Fast Sampling for Large-scale Linear Inverse Problems via Enlarging Principle Submatrix

Fen Wang, Taihao Li, Ying Du, Yu-Ping Ruan
Institute of Artificial Intelligence, Zhejiang Lab
Hangzhou, China
fenwang@zhejianglab.com

Gene Cheung
Department of EECS, York University
Toronto, Canada
genec@yorku.ca

***Abstract*—Sampling for linear inverse problems is the active selection of partial observations so that the entire physical signal with linear model can be well recovered. In this paper, we will propose a fast sampling algorithm for large-scale linear inverse problems. Specifically, assuming that the field signal $\mathbf{f}$ is represented by a linear model $\mathbf{f} = \mathbf{\Phi}\mathbf{g}$, it can be estimated from partial noisy samples via an unbiased least-squares (LS) method, whose expected mean square error (MSE) depends on chosen samples. First, we formulate an approximate MSE problem, and then prove it is equivalent to an optimization of a principle submatrix of $\mathbf{\Phi}\mathbf{\Phi}^\top$ indexed by sample set. By enlarging the submatrix greedily, we solve the proposed combinatorial problem with simple matrix-vector multiplications, leveraging a matrix inverse lemma. To further reduce complexity, we reuse results in the previous greedy step for warm start, so that candidates can be evaluated via lightweight vector-vector multiplications. Extensive experiments show that our proposed sampling method achieved the least execution time and the best performance compared to state-of-the-art schemes, especially for large-size scenarios.**

***Index Terms*—Fast sampling, linear inverse problem, greedy approach**

I. Introduction

Due to limited sensing resources and expensive acquisition cost, sampling is the core problem in signal processing to choose a subset of locations to sense physical fields, such as temperature, humidity and transportation congestion [1]–[3], so that the unobserved signal can be recovered in high accuracy [4]–[7]. Assuming that the physical signal is linearly modelled by $\mathbf{f} = \mathbf{\Phi}\mathbf{g}$ with low-dimensional parameters [8], it can be recovered from partial noisy observations by solving a *linear inverse problem* using the least square (LS) method, whose *mean square error* (MSE) is a function of matrix $\mathbf{\Phi}$ and the sampling set [9]. Sampling for linear inverse problem is to decide the noisy observations actively such that the formulated MSE can be reduced.

To solve the combinatorial sampling problem, [10] relaxed the nonconvex optimization into a convex one, so that it can be solved by fast interior point methods. However, this relaxation performed poorly when the sampling budget was very small. To achieve better MSE performance, greedy algorithms were designed to select sample locations one-by-one by solving different local optimization problems [11]–[13]. Authors in [11] proposed one near-optimal sampling algorithm for the frame-potential-based cost function, which preserved theoretically bounded performance. In parallel, authors in [12] used the worst-case of MSE function—E-optimality criterion [14]—as objective to choose samples greedily. However, in each greedy search, it required computing eigenspace of the selected partial matrix, which is expensive when the eigenspace dimension is large. Recently, [13] proposed a fast algorithm to minimize an approximate MSE criterion, which still suffered from multiple matrix inverse computations and thus is not practical for large-scale linear inverse problems.

Towards fast and effective sampling, in this paper we propose a greedy approach with simple vector-vector multiplications. Specifically, first we formulate an approximate MSE problem using a small spectral shift, and then prove it is equivalent to an optimization of a principle submatrix of $\mathbf{\Phi}\mathbf{\Phi}^\top$ indexed by samples. We devise a fast greedy algorithm for this problem without matrix inverse computation based on a matrix inverse lemma [15]. Using computed results in the previous greedy step for warm start, we design an accelerated strategy to evaluate each candidate via simple vector-vector multiplications. Extensive experiments show that our proposed scheme is the fastest one among popular sampling algorithms for large-scale linear inverse problems, along with the best MSE performance.

Key Notations: Denote by bold lowercase letters ($\mathbf{x}$) and bold uppercase letters ($\mathbf{A}$) vectors and matrices, respectively. The entries in matrix $\mathbf{A}$ is a_{ij}. Matrix $\mathbf{I}$ is an identity matrix, whose dimension depends on context. $\mathbf{A}_{\mathcal{S}_1,\mathcal{S}_2}$ is a sub-matrix of matrix $\mathbf{A}$ with rows and columns indexed by $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively. $\mathbf{A}_{\mathcal{S},\mathcal{S}}$ is simplified to $\mathbf{A}_{\mathcal{S}}$.

II. Preliminaries

In the physical signal processing domain, the high-dimensional field signal $\mathbf{f} \in \mathbb{R}^N$ is conventionally modelled by a linear equation [16]:

$$\mathbf{f} = \mathbf{\Phi}\mathbf{g}, \tag{1}$$

where $\mathbf{g} \in \mathbb{R}^K$ is the parameter vector where $K \ll N$.

Given limited measuring resources, we select M samples out of N total signal, where $M \ll N$. We first define a

This work was supported by the China Postdoctoral Science Foundation under Grant No.2021TQ0309, and by the Major Scientific Project of Zhejiang Lab under Grant No.2020KB0AC01.

Corresponding Author: Taihao Li (lith@zhejianglab.com) sampling matrix $\mathbf{C} \in \{0,1\}^{M \times N}$ associated with sampling set $\mathcal{S} \subset \{1, \ldots, N\}$ as follows [17]:

$$c_{ij} = \begin{cases} 1, & \text{if } j = \mathcal{S}\{i\}; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Thus, the observed noisy signal can be expressed as

$$\mathbf{y} = \mathbf{C}\mathbf{f} + \mathbf{n} = \mathbf{C}\mathbf{\Phi}\mathbf{g} + \mathbf{n} \in \mathbb{R}^M, \tag{3}$$

where $\mathbf{n}$ is an i.i.d Gaussian noise signal with covariance matrix $\sigma^2\mathbf{I}$.

Using observed partial signal $\mathbf{y}$, we can estimate the original signal $\mathbf{f}$ by solving a linear inverse problem. Specifically, assuming the coefficient matrix $\mathbf{C\Phi}$ is full column rank[1], the parameter vector can be estimated as $\hat{\mathbf{g}} = (\mathbf{C\Phi})^\dagger\mathbf{y}$, where $\dagger$ denotes the pseudo inverse computation, and $\hat{\mathbf{g}}$ is an unbiased estimator of $\mathbf{g}$ with minimum variance [9]. Given $\hat{\mathbf{g}}$, the estimated target signal is $\hat{\mathbf{f}} = \mathbf{\Phi}\hat{\mathbf{g}}$.

Since $(\mathbf{C\Phi})^\dagger = \left[(\mathbf{C\Phi})^\top(\mathbf{C\Phi})\right]^{-1}(\mathbf{C\Phi})^\top$ for full column-rank matrix, the expected MSE of this least square (LS) solution is [17]

$$\text{MSE}(\hat{\mathbf{g}}) = \mathbb{E}\left(\|\hat{\mathbf{g}} - \mathbf{g}\|_2^2\right) = \sigma^2\text{Tr}\left[(\mathbf{C\Phi})^\top(\mathbf{C\Phi})\right]^{-1}. \tag{4}$$

Hence, a MSE-based sampling problem for placing sensors is to select M samples to minimize the expected MSE[2]. This is also called *A-optimality* in experimental design [14]:

$$\min_{\mathbf{C}} \ \text{Tr}\left[(\mathbf{C\Phi})^\top(\mathbf{C\Phi})\right]^{-1} = \sum_{k=1}^{K} \frac{1}{\lambda_k} \tag{5}$$

where $\lambda_K \geq \cdots \geq \lambda_2 \geq \lambda_1$ are the eigenvalues of matrix $(\mathbf{C\Phi})^\top(\mathbf{C\Phi})$ [17]. Since $\mathbf{C\Phi}$ has full-column rank and $(\mathbf{C\Phi})^\top(\mathbf{C\Phi})$ is positive semi-definite (PSD) by definition, $(\mathbf{C\Phi})^\top(\mathbf{C\Phi})$ has the property that $\lambda_k > 0, \forall k$.

III. Fast Sampling Strategy to minimize an approximate MSE problem

In this section, we at first propose an augmented A-optimality criterion as our sampling objective. Then, we propose to mitigate the large matrix inverse in each greedy step based on matrix inverse formula. For fast sampling, we propose one strategy to reduce the computation burden based on warm starts. At last, we will analyze the complexity of our proposed sampling algorithm.

A. Proposed Sampling Objective

First, we propose a modified A-optimality sampling criterion that closely approximates the original problem (5) by adding a small constant shift:

$$\min_{\mathbf{C}} \ \text{Tr}\left[(\mathbf{C\Phi})^\top(\mathbf{C\Phi}) + \mu\mathbf{I}\right]^{-1} = \sum_{k=1}^{K} \frac{1}{\lambda_k + \mu} \tag{6}$$

where $\mu > 0$ is a small shift parameter. This shifted sampling objective was also adopted in sensor placement [13] and graph sampling [17]. We will present a faster algorithm minimizing this objective than ones in [13] and [17] in this paper.

It can be proven that the modified optimization (6) has the same optimal solution(s) if $(\mathbf{\Phi\Phi}^\top + \mu\mathbf{I})_\mathcal{S}$ replaces $(\mathbf{C\Phi})^\top(\mathbf{C\Phi}) + \mu\mathbf{I}$ in (6). We formally state this in the following proposition:

Proposition 1: Denote the eigenvalues of matrix $(\mathbf{C\Phi})^\top(\mathbf{C\Phi})$ by $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_K$, and assume rank$(\mathbf{C\Phi}) = K$. When $|\mathcal{S}| = M \geq K$, an optimal sampling set for problem (6) is also optimal to the following optimization:

$$\min_{\mathcal{S}} \ \text{Tr}\left[(\mathbf{\Phi\Phi}^\top + \mu\mathbf{I})_\mathcal{S}\right]^{-1} \tag{7}$$

where the relationship between $\mathbf{C}$ and $\mathcal{S}$ is defined in (2).

Proof: When $M = K$, matrix $\mathbf{C\Phi}$ is a square matrix, so the eigenvalues of $(\mathbf{C\Phi})^\top(\mathbf{C\Phi})$ and $(\mathbf{C\Phi})(\mathbf{C\Phi})^\top = (\mathbf{\Phi\Phi}^\top)_\mathcal{S}$ are the same. Therefore, problem (6) and (7) have equivalent optimal solution; When $M > K$, assume first that the singular value decomposition (SVD) of matrix $\mathbf{C\Phi} \in \mathbb{R}^{M \times K}$ is $\mathbf{C\Phi} = \mathbf{U\Sigma V}^\top$ where $\mathbf{U} \in \mathbb{R}^{M \times M}$ and $\mathbf{V} \in \mathbb{R}^{K \times K}$ are orthogonal left and right eigenvectors respectively. The singular value matrix $\mathbf{\Sigma} \in \mathbb{R}^{M \times K}$ has the form:

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_K \\ & \mathbf{0} & \end{bmatrix} \tag{8}$$

where $\sigma_k \neq 0, \forall k$, since rank$(\mathbf{C\Phi}) = K$ and $M > K$.

Then, the eigenvalues of $(\mathbf{C\Phi})^\top(\mathbf{C\Phi}) = \mathbf{V\Sigma}^\top\mathbf{\Sigma V}^\top$ are $\{\sigma_1^2, \cdots, \sigma_K^2\}$, where $\lambda_k = \sigma_k^2 > 0, \forall k$. And, the eigenvalues of matrix $(\mathbf{\Phi\Phi}^\top)_\mathcal{S} = (\mathbf{C\Phi})^\top(\mathbf{C\Phi}) = \mathbf{U\Sigma\Sigma}^\top\mathbf{U}^\top$ are $\{\underbrace{0, \ldots, 0}_{M-K}, \lambda_1, \ldots, \lambda_K\}$, which indicates the eigenvalues of $(\mathbf{\Phi\Phi}^\top)_\mathcal{S} + \mu\mathbf{I}$ are

$$\{\underbrace{\mu, \ldots, \mu}_{M-K}, \lambda_1 + \mu, \ldots, \lambda_K + \mu\} \tag{9}$$

Therefore,

$$\text{Tr}\left[(\mathbf{\Phi\Phi}^\top)_\mathcal{S} + \mu\mathbf{I}\right]^{-1} = \frac{M-K}{\mu} + \sum_{k=1}^{K} \frac{1}{\lambda_k + \mu} \tag{10}$$

where $\lambda_k + \mu > 0$ since μ is strictly positive and $\lambda_k > 0$.

Combined with equation (6), we will know that

$$\text{Tr}\left[(\mathbf{\Phi\Phi}^\top)_\mathcal{S} + \mu\mathbf{I}\right]^{-1} = \frac{M-K}{\mu} + \text{Tr}\left[(\mathbf{C\Phi})^\top(\mathbf{C\Phi}) + \mu\mathbf{I}\right]^{-1}$$

whose left part can be rewritten as equation (7).

Notice that i) M, K and μ are fixed constants not affected by optimization, and ii) optimization variables $\mathbf{C}$ in (6) or $\mathcal{S}$ in (7) have the same degrees of freedom. We can thus conclude that optimal solutions to (6) and to (7) are the same. □

Remark: This formulation was also derived using Neumann series theorem [19] for sampling of bandlimited graph signals,

[1] One necessary but not sufficient condition of rank$(\mathbf{C\Phi}) = K$ is $M \geq K$ [18].

[2] A-optimality formulation minimizes the expected MSE directly, while other cost functions, like E-optimality or D-optimality criterion, minimize proxies of MSE [14].

where $\mathbf{\Phi}$ is formed using the first K orthogonal eigenvectors of a combinatorial graph Laplacian matrix in [17]. In contrast, in this paper $\mathbf{\Phi}$ is a general measurement matrix for modelling physical filed, which in general does not have the orthogonal property. Thus, we derive the more general result differently here.

B. Greedy Algorithm without Matrix Inversion

Given that the sampling problem (7) is combinatorial in nature, we employ a greedy approach to optimally choose one sample at a time. Specifically, assuming that we have obtained set $\mathcal{S}_t$ after t iterations, to decide the $(t+1)$-th sample, we solve the following local optimization problem:

$$\min_{i \in \mathcal{S}_t^c} \quad \underbrace{\mathrm{Tr}\left[(\mathbf{\Phi}\mathbf{\Phi}^\top + \mu\mathbf{I})_{\mathcal{S}_t \cup \{i\}}\right]^{-1}}_{f(\mathcal{S}_t \cup \{i\})} \tag{11}$$

where $\mathcal{S}_0 = \emptyset$.

If we compute the objective directly to evaluate each candidate solution $i \in \mathcal{S}_t^c$, we have to perform matrix inversion, with complexity at most $\mathcal{O}(M^3)$. To mitigate large matrix inversion, we introduce the next greedy strategy based on one matrix inverse formula [15].

For notation simplicity, we first define $\mathbf{Q} = \mathbf{\Phi}\mathbf{\Phi}^\top + \mu\mathbf{I}$. Since matrix $\mathbf{Q}$ is symmetric, with appropriate permutation, its sub-matrix $\mathbf{Q}_{\mathcal{S}_t \cup \{i\}}$ can be expressed as

$$\mathbf{Q}_{\mathcal{S}_t \cup \{i\}} = \begin{bmatrix} \mathbf{Q}_{\mathcal{S}_t} & \mathbf{Q}_{\mathcal{S}_t,\{i\}} \\ \mathbf{Q}_{\{i\},\mathcal{S}_t} & q_{ii} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{\mathcal{S}_t} & \mathbf{p}_{t,i} \\ \mathbf{p}_{t,i}^\top & q_{ii} \end{bmatrix}, \tag{12}$$

where $\mathbf{p}_{t,i} = \mathbf{Q}_{\mathcal{S}_t,\{i\}} \in \mathbb{R}^t$.

Matrix inverse $\mathbf{Q}^{-1}_{\mathcal{S}_t \cup \{i\}}$ in equation (11) can be obtained using $\mathbf{Q}^{-1}_{\mathcal{S}_t}$ via the *matrix inversion formula* [15]:

$$\mathbf{Q}^{-1}_{\mathcal{S}_t \cup \{i\}} = \begin{bmatrix} \mathbf{Q}^{-1}_{\mathcal{S}_t} + h_i^{-1}\mathbf{Q}^{-1}_{\mathcal{S}_t}\mathbf{p}_{t,i}\mathbf{p}_{t,i}^\top\mathbf{Q}^{-1}_{\mathcal{S}_t} & -h_i^{-1}\mathbf{Q}^{-1}_{\mathcal{S}_t}\mathbf{p}_{t,i} \\ -h_i^{-1}\mathbf{p}_{t,i}^\top\mathbf{Q}^{-1}_{\mathcal{S}_t} & h_i^{-1} \end{bmatrix} \tag{13}$$

where $h_i = q_{ii} - \mathbf{p}_{t,i}^\top\mathbf{Q}^{-1}_{\mathcal{S}_t}\mathbf{p}_{t,i}$ is a scalar.

Therefore,

$$\begin{aligned} f(\mathcal{S}_t \cup \{i\}) &= \mathrm{Tr}\left(\mathbf{Q}^{-1}_{\mathcal{S}_t}\right) + h_i^{-1}\mathrm{Tr}\left(\mathbf{Q}^{-1}_{\mathcal{S}_t}\mathbf{p}_{t,i}\mathbf{p}_{t,i}^\top\mathbf{Q}^{-1}_{\mathcal{S}_t}\right) + h_i^{-1} \\ &= f(\mathcal{S}_t) + h_i^{-1}\|\mathbf{Q}^{-1}_{\mathcal{S}_t}\mathbf{p}_{t,i}\|_2^2 + h_i^{-1} \end{aligned}$$

Because $f(\mathcal{S}_t)$ is a constant not affected by the selection of candidate i, during the $(t+1)$-th greedy step given input $\mathcal{S}_t$, the sampling problem (11) can be simplified as:

$$\begin{aligned} \min_{i \in \mathcal{S}_t^c} \quad & h_i^{-1}\|\mathbf{Q}^{-1}_{\mathcal{S}_t}\mathbf{p}_{t,i}\|_2^2 + h_i^{-1} \\ \text{s.t.} \quad & h_i = q_{ii} - \mathbf{p}_{t,i}^\top\mathbf{Q}^{-1}_{\mathcal{S}_t}\mathbf{p}_{t,i};\ \mathbf{p}_{t,i} = \mathbf{Q}_{\mathcal{S}_t,\{i\}} \end{aligned} \tag{14}$$

where $\mathbf{Q}^{-1}_{\mathcal{S}_t}$ is already computed during the previous iteration.

Compared to solving problem (11), optimizing problem (14) needs to compute matrix-vector product $\mathbf{Q}^{-1}_{\mathcal{S}_t}\mathbf{p}_{t,i}$ with complexity $\mathcal{O}(t^2)$, given known $\mathbf{Q}^{-1}_{\mathcal{S}_t}$.

To optimize problem (14), we can compute $\mathbf{Q} = \mathbf{\Phi}\mathbf{\Phi}^\top + \mu\mathbf{I}$ once with complexity $\mathcal{O}(KN^2)$ and then query its partial entries q_{ii} and $\mathbf{p}_{t,i}$ for greedy evaluation. Next, we will propose to compute the involved elements on the fly inside greedy search without first computing $\mathbf{Q}$ and further reduce the evaluation complexity.

Algorithm 1 Fast MSE-based sampling (FMBS)

```
Input: $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_N]^\top$, sample size $M$ and $\mu$
Initialization: $\mathcal{S} = \emptyset$
Compute $q_{ii} = \|\boldsymbol{\phi}_i\|_2^2 + \mu, \forall i$
Select the first node by $i^* = \mathrm{argmax}_i q_{ii}$
Update $\mathcal{S} \leftarrow \mathcal{S} \cup \{i^*\}$
While $|\mathcal{S}| < M$
$\forall i \in \mathcal{S}^c$, compute
    If $|\mathcal{S}| = 1$
        $\mathbf{p}_i = \boldsymbol{\phi}_{i^*}^\top\boldsymbol{\phi}_i$ and $\mathbf{r}_i = \frac{1}{q_{i^*i^*}}\mathbf{p}_i$
    else
        $\alpha = \mathbf{p}_i^\top\mathbf{r}_{i^*}/h_{i^*}$ and $\beta = \boldsymbol{\phi}_{i^*}^\top\boldsymbol{\phi}_i/h_{i^*}$
        $\mathbf{r}_i = \begin{bmatrix} \mathbf{r}_i + \alpha\mathbf{r}_{i^*} - \beta\mathbf{r}_{i^*} \\ -\alpha + \beta \end{bmatrix}$
        $\mathbf{p}_i = [\mathbf{p}_i^\top \quad \beta h_{i^*}]^\top$
    end If
    $h_i = q_{ii} - \mathbf{p}_i^\top\mathbf{r}_i$
Select $i^* = \arg\min_{i \in \mathcal{S}^c} h_i^{-1}\|\mathbf{r}_i\|_2^2 + h_i^{-1}$
Update $\mathcal{S} \leftarrow \mathcal{S} \cup \{i^*\}$
end While
Return $\mathcal{S}$
```

C. Evaluation Complexity Reduction

We first write input matrix $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_N]^\top$, where $\boldsymbol{\phi}_i^\top$ is the i-th row in matrix $\mathbf{\Phi}$. Then we can compute q_{ii} as

$$q_{ii} = \boldsymbol{\delta}_i^\top\mathbf{Q}\boldsymbol{\delta}_i = \boldsymbol{\delta}_i^\top\mathbf{\Phi}\mathbf{\Phi}^\top\boldsymbol{\delta}_i + \mu = \|\boldsymbol{\phi}_i\|_2^2 + \mu \tag{15}$$

The value of $\mathbf{p}_{t,i}$ can be obtained via

$$\mathbf{p}_{t,i} = \mathbf{Q}_{\mathcal{S}_t,\{i\}} = \mathbf{C}_{(\mathcal{S}_t)}(\mathbf{\Phi}\mathbf{\Phi}^\top + \mu\mathbf{I})\boldsymbol{\delta}_i = \mathbf{C}_{(\mathcal{S}_t)}\mathbf{\Phi}\boldsymbol{\phi}_i + \mu\mathbf{C}_{(\mathcal{S}_t)}\boldsymbol{\delta}_i$$

where $\boldsymbol{\delta}_i$ is the i-th column of identity matrix $\mathbf{I}$, and $\mathbf{C}_{(\mathcal{S}_t)}$ is the sampling matrix corresponding to set $\mathcal{S}_t$, defined in equation (2).

Based on the definition of sampling matrix,

$$\mathbf{C}_{(\mathcal{S}_t)}\mathbf{\Phi}\boldsymbol{\phi}_i = \mathbf{\Phi}_{\mathcal{S}_t,:}\boldsymbol{\phi}_i = [\boldsymbol{\phi}_{\mathcal{S}_t\{1\}}, \ldots, \boldsymbol{\phi}_{\mathcal{S}_t\{t\}}]^\top\boldsymbol{\phi}_i \tag{16}$$

and

$$\mu\mathbf{C}_{(\mathcal{S}_t)}\boldsymbol{\delta}_i = \mathbf{0} \tag{17}$$

since $i \in \mathcal{S}_t^c$.

Using (15) to (17), we can simplify optimization (14) to the following

$$\begin{aligned} \min_{i \in \mathcal{S}_t^c} \quad & h_i^{-1}\|\mathbf{r}_{t,i}\|_2^2 + h_i^{-1} \\ \textit{s.t.} \quad & h_i = \|\boldsymbol{\phi}_i\|_2^2 + \mu - \mathbf{p}_{t,i}^\top\mathbf{r}_{t,i}; \\ & \mathbf{r}_{t,i} = \mathbf{Q}^{-1}_{\mathcal{S}_t}\mathbf{p}_{t,i}; \\ & \mathbf{p}_{t,i} = [\boldsymbol{\phi}_{\mathcal{S}_t\{1\}}, \ldots, \boldsymbol{\phi}_{\mathcal{S}_t\{t\}}]^\top\boldsymbol{\phi}_i \end{aligned} \tag{18}$$

where h_i and $\mathbf{p}_{t,i}$ are computed from input $\mathbf{\Phi}$ without first computing $\mathbf{Q}$ compared to equation (14). However, this formulation still require matrix-vector multiplications for evaluating one candidate.

TABLE I
COMPUTATIONAL COMPLEXITY COMPARISONS OF DIFFERENT SENSOR PLACEMENT METHODS

Method	Convex [10] / SparSenSe [20]	FrameSense [11]	MNEP [12]
Complexity	$\mathcal{O}(i_c N^3)$ / $\mathcal{O}(i_s N^3)$	$\mathcal{O}(N^3)$	$\mathcal{O}(NMK^3)$
Method	MPME [12]	fastMSE [13]	FMBS
Complexity	$\mathcal{O}(NMK^2)$	$\mathcal{O}(NMK^2)$	$\mathcal{O}(NM^2)$

Suppose that the optimal sample in step t is i^*. Then $\mathcal{S}_t = \mathcal{S}_{t-1} \cup \{i^*\}$. For candidate node in unsampled set $i \in \mathcal{S}_t^c$, we can write

$$\mathbf{p}_{t,i} = [\mathbf{\Phi}_{\mathcal{S}_{t-1},:}^\top \quad \boldsymbol{\phi}_{i^*}]^\top \boldsymbol{\phi}_i = [\mathbf{p}_{t-1,i}^\top \quad \boldsymbol{\phi}_{i^*}^\top \boldsymbol{\phi}_i]^\top \tag{19}$$

Thus, using computed result $\mathbf{p}_{t-1,i}$ in last greedy step as warm start, the computation of $\mathbf{p}_{t,i}$ only requires once vector-vector multiplication $\boldsymbol{\phi}_{i^*}^\top \boldsymbol{\phi}_i$. Further, for $i \in \mathcal{S}_t^c$,

$$\begin{aligned}
\mathbf{r}_{t,i} &= \mathbf{Q}_{\mathcal{S}_t}^{-1} \mathbf{p}_{t,i} = \mathbf{Q}_{\mathcal{S}_{t-1} \cup \{i^*\}}^{-1} \mathbf{p}_{t,i} \\
&= \begin{bmatrix} \mathbf{Q}_{\mathcal{S}_{t-1}}^{-1} + h_{i^*}^{-1} \mathbf{r}_{t-1,i^*} \mathbf{r}_{t-1,i^*}^\top & -h_{i^*}^{-1} \mathbf{r}_{t-1,i^*} \\ -h_{i^*}^{-1} \mathbf{r}_{t-1,i^*}^\top & h_{i^*}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{p}_{t-1,i} \\ \boldsymbol{\phi}_{i^*}^\top \boldsymbol{\phi}_i \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{r}_{t-1,i} + h_{i^*}^{-1} \mathbf{r}_{t-1,i^*} \mathbf{r}_{t-1,i^*}^\top \mathbf{p}_{t-1,i} - h_{i^*}^{-1} \boldsymbol{\phi}_{i^*}^\top \boldsymbol{\phi}_i \mathbf{r}_{t-1,i^*} \\ -h_{i^*}^{-1} \mathbf{r}_{t-1,i^*}^\top \mathbf{p}_{t-1,i} + h_{i^*}^{-1} \boldsymbol{\phi}_{i^*}^\top \boldsymbol{\phi}_i \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{r}_{t-1,i} + \alpha \mathbf{r}_{t-1,i^*} - \beta \mathbf{r}_{t-1,i^*} \\ -\alpha + \beta \end{bmatrix}
\end{aligned} \tag{20}$$

where

$$\alpha = h_{i^*}^{-1} \mathbf{p}_{t-1,i}^\top \mathbf{r}_{t-1,i^*} \tag{21}$$

$$\beta = h_{i^*}^{-1} \boldsymbol{\phi}_{i^*}^\top \boldsymbol{\phi}_i \tag{22}$$

and

$$h_{i^*} = q_{i^* i^*} - \mathbf{p}_{t-1,i^*}^\top \mathbf{r}_{t-1,i^*} \tag{23}$$

Remark: Based on equations (20) to (23), for computing $\mathbf{r}_{t,i}$, we need to reuse the computed warm starts in the last greedy step h_{i^*}, $\mathbf{r}_{t-1,i}$ and $\mathbf{p}_{t-1,i}$ and compute two new vector-vector multiplications, *i.e.*, $\mathbf{p}_{t-1,i}^\top \mathbf{r}_{t-1,i^*}$ and $\boldsymbol{\phi}_{i^*}^\top \boldsymbol{\phi}_i$.

Therefore, according to equations (19) and (20), the computations of $\mathbf{p}_{t,i}$ and $\mathbf{r}_{t-i}$ only require four vector-vector multiplications for evaluating one candidate. We write the complete greedy procedure in Algorithm 1, given matrix $\mathbf{\Phi}$ as input, where subscript t is abbreviated for simplicity.

D. Complexity Analysis

Given sampling budget M and the number of unsampled candidates $\mathcal{O}(N)$, the complexity of the proposed method is $\mathcal{O}(NM^2)$, because the complexity of vector-vector multiplications in each greedy search is at most $\mathcal{O}(M)$. We call this method *Fast MSE-based Sampling* (FMBS). We compare the proposed method with the following popular methods: convex relaxation-based (Convex) [10], sparse-aware sensor selection (SparSenSe) [20], FrameSense [11], minimum nonzero eigenvalue pursuit (MNEP) [12], maximal projection on minimum eigenspace (MPME) [12] and fast MSE pursuit-based (fastMSE) sampling [13]. The computational complexities of those methods and the proposed FMBS are illustrated in Table. I, where some results are borrowed from [12] and [13]. The parameter i_c is the iteration number of the interior-point method used in paper [10], which is typically tens. Similarly, i_s is the iteration number in the SparSenSe method. Table. I shows that our proposed algorithm has the lowest theoretical complexity. Corresponding empirical execution time and performance comparisons will be discussed in Section IV.

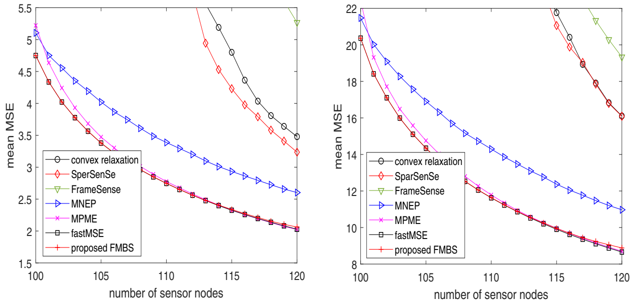

Fig. 1. Mean MSE of different sampling algorithms in terms of sample budget, where the left measurement matrix $\mathbf{\Phi}$ is generated from model 1 and the right one is generated from model 2.

TABLE II
EXECUTION TIME FOR DIFFERENT SAMPLING METHODS IN TERMS OF GIVEN BUDGET M, WHEN $N = 1000$ AND $K = 100$.

	100	105	110	115	120
Convex [10]	23.69	23.60	23.35	23.14	22.63
FrameSense [11]	0.225	0.222	0.222	0.222	0.223
MNEP [12]	45.50	53.47	61.41	69.27	77.10
MPME [12]	0.090	0.093	0.099	0.104	0.112
fastMSE [13]	0.071	0.072	0.077	0.079	0.085
FMBS	**0.064**	**0.067**	**0.071**	**0.075**	**0.082**

IV. EXPERIMENTATION

In this section, we will present extensive experimental results to evaluate the performance and efficacy of the proposed FMBS method. The measurement matrix $\mathbf{\Phi} \in \mathbb{R}^{N \times K}$ in simulations can be generated from the following models [12]:

Model 1: $\mathbf{\Phi} \in \mathbb{R}^{1000 \times 100}$ is a Gaussian random matrix with i.i.d. components $\phi_{ij} \sim \mathcal{N}(0, 1)$, and the variance of the noise $\sigma^2 = 1$;

Model 2: $\mathbf{\Phi} \in \mathbb{R}^{1000 \times 100}$ is a Bernoulli random matrix with i.i.d. components $\phi_{ij} \sim \mathcal{B}(1, 1.5)$ with $\mathcal{B}$ representing the Binomial distribution, and the variance of the noise $\sigma^2 = 1$;

In our paper, we compute the averaged MSE value for each sampling algorithm via [12]:

$$\overline{\text{MSE}} = \frac{1}{L} \sum_{i=1}^{L} \text{Tr} \left[(\mathbf{C}_i \mathbf{\Phi}_i)^\top (\mathbf{C}_i \mathbf{\Phi}_i) \right]^{-1} \tag{24}$$

where L is the number of Monte-Carlo simulations and $\mathbf{C}_i$ is the sampling matrix obtained by different sampling methods at the i-th Monte-Carlo simulation.

We simulated the algorithms listed in Table. I for performance and execution time comparisons. The parameter μ was set to be 10^{-4} for the proposed method and fastMSE method. Fig. 1 demonstrates the mean MSE performance of

TABLE III
EXECUTION TIME FOR DIFFERENT SAMPLING ALGORITHMS WITH FIELD SIZE $N = 1000$ AND $K = M$.

	100	200	300	400	500
MNEP [12]	46.11	386.1	1258	2882	5444
MPME [12]	0.093	0.659	1.831	4.376	7.86
fastMSE [13]	0.085	0.567	1.450	2.898	4.69
FMBS	**0.069**	**0.318**	**0.768**	**1.340**	**1.95**

TABLE IV
EXECUTION TIME FOR DIFFERENT SAMPLING METHODS WITH VARYING N AND $M = K = 10\%N$.

	2e3	4e3	6e3	8e3	1e4
MPME [12]	0.873	7.24	26.2	68.8	156
fastMSE [13]	1.035	8.46	31.2	80.2	171
FMBS	**0.863**	**6.84**	**22.4**	**53.0**	**103**

all simulated algorithms after 10 trails. Fig. 1 validates that the proposed FMBS achieved the same performance as fastMSE method, which was better than all other competing popular methods, especially when the sampling budget was small. As stated after equation (6), fastMSE sampling optimized the problems (6) directly by greedy search, which has the same greedy solution as solving problem (7).

To evaluate experimental complexities of different methods, we recorded the execution time of different sampling methods under model 1, and illustrated the results in Table. II, where the best results are marked in bold. For this simulation, all experiments were performed on a desktop with Intel Core i7-9700 and 16GB of RAM on Windows 10 for counting time (in seconds). Table. II shows that the proposed FMBS cost the least sampling time for all simulated sampling budgets, including the method fastMSE with the same MSE performance.

Further, we recorded the sampling time of MNEP, MPME, fastMSE and FMBS sampling methods on measurement matrix generated from model 1 with size $N = 1000$, but with varied K. Convex and FrameSense sampling were not simulated since their performance were not competitive. We set $M = K$ to fulfill the full column-rank requirement. The averaged execution time is presented in Table. III, which indicates the proposed FMBS was the fastest one among all simulated schemes.

At last, for sampling signals in large-scale problems, we performed relatively fast methods MPME and fastMSE as comparisons, where the size of target signal N was varying from 2×10^3 to 10^4, and sampling budget M is set to be $10\%N$. The dimension of parameter vector was $K = M$. Sampling time results of simulated methods were presented in Table. IV, where the best results were marked in bold. It can be observed from Table. IV that our proposed method had the least sampling time in large-scale problems.

V. CONCLUSION

In this paper, we proposed a fast sampling method for large-scale linear inverse problems. Specifically, assuming the field signal $\mathbf{f}$ is modelled by a linear model $\mathbf{f} = \mathbf{\Phi}\mathbf{g}$, it can be estimated from partial noisy samples via an unbiased least-square (LS) method, whose expected mean square error (MSE) is a function of measurement matrix $\mathbf{\Phi}$ and sample set. We formulated an approximate MSE problem, and then proved it is equivalent to a problem of a sample-dependent principle submatrix of $\mathbf{\Phi}\mathbf{\Phi}^\top$. We proposed a fast greedy algorithm using simple vector-vector multiplications via leveraging matrix inverse lemma and greedy warm starts. Extensive experiments have validated superiority of our proposed method compared to existing popular methods.

REFERENCES

[1] A. Hashemi, M. Ghasemi, H. Vikalo, and U. Topcu, "Randomized greedy sensor selection: Leveraging weak submodularity," *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 199–212, 2020.

[2] D. Lu, Y. Zheng, A. Penirschke, and R. Jakoby, "Humidity sensors based on photolithographically patterned pva films deposited on saw resonators," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 13–14, 2016.

[3] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.

[4] G. Ortiz-Jiménez, M. Coutino, S. P. Chepuri, and G. Leus, "Sparse sampling for inverse problems with tensors," *IEEE Transactions on Signal Processing*, vol. 67, no. 12, pp. 3272–3286, 2019.

[5] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies." *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3775–3789, 2016.

[6] F. Wang, Y. Wang, G. Cheung, and C. Yang, "Graph sampling for matrix completion using recurrent gershgorin disc shift," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2814–2829, 2020.

[7] L. F. O. Chamon and A. Ribeiro, "Greedy sampling of graph signals," *IEEE Transactions on Signal Processing*, vol. 66, no. 1, pp. 34–47, 2018.

[8] A. Bourrier, M. E. Davies, T. Peleg, P. Pérez, and R. Gribonval, "Fundamental performance limits for ideal decoders in high-dimensional linear inverse problems," *IEEE Transactions on Information Theory*, vol. 60, no. 12, pp. 7928–7946, 2014.

[9] S. M. Kay, *Fundermentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, 1993.

[10] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2008.

[11] J. Ranieri, A. Chebira, and M. Vetterli, "Near-optimal sensor placement for linear inverse problems," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1135–1146, 2014.

[12] C. Jiang, Y. C. Soh, and H. Li, "Sensor placement by maximal projection on minimum eigenspace for linear inverse problems," *IEEE Transactions on Signal Processing*, vol. 64, no. 21, pp. 5595–5610, 2016.

[13] C. Jiang, Z. Chen, R. Su, and Y. C. Soh, "Group greedy method for sensor placement," *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp. 2249–2262, 2019.

[14] F. Pukelsheim, *Optimal design of experiments*. SIAM, 1993, vol. 50.

[15] D. S. Bernstein, *Matrix Mathematics*. Princeton University Press, 2005.

[16] S. Liu, N. Cao, and P. K. Varshney, "Sensor placement for field estimation via poisson disk sampling," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016, pp. 520–524.

[17] F. Wang, G. Cheung, and Y. Wang, "Low-complexity graph sampling with noise and signal reconstruction via Neumann series," *IEEE Transactions on Signal Processing*, vol. 67, no. 21, pp. 5511–5526, 2019.

[18] A. Anis, A. El Gamal, A. S. Avestimehr, and A. Ortega, "A sampling theory perspective of graph-based semi-supervised learning," *IEEE Transactions on Information Theory*, vol. 65, no. 4, pp. 2322–2342, April 2019.

[19] R. A. Horn, R. A. Horn, and C. R. Johnson, *Matrix analysis*. Cambridge university press, 1990.

[20] H. Jamali-Rad, A. Simonetto, and G. Leus, "Sparsity-aware sensor selection: Centralized and distributed algorithms," *IEEE Signal Processing Letters*, vol. 21, no. 2, pp. 217–220, 2014.