Optimal randomized multilevel Monte Carlo estimators for repeatedly nested expectations

Guanyang Wang

Joint with: Yasa Syed
Department of Statistics
Rutgers University - New Brunswick
arxiv: 2301.04095 (ICML 2023)

June 6, 2023

Road map

- Set-up
- Current progress
- The algorithm
- Empirical results
- Discussion

Road map

- Set-up
- Current progress
- The algorithm
- Empirical results
- Discussion

Expectation, nested expectation, repeatedly nested expectation

- Expectation: $\mathbb{E}[g]$

Expectation, nested expectation, repeatedly nested expectation

- Expectation: $\mathbb{E}[g]$
  - Sample $X_1, X_2, \ldots, X_n$

Expectation, nested expectation, repeatedly nested expectation

- Expectation: $\mathbb{E}[g]$
  - Sample $X_1, X_2, \ldots, X_n$
  - Monte Carlo estimate: $\sum_{i=1}^{n} g(X_i)/n$

Expectation, nested expectation, repeatedly nested expectation

- Expectation: $\mathbb{E}[g]$
  - Sample $X_1, X_2, \ldots, X_n$
  - Monte Carlo estimate: $\sum_{i=1}^{n} g(X_i)/n$
- Nested expectation: $\mathbb{E}_x[g_0(\mathbb{E}_y[g_1(y) \mid x])]$

Expectation, nested expectation, repeatedly nested expectation

- Expectation: $\mathbb{E}[g]$
  - Sample $X_1, X_2, \ldots, X_n$
  - Monte Carlo estimate: $\sum_{i=1}^n g(X_i)/n$
- Nested expectation: $\mathbb{E}_x[g_0(\mathbb{E}_y[g_1(y) \mid x])]$
  - Expectation of a function of (conditional) expectation

Expectation, nested expectation, repeatedly nested expectation

- Expectation: $\mathbb{E}[g]$
  - Sample $X_1, X_2, \ldots, X_n$
  - Monte Carlo estimate: $\sum_{i=1}^{n} g(X_i)/n$
- Nested expectation: $\mathbb{E}_x[g_0(\mathbb{E}_y[g_1(y) \mid x])]$
  - Expectation of a function of (conditional) expectation
  - Linear g: the problem is equivalent to the standard Monte Carlo

Expectation, nested expectation, repeatedly nested expectation

- Expectation: $\mathbb{E}[g]$
  - Sample $X_1, X_2, \ldots, X_n$
  - Monte Carlo estimate: $\sum_{i=1}^{n} g(X_i)/n$
- Nested expectation: $\mathbb{E}_x[g_0(\mathbb{E}_y[g_1(y) \mid x])]$
  - Expectation of a function of (conditional) expectation
  - Linear g: the problem is equivalent to the standard Monte Carlo
  - Nonlinear g: much harder

Expectation, nested expectation, repeatedly nested expectation

- Expectation: $\mathbb{E}[g]$
    - Sample $X_1, X_2, \ldots, X_n$
    - Monte Carlo estimate: $\sum_{i=1}^{n} g(X_i)/n$
- Nested expectation: $\mathbb{E}_x[g_0(\mathbb{E}_y[g_1(y) \mid x])]$
    - Expectation of a function of (conditional) expectation
    - Linear g: the problem is equivalent to the standard Monte Carlo
    - Nonlinear g: much harder
- Repeatedly nested expectation: Today's talk

Set-up: an example

- Consider a process $(y^{(0)}, y^{(1)}, y^{(2)})$:

$$y^{(0)} \sim \mathrm{Norm}(\pi/2, 1),\ \ y^{(1)} \sim \mathrm{Norm}(y^{(0)}, 1),\ \ y^{(2)} \sim \mathrm{Norm}(y^{(1)}, 1).$$

Set-up: an example

- Consider a process $(y^{(0)}, y^{(1)}, y^{(2)})$:

$$y^{(0)} \sim \mathrm{Norm}(\pi/2, 1), \;\; y^{(1)} \sim \mathrm{Norm}(y^{(0)}, 1), \;\; y^{(2)} \sim \mathrm{Norm}(y^{(1)}, 1).$$

- Consider functions of this process:

$$\begin{aligned} g_0(y^{(0)}, z) &:= \sin\left(y^{(0)} + z\right) \\ g_1(y^{(0:1)}, z) &:= \sin\left(y^{(1)} - z\right) \\ g_2(y^{(0:2)}) &:= y^{(2)}. \end{aligned}$$

Set-up: an example

- Consider a process $(y^{(0)}, y^{(1)}, y^{(2)})$:

$$y^{(0)} \sim \mathrm{Norm}(\pi/2, 1),\ \ y^{(1)} \sim \mathrm{Norm}(y^{(0)}, 1),\ \ y^{(2)} \sim \mathrm{Norm}(y^{(1)}, 1).$$

- Consider functions of this process:

$$\begin{aligned} g_0(y^{(0)}, z) &:= \sin\left(y^{(0)} + z\right) \\ g_1(y^{(0:1)}, z) &:= \sin\left(y^{(1)} - z\right) \\ g_2(y^{(0:2)}) &:= y^{(2)}. \end{aligned}$$

- The goal is to estimate $\gamma_0 = \mathbb{E}\left[g_0\left(y^{(0)}, \gamma_1\left(y^{(0)}\right)\right)\right]$, where:

$$\begin{aligned} \gamma_1(y^{(0)}) &= \mathbb{E}\left[g_1\left(y^{(0:1)}, \gamma_2\left(y^{(0:1)}\right)\right) \mid y^{(0)}\right] \\ \gamma_2(y^{(0:1)}) &= \mathbb{E}\left[g_2\left(y^{(0:2)}\right) \mid y^{(0:1)}\right] \end{aligned}$$

Set-up: general setting (Rainforth et al. 2018)

- Fix $D > 0$ and real-valued functions $g_0, \ldots, g_D$, a process $(y^{(0)}, \ldots, y^{(D)})$
- Goal: estimate the repeatedly nested expectation (RNE):

$$\gamma_0 = \mathbb{E}\left[g_0\left(y^{(0)}, \gamma_1\left(y^{(0)}\right)\right)\right],$$

where for $d \in \{1, \ldots, D-1\}$, we have:

$$\gamma_d(y^{(0:d-1)}) = \mathbb{E}\left[g_d\left(y^{(0:d)}, \gamma_{d+1}\left(y^{(0:d)}\right)\right) \mid y^{(0:d-1)}\right],$$

and for $d = D$:

$$\gamma_D(y^{(0:D-1)}) = \mathbb{E}\left[g_D\left(y^{(0:D)}\right) \mid y^{(0:D-1)}\right].$$

RNE as a Russian Doll

Applications

- Optimal stopping: $g_d(y^{(0:d)}, u) := \max\{y^{(d)}, u\}$ for $0 \leq d \leq D-1$, and $g_D(y^{(0:D)}) := y^{(D)}$.
- $D = 2$: Risk estimation for the credit valuation adjustment
- $D = 1$: experimental design, portfolio risk management, stochastic and bilevel optimization
- Other applications: probabilistic programs, numerical PDEs, physics and chemistry
- Other name: nonlinear Monte Carlo

Road map

- Set-up of the problem
- Current progress
- The algorithm
- Empirical results
- Discussion

The nested Monte Carlo estimator

- Question: How will you estimate γ_0?

The nested Monte Carlo estimator

- Question: How will you estimate γ_0?
- Answer: One can first estimate γ_D, then plug-in this estimate to estimate γ_{D-1}, ...

The nested Monte Carlo estimator

- Question: How will you estimate γ_0?
- Answer: One can first estimate γ_D, then plug-in this estimate to estimate $\gamma_{D-1}, \ldots$
- Question: What is the cost to get an estimator with ϵ-error?

The nested Monte Carlo estimator

- Question: How will you estimate γ_0?
- Answer: One can first estimate γ_D, then plug-in this estimate to estimate $\gamma_{D-1}, \ldots$
- Question: What is the cost to get an estimator with ϵ-error?
- Hint: Standard Monte Carlo has cost $O(1/\epsilon^2)$.

The nested Monte Carlo estimator

- Question: How will you estimate γ_0?
- Answer: One can first estimate γ_D, then plug-in this estimate to estimate γ_{D-1}, ...
- Question: What is the cost to get an estimator with ϵ-error?
- Hint: Standard Monte Carlo has cost $O(1/\epsilon^2)$.
- Heuristic calculation:
  $O(1/\epsilon^2)$ for each d, therefore $O(1/\epsilon^2)^{D+1} = O(1/\epsilon^{2D+2})$ in total.

Current progress on the problem

- Rainforth et al. (2018): $O(1/\epsilon^{2D+2})$ or $O(1/\epsilon^{D+2})$ depending on conditions on $\{g_d\}$

Current progress on the problem

- Rainforth et al. (2018): $O(1/\epsilon^{2D+2})$ or $O(1/\epsilon^{D+2})$ depending on conditions on $\{g_d\}$
- Giles (2017): $O(1/\epsilon^2)$ when $D = 1$ and $\{g_0, g_1\}$ behaves 'nice'

Current progress on the problem

- Rainforth et al. (2018): $O(1/\epsilon^{2D+2})$ or $O(1/\epsilon^{D+2})$ depending on conditions on $\{g_d\}$
- Giles (2017): $O(1/\epsilon^2)$ when $D = 1$ and $\{g_0, g_1\}$ behaves 'nice'
- Our algorithm:

Current progress on the problem

- Rainforth et al. (2018): $O(1/\epsilon^{2D+2})$ or $O(1/\epsilon^{D+2})$ depending on conditions on $\{g_d\}$
- Giles (2017): $O(1/\epsilon^2)$ when $D = 1$ and $\{g_0, g_1\}$ behaves 'nice'
- Our algorithm:
    1. $O(1/\epsilon^2)$ for arbitrary D if $\{g_d\}_{d=0}^{D-1}$ follow a second-order smoothness condition.

Current progress on the problem

- Rainforth et al. (2018): $O(1/\epsilon^{2D+2})$ or $O(1/\epsilon^{D+2})$ depending on conditions on $\{g_d\}$
- Giles (2017): $O(1/\epsilon^2)$ when $D = 1$ and $\{g_0, g_1\}$ behaves 'nice'
- Our algorithm:
  1. $O(1/\epsilon^2)$ for arbitrary D if $\{g_d\}_{d=0}^{D-1}$ follow a second-order smoothness condition.
  2. $O(1/\epsilon^{(2+0.00\cdots 1)})$ for arbitrary D if $\{g_d\}_{d=0}^{D-1}$ follow a Lipschitz smoothness condition.

Road map

- Set-up of the problem
- Current progress
- The algorithm
- Empirical results
- Discussion

The algorithm: High-level idea

- Main idea: an unbiased estimator of γ_0 with finite variance + cost

The algorithm: High-level idea

- Main idea: an unbiased estimator of γ_0 with finite variance + cost
- Unbiasedness $\to O(1/\epsilon^2)$ cost:
  Constructing n i.i.d. unbiased estimators $R_1, \ldots, R_n$ for γ_0. Let our estimator be $R := \frac{1}{n}\sum_{i=1}^{n} R_i$. Then,

$$\mathbb{E}[(R-\gamma_0)^2] = \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n} R_i - \gamma_0\right)^2\right] = \frac{1}{n}\mathrm{Var}(R_1).$$

  Finally, taking $n = \mathrm{Var}(R_1)/\epsilon^2$.

The algorithm: High-level idea

- Main idea: an unbiased estimator of γ_0 with finite variance + cost
- Unbiasedness $\to$ $O(1/\epsilon^2)$ cost:
  Constructing n i.i.d. unbiased estimators $R_1, ..., R_n$ for γ_0. Let our estimator be $R := \frac{1}{n}\sum_{i=1}^{n} R_i$. Then,

$$\mathbb{E}[(R-\gamma_0)^2] = \mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^{n} R_i - \gamma_0\right)^2\right] = \frac{1}{n}\mathrm{Var}(R_1).$$

  Finally, taking $n = \mathrm{Var}(R_1)/\epsilon^2$.
- Unbiasedness $\to$ parallel implementation:
  Closely related to Glynn & Rhee, Jacob, O'Leary & Atchadé.

The algorithm: $D = 1$ case

- Target: Unbiasedly estimate

$$\mathbb{E}\left[g_0\left(y^{(0)}, \mathbb{E}\left[g_1(y^{(0)}, y^{(1)}) \mid y^{(0)})\right]\right)\right]$$

The algorithm: $D = 1$ case

- Target: Unbiasedly estimate

$$\mathbb{E}\left[g_0\left(y^{(0)}, \mathbb{E}\left[g_1(y^{(0)}, y^{(1)}) \mid y^{(0)}\right]\right)\right]$$

- When $y^{(0)}$ fixed $\Leftrightarrow$ Unbiasedly estimate $g(\mathbb{E}[f(X)]])$

The algorithm: $D = 1$ case

- Target: Unbiasedly estimate

$$\mathbb{E}\left[g_0\left(y^{(0)}, \mathbb{E}\left[g_1(y^{(0)}, y^{(1)}) \mid y^{(0)})\right]\right)\right]$$

- When $y^{(0)}$ fixed $\Leftrightarrow$ Unbiasedly estimate $g(\mathbb{E}[f(X)]])$
- Technique: Randomized Multilevel Monte Carlo by McLeish, Glynn & Rhee

The algorithm: $D = 1$ case

- Trick one: telescoping sum
  Suppose $X_1, X_2, \ldots, X_n, \ldots$ i.i.d., let $S_n = \sum_{i=1}^{n} f(X_i)$

$$
\begin{aligned}
g(\mathbb{E}[f]) &= \lim_{l\to\infty} \mathbb{E}\left[g\left(\frac{S_{k_l}}{k_l}\right)\right] \qquad \text{(by LLN)} \\
&= \sum_{l=1}^{\infty} \mathbb{E}\left[g\left(\frac{S_{k_l}}{k_l}\right) - g\left(\frac{S_{k_{l-1}}}{k_{l-1}}\right)\right] \quad (S_0/0 := 0) \\
&= \sum_{l=1}^{\infty} \mathbb{E}\left[\Delta_l\right]
\end{aligned}
$$

The algorithm: $D = 1$ case

- Trick two: Randomize level l
  Sample N with $\mathbb{P}(N = n) = p_n$, and samples $i.i.d.$ random variables $X_1, ..., X_{k_N}$. The 'final' estimator is Δ_N / p_N.

The algorithm: $D = 1$ case

- Trick two: Randomize level l
  Sample N with $\mathbb{P}(N = n) = p_n$, and samples $i.i.d.$ random variables $X_1, \ldots, X_{k_N}$. The 'final' estimator is Δ_N / p_N.
- Sanity check:

$$\begin{aligned}\mathbb{E}[\Delta_N / p_N] &= \mathbb{E}[\mathbb{E}[\Delta_N / p_N \mid N]] \quad \text{(law of iterated expectation)} \\ &= \sum_{n=1}^{\infty} \frac{\mathbb{E}[\Delta_n]}{p_n} \cdot p_n = \sum_{n=1}^{\infty} \mathbb{E}[\Delta_n].\end{aligned}$$

The algorithm: $D = 1$ case

- Trick two: Randomize level l
  Sample N with $\mathbb{P}(N = n) = p_n$, and samples $i.i.d.$ random variables $X_1, ..., X_{k_N}$. The 'final' estimator is Δ_N / p_N.
- Sanity check:

$$\begin{aligned}\mathbb{E}[\Delta_N/p_N] &= \mathbb{E}[\mathbb{E}[\Delta_N/p_N \mid N]] \quad \text{(law of iterated expectation)}\\ &= \sum_{n=1}^{\infty} \frac{\mathbb{E}[\Delta_n]}{p_n} \cdot p_n = \sum_{n=1}^{\infty} \mathbb{E}[\Delta_n].\end{aligned}$$

- Other trick: Antithetic design for Δ_l to reduce variance

The algorithm: General D

- Idea: Run an recursive algorithm on D.

The algorithm: General D

- Idea: Run an recursive algorithm on D.
- More precisely:

The algorithm: General D

- Idea: Run an recursive algorithm on D.
- More precisely:
  - Unbiased estimator for $\gamma_D(y^{(0:D-1)}) = \mathbb{E}\left[g_D\left(y^{(0:D)}\right) \mid y^{(0:D-1)}\right]$ is easy

The algorithm: General D

- Idea: Run an recursive algorithm on D.
- More precisely:
  - Unbiased estimator for $\gamma_D(y^{(0:D-1)}) = \mathbb{E}\left[g_D\left(y^{(0:D)}\right) \mid y^{(0:D-1)}\right]$ is easy
  - $\gamma_{D-1}(y^{(0:D-2)}) \Leftrightarrow D = 1$ case (with unbiased estimator of $\gamma_D(y^{(0:D-1)})$ as input).

The algorithm: General D

- Idea: Run an recursive algorithm on D.
- More precisely:
  - Unbiased estimator for $\gamma_D(y^{(0:D-1)}) = \mathbb{E}\left[g_D\left(y^{(0:D)}\right) \mid y^{(0:D-1)}\right]$ is easy
  - $\gamma_{D-1}(y^{(0:D-2)}) \Leftrightarrow D = 1$ case (with unbiased estimator of $\gamma_D(y^{(0:D-1)})$ as input).
  - $\gamma_{D-2}(y^{(0:D-3)}) \Leftrightarrow D = 1$ case (with unbiased estimator of $\gamma_{D-1}(y^{(0:D-2)})$ as input).

The algorithm: General D

- Idea: Run an recursive algorithm on D.
- More precisely:
  - Unbiased estimator for $\gamma_D(y^{(0:D-1)}) = \mathbb{E}\left[g_D\left(y^{(0:D)}\right) \mid y^{(0:D-1)}\right]$ is easy
  - $\gamma_{D-1}(y^{(0:D-2)}) \Leftrightarrow D = 1$ case (with unbiased estimator of $\gamma_D(y^{(0:D-1)})$ as input).
  - $\gamma_{D-2}(y^{(0:D-3)}) \Leftrightarrow D = 1$ case (with unbiased estimator of $\gamma_{D-1}(y^{(0:D-2)})$ as input).
  - $\gamma_D \Rightarrow \gamma_{D-1} \Rightarrow \gamma_{D-2} \Rightarrow \ldots \Rightarrow \gamma_0$

Road map

- Set-up of the problem
- Current progress
- The algorithm
- Empirical results
- Discussion

Empirical results: recall the problem

- Consider a process $(y^{(0)}, y^{(1)}, y^{(2)})$:

$$y^{(0)} \sim \text{Norm}(\pi/2, 1),\ \ y^{(1)} \sim \text{Norm}(y^{(0)}, 1),\ \ y^{(2)} \sim \text{Norm}(y^{(1)}, 1).$$

- Consider functions of this process:

$$\begin{aligned} g_0(y^{(0)}, z) &:= \sin\left(y^{(0)} + z\right) \\ g_1(y^{(0:1)}, z) &:= \sin\left(y^{(1)} - z\right) \\ g_2(y^{(0:2)}) &:= y^{(2)}. \end{aligned}$$

- The goal is to estimate $\gamma_0 = \mathbb{E}\left[g_0\left(y^{(0)}, \gamma_1\left(y^{(0)}\right)\right)\right]$, where:

$$\begin{aligned} \gamma_1(y^{(0)}) &= \mathbb{E}\left[g_1\left(y^{(0:1)}, \gamma_2\left(y^{(0:1)}\right)\right) \mid y^{(0)}\right] \\ \gamma_2(y^{(0:1)}) &= \mathbb{E}\left[g_2\left(y^{(0:2)}\right) \mid y^{(0:1)}\right] \end{aligned}$$

Empirical results

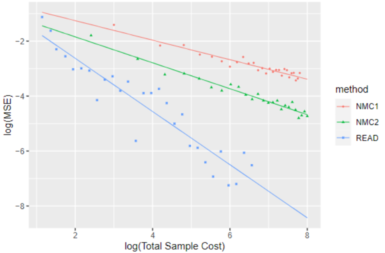

Figure 1. The comparison on the empirical MSEs of estimating the RNE among READ (blue), NMC1 (red), and NMC2 (green). All the logarithms are of the base 10. Each method's empirical errors are calculated based on 20 independent repetitions.

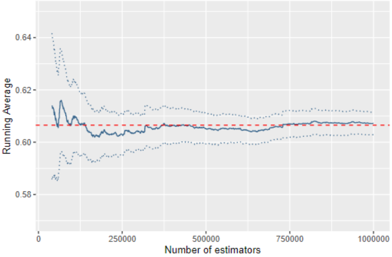

Figure 2. The trace plot (solid blue curve) of the running averages of READ. The blue dotted curves are the 95% confidence intervals. The red dashed line is the ground truth $\exp(-1/2)$.

Road map

- Set-up of the problem
- Current progress
- The algorithm
- Empirical results
- Discussion

Discussion

- Take home message: an $O(1/\epsilon^2)$ estimator for RNEs.
- Key technique: unbiased Monte Carlo
- Remark:
  - Our estimator has excellent dependency on ϵ
  - But poor dependency on D.
- Our QR code:

References

- Syed & Wang, Optimal randomized multilevel Monte Carlo estimators for multilevel nested expectations, *ICML (2023)*.
- Giles, MLMC for Nested Expectations. *Contemporary Computational Mathematics. Springer, Cham. (2018)*
- Rainforth, et al., On Nesting Monte Carlo Estimators, *ICML (2018)*.
- Jacob, O'Leary, & Atchadé, Unbiased Markov chain Monte Carlo with couplings, *JRSS-B (2020)*
- Rhee & Glynn, Unbiased estimation with square root convergence for SDE models, *OR (2015)*
- McLeish, A general method for debiasing a Monte Carlo estimator, *Monte Carlo Methods and Applications (2012)*

Current progress on the problem

- Our algorithm provides a $O(1/\epsilon^2)$ sampling complexity with ϵ-RMSE for arbitrary depth if the functions $\{g_d\}_{d=0}^{D-1}$ follow a last-component bounded second-derivative condition (LBS). Specifically, a function $f : \mathbb{R}^{k+2} \to \mathbb{R}$ satisfies the LBS assumption if there exists a $C_f < \infty$ such that
$$\sup_{y^{(0:k+1)}} \left|\partial_{k+1}^2 f(y^{(0:k+1)})\right| < C_f.$$
- Our algorithm provides a $O(1/\epsilon^{2(1+\delta)})$ sampling complexity with ϵ-MAE for arbitrary depth if $\{g_d\}_{d=0}^{D-1}$ follow a last-component bounded Lipschitz assumption (LBL). Specifically, a function $f : \mathbb{R}^{k+2} \to \mathbb{R}$ satisfies the LBL assumption if for some $L_f < \infty$ for all $x, z \in \mathbb{R}$
$$\sup_{y^{(0:k)}} \left|f(y^{(0:k)}, x) - f(y^{(0:k)}, z)\right| \le L_f|x - z|.$$

The case D arbitrary: theoretical guarantees (LBS)

Theorem

Suppose for every $d \in \{0, 1, \ldots, D-1\}$, the function g_d satisfies the LBS assumption, and $r_d := 1 - 2^{-k_d}$ satisfies $k_d \in \left(1, \frac{2^{d+1}}{2^{d+1}-1}\right)$. Moreover, suppose $\|g_D(y^{(0:D)})\|_{\pi, 2^{D+1}} < \infty$. Then for every $0 \le d \le D$, the output $R_d(y^{(0:d-1)})$ of our algorithm with inputs $\{depth = d, \, trajectory = y^{(0:d-1)}, \mathcal{S}, \, parameters \, (r_d, \ldots, r_{D-1})\}$ has the following properties:

- *For π-almost surely every fixed $y^{(0:d-1)}$,*

$$\mathbb{E}\left[R_d(y^{(0:d-1)}) \mid y^{(0:d-1)}\right] = \gamma_d(y^{(0:d-1)}).$$

- *The expected computational cost of R_d is finite.*
- *The output has finite 2^{d+1}-th moment, i.e.,*

$$\mathbb{E}_\pi\left[\left|R_d(y^{(0:d-1)})\right|^{2^{d+1}}\right] < \infty \quad \text{for} \quad 0 \le d \le D.$$

The case D arbitrary: theoretical guarantees (LBL)

Theorem

Fix any $0 < \delta < 1/2$. *Suppose for every* $d \in \{0, 1, \dots, D-1\}$, *the function* g_d *satisfies the LBL assumption, and* $r_d := 1 - 2^{-k_d}$ *satisfies* $k_d \in \left(1, \left(\frac{2^{d+2}-3\delta}{2^{d+3}-3\delta}\right)\left(\frac{2^{d+1}-\delta}{2^d-\delta}\right)\right)$. *Moreover, suppose* $\|g_D(y^{(0:D)})\|_{\pi,2} < \infty$. *Then for every* $0 \le d \le D$, *the output* $R_d(y^{(0:d-1)})$ *of our algorithm with inputs* $\{depth = d,\ trajectory = y^{(0:d-1)},\ \mathcal{S},\ parameters\ (r_d, \dots, r_{D-1})\}$:

- *For almost surely every fixed* $y^{(0:d-1)}$,

$$\mathbb{E}\left[R_d(y^{(0:d-1)}) \mid y^{(0:d-1)}\right] = \gamma_d(y^{(0:d-1)}).$$

- *The expected computational cost of* R_d *is finite.*
- *The output has finite* $(2 - \delta/2^d)$*-th moment, i.e.,*

$$\mathbb{E}_\pi\left[|R_d(y^{(0:d-1)})|^{(2-\delta/2^d)}\right] < \infty \quad \text{for} \quad 0 \le d \le D.$$

The case D arbitrary: theoretical guarantees (LBL)

Theorem

Let the assumptions for the LBL theorem hold. Fix $0 < \delta < 1/2$*. For any* $\epsilon > 0$*, there exists an estimator* R *with expected sampling complexity* $O(1/\epsilon^{2(1+\delta)})$ *such that the mean absolute error* $\mathbb{E}[|R - \gamma_0|] \leq \epsilon$*.*

Proof.

(Sketch) Use same procedure as in the previous theorem, however there is a lack of finite variance, and so we use the Marcinkiewicz-Zygmund LLN for the finite $(2 - \delta)$-th moment case. The result then follows. $\square$