Meta-learning Control Variates: Variance Reduction with Limited Data

Zhuo Sun[1, 2]

[1]Department of Statistical Science, University College London
[2]The Alan Turing Institute

2023

Collaborators

Chris J. Oates
(Newcastle & Turing)

François-Xavier Briol
(UCL & Turing)

Sun, Z., Oates, C. J. Briol, F-X. (2023).
Meta-learning Control Variates: Variance Reduction with Limited Data.
arXiv:2303.04756. In Proc. of UAI 2023.

Problem of Interest

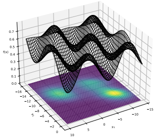
- Consider a finite (but possibly large) number, T, of integration tasks

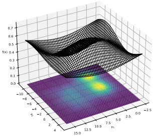
$$\Pi_1[f_1], \ldots, \Pi_T[f_T]. \tag{1}$$

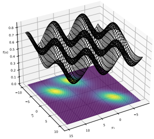
Denote by $\mathcal{T}_t := \{f_t, \pi_t\}$ the components of the t^{th} task:

i). an integrand $f_t \in \mathcal{L}^2(\pi_t)$; a density $\pi_t : \mathcal{X} \to [0, \infty)$;

ii). only have access to very limited data.







Preliminary

- **Monte Carlo (MC) estimator** for each task:

$$\hat{\Pi}^{\mathrm{MC}}[f] := \frac{1}{N}\sum_{i=1}^{N} f(x_i), \qquad \{x_i\}_{i=1}^{N} \sim \Pi.$$

Cons ☹***:*** large variance $N^{-1}\mathbb{V}_\pi[f]$ (CLT).

- **Control Variates (CVs)**:

Estimate $\Pi[f]$ by $\Pi[f-g]+\Pi[g]$ where $g \in \mathcal{L}^2(\pi)$: $\Pi[g]$ can be exactly computed (Stein) and $\mathbb{V}_\pi[f-g]$ is small (CLT).

➛ *Step 1. Choose $\mathcal{G}$ such that $\Pi[g]$ can be exactly computed for all $g \in \mathcal{G}$.*

✓ **Stein operators** $\mathcal{S}_\pi$**:** $g(\cdot;\gamma) := \mathcal{S}_\pi[u(\cdot)] + \gamma_0$ with $\Pi[\mathcal{S}_\pi[u]] = 0$.

✓ **Parametric Spaces:** $u := u_{\gamma_{1:p}}$.

➛ *Step 2. Select a $\hat{g}_m$ from $\mathcal{G}$ by minimising $J_S(\gamma)$.*

$$J_S(\gamma) := \underbrace{\frac{1}{m}\sum_{i=1}^{m}\left(f(x_i) - g(x_i;\gamma)\right)^2}_{\text{empirical est. of } \mathbb{V}_\pi[f-g]}. \tag{2}$$

Preliminary

- **Monte Carlo (MC) estimator** for each task:

$$\hat{\Pi}^{\text{MC}}[f] := \frac{1}{N}\sum_{i=1}^{N} f(x_i), \qquad \{x_i\}_{i=1}^{N} \sim \Pi.$$

Cons ☹: large variance $N^{-1}\mathbb{V}_\pi[f]$ (CLT).

- **Control Variates (CVs)**:

Estimate $\Pi[f]$ by $\Pi[f-g]+\Pi[g]$ where $g \in \mathcal{L}^2(\pi)$: $\Pi[g]$ can be exactly computed (Stein) and $\mathbb{V}_\pi[f-g]$ is small (CLT).

➸ *Step 1. Choose $\mathcal{G}$ such that $\Pi[g]$ can be exactly computed for all $g \in \mathcal{G}$.*

✓ **Stein operators** $\mathcal{S}_\pi$**:** $g(\cdot;\gamma) := \mathcal{S}_\pi[u(\cdot)] + \gamma_0$ with $\Pi[\mathcal{S}_\pi[u]] = 0$.

✓ **Parametric Spaces:** $u := u_{\gamma_{1:p}}$.

➸ *Step 2. Select a $\hat{g}_m$ from $\mathcal{G}$ by minimising $J_S(\gamma)$.*

$$J_S(\gamma) := \underbrace{\frac{1}{m}\sum_{i=1}^{m}\left(f(x_i) - g(x_i;\gamma)\right)^2}_{\text{empirical est. of } \mathbb{V}_\pi[f-g]}. \tag{2}$$

Preliminary

- **Monte Carlo (MC) estimator** for each task:

$$\hat{\Pi}^{\mathrm{MC}}[f] := \frac{1}{N}\sum_{i=1}^{N} f(x_i), \qquad \{x_i\}_{i=1}^{N} \sim \Pi.$$

Cons ☹***:*** large variance $N^{-1}\mathbb{V}_\pi[f]$ (CLT).

- **Control Variates (CVs)**:

Estimate $\Pi[f]$ by $\Pi[f-g]+\Pi[g]$ where $g \in \mathcal{L}^2(\pi)$: $\Pi[g]$ can be exactly computed (Stein) and $\mathbb{V}_\pi[f-g]$ is small (CLT).

➳ *Step 1. Choose $\mathcal{G}$ such that $\Pi[g]$ can be exactly computed for all $g \in \mathcal{G}$.*

✓ **Stein operators $\mathcal{S}_\pi$:** $g(\cdot;\gamma) := \mathcal{S}_\pi[u(\cdot)] + \gamma_0$ with $\Pi[\mathcal{S}_\pi[u]] = 0$.

✓ **Parametric Spaces:** $u := u_{\gamma_{1:p}}$.

➳ *Step 2. Select a $\hat{g}_m$ from $\mathcal{G}$ by minimising $J_S(\gamma)$.*

$$J_S(\gamma) := \underbrace{\frac{1}{m}\sum_{i=1}^{m}\left(f(x_i) - g(x_i;\gamma)\right)^2}_{\text{empirical est. of } \mathbb{V}_\pi[f-g]}. \qquad (2)$$

Preliminary

- **Monte Carlo (MC) estimator** for each task:

$$\hat{\Pi}^{\mathrm{MC}}[f] := \frac{1}{N}\sum_{i=1}^{N} f(x_i), \qquad \{x_i\}_{i=1}^{N} \sim \Pi.$$

Cons ☹**:** large variance $N^{-1}\mathbb{V}_\pi[f]$ (CLT).

- **Control Variates (CVs)**:

Estimate $\Pi[f]$ by $\Pi[f-g]+\Pi[g]$ where $g \in \mathcal{L}^2(\pi)$: $\Pi[g]$ can be exactly computed (Stein) and $\mathbb{V}_\pi[f-g]$ is small (CLT).

➸ *Step 1. Choose $\mathcal{G}$ such that $\Pi[g]$ can be exactly computed for all $g \in \mathcal{G}$.*

✓ **Stein operators $\mathcal{S}_\pi$:** $g(\cdot;\gamma) := \mathcal{S}_\pi[u(\cdot)] + \gamma_0$ with $\Pi[\mathcal{S}_\pi[u]] = 0$.

✓ **Parametric Spaces:** $u := u_{\gamma_{1:p}}$.

➸ *Step 2. Select a $\hat{g}_m$ from $\mathcal{G}$ by minimising $J_S(\gamma)$.*

$$J_S(\gamma) := \underbrace{\frac{1}{m}\sum_{i=1}^{m}\left(f(x_i) - g(x_i;\gamma)\right)^2}_{\text{empirical est. of } \mathbb{V}_\pi[f-g]}. \tag{2}$$

Preliminary

- **Monte Carlo (MC) estimator** for each task:

$$\hat{\Pi}^{\mathrm{MC}}[f] := \frac{1}{N}\sum_{i=1}^{N} f(x_i), \qquad \{x_i\}_{i=1}^{N} \sim \Pi.$$

Cons ☹**:** large variance $N^{-1}\mathbb{V}_\pi[f]$ (CLT).

- **Control Variates (CVs)**:

Estimate $\Pi[f]$ by $\Pi[f-g]+\Pi[g]$ where $g \in \mathcal{L}^2(\pi)$: $\Pi[g]$ can be exactly computed (Stein) and $\mathbb{V}_\pi[f-g]$ is small (CLT).

➸ *Step 1. Choose $\mathcal{G}$ such that $\Pi[g]$ can be exactly computed for all $g \in \mathcal{G}$.*

✓ **Stein operators $\mathcal{S}_\pi$:** $g(\cdot;\gamma) := \mathcal{S}_\pi[u(\cdot)] + \gamma_0$ with $\Pi[\mathcal{S}_\pi[u]] = 0$.

✓ **Parametric Spaces:** $u := u_{\gamma_{1:p}}$.

➸ *Step 2. Select a $\hat{g}_m$ from $\mathcal{G}$ by minimising $J_S(\gamma)$.*

$$J_S(\gamma) := \underbrace{\frac{1}{m}\sum_{i=1}^{m}\left(f(x_i) - g(x_i;\gamma)\right)^2}_{\text{empirical est. of } \mathbb{V}_\pi[f-g]}. \tag{2}$$

Preliminary

- **Monte Carlo (MC) estimator** for each task:

$$\hat{\Pi}^{\mathrm{MC}}[f] := \frac{1}{N}\sum_{i=1}^{N} f(x_i), \qquad \{x_i\}_{i=1}^{N} \sim \Pi.$$

Cons ☹***:*** large variance $N^{-1}\mathbb{V}_\pi[f]$ (CLT).

- **Control Variates (CVs)**:

Estimate $\Pi[f]$ by $\Pi[f-g]+\Pi[g]$ where $g \in \mathcal{L}^2(\pi)$: $\Pi[g]$ can be exactly computed (Stein) and $\mathbb{V}_\pi[f-g]$ is small (CLT).

  - *Step 1. Choose $\mathcal{G}$ such that $\Pi[g]$ can be exactly computed for all $g \in \mathcal{G}$.*
    - ✓ **Stein operators $\mathcal{S}_\pi$:** $g(\cdot;\gamma) := \mathcal{S}_\pi[u(\cdot)] + \gamma_0$ with $\Pi[\mathcal{S}_\pi[u]] = 0$.
    - ✓ **Parametric Spaces:** $u := u_{\gamma_{1:p}}$.

  - *Step 2. Select a $\hat{g}_m$ from $\mathcal{G}$ by minimising $J_S(\gamma)$.*

$$J_S(\gamma) := \underbrace{\frac{1}{m}\sum_{i=1}^{m}\left(f(x_i) - g(x_i;\gamma)\right)^2}_{\text{empirical est. of } \mathbb{V}_\pi[f-g]}. \qquad (2)$$

Control Variates Cont'd

- *Step 3. Construct a CV estimator with the remaining $N - m$ samples:*

$$\hat{\Pi}^{\mathrm{CV}}[f] := \hat{\Pi}^{\mathrm{MC}}[\underbrace{f - \hat{g}_m}_{\text{var. minimised!}}] + \Pi[\hat{g}_m] \tag{3}$$
$$= \frac{1}{N-m} \sum_{i=m+1}^{N} (f(x_i) - \hat{g}_m(x_i)) + \Pi[\hat{g}_m].$$

CLT: $\sqrt{N-m}\left(\hat{\Pi}^{\mathrm{CV}}[f] - \Pi[f]\right) \xrightarrow{d} \mathcal{N}(0, \mathbb{V}_{\Pi}[f - \hat{g}_m])$.
$\Longrightarrow \hat{g}_m \approx f$ means $\mathbb{V}_{\Pi}[f - \hat{g}_m]$ **close to zero and fast convergence rate!**

***Cons* ☹:** need a large number of samples; ignore potential relationship among T tasks.

Control Variates Cont'd

- *Step 3. Construct a CV estimator with the remaining $N - m$ samples:*

$$\hat{\Pi}^{\text{CV}}[f] := \hat{\Pi}^{\text{MC}}[\underbrace{f - \hat{g}_m}_{\text{var. minimised!}}] + \Pi[\hat{g}_m] \tag{3}$$

$$= \frac{1}{N-m} \sum_{i=m+1}^{N} (f(x_i) - \hat{g}_m(x_i)) + \Pi[\hat{g}_m].$$

CLT: $\sqrt{N-m}\left(\hat{\Pi}^{\text{CV}}[f] - \Pi[f]\right) \xrightarrow{d} \mathcal{N}(0, \mathbb{V}_\Pi[f - \hat{g}_m])$.
$\implies \hat{g}_m \approx f$ means $\mathbb{V}_\Pi[f - \hat{g}_m]$ **close to zero and fast convergence rate!**

***Cons* ☹:** need a large number of samples; ignore potential relationship among T tasks.

Control Variates Cont'd

- *Step 3. Construct a CV estimator with the remaining $N - m$ samples:*

$$\hat{\Pi}^{\mathrm{CV}}[f] := \hat{\Pi}^{\mathrm{MC}}[\underbrace{f - \hat{g}_m}_{\text{var. minimised!}}] + \Pi[\hat{g}_m] = \frac{1}{N-m}\sum_{i=m+1}^{N}(f(x_i) - \hat{g}_m(x_i)) + \Pi[\hat{g}_m]. \tag{3}$$

CLT: $\sqrt{N-m}\left(\hat{\Pi}^{\mathrm{CV}}[f] - \Pi[f]\right) \xrightarrow{d} \mathcal{N}(0, \mathbb{V}_{\Pi}[f - \hat{g}_m])$.
$\Longrightarrow \hat{g}_m \approx f$ means $\mathbb{V}_{\Pi}[f - \hat{g}_m]$ **close to zero and fast convergence rate!**

Cons ☹: need a large number of samples; ignore potential relationship among T tasks.

Related Work

- **Vector-valued Control Variates (vv-CVs)** [Sun et al., 2021]:
  - Reformat (1) as a vector-valued integration task

$$\Pi[f] := (\Pi_1[f_1], \ldots, \Pi_T[f_T])^\top .$$

  - Derive matrix-valued Stein kernels K_0: $\Pi_t[g_t] = 0$ for $t \in [T]$ and $g \in \mathcal{H}_{K_0}$.

Pros ☺***:*** exploit the relationship among integration tasks.

Cons ☹***:*** computational cost between $\mathcal{O}(T^4)$ and $\mathcal{O}(T^6)$.

Z. Sun, A. Barp, and F-X. Briol. “Vector-Valued Control Variates”. In ICML 2023.

Motivation

The key challenge remains to be solved:

- How can we construct CVs at scale, sharing information across a large number of tasks even with limited samples?

Answer in brief:

- Re-frame selecting effective CVs as optimisation tasks.
- Utilise meta-learning to learn CVs fast.

Motivation

The key challenge remains to be solved:

- How can we construct CVs at scale, sharing information across a large number of tasks even with limited samples?

Answer in brief:

- Re-frame selecting effective CVs as optimisation tasks.
- Utilise meta-learning to learn CVs fast.

Motivation

The key challenge remains to be solved:

- How can we construct CVs at scale, sharing information across a large number of tasks even with limited samples?

Answer in brief:

- Re-frame selecting effective CVs as optimisation tasks.
- Utilise meta-learning to learn CVs fast.

Motivation

The key challenge remains to be solved:

- How can we construct CVs at scale, sharing information across a large number of tasks even with limited samples?

Answer in brief:

- Re-frame selecting effective CVs as optimisation tasks.
- Utilise meta-learning to learn CVs fast.

Our Proposed Method: Meta-learning Control Variates

- **Set-up:** For each task $\mathcal{T}_t := \{f_t, \pi_t\}$, we split the data D_t into two disjoint sets S_t and Q_t,

$$S_t := \{x_j, \nabla \log \pi_t(x_j), f_t(x_j)\}_{j=1}^{m_t}, \qquad Q_t := \{x_j, \nabla \log \pi_t(x_j), f_t(x_j)\}_{j=m_t+1}^{N_t}.$$

- **Two steps:**
  1. Learning a Meta-CV;
  2. Task-specific CVs from the Meta-CV.

Our Proposed Method: **Meta-**learning **C**ontrol **V**ariates

- **Set-up:** For each task $\mathcal{T}_t := \{f_t, \pi_t\}$, we split the data D_t into two disjoint sets S_t and Q_t,

$$S_t := \{x_j, \nabla \log \pi_t(x_j), f_t(x_j)\}_{j=1}^{m_t}, \qquad Q_t := \{x_j, \nabla \log \pi_t(x_j), f_t(x_j)\}_{j=m_t+1}^{N_t}.$$

- **Two steps:**
  1. Learning a Meta-CV;
  2. Task-specific CVs from the Meta-CV.

Step I: Learning a Meta-CV

- An *idealised Meta-CV* as a CV whose parameters γ satisfy,

$$\arg\min_{\gamma \in \mathbb{R}^{p+1}} \mathbb{E}_t[\mathcal{J}_t(\gamma)] \text{ with } \mathcal{J}_t(\gamma) := \overbrace{J_t(\text{UPDATE}_L(\gamma, \nabla_\gamma \underbrace{J_t(\gamma)}_{J_{S_t}}; \alpha))}^{J_{Q_t}}$$

where $\mathbb{E}_t$ denotes expectation with respect to a uniformly sampled task index $t \in \{1, \ldots, T\}$.

- $\text{UPDATE}_L(;\alpha) \rightarrow$ L-step gradient descent with step size α.
- Optimising $\rightarrow$ gradient-based bi-level optimisation [Finn et al., 2017] with J_{S_t} and J_{Q_t} as in (2).
- $g(\cdot; \hat{\gamma}_{\text{meta}}) \rightarrow$ the so-called *Meta-CV*.

C. Finn, P. Abbeel, and S. Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In ICML (2017).

Step I: Learning a Meta-CV

- An *idealised Meta-CV* as a CV whose parameters γ satisfy,

$$\arg\min_{\gamma \in \mathbb{R}^{p+1}} \mathbb{E}_t[\mathcal{J}_t(\gamma)] \text{ with } \mathcal{J}_t(\gamma) := \overbrace{J_t(\text{UPDATE}_L(\gamma, \nabla_\gamma \underbrace{J_t(\gamma)}_{J_{S_t}}; \alpha))}^{J_{Q_t}}$$

where $\mathbb{E}_t$ denotes expectation with respect to a uniformly sampled task index $t \in \{1, \ldots, T\}$.

  - $\text{UPDATE}_L(;\alpha) \rightarrow L$-step gradient descent with step size α.
  - Optimising $\rightarrow$ gradient-based bi-level optimisation [Finn et al., 2017] with J_{S_t} and J_{Q_t} as in (2).
  - $g(\cdot;\hat{\gamma}_{\text{meta}}) \rightarrow$ the so-called *Meta-CV*.

C. Finn, P. Abbeel, and S. Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In ICML (2017).

Step I: Learning a Meta-CV

- An *idealised Meta-CV* as a CV whose parameters γ satisfy,

$$\arg\min_{\gamma \in \mathbb{R}^{p+1}} \mathbb{E}_t[\mathcal{J}_t(\gamma)] \text{ with } \mathcal{J}_t(\gamma) := \overbrace{J_t(\text{UPDATE}_L(\gamma, \nabla_\gamma \underbrace{J_t(\gamma)}_{J_{S_t}}; \alpha))}^{J_{Q_t}}$$

where $\mathbb{E}_t$ denotes expectation with respect to a uniformly sampled task index $t \in \{1, \ldots, T\}$.

- $\text{UPDATE}_L(;\alpha) \rightarrow L$-step gradient descent with step size α.
- Optimising $\rightarrow$ gradient-based bi-level optimisation [Finn et al., 2017] with J_{S_t} and J_{Q_t} as in (2).
- $g(\cdot; \hat{\gamma}_{\text{meta}}) \rightarrow$ the so-called *Meta-CV*.

C. Finn, P. Abbeel, and S. Levine. “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks”. In ICML (2017).

Step I: Learning a Meta-CV

- An *idealised Meta-CV* as a CV whose parameters γ satisfy,

$$\arg\min_{\gamma \in \mathbb{R}^{p+1}} \mathbb{E}_t[\mathcal{J}_t(\gamma)] \text{ with } \mathcal{J}_t(\gamma) := \overbrace{J_t(\text{UPDATE}_L(\gamma, \nabla_\gamma \underbrace{J_t(\gamma)}_{J_{S_t}}; \alpha))}^{J_{Q_t}}$$

where $\mathbb{E}_t$ denotes expectation with respect to a uniformly sampled task index $t \in \{1, \ldots, T\}$.

- $\text{UPDATE}_L(;\alpha) \rightarrow L$-step gradient descent with step size α.
- Optimising $\rightarrow$ gradient-based bi-level optimisation [Finn et al., 2017] with J_{S_t} and J_{Q_t} as in (2).
- $g(\cdot; \hat{\gamma}_{\text{meta}}) \rightarrow$ the so-called *Meta-CV*.

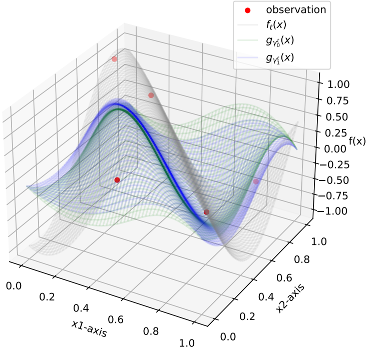
C. Finn, P. Abbeel, and S. Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks". In ICML (2017).

Step II: Task-specific CVs from the Meta-CV

- **Task-specific CVs** $g(\cdot;\hat{\gamma}_L^t)$ **for** $\Pi_t[f_t]$**:**
  - $\hat{\gamma}_L^t \leftarrow \text{UPDATE}_L\left(\hat{\gamma}_{\text{meta}}, \nabla_\gamma J_{S_t}(\hat{\gamma}_{\text{meta}}); \alpha\right)$.
  - Estimate $\Pi_t[f_t]$ with Q_t by:

$$\hat{\Pi}_t^{\text{CV}}[f_t] := \hat{\Pi}_t^{\text{MC}}[f_t - g(\cdot;\hat{\gamma}_L^t)] + \Pi[g(\cdot;\hat{\gamma}_L^t)]$$

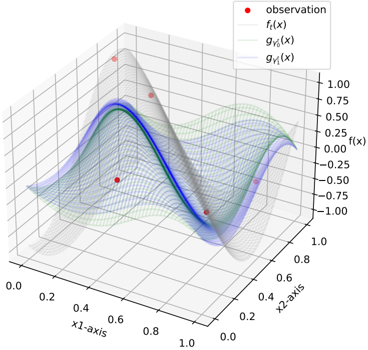
$$= \frac{1}{N-m}\sum_{i=m+1}^{N}\left(f_t(x_i) - g(x_i;\hat{\gamma}_L^t)\right) + \Pi_t[g(\cdot;\hat{\gamma}_L^t)].$$



Step II: Task-specific CVs from the Meta-CV

- **Task-specific CVs $g(\cdot;\hat{\gamma}_L^t)$ for $\Pi_t[f_t]$:**
  - $\hat{\gamma}_L^t \leftarrow \text{UPDATE}_L\left(\hat{\gamma}_{\text{meta}}, \nabla_\gamma J_{S_t}(\hat{\gamma}_{\text{meta}}); \alpha\right)$.
  - Estimate $\Pi_t[f_t]$ with Q_t by:

$$\hat{\Pi}_t^{\text{CV}}[f_t] := \hat{\Pi}_t^{\text{MC}}[f_t - g(\cdot;\hat{\gamma}_L^t)] + \Pi[g(\cdot;\hat{\gamma}_L^t)]$$
$$= \frac{1}{N-m}\sum_{i=m+1}^{N}\left(f_t(x_i) - g(x_i;\hat{\gamma}_L^t)\right) + \Pi_t[g(\cdot;\hat{\gamma}_L^t)].$$



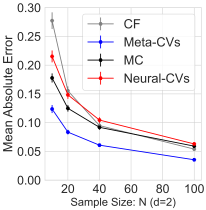
Experiments — A Synthetic Example

Consider integrands of the form:

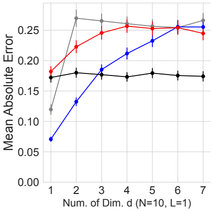
$$f_t(x; a_t) = \cos\left(2\pi a_{t,1} + \sum_{i=1}^{d} a_{t,i+1} x_i\right),$$

with parameters $a_t \in \mathbb{R}^{d+1}$, and let π_t be the uniform distribution on $\mathcal{X} = [0, 1]^d$.

- a_t controls the difficulty: larger $a_t \rightarrow$ larger frequency.
- sample tasks $\Longleftrightarrow$ sample $a_t \sim \rho$.



Effect of N_t per task.



Effect of Dimension d.

Marginalization in Hierarchical Gaussian Processes

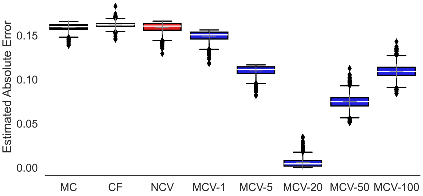
Sarcos robot arm: a canonical example for hierarchical Gaussian processes regression.

Bayesian posterior predictive mean at an unseen state z^*:

$$\mathbb{E}[Y^*|y_{1:q}] = \mathbb{E}_{X\sim\pi(\cdot|y_{1:q})}[\mathbb{E}[Y^*|y_{1:q}, X]].$$

- Integrand: $f(x; z^*) = \mathbb{E}[Y^*|y_{1:q}, x] = K_{z^*,q}(x)(K_{q,q}(x) + \sigma^2 I_q)^{-1} y_{1:q}$.
- Posterior of kernel hyperparameters $\pi(x|y_{1:q})$.
- Each state $z*$ corresponds to a task.

Expensive integrand f: $\mathcal{O}(q^3)$ operations per evaluation.



MCV-L: Meta-CVs with L inner updates.

Theoretical Analysis

Theorem

Let $\hat{\gamma}_{\text{meta}}$ be the output of the propose algorithm with gradient descent steps with model hyper-parameters {...} Then, under {...} assumptions:

$$\mathbb{E}[\|\mathbb{E}_t[\nabla \mathcal{J}_t(\hat{\gamma}_{\text{meta}})]\|_2] = \mathcal{O}\left(\sqrt{\frac{1}{I_{tr}} + \frac{1}{B}}\right).$$

Corollary

Further suppose that there exists $\mu > 0$ such that for all t and all γ, $\nabla^2 J_{Q_t}(\gamma) \succeq \mu I_{p+1}$ where I_{p+1} is an identity matrix of size $p+1$. Then there exist constants $C_1, C_2 > 0$ such that

$$\mathbb{E}[\mathbb{E}_t[\|\hat{\gamma}_\epsilon - \gamma_t^*\|_2]] \leqslant \frac{C_1}{\mu}\epsilon + \frac{C_2}{\mu},$$

where γ_t^* is the (unique) minimiser of $\gamma \mapsto J_{Q_t}(\gamma)$ ...

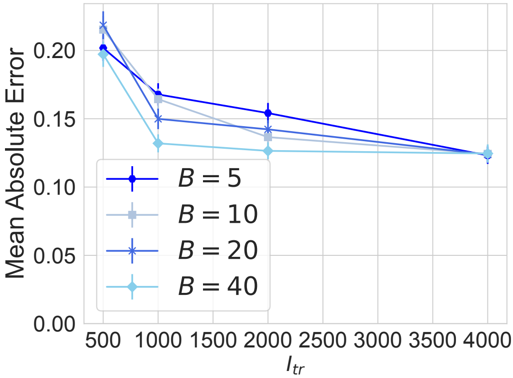
K. Ji, J. Yang, and Y. Liang. "Theoretical Convergence of Multi-Step Model-Agnostic Meta-Learning.". J. Mach. Learn. Res. 23 (2022).

Theoretical Analysis (Cont'd)

Back to the synthetic example:

$$f_t(x; a_t) = \cos\left(2\pi a_{t,1} + \sum_{i=1}^{d} a_{t,i+1} x_i\right),$$

with parameters $a_t \in \mathbb{R}^{d+1}$. π_t is the uniform distribution on $\mathcal{X} = [0,1]^d$.



Conclusion

- **Meta-CVs** work well for variance reduction with limited data by sharing information among tasks.
- **Meta-CVs** is scalable in T and N_t.

Find more (theories and experiments) in the paper:

Sun, Z., Oates, C. J. Briol, F-X. (2023). Meta-learning Control Variates: Variance Reduction with Limited Data. In Proc. of UAI 2023.