Compressive Sensing by Shortest-Solution Guided Decimation

Mutian Shen\textsuperscript{3}, Pan Zhang\textsuperscript{1,4}, and Hai-Jun Zhou\textsuperscript{1,2,4}\textsuperscript{†}

\textsuperscript{1}Key Laboratory for Theoretical Physics, Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100190, China
\textsuperscript{2}School of Physical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
\textsuperscript{3}School of the Gifted Young, University of Science and Technology of China, Hefei 230026, China
\textsuperscript{4}Synergetic Innovation Center for Quantum Effects and Applications, Hunan Normal University, Changsha, Hunan 410081, China

(Dated: 25 September, 2017 (version 1); 05 October, 2017 (version 2))

Compressed sensing, which reconstructs a signal by finding sparse solutions to underdetermined linear systems, is an important problem in many fields of science and engineering. In this work we propose a deterministic and non-parametric algorithm, Shortest-Solution guided Decimation (SSD), to construct support of the sparse solution in an iterative way guided by the shortest Euclidean-length solution of the recursively decimated linear equation under orthogonal projections. The most significant feature of SSD is its insensitivity to correlations in the sampling matrix. Using extensive numerical experiments we show that SSD greatly outperforms \( \ell_1 \)-based methods, Orthogonal Matching Pursuit, and Approximate Message Passing when the sampling matrix contains strong correlations. This nice property of correlation tolerance makes SSD a versatile and robust tool for different types of real-world signal acquisition tasks.

I. INTRODUCTION

Real-world signals such as images, voice streams and text documents are highly compressible due to their intrinsic sparsity. Recent intensive efforts from diverse fields (computer science, engineering, mathematics and physics) have established the feasibility of merging data compression with data acquisition to achieve high efficiency of sparse information retrieval \([1, 2]\). This integrated signal processing framework is called compressed sensing or compressed sampling \([3, 5]\). At the core of this concept is an underdetermined linear equation

\[
Dh = z ,
\]

with \( D \) being an \( M \times N \) real matrix \( (M < N) \), \( h \equiv (h_1, h_2,\ldots,h_N)^T \) an \( N \)-dimensional real column vector, and \( z \equiv (z_1, z_2,\ldots,z_M)^T \) an \( M \)-dimensional real column vector. (Throughout this paper we use uppercase and lowercase bold letters to denote matrices and vectors, respectively.) \( z \) is the result of \( M \) sampling operations (measurements) on an \( N \)-dimensional hidden signal \( h^0 \); in matrix form, this is

\[
z \equiv Dh^0 .
\]

\( h^0 \equiv (h_1^0, h_2^0,\ldots,h_N^0)^T \) is referred to as a planted solution of \((1)\). Given a matrix \( D \) and an observed vector \( z \), the task is to reconstruct the planted solution \( h^0 \).

The sampling process \((2)\) has compression ratio \( \alpha \equiv M/N < 1 \). If the number \( N_{nz} \) of non-zero entries in \( h^0 \) exceeds \( M \), some information must be lost in compression and it is then impossible to completely recover \( h^0 \) from \( z \).

However, if the sparsity \( \rho \equiv N_{nz}/N \) is sufficiently below the compression ratio \( (\rho < \alpha) \), \( h^0 \) can be faithfully recovered by treating \((1)\) as the sparse representation problem of obtaining a solution \( h \) with the least number of non-zero entries \([6, 7]\).

As \( \ell_0 \)-norm minimization is intractable, many different heuristic ideas have instead been explored for this sparse recovery task \([1, 7]\). These empirical methods form three major clusters: greedy deterministic algorithms for \( \ell_0 \)-minimization, convex relaxations such as \( \ell_1 \)-based methods, and physics-inspired message-passing methods for approximate Bayesian inference. Representative and most popular algorithms of these categories are Orthogonal Matching Pursuit (OMP) \([8, 13]\), \( \ell_1 \)-based methods \([3, 6, 16–19]\), and Approximate Message Passing (AMP) \([20, 21]\), respectively. (In statistical physics the AMP method is known as the Thouless-Anderson-Palmer equation \([22]\).) To achieve good reconstruction performance, these algorithms generally assume that the sampling matrix \( D \) has the restricted isometric property (RIP) \([23]\). This requires \( D \) to be sufficiently random, uncorrelated and incoherent, which is not necessarily easy to meet in real-world applications. In many situations the matrix \( D \) is intensionally designed to be highly structured. For instance, \( D \) in the closely related dictionary learning problem often contains several complete sets of orthogonal base column vectors to increase the representation capacity, leading to considerable correlations between the columns \([6, 16]\). An efficient and robust algorithm applicable for correlated sampling matrices is indeed highly desirable.

In this work we introduce a simple deterministic algorithm, Shortest-Solution guided Decimation (SSD), that is rather insensitive to the statistical property of the sampling matrix \( D \). This algorithm has the same structure as the celebrated OMP algorithm \([8, 13]\). It selects a single column of \( D \) at each iteration step. One of the ma-
or differences is on how to choose the matrix columns: While OMP selects the column that has the largest magnitude of inner product with the residual of vector \( \mathbf{z} \), SSD selects a column according to the shortest Euclidean-length solution of the linear equation (which is recursively decimated under orthogonal projections).

On random uncorrelated matrices SSD works better than OMP and \( \ell_1 \)-based methods, but it is still worse than AMP which for uncorrelated Gaussian matrices is proven to be Bayes-optimal under correct priors \([20, 24]\).

We demonstrate that, since SSD exploits the singular value spectrum of the decimated linear equation, its performance is unaffected by the correlations in the matrix as opposed to other popular methods such as \( \ell_1 \)-based methods, OMP and AMP, which fail completely for highly correlated matrices. This remarkable feature makes SSD a versatile and robust tool for different types of practical compressed sensing and sparse approximation tasks.

II. THE GENERAL DENSE SOLUTION

When \( \alpha < 1 \) so the number \( M \) of constraints is less than the number of degrees of freedom \( N \), the linear equation \((1)\) has infinitely many solutions and most of them are dense (all the entries of \( \mathbf{h} \) are non-zero). To help searching for sparse solutions let us first derive a general solution for this equation.

By singular value decomposition (SVD), the sampling matrix can be expressed as a product of three matrices:

\[
\mathbf{D} = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^T.
\]

Here \( \mathbf{U} \) is an \( M \times M \) orthonormal matrix, \( \mathbf{U} \mathbf{U}^T = \mathbf{I}_{M \times M} \) with \( \mathbf{I}_{M \times M} \) being the \( M \)-dimensional identity matrix; \( \mathbf{\Lambda} \) is an \( M \times N \) diagonal matrix

\[
\mathbf{\Lambda} = \begin{bmatrix}
\lambda_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & \lambda_2 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 0 & \cdots & \cdots & \cdots & \cdots & 0
\end{bmatrix};
\]

and \( \mathbf{V} \) is an \( N \times N \) orthonormal matrix, \( \mathbf{V} \mathbf{V}^T = \mathbf{V}^T \mathbf{V} = \mathbf{I}_{N \times N} \). All the singular values \( \lambda_\mu (\mu = 1, 2, \ldots, M) \) in \( \mathbf{\Lambda} \) are non-negative and \( \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_M \). The columns of \( \mathbf{U} \equiv (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M) \) form a complete set of unit base vectors for the \( M \)-dimensional real space, namely the vectors \( \mathbf{u}_\nu \equiv (u_1^\nu, u_2^\nu, \ldots, u_M^\nu)^T \) satisfy \( \langle \mathbf{u}_\nu, \mathbf{u}_\nu' \rangle = \delta_\nu^\nu' \). Here and in following discussions \((a, b)\) denotes the inner product of two vectors \( \mathbf{a} \) and \( \mathbf{b} \) (e.g., \( \langle \mathbf{w}, \mathbf{v} \rangle \equiv \sum_{\mu} w_\mu v_\mu^* \)) and \( \delta_\nu^\nu' \) is the Kronecker symbol, \( \delta_\nu^\nu' = 0 \) for \( \mu \neq \nu \) and \( \delta_1^1 = 1 \) for \( \nu = 1 \). Similarly, the orthonormal matrix \( \mathbf{V} \equiv (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N) \) is formed by a complete set of unit base vectors \( \mathbf{v}_\nu \equiv (v_1^\nu, v_2^\nu, \ldots, v_N^\nu)^T \) for the \( N \)-dimensional real space, with \( \langle \mathbf{v}_\nu, \mathbf{v}_\nu' \rangle = \delta_\nu^\nu' \).

Expanding the observed vector \( \mathbf{z} \) as a linear combination of base vectors \( \mathbf{u}_\nu \), we see that a general solution of \((1)\) is

\[
\mathbf{h} = \mathbf{g} + \sum_{\nu=M+1}^{N} c_\nu \mathbf{v}_\nu.
\]

Each of the \( N - M \) coefficients \( c_\nu \) is a free parameter that can take any real value; \( \mathbf{g} \equiv (g_1, g_2, \ldots, g_N)^T \) is a \( N \)-dimensional column vector uniquely determined by \( \mathbf{z} \):

\[
\mathbf{g} = \sum_{\mu=1}^{M} \Theta(\lambda_\mu) \frac{\langle \mathbf{z}, \mathbf{u}_\mu \rangle}{\lambda_\mu} \mathbf{u}_\mu \equiv \mathbf{D}^+ \mathbf{z},
\]

where \( \Theta(x) \) is the Heaviside function, \( \Theta(x) = 1 \) for \( x > 0 \) and \( \Theta(x) = 0 \) for \( x \leq 0 \); \( \mathbf{D}^+ \) is the (Moore-Penrose) pseudo-inverse of \( \mathbf{D} \). We call \( \mathbf{g} \) the guidance vector. We will rank the \( N \) entries of \( \mathbf{g} \) in descending order of their magnitude, and the index \( l \in \{1, 2, \ldots, N\} \) corresponding to the maximum absolute entry \( |g_l| \) of \( \mathbf{g} \) will be referred to as the leading index.

The guidance vector \( \mathbf{g} \) is perpendicular to each of the \( N - M \) base vectors \( \mathbf{v}_\nu \) with index \( \nu > M \). The remainder term \( \mathbf{r} \equiv \sum_{\nu>M} c_\nu \mathbf{v}_\nu \) of the general solution \((5)\) therefore is perpendicular to \( \mathbf{g} \). A simple geometric interpretation of \((5)\) is illustrated in Fig. 1. The solution \( \mathbf{h} \) can move freely within the \((N-M)\)-dimensional sub-space spanned by the base vectors \( \mathbf{v}_\nu \) with index \( \nu > M \) (it is called the null space or the kernel of \( \mathbf{D} \)); this subspace is perpendicular to the guidance vector \( \mathbf{g} \), which itself is a vector within the \( M \)-dimensional subspace spanned by the base vectors \( \mathbf{v}_\nu \) with \( \mu \leq M \).

FIG. 1: Geometric interpretation on the general solution \((5)\). The solution \( \mathbf{h} \equiv (h_1, h_2, \ldots, h_N)^T \) is the sum of two parts, the guidance vector \( \mathbf{g} \equiv (g_1, g_2, \ldots, g_N)^T \) and the remainder \( \mathbf{r} \). \( \mathbf{g} \) lies within a \( M \)-dimensional subspace, while \( \mathbf{r} \) lies within the \((N-M)\)-dimensional null space of matrix \( \mathbf{D} \). The null space (signified by the elliptic shaded area) is perpendicular to \( \mathbf{g} \). Each dashed line represents one of the \( N \) coordinate axes. In this schematic figure \( g_1 \) has the largest magnitude among all the entries of \( \mathbf{g} \), so the leading index \( l = 1 \). The sparsest solution \( \mathbf{h} \) of \((1)\) is very likely to have large projection on the leading coordinate axis \( l \) (the filled circular points are examples of such candidate solutions), while it is unlikely to have zero projection on the coordinate axis \( l \) (the filled square points are examples of these unlikely solutions).
Notice that the guidance vector $g$ is nothing but the shortest Euclidean-length solution. That is, the $\ell_2$-norm $\|g\|_2 \equiv \sqrt{\sum_j g_j^2}$ achieves the minimum value among all the solutions of (1). This is easily proved by setting all the coefficients $c_\nu$ of (5) to zero. Therefore we may compute $g$ by the LQ decomposition method which is more economic than SVD. Through LQ decomposition we can express the sampling matrix as $D = LQ^T$, with $Q \equiv (q^1, q^2, \ldots, q^M)$ being an $N \times M$ orthogonal matrix (the column vectors $q^\mu$ satisfying $(q^\mu, q^{\nu'}) = \delta^\mu_{\nu'}$ for $\mu, \nu \in \{1, 2, \ldots, M\}$) and $L$ being an $M \times M$ lower triangular matrix. The guidance vector $g$ is then a linear combination of the unit vectors $q^\mu$:

$$g = \sum_{\mu=1}^M a_\mu q^\mu,$$

and the coefficient vector $a \equiv (a_1, a_2, \ldots, a_M)^T$ is easily fixed by solving the lower-triangular linear equation

$$La = z .$$

\section{III. THE GUIDANCE VECTOR AS A CUE}

The guidance vector $g$, being the shortest (Euclidean-length) solution of (1), is dense and is not the planted solution $h^0$ we are aiming to reconstruct. But we ask an interesting question: Does $g$ bring some reliable clues on which entries of $h^0$ are highly likely to be non-zero? To explore this question we need to know how $g$ is related to $h^0$. Since the observed vector $z$ encodes the information of $h^0$ through $z = Dh^0$, we get from (10) that

$$g = \sum_{\mu=1}^M (h^0, v^\mu) v^\mu \equiv Ph^0 .$$

Here the $N \times N$ matrix $P$ is a projection operator:

$$P \equiv \sum_{\mu=1}^M (v^1, v^\mu) (v^1, v^\mu)^T (v^1, v^2, \ldots, v^N) .$$

It projects $h^0$ to the subspace spanned by the first $M$ base vectors $v^\mu$ ($1 \leq \mu \leq M$) of matrix $V$. A diagonal entry of $P$ is $P_{ii} = \sum_{\mu=1}^M (v^i, v^\mu)^2$, while the expression for an off-diagonal entry is $P_{ij} = \sum_{\mu=1}^M v^i_\mu v^j_\mu$.

Notice that, had the summation in (10) included all the $N$ base vectors, $P$ would have been the identity matrix ($P_{ij} = \delta^i_j$), and then $g$ would have been identical to $h^0$. Because only $M$ base vectors are included in (10) the non-diagonal entries of $P$ no longer vanish, but their magnitudes are still markedly smaller than those of the diagonal entries. Since $v^\mu$ is a unit vector we may assume that $v^\mu_i \equiv \pm \frac{1}{\sqrt{N}}$, then we estimate $P_{ii} \approx \frac{M}{N} = \alpha$. Assuming $v^\mu_i$ and $v^\mu_j$ to be largely independent of each other, we estimate that $P_{ij} \approx \pm \frac{\sqrt{\alpha}}{\sqrt{N}}$, or $P_{ij} \approx \pm \frac{\sqrt{\alpha}}{\sqrt{N}}\sqrt{\rho N} a_0 \approx \pm \sqrt{\alpha \rho a_0}$, where $a_0 \equiv \sqrt{\frac{1}{\rho N} \sum_{i=1}^N (h^0_i)^2}$ is the expected value of the non-zero entries of $h^0$. Notice that $g_i^{(b)}$ does not depend much on the index $i$ and it is of the same order as the first term $g_i^{(a)}$ (or $\alpha h_i^0$) of (11). For the leading index $i$ of $g$, since $g_i$ must have the maximum magnitude among all the entries $g_i$, we expect that $g_i^{(a)}$ will have the same sign as $g_i^{(b)}$ and it will have considerably large magnitude. It follows that $h_i^0$ is very likely to be non-zero and also that $|h_i^0| \gtrsim a_0$.

The above theoretical analysis suggests that the vector $g$ is very helpful for us to guess which entries of the planted solution $h^0$ have large magnitudes. The validity of this theoretical insight has been confirmed by our numerical simulation results (Fig. 2). We find that indeed the guidance vector $g$ contains valuable clues about the non-zero entries of $h^0$. If an index $i$ is ranked on the top with respect to the magnitude of $g_i$, the corresponding

FIG. 2: The guidance vector $g$ reveals which entries of the planted solution $h^0$ are most likely to be non-zero. The underlying Gaussian sampling matrices $D$ have compression ratio $\alpha = 0.2$. (a): Rank curves for three guidance vectors $g$ corresponding to three $h^0$ samples containing $\rho N$ non-zero entries with $\rho = 0.05, 0.1$ and 0.2, respectively. Dimension of $h^0$ is $N = 10^3$ and the $N$ indexes are ranked according to the entry magnitudes $|g_i|$ of $g$. The corresponding curves in (b) show the fractions $f(n)$ of non-zero entries of $h^0$ among the $n$ top-ranked indexes. (c): Probability $p_1$ of $h^0$ being non-zero $(l$ being the leading index of $g)$. $p_1$ is evaluated over $10^4$ random $h^0$ samples, each of which containing $\rho N$ non-zero entries uniformly distributed in $(-1, +1)$. The dimensionality is $N = 10^3$ (plus symbols) or $N = 10^4$ (cross symbols). As a comparison, the dashed line shows the probability of an entry of $h^0$ being non-zero if it is chosen uniformly at random among all the $N$ entries.
value $h_0^0$ has a high probability to be non-zero (Fig. 2a and 2b). This is especially true for the leading index $l$ of $g$. We also observe that, both for $\rho < \alpha$ and for $\rho \approx \alpha$ (Fig. 2c), the probability of $h_0^0$ being non-zero becomes more and more close to unity as the size $N$ increases. This indicates the cue offered by $g$ is more reliable for larger-sized compressed sensing problems.

Encouraged by these initial results, we now proceed to propose a simple algorithm for solving equation (1).

IV. SHORTEST-SOLUTION GUIDED DECIMATION (SSD)

Let us denote the sampling matrix as a collection of column vectors, $D \equiv (d^1, d^2, \ldots, d^N)$ with $d^\mu$ being the $\mu$-th column. With respect to the leading index $l$ of $g$ the linear equation (1) is rewritten as

$$h_0^l, d^l + \sum_{\mu \neq l} h_\mu d^\mu = z .$$

(12)

Therefore, if all the other $N-1$ entries $h_\mu$ of the vector $h$ are known, $h_l$ is uniquely determined as

$$h_l = \frac{\langle z, d^l \rangle}{\langle d^l, d^l \rangle} - \sum_{\mu \neq l} h_\mu \frac{\langle d^\mu, d^l \rangle}{\langle d^\mu, d^\mu \rangle} .$$

(13)

Let us denote by $h^{\mathcal{V}} = (h_1, \ldots, h_{l-1}, h_{l+1}, \ldots, h_N)^T$ the $(N-1)$-dimensional vector formed by deleting $h_l$ from $h$. This vector must satisfy the following linear equation

$$B h^{\mathcal{V}} = y .$$

(14)

$B \equiv (b^1, \ldots, b^{l-1}, b^{l+1}, \ldots, b^N)$ is an $M \times (N-1)$ matrix decimated from $D$ with its column vectors $b^\nu$ being

$$b^\nu \equiv d^\nu - \frac{\langle d^\nu, d^l \rangle}{\langle d^l, d^l \rangle} d^l \quad (\mu \neq l) .$$

(15)

The $M$-dimensional vector $y$ is the residual of $z$:

$$y \equiv z - \frac{\langle z, d^l \rangle}{\langle d^l, d^l \rangle} d^l .$$

(16)

This residual vector $y$ and all the decimated column vectors $b^\nu$ are perpendicular to $d^l$.

Equation (14) has the identical form as the original linear problem (1). If $y$ is a zero vector, we can simply set $h^{\mathcal{V}}$ to be a zero vector too and then a solution $h$ with a single non-zero entry $h_l$ is obtained by (13). On the other hand if the residual $y$ is non-zero, we can obtain the shortest Euclidean-length solution of (14) as the new guidance vector. A new leading index (say $m$) will then be identified, and the corresponding entry $h_m$ is expressed by the remaining $N-2$ entries of $h$ as

$$h_m = \frac{\langle y, b^m \rangle}{\langle b^m, b^m \rangle} - \sum_{\mu \neq l,m} h_\mu \frac{\langle b^\mu, b^m \rangle}{\langle b^m, b^m \rangle} .$$

(17)

This Shortest-Solution guided Decimation (SSD) process will terminate within a number $K$ of steps ($K \leq M$ for sure). We will then achieve a unique solution for the $K$ selected entries $h_1, h_m, \ldots$ by backtracking the $K$ derived equations such as (17) and (13), setting all the other $N-K$ entries of $h$ to be exactly zero. This solution $h$ will have at most $K$ non-zero entries. We provide the pseudo-code of SSD in Algorithm 1 and the corresponding MATLAB code in Appendix A. The longer C++ implementation is provided at power.itp.ac.cn/~zhouhj/codes.html.

An illustration, we show in Fig. 3 the traces of two SSD processes obtained on a $2000 \times 10000$ Gaussian matrix $D$ for two planted solutions $h^0$ with 600 non-zero entries. We see that the leading index $l$ is not always reliable, sometimes the true value of $h_0^0$ is actually zero. But the SSD algorithm is robust to these guessing mistakes. As long as the number of such mistakes is small they will still be corrected by the backtrack process of Algorithm 1 (Fig. 3a and 3b). But if these guessing mistakes are too numerous (Fig. 3c and 3d), the backtrack process is unable to correct all of them and the resulting solution $h$ will be dense. For Algorithm 1 to be successful, the only requirement is that the indexes of all the non-zero en-
FIG. 3: Two traces of the SSD process on a single Gaussian matrix with $M = 2000$ rows and $N = 10000$ columns. The planted solutions $h^0$ for the left [(a), (b)] and right [(c), (d)] columns are different, but they both have the same number $N_{n = 600}$ of non-zero entries (each of them is an i.i.d random value uniformly distributed within $(-1, 1)$). The horizontal axis denotes the step of the decimation. $|x_i|$ is the $l^1$-norm of the residual signal vector $z$. For the left column, the decimation stops after $K = 629$ steps with all the non-zero entries of $h^0$ being identified (the 29 mistakes are then all corrected by the followed backtrack process). For the right column, the decimation stops after $K = 1948$ steps with only 374 non-zero entries of $h^0$ being identified (the followed backtrack process reports a solution that is dramatically different from $h^0$).

entries of $h^0$ are removed from the working index set $\Phi$ in no more than $M$ steps. If this condition is satisfied the backtrack process of the algorithm will then completely recover the planted solution $h^0$.

The SSD reconstruction algorithm has two major differences with the greedy OMP algorithm [10–13]: (i) the matrix $D$ is recursively decimated in SSD but it is fixed OMP; and (ii) SSD computes the guidance vector from the residual signal vector $z$ through (6) or (7) with $D$ being the decimated matrix, but in OMP this is simply $g = D^Tz$ (with $D$ being the original matrix). The second difference makes SSD run slower than OMP. On the other hand, the results in the next section will demonstrate that the guidance vector of SSD is much better than that of OMP, especially on correlated sampling matrices.

V. COMPARATIVE RESULTS

We now test the performance of the SSD algorithm on sparse reconstruction tasks involving both uncorrelated and correlated sampling matrices. As the measure of correlations we consider the condition number $Q \equiv \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}$, which is the ratio between the maximum ($\lambda_{\text{max}}$) and the minimum ($\lambda_{\text{min}}$) singular value of the sampling matrix $D$. An uncorrelated matrix has condition number $Q \approx 1$, while a highly correlated or structured matrix has condition number $Q \gg 1$. Notice that the RIP condition is severely violated for matrices with large $Q$ values. Two types of sampling matrix $D$ are examined in our simulations:

**Gaussian:** Each entry of the matrix is an identically and independently distributed (i.i.d) random real value drawn from the Gaussian distribution with mean zero and unit variance. Every two different columns $d^0$ and $d^0$ of such a random matrix are almost orthogonal to each other; that is, $\frac{1}{M} |\langle d^0, d^0 \rangle| \approx \frac{1}{\sqrt{M}}$. The condition number $Q$ of a Gaussian matrix is close to 1, and the RIP condition is satisfied.

**Correlated:** The matrix $D$ is obtained as the product of an $M \times R$ matrix $D_1$ and a $R \times N$ matrix $D_2$, so $D = D_1D_2$. Both $D_1$ and $D_2$ are Gaussian random matrices as described above. As the rank number $R$ approaches $M$ from above, the entries in the composite matrix $D$ become more and more correlated and the condition number $Q$ increases quickly.

The planted solutions $h^0$ are sparse random vectors with a fraction $\rho$ of non-zero entries. The positions of the $\rho N$ non-zero entries are chosen completely at random from the $N$ possible positions. Two types of planted solutions (Gaussian or uniform) are employed in this paper. The non-zero entries of $h^0$ are i.i.d random real values drawn from the Gaussian distribution with mean zero and unit variance or from the uniform distribution over $(-1, +1)$, for the Gaussian-type and uniform-type $h^0$ respectively. The relative difference $\Delta$ between $h^0$ and the reconstructed solution $h$ of the algorithm is defined as

$$\Delta \equiv \frac{\sqrt{\sum_{i=1}^{N} (h_i - h_i^0)^2}}{\sqrt{\sum_{i=1}^{N} (h_i)^2}}.$$

If this relative distance is considerably small (that is, $\Delta \leq 10^{-5}$), we claim that the planted solution $h^0$ has been correctly recovered by the algorithm.

Many heuristic algorithms have been designed to solve the compressed sensing problem. Here we choose three representative algorithms for the comparative study, namely Minimum $l_1$-norm (ML1), Orthogonal Matching Pursuit (OMP), and Approximate Message Passing (AMP). The algorithm ML1 tries to find a solution of (4) with minimized $l_1$-norm $\sum_{i=1}^{N} |h_i|$. Here we use the MATLAB toolbox L1 MAGIC [25]. The stop criterion for the primal-dual routine of this code is set to be $\text{pdtol} = 10^{-3}$ following the literature. The OMP algorithm is implemented according to [12] [13]. The AMP algorithm is implemented according to [20] [21]. The Gaussian-Bernoulli prior is assumed for the planted solution $h^0$ in the AMP algorithm as in [20] [21], and the sparsity $\rho$ of $h^0$ is revealed to AMP as input.

We fix the compression ratio to $\alpha = 0.2$ in our numerical experiments. Two different values of $N$ are con-
considered, $N = 10^3$ and $N = 10^4$ (and correspondingly, $M = 200$ and $M = 2000$).

### A. Gaussian sampling matrix

The performance of SSD on Gaussian sampling matrices is compared with those of ML1 (Fig. 4), OMP (Fig. 5), and AMP (Fig. 6). Our simulation results demonstrate that, at fixed compression ratio $\alpha$ the transition between successful and unsuccessful reconstruction becomes sharper as $N$ increases. At $N = 10^4$ and $\alpha = 0.2$, when the planted solutions are Gaussian-type, SSD can successfully reconstruct the planted solutions with high probability (e.g., success rate $s > 50\%$) if the signal sparsity is $\rho \leq 0.078$. The corresponding values for ML1, OMP and AMP are, respectively, $\rho \leq 0.04, 0.07$ and $0.093$. When the planted solutions are uniform-type, SSD can successfully reconstruct at sparsity $\rho \leq 0.067$. The corresponding values for ML1, OMP and AMP are, respectively, $\rho \leq 0.045, 0.065$ and $0.079$.

It is encouraging to observe that SSD achieves much better performance than the $\ell_1$-based algorithm ML1. In comparison with ML1, the SSD algorithm is closer to $\ell_0$-norm minimization since it tries to find the non-zero entries of $h^0$ but does not care about the magnitude of these entries.

The SSD algorithm slightly outperforms OMP on the Gaussian matrices. Because the different columns of a Gaussian random matrix $D$ are almost orthogonal to each other, the pseudo-inverse $D^+$ of this matrix does not differ very much from the transpose $D^T$. This could explain why SSD only weakly improves over OMP. Similar small improvements of SSD over OMP are observed on uniform sampling matrices whose entries are i.i.d real values uniformly distributed in $(-1, +1)$. These comparative results suggest that the SSD iteration process makes less guessing mistakes than the OMP iteration does.

The AMP algorithm performs considerably better than SSD on the Gaussian matrices (Fig. 6). This may not be surprising since AMP is an global optimization approach. Compared with SSD, the AMP algorithm also takes more additional information of $h^0$ as input, including its sparsity level $\rho$ and the probability distribution of its non-zero entries, which may not be available in some real-world applications.
FIG. 7: Comparing SSD and OMP on a single correlated measurement matrix $D$ with $N = 1000$ columns and $M = 280$ rows. The protocol of generating this matrix is described in the main text. The planted solution $h^0$ is a uniform-type random sparse vector with $N_{nz} = 50$ non-zero entries. The main figure shows how the $\ell_2$-norm $||z||_2$ of the residual signal vector $z$ changes with the decimation step (plus points, SSD results; cross points, OMP results). The inset compares the entries $h^0_i$ of the planted solution $h^0$ and the corresponding entries $h_i$ of the recovered solution $h$.

B. Correlated sampling matrix

Although SSD, in comparison with AMP, does not give very impressive results on Gaussian random matrices, it works much better than AMP (and also OMP and ML1) on correlated matrices. Let us first consider an ill-conditioned $280 \times 1000$ matrix $D$ obtained by multiplying two random Gaussian matrices of size $280 \times 200$ and $200 \times 1000$, respectively. The condition number of this matrix $D$ is $Q = \infty$. The message-passing iteration of AMP quickly diverges on this instance and the algorithm then completely fails, even for very small sparsity $\rho = 0.005$ (similar divergence problem is experienced for the ML1 algorithm). OMP has no difficulty in constructing a solution $h$, but its solution is rather dense and is much different from the planted solution $h^0$ (Fig. 7, cross points). On the other hand, SSD is not affected by the strong correlations in $D$ and it fully reconstructs $h^0$ (Fig. 7, plus points). From Fig. 7 we see that SSD initially causes a smaller decrease in the Euclidean length ($\ell_2$ norm) of the signal vector $z$ than OMP does, but the slope of $||z||_2$ decreasing keeps roughly the same in later iterations.

Figure 8 shows how the performances of SSD and OMP change with the sparsity $\rho$ of the planted solutions $h^0$ on another larger matrix ($N = 10^4$, $\alpha = 0.2$, condition number $Q = 3.84 \times 10^5$). (The results of AMP are not shown because it always fails to recover $h^0$.) We observe that OMP works only for sparsity $\rho < 0.02$, while SSD is successful up to sparsity $\rho \approx 0.072$ for Gaussian-type $h^0$ and up to $\rho \approx 0.06$ for uniform-type $h^0$. Comparing Fig. 8 with Fig. 7, we see that SSD performs almost equally good on the highly correlated matrix but OMP is very sensitive to these correlations.

To see how the algorithmic performance is affected by the condition number $Q$ of the sampling matrices, we generate a set of 16 correlated $200 \times 1000$ matrices $D$ according to the described protocol using different values of $R = 200, 208, 220, 240, 260, 280, 300, 350, 400, 450, 500, 700, 1000, 2000, 3000, 8000$. These matrices are labeled with index ranging from 1 to 16 in Fig. 9 and their corresponding condition numbers are $Q \approx 318.37, 101.68, 42.95, 24.76, 17.72, 13.74, 10.93, 8.96, 7.27, 6.45, 5.81, 4.47, 4.00, 3.14, 2.97$ and 2.73, respectively. We find that SSD is very robust to correlations (Fig. 9 and 9g) but the other algorithms are not: the performances of ML1, OMP and AMP all severely deteriorate with the increasing of $Q$ (Fig. 9g - 9i).

We have also tested the algorithms on correlated matrices generated by some other more complicated protocols. These additional simulation results (not shown here) further confirm the high degree of insensitivity of SSD. This rather peculiar property should be very desirable in practical applications, because it greatly relaxes the requirements on the sampling matrix.

VI. CONCLUSION AND DISCUSSIONS

In this paper we introduced the Shortest-Solution Decimation (SSD) algorithm for the compressed sensing problem and tested it on uncorrelated and correlated sampling matrices. The SSD algorithm is a geometry-inspired deterministic algorithm that does not explicitly try to minimize a cost function. SSD outperforms OMP in recovering sparse planted solutions; and it is especially competitive in treating highly correlated or structured matrices, on which the tested other repre-
FIG. 9: Comparing the algorithmic performances on 16 correlated 200 × 1000 sampling matrices \( D \) indexed from 1 (pluses, most highly correlated) to 16 (decorated circles, least correlated). Left column [(a), (c), (e), (g)] corresponds to Gaussian-type planted solutions \( h^0 \), right column [(b), (d), (f), (h)] corresponds to uniform-type \( h^0 \). The simulation results are obtained by SSD [(a), (b)], ML1 [(c), (d)], OMP [(e), (f)] and AMP [(g), (h)], respectively. \( \rho \) is the sparsity of \( h^0 \); \( s \) is the fraction of successfully reconstructed solutions among 100 random input \( h^0 \) samples.

Rigorous theoretical understanding is largely absent on why the SSD algorithm is highly tolerant to structural correlations in the sampling matrix. We feel that (i) viewing the sparse recovery problem from the angle of eigen-subspace projection \( (\ref{9}) \) and (ii) recursively adjusting this eigen-projection by modifying the matrix \( D \) and the signal vector \( z \) [Algorithm \ref{alg1}] are crucial to make SSD insensitive to correlations. On the other hand, recursively changing the sampling matrix \( D \) during the reconstruction process seems to pose a big challenge for mathematical analysis. We hope the empirical results of this paper will stimulate more theoretical efforts on the SSD dynamics.

In this proof-of-concept paper we only considered the ideal noise-free situation; but for practical applications it will be necessary to take into account possible uncertainty in the sampling matrix \( D \) and the unavoidable measurement noise in the signal vector \( z \). Time complexity of the SSD algorithm is another important issue to be addressed in future studies. To accelerate the SSD process an easy adaptation is to fix a small fraction of the active indexes (instead of just one of them) in each decimation step. We also notice that the repeated operation of computing the guidance vector \( g \) may be accomplished in an approximate perturbative way. These and other possible ways of optimizing SSD will be explored in the near future. Extensions of the algorithmic ideas of this work to complex-valued compressed sensing problems also need to be carried out.

Appendix A: A MATLAB implementation of SSD

```matlab
function [ h ] = ssd( D, z , epsilon )
[m, n]= size( D ) ;
D0=D ; z0=z ;
g=pinv( D )*z ;
support =[] ;
remain =1:n ;
while norm(z)>epsilon
[-1]=max( abs( g ) ) ;
support=[support,remain(1)] ;
a=D(:,1) ;
a_unit=a/(a'*a) ;
z=z-(z'*a)*a_unit ;
D=D-repmat(a'*D,m,1).*repmat(a_unit,1,size(D,2)) ;
D(:,1)=[] ;
remain(1)=[] ;
g=pinv(D)*z ;
end
h=zeros(n,1) ;
h(support)=pinv(D0(:,support))*z0 ;
end
```

In this MATLAB code the guidance vector \( g \) is obtained through \( (\ref{8}) \) which involves computing the pseudoinverse of the matrix \( D \) by SVD. An alternative way of implementing the SSD algorithm is to use \( (\ref{7}) \) and \( (\ref{9}) \) which are based on LQ decomposition approach. The corresponding MATLAB and C++ codes can be found at power.itp.ac.cn/~zhouhj/codes.html.
Acknowledgement

H.J.Z. thanks Jing He, Fengyao Hou and Hongbo Jin for help in computer simulations; P.Z. acknowledges helpful discussions with Dong Liu, Xiangming Meng and Chuang Wang; M.S. acknowledges the hospitality of ITP-CAS during his stay as an intern student. H.J.Z. and P.Z. acknowledge the hospitality of Kavli Institute for Theoretical Sciences (KITs-UCAS) during the workshop “Machine Learning and Many-Body Physics” (June 28–July 7, 2017). This research was supported by the National Natural Science Foundation of China (grant numbers 11421063 and 11647601) and the Chinese Academy of Sciences (grant number QYZDJ-SSW-SYS018). The numerical computations were carried out at the HPC Cluster of ITP-CAS and the Tianhe-2 platform of the National Supercomputer Center in Guangzhou.

[1] G.-M. Shi, D.-H. Liu, D.-H. Gao, Z. Liu, J. Lin, and L.-J. Wang, “Advances in theory and application of compressed sensing,” Acta Electronica Sinica, vol. 37, pp. 1070–1081, 2009.
[2] S. Foucart and H. Rauhut, A Mathematical Introduction to Compressive Sensing. New York: Springer, 2013.
[3] E. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” IEEE Trans. Inf. Theory, vol. 52, pp. 489–509, 2006.
[4] D. L. Donoho, “Compressed sensing,” IEEE Trans. Inf. Theory, vol. 52, pp. 1289–1306, 2006.
[5] A. C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss, “Near-optimal sparse fourier representations via sampling,” in Proc. 34th Annual ACM Symposium on Theory of Computing (STOC ’02). Montreal, Quebec, Canada: ACM, New York, 2002, pp. 152–161.
[6] D. L. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell_1$ minimization,” Proc. Natl. Acad. Sci. USA, vol. 100, pp. 2197–2202, 2003.
[7] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, “A survey of sparse representation: Algorithms and applications,” IEEE Access, vol. 3, pp. 490–530, 2015.
[8] S. Chen, S. A. Billings, and W. Luo, “Orthogonal least squares methods and their application to non-linear system identification,” Int. J. Control, vol. 50, pp. 1873–1896, 1989.
[9] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” IEEE Trans. Signal Processing, vol. 41, pp. 3397–3415, 1993.
[10] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in Proc. 27th Asilomar Conference on Signals, Systems and Computers. IEEE, 1993, pp. 40–44.
[11] G. M. Davis, S. G. Mallat, and Z. Zhang, “Adaptive time-frequency decompositions,” Optical Engineering, vol. 33, pp. 2183–2191, 1994.
[12] J. A. Tropp, “Greed is good: Algorithmic results for sparse approximation,” IEEE Trans. Inf. Theory, vol. 50, pp. 2231–2242, 2004.
[13] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” IEEE Trans. Inf. Theory, vol. 53, pp. 4655–4666, 2007.
[14] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” IEEE Trans. Inf. Theory, vol. 55, pp. 2230–2249, 2009.
[15] S. Chatterjee, K. V. S. Hari, P. Händel, and M. Skoglund, “Projection-based atom selection in orthogonal matching pursuit for compressive sensing,” in Proceedings of the 2012 National Conference on Communications. Kharagpur, India (3–5 Feb., 2012): IEEE, 2012.
[16] R. Gribonval and M. Nielsen, “Sparse representations in unions of bases,” IEEE Trans. Inf. Theory, vol. 49, pp. 3320–3325, 2003.
[17] D. L. Donoho, “For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution,” Commun. Pure and Applied Math., vol. 59, pp. 797–829, 2006.
[18] R. Tibshirani, “Regression shrinkage and selection via the lasso,” J. R. Statist. Soc. B, vol. 58, pp. 267–288, 1996.
[19] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” SIAM review, vol. 43, pp. 120–159, 2001.
[20] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” Proc. Natl. Acad. Sci. USA, vol. 106, pp. 18914–18919, 2009.
[21] F. Krzakala, M. Mézard, F. Sausset, Y. F. Sun, and L. Zdeborová, “Statistical-physics-based reconstruction in compressed sensing,” Phys. Rev. X, vol. 2, p. 021005, 2012.
[22] D. J. Thouless, P. W. Anderson, and R. G. Palmer, “Solution of ‘solvable model of a spin glass’,” Phil. Mag., vol. 35, pp. 593–601, 1977.
[23] E. J. Candès and T. Tao, “Near optimal signal recovery from random projections: universal encoding strategies?” IEEE Trans. Inf. Theory, vol. 52, pp. 5406–5425, 2006.
[24] M. Bayati and A. Montanari, “The dynamics of message passing on dense graphs, with applications to compressed sensing,” IEEE Trans. Inf. Theory, vol. 57, pp. 764–785, 2011.
[25] E. Candès and J. Romberg, “L1-magic: Recovery of sparse signals via convex programming,” URL: www.acm.caltech.edu/l1magic/downloads/l1magic.pdf, Tech. Rep., 2005.
[26] J. Otsuki, M. Ohzeki, H. Shinaoka, and K. Yoshimi, “Sparse modeling approach to analytical continuation of imaginary-time quantum monte carlo data,” Phys. Rev. E, vol. 95, p. 061302(R), 2017.
[27] J. Ma and L. Ping, “Orthogonal amp,” IEEE Access, vol. 5, pp. 2020–2033, 2017.
[28] S. Rangan, P. Schniter, and A. Fletcher, “Vector approximate message passing,” arXiv:1610.03082, 2016.