

Operations Research: MATH 5810-001

1.2.2. Computing a Pseudoinverse using the Infinity Norm

Devanshu Agrawal

November 16, 2015

Executive Summary

A matrix is an array of numbers that describes a deformation of coordinate space. Matrices form an algebra, and hence it is possible to write equations involving sums and products of matrices. In order to solve such equations, it is often necessary to compute the multiplicative inverse of a matrix or to “invert” a matrix. The inverse of a matrix is a matrix whose product with the original matrix is the identity matrix. The identity matrix is simply the matrix whose product with any other matrix is just that matrix.

Unfortunately, it turns out that every matrix is not invertible. The next best thing, however, is a pseudoinverse, and every matrix has a pseudoinverse.

In this report, we address the problem of computing a pseudoinverse given any matrix (invertible or not). We do this by making the product of a matrix with its pseudoinverse as close as possible to the identity matrix. “Closeness” is measured by the infinity norm, which measures the maximum stretching that a matrix causes on coordinate space.

We show that the problem is a linear optimization problem. Our solution is an algorithm that purports to return the pseudoinverse of any given matrix. We implemented our algorithm in the modeling language GLPK and tested it by computing a pseudoinverse of an example matrix that is not invertible.

Our solution for the pseudoinverse of the example non-invertible matrix disagreed with the pseudoinverse obtained from other matrix algebra software. But we found that our algorithm does produce correct pseudoinverses for invertible matrices that we tested. We conclude that our algorithm produces solutions that minimize the infinity norm

as desired, but in the case of non-invertible matrices, our solutions do not always coincide with the pseudoinverses obtained from other methods.

1. Problem Description

Let A be any square matrix; A can be invertible or non-invertible. The problem is to compute a pseudoinverse X for A by minimizing the infinity norm $\|I - AX\|_\infty$ as a function of X . The general solution to this problem must then be applied to compute a pseudoinverse of the non-invertible matrix

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 \end{bmatrix}.$$

The solution to the problem will be an algorithm that computes the pseudoinverse of a matrix by minimizing an infinity norm as described in the problem statement. A pseudoinverse for the above example matrix is then obtained.

2. Background and Resources

Let $M_n(\mathbb{R})$ be the set of all $n \times n$ matrices with real entries. Let $I \in M_n(\mathbb{R})$ be the $n \times n$ identity matrix. Given a matrix $A \in M_n(\mathbb{R})$, let a_{ij} be the (i, j) th entry of A . The infinity norm of A is defined by

$$\|A\|_\infty = \max_{i=1}^n \sum_{j=1}^n |a_{ij}|.$$

We will solve the problem by setting up a linear program. A linear program is an optimization problem of the form

Minimize $z = c_1x_1 + \cdots + c_nx_n$ subject to

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + \cdots + a_{2n}x_n &\leq b_2 \\ &\vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n &\leq b_m. \end{aligned}$$

We will solve our linear program using the modeling language GLPK.

3. Description of General Solution

Let $A \in M_n(\mathbb{R})$ be a matrix with entries a_{ij} . Let $I \in M_n(\mathbb{R})$ be the identity matrix with entries $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise. Let $X \in M_n(\mathbb{R})$ be a variable matrix with entries x_{ij} . Define the additional variables ν and μ_{ij} , $i, j = 1, \dots, n$. We claim that an optimal solution X that minimizes $\|I - AX\|_\infty$ can be obtained as an optimal solution to the following linear program:

Minimize $z = \nu$ subject to

$$\delta_{ij} - \sum_{k=1}^n a_{ik}x_{kj} \leq \mu_{ij} \quad (i, j = 1, \dots, n) \quad (\text{Constraint I})$$

$$-\delta_{ij} + \sum_{k=1}^n a_{ik}x_{kj} \leq \mu_{ij} \quad (i, j = 1, \dots, n) \quad (\text{Constraint II})$$

$$\sum_{j=1}^n \mu_{ij} \leq \nu \quad (i = 1, \dots, n). \quad (\text{Constraint III})$$

To prove our claim, suppose X is an optimal solution to the above linear program. The matrix $U = I - AX$ has entries

$$u_{ij} = \delta_{ij} - \sum_{k=1}^n a_{ik}x_{kj},$$

where the sum over k performs matrix multiplication. By Constraints I and II, we have

$$|u_{ij}| \leq \mu_{ij}, \quad (i, j = 1, \dots, n).$$

Summing both sides over j , we have

$$\sum_{j=1}^n |u_{ij}| \leq \sum_{j=1}^n \mu_{ij}, \quad (i = 1, \dots, n).$$

By Constraint III, we have

$$\sum_{j=1}^n |u_{ij}| \leq \nu, \quad (i, j = 1, \dots, n).$$

Since this holds for all i , then

$$\max_{i=1}^n \sum_{j=1}^n |u_{ij}| \leq \nu.$$

Recalling the definition of infinity norm, we have $\|U\|_\infty = \|I - AX\|_\infty \leq \nu$. Since ν is minimized, then clearly $\|I - AX\|_\infty$ is minimized as well.

We implemented our linear program in GLPK (see A1. for the code).

4. Application to the Specific Context

Solving our linear program for the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 \end{bmatrix},$$

we obtained the following solution:

$$X = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

5. Interpretations and Conclusions

We found that our solution for the example matrix A given in the preceding section disagrees with the pseudoinverse returned by the “pinv” function in numpy (a package for the programming language python). We found, however, that our algorithm produces correct results for invertible matrices; if A is an invertible matrix, then the pseudoinverse of A returned by our algorithm agrees with the true inverse of A (this is true for at least a few examples that we tested). Therefore, we believe that our algorithm is to a large extent correct. While $X = 0$ for the example matrix A in the preceding section is an optimal solution, we suspect that the correct pseudoinverse of A is a different optimal solution. Our task for the future is therefore to see whether it is possible to move between optimal solutions until the true pseudoinverse for a matrix is obtained.

A1. Code for the General Solution

```
# Pseudoinverse

/* Model Section */

param n; # dimension of matrix A
set N := 1..n;
```

```

param A{i in N, j in N}; # matrix whose
    pseudoinverse is desired

param I{i in N, j in N} := # identity matrix
if i=j then
1
else
0;

var X{i in N, j in N}; # pseudoinverse of A
var mu{i in N, j in N};
var nu;

minimize Z: # minimize ||I-AX||_infty
nu;

s.t. C1{i in N, j in N}:
I[i,j] - sum{k in N} A[i,k]*X[k,j] <= mu[i,j];

s.t. C2{i in N, j in N}:
-I[i,j] + sum{k in N} A[i,k]*X[k,j] <= mu[i,j];

s.t. C3{i in N}:
sum{j in N} mu[i,j] <= nu;

solve;

for {i in N}{
printf{j in N}: "%5g", X[i,j];
printf "\n";
}

data;

param n := 4;

param A:
1 2 3 4 :=
1 1 2 3 4
2 2 3 4 5
3 3 4 5 6

```

```
4 4 5 6 7;
```

```
end;
```