Distributionally Robust Bayesian Quadrature Optimization

Thanh Tang Nguyen* Sunil Gupta Huong Ha Santu Rana Svetha Venkatesh

Applied Artificial Intelligence Institute, Deakin University, Australia

26-28 August, 2020

Outline

1. Introduction and Motivation
2. Problem setting
3. Our approach
4. Experiments
5. Conclusion

Motivation

- Making robust decisions under the context uncertainty is critical in many applications
- A concrete example: hyperparameter selection of an machine learning algorithm using cross-validation
  - **Goal**: select robust hyperparameters that generalize well to test set
  - Contexts here = folds
  - The variance across contexts might be high; Ignoring this uncertainty → sub-optimal and non-robust decisions

Introduction

- **Main general question**: How to achieve robustness with guarantee when making decisions with *spurious* rewards?

→ We study it in a concrete setting:

Learning to optimize under uncertain contexts where the context distribution is misspecified

Problem setting

We consider the stochastic black-box optimization problem:

$$\max_{x\in\mathcal{X}\subset\mathbb{R}^d} g(x) := \max_{x\in\mathcal{X}} \mathbb{E}_{P_0(w)}[f(x,w)]$$

where

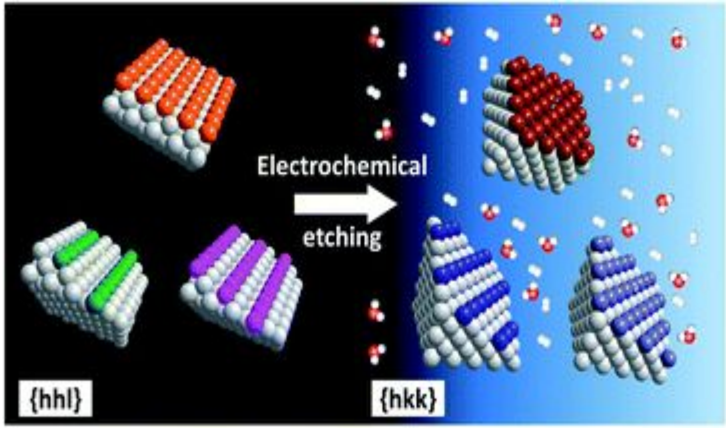
- f: an expensive, derivative-free, black-box function;
- $P_0(w)$: a distribution over context
- We consider the **distributional uncertainty** setting: $P_0(w)$ is unknown except for a set of its empirical samples $\{w_1, w_2, \ldots, w_n\}$
- **Goal**: Find a robust solution (w.r.t. $P_0(w)$) under the distributional uncertainty

Why the distributional uncertainty setting important?

Real-world scenarios: Learning to optimize under uncertain contexts where we do not know the context distribution $P_0(w)$ but its empirical samples

E.g. 1: Alloy design

- Combine several elements for desirable properties
- Alloy elements contain impurities
- Measuring impurities is expensive and we do not know the impurity distribution but only a set of samples
- **Goal**: Alloy design distributionally robust w.r.t. impurities

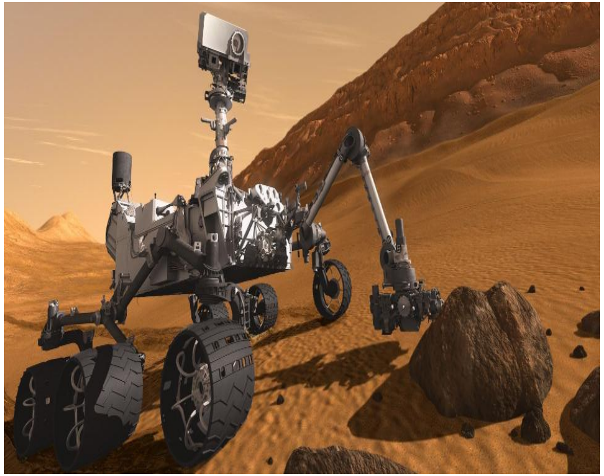


Why the distributional uncertainty setting important?

Real-world scenarios: Learning to optimize under uncertain contexts where we do not know the context distribution $P_0(w)$ but its empirical samples

E.g. 2: Robust control in reinforcement learning

- **Goal**: learn an optimal policy that is robust to unknown environment variables
- Env variables = unobserved state features determined randomly by the environment
- Unknown of the env variable distribution, obtaining its samples is expensive via previous catastrophic events



Why the distributional uncertainty setting important?

Real-world scenarios: Learning to optimize under uncertain contexts where we do not know the context distribution $P_0(w)$ but its empirical samples

E.g. 3: Cross-validation hyperparameter tuning

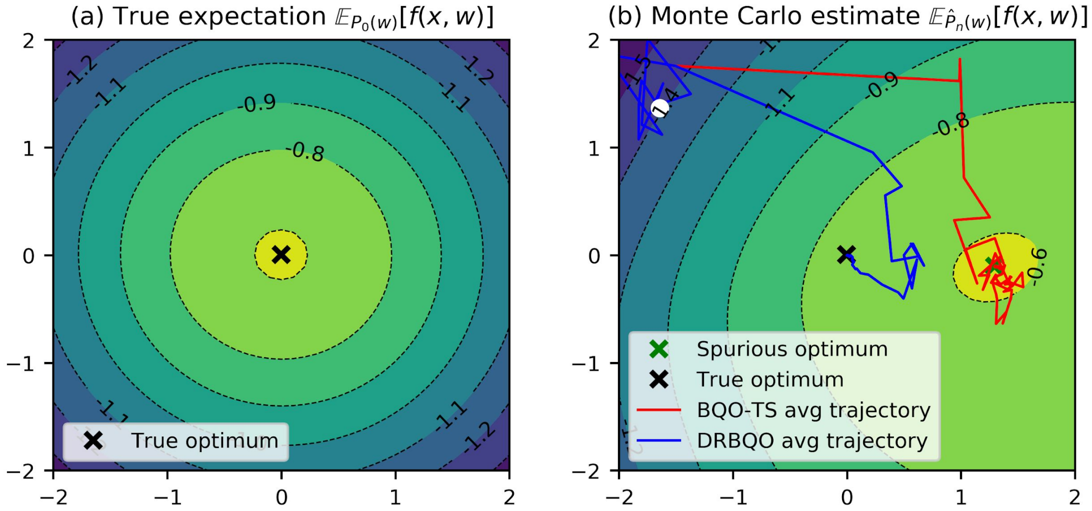
- **Goal**: Find robust hyperparameters of a machine learning algorithm that can generalize well to the test set.

Approaches

- **Naive approach**: Plug in the empirical distribution

$$g_{mc}(x) = \frac{1}{n}\sum_{i=1}^{n} f(x, w_i)$$

But ...



Our approach

Distributionally robust stochastic black-box optimization:

- **Intuition**: Optimize the expected function under the most <u>adversarial</u> distribution over some <u>uncertainty set</u>
- **Formally**,

$$\max_{x \in \mathcal{X}} \min_{P \in \mathcal{P}_{n,\rho}} \mathbb{E}_{P(w)}[f(x,w)],$$

where

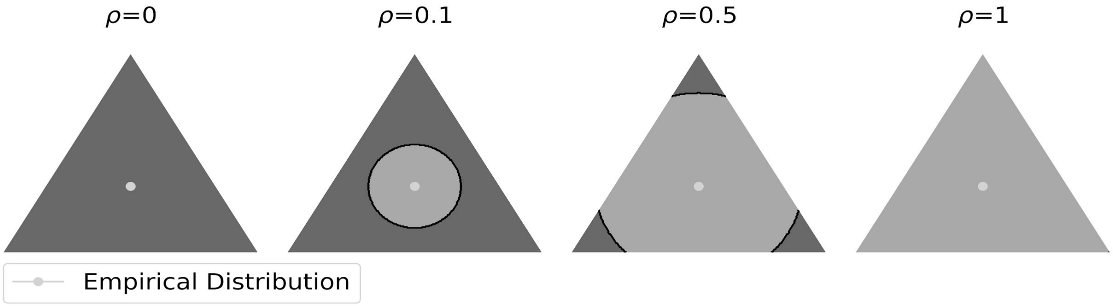
- <u>Uncertainty set</u> within a radius $\mathcal{P}_{n,\rho} = \{P | D(P, \hat{P}_n) \leq \rho\}$
- <u>Empirical</u> distribution $\hat{P}_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{w_i}$
- <u>Divergence</u> $D(\cdot, \cdot)$

Our approach

- We choose D to be χ^2 **divergence**:

$$D_{\chi^2}(P,Q) = \frac{1}{2}\int(\frac{dP}{dQ} - 1)^2 dQ$$

- **Why**? → DRBQO is equivalent to variance penalization (Theorem 1 in the main paper)
- **Intuition**:



Solving the distributionally robust stochastic black-box opt

Algorithm 1: DRBQO: Distributionally Robust Bayesian quadrature optimization

Input: Prior $\mathtt{GP}(\mu_0, k)$, horizon T, fixed sample set S_n, confidence radius $\rho \geq 0, C_0 = k$.

```
for t = 1 to T do
    /* Posterior sampling                                    */
    Sample f̃_t ~ GP(μ_{t-1}, C_{t-1}).
    /* A surrogate DR optimization                           */
    Choose x_t ∈ argmax_{x∈X} min_{P∈P_{n,ρ}} E_P[f̃_t(x, w)].
    /* Highest posterior variance                            */
    Choose w_t = argmax_{w∈S_n} C_{t-1}(x_t, w; x_t, w).
    Observe reward ŷ_t ← f(x_t, w_t) + ε_t.
    Perform update GP to get μ_t and C_t.
end
```

Output: $\arg\max_{x \in \{x_1, \ldots, x_T\}} \min_{P \in \mathcal{P}_{n,\rho}} \mathbb{E}_P[\mu_T(x, w)]$.

1. Build a <u>Gaussian Process (GP)</u> to model f
2. <u>Thompson sampling</u> to draw a <u>surrogate function</u> $\tilde{f}_t$ for f
3. Solve the <u>surrogate distributionally robust optimization</u> for $\tilde{f}_t$
4. Sample new input and update the GP

→ <u>simple</u> yet <u>highly flexible</u> and <u>robustness with theoretical guarantee</u>!

Solving the distributionally robust stochastic black-box opt

Algorithm 1: `DRBQO`: Distributionally Robust Bayesian quadrature optimization

Input: Prior $\texttt{GP}(\mu_0, k)$, horizon T, fixed sample set S_n, confidence radius $\rho \geq 0, C_0 = k$.

for $t = 1$ *to* T **do**

  `/* Posterior sampling */`

  Sample $\tilde{f}_t \sim \texttt{GP}(\mu_{t-1}, C_{t-1})$.

  `/* A surrogate DR optimization */`

  Choose $x_t \in \arg\max_{x \in \mathcal{X}} \min_{P \in \mathcal{P}_{n,\rho}} \mathbb{E}_P[\tilde{f}_t(x, w)]$.

  `/* Highest posterior variance */`

  Choose $w_t = \arg\max_{w \in S_n} C_{t-1}(x_t, w; x_t, w)$.

  Observe reward $\hat{y}_t \leftarrow f(x_t, w_t) + \epsilon_t$.

  Perform update $\texttt{GP}$ to get μ_t and C_t.

end

Output: $\arg\max_{x \in \{x_1, \ldots, x_T\}} \min_{P \in \mathcal{P}_{n,\rho}} \mathbb{E}_P[\mu_T(x, w)]$.

Solving the distributionally robust stochastic black-box opt

Algorithm 1: DRBQO: Distributionally Robust Bayesian quadrature optimization

Input: Prior $\mathtt{GP}(\mu_0, k)$, horizon T, fixed sample set S_n, confidence radius $\rho \geq 0, C_0 = k$.

for $t = 1$ *to* T **do**

/* Posterior sampling */

Sample $\tilde{f}_t \sim \mathtt{GP}(\mu_{t-1}, C_{t-1})$.

/* A surrogate DR optimization */

Choose $x_t \in \arg\max_{x \in \mathcal{X}} \min_{P \in \mathcal{P}_{n,\rho}} \mathbb{E}_P[\tilde{f}_t(x, w)]$.

/* Highest posterior variance */

Choose $w_t = \arg\max_{w \in S_n} C_{t-1}(x_t, w; x_t, w)$.

Observe reward $\hat{y}_t \leftarrow f(x_t, w_t) + \epsilon_t$.

Perform update $\mathtt{GP}$ to get μ_t and C_t.

end

Output: $\arg\max_{x \in \{x_1, \ldots, x_T\}} \min_{P \in \mathcal{P}_{n,\rho}} \mathbb{E}_P[\mu_T(x, w)]$.

1. Build a Gaussian Process (GP) to model f

Solving the distributionally robust stochastic black-box opt

Algorithm 1: DRBQO: Distributionally Robust Bayesian quadrature optimization

Input: Prior $\mathtt{GP}(\mu_0, k)$, horizon T, fixed sample set S_n, confidence radius $\rho \geq 0, C_0 = k$.

```
for t = 1 to T do
    /* Posterior sampling */
    Sample f̃_t ~ GP(μ_{t-1}, C_{t-1}).
    /* A surrogate DR optimization */
    Choose x_t ∈ argmax_{x∈X} min_{P∈P_{n,ρ}} E_P[f̃_t(x, w)].
    /* Highest posterior variance */
    Choose w_t = argmax_{w∈S_n} C_{t-1}(x_t, w; x_t, w).
    Observe reward ŷ_t ← f(x_t, w_t) + ε_t.
    Perform update GP to get μ_t and C_t.
end
```

Output: $\arg\max_{x \in \{x_1, \ldots, x_T\}} \min_{P \in \mathcal{P}_{n,\rho}} \mathbb{E}_P[\mu_T(x, w)]$.

1. Build a Gaussian Process (GP) to model f
2. Thompson sampling to draw a surrogate function $\tilde{f}_t$ for f

Solving the distributionally robust stochastic black-box opt

Algorithm 1: DRBQO: Distributionally Robust Bayesian quadrature optimization

---

Input: Prior $\mathtt{GP}(\mu_0, k)$, horizon T, fixed sample set S_n, confidence radius $\rho \geq 0, C_0 = k$.

for $t = 1$ *to* T **do**

```
    /* Posterior sampling                          */
```
  Sample $\tilde{f}_t \sim \mathtt{GP}(\mu_{t-1}, C_{t-1})$.
```
    /* A surrogate DR optimization                 */
```
  Choose $x_t \in \arg\max_{x \in \mathcal{X}} \min_{P \in \mathcal{P}_{n,\rho}} \mathbb{E}_P[\tilde{f}_t(x, w)]$.
```
    /* Highest posterior variance                  */
```
  Choose $w_t = \arg\max_{w \in S_n} C_{t-1}(x_t, w; x_t, w)$.

  Observe reward $\hat{y}_t \leftarrow f(x_t, w_t) + \epsilon_t$.

  Perform update GP to get μ_t and C_t.

end

Output: $\arg\max_{x \in \{x_1, \ldots, x_T\}} \min_{P \in \mathcal{P}_{n,\rho}} \mathbb{E}_P[\mu_T(x, w)]$.

---

1. Build a <u>Gaussian Process (GP)</u> to model f
2. <u>Thompson sampling</u> to draw a <u>surrogate function</u> $\tilde{f}_t$ for f
3. Solve the <u>surrogate distributionally robust optimization</u> for $\tilde{f}_t$

Solving the distributionally robust stochastic black-box opt

Algorithm 1: DRBQO: Distributionally Robust Bayesian quadrature optimization

Input: Prior $\texttt{GP}(\mu_0, k)$, horizon T, fixed sample set S_n, confidence radius $\rho \geq 0, C_0 = k$.

```
for t = 1 to T do
    /* Posterior sampling                                  */
    Sample f̃_t ~ GP(μ_{t-1}, C_{t-1}).
    /* A surrogate DR optimization                         */
    Choose x_t ∈ argmax_{x∈X} min_{P∈P_{n,ρ}} E_P[f̃_t(x, w)].
    /* Highest posterior variance                          */
    Choose w_t = argmax_{w∈S_n} C_{t-1}(x_t, w; x_t, w).
    Observe reward ŷ_t ← f(x_t, w_t) + ε_t.
    Perform update GP to get μ_t and C_t.
end
```

Output: $\arg\max_{x \in \{x_1, \ldots, x_T\}} \min_{P \in \mathcal{P}_{n,\rho}} \mathbb{E}_P[\mu_T(x, w)]$.

1. Build a Gaussian Process (GP) to model f
2. Thompson sampling to draw a surrogate function $\tilde{f}_t$ for f
3. Solve the surrogate distributionally robust optimization for $\tilde{f}_t$
4. Sample new input and update the GP

Solving the distributionally robust stochastic black-box opt

Algorithm 1: DRBQO: Distributionally Robust Bayesian quadrature optimization

Input: Prior $\mathtt{GP}(\mu_0, k)$, horizon T, fixed sample set S_n, confidence radius $\rho \geq 0, C_0 = k$.

```
for t = 1 to T do
    /* Posterior sampling */
    Sample f̃_t ~ GP(μ_{t-1}, C_{t-1}).
    /* A surrogate DR optimization */
    Choose x_t ∈ argmax_{x∈X} min_{P∈P_{n,ρ}} E_P[f̃_t(x, w)].
    /* Highest posterior variance */
    Choose w_t = argmax_{w∈S_n} C_{t-1}(x_t, w; x_t, w).
    Observe reward ŷ_t ← f(x_t, w_t) + ε_t.
    Perform update GP to get μ_t and C_t.
end
```

Output: $\arg\max_{x \in \{x_1, \ldots, x_T\}} \min_{P \in \mathcal{P}_{n,\rho}} \mathbb{E}_P[\mu_T(x, w)]$.

1. Build a Gaussian Process (GP) to model f
2. Thompson sampling to draw a surrogate function $\tilde{f}_t$ for f
3. Solve the surrogate distributionally robust optimization for $\tilde{f}_t$
4. Sample new input and update the GP

→ simple yet highly flexible and robustness with theoretical guarantee!

DRBQO: Robustness with theoretical guarantee

TL;DR:

DRBQO can find the robust solution in sublinear time!

Theorem 2. *Assume $\mathcal{X}$ is a finite subset of $\mathbb{R}^d$, and $\mathcal{P}_{n,\rho}$ is a finite subset of the χ^2 ball of radius ρ. Let π^{DRBQO} be the DRBQO policy presented in Algorithm 1, γ_T be the maximum information gain defined in Srinivas et al. [2010], then for all $T \in \mathbb{N}$,*

$$BayesRegret(T, \pi^{DRBQO}) \leq 1 + \frac{\left(\sqrt{2\log\frac{(1+T^2)|\mathcal{X}||\mathcal{P}_{n,\rho}|}{\sqrt{2\pi}}} + B\right)\sqrt{2\pi}}{|\mathcal{X}||\mathcal{P}_{n,\rho}|} + \frac{2\gamma_T\sqrt{(1+2\rho)n}}{1+\sigma^{-2}} + 2\sqrt{T\gamma_T(1+\sigma^{-2})^{-1}\log\frac{(1+T^2)|\mathcal{X}||\mathcal{P}_{n,\rho}|}{\sqrt{2\pi}}}.$$

Experiments

Goal: To illustrate if DRBQO can successfully avoid *spurious* solutions and find the *robust* solution as compared to the BQO baselines

- **Synthetic experiment**: Maximizing the expected logistic function
- **Real-world experiment**: Hyperparameter optimization via cross-validation

Synthetic experiment: Setup

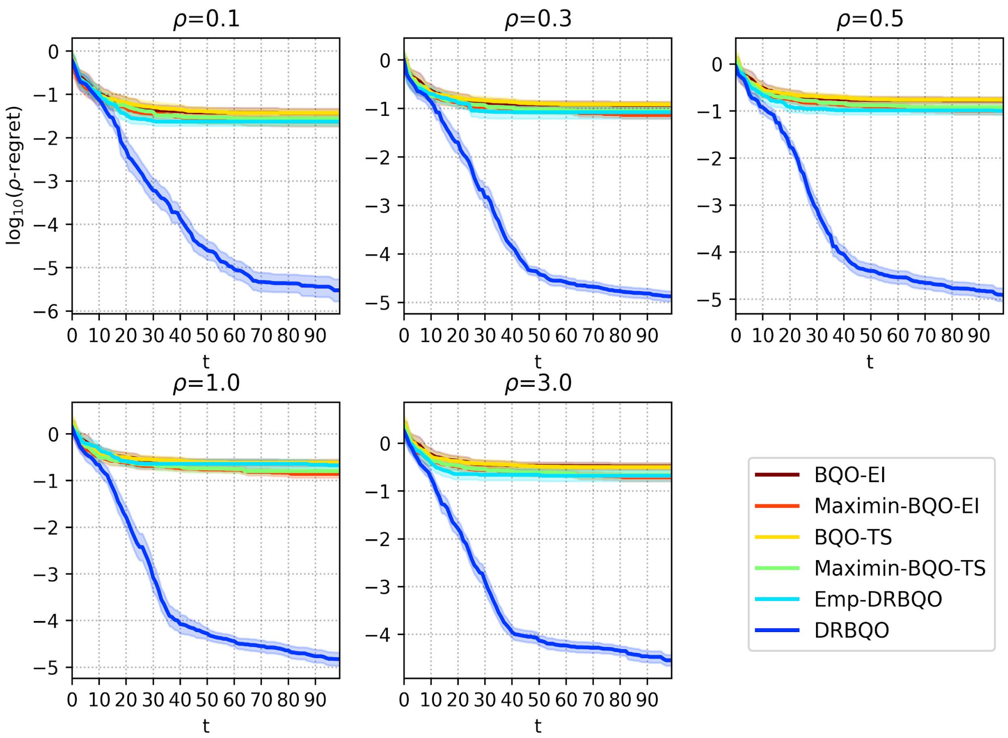
- **Goal**: Maximize the expected logistic function

$$g(x) = \mathbb{E}_{w \sim \mathcal{N}(0;I)} \left[-\log(1 + e^{x^T w}) \right]$$

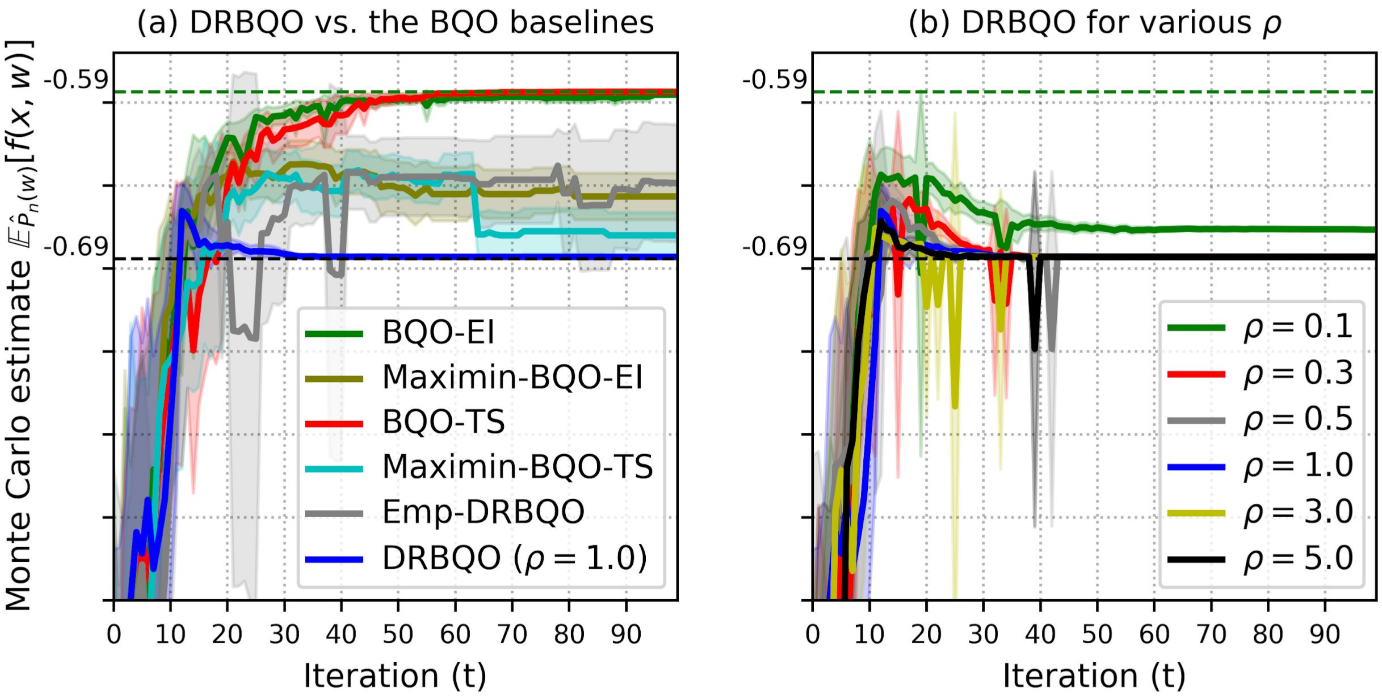
under **distributional uncertainty**

- **Distributional uncertainty**: We sample n=10 values $w \sim \mathcal{N}(0, I)$ and fix this set for the empirical distribution

Synthetic experiment: Results



Synthetic experiment: Results



Cross-validation hyperparameter tuning

- **Setting**: n-fold cross-validation hyperparameter tuning
- **Standard BQO**: optimize for the average n-fold values
- **DRBQO**: optimize under the most adversarial distribution of the n folds

Methods	ElasticNet	CNN
MTBO	8.576 ± 0.080	1.712 ± 0.263
BQO-EI	9.166 ± 0.433	1.634 ± 0.157
BQO-TS	8.625 ± 0.116	1.820 ± 0.227
DRBQO($\rho = 0.1$)	$\mathbf{8.450} \pm 0.022$	1.968 ± 0.310
DRBQO($\rho = 0.3$)	$\mathbf{8.505} \pm 0.082$	$\mathbf{1.495} \pm 0.106$
DRBQO($\rho = 0.5$)	$\mathbf{8.515} \pm 0.075$	1.869 ± 0.232
DRBQO($\rho = 1$)	$\mathbf{8.526} \pm 0.065$	$\mathbf{1.444} \pm 0.071$
DRBQO($\rho = 3$)	$\mathbf{8.387} \pm 0.013$	$\mathbf{1.374} \pm 0.066$
DRBQO($\rho = 5$)	$\mathbf{8.380} \pm 0.022$	$\mathbf{1.321} \pm 0.061$

Table 1: Classification error (%) of ElasticNet and CNN on the MNIST test set tuned by different algorithms. Each bold number in the DRBQO group denotes the classification error that is smaller than any corresponding number in the baseline group.

Conclusion

- **DRBQO** that efficiently seeks for the robust solution under the distributional uncertainty
- **DRBQO** = Distributionally robust optimization + Thompson sampling
- **DRBQO**: flexibility to control the conservativeness against distributional perturbation
- Empirical effectiveness + theoretical convergence via Bayesian regret
- More details @ https://arxiv.org/abs/2001.06814

Thank you for your listening!

Q & A