Pseudocode and Diagrams for Differential Privacy Methods

Contents

1 Introduction

This document presents the pseudocode and conceptual diagrams of the differential privacy methods implemented for quantitative and categorical variables. The included algorithms are:

- **Quantitative Data**:
  - Duchi et al. Mechanism (`duchi`)
  - Laplace Mechanism (`laplace`)
  - Piecewise Mechanism (`piecewise`)
  - Multidimensional Duchi Mechanism (`multidimensional_duchi`)
  - Custom Multidimensional Mechanism (`multidimensional`)
- **Categorical Data**:
  - Direct Encoding (`direct_encoding`)
  - Optimized Unary Encoding (OUE) (`optimized_unary_encoding`)
  - RAPPOR (`rappor`)

Below, the pseudocode and diagram for each of these methods are presented.

2 Pseudocode and Diagrams for Quantitative Methods

2.1 Duchi et al. Mechanism (`duchi`)

Algorithm 1 Duchi et al. Mechanism

Require: Input vector t_i with values in $[-1, 1]$, privacy budget ϵ

Ensure: Privatized vector t_i^*

for each element t_i in the input vector **do**
  Clamp t_i to the range $[-1, 1]$
  Compute e^ϵ
  Compute $p = \frac{e^\epsilon}{e^\epsilon + 1}$
  Compute probability $q = \frac{1}{e^\epsilon + 1}$
  Generate uniform random variable u in $[0, 1]$
  if $u < \frac{1 + t_i}{2}$ **then**
    Set $t_i^* = 1$ with probability p, -1 with probability q
  else
    Set $t_i^* = -1$ with probability p, 1 with probability q
  end if
end for
return Vector t_i^*

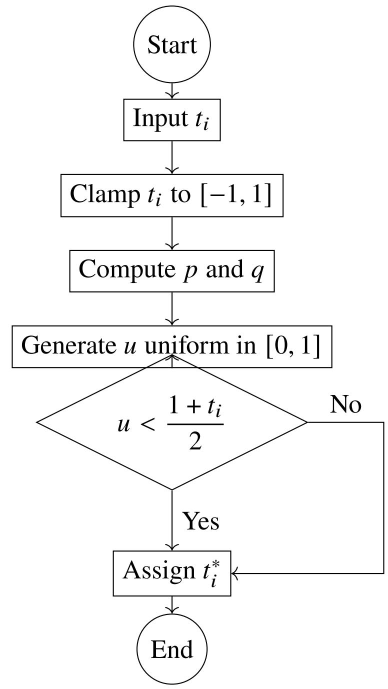

Figure 1: Diagram of the Duchi et al. Mechanism

2.2 Laplace Mechanism (`laplace`)

Algorithm 2 Laplace Mechanism

Require: Input vector t_i, privacy budget ϵ, sensitivity s

Ensure: Privatized vector t_i^*

1: **for** each element t_i in the input vector **do**

2: Generate Laplace noise b with mean 0 and scale $\frac{s}{\epsilon}$

3: Calculate $t_i^* = t_i + b$

4: **end for**

5: **return** Vector t_i^*

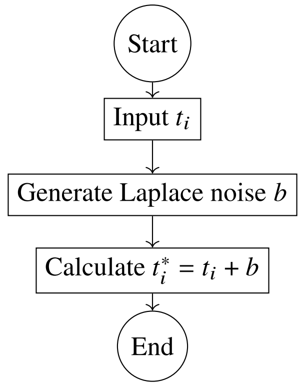

Figure 2: Diagram of the Laplace Mechanism

2.3 Piecewise Mechanism (`piecewise`)

Algorithm 3 Piecewise Mechanism

Require: Input vector t_i with values in $[-1, 1]$, privacy budget ϵ

Ensure: Privatized vector t_i^*

Compute $C = \dfrac{e^{\epsilon/2} + 1}{e^{\epsilon/2} - 1}$
for each element t_i in the input vector **do**
    Compute $l(t_i)$ and $r(t_i)$
    Generate u uniform in $[0, 1]$
    if $u \le \dfrac{e^{\epsilon/2}}{e^{\epsilon/2} + 1}$ **then**
        Generate t_i^* uniformly in $[l(t_i), r(t_i)]$
    else
        Randomly select $[-C, l(t_i)]$ or $[r(t_i), C]$
        Generate t_i^* uniformly in the selected interval
    end if
end for
return Vector t_i^*

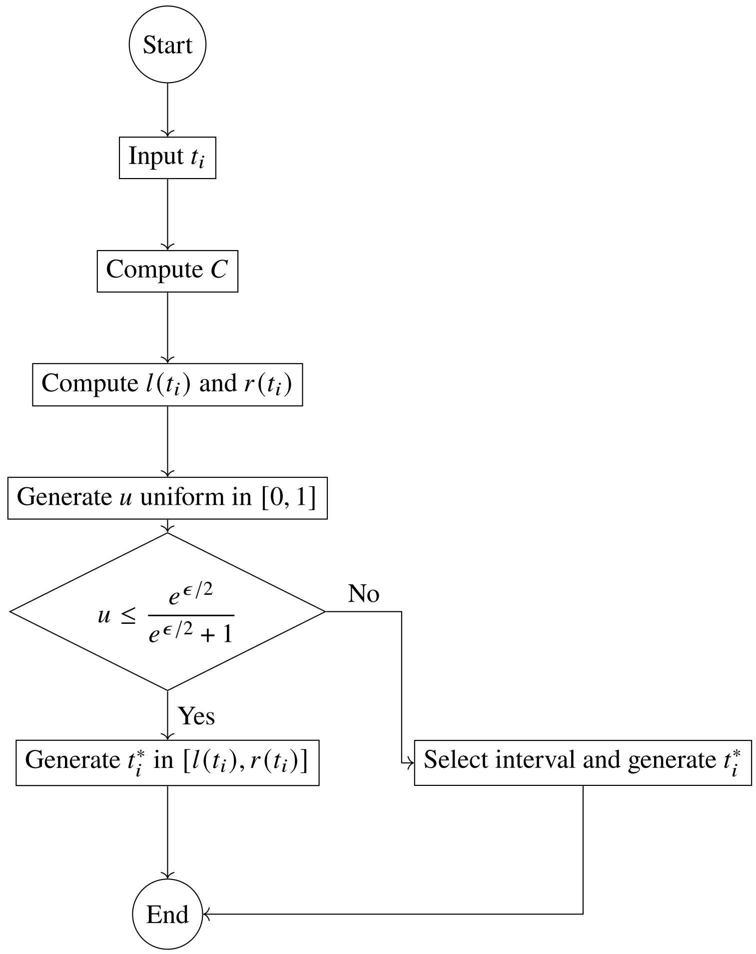

Figure 3: Diagram of the Piecewise Mechanism

2.4 Multidimensional Duchi Mechanism (`multidimensional_duchi`)

Algorithm 4 Multidimensional Duchi Mechanism

Require: Input matrix T of dimension $n \times d$, privacy budget ϵ

Ensure: Privatized matrix T^*

```
for each row t_i in T do
    Generate random vector v_i ∈ {−1, 1}^d
    Calculate s_i = e^ϵ / (e^ϵ + 1)
    Generate Bernoulli random variable u_i with probability s_i
    if u_i = 1 then
        Assign t_i^* = v_i
    else
        Assign t_i^* = −v_i
    end if
end for
return Matrix T^*
```

where $s_i = \frac{e^\epsilon}{e^\epsilon + 1}$.

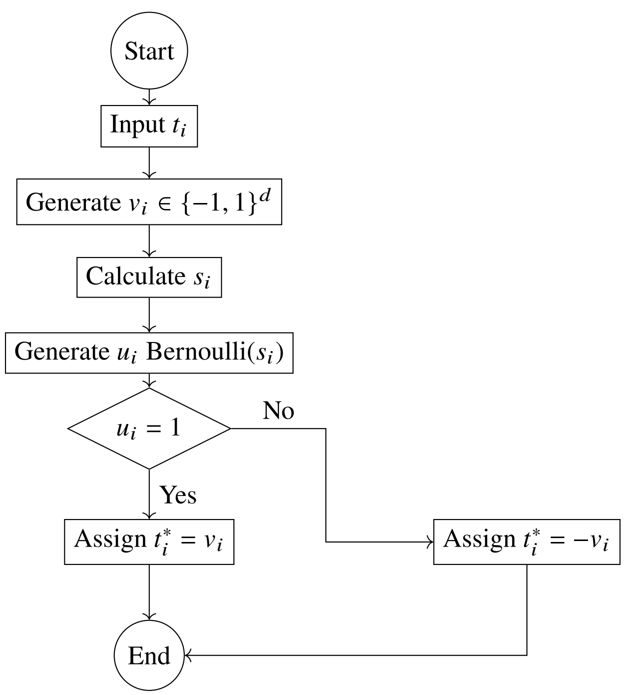

Figure 4: Diagram of the Multidimensional Duchi Mechanism

2.5 Custom Multidimensional Mechanism (`multidimensional`)

Algorithm 5 Custom Multidimensional Mechanism

Require: Input matrix T of dimension $n \times d$, privacy budget ϵ, unidimensional mechanism M, constant C

Ensure: Privatized matrix T^*

1: Calculate $k = \max\left(1, \min\left(d, \left\lfloor \frac{\epsilon}{2.5} \right\rfloor\right)\right)$
2: **for** each row t_i in T **do**
3: Select k random indices without replacement from $[1, d]$
4: **for** each selected index j **do**
5: Apply mechanism M to element t_{ij} with budget $\frac{\epsilon}{k}$
6: Multiply result by $\frac{d}{k}$ and assign to t^*_{ij}
7: **end for**
8: **end for**
9: **return** Matrix T^*

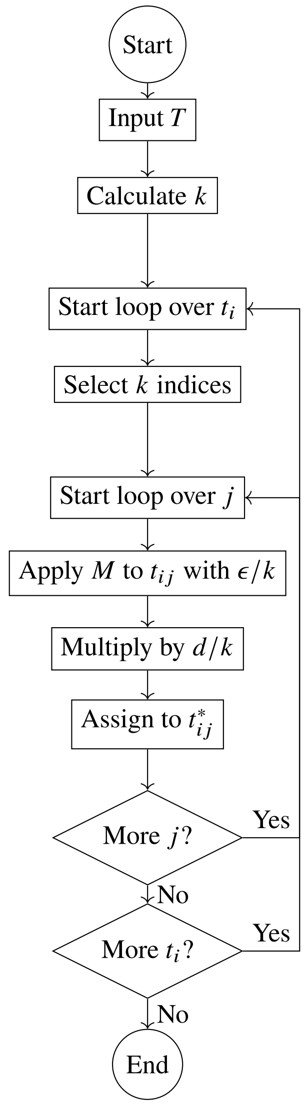

Figure 5: Diagram of the Custom Multidimensional Mechanism

3 Pseudocode and Diagrams for Categorical Methods

3.1 Direct Encoding (`direct_encoding`)

Algorithm 6 Direct Encoding

Require: Categorical data X, privacy budget ϵ

Ensure: Privatized data X^*

Obtain the number of categories k

Calculate $p = \dfrac{e^\epsilon}{e^\epsilon + k - 1}$

Calculate $q = \dfrac{1}{e^\epsilon + k - 1}$

for each element x_i in X **do**

    For each category c_j:

        Assign probability p if $x_i = c_j$, q otherwise

    Generate category x_i^* based on the assigned probabilities

end for

return Privatized data X^*

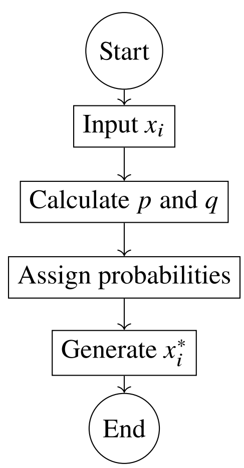

Figure 6: Diagram of Direct Encoding

3.2 Optimized Unary Encoding (OUE) (`optimized_unary_encoding`)

Algorithm 7 Optimized Unary Encoding (OUE)

Require: Categorical data X, privacy budget ϵ

Ensure: Privatized data X^*

Obtain the number of categories d
Represent each data point x_i as a unary vector u_i
Set $p = 0.5$
Calculate $q = \frac{1}{e^{\epsilon} + 1}$
for each unary vector u_i **do**
    for each bit u_{ij} in u_i **do**
        if $u_{ij} = 1$ **then**
            Perturb u_{ij} with probability p
        else
            Perturb u_{ij} with probability q
        end if
    end for
    Reconstruct x_i^* from the perturbed vector
end for
return Privatized data X^*

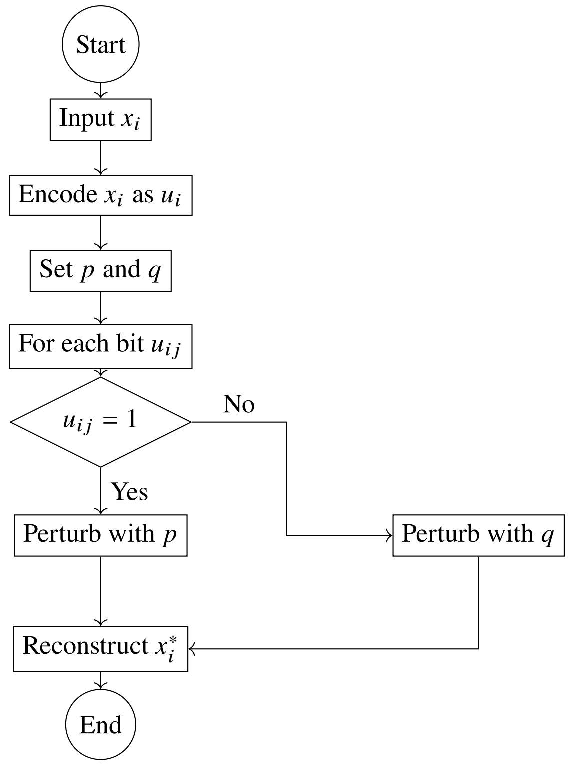

Figure 7: Simplified Diagram of Optimized Unary Encoding (OUE)

3.3 RAPPOR (`rappor`)

Algorithm 8 RAPPOR

Require: Categorical data X, privacy budget ϵ

Ensure: Privatized data X^*

Obtain the number of categories d
Represent each data point x_i as a unary vector u_i
Calculate $f = \frac{1}{e^{\epsilon} + 1}$
for each unary vector u_i **do**
    for each bit u_{ij} in u_i **do**
        if $u_{ij} = 1$ **then**
            Perturb u_{ij} with probability $1 - f$
        else
            Perturb u_{ij} with probability f
        end if
    end for
    Reconstruct x_i^* from the perturbed vector
end for
return Privatized data X^*

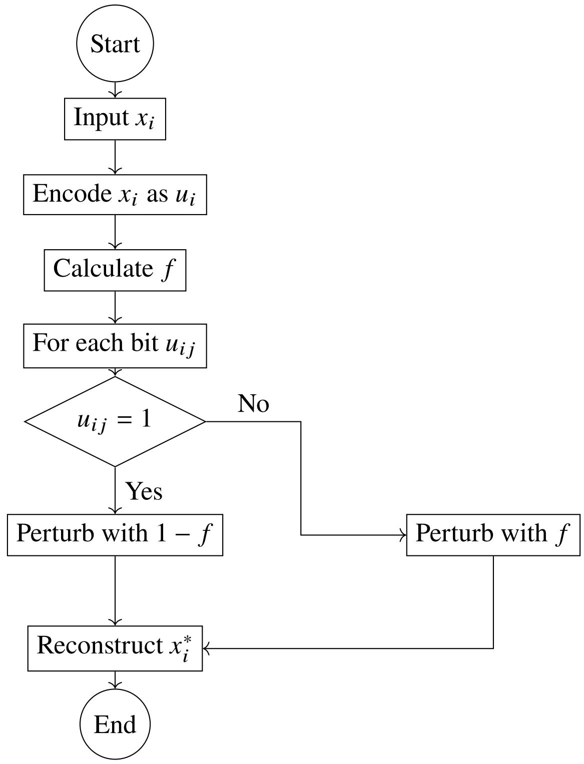

Figure 8: Simplified Diagram of RAPPOR