

# Trustworthy Machine Learning From an Optimization Lens

Meisam Razaviyayn

Lecture 12: DP-FTRL and Denoising DP-SGD

[razaviya@usc.edu](mailto:razaviya@usc.edu)

# (Approximate) differential privacy

**Definition:** Let  $\epsilon, \delta > 0$  and  $X$  be the set of possible datasets. A randomized algorithm  $M(\cdot): X \rightarrow O$  is said to be  $(\epsilon, \delta)$  – differentially private if

$$\Pr(M(D) \in \Omega) \leq e^\epsilon \Pr(M(D') \in \Omega) + \delta \text{ for all } \Omega \subseteq O \text{ and all neighboring datasets } D, D' \in X$$

## Connection to pure DP

- **Definition:** Consider two fixed datasets  $D, D'$ , and a randomized mechanism  $M$ . The **privacy loss random variable (PLRV)** draws an outcome  $o$  from  $M(D)$  and outputs  $\ln\left(\frac{P(M(D)=o)}{P(M(D')=o)}\right)$ . In other words, the random variable takes the value of  $\ln\left(\frac{P(M(D)=o)}{P(M(D')=o)}\right)$  with probability  $P(M(D) = o)$ .
- **Theorem:** If privacy loss is bounded by  $\epsilon$  with probability  $\geq 1 - \delta$ , then the algorithm is  $(\epsilon, \delta)$  – DP

Converse? Proof

# Gaussian Mechanism

**Definition:** Let  $f: X \rightarrow R^k$ . The  $\ell_2$  – sensitivity of  $f$  is defined as

$$\Delta_2 = \sup_{D, D'} \|f(D) - f(D')\|_2$$

where the supremum is taken over all neighboring datasets  $D$  and  $D'$

**Recall:** Zero-mean Gaussian distribution with variance  $\sigma^2$ :  $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$

**Theorem:** Let  $f: X \rightarrow R^k$  have the  $\ell_2$  – sensitivity  $\Delta_2$  and

$$M(D) = f(D) + (Z_1, \dots, Z_k)$$

Where  $Z_1, \dots, Z_k$  are iid Gaussian with variance  $\sigma = \frac{\Delta_2 \sqrt{2 \ln \frac{1.25}{\delta}}}{\epsilon}$ . Then  $M(\cdot)$  is  $(\epsilon, \delta)$  – DP.

**Example:** *Mean estimation:* If data is bounded ( $\|x_i\|_2 \leq c$ ), the sensitivity of the mean (assuming neighboring datasets defined by replacement) is  $\Delta_2 = 2c/n$

# Properties of approximate DP

**Post-Processing:** Let  $M: X \rightarrow O$  be  $(\epsilon, \delta)$  - DP and let  $G: O \rightarrow T$  be an arbitrary (potentially randomized) mapping. Then,  $G(M(\cdot))$  is also  $(\epsilon, \delta)$  - DP.

Use cases: Integer optimization/decisions, projections involved, etc.

**Group Privacy:** Let  $M: X \rightarrow O$  be  $(\epsilon, \delta)$  - DP and let  $D$  and  $D'$  be **two datasets that differ in  $k$  entries**. Then,

for any  $\Omega \in O$ , we have  $\Pr(M(D) \in \Omega) \leq e^{k\epsilon} \Pr(M(D') \in \Omega) + \delta \frac{e^{k\epsilon} - 1}{e^\epsilon - 1}$ .

Proof?

# Remarks on composition

**Basic Adaptive Composition:** Let  $M = (M_1, \dots, M_T)$  be a sequence of algorithms where  $M_i$  is  $(\epsilon_i, \delta_i)$  - DP.

The algorithms may be chosen adaptively. Then,  $M(\cdot)$  is  $(\sum_i \epsilon_i, \sum_i \delta_i)$  - DP.

**Advanced Composition:** Let  $M = (M_1, \dots, M_T)$  be a sequence of  $(\epsilon, \delta)$  - DP algorithms (may be chosen

adaptively). Then, for any  $\delta' > 0$ ,  $M(\cdot)$  is  $(\epsilon', T\delta + \delta')$  - DP where  $\epsilon' = \epsilon \sqrt{2T \ln \left( \frac{1}{\delta'} \right)} + \frac{T\epsilon(e^\epsilon - 1)}{e^\epsilon + 1}$ .

- See [Kairouz et al., 2015] for slightly tighter results that also hold when we have  $(\epsilon_i, \delta_i)$ -DP
- BUT these composition results are typically not tight in practice (Why?)
- Some variants of  $(\epsilon, \delta)$ -DP, such as Rényi DP [Mironov, 2017] and zero-concentrated DP (zCDP) [Bun and Steinke, 2016], can enable tighter bounds for specific mechanisms (such as Gaussian mechanism)

# Applying DP to practical ML models

- Privatizing the model
  - Privatizing training data
  - **DP training/optimization**
  - Synthetic data
  - ...
  
- Privatizing the output



# Differentially private optimization

- Assume we want to train a model by solve

$$\min_w \left( L(\mathbf{w}, \mathbf{X}) \triangleq \frac{1}{n} \sum_{i=1}^n \ell(\mathbf{w}, \mathbf{x}_i) + R(\mathbf{w}) \right)$$

- Our algorithm returns  $w^*$  that may reveal sensitive data.
- How can we solve this optimization problem in a DP fashion?
  - Output perturbation
  - Exponential mechanism
  - Objective perturbation
  - Privatizing the algorithm: DP-SGD

# DP optimization via output perturbation

- Consider the ERM setting

$$\min_w \left( L(\mathbf{w}, \mathbf{X}) \triangleq \frac{1}{n} \sum_{i=1}^n \ell(\mathbf{w}, \mathbf{x}_i) + R(\mathbf{w}) \right)$$

- Output perturbation:  $w_{priv} = w_R^* + z$

- By optimizing the regularizer, we obtain the utility bound

# Output Perturbation: Utility Bound

- Goal:  $\min_w \left( L(w) = \frac{1}{n} \sum_{i=1}^n \ell(w, x_i) \right)$

**Theorem:** Let  $\ell(\cdot, x)$  be  $L$  – Lipschitz and convex and  $\text{diam}(W) = D$ . Then, there exists  $\mu > 0$  s.t.

$$E[L(w_{priv})] - \min_w L(w) = \tilde{O} \left( LD \left( \frac{\sqrt{d}}{\epsilon n} \right)^{1/2} \right)$$

- We ignored  $\log \left( \frac{1}{\delta} \right)$  term
- Trivial algorithm provides the bound  $L(w_{priv}) - \min_w L(w) \leq LD$
- Excess risk for output perturbation in the convex setting:  $\tilde{O} \left( LD \min \left\{ 1, \left( \frac{\sqrt{d}}{\epsilon n} \right)^2 \right\} \right)$

# Limitations of output perturbation

- Only applicable to strongly convex setting or problems with similar behaviors (why?)
- Requires Lipschitz loss functions
- We may want to maintain privacy during training
- Another idea: instead of adding noise to the solution, can we add noise to the objective?

# Objective perturbation

- Introduced by Chaudhuri, Monteleoni, and Sarwate (2011).
- Idea: Instead of adding noise to the solution, we perturb the goal of the training itself
- Standard ERM Objective:  $J(w, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n \ell(w, x_i) + R(w)$
- Perturbed ERM Objective:  $\tilde{J}(w, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n \ell(w, x_i) + R(w) + \frac{1}{n} b^T w$  where  $b \sim N(0, \sigma^2 I)$
- Added noise effectively “tilts” the loss landscape
- Excess risk for objective perturbation in the convex setting:  $\tilde{O} \left( LD \min \left\{ 1, \frac{\sqrt{d}}{\epsilon n} \right\} \right)$

# Limitations of objective perturbation

- Requires strong convexity

- Is the strong convexity necessary?

- Example: median estimation  $J(w, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^n |w - x_i|$

- $\mathcal{D} = \{0, 100\}$  and  $\mathcal{D} = \{0, 101\}$



With probability 1/2  
catastrophic privacy failure

- Requires Lipschitz loss functions

- We may want to maintain privacy during training

- Another idea: instead of perturbing the solution or objective, **perturb the steps of the algorithm**

# Perturbing the algorithm:

➤ Training Goal:  $\min_{w \in W} \left( L(w) = \frac{1}{n} \sum_{i=1}^n \ell(w, x_i) \right)$

➤ (Projected) gradient descent algorithm

For  $t = 0, 1, \dots, T - 1$ :

Update  $w^{t+1} \leftarrow Proj_W(w^t - \alpha_t \nabla L(w^t))$

Return  $w^T$

➤ The only part that depends on data is the gradient. Let's privatize the gradient

For  $t = 0, 1, \dots, T - 1$ :

Draw a random variable  $z^t \sim N(0, \sigma^2 I)$

Update  $w^{t+1} \leftarrow Proj_W(w^t - \alpha_t (\nabla L(w^t) + z^t))$

Return  $w^T$

# Privacy analysis: how much noise should we add?

**Recall:** Let  $M = (M_1, \dots, M_T)$  be a sequence of  $(\epsilon', \delta')$  – DP algorithms (may be chosen adaptively). To

guarantee target privacy level  $(\epsilon, T\delta' + \delta_0)$  with  $0 < \epsilon < 1$  and  $\delta_0 > 0$ , it suffices to choose  $\epsilon' \leq \frac{\epsilon}{\sqrt{8T \ln(1/\delta_0)}}$

➤ Each step of the DP-GD algorithm should be  $(\epsilon', \delta')$  – DP

$$\text{with } \epsilon' = \frac{\epsilon}{\sqrt{8T \log\left(\frac{1}{\delta}\right)}} \text{ and } \delta' = \frac{\delta}{T+1}.$$

For  $t = 0, 1, \dots, T - 1$ :

Draw a random variable  $z^t \sim N(0, \sigma^2 I)$

Update  $w^{t+1} \leftarrow \text{Proj}_W(w^t - \alpha_t(\nabla L(w^t) + z^t))$

Return  $w^T$

➤ **Recall** post-processing and Gaussian mechanism: we need a noise with  $\sigma = \frac{\Delta_2}{\epsilon'} \sqrt{2 \log(1.25/\delta')}$

➤ Sensitivity of  $\nabla L(w^t) \rightarrow \Delta_2 = \frac{2L}{n}$

➤ Noise variance:  $\sigma = \frac{8L\sqrt{T}}{\epsilon n} \left( \log\left(\frac{1.25}{\delta}\right) + \log(T + 1) \right)$

How close to optimal solution can we get?

# DP-GD: excess risk

**Lemma [Shamir & Zhang 2013]:** Consider  $\min_{w \in W} F(w)$  with convex  $F(\cdot)$  and bounded closed convex domain with  $\text{diam}(W) \leq D$ . Assume we run SGD  $w^{t+1} \leftarrow \text{Proj}(w^t - \eta_t g(w^t))$  where  $E[\|g(w^t)\|^2] \leq G^2$ ,  $E[g(w^t)] \in \partial F(w^t)$ , and  $\eta_t = \frac{c}{\sqrt{t}}$  with  $c > 0$ . Then, for any iteration  $T$ , we have  $E[F(w^T)] - F(w^*) \leq \left(\frac{D^2}{c} + cG^2\right) (2 + \log T) / \sqrt{T}$

- Optimizing  $c$ , we get  $E[F(w^T)] - F(w^*) \leq 2DG(2 + \log t) / \sqrt{T}$
- Let's use this lemma in our DP-GD context:
- In DP-GD, we have  $G^2 \leq L^2 + d\sigma^2$
- From privacy analysis:  $\sigma = \frac{8L\sqrt{T}}{\epsilon n} (\log\left(\frac{1.25}{\delta}\right) + \log(T + 1))$
- Optimizing  $T$ , for DP-GD, we obtain  $E[L(w^T)] - \min_w L(w) = \tilde{O}\left(\frac{LD\sqrt{d}}{\epsilon n}\right)$

# DP-GD

➤ Goal:  $\min_w \left( L(w) = \frac{1}{n} \sum_{i=1}^n \ell(w, x_i) \right)$

➤ DP-GD Algorithm:

For  $t = 0, 1, \dots, T - 1$ :

Draw a random variable  $z^t \sim N(0, \sigma^2 I)$

Update  $w^{t+1} \leftarrow \text{Proj}_W(w^t - \alpha_t(\nabla L(w^t) + z^t))$

Return  $w^T$

**Theorem:** Let  $\ell(\cdot, x)$  be  $L$  – Lipschitz and convex and  $\text{diam}(W) = D$ . Assume  $\epsilon < 1$ . Then, by choosing  $T, \alpha_t, \sigma^2$  appropriately,  $w^T$  is  $(\epsilon, \delta)$  – DP and

$$E[L(w^T)] - \min_w L(w) = \tilde{O}\left(\frac{LD\sqrt{d}}{\epsilon n}\right)$$

- We ignored  $\log\left(\frac{1}{\delta}\right)$  and other logarithmic terms
- Trivial algorithm provides the bound  $L(w_{\text{priv}}) - \min_w L(w) \leq LD$
- Excess risk for DP-GD in the convex setting:  $\tilde{O}\left(LD \min\left\{1, \frac{\sqrt{d}}{\epsilon n}\right\}\right)$

# From DP-GD to DP-SGD

➤ Goal:  $\min_w \left( L(w) = \frac{1}{n} \sum_{i=1}^n \ell(w, x_i) \right)$

- Computing the entire gradient can be costly
- How about if we add noise to the gradient obtained from a batch

## DP-GD Algorithm:

For  $t = 0, 1, \dots, T - 1$ :

Draw a random variable  $z^t \sim N(0, \sigma^2 I)$

Update  $w^{t+1} \leftarrow \text{Proj}_W(w^t - \alpha_t (\nabla L(w^t) + z^t))$

Return  $w^T$

## DP-SGD Algorithm:

For  $t = 0, 1, \dots, T - 1$ :

Draw a random variable  $z^t \sim N(0, \sigma^2 I)$

Draw a batch of data  $B$  and let  $g^t = \frac{1}{|B|} \sum_{i \in B} \nabla \ell(w, x_i)$

Update  $w^{t+1} \leftarrow \text{Proj}_W(w^t - \alpha_t (g^t + z^t))$

Return  $w^T$

- Which parts of the analysis would change?

# DP-GD to DP-SGD: Privacy amplification by subsampling

**Lemma [Balle et al. 2018]:** Let  $S: X^n \rightarrow X^m$  be a sampling mechanism that samples  $m$  samples out of  $n$  samples uniformly at random without replacement. Let  $A$  be an  $(\epsilon, \delta)$  – DP mechanism. Then the mechanism  $A(S(\cdot))$  is  $\left(\ln\left(1 + \frac{m}{n}(e^\epsilon - 1)\right), \frac{m}{n}\delta\right)$  – DP.

- The amplification is because **the sampling procedure has randomness and is secret**.
- When  $\epsilon < 1$ , we can use the approximation  $\ln\left(1 + \frac{m}{n}(e^\epsilon - 1)\right) \leq \frac{2m}{n}\epsilon$
- We can also get a similar result using Poisson subsampling

# Proof of Privacy Amplification By Subsampling

- Proof **assuming Poisson subsampling** with parameter  $\gamma$ 
  - Each data point is selected with probability  $\gamma$  independent of other samples
- Let  $D' = D \cup \{x\}$ . Then,

$$\begin{aligned}P(A(S(D')) \in \Omega) &= P(A(S(D')) \in \Omega \mid x \in S(D')) P(x \in S(D')) + P(A(S(D')) \in \Omega \mid x \notin S(D')) P(x \notin S(D')) \\ &\leq (e^\epsilon P(A(S(D)) \in \Omega) + \delta) \gamma + P(A(S(D)) \in \Omega) (1 - \gamma) \\ &\leq (1 - \gamma + \gamma e^\epsilon) P(A(S(D)) \in \Omega) + \gamma \delta\end{aligned}$$

- Similar argument for the opposite direction
- Exercise: do the analysis for privacy amplification for uniform sampling without replacement
  - Notion of neighboring datasets is different

# ERM: DP-SGD vs Output Perturbation

➤ So far, we studied approximate DP ( $(\epsilon, \delta)$  - DP)

➤ With output perturbation mechanism, we obtain  $E[L(w_{priv})] - \min_w L(w) = \tilde{O} \left( LD \min \left\{ 1, \left( \frac{\sqrt{d}}{\epsilon n} \right)^{\frac{1}{2}} \right\} \right)$

➤ With DP-SGD, we have  $E[L(w^T)] - \min_w L(w) = \tilde{O} \left( LD \min \left\{ 1, \frac{\sqrt{d}}{\epsilon n} \right\} \right)$

➤ Which bound is better?

➤ DP-SGD is almost tight in this case. Moreover, it can also be used in non-convex optimization problems

➤ How to do the analysis?

# Assumptions and extensions

➤ Goal:  $\min_w \left( L(w) = \frac{1}{n} \sum_{i=1}^n \ell(w, x_i) \right)$

➤ Do we need the loss function to be Lipschitz?

For  $t = 0, 1, \dots, T - 1$ :

Draw a mini-batch of data  $B_t$  from training data  $D$

Compute **clipped gradient**  $g_t = \frac{1}{|B_t|} \sum_{i \in B_t} \text{clip}(\nabla \ell(w, x_i), C)$

Draw a random variable  $z_t \sim N(0, \sigma^2 I)$

Privatize the (mini-batch) gradient  $\tilde{g}_t = g_t + z_t$

Update  $w_t = \text{Optimizer}(w_t, s_t, \tilde{g}_t)$

Return  $w^T$

SGD

$$g_t = \frac{1}{|B_t|} \sum_{i \in B_t} \nabla \ell(w, x_i)$$

➤ Extension to other (gradient-based) optimizers is straightforward

➤ DP-Adam, DP-AdamW, DP-SVRG, DP-Adafactor, ...

# Discussions on DP-SGD

➤ Goal:  $\min_w \left( L(w) = \frac{1}{n} \sum_{i=1}^n \ell(w, x_i) \right)$

- No assumption on the convexity/strong convexity
- No assumption on Lipschitzness of the loss function → Practical
- Drawback
  - Relies on (advanced) composition theorems (which are loose)
  - How can we fix the issue?
  - Why are composition theorems are loose in practice?

# Analysis of PLRV for Composition

- **Recall: Privacy Loss Random Variable (PLRV):**  $L = \log \left( \frac{P(M(D)=o)}{P(M(D')=o)} \right)$  with probability  $P(M(D) = o)$
- **Connection to DP:**  $M(\cdot)$  is  $(\epsilon, \delta)$ -DP if  $P(L > \epsilon) \leq \delta$
- **Composition as a summation:**
  - The total privacy loss of the composed mechanism:  $L_{tot} = \sum_{i=1}^T L_i$
- In advanced composition, we used Chernoff bound to bound the tail of the distribution of  $L_{tot}$
- Can we do better?
  - We can numerically compute the probability of the tail → Numerical Accountant

# Numerical Accountant and the Fourier Accountant

- Need to compute the distribution of  $L_{tot} = \sum_{i=1}^T L_i$
- Therefore, we need to (Numerically) compute the convolution of the distributions
- Assuming discretization with  $k$  bins, it requires  $O(k^2)$  computational complexity
- **The Fourier Accountant (Koskela et al., 2020):**
  - **Key Idea:** Convolution in the time/loss domain is multiplication in the frequency domain.
  - **Workflow:**
    - Discretize the PLRV distribution of a single step.
    - Use Fast Fourier Transform (FFT).
    - Compose by element-wise multiplication in the frequency domain.
    - Inverse FFT to get the composed PLRV distribution.
- Complexity of Fourier Transformation:  $O(k \log k)$

① DP-FTRL / Matrix Mechanism / Tree aggregation

② Densifying

---

DP-SGD

Assume batch size = 1

The amount of noise added =  $\begin{cases} O(\frac{\sqrt{T}}{\epsilon}) \\ O(\frac{1}{\epsilon}) \end{cases}$

SGD output

$w_1$

$$w_2 = w_1 - \eta g_1$$

$$w_3 = w_2 - \eta \tilde{g}_2 = w_1 - \eta g_1 - \eta g_2$$

$$w_4 = \dots = w_1 - \eta g_1 - \eta g_2 - \eta g_3$$

$\vdots$

$$w_t = \dots = w_1 - \eta \sum_{i=1}^t g_i$$

For Privatizing SGD, we need to privatize

$$g_1$$

$$g_1 + g_2$$

$$g_1 + g_2 + g_3$$

$$g_1 + g_2 + g_3 + g_4$$

⋮

Batch size = 1

$$d = 1$$

$$\begin{bmatrix} g_1 \\ g_1 + g_2 \\ g_1 + g_2 + g_3 \\ g_1 + g_2 + g_3 + g_4 \\ \vdots \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}}_A \underbrace{\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ \vdots \end{bmatrix}}_x$$

Privatize  $Ax$

Solution 1  $\Delta(A) = \|Ax - Ax'\|_2 = O(\sqrt{T})$

$$A = BW$$

We want to compute  $Ax = BWx$

So we can privatize by computing  $B(Wx + z_w)$

How much noise I should add?

What is the accuracy of my entire privatized procedure?

$$\sigma \approx \Delta_2(w)$$

what is the accuracy of my privatization:

$\sim$  privatize version of Ax

$$Ax = B(Wx + z_w) = BWx + Bz_w = \underbrace{Ax} + \underbrace{Bz_w}_{\text{Added noise}}$$

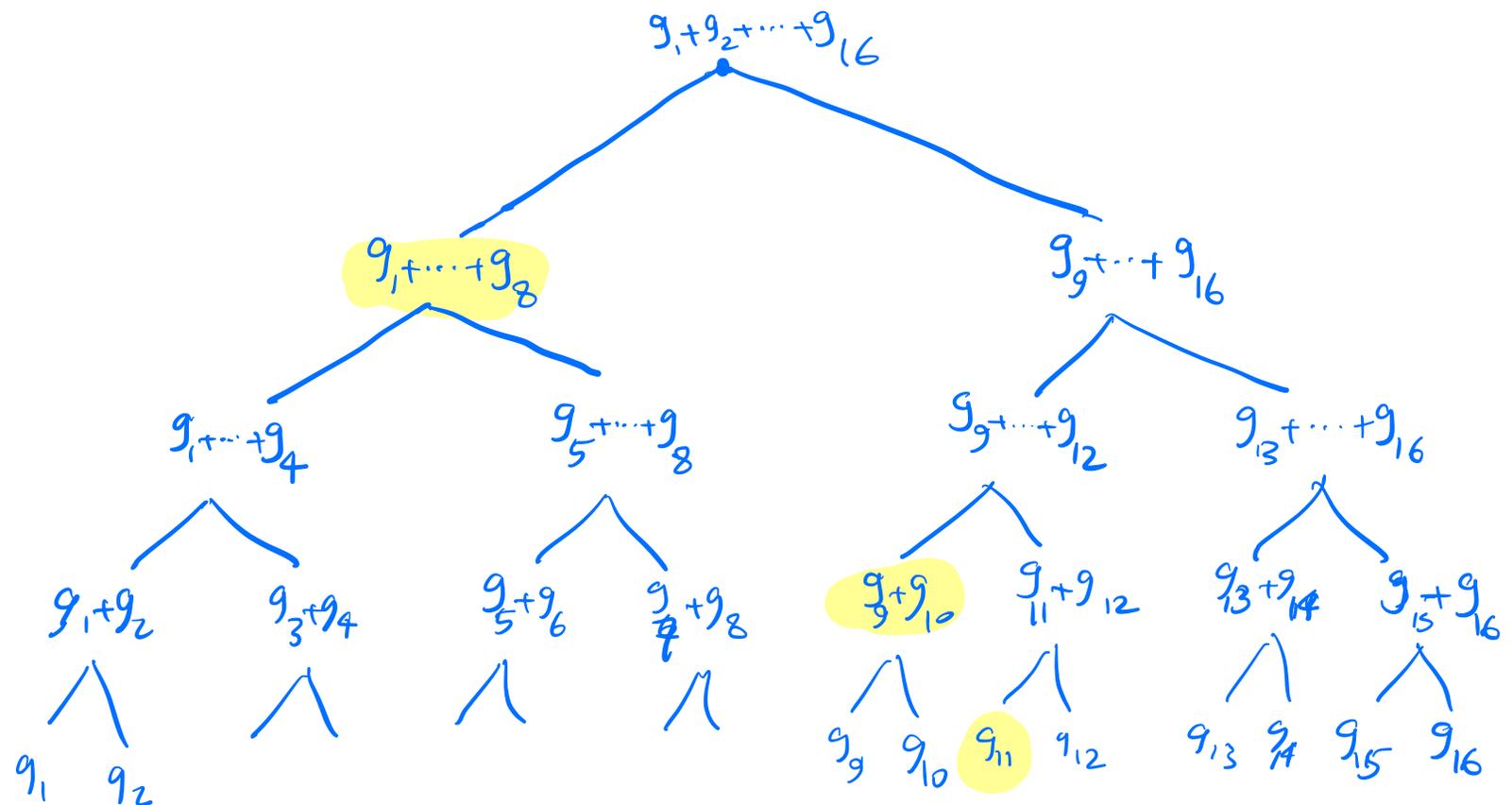
$$\text{error} \in \mathbb{E}[\|Ax - \tilde{Ax}\|^2]$$

$$\text{error} = \sigma^2 \text{Tr}(BB^T) = \|W\|_2^2 \text{Tr}(BB^T)$$

$$\min \|W\|_2^2 \text{Tr}(BB^T)$$

$$\text{s.t. } A = BW$$

Tree aggregation  $T=16$



$\sum_{i=1}^t g_i$  can be written as sum of values in  $\log(T)$  # of nodes at most

$$g_1 + \dots + g_n$$

Algorithm

Generate noise  $N(0, \sigma^2)$  for all the nodes in the tree

Compute privatized gradients by clipping gradient + adding precomputed noise of the tree nodes

$$w_{11} = w_1 - \eta \sum_{i=1}^{11} g_i = w_1 - \eta \left( \left( \sum_{i=1}^8 g_i + z_{1:8} \right) + \left( \sum_{i=9}^{10} g_i + z_{9:10} \right) + (g_{11} + z_{11}) \right)$$

$$\text{DP-SGD: } \theta_{t+1} = \theta_t - \eta (g_t + z_t) \quad \leftarrow \text{independent noise}$$

$$\text{DP-FTRL: } \theta_{t+1} = \theta_t - \eta \left( \tilde{s}_t - \tilde{s}_{t-1} \right) = \theta_t - \eta \tilde{z}_t$$

$\swarrow$                        $\searrow$   
 Privatize estimate      Privatized estimate  
 of  $g_1 + \dots + g_t$       of  $g_1 + \dots + g_{t-1}$

$$\begin{aligned} \theta_{11} &= \theta_1 - \eta \text{Privat} \left( \sum_{i=1}^{11} g_i \right) \\ &= \theta_1 - \eta \left( \left( \sum_{i=1}^8 g_i + z_{1:8} \right) + \left( \sum_{i=9}^{10} g_i + z_{9:10} \right) \right) \end{aligned}$$

$$\begin{aligned} \theta_{12} &= \theta_1 - \eta \text{Privat} \left( \sum_{i=1}^{11} g_i \right) = \\ &= \theta_1 - \eta \left( \left( \sum_{i=1}^8 g_i + z_{1:8} \right) + \left( \sum_{i=9}^{10} g_i + z_{9:10} \right) + (g_{11} + z_{11}) \right) \end{aligned}$$

$$\theta_{12} = \theta_{11} - \eta (g_{11} + z_{11})$$

$$\begin{aligned} \theta_{13} &= \theta_{11} - \eta \text{Private} \left( \sum_{i=1}^{12} g_i \right) \\ &= \theta_{11} - \eta \left( \left( \sum_{i=1}^8 g_i + z_{1:8} \right) + \left( \sum_{i=9}^{12} g_i + z_{9:12} \right) \right) \end{aligned}$$

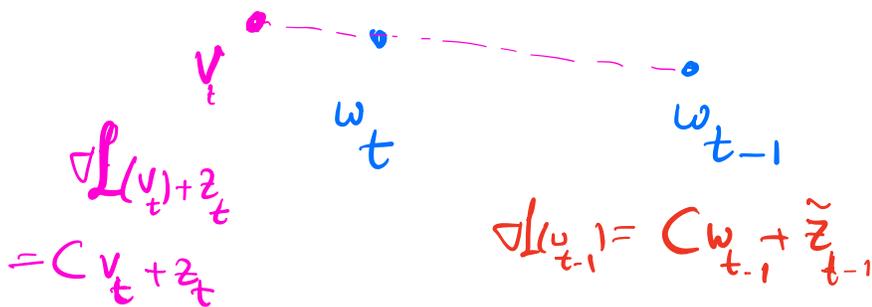
$$\theta_{13} = \theta_{12} - \eta \left( g_{12} + \underbrace{z_{9:12} - z_{9:10} - z_{11}}_{\tilde{z}_{13}} \right)$$

Denoising

$L(w)$  quadratic

$\nabla L(w)$  linear

$$\nabla L(w) = Cw$$



$$k(C v_t + z_t) + (1-k)(C w_{t-1} + \tilde{z}_{t-1}) = C w_t + \text{noise}$$