Week 9 Discussion

CAS CS 132: Geometric Algorithms

October 30, 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 LU Factorization by Hand (Warm Up)

Compute the LU factorization of the following matrix. Then verify that your factorization is correct by carrying out the matrix multiplication LU.

$$\begin{bmatrix} 1 & 2 & -3 \\ -1 & -1 & 4 \\ 2 & 3 & -6 \end{bmatrix}$$

Solution.

2 Matrix Algebra

A. Suppose A and B are invertible matrices and $AB^TXA^{-1}B = I$. Solve for X in terms of A and B.

B. Show that $A + A^T$ is symmetric for any square matrix A. That is, show that $(A + A^T)^T = A + A^T$.

C. Show that if A and B are symmetric and $AB = BA$ then AB is symmetric.

Solution.

3 NumPy and SciPy

A. Construct a random 10×10 matrix and a random vector in $\mathbb{R}^{10}$ using

```python
import numpy
import scipy

a = numpy.random.rand(10, 10)
b = numpy.random.rand(10)
```

Then construct its inverse and its LU factoration using `numpy.linalg.inv` and `scipy.linalg.lu_factor`, respectively. Finally, solve equation `ax = b` in three ways:

```python
s1 = numpy.linalg.solve(a, b)
s2 = a_inv @ b
s3 = scipy.lu_solve(lu) # where lu is the result of lu_factor
```

Verify that these three solutions are the same. Note, this process is not guaranteed to succeed since not all square matrices are invertible, but it is incredibly unlikely that a random 10×10 matrix is not invertible.

B. Construct a random integer matrix with entries between 1 and 10 as follows:

```python
a = numpy.random.randint(1, 11, (10, 10))
```

Write a function called `norm_col` which divides a NumPy vector by the sum of its entries (you don't need to consider the case in which the input is the zero vector). Then read about the function `numpy.apply_along_axis` in the NumPy documentation. Check the value of

```python
numpy.apply_along_axis(norm_col, 0, a)
```