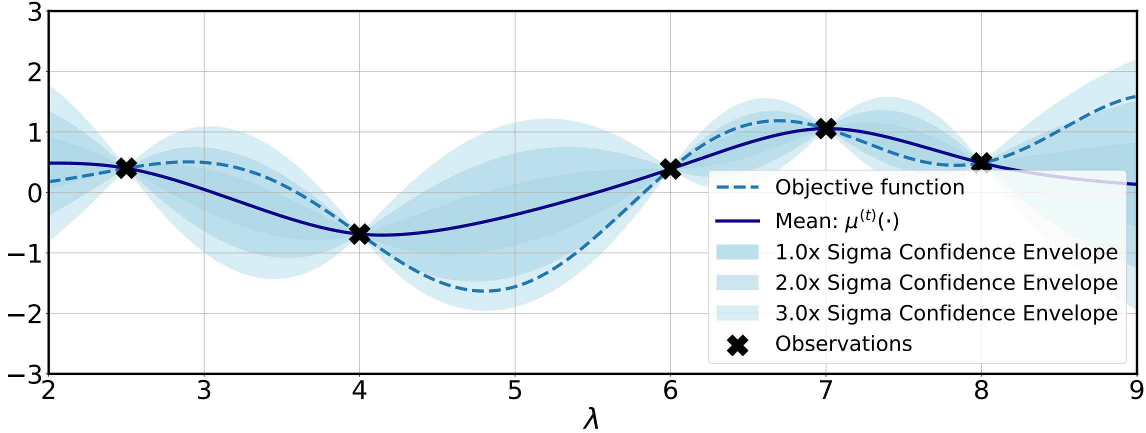
Bayesian Optimization for Hyperparameter Optimization

Computationally Expensive Acquisition Functions

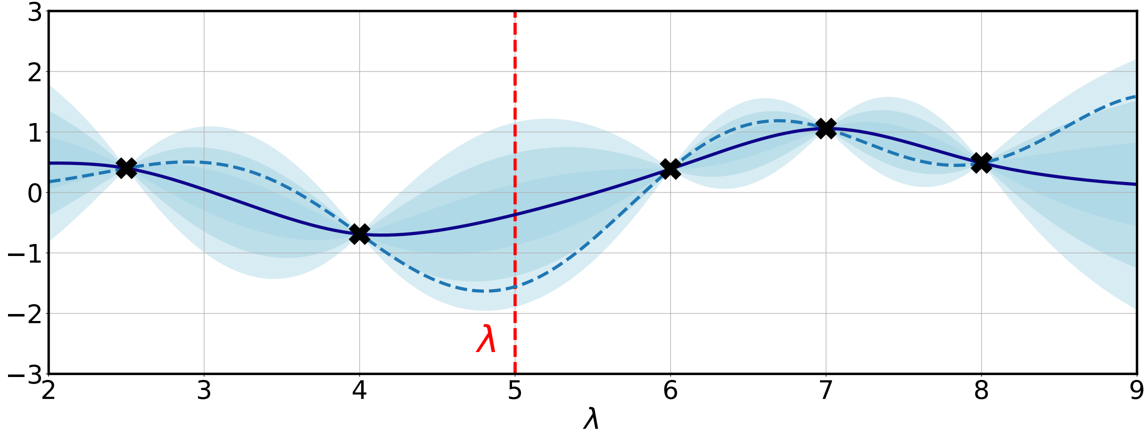
Bernd Bischl Frank Hutter Lars Kotthoff
Marius Lindauer Joaquin Vanschoren

A Computationally Expensive Step: One-Step Look Ahead



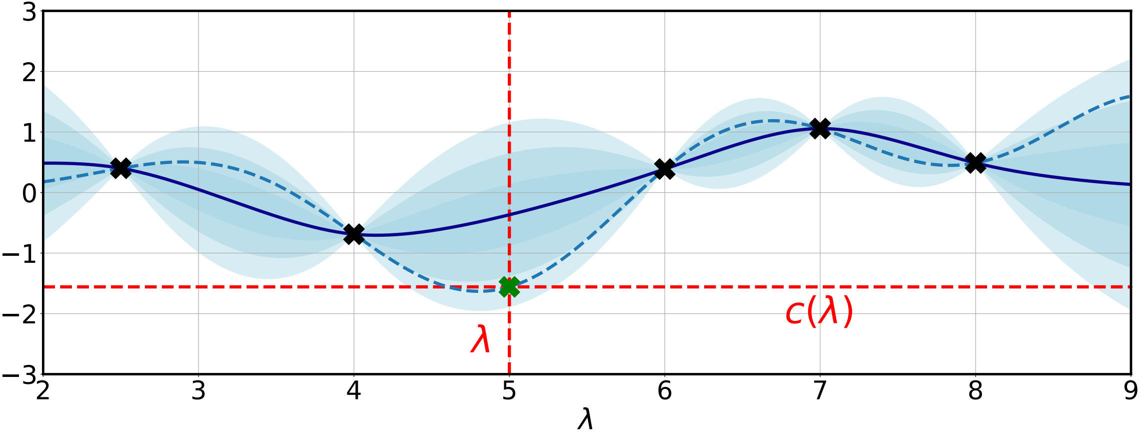
Given the surrogate $\hat{c}^{(t)}$ fit at iteration t

A Computationally Expensive Step: One-Step Look Ahead



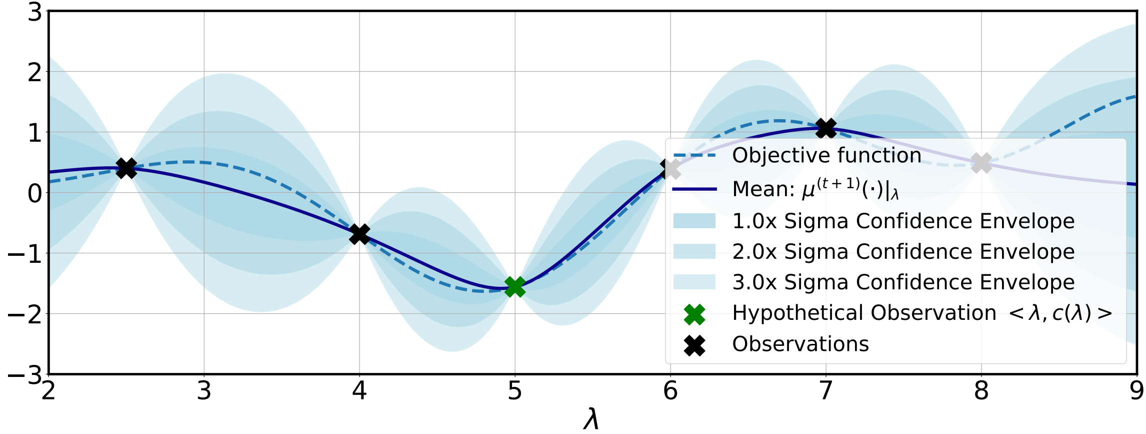
Imagine that we sample at a random configuration $\boldsymbol{\lambda}$

A Computationally Expensive Step: One-Step Look Ahead



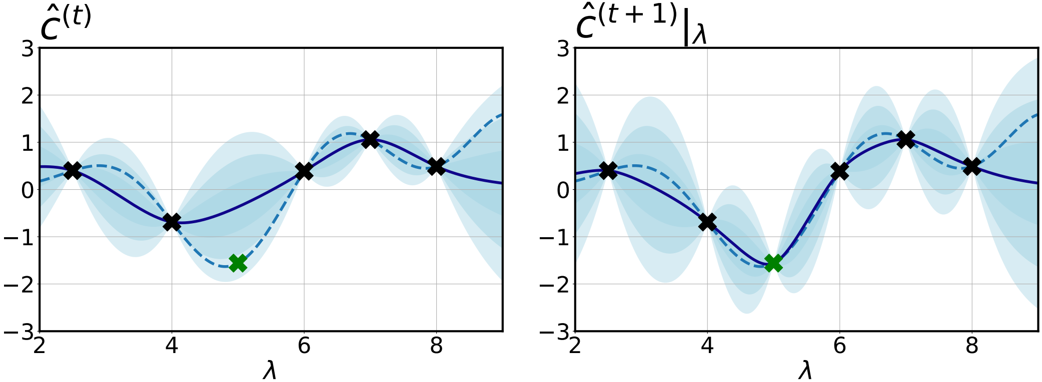
We would then observe the cost $c(\boldsymbol{\lambda})$ at this imaginary configuration $\boldsymbol{\lambda}$

A Computationally Expensive Step: One-Step Look Ahead



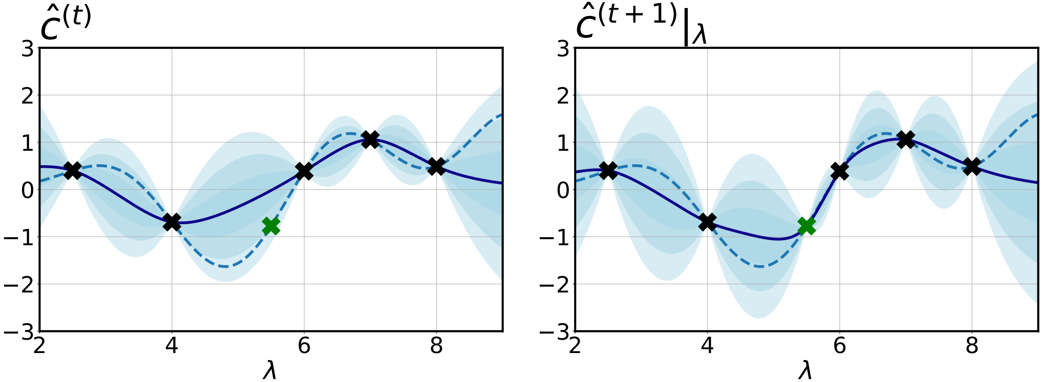
With this hypothetical data point at λ, we'd have this 1-step lookahead surrogate $\hat{c}^{(t+1)}|_{\boldsymbol{\lambda}}(\cdot)$

Visualization of How Different the Lookahead Surrogate Can Be



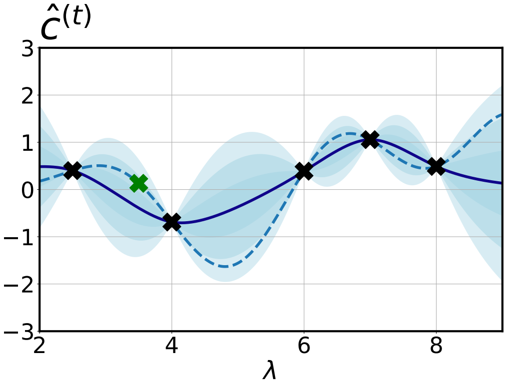
A comparison of $\hat{c}^{(t)}(\cdot)$ and $\hat{c}^{(t+1)}|_{\boldsymbol{\lambda}}(\cdot)$ for a given $\boldsymbol{\lambda}$.

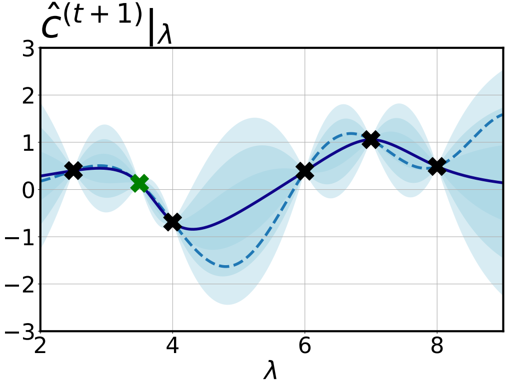
Visualization of How Different the Lookahead Surrogate Can Be



A comparison of $\hat{c}^{(t)}(\cdot)$ and $\hat{c}^{(t+1)}|_{\boldsymbol{\lambda}}(\cdot)$ for a given $\boldsymbol{\lambda}$.

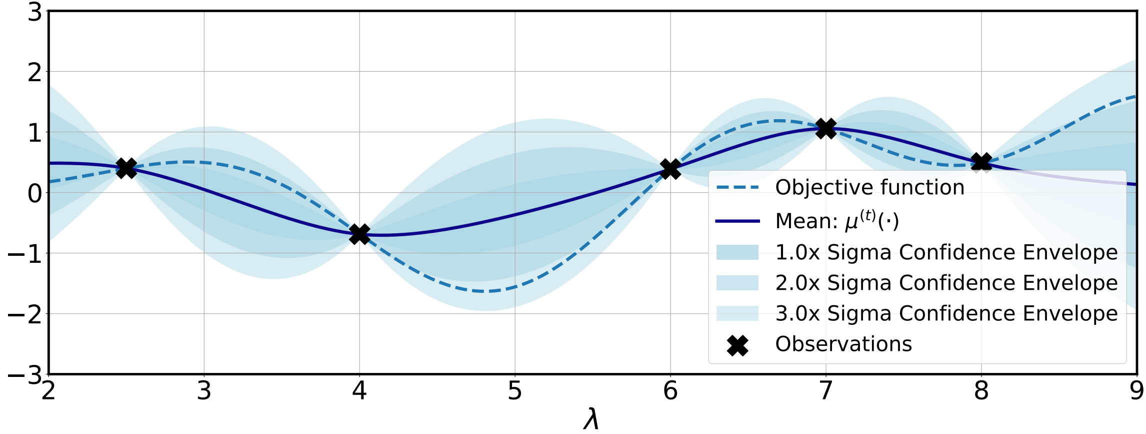
Visualization of How Different the Lookahead Surrogate Can Be





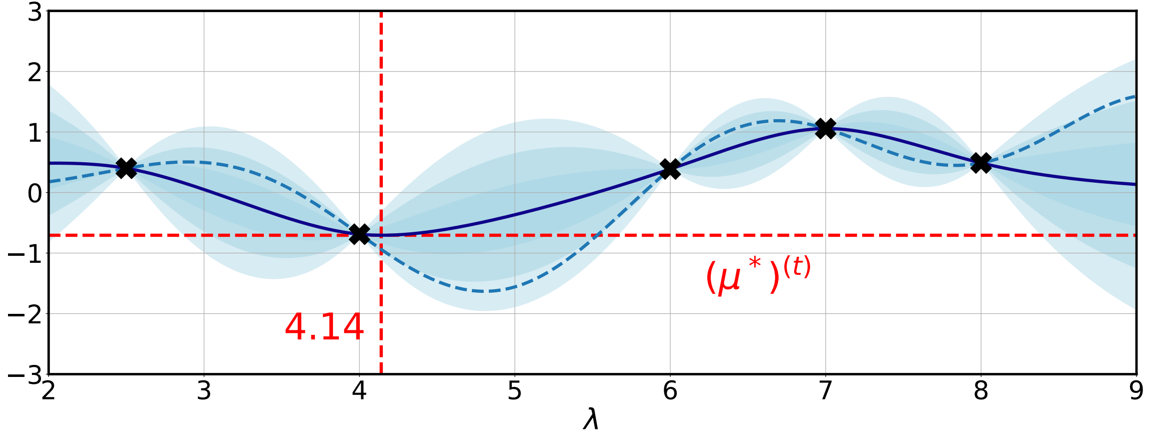
A comparison of $\hat{c}^{(t)}(\cdot)$ and $\hat{c}^{(t+1)}|_{\boldsymbol{\lambda}}(\cdot)$ for a given $\boldsymbol{\lambda}$.

Knowledge Gradient (KG): Concept



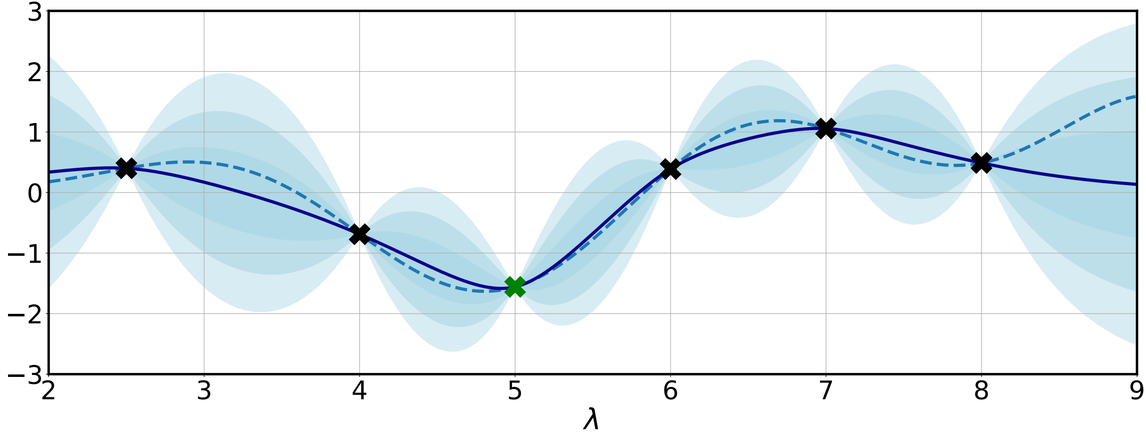
Given the surrogate $\hat{c}(\boldsymbol{\lambda}) = \mathcal{N}(\mu(\boldsymbol{\lambda}), \sigma^2(\boldsymbol{\lambda}))$ fit at iteration t

Knowledge Gradient (KG): Concept



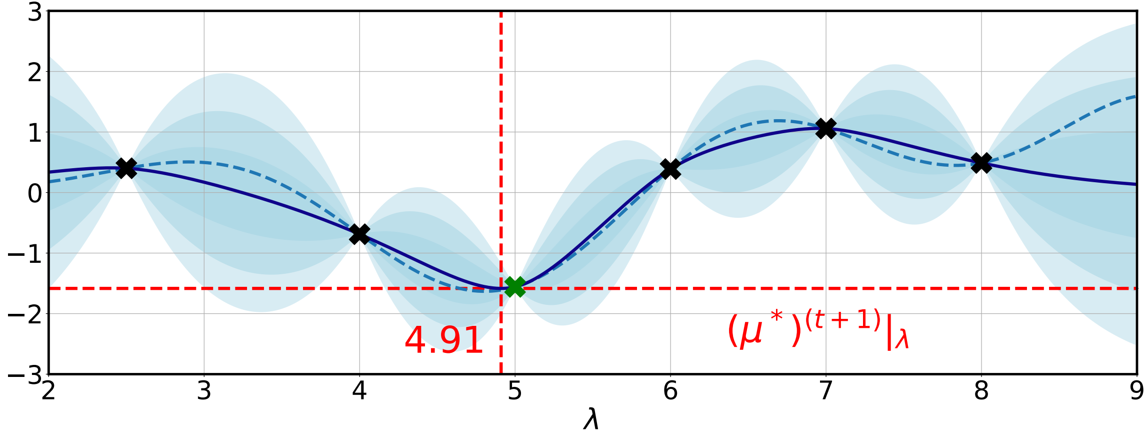
If we are risk-neutral, we'd return $\arg\min_{\boldsymbol{\lambda}} (\mu(\boldsymbol{\lambda}))^{(t)}$ as incumbent, with value $(\mu^*)^{(t)}$

Knowledge Gradient (KG): Concept



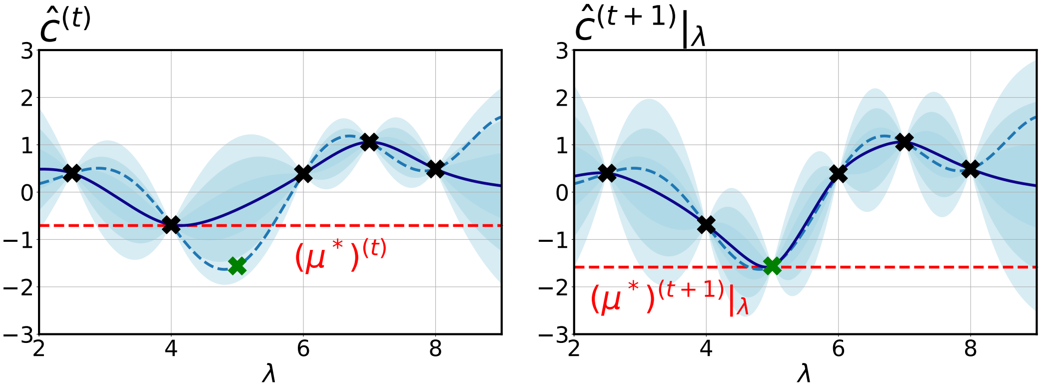
If we perform a one-step look-ahead for configuration $\boldsymbol{\lambda}$, we would get $\hat{c}^{(t+1)}|_{\boldsymbol{\lambda}}$

Knowledge Gradient (KG): Concept



We would then be interested in the minimum of the updated mean function $(\mu^*)^{(t+1)} \mid \boldsymbol{\lambda}$

Knowledge Gradient (KG): Concept



The Knowledge Gradient is then the expectation of the improvement $(\mu^*)^{(t+1)} - (\mu^*)^{(t+1)}|_{\boldsymbol{\lambda}}$

Knowledge Gradient (KG): Formal Definition

- The Knowledge Gradient is the expectation of the improvement $(\mu^*)^{(t+1)} - (\mu^*)^{(t+1)}|_{\boldsymbol{\lambda}}$:

$$u_{KG}^{(t)}(\boldsymbol{\lambda}) = \mathbb{E}\left[(\mu^*)^{(t)} - (\mu^*)^{(t+1)}\Big|_{\boldsymbol{\lambda}^{(t)}=\boldsymbol{\lambda}}\right]$$
$$= \min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}} \mu^{(t)}\left(\boldsymbol{\lambda}' \middle| \mathcal{D}^{(t-1)}\right) - \mathop{\mathbb{E}}_{\tilde{c}\sim\hat{c}(\lambda)^{(t)}}\left[\min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}} \mu^{(t+1)}\left(\boldsymbol{\lambda}' \mid \mathcal{D}^{(t-1)} \cup \{\langle \boldsymbol{\lambda}, \tilde{c}\rangle\}\right)\right]$$

Knowledge Gradient (KG): Formal Definition

- The Knowledge Gradient is the expectation of the improvement $(\mu^*)^{(t+1)} - (\mu^*)^{(t+1)}|_{\boldsymbol{\lambda}}$:

$$
\begin{aligned}
u_{KG}^{(t)}(\boldsymbol{\lambda}) &= \mathbb{E}\left[(\mu^*)^{(t)} - (\mu^*)^{(t+1)}\Big|_{\boldsymbol{\lambda}^{(t)}=\boldsymbol{\lambda}}\right] \\
&= \min_{\boldsymbol{\lambda}'\in\boldsymbol{\Lambda}} \mu^{(t)}\left(\boldsymbol{\lambda}'\middle|\mathcal{D}^{(t-1)}\right) - \mathop{\mathbb{E}}_{\tilde{c}\sim\hat{c}(\lambda)^{(t)}}\left[\min_{\boldsymbol{\lambda}'\in\boldsymbol{\Lambda}} \mu^{(t+1)}\left(\boldsymbol{\lambda}' \mid \mathcal{D}^{(t-1)} \cup \{\langle\boldsymbol{\lambda}, \tilde{c}\rangle\}\right)\right]
\end{aligned}
$$

$$
\boxed{\text{Choose } \boldsymbol{\lambda}^{(t)} = \arg\max_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}}(u_{KG}^{(t)}(\boldsymbol{\lambda}))}
$$

Knowledge Gradient: Pseudocode for Monte Carlo Approximation

$$u_{KG}^{(t)}(\boldsymbol{\lambda}) = const - \mathop{\mathbb{E}}_{\tilde{c} \sim \hat{c}(\lambda)^{(t)}} \left[\min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}} \mu^{(t+1)} \left(\boldsymbol{\lambda}' \mid \mathcal{D}^{(t-1)} \cup \{\langle \boldsymbol{\lambda}, \tilde{c} \rangle\} \right) \right]$$

Sampling Based Knowledge Gradient Acquisition Function

Require: Surrogate $\hat{c}$, candidate configuration $\boldsymbol{\lambda}$, dataset $\mathcal{D}$

Result : Utility $u(\boldsymbol{\lambda})$

```
for s = 1 to S do
    Sample c̃_s ~ ĉ(λ)
    Update ĉ with {⟨λ, c̃_s⟩} to yield ĉ_s = N(μ_s, σ_s^2)
    e[s] ← min_{λ'∈Λ} μ_s
u ← const − (1/S) Σ_{s=1}^{S} e[s]
```

for $s = 1$ **to** S **do**
- Sample $\tilde{c}_s \sim \hat{c}(\boldsymbol{\lambda})$
- Update $\hat{c}$ with $\{\langle \boldsymbol{\lambda}, \tilde{c}_s \rangle\}$ to yield $\hat{c}_s = \mathcal{N}(\mu_s, \sigma_s^2)$
- $e[s] \leftarrow \min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}} \mu_s$

$u \leftarrow const - \frac{1}{S} \sum_{s=1}^{S} e[s]$

Knowledge Gradient: Pseudocode for Monte Carlo Approximation

$$u_{KG}^{(t)}(\boldsymbol{\lambda}) = const - \underset{\tilde{c} \sim \hat{c}(\lambda)^{(t)}}{\mathbb{E}} \left[\min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}} \mu^{(t+1)} \left(\boldsymbol{\lambda}' \mid \mathcal{D}^{(t-1)} \cup \{\langle \boldsymbol{\lambda}, \tilde{c} \rangle\} \right) \right]$$

Sampling Based Knowledge Gradient Acquisition Function

Require: Surrogate $\hat{c}$, candidate configuration $\boldsymbol{\lambda}$, dataset $\mathcal{D}$

Result : Utility $u(\boldsymbol{\lambda})$

```
for s = 1 to S do
    Sample c̃_s ~ ĉ(λ)
    Update ĉ with {⟨λ, c̃_s⟩} to yield ĉ_s = N(μ_s, σ_s²)
    e[s] ← min_{λ'∈Λ} μ_s
u ← const − (1/S) Σ_{s=1}^{S} e[s]
```

This sampling view is useful for intuition;

but in practice, there are more efficient ways to optimize KG [Frazier 2018]

Entropy Search Preliminaries

- Key idea: Evaluate $\boldsymbol{\lambda}$ which most reduces our uncertainty about the location of $\boldsymbol{\lambda}^*$

Entropy Search Preliminaries

- Key idea: Evaluate $\boldsymbol{\lambda}$ which most reduces our uncertainty about the location of $\boldsymbol{\lambda}^*$

- We'll use the p_{min} distribution to characterize the location of $\boldsymbol{\lambda}^*$:

$$p_{min}(\boldsymbol{\lambda}^*|\mathcal{D}) = p(\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}} (\hat{c}(\boldsymbol{\lambda}')|\mathcal{D}))$$

Entropy Search Preliminaries

- Key idea: Evaluate $\boldsymbol{\lambda}$ which most reduces our uncertainty about the location of $\boldsymbol{\lambda}^*$

- We'll use the p_{min} distribution to characterize the location of $\boldsymbol{\lambda}^*$:

$$p_{min}(\boldsymbol{\lambda}^*|\mathcal{D}) = p(\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}}(\hat{c}(\boldsymbol{\lambda}')|\mathcal{D}))$$

- Our uncertainty is then captured by the entropy $H(p_{min}(\cdot|\mathcal{D}))$ of the p_{min} distribution

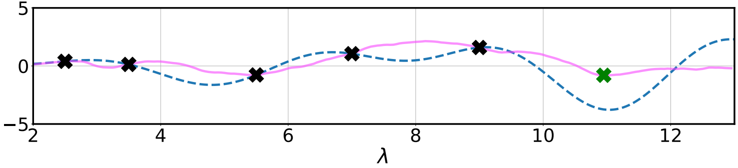
Entropy Search Preliminaries

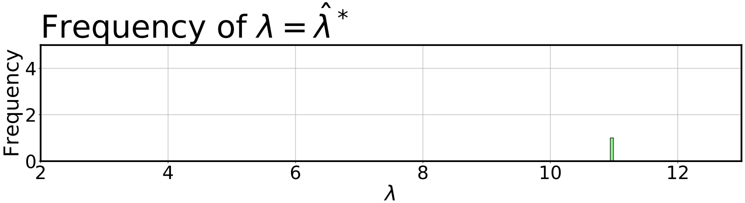
- Key idea: Evaluate $\boldsymbol{\lambda}$ which most reduces our uncertainty about the location of $\boldsymbol{\lambda}^*$
- We'll use the p_{min} distribution to characterize the location of $\boldsymbol{\lambda}^*$:

$$p_{min}(\boldsymbol{\lambda}^*|\mathcal{D}) = p(\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}}(\hat{c}(\boldsymbol{\lambda}')|\mathcal{D}))$$

- Our uncertainty is then captured by the entropy $H(p_{min}(\cdot|\mathcal{D}))$ of the p_{min} distribution
- Minimizing $H(p_{min}(\cdot|\mathcal{D}))$ yields a peaked p_{min} distribution, i.e., strong knowledge about the location of $\boldsymbol{\lambda}^*$

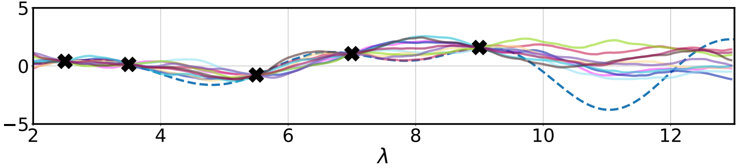
Entropy Search: Visualization of the p_{min} Distribution

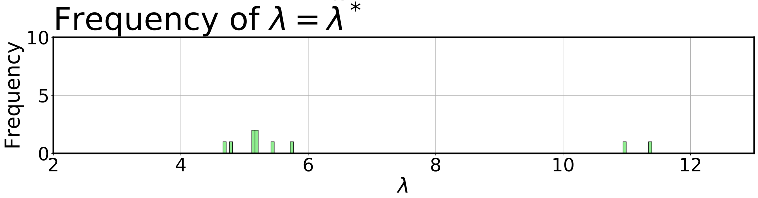




For each sample drawn from $\hat{c}$, we can compute where $\boldsymbol{\lambda}^*$ lies

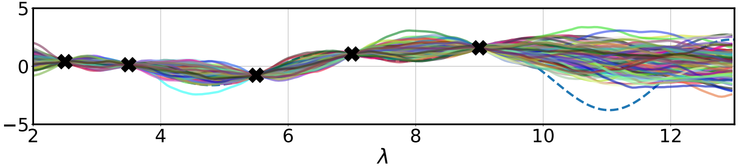
Entropy Search: Visualization of the p_{min} Distribution

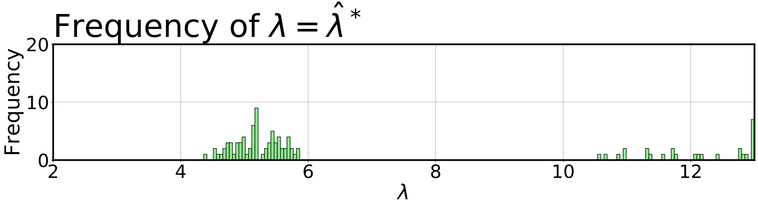




For each sample drawn from $\hat{c}$, we can compute where $\boldsymbol{\lambda}^*$ lies

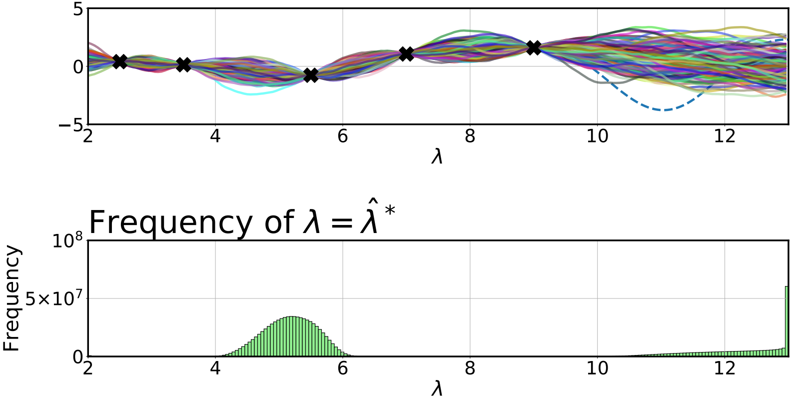
Entropy Search: Visualization of the p_{min} Distribution





From many samples we can approximate the p_{min} distribution

Entropy Search: Visualization of the p_{min} Distribution



From many samples we can approximate the p_{min} distribution

Entropy Search: Formal Definition

- The p_{min} distribution characterizes the location of $\boldsymbol{\lambda}^*$:

$$p_{min}(\boldsymbol{\lambda}^*|\mathcal{D}) = p(\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}}(\hat{c}(\boldsymbol{\lambda}')|\mathcal{D}))$$

Entropy Search: Formal Definition

- The p_{min} distribution characterizes the location of $\boldsymbol{\lambda}^*$:

$$p_{min}(\boldsymbol{\lambda}^*|\mathcal{D}) = p(\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}}(\hat{c}(\boldsymbol{\lambda}')|\mathcal{D}))$$

- Our uncertainty about the location of $\boldsymbol{\lambda}^*$ is captured by the entropy $H(p_{min}(\cdot|\mathcal{D}))$ of the p_{min} distribution

Entropy Search: Formal Definition

- The p_{min} distribution characterizes the location of $\boldsymbol{\lambda}^*$:

$$p_{min}(\boldsymbol{\lambda}^*|\mathcal{D}) = p(\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}}(\hat{c}(\boldsymbol{\lambda}')|\mathcal{D}))$$

- Our uncertainty about the location of $\boldsymbol{\lambda}^*$ is captured by the entropy $H(p_{min}(\cdot|\mathcal{D}))$ of the p_{min} distribution

- Entropy search aims to minimize $H(p_{min})$, to yield a peaked p_{min} distribution:

$$u_{ES}(\boldsymbol{\lambda}) = H(p_{min}(\cdot|\mathcal{D})) - \mathop{\mathbb{E}}_{\tilde{c} \sim \hat{c}(\lambda)^{(t)}} H(p_{min}(\cdot|\mathcal{D} \cup \{\langle \boldsymbol{\lambda}, \tilde{c} \rangle\}))$$

$$\boxed{\text{Choose } \boldsymbol{\lambda}^{(t)} = \arg\max_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}}(u_{ES}^{(t)}(\boldsymbol{\lambda}))}$$

Entropy Search: Pseudocode for Monte Carlo Approximation

$$u_{ES}(\boldsymbol{\lambda}) = const - \mathop{\mathbb{E}}_{\tilde{c} \sim \hat{c}(\lambda)^{(t)}} H(p_{min}(\cdot | \mathcal{D} \cup \{\langle \boldsymbol{\lambda}, \tilde{c} \rangle\}))$$

Sampling Based Entropy Search Acquisition Function

Require : Surrogate $\hat{c}$, candidate configuration $\boldsymbol{\lambda}$, finite set of representer points $\boldsymbol{\Lambda}_r$, dataset $\mathcal{D}$

Result : Utility $u(\boldsymbol{\lambda})$

for $s = 1$ **to** S **do**

  Sample $\tilde{c}_s \sim \hat{c}(\boldsymbol{\lambda})$; $\hat{c}_s \leftarrow$ Update $\hat{c}$ with $\{\langle \boldsymbol{\lambda}, \tilde{c}_s \rangle\}$

  Initialize $F[\boldsymbol{\lambda}] = 0 \quad \forall \boldsymbol{\lambda}' \in \boldsymbol{\Lambda}_r$

  for $n = 1$ **to** N **do**

    Sample $g_n \sim \hat{c}_s$

    $\boldsymbol{\lambda}_s \leftarrow \arg\min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}_r} g_n$

    $F[\boldsymbol{\lambda}_s] \leftarrow F[\boldsymbol{\lambda}_s] + 1$

  $p_{min,s}(\boldsymbol{\lambda}') \leftarrow F_{\boldsymbol{\lambda}'}/N \quad \forall \boldsymbol{\lambda}' \in \boldsymbol{\Lambda}_r$

  $H_s \leftarrow H(p_{min,s})$, computed as $-\sum_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}_r} p_{min,s}(\boldsymbol{\lambda}') \log p_{min,s}(\boldsymbol{\lambda}')$

$u \leftarrow const - \frac{1}{S} \sum_{s=1}^{S} H_s$

Entropy Search: Variations

- The sample-based approximation is slow; for a faster approximation with expectation propagation see the original ES paper [Hennig et al. 2012]
- Predictive Entropy Search [Hernández-Lobato et al. 2014] is a frequently-used equivalent formulation that gives rise to more convenient approximations
- Max-Value Entropy Search [Wang and Jegelka 2017] is a recent variant that is cheaper to compute and has similar behavior
- Further reading and summary for ES: [Metzen 2016]

Questions to Answer for Yourself / Discuss with Friends

- Repetition. Describe the similarities and differences between KG and EI.
- Discussion. When is there an incentive for entropy search to sample at $\max(p_{min})$?