

# EE263: Introduction to Linear Dynamical Systems

## Final Exam

Name: \_\_\_\_\_

SUID: \_\_\_\_\_

Please circle the appropriate option for each of the following:

**Grading option:**      Letter grade              Credit/No credit

**Date:**                      August 14-15 (5pm)    August 15-16 (10am)

                                    August 15-16 (5pm)    Other (please specify):

I acknowledge and accept the Honor Code.

*(Signed)* \_\_\_\_\_

(For EE263 staff only)

Problem	Score
1	/20
2	/20
3	/20
4	/20
5	/20
6	/20
Total	/120

This is a 24-hour take-home final exam. If you are a local student, please submit your exam in Bytes Cafe (in the Packard building) 24 hours after you pick it up; if you are an SCPD student, please email your solutions to the SCPD distribution list:

`scpd-distribution@lists.stanford.edu`

as a single PDF file 24 hours after you receive the exam.

- You may use any books, notes, or computer programs (such as `MATLAB`), but you may not discuss the exam with others until Tuesday August 19, when everyone has taken the exam. The only exception is that you can ask the course staff for clarification by emailing to the staff email address `ee263-sum1314-staff@lists.stanford.edu`. Do not ask about the exam questions on Piazza! However, we've tried pretty hard to make the exam unambiguous and clear, so we're unlikely to say much.
- Since you have 24 hours, we expect your solutions to be legible, neat, and clear. Do not hand in your rough notes, and please try to simplify your solutions as much as you can. We will deduct points from solutions that are technically correct, but much more complicated than they need to be.
- Please check your email a few times during the exam, just in case we need to send out a clarification or other announcement.
- Attach the official exam cover page to your exam, and assemble your solutions to the problems in order: that is, problem 1, problem 2, etc. Start each solution on a new page. We will not go hunting for your work if the problems are not arranged in order; make sure that the code for each problem appears with the corresponding discussion and solution. (In particular, do not put all the code at the end.)
- Please make a copy of your exam before handing it in. We have never lost one, but it might occur.
- When a problem involves some computation, we do not want just the final answers. We want a clear discussion and justification of exactly what you did, the `MATLAB` source code that produces the result, and the final numerical result.
- Some of the problems are described in a practical setting, such as energy consumption, population dynamics, or wireless communications. *You do not need to understand anything about the application area to solve these problems.* We've taken special care to make sure all the information and math needed to solve the problem are given in the problem description.
- Some of the problems require you to download and run a `MATLAB` file to generate problem data. These files can be found at the URL

`http://www.stanford.edu/class/ee263/fs14/FILENAME`

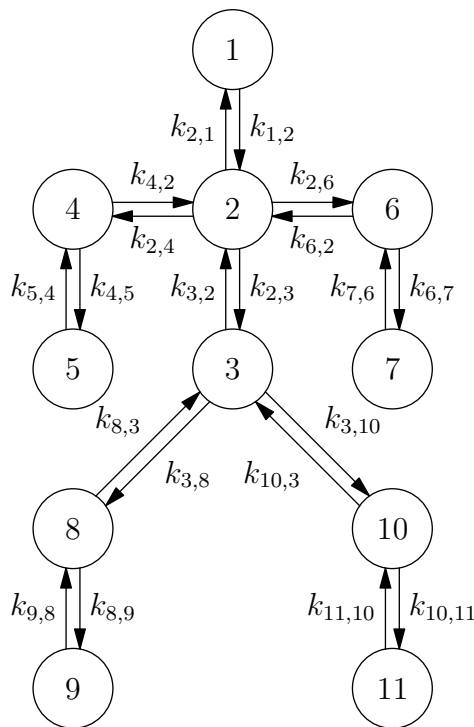
where you should substitute the particular filename (given in the problem) for `FILENAME`. *There are no links on the course web page pointing to these files, so you'll have to type in the whole URL yourself.*

# 1 Analysis of diffusion in a compartment model

Suppose we are interested in the concentration of some substance as a function of time at different locations in a system. For example, we may be interested in the concentration of a drug as a function of time in different parts of the body. Compartment models are commonly used to model such a situation. We divide the system into  $n$  discrete units, called compartments, and we assume that the concentration of the substance of interest is approximately constant throughout each compartment. Let  $x_i(t) \in \mathbb{R}$  be the concentration in compartment  $i$  at time  $t$ , and let  $x(t) = (x_1(t), \dots, x_n(t)) \in \mathbb{R}^n$  denote the vector of concentrations at time  $t$ . We assume that the concentration in compartment  $i$  evolves in time according to the equation

$$x_i(t + 1) = x_i(t) - \sum_{j \neq i} k_{ij} x_i(t) + \sum_{j \neq i} k_{ji} x_j(t), \quad i = 1, \dots, n,$$

where  $k_{ij}$  denotes the rate of diffusion from compartment  $i$  to compartment  $j$ . (Note that  $k_{ij}$  and  $k_{ji}$  need not be equal.) The first summation gives the total diffusion out of compartment  $i$  at time  $t$ , while the second summation gives the total diffusion into compartment  $i$  at time  $t$ . We assume that the diffusion coefficients  $k_{ij}$  are known. We can represent such a model using a graph, as in [figure 1](#).



**Figure 1** – a compartment model of the human body

Each of the nodes in the graph represents one of the compartments. If there is no edge from node  $i$  to node  $j$ , then there is no diffusion directly from compartment  $i$  to compartment  $j$ : that is,  $k_{ij} = 0$ . Each edge in the graph is labeled with the corresponding diffusion rate.

(a) Find a matrix  $A \in \mathbb{R}^{n \times n}$  such that

$$x(t + 1) = Ax(t).$$

- (b) For the rest of the problem, consider the specific compartment model with diffusion coefficients defined by the file `compartment_model_diffusion_data.m`. (The corresponding graph is the one given in [figure 1](#).) Consider any initial condition  $x(0)$  normalized such that  $\mathbf{1}^\top x(0) = 1$ . Compute the limit

$$\bar{x} = \lim_{t \rightarrow \infty} x(t),$$

which is the steady-state concentration profile. State any additional assumptions on  $x(0)$  that are needed to justify your analysis.

- (c) Let the difference between the current concentration profile  $x(t)$ , and the steady-state concentration profile  $\bar{x}$  be

$$e(t) = x(t) - \bar{x}.$$

Compute the limit

$$r = \lim_{t \rightarrow \infty} \frac{\|e(t+1)\|}{\|e(t)\|},$$

which gives the asymptotic rate at which  $x(t)$  converges to its steady-state value. State any additional assumptions on  $x(0)$  that are needed to justify your analysis.

- (d) Simulate the system for  $t = 0, \dots, T = 500$  starting from the initial condition  $x(0) = e_7$ . (Intuitively, the substance is introduced into the body at time  $t = 0$  via an intravenous injection into the left forearm.) Submit plots of  $x_{10}(t)$  and  $\|e(t)\|$  as functions of time. Use linear scales for  $t$  and  $x_{10}(t)$ , and a logarithmic scale for  $\|e(t)\|$ . Briefly comment on your plots in relation to the theoretical results you derived above.

## 2 Pairs of uncertainty ellipses

We want to estimate an unknown vector  $x \in \mathbb{R}^2$ . We have access to  $M$  noisy measurements of  $x$ :

$$y_i = A_i x + v_i, \quad i = 1, \dots, M,$$

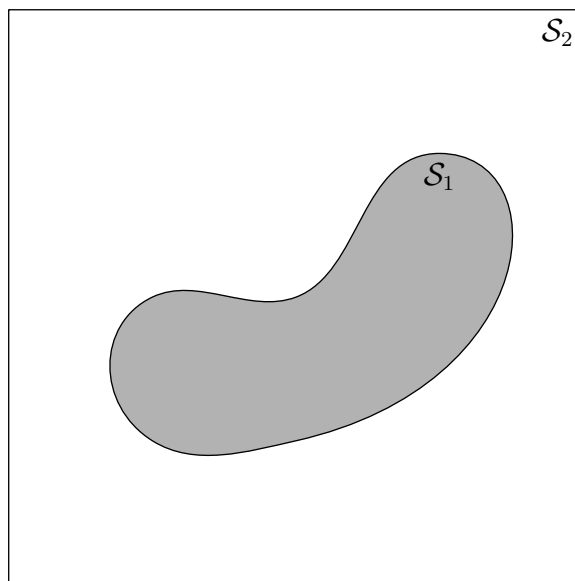
where  $y_i \in \mathbb{R}^{m_i}$  is the  $i$ th noisy measurement of  $x$ ,  $A_i \in \mathbb{R}^{m_i \times 2}$  is a skinny, full-rank matrix describing the  $i$ th measurement of  $x$ , and  $v_i \in \mathbb{R}^{m_i}$  is the noise in the  $i$ th measurement. We assume that  $\|v_i\| \leq \alpha_i$  for some known value of  $\alpha_i > 0$ . We want to choose the pair of measurements that gives us the best estimate of  $x$ . The file `uncertainty_ellipse_pairs_data.m` defines some example data.

- (a) The set of vectors  $x$  that are consistent with the  $i$ th measurement is an ellipse,  $\mathcal{E}_i$ . We know that there are two ways of representing such an ellipse:

$$\mathcal{E}_i = \{x \in \mathbb{R}^2 : (x - c_i)^\top S_i (x - c_i) \leq 1\} = \{d_i + B_i z : \|z\| \leq 1\},$$

where  $S_i \in \mathbb{S}_{++}^2$ ,  $B_i \in \mathbb{R}^{2 \times 2}$ , and  $c_i, d_i \in \mathbb{R}^2$ . Give expressions for  $S_i$ ,  $B_i$ ,  $c_i$ , and  $d_i$  in terms of  $y_i$ ,  $A_i$ , and  $\alpha_i$ . Submit a plot with all  $M$  ellipses on a single set of axes.

- (b) Recall that the area of an ellipse is  $\pi r_1 r_2$ , where  $r_1$  and  $r_2$  are the lengths of the semimajor and semiminor axes of the ellipse, respectively. Explain how to compute the area of  $\mathcal{E}_i$ . Report the area of each of the ellipses, and the indices  $i$  of the two ellipses with the smallest areas.
- (c) *Monte-Carlo area estimation.* Suppose we want to estimate the area of a set  $\mathcal{S}_1 \subseteq \mathbb{R}^2$ . Moreover, suppose that  $\mathcal{S}_1$  is a subset of another set  $\mathcal{S}_2 \subseteq \mathbb{R}^2$ , and that we know the area of  $\mathcal{S}_2$ . This situation is illustrated in [figure 2](#).



**Figure 2** – a complicated set  $\mathcal{S}_1$  that is a subset of a simple set  $\mathcal{S}_2$

If we choose points uniformly at random in  $\mathcal{S}_2$ , then the probability that a given point lies in  $\mathcal{S}_1$  is

$$p = \frac{\text{area}(\mathcal{S}_1)}{\text{area}(\mathcal{S}_2)}.$$

If we sample  $N$  such points  $z_1, \dots, z_N$  independently, then an estimate of  $p$  is

$$\hat{p} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(z_i \in \mathcal{S}_1),$$

where  $\mathbf{1}(z_i \in \mathcal{S}_1)$  is an indicator function:

$$\mathbf{1}(z_i \in \mathcal{S}_1) = \begin{cases} 1 & z_i \in \mathcal{S}_1, \\ 0 & \text{otherwise.} \end{cases}$$

We can use this estimate of  $p$  to estimate the area of  $\mathcal{S}_1$ :

$$\widehat{\text{area}}(\mathcal{S}_1) = \hat{p} \text{area}(\mathcal{S}_2).$$

Intuitively, we randomly generate points in  $\mathcal{S}_2$ , and our estimate of the area of  $\mathcal{S}_1$  is the area of  $\mathcal{S}_2$  times the fraction of the random points that were in  $\mathcal{S}_1$ . This procedure is called a Monte-Carlo method, after the famous Monte Carlo Casino in Monaco. (Monte-Carlo methods are a broad class of tools used for randomized estimation; in this problem, you only need to know about the Monte-Carlo method for area estimation described above.) Use Monte-Carlo area estimation with  $N = 100,000$  samples to estimate the areas of the ellipses  $\mathcal{E}_i$ . Explain how you chose the set  $\mathcal{S}_2$ , and computed its area. (You can choose a different  $\mathcal{S}_2$  for each  $\mathcal{E}_i$ ). Report your estimates of the areas of the ellipses, as well as the RMS error of your estimates.

- (d) The set of vectors  $x$  that are consistent with two measurements  $i$  and  $j$  is the intersection of the corresponding ellipses,  $\mathcal{E}_i$  and  $\mathcal{E}_j$ . For every pair  $i$  and  $j$ , compute the area of  $\mathcal{E}_i \cap \mathcal{E}_j$  using Monte-Carlo area estimation with  $N = 100,000$  samples. How do you choose the set  $\mathcal{S}_2$  used in the Monte-Carlo method? Report the indices  $i$  and  $j$  of the pair of ellipses whose intersection has the smallest estimated area. Also report your estimate of the corresponding area of the intersection.
- (e) Is choosing the pair of measurements that gives the smallest uncertainty region as simple as choosing the two measurements that independently give the smallest uncertainty regions? Briefly comment on the intuition behind your results above.

### 3 Maximum-variance affine subspace

You are given data points  $x_1, \dots, x_N \in \mathbb{R}^n$ , and you want to find an affine subspace  $S \subseteq \mathbb{R}^n$  of dimension  $k$  that captures as much of the variation in the data points as possible. More precisely, let  $\hat{x}_i(S)$  denote the projection of the data point  $x_i$  onto the subspace  $S$ . The mean of the projected data points is

$$\bar{x}(S) = \frac{1}{N} \sum_{i=1}^N \hat{x}_i(S),$$

and the variance of the projected points is

$$V(S) = \frac{1}{N} \sum_{i=1}^N \|\hat{x}_i(S) - \bar{x}(S)\|^2.$$

You want to choose the subspace  $S$  in order to maximize the variance  $V(S)$  of the projected points.

- (a) We can parameterize the affine subspace  $S$  using a vector  $a \in \mathbb{R}^n$ , and a matrix  $Q \in \mathbb{R}^{n \times k}$  with orthonormal columns:

$$S = \{a + Qz : z \in \mathbb{R}^k\}.$$

Explain how to choose  $a$  and  $Q$  in order to maximize  $V(S)$ .

- (b) A common task in data science is clustering: dividing the data points  $x_1, \dots, x_N$  into groups of “similar” points. Clustering is often easier for low-dimensional data than for high-dimensional data. For example, we can often cluster one- or two-dimensional data visually using a scatter plot. (You do not need to know about clustering algorithms in this problem; any clustering analysis will be done by inspection of scatter plots.) One approach to clustering high-dimensional data is to find a low-dimensional representation of the data, and then cluster the low-dimensional representation. This approach works well if the low-dimensional representation contains most of the variation in the data. Running the script `maximum_variance_affine_subspace_data.m` defines the following variables.

- $n$ , the dimension of the data
- $N$ , the number of data points
- $X$ , an  $n \times N$  matrix whose columns are the data points

Find the affine subspace  $S$  of dimension  $k = 2$  that maximizes  $V(S)$ . Note that the variance of the original (unprojected) data points is  $V(\mathbb{R}^n)$ . Report the fraction of the variance in the data points that is captured by the subspace  $S$ :

$$\frac{V(S)}{V(\mathbb{R}^n)}.$$

Make a two-dimensional scatter plot of the projected data points  $\hat{x}_1, \dots, \hat{x}_N$ . (Since the columns of  $Q$  are an orthonormal basis for  $S$ , you can use the columns of  $Q$  as the axes

for your scatter plot.) For comparison, report the fraction of the variance in the data points that is captured by the subspace  $\tilde{S} = \text{span}(e_1, e_2)$ , and make a two-dimensional scatter plot of the projections of the data points onto the subspace  $\tilde{S}$ . Compare the scatter plots showing the projections of the data onto  $S$  and  $\tilde{S}$ . In particular, comment on whether the plots can be used to identify clusters.

## 4 Model-order reduction for linear equations

Suppose  $x \in \mathbb{R}^n$  is a vector of unknown parameters, and  $y_1 \in \mathbb{R}^{m_1}$  and  $y_2 \in \mathbb{R}^{m_2}$  are linear functions of  $x$ :

$$y_1 = A_1 x \quad \text{and} \quad y_2 = A_2 x,$$

where  $A_1 \in \mathbb{R}^{m_1 \times n}$ , and  $A_2 \in \mathbb{R}^{m_2 \times n}$  are known, and  $A_1$  is skinny and full rank. We are given a noisy measurement  $\tilde{y}_1 = y_1 + v \in \mathbb{R}^{m_1}$  of  $y_1$ , where  $v \in \mathbb{R}^{m_1}$  is measurement noise, and we want to use this noisy measurement to estimate  $y_2$ . Executing the script `model_order_reduction_linear_equations_data.m` defines the following variables.

- `m1`, `m2`, and `n`, the problem dimensions  $m_1$ ,  $m_2$ , and  $n$
  - `xtrue`, the true value of  $x \in \mathbb{R}^n$
  - `A1` and `A2`, the matrices  $A_1 \in \mathbb{R}^{m_1 \times n}$  and  $A_2 \in \mathbb{R}^{m_2 \times n}$
  - `y1tilde`, the noisy measurement  $\tilde{y}_1 \in \mathbb{R}^{m_1}$  of  $y_1$
- (a) Let  $\hat{x}$  be the least-squares approximate solution of  $\tilde{y}_1 = A_1 x$ . Form the estimate  $\hat{y}_2 = A_2 \hat{x}$ . Report the relative errors

$$\epsilon_x = \frac{\|x - \hat{x}\|}{\|x\|} \quad \text{and} \quad \epsilon_{y_2} = \frac{\|y_2 - \hat{y}_2\|}{\|y_2\|}.$$

- (b) Make a stem plot of the singular values of the matrix

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \in \mathbb{R}^{(m_1+m_2) \times n}.$$

How many “significant” singular values does  $A$  have?

- (c) Let  $p$  be the number of significant singular values of  $A$ , and write the singular-value decomposition of  $A$  as

$$A = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} [V_1 \quad V_2]^\top,$$

where

$$U_{11} \in \mathbb{R}^{m_1 \times p}, \quad U_{12} \in \mathbb{R}^{m_1 \times (n-p)}, \quad U_{21} \in \mathbb{R}^{m_2 \times p}, \quad U_{22} \in \mathbb{R}^{m_2 \times (n-p)}$$

$$\Sigma_1 \in \mathbb{R}^{p \times p}, \quad \Sigma_2 \in \mathbb{R}^{(n-p) \times (n-p)}, \quad V_1 \in \mathbb{R}^{n \times p}, \quad \text{and} \quad V_2 \in \mathbb{R}^{n \times (n-p)}.$$

The best rank- $p$  approximation of  $A$  is

$$\hat{A} = \begin{bmatrix} \hat{A}_1 \\ \hat{A}_2 \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} [V_1 \quad V_2]^\top,$$

where we have set the block  $\Sigma_2$  of “insignificant” singular values equal to zero. Explain how to use the measurement  $\tilde{y}_1$  and the approximate model  $y_1 \approx \hat{A}_1 x$  to estimate the vector  $z = V_1^\top x \in \mathbb{R}^p$ . Report your estimate  $\hat{z}$ , the true value of  $z$ , and the relative error

$$\epsilon_z = \frac{\|z - \hat{z}\|}{\|z\|}.$$

- (d) Explain how to use  $\hat{z}$  to estimate  $y_2$ . Let this estimate be  $\hat{y}_2^{(z)}$ . Report the relative error

$$\epsilon_{y_2}^{(z)} = \frac{\|y_2 - \hat{y}_2^{(z)}\|}{\|y_2\|}.$$

- (e) Briefly explain the intuition behind your results, and why the term “model-order reduction” might be used to describe the method of estimating  $y_2$  using  $z$ .

## 5 Voting patterns in the United States Senate

The file `senate_voting_patterns_data.m` contains the results of every roll-call vote held by the Senate of the 107th United States Congress (that is, in the years 2001–2002). In particular, running `senate_voting_patterns_data.m` defines the following variables.

- `senators`, a  $102 \times 3$  cell array (there are 102 rows because Senator Wellstone died during his term, and was replaced by Senator Barkley; and Senator Jeffords, who switched parties during his term, recorded votes as both a Republican and an Independent)
  - `senators{i,1}` is the last name of the  $i$ th Senator
  - `senators{i,2}` is the first name of the  $i$ th Senator
  - `senators{i,3}` is the party of the  $i$ th Senator
- `votes`, a  $102 \times 633$  matrix where `votes(i,j)` is the vote cast by Senator  $i$  in vote  $j$  (+1 if Senator  $i$  voted “Yea” on vote  $j$ ,  $-1$  if Senator  $i$  voted “Nay” on vote  $j$ , and 0 if Senator  $i$  did not participate in vote  $j$ )

- (a) Make a stem plot of the singular values of `votes`. How many significant singular values are there? The fraction of the variation in the voting data associated with the first two singular values is

$$f_2 = \frac{\sigma_1^2 + \sigma_2^2}{\sum_{i=1}^r \sigma_i^2}.$$

What is  $f_2$  for the Senate voting data?

- (b) Missing votes will complicate our subsequent analysis, so we use a low-rank model of the voting matrix to guess the missing votes. Initialize  $A$  to the observed voting matrix; repeat the following steps until convergence:
- replace  $A$  with its best rank-two approximation;
  - replace the entries of  $A$  corresponding to the known votes with their true values;
  - replace the entries of  $A$  corresponding to the unknown votes by setting nonnegative entries to  $+1$ , and negative entries to  $-1$ .

Report the numbers of  $+1$ s and  $-1$ s in your final matrix  $A$ .

- (c) Let  $A = U\Sigma V^T$  be the singular-value decomposition of  $A$ . We want to find interpretations for the first two left singular vectors,  $u_1$  and  $u_2$ .
- (i) The function `spatial_plot`, which is available on the course website, can be used to generate color-coded scatter plots. Use this function to make a scatter plot of  $(u_1)_i$  versus  $(u_2)_i$  where Republicans, Democrats, and Independents are represented by red, blue, and green markers, respectively. You can make such a plot using the command

```
spatial_plot(u1 , u2 , z , 3 , eye(3));
```

where  $\mathbf{u}_1$  and  $\mathbf{u}_2$  are the first two left singular vectors of  $A$ , and  $\mathbf{z}$  is a vector where  $\mathbf{z}(i)$  is 0 if Senator  $i$  is a Republican, 1 if Senator  $i$  is an Independent, and 2 if Senator  $i$  is a Democrat.

- (ii) Make a scatter plot showing the fraction of votes in which each Senator voted with the majority of the other Senators. You can make such a plot using the command

```
spatial_plot(u1 , u2 , z , 10);
```

where  $\mathbf{u}_1$  and  $\mathbf{u}_2$  are the first two left singular vectors of  $A$ , and  $\mathbf{z}$  is a vector where  $\mathbf{z}(i)$  is the fraction of votes  $j$  for which

$$A_{ij} = \text{sgn}\left(\sum_{i=1}^{102} A_{ij}\right).$$

(By default, `spatial_plot` uses the `cool` color map, which is pink for large values, and blue for small values.)

Based on these plots, give intuitive interpretations of  $(u_1)_i$  and  $(u_2)_i$ .

- (d) We can also find interpretations for the first two right singular vectors.

- (i) Make a scatter plot of  $(v_1)_j$  versus  $(v_2)_j$  where the colors indicate the total support received by vote  $j$ . In particular, the color should correspond to

$$z_j = \sum_{i=1}^{102} A_{ij}.$$

- (ii) Make a scatter plot of  $(v_1)_j$  versus  $(v_2)_j$  where the colors indicate the partisan support received by vote  $j$ . In particular, let  $R \subseteq \{1, \dots, 102\}$  be the set of Republicans, and let  $D \subseteq \{1, \dots, 102\}$  be the set of Democrats. The color in your plot should correspond to

$$z_j = \frac{1}{|R|} \sum_{i \in R} A_{ij} - \frac{1}{|D|} \sum_{i \in D} A_{ij}.$$

Based on these plots, give intuitive interpretations of  $(v_1)_i$  and  $(v_2)_i$ .

## 6 Facial recognition

Suppose we are given  $F_1, \dots, F_N \in \mathbb{R}^{m \times n}$ , where each  $F_k$  represents an image of a face. Moreover, we assume that we know whose face is shown in each of these images (often we will have more than one image of each person's face). Then, we are given another image  $Y \in \mathbb{R}^{m \times n}$  not in the data set  $F_1, \dots, F_N$ , and we want to determine whose face is shown in the new image. The file `facial_recognition_data.mat` contains the following variables.

- $N$ ,  $m$ , and  $n$ , the problem dimensions
  - $F$ , a cell array of length  $N$ , where  $F\{k\}$  is an  $m \times n$  matrix encoding the  $k$ th sample image
  - $Y$ , an  $m \times n$  matrix encoding the image that we want to identify
  - $r$ , the number of singular vectors to use for our low-rank approximation (see below for details)
- (a) Submit a plot of the image  $Y$  that we want to identify. You can generate such a plot in MATLAB using the following code.

```
figure();  
colormap gray;  
imagesc(Y);  
axis off;
```

- (b) Find the image in the data set  $F_1, \dots, F_n$  that is closest to  $Y$  in the least-squares sense: that is, find the value of  $k$  that minimizes

$$\|Y - F_k\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n (Y_{ij} - (F_k)_{ij})^2.$$

Report your value of  $k$ , and submit a plot of the image  $F_k$ . Were you able to correctly identify the subject?

- (c) Let  $f_k \in \mathbb{R}^{mn}$  be a vectorization of the matrix  $F_k \in \mathbb{R}^{m \times n}$  for  $k = 1, \dots, N$ . (We can form such a vectorization in MATLAB using the expression `F{k}(:)`.) Let  $\bar{f} \in \mathbb{R}^{mn}$  be the average value of the  $f_k$ :

$$\bar{f} = \frac{1}{N} \sum_{k=1}^N f_k.$$

Form the matrix  $X \in \mathbb{R}^{(mn) \times N}$  such that

$$X = [f_1 - \bar{f} \quad \cdots \quad f_N - \bar{f}].$$

We know that the left singular vectors of  $X$  give us a low-dimensional approximation of  $\text{span}(f_1, \dots, f_N)$ . Compute the first  $r$  singular values and left singular vectors of  $X$ . Make a plot of

$$p_i = \frac{\sum_{\tilde{i}=1}^i \sigma_{\tilde{i}}^2}{\sum_{\tilde{i}=1}^N \sigma_{\tilde{i}}^2}, \quad i = 1, \dots, r$$

versus  $i$ . Explain how to compute  $\sum_{i=1}^N \sigma_i^2$  without computing all of the singular values of  $X$ . We can think of  $p_i$  as the proportion of the variation in the sample images that is captured by the first  $i$  left singular vectors of  $X$ . Report  $p_r$ .

- (d) The following code plots the first  $r$  images in the data set.

```
nrow = 4;
ncol = 5;
figure();
colormap gray;
for k = 1:r
    subplot(nrow,ncol,k);
    imagesc(F{k});
    axis off;
end
```

Adapt this code to plot the images corresponding to the first  $r$  left singular vectors of  $X$ . (The MATLAB code `U = reshape(u, [m n])` converts a vector  $u \in \mathbb{R}^{mn}$  into a matrix  $U \in \mathbb{R}^{m \times n}$ .) Submit your plot, and briefly comment on what you see. (The left singular vectors of  $X$  are often called eigenfaces in the literature.)

- (e) Let  $y \in \mathbb{R}^{mn}$  be the vectorization of the image  $Y$ , and let  $U_d \in \mathbb{R}^{(mn) \times d}$  denote the matrix whose columns are the first  $d$  left singular vectors of  $X$ . Define the vector  $\hat{y}_d \in \mathbb{R}^{mn}$  such that

$$\hat{y}_d = \bar{f} + U_d U_d^T (y - \bar{f}).$$

Show that  $\hat{y}_d$  is the projection of  $y$  onto the  $d$ -dimensional affine subspace that minimizes the total residual of the projections of  $f_1, \dots, f_N$ . (Recall that the minimum-residual affine subspace of dimension  $d$  is  $\bar{f} + \text{range}(U_d)$ .) Submit a plot with the images corresponding to  $\hat{y}_d$  for  $d = 1, \dots, r$ .

- (f) Let  $\hat{f}_k \in \mathbb{R}^{mn}$  denote the vector

$$\hat{f}_k = \bar{f} + U_r U_r^T (f_k - \bar{f}).$$

Show that the value of  $k$  that minimizes

$$\|\hat{f}_k - \hat{y}_r\|^2 = \sum_{i=1}^{mn} ((\hat{f}_k)_i - (\hat{y}_r)_i)^2$$

is the same as the value of  $k$  that minimizes

$$\|U_r^T (f_k - y)\|.$$

(Note that using  $\|U_r^T (f_k - y)\|$  is much more efficient than using  $\|\hat{f}_k - \hat{y}_r\|^2$ .) Report your value of  $k$ , and make a plot of the image  $F_k$ . Were you able to correctly identify the subject?

- (g) In a real-world facial-recognition system, we may have millions of sample images, and  $r$  may be in the hundreds or thousands. In terms of storage requirements, explain why a system based on the eigenfaces  $U_1, \dots, U_r$  may be preferable to a system based on the sample images  $F_1, \dots, F_N$ . (It is sufficient to give a simple argument based on how many numbers are needed to store  $U_r$  versus how many numbers are needed to store  $F_1, \dots, F_N$ .)