Week 14 Discussion Solutions

CAS CS 132: Geometric Algorithms

December 4, 2023

During discussion sections, we will go over three problems.

- The first will be a warm-up question, to help you verify your understanding of the material.
- The second will be a solution to a problem on the assignment of the previous week.
- The third will be a problem similar to one on the assignment of the following week.

The remainder of the time will be dedicated to open Q&A.

1 Basis of the column space (Warm up)

Consider the following matrices. Note that A' is an echelon from of A.

$$A = \begin{bmatrix} 1 & 1 & 2 & 0 & 2 \\ 3 & 4 & 9 & -2 & 5 \\ -2 & -3 & -7 & 2 & -2 \\ 2 & 2 & 4 & 0 & 5 \end{bmatrix} \qquad A' = \begin{bmatrix} 1 & 0 & -1 & 2 & 0 \\ 0 & 1 & 3 & -2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

A. Use the echelon form above to find a basis of $\text{Col}\, A$ made up of columns of A.

B. Write down a NumPy expression in terms of `A` (a 2D NumPy array representing the matrix A above) for the matrix whose columns are the basis vectors you found in the previous part.

C. Let A_i be the matrix whose columns are the first i columns of A. For example,

$$A_3 = \begin{bmatrix} 1 & 1 & 2 \\ 3 & 4 & 9 \\ -2 & -3 & -7 \\ 2 & 2 & 4 \end{bmatrix}$$

Find $\text{rank}(A_i)$ for each i using the echelon form above.

D. Write down a NumPy expression for $\text{rank}(A_i)$ in terms of `A` and `i` and the NumPy function `numpy.linalg.matrix_rank`, which returns the rank of its argument.

E. Let B be an arbitrary $m \times 5$ matrix and let B_i be the matrix whose columns are the first i columns of B. Further suppose that $\text{rank}(B_1) = 1$, $\text{rank}(B_2) = 1$, $\text{rank}(B_3) = 2$, $\text{rank}(B_4) = 3$, and $\text{rank}(B_5) = 3$. Which columns of B form a basis of $\text{Col}\, B$?

F. Use the previous parts to describe in an informal procedure you can use to find a basis for the column space of a small matrix using Python.

Solution.

A. The columns of A which form a basis of $\text{Col}\, A$ the pivot columns of A. Therefore,

$$\left\{ \begin{bmatrix} 1 \\ 3 \\ -2 \\ 2 \end{bmatrix}, \begin{bmatrix} 1 \\ 4 \\ -3 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 5 \\ -2 \\ 5 \end{bmatrix} \right\}$$

is a basis for $\text{Col}\, A$.

B. It is possible to index NumPy arrays with lists, which can be used to pick out rows or columns of a matrix:

```
A[:,[0, 1, 4]]
```

C. The linear dependence relations between columns are preserved by row reductions. For example, since

$$-1\begin{bmatrix}1\\0\\0\\0\end{bmatrix}+3\begin{bmatrix}0\\1\\0\\0\end{bmatrix}=\begin{bmatrix}-1\\3\\0\\0\end{bmatrix}$$

we also have that

$$-1\begin{bmatrix}1\\3\\-2\\2\end{bmatrix}+3\begin{bmatrix}1\\4\\-3\\2\end{bmatrix}=\begin{bmatrix}2\\9\\-7\\4\end{bmatrix}$$

We can use this to determine the ranks of each matrix: $\text{rank}(A_1) = 1$, $\text{rank}(A_2) = 2$, $\text{rank}(A_3) = 2$, $\text{rank}(A_4) = 2$, and $\text{rank}(A_5) = 3$.

D. We can use array slices to achieve this:

```
numpy.linalg.matrix_rank(A[:,:i])
```

E. If $\text{rank}(B_i) = \text{rank}(B_{i+1})$ then the $(i + 1)$th column of B lies in the span of the first i columns. Therefore, we can construct a basis for $\operatorname{Col} B$ by choosing the columns for which the rank increases. In this case, the first column, the third column and the fourth column.

F. If we want to find a basis for the column space of a small matrix, we can find the ranks of increasingly large subsets of columns and see which ones increase the rank.

2 Boundary Reflection without a Matrix

Suppose that A is a $n \times n$ matrix and $\mathbf{z}$ is a vector in $\mathbb{R}^n$ whose ith component is 1 if the ith column of A is $\mathbf{0}$, and 0 otherwise, e.g.,

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -3 & 0 \\ 0 & 2 & 0 \end{bmatrix} \qquad \mathbf{z} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

For an arbitrary vector $\mathbf{v}$ in $\mathbb{R}^n$, write down an expression in terms of A, $\mathbf{z}$ and $\mathbf{v}$ for the vector

$$A'\mathbf{v}$$

where A' is the same as A, but every all-zeros column of A is replaced with the vector $c\mathbf{1}$ for some scalar c, e.g., as it pertains to the example above,

$$A' = \begin{bmatrix} c & 1 & c \\ c & -3 & c \\ c & 2 & c \end{bmatrix}$$

Furthermore, write down a NumPy expression which computes this *without* using the function `numpy.ones`.

Solution. The difference between $A\mathbf{v}$ and $A'\mathbf{v}$ in the example above is the vector

$$c \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = c \begin{bmatrix} v_1 + v_3 \\ v_1 + v_3 \\ v_1 + v_3 \end{bmatrix}$$

The important point of recognition here is that $v_1 + v_3 = \mathbf{z} \cdot \mathbf{v}$. Therefore,

$$A'\mathbf{v} = A\mathbf{v} + c(\mathbf{z} + \cdot\mathbf{v})\mathbf{1}$$

If we translate this directly into a NumPy expression we get

```
A @ v + c * (z @ v) * numpy.ones(A.shape[0])
```

but we don't actually need the call to `numpy.ones` since adding a number to a 1D NumPy vector is the same as adding that number to each entry of the vector. So we can equivalently write this as

```
A @ v + c * (z @ v)
```

3 Multiple Least Squares Solutions

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

A. Find the orthogonal projection $\hat{\mathbf{b}}$ onto $\operatorname{Col} A$. (*Hint.* Note that the columns of A are linearly dependent. It will be easier to do the computation if you take the last three columns of A to find the projection.)

B. Find a general form solution for the homogeneous equation $A^T A\mathbf{x} = \mathbf{0}$. Then write this general form solution as a linear combination of vectors with free variables as weights.

C. Find the normal equations for the system $A\mathbf{x} = \mathbf{b}$.

D. Using the normal equations find a general form solution for the set of least squares solutions of $A\mathbf{x} = \mathbf{b}$. Then write this general form solution as a linear combination of vectors with free variables **and the scalar 1** as weights.

Solution. **Note for TFs/TAs:** Feel free to give the reduced forms of the matrices below as you solve each step if you don't want to commit discussion time to having students solve them.

A. $\operatorname{Col} A$ is spanned by the last three columns of A, so we can use this matrix, call it C, to build a projection onto $\operatorname{Col} A$:

$$\hat{\mathbf{b}} = C(C^T C)^{-1} C^T \mathbf{b}$$

Then

$$C^T \mathbf{b} = \begin{bmatrix} -3+1 \\ 0+2 \\ 5+1 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \\ 6 \end{bmatrix}$$

and $C^T C = 2I$ so $(C^T C)^{-1} = (0.5)I$ and

$$C(0.5C^T\mathbf{b}) = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \\ 3 \\ 3 \end{bmatrix}$$

B. The reduced echelon from of the augmented matrix for the system $A^T A\mathbf{x} = \mathbf{0}$ is

$$\begin{bmatrix} 6 & 2 & 2 & 2 & 0 \\ 2 & 2 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

which has the general form solution

$$\begin{aligned} x_1 &= -x_4 \\ x_2 &= x_4 \\ x_3 &= x_4 \\ x_4 &\text{ is free} \end{aligned}$$

which can be written as

$$x_4 \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

C. The normal equations are given by

$$\begin{bmatrix} 6 & 2 & 2 & 2 \\ 2 & 2 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 2 & 0 & 0 & 2 \end{bmatrix} \mathbf{x} = \begin{bmatrix} -4 \\ -4 \\ 2 \\ 6 \end{bmatrix}$$

D. The reduced echelon form of the augmented matrix of the normal equations is

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 3 \\ 0 & 1 & 0 & -1 & 5 \\ 0 & 0 & 1 & -1 & -2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

which has the general form

$$\begin{aligned} x_1 &= 3 - x_4 \\ x_2 &= 5 + x_4 \\ x_3 &= -2 + x_4 \\ x_4 &\text{ is free} \end{aligned}$$

We can then write this as

$$1 \begin{bmatrix} 3 \\ 5 \\ -2 \\ 0 \end{bmatrix} + x_4 \begin{bmatrix} -1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Note the similarity with the linear combination for the null space. We may remember from a previous assignment that **solutions sets are affine spaces which can be represented as the null space translated by some vector.** In other words, every solution to a system of linear equations is some fixed vector plus a vector in the null space. Think about how this would be useful for finding *maximal* sets of linearly independent vectors.