Assigned: 09 September

Homework #2

EE 541: Fall 2024

Due: Friday, 20 September at 22:00. Submission instructions will follow separately on brightspace.

1. **Linear MMSE Estimation**. Consider the problem of estimating a scalar random variable Y from a vector observation $\mathbf{X} \in \mathbb{R}^n$. We want to find the linear MMSE estimator $\hat{Y} = \mathbf{w}^T\mathbf{X}$ that minimizes the mean squared error (MSE), $\mathbb{E}[(Y - \hat{Y})^2]$.

   (a) Given two zero-mean jointly Gaussian random variables X and Y with covariance matrix

   $$\mathbf{K} = \begin{bmatrix} 5 & 2 \\ 2 & 4 \end{bmatrix},$$

   find the linear MMSE estimator $\hat{Y} = w^*X$ for Y given X. That is, find the optimal weight w^* that minimizes the MSE.

   (b) Calculate the minimum mean squared error (MMSE) achieved by the optimal estimator $\hat{Y} = w^*X$.

   (c) Show that for jointly Gaussian random variables, the linear MMSE estimator found in part (a) is equivalent to the conditional expectation $\mathbb{E}[Y|X]$. In other words, prove that $w^*X = \mathbb{E}[Y|X]$.

   (d) Now suppose X and Y are not jointly Gaussian but have the same covariance matrix $\mathbf{K}$ as above. Find the linear MMSE estimator $\hat{Y} = \tilde{w}X$ in this case. Is the MMSE achieved by $\tilde{w}$ different from the jointly Gaussian case in part (b)? Explain why or why not.

2. **Eigenanalysis of Covariance Matrix and PCA**. Consider a zero-mean random vector $\mathbf{X} \in \mathbb{R}^3$ with covariance matrix

   $$\mathbf{K} = \begin{bmatrix} 4 & -1 & 2 \\ -1 & 5 & -1 \\ 2 & -1 & 3 \end{bmatrix}.$$

   (a) Find the eigenvalues λ_k and orthonormal eigenvectors $\mathbf{e}_k$ of $\mathbf{K}$.

   (b) Show that the covariance matrix $\mathbf{K}$ can be expressed in terms of its eigenvalues and eigenvectors using the *spectral decomposition* (this is a special case of Mercer's theorem):

   $$\mathbf{K} = \sum_{k=1}^{3} \lambda_k \mathbf{e}_k \mathbf{e}_k^T.$$

(c) Express $\mathbf{X}$ using its Karhunen-Loève expansion (KL expansion), *i.e.,*

$$\mathbf{X} = \sum_{k=1}^{3} Z_k \mathbf{e}_k,$$

where Z_k are uncorrelated random variables with zero mean and variance equal to the corresponding eigenvalues λ_k. This expansion is closely related to Principal Component Analysis (PCA), where the eigenvectors of the covariance matrix are called principal components and the eigenvalues represent the variance, often intepreted as "power," captured by each component.

(d) Suppose you want to approximate $\mathbf{X}$ using only its two dominant eigenmodes (*i.e.,* the two principal components with the largest eigenvalues). Write the approximation $\tilde{\mathbf{X}}$ in terms of the eigenvectors and eigenvalues of $\mathbf{K}$. This is an example of dimensionality reduction using PCA.

(e) What is the mean squared error (MSE) of the approximation in (d), *i.e.,*

$$\mathbb{E}[\|\mathbf{X} - \tilde{\mathbf{X}}\|^2]?$$

Express your answer in terms of the eigenvalues. This MSE is related to the concept of reconstruction error in PCA and the total variance captured by the selected principal components.

3. The secant method is an iterative root-finding algorithm. It uses a sequence of secant line roots to approximate c such that $f(c) = 0$ for a continuous function f. Unlike Newton's method it does not require knowledge or evaluation of the derivative f'. The secant method is defined by the recurrence:

$$x_n = x_{n-1} - f(x_{n-1}) \frac{x_{n-1} - x_{n-2}}{f(x_{n-1}) - f(x_{n-2})}.$$

Write a python script that uses the secant method to approximate the root of a continuous function f in the interval $[a, b]$. You may assume that f has at most one root in $[a, b]$. Use $|x_{k+1} - x_k| < 10^{-10}$ as the convergence criterion. Let N be the number of iterations to reach convergence. Output N followed by the three root approximations x_{N-2}, x_{N-1}, x_N. **Output each number to its own line** and use precision sufficient to show convergence.

Import the function f from a file named `func.py` in the same directory as your script — *i.e.,* `from func import f`. You may assume that f is continuous on $[a, b]$ and that `func.f(x)` returns a scalar float for all $x \in [a, b]$.

Your script should accept `a` and `b` as two numeric command line arguments, *i.e.,* `python hw3p1.py "1.1" "1.4"`. Your script must validate that `a` and `b` are numeric, verify that `a < b`, and check that `f(a)f(b) < 0` — see Bolzano's Theorem. Write "`Range error`" to `STDERR` (standard error) if any of these three conditions fail and immediately terminate.

Your script should not produce any output except as described above.

4. Raman spectroscopy is a technique that uses inelastic scattering of light to identify unknown chemical substances. Spectral "peaks" indicate vibrational and rotational modes and are of special importance because they act like a chemical fingerprint. Raman spectroscopy measures photon intensity vs.

Raman shift. The Raman shift relates the frequencies of the exciting laser and the scattered photons and is often reported as a wavenumber — the frequency difference in wavelengths per cm (*i.e.,* , cm^{-1}).

- Generate a *molecular fingerprint* using the spectroscopic data in `raman.rod`. The file contains intensity vs. wavenumber data for an unknown chemical sample. A Raman Open Database (`ROD`) file includes content in addition to the raw intensity data:

```
# content
more content
_raman_spectrum.intensity
wavenumber1 intensity1
wavenumber2 intensity2
...
wavenumbern intensityn
```

  Use string matching to ignore all lines before `_raman_spectrum.intensity`. Load valid (`wavenumber, intensity`) pairs until the first invalid intensity line (or upon reaching the end of file).

  Use the method below to estimate the wavenumbers of all spectral peaks. You may use any standard `NumPy` or `SciPy` packages or experiment with your own algorithms.

- First detect peaks in the raw spectral data. Use the peak locations to focus on regions of interest within the spectrum. For instance: if you detect peaks at x_1 cm^{-1} and x_2 cm^{-1} use regions of interest: $[x_1 - n_1, x_1 + n_1]$ and $[x_2 - n_2, x_2 + n_2]$. Experiment to find "good" widths n_1, n_2, etc. Then use a spline to interpolate intensity within each region of interest. Calculate zero-crossings of the derivative to estimate wavenumbers with maximum intensity.

  (a) Print the wavenumber estimates for the eight largest spectral peak to STDOUT sorted by magnitude (largest first).

  (b) Create a figure that shows the Raman data (intensity vs. wavenumber) and mark each of the maximum intensity values.

  (c) Produce a "zoomed-in" figure for the "regions of interest" corresponding to the four largest peaks. Plot the raw spectral data and overlay your interpolating function. Use a marker to show the wavenumber with maximal intensity.