors are represented by bold uppercase and lowercase letters, respectively, while scalars are represented by lowercase letters. The $n \times n$ identity matrix is represented by $\mathbf{I}_n$. The operator $\mathrm{vec}\{\cdot\}$ stacks the columns of a matrix into a column vector, while the operator $\mathrm{unvec}\{\cdot\}$ generates a matrix with the corresponding dimension from the entries of the bracketed vector. The operators $\Re\{\cdot\}$ and $\Im\{\cdot\}$ return respectively the real and imaginary parts of the bracketed argument. The function $\mathbf{1}\{\cdot\}$ is the indicator function. The Hadamard and Kronecker products are denoted by $\odot$ and $\otimes$, respectively.

II. System Model and Preliminaries

A. System Model for One-Bit DOA Estimation

Let a uniform linear array (ULA) with M antenna elements receive narrow-band far-field signals from K sources with directions of $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_K]^T$. Each antenna element deploys two one-bit ADCs for converting the real and imaginary parts of received signals. The received signals over N time instants are then expressed as

$$\mathbf{Y} = \mathcal{Q}\left(\mathbf{A}(\boldsymbol{\theta})\mathbf{S} + \mathbf{Z}\right) \tag{1}$$

where $\mathcal{Q}(\cdot) \triangleq \mathrm{sign}(\Re\{\cdot\}) + j\mathrm{sign}(\Im\{\cdot\})$ represents the element-wise one-bit quantizer, $\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \ldots, \mathbf{a}(\theta_K)] \in \mathbb{C}^{M\times K}$ and $\mathbf{a}(\theta_k) = [1, e^{-j2\pi d \sin(\theta_k)/\lambda}, \ldots, e^{-j2\pi(M-1)d\sin(\theta_k)/\lambda}]^T \in \mathbb{C}^M$ are the steering matrix of $\boldsymbol{\theta}$ and steering vector of θ_k, respectively. In addition, $d = \lambda/2$ is the inter-element spacing, λ is the received signals' carrier wavelength, and $\mathbf{S} \in \mathbb{C}^{K\times N}$ is the incident source signals. Noise $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N] \in \mathbb{C}^{M\times N}$ is assumed to be complex Gaussian obeying $\mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I}_M)$. The goal is to estimate K DOAs $\theta_1, \ldots, \theta_K$ using $\mathbf{Y}$.

B. AdaBoost Framework

In binary supervised learning, a training set $\mathcal{D}$ is formed by m training examples $\{\mathbf{x}_j \in \mathbb{R}^n\}_{j=1,\ldots,m}$ and m binary class labels $y_j \in \{1, -1\}_{j=1,\ldots,m}$. When the training set $\mathcal{D}$ is linearly separable, the hyperplane defined by $f(x) = \mathbf{h}^T\mathbf{x} + b = 0$ splits the data space into two disjoint regions. Here, $\mathbf{h}$ and b are the weight vector and bias, respectively. The relation of $\mathbf{x}_j$ and y_j for $j = 1, \ldots, m$ can be expressed as $y_j = g(\mathbf{x}_j) = \mathrm{sign}(\mathbf{h}^T\mathbf{x}_j + b)$, where $g(\mathbf{x}_j)$ serves as the binary classifier. The similarity between the definition of $g(x)$ and the real-valued transformation of (1) can be used for estimating $\mathbf{S}$ via binary classification approaches.

AdaBoost is an effective classification method that works in an iterative manner by combining multiple *weak classifiers* using a *forward stage-wise additive modeling* to create a *strong classifier* [27], [28]. Here, a weak classifier refers to a classifier that performs slightly better than random guessing. At the beginning, AdaBoost assigns identical weights to all training examples and trains the first weak classifier. After each iteration, it increases the weights of the misclassified examples, ensuring that the next weak classifier is trained on the updated weighted training examples. In the r-th iteration, the r-th hyperplane weight vector (i.e., the r-th weak classifier) $\mathbf{h}^{(r)}$ is trained using the training examples and the AdaBoost weight vector $\boldsymbol{\beta}^{(r)} = [\beta_1^{(r)}, \beta_2^{(r)}, \ldots, \beta_m^{(r)}]^T$. Then, the weighted classification error is computed as $\epsilon^{(r)} = \sum_{j=1}^m \beta_j^{(r)} \mathbf{1}\{\mathrm{sign}\left((\mathbf{h}^{(r)})^T\mathbf{x}_j\right) \neq y_j\}$. The weight of the r-th weak classifier in producing the final strong classifier is defined as $\alpha^{(r)} = \frac{1}{2}\ln\left(\frac{1-\epsilon^{(r)}}{\epsilon^{(r)}}\right)$. Next, the AdaBoost weights corresponding to misclassified training examples are increased for the next iteration as $\boldsymbol{\beta}^{(r+1)} = \beta_j^{(r)} \exp(\alpha^{(r)}\mathbf{1}\{\mathrm{sign}\left((\mathbf{h}^{(r)})^T\mathbf{x}_j\right) \neq y_j\})$, and they are normalized to sum up to 1. After a predefined number of iterations R, the final hyperplane weight vector (strong classifier) is constructed as $\mathbf{h} = \sum_{r=1}^R \alpha^{(r)}\mathbf{h}^{(r)}$.

Exploiting the similarity between the one-bit parameter estimation and binary classification problem, we propose an AdaBoost-based DOA estimators in the sequel.

III. CS-AdaBoost

We begin with constructing an over-complete dictionary $\underline{\mathbf{A}}(\underline{\boldsymbol{\theta}}) = [\mathbf{a}(\underline{\theta}_1), \mathbf{a}(\underline{\theta}_2), \ldots, \mathbf{a}(\underline{\theta}_Q)] \in \mathbb{C}^{M\times Q}$ by discretizing the DOA range $[-\frac{\pi}{2}, \frac{\pi}{2}]$ into Q ($Q \gg K$) equidistant angular grid points $\underline{\boldsymbol{\theta}} = [\underline{\theta}_1, \underline{\theta}_2, \ldots, \underline{\theta}_Q]^T$. Then, we can recast (1) as

$$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] = \mathcal{Q}\left(\underline{\mathbf{A}}(\underline{\boldsymbol{\theta}})\underline{\mathbf{S}} + \mathbf{Z}\right) \tag{2}$$

where $\underline{\mathbf{S}} \in \mathbb{C}^{Q\times N}$ is the extended source signal matrix. The extended source signal $\underline{\mathbf{S}} \in \mathbb{C}^{Q\times N}$ defined in (2) can be viewed as a row sparse matrix. This property of $\underline{\mathbf{S}}$ is taken into account in designing the proper weak classifiers within a CS AdaBoost-based one-bit DOA estimator. Particularly, we use the hard-thresholding operator in the second stage of weak classifiers to ensure the estimated $\underline{\mathbf{S}}$ in each iteration is a K row sparse matrix. Towards this end, after applying the vectorization operator on (2), we obtain

$$\begin{aligned}\mathbf{y} = \mathrm{vec}\{\mathbf{Y}\} &= \mathcal{Q}\Big((\mathbf{I}_N \otimes \underline{\mathbf{A}}(\underline{\boldsymbol{\theta}}))\underline{\mathbf{s}} + \mathbf{z}\Big) \\ &= \mathcal{Q}\left(\underline{\boldsymbol{\Psi}}(\underline{\boldsymbol{\theta}})\underline{\mathbf{s}} + \mathbf{z}\right)\end{aligned} \tag{3}$$

where $\underline{\mathbf{s}} \triangleq \mathrm{vec}\{\underline{\mathbf{S}}\} \in \mathbb{C}^{QN}$, $\mathbf{z} \triangleq \mathrm{vec}\{\mathbf{Z}\} \in \mathbb{C}^{MN}$, and $\underline{\boldsymbol{\Psi}}(\underline{\boldsymbol{\theta}}) \triangleq (\mathbf{I}_N \otimes \underline{\mathbf{A}}(\underline{\boldsymbol{\theta}})) \in \mathbb{C}^{MN\times QN}$. Converting (3) into the real domain, we have

$$\mathbf{y}_{\mathrm{R}} = \mathrm{sign}\left(\underline{\boldsymbol{\Psi}}_{\mathrm{R}}\underline{\mathbf{s}}_{\mathrm{R}} + \mathbf{z}_{\mathrm{R}}\right) \tag{4a}$$

where

$$\begin{aligned}\mathbf{y}_{\mathrm{R}} &\triangleq \left[\Re\{\mathbf{y}\}^T, \Im\{\mathbf{y}\}^T\right]^T = [y_{\mathrm{R},1}, y_{\mathrm{R},2}, \ldots, y_{\mathrm{R},2MN}]^T \\ &\in \{\pm 1\}^{2MN}\end{aligned} \tag{4b}$$

$$\begin{aligned}\underline{\boldsymbol{\Psi}}_{\mathrm{R}} &\triangleq \begin{bmatrix}\Re\{\underline{\boldsymbol{\Psi}}(\underline{\boldsymbol{\theta}})\} & -\Im\{\underline{\boldsymbol{\Psi}}(\underline{\boldsymbol{\theta}})\} \\ \Im\{\underline{\boldsymbol{\Psi}}(\underline{\boldsymbol{\theta}})\} & \Re\{\underline{\boldsymbol{\Psi}}(\underline{\boldsymbol{\theta}})\}\end{bmatrix} \\ &= \left[\underline{\boldsymbol{\psi}}_{\mathrm{R},1}, \underline{\boldsymbol{\psi}}_{\mathrm{R},2}, \ldots, \underline{\boldsymbol{\psi}}_{\mathrm{R},2MN}\right]^T \in \mathbb{R}^{2MN\times 2QN}\end{aligned} \tag{4c}$$

$$\underline{\mathbf{s}}_{\mathrm{R}} \triangleq \left[\Re\{\underline{\mathbf{s}}\}^T, \Im\{\underline{\mathbf{s}}\}^T\right]^T \tag{4d}$$

$$\mathbf{z}_{\mathrm{R}} \triangleq \left[\Re\{\mathbf{z}\}^T, \Im\{\mathbf{z}\}^T\right]^T \ . \tag{4e}$$

For ease of presentation, the dependency of $\underline{\boldsymbol{\Psi}}_{\mathrm{R}}$ on $\underline{\boldsymbol{\theta}}$ is dropped in (4). Estimating $\underline{\mathbf{s}}_{\mathrm{R}}$ in (4) can be viewed as a binary classification problem, where $\{\underline{\boldsymbol{\psi}}_{\mathrm{R},j}\}$ and $\{y_{\mathrm{R},j}\}$ for $j = 1, \ldots, 2MN$ are the training examples and binary class labels, respectively, and $\underline{\mathbf{s}}_{\mathrm{R}}$ is the weight vector of the separating hyperplane. We propose a CS AdaBoost-based method to estimate $\underline{\mathbf{s}}_{\mathrm{R}}$ in (4). A two-stage weak classifier is used in each iteration of the CS-AdaBoost method. The procedure is described in Algorithm 1. In the first stage of the r-th iteration, we solve the following weighted underdetermined system of linear equations:

$$\mathbf{y}_{\mathrm{R}} = \mathbf{W}_{\mathrm{R}}^{(r)} \underline{\mathbf{\Psi}}_{\mathrm{R}} \underline{\mathbf{s}}_{\mathrm{R}}^{(r,1)} \tag{5a}$$

where

$$\mathbf{W}_{\mathrm{R}}^{(r)} \triangleq \mathrm{diag}\left\{\left[\frac{1}{\sqrt{\beta_1^{(r)}}}, \frac{1}{\sqrt{\beta_2^{(r)}}}, \ldots, \frac{1}{\sqrt{\beta_{2MN}^{(r)}}}\right]\right\} \tag{5b}$$

$$\beta_j^{(r)} \triangleq \frac{\beta_j^{(r-1)} \exp\left(\alpha^{(r-1)} \mathbf{1}\left\{\mathrm{sign}\left((\underline{\mathbf{s}}_{\mathrm{R}}^{(r-1,2)})^T \underline{\boldsymbol{\psi}}_{\mathrm{R},j}\right) \neq y_{\mathrm{R},j}\right\}\right)}{\sum_{j=1}^{2MN} \beta_j^{(r)}}$$

$$\text{for} \quad j = 1, \ldots, 2MN \tag{5c}$$

$$\alpha^{(r-1)} \triangleq \frac{1}{2} \ln\left(\frac{1-\epsilon^{(r-1)}}{\epsilon^{(r-1)}}\right) \tag{5d}$$

$$\epsilon^{(r-1)} \triangleq \sum_{j=1}^{2MN} \beta_j^{(r-1)} \mathbf{1}\left\{\mathrm{sign}\left((\underline{\mathbf{s}}_{\mathrm{R}}^{(r-1,2)})^T \underline{\boldsymbol{\psi}}_{\mathrm{R},j}\right) \neq y_{\mathrm{R},j}\right\} \tag{5e}$$

where the roles of β_j's, α's, and ϵ's are elaborated in Section II. In (5), the classification error of the $(r-1)$-th weak classifier is taken into account through the exponential relation of $\beta_j^{(r)}$'s in (5c) for designing the weight matrix $\mathbf{W}_{\mathrm{R}}^{(r)}$ and obtaining the r-th weak classifier. Among infinite number of solutions for the underdetermined system of linear equations in (5), we select the least ℓ_2-norm solution obtained by solving the following optimization problem:

$$\min_{\underline{\mathbf{s}}_{\mathrm{R}}^{(r,1)}} \left\|\underline{\mathbf{s}}_{\mathrm{R}}^{(r,1)}\right\|_2 \tag{6a}$$

$$\text{subject to} \quad \mathbf{y}_{\mathrm{R}} = \mathbf{W}_{\mathrm{R}}^{(r)} \underline{\mathbf{\Psi}}_{\mathrm{R}} \underline{\mathbf{s}}_{\mathrm{R}}^{(r,1)} \tag{6b}$$

The solution of (6) is given as

$$\begin{aligned} \underline{\mathbf{s}}_{\mathrm{R}}^{(r,1)} &= \left(\mathbf{W}_{\mathrm{R}}^{(r)} \underline{\mathbf{\Psi}}_{\mathrm{R}}\right)^T \left(\mathbf{W}_{\mathrm{R}}^{(r)} \underline{\mathbf{\Psi}}_{\mathrm{R}} \left(\mathbf{W}_{\mathrm{R}}^{(r)} \underline{\mathbf{\Psi}}_{\mathrm{R}}\right)^T\right)^{-1} \mathbf{y}_{\mathrm{R}} \\ &= \underline{\mathbf{\Psi}}_{\mathrm{R}}^T \left(\underline{\mathbf{\Psi}}_{\mathrm{R}} \underline{\mathbf{\Psi}}_{\mathrm{R}}^T\right)^{-1} \left(\mathbf{W}_{\mathrm{R}}^{(r)}\right)^{-1} \mathbf{y}_{\mathrm{R}} \ . \end{aligned} \tag{7}$$

Since calculating $\left(\underline{\mathbf{\Psi}}_{\mathrm{R}} \underline{\mathbf{\Psi}}_{\mathrm{R}}^T\right)^{-1}$ is computationally expensive, weak classifiers can be approximate estimators, and scaling weak classifiers by a positive number does not affect the final result due to the one-bit quantization, we approximate (7) by setting $\left(\underline{\mathbf{\Psi}}_{\mathrm{R}} \underline{\mathbf{\Psi}}_{\mathrm{R}}^T\right)^{-1} = \mathbf{I}_{2MN}$ as

$$\underline{\mathbf{s}}_{\mathrm{R}}^{(r,1)} \approx \underline{\mathbf{\Psi}}_{\mathrm{R}}^T \left(\mathbf{W}_{\mathrm{R}}^{(r)}\right)^{-1} \mathbf{y}_{\mathrm{R}} = \underline{\mathbf{\Psi}}_{\mathrm{R}}^T \left(\underline{\boldsymbol{\beta}}^{(r)} \odot \mathbf{y}_{\mathrm{R}}\right) \tag{8a}$$

where

$$\underline{\boldsymbol{\beta}}^{(r)} \triangleq \left[\sqrt{\beta_1^{(r)}}, \sqrt{\beta_2^{(r)}}, \ldots, \sqrt{\beta_{2MN}^{(r)}}\right]^T . \tag{8b}$$

We use $\underline{\mathbf{s}}_{\mathrm{R}}^{(r,1)}$ to construct the complex-valued $\underline{\mathbf{s}}^{(r,1)}$ based on (4d), and then form $\underline{\mathbf{S}}^{(r,1)} = \mathrm{unvec}\{\underline{\mathbf{s}}^{(r,1)}\} \in \mathbb{C}^{Q \times N}$.

Algorithm 1 CS-AdaBoost

Input: $\mathbf{Y}$, M, N, K, Q, and R.
Output: $\boldsymbol{\theta}$.

1: Discretize $[-\frac{\pi}{2}, \frac{\pi}{2}]$ into Q equidistant grid points $\underline{\boldsymbol{\theta}} = [\underline{\theta}_1, \underline{\theta}_2, \ldots, \underline{\theta}_Q]^T$.
2: Form $\underline{\mathbf{A}}(\underline{\boldsymbol{\theta}})$.
3: Form $\underline{\mathbf{\Psi}}(\underline{\boldsymbol{\theta}}) = (\mathbf{I}_N \otimes \underline{\mathbf{A}}(\underline{\boldsymbol{\theta}}))$, and $\underline{\mathbf{\Psi}}_{\mathrm{R}} = \left[\underline{\boldsymbol{\psi}}_{\mathrm{R},1}, \underline{\boldsymbol{\psi}}_{\mathrm{R},2}, \ldots, \underline{\boldsymbol{\psi}}_{\mathrm{R},2MN}\right]^T$ as in (4c).
4: Form $\mathbf{y} = \mathrm{vec}\{\mathbf{Y}\}$, and $\mathbf{y}_{\mathrm{R}} = [y_{\mathrm{R},1}, y_{\mathrm{R},2}, \ldots, y_{\mathrm{R},2MN}]^T$ as in (4b).
5. Initialize the AdaBoost weights $\beta_j^{(1)} = 1/2MN$ for $j = 1, 2, ..., 2MN$.
for $r = 1$ to R **do**
6.1: Form $\underline{\boldsymbol{\beta}}^{(r)}$ as in (8b).
6.2: Compute $\underline{\mathbf{s}}_{\mathrm{R}}^{(r,1)} = \underline{\mathbf{\Psi}}_{\mathrm{R}}^T \left(\underline{\boldsymbol{\beta}}^{(r)} \odot \mathbf{y}_{\mathrm{R}}\right)$.
6.3: Construct the complex-valued $\underline{\mathbf{s}}^{(r,1)}$ based on (4d), and form $\underline{\mathbf{S}}^{(r,1)} = \mathrm{unvec}\{\underline{\mathbf{s}}^{(r,1)}\}$.
6.4: Use (9) to obtain $\underline{\mathbf{S}}^{(r,2)}$.
6.5: Construct $\underline{\mathbf{s}}^{(r,2)} = \mathrm{vec}\{\underline{\mathbf{S}}^{(r,2)}\}$ and $\underline{\mathbf{s}}_{\mathrm{R}}^{(r,2)} = \left[\Re\{\underline{\mathbf{s}}^{(r,2)}\}^T, \Im\{\underline{\mathbf{s}}^{(r,2)}\}^T\right]^T$.
6.6: Compute error $\epsilon^{(r)} = \sum_{j=1}^{2MN} \beta_j^{(r)} \mathbf{1}\left\{\mathrm{sign}\left((\underline{\mathbf{s}}_{\mathrm{R}}^{(r,2)})^T \underline{\boldsymbol{\psi}}_{\mathrm{R},j}\right) \neq y_{\mathrm{R},j}\right\}$.
6.7: Compute $\alpha^{(r)} = \frac{1}{2} \ln\left(\frac{1-\epsilon^{(r)}}{\epsilon^{(r)}}\right)$.
6.8: Update $\beta_j^{(r+1)} = \beta_j^{(r)} \exp\left(\alpha^{(r)} \mathbf{1}\left\{\mathrm{sign}\left((\underline{\mathbf{s}}_{\mathrm{R}}^{(r,2)})^T \underline{\boldsymbol{\psi}}_{\mathrm{R},j}\right) \neq y_{\mathrm{R},j}\right\}\right)$, $\forall j$.
6.9: Compute $c^{(r+1)} = \sum_{j=1}^{2MN} \beta_j^{(r+1)}$ and normalize weights as $\beta_j^{(r+1)} = \frac{\beta_j^{(r+1)}}{c^{(r+1)}}$, $\forall j$.
end for
7: Compute $\underline{\mathbf{s}}_{\mathrm{R}} = \sum_{r=1}^{R} \alpha^{(r)} \underline{\mathbf{s}}_{\mathrm{R}}^{(r,2)}$.
8: Construct the complex-valued $\underline{\mathbf{s}}$ based on (4d), and form $\underline{\mathbf{S}} = \mathrm{unvec}\{\underline{\mathbf{s}}\}$.
9: Obtain $\mathbf{f}_{\mathrm{cs}} \in \mathbb{R}^Q$ by computing ℓ_2-norm of each row of $\underline{\mathbf{S}}$.
10: Pick the K DOAs in $\underline{\boldsymbol{\theta}}$ which correspond to the K largest peaks of $\mathbf{f}_{\mathrm{cs}}$ as $\boldsymbol{\theta}$.

To enforce the K-row sparsity characteristic, the second stage of the r-th iteration is designed as

$$\underline{\mathbf{S}}^{(r,2)} = \mathrm{Hard}_K\left(\underline{\mathbf{S}}^{(r,1)}\right) \tag{9}$$

where $\mathrm{Hard}_K(\cdot)$ first computes the ℓ_2-norm of each row of the bracketed matrix to produce a vector, and then it preserves K rows of the bracketed matrix which correspond to the K largest peaks of the vector and sets the other entries to 0. The final part of the r-th weak classifier employed in the CS-AdaBoost method is to construct $\underline{\mathbf{s}}^{(r,2)} = \mathrm{vec}\{\underline{\mathbf{S}}^{(r,2)}\}$ and

$\underline{\mathbf{s}}_{\mathrm{R}}^{(r,2)} = \left[\Re\{\underline{\mathbf{s}}^{(r,2)}\}^T, \Im\{\underline{\mathbf{s}}^{(r,2)}\}^T\right]^T$. The strong classifier after R iterations is computed as $\underline{\mathbf{s}}_{\mathrm{R}} = \sum_{r=1}^{R} \alpha^{(r)} \underline{\mathbf{s}}_{\mathrm{R}}^{(r,2)}$. We form $\underline{\mathbf{s}}$ via (4d) and also obtain $\underline{\mathbf{S}} = \mathrm{unvec}\{\underline{\mathbf{s}}\}$. We compute ℓ_2-norm of each row of $\underline{\mathbf{S}}$ to obtain $\mathbf{f}_{\mathrm{cs}} \in \mathbb{R}^Q$. We then select the K DOAs in $\underline{\boldsymbol{\theta}}$ which correspond to the K largest peaks of $\mathbf{f}_{\mathrm{cs}}$.

It is worth mentioning that the computational complexity order for Algorithm 1 is $\mathcal{O}\Big[RMN\,(1+NQ)\Big]$.

IV. Simulation Results

This section presents numerical simulation examples to evaluate the performance of the proposed CS-AdaBoost one-bit DOA estimator and compare it to the state-of-the-art algorithms, including the one-bit MUSIC and root-MUSIC [8], CBIHT [13], Gr-SBL [15], and OGIR [24] methods. Furthermore, high-precision (HP) root- MUSIC [30] (i.e., when high-precision ADCs are used) and HP CRB [31] are considered as benchmarks. The numbers of iterations and angular grid points for the proposed CS-AdaBoost method are set to $R = 20$ and $Q = 360$, respectively. The numbers of sources, snapshots, and antennas are represented by K, N, and M, respectively. The mean squared error (MSE) is defined as[1]

$$\mathrm{MSE} \triangleq 10\log_{10}\frac{1}{PK}\sum_{i=1}^{P}\sum_{k=1}^{K}(\hat{\theta}_{k,i}-\theta_k)^2$$

where $\theta_{k,i}$ is the k-th DOA estimated in the i-th trial. The number of trials used to compute the MSE is $P = 1000$ throughout this section.

In Fig. 1, we evaluate the performance of the methods tested for the setup of four uncorrelated sources with $\boldsymbol{\theta} = [-32°, -12°, 5°, 37°]$, $M = 40$, and $N = 10$. In Fig. 1, OGIR has the best performance and one-bit root-MUSIC has the worst. All other one-bit DOA estimators perform almost similarly. The performance of the proposed CS-AdaBoost one-bit DOA estimator can be improved if Q is increased.

In Fig. 2, a challenging setup is considered, where $\boldsymbol{\theta} = [-32°, -30°, 5°, 37°]$, $M = 40$, and $N = 40$. For the first two directions, the correlation coefficient is $\rho = 0.95$. Fig. 2 demonstrates that the performance of the proposed CS-AdaBoost one-bit DOA estimator is considerably better than those of other one-bit DOA estimators tested. Furthermore, the proposed CS-AdaBoost method beats the HP root-MUSIC method at low SNRs in this scenario. This comes from the nonlinearity imposed by the one-bit quantizers, which undermines the noise effect at low SNR regimes to some extend.

Fig. 3 shows the performance of the methods tested when the number of snapshots varies for the setup $\boldsymbol{\theta} = [-32°, -30°, 5°, 37°]$, $\rho = 0.95$, $M = 40$, and SNR $= 5$ dB. It can be seen that the proposed CS-AdaBoost one-bit DOA estimator has the best performance among all one-bit DOA estimators tested.

[1]The unit of directions is radian here.

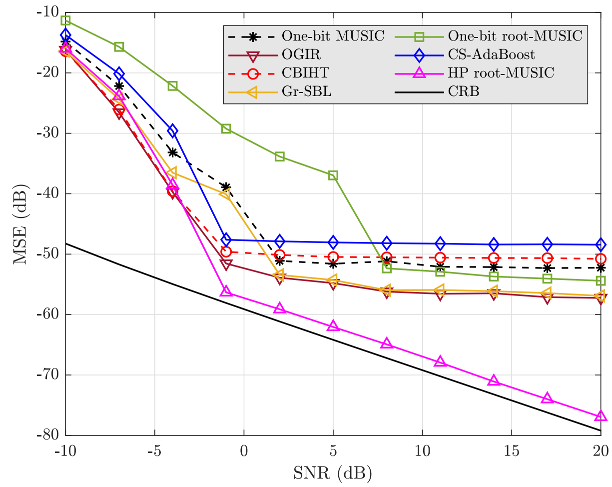

Fig. 1: MSE vs. SNR for $K = 4$ uncorrelated sources with $\boldsymbol{\theta} = [-32°, -12°, 5°, 37°]$, $M = 40$, and $N = 10$.

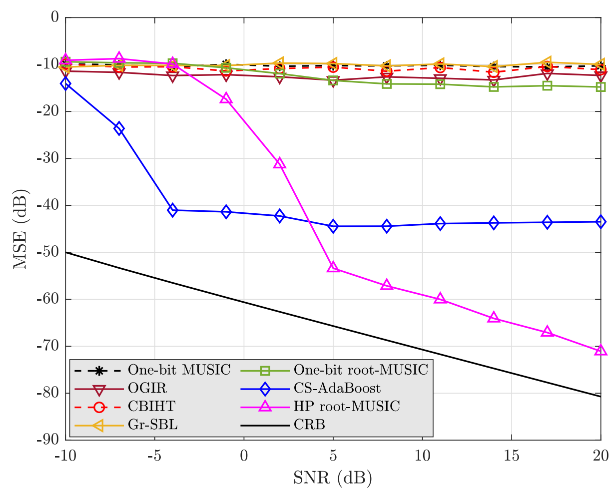

Fig. 2: MSE vs. SNR for $K = 4$ partly correlated sources with $\boldsymbol{\theta} = [-32°, -30°, 5°, 37°]$, $\rho = 0.95$, $M = 40$, and $N = 40$. The proposed CS-AdaBoost method significantly outperforms other one-bit DOA estimators in this scenario.

V. Conclusion

In this paper, we have developed an one-bit DOA estimators, named CS-AdaBoost, which employs two-stage weak classifiers within iterations of an AdaBoost framework to build a strong DOA estimator. It begins with building an over-complete dictionary by discretizing the angular interval into equidistance grid points. The source signal matrix is then estimated using the AdaBoost framework where in each iteration, a weighted least norm estimation is used as the first stage and a hard-thresholding operator is applied for sparsity enforcing as the second stage. Numerical results demonstrate the outstanding superiority of the CS-AdaBoost one-bit DOA estimator, par-

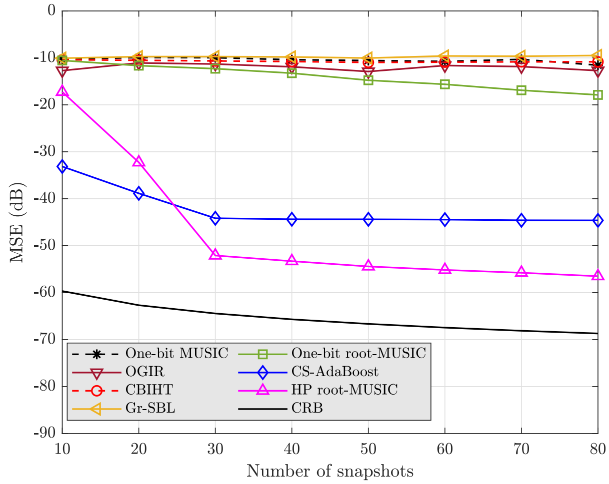

Fig. 3: MSE vs. the number of snapshots for $K = 4$ partly correlated sources with $\boldsymbol{\theta} = [-32°, -30°, 5°, 37°]$, $\rho = 0.95$, $M = 40$, and SNR $= 5$ dB. The proposed CS-AdaBoost method significantly outperforms other one-bit DOA estimators in this scenario.

ticularly in resolving closely spaced and correlated sources, compared to state-of-the-art methods.

References

[1] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 4, pp. 539–550, Apr. 1999.

[2] B. Murmann, "Energy limits in A/D converters," in *Proc. IEEE Faible Tension Faible Consommation*, Paris, France, Jun. 2013, pp. 1–4.

[3] C.-L. Liu and P. P. Vaidyanathan, "One-bit sparse array DOA estimation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, New Orleans, LA, USA, Mar. 2017, pp. 3126–3130.

[4] O. Bar-Shalom and A. J. Weiss, "DOA estimation using one-bit quantized measurements," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 3, pp. 868–884, Jul. 2002.

[5] M. Stein, K. Barbe, and J. A. Nossek, "DOA parameter estimation with 1-bit quantization bounds, methods and the exponential replacement," in *Proc. 20th Int. ITG Workshop Smart Antennas*, 2016, pp. 1–6.

[6] S. Sedighi, B. S. Mysore R., M. Soltanalian, and B. Ottersten, "On the performance of one-bit DoA estimation via sparse linear arrays," *IEEE Trans. Signal Process.*, vol. 69, pp. 6165–6182, Oct. 2021.

[7] J. H. Van Vleck and D. Middleton, "The spectrum of clipped noise," *Proc. IEEE*, vol. 54, no. 1, pp. 2–19, Jan. 1966.

[8] X. Huang and B. Liao, "One-bit MUSIC," *IEEE Signal Process. Lett.*, vol. 26, no. 7, pp. 961–965, Jul. 2019.

[9] X. Huang, S. Bi, and B. Liao, "Direction-of-arrival estimation based on quantized matrix recovery," *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 349–353, Feb. 2020.

[10] I. Yoffe, N. Regev, and D. Wulich, "On direction of arrival estimation with 1-bit quantizer," in *Proc. IEEE Radar Conf.*, 2019, pp. 1–6.

[11] K. Yu, Y. D. Zhang, M. Bao, Y. Hu, and Z. Wang, "DOA estimation from one-bit compressed array data via joint sparse representation," *IEEE Signal Process. Lett.*, vol. 23, no. 9, pp. 1279–1283, Sep. 2016.

[12] P. Xiao, B. Liao, and N. Deligiannis, "DeepFPC: A deep unfolded network for sparse signal recovery from 1-bit measurements with application to DOA estimation," *Signal Process.*, vol. 176, Nov. 2020, Art. no. 107699.

[13] C. Stöckle, J. Munir, A. Mezghani, and J. A. Nossek, "1-bit direction of arrival estimation based on compressed sensing," in *Proc. IEEE 16th Int. Workshop Signal Process. Adv. Wireless Commun.*, 2015, pp. 246–250.

[14] P. Wang, H. Yang, and Z. Ye, "1-Bit direction of arrival estimation via improved complex-valued binary iterative hard thresholding," *Digit. Signal Process.*, vol. 120, Jan. 2022, Art. no. 103265.

[15] X. Meng and J. Zhu, "A generalized sparse Bayesian learning algorithm for 1-bit DOA estimation," *IEEE Commun. Lett.*, vol. 22, no. 7, pp. 1414–1417, Jul. 2018.

[16] X. Meng, S. Wu, and J. Zhu, "A unified Bayesian inference framework for generalized linear models," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 398–402, Mar. 2018.

[17] X.-P. Li, Z.-L. Shi, L. Huang, A. M.-C. So, and H. C. So, "ROCS: Robust one-bit compressed sensing with application to direction of arrival," *IEEE Trans. Signal Process.*, vol. 72, pp. 2407–2420, Apr. 2024.

[18] K. N. Ramamohan, S. P. Chepuri, D. F. Comesana, and G. Leus, "Blind calibration of sparse arrays for DOA estimation with analog and one-bit measurements," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 4185–4189.

[19] Z. Cheng, S. Chen, Q. Shen, J. He, and Z. Liu, "Direction finding of electromagnetic sources on a sparse cross-dipole array using one-bit measurements," *IEEE Access*, vol. 8, pp. 83131–83143, 2020.

[20] C. Zhou, Y. Gu, Z. She, and M. Haardt, "Direction-of-arrival estimation for coprime arrays via coarray correlation reconstruction: A one-bit perspective," in *Proc. IEEE 11th Sensor Array Multichannel Signal Process. Workshop*, 2020, pp. 1–4.

[21] C. Zhou, Z. Zhang, F. Liu, and B. Li, "Gridless compressive sensing method for line spectral estimation from 1-bit measurements," *Digit. Signal Process.*, vol. 60, pp. 152–162, Jan. 2017.

[22] Z. Wei, W. Wang, F. Dong, and Q. Liu, "Gridless one-bit direction-of-arrival estimation via atomic norm denoising," *IEEE Commun. Lett.*, vol. 24, no. 10, pp. 2177–2181, Oct. 2020.

[23] W.-G. Tang, H. Jiang, and Q. Zhang, "One-bit gridless DOA estimation with multiple measurements exploiting accelerated proximal gradient algorithm," *Circ., Syst. Signal Process.*, vol. 41, no. 2, pp. 1100–1114, 2022.

[24] L. Feng, L. Huang, Q. Li, Z.-Q. He, and M. Chen, "An off-grid iterative reweighted approach to one-bit direction of arrival estimation," *IEEE Trans. Vehic. Tech.*, vol. 72, no. 6, pp. 8134–8139, Jun. 2023.

[25] M. Hong, M. Razaviyayn, Z. Q. Luo, and J. S. Pang, "A unified algorithmic framework for block-structured optimization involving Big Data: With applications in machine learning and signal processing," *IEEE Signal Process. Mag.*, vol. 33, no. 1, pp. 57–77, Jan. 2016.

[26] Y. Gao, D. Hu, Y. Chen, and Y. Ma, "Gridless 1-b DOA estimation exploiting SVM approach," *IEEE Commun. Lett.*, vol. 21, no. 10, pp. 2210–2213, Oct. 2017.

[27] T. Hastie, R. Tibshirani, and J. H. Friedman, *The elements of statistical learning: Data mining, inference, and prediction*. Springer, 2009.

[28] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[29] M. Esfandiari, S. A. Vorobyov, and R. W. Heath Jr., "AdaBoost-based efficient channel estimation and data detection in one-bit massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 13935–13945, Oct. 2024.

[30] A. J. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *Proc. ICASSP*, Boston, MA, 1983, pp. 336–339.

[31] P. Stoica, E. G. Larsson, and A. B. Gershman, "The stochastic CRB for array processing: A textbook derivation," *IEEE Signal Process. Lett.*, vol. 8, no. 5, pp. 148–150, May 2001.