# ECON526: Problem Set 2 (Solutions)

Jesse Perla

## — INSTRUCTOR SOLUTIONS —

### Instructions

- Ensure you modify the field above with your **name and student number above immediately**
- Modify directly and do not rename the file as Canvas will automatically append your name to it.
- Submit both the `.ipynb` and an exported `html` version of this file.
- See here for instructions.
- Do not change the filenames. Canvas automatically appends your name/student number to submitted files.

### Setup

Use the following packages and imports

```
import numpy as np
import matplotlib.pyplot as plt
import scipy
from numpy.linalg import cond, matrix_rank, norm
from scipy.linalg import inv, solve, det, eig, lu, eigvals
from scipy.linalg import solve_triangular, eigvalsh, cholesky
```

## Question 1

### Question 1.1

Generate a random matrix $A \in \mathbb{R}^{10\times 10}$ and random vector $b \in \mathbb{R}^{10}$ of uniformly distributed floating points between 0 and 1. Hint: use `np.random.rand(10, 10)`

and solve $Ax = b$ as a linear system with

1. `scipy.linalg.solve`
2. `scipy.linalg.inv`

Modify

```
N = 10
# modify here
```

What matrix decomposition would `scipy.linalg.solve` likely use in this case?

SOLUTIONS

```
# SOLUTIONS
N = 10
A = np.random.rand(N, N)
b = np.random.rand(N)
x = solve(A, b)
x_inv = inv(A) @ b
print(f"norm of difference is {norm(x - x_inv)}")
```

```
norm of difference is 2.238801616996866e-15
```

SOLUTIONS

An LU decomposition (with pivots) because the matrix is square but no other structure is given

### Question 1.2

The product of any matrix with its transpose is symmetric. Prove it. Hint: the definition of symmetric matrix $B$ is if $B = B^T$. Use this to take the transpose of $B \equiv A^T A$.

SOLUTIONS

$$B^T = (A^T A)^T = (A)^T (A^T)^T = A^T A = B$$

### Question 1.3

Using your matrix $A$ from before, construct the symmetric $B = A^T A$.

Verify it is symmetric, and then find out if it is positive definite. Hint: you will need to use `eigvals` or `eigs`.

```
# modify here, using A from above
```

SOLUTIONS

```
# SOLUTIONS
B = A.T @ A
Lambda = eigvals(B)
print(f"||B - B^T|| = {norm(B - B.T)}")
print(f"eigenvalues of B are {Lambda}")
print(f"min eigenvalue is {np.real(Lambda.min())}")
# print(f"max abs eigenvalue (spectral radius) is {np.abs(Lambda).max()}")
```

```
||B - B^T|| = 0.0
eigenvalues of B are [2.38717923e+01+0.j 2.21333899e+00+0.j 1.24332424e+00+0.j
 9.38326421e-01+0.j 8.24352685e-01+0.j 6.10684667e-01+0.j
 3.91924339e-01+0.j 1.64192459e-03+0.j 3.02795354e-02+0.j
 1.58407983e-01+0.j]
min eigenvalue is 0.0016419245860238934
```

### Question 1.4

Now solve the system $Bx = b$ using `solve` and `inv`. If the matrix was shown to be symmetric or positive definite before, then use that in your solution

```
# modify here
```

SOLUTIONS

```
# SOLUTIONS
x = solve(B, b, assume_a="pos")
x_inv = inv(B) @ b
print(f"norm of difference is {norm(x - x_inv)}")
```

```
norm of difference is 1.3545175902220895e-14
```

## Question 2

### Question 2.1

Take the matrix $A \in \mathbb{R}^{100 \times 5}$

Check if it is full rank

```
# modify here
N = 100
K = 5
A = np.random.rand(N, K)
```

SOLUTIONS

```
# SOLUTIONS
matrix_rank(A) == K
```

```
np.True_
```

### Question 2.2

Take that previous matrix in Q2.1 and append a new column to it, so that it is now $\hat{A} \in \mathbb{R}^{100 \times 6}$ such that the matrix will still have a rank of 5 and not 6. Hint: lots of ways to append a vector to a matrix in numpy, including `numpy.column_stack` and `numpy.concatenate`

```
# modify here
```

SOLUTIONS

```
# SOLUTIONS
# One approach: just make a copy of the first column, which is exactly
# linearly dependent on that same column
A_col1 = A[:, 0]
A_hat = np.column_stack((A, A_col1))
print(f"shape of A_hat is {A_hat.shape}, rank is {matrix_rank(A_hat)}")
```

```
shape of A_hat is (100, 6), rank is 5
```

### Question 2.3

Take the $A$ and the $\hat{A}$ from before, and form $B = AA^T$ and $\hat{B} = \hat{A}\hat{A}^T$. What are the ranks?

```
# modify here
```

Could we do a cholesky decomposition of this matrix? Check and/or explain why not if you can't

SOLUTIONS

```
# SOLUTIONS
B = A @ A.T
B_hat = A_hat @ A_hat.T
print(f"size of B is {B.shape}")
print(f"rank of B is {matrix_rank(B)}")
print(f"rank of B_hat is {matrix_rank(B_hat)}")
```

```
size of B is (100, 100)
rank of B is 5
rank of B_hat is 5
```

SOLUTIONS

- `cholesky(B)` would fail
- The matrix isn't positive definite.
- We don't even need to check because we know it isn't full rank
- hence has zero eigenvalues

## Question 3

Prove that if you have an eigendecomposition $A = Q\Lambda Q^{-1}$, then $A \cdot A = A^2 = Q\Lambda^2 Q^{-1}$ where $\Lambda^2$ is the componentwise square of the eigenvalues. Hint: $Q^{-1}Q = I$.

SOLUTIONS

Given $A = Q\Lambda Q^{-1}$ and that $Q^{-1}Q = I$

$$AA = Q\Lambda Q^{-1}Q\Lambda Q^{-1} = Q\Lambda\Lambda Q^{-1} = Q\Lambda^2 Q^{-1}$$

and $\Lambda\Lambda$ is diagonal with the squared eigenvalues since $\Lambda$ is diagonal.

## Question 4

### Question 4.1

Take the vector $\hat{x}_1 \in \mathbb{R}^2$

```
x_hat_1 = np.array([1, 2])
```

Verify that it is not a unit length vector (i.e. $\|\hat{x}_1\| \neq 1$) then create a new $x_1$ that is a unit length vector in the same direction as $\hat{x}_1$ (i.e. $||x_1|| = 1$)

```
# modify here
# x_1 = ...
```

SOLUTIONS

```
# SOLUTIONS
print(f"norm of x_hat_1 is {norm(x_hat_1)} (not 1)")
x_1 = x_hat_1 / norm(x_hat_1)
print(f"norm of x_1 is {norm(x_1)}")
```

```
norm of x_hat_1 is 2.23606797749979 (not 1)
norm of x_1 is 0.9999999999999999
```

### Question 4.2

Now find a $x_2$ which is also a unit length vector, but is orthogonal to $x_1$. Check it with `np.dot(x_1, x_2)` approx 0 and `norm(x_2)` approx 1. Hint: many ways to do this by hand in $\mathbb{R}^2$ and fulfill the requirements, such as simple rotations.

```
# modify here
# x_2 = ...
```

SOLUTIONS

```
# SOLUTIONS
x_2 = np.array([-x_1[1], x_1[0]]) # rotate by 90 degrees
print(f"norm of x_2 is {norm(x_2)} and dot product is {np.dot(x_1, x_2)}")
```

```
norm of x_2 is 0.9999999999999999 and dot product is 0.0
```

### Question 4.3

The vectors $x_1$ and $x_2$ are now an orthonormal set. Form the matrix $Q = \begin{bmatrix} x_1 & | & x_2 \end{bmatrix}$ and verify the condition for orthonormality (i.e. $Q^T Q = I \implies Q^{-1} = Q^T$)

```
# Q = np.column_stack((x_1, x_2))
# modify here
```

SOLUTIONS

```
# SOLUTIONS
Q = np.column_stack((x_1, x_2))
print(f"Q^T Q = \n{Q.T@Q}")
print(f"inv(Q) - Q^T = \n{inv(Q) - Q.T}")
```

```
Q^T Q =
[[1.00000000e+00 1.25949234e-17]
 [1.25949234e-17 1.00000000e+00]]
inv(Q) - Q^T =
[[0.00000000e+00 1.11022302e-16]
 [0.00000000e+00 0.00000000e+00]]
```

### Question 4.4

Create a matrix $A$ such that: 1. $Q$ from the previous question are its eigenvectors 2. The spectral radius of $A$ is 1.0 3. $A$ is positive definite.

Hint: create a matrix of eigenvalues $\Lambda$ and then do an eigendecomposition in reverse

```
# modify here
```

SOLUTIONS

```
# SOLUTIONS
Lambda = np.array([1.0, 0.1]) # largest is 1, all in (0, 1]
A = Q @ np.diag(Lambda) @ Q.T # Q orthogonal, so inv(Q) = Q.T
print(A)
print(f"eigenvalues of A are {eigvalsh(A)}") # can use since symmetric
```

```
[[0.28 0.36]
 [0.36 0.82]]
eigenvalues of A are [0.1 1. ]
```