SE 165-265 Homework #8
Assigned May 22nd, 2024; due May 29th, 2024 (11:59 pm Pacific Time)

Assignment Goal:

1. Develop a data projection based on Fisher's linear discriminant that will take a multi-dimension feature vector and project it as a scalar on a line where the project attempts to maximize the separation in the class means while minimizing the class variances. The damage-sensitive features that will be used are the coefficients of an autoregressive model.

2. Reinforce material on supervised learning approaches to the statistical classification of the damage detection process. Reinforce the material on the use of time-series model parameters as damage-sensitive features.

<u>Background Reading:</u>
-Structural Health Monitoring: A Machine Learning Perspective, Chapter *7.11 Time Series Models*
-A. Farag, S. Elhabian, A Tutorial on Data Reduction Linear Discriminant Analysis
http://www.cvip.louisville.edu/wordpress/wp-content/uploads/2010/01/LDA-Tutorial-1.pdf

<u>Introduction</u>
We will continue to use experimental data from the 4-story structure. You can refer to Homework #1 to recall how the data were acquired. Recall that autoregressive (AR) models are a type of linear time-series model that can be fit to our measured response data. The parameters (coefficients) of these models can be used as damage sensitive features. The features are often of high dimension (i.e. the order of the AR model). This homework will examine a method to reduce the dimension of these features in a manner that best allows for correct classification of the damage cases.

With the 4-story structure, the first 9 states correspond to a system that is linear, but with different stiffness and mass values to simulate E&O variability one might encounter in real-world applications. Damage that is introduced with the bumper mechanism in states 10-17 results in nonlinear system response somewhat analogous to crack opening and closing. Fitting AR models to data from these cases results in a model that is the best linear approximation to the data from the nonlinear system. Therefore, we would also expect the AR models to change as a result of this transition from a linear to nonlinear system, which results from the introduction of the simulated damage. We will develop a method that allows us to distinguish the AR coefficient obtained from data acquired on the undamaged structure from the coefficients obtained from data acquired from the damaged structure.

<u>Task 0</u>: <u>Load data and create arrays (as in the previous assignments)</u>.

Download the file **data3SS2009.mat** from the SE 165-265 CANVAS Data File folder, 4-story structure data. **Load** this file in Matlab. This file contains measurements from all 5 sensors.

Loading this file will give you a 3-D matrix called **dataset** ($\mathbf{8192 \times 5 \times 850}$), where 8,192 corresponds to the data points, 5 corresponds to channels 1 to 5, and 850 is the number of measurements (50 measurements each for 17 states). Recall from HW 1 that states 1-9 are considered undamaged, and states 10-17 are damaged.

Recall that the data includes the (shaker input), **Channels 2 - 5** (response data from the accelerometers on the floors 1-4). In this assignment, **we will work only with data from channel 5 and the data from States 1, 10, 12 and 14.:**

Although not necessary, I first saved the data from channels 1-5 in new 8192x850 arrays called:

Channel 1 = **input**
Channel 2 = **sensor1**
Channel 3 = **sensor2**
Channel 4 = **sensor3**
Channel 5 = **sensor4**

In this assignment, we will not work with the input data.

We will work with sensors 1 to 4 (channels 2 to 5 from the 'dataset' matrix). We will need all 50 measurements from states 1, 10, 12, and 14. We will be able to observe how the damage sensitive features behave as the damage level increases and for the different sensor locations.

You will complete the tasks below for all 4 sensors.

<u>Task 1: Fit an AR model to the data from states 1, 10, 12 and 14</u>.

For all 4 sensors, fit AR(30) models to all time histories (all 50 measurements) from <u>states 1, 10, 12 and 14</u>. Store the AR(30) coefficients as column vectors in a 30x50 matrix for each state.

Use the **lpc** command in Matlab (specify order 30 model, e.g. a=lpc (x,30)) it estimates the AR coefficients by minimizing the following function: $\mathbf{(x_i - a_1 x_{i-1} - a_2 x_{i-2} - a_3 x_{i-3} - a_4 x_{i-4} - \ldots - a_{10} x_{i-10})^2}$**,** which is the square of the residual error. Note that you will have to specify x as double precision using the **double** function in Matlab before using the **lpc** command.

The LPC command returns a vector a={1, $-a_1$, $-a_2$, …$-a_{10}$). The first value of one is the coefficient on the x_i term and has to be eliminated by the command a(1)=[ ]. Also, the AR coefficients are in reverse order and of opposite sign to the ones we'd obtain by solving the least-squares problem (using **pinv**, as we did in HW 7). We won't worry about this ordering as we are only looking at the relative values of the coefficients.

Create a 2x2 grid and, for each of the 4 sensors, **plot the AR(30) coefficients for the first data sample (first measurement) from States 1, 10, 12 & 14 with the coefficient index on the x-axis and the coefficient value on they-axis.**

Use a legend to identify the four different sets of AR coefficients as follows: State 1 – red, State 10 – blue, State 12 – green, and State 14 – black.

Task 2: Calculate the Mean and covariance matrices for the Fisher Discriminant.

The Fisher projection of the AR(30) features (coefficients we just calculated) will be based on the data from States 1 and 10. The goal is to find a project vector, w, that maximizes the objective function J(w):

$$J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_{10}|}{\tilde{s}_1^2 + \tilde{s}_{10}^2} = \frac{w^T S_B w}{w^T S_w w}$$

Where:

$\tilde{\mu}_1 and \tilde{\mu}_{10}$ are the mean of the projected features from State 1, and State 10, respectively.

$\tilde{s}_1 and \tilde{s}_{10}$ are the scatter of the projected features from State 1 State 10, respectively.

S_b is the between-class scatter matrix, and

S_w is the within-class scatter matrix (we will calculate it in Task 3).

To develop this projection, <u>first calculate the mean vectors and the covariance matrices</u> for all 50 AR(30) feature vectors corresponding to these two states (state 1 and state 10). The mean vectors should be 30x1 column vectors. The covariance matrices should have dimension 30x30.

Create a 2x2 grid and, for each of the 4 sensors**, plot the two mean vectors** (state 1 and state 10) with the x-axis being the vector component index and they-axis being the value of the mean vector components. Provide a legend on the plot with the State 1 mean vector plotted in red and the State 10 mean vector plotted in blue.

Task 3 Calculate the within-class and between-class scatter matrices

Now use the means vectors and covariance matrices from Task 2 to calculate the within-class and between-class scatter matrices.

The within-class scatter matrix, S_w, is based on the covariance matrices, S_i for the features from State the two states w_i, i.e. States1 and State 10.

$$S_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

$$S_w = S_1 + S_2$$

The between-class scatter matrix, S_B, is based on the mean vectors from States 1 and 10

$$S_B = (\mu_1 - \mu_{10})(\mu_1 - \mu_{10})^T$$

Task 4 Solve the eigenvalue problem to find the Fisher project vector

Solve the eigenvalue problem defined by the within-class and between-class scatter matrices to find the Fisher project vector, w. Refer to Lecture #17 if needed.

$$S_w^{-1} S_B w = \lambda w$$

Note this vector is the one that corresponds to the largest eigenvalue. The matrix is of rank 1, at most, so there should be only one non-zero eigenvalue.

Create a 2x2 grid, and for each of the 4 sensors, **plot the Fisher projection vector** (w) with the x-axis being the vector component index and the y-axis being the value of the vector value.

Task 5 Project the AR(30) features for all the classes onto the Fisher coordinate

For each sensor, to project the data onto the Fisher coordinate, multiply the features (AR30 coefficients) by the transpose of the projection vector:

$$y = w^T * [x]$$

Note, use the same projection vector, w, (developed with data from States 1 & 10) for all the projections.

Task 6 Estimate the density function of the projected data sets using a kernel density estimator.

Now estimate the density functions of the four projected data sets using a kernel density estimate (**ksdensity** function in Matlab).

Create a 2x2 grid, and for each of the 4 sensors, plot the density functions for the projected data from States 1, 10, 12 &14 on a single plot.

Use a legend to identify the four different density functions with State 1 –red, State 10 – blue, State 12 – green and State 14 – black.