Abstract

The generalized conditioned inverse is an order \( N^3 \) method to generate an inverse of a singular or near singular matrix, using complete pivoting. Instead of eigenvectors, the inverse is expressed in terms of a left and right representative space, such that the dimension of the problem is reduced in a trivial manner, limiting the bookkeeping. The null space vectors can be constructed to test the existence of a solution.

1 Introduction

The inverse of a singular or ill-posed matrix is hard problem, which is generally solved by an expensive spectral decomposition. After which the inverse are the transpose of the left and right eigenvectors, combined with the reciprocal of the non-zero eigenvalues. The zero eigenvalues are kept zero in the inverse, such that the inverse projects on the image space of the original matrix. Generally, spectral decompositions, besides being expensive, are also prone to numerical instabilities. We propose another type of inverse, guided by complete pivoting.

In some cases, the linear equation might not be strictly singular, but ill-posed, due to small eigenvalues, or columns or rows which are nearly parallel. In the inverse, they yield large errors, while the underlying problem might be respectable. It could just be that in the process of discretization of a continuous problem, or the linearization of a general non-linear problem some wrong choices were made. For such cases one would like to wield out the bad components, and let the inverse depend on the proper directions spanned by the linear equation only. A tolerance should truncate at a given level. With complete pivoting, there is a restriction on all successive pivots given a particular pivot. Since the product of pivots is the determinant of the matrix, the pivot itself serves as bound on the norm of the submatrix. Therefore, complete pivoting is an appropriate mean to truncate an ill-posed problem to a proper one.

Pivoting is the successive operation of a weighted row subtraction \( A_i \rightarrow A_{i+1} \) on a matrix \( A \), such that a particular column contains all zeros except for the pivot in one row. If the choice of pivot is the maximal element in all rows and columns, it is called complete pivoting. If for a given column the maximal element is picked, it is called partial pivoting. The set of subtraction operations define a left inverse. If one uses column subtraction instead of row subtraction, one would yield the right inverse, which should be identical, up to numerical round-off, and singularity ambiguities. General one seeks the left inverse of a matrix in the case of a linear equation of the form

\[
Ax = b \quad \Rightarrow \quad x = A^{-1}Ax = A^{-1}b
\]

with the unknown \( x \).

In this paper we first construct an inverse for the singular or ill-posed matrix \( A \). For an arbitrary \( b \) it might not have a solution, but assuming \( b \) has a solution, it can be constructed with a rectangular, truncated inverse matrix, dropping the non-pivoted columns. Afterwards, we discuss the construction of the null space, such that it can be tested whether the singular equation \( Ax = b \) has a solution and what
the subspace of solutions is, since any vector \( w \) in the right null space of \( A \) can be added to a solution \( x \) of the linear equation. Different choices for this vector \( w \) correspond to different criteria, such a least square fitting.

2 Complete pivoting

Complete pivoting, in Gauss elimination, is picking the largest element of the submatrix, to sweep that particular column of the submatrix, by subtracting the weighted row from all the other rows:

\[
(A | A^{-1}) = \begin{pmatrix}
\ast & \ast & \ast & \ast & 1 & 0 & 0 & 0 \\
\ast & \ast & \ast & \ast & 0 & 1 & 0 & 0 \\
\ast & \ast & \ast & \ast & 0 & 0 & 1 & 0 \\
\ast & \ast & \ast & \ast & 0 & 0 & 0 & 1 \\
\end{pmatrix}
\]

\[
\rightarrow (A_1 | A_1^{-1}) = \begin{pmatrix}
\ast & 0 & \ast & \ast & 1 & * & 0 & 0 \\
\ast & 1 & \ast & \ast & 0 & \square^{-1} & 0 & 0 \\
\ast & 0 & \ast & \ast & 0 & * & 1 & 0 \\
\ast & 0 & \ast & \ast & 0 & * & 0 & 1 \\
\end{pmatrix}
\]

where the box \( \square \) indicates the largest element, and the right hand side is the defines the set of operations, which will yield the inverse. Clearly, if row and column coordinate of the pivots are stored, the elimination can be done with a single array storage, where the column with the row number of the pivot is stored in the column with the column number of the pivot. Eventually, a permutation matrix remains, with a permutated inverse on the right hand side.

Complete pivoting gives a bound on the pivots. In the case of

\[
A = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 \\
-1 & 1 & 1 & 1 & 1 \\
0 & -1 & 1 & 1 & 1 \\
0 & 0 & -1 & 1 & 1 \\
0 & 0 & 0 & -1 & 1 \\
\end{pmatrix}
\]

the first pivot is one while all the others are two, which is also the upper bound on all the pivots: 1, 2, 2, \ldots, 2. The determinant follows immediately for such a matrix: \( \det A_{n \times n} = 2^{n-1} \).

In the case of a singular, or near-singular, matrix, the pivot has a value below the tolerance. Therefore, all other pivots will be below the threshold, which is two times the tolerance. A strict requirement, only fulfilled with complete pivoting. The corresponding small rows are set to zero. In permuted form the elimination matrix has the form:

\[
A_m = \begin{pmatrix}
1 & * & * & \cdots & 0 \\
\vdots & \ddots & \ast & \vdots & \vdots \\
0 & \cdots & 1 & \ast & \cdots & \ast \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & 0 & 0 & \cdots & 0 \\
\end{pmatrix}
\]

Back eliminating will bring it to the form:

\[
A_{2m} = \begin{pmatrix}
1 & \cdots & 0 & * & \cdots & * \\
\vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & 1 & * & \cdots & \ast \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & 0 & 0 & \cdots & 0 \\
\end{pmatrix}
\equiv \begin{pmatrix}
1 & \ast & 0 \\
\vdots & \vdots & \vdots \\
0 & 0 & 0 \\
\end{pmatrix}
\]

The first \( m \) rows span the domain. The last \( n - m \) rows are in the null space. The associated partial inverse has the form:

\[
A_{2m}^{-1} = \begin{pmatrix}
* & 0 \\
* & 1 \\
\end{pmatrix}
\]

which still spans the full space. If the singular matrix equation:

\[
Ax = b
\]

is known to have a solution, then the projection of \( b \) on the pivoted rows is sufficient to determine \( x \). Moreover, it is an accurate method to determine the solution, since the pivots were the largest coefficients in the domain space. Therefore, ignoring the \( (n - m) \times (n - m) \) unit matrix in the lower right corner will yield a rectangular, or singular inverse:

\[
A_{\text{truncated}}^{-1} b = \begin{pmatrix}
* & 0 \\
* & 1 \\
\end{pmatrix} b = x
\]

If the linear equation \( Ax = b \) is known to have a solution, the truncated \( A_{\text{truncated}}^{-1} \) is an appropriate
method to find a solution \( x \). Only the pivoted entries of \( b \) are required to uniquely determine a solution.

In the case the right inverse, instead of the left inverse is required, the appropriate truncation is not to the pivoted columns, but the pivoted rows of \( A^{-1} \):

\[
AA^{-1}_{\text{doubly truncated}} = A \begin{pmatrix} * & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}
\]

which yields a solution in the pivoted spaces for both \( b \) and \( x \). In effect, it turned the ill-posed problem into a well-posed problem in a straight projection on components of the total vectors, which has clear numerical and bookkeeping advantages above complicated rotations onto a subspace.

### 3 Null spaces

In the case it is not known if the singular matrix equation has a solution, the null space has to be determined, such that \( x \) and \( b \) can be projected on the space perpendicular to the right and left null space in the case of a generalized inverse. In the case \( b \) is perpendicular to the left null space, a solution exists. A vector in the right null space can be added to a solution \( x \), the compound vector is still a solution. The Moore-Penrose inverse picks the solution \( x \) which is perpendicular to the right null space. The right null space is the space perpendicular to the row vectors:

\[
\begin{pmatrix} 1 & \mathbf{R} \end{pmatrix} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_m \end{pmatrix}
\]

where

\[
\mathbf{v}_i = \underbrace{0, \ldots, 0, 1, 0, \ldots, 0}_{i-1}, \underbrace{r_{i1}, r_{i2}, \ldots, r_{i(n-m)}}_{n-m}
\]

Direct orthogonalization, through Gram-Schmidt, will generally not give a very accurate result. Furthermore, in general, one expects the null space to be much smaller than the total space, \((n - m) \ll m \approx n\), hence constructing the \((n - m)\) null space vectors by exclusion starting with \( n \) vectors, is an expensive task. From the expression of the \( v_i \)'s it is already clear that the null space, perpendicular to the \( v_i \)'s is spanned by: \((j \in \{1, \ldots, (n - m)\})

\[
\mathbf{w}_j = \begin{pmatrix} r_{1j}, r_{2j}, \ldots, r_{mj}, 0, \ldots, 0, -1, 0, \ldots, 0 \end{pmatrix}_{m \atop j-1 \atop n-m-j-1}
\]

These vectors are easily orthonormalized by constructing the covariance matrix \( C \) of the non-pivoted rows:

\[
C_{ij} = \delta_{ij} + \sum_{k=1}^{n-m} r_{ki} r_{kj}
\]

which is a positive definite symmetric matrix that can be diagonalized by appropriate means. The orthonormal basis of the right null space is given by:

\[
\text{RIGHT NULL} = \left\{ \frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^{n-m} u_j^i \mathbf{w}_j \right\}_{i=1}^{n-m}
\]

where \( \lambda_i \) and \( u_i \) are the eigenvalues and eigenvectors of \( C \), for \( i \in \{1, \ldots, (n - m)\} \).

In the case that \( A \) is a symmetric matrix the left null space equals the right null space. However, there is no guarantee that the row and column pivoting is equal for a symmetric matrix, although, in all practical cases it turned out that way.

For a general matrix \( A \), finding the right null space is a separate task. Given the partial inverted linear equation, after \( 2m \) complete pivoting steps, ignoring the row and column permutation matrices for a moment:

\[
A_{2m} x = \begin{pmatrix} 1 & \mathbf{R} \\ 0 & 0 \end{pmatrix} x = \begin{pmatrix} * & 0 \\ \mathbf{S} & 1 \end{pmatrix} b = A_{2m}^{-1} b
\]

The left null space correspond to the lower half of the equations; the lower \((n - m)\) rows, which are linear combinations of the other rows. In order for a solution to exists:

\[
0 = \begin{pmatrix} \mathbf{S} & 1 \end{pmatrix} b \equiv \begin{pmatrix} \mathbf{y}^1 \\ \vdots \\ \mathbf{y}^{n-m} \end{pmatrix} b
\]

Hence the rows \( y^j \) of \((\mathbf{S}|1)\) span the left null space of \( A \).
These rows can be converted to an orthonormal basis through a similar procedure as used for the right null space. The covariance matrix $C'$ is:

$$C_{ij}' = \delta_{ij} + \sum_{k=1}^{n-m} s_{ki}s_{kj}$$

and the rows $\{y^j\}_{j=1}^{n-m}$ are given by:

$$y^j = (s_{1j}, s_{2j}, \ldots, s_{mj}, 0, \ldots, 0, 1, 0, \ldots, 0)$$

Such that the null space is given by:

$$\text{LEFT NULL} = \left\{ \frac{1}{\sqrt{\lambda_i'}} \sum_{j=1}^{n-m} u_{ij}' y^j \right\}_{i=1}^{n-m}$$

where $\lambda_i'$ and $u_{ij}'$ are the eigenvalues and eigenvectors of $C'$. Both the covariance matrices $C$ and $C'$, used to construct the right and left null space orthonormal basis are proper positive definite matrices since they correspond to the unit matrix plus a positive matrix. Hence, recovering the orthonormal basis should not be too involved.

4 Conclusion

For the generalized inverse of an ill-posed, but full matrix, the small components were dropped. The use of complete pivoting is essential to separate all small components from the well-posed part. The generalized inverse does not necessarily have to be the projection on the left and right image space. Instead, a projection on large, i.e., pivoted, coefficients is used. This is called the representative space. The projection perpendicular to the left and right null spaces is a separate task, for which the null space, rather than the image space, is generally much more appropriate. The left and right null spaces, assumed to have low dimension compared to the rank of the matrix, are constructed through well-posed symmetric and positive definite eigenvalue problems of the dimension of the null space.

In the course of automating tasks for complex analysis, one will encounter exceptions of the type of ill-posed matrices. It does not necessarily mean the problem is ill-defined, but that the numerical implementation is not tuned to specific problems. The ability to handle and reduce such matrices will make automatic tools for complex analysis more robust and versatile.

Acknowledgement

This research was financially supported by the Dutch Technology Foundation STW (project grant TWI.6012).

References