Gaussian Process Methods in Machine Learning

Jonathan Scarlett
scarlett@comp.nus.edu.sg

Lecture 3: Advanced Bayesian Optimization Methods

CS6216, Semester 1, AY2021/22

Outline of Lectures

- Lecture 0: Bayesian Modeling and Regression
- Lecture 1: Gaussian Processes, Kernels, and Regression
- Lecture 2: Optimization with Gaussian Processes
- **Lecture 3: Advanced Bayesian Optimization Methods**
- Lecture 4: GP Methods in Non-Bayesian Settings

Outline: This Lecture

- This lecture
  1. Practical twists on Bayesian optimization
  2. Level-set estimation
  3. One-step lookahead algorithms
  4. Truncated variance reduction

Recap 1: Black-Box Function Optimization

Black-box **function optimization:**

$$\mathbf{x}^\star \in \arg\max_{\mathbf{x} \in D \subseteq \mathbb{R}^d} f(\mathbf{x})$$

- Setting:
  - Unknown "reward" function f
  - Expensive evaluations of f
  - Noisy evaluations
  - Choose $\mathbf{x}_t$ based on $\{(\mathbf{x}_{t'}, y_{t'})\}_{t'<t}$

$$y_t = f(\mathbf{x_t}) + z_t$$

$$z_t \sim N(0, \sigma^2)$$

Recap 2: Bayesian Optimization (BO) Template

A general BO template [Shahriari *et al.*, 2016]

for $t = 1, 2, \dots, T$ **do**

  choose new $\mathbf{x}_t$ by optimizing an ***acquisition function*** $\alpha(\cdot)$

$$\mathbf{x}_t \in \arg\max_{\mathbf{x} \in D} \alpha(\mathbf{x}; \mathcal{D}_{t-1})$$

  where $\mathcal{D}_{t-1}$ is the data collected up to time $t-1$

  query objective function f to obtain $y_t = f(\mathbf{x}_t) + z_t$

  augment data $\mathcal{D}_t = \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, y_t)\}$

  update the GP model

end for

make final recommendation $\hat{\mathbf{x}}$ *(if considering simple regret)*

Twists

Practical variations along the same theme:

Pointwise costs: *Choosing point* $\mathbf{x}$ *incurs a cost* $c(\mathbf{x})$ [Snoek *et al.*, 2012]

- **Examples:** Advertising costs, sensor power consumption

Twists

Practical variations along the same theme:

Heteroscedastic noise: *Choosing point* $\mathbf{x}$ *incurs noise* $\sigma^2(\mathbf{x})$ [Goldberg *et al.*, 1997]

- **Example:** Different sensing quality

Twists

Practical variations along the same theme:

Multi-fidelity: *Alternative evaluations* $f_1, \dots, f_K$ *related to* f [Swersky *et al.*, 2013]

- **Example:** Varying data set sizes in automated machine learning

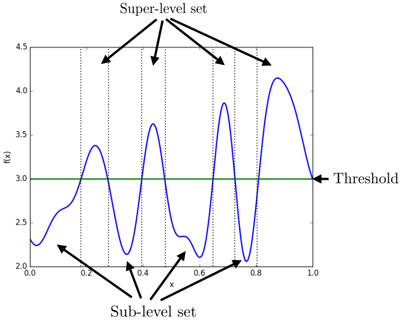
Another Twist: Level-Set Estimation

Level-set estimation: *Estimate the super- and sub-level sets* [Gotovos *et al.*, 2013]

$$S_{\text{super}}(f) := \{\mathbf{x} : f(\mathbf{x}) > h\}, \qquad S_{\text{sub}}(f) := \{\mathbf{x} : f(\mathbf{x}) < h\}$$

for some threshold h

- **Example:** Find all hotspots in environmental monitoring



Accommodating the BO Twists: Lookahead Algorithms

- Mostly heuristic BO approaches
  - Entropy search (ES): [Hennig *et al.*, 2012]

$$\mathbf{x}_t \approx \arg\min_{\mathbf{x}_t \in D} \mathbb{E}_{y_t}\left[H(\mathbf{x}^\star \mid \{\mathbf{x}_i, y_i\}_{i=1}^t)\right]$$

H: entropy function

Accommodating the BO Twists: Lookahead Algorithms

- Mostly heuristic BO approaches
  - Entropy search (ES): [Hennig *et al.*, 2012]

$$\mathbf{x}_t \approx \arg\min_{\mathbf{x}_t \in D} \mathbb{E}_{y_t}\Big[H(\mathbf{x}^\star \mid \{\mathbf{x}_i, y_i\}_{i=1}^t)\Big]$$

H: entropy function

  - Minimum regret search (MRS): [Metzen, 2016]

$$\mathbf{x}_t \approx \arg\min_{\mathbf{x} \in D} \mathbb{E}_{y_t}\left[\mathbb{E}_{\mathbf{x}^*}\left[\text{regret} \,\middle|\, \{\mathbf{x}_i, y_i\}_{i=1}^t\right]\right]$$

Accommodating the BO Twists: Lookahead Algorithms

- Mostly heuristic BO approaches
  - Entropy search (ES): [Hennig *et al.*, 2012]

$$\mathbf{x}_t \approx \arg\min_{\mathbf{x}_t \in D} \mathbb{E}_{y_t}\left[H(\mathbf{x}^\star \mid \{\mathbf{x}_i, y_i\}_{i=1}^t)\right]$$

H: entropy function

  - Minimum regret search (MRS): [Metzen, 2016]

$$\mathbf{x}_t \approx \arg\min_{\mathbf{x} \in D} \mathbb{E}_{y_t}\left[\mathbb{E}_{\mathbf{x}^*}\left[\text{regret} \,\middle|\, \{\mathbf{x}_i, y_i\}_{i=1}^t\right]\right]$$

  - Multi-step lookahead: approximation of the ideal lookahead loss function [Osborne *et al.*, 2009, Gonzalez *et al.*, 2016]

Accommodating the BO Twists: Lookahead Algorithms

- Mostly heuristic BO approaches
  - Entropy search (ES): [Hennig *et al.*, 2012]

$$\mathbf{x}_t \approx \arg\min_{\mathbf{x}_t \in D} \mathbb{E}_{y_t}\left[H(\mathbf{x}^\star \mid \{\mathbf{x}_i, y_i\}_{i=1}^t)\right]$$

H: entropy function

  - Minimum regret search (MRS): [Metzen, 2016]

$$\mathbf{x}_t \approx \arg\min_{\mathbf{x} \in D} \mathbb{E}_{y_t}\left[\mathbb{E}_{\mathbf{x}^*}\left[\text{regret} \,\Big|\, \{\mathbf{x}_i, y_i\}_{i=1}^t\right]\right]$$

  - Multi-step lookahead: approximation of the ideal lookahead loss function [Osborne *et al.*, 2009, Gonzalez *et al.*, 2016]

Advantages: Versatility with point-wise costs, non-uniform noise, multi-fidelity scenarios; can improve on baseline algorithms even without these twists.

Disadvantages: Expensive to compute; no theory; no LSE

Note:

Lookahead algorithms tend to be more versatile with respect to interesting twists on the optimization problem

- **Example.**
  - Minimize entropy $\iff$ maximize reduction in entropy
  - Extension: Maximize reduction in entropy per unit cost

More on Entropy Search

- Entropy search and its variants are particularly popular:

$$\mathbf{x}_t \approx \arg\min_{\mathbf{x}_t \in D} \mathbb{E}_{y_t}\left[H(\mathbf{x}^\star \,|\, \{\mathbf{x}_i, y_i\}_{i=1}^t) \right]$$

H: entropy function

  - Interpretation: Choose the point that makes us least uncertain (i.e., minimizes entropy) about the optimizer $\mathbf{x}^*$

More on Entropy Search

- Entropy search and its variants are particularly popular:

$$\mathbf{x}_t \approx \arg\min_{\mathbf{x}_t \in D} \mathbb{E}_{y_t}\left[H(\mathbf{x}^\star \mid \{\mathbf{x}_i, y_i\}_{i=1}^t)\right]$$

H: entropy function

  - Interpretation: Choose the point that makes us least uncertain (i.e., minimizes entropy) about the optimizer $\mathbf{x}^*$

- **Difficulty.** Cannot compute $\mathbb{E}_{y_t}\left[H(\mathbf{x}^\star \mid \{\mathbf{x}_i, y_i\}_{i=1}^t)\right]$ exactly
  - Need to approximate, typically using Monte Carlo methods
  - Particularly difficult for higher dimensions, e.g., $\mathbf{x} \in \mathbb{R}^d$ for $d > 10$

More on Entropy Search

- Entropy search and its variants are particularly popular:

$$\mathbf{x}_t \approx \arg\min_{\mathbf{x}_t \in D} \mathbb{E}_{y_t}\left[H(\mathbf{x}^\star \mid \{\mathbf{x}_i, y_i\}_{i=1}^t)\right]$$

H: entropy function

  - Interpretation: Choose the point that makes us least uncertain (i.e., minimizes entropy) about the optimizer $\mathbf{x}^*$

- **Difficulty.** Cannot compute $\mathbb{E}_{y_t}\left[H(\mathbf{x}^\star \mid \{\mathbf{x}_i, y_i\}_{i=1}^t)\right]$ exactly
  - Need to approximate, typically using Monte Carlo methods
  - Particularly difficult for higher dimensions, e.g., $\mathbf{x} \in \mathbb{R}^d$ for $d > 10$

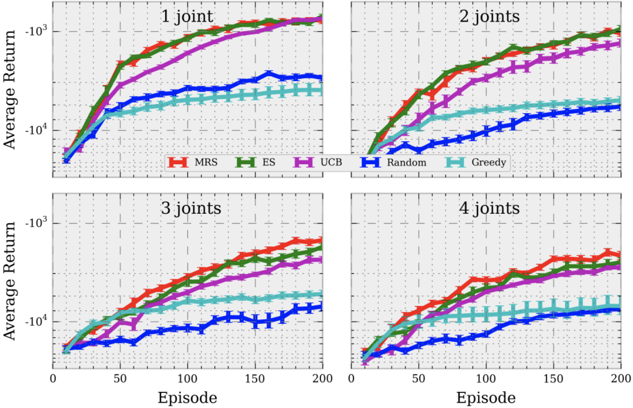
- **Alternative: Max-value entropy search.**

$$\mathbf{x}_t \approx \arg\min_{\mathbf{x}_t \in D} \mathbb{E}_{y_t}\left[H(f^\star \mid \{\mathbf{x}_i, y_i\}_{i=1}^t)\right]$$

  - Intuition: Low uncertainty in $f^* = f(\mathbf{x}^*)$ should mean we have found $\mathbf{x}^*$
  - Now approximating entropy is easier – only one-dimensional

Experimental Example

- Performance plots from [Metzen, 2016] for robot control task:



Accommodating the Twists: Level-Set Estimation

- Limited literature
  - Confidence-bound based LSE algorithm: [Gotovos *et al.*, 2013]

$$\mathbf{x}_t = \arg\max_{\mathbf{x} \in M_{t-1}} \min \Big\{ u_t(\mathbf{x}) - h, h - \ell_t(\mathbf{x}) \Big\}$$

u_t/l_t:	upper/lower confidence bounds
M_t:	the set of unclassified points
h:	the level-set threshold

  - Analogous to, but distinct from, the GP-UCB algorithm for BO
  - **Intuition:** Resolve uncertainty of points whose confidence interval crosses h

Accommodating the Twists: Level-Set Estimation

- Limited literature
  - Confidence-bound based LSE algorithm: [Gotovos *et al.*, 2013]

$$\mathbf{x}_t = \arg\max_{\mathbf{x} \in M_{t-1}} \min \Big\{ u_t(\mathbf{x}) - h, h - \ell_t(\mathbf{x}) \Big\}$$

u_t/l_t:	upper/lower confidence bounds
M_t:	the set of unclassified points
h:	the level-set threshold

    - Analogous to, but distinct from, the GP-UCB algorithm for BO
    - **Intuition:** Resolve uncertainty of points whose confidence interval crosses h

  - Straddle heuristic: [Bryan *et al.*, 2006]

$$\mathbf{x}_t = \arg\max_{\mathbf{x} \in D} 1.96\sigma_{t-1}(\mathbf{x}) - |\mu_{t-1}(\mathbf{x}) - h|$$

Accommodating the Twists: Level-Set Estimation

- Limited literature
  - Confidence-bound based LSE algorithm: [Gotovos *et al.*, 2013]

$$\mathbf{x}_t = \operatorname*{arg\,max}_{\mathbf{x}\in M_{t-1}} \min\Big\{u_t(\mathbf{x}) - h, h - \ell_t(\mathbf{x})\Big\}$$

u_t/l_t: upper/lower confidence bounds
M_t: the set of unclassified points
h: the level-set threshold

    - Analogous to, but distinct from, the GP-UCB algorithm for BO
    - **Intuition:** Resolve uncertainty of points whose confidence interval crosses h

  - Straddle heuristic: [Bryan *et al.*, 2006]

$$\mathbf{x}_t = \operatorname*{arg\,max}_{\mathbf{x}\in D} 1.96\sigma_{t-1}(\mathbf{x}) - |\mu_{t-1}(\mathbf{x}) - h|$$

Advantages: Versatility in the sense of handling level-set estimation

Disadvantages: No theory (Straddle); lacking in other versatility (costs, non-uniform noise, multi-fidelity)

Accommodating the Twists *with Guarantees*: TruVaR

Truncated Variance Reduction (TruVaR) algorithm: [Bogunovic *et al.*, 2016]

- Unified BO and LSE
- Versatility to handle all of the above twists
- Theoretical guarantees

TruVaR Intuition (for optimization):

- Use confidence bounds to keep track of potential maximizers
- Choose points that shrink their uncertainty

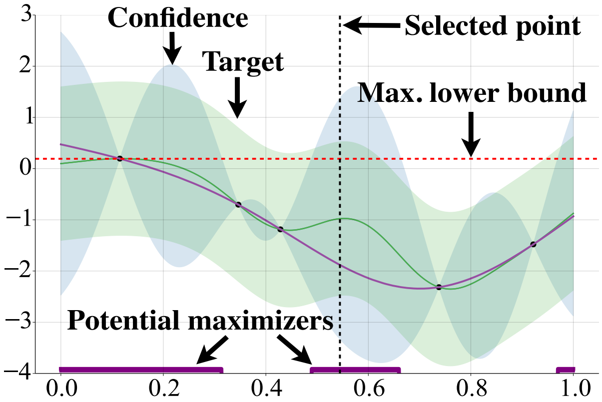
Modified Template for Choosing $\mathbf{x}_t$ Based on $\{(\mathbf{x}_{t'}, y_{t'})\}_{t'<t}$

A general TruVaR template:

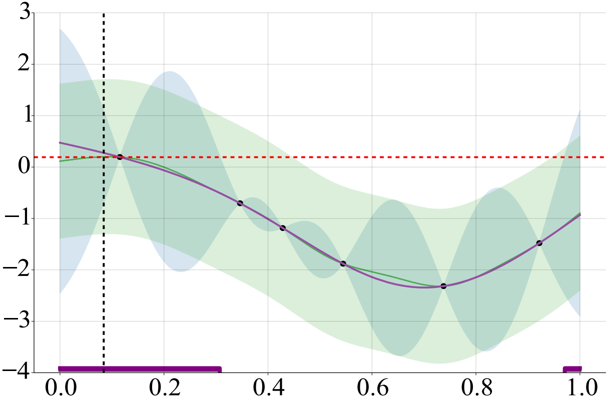
- Choose $\mathbf{x}_t$ to shrink the posterior variance within[1] M_t below a target η
- For each point chosen,
  1. Update M_t via confidence bounds
  2. If the target η is reached within M_t, then set $\eta \leftarrow \frac{\eta}{\text{const.}}$

[1] M_t: potential maximizers (BO) or unclassified points (LSE)

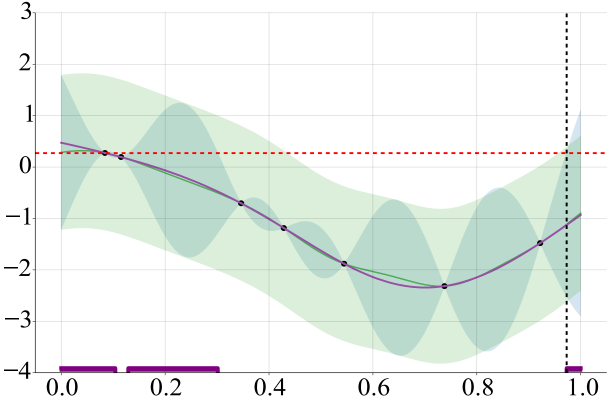
TruVaR: Intution



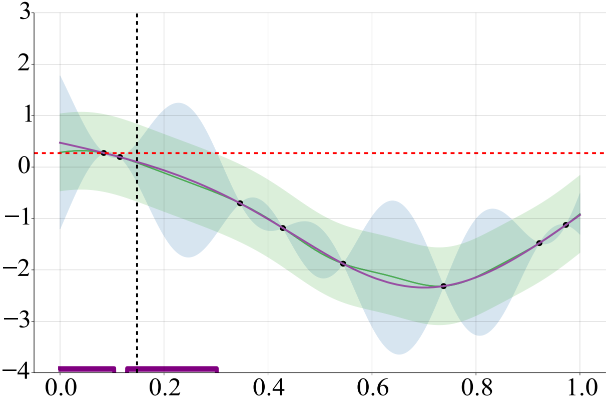
TruVaR: Intution



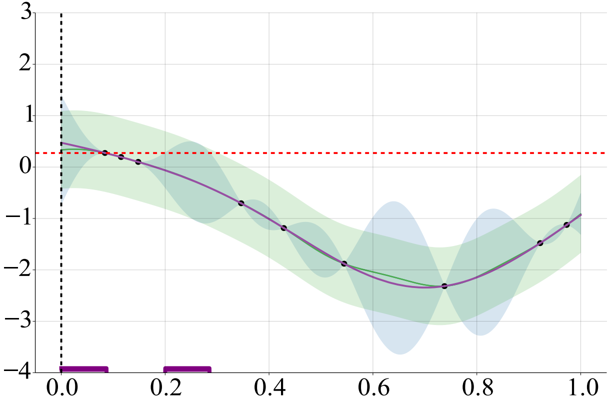
TruVaR: Intution



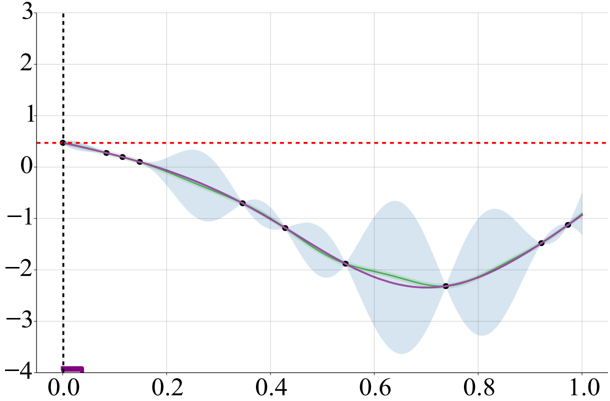
TruVaR: Intution



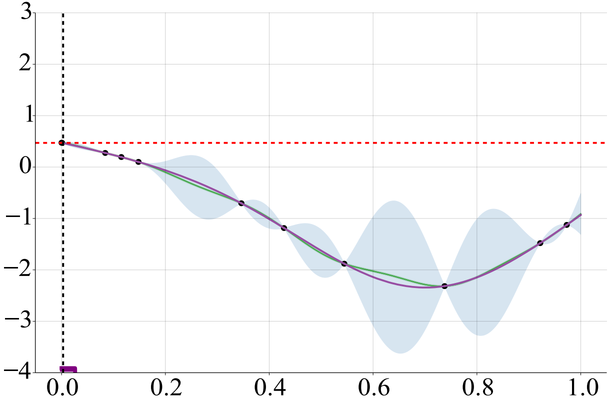
TruVaR: Intution



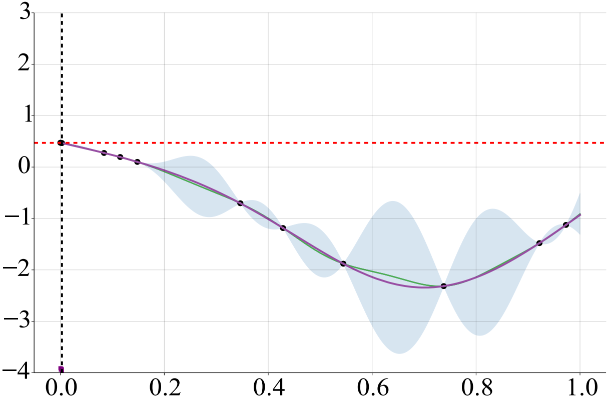
TruVaR: Intution



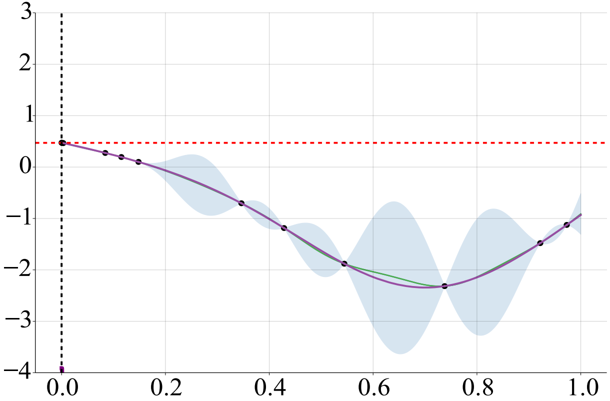
TruVaR: Intution



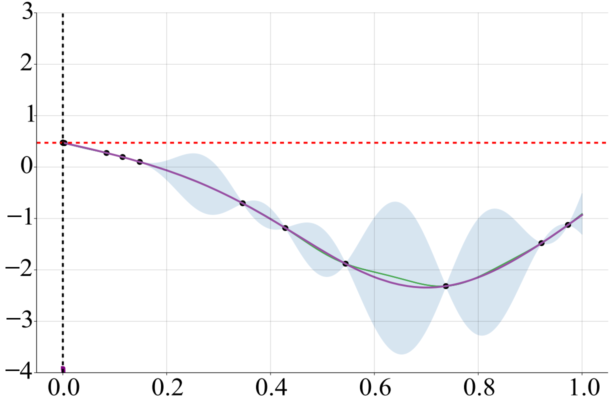
TruVaR: Intution



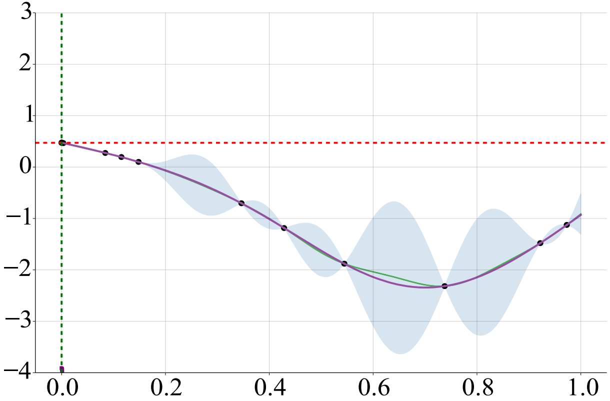
TruVaR: Intution



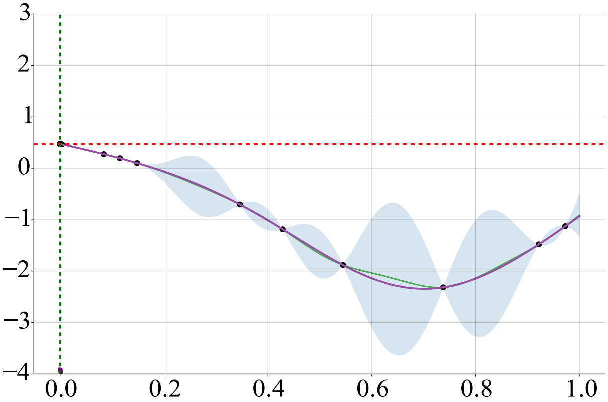
TruVaR: Intution



TruVaR: Intution



TruVaR: Intution



TruVaR: Acquisition Function

- Acquisition function based on variance reduction per cost

$$\arg\max_{x \in D} \frac{\sum_{\overline{\mathbf{x}} \in M_{t-1}} \max\{\beta_{(i)} \sigma^2_{t-1}(\overline{\mathbf{x}}), \eta^2_{(i)}\} - \sum_{\overline{\mathbf{x}} \in M_{t-1}} \max\{\beta_{(i)} \sigma^2_{t-1|\mathbf{x}}(\overline{\mathbf{x}}), \eta^2_{(i)}\}}{c(\mathbf{x})}$$

$\sigma^2_{t-1|\mathbf{x}}$: posterior variance given all points up to time $t-1$, and $\mathbf{x}$

$\beta_{(i)}$: exploration parameter

The set of potential maximizers M_t

- BO

$$M_t = \left\{ \mathbf{x} \in M_{t-1} \; : \; u_t(\mathbf{x}) \geq \max_{\overline{\mathbf{x}} \in M_{t-1}} \ell_t(\overline{\mathbf{x}}) \right\}$$

$u_t(\mathbf{x})/l_t(\mathbf{x})$: upper/lower confidence bounds

- LSE

$$M_t = \left\{ \mathbf{x} \in M_{t-1} \; : \; u_t(\mathbf{x}) \geq h \text{ and } \ell_t(\mathbf{x}) \leq h \right\}$$

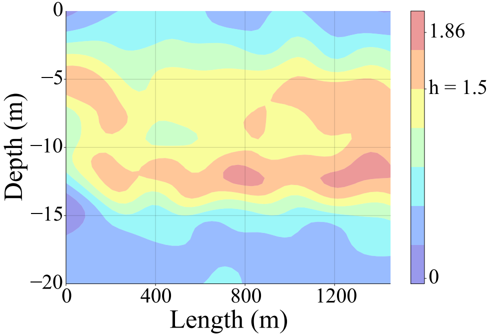
Numerical Evidence

- Real and synthetic data
- Acronyms

LSE	Level-set estimation algorithm	[Gotovos *et al.*, 2013]
STR	Straddle heuristic	[Bryan *et al.*, 2006]
VAR	Maximum variance rule	[Gotovos *et al.*, 2013]
EI	Expected improvement	[Mockus *et al.*, 1978]
GP-UCB	Gaussian process upper confidence bound	[Srinivas *et al.*, 2012]

Numerical Evidence 1: Level-Set Estimation (I)

- Lake Zürich chlorophyll concentration via an autonomous vehicle:



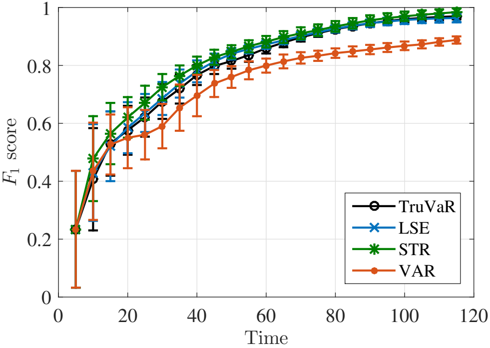
- Evaluate performance with the F_1 score:

$$F_1 = \frac{\#\text{true positives}}{\#\text{true positives} + \frac{1}{2}\Big(\#\text{false positives} + \#\text{false negatives}\Big)} \in \big[0, 1\big]$$

where "positive" means above the level-set h.

Numerical Evidence 1: Level-Set Estimation (II)

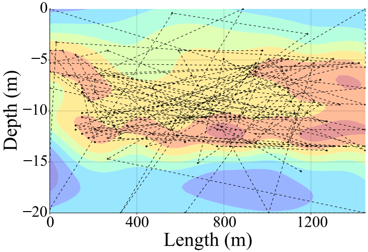
- Classification performance (unit-cost):



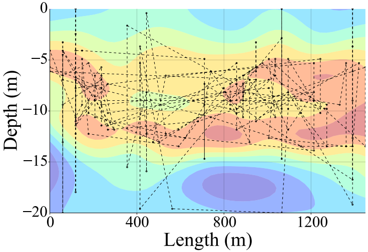
Numerical Evidence 1: Level-Set Estimation (III)

- Cost function: (i) Penalizes distance traveled; (ii) Penalizes deeper measurements

LSE algorithm [Gotovos *et al.*, 2013]:

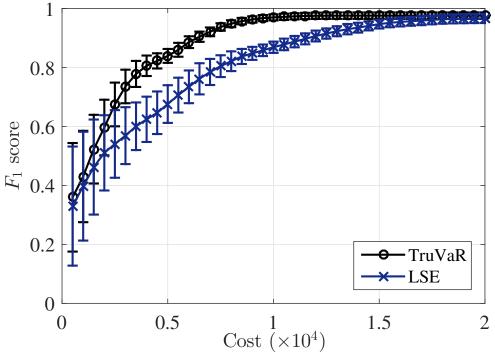


TruVaR:



Numerical Evidence 1: Level-Set Estimation (IV)

- Classification performance (non-unit cost):



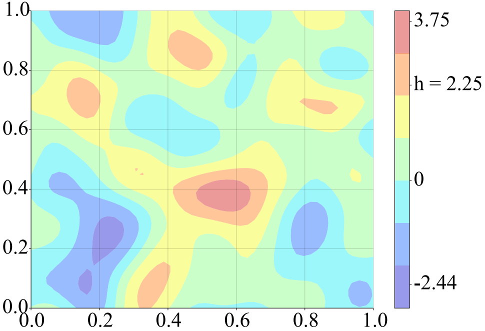
Numerical Evidence 2: Level-Set Estimation (I)

- Twist: Choice of the noise level
  - Noise levels $\{10^{-6}, 10^{-3}, 0.05\}$
  - Corresponding costs $\{15, 10, 2\}$

- Synthetic simulation
  - Function drawn from GP with squared-exponential kernel
  - True kernel used in algorithms

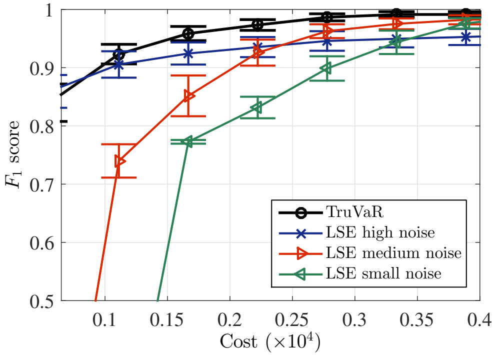
Numerical Evidence 2: Level-Set Estimation (II)

- Synthetic function drawn from GP:



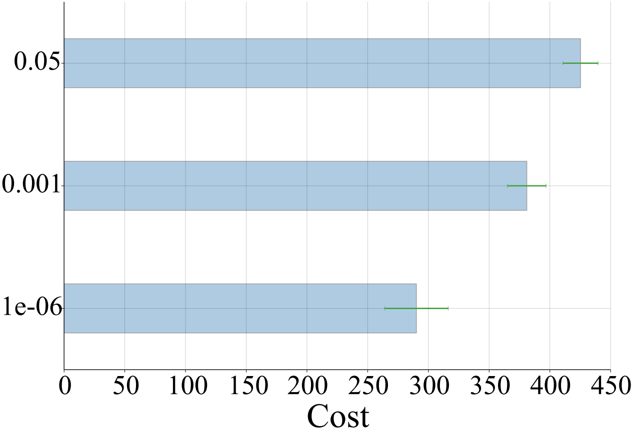
Numerical Evidence 2: Level-Set Estimation (III)

- ***Oracle-level*** classification performance:



Numerical Evidence 2: Level-Set Estimation (IV)

- Cost incurred for each noise level:

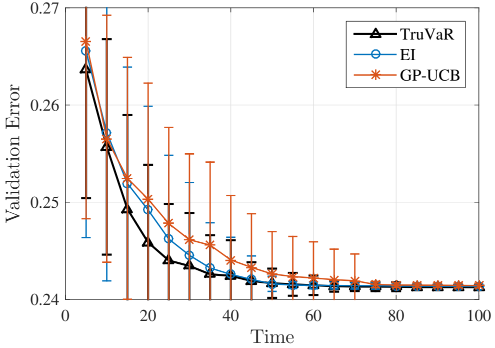


- TruVaR gradually switches between different levels:

high noise / low cost $\Longrightarrow$ medium noise / cost $\Longrightarrow$ low noise / high cost

Numerical Evidence 3: Bayesian Optimization

- Hyper-parameter tuning: SVM on grid dataset [Snoek *et al.*, 2012]
  - Tuning 3 hyperparameters for SVM algorithm
  - GP kernel estimated online using maximum-likelihood
- Generalization error:



TruVaR – Batch Setting

- In the batch setting, we choose $B > 1$ points at each time, evaluate them in parallel, and observe the B observations [Azimi *et al.*, 2010]
  - Example 1: Equipment allows running scientific experiments in parallel
  - Example 2: f is a computation, and we have multiple computing cores

TruVaR – Batch Setting

- In the batch setting, we choose $B > 1$ points at each time, evaluate them in parallel, and observe the B observations [Azimi *et al.*, 2010]
  - Example 1: Equipment allows running scientific experiments in parallel
  - Example 2: f is a computation, and we have multiple computing cores

- With $B = 1$, we can interpret TruVaR as greedily minimizing

$$\sum_{\overline{\mathbf{x}} \in M_{t-1}} \max\{\beta_{(i)} \sigma^2_{t-1|\mathbf{x}}(\overline{\mathbf{x}}), \eta^2_{(i)}\}$$

with respect to $\mathbf{x}$

- **A simple batch extension:** In each round, run B steps of the greedy algorithm minimizing this function

Epilogue: Theoretical Performance

Definition: Numerical ϵ-accuracy

- (BO) The reported point after T rounds satisfies $f(\hat{\mathbf{x}}_T) \geq f(\mathbf{x}^\star) - \epsilon$
- (LSE) The classification after T rounds is correct for points at least $\frac{\epsilon}{2}$-far from h

Epilogue: Theoretical Performance (I)

- Generalize the canonical notion of rounds T to costs C to shrink variance:

$$C^*(\xi, M) = \min_S \Big\{ c(S) \,:\, \max_{\overline{x} \in M} \sigma_S(\overline{x}) \le \xi \Big\},$$

σ_S^2: posterior variance given points in S

Theorem

For a finite domain D, under a submodularity assumption, if TruVaR *is run until the cumulative cost reaches*

$$C_\epsilon = \sum_{i \,:\, 4\eta_{(i-1)} > \epsilon} C^*\left(\frac{\eta_{(i)}}{\beta_{(i)}^{1/2}}, \overline{M}_{(i-1)} \right) \log \frac{|\overline{M}_{(i-1)}| \beta_{(i)}}{\eta_{(i)}^2},$$

for suitable $\beta_{(i)}$, then with probability at least $1 - \delta$ we have ϵ-accuracy.
In the cumulative cost, the outer bounds on M_t are defined as

$$\overline{M}_{(i)} := \Big\{ \mathbf{x} \,:\, f(\mathbf{x}) \ge f(\mathbf{x}^\star) - 4\eta_{(i)} \Big\} \qquad \text{(BO)}$$

$$\overline{M}_{(i)} := \Big\{ \mathbf{x} \,:\, |f(\mathbf{x}) - h| \le 2\eta_{(i)} \Big\} \qquad \text{(LSE)}$$

Epilogue: Theoretical Performance (II)

Corollary

There exist $\beta_{(i)}$ such that we have ϵ-accuracy with probability at least $1-\delta$ once

$$T \geq O^*\Big(\frac{\sigma^2\gamma_T}{\epsilon^2} + \frac{C_1\gamma_T}{\sigma^2}\Big)$$

where $C_1 = \frac{1}{\log(1+\sigma^{-2})}$, and

$$\gamma_T = \max_{S\,:\,|S|=T} I(f;\mathbf{y}_S)$$

is the maximum amount of information $\mathbf{y}_S = (y_1,\dots,y_T)$ can reveal about f upon querying points $S = (\mathbf{x}_1,\dots,\mathbf{x}_T)$

- New: Improved dependence on the noise level in BO
  - For small σ and $\epsilon \ll \sigma$, existing bound (GP-UCB): $T \geq O^*\big(\frac{C_1\gamma_T}{\epsilon^2}\big)$

Epilogue: Theoretical Performance (III)

- Multi-noise setup:
  - noise levels $\sigma^2(1), \dots, \sigma^2(K)$
  - sampling costs $c(1), \dots, c(k)$

Corollary

For each $k = 1, \cdots, K$, let $T^*(k)$ denote the smallest value of T such that

$$T \geq \Omega^*\Big(\frac{\sigma(k)^2 \gamma_T}{\epsilon^2} + \frac{C_1(k)\gamma_T}{\sigma(k)^2}\Big)$$

where $C_1 = \frac{1}{\log(1+\sigma(k)^{-2})}$.

There exist choices of $\beta_{(i)}$ such that we have ϵ-accuracy with probability at least $1-\delta$ once the cumulative cost reaches

$$\min_k c(k) T^*(k)$$

- As good as sticking to any fixed noise/cost pair a posteriori!

Epilogue: Theoretical Performance (IV)

- Recall: Minimum cost required to shrink variance

$$C^*(\xi, M) = \min_S \Big\{ c(S) \,:\, \max_{\overline{\mathbf{x}} \in M} \sigma_S(\overline{\mathbf{x}}) \le \xi \Big\},$$

σ_S^2: posterior variance given points in S

- In a single epoch, TruVaR greedily maximizes a submodular set function

$$g(S) = -\sum_{\overline{\mathbf{x}} \in M_{t-1}} \max\{\beta_{(i)} \sigma^2_{t-1|S}(\overline{\mathbf{x}}), \eta^2_{(i)}\}$$

- By submodularity, our incurred cost is within a logarithmic factor of the optimum:

$$C_{(i)} \le C^* \bigg(\frac{\eta_{(i)}}{\beta_{(i)}^{1/2}}, \overline{M}_{(i-1)} \bigg) \log \frac{|\overline{M}_{(i-1)}| \beta_{(i)}}{\eta^2_{(i)}}$$

- Sum over the epochs i to obtain the theory

Further Reading

- Tutorials/classes:
  - Taking the Human Out of the Loop: A Review of Bayesian Optimization (Shahriari *et al.*, 2016)
  - A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning (Brochu *et al.*, 2010)
  - Lectures on Gaussian Processes & Bayesian optimization by Nando de Freitas (available on YouTube)
- Other:
  - Various papers referenced at the end of each set of slides (this & previous ones)
  - Popular GP book: Gaussian Processes for Machine Learning (Rasmussen, 2006)
  - My papers: `http://www.comp.nus.edu.sg/~scarlett/`

Useful Programming Packages

- **Useful libraries:**
  - Python packages (some with other methods beyonds GPs):
    - GPy and GPyOpt
    - Spearmint
    - BayesianOptimization
    - PyBo
    - HyperOpt
    - MOE
  - Packages for other languages:
    - GPML for MATLAB
    - GPFit and rBayesianOptimization for R

References I

[1] Ilija Bogunovic, Jonathan Scarlett, Andreas Krause, and Volkan Cevher.
Truncated variance reduction: A unified approach to Bayesian optimization and level-set estimation.
In *Conf. Neur. Inf. Proc. Sys. (NIPS)*, 2016.

[2] Eric Brochu, Vlad M. Cora, and Nando de Freitas.
A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning.
http://arxiv.org/abs/1012.2599, 2010.

[3] Brent Bryan, Robert C Nichol, Christopher R Genovese, Jeff Schneider, Christopher J Miller, and Larry Wasserman.
Active learning for identifying function threshold boundaries.
In *Conf. Neur. Inf. Proc. Sys. (NIPS)*, pages 163–170, 2006.

[4] Paul W Goldberg, Christopher KI Williams, and Christopher M Bishop.
Regression with input-dependent noise: A Gaussian process treatment.
Adv. Neur. Inf. Proc. Sys. (NIPS), 10:493–499, 1997.

[5] Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause.
Active learning for level set estimation.
In *Int. Joint. Conf. Art. Intel.*, 2013.

References II

[6] Philipp Hennig and Christian J Schuler.
Entropy search for information-efficient global optimization.
J. Mach. Learn. Research, 13(1):1809–1837, 2012.

[7] Jan Hendrik Metzen.
Minimum regret search for single-and multi-task optimization.
In *Int. Conf. Mach. Learn. (ICML)*, 2016.

[8] J Moćkus, V Tiesis, and A Źilinskas.
The application of Bayesian methods for seeking the extremum. vol. 2, 1978.

[9] Carl Edward Rasmussen.
Gaussian processes for machine learning.
MIT Press, 2006.

[10] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando de Freitas.
Taking the human out of the loop: A review of Bayesian optimization.
Proc. IEEE, 104(1):148–175, 2016.

[11] Jasper Snoek, Hugo Larochelle, and Ryan P Adams.
Practical Bayesian optimization of machine learning algorithms.
In *Adv. Neur. Inf. Proc. Sys.* 2012.

References III

[12] N. Srinivas, A. Krause, S.M. Kakade, and M. Seeger.
Information-theoretic regret bounds for Gaussian process optimization in the bandit setting.
IEEE Trans. Inf. Theory, 58(5):3250–3265, May 2012.

[13] Kevin Swersky, Jasper Snoek, and Ryan P Adams.
Multi-task Bayesian optimization.
In *Adv. Neur. Inf. Proc. Sys. (NIPS)*, pages 2004–2012, 2013.