Contextual Stochastic Bilevel Optimization

Yifan Hu[1,2], **Jie Wang**[3], Yao Xie[3], Andreas Krause[2], Daniel Kuhn[1]

[1] EPFL, [2] ETH Zurich, [3] Gatech

Paper

Yifan's Homepage

Introduction

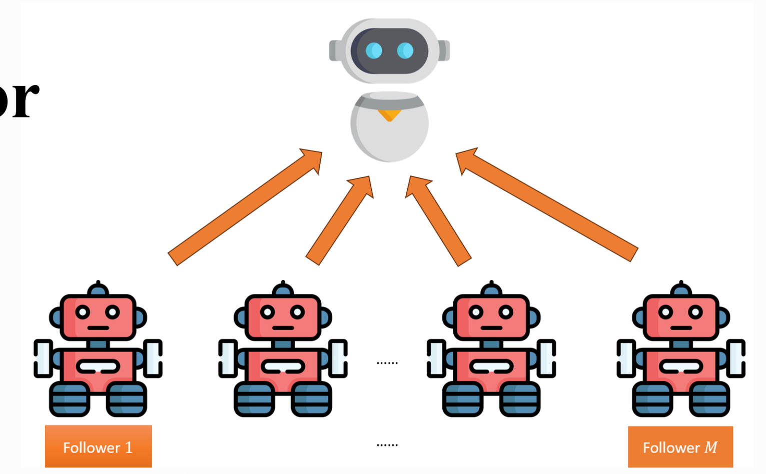
$$\min_{x\in\mathbb{R}^{d_x}} \quad F(x) := \mathbb{E}_{\xi\sim\mathbb{P}_\xi}[f(x, y^*(x;\xi); \xi)] \quad \text{(upper-level)}$$

$$\text{where} \quad y^*(x;\xi) := \operatorname*{argmin}_{y\in\mathbb{R}^{d_y}} \mathbb{E}_{\eta\sim\mathbb{P}_{\eta|\xi}}[g(x,y;\eta,\xi)] \quad \forall\, \xi \quad \text{(lower-level)}$$

- $\xi\sim\mathbb{P}_\xi$: contextual information; $\eta\sim\mathbb{P}_{\eta|\xi}$: conditional distributions
- lower-level: contextual stochastic optimization

Moviation

I: Find a shared model parameter for multiple similar tasks/individuals.

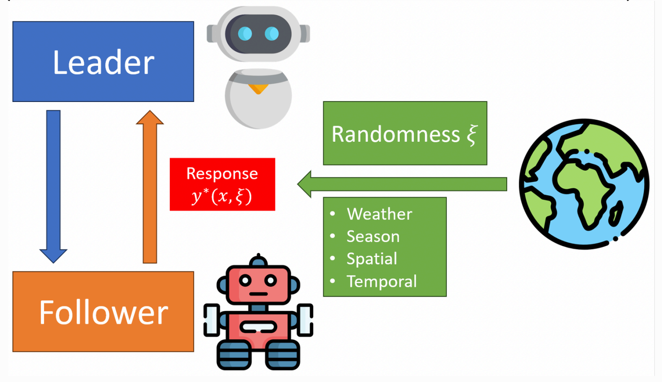


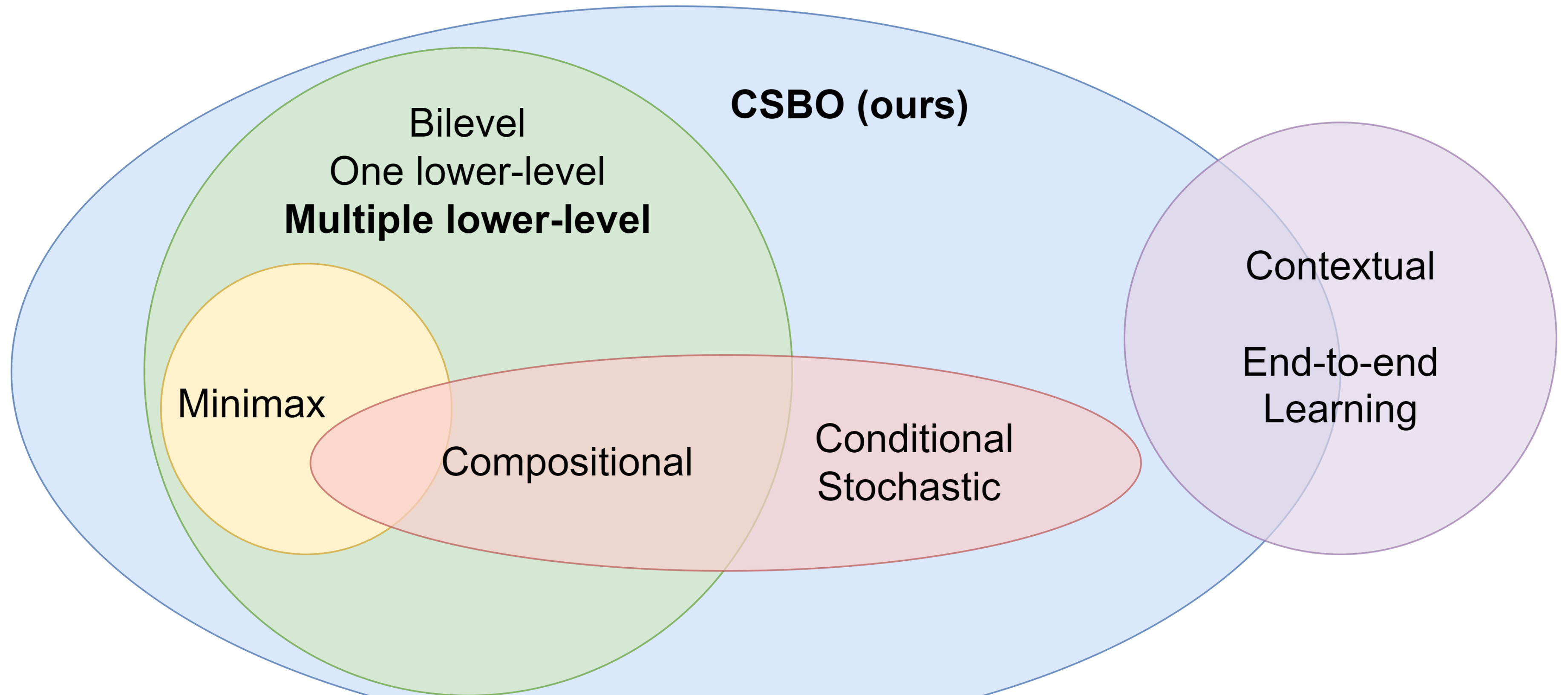
$$\min_x \quad \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_{\eta\sim P_i}\left[l(y_i^*(x);\eta)\right]$$

$$\text{where} \quad y_i^*(x) = \operatorname*{argmin}_{y_i} \mathbb{E}_{\eta\sim P_i}\left[l(y_i;\eta) + \frac{\lambda}{2}\|y_i - x\|^2\right], \forall i\in[M].$$

- Complexity depends linearly on M (Guo and Yang 2021)

II: Optimal Response to Side Information.





Algorithm Design

$$\nabla F(x) = \mathbb{E}\Big[\nabla_1 f(x, y^*(x;\xi);\eta,\xi) - \Big(\mathbb{E}_{\eta'\sim\mathbb{P}_{\eta|\xi}}\nabla_{12}^2 g(x, y^*(x;\xi);\eta',\xi)\Big)\times [\mathbb{E}_{\eta\sim\mathbb{P}_{\eta|\xi}}\nabla_{22}^2 g(x,y^*(x;\xi);\eta,\xi)]^{-1}\times\nabla_2 f(x, y^*(x;\xi);\eta,\xi)\Big].$$

Challenges:

- Estimate Hessian inverse
- Estimate the optimal response $y^*(x;\xi)$

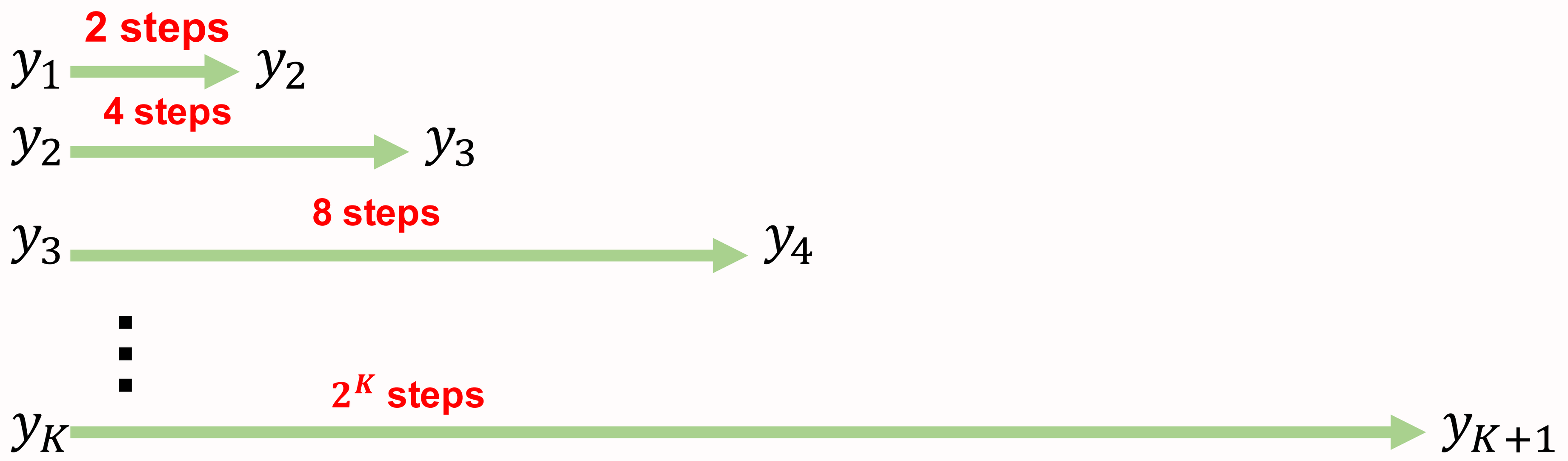
Matrix Inverse Estimation: Neumann Series

For random matrix A such that $0\prec A\prec I$:

$$[\mathbb{E}A]^{-1} = \sum_{i=0}^{\infty}(I-\mathbb{E}A)^i = \sum_{i=0}^{\infty}\prod_{n=1}^{i}\mathbb{E}(I-A_n) \approx \sum_{i=0}^{N}\prod_{n=1}^{i}\mathbb{E}(I-A_n).$$

- Bias: exponentially decreasing in N, i.e., $N=\mathcal{O}(\log(\epsilon^{-1}))$

Optimal Response Estimation: Epoch SGD



- Gradient estimator: $\widehat{v}(x; y_{K+1})$.
- **Con:** Need $\mathcal{O}(\epsilon^{-2})$ operations to get y_{K+1}.

Random Sampling Gradient Estimator (Hu et al. 2021)

$$\widehat{v}(x;y_{K+1}) = \widehat{v}(x;y_1) + \sum_{k=1}^{K}[\widehat{v}(x;y_{k+1}) - \widehat{v}(x;y_k)]$$

$$= \widehat{v}(x;y_1) + \sum_{k=1}^{K} p_k\frac{\widehat{v}(x;y_{k+1}) - \widehat{v}(x;y_k)}{p_k} = \mathbb{E}_{k\sim\mathbb{P}_k}\left[\widehat{v}(x;y_1) + \frac{\widehat{v}(x;y_{k+1}) - \widehat{v}(x;y_k)}{p_k}\right].$$

- Sample k according to pmf $p_k\propto 2^{-k}$, $\sum_{k=1}^{K} p_k = 1$. Construct estimator $\widehat{v}(x) = \widehat{v}(x;y_1) + \frac{\widehat{v}(x;y_{k+1}) - \widehat{v}(x;y_k)}{p_k}$.
- **High probability**: generate small k, **Low probability**: generate large k.
- **Per-iteration cost reduction**: from $\mathcal{O}(2^K) = \mathcal{O}(\epsilon^{-2})$ to $\mathcal{O}(K) = \widetilde{\mathcal{O}}(1)$.
- **Variance reduction** effect as $\widehat{v}(x;y_{k+1}) - \widehat{v}(x;y_k)\to 0$ for large k.

Takeaway

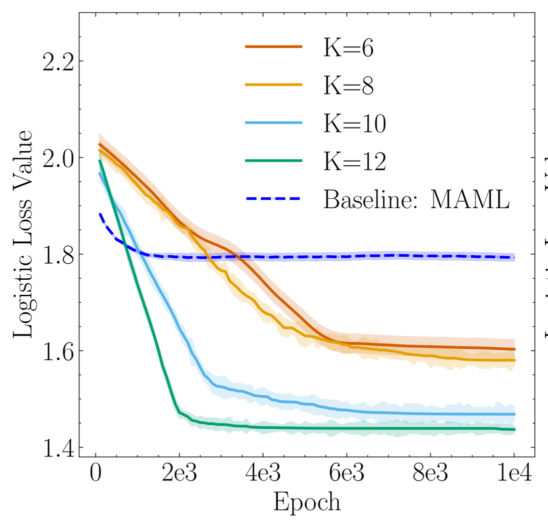
To find an ϵ-stationary point, the sample complexity is

- For vanilla SGD, it is $\widetilde{\mathcal{O}}(\epsilon^{-6})$.
- For random sampling method, it is $\widetilde{\mathcal{O}}(\epsilon^{-4})$.

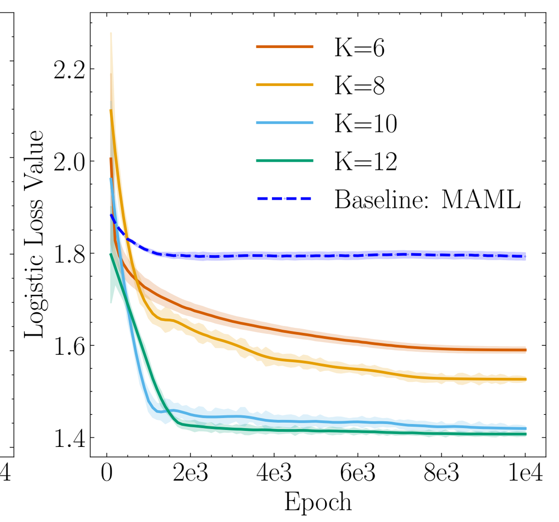
Remark: No dependence on number of tasks or individuals M.

Numerical Study: Meta-learning on Mini-ImageNet

Vanilla SGD



Random Sampling



Run Time Per Iteration

K	Vanilla	Random Sampling
6	**2.65e-2**	2.73e-2
8	7.23e-2	**3.41e-2**
10	2.48e-1	**4.93e-2**
12	9.38e-1	**1.08e-1**