AI 300 Prerequisite Test *

Beaver-Edge AI Institute

If you are not sure whether you are ready to take AI 300, please complete this prerequisites test. There is no time limit on solving problems below.

If you can solve 70% of the problems, you are ready to take AI 300.

1. Define

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_0 & \cdots \mathbf{a}_{N-1} \end{bmatrix} \in \mathbb{R}^{M \times N}.$$

   For any vector $\mathbf{x} \in \mathbb{R}^{M \times 1}$, compute its projection to the subspace spanned by all column vectors in $\mathbf{A}$.

2. Compute the trace and the determinant of the following matrix:

$$\begin{bmatrix} 2 & -1 & 3 & 5 \\ 0 & 2 & 4 & 1 \\ -3 & 1 & 0 & 2 \\ 4 & -3 & 1 & 3 \end{bmatrix}.$$

3. Do the eigendecomposition of the following matrix

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

4. Do the spectral decomposition of the following matrix

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

5. Let $\mathbf{A} \in \mathbb{N}^{M \times N}$. Prove that $\mathbf{A}\mathbf{A}^\top$ is positive semi-definite.

6. For any matrix $\mathbf{A} \in \mathbb{N}^{M \times N}$, use the singular value decomposition to find four subspaces.

7. Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N \times N}$ be two real-valued square matrices with smallest eigenvalues $\underline{\lambda}_A$ and $\underline{\lambda}_B$, respectively. Define $\mathbf{C} = \mathbf{A} + \mathbf{B}$. Prove that the smallest eigenvalue of $\mathbf{C}$ is lower bounded by $\underline{\lambda}_A + \underline{\lambda}_B$.

8. Prove that for any matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$,

$$\operatorname{rank}\left(\mathbf{A}\mathbf{A}^\top\right) = \operatorname{rank}\left(\mathbf{A}\right).$$

9. Let $\mathbf{A} \in \mathbb{R}^{M \times N}$ be with rank r and the following singular value decomposition (SVD) $\mathbf{A} = \sum_{i=0}^{r-1} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$.

   Solve the equation

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

   where $\mathbf{b} \in \mathbb{R}^{M \times 1}$. Express your solution with the SVD of $\mathbf{A}$.

10. This is a coding task.

    Write a function satisfying

    (a) The input is a real-valued square matrix that is guaranteed to have all real and distinct eigenvalues.

    (b) The output consists of three items: the dominant eigenvalue (its absolute value is the largest among all eigenvalues), the associated right eigenvector, and the associated left eigenvector.

11. This is a coding task.

    Write a function satisfying

    (a) The input has two arguments. One is a 3-d NumPy array that stores a colorful image data. One is a positive integer k.

    (b) The output is a 3-d NumPy array that stores a colorful image obtained from the following transformation of the input data:

      i. Do singular value decomposition on each color channel of the input image.

      ii. Keep top-k components in each channel.

    (c) While calling this function, display the output image.

12. Use NumPy to write a function that does the following tasks:

    (a) Input: an arbitrary number of positional arguments `*args`.

    (b) The body of the function:

      i. Create a NumPy array object that has shape `*args` and data type `uint8`.

      ii. Move axis 1 to the last position.

      iii. Flatten the array.

      iv. Normalize data within -1 and 1. Print and output it.

      To test your function, try the following inputs: `[1, 3, 4]`, `[3, 5, 2, 4]`.

13.

(a) Write a function that uses NumPy to generate the following lattice points:

$$\left\{(x_0, x_1, \cdots, x_{K-1}) \in \{0, 0.01, 0.02, \cdots, 1\}^K\right\},$$

where the dimension K is the input.
The output is a NumPy array with shape `(101**K, K)`.

(b) Set $K = 2$. Define

$$f(x_0, x_1) = \begin{cases} 3 & \text{if } \mathbf{1}\{x_0 + x_1 \leq 1\} \\ 1 & \text{elsewhere} \end{cases}.$$

Write code to generate this output. No loop is allowed.

(c) For those points whose functional values are 3, plot them with the red color. For those points whose functional values are 1, plot them with the blue color.

14. Let

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^{(0),\top} \\ \mathbf{x}^{(1),\top} \\ \vdots \\ \mathbf{x}^{(N-1),\top} \end{bmatrix} \in \mathbb{R}^{N\times d}.$$

Write one line of code with NumPy to compute the following empirical covariance matrix

$$\frac{1}{N-1}\sum_{n=0}^{N-1} \hat{\mathbf{x}}^{(n)}\hat{\mathbf{x}}^{(n),\top},$$

where

$$\hat{\mathbf{x}}^{(n)} = \mathbf{x}^{(n)} - \frac{1}{N}\sum_{m=0}^{N-1}\mathbf{x}^{(m)}.$$

Loop is not allowed.

15.

(a) Write one line of code with NumPy to generate an array with shape `(N,N)`, rank at most r, and each entry that is uniformly distributed between 0 and 1.

(b) Use NumPy to do singular value decomposition of your generated 2-dim array.

16. You are given inputs μ with shape `(d,)` and $\mathbf{\Sigma}$ with shape `(d,d)` and is guaranteed to be positive definite.

Write code with NumPy to generate a 2-dim array with shape `(N,d)`, denoted as `data`, such that `data[n]` for any $n = 0, \cdots, N-1$ is a normal random variable with mean μ and covariance $\mathbf{\Sigma}$.