

Computation and applications of sparse generalized inverses

Ananias Sousa Machado
Universidade Federal do Rio de Janeiro
ananiasmachado@poli.ufrj.br

Marcia Fampa
Universidade Federal do Rio de Janeiro
fampa@cos.ufrj.br

Jon Lee
University of Michigan
jonxlee@umich.edu

ABSTRACT

Given $A \in \mathbb{R}^{m \times n}$, its Moore-Penrose (M-P) pseudoinverse $H \in \mathbb{R}^{n \times m}$ satisfies the properties: (P1) $AHA = A$, (P2) $HAH = H$, (P3) $AH = (AH)^\top$ and (P4) $HA = (HA)^\top$. We say that H is a generalized inverse of A if it satisfies (P1). We propose different approaches for computing sparse generalized inverses that can be used to efficiently compute least-squares solutions of overdetermined systems of linear equations and minimum 2-norm solutions of underdetermined systems, with comprehensive applications. We investigate mathematical models for computing these generalized inverses based on a theoretical study of the representation of proper subsets of the M-P properties as reduced systems of linear equations. We propose an ADMM (Alternating Direction Method of Multipliers) algorithm for computing generalized inverses that can be used to compute least-squares and minimum 2-norm solutions. We investigate the generalized Tikhonov problem and use its analysis to propose an alternative approach for computing least-squares solutions using generalized inverses.

KEYWORDS. generalized inverse. sparse optimization. least-squares. 2-norm minimization

OC- Continuous Optimization

1. Introduction

Generalized inverses are a key tool in matrix algebra and its applications. In particular, the Moore-Penrose (M-P) pseudoinverse can be used to calculate the least-squares solution of an overdetermined system of linear equations and the 2-norm minimizing solution of an under-determined system of linear equations. Sparse optimization aims at finding sparse solutions of optimization problems, often for computational efficiency in the use of the output of the optimization.

When a real matrix $A \in \mathbb{R}^{m \times n}$ is not square or not invertible, we consider generalized inverses of A (see [Rao and Mitra \[1972\]](#), for example). The most well-known generalized inverse is the M-P pseudoinverse, independently discovered by E.H. Moore and R. Penrose. If $A = U\Sigma V^\top$ is the real singular-value decomposition of A , then the M-P pseudoinverse of A can be defined as $A^\dagger := V\Sigma^\dagger U^\top$, where $\Sigma^\dagger$ has the shape of the transpose of the diagonal matrix Σ , and is derived from Σ by taking reciprocals of the non-zero (diagonal) elements of Σ (i.e., the non-zero singular values of A). The M-P pseudoinverse, a central object in matrix theory, has many concrete uses.

We seek to define different tractable sparse generalized inverses, based on the following fundamental characterization of the M-P pseudoinverse.

Theorem 1 ([Penrose \[1955\]](#)). For $A \in \mathbb{R}^{m \times n}$, the M-P pseudoinverse $A^\dagger$ is the unique $H \in \mathbb{R}^{n \times m}$ satisfying:

$$\text{(generalized inverse)} \quad AHA = A \quad (\text{P1})$$

$$\text{(reflexive)} \quad HAH = H \quad (\text{P2})$$

$$\text{(ah-symmetric)} \quad (AH)^\top = AH \quad (\text{P3})$$

$$\text{(ha-symmetric)} \quad (HA)^\top = HA \quad (\text{P4})$$

The terms “generalized inverse” and “reflexive” are completely standard. While less standard, “ah-symmetric” and “ha-symmetric” are convenient mnemonics. If H satisfies P3 and P4, we say that H is ah-ha-symmetric. Next, we present a key result for our developments.

Theorem 2 ([Ben-Israel and Greville \[2003\]](#), p. 208, Ex. 14 (no proof); also see [Ponte et al. \[2024b\]](#) (with proof)).

For rank r matrix $A \in \mathbb{R}^{m \times n}$, consider the full singular-value decomposition $A = U\Sigma V^\top$ with

$$\Sigma =: \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix},$$

$\begin{matrix} r \times r & r \times (n-r) \\ (m-r) \times r & (m-r) \times (n-r) \end{matrix}$

and D diagonal. For $H \in \mathbb{R}^{n \times m}$, let $\Gamma := V^\top H U$ (so $H = V\Gamma U^\top$), where we block partition Γ as

$$\Gamma =: \begin{bmatrix} X & Y \\ Z & W \end{bmatrix}.$$

$\begin{matrix} r \times r & r \times (m-r) \\ (n-r) \times r & (n-r) \times (m-r) \end{matrix}$

We have that

- (i) P1 is equivalent to $X = D^{-1}$.
- (ii) If P1 is satisfied, then P2 is equivalent to $ZDY = W$.
- (iii) If P1 is satisfied, then P3 is equivalent to $Y = 0$.
- (iv) If P1 is satisfied, then P4 is equivalent to $Z = 0$.

Organization and Contributions: Throughout §§2, 3 and 4, we present theoretical results on the expression of some proper subsets of M-P properties as a reduced system of equations. The subsets of properties are specifically selected so the generalized inverses satisfying them can be used to compute the least-squares solution of an overdetermined system of linear equations, or the 2-norm minimizing solution of an underdetermined system of linear equations. Some of the theoretical results presented here can be found in the literature, but we include them for the sake of completeness and also because we could not find proofs for some. In §5, we present an ADMM algorithm (see [Boyd et al. \[2011\]](#), for example) for computing ah-ha-symmetric generalized inverses, which we show can be used to compute both least-squares and minimum 2-norm solutions. In §6, we present the generalized Tikhonov regularization problem $\min_{\theta \in \mathbb{R}^n} \{\|A\theta - b\|_2^2 + \lambda \|L\theta\|_2^2\}$, $\lambda \geq 0$. We prove that its solution can be obtained with a generalized inverse of $\hat{A} = A^\top A + \lambda L^\top L$. Furthermore, we prove that generalized inverses of $\hat{A}$ are related to generalized inverses of A when $\lambda = 0$. In §7, we apply the results of §§2 and 3 to compute sparse generalized inverses efficiently, using the reduced systems of equations considered in §§2 and 3. We investigate the performance of the ADMM algorithm presented in §5, comparing it to other solution methods. Using the results of §6, we propose an alternative approach for computing least-squares solutions using generalized inverses and show that it can be numerically advantageous for sparse matrices.

Notation: $\|\cdot\|_2$ denotes vector 2-norm and $\|\cdot\|_1$ denotes vector 1-norm. $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. For brevity, we write $\|H\|_1$ to mean $\|\text{vec}(H)\|_1$. $\|H\|_0$ refers to the number of nonzero elements in the matrix H . $\mathcal{R}(A)$ denotes the range of matrix A , and $\mathcal{K}(A)$ denotes the kernel of A . We say that H satisfies $P_i + P_j$ if H satisfies both properties P_i and P_j , with $i, j \in \{1, 2, 3, 4\}$.

2. Sparse universal least-squares solvers

A universal least-squares solver for a matrix $A \in \mathbb{R}^{m \times n}$ is a matrix $H \in \mathbb{R}^{n \times m}$ such that $\hat{\theta} := Hb$ is a solution to the least-squares (estimation) problem

$$\min \|A\theta - b\|_2^2, \quad (\text{LSE})$$

for every $b \in \mathbb{R}^m$.

Proposition 1 (see e.g. [Rohde \[1964\]](#)). If H satisfies $P1 + P3$, then H is a universal least-squares solver for A .

Previously, we used several optimization ideas over the constraints $P1 + P3$ to seek sparse universal least-squares solvers for A . We are particularly interested in the case where A does not have full column rank; see [Fampa and Lee \[2018\]](#); [Xu et al. \[2021\]](#); [Ponte et al. \[2024b\]](#); [Fampa et al. \[2021a\]](#); [Ponte et al. \[2024a\]](#). Our methods also consider low norm (for numerical considerations) and low rank (for a kind of explainability) for H .

The well-known normal equations are the least-squares optimality conditions: $A^\top A\theta = A^\top b$. We are looking for an H so that the least-squares solution is $\hat{\theta} := Hb$. Substituting this into the normal equations gives $A^\top AHb = A^\top b$. Replacing b with all of the standard unit vectors, we see that we get simply

$$A^\top AH = A^\top. \quad (\text{PLS})$$

So, we could forget about the M-P properties $P1 + P3$, and simply look for sparse solutions of PLS. Ideally, the objective function would be $\|H\|_0$, however, because the 0-norm is non-convex, it would lead to an intractable problem. A form of making this problem

become tractable is replacing the 0-norm by the 1-norm as a way of inducing sparsity. So, considering the M-P properties, the problem would become $\min\{\|H\|_1 : P1 + P3\}$, which can be reformulated as a linear-optimization problem. Because $A \in \mathbb{R}^{m \times n}$ and $H \in \mathbb{R}^{n \times m}$, the constraint $P1+P3$ consists of $mn + m^2$ equations, where mn equations are from $P1$ and m^2 equations are from $P3$. Considering that the expected use of least-squares models are in scenarios where $m \gg n$ and A is not necessarily sparse, we can expect a dense matrix equation of constraints, and thus, the numerical resolution of the problem becomes difficult as the size of the instances increase. Seeking to find a solution to make bigger instances of the problem more tractable by reducing the number of equations in the constraints, we study representing the pair of properties $P1+P3$ as the linear system PLS, with less equations.

In Prop. 1, we saw that $\|A\theta - b\|_2^2$ is smallest for $\hat{\theta} := Hb$, where H satisfies $P1+P3$. In Lem. 1, we verify the equivalence between $P1+P3$ and PLS. This result is mentioned, for example, in Rao and Mitra [1972] and in Tian [2019], which presents a list of equivalences between systems of equations with respect to generalized inverses. Finally, in Thm. 3, we present a fundamental and well-known result characterizing the solution of LSE with PLS, using Lagrangian duality. The results confirm that we can alternatively solve $\min\{\|H\|_1 : PLS\}$ instead of $\min\{\|H\|_1 : P1 + P3\}$ when searching for sparse matrices H to solve LSE.

Lemma 1. H satisfies $P1+P3$ if and only if H satisfies PLS.

Proof.

1. ($\Rightarrow$) If H satisfies $P1+P3$, then H satisfies PLS.

Because H satisfies $P1$, we have $AHA = A \Rightarrow A^\top H^\top A^\top = A^\top \Rightarrow A^\top AH = A^\top$, where from the second to the third equation we used that H satisfies $P3$.

2. ($\Leftarrow$) If H satisfies PLS, then H satisfies $P1$ and $P3$.

Because H satisfies PLS, we have $A^\top AH = A^\top \Rightarrow H^\top A^\top AH = H^\top A^\top \Rightarrow (AH)^\top AH = (AH)^\top$. Because the last left-hand side is symmetric, we have that AH is symmetric, thus H satisfies $P3$. We also have $A^\top AH = A^\top \Rightarrow A^\top H^\top A^\top = A^\top \Rightarrow AHA = A$, where from the first to the second equation, we used that H satisfies $P3$. Then, H also satisfies $P1$. $\square$

Theorem 3. $\hat{\theta} := Hb$ solves LSE for every b if and only if H satisfies PLS.

Proof. Let $f(\theta) := \|A\theta - b\|_2^2$ be the objective function of LSE. Because LSE is convex, the optimal solution is such that the gradient of f evaluated at the optimal solution vanishes, with $\nabla f(\theta) = 2A^\top A\theta - 2A^\top b$.

1. ($\Rightarrow$) If $\hat{\theta} := Hb$ solves LSE, then H satisfies PLS.

Considering that the gradient of f vanishes at $\hat{\theta} := Hb$, we have $A^\top AHb = A^\top b$. Plugging in all standard unit vectors for b , we have $A^\top AH = A^\top$, then H satisfies PLS.

2. ($\Leftarrow$) If H satisfies PLS, then $\hat{\theta} := Hb$ solves LSE.

Because H satisfies PLS, by Lem. 1, we have that H satisfies $P1+P3$. Thus $\nabla f(Hb) = 2A^\top AHb - 2A^\top b = 2A^\top H^\top A^\top b - 2A^\top b = 2A^\top b - 2A^\top b = 0$, where we first used that H satisfies $P3$, and then that it satisfies $P1$. $\square$

3. Sparse universal minimum 2-norm solvers

A universal minimum 2-norm solver for a matrix $A \in \mathbb{R}^{m \times n}$ is a matrix $H \in \mathbb{R}^{n \times m}$ such that $\hat{\theta} := Hb$ is a solution to the minimum 2-norm problem

$$\min\{\|\theta\|_2^2 : A\theta = b\}, \quad (\text{M2N})$$

for every $b \in \mathcal{R}(A)$.

Proposition 2 (see e.g. Rohde [1964]). If H satisfies P1+P4, then H is a universal minimum 2-norm solver for A .

Following the ideas of §2, we study formulations to compute sparse universal minimum 2-norm solvers. In Prop. 2, we saw that H is a universal minimum 2-norm for A when it satisfies P1+P4. In Lem. 2, we verify the well-known equivalence between P1+P4 and the reduced linear system PMN. Finally, in Thm. 4, we characterize the solution of M2N with PMN, using Lagrangian duality. The results confirm that we can solve $\min\{\|H\|_1 : \text{PMN}\}$ instead of $\min\{\|H\|_1 : \text{P1+P4}\}$ when searching for sparse matrices H to solve M2N.

Lemma 2. H satisfies P1+P4 if and only if H satisfies PMN, where

$$AA^\top H^\top = A. \quad (\text{PMN})$$

Proof. The result follows directly from Lem. 1 and from the straightforward statements: (i) H satisfies P1 for A if and only if $H^\top$ satisfies P1 for $A^\top$; (ii) H satisfies P4 for A if and only if $H^\top$ satisfies P3 for $A^\top$. $\square$

Theorem 4. $\hat{\theta} := Hb$ solves M2N for every $b \in \mathcal{R}(A)$ if and only if H satisfies PMN.

Proof. To solve the convex optimization problem M2N, we consider the associated Lagrangian function $\mathcal{L}(\theta, \lambda) := \frac{1}{2}\theta^\top \theta + \lambda^\top (b - A\theta)$, and we note that $\nabla_\theta \mathcal{L}(\theta, \lambda) = \theta - A^\top \lambda$.

In the remainder of this proof, we consider the result of Lem. 2, i.e., we consider that PMN is equivalent to P1+P4.

1. ($\Rightarrow$): If $\hat{\theta} := Hb$ solves M2N, for every $b \in \mathcal{R}(A)$, then H satisfies PMN.

Because $\hat{\theta} := Hb$ solves M2N, for every $b \in \mathcal{R}(A)$, we have $AHb = b$. Because this equation holds for every $b \in \mathcal{R}(A)$, we can plug in all the columns of A , and we arrive at $AHA = A$. Thus, H satisfies P1.

Also, because $\hat{\theta} := Hb$ solves M2N, $\nabla_\theta \mathcal{L}(Hb, \lambda) = Hb - A^\top \lambda$ vanishes for some λ , hence, $Hb = A^\top \lambda$ for some λ . Because this equation holds for every $b \in \mathcal{R}(A)$, we can plug in all the columns of A , and we arrive at $HA = A^\top \Lambda$. Because H satisfies P1, we have $A = AHA = AA^\top \Lambda \Rightarrow A = AA^\top \Lambda$. Thus, by Lem. 2, $\Lambda^\top$ satisfies P1+P4.

Now we show that this implies that H satisfies P4. From $HA = A^\top \Lambda$, we have $HA = A^\top \Lambda \Rightarrow HA(HA)^\top = (A\Lambda^\top A)^\top H^\top \Rightarrow HA(HA)^\top = A^\top H^\top \Rightarrow HA(HA)^\top = (HA)^\top$, where from the second to the third equation we used that $\Lambda^\top$ satisfies P1. Because the last left-hand side is symmetric, we have that HA is symmetric. Therefore, H satisfies P4.

2. ($\Leftarrow$): If H satisfies PMN, then $\hat{\theta} := Hb$ is a solution of M2N, for every $b \in \mathcal{R}(A)$.

We want to prove that if H satisfies PMN and b is in the range of A , then there exists $\lambda \in \mathbb{R}^m$ such that $\nabla_\theta \mathcal{L}(Hb, \lambda) = Hb - A^\top \lambda = 0$. Because b is in the range of A , there exists $y \in \mathbb{R}^n$ such that $b = Ay$, and then $HAy - A^\top \lambda$. Now, we claim that $\lambda = H^\top y$ is such that the gradient vanishes. We have $\nabla_\theta \mathcal{L}(Hb, H^\top y) = Hb - A^\top H^\top y = Hb - (HA)^\top y = Hb - HAy = 0$, where from the second to the third equation we used that H satisfies P4. $\square$

4. Sparse universal solvers for least-squares and minimum 2-norm problems

We have seen that H is a universal least-squares solver (for b not in the range of A) if and only if H satisfies P1+P3, and H is universal minimum 2-norm solver (for $b \in \mathcal{R}(A)$) if and only if H satisfies P1+P4. So, if we want a single sparse matrix H that is both a universal least-squares solver and a universal minimum 2-norm solver, we want a sparse ah-ha symmetric generalized inverse. From Lem. 1 we have that H is an ah-symmetric

generalized inverse if and only if H satisfies PLS, and from Lem. 2 we have that H is a ha-symmetric generalized inverse if and only if H satisfies PMN, it turns out that we can transpose the first equation, sum with the second equation and obtain a reduced system of equations that H satisfies it if and only if H is a ah-ha symmetric generalized inverse. Next, we present a theorem, which we did not find in the literature, that characterizes ah-ha symmetric reflexive generalized inverses.

Lemma 3. H satisfies P1+P3+P4 if and only if H satisfies PMX, where

$$AA^T H^T + H^T A^T A = 2A. \quad (\text{PMX})$$

Proof.

1. ($\Rightarrow$) If H satisfies P1+P3+P4, then H satisfies $AA^T H^T + H^T A^T A = 2A$.

Because H satisfies P1, we have $AHA + AHA = 2A \Rightarrow AA^T H^T + H^T A^T A = 2A$, where we first used that H satisfies P4 and then that H satisfies P3.

2. ($\Leftarrow$) If H satisfies $AA^T H^T + H^T A^T A = 2A$, then H satisfies P1+P3+P4.

First, we consider the full singular-value decomposition of A , given by $A = U\Sigma V^T$ and the representation of H given by $H = V\Gamma U^T$ (see Thm. 2). Because H satisfies $AA^T H^T + H^T A^T A = 2A$, we have

$$\begin{aligned} (U\Sigma V^T)(V\Sigma^T U^T)(U\Gamma^T V^T) + (U\Gamma^T V^T)(V\Sigma^T U^T)(U\Sigma V^T) &= 2U\Sigma V^T \Rightarrow \\ U^T U \Sigma \Sigma^T \Gamma^T V^T V + U^T U \Gamma^T \Sigma^T \Sigma V^T V &= 2U^T U \Sigma V^T V \Rightarrow \Sigma \Sigma^T \Gamma^T + \Gamma^T \Sigma^T \Sigma = 2\Sigma \Rightarrow \\ \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} D^T & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X^T & Z^T \\ Y^T & W^T \end{bmatrix} + \begin{bmatrix} X^T & Z^T \\ Y^T & W^T \end{bmatrix} \begin{bmatrix} D^T & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} &= 2 \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \Rightarrow \\ \begin{bmatrix} DD^T X^T + X^T D^T D & DD^T Z^T \\ Y^T D^T D & 0 \end{bmatrix} &= 2 \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}, \end{aligned}$$

where from the first to the second equation we pre-multiplied by U^T and post-multiplied by V . Because D is diagonal with non-zero diagonal elements, the only solution to the equation is $Z = 0$ and $Y = 0$. Finally, from $DD^T X^T + X^T D^T D = 2D$ we have

$$x_{ij}d_{jj}^2 + d_{ii}^2 x_{ij} = \begin{cases} 2d_{ii}, & \text{if } i = j; \\ 0, & \text{otherwise.} \end{cases}$$

Thus, $x_{ij} = 0$ for all $i \neq j$, and $x_{ii} = \frac{1}{d_{ii}}$ for all i . So, $X = D^{-1}$. Hence, by Thm. 2, H satisfies P1. As H satisfies P1, $Y = 0$ and $Z = 0$, we have again by Thm. 2 that H satisfies P3 and P4. $\square$

5. ADMM for computing sparse ah-ha symmetric generalized inverses

Consider the following convex-optimization problem aimed at inducing sparsity at universal solvers for the least-squares and the minimum 2-norm problems, or equivalently, at ah-ha symmetric generalized inverses:

$$\min_{H \in \mathbb{R}^{n \times m}} \{ \|H\|_1 : \text{P1} + \text{P3} + \text{P4} \} \quad (P_{134}^1)$$

From Thm. 2, we see that any feasible solution to P_{134}^1 satisfies $H = V\Gamma U^T$, where $\Gamma =$:

$$\begin{bmatrix} D^{-1} & 0 \\ 0 & W \end{bmatrix} \begin{matrix} r \times r & r \times (m-r) \\ (n-r) \times r & (n-r) \times (m-r) \end{matrix}. \text{ Then, defining } V =: \begin{bmatrix} V_1 & V_2 \\ n \times r & n \times (m-r) \end{bmatrix}, U =: \begin{bmatrix} U_1 & U_2 \\ m \times r & m \times (m-r) \end{bmatrix}, \text{ we reformulate } P_{134}^1 \text{ as}$$

$$\min_{W \in \mathbb{R}^{(n-r) \times (m-r)}} \{ \|V_1 D^{-1} U_1^T + V_2 W U_2^T\|_1 \} \quad (\mathcal{P}_{134}^1)$$

We remark that U_1 , U_2 and V_2 are full column rank matrices and $U_1^\top U_1 = I$, $U_2^\top U_2 = I$ and $V_2^\top V_2 = I$, then $U_1^\dagger = (U_1^\top U_1)^{-1} U_1^\top = U_1^\top$, $U_2^\dagger = (U_2^\top U_2)^{-1} U_2^\top = U_2^\top$ and $V_2^\dagger = (V_2^\top V_2)^{-1} V_2^\top = V_2^\top$.

Next, we consider the ADMM method to solve problem $\mathcal{P}_{134}^1$. Because problem $\mathcal{P}_{134}^1$ is very similar to problem $\mathcal{P}_{123}^1$ investigated in [Ponte et al. \[2024a\]](#), our goal is to adapt the ADMM algorithm discussed in [\[Ponte et al., 2024a, Sect. 2.1\]](#) to $\mathcal{P}_{134}^1$ and verify its computational efficiency, running numerical experiments. Initially, introducing a variable $E \in \mathbb{R}^{n \times m}$, we rewrite $\mathcal{P}_{134}^1$ as

$$\min\{\|E\|_1 : E = V_1 D^{-1} U_1^\top + V_2 W U_2^\top\}. \quad (1)$$

The augmented Lagrangian function associated to (1) is

$$\begin{aligned} \mathcal{L}_\rho(W, E, \Lambda) &:= \|E\|_1 + \langle \Theta, V_1 D^{-1} U_1^\top + V_2 W U_2^\top - E \rangle + \frac{\rho}{2} \|V_1 D^{-1} U_1^\top + V_2 W U_2^\top - E\|_F^2 \\ &= \|E\|_1 + \frac{\rho}{2} \|V_1 D^{-1} U_1^\top + V_2 W U_2^\top - E + \Lambda\|_F^2 - \frac{\rho}{2} \|\Lambda\|_F^2, \end{aligned}$$

where $\rho > 0$ is the penalty parameter, $\Theta \in \mathbb{R}^{n \times m}$ is the Lagrange multiplier, and Λ is the scaled Lagrange multiplier, that is $\Lambda := \Theta/\rho$. We will apply the ADMM method to $\mathcal{P}_{134}^1$, by iteratively solving, for $k = 0, 1, \dots$,

$$W^{k+1} := \operatorname{argmin}_W \mathcal{L}_\rho(W, E^k, \Lambda^k), \quad (2)$$

$$E^{k+1} := \operatorname{argmin}_E \mathcal{L}_\rho(W^{k+1}, E, \Lambda^k), \quad (3)$$

$$\Lambda^{k+1} := \Lambda^k + V_1 D^{-1} U_1^\top + V_2 W^{k+1} U_2^\top - E^{k+1}.$$

Next, we detail how to solve the subproblems above.

5.1. Update W

To update W , we consider (2). Defining $J := E^k - V_1 D^{-1} U_1^\top - \Lambda^k$, we solve

$$W^{k+1} = \operatorname{argmin}_W \{\|J - V_2 W U_2^\top\|_F^2\}. \quad (4)$$

Proposition 3. The solution of (4) is given by $W^{k+1} = V_2^\top J U_2$.

Proof. We have $\nabla_W \|J - V_2 W U_2^\top\|_F^2 = -2V_2^\top J U_2 + 2V_2^\top V_2 W U_2^\top U_2 = -2V_2^\top J U_2 + 2W U_2^\top U_2$. Then, the result simply follows because the gradient is zero at the minimum. $\square$

5.2. Update E

To update E , we consider (3). Defining $Y := V_1 D^{-1} U_1^\top + V_2 W^{k+1} U_2^\top + \Lambda^k$, we solve

$$E^{k+1} = \operatorname{argmin}_E \{\|E\|_1 + \frac{\rho}{2} \|E - Y\|_F^2\}. \quad (5)$$

Proposition 4 (see [\[Boyd et al., 2011, Sect. 4.4.3\]](#)). The solution of (5) is given by $E_{ij}^{k+1} = S_{1/\rho}(Y_{ij})$, for $i = 1, \dots, n$ and $j = 1, \dots, m$, where the soft thresholding operator S is defined as

$$S_\kappa(a) := \begin{cases} a - \kappa, & a > \kappa; \\ 0, & |a| \leq \kappa; \\ a + \kappa, & a < -\kappa. \end{cases}$$

5.3. Initialization of the variables

Initialization is quite important for obtaining rapid convergence. To initialize Λ , we consider the dual problem of (1),

$$\max_{\Theta} \{\operatorname{tr}(D^{-1} V_1^\top \Theta U_1) : V_2^\top \Theta U_2 = 0, \|\Theta\|_\infty \leq 1\}. \quad (6)$$

Then, we set $\hat{\Theta} := \frac{1}{\|V_1 U_1^\top\|_\infty} V_1 U_1^\top$, which is feasible to (6) due to the orthogonality of V , and $\Lambda^0 := \hat{\Theta}/\rho$. From Thm. 2, we see that ah-ha symmetric generalized inverses of A can be written as $V_1 D^{-1} U_1^\top + V_2 W U_2^\top$. We also see that the M-P pseudoinverse $A^\dagger$ can be written as $V_1 D^{-1} U_1^\top$, and we recall that $A^\dagger$ is the generalized inverse of A with minimum Frobenius norm. Then, aiming at obtaining $W^1 = 0$ when solving (4) at the first iteration of the algorithm, and consequently starting the algorithm with a Frobenius-norm minimizing ah-ha symmetric generalized inverse, we set $E^0 := V_1 D^{-1} U_1^\top + \Lambda^0$.

5.4. Stopping criteria

We consider a stopping criterion from [Boyd et al., 2011, Sect. 3.3.1], and select an absolute tolerance ϵ^{abs} and a relative tolerance ϵ^{rel} . The algorithm stops at iteration k if

$$\begin{aligned} \|r^k\|_F &\leq \epsilon^{\text{abs}} \sqrt{nm} + \epsilon^{\text{rel}} \max\{\|E^k\|_F, \|V_2 W^k U_2^\top\|_F, \|V_1 D^{-1} U_1^\top\|_F\}, \\ \|s^k\|_F &\leq \epsilon^{\text{abs}} \sqrt{(n-r)r} + \epsilon^{\text{rel}} \rho \|V_2^\top \Lambda^k U_2\|_F, \end{aligned}$$

where $r^k := V_1 D^{-1} U_1^\top + V_2 W^k U_2^\top - E^k$, $s^k := \rho V_2^\top (E^k - E^{k-1}) U_2$ are primal and dual residuals.

6. Generalized Tikhonov regularization

Given $\lambda \geq 0$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $L \in \mathbb{R}^{p \times n}$, we consider the following well-known generalization of the Tikhonov regularization problem

$$\min_{\theta \in \mathbb{R}^n} \{ \|A\theta - b\|_2^2 + \lambda \|L\theta\|_2^2 \}. \quad (\text{G}_L)$$

We would like to mention that for $L := I_n$, this is commonly known as ridge regression or regression with ℓ_2 -regularization. And for $L := I_n$, but replacing the squared 2-norm with the 1-norm, we obtain instead what is commonly known as lasso regression or regression with ℓ_1 -regularization.

From convexity, we have that $\hat{\theta}$ solves G_L if and only if the gradient of the objective function of G_L vanishes at $\hat{\theta}$, that is

$$A^\top A \hat{\theta} + \lambda L^\top L \hat{\theta} - A^\top b = 0 \Leftrightarrow (A^\top A + \lambda L^\top L) \hat{\theta} = A^\top b. \quad (7)$$

Also, from the convexity of the quadratic objective function of G_L , we have that the linear system (7) is solvable for every $\lambda \geq 0$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $L \in \mathbb{R}^{p \times n}$.

Theorem 5. If $\hat{H}$ is a generalized inverse of $\hat{A} := A^\top A + \lambda L^\top L$, then $\hat{\theta} := \hat{H} A^\top b$ solves G_L .

Proof. Let $f(\theta) = \|A\theta - b\|_2^2 + \lambda \|L\theta\|_2^2$. We show that the gradient of f evaluated at $\hat{\theta} = \hat{H} A^\top b$ vanishes. We have $\nabla_\theta f(\hat{H} A^\top b) = 2(A^\top A + \lambda L^\top L) \hat{H} A^\top b - 2A^\top b = 2\hat{A} \hat{H} A^\top b - 2A^\top b$. Because (7) is solvable, $A^\top b$ is in the range of $\hat{A}$, thus, there exists $y \in \mathbb{R}^n$ such that $\hat{A} y = A^\top b$. Replacing $A^\top b$ with $\hat{A} y$ in the expression of the gradient we have, $\nabla_\theta f(\hat{H} A^\top b) = 2\hat{A} \hat{H} \hat{A} y - 2\hat{A} y = 2\hat{A} y - 2\hat{A} y = 0$, where we used that $\hat{H}$ satisfies P1 for $\hat{A}$. $\square$

The result above does not seem to be well known. We now study the case when $\lambda = 0$.

Theorem 6. There are generalized inverses H and $\hat{H}$ of A and $\hat{A}$, respectively, with $H = \hat{H} A^\top$.

Proof. First, we consider the full singular-value decomposition of $A \in \mathbb{R}^{m \times n}$, given by $A = U \Sigma V^\top$ and the representation of $H \in \mathbb{R}^{n \times m}$ given by $H = V \Gamma U^\top$ (see Thm. 2), where $\Sigma =:$

$$\begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}_{\substack{r \times r & r \times (n-r) \\ (m-r) \times r & (m-r) \times (n-r)}} \text{ and } \Gamma =: \begin{bmatrix} X & Y \\ Z & W \end{bmatrix}_{\substack{r \times r & r \times (m-r) \\ (n-r) \times r & (n-r) \times (m-r)}}, \text{ with } D \text{ diagonal. Note that } A^\top A = V \Sigma^2 V^\top \text{ is}$$

the full singular-value decomposition of $A^\top A$. Let $\hat{H} \in \mathbb{R}^{m \times m}$ and $\hat{H} = V \Delta V^\top$, where $\Delta =:$

$\begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix}$ where $\Delta_{11} \in \mathbb{R}^{r \times r}$, $\Delta_{12} \in \mathbb{R}^{r \times (n-r)}$, $\Delta_{21} \in \mathbb{R}^{(n-r) \times r}$, and $\Delta_{22} \in \mathbb{R}^{(n-r) \times (n-r)}$. Applying Thm. 2 to A and $A^\top A$, we have that H is a generalized inverse of A if and only if $X = D^{-1}$, and $\hat{H}$ is a generalized inverse of $A^\top A$ if and only if $\Delta_{11} = D^{-2}$. Then, $H = V \begin{bmatrix} D^{-1} & Y \\ Z & W \end{bmatrix} U^\top$ and $\hat{H}A^\top = V \begin{bmatrix} D^{-2} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix} VV^\top \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} U^\top = V \begin{bmatrix} D^{-1} & 0 \\ \Delta_{21}D & 0 \end{bmatrix} U^\top$. Thus, we have that $H = \hat{H}A^\top \Leftrightarrow Y = W = 0, Z = \Delta_{21}D$. $\square$

Corollary 1. If $H = \hat{H}A^\top$ holds with H being a generalized inverse of A and $\hat{H}$ being a generalized inverse of $A^\top A$, then H is an ah-symmetric reflexive generalized inverse of A .

Proof. The result follows by noting that $X = D^{-1}$, $Y = 0$, $W = 0$ and applying Thm. 2. $\square$

7. Numerical Experiments

The programming language used was Julia v1.11.3 (using VSCode v1.97.1), and we ran the experiments on “zebra”, a 16-core machine (running Windows Server 2019 Standard) Intel Xeon E5-2667 v4 processors running at 3.20GHz with 8 cores each, and 128 GB of memory. Unless otherwise noted, we solved the optimization problems involved in our experiments with Gurobi v12.0.1, using parameters: Barrier convergence tolerance, Feasibility tolerance, and Optimality tolerance: 10^{-5} . The time limit to solve each instance was 1200 seconds. To compute the 0-norm, we consider the tolerance for non-zero element as 10^{-5} . All results presented are averages for 5 instances. The symbol ‘*’ indicates that it was not possible to solve at least one instance with the given configuration, in our time limit.

We used the MATLAB function sprand to generate the matrices for our test instances. It generates a random $m \times n$ dimensional matrix A with rank r and approximate density d , and with the r positive singular values given by the input vector rc . Following Fampa et al. [2021b], we selected rc as the decreasing vector $M \times (\rho^1, \rho^2, \dots, \rho^r)$, where $M = 2$, and $\rho = (1/M)^{(2/(r+1))}$.

To compare the different formulations proposed in §2 to compute sparse universal least-squares solvers, we generated 50 dense matrices A (5 for each triple (m, n, r) , for all, we have $d = 1$). We set $m = 60, 120, \dots, 540$, $n = 0.5m$, $r = 0.25m$. In Table 1, we present results comparing the use of equations PLS to the use of P1+P3 when seeking sparse universal least-squares solvers. We consider the following problems:

$$\begin{aligned} \min_{H \in \mathbb{R}^{n \times m}} \{ \|H\|_1 : P1 + P3 \} & \quad (P_{13}^1) \\ \min_{H \in \mathbb{R}^{n \times m}} \{ \|H\|_1 : PLS \} & \quad (P_{PLS}^1) \end{aligned}$$

m	n	r	$\frac{\ H\ _0}{\ A^\top\ _0}$	$\frac{\ H\ _1}{\ A^\top\ _1}$	$\frac{r(H)}{r(A^\top)}$	Time (sec)	
						P_{13}^1	P_{PLS}^1
60	30	15	0.502	0.839	1.893	24.802	0.166
120	60	30	0.505	0.821	1.927	1034.507	1.782
180	90	45	0.511	0.820	1.933	*	8.389
240	120	60	0.508	0.820	1.963	*	28.777
300	150	75	0.514	0.823	1.963	*	64.199
360	180	90	0.516	0.813	1.962	*	128.553
420	210	105	0.517	0.815	1.970	*	244.081
480	240	120	0.519	0.816	1.962	*	497.880
520	260	130	0.519	0.813	1.982	*	635.693
540	270	135	*	*	*	*	*

Table 1: Comparison of P_{13}^1 and P_{PLS}^1

The 0-norm and 1-norm of the solutions found by P_{13}^1 and P_{PLS}^1 were approximately the same, as expected, so we present only the average 0-norm and 1-norm found by P_{PLS}^1 . As can be observed in Table 1, problem P_{13}^1 was not solvable within the time limit for instances of $m \geq 180$, while P_{PLS}^1 was solvable within the time limit for instances of size $m \leq 520$. It is also worth noting that the time taken to solve P_{PLS}^1 was at least 2 orders of magnitude less than the time taken to solve P_{13}^1 , and that for the biggest instance solved by P_{13}^1 the time taken by P_{PLS}^1 was 3 orders of magnitude less. The solution of P_{PLS}^1 was expected to be more efficient than P_{13}^1 due to the reduced number of constraints, however, such a difference in efficiency was not expected. By analyzing the trend in the time taken to solve P_{PLS}^1 we would expect that the instances of size $m = 540$ would be solvable, however, this was not the case, which shows that this trend is nonlinear in m . We note that the solutions obtained are much sparser than the M-P pseudoinverse and with a smaller 1-norm, satisfying our goal.

We also solved problems $\min_{H \in \mathbb{R}^{n \times m}} \{\|H\|_1 : P1 + P4\}$ and $\min_{H \in \mathbb{R}^{n \times m}} \{\|H\|_1 : PMN\}$ to compare the use of equations PMN to the use of P1+P4, when seeking sparse universal minimum 2-norm solvers. The comparison results were similar to those presented in Table 1.

We now discuss numerical experiments that show the efficacy of the ADMM algorithm to solve $\mathcal{P}_{134}^1$, presented in §5. We compare it to the solver Gurobi when applied to different optimization problems to compute minimum 1-norm ah-ha symmetric generalized inverses, using the systems of equations PMN and PMX, to equivalently model P1+P3+P4. We solve $\mathcal{P}_{134}^1$ using two versions of the proposed ADMM algorithm, one with the proposed stopping criterion, and another with a fixed primal and dual tolerance to make a fairer comparison with the solver Gurobi. The problems solved by Gurobi are the following.

$$\begin{aligned} \min_{H \in \mathbb{R}^{n \times m}} \{\|H\|_1 : P1 + P3 + P4\} & \quad (P_{134}^1) \\ \min_{H \in \mathbb{R}^{n \times m}} \{\|H\|_1 : PMN + P3\} & \quad (P_{PMN,3}^1) \\ \min_{H \in \mathbb{R}^{n \times m}} \{\|H\|_1 : PMX\} & \quad (P_{PMX}^1) \end{aligned}$$

The parameters used for ADMM₁ are: $\epsilon^{\text{abs}} = \epsilon^{\text{rel}} := 10^{-5}$, $\rho := 3$; for ADMM₁ ^{ϵ} are: $\epsilon := 10^{-5}$, $\rho := 3$, where ADMM₁ is the proposed ADMM algorithm with the proposed stopping criterion, ADMM₁ ^{ϵ} is the proposed ADMM algorithm with fixed primal and dual tolerance ϵ . For these experiments, we randomly generated 140 test instances. More specifically, we generated 140 dense (density $d = 1.0$) square matrices with the MATLAB function `sprand`, in the same way described previously. We set $m = 60, 80, 100, \dots, 600$, $n = m$, $r = 0.25m$.

m	r				Time (sec)		
		$\frac{\ H\ _0}{\ A^\top\ _0}$	$\frac{\ H\ _1}{\ A^\top\ _1}$	$\frac{r(H)}{r(A^\top)}$	P_{134}^1	$P_{PMN,3}^1$	P_{PMX}^1
60	15	0.452	0.880	3.907	261.277	13.478	19.047
80	20	0.452	0.888	3.91	*	137.975	131.492
100	25	0.454	0.888	3.912	*	418.176	*
120	30	*	*	*	*	*	*

Table 2: Comparison of P_{134}^1 , $P_{PMN,3}^1$, and P_{PMX}^1

In Table 2, we see that the times to solve problems $P_{PMN,3}^1$ and P_{PMX}^1 are both one order of magnitude less than the time to solve P_{134}^1 , which shows that using either equations PMN+P3 or PMX allows us to solve the problem much more efficiently. Comparing the times to solve $P_{PMN,3}^1$ and P_{PMX}^1 , we observe that for the given time limit, $P_{PMN,3}^1$ solved the problem for instances of $m = 60$ faster than P_{PMX}^1 , but the latter solved the problem faster

for instances of $m = 80$, which does not make it clear which equations solve the problem faster in general, however, as $P_{\text{PMN},3}^1$ solved all instances of $m = 100$ within the time limit of 1200 while P_{PMX}^1 did not, it seems that solving problem $P_{\text{PMN},3}^1$ is the most efficient choice.

m	r	ADMM ₁				ADMM ₁ ^ε			
		$\frac{\ H\ _0}{\ A^\top\ _0}$	$\frac{\ H\ _1}{\ A^\top\ _1}$	$\frac{r(H)}{r(A^\top)}$	Time (sec)	$\frac{\ H\ _0}{\ A^\top\ _0}$	$\frac{\ H\ _1}{\ A^\top\ _1}$	$\frac{r(H)}{r(A^\top)}$	Time (sec)
60	15	0.465	0.880	3.893	1.135	0.452	0.880	3.907	20.693
120	30	0.463	0.888	3.933	24.36	0.454	0.888	3.933	235.741
180	45	0.466	0.882	3.924	31.245	0.456	0.882	3.929	411.739
240	60	0.467	0.886	3.943	38.157	0.456	0.886	3.943	682.229
300	75	0.470	0.885	3.949	48.25	*	*	*	*
360	90	0.474	0.882	3.958	61.29	*	*	*	*
420	105	0.477	0.883	3.962	130.595	*	*	*	*
480	120	0.480	0.885	3.963	176.703	*	*	*	*
540	135	0.485	0.885	3.964	241.725	*	*	*	*
600	150	0.484	0.883	3.965	293.959	*	*	*	*

Table 3: Comparison of ADMM₁ and ADMM₁^ε for $\mathcal{P}_{134}^1$

In Table 3, we compare the two versions of the ADMM algorithm to solve $\mathcal{P}_{134}^1$ for all instances with $m = 60, 120, \dots, 600$. We observe that ADMM₁ managed to solve the problems for all instances, while ADMM₁^ε managed to solve all 5 instances only for $m \leq 240$. Comparing the stats in Table 3, we have that the solutions found by ADMM₁^ε had smaller 0-norm and similar 1-norm, which shows that ADMM₁^ε finds better solutions than ADMM₁, however, it comes at the cost of having a time one order of magnitude bigger than ADMM₁.

Comparing the results in Tables 2 and 3, we see that for the instances solvable by Gurobi, all the methods found very similar solutions with relation to the 1-norm, however, Gurobi and ADMM₁^ε found solutions with smaller 0-norm. However, the highest value of m for which Gurobi solved the problems for all 5 instances was $m = 100$, which is smaller than the highest value of m for which ADMM₁^ε solved the problems for all instances ($m = 240$). Besides, considering that ADMM₁ managed to solve all instances, we can conclude that Gurobi is not a scalable method to solve this problem compared to ADMM, especially if we allow the tolerance given by the stopping criterion proposed in §5.

Finally, from the discussion in §6, we see that: (i) $\hat{H}$ is a generalized inverse of $\hat{A} := A^\top A$, if and only if $\hat{\theta} := \hat{H}A^\top b$ solves the least-squares problem G_L with $\lambda = 0$, for all b , (ii) H is an ah-symmetric generalized inverse of A , if and only if $\hat{\theta} := Hb$ solves the least-squares problem G_L with $\lambda = 0$, for all b . To compare Approaches (i) and (ii) to solving LSE, we generated 20 matrices A (5 for each (m, n, r, d)) with the MATLAB function sprand. We set $m = 120$, $n = 40, 60$, $r = 0.75n$, and density $d = 0.1, 0.25$. Because $\hat{A}$ is symmetric, we seek symmetric generalized inverses of $\hat{A}$ in Approach (i). Considering Corollary 1, we seek ah-symmetric reflexive generalized inverses of A in Approach (ii). We solve:

$$\begin{aligned} \min_{\hat{H} \in \mathbb{R}^{n \times n}} \{ \|\hat{H}\|_1 : \hat{A}\hat{H}\hat{A} = \hat{A}, \hat{H} = \hat{H}^\top \}. & \quad (\hat{P}_{1,\text{sym}}^1) \\ \min \{ \|H\|_1 : P1 + P2 + P3 \}. & \quad (P_{123}^1) \end{aligned}$$

For P_{123}^1 , we apply an ADMM algorithm from Ponte et al. [2024a]. We note that with Approach (i), the number of multiplications performed to compute Hb (assuming b is dense) is $\|H\|_0$, while with (ii), the number of multiplications performed to compute $\hat{H}A^\top b$ is $\|\hat{H}\|_0 + \|A\|_0$ (if we first compute $y := A^\top b$ and then compute $\hat{H}y$). From Table 4, we

see that we always have $\|\hat{H}\|_0 + \|A\|_0 < \|H\|_0$. The average reduction in the number of multiplications using Approach (ii) is over 50%. From these experiments, we see that Approach (ii) can be numerically advantageous for solving LSE with sparse matrices A .

n, d	$\ A\ _0$	$\hat{P}_{1,\text{sym}}^1$			P_{123}^1		
		$\ \hat{H}\ _0$	$\ \hat{H}\ _1$	Time	$\ H\ _0$	$\ H\ _1$	Time
40, 0.10	562.0	894.8	40.814	3.897	4317.4	112.853	3.225
40, 0.25	1273.6	894.8	16.280	2.471	4560.6	92.583	3.266
60, 0.10	854.0	2016.6	94.778	20.520	6521.4	204.301	4.202
60, 0.25	1900.4	2016.6	43.332	25.130	6612.2	162.332	3.543

Table 4: Comparison of $\hat{P}_{1,\text{sym}}^1$ and P_{123}^1

References

- Ben-Israel, A. and Greville, T. N. (2003). Generalized Inverses: Theory and Applications. Springer, 2 edition.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122.
- Fampa, M. and Lee, J. (2018). On sparse reflexive generalized inverses. *Operations Research Letters*, 46(6):605–610.
- Fampa, M., Lee, J., and Ponte, G. (2021a). Trading off 1-norm and sparsity against rank for linear models using mathematical optimization: 1-norm minimizing partially reflexive ah-symmetric generalized inverses. *Open Journal of Mathematical Optimization*, 2:1–14.
- Fampa, M., Lee, J., Ponte, G., and Xu, L. (2021b). Experimental analysis of local searches for sparse reflexive generalized inverses. *Journal of Global Optimization*, 81:1057–1093.
- Penrose, R. (1955). A generalized inverse for matrices. *Proceedings of the Cambridge Philosophical Society*, 51:406–413.
- Ponte, G., Fampa, M., Lee, J., and Xu, L. (2024a). Good and fast row-sparse ah-symmetric reflexive generalized inverses. <https://arxiv.org/abs/2401.17540>.
- Ponte, G., Fampa, M., Lee, J., and Xu, L. (2024b). On computing sparse generalized inverses. *Operations Research Letters*, 52:107058.
- Rao, C. R. and Mitra, S. K. (1972). Generalized inverse of a matrix and its applications. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability. Volume I: Theory of Statistics* (Lucien M. Le Cam, Jerzy Neyman, Elizabeth L. Scott, eds., University of California Press), p. 601–620.
- Rohde, C. A. (1964). Contributions to the Theory, Computation and Application of Generalized Inverses. PhD thesis, University of North Carolina, Raleigh, N.C.
- Tian, Y. (2019). Reverse order laws for generalized inverses of products of two or three matrices with applications. <https://arxiv.org/abs/1912.05948>.
- Xu, L., Fampa, M., Lee, J., and Ponte, G. (2021). Approximate 1-norm minimization and minimum-rank structured sparsity for various generalized inverses via local search. *SIAM Journal on Optimization*, 31(3):1722–1747.