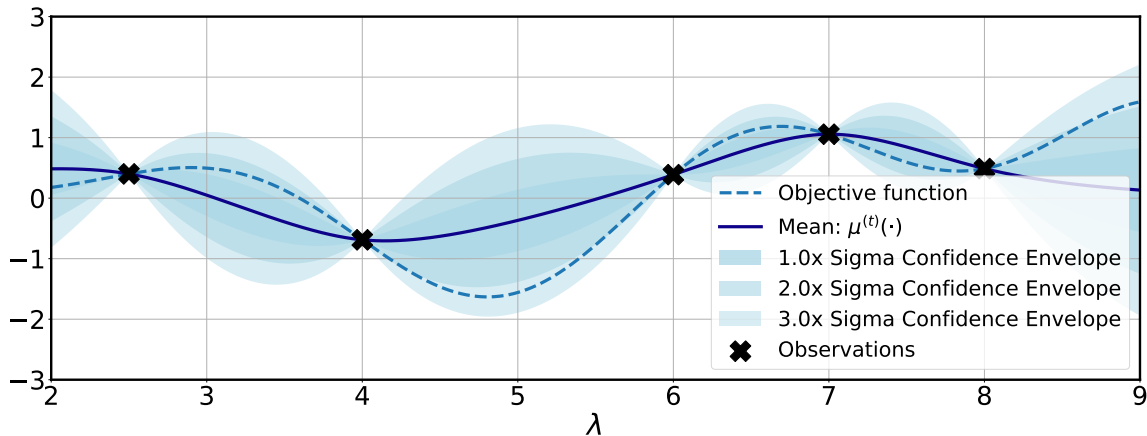


AutoML: Bayesian Optimization for HPO

Computationally Expensive Acquisition Functions

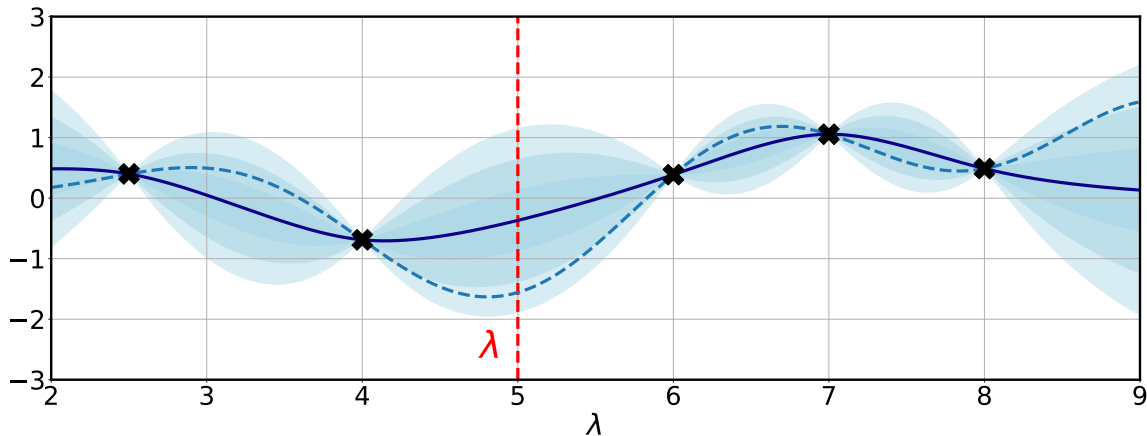
Bernd Bischl Frank Hutter Lars Kotthoff
Marius Lindauer Joaquin Vanschoren

A Computationally Expensive Step: One-Step Look Ahead



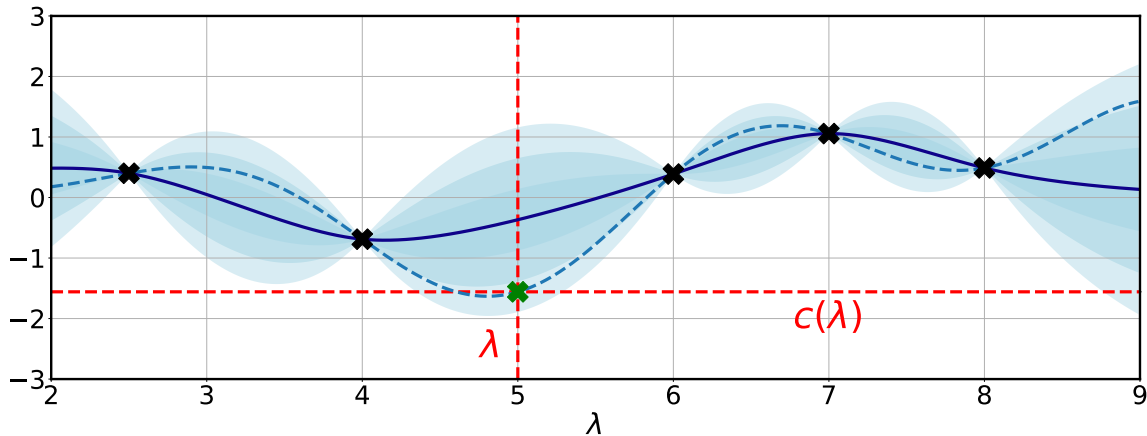
Given the surrogate $\hat{c}^{(t)}$ fit at iteration t

A Computationally Expensive Step: One-Step Look Ahead



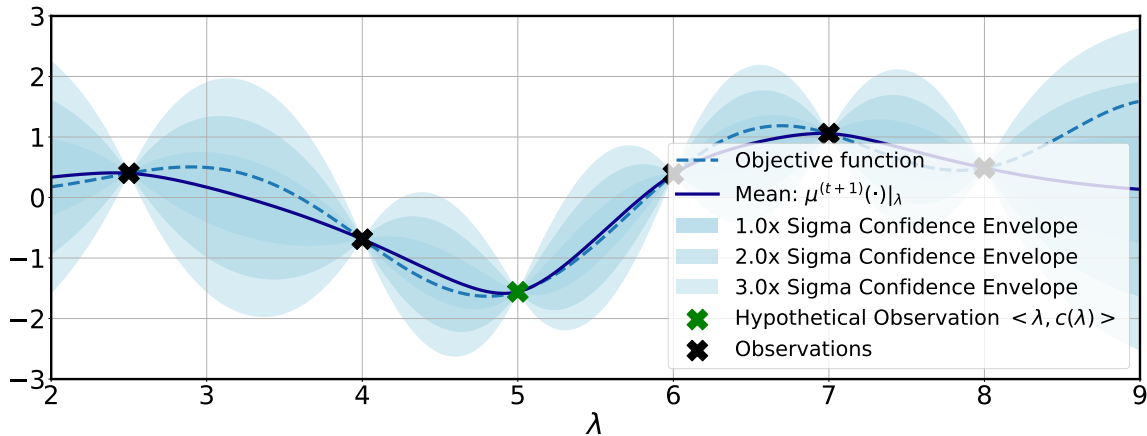
Imagine that we sample at a random configuration λ

A Computationally Expensive Step: One-Step Look Ahead



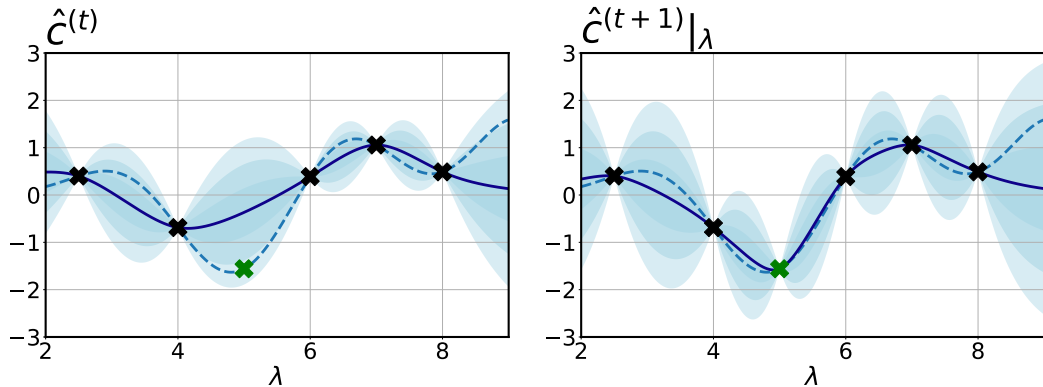
We would then observe the cost $c(\lambda)$ at this imaginary configuration λ

A Computationally Expensive Step: One-Step Look Ahead



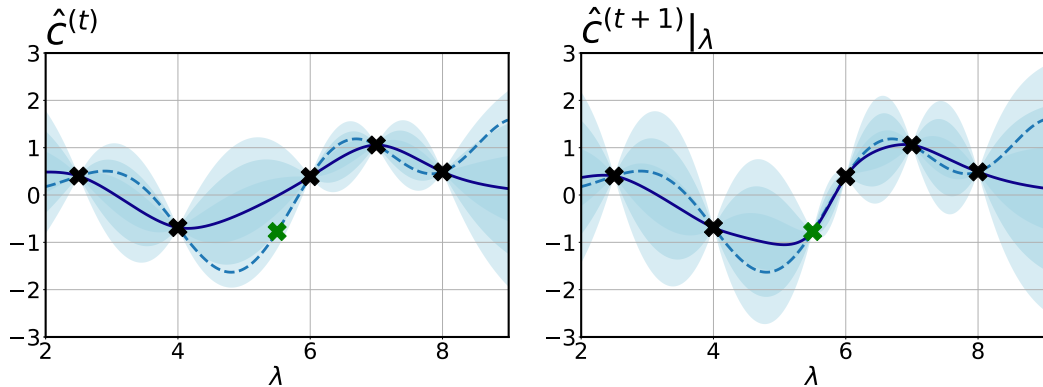
With this hypothetical data point at λ , we'd have this 1-step lookahead surrogate $\hat{c}^{(t+1)}|_{\lambda}(\cdot)$

Visualization of How Different the Lookahead Surrogate Can Be



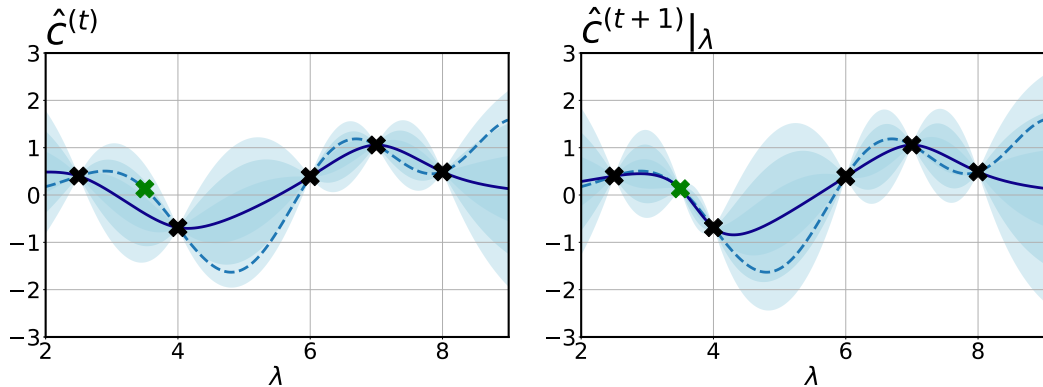
A comparison of $\hat{c}^{(t)}(\cdot)$ and $\hat{c}^{(t+1)}|_{\lambda}(\cdot)$ for a given λ .

Visualization of How Different the Lookahead Surrogate Can Be



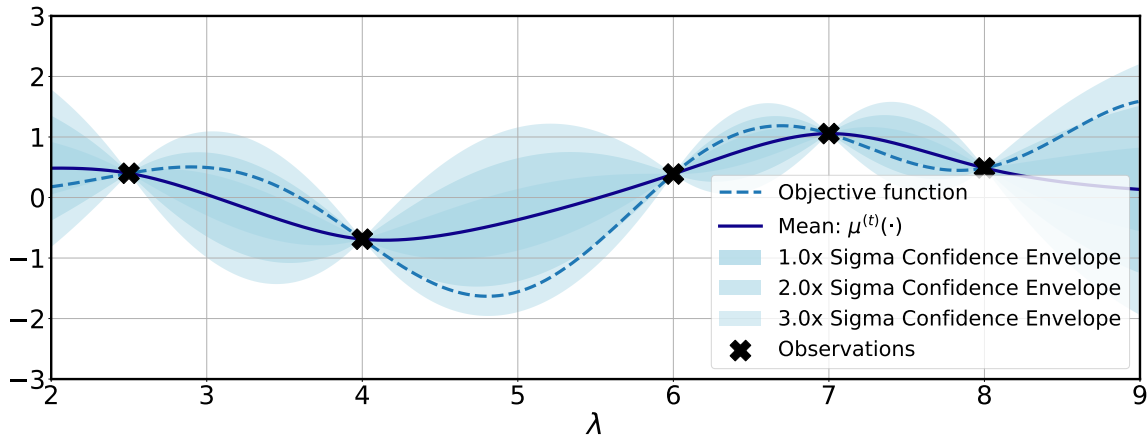
A comparison of $\hat{c}^{(t)}(\cdot)$ and $\hat{c}^{(t+1)}|_{\lambda}(\cdot)$ for a given λ .

Visualization of How Different the Lookahead Surrogate Can Be



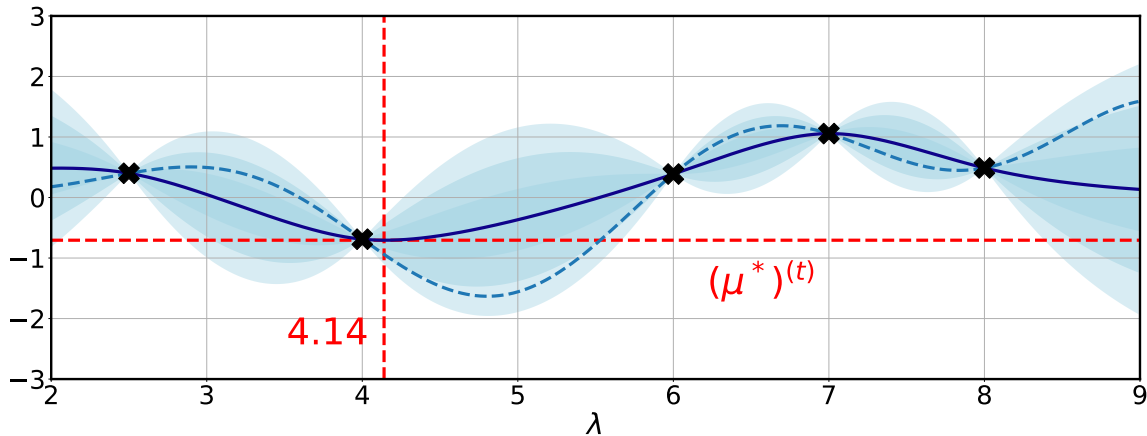
A comparison of $\hat{c}^{(t)}(\cdot)$ and $\hat{c}^{(t+1)}|_{\lambda}(\cdot)$ for a given λ .

Knowledge Gradient (KG): Concept



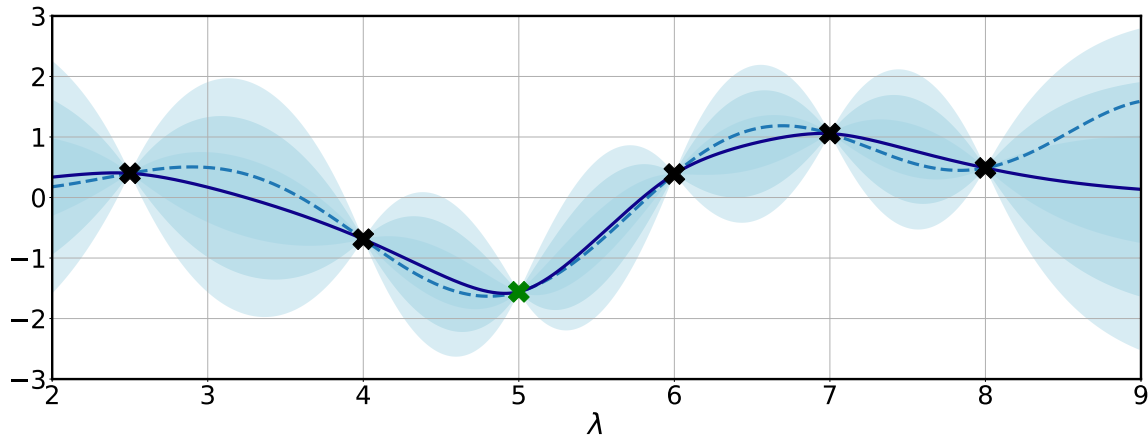
Given the surrogate $\hat{c}(\lambda) = \mathcal{N}(\mu(\lambda), \sigma^2(\lambda))$ fit at iteration t

Knowledge Gradient (KG): Concept



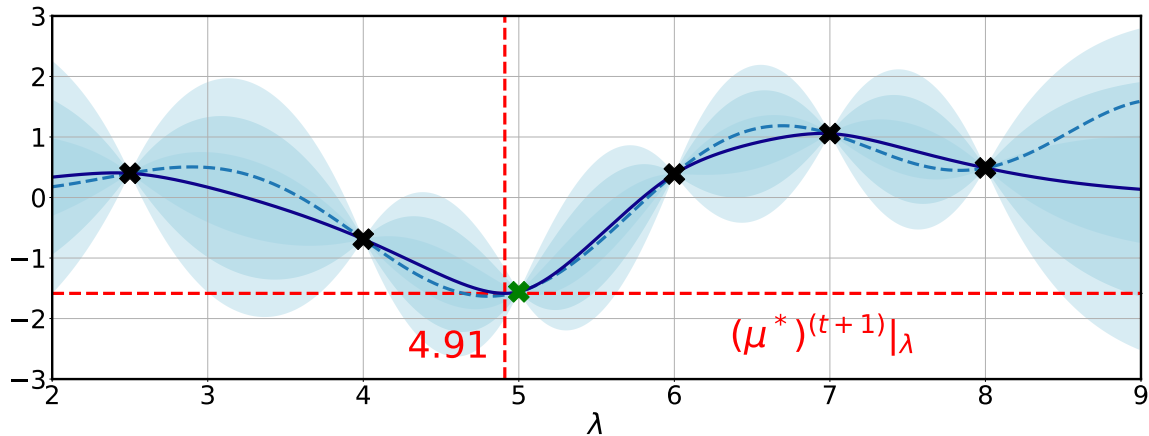
If we are risk-neutral, we'd return $\arg \min_{\lambda} (\mu(\lambda))^{(t)}$ as incumbent, with value $(\mu^*)^{(t)}$

Knowledge Gradient (KG): Concept



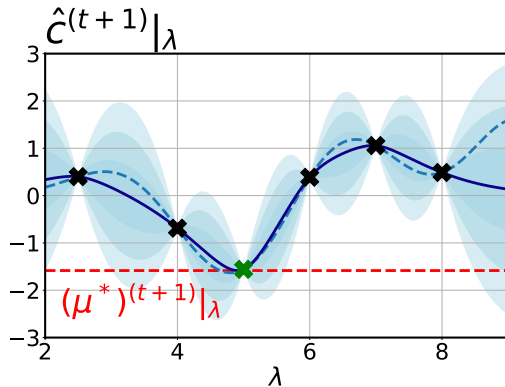
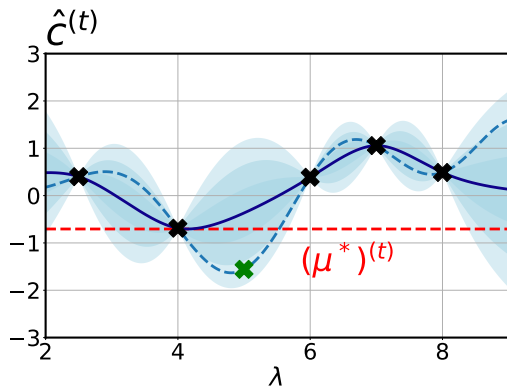
If we perform a one-step look-ahead for configuration λ , we would get $\hat{c}^{(t+1)}|_{\lambda}$

Knowledge Gradient (KG): Concept



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

Knowledge Gradient (KG): Concept



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

Knowledge Gradient (KG): Formal Definition

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

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

Knowledge Gradient (KG): Formal Definition

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

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

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

Knowledge Gradient: Pseudocode for Monte Carlo Approximation

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

Sampling Based Knowledge Gradient Acquisition Function

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

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

- 1 **for** $s = 1$ **to** S **do**
 - 2 Sample $\tilde{c}_s \sim \hat{c}(\boldsymbol{\lambda})$
 - 3 Update \hat{c} with $\{\langle \boldsymbol{\lambda}, \tilde{c}_s \rangle\}$ to yield $\hat{c}_s = \mathcal{N}(\mu_s, \sigma_s^2)$
 - 4 $e[s] \leftarrow \min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}} \mu_s$
 - 5 $u \leftarrow \text{const} - \frac{1}{S} \sum_{s=1}^S e[s]$
-

Knowledge Gradient: Pseudocode for Monte Carlo Approximation

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

Sampling Based Knowledge Gradient Acquisition Function

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

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

- 1 **for** $s = 1$ **to** S **do**
 - 2 Sample $\tilde{c}_s \sim \hat{c}(\boldsymbol{\lambda})$
 - 3 Update \hat{c} with $\{\langle \boldsymbol{\lambda}, \tilde{c}_s \rangle\}$ to yield $\hat{c}_s = \mathcal{N}(\mu_s, \sigma_s^2)$
 - 4 $e[s] \leftarrow \min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}} \mu_s$
 - 5 $u \leftarrow \text{const} - \frac{1}{S} \sum_{s=1}^S e[s]$
-

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 λ which most reduces our uncertainty about the location of λ^*

Entropy Search Preliminaries

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

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

Entropy Search Preliminaries

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

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

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

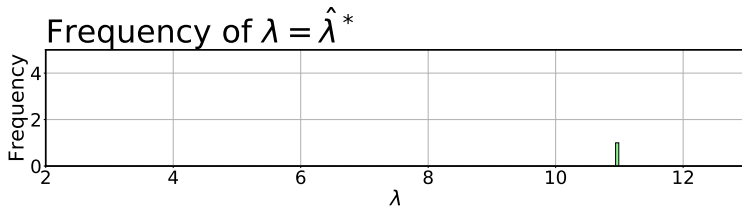
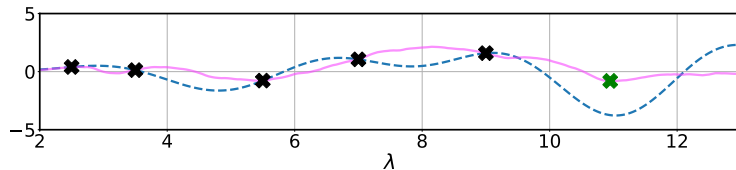
Entropy Search Preliminaries

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

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

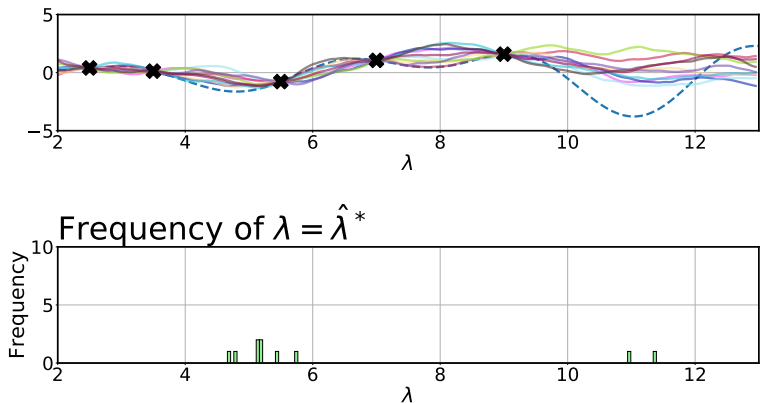
- Our uncertainty is then captured by the entropy $H(p_{min}(\cdot|\mathcal{D}))$ of the p_{min} distribution
- Minimizing $H(p_{min}(\cdot|\mathcal{D}))$ yields a peaked p_{min} distribution, i.e., strong knowledge about the location of λ^*

Entropy Search: Visualization of the p_{min} Distribution



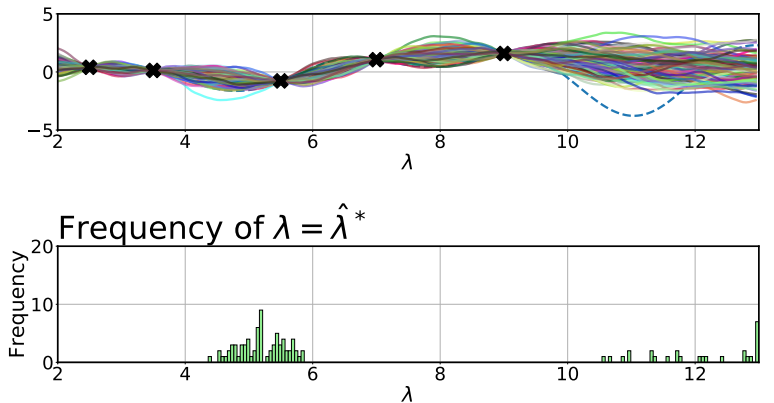
For each sample drawn from \hat{c} , we can compute where λ^* lies

Entropy Search: Visualization of the p_{min} Distribution



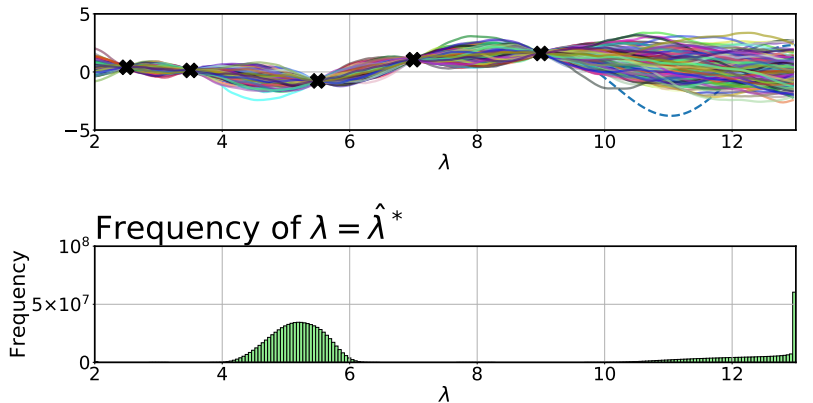
For each sample drawn from \hat{c} , we can compute where λ^* lies

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 λ^* :

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

Entropy Search: Formal Definition

- The p_{min} distribution characterizes the location of λ^* :

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

- Our uncertainty about the location of λ^* 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 λ^* :

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

- Our uncertainty about the location of λ^* 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:

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

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

Entropy Search: Pseudocode for Monte Carlo Approximation

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

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 $u(\boldsymbol{\lambda})$

```
1 for  $s = 1$  to  $S$  do
2   Sample  $\tilde{c}_s \sim \hat{c}(\boldsymbol{\lambda})$ ;  $\hat{c}_s \leftarrow$  Update  $\hat{c}$  with  $\{\langle \boldsymbol{\lambda}, \tilde{c}_s \rangle\}$ 
3   Initialize  $F[\boldsymbol{\lambda}] = 0 \quad \forall \boldsymbol{\lambda}' \in \boldsymbol{\Lambda}_r$ 
4   for  $n = 1$  to  $N$  do
5     Sample  $g_n \sim \hat{c}_s$ 
6      $\boldsymbol{\lambda}_s \leftarrow \arg \min_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}_r} g_n$ 
7      $F[\boldsymbol{\lambda}_s] \leftarrow F[\boldsymbol{\lambda}_s] + 1$ 
8    $p_{\min,s}(\boldsymbol{\lambda}') \leftarrow F[\boldsymbol{\lambda}_s] / N \quad \forall \boldsymbol{\lambda}' \in \boldsymbol{\Lambda}_r$ 
9    $H_s \leftarrow H(p_{\min,s})$ , computed as  $-\sum_{\boldsymbol{\lambda}' \in \boldsymbol{\Lambda}_r} p_{\min,s}(\boldsymbol{\lambda}') \log p_{\min,s}(\boldsymbol{\lambda}')$ 
10  $u \leftarrow \text{const} - \frac{1}{S} \sum_{s=1}^S H_s$ 
```

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 $\max(p_{min})$?