Mathematics and Statistics for Data Analysis

Homework 3

Part 1: Geometric Foundations and Best Approximation

Instructions

You may discuss the problems with your peers, but your final write-up must be your own. Clearly state any assumptions you make. All mathematical expressions and proofs should be written clearly.

---

1. **Conceptual Summary: The "Story" of Geometric Spaces**
   The provided lecture notes build a "story" or hierarchy of mathematical spaces, starting from the most basic (Vector Spaces) and adding structure (Norms, Inner Products, Completeness) to reach the most powerful (Hilbert Spaces). This structure is then used to solve a fundamental problem (Best Approximation).

   In 2-3 paragraphs, summarize this narrative. Your summary must answer the following questions:

   - What specific properties (i.e., axioms) does each new space (Normed, Inner Product, Hilbert) add to the one before it?
   - Why is "completeness" (the jump from an Inner Product Space to a Hilbert Space) so critical when dealing with infinite-dimensional spaces like $L_2(\mathbb{R})$?
   - How does the final structure of a Hilbert Space provide a practical "toolkit" (specifically, the Orthogonality Principle) to solve the best approximation problem?

2. **Testing the Foundations: Subspace Axioms**
   A subset $\mathcal{T}$ of a vector space $\mathcal{S}$ is a subspace if it satisfies the closure property: for any $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{T}$ and any scalars $a, b \in \mathbb{F}$, the linear combination $a\boldsymbol{x} + b\boldsymbol{y}$ is also in $\mathcal{T}$.

   For each of the following, determine if the subset $\mathcal{T}$ is a subspace of the given vector space $\mathcal{S}$. Prove your answer by either showing that the closure property holds or by providing a specific counter-example.

   (a) **The Space:** $\mathcal{S} = \mathbb{R}^3$ (standard vector addition and scalar multiplication).
   The Subset: $\mathcal{T} = \{\boldsymbol{x} \in \mathbb{R}^3 : x_1 - 2x_2 + x_3 = 1\}$.

   (b) **The Space:** $\mathcal{S} = C[a, b]$ (the space of all bounded, continuous functions on the interval $[a, b]$).
   The Subset: $\mathcal{T} = \{f(t) \in \mathcal{S} : f(t) \geq 0 \text{ for all } t \in [a, b]\}$.

(c) **The Space:** $\mathcal{S} = L_2[a, b]$ (the space of square-integrable functions on $[a, b]$).
The Subset: $\mathcal{T} = \left\{ f(t) \in \mathcal{S} : \int_a^b f(t)dt = 0 \right\}$ (The set of functions with zero mean).

3. **Proving Norms and Non-Norms**
This problem tests your understanding of the three fundamental axioms of a norm:

- **Non-negativity & Definiteness:** $\|\boldsymbol{x}\| \geq 0$ and $\|\boldsymbol{x}\| = 0 \iff \boldsymbol{x} = \mathbf{0}$
- **Homogeneity:** $\|a\boldsymbol{x}\| = |a| \cdot \|\boldsymbol{x}\|$
- **Triangle Inequality:** $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$

(a) **The Induced Norm:** Prove that if $\langle \cdot, \cdot \rangle$ is a valid inner product, then $\|\boldsymbol{x}\| := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$ is a valid norm. (*Hint: You may use the Cauchy-Schwarz Inequality,* $|\langle \boldsymbol{x}, \boldsymbol{y} \rangle| \leq \|\boldsymbol{x}\| \cdot \|\boldsymbol{y}\|$*, without proof.*)

(b) **A Counter-Example:** The function $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_2^2 = \sum_{n=1}^{N} |x_n|^2$ is **not** a valid norm. Which of the three axioms does it fail? Provide a simple numerical counter-example in $\mathbb{R}^2$ to prove your claim.

(c) **The Reverse Triangle Inequality:** Prove that for any valid norm, the following inequality holds for all vectors $\boldsymbol{x}, \boldsymbol{y}$:

$$\|\boldsymbol{x}\| - \|\boldsymbol{y}\| \leq \|\boldsymbol{x} - \boldsymbol{y}\|$$

(*Hint: Start with* $\|\boldsymbol{x}\| = \|(\boldsymbol{x} - \boldsymbol{y}) + \boldsymbol{y}\|$ *and apply the triangle inequality.*)

4. **The Parallelogram Law: Linking Norms and Inner Products**
A norm is induced by an inner product if and only if it satisfies the **Parallelogram Law**.

(a) **Proof:** Using only the axioms of an inner product and $\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$, prove the Parallelogram Law:

$$\|\boldsymbol{x} + \boldsymbol{y}\|^2 + \|\boldsymbol{x} - \boldsymbol{y}\|^2 = 2\|\boldsymbol{x}\|^2 + 2\|\boldsymbol{y}\|^2$$

(b) **Application:** Use the Parallelogram Law to prove that the l_∞ norm ($\|\boldsymbol{x}\|_\infty = \max_n |x_n|$) in $\mathbb{R}^N$ (for $N \geq 2$) is **not** induced by any inner product. (*Hint: You must find two specific vectors* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$ *that violate the law.*)

(c) **Weighted Inner Products:** Consider $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathbf{Q}} = \boldsymbol{x}^T \mathbf{Q} \boldsymbol{y}$ for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$. What properties must the matrix $\mathbf{Q} \in \mathbb{R}^{N \times N}$ have for this to be a valid inner product? Justify your answer by relating the required properties to the inner product axioms.

5. **Best Approximation in a Function Space**
Let our Hilbert Space be $\mathcal{S} = L_2[0, 1]$ with the inner product $\langle f, g \rangle = \int_0^1 f(t)g(t)dt$.

We want to find the best approximation of $f(t) = t^3$ within the 2-dimensional subspace $\mathcal{T} = \text{span}\{v_1(t), v_2(t)\}$, where $v_1(t) = 1$ and $v_2(t) = t$. The best approximation is $\hat{f}(t) = a_1 v_1(t) + a_2 v_2(t) = a_1 + a_2 t$.

(a) **Set up the Normal Equations:** Write down the two integral equations given by the Orthogonality Principle: $\left\langle f - \hat{f}, v_1 \right\rangle = 0$ and $\left\langle f - \hat{f}, v_2 \right\rangle = 0$.

(b) **Compute G and $\boldsymbol{b}$:** Show that the equations from (a) can be written as $\mathbf{G}\boldsymbol{a} = \boldsymbol{b}$. Compute all entries of the 2×2 Gram Matrix $\mathbf{G}$ ($G_{ij} = \langle v_j, v_i \rangle$) and the 2×1 vector $\boldsymbol{b}$ ($b_i = \langle f, v_i \rangle$).

(c) **Solve for Coefficients:** Solve $\mathbf{G}\boldsymbol{a} = \boldsymbol{b}$ to find a_1 and a_2.

(d) **Calculate the Error:** What is the final $\hat{f}(t)$? Calculate the energy of the approximation error, $\left\| f - \hat{f} \right\|_2^2$.

6. **Stability, Computation, and the Dual Basis**
This problem synthesizes concepts from the second half of the lecture.

(a) **Stability:** The stability inequality is $A \cdot \|\boldsymbol{\alpha}\|_2^2 \leq \|x\|^2 \leq B \cdot \|\boldsymbol{\alpha}\|_2^2$. Explain how a *tiny* smallest eigenvalue A of the Gram matrix $\mathbf{G}$ leads to the "unstable" behavior shown in the "Nothing Polynomial" example.

(b) **The Dual Basis Connection:** For an **orthonormal basis** $\{v_n\}$, $\hat{f} = \sum \langle f, v_n \rangle v_n$. For a **general basis**, coefficients are computed using the dual basis $\{\tilde{v}_n\}$ as $a_n = \langle f, \tilde{v}_n \rangle$. Prove that for an orthonormal basis, these two formulas are identical (i.e., show $v_n = \tilde{v}_n$).

(c) **Synthesis:** Explain mathematically *why* an ill-conditioned Gram matrix $\mathbf{G}$ (with a tiny eigenvalue A) *must* produce a dual basis $\{\tilde{v}_n\}$ with large norms. (*Hint:* $\tilde{v}_n = \sum_l H_{n,l} v_l$ *where* $\mathbf{H} = \mathbf{G}^{-1}$. *What is the relationship between the eigenvalues of* $\mathbf{G}$ *and* $\mathbf{H}$*?*)

Part 2: EVD, SVD, and Applications

Instructions

This assignment focuses on the Eigenvalue and Singular Value Decompositions (EVD/SVD) and their applications in solving linear systems, regularization, and data analysis.

1. **EVD vs. SVD: Foundations and Geometry**

   (a) **EVD for Symmetric Matrices:** For a real, symmetric $\mathbf{A} \in \mathbb{R}^{N\times N}$, the EVD is $\mathbf{A} = \mathbf{V}\Lambda\mathbf{V}^T$. What are the specific properties of $\mathbf{V}$ and Λ? What does this decomposition tell you about the action of $\mathbf{A}$ in the coordinate system defined by the columns of $\mathbf{V}$?

   (b) **The SVD for General Matrices:** For *any* $\mathbf{A} \in \mathbb{R}^{M\times N}$, the SVD is $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$. Explain the geometric interpretation of this as a three-step process (rotation, scaling, rotation) by defining $\mathbf{U}$, Σ, and $\mathbf{V}$ and their roles.

   (c) **Connecting EVD and SVD:** How are the right singular vectors $\mathbf{V}$ and singular values Σ of $\mathbf{A}$ related to the EVD of $\mathbf{A}^T\mathbf{A}$? How are the left singular vectors $\mathbf{U}$ related to the EVD of $\mathbf{A}\mathbf{A}^T$?

   (d) **When SVD = EVD:** Under what specific conditions on a matrix $\mathbf{A}$ will its SVD ($\mathbf{U}\Sigma\mathbf{V}^T$) be identical to its EVD ($\mathbf{V}\Lambda\mathbf{V}^T$)? (This requires $\mathbf{U} = \mathbf{V}$ and $\Sigma = \Lambda$).

2. **The Least-Squares Problem and the Pseudoinverse**
   The least-squares problem seeks to find $\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\|_2^2$.

   (a) **Deriving the Solution:** Starting from $\boldsymbol{x} = \mathbf{V}\boldsymbol{\alpha}+\mathbf{V}_0\boldsymbol{\alpha}_0$ and $\boldsymbol{y} = \mathbf{U}\boldsymbol{\beta}+\mathbf{U}_0\boldsymbol{\beta}_0$, derive the full Moore-Penrose pseudoinverse solution $\hat{\boldsymbol{x}} = \mathbf{A}^\dagger\boldsymbol{y} = \mathbf{V}\Sigma^{-1}\mathbf{U}^T\boldsymbol{y}$. Show your work in minimizing the residual $\|\boldsymbol{r}\|_2^2$ and finding the minimum-norm solution.

   (b) **Properties of the Solution:** The notes state this $\hat{\boldsymbol{x}} = \mathbf{A}^\dagger\boldsymbol{y}$ solution is optimal because it satisfies three conditions (one of which is minimizing the norm). List these three conditions.

   (c) **Relating Special Cases:** Prove that for a "tall" matrix $\mathbf{A}$ (full column rank $R = N < M$), the general SVD formula $\mathbf{A}^\dagger = \mathbf{V}\Sigma^{-1}\mathbf{U}^T$ simplifies to the well-known formula $\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$.

3. **Instability and Regularization**

   (a) **The Instability Problem:** When solving $\hat{\boldsymbol{x}}_{ls} = \mathbf{A}^\dagger\boldsymbol{y}$ with noisy data $\boldsymbol{y} = \mathbf{A}\boldsymbol{x}_0 + \boldsymbol{e}$, derive the formula for the expected reconstruction error $\mathbb{E}[\left\|\mathbf{A}^\dagger\boldsymbol{e}\right\|_2^2]$ assuming white noise. Explain in one sentence why small singular values σ_r are "dangerous."

   (b) **Tikhonov Regularization:** The Tikhonov solution is $\hat{\boldsymbol{x}}_{\text{tik}} = \sum_{r=1}^{R} \frac{\sigma_r}{\sigma_r^2+\delta} \langle \boldsymbol{y}, \boldsymbol{u}_r \rangle \boldsymbol{v}_r$. Decompose the total error $\hat{\boldsymbol{x}}_{\text{tik}} - \boldsymbol{x}_0$ into its two components: **Noise Error** and **Approximation Error (Bias)**.

(c) **Truncated SVD vs. Tikhonov:** Compare the error terms from (b) to the error terms for Truncated SVD. How does the "damped multiplier" of Tikhonov differ from the "hard threshold" of Truncated SVD? Which singular values (r) contribute to the approximation bias in each method?

4. **Total Least Squares (TLS)**

   (a) **A Different Error Model:** What is the fundamental assumption that Ordinary Least Squares (LS) makes about the error? How does the Total Least Squares (TLS) error model differ?

   (b) **The SVD Solution to TLS:** The TLS problem is solved by finding the best rank-N approximation of the *augmented matrix* $\mathbf{C} = [\mathbf{A}|\boldsymbol{y}]$. Explain *why* the solution $\hat{\boldsymbol{x}}_{TLS}$ is constructed from $\boldsymbol{z}_{N+1}$, the right singular vector of $\mathbf{C}$ corresponding to the *smallest* singular value γ_{N+1}.

5. **PCA and Low-Rank Approximation**

   (a) **The "Grand Connection":** Both PCA and TLS are low-rank approximation problems.

      (a) In PCA, what matrix are we finding the best rank-K approximation of?

      (b) In TLS, what matrix are we finding the best rank-N approximation of?

   (b) **PCA: SVD vs. Covariance:** Prove that finding the eigenvectors of the covariance matrix $\mathbf{S} = \frac{1}{N}\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$ is equivalent to finding the *left singular vectors* $\mathbf{U}$ of the centered data matrix $\tilde{\mathbf{X}}$.

6. **The Grammian Operator and Riesz Bases**

   (a) **The Gram Matrix:** Given a basis $\{v_1, \ldots, v_N\}$, the Gram matrix $\mathbf{G}$ is $\mathbf{G}_{n,m} = \langle v_n, v_m \rangle$. What is the relationship between the *invertibility* of $\mathbf{G}$ and the *linear independence* of the basis vectors $\{v_n\}$?

   (b) **Stability and the Riesz Basis:** For an infinite basis $\{v_n\}_{n=1}^{\infty}$, the **Riesz Basis Condition** is:

$$A\sum_{n=1}^{\infty}|\alpha_n|^2 \leq \left\|\sum_{n=1}^{\infty}\alpha_n v_n\right\|^2 \leq B\sum_{n=1}^{\infty}|\alpha_n|^2 \quad (\text{for } 0 < A \leq B < \infty)$$

   Why is this condition much stronger than simple linear independence when $N \to \infty$? What specific problem, discussed in the notes, does it prevent?

Part 3:Computational Applications (Coding)

13. **Gram Matrix, Least Squares, and Stability via Truncated SVD (Code:** `hw_code_monomial.py`**)**
Goal: Approximate $f(t) = e^t$ on $[0, 1]$ using the monomial basis $\{1, t, t^2\}$. You will construct the least squares problem explicitly, analyze its conditioning, and stabilize it using truncated SVD.

(a) **Build the discrete least squares system:** Sample $f(t) = e^t$ at $N = 50$ uniformly spaced points in $[0, 1]$. Construct the design matrix $A \in \mathbb{R}^{N\times 3}$ where $A_{i,:} = [1, t_i, t_i^2]$. Compute the Gram matrix $= A^\top A$ and the right-hand side $\mathbf{b} = A^\top \mathbf{y}$, where $\mathbf{y} = [f(t_1), \ldots, f(t_N)]^\top$. Solve $\mathbf{a} = \mathbf{b}$ for $\mathbf{a}_{\text{LS}}$ and compute $\kappa()$.

(b) **Implement truncated SVD regularization:** Compute the SVD of $A = U\Sigma V^\top$. Manually construct the truncated pseudo-inverse $A^\dagger_{R'}$ using only the $R' = 2$ largest singular values (set others to zero). Compute $\mathbf{a}_{\text{TSVD}} = A^\dagger_{R'}\mathbf{y}$.

(c) **Analyze and compare:** Compute $\|\mathbf{a}_{\text{LS}}\|_2$ and $\|\mathbf{a}_{\text{TSVD}}\|_2$. Plot both polynomial approximations against $f(t)$. In a short written response (in code comments or separate), explain why discarding the smallest singular value improves stability.

(d) **Error Analysis:** Compute the residual norm $\|A\mathbf{a}_{\text{LS}} - \mathbf{y}\|_2$ and $\|A\mathbf{a}_{\text{TSVD}} - \mathbf{y}\|_2$. Additionally, compute the residual for truncation levels $R' = 1, 2, 3$ and plot residual norm vs. R'. Explain the trade-off between stability and accuracy.

14. **When Does Total Least Squares Help? (Code:** `hw_code_tls.py`**)**
Goal: Investigate the performance of OLS and TLS under different noise models. You will run two experiments and analyze when TLS is beneficial.

Experiment 1 Only the output y is corrupted by noise (input x is clean).

(a) Generate clean x, compute clean $y = \theta_{\text{true}}x$, then add noise only to y.

(b) Compute OLS and TLS estimates.

(c) Compare $|\hat{\theta}_{\text{OLS}} - \theta_{\text{true}}|$ and $|\hat{\theta}_{\text{TLS}} - \theta_{\text{true}}|$.

Experiment 2 Both input x and output y are corrupted by noise of equal magnitude.

(d) Generate clean (x, y), then add independent noise to both.

(e) Compute OLS and TLS estimates.

(f) Compare parameter errors again.

(g) **Analysis:** Based on your results, explain:

- Why TLS performs worse in Experiment 1,
- Why TLS performs better in Experiment 2,
- What assumption OLS makes that is violated in Experiment 2.

15. **Understanding PCA Step-by-Step: Geometry, Decorrelation, and Compression (Code:** `hw_code_pca.py`**)**
Goal: Use a small 16×16 image to explore PCA through three complementary perspectives. For each step below, your code must **print numerical results** and **generate a plot**.

(a) **Geometric Transformation (Rotation to Principal Axes)**
Center the image data and compute its SVD: $\tilde{X} = U\Sigma V^\top$. The matrix V defines a rotation. Project the data: $Z = \tilde{X}V$.
Print:

- Shape of X, $\tilde{X}$, and Z,
- First principal component vector v_1 (first column of V). **Plot:** Original image and the rotated data matrix Z (as an image).

(b) **Statistical Decorrelation**
Compute the covariance matrix of the original centered data ($C_{\text{orig}} = \frac{1}{n}\tilde{X}^\top \tilde{X}$) and of the rotated data ($C_{\text{pca}} = \frac{1}{n}Z^\top Z$).
Print:

- Frobenius norm of off-diagonal entries of C_{orig},
- Frobenius norm of off-diagonal entries of C_{pca}. **Plot:** Heatmaps of both covariance matrices side by side.

(c) **Dimensionality Reduction (Compression)**
Reconstruct the image using only the top $k = 3$ principal components.
Print:

- Reconstruction error $\|X - \hat{X}\|_F$,
- Original size (256 numbers) vs. PCA storage cost ($17k + 16 = 67$ for $k = 3$),
- Compression ratio. **Plot:** Original image, reconstructed image ($k = 3$), and cumulative variance explained curve.

(d) **PCA via Covariance Method vs. SVD Method**
Implement PCA using the classical covariance method:

(a) Compute the covariance matrix $C = \frac{1}{n}\tilde{X}^\top \tilde{X}$,
(b) Perform eigendecomposition $C = V_{\text{cov}} \Lambda V_{\text{cov}}^\top$,
(c) Use the top $k = 3$ eigenvectors to reconstruct the image.

Print:

- Reconstruction error from the covariance method,
- Difference in reconstruction between SVD and covariance methods: $\|\hat{X}_{\text{SVD}} - \hat{X}_{\text{cov}}\|_F$. **Analysis:** Explain why the SVD method is numerically superior, especially when the data matrix is wide (more features than samples) or ill-conditioned.

16. **Gramian Approach to Kernel Regression (Code:** `hw_code_gramian_kernel.py`**)**
Goal: Use the Gram Matrix (Kernel Matrix) $\mathbf{K}$ to solve a regularized regression problem (Ridge Regression) in a lifted feature space.

(a) **Compute Gramian:** The starter code provides the feature matrix $\mathbf{\Phi}$. Compute the Gramian (or Kernel Matrix) $\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^T$.

(b) **Solve Dual Problem:** Implement the solution for the dual coefficients $\boldsymbol{\alpha}$: $\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1}\boldsymbol{y}$.

(c) **Prediction and Comparison:** Calculate the prediction $\hat{\boldsymbol{y}}_{\text{Gramian}} = \mathbf{K}\boldsymbol{\alpha}$. The script will compare this to the OLS/Primal solution. *Report the MSE for the Gramian prediction.*

17. **Kernel Ridge Regression via Gram Matrix (Code:** `hw_code_gramian_kernel.py`**)**
Goal: Solve a regression problem in a high-dimensional (or infinite-dimensional) feature space using the kernel trick, without explicitly computing feature vectors.

(a) **Kernel Matrix Construction:** Define a radial basis function (RBF) kernel $k(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$. Compute the Gram (kernel) matrix $\mathbf{K} \in \mathbb{R}^{M \times M}$ where $K_{ij} = k(x_i, x_j)$.

(b) **Dual Solution:** Solve for the dual coefficients $\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \boldsymbol{y}$.

(c) **Prediction and Analysis:** Compute predictions $\hat{y}_i = \sum_{j=1}^{M} \alpha_j k(x_j, x_i)$. Compare the kernel solution to an explicit polynomial regression (limited to low-degree features). Report MSE for both.
Explain: Why is the kernel method more powerful when the true relationship is highly non-linear?

18. **Regression via Explicit Features and Kernel Gram Matrix (Code:** `hw_code_gramian_kern`
Goal: Compare multiple regression approaches on a non-linear dataset: (1) simple linear OLS, (2) polynomial regression, (3) ridge regression using explicit features via Gram matrix, and (4) kernel ridge regression using an RBF kernel.

(a) **Data and Baselines:** Generate noisy observations from $y = 2\sin(2x) + 0.5x^3$. Fit:

- A linear OLS model using features $\phi_{\text{lin}}(x) = [1, x]$,
- A degree-5 polynomial model using $\phi_{\text{poly}}(x) = [1, x, \ldots, x^5]$.

(b) **Gramian Ridge (Explicit Features):** Using the same linear features as OLS, compute the Gram matrix $K = \Phi_{\text{lin}} \Phi_{\text{lin}}^{\top}$ and solve the dual problem $\alpha = (K + \lambda I)^{-1} y$. Predict using $\hat{y} = K\alpha$.

(c) **Kernel Ridge Regression (RBF):** Define the RBF kernel $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$. Compute the kernel Gram matrix, solve for α, and predict.

(d) **Analysis:** Compute MSE for all methods against the true function. Plot all predictions. Explain:

- Why OLS and Gramian-Linear give identical results,
- Why the RBF kernel outperforms polynomial regression,
- When the Gram matrix approach is essential (e.g., infinite-dimensional features).