EE263 Homework 4

Fall 2025

4.660. Norm preserving implies orthonormal columns. In lecture we saw that if $A \in \mathbb{R}^{m\times n}$ has orthonormal columns, *i.e.*, $A^\mathsf{T}A = I$, then for any vector $x \in \mathbb{R}^n$ we have $\|Ax\| = \|x\|$. In other words, multiplication by such a matrix preserves norm.

Show that the converse holds: If $A \in \mathbb{R}^{m\times n}$ satisfies $\|Ax\| = \|x\|$ for all $x \in \mathbb{R}^n$, then A has orthonormal columns (and in particular, $m \geq n$).

Hint. Start with $\|Ax\|^2 = \|x\|^2$, and try $x = e_i$, and also $x = e_i + e_j$, for all $i \neq j$.

Solution. Suppose that $\|Ax\| = \|x\|$ for all x. Then

$$\|Ax\|^2 = x^\mathsf{T}(A^\mathsf{T}A)x = \|x\|^2 = x^\mathsf{T}x$$

for all x. Let's start with $x = e_i$. The equation above then reduces to

$$e_i^\mathsf{T}(A^\mathsf{T}A)e_i = (A^\mathsf{T}A)_{ii} = 1,$$

and we see that the diagonal entries in $A^\mathsf{T}A$ are all one. Now let's plug in $x = e_i + e_j$, to get

$$\begin{aligned}(e_i+e_j)^\mathsf{T}(A^\mathsf{T}A)(e_i+e_j) &= e_i^\mathsf{T}(A^\mathsf{T}A)e_i + e_j^\mathsf{T}(A^\mathsf{T}A)e_i + e_i^\mathsf{T}(A^\mathsf{T}A)e_j + e_j^\mathsf{T}(A^\mathsf{T}A)e_j\\ &= 2 + 2e_j^\mathsf{T}(A^\mathsf{T}A)e_i\\ &= (e_i+e_j)^\mathsf{T}(e_i+e_j)\\ &= 2.\end{aligned}$$

We conclude that $e_j^\mathsf{T}(A^\mathsf{T}A)e_i = (A^\mathsf{T}A)_{ij} = 0$, for $j \neq i$. Thus, the off-diagonal elements of $A^\mathsf{T}A$ are zero. So we have $A^\mathsf{T}A = I$.

7.1060. Curve-smoothing. We are given a function $F : [0,1] \to \mathbb{R}$ (whose graph gives a curve in $\mathbb{R}^2$). Our goal is to find another function $G : [0,1] \to \mathbb{R}$, which is a *smoothed* version of F. We'll judge the smoothed version G of F in two ways:

- *Mean-square deviation from F,* defined as

$$D = \int_0^1 (F(t) - G(t))^2\, dt.$$

- *Mean-square curvature,* defined as

$$C = \int_0^1 G''(t)^2\, dt.$$

We want *both* D and C to be small, so we have a problem with two objectives. In general there will be a trade-off between the two objectives. At one extreme, we can choose $G = F$, which makes $D = 0$; at the other extreme, we can choose G to be an affine function (*i.e.*, to have $G''(t) = 0$ for all $t \in [0,1]$), in which case $C = 0$. The problem is to identify the optimal trade-off curve between C and D, and explain how to find smoothed functions G

on the optimal trade-off curve. To reduce the problem to a finite-dimensional one, we will represent the functions F and G (approximately) by vectors $f,\ g \in \mathbb{R}^n$, where

$$f_i = F(i/n), \quad g_i = G(i/n).$$

You can assume that n is chosen large enough to represent the functions well. Using this representation we will use the following objectives, which approximate the ones defined for the functions above:

- *Mean-square deviation,* defined as

$$d = \frac{1}{n}\sum_{i=1}^{n}(f_i - g_i)^2.$$

- *Mean-square curvature,* defined as

$$c = \frac{1}{n-2}\sum_{i=2}^{n-1}\left(\frac{g_{i+1} - 2g_i + g_{i-1}}{1/n^2}\right)^2.$$

In our definition of c, note that

$$\frac{g_{i+1} - 2g_i + g_{i-1}}{1/n^2}$$

gives a simple approximation of $G''(i/n)$. You will only work with this approximate version of the problem, *i.e.*, the vectors f and g and the objectives c and d.

a) Explain how to find g that minimizes $d + \mu c$, where $\mu \geq 0$ is a parameter that gives the relative weighting of sum-square curvature compared to sum-square deviation. Does your method always work? If there are some assumptions you need to make (say, on rank of some matrix, independence of some vectors, etc.), state them clearly. Explain how to obtain the two extreme cases: $\mu = 0$, which corresponds to minimizing d without regard for c, and also the solution obtained as $\mu \to \infty$ (*i.e.*, as we put more and more weight on minimizing curvature).

b) Get the file `curve_smoothing.json` from the course web site. This file defines a specific vector f that you will use. Find and plot the optimal trade-off curve between d and c. Be sure to identify any critical points (such as, for example, any intersection of the curve with an axis). Plot the optimal g for the two extreme cases $\mu = 0$ and $\mu \to \infty$, and for three values of μ in between (chosen to show the trade-off nicely). On your plots of g, be sure to include also a plot of f, say with dotted line type, for reference.

Solution.

a) Let's start with the two extreme cases. When $\mu = 0$, finding g to minimize $d + \mu c$ reduces to finding g to minimize d. Since d is a sum of squares, $d \geq 0$. Choosing $g = f$ trivially achieves $d = 0$. This makes perfect sense: to minimize the deviation measure, just take the smoothed version to be the same as the original function. This yields zero deviation, naturally, but also, it yields no smoothing! Next, consider the extreme case where $\mu \to \infty$. This means we want to make the curvature as small as possible. Can we drive it to zero? The answer is yes, we can: the curvature is zero if and only if g is an affine function, *i.e.*, has the form $g_i = ai + b$ for some constants a and b. There are lots of vectors g that have this form; in fact, we have one for every pair of numbers a, b. All of these vectors g make c zero. Which one do we choose? Well, even if μ is huge, we still have a small contribution to $d + \mu c$ from d, so among all g that make $c = 0$, we'd like the one that minimizes d. Basically, we want to find the best affine approximation, in the sum of squares sense, to f. We want to find a and b that minimize

$$\left\| f - A \begin{bmatrix} a \\ b \end{bmatrix} \right\| \text{ where } A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \\ \vdots & \vdots \\ n & 1 \end{bmatrix}.$$

For $n \geq 2$, A is skinny and full rank, and a and b can be found using least-squares. Specifically, $[a\ b]^\mathsf{T} = (A^\mathsf{T} A)^{-1} A^\mathsf{T} f$. In the general case, minimizing $d + \mu c$, is the same as choosing g to minimize

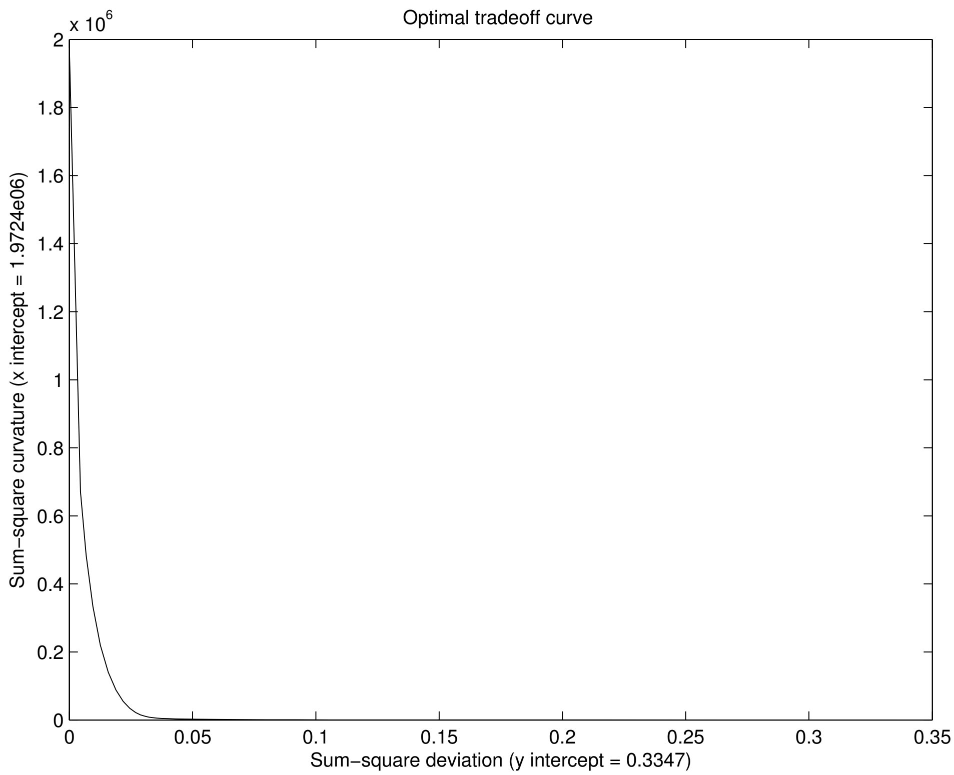
$$\left\| \frac{1}{\sqrt{n}} I g - \frac{1}{\sqrt{n}} f \right\|^2 + \mu \left\| \underbrace{\frac{n^2}{\sqrt{n-2}} \begin{bmatrix} -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 & 2 & -1 \end{bmatrix}}_{S} g \right\|^2.$$

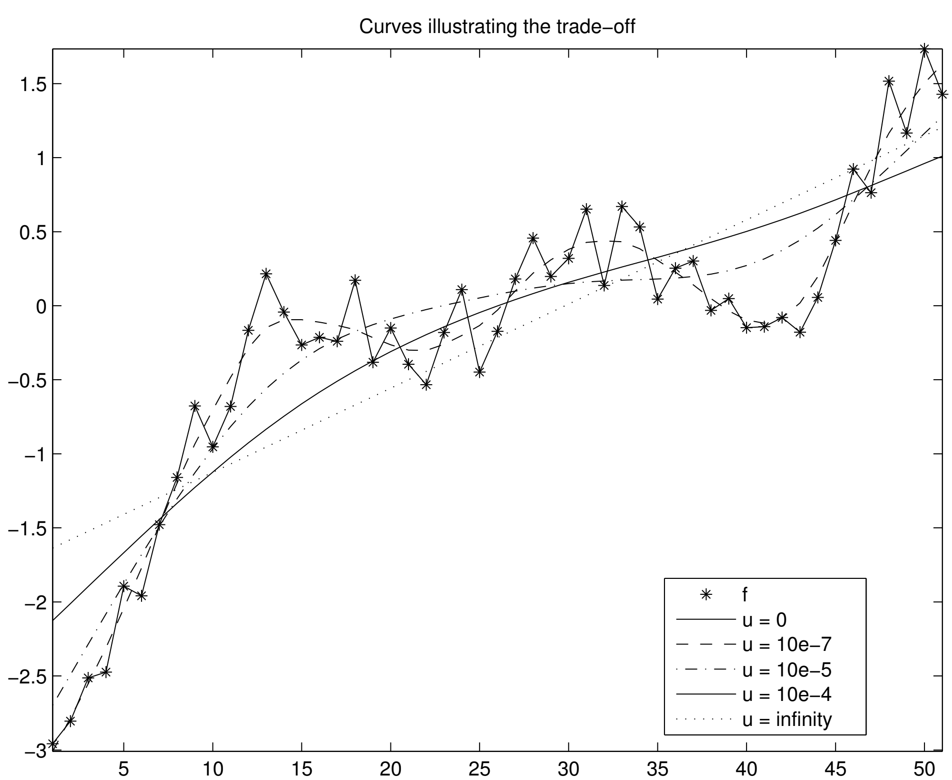
This is a multi-objective least-squares problem. The minimizing g is

$$g = (\tilde{A}^\mathsf{T} \tilde{A})^{-1} \tilde{A}^\mathsf{T} \tilde{y} \text{ where } \tilde{A} = \begin{bmatrix} \frac{I}{\sqrt{n}} \\ \sqrt{\mu} S \end{bmatrix} \text{ and } \tilde{y} = \begin{bmatrix} \frac{f}{\sqrt{n}} \\ 0 \end{bmatrix}.$$

The inverse of $\tilde{A}$ always always exists because I is full rank. The expression can also be written as $g = (\frac{I}{n} + \mu S^\mathsf{T} S)^{-1} \frac{f}{n}$.

b) The following plots show the optimal trade-off curve and the optimal g corresponding to representative μ values on the curve.





The following matlab code finds and plots the optimal trade-off curve between d and c. It also finds and plots the optimal g for representative values of μ. As expected, when $\mu = 0$, $g = f$ and no smoothing occurs. At the other extreme, as μ goes to infinity, we get an affine approximation of f. Intermediate values of μ correspond to approximations of f with different degrees of smoothness.

```
using LinearAlgebra
using Plots
using ToeplitzMatrices

include("readclassjson.jl")
data = readclassjson("curve_smoothing_data.json")
f = data["f"]
n = data["n"]

S = Toeplitz(vec([-1; zeros(n-3,1)]), vec([-1; 2; -1; zeros(n-3,1)]));
S = S*n^2/(sqrt(n-2));
I_n = 1*Matrix(I, n, n)
g_no_deviation = f;

error_curvature = []
error_deviation = []
append!(error_curvature, norm(S*g_no_deviation)^2)
append!(error_deviation, 0)

u = 10 .^(range(-8,stop=-3,length=30));


for i = eachindex(u)
    A_tilde = [1/sqrt(n)*I_n; sqrt(u[i])*S];
    y_tilde = [1/sqrt(n)*f; zeros(n-2,1)];
    g = A_tilde\y_tilde;
    append!(error_deviation, norm(1/sqrt(n)*I_n*g-f/sqrt(n))^2);
    append!(error_curvature, norm(S*g)^2);
end

a1 = collect(1:n);
a2 = ones(n,1);
A = [a1 a2];
affine_param = inv(A'*A)*A'*f;

g_no_curvature = []
for i = 1:n
    append!(g_no_curvature, affine_param[1]*i+affine_param[2])
end
```

```
g_no_curvature = g_no_curvature';
append!(error_deviation, 1/n*norm(vec(g_no_curvature)-f)^2);
append!(error_curvature, 0);

plot(error_deviation, error_curvature, label = "");
xlabel!("Sum-square deviation (y intercept = 0.3347)");
ylabel!("Sum-square curvature (x intercept = 1.9724e06)");
title!("Optimal tradeoff curve");
savefig("Optimal_tradeoff_curve.png")
u1 = 10e-7;
A_tilde = [1/sqrt(n)*I_n;sqrt(u1)*S];
y_tilde = [1/sqrt(n)*f;zeros(n-2,1)];
g1 = A_tilde\y_tilde;
u2 = 10e-5;
A_tilde = [1/sqrt(n)*I_n;sqrt(u2)*S];
y_tilde = [1/sqrt(n)*f;zeros(n-2,1)];
g2 = A_tilde\y_tilde;
u3 = 10e-4;
A_tilde = [1/sqrt(n)*I_n;sqrt(u3)*S];
y_tilde = [1/sqrt(n)*f;zeros(n-2,1)];
g3 = A_tilde\y_tilde;

scatter(f, label = "f", marker = (:star, 5), color = :black);
plot!(g_no_deviation, label = "mu = 0", line = (:solid, 2),);
plot!(g1, label = "mu = 10e-7", line = (:dash, 2),);
plot!(g2, label = "mu = 10e-5", line = (:dashdot, 2),);
plot!(g3, label = "mu = 10e-4", line = (:dashdotdot, 2),);
plot!(vec(g_no_curvature), label = "mu = inf", line = (:dot, 2),);
title!("Curves illustrating the trade-off");
savefig("Curves.png")
```

Note: Several exams had a typo that defined

$$c = \frac{1}{n-1} \sum_{i=2}^{n-1} \left(\frac{g_{i+1} - 2g_i + g_{i-1}}{1/n^2} \right)^2$$

instead of

$$c = \frac{1}{n-2} \sum_{i=2}^{n-1} \left(\frac{g_{i+1} - 2g_i + g_{i-1}}{1/n^2} \right)^2.$$

The solutions above reflect the second definition. Full credit was given for answers consistent with either definition. *Some common errors*

- Several students tried to approximate f using low-degree polynomials. While fitting f to a polynomial does smooth f, it does not necessarily minimize $d + \mu c$ for some $\mu \geq 0$, nor does it illustrate the trade-off between curvature and deviation.

- In explaining how to find the g that minimizes $d + \mu c$ as $\mu \to \infty$, many people correctly observed that if $g \in \text{null}(S)$, then $c = 0$. For full credit, however, solutions had to show how to choose the vector in null(S) that minimizes d.
- Many people chose to zoom in on a small section of the trade-off curve rather than plot the whole range from 0 to $\mu \to \infty$. Those solutions received full-credit provided they calculated the intersections with the axes (i.e. provided they found the minimum value for $d + \mu c$ when $\mu = 0$ and when $\mu \to \infty$).

8.1130. Modifying measurements to satisfy known conservation laws. A vector $y \in \mathbb{R}^n$ contains n measurements of some physical quantities $x \in \mathbb{R}^n$. The measurements are good, but not perfect, so we have $y \approx x$. From physical principles it is known that the quantities x must satisfy some linear equations, *i.e.*,

$$a_i^\mathsf{T} x = b_i, \qquad i = 1, \ldots, m,$$

where $m < n$. As a simple example, if x_1 is the current in a circuit flowing into a node, and x_2 and x_3 are the currents flowing out of the node, then we must have $x_1 = x_2 + x_3$. More generally, the linear equations might come from various conservation laws, or balance equations (mass, heat, energy, charge ...). The vectors a_i and the constants b_i are known, and we assume that $a_1, \ldots, a_m$ are independent. Due to measurement errors, the measurement y won't satisfy the conservation laws (*i.e.*, linear equations above) exactly, although we would expect $a_i^\mathsf{T} y \approx b_i$. An engineer proposes to adjust the measurements y by adding a correction term $c \in \mathbb{R}^n$, to get an adjusted estimate of x, given by

$$y_{\text{adj}} = y + c.$$

She proposes to find the smallest possible correction term (measured by $\|c\|$) such that the adjusted measurements y_{adj} satisfy the known conservation laws. Give an explicit formula for the correction term, in terms of y, a_i, b_i. If any matrix inverses appear in your formula, explain why the matrix to be inverted is nonsingular. Verify that the resulting adjusted measurement satisfies the conservation laws, *i.e.*, $a_i^\mathsf{T} y_{\text{adj}} = b_i$.

Solution. The correction c must satisfy the linear equations

$$a_i^\mathsf{T} y_{\text{adj}} = a_i^\mathsf{T} y + a_i^\mathsf{T} c = b_i, \quad i = 1, \ldots, m,$$

i.e.,

$$a_i^\mathsf{T} c = b_i - a_i^\mathsf{T} y \quad i = 1, \ldots, m.$$

We'll write that as $Ac = b - Ay$, where $A \in \mathbb{R}^{m \times n}$ has rows $a_1, \ldots, a_m$. We need to find the smallest c that satisfies these linear equations, *i.e.*, the least-norm solution of $Ac = b - Ay$. That is given by

$$c = A^\mathsf{T}(AA^\mathsf{T})^{-1}(b - Ay).$$

Here we take the inverse of AA^T, which is invertible since A is fat and full rank. Let's check that the conservation laws are satisfied, *i.e.*, $Ay_{\text{adj}} = b$:

$$\begin{aligned} Ay_{\text{adj}} &= A(y + c) \\ &= Ay + AA^\mathsf{T}(AA^\mathsf{T})^{-1}(b - Ay) \\ &= b \end{aligned}$$

as required.