NYU | COURANT

MATH-GA.2711 Machine Learning & Computational Statistics

Homework 2

Profs. Ivailo Dimov
Due: February 8, 2026

Instruction

This homework is to be done individually. No collaboration and/or code sharing permitted.

Questions

1. Implementing and Pipelining OLS in Sklearn
   (a) Using numpy, generate a matrix $\mathbf{X} \in \mathbb{R}^{T\times N}$ whose entries are $\mathcal{N}(0,1)$.
   (b) Write a method linmodel() that outputs the target $\mathbf{y}$ defined by $\mathbf{y} := \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ where $\mathbf{X} \in \mathbb{R}^{T\times N}$, $\boldsymbol{\beta} \in \mathbb{R}^N$ and $\sigma \in \mathbb{R}_+$ are the inputs.
   (c) Let us assume $T > N$. Write a method fit() that solves the linear system $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}$ using the normal equations, returning the solution $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$.
   (d) Let $T = 1000$ and $N = 3$. Using nested for loops, test the performance of your fit() method by instantiating your class with a linear model whose parameters are within the ranges $\sigma = 0.01, 0.1, 1$. To generate your $\mathbf{y}$, use $\boldsymbol{\beta} := [0.01, 0.1, 1]$. What is the $\hat{\boldsymbol{\beta}}$ that you obtain by running the fit() function?
   (e) How noisy is your estimator as a function of the sample noise σ? Explain.
   (f) If you were to simulate the data K times, each time you would get a different estimate of $\hat{\boldsymbol{\beta}}$. Suppose $K = 100$. What is standard deviation of the distribution of $\hat{\beta}_i$ $(i = 1, 2, 3)$ of the K estimates?
   (g) Define $\hat{\mathbf{y}} := \mathbf{X}\hat{\boldsymbol{\beta}}$. (i) Create a scatter plot of $\mathbf{y}$ and $\hat{\mathbf{y}}$ when $\sigma = 0.1$. (ii) What happens to the scatter plot for various values of σ? Explain.
   (h) Extra credit: make the scatter plot in (g) interactive as a function of $\sigma \in [0, 1]$ and $\beta_i \in [0, 1]$ $(i = 1, 2, 3)$.

2. OLS geometry: projection and residual orthogonality.
   Let $\mathbf{X} \in \mathbb{R}^{n\times p}$ have full column rank. Define the *projection* and *annihilator* matrices

$$\mathbf{P} := \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}', \qquad \mathbf{M} := \mathbf{I}_n - \mathbf{P}.$$

   Let $\widehat{\boldsymbol{\beta}} := (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$, $\widehat{\mathbf{y}} := \mathbf{X}\widehat{\boldsymbol{\beta}}$, and $\mathbf{r} := \mathbf{y} - \widehat{\mathbf{y}}$.

(a) Prove that $\mathbf{P}$ and $\mathbf{M}$ are each symmetric and idempotent. Show that the OLS residual orthogonal to the column space of $\mathbf{X}$:

$$\mathbf{X}'\mathbf{r} = 0.$$

Do you expect orthogonality to hold out of sample? Justify.

(b) Using only NumPy, write code that:

1. constructs $\mathbf{P}$ and $\mathbf{M}$ *without* explicitly computing $(\mathbf{X}'\mathbf{X})^{-1}$ (use `np.linalg.solve`),
2. verifies numerically (up to tolerance) that $\mathbf{P} \approx \mathbf{P}'$, $\mathbf{P}^2 \approx \mathbf{P}$, $\mathbf{P}^2 \approx \mathbf{P}$,
3. verifies $\mathbf{P}'(\mathbf{M}\mathbf{y}) \approx 0$ for a random $\mathbf{y} \in \mathbf{R}^n$,
4. verifies the decomposition $\mathbf{y} \approx \mathbf{P}\mathbf{y} + \mathbf{M}\mathbf{y}$.

Report the maximum absolute deviation in each check.

3. Unbiasedness of s^2 and degrees of freedom.
Consider the classical linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \qquad \mathbb{E}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \mathbf{0}, \qquad \mathbb{V}[\boldsymbol{\varepsilon} \mid \mathbf{X}] = \sigma^2 \mathbf{I}_n.$$

Let $\widehat{\boldsymbol{\beta}}$ be the OLS estimator, $\mathbf{r} := \mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}$ be the residual vector, and define

$$s^2 := \frac{\mathbf{r}'\mathbf{r}}{n-p}.$$

(a) Show that

$$\mathbf{r} = \mathbf{M}\boldsymbol{\varepsilon} \quad \text{and} \quad \mathbf{r}'\mathbf{r} = \boldsymbol{\varepsilon}'\mathbf{M}\boldsymbol{\varepsilon}.$$

(b) Show that

$$\mathbb{E}[\boldsymbol{\varepsilon}'\mathbf{M}\boldsymbol{\varepsilon} \mid \mathbf{X}] = \sigma^2 \operatorname{Tr}(\mathbf{M}) \quad \text{and} \quad \mathbb{E}[s^2 \mid \mathbf{X}] = \sigma^2.$$

(c) Fix n, p, draw a full-rank $\mathbf{X}$, fix $\boldsymbol{\beta}$ and σ^2, and run a Monte Carlo experiment:

1. simulate many $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n)$ and compute $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$,
2. compute $\widehat{\boldsymbol{\beta}}$ and s^2 each time,
3. report the empirical mean of s^2 and compare it to σ^2,
4. repeat with the *incorrect* estimator $\widetilde{s}^2 := \frac{\mathbf{r}'\mathbf{r}}{n}$ and compare the bias.

4. Chi-square quadratic forms from idempotent matrices.
State precisely the result relating the quadratic form $\mathbf{z}^\top \mathbf{A}\mathbf{z}$ to a chi-square distribution, including its degrees of freedom

(a) For a random full-rank $\mathbf{X} \in \mathbb{R}^{n \times p}$, construct

$$\mathbf{P} := \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}', \qquad \mathbf{M} := \mathbf{I}_n - \mathbf{P}.$$

Verify (numerically) that $\mathbf{M}$ is symmetric and idempotent. Compute and report $\operatorname{Tr}(\mathbf{M})$.

(b) Draw B i.i.d. samples $\mathbf{z}^{(b)} \sim \mathcal{N}(0, \mathbf{I}_n)$ and compute

$$q^{(b)} := (\mathbf{z}^{(b)})' \mathbf{M} \mathbf{z}^{(b)}.$$

Using the simulated $\{q^{(b)}\}_{b=1}^{B}$, estimate the empirical mean and variance of q. Compare them to the chi-square benchmarks:

$$\mathbb{E}[q] = \mathrm{Tr}(\mathbf{M}), \qquad \mathbb{V}[q] = 2\ \mathrm{Tr}(\mathbf{M}).$$

Report absolute and relative errors.