hw1

Name: Ruffin White
ID: A53229660

January 25, 2018

Contents

HW1 : Math for Robotics

Author: Ruffin White
Course: CSE291
Date: Jan 25 2018

1.

> Implement the $PA = LDU$ decomposition algorithm by yourself (i.e. do not just call a built-in function in Matlab or Python. You may assume the matrix A is square and of full rank. Show that your implementation is functional.

See original jupyter notebook for code.
We'll use test this with the given A:

$$A = \begin{bmatrix} 4 & 7 & 0 \\ 2 & 2 & -6 \\ 3 & 2 & 1 \end{bmatrix}$$

Then run our function to compute P, L D, and U matrix for the following form:

$$A = PLDU$$

$$P = \begin{bmatrix} 1. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & 1. \end{bmatrix} \quad L = \begin{bmatrix} 1. & 0. & 0. \\ 0.5 & 1. & 0. \\ 0.75 & 2.16666667 & 1. \end{bmatrix} \quad D = \begin{bmatrix} 4. & 0. & 0. \\ 0. & -1.5 & 0. \\ 0. & 0. & 14. \end{bmatrix} \quad U = \begin{bmatrix} 1. & 1.75 & 0. \\ -0. & 1. & 4. \\ 0. & 0. & 1. \end{bmatrix}$$

If we then plug the found matixs back into the eq above, we see that we get back the orignal A:

$$A = \begin{bmatrix} 4. & 7. & 0. \\ 2. & 2. & -6. \\ 3. & 2. & 1. \end{bmatrix}$$

Comparison

Now let us compare our implentation to the linuar algrabra library in scipy.linalg.lu using a previous example.

The scipy implementation returns only P, L and U for the form:

$$A = PLU$$

$$P = \begin{bmatrix} 1. & 0. & 0. \\ 0. & 0. & 1. \\ 0. & 1. & 0. \end{bmatrix} \quad L = \begin{bmatrix} 1. & 0. & 0. \\ 0.75 & 1. & 0. \\ 0.5 & 0.46153846 & 1. \end{bmatrix} \quad U = \begin{bmatrix} 4. & 7. & 0. \\ 0. & -3.25 & 1. \\ 0. & 0. & -6.46153846 \end{bmatrix}$$

We can recompute U and D by extracting the diagle along U for D, then normalizing U along the diagonal.

$$A = PLDU$$

$$P = \begin{bmatrix} 1. & 0. & 0. \\ 0. & 0. & 1. \\ 0. & 1. & 0. \end{bmatrix} \quad L = \begin{bmatrix} 1. & 0. & 0. \\ 0.75 & 1. & 0. \\ 0.5 & 0.46153846 & 1. \end{bmatrix} \quad D = \begin{bmatrix} 4. & 0. & 0. \\ 0. & -3.25 & 0. \\ 0. & 0. & -6.46153846 \end{bmatrix} \quad U = \begin{bmatrix} 1. & 1.75 & 0. \\ -0. & 1. & -0.30769231 \\ -0. & -0. & 1. \end{bmatrix}$$

Agian, we get back the same A by re-multiplying the matrices as in the eq above.

$$A = \begin{bmatrix} 4. & 7. & 0. \\ 2. & 2. & -6. \\ 3. & 2. & 1. \end{bmatrix}$$

2.

Compute the $PA = LDU$ decomposition and the SVD decomposition for each of the following matrices: (you can use your own LDU implementation and it is OK to use a pre-defined implementation for SVD).

Again, runing our custom $PA = LDU$ decomposition function on the following matrices, along with SVD, then reconstruction A from the returned matrices to note behavior.

a.

$$A_1 = \begin{bmatrix} 4 & 7 & 0 \\ 2 & 2 & -6 \\ 3 & 2 & 1 \end{bmatrix}$$

$$P = \begin{bmatrix} 1. & 0. & 0. \\ 0. & 1. & 0. \\ 0. & 0. & 1. \end{bmatrix} \quad L = \begin{bmatrix} 1. & 0. & 0. \\ 0.5 & 1. & 0. \\ 0.75 & 2.16666667 & 1. \end{bmatrix} \quad D = \begin{bmatrix} 4. & 0. & 0. \\ 0. & -1.5 & 0. \\ 0. & 0. & 14. \end{bmatrix} \quad U = \begin{bmatrix} 1. & 1.75 & 0. \\ -0. & 1. & 4. \\ 0. & 0. & 1. \end{bmatrix}$$

$$A = \begin{bmatrix} 4. & 7. & 0. \\ 2. & 2. & -6. \\ 3. & 2. & 1. \end{bmatrix}$$

$$U = \begin{bmatrix} 0.83108771 & 0.36392168 & 0.42054041 \\ 0.45440389 & -0.88032036 & -0.13621004 \\ 0.3206405 & 0.30429769 & -0.89699085 \end{bmatrix} \quad S = \begin{bmatrix} 9.33057832 & 0. & 0. \\ 0. & 5.78987054 & 0. \\ 0. & 0. & 1.55489788 \end{bmatrix}$$

$$V = \begin{bmatrix} 0.55678008 & 0.7896298 & -0.25783856 \\ 0.10500046 & 0.24100823 & 0.96482638 \\ -0.82399688 & 0.56426927 & -0.05127709 \end{bmatrix}$$

$$A = \begin{bmatrix} 4.00000000e+00 & 7.00000000e+00 & 7.99719281e-16 \\ 2.00000000e+00 & 2.00000000e+00 & -6.00000000e+00 \\ 3.00000000e+00 & 2.00000000e+00 & 1.00000000e+00 \end{bmatrix}$$

Here we'll note that none of the values in U, S, V are terribly small or large, so we are not too close too a singularity. Additinaly, our reconstruction of A is quite accurate, with some small rounding error for value 0 in A.

b.

$$A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. \\ 0. & 0. & 0. & 1. & 0. \\ 0. & 0. & 0. & 0. & 1. \end{bmatrix} \quad L = \begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. \\ 0. & 1. & 0. & 1. & 0. \\ 1. & 0. & 0. & 0. & 1. \end{bmatrix} \quad D = \begin{bmatrix} 1. & 0. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. \\ 0. & 0. & 0. & -1. & 0. \\ 0. & 0. & 0. & 0. & -1. \end{bmatrix} \quad U = \begin{bmatrix} 1. & 0. & 0. & 0. & 1. \\ 0. & 1. & 0. & 1. & 0. \\ 0. & 0. & 1. & 0. & 0. \\ -0. & -0. & -0. & 1. & -0. \\ -0. & -0. & -0. & -0. & 1. \end{bmatrix}$$

$$A = \begin{bmatrix} 1. & 0. & 0. & 0. & 1. \\ 0. & 1. & 0. & 1. & 0. \\ 0. & 0. & 1. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. \\ 1. & 0. & 0. & 0. & 0. \end{bmatrix}$$

$$U = \begin{bmatrix} -0.85065081 & 0. & 0. & -0.52573111 & 0. \\ 0. & -0.85065081 & 0. & 0. & 0.52573111 \\ 0. & 0. & 1. & 0. & 0. \\ 0. & -0.52573111 & 0. & 0. & -0.85065081 \\ -0.52573111 & 0. & 0. & 0.85065081 & 0. \end{bmatrix}$$

$$V = \begin{bmatrix} 1.61803399 & 0. & 0. & 0. & 0. \\ 0. & 1.61803399 & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. \\ 0. & 0. & 0. & 0.61803399 & 0. \\ 0. & 0. & 0. & 0. & 0.61803399 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.85065081 & 0. & 0. & 0. & -0.52573111 \\ -0. & -0.85065081 & -0. & -0.52573111 & 0. \\ 0. & 0. & 1. & 0. & 0. \\ 0.52573111 & 0. & 0. & 0. & -0.85065081 \\ -0. & -0.52573111 & -0. & 0.85065081 & 0. \end{bmatrix}$$

$$A = \begin{bmatrix} 1.00000000e+00 & 0.00000000e+00 & 0.00000000e+00 & 0.00000000e+00 & 1.00000000e+00 \\ 0.00000000e+00 & 1.00000000e+00 & 0.00000000e+00 & 1.00000000e+00 & 0.00000000e+00 \\ 0.00000000e+00 & 0.00000000e+00 & 1.00000000e+00 & 0.00000000e+00 & 0.00000000e+00 \\ 0.00000000e+00 & 1.00000000e+00 & 0.00000000e+00 & -2.68028381e-17 & 0.00000000e+00 \\ 1.00000000e+00 & 0.00000000e+00 & 0.00000000e+00 & 0.00000000e+00 & 4.52894042e-17 \end{bmatrix}$$

Again, nothing to extream in U, S, or V. All values of S are well within the same magnitude.

Intrestingly, I think I may have a small error in my permutation matrix generation, as I need to not swap `np.absolute(pivot_search_space)` to get this PLDU to reconstruct corectly, so perhaps its an index error elseware this corner case touches.

```
abs_pivot_search_space = np.absolute(pivot_search_space)
```

c.

$$A_3 = \begin{bmatrix} 2 & 2 & 5 \\ 3 & 2 & 5 \\ 1 & 1 & 5 \end{bmatrix}$$

$$P = \begin{bmatrix} 0. & 1. & 0. \\ 1. & 0. & 0. \\ 0. & 0. & 1. \end{bmatrix} \quad L = \begin{bmatrix} 1. & 0. & 0. \\ 0.66666667 & 1. & 0. \\ 0.33333333 & 0.5 & 1. \end{bmatrix} \quad D = \begin{bmatrix} 3. & 0. & 0. \\ 0. & 0.66666667 & 0. \\ 0. & 0. & 2.5 \end{bmatrix} \quad U = \begin{bmatrix} 1. & 0.66666667 & 1.66666667 \\ 0. & 1. & 2.5 \\ 0. & 0. & 1. \end{bmatrix}$$

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

$$U = \begin{bmatrix} -0.58592436 & -0.04442838 & -0.80914693 \\ -0.62305157 & -0.61376658 & 0.48486836 \\ -0.51816926 & 0.78823645 & 0.33193962 \end{bmatrix} \quad S = \begin{bmatrix} 9.79103061 & 0. & 0. \\ 0. & 1.4162264 & 0. \\ 0. & 0. & 0.36058604 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.36351359 & -0.29987866 & -0.88200378 \\ -0.8063118 & -0.37293011 & 0.45911264 \\ 0.46660385 & -0.87806373 & 0.10623054 \end{bmatrix}$$

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

The reconstruction of A is spot on, and again nothing too small in S.

3.

Solve the following system of equations $Ax = b$ given the below values for A and b. For each system specify if it has zero, one or more solutions. For the systems with zero solutions give the SVD solution. Relate your answers to the SVD decomposition.

a.

$$A = \begin{bmatrix} 2 & 1 & 3 \\ 2 & 1 & 2 \\ 5 & 5 & 5 \end{bmatrix} \quad b = \begin{bmatrix} 5 \\ -5 \\ 0 \end{bmatrix}$$

By solving for the system of equations, we arive at a well defined solution:

$$x = \begin{bmatrix} -15. \\ 5. \\ 10. \end{bmatrix}$$

$$Det(A) = 5.0 \quad A^{-1} = \begin{bmatrix} -1.00000000e+00 & 2.00000000e+00 & -2.00000000e-01 \\ -1.11022302e-16 & -1.00000000e+00 & 4.00000000e-01 \\ 1.00000000e+00 & -1.00000000e+00 & 0.00000000e+00 \end{bmatrix}$$

Here we see that the determinant of A is well behaved, meaing that A is not very singular.

$$U = \begin{bmatrix} -0.36351359 & 0.8063118 & 0.46660385 \\ -0.29987866 & 0.37293011 & -0.87806373 \\ -0.88200378 & -0.45911264 & 0.10623054 \end{bmatrix} \quad S = \begin{bmatrix} 9.79103061 & 0. & 0. \\ 0. & 1.4162264 & 0. \\ 0. & 0. & 0.36058604 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.58592436 & -0.51816926 & -0.62305157 \\ 0.04442838 & -0.78823645 & 0.61376658 \\ -0.80914693 & 0.33193962 & 0.48486836 \end{bmatrix}$$

$$x = \begin{bmatrix} -12.39094986 \\ 10.23654626 \\ 9.57483588 \end{bmatrix}$$

With the SVD solution close to the original, it is more likely that only one solution exist.

b.

$$A = \begin{bmatrix} 4 & 7 & 0 \\ 2 & 2 & -6 \\ 1 & 2 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}$$

By solving for the system of equations, we arive at a ill defined solution:

$$x = \begin{bmatrix} 1.05083991e+17 \\ -6.00479950e+16 \\ 1.50119988e+16 \end{bmatrix}$$

$$Det(A) = -3.33066907388e-16 \quad A^{-1} = \begin{bmatrix} -4.20335965e+16 & 2.10167983e+16 & 1.26100790e+17 \\ 2.40191980e+16 & -1.20095990e+16 & -7.20575940e+16 \\ -6.00479950e+15 & 3.00239975e+15 & 1.80143985e+16 \end{bmatrix}$$

However, we see that the determinant of A is quite close to 0, meaing that A is almost singular.

$$U = \begin{bmatrix} -0.84780042 & 0.42857143 & -0.31234752 \\ -0.49083182 & -0.85714286 & 0.15617376 \\ -0.20079484 & 0.28571429 & 0.93704257 \end{bmatrix} \quad S = \begin{bmatrix} 9.05538514e+00 & 0.00000000e+00 & 0.00000000e+00 \\ 0.00000000e+00 & 5.74456265e+00 & 0.00000000e+00 \\ 0.00000000e+00 & 0.00000000e+00 & 1.05319976e-16 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.50507627 & -0.80812204 & 0.30304576 \\ 0.04973647 & 0.32328708 & 0.94499299 \\ 0.86164044 & -0.49236596 & 0.12309149 \end{bmatrix}$$

$$x = \begin{bmatrix} 2.24685756e+15 \\ 7.00641587e+15 \\ 9.12631296e+14 \end{bmatrix}$$

Here because A is close to singular, and that solution from SVD is quite far from the one found, it is more likly that more solution are possible.

c.

$$A = \begin{bmatrix} 4 & 7 & 0 \\ 2 & 2 & -6 \\ 1 & 2 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 18 \\ -12 \\ 8 \end{bmatrix}$$

By solving for the system of equations, we arive at a well defined solution:

$$x = \begin{bmatrix} 4.5 \\ 0. \\ 3.5 \end{bmatrix}$$

$$Det(A) = -3.33066907388e-16 \quad A^{-1} = \begin{bmatrix} -4.20335965e+16 & 2.10167983e+16 & 1.26100790e+17 \\ 2.40191980e+16 & -1.20095990e+16 & -7.20575940e+16 \\ -6.00479950e+15 & 3.00239975e+15 & 1.80143985e+16 \end{bmatrix}$$

However, we see that the determinant of A is quite close to 0, meaing that A is almost singular.

$$U = \begin{bmatrix} -0.84780042 & 0.42857143 & -0.31234752 \\ -0.49083182 & -0.85714286 & 0.15617376 \\ -0.20079484 & 0.28571429 & 0.93704257 \end{bmatrix} \quad S = \begin{bmatrix} 9.05538514e+00 & 0.00000000e+00 & 0.00000000e+00 \\ 0.00000000e+00 & 5.74456265e+00 & 0.00000000e+00 \\ 0.00000000e+00 & 0.00000000e+00 & 1.05319976e-16 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.50507627 & -0.80812204 & 0.30304576 \\ 0.04973647 & 0.32328708 & 0.94499299 \\ 0.86164044 & -0.49236596 & 0.12309149 \end{bmatrix}$$

$$x = \begin{bmatrix} -2.24146804 \\ 1.08133057 \\ -2.78315274 \end{bmatrix}$$

4.

A frequent problem in perception is that you see an object from multiple positions. Consider that you have recorded two datasets $\{a_i\}$ and $\{b_i\}$ with $i = 1 \dots n$ and we assume we have established correspondance between the data points. How could you use SVD or similar techniques to compute the relatively transformation between the datasets a and b assuming that the datasets have many data points and you are expected to use all the data-points.

If we take a look at the homogeneous coordinate transformation that translats and rotates the point from one cordenate frame to the other:

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1a} \\ y_{1a} \\ z_{1a} \\ 1 \end{bmatrix} = \begin{bmatrix} x_{1b} \\ y_{1b} \\ z_{1b} \\ 1 \end{bmatrix}, \quad \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{2a} \\ y_{2a} \\ z_{2a} \\ 1 \end{bmatrix} = \begin{bmatrix} x_{2b} \\ y_{2b} \\ z_{2b} \\ 1 \end{bmatrix}$$

We'll see that we can append such transformation together into one large operation.

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1a} & \dots & x_{na} \\ y_{1a} & \dots & y_{na} \\ z_{1a} & \dots & z_{na} \\ 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} x_{1b} & \dots & x_{nb} \\ y_{1b} & \dots & y_{nb} \\ z_{1b} & \dots & z_{nb} \\ 1 & \dots & 1 \end{bmatrix}$$

By transposing the equalities above, we can more clearly see the set of linear set of equations we can use to solve for the R and T components of the unknown transformation:

$$\begin{bmatrix} x_{1a} & y_{1a} & z_{1a} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{na} & y_{na} & z_{na} & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^T = \begin{bmatrix} x_{1b} & y_{1b} & z_{1b} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_{nb} & y_{nb} & z_{nb} & 1 \end{bmatrix}$$

It should be noted that a minimum of 8 associated points forming 8 equations are necessary to solve for the 8 unknown in the rotation and translation of the 3D transformation.

Given the points may be noise, i.e. mensuments made from one references frame may be subject to error, it is perhaps best to Random sample consensus (RANSAC) as an iterative method to estimate the coefficients within the homogeneous transformation matrix. In this way we can ensure that observed data containing outliers from mesurment noise are to be accorded no influence on eventual estimate.