

Student Number: 21000790

Question 1: The cooked books

a) With the cooked data set, how would you estimate  $\lambda$ ? [4 points]

Let the honest distribution of arrivals be  $X_i$ , we know that  $X \sim \text{Pois}(\lambda)$ .

In the cooked distribution,  $X = 0$  and  $X = 1$  become  $X_{\text{cooked}} = 1$ , while other  $X > 1$  remains the same. The expected number of arrivals to boutique shop therefore is:

$$E(X_{\text{cooked}}) = 1 \cdot P(X_{\text{cooked}} = 1) + \sum_{x=2}^{\infty} x \cdot P(X = x)$$

where on the LHS:

The first term is:

$$P(X_{\text{cooked}} = 1) = P(X = 0) + P(X = 1) \\ = e^{-\lambda} + \lambda e^{-\lambda}$$

The second term is:

$$\sum_{x=2}^{\infty} x \cdot \frac{e^{-\lambda} \lambda^x}{x!} = \sum_{x=1}^{\infty} x \cdot \frac{e^{-\lambda} \lambda^x}{x!} - (0 \cdot P(X = 0) + 1 \cdot P(X = 1)) \\ = \lambda - (0 \cdot P(X = 0) + 1 \cdot P(X = 1)) \\ = \lambda - (0 \cdot e^{-\lambda} + 1 \cdot \lambda e^{-\lambda}) \\ = \lambda - \lambda e^{-\lambda}$$

Therefore:

$$E(X_{\text{cooked}}) = e^{-\lambda} + \lambda e^{-\lambda} + \lambda - \lambda e^{-\lambda} \\ = e^{-\lambda} + \lambda$$

Given independent variables representing the arrivals for each day for the 'cooked' process:  $X_1, X_2, \dots, X_n$ , the law of large numbers says that:

$$\frac{1}{n}(X_1 + X_2 + \dots + X_n) \rightarrow E(X_1) = e^{-\lambda} + \lambda$$

As  $n \rightarrow \infty$ , the LHS converges in probability to the true mean parameter of the 'cooked' process. On the RHS, the term  $e^{-\lambda}$  is a correction on the honest process. We can solve for  $\lambda$  to obtain a consistent, unbiased estimate for  $\lambda$  ( $\lambda > 0$ ). However, because the equation needs to be solved numerically and involves approximation, the solution can introduce some bias. But overall this is a reasonable estimation method.

b) Test your method, in the case and  $\lambda = 1$  [3 points]

```
# Function to solve for lambda from the mean of a 'cooked' sample
solve_for_lambda <- function(x_bar) {
  equation <- function(lambda) {
    return(exp(-lambda) + lambda - x_bar)
  }
  return(unroot(equation, interval = c(0, 10))$root)
}

# Parameters
lambda_true <- 1
N <- 100 # Number of samples
n <- 300 # Sample size
set.seed(1234)

# Array to store estimated lambda
estimated_lambdas <- numeric(N)

# Generate and analyse samples
for (i in 1:N) {
  sample <- rpois(n, lambda_true)
  sample[sample == 0] <- 1 # Cook the data
  x_bar <- mean(sample)
  estimated_lambdas[i] <- solve_for_lambda(x_bar)
}

# Compute the average of estimated lambda values
average_estimated_lambda <- mean(estimated_lambdas)
print(paste("Average estimated lambda:", average_estimated_lambda))

[1] "Average estimated lambda: 1.00147656898498"

print(paste("Bias:", average_estimated_lambda - lambda_true))

[1] "Bias: 0.00147656898498272"
```

c) Now, use your method to estimate  $\lambda$  for the given data [3 points]

```
x_bar_c <- mean(data$x)
estimated_lambda_c <- solve_for_lambda(x_bar_c)
print(paste("Estimated lambda:", estimated_lambda_c))

[1] "Estimated lambda: 2.22901609606056"
```

Question 2: The half parabola

a) Sampling from the region R and the plot of x-coordinates [3 points]

We use rejection sampling method to sample for values of U. We will generate two uniform random numbers,  $u$  and  $v$ , where  $u$  is drawn from the interval  $[0, 1]$  and  $v$  is drawn from  $[0, f(u)]$ . A sampled point  $(u, v)$  is accepted if  $v \leq f(u)$ .

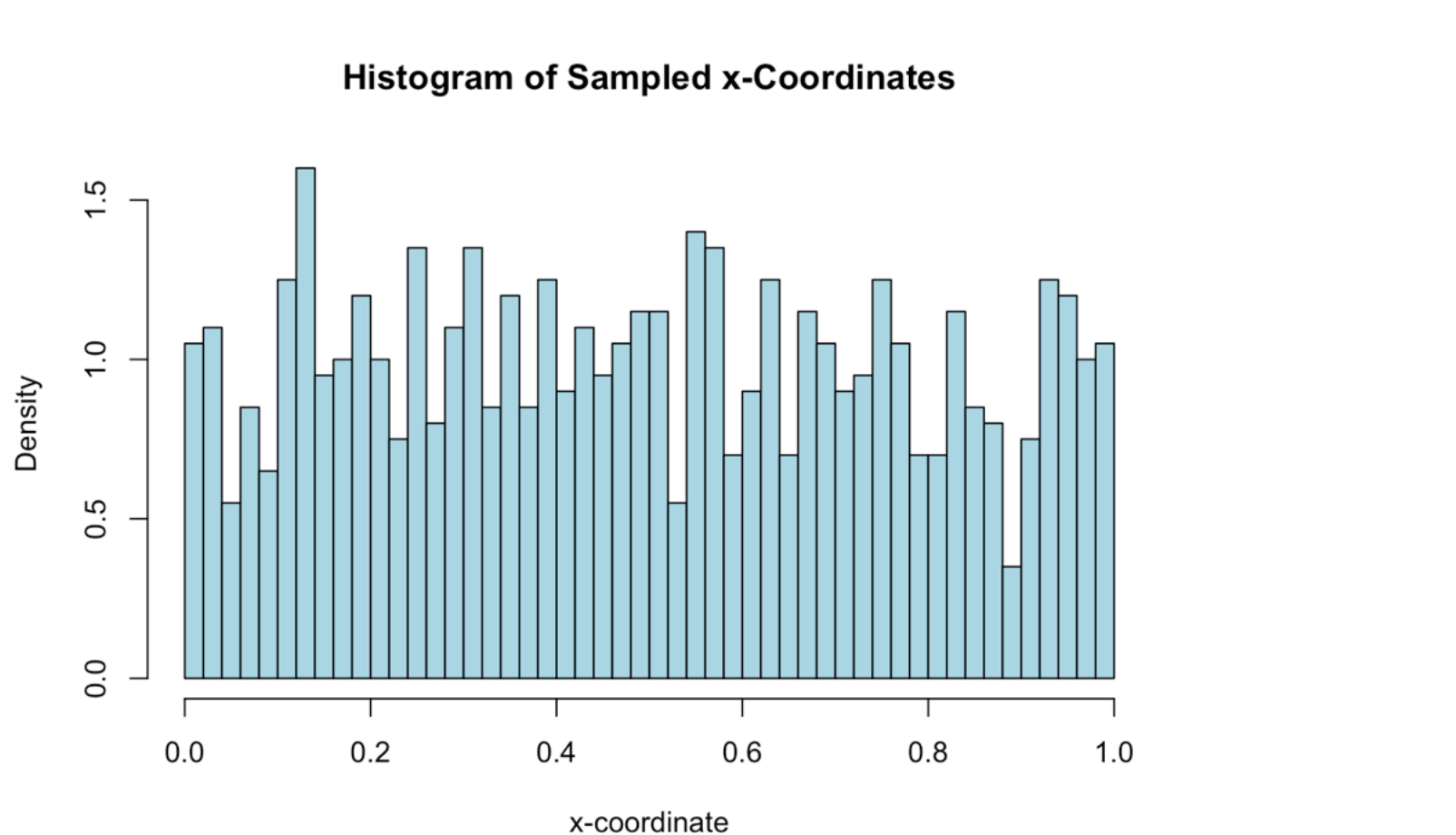
```
# Define the function f(x)
f <- function(x) {
  return(3/2 * (1 - x^2))
}

# Number of points to sample
n_samples <- 1000
set.seed(1234)

# Initialise vector to save accepted x-coordinates
x_points <- numeric(n_samples)

# Sample points using the rejection sampling method
for (i in 1:n_samples) {
  repeat {
    u <- runif(1, 0, 1)
    v <- runif(1, 0, f(u))
    if (v <= f(u)) {
      x_points[i] <- u
      break
    }
  }
}

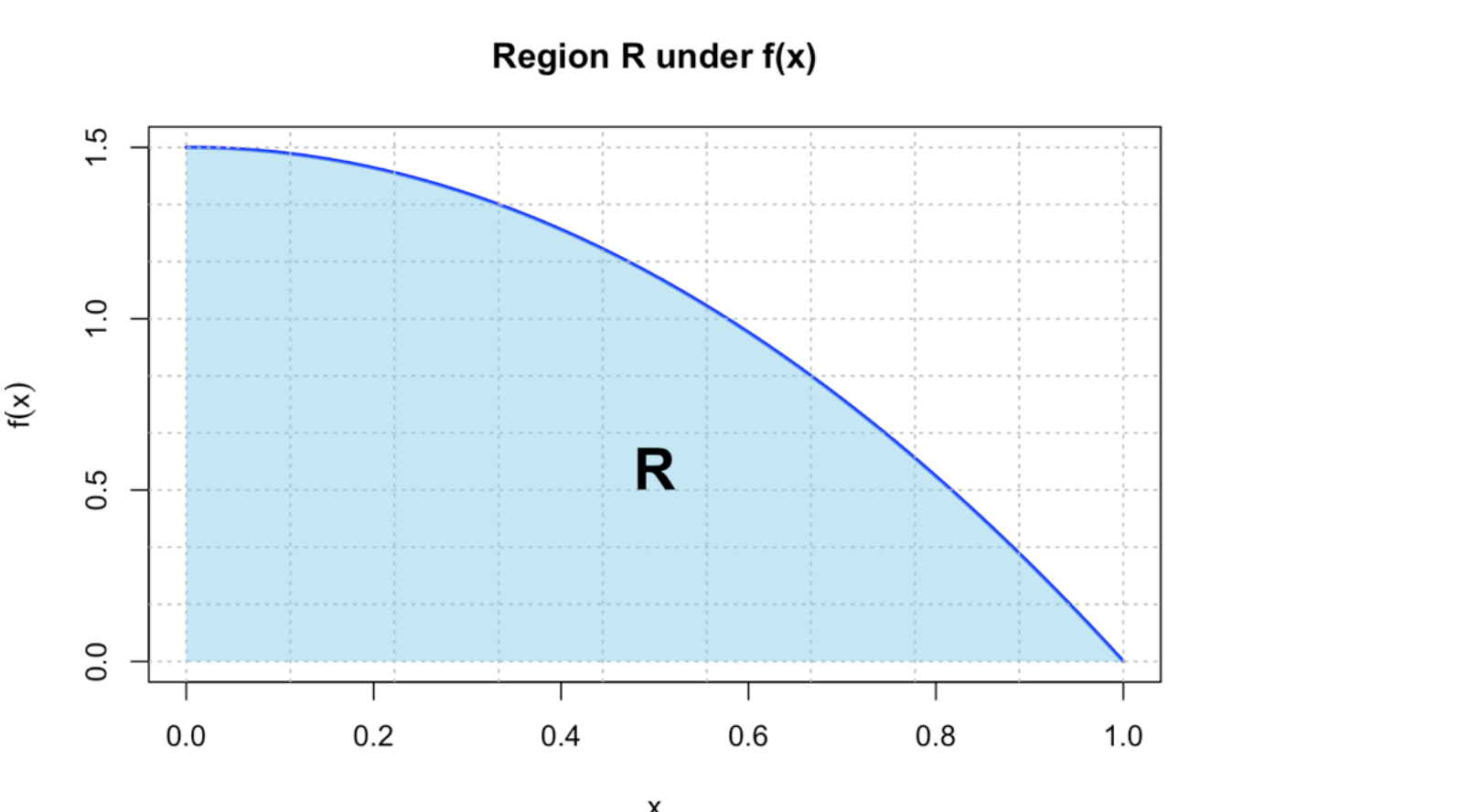
# Plotting the histogram of x-coordinates
hist(x_points, prob = T, breaks = 50, col = "lightblue", main = "Histogram of Sampled x-coordinates")
```



b) Distribution of x-coordinates of U [1 point]

The x-coordinates of U are chosen uniformly through  $u \sim U(0, 1)$ , yet was subject to the acceptance condition. Therefore, X is distributed non-uniformly depends on the acceptance probability: more likely towards 0 and less likely toward 1 - observed from the plot below the area of R compared to the overall rectangle.  $f_X(x)$  is the acceptance probability: The value of each strip of  $f(x)$  divided by  $\frac{3}{2}$ .

$$f_X(x) = \begin{cases} 1 - x^2 & \text{if } 0 \leq x < 1, \\ 0 & \text{otherwise.} \end{cases}$$



c) Are X and Y independent? [1 point]

X and Y are not independent. The range of possible values for Y depends on the value of X, since Y is uniformly distributed between 0 and  $f(X)$ .

Question 3: The Casino

Let B be the Blackjack state, and R be the Roulette state. The transition matrix is:

$$P = \begin{bmatrix} P_{BB} & P_{BR} \\ P_{RB} & P_{RR} \end{bmatrix} = \begin{bmatrix} 0.43 & 0.57 \\ \frac{26}{37} & \frac{1}{37} \end{bmatrix}$$

a) Solving by simulation [5 points]

```
# Make sure of reproducibility
set.seed(1234)

# Define the number of steps
N <- 10000

# Initialise variables to track state and winnings
state <- 1 # Start at blackjack
total <- 0

# Define the transition probabilities and betting outcomes
P_BB <- 0.43
P_RR <- 1/(36+1)

blackjack_win <- 5
blackjack_loss <- -5
roulette_win <- 35
roulette_loss <- -1

# Simulation
for (i in 1:N) {
  if (state == 1) { # If playing blackjack
    if (runif(1) < P_BB) {
      total <- total + blackjack_win # Win at blackjack
    } else {
      total <- total + blackjack_loss # Lose at blackjack and switch to roulette
      state <- 2
    }
  } else { # If playing roulette
    if (runif(1) < P_RR) {
      total <- total + roulette_win # Win at roulette
    } else {
      total <- total + roulette_loss # Lose at roulette and switch to blackjack
      state <- 1
    }
  }
}

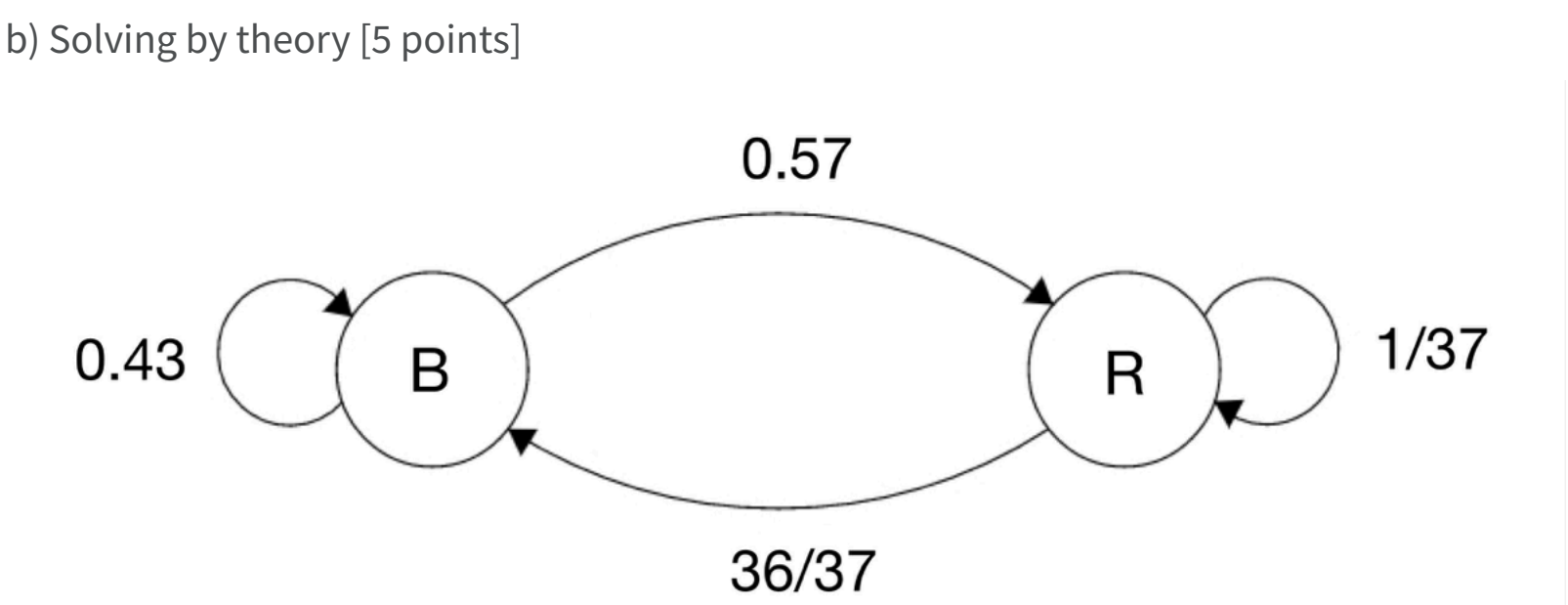
# Output the results
print(paste("S_N is: ", total, "pounds"))

[1] "S_N is: -4806 pounds"

print(paste("Average win per step is: ", total/N, "pounds"))

[1] "Average win per step is: -0.4806 pounds"
```

b) Solving by theory [5 points]



The Markov chain in this problem has finite state space, is irreducible, and has periodicity 1 (can be seen from the figure above,  $\text{per}(R) = \gcd(1, 2) = 1$ , and everything else has the same periodicity as they share the same class). It therefore has a stationary distribution, which can be calculated as follow:

```
# Define the transition matrix with the given probabilities
P <- matrix(c(0.43, 0.57, 36/37, 1/37), nrow = 2)

# Calculate eigenvalues and eigenvectors of the transposed transition matrix
E = eigen(P) # Eigen decomposition to find stationary distribution

# Extract the eigenvector associated with the largest eigenvalue
EE = E$vectors[,1] # Assuming the first column corresponds to eigenvalue 1

# Normalise the eigenvector to sum to 1 to get the stationary distribution
F = EE / sum(EE) # Normalisation step to get probabilities

# Output the stationary distribution
F

[1] 0.6305833 0.3694167
```

Therefore the stationary distribution, or the proportion of time spent playing two games in the long-run is:

$$P = [\lambda_B \quad \lambda_R] = [0.6305 \quad 0.3694]$$

The expected gain from each game are:

$$\text{Blackjack} = 0.43 * 5 + 0.57 * -5 = -0.70$$

$$\text{Roulette} = \frac{1}{37} * 35 + \frac{36}{37} * -1 = \frac{-1}{37}$$

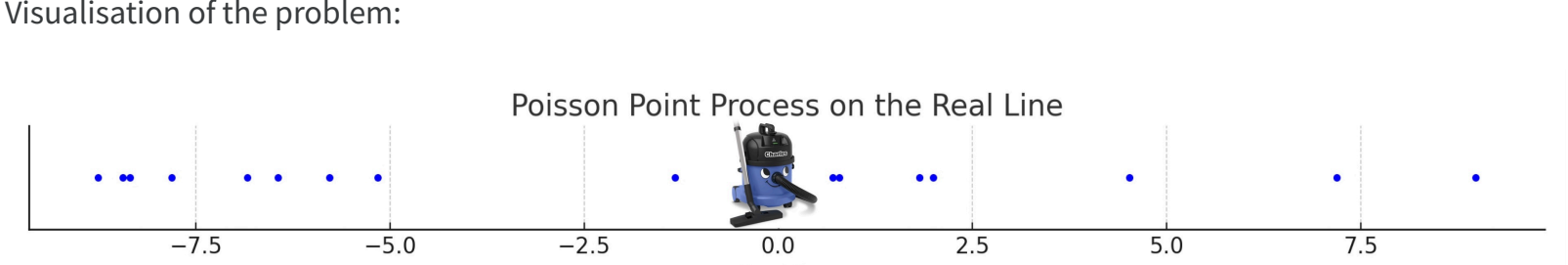
Expected win in the long run for each step is:

$$\text{Blackjack} * \lambda_B + \text{Roulette} * \lambda_R = -0.451$$

So the theoretical result for steply win is -0.451 pound compared to the simulated result, which is -0.481 pound.

Question 4: The Poisson collector [5 points]

Visualisation of the problem:



Because the dust follow a Poisson process, the distance between each pair of them is exponential distributed, with  $\lambda = 1$ . Providing that it starts at 0, we want the probability that it heads to the right twice. We also want that it only moves to the nearest dust point.

The strategy is to leverage the law of large numbers. We simulate the situation where the vacuum heads right twice consecutively,  $N$  times, count the number of times the event happens, then divide by  $N$ . As  $N \rightarrow \infty$ , we have a reasonable estimate for  $P$ .

```
vacuum <- function() {
  # Step 1: Vacuum the first piece of dust
  first_positive <- rexp(1, rate = 1)
  first_negative <- rexp(1, rate = 1)

  # If the first piece of dust is in the positive direction
  if (first_positive < first_negative) {

    # Step 2: Vacuum the second piece of dust
    second_positive <- rexp(1, rate = 1) # 1 unit to the right
    second_negative <- rexp(1, rate = 1) # 1 unit to the left

    # If the second piece of dust is also in the positive direction
    if (second_positive < second_negative) {
      return(1) # Indicates two successive moves in the positive direction
    }
  }

  return(0)
}

# Perform the simulations N times
results <- replicate(100000, vacuum())

# Estimate the probability
probability <- mean(results)
print(paste("The estimated probability is: ", probability))

[1] "The estimated probability is: 0.250570000000001"
```

Question 5: Coupling [5 points]

Let  $X$  be a Poisson random variable with mean  $\lambda$  and  $Y$  be a Poisson random variable with mean  $\lambda + \varepsilon$ . We construct a coupling  $(X, Y)$  by defining  $Y = X + Z$ , where  $Z$  is an independent  $\text{Poisson}(\varepsilon)$  random variable. This ensures  $Y$  has the desired mean  $\lambda + \varepsilon$ .

The total variation distance between  $X$  and  $Y$  in terms of this coupling is bounded by, from Theorem 6.1:

$$d_{TV}(X, Y) \leq 2 * P(X \neq Y) = P(Z > 0)$$

In order words,  $X$  and  $Y$  differ if and only if  $Z$ , the extra Poisson noise, is greater than zero.

Since  $Z$  follows Poisson,  $P(Z > 0) = 1 - e^{-\varepsilon}$ .

As  $\varepsilon \rightarrow 0$ ,  $e^{-\varepsilon} \rightarrow 1$ , making  $P(Z > 0) \rightarrow 0$ . Consequently,  $d_{TV}(X, Y) \rightarrow 0$  as  $\varepsilon \rightarrow 0$ , showing the convergence of the total variational distance.