Near-Optimal Bayesian Localization Via Incoherence and Sparsity

Volkan Cevher, Petros Boufounos, Richard Baraniuk, Anna Gilbert, Martin Strauss

Abstract

This paper exploits recent development in sparse approximation and compressive sensing to efficiently perform localization in a sensor network. We introduce a Bayesian framework for the localization problem and provide sparse approximations to its optimal solution. By exploiting the spatial sparsity of the posterior density, we demonstrate that the optimal solution can be computed using fast sparse approximation algorithms. We show that exploiting the signal sparsity can reduce the sensing and computational cost on the sensors, as well as the communication bandwidth. We further illustrate that the sparsity of the source locations can be exploited to decentralize the computation of the source locations and reduce the sensor communications even further. We also discuss how recent results in 1-bit compressive sensing can significantly reduce the amount of inter-sensor communications by transmitting only the intrinsic timing information. Finally, we develop a computationally efficient algorithm for bearing estimation using a network of sensors with provable guarantees.

IPSN 2009

Near-Optimal Bayesian Localization
via Incoherence and Sparsity

Volkan Cevher[*]
Rice University

Petros Boufounos
MERL

Richard G. Baraniuk
Rice University

Anna C. Gilbert
University of Michigan

Martin J. Strauss
University of Michigan

ABSTRACT

This paper exploits recent developments in sparse approximation and compressive sensing to efficiently perform localization in a sensor network. We introduce a Bayesian framework for the localization problem and provide sparse approximations to its optimal solution. By exploiting the spatial sparsity of the posterior density, we demonstrate that the optimal solution can be computed using fast sparse approximation algorithms. We show that exploiting the signal sparsity can reduce the sensing and computational cost on the sensors, as well as the communication bandwidth. We further illustrate that the sparsity of the source locations can be exploited to decentralize the computation of the source locations and reduce the sensor communications even further. We also discuss how recent results in 1-bit compressive sensing can significantly reduce the amount of inter-sensor communications by transmitting only the intrinsic timing information. Finally, we develop a computationally efficient algorithm for bearing estimation using a network of sensors with provable guarantees.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Distributed networks; G.1.6 [**Numerical Analysis**]: Optimization—*Con-strained optimization, convex programming, nonlinear programming*; G.1.2 [**Approximation**]: Nonlinear approximation—*Sparse approximation*

General Terms

Theory, Algorithm, Performance.

Keywords

Sparse approximation, spatial sparsity, localization, bearing estimation, sensor networks.

1. INTRODUCTION

Source localization using a network of sensors is a classical problem with diverse applications in tracking, habitat monitoring, etc. A solution to this problem in practice must satisfy a number of competing resource constraints, such as estimation accuracy, communication and energy costs, signal sampling requirements and computational complexity. A plethora of localization solutions exists with emphasis on one or some of these constraints.

Unfortunately, most of the localization solutions do not provide an end-to-end solution starting from signals to location estimates with rigorous guarantees or they are computationally inefficient. For instance, in a number of emerging applications, such as localizing transient events (e.g., sniper fire [21]) or sources hidden in extremely large bandwidths, sampling source signals at Nyquist rate is extremely expensive and difficult for resource-constrained sensors. Even if the sources can be sampled at the Nyquist rate, in many cases, we end up with too many samples and must compress to store or communicate them. Furthermore, even if a processor in the network can receive such a large amount of data, its localization algorithms to locate the targets is frequently slow, inefficient, and high-dimensional. These algorithms make it infeasible for the required computation to be distributed at the sensor level as reasonable hardware costs.

This paper re-examines the problem of target localization in sensor networks and uses recent results in sparse

approximation and compressive sensing (CS) to provide a fundamentally different approach with near Bayesian optimality guarantees and highly efficient end-to-end algorithms that are both rigorous theoretically and practical in real settings. In particular, we show that (*i*) the Bayesian model order selection formulation to determine the number of sources naturally results in a sparse approximation problem, (*ii*) the sparse localization solution naturally lends itself to decentralized estimation, (*iii*) it is possible to reduce communication significantly by exploiting the spatial sparsity of the sources as well as 1-bit quantization schemes, and (*iv*) a simple greedy (matching pursuit) algorithm provides provable recovery guarantees for special localization cases.

We introduce a Bayesian framework for target localization using graphical models in Sect. 3. Optimal localization under this model is performed by computing the maximum a posteriori (MAP) estimate of the number of sources and the source locations. The model is quite general and can be applied in a variety of localization scenarios.

We develop a discretization of the optimal Bayesian solution that exploits the sparsity of the posterior density functions in Sect. 5. This discretization enables the use of very efficient optimization algorithms to jointly compute a near-optimal MAP estimate of the number of sources and their locations. As with the Bayesian framework itself, this discretization is quite general.

We exploit source signal sparsity, when available, in two ways in Sec 6.1. First, we reduce the analog-to-digital sampling requirements on the sensor, and therefore its cost, using CS techniques. Second, we reduce the amount of communication required per sensor, and therefore its power consumption, by compressing the signal at each sensor as it is acquired. The latter allows us to efficiently transmit the data to a central location to localize the sources.

We exploit source incoherence and spatial sparsity to efficiently decentralize the localization problem using CS in Sect. 6.2. Each sensor can build a local representation (dictionary) for the problem and use compressive measurements of the remaining sensor network data to efficiently localize the sources while using limited communication bandwidth. Such decentralized processing does not require that the source signals be sparse, only that there are few sources distributed in space.

We capitalize on recent 1-bit CS results to further reduce the communication requirements for distributed localization in Sect. 7. Specifically, by transmitting only the sign of the compressive measurements we eliminate all the amplitude information from the sensor data and communicate only phase and timing information. Thus, it is not necessary that the amplitude gain of each sensor or the received-signal-strength (RSS) be known when the localization is performed. Timing information enables more robust and accurate recovery of the source locations as compared with communication-constrained approaches that transmit the re-

ceived signal strength at each sensor.

Our experiments with both real and simulated data in Sect. 9 indicate that our theoretical approach has practical significance. We show that in realistic situations, our new Bayesian framework reduces the amount of data collection and communication by a significant margin with graceful or no degradation in the localization accuracy.

Prior Work. Related localization approaches have been considered in [11, 18, 19, 7, 13, 6]. In [11], spatial sparsity is used to improve localization performance; however the computational complexity of the presented algorithm is high, since it uses the high-dimensional received signals. Dimensionality reduction through principal components analysis was proposed in [18] to optimize a maximum likelihood cost; however, this technique is contingent on knowledge of the number of sources present for acceptable performance and also requires the transmission of all the sensor data to a central location to perform singular value decomposition. Similar to [18], we do not assume the source signals are incoherent. In [19], along with the spatial sparsity assumption, the authors assume that the received signals are also sparse in some known basis and perform localization in near and far fields; however, similar to [11], the authors use the high-dimensional received signals and the proposed method has high complexity and demanding communication requirements. Moreover, the approach is centralized and is not suitable for resource constrained sensor network settings. CS was employed for compression in [7,13], but the method was restricted to far-field bearing estimation. In [6], the authors extend CS-based localization setting to near-field estimation with a maximum likelihood formulation, and examine the constraints necessary for accurate estimation in the number of measurements and sensors taken, the allowable amount of quantization, the spatial resolution of the localization grid, and the conditions on the source signals. In this paper, we build on the preliminary results in [6] by showing the optimality of the sparse approximation approaches in Bayesian inference. Compared to [6], we also provide a new 1-bit framework and an efficient matching pursuit algorithm with provable guarantees for cases where the sensors network to calculate source bearings.

2. COMPRESSIVE SENSING BACKGROUND

Compressive sensing (CS) exploits sparsity to acquire high-dimensional signals using very few linear measurements [1, 9, 5]. Specifically, consider a vector $\boldsymbol{\theta}$ in an N-dimensional space which is K-sparse, i.e., has only K non-zero components. Using compressive sensing, this vector can be sampled and reconstructed with only $M = O(K \log(N/K))$ linear measurements:

$$\chi = \boldsymbol{\Phi\theta} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{\Phi}$ is a $M \times N$ measurement matrix, χ are the measurements and $\boldsymbol{n}$ is the measurement noise.

The sparse vector $\boldsymbol{\theta}$ can subsequently be recovered from the measurements using the following convex optimization:

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_1 + \lambda \|\chi - \boldsymbol{\Phi}\boldsymbol{\theta}\|_2^2, \qquad (2)$$

where the ℓ_p norm is defined as $\|\boldsymbol{\theta}\|_p = \left(\sum_i |\theta_i|^p\right)^{1/p}$, and λ is a relaxation parameter that depends on the noise variance. It can also be recovered using greedy algorithms, such as the ones in [23,24] and references within.

In the absence of noise and under certain conditions on $\boldsymbol{\Phi}$, both convex optimization and several greedy algorithms exactly recover $\boldsymbol{\theta}$ [5]. This formulation is robust even if the vector is not sparse but compressible, i.e., has very few significant coefficients and can be well approximated by a K-sparse representation [1,9,5].

A sufficient but not necessary condition on $\boldsymbol{\Phi}$ to recover the signal using (2) is a restricted isometry property (RIP) or order $2K$. This property states that there is a sufficiently small and positive δ such that for any $2K$ sparse signal $\boldsymbol{\theta}$:

$$(1-\delta)\|\boldsymbol{\theta}\|_2^2 \le \|\boldsymbol{\Phi}\boldsymbol{\theta}\|_2^2 \le (1+\delta)\|\boldsymbol{\theta}\|_2^2. \qquad (3)$$

Although in general it is combinatorially complex to verify the RIP on an arbitrary measurement matrix $\boldsymbol{\Phi}$, a surprising result in CS is that a randomly generated $\boldsymbol{\Phi}$ with $M = O(K\log(N/K))$ rows satisfies the RIP with overwhelming probability.

The same framework applies if a vector is sparse in a sparsity-inducing basis or dictionary $\boldsymbol{\Psi}$ instead of the canonical domain. Specifically, if $\mathbf{a} = \boldsymbol{\Psi}\boldsymbol{\theta}$, where $\mathbf{a}$ is the measured signal instead of $\boldsymbol{\theta}$, and $\boldsymbol{\Psi}$ is the sparsity-inducing dictionary, then (1) becomes

$$\chi = \boldsymbol{\Phi}\mathbf{a} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\theta} + \mathbf{n} \qquad (4)$$

Thus the problem is reformulated as the recovery of a sparse $\boldsymbol{\theta}$ from χ, acquired using the measurement matrix $\boldsymbol{\Phi}\boldsymbol{\Psi}$.

If the sparsity-inducing basis is the Fourier basis, then it is possible to sample and reconstruct the signal using extremely efficient Fourier sampling algorithms, such as the ones presented in [10]. The advantage of these algorithms is that they can operate with complexity that is sublinear in the dimensionality of the signal, making them appropriate for very large signals in computationally constrained environments.

3. BAYESIAN INFERENCE FOR LOCALIZATION

3.1 Problem set up and notation

Our objective is to determine the locations of K sources in a known planar deployment area using the signals received by a network of Q sensors. We assume that neither the number of sources K nor the source signals are known. We denote the horizontal h and vertical v location of q-th sensor ($q = 1, \ldots, Q$) by $\boldsymbol{s}_q = (s_{qh}, s_{qv})'$ with respect to a known origin. We assume that the sensor network is calibrated so

that the location of each sensor is known across the network. We also assume that the local clocks of the sensors are synchronized within $\pm\delta$ seconds.

We denote the received signal vector at the q-th sensor by $\boldsymbol{z}_q$ and its Fourier transform by $\boldsymbol{Z}_q = \mathcal{F}\boldsymbol{z}_q$, where $\mathcal{F}$ is the Fourier transform operator. The time vector $\boldsymbol{z}_q$ is formed by concatenating T received signal samples $z_q(t)$ at times $t = t_1, \ldots, t_T$. Similarly, the frequency vector is formed by concatenating T Fourier samples $Z_q(\omega)$ at frequencies $\omega = \omega_1, \ldots, \omega_T$, corresponding to the time vector $\boldsymbol{z}_q$. We then denote the unknown source signal vectors, their Fourier transforms and locations by $\boldsymbol{y}_k$, $\boldsymbol{Y}_k$ and $\boldsymbol{x}_k = (x_{kh}, x_{kv})'$, respectively for $k = 1, \ldots, K$. Finally, we represent the full sensor network data by the $(Q \times T) \times 1$ dimensional vector $\boldsymbol{Z} = (\boldsymbol{Z}_1', \ldots, \boldsymbol{Z}_Q')'$ in frequency and similarly by $\boldsymbol{z}$ in time.

3.2 Signal Propagation and the Sensor Observations

We denote $\mathcal{A}$ as the signal propagation operator, which takes a source signal $\boldsymbol{y}$ and its location $\boldsymbol{x}$ and calculates in the observed signal $\boldsymbol{z}$ at a location $\boldsymbol{s}$ via

$$\boldsymbol{z} = \mathcal{A}_{\boldsymbol{x} \to \boldsymbol{s}}\left[\boldsymbol{y}\right]. \qquad (5)$$

In an isotropic medium with a propagation speed of c, $\mathcal{A}$ is a linear operator, known as the Green's function, with a particularly distinguished form in the frequency domain:

$$\mathcal{A}_{\boldsymbol{x} \to \boldsymbol{s}} : Z(\omega) = \frac{1}{\|\boldsymbol{x} - \boldsymbol{s}\|^\alpha} \exp\left(-j\omega \frac{\|\boldsymbol{x} - \boldsymbol{s}\|}{c}\right) Y(\omega), \quad (6)$$

where $j = \sqrt{-1}$ and α is the attenuation constant that depends on the nature of the propagation.

In the sequel, we assume that $\mathcal{A}$ is the Green's function and $\alpha = 1$ (spherical propagation). In this specific case, (5) can be represented with linear matrix equation in frequency domain due to (6). Hence, without loss of generality, we will discuss the location problem and its solution in the frequency domain. In general localization problems, $\mathcal{A}$ must be learned or simulated to account for unisotropic media and multipath. Note that the algorithms in this paper can be modified to handle different operators as long as they are linear.

When $\mathcal{A}$ is the Green's function, the sensor network data $\boldsymbol{Z}$ can be written as a superposition of the source signals:

$$\boldsymbol{Z} = \sum_{k=1}^{K} \boldsymbol{A}(\boldsymbol{x}_k)\boldsymbol{Y}_k + \boldsymbol{N} \qquad (7)$$

where $\boldsymbol{N}$ is an additive noise, and $\boldsymbol{A}$ is the mixing matrix for the sensor network due to (6) with the following form:

$$\boldsymbol{A}(\boldsymbol{x}_k) = \begin{bmatrix} \boldsymbol{A}_1(\boldsymbol{x}_k) \\ \vdots \\ \boldsymbol{A}_Q(\boldsymbol{x}_k) \end{bmatrix}_{(QT \times T)}, \qquad (8)$$

$$[\boldsymbol{A}_q(\boldsymbol{x}_k)]_{lm} = \begin{cases} \frac{1}{\|\boldsymbol{x}_k - \boldsymbol{s}_q\|} \exp\left(-j\omega_l \frac{\|\boldsymbol{x}_k - \boldsymbol{s}_q\|}{c}\right), & l = m, \\ 0, & l \ne m, \end{cases} \quad (9)$$

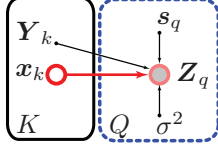where $l = 1, \ldots, T$; $m = 1, \ldots, T$; and $q = 1, \ldots, Q$.

Figure 1: A visual representation of the localization problem as a directed acyclic graph.

3.3 Graphical Model and the Inference Problem

We cast the localization problem as a latent variable estimation problem in Bayesian inference. To summarize the inter-dependencies amongst the relevant variables, we visualize the localization problem in Fig. 1 with a directed acyclic graphical model. In the graphical model, the dashed box denotes the set of Q sensor observations $\boldsymbol{Z}_q$, which are assumed to be independent, in plate notation [2]. The shaded node within the dashed box represents the observed variables. The nodes in the solid box represents the latent variables, namely, the number of targets K, the k-th source signal $\boldsymbol{Y}_k$ and its location $\boldsymbol{x}_k$. The deterministic components of the problem are shown with solid dots, such the sensor positions $\boldsymbol{s}_q$'s and — as an example — the additive noise variance σ^2 at the sensors. In Fig. 1, arrows indicate the causal relationships, where the distribution of the variables at the head of an arrow depends on the variables on at the tail.

Within the graphical model of Fig. 1, the latent variable pair $(K, \boldsymbol{Y})$ defines a model $\mathcal{M}_K$, labeled by K, where $\boldsymbol{Y} = (\boldsymbol{Y}'_1, \ldots, \boldsymbol{Y}'_K)'$. Note that the length of the vector of source signals $\boldsymbol{Y}$ depends on K. Each model $\mathcal{M}_K$ refers to a different probability density function (PDF) over the observed variables as we vary K and $\boldsymbol{Y}$. The comparison of posterior density of each model $\mathcal{M}_K$ for different values of K enables us the determine the number of targets and their source vectors.

Given the sensor network observations $\boldsymbol{Z}$, the posterior density of $\mathcal{M}_K$ can be determined using Bayes' rule:

$$p(\mathcal{M}_K|\boldsymbol{Z}) \propto p(\boldsymbol{Z}|\mathcal{M}_K)p(\mathcal{M}_K), \qquad (10)$$

where $p(\mathcal{M}_K)$ is the prior distribution of each model $\mathcal{M}_K$, and $p(\boldsymbol{Z}|\mathcal{M}_K)$ is the model evidence distribution. In the localization problem, we assume that the sources $\boldsymbol{Y}$ are unknown parameters that are uniformly distributed in their natural space. Hence, the model prior only depends on the number of targets K:

$$p(\mathcal{M}_K) \propto p(K). \qquad (11)$$

This prior $p(K)$ incorporates known information on the number of targets. In this paper we use an exponential prior, which penalizes large numbers of targets:

$$p(K) \propto \exp{-\lambda K}. \qquad (12)$$

In general, we do not have a direct expression for the model evidence $p(\boldsymbol{Z}|\mathcal{M}_K)$. To determine $p(\boldsymbol{Z}|\mathcal{M}_K)$, we marginalize the latent variables $\boldsymbol{X} = (\boldsymbol{x}'_1, \ldots, \boldsymbol{x}_K)'$ via

$$p(\boldsymbol{Z}|\mathcal{M}_K) = \int p(\boldsymbol{Z}|\boldsymbol{X}, \mathcal{M}_K)p(\boldsymbol{X}|\mathcal{M}_K)d\boldsymbol{X}. \qquad (13)$$

by using $p(\boldsymbol{Z}|\boldsymbol{X}, \mathcal{M}_K)$ and $p(\boldsymbol{X}|\mathcal{M}_K)$, which are the probability density function (PDF) of the sensor network data and the prior distribution of the source locations, respectively.

The PDF $p(\boldsymbol{Z}|\boldsymbol{X}, \mathcal{M}_K)$ is determined by the physics of signal propagation and the sensor observations. Assuming i.i.d. zero means Gaussian noise with variance σ^2 at the sensors, we obtain via (7) that

$$p(\boldsymbol{Z}|\boldsymbol{X}, \mathcal{M}_K) \sim \mathcal{N}\left(\boldsymbol{Z}\,\middle|\,\sum_{k=1}^{K}\boldsymbol{A}(\boldsymbol{x}_k)\boldsymbol{Y}_k, \sigma^2\boldsymbol{I}\right), \qquad (14)$$

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is shorthand notation for the Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

On the other hand, the prior distribution on the target locations summarizes our prior knowledge on the locations. A quick inspection of the graphical model (Fig. 1) reveals that $\boldsymbol{Y}_k$ and the $\boldsymbol{x}_k$'s are independent. Hence, the prior distribution of the source locations has the following form:

$$p(\boldsymbol{X}|\mathcal{M}_K) = p(\boldsymbol{X}|K). \qquad (15)$$

If we have prior information on the source locations, then it can be incorporated in the above. However, for the remainder of this paper we assume a uniform prior on the locations.

In the optimal Bayesian source location estimation, we first choose the single most probable model among $\mathcal{M}_K$ alone to make a good prediction. Hence, we focus on the maximum a posteriori (MAP) estimate of the PDF in (10):

$$\widehat{\mathcal{M}}_K = \arg\max p(\mathcal{M}_K|\boldsymbol{Z}). \qquad (16)$$

Then, given this MAP estimate of the model, localization becomes an inference problem from the posterior of the target locations. Specifically, the MAP estimate of the locations can be obtained as

$$\widehat{\boldsymbol{X}} = \arg\max p(\boldsymbol{X}|\boldsymbol{Z}, \widehat{\mathcal{M}}_K). \qquad (17)$$

We emphasize that the MAP estimate under the model assumptions can only be unique up to a permutation of the sources since a re-indexing of the source locations $\boldsymbol{x}_k$ does not change the problem or the data. Therefore, (17) is symmetric to permutations of $\boldsymbol{X}$.

4. THE PRICE OF OPTIMALITY: SAMPLING, COMMUNICATION AND COMPUTATIONAL CHALLENGES

To realize the Bayesian solution in a sensor network, we must (*i*) sample the received signals at their Nyquist rate, (*ii*) communicate the sensor observations $\boldsymbol{Z}$ to a collection point, and (*iii*) solve the optimization problems corresponding to (16) and (17). In this section, we discuss these issues in detail and describe how they are traditionally handled.

In numerous localization applications, such as acoustic vehicle tracking, human speaker localization, etc. [7,8], the necessary source Nyquist rate is typically quite low. Hence, the cost and form factor of the required analog-to-digital converter (ADC) hardware in each sensor are quite manageable. However, in a number of emerging applications, such as localizing transient events (e.g., sniper fire [21]) or sources hidden in extremely large bandwidths, Nyquist rate sampling is extremely expensive and difficult.

Even if the sources can be sampled at the Nyquist rate, it is often necessary to compress the samples before storing or communicating them. Compression reduces the storage requirements of a T-dimensional signal by representing it in a domain where most of the coefficients are zero or close to zero, i.e., in a domain where the signal is sparse or compressible, respectively (for example, the Fourier, DCT, or wavelet domain). Classical compression then encodes only the magnitude and location of the most significant coefficients. Compressive sensing (CS) addresses the inefficiency of the classical *sample-then-compress* scheme by developing theory and hardware to directly obtain a compressed representation of sparse or compressible signals [5,9].

The sensor network communications necessary to solve (16) and (17) in a centralized manner scale with the product of the source sparsity and the number of sensors (see [8] for an example application). The communication requirements can still be quite demanding on the resources, for example, in practical battery operated or wireless sensor network. Hence, lossy compression of the observed signals is typically used. As an example, the received-signal-strength (RSS) of the observed signals can be used as an aggressive compression scheme. Such aggressive lossy compression schemes focus on distilling the observed signals to the smallest sketch possible but may result in significant accuracy losses in the target location estimates. The focus of Sect. 6 is on new signal compression schemes designed to maintain information necessary for the localization problem.

Although required by the optimal Bayesian solution centralized processing has many disadvantages, such as creating communication bottlenecks and catastrophic points of failure. Moreover, the resulting optimization problems are high dimensional. Hence, approximate inference methods on graphical models such as (loopy) belief propagation, junction-tree algorithms, and variational methods based on convex duality are often used to distribute the resulting inference problem over the individual sensors of the network [15, 12, 20]. As a result, the computation load of the inference task is also distributed across the sensors.

The underlying message passing mechanism of distributed (approximate) inference methods require communication of local beliefs on the latent variables whose size is proportional to the desired resolution of the latent variables. However, further compression can be achieved by parameterizing the problem with certain kernel basis functions, fitting Gaussian mixtures via fast Gauss transform, or variational approximations [12,15,20]. The resulting approximation algorithms inherit the estimation (or *divergence*) guarantees, such as bounded distortion, of the underlying approximation engine as well as its disadvantages, such as unknown number of communication loops.

5. APPROXIMATE INFERENCE FOR LOCALIZATION

Section 3.3 demonstrated that the Bayesian solution to the localization problem involves the optimization problems (16) and (17). The first corresponds to a model order selection problem which determines the number of targets K, and the second corresponds to the location inference problem which determines the target locations given K. Unfortunately both optimizations are difficult to solve analytically.

In this section we describe a computational approach that uses a discretization of the source location grid to efficiently and accurately compute the optimal solution. Our approach exploits the incoherence of the sources to factorize the optimization problem and the sparsity of the posterior density to compute its sparse approximation on the location grid. Using this sparse approximation we jointly estimate both the number of sources and their locations.

Although we can incorporate a variety of priors, for clarity of the derivations we only treat the case of a uniform prior on the source locations:

$$p(\boldsymbol{X}|K) \propto 1. \tag{18}$$

Thus, the posterior distribution of the target locations is

$$p(\boldsymbol{X}|\boldsymbol{Z}, \mathcal{M}_K) \propto p(\boldsymbol{Z}|\boldsymbol{X}, \mathcal{M}_K) \sim \mathcal{N}\left(\boldsymbol{Z}\left|\sum_{k=1}^{K} \boldsymbol{A}(\boldsymbol{x}_k)\boldsymbol{Y}_k, \sigma^2 \boldsymbol{I}\right.\right). \tag{19}$$

5.1 Estimation of Source Signals

The Bayesian model order selection problem (16) is equivalent to the following optimization via (11):

$$\widehat{\mathcal{M}}_K = \arg\max_K p(K) \max_{\boldsymbol{Y}} \int p(\boldsymbol{Z}|\boldsymbol{X}, \mathcal{M}_K)d\boldsymbol{X}, \tag{20}$$

To solve (20), we make the following observations.

Observation 1: The maximization of the posterior PDF, given in (19), requires us to solve the following least squares problem:

$$\widehat{\boldsymbol{Y}} = \arg\min_{\boldsymbol{Y}} E(\boldsymbol{X}, \boldsymbol{Y}), \text{ where} \tag{21}$$

$$E(\boldsymbol{X}, \boldsymbol{Y}) = \boldsymbol{Z}'\boldsymbol{Z} - 2\boldsymbol{Z}'\left(\sum_{k=1}^{K} \boldsymbol{A}(\boldsymbol{x}_k)\boldsymbol{Y}_k\right) + \left\|\sum_{k=1}^{K} \boldsymbol{A}(\boldsymbol{x}_k)\boldsymbol{Y}_k\right\|^2. \tag{22}$$

If the sources satisfy the following factorization

$$\left\|\sum_{k=1}^{K} \boldsymbol{A}(\boldsymbol{x}_k)\boldsymbol{Y}_k\right\|^2 \approx \sum_{k=1}^{K} \boldsymbol{Y}'_k \boldsymbol{A}'(\boldsymbol{x}_k)\boldsymbol{A}(\boldsymbol{x}_k)\boldsymbol{Y}_k \tag{23}$$

then it is easy to prove that the solution to (21) is given by

$$\widehat{\boldsymbol{Y}}_k = \boldsymbol{A}^\dagger(\boldsymbol{x}_k)\boldsymbol{Z}, \tag{24}$$

where $\dagger$ is the pseudo inverse. When the sources (*i*) have fast decaying autocorrelations and (*ii*) are sufficiently separated in space [6,13], the factorization in (23) is quite accurate.

Observation 2: The optimal source estimates $\widehat{Y}_k$'s in (24) for $k = 1, \ldots, K$ are independent of K given x_k. Then, the maximization operation with respect to Y in (20) can be moved into the integral. The resulting objective automatically ties the source signal estimates with the location estimates and modifies the model selection problem (20) to

$$\widehat{K} = \arg\max_K p(K) \int p(Z|X, K, \widehat{Y}) dX, \text{ where} \quad (25)$$

$$p(Z|X, K, \widehat{Y}) \sim \mathcal{N}\left(Z \left| \sum_{k=1}^{K} A(x_k) A^\dagger(x_k) Z, \sigma^2 I \right. \right). \quad (26)$$

5.2 Discretization of the Source Locations

The Bayesian formulation in Sec. 3 defines the source locations as continuous random vectors in the 2-D plane. In this section, we discretize the plane using an N-point spatial grid. We assume we have a sufficiently dense grid so that each target location x_k is located at one of the N grid points. We then define an N-dimensional grid selector vector $\boldsymbol{\theta}$ with components θ_i that are 1 or 0 depending on whether or not a source is present at grid point i. With this notation, note that the number of sources K is equal to the ℓ_0 norm of $\boldsymbol{\theta}$, which is defined as the number of non-zero elements in the vector. Since $\boldsymbol{\theta}$ is a vector of ones and zeros, the number of ones is also equal to its ℓ_1 norm, defined as $\|\boldsymbol{\theta}\|_1 = \sum_{i=1}^{N} |\theta_i| = K$, where θ_i corresponds to the i-th grid point.

With a slight abuse of notation, we will use $\boldsymbol{\theta}$ interchangeably with X in sequel. By $\boldsymbol{\theta}$, we will refer to either the grid points or the actual physical locations corresponding the nonzero elements of $\boldsymbol{\theta}$, depending on the context.

5.3 Joint Model Selection and Posterior Estimation

Using the discretization in Sec. 5.2, we define a dictionary $\boldsymbol{\Psi}$, whose column i is equal to $A(\theta_i)A^\dagger(\theta_i)Z$. This columns of this dictionary describes how the single source signal would be observed at the sensors if it was located at grid point i. It is possible to show that if the source signals are uncorrelated with each other or have rapidly decaying autocorrelations then the dictionary $\boldsymbol{\Psi}$ is incoherent and well conditioned for sparse approximation [6].

Via (26), the integral in (25) is then lower-bounded by

$$\int p(Z|X, K, \widehat{Y}) dX \geq \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2} \left\| Z - \boldsymbol{\Psi}\widehat{\boldsymbol{\theta}}_K \right\|^2 \right\}, \quad (27)$$

where $\widehat{\boldsymbol{\theta}}_K$ is the best K-sparse vector that minimizes $\|Z - \boldsymbol{\Psi}\boldsymbol{\theta}\|^2$. The intuition behind this lower bound is straight forward. We first approximate the continuous integral by a discrete summation over the grid locations $\boldsymbol{\theta}$. Then, by only keeping K-heavy hitters of $\boldsymbol{\theta}$ (e.g., the best K columns that maximize joint the posterior), we arrive at (27).

We use (27) and (25) to determine the model order:

$$\widehat{K} \approx \arg\min_K \left\| Z - \boldsymbol{\Psi}\widehat{\boldsymbol{\theta}}_K \right\|^2 - 2\sigma^2 \log p(K). \quad (28)$$

Using the exponential prior on K from (12) and substituting the ℓ_1 norm of $\boldsymbol{\theta}$ for K, this optimization becomes

$$\widehat{\boldsymbol{\theta}} \approx \arg\min_{\boldsymbol{\theta}} \|Z - \boldsymbol{\Psi}\boldsymbol{\theta}\|^2 + 2\sigma^2 \lambda \|\boldsymbol{\theta}\|_1, \quad (29)$$

where $\boldsymbol{\theta}$ is a vector of 0 and 1. This optimization jointly solves for the number of sources K and their locations.

The discrete nature of $\boldsymbol{\theta}$ makes (29) a combinatorial optimization problem. To solve it we heuristically relax it and allow $\boldsymbol{\theta}$ to take continuous positive values. Thus the optimization becomes a minimization problem easily solved using linear programming, basis pursuit, or greedy algorithms (for examples see [5] and references within). In practice we observe that this relaxation performs very well. There are guaranteed branch and bound methods to compute the combinatorial minimization using the relaxation results, but they are beyond the scope of this paper.

6. EXPLOITING COMPRESSED SENSING

This section examines how sparsity and compressive sensing can be exploited in sensor networks for source localization. Source signal sparsity, when available, reduces the sensing cost and the communication burden for each sensor. Spatial sparsity distributes the computation of the localization algorithm and subsequently increases the robustness of the network.

6.1 Signal Sparsity

When the signals are sparse in the frequency domain (i.e., have very few significant frequency components), recent results in CS enable the use of cheaper sensors for digital data acquisition. Two promising methods are random demodulation and random sampling [17,16]. Both methods can be efficiently implemented in hardware. Furthermore, random sampling enables very efficient greedy reconstruction algorithms that recover the signal with computational complexity sublinear to the signal dimension.

Furthermore, if the source signals are sparse, then the sensors do not need to communicate the entire received signals to the processing center. Communication resources can be saved by transmitting only the significant frequency components of the sensed data and their locations on the frequency grid. If the signal is compressively sampled, then the CS reconstruction algorithms provide these components at their output. If a classical uniform Nyquist-rate sensor is used instead, then the sparse components can be identified using a very low-cost FFT operation. In either case $\|Z_{|\boldsymbol{\omega}} - \boldsymbol{\Psi}_{|\boldsymbol{\omega}}\boldsymbol{\theta}\|$ can be used instead of $\|Z - \boldsymbol{\Psi}\boldsymbol{\theta}\|$, where $\boldsymbol{\omega}$ is the frequency support of the signals and $_{|\boldsymbol{\omega}}$ selects the vector or matrix rows only in this frequency support.

6.2 Spatial Sparsity and Decentralized Processing

Decentralize for Robustness. The minimization of (21) with respect to the unknown sources $\boldsymbol{Y}$ requires us to collect the sensor network data $\boldsymbol{Z}$ at a central location, which is undesirable in some cases. To overcome the need for centralized processing, consider the following upper-bound to the objective function in (21):

$$
\begin{aligned}
\min_{\boldsymbol{Y}} \|\boldsymbol{Z} - \sum_k \boldsymbol{A}(\boldsymbol{x}_k)\boldsymbol{Y}_k\|^2 &= \min_{\boldsymbol{Y}} \sum_{q=1}^{Q} \|\boldsymbol{Z}_q - \sum_k \boldsymbol{A}_q(\boldsymbol{x}_k)\boldsymbol{Y}_k\|^2 \\
&\leq \min_{\forall q} \sum_{\forall i \backslash q} \left\| \boldsymbol{Z}_i - \sum_k \boldsymbol{A}_i(\boldsymbol{x}_k)\boldsymbol{A}_q^{\dagger}(\boldsymbol{x}_k)\boldsymbol{Z}_q \right\|^2 \\
&= \min_{\forall q} \left\| \boldsymbol{Z} - \widehat{\boldsymbol{\Psi}}_q \boldsymbol{\theta} \right\|^2
\end{aligned}
$$

(30)

where the i-th column of $\widehat{\boldsymbol{\Psi}}_q$ is defined by $\boldsymbol{A}(\boldsymbol{\theta}_i)\boldsymbol{A}_q^{\dagger}(\boldsymbol{\theta}_i)\boldsymbol{Z}_q$ for each grid point i. The upper-bound in (30) is obtained by (*i*) simply factoring the objective across the sensors, (*ii*) independently optimizing the individual factors at each sensor, and (*iii*) choosing the minimum objective across all the sensors. Since each factorization requires local data, the computation is distributed across all the sensors.

Given that we can calculate approximate source estimates individually at each sensor, it is also natural that we distribute the model order selection problem among the sensors (28). The key idea is that when we plug the local signal estimates to solve the model order selection (28), the new objective function with respect to the source locations is still a surrogate to the original problem:

$$
\widehat{K} \approx \arg\min_{K} \left\| \boldsymbol{Z} - \widehat{\boldsymbol{\Psi}}_q \widehat{\boldsymbol{\theta}}_K \right\|^2 - 2\sigma^2 \log p(K),
$$

(31)

The objective value of the optimization problem (31) provides a score for us to rank all the local solutions across the sensor network. Then, sensor network chooses the minimum score solution among all the sensors via (30).

Enter CS to reduce communication. Since we know the desired $\boldsymbol{\theta}$ is sparse, we can use a Gaussian random matrix $\boldsymbol{\Phi}$ for dimensionality reduction. Via RIP, which was discussed in Sect. 2, we have

$$
\frac{1}{1+\delta}\|\boldsymbol{\Phi}\boldsymbol{Z} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\theta}\|^2 \leq \|\boldsymbol{Z} - \boldsymbol{\Psi}\boldsymbol{\theta}\|^2 \leq \frac{1}{1-\delta}\|\boldsymbol{\Phi}\boldsymbol{Z} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\theta}\|^2.
$$

(32)

Required isometry is proportional $\mathcal{O}(K \log \frac{N}{K})$ to the desired spatial sparsity K of $\boldsymbol{\theta}$.

7. SUPPORT RECOVERY FROM QUANTIZED MEASUREMENTS

Transmission of the network data $\boldsymbol{Z}$ requires that the continuous-amplitude values be quantized to a certain precision. In this section we describe how quantization to 1-bit values can be effectively used to transmit the sensor data. A 1-bit quantization scheme was used in [7], using standard CS

reconstruction methods to reduce the communication bandwidth. The approach we propose here also uses 1-bit quantization but is modeled after the 1-bit CS theory [3].

The proposed algorithm eliminates received-signal-strength (RSS) information from the data and uses no communication bandwidth to transmit them. Instead it transmits only the more robust timing and phase information in the signal. This enables significantly more accurate localization even in far-field and bearing estimation configurations.

The 1-bit data also eliminate signal amplitude information, which is not necessary for localization using our algorithm. Thus it is not necessary to perform accurate sensor calibration to have a common amplitude reference. Furthermore, no communication resources are used to transmit unnecessary information.

7.1 1-bit Quantization

The 1-bit data transmitted though the network is the sign of the CS measurements, henceforth denoted using ζ:

$$
\zeta \equiv \text{sign}(\boldsymbol{\Phi}\boldsymbol{Z}) = \text{sign}(\boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\theta})
$$

(33)

where $\text{sign}(\boldsymbol{x})$ is a vector of $+1$ or -1 if the corresponding element of $\boldsymbol{x}$ is positive or negative, respectively. This vector, however, only indicates the sign of each measurement. Directly using it in the optimization (29) as a substitute of $\boldsymbol{Z}$ would result to suboptimal solutions. Instead, the localization algorithm should only ensure that the recovered location information $\boldsymbol{\theta}$ is consistent with the measurements, assuming no measurement noise. Furthermore, the location information $\boldsymbol{\theta}$ is a positive quantity, which should also be enforced as a constraint in the reconstruction algorithm.

7.2 1-bit Localization Algorithm

The 1-bit information eliminates the amplitude information of the signal since $\text{sign}(\boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\theta}) = \text{sign}(c\boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{\theta})$ for any positive constant c. Thus, determining the sparse location data by minimizing the ℓ_1 norm consistent with the signs would result to a zero signal. An additional reconstruction constraint is necessary for to recover the location of the sources. Since the ℓ_1 norm of the signal is used in (29) as a proxy for the source sparsity, [3] constrains the ℓ_2 norm of the signal such that $\|\boldsymbol{\theta}\|_2 = 1$. Thus, the location information is recovered by solving the following optimization:

$$
\widehat{\boldsymbol{\theta}} = \arg\min_{\widehat{\boldsymbol{\theta}}} \|\widehat{\boldsymbol{\theta}}\|_1, \text{ s.t. } \text{sign}(\boldsymbol{\Phi}\boldsymbol{\Psi}\widehat{\boldsymbol{\theta}}) = \zeta, \ \text{sign}(\widehat{\boldsymbol{\theta}}) = +1 \text{ and } \|\widehat{\boldsymbol{\theta}}\|_2 = 1.
$$

The resulting location vector can subsequently be scaled to have the desired properties.

Imposing sign measurements as hard constraints in the reconstruction has the potential to generate infeasible problems in the presence of measurement noise or errors in the transmission. However, under the assumption of Gaussian measurement noise, it can be shown that the hard constraints can be softened and approximated by a one-sided quadratic

function:

$$f(x) = \begin{cases} 0, & x \geq 0 \\ x^2, & x < 0. \end{cases} \tag{34}$$

Thus, the optimization in (34) can be relaxed to:

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\widehat{\boldsymbol{\theta}}} \|\widehat{\boldsymbol{\theta}}\|_1 + \lambda_1 \sum_l f(\zeta_l(\boldsymbol{\Phi\Psi}\widehat{\boldsymbol{\theta}})_l) + \lambda_2 \sum_k f((\widehat{\boldsymbol{\theta}})_k) \text{ s.t. } \|\widehat{\boldsymbol{\theta}}\|_2 = 1, \tag{35}$$

where λ_1 and λ_2 are the relaxation parameters. The optimization in (35) can be efficiently computed using the algorithm in [3]. Under this relaxation, the optimization in (35) is the 1-bit equivalent of the optimization in (29) and solves essentially the same problem when the data is quantized to 1-bit sign information.

8. SPECIAL CASE: BEARING ESTIMATION

The Bayesian localization framework was derived assuming we localize sources on a 2D-grid. The optimization problems and the discussed solution algorithms exploited the spatial sparsity of the sources on the location grid. This framework is general enough that it includes localizing targets in 1D-grids as a special case, e.g., the bearing estimation of sources using a network of sensors, which is the focus of this section.

Note that in localization problems, it is customary to assume that (*i*) the propagation medium is isotropic and (*ii*) the sources are isotropic point sources. In reality, these two assumptions are somewhat idealistic: we almost always have a non-isotropic medium, and most sources are directional. Hence, the source propagation can be assumed to be uniform only within a small cone, as illustrated in Fig. 2. When the data from all of the sensors is fused for localization, the directional nature of the signal propagation discrepancies might cause estimation errors in localization. This observation motivates a specialized localization algorithm for a collection of nearby sensors (e.g., Mica nodes [14]) when a sensor management system can self-organize the sensors to initiate bearing estimation.

8.1 Far-field localization

We assume that there are K targets that are sufficiently far away from the sensors so that they can be considered on a ring of radius R_2 concentric with the sensor disk (e.g., see Fig. 3). Hence, to localize these targets, it suffices to estimate their bearing with respect to the sensors. We assume that the sources are restricted to N equally-spaced grid points on the circle, where angles are measured with respect to the horizontal axis.

Without loss of generality, the K targets transmit at one frequency ω with a corresponding wavelength $\lambda = c/\omega$. When the targets transmit at different frequencies, it presents a simpler problems since we can then solve for the bearings at each frequency separately. For analysis, we assume that Q sensors are placed uniformly at random within a concentric
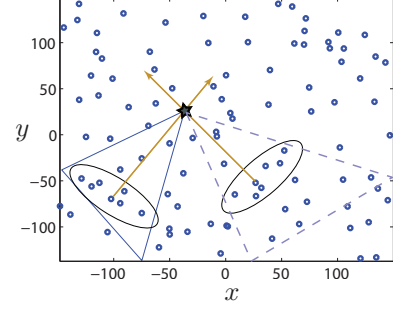

Figure 2: Bearing estimation example: The source location is marked with a star, and the sensor locations are shown with circles. The sensors within the solid and dashed triangles experience similar propagation for their observed signals. The bearing of the source can be estimated from these sensors using the Bayesian bearing estimation framework.
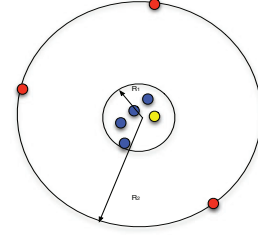

Figure 3: Far-field scenario in which targets (red) are placed arbitrarily on a ring at some large distance R_2 from a field of sensors (blue) which are placed uniformly at random within a disk of radius R_1. The yellow sensor is the query sensor.

sensor disk of radius R_1 with polar coordinates (r_p, ψ_p). The Q sensors send their lists to a central processing unit which builds a dictionary $\boldsymbol{\Psi}$ and then runs the Bearing Pursuit algorithm of Fig. 4. We assume that the central unit knows the locations of the the other P sensors so that it can use this information to build the dictionary. See

Next, we describe the dictionary matrix $\boldsymbol{\Psi}$ which we need to form to determine the bearings. With the single frequency assumption, the dictionary matrix $\boldsymbol{\Psi}$ can be analytically built. In fact, due to geometry, the (p, j) entry in $\boldsymbol{\Psi}$ is simply

$$\boldsymbol{\Psi}_{p,j} = e^{2\pi i r_p/\lambda \cos(\psi_p - \theta_j)}.$$

For any p, j, note that we have $|\boldsymbol{\Psi}_{p,j}| = 1$. Then, the bearing estimation can be obtained via

$$\boldsymbol{\Psi}\theta = y. \tag{36}$$

We note that the matrix $\boldsymbol{\Psi}$ is *not* the result of applying a Johnson-Lindenstrauss (JL) matrix to each sensor's observations. The dictionary does, however, have both sufficient ran-

```
            Algorithm:  Bearing Pursuit

Inputs:  Number K of sources, dictionary Ψ,
and y vector of measurements
Output:  List L of O(K) locations

T = O(K) // size of list maintained
r = 0 // current representation
For each iteration j = 0,1,...,O(log K) {
    form vector z = Ψ*y,
    retain top T entries in z,
    update representation r = r + z,
    prune r to maintain list of size T,
    update measurements y = y − Ψr.   }
```

Figure 4: Pseudocode for Bearing Pursuit Algorithm.

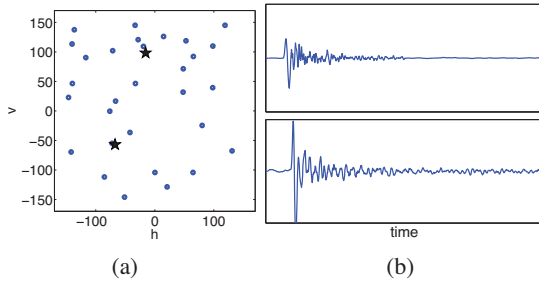(a)   (b)

Figure 5: (a) Sensor network simulation topology for 2-D source localization. (b) Source signals used in the simulations.

domness and sufficient structure which Bearing pursuit can exploit to solve Equation (36) for the non-zero entries in θ. Appendix 10 proves that Bearing pursuit correctly identifies the bearing of the sources.

9. EXPERIMENTS
9.1 Near-Optimal Spatial Localization

To demonstrate the localization framework and the optimization algorithms, we simulate a sensor network where we use $Q = 30$ sensors, randomly deployed on a 150×150 m^2 deployment area, to localize two targets. In Fig. 5(a), the target locations are shown with stars whereas the sensor locations are shown with circles. In this experiment, our sources are actual recoded projectile shots, as shown in Fig. 5(b). The power of the first source (top) is approximately one third of the power of the second source (bottom).

We simulate the decentralized message passing scheme as discussed in Sect. 6.2. We compress the dimensions of the observed signals 50:1 from their Nyquist rate using Gaussian random matrices for transmission (2% compression) at each sensor. Given the compressive measurements from other sensors in the network, each sensor proceeds to solve (31) locally. Finally, each solution along with the objective score is passed across the sensor network. Only the solution with the minimum score among all the sensors is kept during transmission.

(a)   (b)
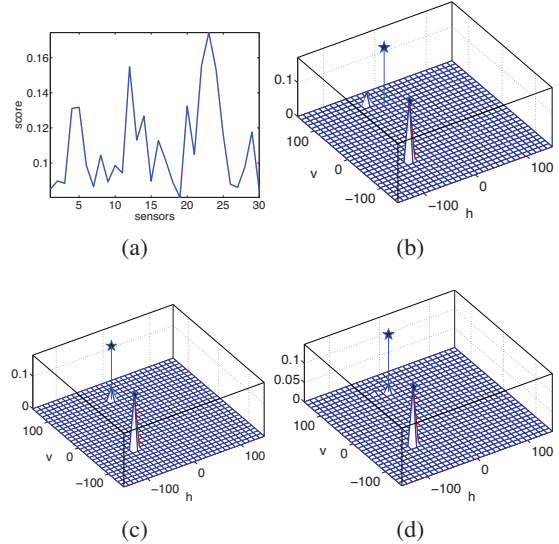
(c)   (d)

Figure 6: (a) Estimated local scores for the localization solutions. (b) The localization solution corresponding to the best score ($q = 19$). The true target locations are marked with stars. (c) The localization solution corresponding to sensor #27. (d) The mean of all the localization vectors.

Figure 6 summarizes the estimation results. Note that the heights of the peaks are approximately proportional to the source powers. In Fig. 6(a), the locally computed data score values are shown. The scores vary because the dictionaries are built using the observed signals themselves, which include both sources. In 6(b), we illustrate the localization result for the sensor with the best local score. Even in the presence of additive noise in the observed signals and the high amount of compression, the resulting location estimates are satisfactory. In 6(c), we randomly selected sensor # 19 and plotted its localization output. Given the ground truth, shown with stars in the plot, the localization output of sensor # 19 is much better than the sensor with the best local score. However, with Monte Carlo run, we expect data fusion scheme to perform better on the average. For completeness, we show the average of all the localization outputs from the sensor network.

Finally, Fig. 7 summarizes the localization results by solving the optimization problem (35). In contrast to the results in Fig. 6, the results in Fig. 7 only use the sign measurements of the compressive samples. Hence, the compression is 800:1. Unfortunately, we do not have a score function for the 1-bit support recovery results derive from the Bayesian framework. Therefore, we heuristically use the mean of all the sensor estimates, as shown in Fig. 7(a). The two targets are in the solution as expected along with some spurious peaks due to noise and the drastic compression rates. Figures 7(b) and (c) show the localization results from two
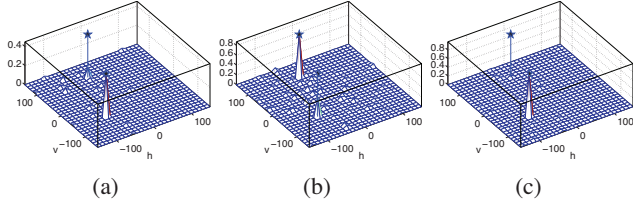
Figure 7: 1**-bit estimation results. (a) Average of the sparse solutions across the sensor network. (b) The localization vector of the sensor with the best score. (c) A randomly selected sensor output. The true target locations are marked with stars.**

random sensors in the sensor network. Note that the less powerful target is missed in Fig. 7(c).

9.2 Bearing Estimation

In this section, we demonstrate the bearing estimation performance of the proposed algorithms in Secs. 7 and 8. Our focus is to demonstrate the potential reductions in communication cost as well as computational efficiency in obtaining bearing estimates in a wireless setting rather than argue regarding sampling efficiency. For this experiment, we have collected acoustic vehicle data for a convoy of five vehicles traveling on an oval track. Since the spectral support of the vehicle signals usually lie in the range of 0–500Hz, existing sampling hardware suffices for this task. Therefore, we collected data using 10 sensors, which uniformly sampled data at a sampling rate of 4410Hz. The network reports its bearing estimates at twice per second. Hence, the number of signal samples per bearing estimate is 2205.

In the experiment, after the sensors collect Nyquist samples, they create local dictionaries as described in 6.2 and calculate the random projections of their data with a pre-stored Gaussian random matrix of dimensions 100×2205, which is different at each sensor. At each sensor, we create a uniform grid in the bearing domain $[0, \pi)$ using $N = 180$ discrete locations.

Figures 8(a)–(d) summarize the results of four different approaches. In Fig. 8(a) and (b), the sensors record the zero crossing information of each measurement as $0/1$ bits, where 0 corresponds to a negative signal sample and 1 corresponds to a positive signal sample. Then, to determine what each bit corresponds to as a signal value, each sensor calculates the absolute value of their randomly projected measurements and estimate their mean, denoted by μ. The resulting number and its negative corresponds to what the bits 1 and 0 encode, respectively.

In the intersensor transmissions, sensors transmit the 1-bit information of 100 compressive samples as well as the single number μ, which can be quantized up to the desired level of accuracy. Note that this transmission bandwidth, which is on the order of hundred bits, is significantly smaller than what would take transmit the observed signal itself,
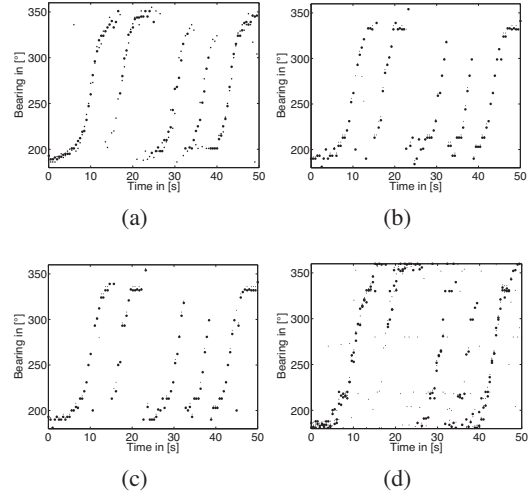

Figure 8: (a) Baseline bearing tracks. (b) Bearing pursuit results with the 1**-bit messaging scheme described in Sect.7. (c) Bearing pursuit results when the compressive samples of the source signals are used. (d) Result of** 1**-bit CS optimization problem** (35)**.**

which is 2205-dimensional even if it is compressed. As an example, with 10:1 compression and 8-bit quantization on the signal values, we would need at least $10\times$ the communication to transmit the observed signals. Since the signal messages from multiple sensors need to be accumulated across the sensors, this may create bottlenecks across the network. In contrast, the 1-bit intersensor messages require a communication bandwidth that is on the order of bandwidths typically used by conventional RSS localization algorithms.

The result in Fig. 8(a) has been previously reported in [7] and serves as a baseline for the comparison since it uses computationally costly Dantzig selector for recovery [4]. In Fig. 8(b), we use our bearing pursuit algorithm, which has provable guarantees discussed in the appendix. Compared to the Dantzig selector based approach, the tracks of the bearing pursuit have a small loss of accuracy. However, the upshot of our approach is that it can be easily implemented in simple sensor hardware, since it only requires the iteration of a single matrix multiplication and two sorting operations. Moreover, the number of iterations it requires to converge is on the order of source sparsity. Note that the estimation performance of the bearing pursuit algorithm improves only slightly when the compressive measurements are transmitted directly without compression, as shown in Fig. 8(c).

Finally, Fig. 8(d) illustrates the bearing tracks as a result of solving the optimization problem (35). Note that for this optimization problem, the encoding value for the 1-bit measurements μ is not needed. In this experiment, it is somewhat surprising to see that only the phase of the randomly projected measurements is sufficient to obtain the bearing

tracks. This optimization based approach is useful in scenarios when the sensors operate in clusters, where the cluster head can build the source dictionary and the other sensors can directly sample the compressive measurements of the observations. Since only zero crossing information needs to be transmitted, the sensor hardware can be simplified.

10. CONCLUSIONS

In this paper, we have developed a Bayesian formulation of the localization problem and posed it as a sparse recovery problem. Our approach allows us to exploit sparsity in several aspects of the network design: Signal sparsity, when available, allows very efficient sensing and communication of the source signals. Spatial sparsity allows decentralized computation of the source locations and further reduction in the communication cost, even if the source signals themselves are not sparse. It further allows the use of very efficient 1-bit quantization and reconstruction methods that only transmit timing information relevant for localization. In our setting, the randomized compressive measurements that are transmitted between sensor nodes act like fountain codes: As long as "enough" measurements arrive at the receiver, we can recover the required information about the signal. This makes the measurements robust to packet drops. The measurements are also progressive in the sense that each receiver can choose to receive measurements until they can recover to within a desired tolerance. In the special case of bearing estimation, the combination of sparsity and the incoherence of the bearing problem also allows us to provide solid theoretical guarantees on the performance of our algorithms. Our experimental results with synthetic and field data verify and validate our approach.

11. REFERENCES

[1] R. G. Baraniuk. Compressive Sensing. *IEEE Signal Processing Magazine*, 24(4):118–121, 2007.

[2] C. M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[3] P. Boufounos and R. G. Baraniuk. One-Bit Compressive Sensing. In *Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, Mar 2008.

[4] E. Candes and T. Tao. The Dantzig selector: statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, 2007.

[5] E. J. Candès. Compressive sampling. In *Proc. International Congress of Mathematicians*, volume 3, pages 1433–1452, Madrid, Spain, 2006.

[6] V. Cevher, M. Duarte, and R. G. Baraniuk. Distributed Target Localization via Spatial Sparsity. In *European Signal Processing Conference (EUSIPCO)*, Lausanne, Switzerland, Aug 2008.

[7] V. Cevher, A. C. Gurbuz, J. H. McClellan, and R. Chellappa. Compressive wireless arrays for bearing estimation. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, number 2497–2500, Las Vegas, NV, Apr 2008.

[8] J. Chen, L. Yip, J. Elson, H. Wang, D. Maniezzo, R. Hudson, K. Yao, and D. Estrin. Coherent acoustic array processing and localization on wireless sensor networks. *Proceedings of the IEEE*, 91(8):1154–1162, 2003.

[9] D. L. Donoho. Compressed Sensing. *IEEE Trans. on Information Theory*, 52(4):1289–1306, 2006.

[10] A. Gilbert, M. Strauss, and J. Tropp. A tutorial on fast fourier sampling. *IEEE Signal Processing Magazine*, 25(2):57–66, March 2008.

[11] I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *IEEE Trans. Signal Processing*, 45(3):600–616, 1997.

[12] C. Guestrin, P. Bodi, R. Thibau, M. Paski, and S. Madden. Distributed regression: an efficient framework for modeling sensor network data. In *Proc. of the Third International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 1–10. ACM Press New York, NY, USA, 2004.

[13] A. C. Gurbuz, V. Cevher, and J. H. McClellan. A compressive beamformer. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, Nevada, Mar 30 –Apr 4 2008.

[14] J. Hill and D. Culler. Mica: a wireless platform for deeply embedded networks. *Micro, IEEE*, 22(6):12–24, Nov/Dec 2002.

[15] A. T. Ihler. *Inference in Sensor Networks: Graphical Models and Particle Methods*. PhD thesis, Massachusetts Institute of Technology, 2005.

[16] J. Laska, S. Kirolos, Y. Massoud, R. Baraniuk, A. Gilbert, M. Iwen, and M. Strauss. Random sampling for analog-to-information conversion of wideband signals. In *IEEE Dallas Circuits and Systems Workshop, Dallas, TX*, 2006.

[17] J. N. Laska, S. Kirolos, M. F. Duarte, T. Ragheb, R. G. Baraniuk, and Y. Massoud. Theory and implementation of an analog-to-information conversion using random demodulation. In *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS)*, New Orleans, LA, May 2007. To appear.

[18] D. Malioutov, M. Cetin, and A. S. Willsky. A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE Trans. Signal Processing*, 53(8):3010–3022, 2005.

[19] D. Model and M. Zibulevsky. Signal reconstruction in sensor arrays using sparse representations. *Signal Processing*, 86(3):624–638, 2006.

[20] M. Rabbat and R. Nowak. Distributed optimization in sensor networks. In *Proc. 3rd International Workshop on Inf. Processing in Sensor Networks (IPSN)*, pages 20–27. ACM Press New York, NY, USA, 2004.

[21] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based countersniper system. In *Proc. of the 2nd international conference on Embedded networked sensor systems*, pages 1–12. ACM New York, NY, USA, 2004.

[22] E. M. Stein. *Harmonic analysis real-variable methods, orthogonality, and oscillatory integrals*. Princeton University Press, Princeton, NJ, 1993.

[23] J. Tropp and A. C. Gilbert. Signal recovery from partial information via orthogonal matching pursuit. *IEEE Trans. Info. Theory*, 53(12):4655–4666, Dec. 2007.

[24] J. Tropp, D. Needell, and R. Vershynin. Iterative signal recovery from incomplete and inaccurate measurements. In *Information Theory and Applications*, San Diego, CA, Jan. 27–Feb. 1 2008.

Appendix: Correctness of Bearing Pursuit

This appendix proves that the Bearing Pursuit algorithm correctly identifies the bearing of the sources.

LEMMA 1. *With $R_1 > K^2 N \lambda$ and $R_2 > K N R_1$ we have, for any $j \neq j'$,*

$$\left| \mathbb{E}_p(\boldsymbol{\Psi}_{p,j} \bar{\boldsymbol{\Psi}}_{p,j'}) \right| \leq \mathcal{O}\left(\frac{1}{K}\right). \tag{37}$$

PROOF. Let us fix a circle with radius $R_1/2 < r \leq R_1$ around which sensors are placed at uniformly random angles ϕ. Because of the scaling of R_1, we have $r = a \lambda K^2 N$. With

a fixed radius r, we have

$$\left|\mathbb{E}_p(\boldsymbol{\Psi}_{p,j}\bar{\boldsymbol{\Psi}}_{p,j'})\right| = \left|\frac{1}{2\pi}\int_0^{2\pi} e^{2\pi i a K^2 N[\cos(\phi-\theta_j)-\cos(\phi-\theta_{j'})]}\,d\phi\right|.$$

Using basic trigonometric identities, we note that

$$\cos(\phi-\theta_j)-\cos(\phi-\theta_{j'}) = C_1\cos(C_2-\phi)$$

where $C_1 = 2\sin(\frac{\theta_j-\theta_{j'}}{2})$ and $C_2 = \frac{\phi}{2}+\frac{\theta_j+\theta_{j'}}{2}$. Because the target bearings are separated by at least $2\pi/N$ (the hardest case), the constant C_1 is at least $4\pi/N$. Then we can simplify the above expression and obtain

$$\left|\mathbb{E}_p(\boldsymbol{\Psi}_{p,j}\bar{\boldsymbol{\Psi}}_{p,j'})\right| \le \left|\frac{1}{2\pi}\int_0^{2\pi} e^{2\pi i C K^2\cos(C_2-\phi)}\right|,$$

where the constant C is independent of the other parameters. Finally, we observe that this integral satisfies the hypotheses of Proposition 2, Chapter 8 (Oscillatory integrals of the first kind) in [22] and apply the standard method of stationary phase to bound it. We arrive at

$$\left|\frac{1}{2\pi}\int_0^{2\pi} e^{2\pi i C K^2\cos(C_2-\phi)}\right| \le \mathcal{O}\left(\frac{1}{\sqrt{K^2}}\right) = \mathcal{O}\left(\frac{1}{K}\right).$$

$\square$

We claim that Equation 37 is a sufficient condition on $\boldsymbol{\Psi}$ to recover the bearings of the sensors; i.e., to determine the locations of the K non-zero entries in θ.

THEOREM 1. *If $\boldsymbol{\Psi}$ has $\mathcal{O}(K)$ independent rows (i.e., if we place $\mathcal{O}(K)$ sensors uniformly at random in the sensor disk), then each estimate of the form*

$$\widetilde{\theta}_j = (\boldsymbol{\Psi}^*\boldsymbol{\Psi}\theta)_j$$

satisfies

$$\mathbb{E}(\widetilde{\theta}_j) = \theta_j \pm \frac{1}{K}\|\theta\|_1 \tag{38}$$

$$\mathrm{Var}(\widetilde{\theta}_j) \le \frac{1}{K}\|\theta\|_2^2. \tag{39}$$

PROOF. First, we check the expected value of the estimator $\widetilde{\theta}_j$ which corresponds to one sensor.

$$\mathbb{E}(\widetilde{\theta}_j) = \mathbb{E}_p\Big(\boldsymbol{\Psi}_{p,j}^*\sum_{\ell=1}^N \boldsymbol{\Psi}_{p,\ell}\theta_\ell\Big)$$

$$= \mathbb{E}_p\Big(\theta_j + \sum_{\ell\neq j}\boldsymbol{\Psi}_{p,j}^*\boldsymbol{\Psi}_{p,\ell}\theta_\ell\Big)$$

$$= \theta_j + \sum_{\ell\neq j}\mathbb{E}_p(\boldsymbol{\Psi}_{p,j}^*\boldsymbol{\Psi}_{p,\ell})\theta_\ell$$

$$\le \theta_k \pm \frac{1}{K}\|\theta-\theta_j\|_1$$

$$\le \theta_k \pm \frac{1}{K}\|\theta\|_1.$$

If we average over K independent sensor estimators, we retain the same expected value bound. Furthermore, if θ is

1-sparse with support at position j, then $\widetilde{\theta}_j$ is approximately correct in expectation while, if there is no source at position j, then $\widetilde{\theta}_j = (\|\theta-\theta_j\|_1)/k$ is small. Note that this estimate produces a separation: large values of the estimator give us the correct position and small values give us an incorrect position.

Let us check the second moment as well (which is an upper bound on the variance in Equation (39)).

$$\mathbb{E}(\widetilde{\theta}_j^2) = \mathbb{E}_p\Big(\boldsymbol{\Psi}_{p,j}^*\boldsymbol{\Psi}_{p,j}\sum_{\ell,\ell'}\boldsymbol{\Psi}_{p,\ell}^*\boldsymbol{\Psi}_{p,\ell'}\theta_\ell\theta_{\ell'}^*\Big)$$

$$= \sum_{\ell=1}^N |\theta_\ell|^2 + \sum_{\ell\neq\ell'}\theta_\ell\theta_{\ell'}^*\mathbb{E}_p(\boldsymbol{\Psi}_{p,\ell}^*\boldsymbol{\Psi}_{p,\ell'})$$

$$\le \|\theta\|_2^2 + K\|\theta\|_2^2\mathcal{O}\Big(\frac{1}{K}\Big)$$

$$\le \mathcal{O}(1)\|\theta\|_2^2.$$

Note that we use the AMGM to bound the product $\theta_\ell\theta_{\ell'}^*$ by the norm $\|\theta\|_2^2$ and that we have K such terms for a K-sparse position vector θ. So, we have a single instance of an estimator which produces an approximately correct answer and whose second moment we have bounded. If we repeatedly use this estimator for $\mathcal{O}(K)$ independent sensor positions (i.e., look at $\mathcal{O}(K)$ independent instances of the estimator), then we drive down the variance of $\widetilde{\theta}_j$ by the factor $1/K$ and we can estimate θ_j accurately for those positions j that satisfy, simultaneously, $\|\theta_j\| \ge 1/K\|\theta\|_1$ (from the bias in the expectaction) and $|\theta_j|^2 \ge 1/K\|\theta\|_2^2$ (from the variance bound). In particular, we can correctly recover the largest-magnitude θ_j in a K-sparse vector θ. By making R_1 and the number of sensors larger by the factor $\epsilon^{-\mathcal{O}(1)}$, we can recover positions with magnitude within the factor ϵ of largest, which makes the algorithm more robust.

Once we estimate accurately such positions, we add them to our current representation for θ, subtract their estimates from the current set of measurements, and iterate. We omit details. $\square$

Strictly speaking, the proof of Equation 37 restricted our sensors to an annulus with inner and outer radii approximately R_1 (i.e., we use sensors towards the outside of the inner disk). We note that if we place sensors uniformly at random within the inner disk, they will be concentrated on such an annulus and that we can disregard those towards the inside, possibly placing additional sensors at random to get $\mathcal{O}(K)$ in the outer annulus. This rejection sampling increases the number of sensors necessary by a constant factor and we simply include it in the factor $\mathcal{O}(K)$ without loss of generality. We can, therefore, conclude that the Bearing Pursuit algorithm finds the sensors.

COROLLARY 11.1. *With $\mathcal{O}(K)$ uniformly random sensors, Algorithm 4 correctly identifies the bearings of the sensors (i.e., it correctly determines the non-zero entries in θ).*