

Insense: Incoherent Sensor Selection for Sparse Signals

Amirali Aghazadeh, Mohammad Golbabaee, Andrew S. Lan, Richard G. Baraniuk Electrical and Computer Engineering Department, Rice University, Houston, TX 77005 USA

Abstract—Sensor selection, an important problem in the design of signal acquisition systems, refers to the problem of intelligently selecting a small subset of the available sensors to reduce the sensing cost while preserving signal acquisition performance. The majority of sensor selection algorithms find the subset of sensors that best recovers an arbitrary signal $x \in \mathbb{R}^N$ from a number of linear measurements $M > N$. In this paper, we develop a new sensor selection framework for *sparse* signals that finds the subset of sensors that best recovers such signals from a number of measurements $M \ll N$. Our proposed algorithm, Insense, minimizes a coherence-based cost function that is adapted from classical results in sparse recovery theory. Using a range of datasets, including two real-world datasets from microbial diagnostics and structural health monitoring, we demonstrate that Insense significantly outperforms conventional algorithms in the sparse regime.

Index Terms—Sensor selection, coherence, optimization, compressive sensing

I. INTRODUCTION

The accelerating demand for capturing signals at high resolution is driving acquisition systems to employ an increasingly large number of sensing units. However, factors like manufacturing costs, physical limitations, and energy constraints typically define a budget on the total number of sensors that can be implemented in a given system. This budget constraint motivates the design of *sensor selection* algorithms [1] that intelligently select a subset of sensors from a pool of available sensors in order to lower the sensing cost with only a small deterioration in acquisition performance.

In this paper, we build on the classical sensor selection setup, where D available sensors obtain linear measurements of a signal $x \in \mathbb{R}^N$ according to $y = \Phi x$, where $\Phi \in \mathbb{R}^{D \times N}$ with each row corresponding to one sensor. With this setup, the sensor selection problem is one of finding a subset Ω of sensors (i.e., rows of Φ) of size $|\Omega| = M$ such that the signal x can be recovered from its M linear measurements

$$y_\Omega = \Phi_\Omega x \quad (1)$$

with minimal reconstruction error. Here, $\Phi_\Omega \in \mathbb{R}^{M \times N}$ is called the *sensing matrix*; it contains the rows of Φ indexed by Ω .

The lion's share of current sensor selection algorithms [1]–[3] select sensors that best recover an arbitrary signal x from $M > N$ measurements. In this case, (1) is *overdetermined*. Given a subset of sensors Ω , the signal x is recovered simply

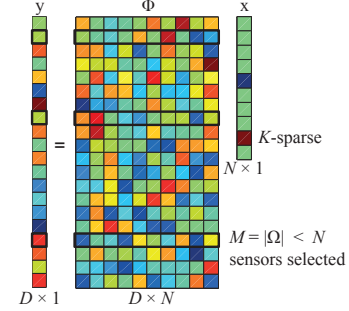


Fig. 1. Schematic of the sensor selection problem for sparse signals on synthetic data. Here, $M = 3$ sensors indexed by $\Omega = \{2, 8, 17\}$ are selected from $D = 20$ available sensors to recover a $K = 2$ -sparse vector $x \in \mathbb{R}^N$, $N = 10$ from the underdetermined linear system $y_\Omega = \Phi_\Omega x$.

by inverting the sensing matrix $\Phi_\Omega^\dagger y_\Omega$, where Φ_Ω^\dagger is the pseudoinverse of Φ_Ω .

Such approaches do not exploit the fact that many real-world signals are (near) *sparse* in some basis [4]. It is now well-known that (near) sparse signals can be accurately recovered from a number of linear measurements $M \ll N$ [5]–[7]. Conventional sensor selection algorithms are not designed to exploit such low-dimensional structure. Indeed, they typically fail to select the appropriate sensors for sparse signals in the *underdetermined* setting when $M < N$.

In this paper, we develop a new sensor selection framework that finds the optimal subset of sensors Ω that best recovers a (near) sparse signal x from $M < N$ linear measurements (see Figure 1). In contrast to the conventional sensor selection setting, here the sensing equation (1) is *underdetermined*, and it can not be simply inverted in closed form.

A key challenge in sensor selection in the underdetermined setting is that we must find a replacement for the cost function that has been so far useful in the classical, overdetermined setting. In the overdetermined setting, the estimation error $x - \hat{x}$ (or the covariance of the estimation error in presence of noise) is typically used as the cost function to find the most informative sensors. This can be obtained in closed form by inverting (1). In the underdetermined setting, however, the recovery of sparse vector x from y_Ω requires a computational scheme (iterative or greedy) [8], [16] and therefore no closed form expression exists. Fortunately, the sparse recovery theory tells us that one can reliably recover a sufficiently sparse vector x from its linear measurements y_Ω when the columns of the sensing matrix Φ_Ω are sufficiently *incoherent* [17]–[19]. Concretely, let $\mu_{ij}(\Phi_\Omega) = \frac{|\langle \phi_i, \phi_j \rangle|^2}{\|\phi_i\|^2 \|\phi_j\|^2}$ define the *coherence* between the columns ϕ_i and ϕ_j in the sensing matrix Φ_Ω .

If the values of $\mu_{ij}(\Phi_\Omega)$ for all pair of column pairs (i, j) are bounded by a certain threshold, then sparse recovery algorithms such as Basis Pursuit (BP) [10], [17], [20] can recover the sparse signal x exactly. This suggests a new cost function for sensor selection. In this paper we advocate minimizing the *average squared coherence* defined by

$$\mu_{\text{avg}}(\Phi_\Omega) = \frac{1}{\binom{N}{2}} \sum_{1 \leq i < j \leq N} \mu_{ij}(\Phi_\Omega) \quad (2)$$

in selecting sensors for sparse signals. The challenge of sensor selection problem for sparse signals is now to formulate an optimization algorithm that selects the subset of sensors with smallest average coherence using the exact definition outlined above.

A. Contributions

We make three distinct contributions in this work. First, we propose and study the sensor selection problem for sparse signals for the same time in the sensor selection literature. We demonstrate that the common cost functions used in sensor selection algorithms are not suitable to select the best subset of sensor to recover sparse signals.

Second, to address this problem, we develop a new sensor selection algorithm called *Incoherent sensor selection* (Insense). Insense uses an efficient optimization algorithm to find a subset of sensors with smallest average coherence among the columns of the selected sensing matrix Φ_Ω . As a subproblem, we develop an efficient optimization algorithm for Insense to project onto the convex set defined by an scaled-boxed simplex (SBS) constraint (see subsection IV-A) which can have independent interest in other problems.

Third, we demonstrate the superior performance of Insense over conventional sensor selection algorithms using an exhaustive set of experimental evaluations involving two diverse datasets from microbial diagnostics and structural health monitoring and six performance metrics (average mutual coherence, worst-case mutual coherence, sparse recovery performance, frame potential, condition number, running time).

We demonstrate that in the case of redundant, coherent, or structured dictionaries Φ , common in real-world applications, Insense finds the best subset of sensors in terms of sparse recovery performance with wide margins even when the conventional algorithms fail to run.

B. Paper organization

We first outline the state-of-the-art greedy and optimization-based sensor selection algorithms and expand on the implications of our proposed sensor selection problem to compressive sensing in Section II and then explain how the cost function they minimize is adapted to select sensors in the overdetermined sensing regime $M \geq N$. Next, we state our coherence minimization formulation to solve the sensor selection problem in Section III. Section IV details the Insense algorithm and Section V presents simulation results on synthetic and real-world datasets. We conclude in Section VI.

II. RELATED WORK

In this section we first describe the conventional sensor selection algorithms. More specifically, we talk about the cost functions they employ to select the most informative sensors. Next, we talk about the implications that our proposed sensor selection problem have to design sensing matrices for compressive sensing applications.

A. Conventional sensor selection algorithms

Existing sensor selection algorithms mainly study the sensor selection problem in the overdetermined regime i.e., when $M \geq N$ [1]–[3], [27]. In this regime, robust signal recovery can be obtained using the least square (LS) solution to the sensing model (1). As a result, these algorithms select a set of sensors that minimize the mean square error (MSE) [28]–[30] or a proxy of the MSE [31]–[33] of the corresponding LS solution.

For instance, Joshi, et al. [1] proposed a convex optimization-based algorithm that minimizes the log-volume of the *confidence ellipsoid* around the LS solution of x . Shamaiah et al. [2] developed a greedy algorithm that outperforms the aforementioned convex approach in terms of MSE. Ranieri et al. [3] attempted to minimize the Frame Potential (FP) of the selected matrix, defined as

$$\text{FP}(\Phi_\Omega) = \sum_{\forall (i,j) \in \Omega, i < j} |\langle \phi^i, \phi^j \rangle|^2, \quad (3)$$

where ϕ^i represents the i^{th} row of Φ . Their greedy algorithm, FrameSense, selects near optimal sensors in terms of MSE when $M \geq N$. Other sensor selection algorithms [34], [35] which assume a non-linear observation model for sensors, also operate in the overdetermined regime.

B. Connections to compressive sensing

Our model for sensor selection has important implications in designing sensing matrices for compressive sensing (CS) applications. Note that CS theory typically resorts to *random* sensing matrices. For instance, it has been shown that many ensembles of random matrices, such as partial Fourier, Bernoulli, and Gaussian matrices, result in sensing matrices with guaranteed sparse recovery [21], [22]. Recently, there have been efforts [36]–[40] in *designing* sensing matrices Φ that outperform the regular random matrices in the CS recovery tasks. For instance, Grassmannian matrices [37] attain the smallest possible mutual coherence.

However, in many practical applications it is extremely difficult to have sensing matrices that exactly match a random or Grassmannian matrix; the sensing matrix is dictated by physical constraints that are specific and unique to each application. In the microbial diagnostic approach of [23], for example, the entries of the sensing matrix Φ are determined by the hybridization affinity of random DNA probes to microbial genomes and do not necessarily follow any well-known random distribution. Similarly, in structural health monitoring systems [24]–[26], the sensing matrix is constrained by the solution to wave equations, which follows specific patterns

that are far from any random distribution. In order to design realizable sensing matrices in practice, a palette of sensors can be *selected* from a dictionary of all physically feasible sensors Φ to achieve good sparse recovery performance.

III. PROBLEM STATEMENT

Consider a set of D sensors taking nonadaptive, linear measurements of a K -sparse (i.e., with K non-zero elements) vector $x \in \mathbb{R}^N$ following the linear system $y = \Phi x$, where $\Phi \in \mathbb{R}^{D \times N}$ is a given sensing matrix. We aim to select a subset Ω of sensors with size $|\Omega| = M \ll D$, such that the sparse vector x can be recovered from its $M < N$ linear measurements $y_\Omega = \Phi_\Omega x$ with minimal reconstruction error. Here, Φ_Ω contains the rows of Φ indexed by Ω and y_Ω contains the entries of y indexed by Ω . This model for the sensor selection problem can be adapted to more general scenarios. For example, if the signal is sparse in a basis called Ψ , we simply consider $\Phi = \Theta\Psi$ as the new sensing matrix, where Θ is the original sensing matrix.

In order to find a subset Ω of sensors (rows of Φ) that best recovers a sparse signal x from y_Ω we aim to select a submatrix $\Phi_\Omega \in \mathbb{R}^{M \times N}$ that attains the lowest average squared coherence

$$\mu_{\text{avg}}(\Phi_\Omega) = \frac{1}{\binom{N}{2}} \sum_{1 \leq i < j \leq N} \frac{|\langle \phi_i, \phi_j \rangle|^2}{\|\phi_i\|^2 \|\phi_j\|^2}, \quad (4)$$

where ϕ_i denotes the i^{th} column of Φ_Ω . Note that other measure for coherence, for example max coherence $\mu_{\text{max}}(\Phi_\Omega) = \max_{i < j} \mu_{ij}$ can also be employed here by adding one more step in the optimization procedure. We however chose to work with average coherence due to its simplicity and comparable sensor selection performance to the max coherence.

We define a diagonal selector matrix $Z = \text{diag}(z)$ with $z = [z_1, z_2, z_3, \dots, z_D]^T$ and $z_i \in \{0, 1\}$ where, $z_i = 1$ indicates that the i^{th} sensor (row) in Φ is selected and $z_i = 0$ otherwise. Next, we reformulate the sensor selection as the following optimization problem

$$\min_{z \in \{0, 1\}^D} \sum_{1 \leq i < j \leq N} \frac{G_{ij}^2}{G_{ii} G_{jj}}, \quad \text{s.t.} \quad G = \Phi^T Z \Phi, \quad \mathbf{1}^T z = M,$$

where $\mathbf{1}$ is the all-one vector. This Boolean optimization problem is combinatorial, since one needs to search over $\binom{D}{M}$ combination of sensors to find the optimal set of sensors Ω . To overcome this issue, we relax the Boolean constraint on z_i and replace it with a box constraint $z_i \in [0, 1]$ to arrive at the following problem

$$\min_{z \in [0, 1]^D} \sum_{1 \leq i < j \leq N} \frac{G_{ij}^2}{G_{ii} G_{jj}}, \quad \text{s.t.} \quad G = \Phi^T Z \Phi, \quad \mathbf{1}^T z = M, \quad (5)$$

which enables us to use an efficient gradient–projection algorithm to find an approximate solution.

IV. THE INSENSE ALGORITHM

We now outline the steps that Insense takes to solve the problem (5). We slightly modify the objective of (5) as follows:

$$f_\epsilon(z) = \sum_{1 \leq i < j \leq N} \frac{G_{ij}^2 + \epsilon_1}{G_{ii} G_{jj} + \epsilon_2} \quad \text{where} \quad G = \Phi^T Z \Phi, \quad (6)$$

where small positive numerical constants $\epsilon_2 < \epsilon_1 \ll 1$ are introduced to make the objective function well defined and bounded over $z \in [0, 1]^D$.

The optimization problem (6) has a smooth and differentiable, albeit non-convex objective function with linear and box constraints on z . We solve this problem with an iterative gradient-projection algorithm (outlined in Alg. (1)). For the objective function $f(z)$, the gradient $\nabla_z f \in \mathbb{R}^M$ is computed using the multidimensional chain rule of derivatives [41] as

$$(\nabla_z f)_i = (\Phi \nabla_G f \Phi^T)_{ii} \quad \text{for} \quad i = 1, \dots, M.$$

The $N \times N$ upper triangular matrix $\nabla_G f$ is the gradient of f in terms of the (auxiliary) variable G at the point $G = \Phi^T Z \Phi$, given by

$$(\nabla_G f)_{ij} = \begin{cases} \frac{2G_{ij}}{G_{ii} G_{jj} + \epsilon_2}, & i < j \\ \sum_{\forall \ell \neq i} \frac{G_{\ell\ell} (G_{i\ell}^2 + \epsilon_1)}{(G_{ii} G_{\ell\ell} + \epsilon_2)^2}, & i = j \\ 0, & \text{elsewhere.} \end{cases} \quad (7)$$

The Insense algorithm proceeds as follows (see algorithm 1). First, the optimization variables G and Z are initialized. Next, Insense performs the following update in iteration k ,

$$z^{k+1} = P_{\text{SBS}}(z^k - \gamma^k \nabla_z f(z^k)), \quad (8)$$

where P_{SBS} denotes the projection onto the convex set defined by the scaled boxed-simplex (SBS) constraints $\mathbf{1}^T z = D$ and $z \in [0, 1]^D$. For certain bounded step size rules (e.g., $\gamma^k \leq 1/L$ where L is the Lipchitz constant of $\nabla_z f$), the sequence $\{z^k\}$ generated by (8) converges to a critical point of the non-convex problem [42]. In our experiments, we use backtracking line search to determine γ^k in each step.

Algorithm 1: Insense

Input: Φ

Output: $Z = \text{diag}(z)$

Initialization:

$z \leftarrow z_0$;

$G \leftarrow \Phi^T Z \Phi$;

while *stoppage criterion* = *false* **do**

1. $k \leftarrow k + 1$;

2. update $\nabla_z f(z^k)$ based on equation (7);

3. $\gamma_k \leftarrow \text{line search}(f, \nabla_z f(z^k), z^k)$;

4. $z^k \leftarrow z^k - \gamma^k \nabla_z f(z^k)$ {gradient step};

5. $z^{k+1} \leftarrow P_{\text{SBS}}(z^k)$ {SBS projection step};

end

A. The SBS projection

We now detail our approach to solve the SBS projection problem

$$\begin{aligned} & \underset{z}{\text{minimize}} \quad \frac{1}{2} \|z - y\|_2^2, \\ & \text{subject to} \quad \sum_i z_i = M, \quad \text{and} \quad z_i \in [0, 1], \forall i. \end{aligned} \quad (9)$$

We emphasize that, for $M > 1$, the SBS projection problem is significantly different from the (scaled-)simplex constraint ($\sum_i z_i = M$) projection problem that has been studied in the literature [43]–[45], due to the additional box constraints $z_i \in [0, 1]$.

The Lagrangian of the problem (9) can be written as

$$\begin{aligned} f(z, \lambda, \alpha, \beta) = & \frac{1}{2} \|z - y\|_2^2 \\ & + \lambda (\sum_i z_i - M) + \sum_i \alpha_i (-z_i) + \sum_i \beta_i (z_i - 1), \end{aligned}$$

where λ, α, β are Lagrange multipliers for the equality and inequality constraints, respectively. The Karush-Kuhn-Tucker (KKT) condition is given by

$$\begin{aligned} z_i - y_i + \lambda - \alpha_i + \beta_i &= 0, \forall i, \\ \sum_i z_i - M &= 0, \\ \alpha_i (-z_i) &= 0, \beta_i (z_i - 1) = 0, \alpha_i, \beta_i \geq 0, 0 \leq z_i \leq 1, \forall i. \end{aligned}$$

According to the complementary slackness condition for the box constraint $z_i \in [0, 1]$, we have the following three cases for x_i :

- (1) $z_i = 0$: $\beta_i = 0, \alpha_i = y_i + \lambda > 0$,
- (2) $z_i = 1$: $\alpha_i = 0, \beta_i = 1 - y_i - \lambda > 0$,
- (3) $z_i \in (0, 1)$: $\alpha_i = \beta_i = 0, z_i = y_i + \lambda$.

Therefore, the key to solve the proximal problem (9) is to find the value of λ . However, this is not an easy task since we do not know which entries in z would fall on the boundary of the box constraint (and are equal to either 0 or 1).

In order to find the entries z_i that are equal to 0, we assume without loss of generality that the values of y are sorted in ascending order, $y_1 \leq y_2 \leq \dots \leq y_D$. We note that for all three cases above, we have $z_i = \max(\min(y_i + \lambda, 1), 0)$. Therefore, $\sum_i z_i$ is a non-decreasing function of λ . We evaluate $\sum_i z_i$ at the following values of λ :

- $\lambda = -y_1$: $z_1 = 0, z_i = \min(y_i - y_1, 1)$ for $i \geq 2$,
- $\lambda = -y_2$: $z_1 = z_2 = 0, z_i = \min(y_i - y_1, 1)$ for $i \geq 3$,
- \dots
- $\lambda = -y_D$: $z_1 = z_2 = \dots = z_D = 0$.

Thus, the entries in z that are equal to 0 correspond to the first K_0 entries in y where K_0 is the largest integer K such that $\sum_i \max(\min(y_i - y_K, 1), 0) \geq M$.

Similarly, we can find the entries in z that are equal to 1 by negating z and y in (9). Let $p = -y$ and assume that its entries are sorted in ascending order, a similar procedure shows that the entries in z that are equal to 1 correspond to the first K_1 entries in p where K_0 is the largest integer K such that $\sum_i \max(\min(p_i - p_K - 1, 0), -1) \geq -M$.

Knowing which entries in z are equal to 0 and 1 respectively, we can solve for the value of λ by looking at the entries with values in $(0, 1)$. Using case (3) above and denoting the index set of these entries by ζ , we have

$$\lambda = \frac{M - K_1 - \sum_{i \in \zeta} y_i}{|\zeta|},$$

and the solution to (9) is given by $z_i = \max(\min(y_i + \lambda, 1), 0)$.

V. EXPERIMENTS

In this section, we experimentally validate the Insense algorithm using a range of synthetic and real-world datasets. In all experiments, we set $\epsilon_1 = 10^{-9}$ and $\epsilon_2 = 10^{-10}$. We initialize Insense with $z_0 = \mathbf{1}/D$ and terminate when the relative change of the cost function $\mu_{\text{avg}}^2(\Phi_\Omega)$ drops below 10^{-7} .

A. Baselines and performance metrics

We compare Insense with several sensor selection algorithms including *Convex Sensor Selection* [1], *Greedy Sensor Selection* [2], *EigenMaps* [27], *FrameSense* [3], and selecting sensors at random. We also compare our algorithm with four greedy sensor selection algorithms which were also used in [3] as baselines. The first three of these greedy algorithms minimize three different information theoretic measures as a proxy to MSE: the determinant, mutual information (MI), and entropy of the selected sensing matrix. The final greedy algorithm directly minimizes the MSE of the LS reconstruction error. We dub these greedy algorithm by the objectives they minimize, i.e., *Determinant-G* [31], *MI-G* [32], *Entropy-G* [33], and *MSE-G* [28]–[30], respectively.

We compare the sensor selection algorithms using the following six metrics:

- The *Average coherence* $\mu_{\text{avg}}(\Phi_\Omega)$.
- The *Maximum coherence* $\mu_{\text{max}}(\Phi_\Omega)$.
- The *Frame potential* $\text{FP}(\Phi_\Omega)$.
- The *Condition number* $\text{CN}(\Phi_\Omega)$.
- The *BP recovery accuracy*.
- *Running time*.

To compute the BP recovery accuracy we run the BP algorithm [9] for multiple trials, where in each trial we generate a K -sparse vector x (with equal loadings on its support) and use BP to recover x from linear, nonadaptive (noiseless) measurements $y = \Phi_\Omega x$. We repeat the same experiment for all $\binom{N}{K}$ sparse vectors x with different support sets and report the percentage of trials x has been exactly recovered. For cases where $\binom{N}{K} > 10,000$, we run the BP algorithm on a smaller random subset of all $\binom{N}{K}$ sparse vectors x . Depending on the task, in some experiments only a portion of the metrics are illustrated.

B. Unstructured datasets

We first tested the sensor section algorithms by applying them on random matrices that do not have a special structure; asymptotically ($N \rightarrow \infty$) random matrices do not that favors certain sensors (rows) to others. We generate three types of

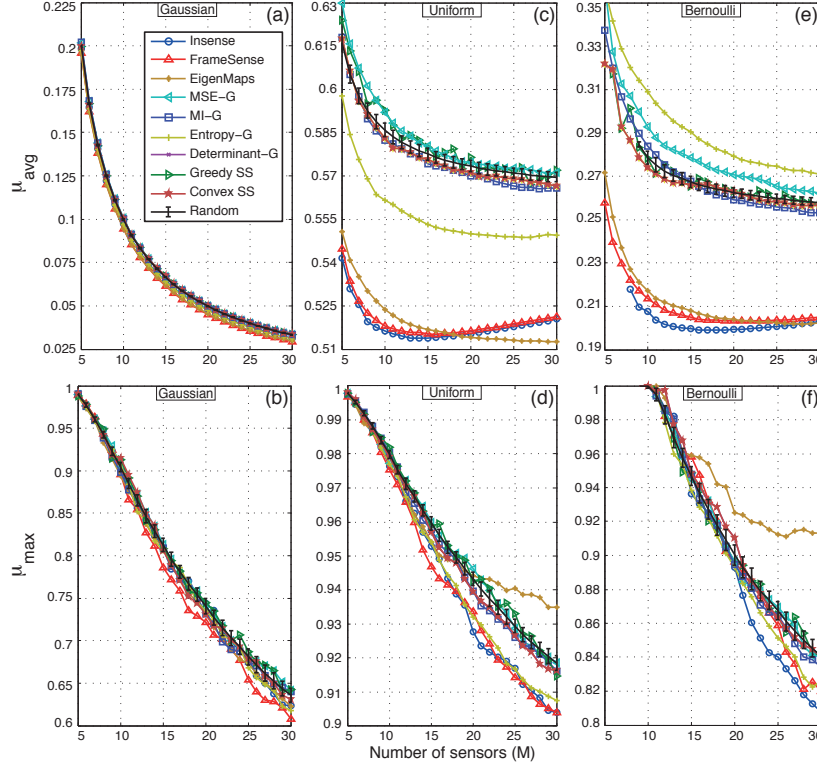


Fig. 2. Comparison of Insense against several baseline algorithms in minimizing the average coherence μ_{avg} and maximum coherence μ_{max} of the selected sensing matrix Φ_{Ω} from random sensing matrices with independent Gaussian (a,b), Uniform (c,d), and Bernoulli (e,f) entries ($D = N = 100$). Results are averaged over 20 trials with different randomly drawn matrices.

random sensing matrices Φ whose entries are drawn from Gaussian, Uniform, and Bernoulli distributions and compare the performance of Insense against other baselines on these matrices.

1) *Random Gaussian matrix:* We start with a random Gaussian sensing matrix Φ . We conduct this experiment for 20 random trials and generate 100×100 matrices, whose entries are independently drawn from a standard normal distribution. We use Insense and other baseline algorithms to select $M = \{5, 6, 7, \dots, 30\}$ sensors. In Figure 2(a,b) we report μ_{avg} and μ_{max} of the selected submatrices Φ_{Ω} , with $|\Omega| = M$.

All sensing algorithms show comparable performances with the random sensor selection strategy, illustrating that only small improvements can be achieved using the sensor selection algorithms.

2) *Random Uniform matrix:* Here, we study sensing matrices drawn from a slightly different family of distributions. We repeat the previous experiment with a sensing matrix Φ whose entries are drawn uniformly at random from $[0, 1]$. Insense outperforms most of the baseline algorithms, including Random, and Convex SS, with a large gap in μ_{avg} (Figure 2(c)). Despite selecting completely different sensors, FrameSense and EigenMaps show comparable performances to Insense in minimizing μ_{avg} on this random dataset. Figure 2(d) indicates a gap between the maximum coherence μ_{max} that Insense achieves compared to other baselines.

3) *Random Bernoulli matrix:* We test the sensor selection algorithms on random matrices with Bernoulli 0/1 entries using sensing matrices Φ whose entries are 0 or 1 with equal

probability.¹ FrameSense and EigenMaps show similar results to Insense, which outperforms the rest of the algorithms in terms of μ_{avg} with a large margin (Figure 2(e)).² Figure 2(f) shows a clear gap between Insense and other baselines in terms of the maximum coherence μ_{max} .

These results on random sensing matrices demonstrate the importance of choosing the proper algorithm for the CS recovery task dealing with different family of random matrices. Insense selects reliable sensors that are consistently better than or comparable to other baseline algorithms on random sensing matrices.

We note that finding the optimal subset of sensors in real-world applications better reflects the power of the Insense algorithm. In contrast to random matrices, sensing matrices in real-world applications have more structure and redundancy in their sensors. Employing a proper sensor selection algorithm in these scenarios can significantly reduce the number of required sensors.

C. Highly structured datasets

We construct two synthetic datasets that better resembles the redundancies and structures in real-world datasets. We also elaborate how these datasets can have real-world implications for problems involving sensors selection.

¹Sensor selection algorithms show similar coherence minimization performance on random matrices with Bernoulli -1/1 entries to Gaussian sensing matrices.

²For the selected matrices Φ_{Ω} which contain columns with all zero entries, the average coherence μ_{avg} is not defined. The missing values in some curves correspond to these instances.

TABLE I

COMPARISON OF THE INSENSE ALGORITHM AGAINST SEVERAL OTHER BASELINE ALGORITHMS IN SELECTING SENSORS FROM A SYNTHETIC DATASET. INSENSE PICKS THE SET OF SENSORS FROM THE GAUSSIAN SUBMATRIX BY MINIMIZING THE AVERAGE COHERENCE μ_{avg} . WHILE FRAMESENSE SELECTS SENSORS FROM THE IDENTITY SUB-MATRIX AND ACHIEVES THE MINIMUM $\text{FP}(\Phi_\Omega)$ POSSIBLE, INSENSE ACHIEVES THE BEST BP RECOVERY PERFORMANCE. FOR THE SELECTED MATRICES Φ_Ω WHICH CONTAIN COLUMNS WITH ALL ZERO ENTRIES, THE AVERAGE COHERENCE μ_{avg} IS NOT DEFINED. THE DASH SIGNS CORRESPOND TO THESE INSTANCES.

Algorithms	$\mu_{\text{avg}}(\Phi_\Omega)$	$\text{FP}(\Phi_\Omega)$	$\text{CN}(\Phi_\Omega)$	BP accuracy %
Insense	0.3061 ± 0.0047	1019 ± 313	1.93 ± 0.19	92.27 ± 1.42
FrameSense	–	0.00 ± 0.00	1.00 ± 0.00	4.00 ± 0.00
EigenMaps	–	0.00 ± 0.00	1.00 ± 0.00	4.00 ± 0.00
MSE-G	0.3872 ± 0.0305	1155 ± 374	11.51 ± 0.93	57.91 ± 1.09
MI-G	–	0.00 ± 0.00	1.00 ± 0.00	4.00 ± 0.00
Entropy-G	–	0.00 ± 0.00	1.00 ± 0.00	4.00 ± 0.00
Determinant-G	–	0.00 ± 0.00	1.00 ± 0.00	4.00 ± 0.00
Greedy SS	–	0.00 ± 0.00	1.00 ± 0.00	4.00 ± 0.00
Convex SS	0.3137 ± 0.0075	2279 ± 470	2.22 ± 0.25	88.64 ± 3.64

1) *Identity - Gaussian matrix*: We construct our first highly structured datasets by concatenating two 50×50 matrices: The first matrix is an identity matrix and the second one is a random matrix with independent Gaussian entries.

This structured sensing matrix has potential implications in real-world problems. In universal microbial diagnostics [23], as an example, the identity and Gaussian matrices symbolize two different types of sensors: the identity matrix corresponds to a set of sensors that are each specific to a single target or atom (column) of the dictionary Φ , whereas the Gaussian matrix corresponds to a set of sensors that are universal to all the atoms of the dictionary.

Lest assume a potential detection scenario where the sparse recovery problem is solved to identify the present targets (columns). Once the sensors are selected from the identity matrix, they lie entirely dormant in detecting a large subset of columns of the dictionary. On the other end, once the sensors are selected from the Gaussian sub-matrix they enable detecting *all* the atoms in the dictionary and thus provide better average sparse recovery performance. Therefore, to achieve a better sparse recovery performance the sensor selection algorithm in this application should select rows (sensors) from Gaussian sub-matrix. We will elaborate more on the details of this application in Subsection V-D1.

We run Insense as well as all the baseline algorithms to select $M = 10$ rows from our structured dataset. We repeat the same experiment 10 times with different random Gaussian matrices and report the results in Table I.

Insense, Convex SS, and MSE-G are the only algorithms that select rows of the Gaussian submatrix. While achieving the minimum $\text{FP}(\Phi_\Omega)$ ($= 0$) by selecting rows from the identity matrix, Framesense performs poorly on BP recovery. The greedy algorithms also select rows from the identity matrix that result in columns with all-zero entries and thus fail to recover most of the entries in x .

Although the rows of the Gaussian sub-matrix are good sensors for CS recovery, Insense selects the rows with smaller column coherence as compared to the rows selected by Convex SS and MSE-G. As a result Insense achieves the best BP recovery performance (Table I) among them.

These results demonstrate that minimizing similarity metrics (such as frame potential, etc.) on the rows is not sufficient

for maximizing CS recovery performance, i.e., when we are interested in sensing in the underdetermined regime ($M < N$).

2) *Uniform - Gaussian matrix*: Knowing the global optimal solution would enable us to evaluate how far the solution obtained by Insense is from the optimal set of sensors as compared to other baselines. However, it is computationally hard to find the optimal sensors for a given matrix Φ of even moderate size ($D, N > 200$).

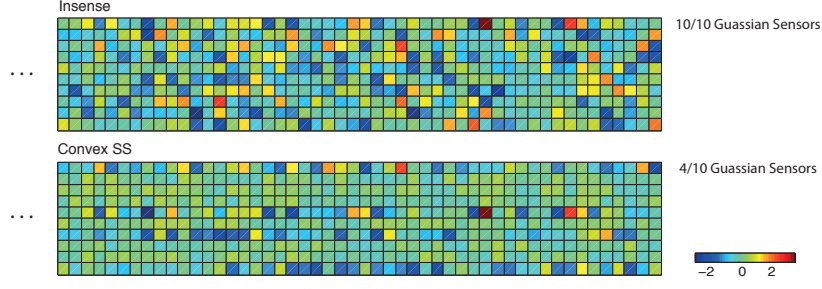
We concatenate a 10×200 matrix with entries independently drawn from the standard normal distribution with a 190×200 matrix with entries independently drawn from the uniform distribution $[0, 1]$. With this construction one would expect that the Gaussian submatrix has the lowest μ_{avg} when we set $M = 10$. We run Insense as well as other baseline algorithms on this dataset and show the results in Figure 3. In all 10 random trials, Insense successfully selects all Gaussian rows and hence find the globally optimal set of sensors. FrameSense, misses on average 10 – 20% of Gaussian sensors. EigenMaps performances similarly to FrameSense. Other baselines, including Convex SS, select only a small portion ($< 20\%$) of the Gaussian sensors. Figure 3(b) also illustrates that Insense achieves better BP recovery performance since it selects all the Gaussian sensors, resulting in minimum average coherence μ_{avg} of the resulting sensing matrix.

D. Real-world datasets

To assess the performance of the sensor selection algorithms, it is best to test them on real-world datasets. In this section, we evaluate the performance of Insense against other baseline algorithms on two real-world dataset from microbial diagnostics and structural health monitoring (SHM) system.

1) *Microbial diagnostics*: Microbial diagnostics seeks rapid and reliable methods to detect and identify microbial organisms in an infectious sample. The presence of the DNA sequence of an organism is typically measured using DNA probes that bind (hybridize) to the target sequence and emit fluorescence signal.

Designing DNA probes for microbial diagnostic settings is an important application of sensor selection algorithms in the underdetermined sensing regime. In the universal microbial sensing (UMD) framework [23], as an example, DNA probes acquire linear measurements from a microbial sample (e.g.,



(a) Instances of sensing matrices Φ_Ω selected by Insense and Convex SS and the number of selected Gaussian sensors.

Algorithms	$\mu_{\text{avg}}(\Phi_\Omega)$	FP(Φ_Ω)	CN(Φ_Ω)	Gaussian sensor ratio %	BP accuracy %
Insense	0.3165 ± 0.0023	9320 \pm 3292	1.46 \pm 0.07	100 \pm 0	58.55 ± 2.64
FrameSense	0.3273 \pm 0.0059	6095 \pm 1708	3.19 \pm 0.92	84 \pm 5	58.15 \pm 2.26
EigenMaps	0.3215 \pm 0.0021	7230 \pm 2319	2.07 \pm 0.12	90 \pm 0	57.60 \pm 3.72
MSE-G	0.5805 \pm 0.0440	78530 \pm 12450	5.99 \pm 0.31	17 \pm 4	49.90 \pm 3.54
MI-G	0.6814 \pm 0.0556	93260 \pm 109250	6.26 \pm 0.77	7 \pm 4	51.60 \pm 5.21
Entropy-G	0.7007 \pm 0.0804	98950 \pm 16216	6.61 \pm 0.48	5 \pm 7	53.70 \pm 5.21
Determinant-G	0.7303 \pm 0.0545	105700 \pm 11228	6.57 \pm 0.31	3 \pm 4	55.50 \pm 4.50
Greedy SS	0.7303 \pm 0.0545	105700 \pm 11228	5.57 \pm 0.31	3 \pm 4	55.50 \pm 4.50
Convex SS	0.5788 \pm 0.1140	75270 \pm 27383	5.97 \pm 0.77	20 \pm 15	54.40 \pm 4.20

(b) Performance of sensor selection algorithms.

Fig. 3. Comparison of the Insense algorithm against several baseline algorithms in selecting $M = 10$ sensors from a synthetic dataset. (a) While Insense picks the optimal set of Gaussian sensors with minimal coherence, Convex SS only selects a portion of Gaussian sensors (4 out of 10 sensors in this illustration). (b) FrameSense attains the lowest frame potential, however, it fails to select all Gaussian sensors. Convex SS misses most of the Gaussian sensors and performs poorly in terms of BP recovery performance. Insense provides the best BP recovery performance.

TABLE II

COMPARISON OF THE PERFORMANCE OF DNA PROBES SELECTED BY INSENSE TO THE PROBES SELECTED BY OTHER BASELINE ALGORITHMS IN IDENTIFYING THE COMPOSITION OF K BACTERIAL SPECIES. DNA SENSORS ARE SELECTED FROM A DICTIONARY COMPRISING LINEAR MEASUREMENTS OF $D = 100$ DNA PROBES FROM GENOME SEQUENCE OF $N = 42$ HUMAN PATHOGENS.

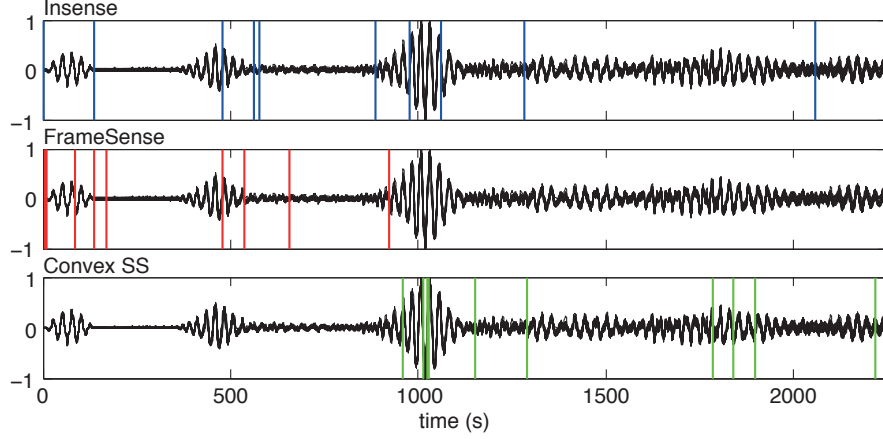
# of present organisms	$K = 2$				$K = 3$				$K = 5$			
# of DNA probes (M)	5	8	12	15	8	12	15	20	12	15	20	25
Insense	25.52	68.33	94.78	99.65	26.46	71.74	93.95	99.53	16.78	51.95	92.71	99.10
FrameSense	27.73	61.83	88.40	95.71	22.70	62.32	82.29	98.36	10.79	35.16	81.92	96.50
EigenMaps	14.97	49.65	84.69	94.66	13.17	54.68	78.09	96.25	6.69	27.47	72.13	95.30
MSE-G	27.26	60.79	91.53	97.91	22.01	67.16	89.15	98.40	14.69	43.26	83.52	97.40
MI-G	26.22	59.98	89.68	96.40	20.96	65.69	84.10	97.39	13.49	37.96	79.72	96.00
Entropy-G	27.96	61.25	91.53	98.61	21.51	66.35	88.96	99.19	14.19	42.86	89.61	97.50
Determinant-G	14.85	46.75	82.13	94.55	12.49	48.97	76.13	96.03	6.29	24.48	72.73	92.81
Greedy SS	25.52	57.54	87.70	96.87	19.72	59.65	84.64	97.34	10.99	36.16	80.22	94.11
Convex SS	15.89	53.36	87.94	98.94	14.29	57.58	87.59	98.89	7.69	38.46	83.52	98.40
Random	25.57	61.53	88.79	96.66	22.37	62.29	86.15	97.72	12.79	38.88	82.94	86.44

bacterial, viral, etc.) in the form of fluorescence energy transfer (FRET), which indicates the hybridization affinity of DNA probes to the organisms present in the sample. Given a dictionary Φ of the hybridization affinities of DNA probes to microbial species, the objective is to recover sparse vector x comprising the concentration of organisms in the sample, from its few linear measurements, similar to the problem setup described around equation (1) in the Introduction.

We run Insense and other sensor selection algorithms on a large sensing matrix comprising the hybridization affinity of $D = 100$ random DNA probes to $N = 42$ bacterial species (as described in [23]). To test the sensor selection algorithm, we first select M probes and construct a sensing matrix Φ_Ω with $|\Omega| = M$. Then, using the selected matrix Φ_Ω , we perform BP recovery algorithms to recover multiple sparse vectors x , whose support set corresponds to the set of present bacterial organisms in a simulated infectious sample. We repeat the same experiment for all $\binom{N}{K}$ sparse vectors

x with $K = \{2, 3, 5\}$ non-zero elements (i.e., bacteria in presence) and report the average BP recovery performance in identifying the composition of the microbial samples in Table II. To report the BP recovery performance on bacterial samples with $K = 5$ organisms, we randomly select 1000 sparse vector x with 5 active elements and report BP recovery performance on selected samples.

DNA probes selected by Insense outperform all the baseline algorithms in identifying the bacterial organisms in presence. Specifically, Insense requires a smaller number of DNA probes than other baselines to achieve almost perfect detection performance (i.e., BP accuracy $> 99\%$), suggesting that Insense is the most cost efficient algorithm to select DNA probes for this application. Moreover, the performance gap between Insense and other baselines enlarges as the number of bacteria species comprising the pathogenic sample K increases, indicating that Insense has a better recovery performance in complex biological samples compared to other baselines.



(a) Illustration of signal profiles from $N = 25$ crack locations (black signals) and $M = 10$ time instances sampled by Insense (vertical blue lines) and two other sensor selection algorithms (vertical red and green lines).

Detection metrics	$\mu_{\text{avg}}(\Phi_{\Omega})$						BP accuracy in localizing cracks %						Running time (s)
# of time samples (M)	5	6	7	8	9	10	5	6	7	8	9	10	10
Insense	0.2491	0.2251	0.2185	0.2127	0.2075	0.2030	40.99	60.49	76.49	86.09	91.39	94.64	4.12
FrameSense	–	0.4051	0.3843	0.3179	0.2717	0.2640	24.41	51.58	63.73	81.92	89.04	91.14	0.18
EigenMaps	0.4341	0.4102	0.4028	0.3950	0.3841	0.3699	25.13	41.16	40.55	51.38	64.52	80.45	0.11
MSE-G	0.9832	0.9779	0.9752	0.9748	0.9707	0.9637	39.50	53.50	63.73	79.18	86.28	93.50	0.97
Entropy-G	0.9694	0.9649	0.9615	0.9568	0.9525	0.9507	40.89	58.95	71.35	80.98	86.50	92.36	0.16
Greedy SS	–	–	–	–	–	–	9.32	14.96	19.55	28.36	45.77	57.43	1.65
Convex SS	0.9702	0.9685	0.9611	0.9571	0.9519	0.9451	38.92	59.56	72.42	82.83	89.98	94.56	4.52
Random	0.7980	0.7976	0.8083	0.8147	0.8111	0.8132	33.84	52.20	67.51	79.60	88.19	93.75	$< 10^{-4}$

(b) Performance of sensor selection algorithms.

Fig. 4. Comparison of the Insense algorithm against several baseline algorithms in sampling M time instances that best localize concurrence of $K = 2$ cracks on a plane in a real structural health monitoring problem. (a) Insense selects times instances which distribute along the signal and capture higher signal variation with lower mutual coherence. (b) Insense provides the best localization performance by minimizing the average mutual coherence between the crack profiles $\mu_{\text{avg}}(\Phi_{\Omega})$ with comparable running time.

2) *Structural Health Monitoring*: Structural health monitoring (SHM) studies the problem of detecting, localizing, and characterizing damages (e.g., cracks, holes, etc.) occurring on or inside structures such as buildings, bridges, pipes, etc [24]. SHM systems continuously monitor the target structure by sending signals that propagate throughout the structure and then analyze the reflected signals measured using an array of sensors.

The classical methods for damage localization calculate *time-of-flight* (TOF), followed by estimating the location of a crack using *triangulation*, i.e., locating the damages by finding the intersection of circles centered at the location of sensors with radii determined by the time the signals are measured at the sensors.

The emergence of dictionary-based methods [25], [26] enables the design of new methods for damage localization that alleviates the computational complexity of calculating damage feature (such as TOF) by solving wave equations, which is a standard approach in traditional methods. Instead, a dictionary of signal profiles is first constructed whose columns correspond to the actual impulse responses of systems with cracks at certain locations on a predefined grid. The location of one or a small number of damages on the structure is then determined by solving a sparse recovery problem where the support of the sparse signal vector x in equation (1) correspond to the damage locations [26].

In sensor dictionary-based methods, the sensors correspond

to time instances the reflected signals are measured. Therefore, applying the sensor selection algorithms to SHM systems will reduce the number of measurements required while preserving the damage localization performance.

To compare Insense against other baseline algorithms, we use SHM data from a crack localization experiment on a metal plate with 25 potential cracks located on a 5×5 square grid. The experimentally measured signals of size $D = 2250$ from all cracks are obtained and stored in a dictionary $\Phi \in \mathbb{R}^{2250 \times 25}$. The crack locations are then identified by solving the sensor selection problem with the experimentally measured sensing matrix Φ .

Figure 4(a) visualizes the $M = 10$ selected time instances for Insense and two other baselines algorithms and Figure 4(b) showcases their performance on BP recovery in localizing $K = 2$ cracks using $M \in \{5, 6, 7, 8, 9, 10\}$ time instances.³

These results show that Insense consistently outperforms other baseline algorithms in terms of crack localization accuracy, and can reduce the number of measurement time instances required in SHM systems by a factor of > 200 . We note that the measurements are not statistically independent in this dataset as they correspond to measurements of the same reflected signal at different time instances, highlighting the superior performance of Insense in applications where the independence assumption in sensing model is slightly violated.

³MI-G and Determinant-G did not run on our machine with 12 GB of RAM due to high memory requirements.

We also compared the running time of Insense against other sensor selection algorithms for this application. Insense runs faster than the other optimization-based algorithm, Convex SS. The greedy sensor selection algorithms run only slightly faster than Insense.

VI. CONCLUSION

We demonstrated that our optimization algorithm, Insense, select sensors that provide superior sparse recovery performance compared to the sensors selected by the conventional sensor selection algorithms, especially in real-world problems such as microbial diagnostics and structural health monitoring. Our results verified that the coherence-based cost function used in Insense is a more natural and appropriate cost function in selecting sensors for (near) sparse signals. With small modification to the optimization algorithm, Insense can be extended to select sensors from large-scale dictionary of sensors in big data problems. We defer this extension to a future work. We believe various other problems in sensors selection will be explored in near future with the increasing demand for larger number of sensing units in large-scale signal acquisition systems.

REFERENCES

- [1] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Sig. Proc.*, vol. 57, no. 2, pp. 451–462, 2009.
- [2] M. Shamaiah, S. Banerjee, and H. Vikalo, "Greedy sensor selection: Leveraging submodularity," in *IEEE Conf. Dec. Ctrl.*, 2010, pp. 2572–2577.
- [3] J. Ranieri, A. Chebira, and M. Vetterli, "Near-optimal sensor placement for linear inverse problems," *IEEE Trans. Sig. Proc.*, vol. 62, no. 5, pp. 1135–1146, 2014.
- [4] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [5] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [6] R. G. Baraniuk, "Compressive sensing," *IEEE Sig. Proc. Mag.*, vol. 24, no. 4, pp. 118–124, 2007.
- [7] E. J. Candès, "Compressive sampling," in *Proc. intl. Congress Mathematicians*, vol. 3, 2006, pp. 1433–1452.
- [8] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Roy. Stat. Soc. B Met.*, pp. 267–288, 1996.
- [9] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.
- [10] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pur. Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [11] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5-6, pp. 877–905, 2008.
- [12] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Sig. Proc.*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [13] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Asilomar Conf. Sig. Sys. Comp.*, 1993, pp. 40–44.
- [14] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Computat. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, 2009.
- [15] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Computat. Harmon. Anal.*, vol. 27, no. 3, pp. 265–274, 2009.
- [16] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proc. Natl. Acad. Sci.*, vol. 106, no. 45, pp. 18 914–18 919, 2009.
- [17] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [18] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 6–18, 2006.
- [19] C. Herzet, C. Soussen, J. Idier, and R. Gribonval, "Exact recovery conditions for sparse representations with partial support information," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7509–7524, 2013.
- [20] R. Gribonval and P. Vandergheynst, "On the exponential convergence of matching pursuits in quasi-incoherent dictionaries," *Information Theory, IEEE Transactions on*, vol. 52, no. 1, pp. 255–261, 2006.
- [21] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Found. Comput. Math.*, vol. 9, no. 3, pp. 317–334, 2009.
- [22] D. M. Needell and R. Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit," *IEEE J. Sel. Top. Sig. Proc.*, vol. 4, no. 2, pp. 310–316, 2010.
- [23] A. Aghazadeh, A. Y. Lin, M. A. Sheikh, A. L. Chen, L. M. Atkins, C. L. Johnson, J. F. Petrosino, R. A. Drezek, and R. G. Baraniuk, "Universal microbial diagnostics using random DNA probes," *Sci. Adv. (in press)*, vol. 2, 2016.
- [24] D. Balageas, C. P. Fritzen, and A. Güemes, *Structural Health Monitoring*. Wiley Online Library, 2006, vol. 493.
- [25] X. Q. Zhou, Y. Xia, and S. Weng, "L1 regularization approach to structural damage detection using frequency data," *Struct. Health Monit.*, pp. 14 759 217–15 604 386, 2015.
- [26] D. Sen, A. Aghazadeh, A. Mousavi, N. S., and B. G. R., "Sparse based data-driven approaches for damage detection in plates," *preprint*, 2016.
- [27] J. Ranieri, A. Vincenzi, A. Chebira, D. Atienza, and M. Vetterli, "EigenMaps: Algorithms for optimal thermal maps extraction and sensor placement on multicore processors," in *IEEE Des. Auto. Conf.*, 2012, pp. 636–641.
- [28] A. Das and D. Kempe, "Algorithms for subset selection in linear regression," in *Proc. Ann. ACM Theory Comput.*, 2008, pp. 45–54.
- [29] D. Golovin, M. Faulkner, and A. Krause, "Online distributed sensor selection," in *Proc. ACM Intl. Conf. Inf. Proc. Sens. Net.*, 2010, pp. 220–231.
- [30] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," *arXiv preprint arXiv:1102.3975*, 2011.
- [31] D. M. Steinberg and W. G. Hunter, "Experimental design: Review and comment," *Technometrics*, vol. 26, no. 2, pp. 71–97, 1984.
- [32] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, 2008.
- [33] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *Proc. Intl. ACM Conf. Inf. Proc. Sens. Net.*, 2004, pp. 36–45.
- [34] I. Ford, D. M. Titterton, and C. P. Kitsos, "Recent advances in nonlinear experimental design," *Technometrics*, vol. 31, no. 1, pp. 49–60x, 1989.
- [35] S. P. Chepuri and G. Leus, "Sparsity-promoting sensor selection for nonlinear measurement models," *IEEE Trans. Sig. Proc.*, vol. 63, no. 3, pp. 684–698, 2015.
- [36] A. Amini and F. Marvasti, "Deterministic construction of binary, bipolar, and ternary compressed sensing matrices," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2360–2370, 2011.
- [37] T. Strohmer and R. W. Heath, "Grassmannian frames with applications to coding and communication," *Appl. Computat. Harmon. Anal.*, vol. 14, no. 3, pp. 257–275, 2003.
- [38] J. Tropp, I. S. Dhillon, R. W. Heath Jr, and T. Strohmer, "Designing structured tight frames via an alternating projection method," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 188–209, 2005.
- [39] M. Elad, "Optimized projections for compressed sensing," *IEEE Trans. Sig. Proc.*, vol. 55, no. 12, pp. 5695–5702, 2007.
- [40] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization," *IEEE Trans. Image Proc.*, vol. 18, no. 7, pp. 1395–1408, 2009.
- [41] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," Technical University of Denmark, Tech. Rep., 2008.
- [42] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods," *Math. Program.*, vol. 137, no. 1-2, pp. 91–129, 2013.
- [43] Y. Chen and X. Ye, "Projection onto a simplex," *arXiv preprint arXiv:1101.6081*, 2011.
- [44] W. Wang and M. A. Carreira-Perpinán, "Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application," *arXiv preprint arXiv:1309.1541*, 2013.

[45] L. Condat, “Fast projection onto the simplex and the ℓ_1 ball,” 2014.

[Online]. Available: <https://hal.archives-ouvertes.fr/hal-01056171>