CS 215 — Assignment 3

Aradhana R (24b1006) Suhas Alladaboina (24b1009)

Question 1

Terminology and Setup.

- $\Omega \subset Z^2$: the pixel grid of the image (set of all integer-coordinate pixel positions).
- $I : \Omega \to R$: the unknown, noise-free grayscale image. Each pixel (x, y) has intensity $I(x, y)$.
- N: number of independent frames captured in burst mode.
- $J_k(x, y)$: the observed intensity at pixel (x, y) in frame k, modeled as

  $$J_k(x, y) = I(x, y) + W_k(x, y), \qquad k = 1, \ldots, N,$$

  where $W_k(x, y)$ are i.i.d. noise terms.
- $W_k(x, y)$: random noise, independent across all pixels and frames, with $E[W_k(x, y)] = 0$.
- **Checkerboard model:** the image is divided into M disjoint square regions (tiles) $R_1, \ldots, R_M \subset \Omega$, with constant intensity per tile:

  $$I(x, y) = \alpha_m, \quad \forall (x, y) \in R_m,$$

  where α_m is the true constant intensity of tile R_m.
- **Hat notation:** A symbol with a hat (e.g. $\widehat{I}$, $\widehat{\alpha}_m$, $\widehat{W}$) denotes an *estimate* of the corresponding true quantity (I, α_m, W), obtained from the observed data $\{J_k(x, y)\}$.

Goal. Estimate the noise distribution from the data $\{J_k(x, y)\}$.

Step 1 — Excluding edge pixels

We remove the border pixels before analysis, since their intensities can lie between two neighboring regions (tiles) and may not represent a pure constant value.

Formally, define

$$\Omega_{\text{edge}} \;=\; \{(x, y) \in \Omega : \exists (x', y') \text{ neighbor of } (x, y) \text{ with } |I(x, y) - I(x', y')| > threshold\}$$

and use only the interior set

$$\Omega_b \;=\; \Omega \setminus \Omega_{\text{edge}}.$$

All subsequent computations (averaging, residuals, histograms) use only the interior pixels. Wherever tiling is used, each tile R_m is used to represent the interior subset after removing borders: $R'_m = R_m \cap \Omega_b$

Step 2 — Estimating the noise-free image

For each pixel in Ω_b, average across the N frames:

$$\widehat{I}(x, y) \;=\; \frac{1}{N} \sum_{k=1}^{N} J_k(x, y), \qquad (x, y) \in \Omega_b$$

which converges to $I(x, y)$ as $N \to \infty$ by SLLN.

Variance reduction via tiles. Because I is constant on each tile, one can further reduce variance by spatial averaging within tiles (restricted to Ω_b):

$$\widehat{\alpha}_m \;=\; \frac{1}{N\,|R'_m|}\sum_{(x,y)\in R'_m}\sum_{k=1}^{N} J_k(x,y), \qquad \boxed{\widehat{I}(x,y)=\widehat{\alpha}_m \text{ for } (x,y)\in R'_m}$$

Step 3 — Extracting the noise from the samples

Define residuals (estimated noise samples) on Ω_b by

$$\widehat{W_k}(x,y) \;=\; J_k(x,y)-\widehat{I}(x,y), \qquad (x,y)\in\Omega_b$$

Since $\widehat{I}(x,y)\to I(x,y)$, the residuals converge to the true noise $W_k(x,y)$ and can be used to estimate its distribution.

Step 4 — Noise distribution via histogram

Form the collection of residuals

$$\mathcal{S} \;=\; \{\widehat{W_k}(x,y) : (x,y)\in\Omega_b,\; k=1,\dots,N\}.$$

To visualize the estimated distribution of the noise, construct a histogram of S. This histogram is our final estimate of the noise distribution. Alternatively, one can also perform kernel density estimation (KDE) to obtain a smooth more accurate estimate of the distribution instead of using a histogram.

Question 2

Part (a)

$$\begin{aligned}
F_V(x) &= P(V\le x)\\
&= P(F^{-1}(U)\le x) && \text{(since } F \text{ is strictly increasing and } V=F^{-1}(U))\\
&= P(U\le F(x)) && \text{(where } U\sim \text{Uniform}(0,1), \text{ so } P(U\le u)=u, \quad u\in[0,1])\\
&= F(x).
\end{aligned}$$

Therefore, V has distribution function F.

If $u_1,\dots,u_n$ are i.i.d. from [0,1] uniform distribution then the values $v_i=F^{-1}(u_i), \quad i=1,\dots,n$ are i.i.d. as well which follow the distribution with CDF F.

Part (b)

$$\begin{aligned}
P(D\ge d) &= P\left\{\max_x \left|\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{Y_i\le x\}-F(x)\right|\ge d\right\}\\
&= P\left\{\max_x \left|\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{F(Y_i)\le F(x)\}-F(x)\right|\ge d\right\}\\
&= P\left\{\max_x \left|\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{U_i\le F(x)\}-F(x)\right|\ge d\right\} \qquad (U_i:=F(Y_i))
\end{aligned}$$

Let $y=F(x)$

$$\begin{aligned}
P(D\ge d) &= P\left\{\max_{0\le y\le 1} \left|\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\{U_i\le y\}-y\right|\ge d\right\}\\
&= P(E\ge d)
\end{aligned}$$

Hence proved.

Part (c)

Because $P(E \geq d) = P(D \geq d)$, the sampling distribution of

$$D = \max_x \left|F_e(x) - F(x)\right|$$

is *independent of* F.

It can therefore be used to see whether the distribution of the observed data matches known distribution F. If the data is actually from F, then $D = \max_x |F_e(x) - F(x)|$ should not be more than the corresponding discrepancy between the empirical CDF of n i.i.d. Uniform$(0, 1)$ variables and the true uniform CDF. Therefore, observing a value D_{obs} that is much larger than this benchmark is an event with small probability.

Question 3

Part (a)

$z_i = ax_i + by_i + c + \varepsilon_i \qquad$ where $\varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$

Likelihood:

$$L(a, b, c; \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{\left(z_i - (ax_i + by_i + c)\right)^2}{2\sigma^2}\right)}$$

Log Likelihood:

$$LL = log(L(a, b, c; \sigma^2)) = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\big(z_i - ax_i - by_i - c\big)^2$$

For fixed σ^2, maximizing LL w.r.t. a,b,c is equivalent to minimizing $\sum_{i=1}^{N}\big(z_i - ax_i - by_i - c\big)^2$

$$\frac{\partial LL}{\partial a} = 0 \implies \sum_{i=1}^{N} 2\big(z_i - ax_i - by_i - c\big)(-x_i) = 0$$

$$\implies \sum_{i=1}^{N} x_i z_i = a\sum_{i=1}^{N} x_i^2 + b\sum_{i=1}^{N} x_i y_i + c\sum_{i=1}^{N} x_i$$

$$\frac{\partial LL}{\partial b} = 0 \implies \sum_{i=1}^{N} 2\big(z_i - ax_i - by_i - c\big)(-y_i) = 0$$

$$\implies \sum_{i=1}^{N} y_i z_i = a\sum_{i=1}^{N} x_i y_i + b\sum_{i=1}^{N} y_i^2 + c\sum_{i=1}^{N} y_i$$

$$\frac{\partial LL}{\partial c} = 0 \implies \sum_{i=1}^{N} 2\big(z_i - ax_i - by_i - c\big)(-1) = 0$$

$$\implies \sum_{i=1}^{N} z_i = a\sum_{i=1}^{N} x_i + b\sum_{i=1}^{N} y_i + c\,N$$

Matrix form:

$$\begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & N \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum x_i z_i \\ \sum y_i z_i \\ \sum z_i \end{bmatrix}$$

Vector form:

$$X = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N & y_N & 1 \end{bmatrix}, \quad \theta = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}$$

$$\implies \quad (X^\top X)\,\theta = X^\top \mathbf{z}$$

Part (b)

$z_i = a_1 x_i^2 + a_2 y_i^2 + a_3 x_i y_i + a_4 x_i + a_5 y_i + a_6 + \varepsilon_i \qquad$ where $\varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$

Likelihood:

$$L(a_1, \ldots, a_6; \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{\left(z_i - (a_1 x_i^2 + a_2 y_i^2 + a_3 x_i y_i + a_4 x_i + a_5 y_i + a_6)\right)^2}{2\sigma^2} \right)}$$

Log Likelihood:

$$LL = log(L(a_1, \ldots, a_6; \sigma^2)) = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(z_i - a_1 x_i^2 - a_2 y_i^2 - a_3 x_i y_i - a_4 x_i - a_5 y_i - a_6\right)^2$$

For fixed σ^2, maximizing LL w.r.t. $a_1, \ldots, a_6$ is equivalent to minimizing

$$\sum_{i=1}^{N}\left(z_i - a_1 x_i^2 - a_2 y_i^2 - a_3 x_i y_i - a_4 x_i - a_5 y_i - a_6\right)^2$$

$$\frac{\partial LL}{\partial a_1} = 0 \implies \sum_{i=1}^{N} x_i^2 z_i = a_1 \sum_{i=1}^{N} x_i^4 + a_2 \sum_{i=1}^{N} x_i^2 y_i^2 + a_3 \sum_{i=1}^{N} x_i^3 y_i + a_4 \sum_{i=1}^{N} x_i^3 + a_5 \sum_{i=1}^{N} x_i^2 y_i + a_6 \sum_{i=1}^{N} x_i^2$$

$$\frac{\partial LL}{\partial a_2} = 0 \implies \sum_{i=1}^{N} y_i^2 z_i = a_1 \sum_{i=1}^{N} x_i^2 y_i^2 + a_2 \sum_{i=1}^{N} y_i^4 + a_3 \sum_{i=1}^{N} x_i y_i^3 + a_4 \sum_{i=1}^{N} x_i y_i^2 + a_5 \sum_{i=1}^{N} y_i^3 + a_6 \sum_{i=1}^{N} y_i^2$$

$$\frac{\partial LL}{\partial a_3} = 0 \implies \sum_{i=1}^{N} x_i y_i z_i = a_1 \sum_{i=1}^{N} x_i^3 y_i + a_2 \sum_{i=1}^{N} x_i y_i^3 + a_3 \sum_{i=1}^{N} x_i^2 y_i^2 + a_4 \sum_{i=1}^{N} x_i^2 y_i + a_5 \sum_{i=1}^{N} x_i y_i^2 + a_6 \sum_{i=1}^{N} x_i y_i$$

$$\frac{\partial LL}{\partial a_4} = 0 \implies \sum_{i=1}^{N} x_i z_i = a_1 \sum_{i=1}^{N} x_i^3 + a_2 \sum_{i=1}^{N} x_i y_i^2 + a_3 \sum_{i=1}^{N} x_i^2 y_i + a_4 \sum_{i=1}^{N} x_i^2 + a_5 \sum_{i=1}^{N} x_i y_i + a_6 \sum_{i=1}^{N} x_i$$

$$\frac{\partial LL}{\partial a_5} = 0 \implies \sum_{i=1}^{N} y_i z_i = a_1 \sum_{i=1}^{N} x_i^2 y_i + a_2 \sum_{i=1}^{N} y_i^3 + a_3 \sum_{i=1}^{N} x_i y_i^2 + a_4 \sum_{i=1}^{N} x_i y_i + a_5 \sum_{i=1}^{N} y_i^2 + a_6 \sum_{i=1}^{N} y_i$$

$$\frac{\partial LL}{\partial a_6} = 0 \implies \sum_{i=1}^{N} z_i = a_1 \sum_{i=1}^{N} x_i^2 + a_2 \sum_{i=1}^{N} y_i^2 + a_3 \sum_{i=1}^{N} x_i y_i + a_4 \sum_{i=1}^{N} x_i + a_5 \sum_{i=1}^{N} y_i + a_6 N$$

Matrix form:

$$\begin{bmatrix} \sum x_i^4 & \sum x_i^2 y_i^2 & \sum x_i^3 y_i & \sum x_i^3 & \sum x_i^2 y_i & \sum x_i^2 \\ \sum x_i^2 y_i^2 & \sum y_i^4 & \sum x_i y_i^3 & \sum x_i y_i^2 & \sum y_i^3 & \sum y_i^2 \\ \sum x_i^3 y_i & \sum x_i y_i^3 & \sum x_i^2 y_i^2 & \sum x_i^2 y_i & \sum x_i y_i^2 & \sum x_i y_i \\ \sum x_i^3 & \sum x_i y_i^2 & \sum x_i^2 y_i & \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i^2 y_i & \sum y_i^3 & \sum x_i y_i^2 & \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i^2 & \sum y_i^2 & \sum x_i y_i & \sum x_i & \sum y_i & N \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} \sum x_i^2 z_i \\ \sum y_i^2 z_i \\ \sum x_i y_i z_i \\ \sum x_i z_i \\ \sum y_i z_i \\ \sum z_i \end{bmatrix}$$

Vector form:

$$X = \begin{bmatrix} x_1^2 & y_1^2 & x_1 y_1 & x_1 & y_1 & 1 \\ x_2^2 & y_2^2 & x_2 y_2 & x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N^2 & y_N^2 & x_N y_N & x_N & y_N & 1 \end{bmatrix}, \quad \theta = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix}, \quad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix}$$

$$\implies \quad (X^\top X)\,\theta = X^\top \mathbf{z}$$

Part (c)

Model variance(σ^2) is not needed for model fitting because it cancels out when the derivative of the log likelihood is equated to zero. Log Likelihood:

$$log(L(\theta;\sigma^2)) = \ell = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}|\mathbf{z} - X\theta|^2$$

Differentiating w.r.t. σ^2 :

$$\frac{\partial \ell}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \frac{|\mathbf{z} - X\theta|^2}{2(\sigma^2)^2} = 0 \implies \hat{\sigma}^2 = \frac{1}{N}\,|\mathbf{z} - X\hat{\theta}|^2$$

where $\hat{\theta} = (X^\top X)^{-1} X^\top \mathbf{z}$

Part (d)

The predicted equation of the plane is z = 10.0022084492 * x + 19.9980223089 * y + 29.9515789228.
The predicted noise variance($\hat{\sigma}^2$) is 23.05696828.

Part (e)

Assume that the model before swaps is

$$z = ax + by + c + \varepsilon, \qquad \varepsilon \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2),$$

and that a small number of indices have had their z values pairwise exchanged.

1. Fit the plane by the above method on the observed data to get $\hat{\theta} = (\hat{a}, \hat{b}, \hat{c})$.
2. Find residuals $d_i = z_i - (\hat{a}x_i + \hat{b}y_i + \hat{c})$ for all i and take magnitudes $|d_i|$.
3. Sort the magnitudes $|d_i|$ in descending order.
4. Because swapped z values are attached to the wrong (x_i, y_i), they should produce unusually large residuals . And as the corruptions are pairwise exchanges, the top of the sorted list should contain pairs of nearly equal large magnitudes i.e. each pair will correspond to the two values of z that were swapped.

To find which pairs to swap: After sorting the magnitudes $|d_i|$ in descending order, estimate the noise variance using the median absolute deviation (MAD) around the median:

$$\sigma' \;=\; 1.4862 \cdot \text{median}_i\big(\,|d_i - \text{median}(d)|\,\big).$$

Flag the indices with $|d_i - median(d)| > 2k\sigma'$ as outliers, where $k \in [2.5, 4]$.

Greedy pairing & swapping: Sort the flagged indices in descending order of $|d_i|$, then form consecutive pairs $(i_{(1)}, i_{(2)}), (i_{(3)}, i_{(4)}), \ldots$. Let $\text{MSE}_{\text{curr}} = \frac{1}{N}\sum_{i=1}^{N} d_i^2$. For each pair (p, q) in order, swap their observed z-values ($z_p \leftrightarrow z_q$) and recompute MSE_{new}. If $\text{MSE}_{\text{new}} < \text{MSE}_{\text{curr}}$, keep the swap and set $\text{MSE}_{\text{curr}} \leftarrow \text{MSE}_{\text{new}}$; otherwise, revert the swap and stop the greedy procedure.

5. After all accepted pairwise swaps are applied, fit the plane on the corrected data to get the final plane.

Why this works: If (i, j) is a true swapped pair, both $|d_i|$ and $|d_j|$ are large and of similar magnitude because each observed z is evaluated at the wrong (x, y). Swapping their z-values strictly reduces the expected sum of squared residuals for that pair. After correcting such pairs and refitting, this further reduces the residuals.

This idea was taken from Wikipedia.

Question 4

(a) Data Generation and Train/Validation Split

We draw $n = 1000$ samples from $N(0, 16)$ and split into:

$$T = \{t_1, \ldots, t_{750}\}, \quad V = \{v_1, \ldots, v_{250}\}, \quad T \cap V = \emptyset.$$

MATLAB code:

```
rng(1);
n = 1000;
X = 4 * randn(1,n);
idx = randperm(n);
T = X(idx(1:(3*n/4)));
V = X(idx((3*n/4 + 1):end));
```

(b) Joint Likelihood Expression

For bandwidth σ, the KDE using T is

$$\hat{p}_T(x;\sigma) = \frac{1}{|T|\,\sigma\sqrt{2\pi}} \sum_{t \in T} \exp\left(-\frac{(x-t)^2}{2\sigma^2}\right).$$

The joint likelihood of validation samples $V = \{v_1, \ldots, v_{250}\}$ is

$$L(\sigma) = \prod_{j=1}^{250} \hat{p}_T(v_j;\sigma).$$

The log-likelihood is

$$LL(\sigma) = \sum_{j=1}^{250} \log \hat{p}_T(v_j;\sigma).$$

(c) Cross-Validation Procedure

We evaluate the validation log-likelihood

$$LL(\sigma) \;=\; \sum_{v \in V} \log \widehat{p}_T(v;\sigma),$$

for bandwidths $\sigma \in \{0.001, 0.1, 0.2, 0.9, 1, 2, 3, 5, 10, 20, 100\}$, where $\widehat{p}_T(x;\sigma)$ is the Gaussian kernel KDE fit on the training set T with bandwidth σ.

Result (LL versus $\log\sigma$). Figure shows the curve of $LL(\sigma)$ against $\log\sigma$. The maximum of this curve identifies the choice of bandwidth:

$$\boxed{\sigma^{\star}_{LL} = 0.9}$$

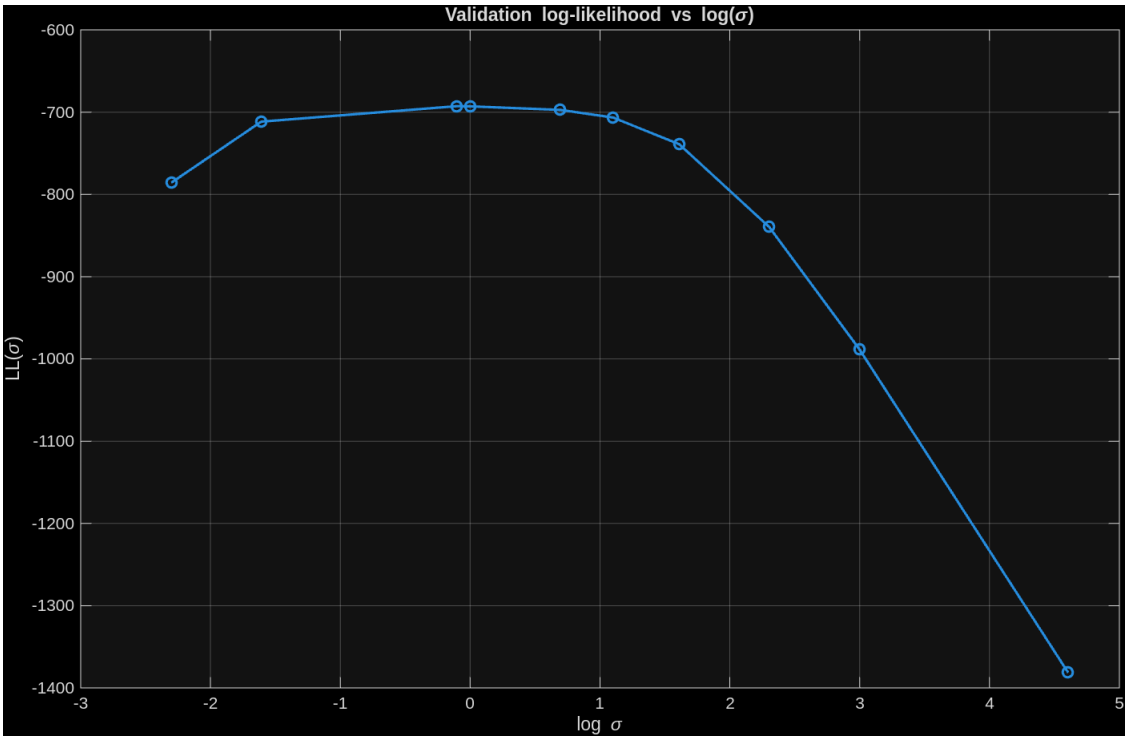

Figure 1: log-likelihood $LL(\sigma)$ as a function of $\log\sigma$.

Density at the best σ. Using $\sigma^{\star}_{LL}$, we evaluate the KDE $\widehat{p}_n(x;\sigma^{\star}_{LL})$ on the grid $x \in [-8 : 0.1 : 8]$ and overlay it with the true density $p(x) = \mathcal{N}(0, 16)$. The KDE closely resembles the true curve as shown in the figure.

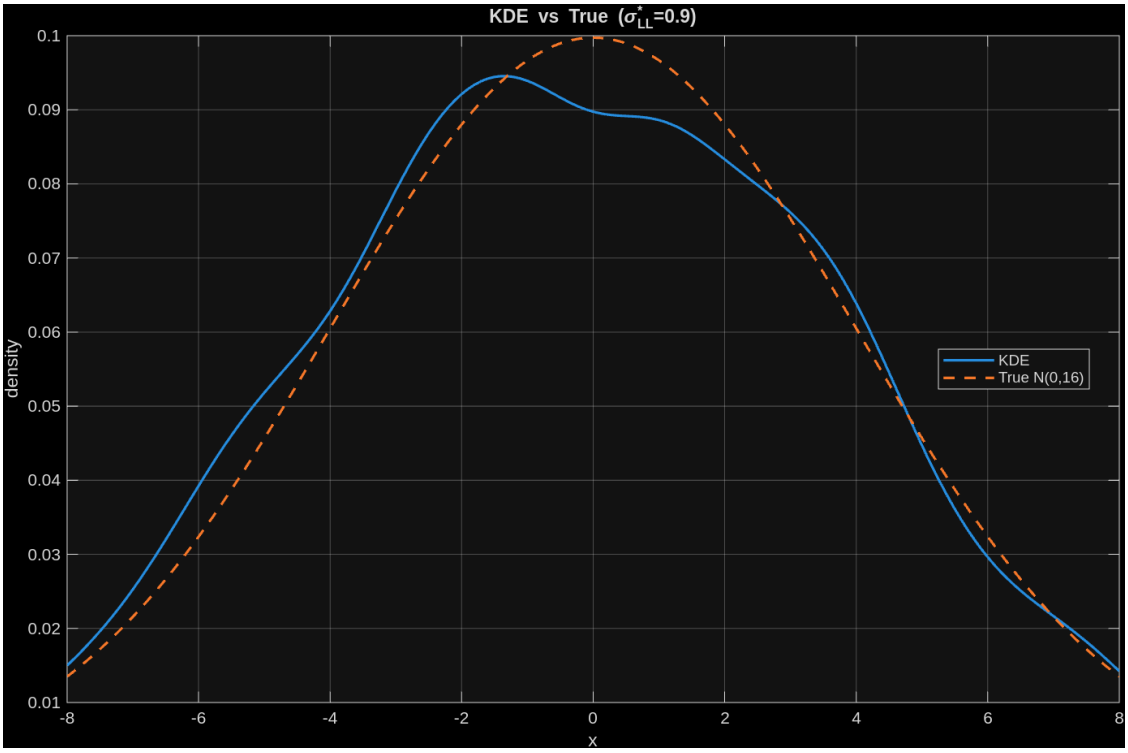

Figure 2: KDE $\widehat{p}_n(x;\sigma^\star_{LL})$ and the true density $p(x)=\mathcal{N}(0,16)$ with $\sigma^\star_{LL}=1.0$.

(d) Ground Truth Comparison

In this experiment, we know the true probability density function

$$p(x)=\mathcal{N}(0,16)=\frac{1}{4\sqrt{2\pi}}\exp\left(-\frac{x^2}{32}\right).$$

Therefore, we can directly compare the estimated density with the ground truth. For each candidate bandwidth σ, we compute the discrepancy

$$D(\sigma)=\sum_{v\in V}\big(p(v)-\hat{p}_T(v;\sigma)\big)^2.$$

Result (D versus $\log\sigma$). Figure shows $D(\sigma)$ as a function of $\log\sigma$. The minimum of this curve occurs at $\sigma^\star_D$

$$\boxed{\sigma^\star_D=1.0}$$

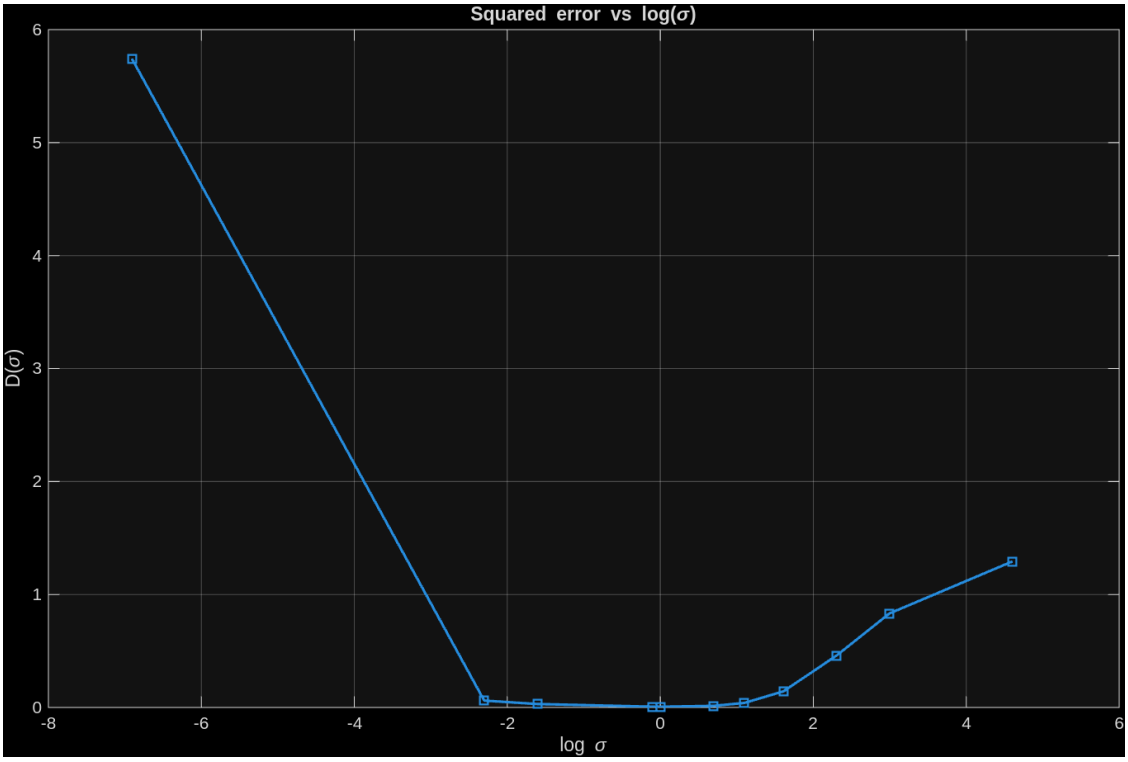

Figure 3: Squared error $D(\sigma)$ versus $\log \sigma$.

Comparison with $\sigma^\star_{LL}$. The squared error for the bandwidth chosen by log-likelihood cross-validation, $\sigma^\star_{LL} = 0.9$ (from part (c)), is

$$D(\sigma^\star_{LL}) = 0.0066,$$

which is close to the minimum $D(\sigma^\star_D)$ value (0.0064), confirming that cross-validation performs well even without access to the true distribution.

Density overlay at the best $\sigma^\star_D$. Figure shows the KDE $\hat{p}_n(x; \sigma^\star_D)$ evaluated on the grid $x \in [-8 : 0.1 : 8]$ and overlaid with the true density $p(x)$. The two curves match closely across the domain.

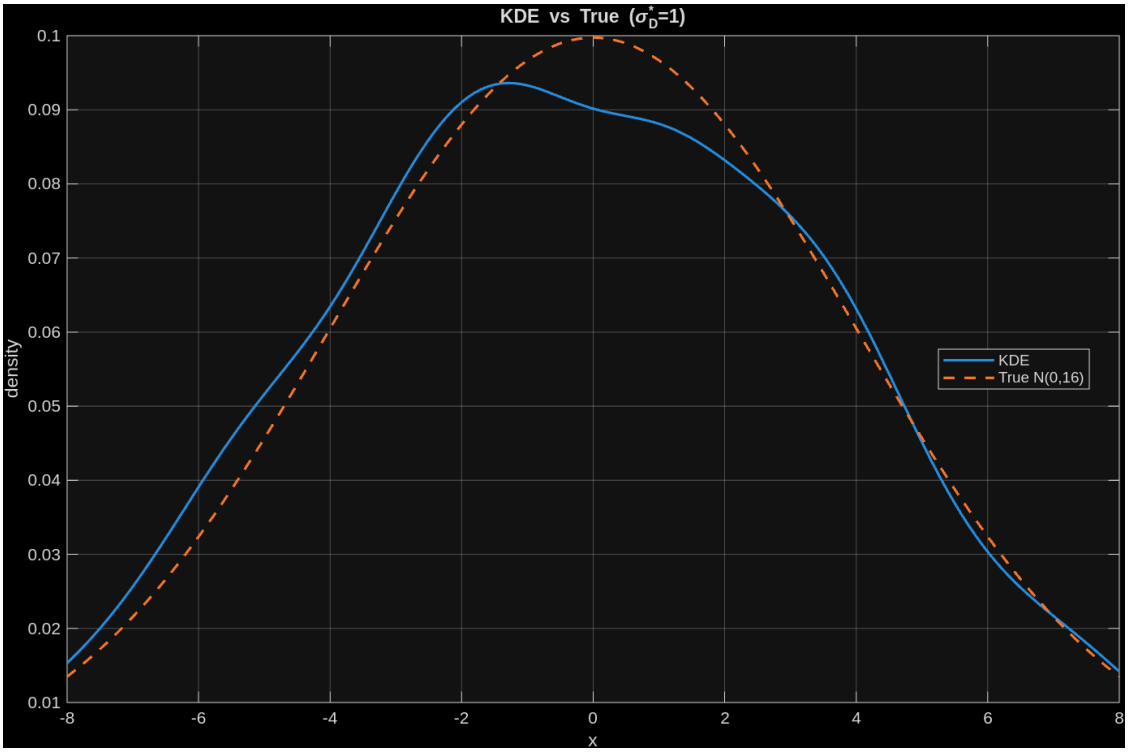

Figure 4: KDE $\hat{p}_n(x; \sigma_D^\star)$ and the true density $p(x) = \mathcal{N}(0, 16)$.

(e) What if $T = V$?

If the training and validation sets coincide, cross-validation fails:

- The procedure always picks the smallest σ in the set.
- For $\sigma \to 0$, each validation sample v_j has a kernel centered at itself, making $\hat{p}_T(v_j; \sigma) \to \infty$, thus, $LL(\sigma)$ increases monotonically as $\sigma \to 0$.

This is why T and V must be disjoint.