Private Stochastic Convex Optimization: Optimal Rates in L_1 Geometry

Hilal Asi
Stanford University

Joint work with:

Vitaly Feldman
Apple

Tomer Koren
Tel Aviv University

Kunal Talwar
Apple

ICML 2021

Stochastic Convex Optimization (SCO)

Samples $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ where $S_i \sim P$

Convex Parameter Space $\mathcal{X} \subseteq \mathbb{R}^d$

Convex loss function $f(x; S) : \mathcal{X} \times \mathcal{S} \to \mathbb{R}$

Population loss $f(x) = \mathbb{E}_{S\sim P}[f(x; S)]$

Goal: find a solution $\hat{x} \in \mathcal{X}$ that minimizes

Excess population risk $$f(\hat{x}) - \min_{x\in\mathcal{X}} f(x)$$

Stochastic Convex Optimization (SCO)

Goal: find a solution $\hat{x} \in \mathcal{X}$ that minimizes

Excess population risk $\quad f(\hat{x}) - \min_{x \in \mathcal{X}} f(x)$

Problem is well-understood

$\mathcal{X}$ is unit ℓ_2 ball

f is 1-Lipschitz

Optimal risk = $\frac{1}{\sqrt{n}}$

$\mathcal{X}$ is unit ℓ_1 ball

f is 1-Lipschitz

wrt ℓ_1 norm $\; f(x) - f(y) \leq \|x - y\|_1$

Optimal risk = $\sqrt{\frac{\log d}{n}}$

Differentially Private Stochastic Convex Optimization (DP-SCO)

Goal: find a solution $\hat{x} \in \mathcal{X}$ that minimizes

Excess population risk $f(\hat{x}) - \min_{x \in \mathcal{X}} f(x)$

Additional constraint: algorithm is (ε, δ)-differentially private

Problem is (relatively) well-understood in ℓ_2-Geometry [BFTT19, FKT20]

$\mathcal{X}$ is unit ℓ_2 ball

f is 1-Lipschitz

Optimal private risk = $\dfrac{1}{\sqrt{n}} + \dfrac{\sqrt{d}}{n\varepsilon}$

This work: what about other geometries?

Private Optimization in ℓ_1-Geometry

This work: DP-SCO in ℓ_1-Geometry

$\mathcal{X}$ is unit ℓ_1 ball

f is 1-Lipschitz $\quad f(x) - f(y) \leq \|x - y\|_1$

Previous work: [JT14, TTZ15] for empirical loss $\quad f_{\mathcal{S}}(x) = \frac{1}{n}\sum_{i=1}^{n} f(x; S_i)$

Empirical risk: $\left(\frac{\text{poly}(\log d)}{n\varepsilon}\right)^{2/3}$

Population risk: $\sqrt{\frac{d}{n}} + \left(\frac{\text{poly}(\log d)}{n\varepsilon}\right)^{2/3}$

Our contributions

1. Optimal rates for DP-SCO in ℓ_1-geometry (with tight lower bounds)

Non-smooth functions: $\sqrt{\frac{\log d}{n}} + \frac{\sqrt{d}}{n\varepsilon}$

smoothness helps in ℓ_1 geometry

Smooth functions: $\sqrt{\frac{\log d}{n}} + \left(\frac{\text{poly}(\log d)}{n\varepsilon}\right)^{2/3}$

Privacy for free even when

$d \gg n$ and $\varepsilon \approx \frac{1}{n^{1/4}}$

2. Optimal rates for DP-SCO in ℓ_p-geometry with $p \in (1,2]$

Non-smooth functions: $\frac{1}{\sqrt{n}} + \frac{\sqrt{d}}{n\varepsilon}$

tight lower bounds from [BGN21]

3. Faster runtime for non-smooth functions in ℓ_2-Geometry

[FKT20]: $O(n^2)$

Our algorithms: $O(n^{3/2})$

Comparison to [BGN21]

1. Optimal rates for DP-SCO in ℓ_1-geometry (with tight lower bounds)

Non-smooth functions: $\sqrt{\frac{\log d}{n}} + \frac{\sqrt{d}}{n\varepsilon}$

Smooth functions: $\sqrt{\frac{\log d}{n}} + \left(\frac{\text{poly}(\log d)}{n\varepsilon}\right)^{2/3}$ [BGN21] $\frac{\log d}{\sqrt{n\varepsilon}}$

2. Optimal rates for DP-SCO in ℓ_p-geometry with $p \in (1,2]$

Non-smooth functions: $\frac{1}{\sqrt{n}} + \frac{\sqrt{d}}{n\varepsilon}$ [BGN21] $\frac{\sqrt{d}}{n^{3/4}\varepsilon}$

Main techniques

Non-smooth case

- Reduction from DP-SCO to strongly convex DP-ERM
- Solve DP-ERM in ℓ_1 geometry using noisy mirror descent

Smooth case

- Private variance-reduced Frank-Wolfe algorithm
- Binary tree allocation of the samples for variance-reduction

Algorithm for Non-Smooth Functions

Two main ingredients

1. Reduction from DP-SCO to strongly convex DP-ERM
2. Solve DP-ERM using noisy mirror descent

Reduction from DP-SCO to DP-ERM

DP-SCO

minimize the population loss $f(x) = \mathbb{E}_{S \sim P}[f(x; S)]$

DP-ERM

minimize the empirical loss $f(x) = \frac{1}{n}\sum_{i=1}^{n} f(x; S_i)$

Optimal algorithms for strongly convex DP-ERM give optimal algorithms for DP-SCO

Reduction from DP-SCO to DP-ERM

Based on iterative-localization [FKT20]

[FKT20] use localization to reduce DP-SCO to stable-ERM

Gives optimal rates for ℓ_2 geometry

Not sufficient for ℓ_1 geometry

Idea:

1. At each iteration, privately solve a regularized ERM problem
2. As the output is accurate, shrink diameter and repeat

Reduction from DP-SCO to DP-ERM

Idea:

1. At each iteration, privately solve a regularized ERM problem
2. As the output is accurate, increase regularization and repeat

Algorithm (sketch)

1. Initialize $x_0 = \mathbf{0}$
2. For $k = 1$ to $\log n$
   - Find x_{k+1} by privately solve the ERM problem: $$\frac{1}{n}\sum_{i=1}^{n} f(x; S_i) + \lambda \|x - x_{k-1}\|^2$$
   - Increase regularization λ by a factor of 2 (shrinks diameter)

Reduction from DP-SCO to DP-ERM

Algorithm (sketch)

1. Initialize $x_0 = \mathbf{0}$
2. For $k = 1$ to $\log n$
   - Find x_{k+1} by privately solve the ERM problem: $\frac{1}{n}\sum_{i=1}^{n} f(x; S_i) + \lambda \|x - x_{k-1}\|^2$
   - Increase regularization λ by a factor of 2 (shrinks diameter)

Main claim (informal)

If algorithm A solves λ-strongly convex DP-ERM with rate $\frac{1}{\lambda n} + \frac{d}{\lambda n^2 \varepsilon^2}$

then the above algorithm has population loss $\frac{1}{\sqrt{n}} + \frac{\sqrt{d}}{n\varepsilon}$

Noisy Mirror Descent for DP-ERM

Noisy Mirror Descent

1. Initialize $x_0 = \mathbf{0}$

2. For $t = 1$ to T

   - Add noise to gradient: $\hat{g}_t = \nabla_x f(x_t; S_t) + \mathcal{N}(0, \sigma^2 \mathbb{I}_d)$

   - Apply mirror descent step: $x_{t+1} = \arg\min \{ \langle \hat{g}_t, x \rangle + \frac{1}{\eta} D_h(x, x_t) \}$

Claim (informal)

Choosing h according to geometry, Noisy MD obtains excess loss $\dfrac{1}{\lambda n} + \dfrac{d}{\lambda n^2 \varepsilon^2}$

ℓ_1 geometry: use $\|x\|_p^2$ with $p = 1 + \dfrac{1}{\log d}$

ℓ_p geometry: use $\|x\|_p^2$ for $p > 1$

Algorithm for Smooth Functions

Main techniques

- Private variance-reduced Frank-Wolfe algorithm
- Exponential mechanism to apply Frank-Wolfe update (choose from d vertices)
- Binary tree allocation of the samples for variance-reduction

Frank-Wolfe Algorithm

Frank-Wolfe

For $t = 1$ to T:

1. $w_t = \arg\min_{x \in B_1} \langle \nabla f(x_t), x \rangle$

2. Set $x_{t+1} = (1-\eta)x_t + \eta w_t$

Main observation [TTZ15]: the minimizer w_t is a vertex of the ℓ_1 ball

Use Exponential mechanism to privately pick best vertex

Empirical risk [TTZ15]: $\left(\frac{\text{poly}(\log d)}{n\varepsilon}\right)^{2/3}$

What about population risk?

Even without privacy, FW achieves only $\frac{1}{n^{1/3}}$

Variance-Reduced Frank-Wolfe Algorithm [YCS19]

Variance-Reduced Frank-Wolfe (sketch)

- $v_0 = \nabla f(x_0; \mathcal{S}_0)$ where $\mathcal{S}_0$ is a set of n samples
- For $t = 1$ to T: $T \approx \sqrt{n}$

1. $v_t = v_{t-1} + \nabla f(x_t; \mathcal{S}_t) - \nabla f(x_{t-1}; \mathcal{S}_t)$ $|\mathcal{S}_k| \approx \sqrt{n}$
2. $w_t = \arg\min_{x \in \mathrm{B}_1} \langle v_t, x \rangle$
3. Set $x_{t+1} = (1 - \eta) x_t + \eta w_t$

Achieves optimal population risk [YCS19] $\frac{1}{\sqrt{n}}$

use it for DP-SCO?

Private Variance-Reduced Frank-Wolfe Algorithm

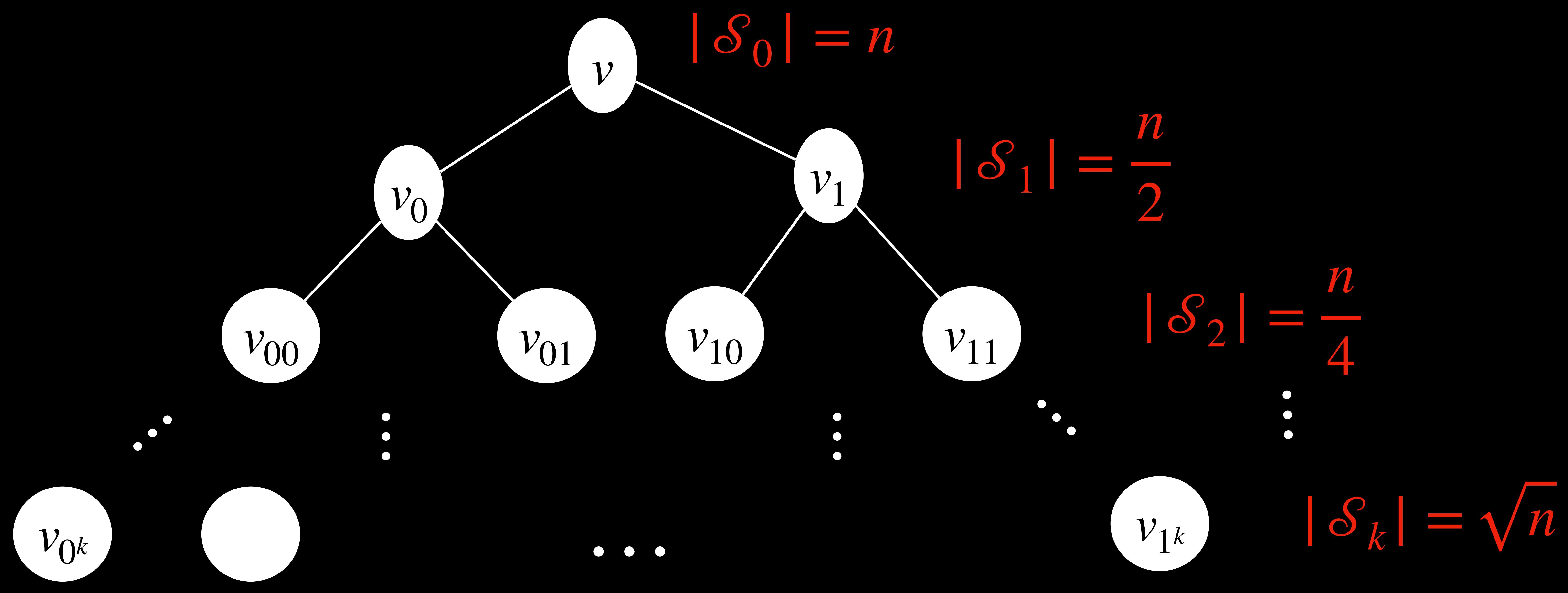
Attempt 1: add noise to privatize v_k

Problem: samples in $\mathcal{S}_1$ are used in $\sqrt{n}$ updates!

Results in sub-optimal bounds $\frac{\log d}{\sqrt{n\varepsilon}}$

Private Variance-Reduced Frank-Wolfe Algorithm

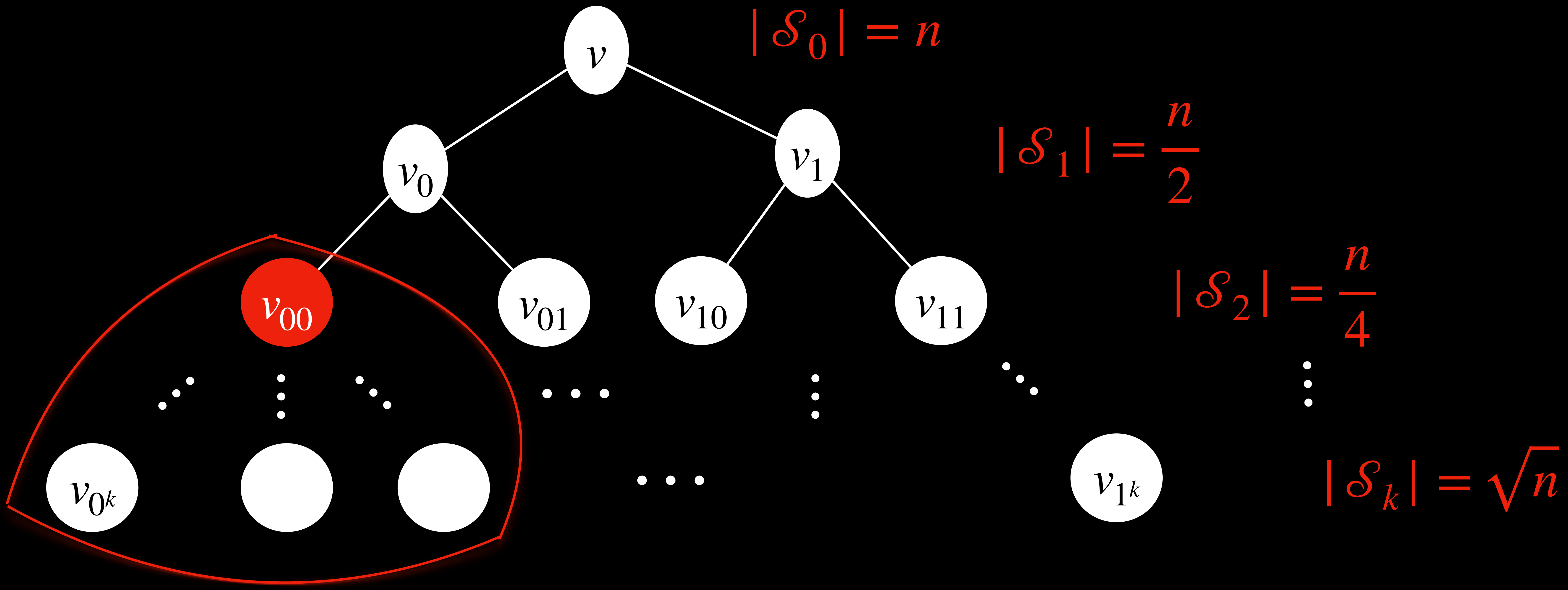
Main idea: allocate the samples so that smaller sets are used in less updates



Use parent's gradient to reduce variance at current vertex

$$v_{01} = v_0 + \nabla f(x_k; \mathcal{S}_{01}) - \nabla f(x_{x-1}; \mathcal{S}_{01})$$

Private Variance-Reduced Frank-Wolfe Algorithm



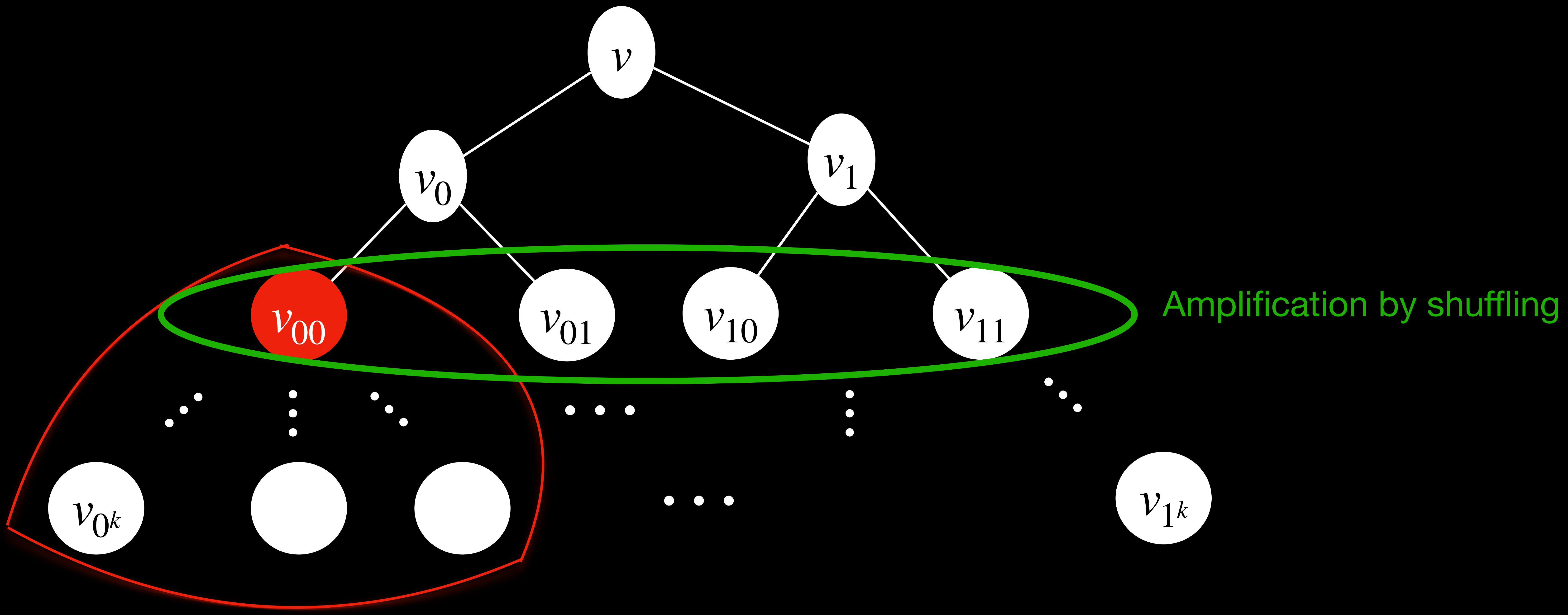
Use parent's gradient to reduce variance at current vertex

$$v_{01} = v_0 + \nabla f(x_k; \mathcal{S}_{01}) - \nabla f(x_{x-1}; \mathcal{S}_{01})$$

Apply FW step on v_k using exponential mechanism

How much noise to add?

Private Variance-Reduced Frank-Wolfe Algorithm



Private Variance-Reduced Frank-Wolfe achieves

Excess population risk $\sqrt{\frac{\log d}{n}} + \left(\frac{\text{poly}(\log d)}{n\varepsilon}\right)^{2/3}$

Open Problems

1. Linear $O(n)$ complexity for non-smooth DP-SCO?

2. Optimal rates for ℓ_p geometry with $p > 2$?

Thanks!