NOTE: Choose any 2 - one on kernels and one on random matrices/projection

1. **Kernels Versus Kernels.** Generate \( n = 400 \) data points \((X_1, Y_1), \ldots, (X_n, Y_n)\) as follows. Take \( X_1, \ldots, X_n \sim \text{Uniform}(-1,1) \). Take \( Y_i = m(X_i) + \sigma(X_i) \epsilon_i \) where \( \epsilon_1, \ldots, \epsilon_n \sim \mathcal{N}(0,1) \),

\[
m(x) = \begin{cases} 
\frac{(x + 2)^2}{2} & -1 \leq x < -0.5 \\
x/2 + 0.875 & -0.5 \leq x < 0 \\
-5(x - 0.2)^2 + 1.075 & 0 \leq x < 0.5 \\
x + 0.125 & 0.5 \leq x < 1 
\end{cases}
\]

and

\[
\sigma(x) = 0.2 - 0.1 \cos(2\pi x).
\]

Randomly split the data into two sets of \( n = 200 \) observations each. The first half is the training data and the second is the testing data.

(a) Estimate \( m \) using kernel regression. Use a Gaussian kernel. Choose the bandwidth by cross-validation (using the test data). Plot the true function, the data and the estimated function. Plot the residuals. Plot the cross-validation function as a function of \( h \).

(b) Now estimate \( m \) using RKHS methods. Specifically, choose \( \hat{m} \) to minimize

\[
\sum_{i=1}^{n} (Y_i - m(X_i))^2 + \lambda \|m\|^2_K
\]

where the kernel \( K \) is \( K(x,y) = e^{-(x-y)^2/\sigma^2} \). There are two tuning parameters, \( \lambda \) and \( \sigma \). Choose both by cross-validation (using the test data). Make the same plots as in (a). Comment on the differences/similarities between the two estimates.

2. **RKHS.** Let \( \mathcal{F} \) denotes all real-valued functions on \([0,1]\) with \( m \) continuous derivatives. Define the kernel

\[
K(x,y) = \sum_{s=0}^{m-1} \frac{x^s y^s}{s! s!} + \int_0^1 \frac{(x-u)^{m-1} (y-u)^{m-1}}{(m-1)! (m-1)!} du
\]

and inner product

\[
\langle f, g \rangle = \sum_{s=0}^{m-1} f^{(s)}(0) g^{(s)}(0) + \int_0^1 f^{(m)}(x) g^{(m)}(x) dx.
\]

Verify that this kernel has the reproducing property: \( \langle K_x, f \rangle = f(x) \). 

Hint: By Taylor’s theorem with remainder, we can write

\[
f(x) = \sum_{s=0}^{m-1} \frac{x^s}{s!} f^{(s)}(0) + \int_0^1 \frac{(x-u)^{(m-1)}}{(m-1)!} f^{(m)}(u) du.
\]
3. **Random Matrices.** Refer to the notes on random matrices.

(a) Prove Lemma 1.

(b) The notes contain a proof sketch for Theorem 4. Fill in the missing details and provide a complete proof.

4. **Low Rank Approximation via Random Projections.** A low rank approximation of an \( m \times n \) matrix \( A \) is another matrix \( A_k \) such that 1) The rank of \( A_k \) is at most \( k \) and 2) \( \|A - A_k\| \) is minimized for some norm. It is well known that for the Frobenius norm \( (\|A\|_F = \sqrt{\sum_{ij} A_{ij}^2}) \), we have \( A_k = \sum_{i=1}^k \sigma_i u_i v_i^T \) where the singular value decomposition (SVD) of \( A \) is \( A = \sum_{i=1}^n \sigma_i u_i v_i^T \). However, the complexity of computing the SVD is \( O(mn^2) \).

We consider an alternate method based on random projections that is much faster. The algorithm is as follows:

1. Let \( R \) be an \( m \times \ell \) matrix such that \( R_{ij} \) are drawn i.i.d from \( N(0,1) \). Also we have that \( \ell \geq c(\log n)/\epsilon^2 \) for some constant \( c > 0 \). Compute \( B = \frac{1}{\sqrt{\ell}} R^T A \).

2. Compute the SVD of \( B \), \( B = \sum_{i=1}^\ell \lambda_i a_i b_i^T \).

3. Return: \( \tilde{A}_k = A \cdot \sum_{i=1}^k b_i b_i^T \).

- Show that with high probability
  \[ \|A - \tilde{A}_k\|_F^2 \leq \|A - A_k\|_F^2 + 2\epsilon\|A_k\|_F^2 \]

The following form of the JL Lemma will be useful: A set of \( n \) vectors \( x_1, \ldots, x_n \) in \( \mathbb{R}^m \) can be projected down to \( R^T x_1, \ldots, R^T x_n \) in \( \mathbb{R}^\ell \) with high probability using a \( m \times \ell \) random matrix \( R \) with i.i.d \( N(0,1) \) entries such that

\[ (1 - \epsilon)\|x_i\|^2 \leq \|R^T x_i\|^2 \leq (1 + \epsilon)\|x_i\|^2 \]

for \( i = 1, \ldots, n \) provided \( \ell \geq c(\log n)/\epsilon^2 \) for some constant \( c > 0 \).

- What is the computational complexity of this procedure?