BEYOND MOORE-PENROSE: SPARSE PSEUDOINVERSE

Ivan Dokmanić, Mihailo Kolundžija and Martin Vetterli

School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland
{ivan.dokmanic,mihailo.kolundzija,martin.vetterli}@epfl.ch

ABSTRACT

Frequently, we use the Moore-Penrose pseudoinverse (MPP) even in cases when we do not require all of its defining properties. But if the running time and the storage size are critical, we can do better. By discarding some constraints needed for the MPP, we gain freedom to optimize other aspects of the new pseudoinverse. A sparser pseudoinverse reduces the amount of computation and storage. We propose a method to compute a sparse pseudoinverse and show that it offers sizable improvements in speed and storage, with a small loss in the least-squares performance. Differently from previous approaches, we do not attempt to approximate the MPP, but rather to produce an exact but sparse pseudoinverse. In the underdetermined (compressed sensing) scenario we prove that the rescaled sparse pseudoinverse yields an unbiased estimate of the unknown vector, and we demonstrate its potential in iterative sparse recovery algorithms, pointing out directions for future research.

Index Terms— Efficient computation, Moore-Penrose pseudoinverse, sparse pseudoinverse

1. INTRODUCTION

In discrete linear inverse problems, we seek to find $\mathbf{x}$ from measurements $\mathbf{y}$, when they are related by a linear system, $\mathbf{y} = \mathbf{A}\mathbf{x}$, $\mathbf{A} \in \mathbb{R}^{m\times n}$. Such problems come in two very different flavors: overdetermined ($m > n$) and underdetermined ($m < n$). An example is computed tomography (CT). If we target a coarse reconstruction resolution (less pixels than rays), the system matrix is tall and we deal with an overdetermined system. In the opposite case the system is underdetermined. Entries of the model matrix quantify how much the ith ray affects the jth pixel.

Consider an overdetermined problem

$$\mathbf{y} = \mathbf{A}_t\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{A}_t \in \mathbb{R}^{m\times n},\ m > n$, $\mathbf{n}$ is noise, and t stands for "tall". In general, when $\mathbf{y}$ is perturbed by noise, it is not anymore in $\mathcal{R}(\mathbf{A}_t)$ and no solution $\widehat{\mathbf{x}}$ such that $\mathbf{y} = \mathbf{A}_t\widehat{\mathbf{x}}$ exists. If $\mathbf{n} = 0$, we can compute $\mathbf{x}$ by inverting any full rank $n \times n$ minor of $\mathbf{A}_t$, assuming that $\mathbf{A}_t$ has full column rank. With $\mathbf{n} \neq 0$ we get the solution corresponding to the orthogonal projection of $\mathbf{y}$ on $\mathcal{R}(\mathbf{A}_t)$ by applying the Moore-Penrose pseudoinverse (MPP). Explicit form of the MPP solution is $\widehat{\mathbf{x}} = \mathbf{A}_t^\dagger\mathbf{y} = (\mathbf{A}_t^\top\mathbf{A}_t)^{-1}\mathbf{A}_t^\top\mathbf{y}$.

The key problem is that the complexity of applying and storing the MPP is $\Theta(mn)$, which is in some cases too expensive. Focus of this paper is on computing a different (exact) pseudoinverse that requires considerably less operations and storage, but does not incur a large hit in output SNR.

MPP of the matrix $\mathbf{A}$ is the unique matrix $\mathbf{A}^\dagger$ satisfying

$$\begin{aligned} \mathbf{A}\mathbf{A}^\dagger\mathbf{A} &= \mathbf{A}, & (\mathbf{A}\mathbf{A}^\dagger)^\top &= \mathbf{A}\mathbf{A}^\dagger, \\ \mathbf{A}^\dagger\mathbf{A}\mathbf{A}^\dagger &= \mathbf{A}^\dagger, & (\mathbf{A}^\dagger\mathbf{A})^\top &= \mathbf{A}^\dagger\mathbf{A}. \end{aligned} \tag{2}$$

Ordinarily we do not reconsider the choice of the pseudoinverse when the least-squares properties are not critical. But if the computation time and the amount of storage are limited, it is worthwhile to replace some of the requirements for MPP with different ones reflecting these constraints. In applications of our interest, the low-power embedded processor used for number crunching can execute only a fraction of operations required to apply the MPP at the target frame rate. The same goes for the required storage (but less dramatically).

Sparse pseudoinverse is useful in the underdetermined case as well. Namely, we show that it can be used as a building block of algorithms for sparse vector recovery (compressed sensing). We focus on iterative algorithms, following an example presented by Montanari at ITW 2012 [1]: Given a budget of k matrix-vector multiplications, what can you say about the solution of an underdetermined linear inverse problem $\mathbf{y} = \mathbf{A}_f\mathbf{x}$, under certain priors on $\mathbf{x}$ (f stands for "fat"). Typically $\mathbf{x}$ is in some sense low-dimensional, for example sparse. These iterative algorithms have two blocks: 1) a matched-filtering-like linear block, followed by 2) a non-linearity. Depending on the application, the non-linearity is either thresholding (iterative hard thresholding (IHT) [2, 3, 4], iterative soft thresholding (IST) [5]), or a more general denoising [1, 6]. Clearly it is desirable to reduce the complexity of each building block, while maintaining the performance. We demonstrate encouraging initial results by using the sparse pseudoinverse in the linear stage and point out directions for future work.

1.1. Prior Art

Replacing full by sparse matrices to reduce the computation cost is not at all a new idea. When solving large systems of linear equations, often coming from partial differential equations, it is favorable to have sparse preconditioning matrices. This includes preconditioners for multipole methods [7] or for the Helmholtz equation [8]. In [9] the author observes that the inverse finite element method (FEM) matrices have many small elements and then simply thresholds them. Approximating the MPP by a sparse matrix is proposed in [10] in the context of MRI reconstruction. The authors use a greedy algorithm adding elements one by one. In [11, 12, 13] the authors sparsify inverse covariance matrices in the context of speech recognition. When the forward matrix is also sparse it is important to

This work was supported by an ERC Advanced Grant – Support for Frontier Research – SPARSAM Nr: 247006, and by the CTI grant Nr: 14371.2 PFNM-NM.

compare our method with known iterative algorithms such as Kaczmarz [14] or randomized Kaczmarz [15]. To take advantage of sparsity in implementations, we can either hardcode the multiplication, or use fast sparse-matrix-vector multiplication algorithms [16, 17] if universality is desired.

1.2. Main Contributions

First, we propose a new method for solving overdetermined linear inverse problems by the sparse pseudoinverse. It outperforms the approaches based on sparse approximations of MPP and the iterative algorithms for an equivalent number of elementary operations. It is best suited for moderate-size systems as computing a pseudoinverse for very large matrices is expensive. A good example and the inspiration for this paper are embedded applications. Our method offers sizable savings in execution time and storage. The savings increase if we can trade off the inverting property for sparsity. We stress that we do not aim at *approximating the MPP* by a sparse matrix. Rather, we are computing an *exact* pseudoinverse with many zeros. We demonstrate that some obvious ideas to reduce the computation, such as inverting any full rank minor, perform much worse.

Second, we propose to use the sparse pseudoinverse in the underdetermined case for sparse recovery or compressed sensing [18]. We show that the sparse pseudoinverse can be a part of iterative algorithms such as [6]. We prove that the expected value of estimating $\mathbf{x}$ by the sparse pseudoinverse is indeed $\mathbf{x}$ for different random ensembles of forward matrices. This property is important for assessing the performance of the sparse pseudoinverse in iterative algorithms. Finding the optimal parameters for these algorithms as was done in [19] is beyond the scope of this paper. Even with an ad hoc choice of parameters, we get results comparable to state-of-the-art, with less computation. The results confirm that alternative pseudoinverses have practical merit in both overdetermined and underdetermined scenarios.

2. OVERDETERMINED CASE

Imagine performing a low-resolution tomographic reconstruction of an object on low-power embedded hardware. The relationship between image elements $\mathbf{x}$ and measurements $\mathbf{y}$ is given as

$$\mathbf{y} = \mathbf{A}_t\mathbf{x} + \mathbf{n}, \tag{3}$$

where the model matrix $\mathbf{A}_t \in \mathbb{R}^{m\times n}$, $m > n$, is determined by the hardware and $\mathbf{n} \in \mathbb{R}^m$ represents the measurement noise. We seek the solution as

$$\widehat{\mathbf{x}} = \mathbf{B}\mathbf{y}\,, \tag{4}$$

for some reconstruction matrix $\mathbf{B} \in \mathbb{R}^{n\times m}$ that is computed offline and stored on the device. Applying a dense $\mathbf{B}$ in (4) is too expensive both in terms of the processing time and the consumed memory. Consequently, we cannot apply the MPP. Note that the optimal solution to (4) in the MSE sense is not given by the MPP, but as [20]

$$\mathbf{B}_{\mathsf{MSE}} = \mathbf{C}_x\mathbf{A}^\top(\mathbf{A}\mathbf{C}_x\mathbf{A}^\top + \mathbf{C}_n)^{-1}\,, \tag{5}$$

where $\mathbf{C}_x$ and $\mathbf{C}_n$ are signal and noise covariance matrices (MPP is a special case of this formula). $\mathbf{B}_{\mathsf{MSE}}$ is also a dense matrix.

In order to save on computation and storage, we seek a sparse $\mathbf{B}$. Sufficient sparsity would enable (4) to be computed rapidly. To this end, we propose to use the *sparse pseudoinverse* of $\mathbf{A}$,

$$\mathsf{spinv}(\mathbf{A}) = \arg\min \|\mathbf{B}\|_1 \;\text{ subject to }\; \mathbf{B}\mathbf{A} = \mathbf{I}_n, \tag{6}$$

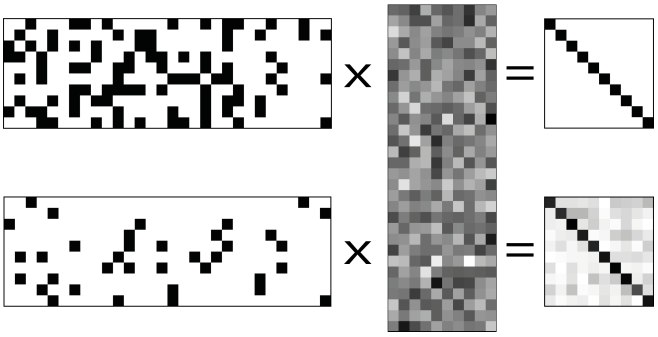

Fig. 1: Illustration of sparse pseudoinverses (exact and approximate). The sparsity of the exact sparse pseudoinverse is 66 %, and the sparsity of the approximate sparse pseudoinverse is 90%.

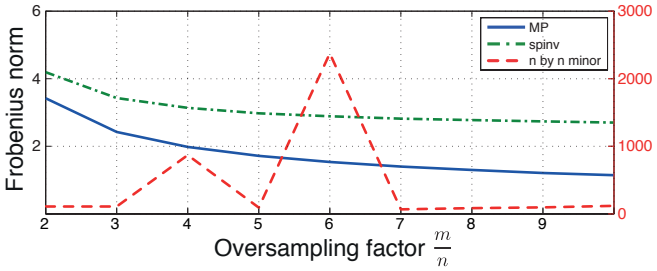

Fig. 2: Frobenius norms for the MPP, the sparse pseudoinverse, and the inverse of an $n \times n$ minor, for $n = 50$, and entries $\sim \mathcal{U}([0, 1])$. Note that the Frobenius norm of the inverse minor has a different scale, shown on the right. Besides, it varies substantially so that 20 realizations are not enough to indicate the true mean.

where $\|\mathbf{B}\|_1$ denotes the entrywise ℓ_1 norm of $\mathbf{B}$. The linear program (6) is separable and can be solved by computing one row of $\mathbf{B}$ at a time. We aim at minimizing the number of non-zeros in $\mathsf{spinv}(\mathbf{A})$. This is in general a non-tractable problem, so we use the standard linear relaxation with the ℓ_1 norm. The logic behind computing $\mathsf{spinv}(\mathbf{A})$ is that the pseudoinverse has nm elements, but we have only n^2 constraints, which leaves us with $nm - n^2$ degrees of freedom. Intuitively, we expect to get a fraction $\frac{nm-n^2}{nm} = 1 - \frac{n}{m}$ of zeros. For very tall matrices the savings can be considerable.

A legitimate question is whether (6) really produces a sparse matrix? The answer is positive: In all the test cases, for both full and sparse forward matrices, the number of zeros in $\mathsf{spinv}(\mathbf{A})$ was at least $nm - n^2$. To further sparsify the matrix, we can apply entrywise hard thresholding, thus sacrificing the inverting property and computing an approximate sparse pseudoinverse,

$$[\mathsf{sspinv}_\tau(\mathbf{A})]_{ij} = \eta_\tau\{[\mathsf{spinv}(\mathbf{A})]_{ij}\}, \tag{7}$$

with $\eta_\tau(u) = u\mathbf{1}_{|u|\geq\tau}$. Both pseudoinverses are illustrated in Fig. 1. To compare the MSE performance with the MPP, we need to compute the Frobenius norm of $\mathsf{spinv}(\mathbf{A})$. It can be shown that the output SNR of any such estimator depends on its Frobenius norm, suggesting that the sparse pseudoinverse will do worse than MPP in terms of MSE. This is summarized in the following simple proposition,

Proposition 1. *Let* $\mathbf{A}_t \in \mathbb{R}^{m\times n}$, $\mathbf{y} = \mathbf{A}_t\mathbf{x} + \mathbf{n}$ *and* $\mathbf{n}$ *be random zero-mean noise such that* $\mathbb{E}[\mathbf{n}\mathbf{n}^\top] = \sigma^2\mathbf{I}_m$. *Then we have*

$$\mathbb{E}\left[\|\mathbf{x} - \mathbf{B}\mathbf{y}\|^2\right] = \sigma^2\|\mathbf{B}\|_F^2 \tag{8}$$

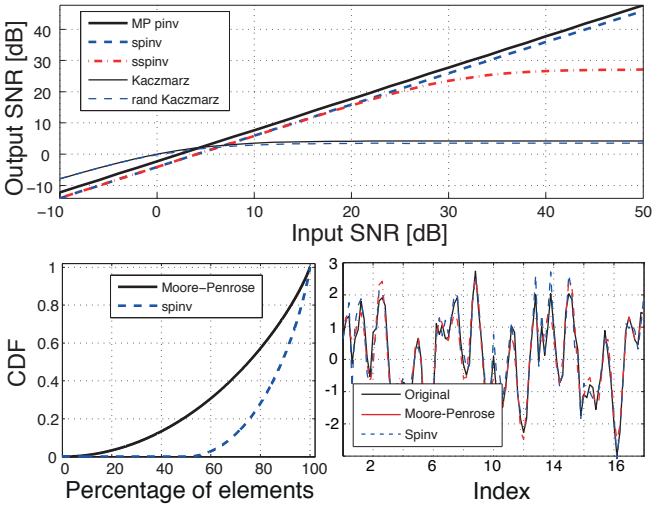

Fig. 3: Output SNR and cumulative distribution functions (CDFs) of matrix element magnitudes for Moore-Penrose pseudoinverse, $\mathsf{spinv}(\mathbf{A})$, $\mathsf{sspinv}(\mathbf{A})$ and iterative methods when the forward matrix is full. Lower-left figure shows a reconstruction example at $\mathsf{SNR} = 10$ dB. CDF threshold for sspinv was set to 0.05.

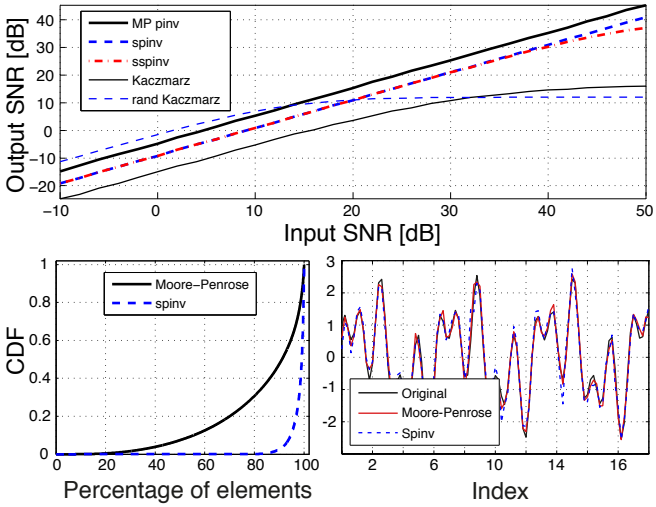

Fig. 4: Output SNR and cumulative distribution functions (CDFs) of matrix element magnitudes for Moore-Penrose pseudoinverse, $\mathsf{spinv}(\mathbf{A})$, $\mathsf{sspinv}(\mathbf{A})$ and iterative methods when the forward matrix is sparse. Lower-left figure shows a reconstruction example at $\mathsf{SNR} = 10$ dB. CDF threshold for sspinv was set to 0.01.

for any $\mathbf{B} \in \mathbb{R}^{n\times m}$ *such that* $\mathbf{B}\mathbf{A}_t = \mathbf{I}_n$.

Thus we cannot do better than the MPP since it minimizes the Frobenius norm under the constraint $\mathbf{B}\mathbf{A}_t = \mathbf{I}_n$. How much worse do we do? We have the following upper bound on the Frobenius norm of the sparse pseudoinverse.

Proposition 2. *Let* $\mathbf{A}_t \in \mathbb{R}^{m\times n}$, $m > n$. *Then* $\|\mathsf{spinv}(\mathbf{A}_t)\|_F \leq \sqrt{nm}\|\mathbf{A}_t^\dagger\|_F$.

In practice we observe far better results, as demonstrated in Fig. 2. For $m = 200$, $n = 50$, we have that $\|\mathsf{spinv}(\mathbf{A}_t)\|_F \approx 1.4\|\mathbf{A}_t^\dagger\|_F$. We see that computing the solution by just inverting a full rank $n \times n$ minor of $\mathbf{A}_t$ does substantially worse. In fact, it is worse than the prediction of Proposition 2 for the spinv.

2.1. Example 1: Full Forward Matrix

Fig. 3 shows the performance of various reconstruction methods with a dense forward matrix. We show results for $\mathsf{spinv}(\mathbf{A})$ and $\mathsf{sspinv}_\tau(\mathbf{A})$. The threshold τ is computed from the empirical CDF of absolute values of matrix entries shown in Fig. 3. The number of non-zeros in sspinv is about two times lower than for $\mathsf{spinv}(\mathbf{A})$, and four times lower than for MPP. As we will see in the next example, spinv and sspinv are particularly interesting in the sparse matrix case. We also include the well-known non-linear iterative methods—Kaczmarz [14] and randomized Kaczmarz algorithm [15]. The number of iterations is selected to have the same operation count as for the the application of $\mathsf{sspinv}(\mathbf{A})$. We sampled 50 realizations of $\mathbf{A}$, with entries $a_{ij} \sim \mathcal{U}([0,1])$. For each matrix, a different random input vector was generated, and 50 different iid Gaussian noise vectors $\mathbf{n}$ were added to simulated measurements $\mathbf{y} = \mathbf{A}\mathbf{x}$. We can see from Fig. 3 that at all but the lowest SNRs, the MPP performs the best. But the cost of applying and storing spinv is 2 times lower, and the cost of applying the sspinv 4 times lower. The spinv and sspinv perform similarly up to 30 dB, and both variants of Kaczmarz algorithm perform worse at all but the lowest tested SNRs. This example shows clearly that the spinv and its sparser variant sspinv are useful for overdetermined problems.

2.2. Example 2: Sparse Forward Matrix

In applications such as tomography we often encounter sparse forward matrices. This makes the Kaczmarz algorithm attractive, especially for large systems. Surprisingly, we demonstrate that the proposed scheme may outperform Kaczmarz even for sparse matrices. We run the simulation like in Example 1, but with the fraction of non-zeros in the forward matrix set to 0.03. This is the sparsity of the model matrix $\mathbf{A}_t$ in our practical setup (but with smaller matrix dimensions). The size of $\mathbf{A}_t$ was 202×101. From Fig. 4, we see that the MPP performs consistently better than spinv at all input SNRs by about 6 dB. Again, it requires at least twice the number of operations. The sparser $\mathsf{sspinv}(\mathbf{A}_t)$ performs the same as $\mathsf{spinv}(\mathbf{A}_t)$ up to higher SNR values (around 40 dB). Note that the sspinv performs better than in the dense case, which is suggested by the CDF. But, sspinv requires **38** times less computation than the MPP. Randomized Kaczmarz algorithm performs the best on the average at low SNRs (below 15 dB), but its performance at higher SNRs is notably inferior to all three linear reconstruction methods. Conventional Kaczmarz has the worst average performance at all tested SNRs.

3. UNDERDETERMINED CASE

The usefulness of spinv extends to the underdetermined case. We first prove an important property of the sparse pseudoinverse which justifies its application in iterative sparse recovery algorithms. This property is satisfied by the MPP as demonstrated in [1]. Some obvious "sparse" formulations, such as inverting a minor, do not satisfy it. We define $\mathsf{spinv}(\mathbf{A})$ for a "fat" $\mathbf{A}$ similarly to the tall case,

$$\mathsf{spinv}(\mathbf{A}) = \arg\min \|\mathbf{B}\|_1 \;\text{ subject to }\; \mathbf{A}\mathbf{B} = \mathbf{I}_m. \qquad (9)$$

With this definition, we can state the following unbiasedness result.

Theorem 1. *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$, $m < n$ be a random matrix with iid entries such that $a_{ij} \sim (-a_{ij})$. Then $\mathbb{E}[\mathsf{spinv}(\mathbf{A})\mathbf{A}] = \frac{m}{n}\mathbf{I}_n$.*

To prove the theorem we need the following lemma,

Lemma 1. *Let $\mathbf{\Pi} \in \mathbb{R}^{n\times n}$ be a permutation matrix, and $\mathbf{\Sigma}$ a modulation matrix, $\mathbf{\Sigma} = \mathbf{diag}(\pm 1, \pm 1, \ldots, \pm 1)$. Then the following two claims hold,*

$$\mathsf{spinv}(\mathbf{A\Pi}) = \mathbf{\Pi}^\top \mathsf{spinv}(\mathbf{A}) \tag{10}$$

$$\mathsf{spinv}(\mathbf{A\Sigma}) = \mathbf{\Sigma}\, \mathsf{spinv}(\mathbf{A}). \tag{11}$$

Proof of the lemma. We only prove the first claim. The second part follows analogously.
Feasibility: $(\mathbf{A\Pi}) \cdot (\mathbf{\Pi}^\top \mathsf{spinv}(\mathbf{A})) = \mathbf{A}\mathsf{spinv}(\mathbf{A}) = \mathbf{I}_m$.
Optimality: Suppose that there exists $\mathbf{B}$ such that $(\mathbf{A\Pi})\mathbf{B} = \mathbf{I}_m$ and $\|\mathbf{B}\|_1 < \|\mathbf{\Pi}^\top \mathsf{spinv}(\mathbf{A})\|_1$. Since

$$(\mathbf{A\Pi})\mathbf{B} = \mathbf{I}_m = \mathbf{A}(\mathbf{\Pi B}),$$

$(\mathbf{\Pi B})$ is a feasible point in minimization for $\mathsf{spinv}(\mathbf{A})$. But

$$\|\mathbf{\Pi B}\|_1 = \|\mathbf{B}\|_1 < \|\mathbf{\Pi}^\top \mathsf{spinv}(\mathbf{A})\|_1 = \|\mathsf{spinv}(\mathbf{A})\|_1, \tag{12}$$

which contradicts the ℓ_1-optimality of $\mathsf{spinv}(\mathbf{A})$. Therefore it must be that $\mathsf{spinv}(\mathbf{A\Pi}) = \mathbf{\Pi}^\top \mathsf{spinv}(\mathbf{A})$. □

Proof of the theorem. Since the matrix elements are iid, for any permutation $\mathbf{\Pi}$, $\mathbf{A}$ is distributed identically to $\mathbf{A\Pi}$. This implies that functions of $\mathbf{A}$ and $\mathbf{A\Pi}$ have the same distributions. Thus $\mathsf{spinv}(\mathbf{A})\mathbf{A} \sim \mathsf{spinv}(\mathbf{A\Pi})\mathbf{A\Pi}$, and we have that

$$\mathbb{E}[\mathsf{spinv}(\mathbf{A})\mathbf{A}] = \mathbb{E}[\mathsf{spinv}(\mathbf{A\Pi})\mathbf{A\Pi}] \tag{13}$$

$$= \mathbb{E}[\mathbf{\Pi}^\top \mathsf{spinv}(\mathbf{A})\mathbf{A\Pi}]. \tag{14}$$

But note that this holds for any permutation, so by linearity of expectation we can write

$$\mathbb{E}[\mathsf{spinv}(\mathbf{A})\mathbf{A}] = \mathbb{E}\left[\frac{1}{n!}\sum_{\mathbf{\Pi}\in\mathcal{P}} \mathbf{\Pi}^\top \mathsf{spinv}(\mathbf{A})\mathbf{A\Pi}\right]$$

$$= \mathbb{E}\left\{\begin{bmatrix} C & B & \cdots & B \\ B & C & \cdots & B \\ \vdots & & \ddots & \vdots \\ B & B & \cdots & C \end{bmatrix}\right\} = \begin{bmatrix} c & b & \cdots & b \\ b & c & \cdots & b \\ \vdots & & \ddots & \vdots \\ b & b & \cdots & c \end{bmatrix}, \tag{15}$$

where $\mathcal{P}$ denotes the set of all $n \times n$ permutation matrices. We can compute the value of c as follows:

$$\mathrm{Tr}\,\mathbb{E}[\mathsf{spinv}(\mathbf{A})\mathbf{A}] = \mathbb{E}[\mathrm{Tr}\,\mathsf{spinv}(\mathbf{A})\mathbf{A}] \tag{16}$$

$$= \mathrm{Tr}\,\mathbf{A}\,\mathsf{spinv}(\mathbf{A}) = \mathrm{Tr}\,\mathbf{I}_m = m. \tag{17}$$

Since $\mathrm{Tr}\,\mathbb{E}[\mathsf{spinv}(\mathbf{A})\mathbf{A}] = nc$, we get $c = \frac{m}{n}$.

To show that $b = 0$, we observe that $\mathbf{A}$ and $\mathbf{A\Sigma}$ have the same distribution. Similarly to the previous part, we conclude that

$$\mathbb{E}[\mathsf{spinv}(\mathbf{A})\mathbf{A}] = \mathbb{E}\left[\frac{1}{2^n}\sum_{\mathbf{\Sigma}\in\mathcal{M}} \mathbf{\Sigma}\mathsf{spinv}(\mathbf{A})\mathbf{A\Sigma}\right] \tag{18}$$

$$= \mathbf{diag}(c_1, c_2, \ldots, c_n). \tag{19}$$

But we already established that $c_i \equiv \frac{m}{n}$ so indeed $\mathbb{E}[\mathsf{spinv}(\mathbf{A})\mathbf{A}] = \frac{m}{n}\mathbf{I}_n$. □

Many input distributions satisfy the assumptions of the theorem, and in particular iid Gaussian. Moreover, the reasoning can be repeated for a wider class of pseudoinverses. For example, nothing changes in the proof if we replace $\|(\cdot)\|_1$ by $\|(\cdot)\|_p$.

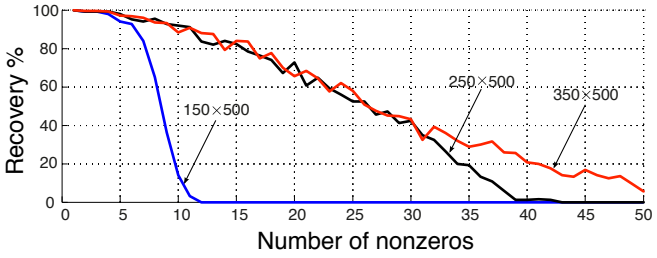

Fig. 5: Percentage of correct recovery with spinv IHT for different matrix sizes, and different sparsities of the unknown vector.

3.1. Example 3: Iterative Hard Thresholding

As a proof-of-concept, we demonstrate initial results on sparse vector recovery for an IHT algorithm with the spinv. We use an ad hoc choice of parameters without attempting to optimize them. A fair comparison would involve searching for an optimal set of parameters. See [19] for an extensive numerical effort to optimally tune a number of popular iterative algorithms. Such large-scale tuning is beyond the scope of the present contribution.

We used the following iterative algorithm for sparse recovery,

$$\widehat{\mathbf{x}}^k = \eta_\tau(\widehat{\mathbf{x}}^{k-1} + \tfrac{n}{m}\mathsf{spinv}(\mathbf{A})(\mathbf{y} - \mathbf{A}\widehat{\mathbf{x}}^{k-1})), \tag{20}$$

where $\eta_\tau(\cdot)$ is the hard thresholding operator defined as $\eta_\tau(u) = u\mathbf{1}_{|u|\geq\tau}$. The threshold was set to retain the s largest magnitude elements, where s is the number of nonzeros in $\mathbf{x}$, assumed known.

Fig. 5 shows the percentage of correct recovery as a function of the number of non-zeros for three different matrix sizes. For each matrix size we sampled five forward matrices and for each matrix 50 random sparse input vectors. The results show that the phase transition occurs at sparsities two or three times lower than for optimally tuned algorithms. This will only improve by finding the optimal parameter set for our algorithms. But the important point is that in the feasible region, where this variant of IHT works, it requires less computation.

4. CONCLUSION AND ONGOING WORK

We have introduced a novel way to compute a matrix pseudoinverse. The proposed pseudoinverse is sparse, thus saving on computation and storage. Unlike earlier approaches, we do not try to approximate the Moore-Penrose pseudoinverse, but we compute an exact pseudoinverse with as few non-zero entries as possible. We show that for overdetermined problems it performs close to the Moore-Penrose pseudoinverse in terms of MSE, while offering considerable savings in terms of computation and storage. By further thresholding the entries, we increase the computational savings, without sacrificing the MSE performance at moderate SNRs. Perhaps surprisingly, we show that in this scenario it outperforms Kaczmarz and randomized Kaczmarz algorithms, for both dense and sparse forward matrices. For the underdetermined (compressed sensing) case, we have proved that the sparse pseudoinverse yields an unbiased estimate of the unknown vector. We demonstrate its applicability in iterative algorithms, with less computation than in the conventional approaches. Further work involves obtaining tight theoretical bounds on the reconstruction performance, and a detailed assessment of the potential of the method in the underdetermined case.

5. REFERENCES

[1] A. Montanari, "Iterative Methods in Statistical Estimation," *Information Theory Workshop*, `http://www.stanford.edu/~montanar/`, Sept. 2012.

[2] T. Blumensath, M. Yaghoobi, and M. E. Davies, "Iterative Hard Thresholding and L0 Regularisation," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, Honolulu, HI, 2007.

[3] T. Blumensath and M. E. Davies, "Iterative Hard Thresholding for Compressed Sensing," *Appl. Comput. Harmon. Anal.*, vol. 27, no. 3, pp. 265–274, Nov. 2009.

[4] K. Qiu and A. Dogandzic, "Sparse Signal Reconstruction via ECME Hard Thresholding," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4551–4569, Sept. 2012.

[5] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[6] M. Bayati, M. Lelarge, and A. Montanari, "Universality in Polytope Phase Transitions and Message Passing Algorithms," *ArXiv e-prints*, July 2012.

[7] J. Lee, J. Zhang, and C.-C. Lu, "Sparse Inverse Preconditioning of Multilevel Fast Multipole Algorithm for Hybrid Integral Equations in Electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 52, no. 9, pp. 2277–2287, 2004.

[8] X. Ping, W. Yu, and T. Cui, "The Approximate Inverse Preconditioner for Finite Element Analysis of Helmholtz Equations," in *Asia Pacific Microw. Conf.*, Singapore, 2009.

[9] B. He and F. L. Teixeira, "Differential Forms, Galerkin Duality, and Sparse Inverse Approximations in Finite Element Solutions of Maxwell Equations," *IEEE Trans. Antennas Propag.*, vol. 55, no. 5, 2007.

[10] A. Samsonov, W. F. Block, and A. S. Field, "Reconstruction of MRI Data Using Sparse Matrix Inverses," in *Allerton Conf. on Comm., Control and Comp.*, Monticello IL, 2007, pp. 1884–1887.

[11] W. Zhang and P. Fung, "Sparse Inverse Covariance Matrices for Low Resource Speech Recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, , no. 99, pp. 1, 2012.

[12] W. Zhang and P. Fung, "Low Resource Speech Recognition with Automatically Learned Sparse Inverse Covariance Matrices," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, Kyoto, 2012.

[13] J. A. Bilmes, "Factored Sparse Inverse Covariance Matrices," in *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, Istanbul, 2000.

[14] S. Kaczmarz, "Angenäherte auflösung von systemen linearer gleichungen," *Bull. Internat. Acad. Polon. Sci. Lettres A*, vol. 35, pp. 355–357, 1937.

[15] T. Strohmer and R. Vershynin, "A Randomized Kaczmarz Algorithm with Exponential Convergence," *J. Fourier Anal. Appl.*, vol. 15, no. 2, pp. 262–278, Apr. 2008.

[16] E.-J. Im, *Optimizing the Performance of Sparse Matrix-Vector Multiplication*, Ph.D. thesis, EECS Department, University of California, Berkeley, Jun 2000.

[17] U. V. Catalyurek and C. Aykanat, "Hypergraph-Partitioning-Based Decomposition for Parallel Sparse-Matrix Vector Multiplication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 10, no. 7, 1999.

[18] E. J. Candes and M. B. Wakin, "An Introduction To Compressive Sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, 2008.

[19] A. Maleki and D. L. Donoho, "Optimally Tuned Iterative Reconstruction Algorithms for Compressed Sensing," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 330–341, 2010.

[20] S. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, Prentice Hall, 1993.