Correlated Noise

Today

- Review: Factorization & Binary Tree Mechanism
- Correlated Noise for DP-GD
- Examples
- Considerations

Review: Factorization Mechanisms

Answering k linear queries on $(x_1, \dots, x_n) \in \mathcal{U}^n$

$$f_j(x) = \frac{1}{n} \sum_{i=1}^{n} \phi_j(x_i)$$

ϕ_j binary predicate

$\phi_j : \mathcal{U} \to \{0,1\}$

$|\mathcal{U}| = m$

Workload:

$$F : \mathcal{U}^n \to [0,1]^k$$

Rewrite dataset in "histogram form"

$$(h_x)_u = \frac{1}{n} \#\{i \in [n] : x_i = u\}$$

$h_x \in [0,1]^m$

Rewrite predicate as vector

$$v_\varphi = (\varphi(u_1), \varphi(u_2), \ldots, \varphi(u_m))$$

Rewrite query as inner product

$$f_j(x) = \langle v_{\varphi_j}, h_x \rangle$$

Write <u>workload matrix</u>

$$F = \begin{bmatrix} — v_{\varphi_1} — \\ — v_{\varphi_2} — \\ \vdots \\ — v_{\varphi_k} — \end{bmatrix}$$

True answers Fh_x, matrix-vector product.

Gaussian Mechanism:

Release $Fh_x + Z$

↖ Gaussian noise $\mathcal{N}(0, \sigma^2 \mathbb{I}_k)$ or Laplace

Factorization Mechanism

Write $F = RM$ matrix factorization

"reconstruction" "measurement"

Release $R(Mh_x + z)$

Observe

(1) Can have much better error, picking what to answer.

(2) True answers plus correlated noise

$$Fh_x + \mathcal{N}(0, \sigma^2 RR^T)$$

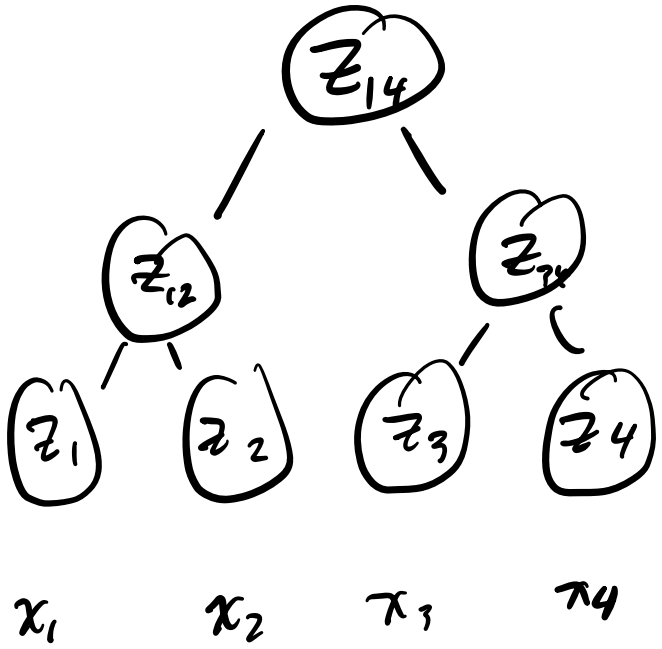
Review: Binary Tree Mechanism

Important instance of above.

Consider T people, each holding a bit x_t.

Task: $\forall t$, estimate prefix sum $s_t = \sum_{\tau=1}^{t} x_\tau$

Binary Tree Mechanism



Time	Output
$t=1$	$(x_1)+(z_1)$
$t=2$	$(x_1+x_2)+(z_{12})$
$t=3$	$(x_1+x_2+x_3)+(z_{12}+z_3)$
$t=4$	$(x_1+x_2+x_3+x_4)+(z_{14})$

Observe

1. Each output needs only $\leq \log T$ noise terms $\Rightarrow$ error $= O(\text{polylog} T)$
2. correlated answers!
3. Extends to setting where $x_i \in \mathbb{R}^d$
4. Can be done sequentially! Don't need to "look ahead" in time.

DP-SGD as Prefix Sums

$$\theta_0$$

$$\theta_1 = \theta_0 - \eta g_0$$

$$\theta_2 = \theta_1 - \eta g_1$$

$$= \theta_0 - \eta g_1 - \eta g_0$$

$$\vdots$$

$$\theta_t = \theta_0 - \eta \sum_{\tau=1}^{t-1} g_\tau = \theta_0 - \eta s_t$$

At each time point, call subroutine to estimate prefix sum

Old goal: $\forall t$, try to estimate g_t as well as possible

New goal: $\forall t$, try to estimate s_t as well as possible.

Correlation can cause noise to cancel out.

Prefix sum matrix

$$S = \begin{bmatrix} — s_0 — \\ — s_1 — \\ \vdots \\ — s_{t-1} — \end{bmatrix} \begin{matrix} \\ \\ \\ \end{matrix} \quad \begin{matrix} d \text{ cols} \\ \\ \\ t \text{ rows} \end{matrix}$$

Gradient matrix

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{t-1} \end{bmatrix} \begin{matrix} d \text{ cols} \\ \\ \\ t \text{ rows} \end{matrix}$$

Prefix sum workload matrix A

$$S = \begin{bmatrix} \quad \end{bmatrix} G$$

$$A \in \mathbb{R}^{T \times T}$$

Then factor $A = BC$, we'll run

$$M(G) = B(CG + Z)$$

$$= BCC^{-1}(CG + Z)$$

$$= A(G + C^{-1}Z)$$

Considerations

1) Adaptivity
2) Privacy
3) How does factorization affect error?
3) Computation
   - → Find good factorization
   - → generate noise at each time step.