DISTRIBUTED PREDICTIVE SUBSPACE PURSUIT

Dennis Sundman, Dave Zachariah, Saikat Chatterjee and Mikael Skoglund

School of Electrical Engineering and ACCESS Linnaeus Centre
KTH - Royal Institute of Technology, SE-10044 Stockholm, Sweden
denniss@kth.se, davez@kth.se, sach@kth.se, skoglund@kth.se

ABSTRACT

In a compressed sensing setup with jointly sparse, correlated data, we develop a distributed greedy algorithm called distributed predictive subspace pursuit. Based on estimates from neighboring sensor nodes, this algorithm operates iteratively in two steps: first forming a prediction of the signal and then solving the compressed sensing problem with an iterative linear minimum mean squared estimator. Through simulations we show that the algorithm provides better performance than current state-of-the-art algorithms.

***Index Terms*—** compressed sensing, greedy algorithms, prediction methods, distributed compressed sensing.

1. INTRODUCTION

Compressed sensing (CS) [1, 2] is a sub-sampling technique where the samples are formed through random linear combinations of an underlying measured signal which is inherently sparse. For a standard single sensor setup problem it has been shown that few random samples are sufficient for reconstruction of the full signal. In the literature there are three popular reconstruction approaches; greedy pursuit algorithms, convex relaxation and Bayesian methods. The greedy pursuit algorithms provide good performance at low computational cost [3, 4, 5, 6, 7].

In CS setups where several sensors are present, efficient reconstruction is possible by communicating and exploiting correlation among jointly sparse data from the different sensors. In the literature, several signal models have been presented for describing correlation structures, for example the common support-set model [8, 9], mixed support-set model [10], mixed signal-model [11]. There exist several centralized algorithms to exploit the correlation [12, 13]. We are particularly interested in algorithms that can solve the CS problem with correlated data from multiple measurements in a distributed manner. In a distributed setup, each sensor node exchanges data with its neighbors and individually (or locally) endeavors for a reliable CS reconstruction. There are various attempts to solve the distributed CS problem using convex relaxation based methods [14, 15, 16, 17]. To the best of the authors' knowledge there exists only one attempt to solve the distributed CS problem based on greedy pursuit methods [18]. While the distributed CS mainly considers spatial- and/or frequency domain correlations, there are some single sensor based algorithms that exploit correlation over time [19, 20, 21, 22].

In this paper, we consider the distributed CS problem where multiple sensor nodes observe sparse signals according to the common support-set model. In addition, we assume that the data from different sensors are correlated. We describe the signal model in detail in section 2.1. This model is applicable in scenarios where sensors track some analog signal, e.g. estimating a shared power spectrum based on samples from the air. For a distributed algorithm, it is important to know about the network topology among the sensor nodes; this is described in section 2.2. Based on the signal model and for a given network, we then develop a distributed algorithm that can solve the distributed CS problem using tools from [18] and applying the predictive subspace pursuit (PrSP) algorithm of [23]. Here we mention that the PrSP algorithm was developed by extending the greedy subspace pursuit (SP) algorithm [4]. We end the paper with some simulation results.

Notation: $\mathbf{A}^*$ and $\mathbf{A}^\dagger$ denote the Hermitian transpose and Moore-Penrose pseudoinverse of matrix $\mathbf{A}$, respectively. $\mathbf{A} \oplus \mathbf{B}$ denotes the direct sum of matrices. $\mathbf{A}_{[\mathcal{I},\mathcal{J}]}$ denotes a sub-matrix of $\mathbf{A}$ with elements from row and column indices listed in ordered sets $\mathcal{I}$ and $\mathcal{J}$, respectively. Similarly, the column vector $\mathbf{x}_{[\mathcal{I}]}$ contains the elements of $\mathbf{x}$ with indices from support set $\mathcal{I}$.

2. DISTRIBUTED COMPRESSED SENSING

We begin by describing the distributed CS problem in a general multiple sensor network setup [11]. We follow this by a description of the signal model. Without details, we mention network topology.

In the distributed CS problem, the sparse signal $\mathbf{x}_l \in \mathbb{C}^N$ at the l'th sensor is measured according to the linear model,

$$\mathbf{y}_l = \mathbf{H}_l \mathbf{x}_l + \mathbf{w}_l \in \mathbb{C}^M, \quad \forall l \in \{1, 2, ..., L\}, \tag{1}$$

where $\mathbf{H}_l \in \mathbb{C}^{M \times N}$ is a known measurement matrix and that $\mathbf{w}_l$ is some measurement noise. In this setup $M < N$ and hence the system is under-determined. The signal vector $\mathbf{x}_l = [x_l(1), \ldots, x_l(N)]$ has $K_l(< M < N)$ non-zero or 'active' components with a set of indices $\mathcal{I}_l = \{i : x_l(i) \neq 0\}$. $\mathcal{I}_l$ is referred to as the 'support set' of $\mathbf{x}_l$. The measurement noise $\mathbf{w}_l$ is assumed to be zero-mean and independent across all l.

The distributed CS reconstruction problem consists of estimating $\mathbf{x}_l$ for each l in a distributed manner. This is done by exploiting some common signal structure and exchanging the common information across the sensor network. The signal structure applied in this paper is described by the signal model in the next section.

2.1. Signal model

In this work, we assume that the set of sparse signals share the same support set and statistical properties across the network. More formally, this is modeled in the following way:

- All nodes share the same support $\mathcal{I}$, where $|\mathcal{I}| \leq K$. That is, $\forall l, \mathcal{I}_l = \mathcal{I}$ and $K_l \leq K$.
- The signal is assumed to be zero-mean, $\mathrm{E}[\mathbf{x}_l] = \mathbf{0}$.

This work was funded in part by VR (Swedish research council).

- The active components are uncorrelated at node l, $\mathrm{E}[\mathbf{x}_{[\mathcal{I}],l}\mathbf{x}^*_{[\mathcal{I}],l}] = \sigma_x^2\mathbf{I}_N$.
- Between nodes l and k, the active components are correlated, $\mathrm{E}[\mathbf{x}_{[\mathcal{I}],l}\mathbf{x}^*_{[\mathcal{I}],k}] = \rho\mathbf{I}_N$.

An example of a scenario that fits this distributed CS problem is a spectrum estimation problem in a cognitive radio network [24, 25]. In such a setup, each node may find common parts of the frequency band that are occupied by some primary transmitter. The amplitudes of occupied frequency bands may be subject to random variations of the local environment (i.e., scattering, fading) but are likely to be correlated. This correlation is in our model captured by the parameter ρ. Furthermore, we assume that $\mathrm{E}[\mathbf{w}_l\mathbf{w}_l^*] = \sigma_w^2\mathbf{I}_M$ and that $\mathbf{H}_l$ has incoherent columns such that $|\mathbf{h}^*_{l,i}\mathbf{h}_{l,j}| \ll |\mathbf{h}^*_{l,i}\mathbf{h}_{l,i}|$ [26]. Without loss of generality it is assumed that the columns have unit norm.

In summary, it is assumed that $\mathbf{H}_l$, K, σ_x^2, ρ and σ_w^2 are given. Next, we consider the properties of the sensor network.

2.2. Network topology

In a distributed setup, we assume that the CS nodes are connected in a joint network where there is at least one connection between any two nodes; otherwise the setup is equivalent to two, or more, disjoint networks. The information exchange between nodes is modeled by a directed graph, where each node l has a set of incoming neighbors, $\mathcal{N}_l^{\text{in}}$, from which it receives information, and a set of outgoing neighbors, $\mathcal{N}_l^{\text{out}}$, to which it transmits information. A thorough study of the network topology and the resulting estimation performance in various settings is beyond the scope of this paper. Instead, for a practical setup, we assume a network consisting of ten nodes and each node has two outgoing neighbors and two incoming neighbors arranged in such a way that the network is connected.

3. DISTRIBUTED PREDICTIVE SUBSPACE PURSUIT

In this work we develop a distributed subspace pursuit algorithm which by exchanging data among its neighbours iteratively improves an estimate of the sparse signal. In the first iteration of node l, local estimates $\hat{\mathbf{x}}_l$ are formed based on $\mathbf{y}_l$ and $\mathbf{H}_l$. Then, these local estimates are sent to the outgoing set of neighbors $\mathcal{N}_l^{\text{out}}$. The received information from the incoming neighbors $\mathcal{N}_l^{\text{in}}$ is used to form a prediction of the local signal $\mathbf{x}_l$. In the second iteration, this prediction is incorporated with the local measurement $\mathbf{y}_l$ by a predictive subspace pursuit to produce a new, improved, local estimate which is again shared over the network. Note that due to the sparse nature of the signal, the distributed algorithm at a sensor l only needs to send at most K integers and real valued numbers ($\widehat{\mathcal{I}}_l$ and $\hat{\mathbf{x}}_{[\mathcal{I}],l}$) at each transmission. Since the core of the distributed algorithm is the predictive subspace pursuit algorithm, we refer to the distributed algorithm by distributed predictive subspace pursuit (DPrSP). A detailed description of DPrSP is given in Section 3.3. But before we describe DPrSP we first describe the underlying PrSP sub-algorithm and formulate the predictor.

3.1. Predictive Subspace Pursuit

In this section, we drop the sub-index l for less notational clutter. The predictive subspace pursuit (PrSP) requires a prediction of the signal, denoted $\hat{\mathbf{x}}^-$, with error covariance matrix $\mathbf{P}^-$. This information is used along with a measurement $\mathbf{y}$ to detect the support set in a weighted matched filter framework, after which the active coefficients are reconstructed by a linear minimum mean square estimator (LMMSE). The PrSP algorithm effectively operates by solving the following optimization problem iteratively over support sets $\mathcal{I}$ according to the subspace pursuit method [4]:

$$\hat{\mathbf{x}}_{[\mathcal{I}],n} = \underset{\mathbf{x}_{[\mathcal{I}]}\in\mathbb{C}^{|\mathcal{I}|}}{\arg\min} \left\| \begin{bmatrix} \mathbf{y}_n \\ \hat{\mathbf{x}}^-_{[\mathcal{I}],n} \end{bmatrix} - \begin{bmatrix} \mathbf{H}_{[\cdot,\mathcal{I}]} \\ \mathbf{I}_{|\mathcal{I}|} \end{bmatrix} \mathbf{x}_{[\mathcal{I}]} \right\|^2_{\mathbf{R}^{-1}\oplus\mathbf{P}^{-1}_{[\mathcal{I},\mathcal{I}],n}} . \quad (2)$$

Due to spatial limitations we omit the details of PrSP and instead refer the reader to the previous work [23].

A function call to incorporate PrSP is defined as follows:

$$(\hat{\mathbf{x}}, \widehat{\mathcal{I}}, \eta) \leftarrow \text{PrSP}(\mathbf{y}, \mathbf{H}, \mathbf{R}, \hat{\mathbf{x}}^-, \mathbf{P}^-),$$

where $\mathbf{R}$ and $\eta = \|\mathbf{y} - \mathbf{H}_{[\cdot,\hat{\mathcal{I}}]}\hat{\mathbf{x}}_{[\hat{\mathcal{I}}]}\|^2$ denotes the noise covariance matrix and measurement residual norm, respectively.

When no side information is given from other nodes, the prediction is the prior $\hat{\mathbf{x}}^- = \mathbf{0}$ with $\mathbf{P}^- = \sigma_x^2\mathbf{I}_N$. With this prediction, the error statistics of the LMMSE employed by the nodes can be approximated. This allows each node l to use the received estimates from $\mathcal{N}_l^{\text{in}}$ as a prediction of its local signal $\mathbf{x}_l$, which is subsequently re-estimated. Next, we formulate a linear predictor based on received estimates.

3.2. Linear predictor

For node l, let $\{\mathbf{z}_n\}_{n=1}^T$ denote the set of received state estimates from incoming nodes $\mathcal{N}_l^{\text{in}}$, where $T = |\mathcal{N}_l^{\text{in}}|$. We now formulate the linear predictor that minimizes the mean square error of the prediction $\hat{\mathbf{x}}_l^-$ given $\{\mathbf{z}_n\}_{n=1}^T$.

Each $\mathbf{z}_n$, with detected support set $\mathcal{I}$, can be approximated by the expressions for the linear minimum mean square error estimator, $\mathbf{z}_n = \mathbf{K}_n\mathbf{y}_n$, where the rows of $\mathbf{K}_n \in \mathbb{C}^{N\times M}$ with indices $\mathcal{I}^c$ consisting of zeros. More specifically,

$$\begin{aligned} \mathbf{z}_{[\mathcal{I}],n} &= \mathbf{K}_{[\mathcal{I},\cdot]}\mathbf{y}_n \\ &= \sigma_w^{-2}\left(\sigma_x^{-2}\mathbf{I}_{|\mathcal{I}|} + \sigma_w^{-2}\mathbf{H}^*_{[\cdot,\mathcal{I}]}\mathbf{H}_{[\cdot,\mathcal{I}]}\right)^{-1}\mathbf{H}^*_{[\cdot,\mathcal{I}]}\mathbf{y}_n \end{aligned} \quad (3)$$

and $\mathbf{z}_{[\mathcal{I}^c],n} = \mathbf{K}_{[\mathcal{I}^c,\cdot]}\mathbf{y}_n = \mathbf{0}\mathbf{y}_n$. Furthermore, let $\mathbf{S}_n$ denote the covariance matrix of $\mathbf{z}_n$, then

$$\begin{aligned} \mathbf{S}_n &= \mathrm{E}[\mathbf{z}_n\mathbf{z}_n^*] \\ &= \mathrm{E}\left[\mathbf{K}_n(\mathbf{H}_n\mathbf{x}_n + \mathbf{w}_n)(\mathbf{H}_n\mathbf{x}_n + \mathbf{w}_n)^*\mathbf{K}_n^*\right] \\ &= \sigma_x^2\mathbf{K}_n\mathbf{H}_n\mathbf{H}_n^*\mathbf{K}_n^* + \sigma_w^2\mathbf{K}_n\mathbf{K}_n^*. \end{aligned} \quad (4)$$

Note that $\mathbf{S}_n$ has at most rank K. Expressions (3) and (4) provide a model of the received estimates at node l. The cross-correlation between $\{\mathbf{z}_n\}_{n=1}^T$ and the local signal $\mathbf{x}_l$, enables the formulation of the linear predictor.

By stacking the received estimates as $\mathbf{u} \triangleq \mathrm{vec}\left([\mathbf{z}_1, \ldots, \mathbf{z}_T]\right)$, the linear prediction equals [27]

$$\hat{\mathbf{x}}^- = \mathbf{R}_{xu}\mathbf{R}_u^\dagger\mathbf{u} \quad (5)$$

with approximate error covariance matrix

$$\mathbf{P}^- = \sigma_x^2\mathbf{I}_N - \mathbf{R}_{xu}\mathbf{R}_u^\dagger\mathbf{R}_{xu}^*. \quad (6)$$

Here

$$\begin{aligned} \mathbf{R}_{xu} &= \left[\mathrm{E}[\mathbf{x}_l\mathbf{z}_1^*] \quad \cdots \quad \mathrm{E}[\mathbf{x}_l\mathbf{z}_T^*]\right] \\ &= \left[\mathrm{E}[\mathbf{x}_l\mathbf{y}_1^*\mathbf{K}_1^*] \quad \cdots \quad \mathrm{E}[\mathbf{x}_l\mathbf{y}_T^*\mathbf{K}_T^*]\right] \\ &= \left[\rho\mathbf{H}_1^*\mathbf{K}_1^* \quad \cdots \quad \rho\mathbf{H}_T^*\mathbf{K}_T^*\right] \in \mathbb{C}^{N\times NT}, \end{aligned} \quad (7)$$

and

$$\mathbf{R}_u = \begin{bmatrix} \mathrm{E}[\mathbf{z}_1\mathbf{z}_1^*] & \cdots & \mathrm{E}[\mathbf{z}_1\mathbf{z}_T^*] \\ \vdots & \ddots & \vdots \\ \mathrm{E}[\mathbf{z}_T\mathbf{z}_1^*] & \cdots & \mathrm{E}[\mathbf{z}_T\mathbf{z}_T^*] \end{bmatrix} \in \mathbb{C}^{NT\times NT}, \tag{8}$$

where

$$\mathrm{E}[\mathbf{z}_m\mathbf{z}_n^*] = \begin{cases} \mathbf{S}_n & \text{for } m = n \\ \rho\mathbf{K}_m\mathbf{H}_m\mathbf{H}_n^*\mathbf{K}_n^* & \text{for } m \neq n \end{cases}. \tag{9}$$

Two problems impede the practicality of this estimator however. First, for large N and T the computational complexity may be prohibitive. Second, when the sensing matrices $\mathbf{H}_n$ differ across the network, it is impractical for each node to store the matrices of its incoming neighbours $\mathcal{N}^{\text{in}}$. To address these problems, we exploit incoherence of the sensing matrix by approximating $\mathbf{H}^*_{[\cdot,\mathcal{I}]}\mathbf{H}_{[\cdot,\mathcal{I}]} \approx \mathbf{I}_{|\mathcal{I}|}$. Then the matrices $\mathbf{R}_u$ and $\mathbf{R}_{xu}$ become sparse which reduces the computational complexity of the estimator drastically. As a result, the estimator model is approximated by

$$\begin{aligned} \mathbf{K}_{[\mathcal{I},\cdot]} &= \sigma_w^{-2}\left(\sigma_x^{-2}\mathbf{I}_{|\mathcal{I}|} + \sigma_w^{-2}\mathbf{H}^*_{[\cdot,\mathcal{I}]}\mathbf{H}_{[\cdot,\mathcal{I}]}\right)^{-1}\mathbf{H}^*_{[\cdot,\mathcal{I}]} \\ &\simeq \sigma_w^{-2}\left(\sigma_x^{-2}\mathbf{I}_{|\mathcal{I}|} + \sigma_w^{-2}\mathbf{I}_{|\mathcal{I}|}\right)^{-1}\mathbf{H}^*_{[\cdot,\mathcal{I}]} \\ &= \frac{\sigma_x^2}{\sigma_x^2+\sigma_n^2}\mathbf{H}^*_{[\cdot,\mathcal{I}]} \\ &= \kappa\mathbf{H}^*_{[\cdot,\mathcal{I}]}, \end{aligned} \tag{10}$$

where $\kappa \triangleq \frac{\sigma_x^2}{\sigma_x^2+\sigma_w^2}$. Further, let $\mathbf{G}(\mathcal{I}_n)$ be a diagonal matrix with elements,

$$g_{jj} = \begin{cases} 1 & \text{for } j \in \mathcal{I}_n \\ 0 & \text{otherwise} \end{cases}. \tag{11}$$

Then

$$\mathbf{K}_n\mathbf{H}_n \simeq \kappa\mathbf{G}(\mathcal{I}_n) \in \mathbb{C}^{N\times N},$$

and

$$\begin{aligned} \mathbf{S}_n &\simeq \sigma_x^2\kappa^2\mathbf{G}(\mathcal{I}_n)\mathbf{G}^*(\mathcal{I}_n) + \sigma_w^2\kappa^2\mathbf{G}(\mathcal{I}_n) \\ &= (\sigma_x^2+\sigma_w^2)\kappa^2\mathbf{G}(\mathcal{I}_n). \end{aligned}$$

These expressions enable the following approximations,

$$\mathbf{R}_{xu} \simeq \rho\kappa\begin{bmatrix}\mathbf{G}^*(\mathcal{I}_1) & \cdots & \mathbf{G}^*(\mathcal{I}_T)\end{bmatrix} \tag{12}$$

and

$$\mathbf{R}_u \simeq \begin{bmatrix} (\sigma_x^2+\sigma_w^2)\kappa^2\mathbf{G}(\mathcal{I}_1) & \cdots & \rho\kappa^2\mathbf{G}(\mathcal{I}_1)\mathbf{G}^*(\mathcal{I}_T) \\ \vdots & \ddots & \vdots \\ \rho\kappa^2\mathbf{G}(\mathcal{I}_T)\mathbf{G}^*(\mathcal{I}_1) & \cdots & (\sigma_x^2+\sigma_w^2)\kappa^2\mathbf{G}(\mathcal{I}_T) \end{bmatrix}, \tag{13}$$

which are sparse matrices that dispel the need for storing sensing matrices $\mathbf{H}_n$. We summarize the approximate optimal linear predictor in function $\texttt{linpred}(\cdot)$.

Function 1 *(Linear predictor) Given a set of received estimates* $\{\mathbf{z}_n\}_{n=1}^T$ *and the signal parameters* σ_x, ρ *and* σ_w *at node* l, *the following function*

$$(\hat{\mathbf{x}}_l^-, \mathbf{P}_l^-,) \leftarrow \mathtt{linpred}(\sigma_x, \rho, \sigma_w, \{\mathbf{z}_n\}_{n=1}^T),$$

computes its output according to (5) *and* (6), *using approximations* (12) *and* (13).

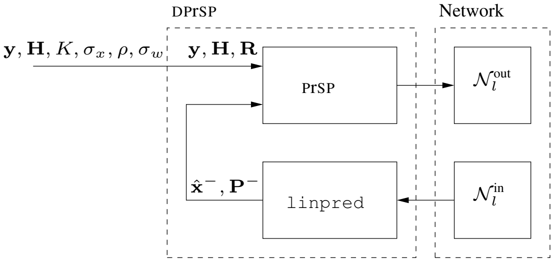
DPrSP
Network
$\mathbf{y}, \mathbf{H}, K, \sigma_x, \rho, \sigma_w$
$\mathbf{y}, \mathbf{H}, \mathbf{R}$
PrSP
$\mathcal{N}_l^{\text{out}}$
$\hat{\mathbf{x}}^-, \mathbf{P}^-$
linpred
$\mathcal{N}_l^{\text{in}}$

Fig. 1. flowchart of DPrSP

3.3. Decentralized estimation

Using PrSP in conjunction with the linear predictor formulated in previous section, we are now ready to develop the DPrSP algorithm which is executed in each node of a sensor network after a measurement is obtained. Algorithm 1 provides a summary; also see Figure 1 for a graphical view.

Algorithm 1 Distributed Predictive Subspace Pursuit
Executed in the l'th node with $\mathcal{N}_l^{in}$ *as incoming neighbours and* $\mathcal{N}_l^{out}$ *as outgoing neighbours*

```
Input: $\mathbf{y}_l$, $\mathbf{H}_l$, $K$, $\sigma_x$, $\rho$, $\sigma_w$
Initialization:
  1: $\hat{\mathbf{x}}_l^- \leftarrow \mathbf{0}$,  $\mathbf{P}_l^- \leftarrow \sigma_x^2\mathbf{I}_N$,  $\mathbf{R}_l \leftarrow \sigma_w^2\mathbf{I}_M$
  2: $\eta_l \leftarrow \infty$,  $\eta_l' \leftarrow \eta_l$,  $T = |\mathcal{N}^{\text{in}}|$
Iteration:
  1: repeat
  2:     $(\hat{\mathbf{x}}_l', \hat{\mathcal{I}}_l', \eta_l') \leftarrow (\hat{\mathbf{x}}_l, \hat{\mathcal{I}}_l, \eta_l)$
  3:     $(\hat{\mathbf{x}}_l, \hat{\mathcal{I}}_l, \eta_l) \leftarrow$ PrSP$(\mathbf{y}_l, \mathbf{H}_l, \mathbf{R}_l, \hat{\mathbf{x}}_l^-, \mathbf{P}_l^-)$
  4:     Transmit: $\hat{\mathbf{x}}_l$ to neighbours $n \in \mathcal{N}_l^{\text{out}}$
  5:     Receive: $\hat{\mathbf{x}}_n$ as $\mathbf{z}_n$ from neighbours $n \in \mathcal{N}_l^{\text{in}}$
  6:     $(\hat{\mathbf{x}}_l^-, \mathbf{P}_l^-) \leftarrow \mathtt{linpred}(\sigma_x, \rho, \sigma_w, \{\mathbf{z}_n\}_{n=1}^T)$
  7: until $(\eta_l \geq \eta_l')$
Output: $\hat{\mathbf{x}}_l$
```

Input to the algorithm is the measurement $\mathbf{y}_l$ and prior knowledge $\mathbf{H}_l$, K, σ_x, ρ, σ_w. In the Initialization phase, the prior estimate $\hat{\mathbf{x}}_l = \mathbf{0}$ is formed. The residual norm η is also initialized and is used in the algorithm as a stopping criterion. For all variables in the algorithm, the prime $'$ is used as symbol for the previous iterations values, for example is η' the value of η from the previous iteration.

In step 3 of the iteration the algorithm in node l forms an estimate $\hat{\mathbf{x}}_l$ based on the measurement $\mathbf{y}_l$ and prior prediction $\mathbf{x}_l^-$, $\mathbf{P}_l^-$, using the underlying predictive subspace pursuit algorithm, PrSP. In step 4, this estimate is then transmitted to outgoing neighbour-nodes $\mathcal{N}_l^{\text{out}}$. In step 5, the node similarly receives estimates from its incoming set of neighbours $\mathcal{N}_l^{\text{in}}$ which are locally collected in vectors denoted by $\mathbf{z}_n$, where $n \in \mathcal{N}_l^{\text{in}}$. In step 6, based on the received signal estimates, the node uses the $\texttt{linpred}$ to form a prediction, which in the next iteration is used as prior knowledge for PrSP.

The algorithm stopping criterion is based on that the residual norm in the l'th node does not decrease.

4. SIMULATIONS

In the simulations we are interested in comparing DPrSP with SP and previous distributed greedy pursuit algorithm DiSP [18]. As a

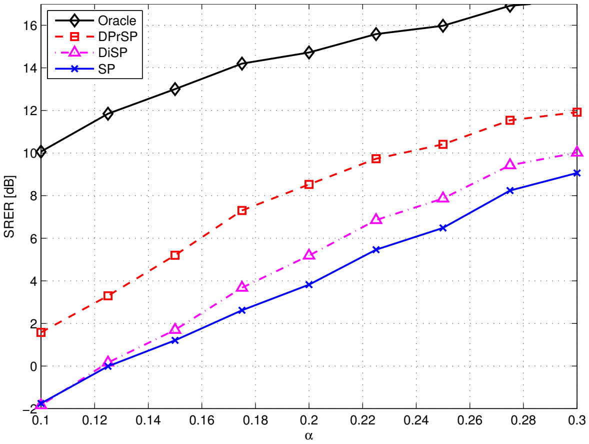

Fig. 2. SRER vs α for DPrSP, SMNR = 10 dB, $\rho = 0.9$

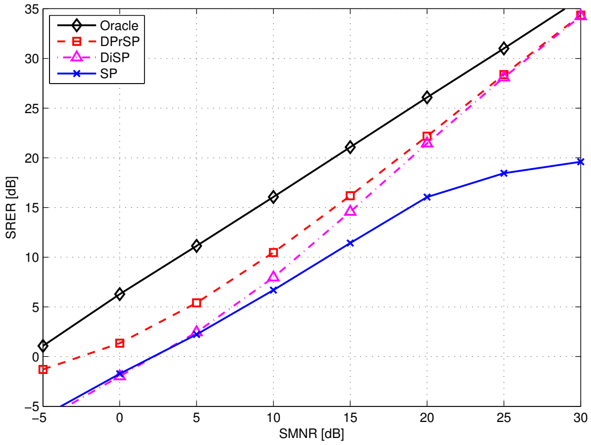

Fig. 3. SRER vs SMNR for $\alpha = 0.25$, $\rho = 0.9$

reference we also show an oracle which, based on knowing the true support-set uses the least-squares approach to find its solution.

All simulations contain noise, described by the signal-to-measurement-noise-ratio (in dB), SMNR $= 10\log_{10}\frac{\mathbb{E}\{\|\mathbf{x}\|_2^2\}}{\mathbb{E}\{\|\mathbf{w}\|_2^2\}} = 10$. Note that we drop the subscript l because we are averaging over all nodes l. To compare the algorithms, the performance measure chosen is the signal-to-reconstruction-error-ratio (in dB) SRER $= 10\log_{10}\frac{\mathbb{E}\{\|\mathbf{x}\|_2^2\}}{\mathbb{E}\{\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2\}}$. We emulate a connected network with a binary connection matrix $\mathbf{C}$, where a "1" in c_{ij} corresponds to a connection from node i to node j. In any CS setup, all sparse signals are expected to be exactly reconstructed if the number of measurements are more than a certain threshold value. The computational complexity to test this uniform reconstruction ability is exponentially high. Instead, we can rely on empirical testing, where SRER is computed for random measurement matrix ensemble. We define the fraction of measurements as $\alpha = \frac{M}{N}$.

Based on the performance SRER, we show the performance for different values of α, SMNR by simulations as follows:

1. Fix the connection matrix $\mathbf{C}$ so that each node has two incoming and two outgoing neighbours.
2. Randomly generate:
   - A set of $M \times N$ sensing matrices $\{\mathbf{A}_l\}_{l=1}^{L}$ where the components are drawn from an i.i.d. Gaussian source ($a_{m,n} \sim \mathcal{N}\left(0, \frac{1}{M}\right)$) and scale the columns of $\mathbf{A}_l$ to unit-norm.
   - Support-sets $\mathcal{I}$ of cardinality K. The support-sets are uniformly chosen from $\{1, 2, ..., N\}$.
   - A set of signal vectors $\{\mathbf{x}_l\}_{l=1}^{L}$ where the non-zero components in the vectors are chosen at random according to the signal model in section 2.1 so that the components are uncorrelated at node l, but correlated between all different nodes, with correlation parameter ρ.
3. Compute the measurements $\mathbf{y}_l = \mathbf{A}_l\mathbf{x}_l + \mathbf{w}_l, \forall l \in \{1, 2, ..., L\}$. Here $\mathbf{w}_l \sim \mathcal{N}(\mathbf{0}, \sigma_l^2\mathbf{I}_M)$.
4. Apply the CS algorithms on the data $\{\mathbf{y}_l\}_{l=1}^{L}$. The connection matrix $\mathbf{C}$ is used to distribute the data in the network.

We generate $2 \cdot 10^2$ sensing matrices and $2 \cdot 10^2$ signal vectors. In all simulations we have a network consisting of ten sensor nodes with two outgoing and two incoming connections. Thus, for the plots, each data-point is evaluated over $4 \cdot 10^5$ realizations.

4.1. Results

For all figures presented here, we are using the common support-set model with correlated data, as described in section 2.1. The DiSP algorithm which we compare against is a more general algorithm and can also operate in other scenarios. Furthermore, the network is restrictively connected, as described in section 2.2 and we may experience better performance with a better connected network. All gains in performance for any user l for DPrSP comes at the price of K integers and real valued numbers ($\hat{\mathcal{I}}_l$ and $\hat{\mathbf{x}}_{[\mathcal{I}],l}$) at each transmission.

In Figure 2, we study how the DPrSP algorithm performs when increasing the number of measurements for each node. The SMNR is set to 10 dB, and we notice a clear increase of performance for DPrSP compared to both a disconnected solution (SP) and the competing DiSP algorithm. At $\alpha = 0.24$, DPrSP performs about 4 dB better than SP and about 2.5 dB better than DiSP. The reason DPrSP outperforms DiSP is because it combats the measurement-noise better since it forms an estimate based on data from many sensor nodes with independent noise. All algorithms scale with increasing α.

In Figure 3, we further see how DPrSP handles noise. For all regions depicted here, it is clear that DPrSP always outperforms SP. What is particularly interesting, however, is the performance gain over DiSP that is achieved in the lower SMNR regions. At SMNR $= 5$, DPrSP is almost 3 dB better than DiSP. At higher SMNR the performance of DPrSP and DiSP converge towards the oracle bound.

While running the simulations we have tried other values of ρ. Due to space limitation we do not show these results here, but we noticed that the algorithm performs well also for smaller values of ρ.

5. RELATION TO PRIOR WORK AND CONCLUSION

We derive a predictor for incorporating the PrSP (from prior work [23]) algorithm to solve a distributed CS (as in prior work [18]) setup where the underlying model is based on the common support-set model (defined in [11]) and the assumption that the data is correlated among the sensor nodes. Based on this predictor we develop a new distributed algorithm that provides improved performance compared to the current distributed greedy search algorithm DiSP. This improved performance is possible by exploiting correlation in the measured signal. The communication overhead for the DPrSP algorithm is slightly higher than that of DiSP although still low.

6. REFERENCES

[1] D.L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, pp. 1289–1306, Apr. 2006.

[2] E.J. Candès and M.B. Wakin, "An introduction to compressive sampling," *IEEE Electron Device Letters*, vol. 25, pp. 21–30, Mar. 2008.

[3] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, pp. 301–321, Apr. 2009.

[4] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, pp. 2230–2249, May 2009.

[5] J.A. Tropp and A.C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, pp. 4655–4666, Dec. 2007.

[6] S. Chatterjee, D. Sundman, M. Vehkaperä, and M.Skoglund, "Projection-based and look-ahead strategies for atom selection," *IEEE Transactions on Signal Processing*, vol. 60, pp. 634–647, Feb. 2012.

[7] D. Sundman, S. Chatterjee, and M. Skoglund, "Look ahead parallel pursuit," in *IEEE Swedish Communication Technologies Workshop (Swe-CTW)*, Stockholm, Sweden, Oct. 2011, pp. 114–117.

[8] S.F. Cotter, B.D. Rao, Kjersti Engan, and K. Kreutz-Delgado., "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Transactions on Signal Processing*, vol. 53, pp. 2477–2488, July 2005.

[9] R. Gribonval, H. Rauhut, K. Schnass, and P. Vandergheynst, "Atoms of all channels, unite! average case analysis of multichannel sparse recovery using greedy algorithms," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 655–687, Dec. 2008.

[10] D. Sundman, S. Chatterjee, and M. Skoglund, "Greedy pursuits for compressed sensing of jointly sparse signals," in *EURASIP European Signal Processing Conference (EUSIPCO)*, Barcelona, Spain, Aug. 2011, pp. 368–372.

[11] M.F. Duarte, S. Sarvotham, D. Baron, M.B. Wakin, and R.G. Baraniuk, "Distributed compressed sensing of jointly sparse signals," in *Annual Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, California, Nov. 2005, pp. 1537–1541.

[12] J.A. Tropp, A.C. Gilbert, and M.J. Strauss, "Simultaneous sparse approximation via greedy pursuit," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Philadelphia, USA, Mar. 2005, pp. 721–724.

[13] J. Chen and X. Huo, "Theoretical results on sparse representations of multiple-measurement vectors," *IEEE Transactions on Signal Processing*, vol. 54, pp. 4634–4643, Dec. 2006.

[14] F. Zeng, C. Li, and Z. Tian, "Distributed compressive spectrum sensing in cooperative multihop cognitive networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, pp. 37–48, Feb. 2011.

[15] J.A. Bazerque and G.B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Transactions on Signal Processing*, vol. 58, pp. 1847–1862, Mar. 2010.

[16] J.F.C. Mota, J.M.F. Xavier, P.M.Q. Aguiar, and M. Puschel, "Distributed basis pursuit," *IEEE Transactions on Signal Processing*, vol. 60, pp. 1942–1956, Apr. 2012.

[17] Q. Ling and T. Zhi, "Decentralized support detection of multiple measurement vectors with joint sparsity," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. 2996–2999.

[18] D. Sundman, S. Chatterjee, and M. Skoglund, "A greedy pursuit algorithm for distributed compressed sensing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 2729–2732.

[19] N. Vaswani, "Kalman filtered compressed sensing," in *Proc. IEEE International Conference on Image Processing (ICIP)*, San Diego, California, USA, Oct. 2008, pp. 893–896.

[20] A. Carmi, P. Gurfil, and D. Kanevsky, "Methods for sparse signal recovery using kalman filtering with embedded pseudo-measurement norms and quasi-norms," *IEEE Transactions on Signal Processing*, vol. 58, pp. 2405–2409, Apr. 2010.

[21] Z. Zhang and B.D. Rao, "Sparse signal recovery in the presence of correlated multiple measurement vectors," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, Texas, USA, Mar. 2010, pp. 3986–3989.

[22] D. Zachariah, S. Chatterjee, and M. Jansson, "Dynamic iterative pursuit," *IEEE Transactions on Signal Processing*, vol. 60, pp. 4967–4972, Sept. 2012.

[23] D. Zachariah, S. Chatterjee, and M. Jansson, "Dynamic subspace pursuit," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 3605–3608.

[24] J. Mitola III and G.Q. Maguire Jr., "Cognitive radio: making software radios more personal," *IEEE Personal Communications Magazine*, vol. 6, pp. 13–18, Aug. 1999.

[25] D. Sundman, S. Chatterjee, and M. Skoglund, "On the use of compressive sampling for wide-band spectrum sensing," in *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Luxor, Egypt, Dec. 2010, pp. 354–359.

[26] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, Haifa, Israel, 2010.

[27] T. Kailath, A.H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, Stanford, California, USA, 2000.