EE263 Final Exam, December 2019

This is a 24 hour take-home exam. Please turn it in using Gradescope 24 hours after you receive it.

- Please turn in your code along with your solutions.
- You may use any books, notes, or computer programs (*e.g.*, Julia, Python or Matlab). You can Google anything. But **you may not discuss the exam with anyone** until Dec. 11, after everyone has taken the exam. The only exception is that you can ask the TAs or Professor Lall for clarification, by emailing to the staff email address `ee263-aut1920-staff@lists.stanford.edu`. 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.
- 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. Be sure to show us your verification that your computed solution satisfies whatever properties it is supposed to, at least up to numerical precision. For example, if you compute a vector x that is supposed to satisfy $Ax = b$ (say), show us the code that checks this, and the result. (This might be done by the code `norm(A*x-b)`; be sure to show us the result, which should be very small.) *We will not check your numerical solutions for you, in cases where there is more than one solution.*
- In the portion of your solutions where you explain the mathematical approach, you *cannot* refer to programming language operators, such as the backslash operator. (You can, of course, refer to inverses of matrices, or any other standard mathematical construct.)
- Some of the problems may be described in a practical setting, (imaging systems, energy consumption, population dynamics, or wireless communications). Others may discuss mathematical areas (Markov chains, differential equations). *You do not need to understand anything about these practical settings or mathematical areas to solve these problems.* We've taken special care to make sure all the information and math needed to solve the problem is given in the problem description.
- Some of the problems require you to download data files. These files can be found at the URL

  `http://ee263.stanford.edu/finalquestions.html`

There are no links on the course web page pointing there, so you'll have to type in the whole URL yourself. The data files are in `json` format. In Julia, use `readclassjson` to load the files. In Matlab, use `jason_helper.m` to load the files. In Python, use `jason_helper.py` to load the files. These are available at the above web page.

- At the same web page as the data files we will post notification of any exam typos or problems. We recommend you check this site occasionally during the exam.
- For problems involving coding, please include your codes as part of your pdf submission and correctly assign them to the corresponding problems on gradescope. Make sure to double check that the pages have been correctly assigned for each problem, and don't wait until the last minute to submit.
- Good luck!

1. Periodic drone motion. You are operating a drone with the goal of continuously observing a particular set of locations. For example, you may be monitoring bird nesting sites to keep track of populations and behavior. You know that the birds tend to feed at particular locations at particular times of day, so you want to have the drone be at those locations at the right times.

We'll model the drone as obeying Newton's laws, which is a reasonable model at this scale. At time t, the drone will be at position $y(t) \in \mathbb{R}^2$ and acted on by a force $u(t) \in \mathbb{R}^2$. We'll discretize, assuming piecewise constant forces, with a sampling period of 1. This gives the equations of motion as a discrete-time linear dynamical system as follows.

$$x(t+1) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} \frac{1}{2} & 0 \\ 1 & 0 \\ 0 & \frac{1}{2} \\ 0 & 1 \end{bmatrix} u(t)$$

The state is $x = (y_1, \dot{y}_1, y_2, \dot{y}_2)$. We have m nesting sites, with the ith nesting site given by the ith column of P, where

$$P = \begin{bmatrix} 0.8 & 1.2 & -0.9 & -1.0 & 0.2 \\ -1.2 & 1.0 & 1.2 & -1.0 & -1.1 \end{bmatrix}$$

We would like to visit these sites at times s_i, where

$$s = \begin{bmatrix} 10.0 & 15.0 & 40.0 & 50.0 & 70.0 \end{bmatrix}$$

a) Our first objective is to have the drone visit the sites. Design a trajectory for which $y(s_i) = p_i$ for all $i = 1, \ldots, m$. We will have the drone start stationary at the origin. We would of course like to minimize battery usage, and so we decide to minimize

$$J_{\text{effort}} = \sum_{t=0}^{T-1} \|u(t)\|^2$$

Here $T = 90$ is the flight time. Find the optimal trajectory of the drone. What is the minimum value of J_{effort}?

Plot u as a function of time. Also make a plot which shows the route the drone takes, and includes the waypoints. This should have axes y_1 and y_2.

b) Actually we want the drone to continually monitor these sites, and so we want the drone to go back to visiting p_1 again after finishing visiting p_m. So we would like to design a periodic trajectory y which satisfies

$$y(t+T) = y(t) \text{ for all } t \geq 0$$

This of course means that the drone will visit the nesting sites periodically, so we will have

$$y(s_i + kT) = p_i \text{ for all } k \geq 0$$

We also do *not* need the drone to start from stationary at the origin. The position and velocity of the drone at time $t = 0$ are also to be chosen. Find the periodic trajectory that minimizes J_{effort}. What is the minimum value of J_{effort}? Make the same two plots as in the previous part.

c) The drone is actually controlled remotely from a base station at the origin, and so we need the drone to stay within range $r = 2$. To encourage this behavior, we decide to add an additional objective, of minimizing

$$J_{\text{range}} = \sum_{t=0}^{T-1} \|y(t)\|^2$$

Of course, we still want the drone to visit the sites at the right times. For positive γ, find the trajectory that minimizes

$$J_{\text{range}} + \gamma J_{\text{effort}}$$

Plot the trade-off curve of J_{range} versus J_{effort}. For each value of γ in

$$\gamma = 1, 10, 100, 10^3, 10^4, 10^5, 10^6$$

plot the route the drone takes, and the waypoints.

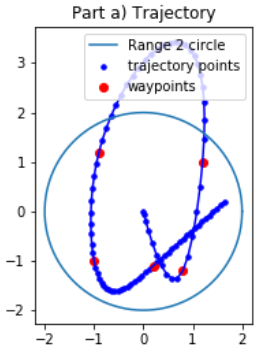
d) Find (by searching) the largest γ such that the drone stays within range? What is the corresponding J_{range} and J_{effort}? For this γ, plot the route the drone takes, and the waypoints.

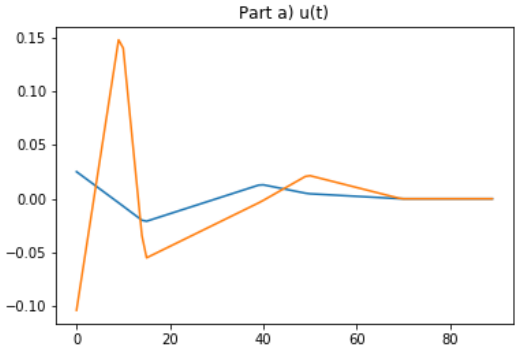
Solution.

a) Using the control matrix from lecture notes, which we will call G, where each block-row corresponds to $[A^{t-1}B \; A^{t-2}B \; ...B \; 0...0]$. Note there is no A^t leading term because we know in this part that the initial state is 0, so the autonomous part will be zero. So we can define G of dimension $20 \times 2T$, with each ith block-row of size $4 \times 2T$ corresponding to the state at the i-th waypoint time ($i = 1, \ldots, 5$). We can define a block matrix C of size 10×20 where each block-row of size 2×20 is responsible for extracting the y_1, y_2 components of the state from the corresponding block of rows from Gu where u is the length $2T$ control vector. Then defining b to be a column vector which consists of the columns of P stacked, our problems amounts to

$$\begin{array}{ll} \min\limits_{u} & \|u\|^2 \\ \text{subject to} & CGu = b \end{array}$$

This is just least norm, so defining $F = CG$ we can solve for $\hat{u} = (FF^T)^{-1}F^Tb$. The minimum value of J_{effort} is 0.141158. The plots follow:





b) Now, we have the additional issue of determining our initial condition. No trouble: we will add it to a block free vector z where we interpret $z = \begin{bmatrix} x(0) \\ u \end{bmatrix}$ Additionally, we must modify the matrix G to G_1, for which each block-row of G_1 includes the autonomous part:

$$[A^t \ A^{t-1}B \ A^{t-2}B \ ...B \ 0...0]$$

and to add another block row of size $4 \times (2T+2)$corresponding to time $T = 90$ (to be specified below), so now G_1 is of dimension $24 \times (2T+4)$. Additionally, we modify C to C_1, which extracts y_1, y_2 from the first 5 blocks of size 4, and then to extract all 4 entries for the final four, corresponding to the final state at time $T = 90$. We note that for the system to be periodic, we need the $x(90) = x(0)$. To embed this constraint, we note that in the final block corresponding to time $T = 90$ of our linear equality constraint we will need to have (with G denoting the block row of G_1 from the previous part with no autonomous part):

$$\begin{bmatrix} A^T & G \end{bmatrix} \begin{bmatrix} x(0) \\ u \end{bmatrix} = x(0) = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} x(0) \\ u \end{bmatrix}$$

Thus, we can specify the last block row of G to be $\begin{bmatrix} A^T - I & G \end{bmatrix}$ so our constraint in that block is

$$\begin{bmatrix} A^T - I & G \end{bmatrix} z = 0$$

We are almost ready to do constrained least norm, but we do not want to consider $\|z\|^2$ since this would include the norm of $y(0)$. So lastly we must construct

$$W = \begin{bmatrix} 0_{2T \times 4} & I_{2T \times 2T} \end{bmatrix}$$

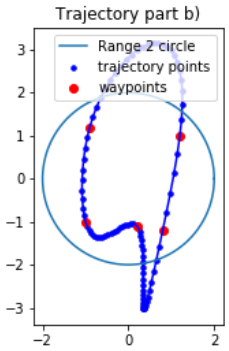
so that $Wz = u$. Lastly, we let

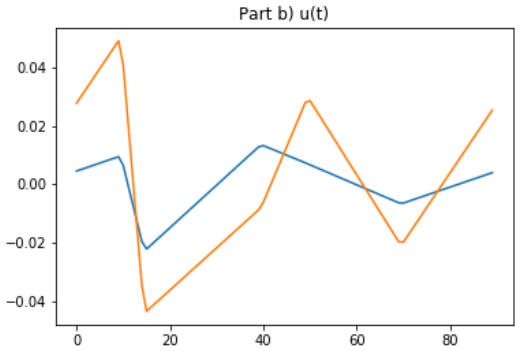
$$b_1 = \begin{bmatrix} b \\ 0_{4 \times 1} \end{bmatrix}$$

to deal with our last constraint block. Finally, our problem is

$$\begin{aligned} \min_{z} \quad & \|Wz\|^2 \\ \text{subject to} \quad & C_1 G_1 z = b_1 \end{aligned}$$

which we know how to solve as a generalized least norm problem. The J_{effort} for the solution is 0.0557285. The same two plots as above follow:





c) To modify our solution from above, all we need is to change W to

$$W_1 = \begin{bmatrix} \sqrt{\gamma} W \\ C_2 G_2 \end{bmatrix}$$

so that

$$\|W_1 z\|^2 = J_{\text{range}} + \gamma J_{\text{effort}}.$$

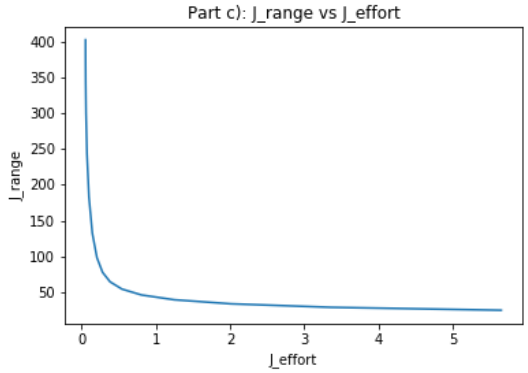
Notice that here the $C_2 \in \mathbb{R}^{2T \times 4T}$ matrix and the $G_2 \in \mathbb{R}^{4T \times (2T+4)}$ matrix are in charge of selecting the positions $y(t)$ all time points $t = 0, 1, \ldots, T-1$ out.

The problem is then

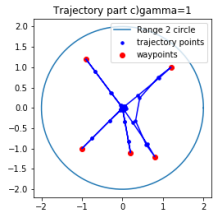
$$\begin{aligned} \min_{z} \quad & \|W_1 z\|^2 \\ \text{subject to} \quad & C_1 G_1 z = b_1, \end{aligned}$$

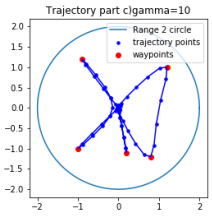
where $C_1,\ G_1,\ b_1$ are the same as in part b).

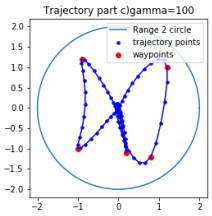
Solving and plotting the tradeoff curve:

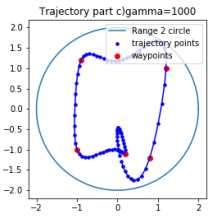


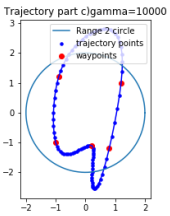
As well as the trajectories for each given γ:

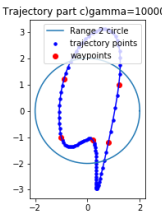


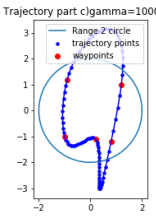








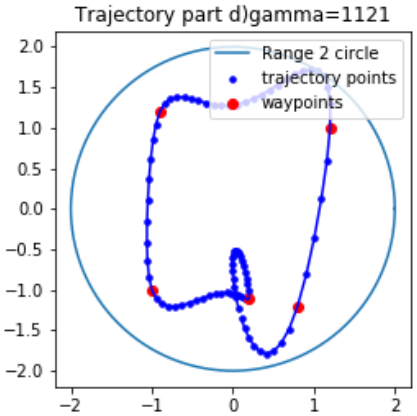




d) By searching with resolution .1, the largest is $\gamma = 1121$. J_{range} and J_{effort} are respectively

155.84516053149727 0.12219935785471174

The plot of the trajectory follows:



2. Affine sets.

a) An affine set $\mathcal{A} \subset \mathbb{R}^m$ is a translated subspace. More precisely it is a set of the form

$$\mathcal{A} = \{x \mid Ax = a\}$$

where $A \in \mathbb{R}^{m \times n}$ and $a \in \mathbb{R}^m$. This way of writing an affine set is called the *implicit form*. An affine set can always be written in another form, called the *explicit form*, as follows. There is always an integer k, a matrix $C \in \mathbb{R}^{n \times k}$ and a vector $c \in \mathbb{R}^n$ such that

$$\mathcal{A} = \{Cz + c \mid z \in \mathbb{R}^k\}$$

Suppose you are given A and a. Give a method for finding k, C and c. What determines the smallest k which can be used?

b) We can also convert from explicit to implicit form. Give an algorithm for finding A, a given C, c.

c) Apply your method to

$$C = \begin{bmatrix} 37 & -100 & -15 \\ -16 & 230 & 20 \\ -66 & 100 & -10 \\ 96 & -34 & 52 \\ 11 & 24 & -20 \end{bmatrix} \qquad c = \begin{bmatrix} 2 \\ 15 \\ 2 \\ -13 \\ 2 \end{bmatrix}$$

d) We now turn to computing the distance between two affine sets. For two sets $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^m$, define the distance between them to be

$$\text{dist}(\mathcal{A}, \mathcal{B}) = \min_{x \in \mathcal{A}, y \in \mathcal{B}} \|x - y\|$$

We have two affine sets in implicit form, which are given by

$$\mathcal{A} = \{x \mid Ax = a\} \qquad \mathcal{B} = \{y \mid By = b\}$$

We are given the matrices $A, B \in \mathbb{R}^{m \times n}$ and the vectors $a, b \in \mathbb{R}^m$, and would like to compute the distance $\text{dist}(\mathcal{A}, \mathcal{B})$. We would like to convert this problem to a least-squares problem. Explain how to find a matrix F and a vector g such that

$$\text{dist}(\mathcal{A}, \mathcal{B}) = \min_z \|Fz + g\|$$

Solution.

a) Let $C \in \mathbb{R}^{m \times k}$ is the null space basis matrix of A. Then, we know that $\mathcal{A} = \{c + Cz \mid z \in \mathbb{R}^k\}$, where c is an arbitrary solution to $Ax = a$. In particular, we can take c to be the least-norm solution $c = A^T(AA^T)^{-1}a$. To find the null space

basis matrix C, we can take the singular value decomposition (SVD) of a matrix $A \in \mathbb{R}^{m \times n}$:

$$A = U\Sigma V^T$$

where the orthogonal matrix $U \in \mathbb{R}^{m \times m}$, the diagonal matrix $\Sigma = \text{diag}\left(\sigma_1, \ldots, \sigma_{\min(m,n)}\right) \in \mathbb{R}^{m \times n}$ where $\sigma_1 \geq \cdots, \geq \sigma_{\min(m,n)} \geq 0$, and the orthogonal matrix $V \in \mathbb{R}^{n \times n}$. σ_i is the i-th singular value of A and the i-th column of U and i-th column of V are the corresponding left singular vector and right singular vector. The smallest k would be the number of singular values equal to zero and C is the columns of V corresponding to singular values equal to zero.

b) We can find the orthonomal basis $D \in \mathbb{R}^{n \times p}$ of $C \in \mathbb{R}^{n \times k}$. We can find this by using SVD of C. Similar to our method on part a), let $C = U\Sigma V^T$, find nonzero singular values and D is the columns of U corresponding to these singular values. Then, we find the orthonormal complement E of D. Note that E is simply the columns of U corresponding to the singular values equal to zero. Here, We have $E^T D = 0$. Then, we can simply take A to be E^T and $a = E^T c$. The reason is that for any $y = Cx + c$, we have $Ay = ACx + Ac = Ac = E^T c$.

c) Here is the solution in Python

```
import numpy as np

C = np.array( [ [37, -100, -15], [-16, 230, 20], [-66, 100, -10], [96, -34, 52],
c = np.array([2, 15, 2, -13, 2])

u, s, vh = np.linalg.svd(C)

r = np.linalg.matrix_rank(np.diag(s))

n,k = np.shape(C)

A = np.zeros((n-r, n))
for i in range(n-r):
    A[i,:] = u[:, r+i]
a = A.dot(c)

print("matrix A is \n", A)
print("vector a is \n", a)
```

Here is the solution in Julia

```
using LinearAlgebra
```

```
C = [37 -100 -15; -16 230 20; -66 100 -10; 96 -34 52; 11 24 -20];

c = [2; 15; 2; -13; 2];

U,s,V = svd(C, full=true);
r = rank(C);
n = 5;
k = 3;

A = zeros(Float64, (n-r, n));
for i =1:1:(n-r)
    A[i,:] = U[:, r+i]';
end
a = A*c;

println("A matrix is")
println(A)
println("a vector is")
println(a)
```

we have

$$A = \begin{bmatrix} -0.0706355 & -0.3281738 & 0.71845135 & 0.41540268 & 0.44562411 \\ -0.78302584 & -0.21801873 & -0.40176604 & -0.05862257 & 0.41771499 \end{bmatrix}$$

$$a = \begin{bmatrix} -8.13596185 \\ -4.04234126 \end{bmatrix}$$

d) Since Z_A is the null space basis matrix of A, we know that $\mathcal{A} = \{x_A + Z_A y_A \,|\, y_A \in \mathbb{R}^k\}$, where x_A is an arbitrary solution to $Ax = a$. In particular, we can take x_A to be the least-norm solution $x_A = A^T(AA^T)^{-1}a$, which is well-defined since A has a full row rank (and hence AA^T is invertible). Similarly, $\mathcal{B} = \{x_B + Z_B y_B \,|\, y_B \in \mathbb{R}^p\}$, where $x_B = B^T(BB^T)^{-1}b$.

Hence using definition, we have

$$\begin{aligned} \text{dist}(\mathcal{A}, \mathcal{B}) &= \inf_{y_A \in \mathbb{R}^k,\ y_B \in \mathbb{R}^p} \|x_A + Z_A y_A - (x_B + Z_B y_B)\| \\ &= \inf_y \|[Z_A, -Z_B]y - (x_B - x_A)\|, \end{aligned}$$

where last expression is obtained by rearanging terms and setting $y = (y_A, y_B)$. Hence the distance can be expressed as a least squares problem of the form minimize $\|Hx - g\|$ with $H = [Z_A, -Z_B]$ and $g = x_B - x_A$. Then, the solution to $\text{dist}(\mathcal{A}, \mathcal{B})$ is $y^* = (H^T H)^{-1} H^T g$.

3. Constraints. For the following problems, if relevant, let $A \in \mathbb{R}^{n\times n}$. For each part, you must justify your answer for credit.

a) Let $f(x) = t^{10}+7t^9-6t^8+\pi t^7-\pi t^6+5t^3+2t-7$ and $\mathcal{C} = \text{span}(\{1, t^2, t^4, t^6, t^8, t^{10}\})$. What is the optimal $g(t)$ to

$$\min_{g(t)\in\mathcal{C}} \int_{-1}^{1} |f(t) - g(t)|^2 dt?$$

b) Find the solution (*i.e.*, the optimal X) to

$$\min_{X\in\mathbb{R}^{n\times n}} \|X - A\|_F^2.$$

c) Find the solution (*i.e.*, the optimal X) to

$$\min_{X\in\mathcal{A}} \|X - A\|_F^2,$$

where $\mathcal{A}$ is the set of all $n \times n$ matrices that are a scalar multiple of the identity matrix.

d) Find the solution (*i.e.*, the optimal X) to

$$\min_{X\in\mathcal{B}} \|X - A\|_F^2,$$

where $\mathcal{B}$ is the set of all $n \times n$ symmetric matrices.

e) Find the solution (*i.e.*, the optimal X) to

$$\min_{X\in\mathcal{D}} \|X - H\|_F^2,$$

where $\mathcal{D} = \{Z \mid Z \succeq \text{diag}(d)\}$, $d \in \mathbb{R}^n$ is a known vector, and $H \in \mathbb{R}^{n\times n}$ is a symmetric matrix. Recall $\text{diag} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ constructs a diagonal matrix with diagonal equal to its argument. For example, $\text{diag}(\mathbf{1}) = I$. Also recall that a matrix $W \succeq 0$ if and only if W is positive semidefinite.

(Fun fact / very small hint: $\mathcal{D}$ is a generalization of the set of positive semidefinite matrices.)

Solution.

a) Optimal $g(t)$ is $g(t) = t^{10} - 6t^8 - \pi t^6 - 7$. The error $e(t) = f(t) - g(t)$ is an odd function, and every signal in $\mathcal{C}$ is an even function. Thus $e(t)$ is orthogonal to $\mathcal{C}$ and $g(t)$ above is optimal.

b) Trivial $X = A$.

c) We can recast the problem as

$$\min_{\gamma \in \mathbb{R}} \|\gamma I - A\|_F^2.$$

Taking the gradient with respect to γ, we get

$$\begin{aligned}
\nabla_\gamma \|\gamma I - A\|_F^2 &= \nabla_\gamma \operatorname{trace}(\gamma^2 I - 2\gamma A + A^2) \\
&= \frac{d}{d\gamma} \operatorname{trace}(\gamma^2 I - 2\gamma A + A^2) \\
&= 2\gamma \operatorname{trace}(I) - 2 \operatorname{trace}(A) \\
&= 2\gamma N - 2 \operatorname{trace}(A).
\end{aligned}$$

Setting this to zero, we see that the optimal γ is $\gamma = \operatorname{trace}(A)/N$. Thus, the optimal X is $X = (\operatorname{trace}(A)/N)I$.

d) Since for every symmetric matrix X we have a decomposition $X = (1/2)(Y + Y^T)$, we can recast the problem as

$$\min_{Y \in \mathbb{R}^{n \times n}} \|(1/2)(Y + Y^T) - A\|_F^2.$$

Thus, the optimal $Y = (1/2)A$. Thus, the optimal $X = (1/2)(A + A^T)$.

e) Let the eigenvalue decomposition of H be given as

$$H = T \Lambda T^{-1},$$

where $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$.

Let $\tilde{\Lambda} \in \mathbb{R}^{n \times n}$ be

$$\tilde{\Lambda} = \operatorname{diag}(\max\{\lambda_1, d_1\}, \ldots, \max\{\lambda_n, d_n\}).$$

Thus, the solution of this problem is given by

$$X = T \tilde{\Lambda} T^{-1}.$$

4. Solving risk-constrained portfolio optimization in closed form. We consider the problem of selecting a portfolio composed of n assets. Here $x_i \in \mathbb{R}$ denotes the investment, measured in dollars, in asset i. If $x_i < 0$ this means that we hold a short position in this asset. The vector $x \in \mathbb{R}^n$ is called the *holding* vector. By holding such a portfolio, we have a mean return, given by $\mu^\mathsf{T} x$. Here $\mu \in \mathbb{R}^n$ is called the *return* vector.

There is a type of risk exposure in holding such a portfolio due to the fluctuations in the returns of the assets. This fluctuation is characterized by the matrix $\Sigma \in \mathbb{R}^{n \times n}$, which is symmetric and positive definite. (One can make a statistical justification for

using the return covariance matrix, but we don't need that in this question.) The amount of risk associated with a particular holding is then $x^\mathsf{T}\Sigma x$.

The goal is to maximize the total return, subject to the constraint that $x^T\Sigma x \leq \rho$, where $\rho > 0$ is a number which expresses our tolerance for overall risk. That is, we would like to solve

$$\begin{array}{ll} \text{maximize} & \mu^\mathsf{T} x \\ \text{subject to} & x^\mathsf{T}\Sigma x \leq \rho \end{array}$$

a) It turns out that this problem can be solved analytically; there is an explicit expression for the optimal x in terms of μ, ρ and Σ. Find such an expression. You may use common techniques from linear algebra (SVD, eigenvalues, norms, etc.)

b) Is the solution unique? Please verify your claim.

c) Show that the optimal holding vector x satisfies $\|x\| \geq \sqrt{\rho/\|\Sigma\|}$.

d) For $\Sigma = I$, provide a geometric interpretation of the solution(s).

Solution.

a) If $\mu = 0$, then any $x \in \mathbb{R}^n$ is an optimal solution. Now assume that $\mu \neq 0$. Let $\Sigma = T\Lambda T^T$ be the eigen-decomposition of the PD matrix Σ, where T is orthogonal and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix consisting of the eigenvalues of Σ. Let $y = T^T x$. Then the constraint is equivalent to $y^T\Lambda y \leq \rho$. Hence we can rewrite the objective as

$$\begin{aligned} \mu^T x &= c^T y \\ &= \sum_{i=1}^n c_i y_i \\ &= \sum_{i=1}^n \frac{c_i}{\sqrt{\lambda_i}}(\sqrt{\lambda_i} y_i) \\ &\leq \sqrt{\sum_{i=1}^n \frac{c_i^2}{\lambda_i}} \sqrt{\sum_{i=1}^n \lambda_i y_i^2} \\ &\leq \sqrt{\rho} \sqrt{\sum_{i=1}^n \frac{c_i^2}{\lambda_i}}. \end{aligned}$$

where $c = T^T\mu$, and the fourth row uses the Cauchy inequality. Here we also used the fact that $\lambda_i > 0$. Since $\mu \neq 0$ and T is orthogonal, we also have $c \neq 0$. Hence the Cauchy inequality becomes equality if and only if $\sqrt{\lambda_i} y_i = \alpha c_i/\sqrt{\lambda_i}$ for

some constant $\alpha > 0$. This implies that $y_i = \alpha \frac{c_i}{\lambda_i}$. Then the inequality on the third row is an equality if and only if $\sum_{i=1}^n \lambda_i y_i^2 = \alpha^2 \sum_{i=1}^n \frac{c_i^2}{\lambda_i} = \rho$, which yields $\alpha = \sqrt{\frac{\rho}{\sum_{i=1}^n c_i^2/\lambda_i}}$. Hence the solution x is

$$x = Ty = \sqrt{\frac{\rho}{\sum_{i=1}^n c_i^2/\lambda_i}} T \begin{bmatrix} c_1/\lambda_1 \\ \vdots \\ c_n/\lambda_n \end{bmatrix} = \frac{\sqrt{\rho}}{\sqrt{c^T \Lambda c}} T\Lambda^{-1} c = \frac{\sqrt{\rho}}{\sqrt{\mu^T T \Lambda^{-1} T^T \mu}} T\Lambda^{-1} T^T \mu$$
$$= \frac{\sqrt{\rho}}{\sqrt{\mu^T \Sigma^{-1} \mu}} \Sigma^{-1} \mu,$$

where we used the fact that $c = T^T \mu$. Notice that this can also be obtained by applying KKT conditions.

b) Yes if $\mu = 0$ – the above derivation already shows that the optimal solution is unique. No if $\mu \neq 0$. We give full credit to all reasonable answers.

c) To see this, first notice that $\|\Sigma\| = \lambda_{\max}(\Sigma)$. Then since T is orthogonal, we have

$$\|x\|^2 = \left\| \sqrt{\frac{\rho}{\sum_{i=1}^n c_i^2/\lambda_i}} \begin{bmatrix} c_1/\lambda_1 \\ \vdots \\ c_n/\lambda_n \end{bmatrix} \right\|^2 = \rho \frac{\sum_{i=1}^n c_i^2/\lambda_i^2}{\sum_{i=1}^n c_i^2/\lambda_i}.$$

Then for the denominator, we have (again by Cauchy inequality)

$$\sum_{i=1}^n c_i^2/\lambda_i \leq \sqrt{\sum_{i=1}^n (c_i/\lambda_i)^2} \sqrt{\sum_{i=1}^n c_i^2}$$
$$= \sqrt{\sum_{i=1}^n (c_i/\lambda_i)^2} \|c\| = \|\mu\| \sqrt{\sum_{i=1}^n (c_i/\lambda_i)^2},$$

where we used the fact that $\|T^T \mu\| = \|\mu\|$.

Hence we arrive at

$$\|x\|^2 \geq \rho \frac{\sqrt{\sum_{i=1}^n (c_i/\lambda_i)^2}}{\|\mu\|}$$
$$\geq \rho \frac{\sqrt{\sum_{i=1}^n c_i^2/\lambda_{\max}(\Sigma)}}{\|\mu\|}$$
$$\geq \rho \frac{\|\mu\|/\|\Sigma\|}{\|\mu\|}$$
$$= \frac{\rho}{\|\Sigma\|}.$$

This completes the proof.

Alternative solution. Simply noticing that $\rho = x^T \Sigma x$ at the optimal solution, we have

$$\rho \leq \|\Sigma\| \|x\|^2,$$

which implies the desired inequality.

d) When $\Sigma = I$, the solution reduces to $x = \frac{\sqrt{\rho}}{\|\mu\|}\mu$, *i.e.*, projection onto the sphere of radius $\sqrt{\rho}$ centered at the origin.

5. Embedding data based on pairwise distances. In this problem, you will design an algorithm to visualize relationships between potentially high dimensional or non-numeric data in lower dimensions. Consider the setting where m objects have been analyzed, and for each object $i = 1, \ldots, m$ we have a *feature vector* $x_i \in \mathbb{R}^n$. Unfortunately n is large, and so we would like to replace these high-dimensional data points with lower dimensional data points $y_1, \ldots, y_m \in \mathbb{R}^d$. Often we want $d = 2$ or $d = 3$ so we can visualize the data. We would like the new data points y_i to roughly preserve the arrangement of the high-dimensional data x_i. Specifically, we want

$$\|y_i - y_j\| \approx \|x_i - x_j\| \qquad \text{for all } i, j$$

a) We construct a matrix of the data $X = \begin{bmatrix} x_1 & x_2 & \ldots & x_m \end{bmatrix}$. Let D^X be the corresponding matrix of squared pairwise distances squared

$$D^X_{ij} = \|x_i - x_j\|^2$$

and let G^X_{ij} be the matrix of pairwise inner products

$$G^X_{ij} = x_i^\mathsf{T} x_j$$

For a square matrix $G \in \mathbb{R}^{m \times m}$, use the notation $g = \operatorname{diag} G$ to mean $g \in \mathbb{R}^m$ and $g_i = G_{ii}$; that is g is the vector of elements on the diagonal of G. Show that

$$D^X = (\operatorname{diag} G^X)\mathbf{1}^\mathsf{T} + \mathbf{1}(\operatorname{diag} G^X)^\mathsf{T} - 2G^X$$

b) Define the square matrix

$$H = I - \frac{\mathbf{1}\mathbf{1}^\mathsf{T}}{m}$$

Suppose $XH = X$. What does this mean geometrically?

c) Suppose $XH = X$ (you may assume this for the rest of the problem). Show that

$$HD^X H = -2G^X$$

Notice this means that we have a one-to-one map between G^X and D^X.

d) Given any symmetric matrix $D \in \mathbb{R}^{m\times m}$, we can ask the question whether the entries of the matrix D could possibly be the pairwise distances between a hypothetical set of data; that is, does there exist n and $X \in \mathbb{R}^{n\times m}$ such that $D = D^X$. Show that this holds if and only if

$$HDH \preceq 0.$$

Also recall that a matrix $W \succeq 0$ if and only if W is positive semidefinite.

e) Now let's address our original problem, of finding points $y_1, \ldots, y_m$ whose pairwise distance are close to those of $x_1, \ldots, x_m$. That is, we are looking to make an embedding of the m data points in d dimensions. Ideally we would like $D^Y \approx D^X$. We know G^X and D^X are related as above, and so

$$G^X - G^Y = -\frac{1}{2}H(D^X - D^Y)H$$

So if $D^X \approx D^Y$ then we must have $G^X \approx G^Y$. We therefore decide to choose $Y \in \mathbb{R}^{d\times m}$ to minimize

$$\|G^X - G^Y\|_F$$

Explain how to do this.

f) And now let's do it! We have provided 3 data sets. From the file `USmap.json`, `Dmap` is the pairwise spherical distances between US cities. Use the technique you developed above to find an embedding in 2 dimensions (this corresponds to flattening a globe onto a map). The list `labels` contains the names of the cities. Please plot your embedding with the cities labeled (what you see should look familiar).

The files `digits2and6.json` and `digits8and9.json` contain `D2and6` and `D8and9` corresponding to the pairwise distances squared of 8x8 images of handwritten digits (the former from a subset of 2's and 6's and the latter from 8's and 9's), as well as the identities of each image (which digit it was) stored in `labels`. Use the same technique to embed the originally 64 dimensional dataset in 2 dimensions, and color the points according to their label (one color for each of the two digits). In one or two sentences comment on what you notice between the two datasets's embeddings.

Solution. Here is the solution.

a) $D_{ij} = \|x_i - x_j\|^2 = \|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2$. Note the middle term in matrix form is $-2X^T X$. The first term agrees with the ij'th entry of $(\operatorname{diag} G^X)\mathbf{1}^T$ and similarly for the latter term.

b) XH operates on X by subtracting the mean from each column. If $XH = X$ this means X already has columns whose mean is 0.

c) Let $Z = (\operatorname{diag} G^X)\mathbf{1}^T$. Note that $\frac{1}{N}Z\mathbf{1}\mathbf{1}^T = Z$, $\frac{1}{N}\mathbf{1}\mathbf{1}^T Z^T = Z^T$. Then

$$\begin{aligned} HDH &= (I - \frac{1}{N}\mathbf{1}\mathbf{1}^T)(Z - 2X^TX + Z^T)(I - \frac{1}{N}\mathbf{1}\mathbf{1}^T) \\ &= Z - \frac{1}{N}\mathbf{1}\mathbf{1}^T Z - Z + \frac{1}{N}\mathbf{1}\mathbf{1}^T Z - 2H^TX^TXH + Z^T - Z^T - \frac{1}{N}Z^T\mathbf{1}\mathbf{1}^T + \frac{1}{N}Z^T\mathbf{1}\mathbf{1}^T \\ &= -2X^TX = -2G^X \end{aligned}$$

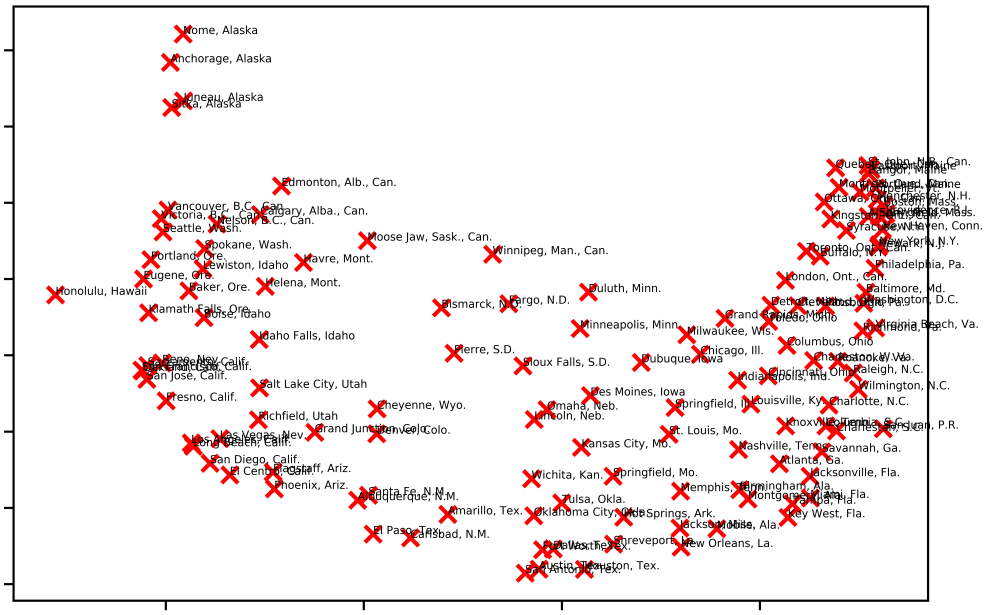
where we used our initial observations and the assumption that $XH = H$.

d) $\Rightarrow$ Assume that there is such a data-set X. Then $D = D^X$ and $HD^XH = -2G^X = -2X^TX$. But for any v, $v^T(-2X^TX)v = -2\|Xv\|^2 \leq 0$ so $HDH \preceq 0$.

$\Leftarrow$ Suppose $HDH \preceq 0$. Then note $-\frac{1}{2}HDX \succeq 0$. Since $-\frac{1}{2}HDH$ is symmetric, we can eigen-decompose: $-\frac{1}{2}HDH = U\Lambda U^T$. Because the matrix is PSD, we know all of the eigenvalues and hence all of the entries in the diagonal matrix Λ are non-negative. Thus, we can define $\Lambda^{1/2}$ by $(\Lambda^{1/2})_{ii} = (\Lambda_{ii})^{1/2}$ and the off diagonal entries zero. But then, consider $X = \Lambda^{1/2}U^T$. $-2X^TX = HDH$ and so we can construct D^X from $G^X = X^TX$ to find $D^X = X$.

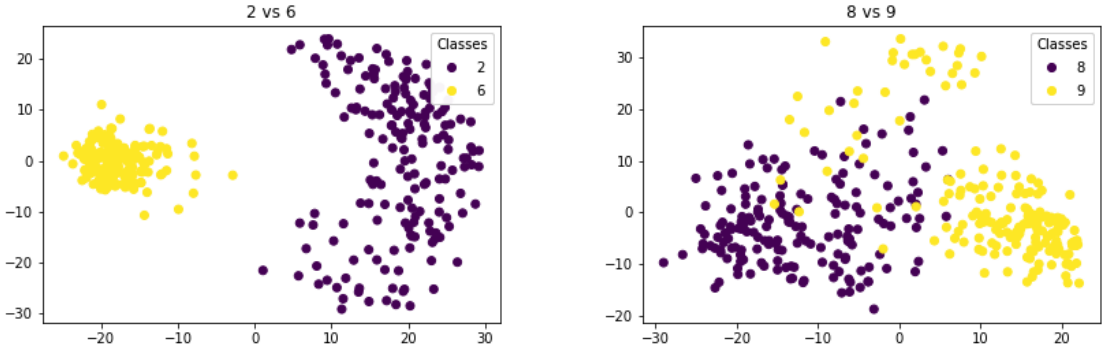
e) We know that the general solution to $\min_{A:\operatorname{rank}(A)\leq p} \|A - B\|_F$ is to take A as the SVD of B with all but the first p entries on the diagonal of Σ zeroed. We know that for a data set Y in d dimensions, Y^TY will have rank at most d. Thus, we can use our insight from the previous problem to take the SVD of $-\frac{1}{2}HDH$ and let $Y = \Lambda_d^{1/2}U_d^T$ where the d subscripts refer to taking a diagonal matrix of size d and only the first d rows of U^T.

f) On the given data, we compute $-\frac{1}{2}HDH$, compute its eigen-decomposition, and take the square root as described above. Plotting, we see for the US map:



which certainly looks familiar.

For the digits we see



The key observation is the 2 and 6 and more easily separated then 8 and 9: this makes sense when you consider the visual similarity between the two pairs. 2's and 6's should be close to other 2's and 6's other and far from each other, but the same is not true for 8's and 9's, which similar both intra- and inter-class.

6. Worst-case confidence ellipsoid. Consider the scenario where we observe

$$y = Ax + e,$$

Here $y \in \mathbb{R}^m$ is measured, $x \in \mathbb{R}^n$ is unknown, and $A \in \mathbb{R}^{m \times n}$ is a matrix that represents the effects of our sensors. You may assume A is skinny and full rank. In a sensing problem, we often measure y and try to determine x. This is made more difficult by e, which represents noise or measurement uncertainty. We do not know the vector e, but we do know that $\|e\| \leq 1$.

a) Since we do not know e, we cannot typically determine x with certainty. Instead, if we know A and y, there is a set of x consistent with the model above. This is called the *worst-case confidence ellipsoid* associated with the measurement. Let S be this set. That is

$$\mathcal{S} = \left\{x \in \mathbb{R}^n \mid y = Ax + e \text{ for some } e \text{ with } \|e\| \leq 1\right\}$$

For example, let

$$A = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \quad y = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

What is $\mathcal{S}$?

Sketch this situation. Clearly label y, the range of A, and the set $\mathcal{S}$.

b) In general there exists $c \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n \times n}$ such that

$$\mathcal{S} = \{x \mid (x - c)^T Q^{-1} (x - c) \leq 1\}$$

Find c and Q.

c) Given

$$A = \begin{bmatrix} -2.3 & -1.1 \\ -1.3 & -1.5 \\ 0.5 & 0.2 \\ 0.2 & 0.4 \\ -0.3 & 0.5 \end{bmatrix}, \qquad y = (0, 1, 0, 0, 1),$$

and $\|e\|_2 \leq 1$, find the set of all possible x consistent with this information.

Solution. Here is the solution.

a) This is parts A and B.
By definition, we have

$$\mathcal{S} = \{x \mid \exists e \text{ such that } \|e\|_2 \leq 1 \text{ and } y = Ax + e\},$$

which is equivalent to

$$\mathcal{S} = \{x \mid \|y - Ax\|_2^2 \leq 1\}.$$

Note that

$$\begin{aligned}
\|y - Ax\|_2^2 \leq 1 &\implies (y - Ax)^T (y - Ax) \leq 1 \\
&\implies y^T y - 2x^T A^T y + x^T A^T A x \leq 1 \\
&\implies y^T y - 2x^T A^T y + x^T A^T A x - y^T y + y^T A (A^T A)^{-1} A^T y \leq 1 - y^T y + y^T A (A^T A)^{-1} A^T y \\
&\implies y^T A (A^T A)^{-1} A^T y - 2x^T A^T y + x^T A^T A x \leq 1 - y^T y + y^T A (A^T A)^{-1} A^T y \\
&\implies (x - (A^T A)^{-1} A^T y)^T (A^T A)(x - (A^T A)^{-1} A^T y) \leq 1 - y^T y + y^T A (A^T A)^{-1} A^T y \\
&\implies (x - (A^T A)^{-1} A^T y)^T \left(\frac{A^T A}{1 - y^T y + y^T A (A^T A)^{-1} A^T y} \right) (x - (A^T A)^{-1} A^T y) \leq 1
\end{aligned}$$

Thus, $Q = \left(\frac{A^T A}{1 - y^T y + y^T A (A^T A)^{-1} A^T y} \right)^{-1}$ and $c = (A^T A)^{-1} A^T y$.

For $A = [3, 1]^T$ and $y = [4, 2]^T$, we have

$$Q = \left(\frac{A^T A}{1 - y^T y + y^T A (A^T A)^{-1} A^T y} \right)^{-1} = 0.06, \qquad c = (A^T A)^{-1} A^T y = 1.4.$$

Most sketches were accepted as long as they conveyed the correct information.

b) We have

$$Q = \begin{bmatrix} -0.3021 & 0.3484 \\ 0.3484 & -0.5686 \end{bmatrix}, \qquad c = (-0.2069, -0.0171).$$