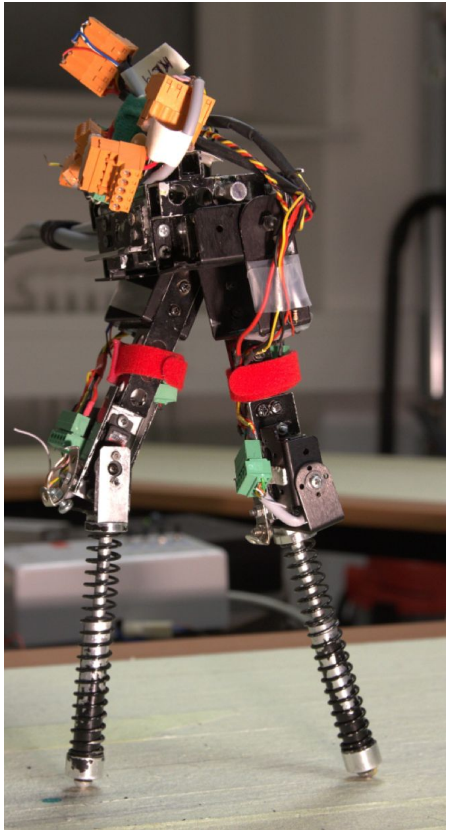
A tutorial on Bayesian optimization

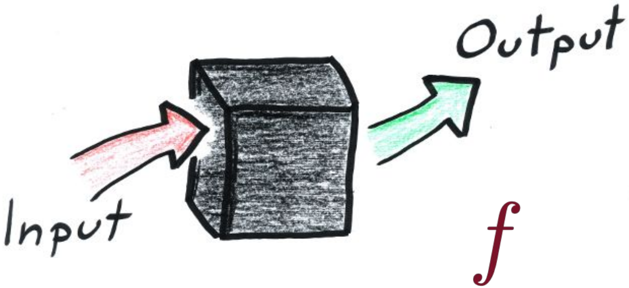
Zi Wang @ Google Brain

Google Research

Blackbox Function Optimization



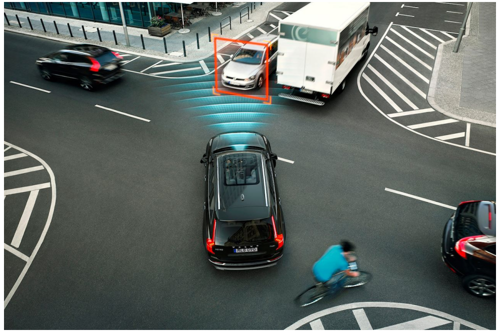
[Calandra et al., 2015]

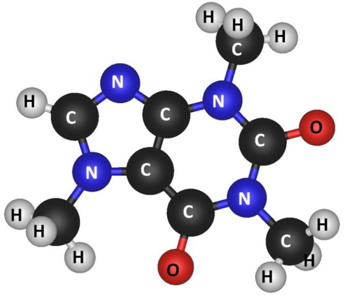


Goal:

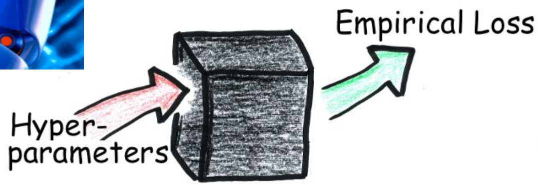
$$x_* = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}} f(x)$$

Blackbox Function Optimization



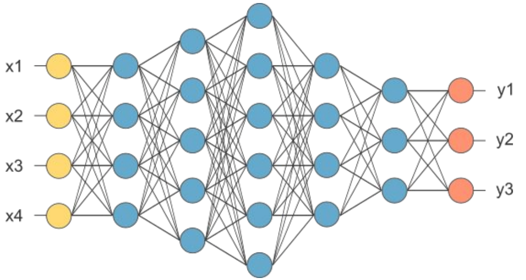


Goal: $x_* = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}} f(x)$



Challenges:

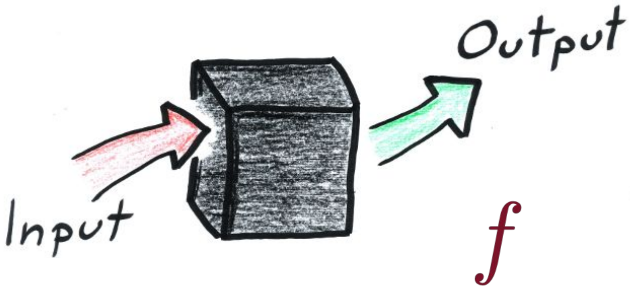
- f is expensive to evaluate
- f is multi-peak
- no gradient information
- evaluations can be noisy



[Snoek et al., 2012; Gonzalez et al., 2015; Hernández-Lobato et al., 2017]

[Koch, 2016]

Blackbox Function Optimization

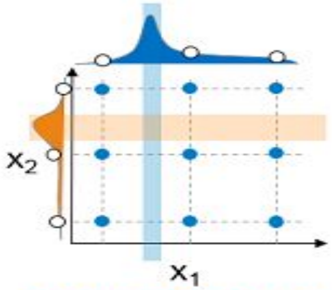


Goal: $x_* = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}} f(x)$

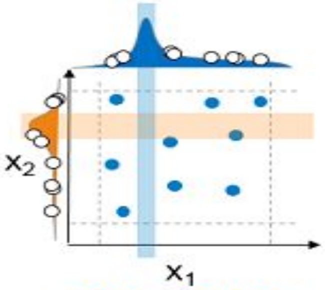
Challenges:

- f is expensive to evaluate
- f is multi-peak
- no gradient information
- evaluations can be noisy

Grid search?



Random search?



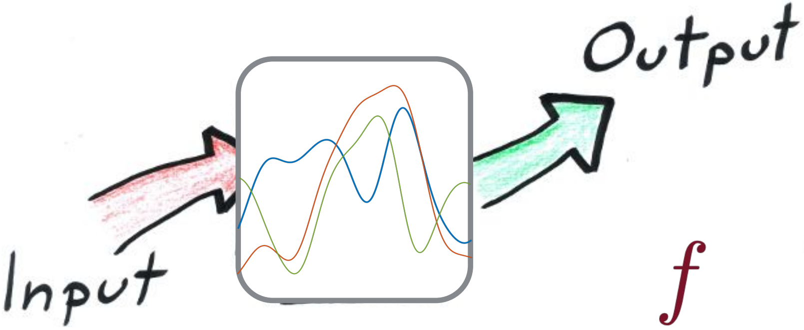
Many evaluations are wasted!

Bayesian Optimization

Idea: build a **probabilistic model** of the function f

LOOP

- choose new query point(s) to evaluate
  decision criterion: acquisition function $\alpha_t(\cdot)$
- update model



$$x_* = \underset{\mathcal{X}\subset\mathbb{R}^d}{\operatorname{argmax}} \; f(x)$$

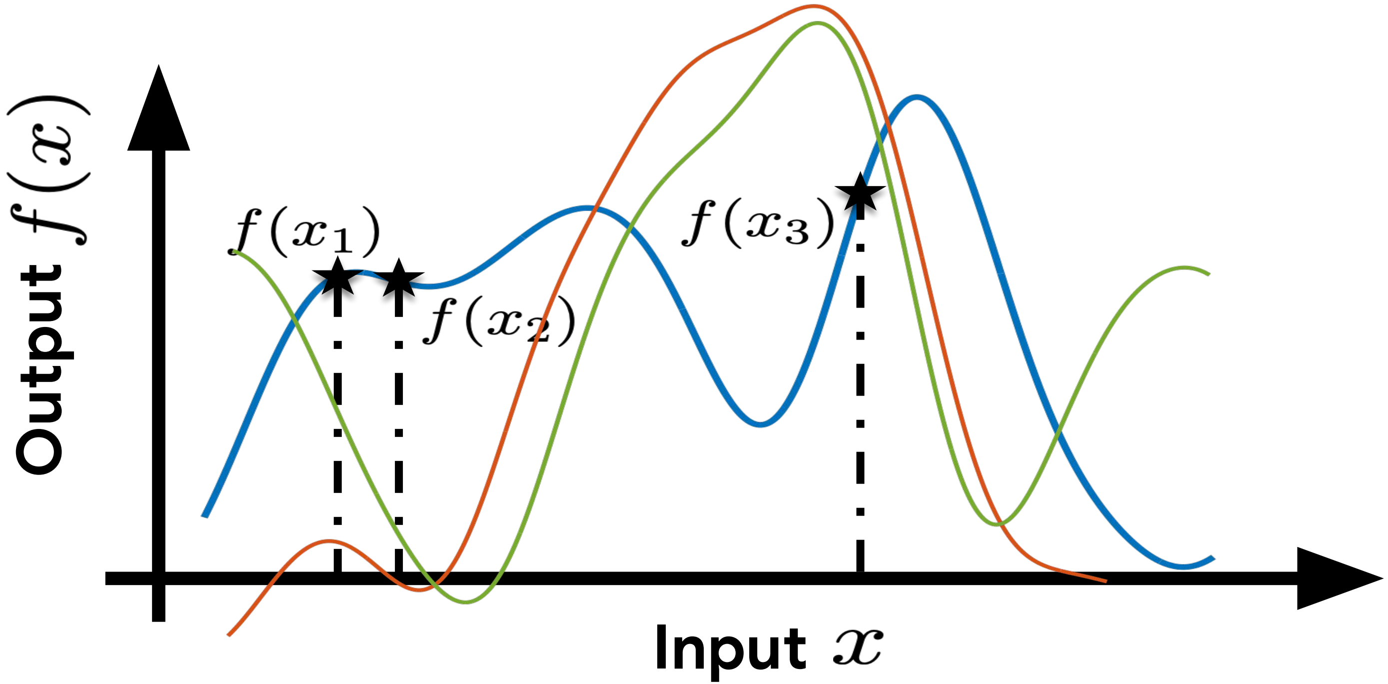
$$\Downarrow$$

$$x_t = \underset{\mathcal{X}\subset\mathbb{R}^d}{\operatorname{argmax}} \; \alpha_t(x)$$

$$t = 1, \cdots, T$$

Gaussian Processes (GPs)

- probability distribution over functions
- any finite set of function values is a multi-variate Gaussian



Gaussian Processes (GPs)

- probability distribution over functions

- any finite set of function values is a multi-variate Gaussian
- kernel function $k(\cdot,\cdot)$; mean function $\mu(\cdot)$

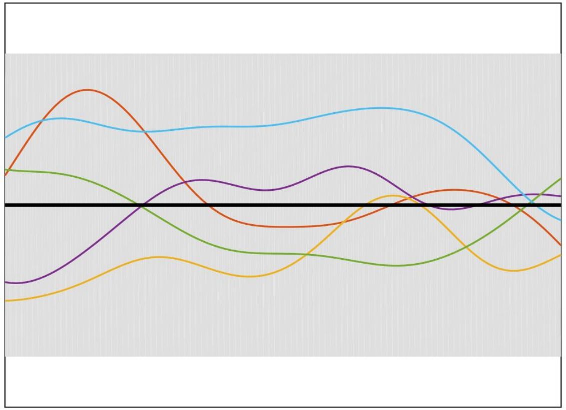
$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu(x_1) \\ \vdots \\ \mu(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1,x_1), & \cdots, & k(x_1,x_n) \\ \vdots, & & \vdots \\ k(x_n,x_1), & \cdots, & k(x_n,x_n) \end{bmatrix} \right)$$

- function $f \sim GP(\mu, k)$; observe noisy output at x_τ

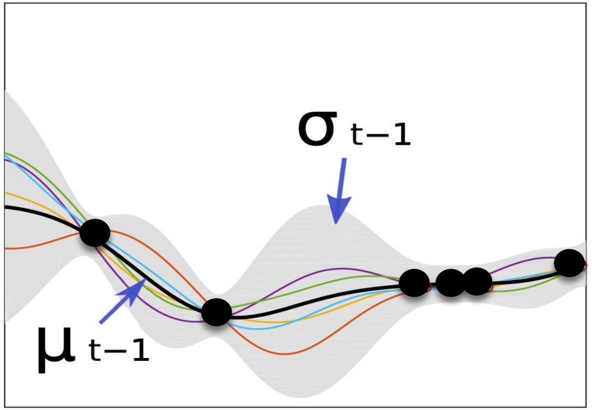
$$y_\tau = f(x_\tau) + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Gaussian Processes (GPs)

Samples from the prior



Samples from the posterior



Given observations $D_t = \{(x_\tau, y_\tau)\}_{\tau=1}^{t-1}$, predict posterior mean and variance in closed form via conditional Gaussian

$$\mu_{t-1}(x) = k_{t-1}(x)^{\mathrm{T}}(K_{t-1} + \sigma^2 I)^{-1} y_{t-1}$$

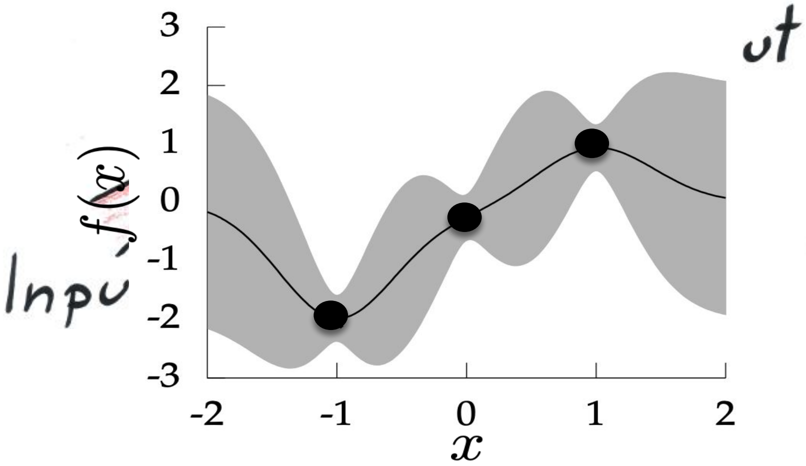
$$\sigma_{t-1}(x)^2 = k(x,x) - k_{t-1}(x)^{\mathrm{T}}(K_{t-1} + \sigma^2 I)^{-1} k_{t-1}(x)$$

Bayesian Optimization

Idea: build a probabilistic model of the function f

LOOP

- choose new query point(s) to evaluate
  decision criterion: acquisition function $\alpha_t(\cdot)$
- update model



$$x_* = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}} f(x)$$

$$x_t = \underset{\mathcal{X} \subset \mathbb{R}^d}{\operatorname{argmax}} \alpha_t(x)$$

$$t = 1, \cdots, T$$

How to design acquisition functions?

Acquisition functions in BayesOpt

Google Research

Examples of acquisition functions in BayesOpt

Prior: $f \sim GP(\mu, k)$

At iteration t,

- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max_x \alpha_t(x)$$

Upper confidence bounds, expected improvement, probability of improvement, entropy search methods…

[Auer, 2002; Srinivas et al., 2010; Kushner, 1964; Mockus, 1974; Hennig & Schuler, 2012; Hernandez-Lobato et al., 2014; Wang&Jegelka, 2017; Hoffman&Zoubin, 2015…]

GP-UCB: an example of acquisition functions

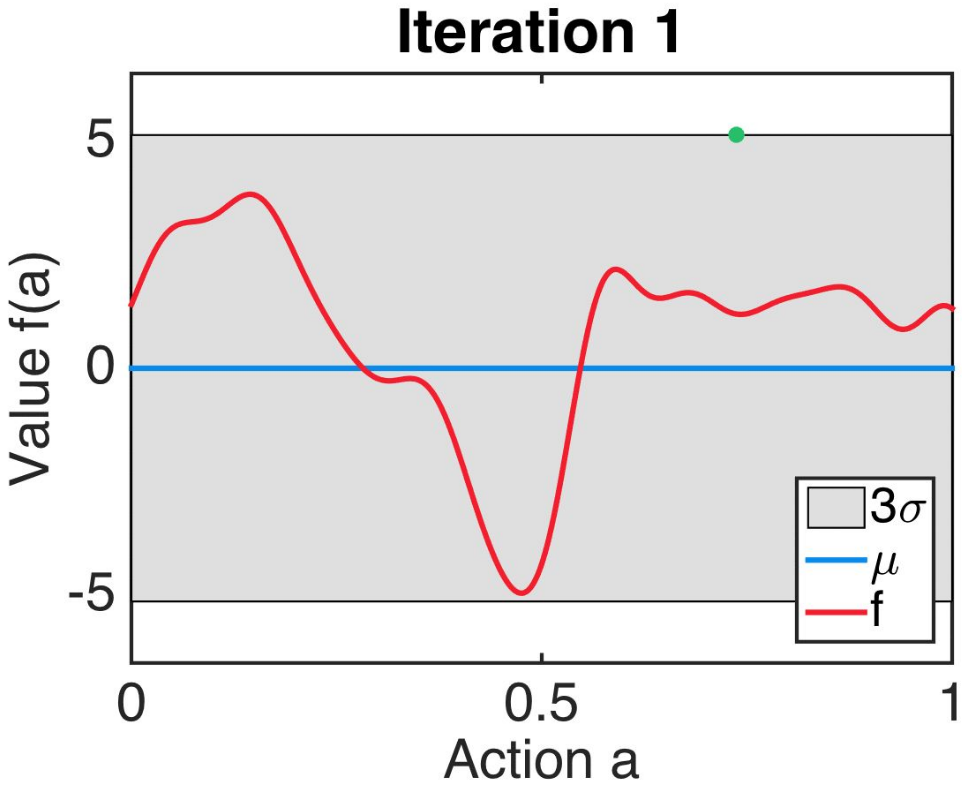
[Auer, 2002; Srinivas et al., 2010]

Prior: $f \sim GP(\mu, k)$

At iteration t,

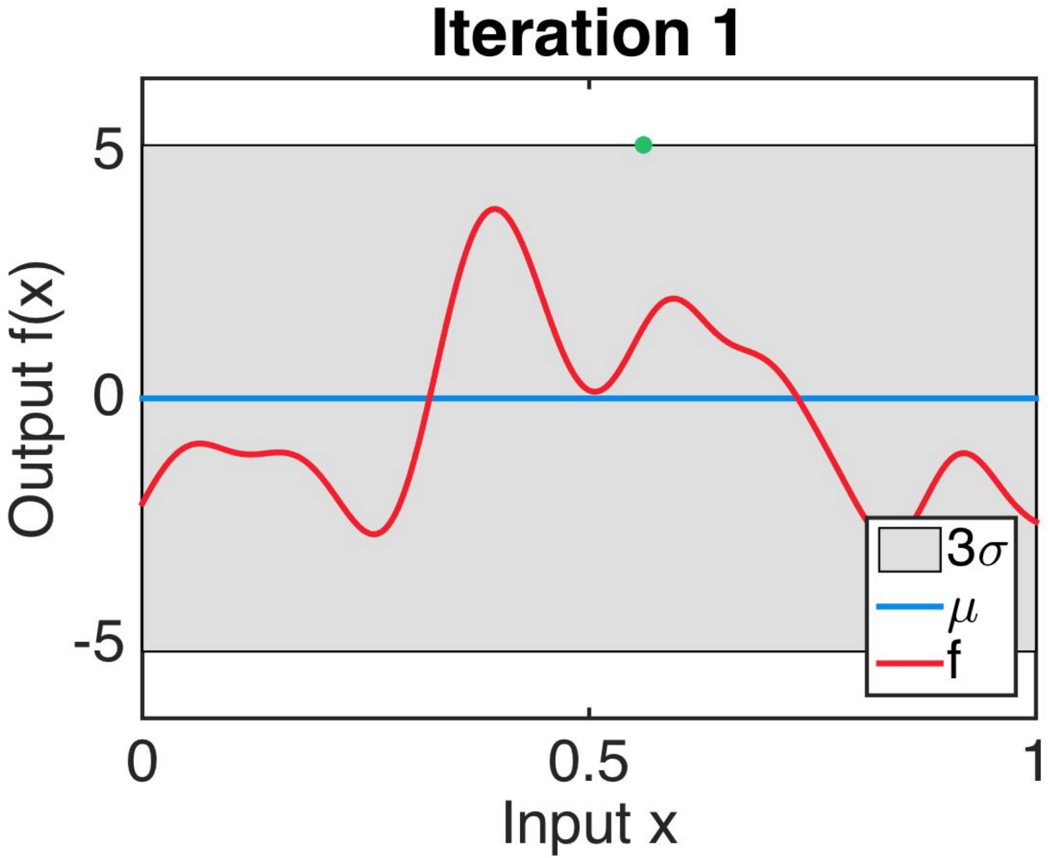
- predict the posterior $\mu_{t-1}(x)$ and $\sigma^2_{t-1}(x)$
- pick an input by optimizing the acquisition function

$$x_t = \arg\max \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$

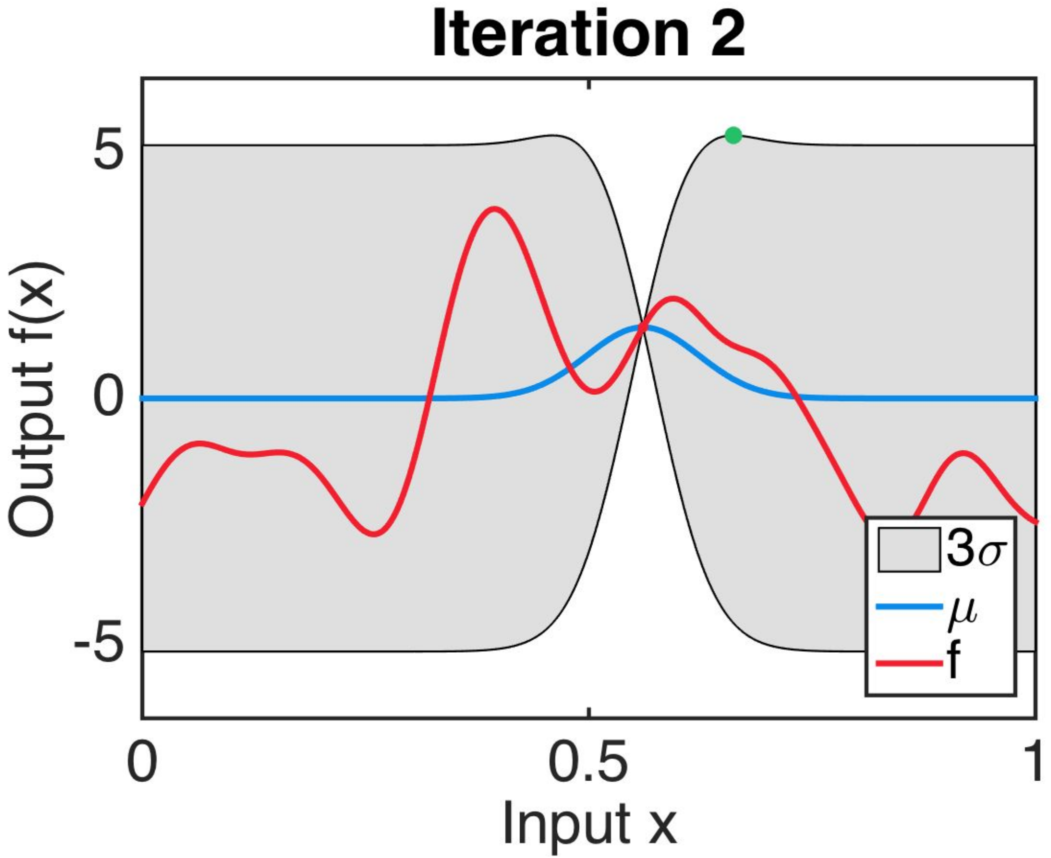


Iteration 1
5
0
-5
Value f(a)
0
0.5
1
Action a
3σ
μ
f

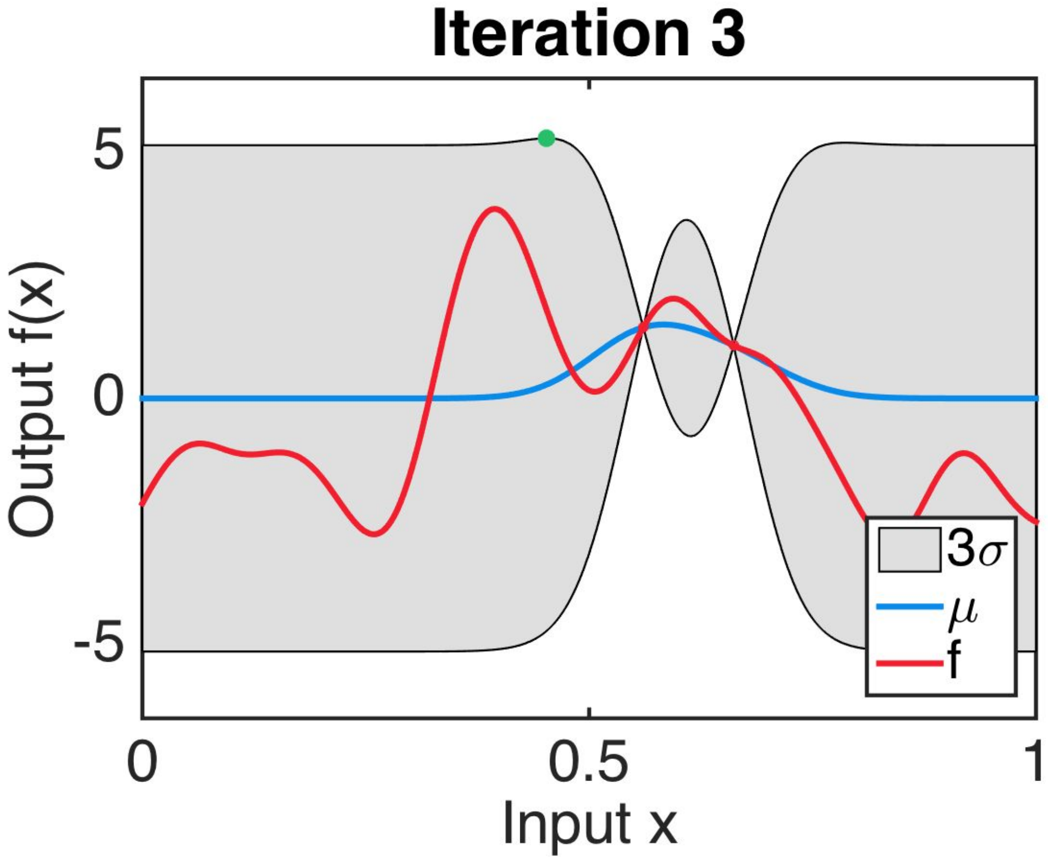
GP-UCB: an example of acquisition functions



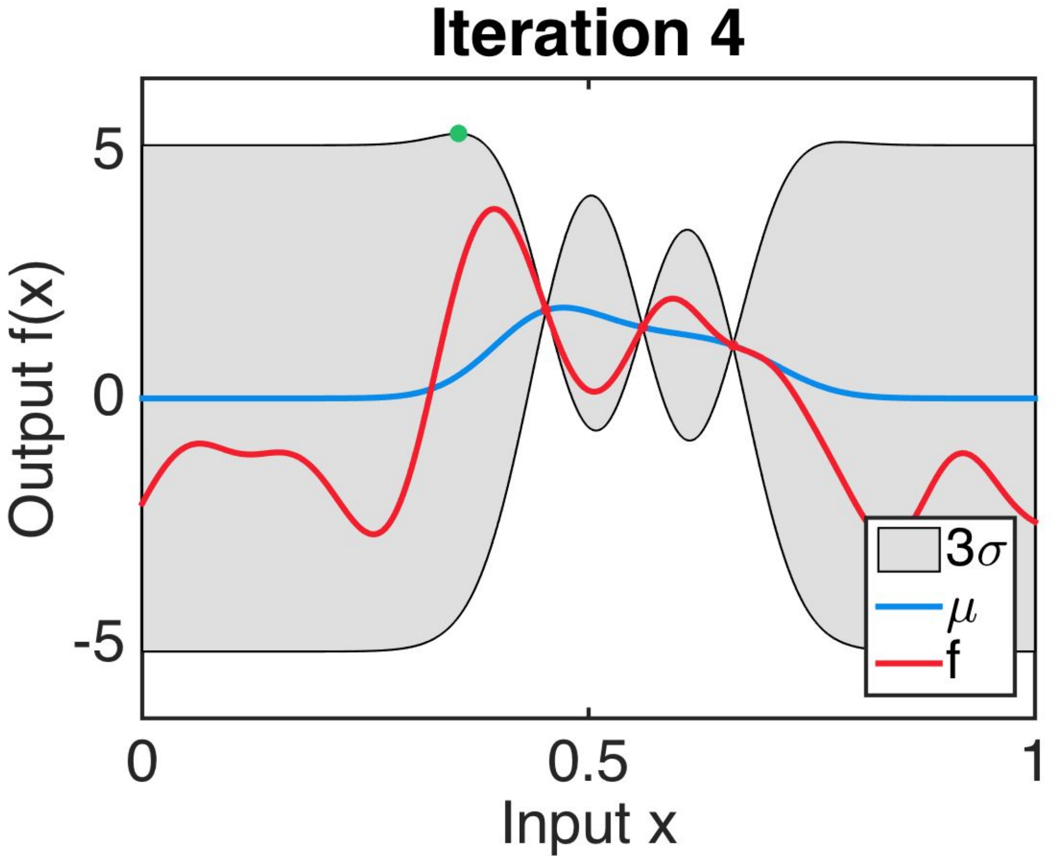
GP-UCB: an example of acquisition functions



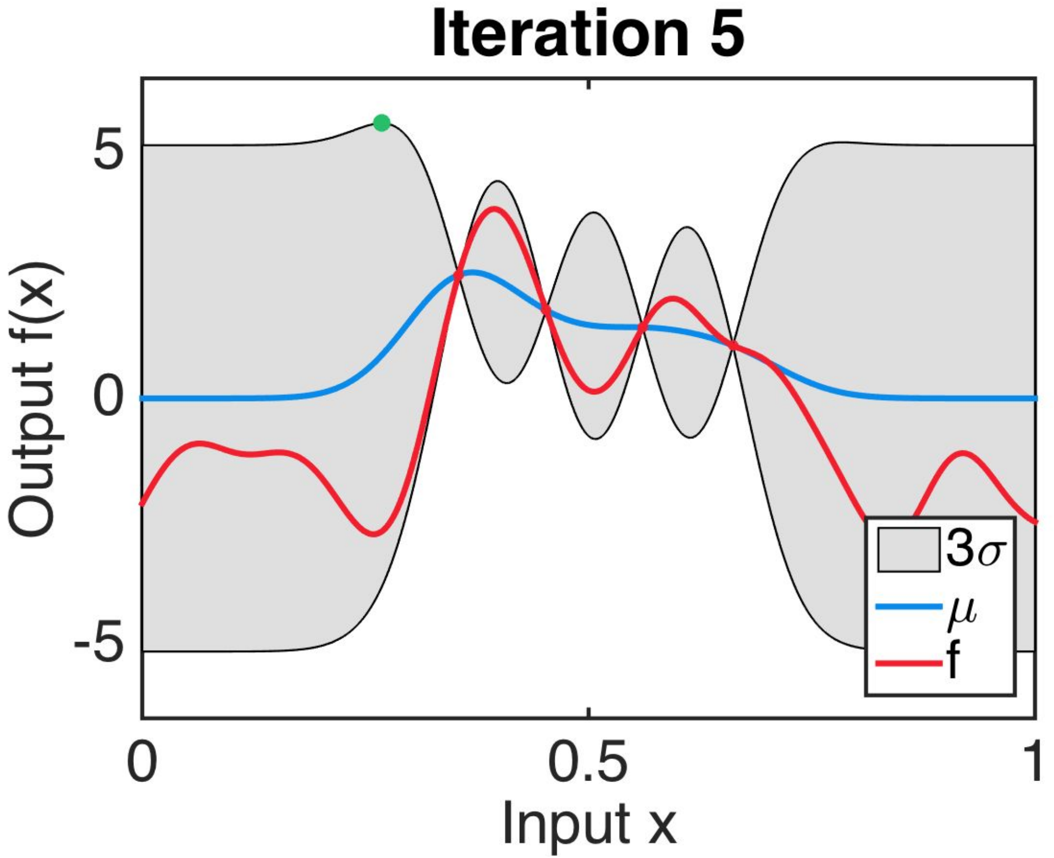
GP-UCB: an example of acquisition functions



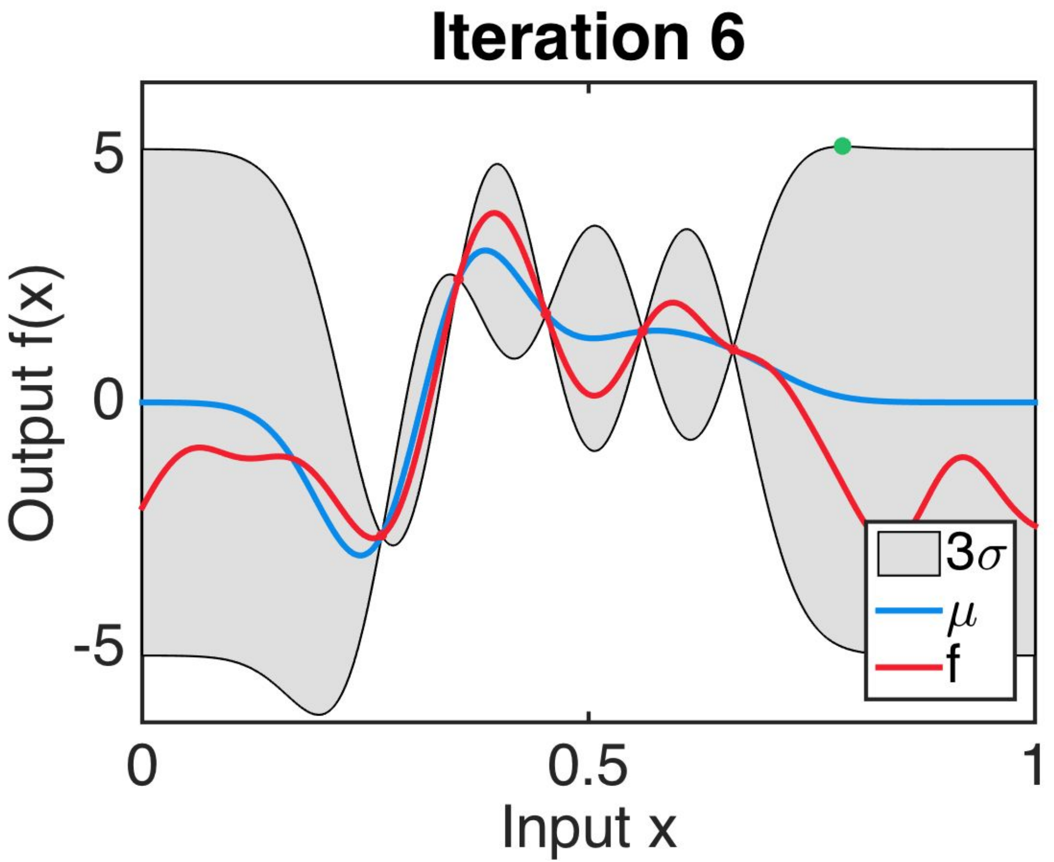
GP-UCB: an example of acquisition functions



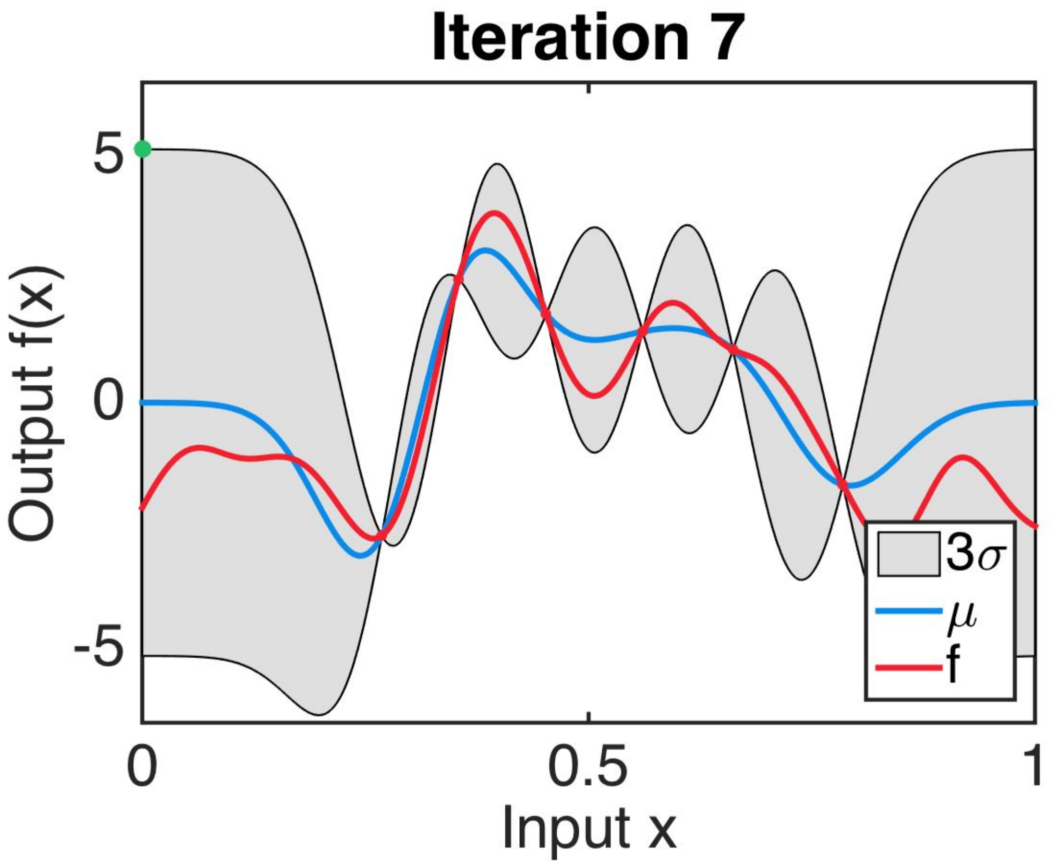
GP-UCB: an example of acquisition functions



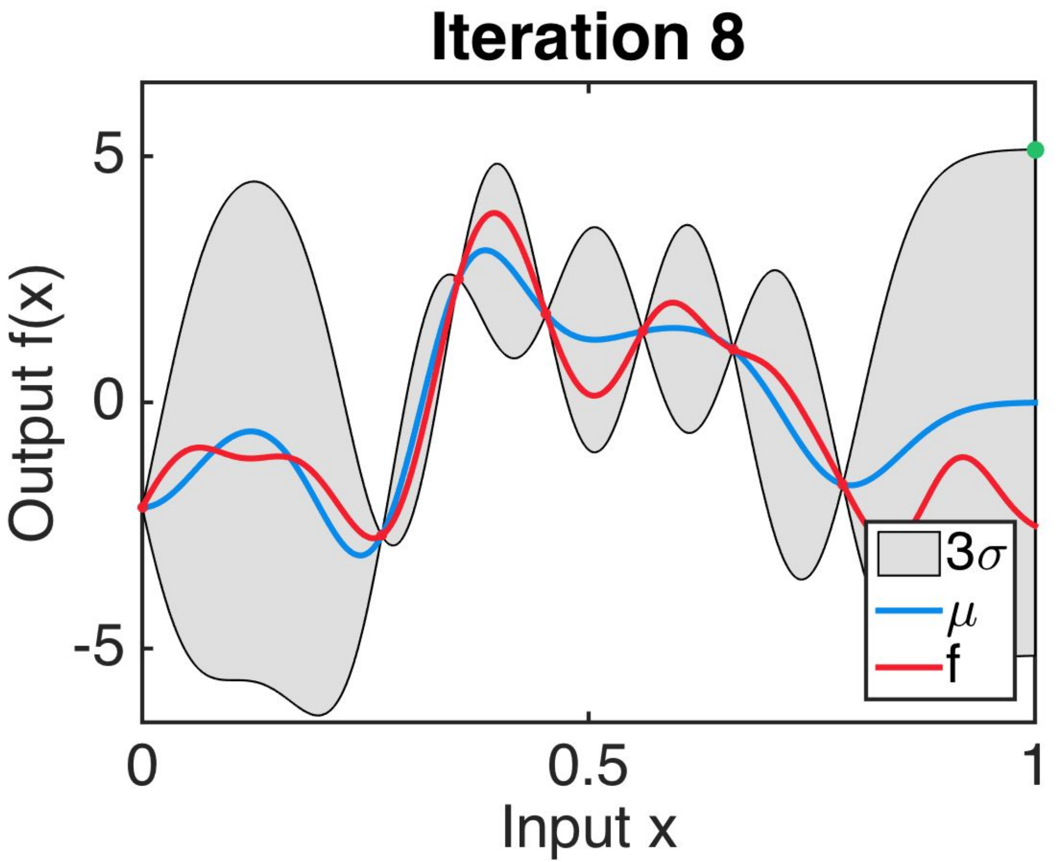
GP-UCB: an example of acquisition functions



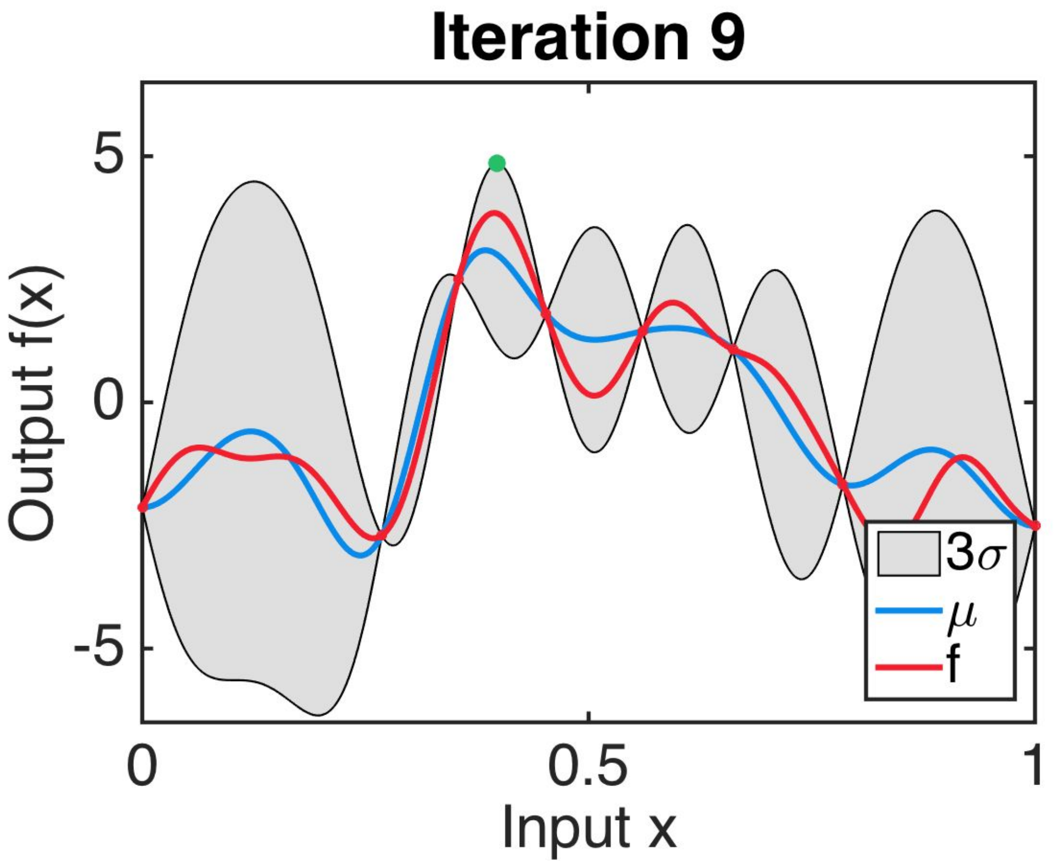
GP-UCB: an example of acquisition functions



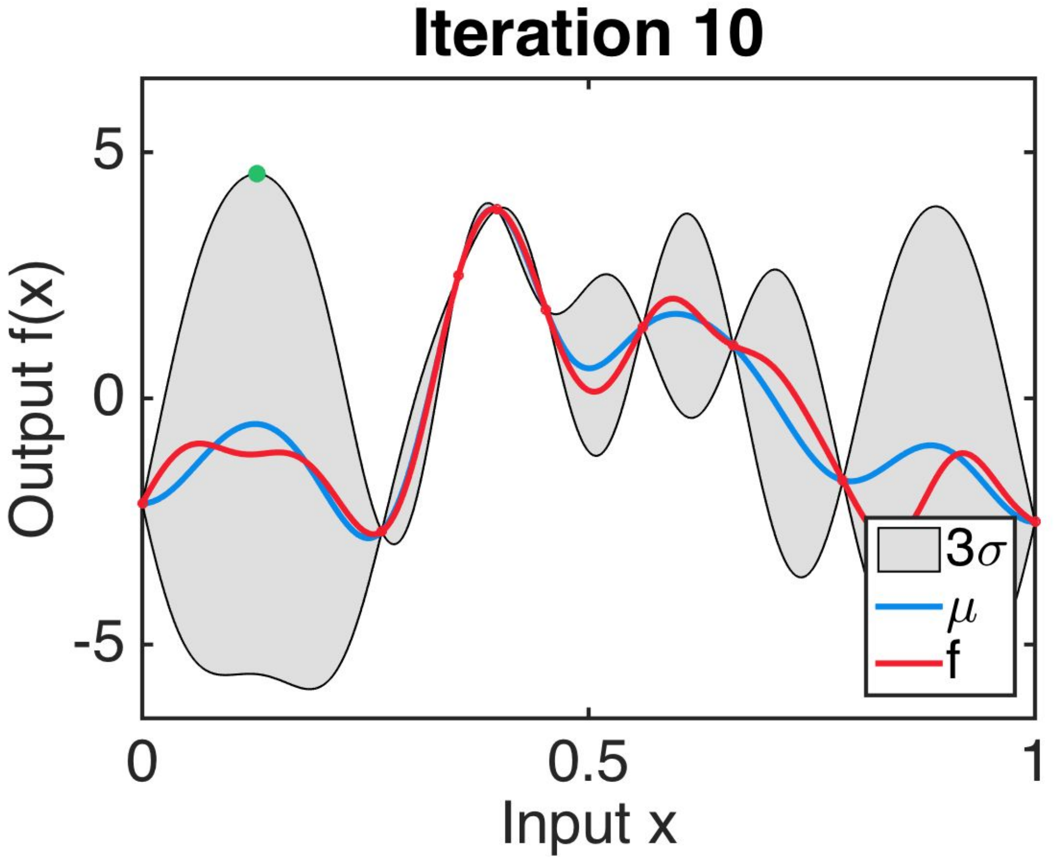
GP-UCB: an example of acquisition functions



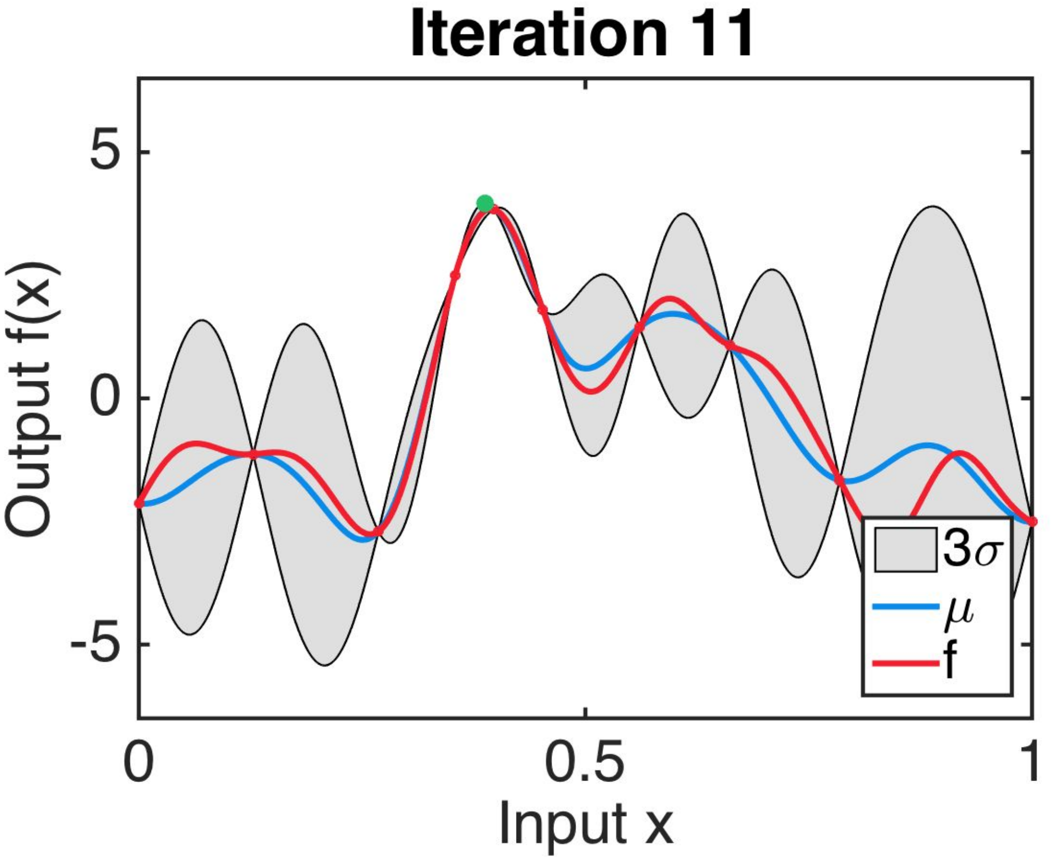
GP-UCB: an example of acquisition functions



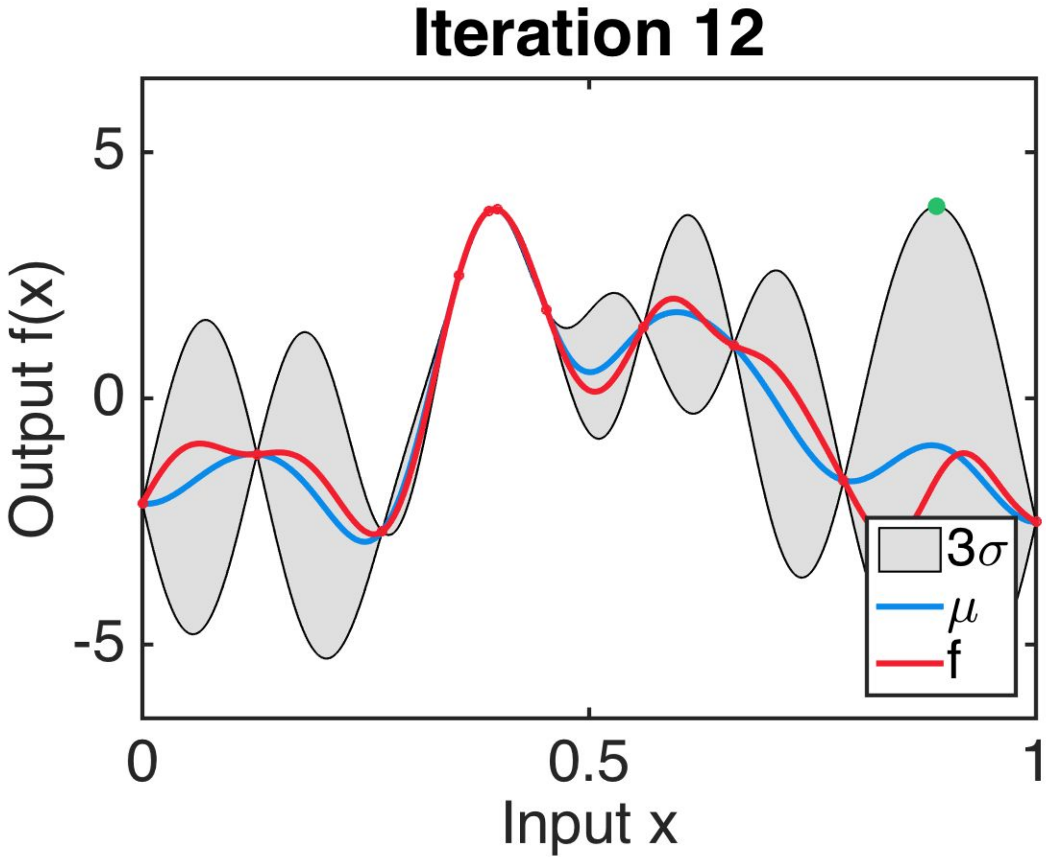
GP-UCB: an example of acquisition functions



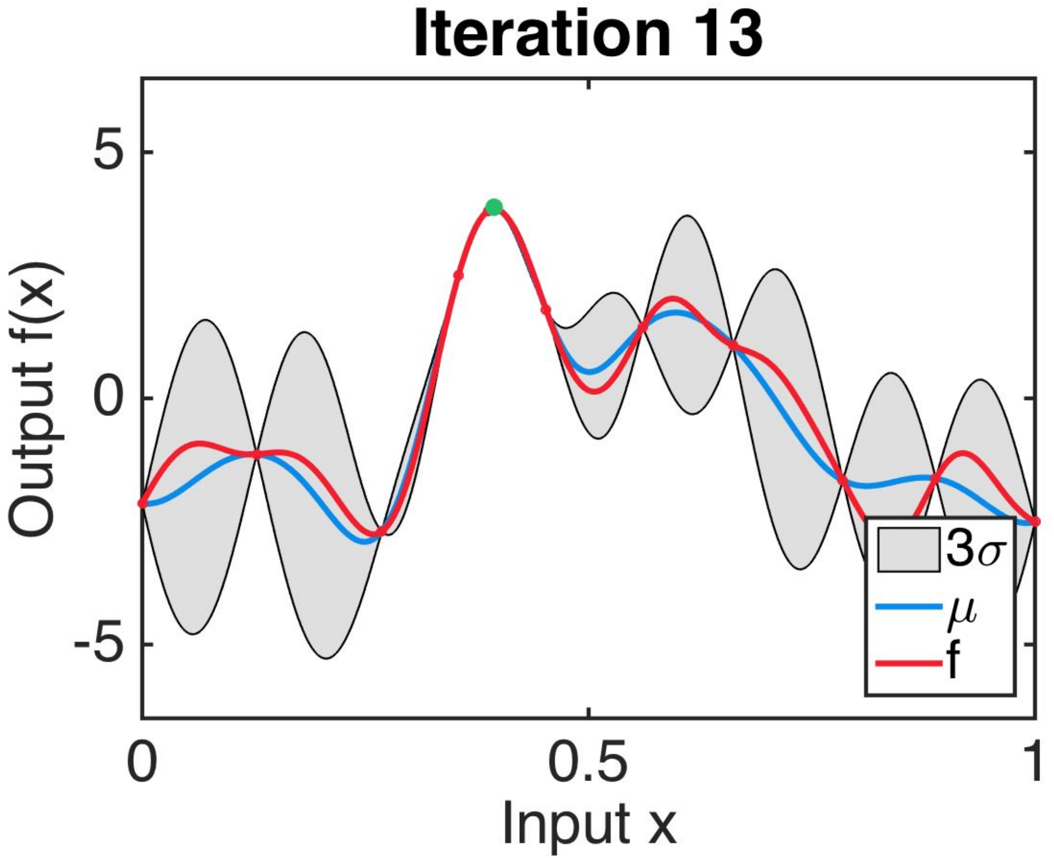
GP-UCB: an example of acquisition functions



GP-UCB: an example of acquisition functions



GP-UCB: an example of acquisition functions

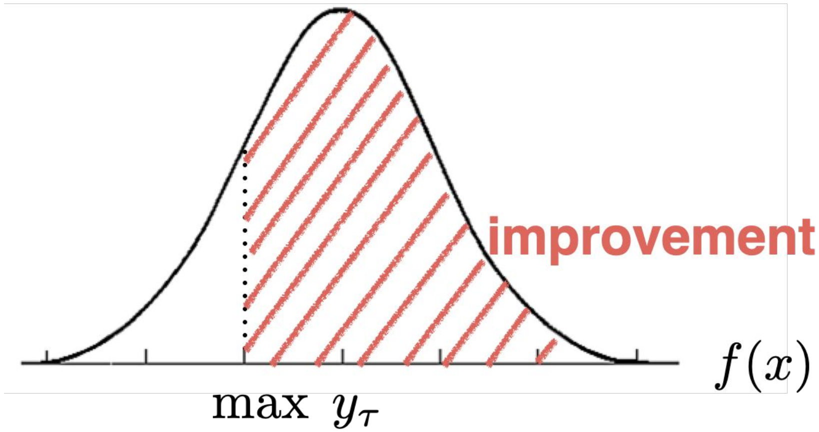


Example of acquisition functions: PI

[Kushner, 1964]

Probability of Improvement (PI)

- Observations: $D_t = \{(x_\tau, y_\tau)\}_{\tau=1}^{t-1}$
- The best observation is $\max\ y_\tau$
- for each x, predict the posterior mean and variance



$$\alpha_t(x) = \Pr[f(x) \geq \max\ y_\tau]$$

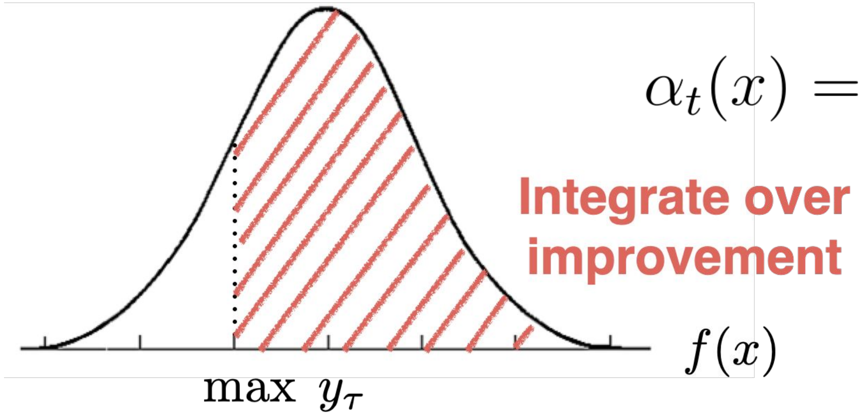
$$\alpha_t(x) = \Pr[f(x) \geq \max\ y_\tau + \epsilon]$$

Example of acquisition functions: EI

[Kushner, 1964]

Expected Improvement (EI)

- Observations: $D_t = \{(x_\tau, y_\tau)\}_{\tau=1}^{t-1}$
- The best observation is $\max\ y_\tau$
- for each x, predict the posterior mean and variance



$$\alpha_t(x) = \mathbb{E}\left[(f(x) - \max y_\tau)_+\right]$$

Entropy Search and Predictive Entropy Search

$$\underset{x_t \in \mathfrak{X}}{\text{maximize}}\ \alpha_t(x_t) \qquad t = 1, \cdots, T$$

$$I(a;b) = H(a) - H(a|b)$$
$$= H(b) - H(b|a)$$

Point to query

Location of global optimum

Observed Data

$$\alpha_t(x) = I(\{x, y\}; x_* \mid D_t)$$

$$= H(p(x_* \mid D_t)) - \mathbb{E}_y[H(p(x_* \mid D_t \cup \{x, y\}))]$$

$$= H(p(y \mid D_t, x)) - \mathbb{E}_{x_*}[H(p(y \mid x_*, D_t, x))]$$

[Hennig & Schuler, 2012; Hernandez-Lobato et al., 2014]

Max-value Entropy Search

[Wang&Jegelka, 2017; Hoffman&Zoubin, 2015]

Point to query

Observed Data

$\cancel{\alpha_t(x) = I(\{x, y\}; x_* \mid D_t)}$ ~~Location of global optimum~~

D-dimensional input space

$\downarrow$

1-dimensional Output space

Global max-value

$$\alpha_t(x) = I(\{x, y\}; y_* \mid D_t)$$

$$= H(p(y \mid D_t, x)) - \mathbb{E}_{y_*}[H(p(y \mid y_*, D_t, x))]$$

Gaussian

Truncated Gaussian

$y \le y_*$

$$\approx \frac{1}{K} \sum_{y_* \in Y_*} \left[\frac{\gamma_{y_*}(x)\psi(\gamma_{y_*}(x))}{2\Psi(\gamma_{y_*}(x))} - \log(\Psi(\gamma_{y_*}(x))) \right]$$

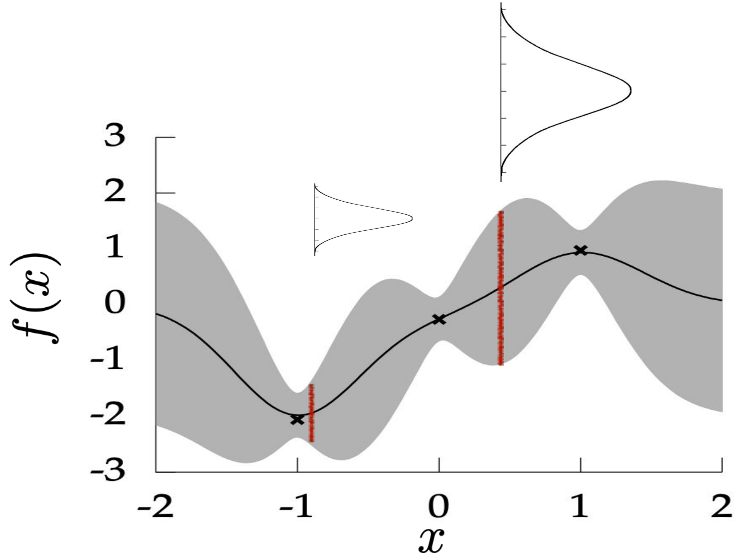
Something Closed-form

Sample y_* with a Gumbel Distribution

Intuition: each f(x) is a Gaussian

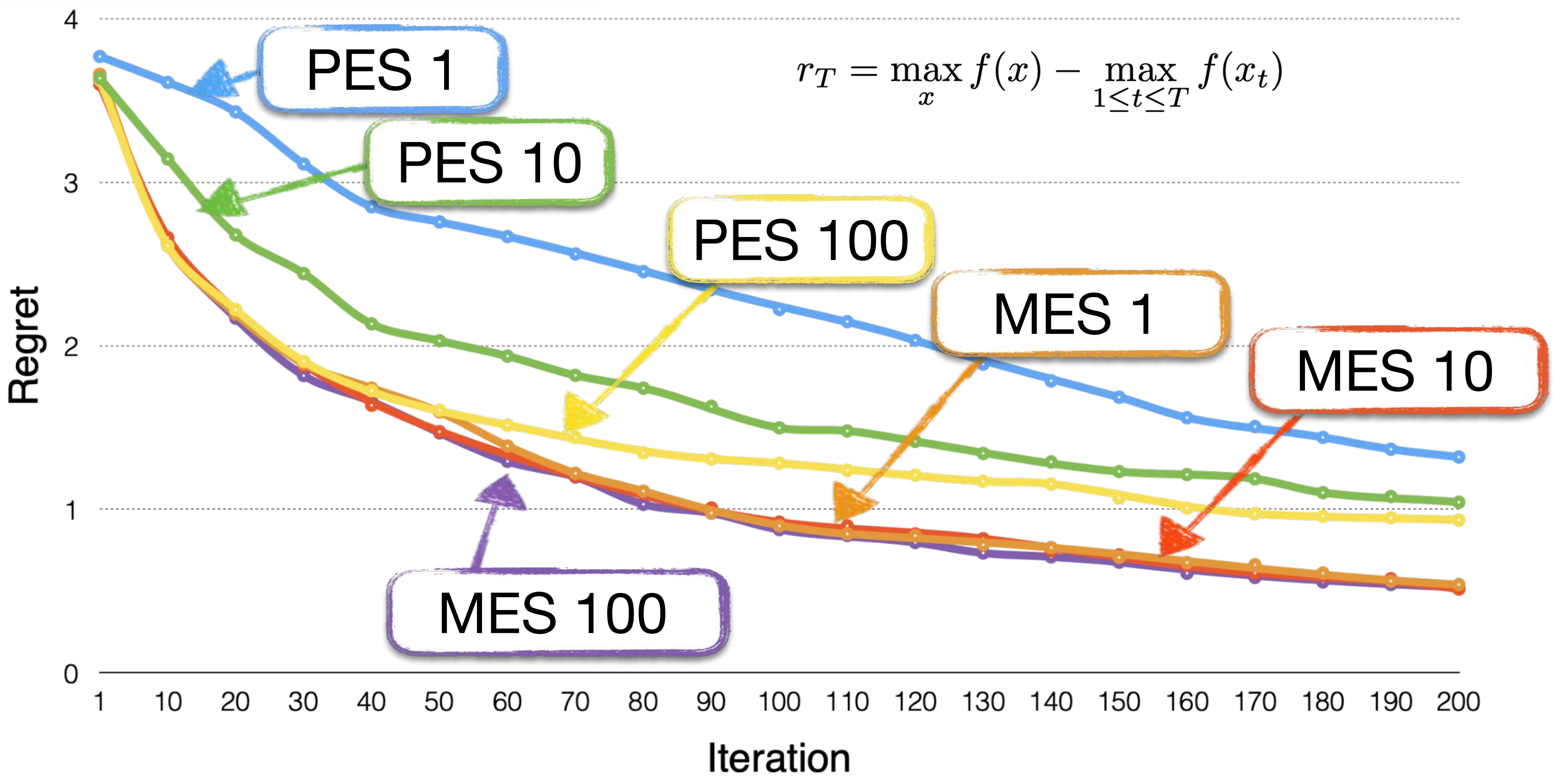
Fisher-Tippett-Gnedenko Theorem

The maximum of a set of i.i.d. Gaussian variables is asymptotically described by a Gumbel distribution.



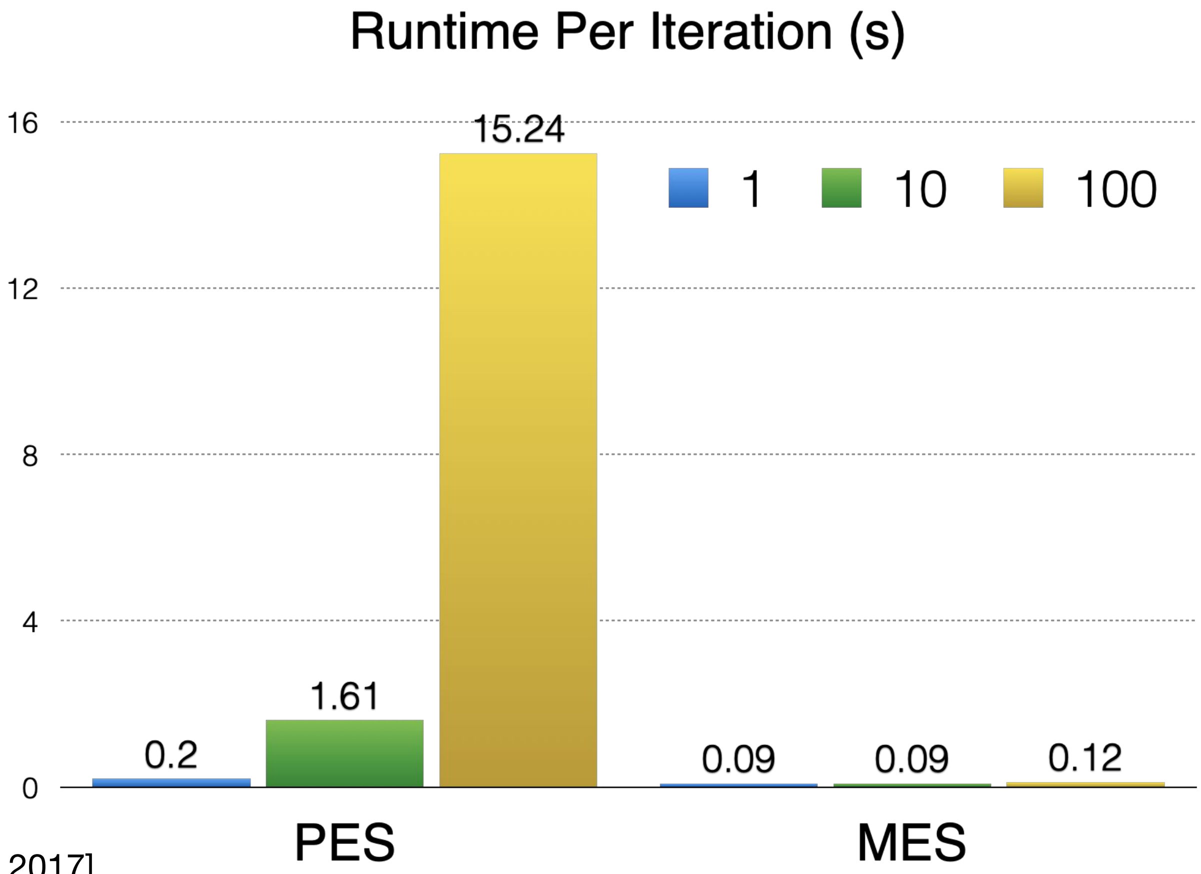
- Sample representative points
- Approximate the max-value of the representative points by a Gumbel distribution [Wang&Jegelka, 2017]

MES gets faster and better empirical results than PES



[Wang&Jegelka, 2017]

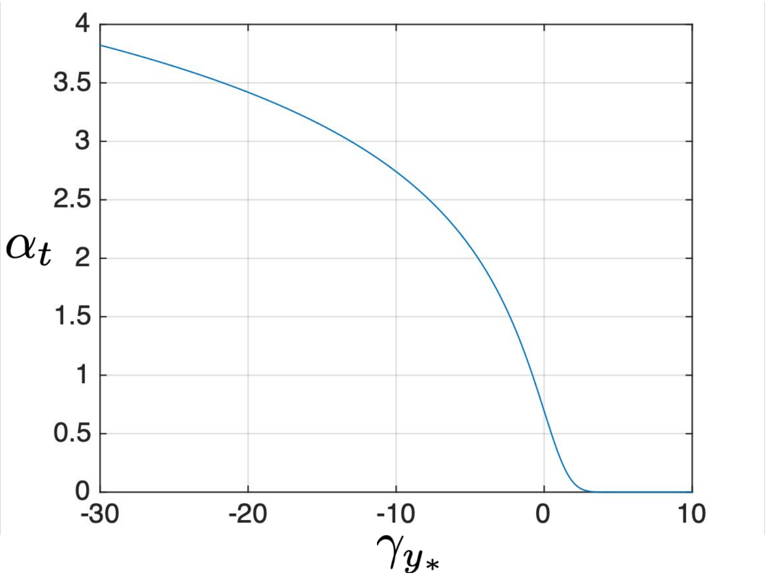
MES gets faster and better empirical results than PES



[Wang&Jegelka, 2017]

Understanding the acquisition function in MES

$$\alpha_t(x) \approx \alpha_t(\gamma_{y_*}(x))$$



$$\gamma_{y_*}(x) = \frac{y_* - \mu_{t-1}(x)}{\sigma_{t-1}(x)}$$

So, $\underset{x}{\text{maximize}}\ \alpha_t(x)$

is equivalent to

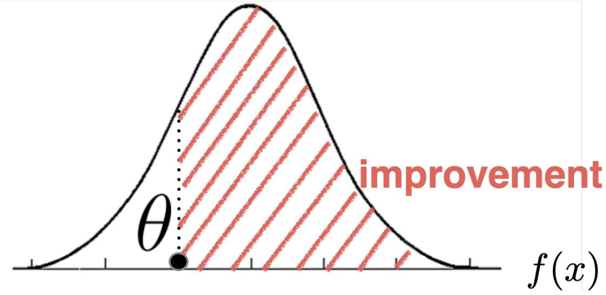
$\underset{x}{\text{minimize}}\ \gamma_{y_*}(x)$.

[Wang&Jegelka, 2017]

Relations among GP-UCB, PI and MES

[Jones, 2001; Wang&Jegelka, 2017]

$$\gamma_\theta(x) = \frac{\theta - \mu_{t-1}(x)}{\sigma_{t-1}(x)}$$



MES $\quad \underset{x}{\text{minimize}}\, \gamma_{y_*}(x)$

PI $\quad \underset{x}{\text{minimize}}\, \gamma_{\max y_\tau + \epsilon}(x)$

GP-UCB $\quad \underset{x}{\text{minimize}}\, \gamma_{\max_{x'} \mu_{t-1}(x') + \beta\sigma_{t-1}(x')}(x)$

GP-UCB, PI and MES are equivalent to $\underset{x}{\text{minimize}}\, \gamma_\theta(x)$ under special cases of θ.

Relations among GP-UCB, PI and MES

[Jones, 2001; Wang&Jegelka, 2017]

$$\gamma_\theta(x) = \frac{\theta - \mu_{t-1}(x)}{\sigma_{t-1}(x)}$$

MES

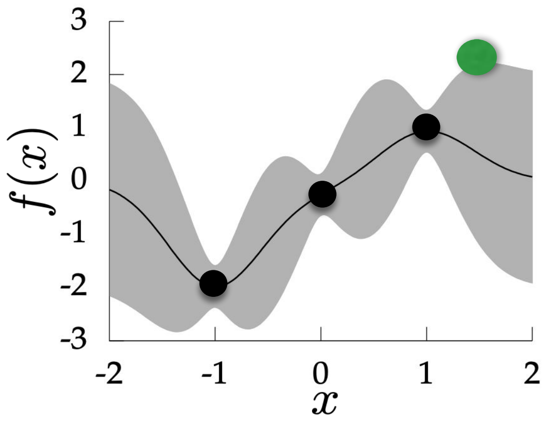
$$\underset{x}{\text{maximize}}\, \mu_{t-1}(x) + \sigma_{t-1}(x) \min_{x'} \gamma_{y_*}(x')$$

PI

$$\underset{x}{\text{maximize}}\, \mu_{t-1}(x) + \sigma_{t-1}(x) \min_{x'} \gamma_{\max y_\tau + \epsilon}(x')$$

GP-UCB

$$\underset{x}{\text{maximize}}\, \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$$



They are also equivalent to $\underset{x}{\text{maximize}}\, \mu_{t-1}(x) + \beta\sigma_{t-1}(x)$ under special cases of β.

Regret bounds for GP-UCB and related methods

[Srinivas et al., 2010; Wang et al., 2016]

Define regret as: $r_T = \max_x f(x) - \max_{1 \le t \le T} f(x_t)$

Key assumptions: $f \sim GP(\mu, k)$

Mean function and kernel are both given.
Optimize in a d-dimensional compact space.

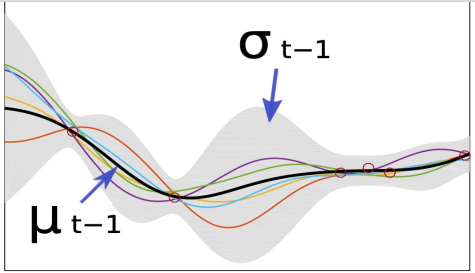
After T iterations, GP-UCB obtains $r_T = O(\sqrt{\frac{d(\log T)^{d+2}}{T}})$.

MES obtains $r_T = O\left(\sqrt{\frac{(\log T)^{d+2}}{T}} + \max_t \min_x \frac{y_* - \mu_{t-1}(x)}{\sigma_{t-1}(x)} \sqrt{\frac{(\log T)^{d+1}}{T}}\right)$.

* For simplicity we only show regret bounds for Gaussian kernels. Regret for other kernels may look different.

Summary of how BayesOpt works

Posterior estimation



$t \leftarrow t+1$

Define an acquisition function

UCB: $\mu_{t-1}(x) + \beta\sigma_{t-1}(x)$

EI: $\mathbb{E}\left[(f(x) - \max y_\tau)_+\right]$

PI: $\Pr[f(x) \geq \max\ y_\tau + \epsilon]$

ES: $I(\{x,y\}; x_* \mid D_t)$

$I(\{x,y\}; y_* \mid D_t)$

and others…

Evaluate f at

$x_t = \arg\max \alpha_t(x)$

Challenges, open problems and some attempts

Selected topics in BayesOpt

- → High dimensional search space
- → Unknown GP prior
- → Parallel evaluations
- → Unknown constraints
- → Applications in robotics

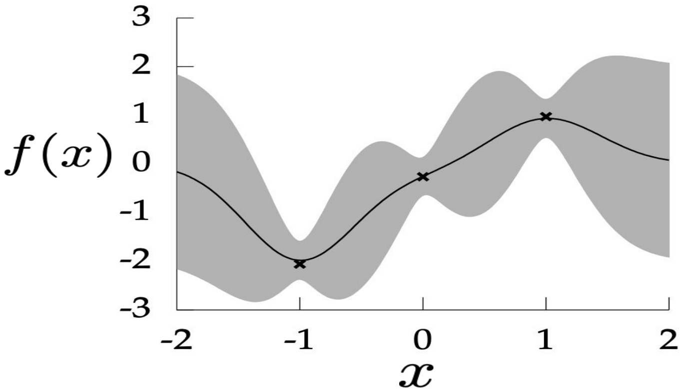
Google Research

Challenges, open problems and some attempts

High dimensional search space

Challenges in high-dimensional BO

- optimizing multi-peak acquisition functions in high dimensions
  computationally challenging



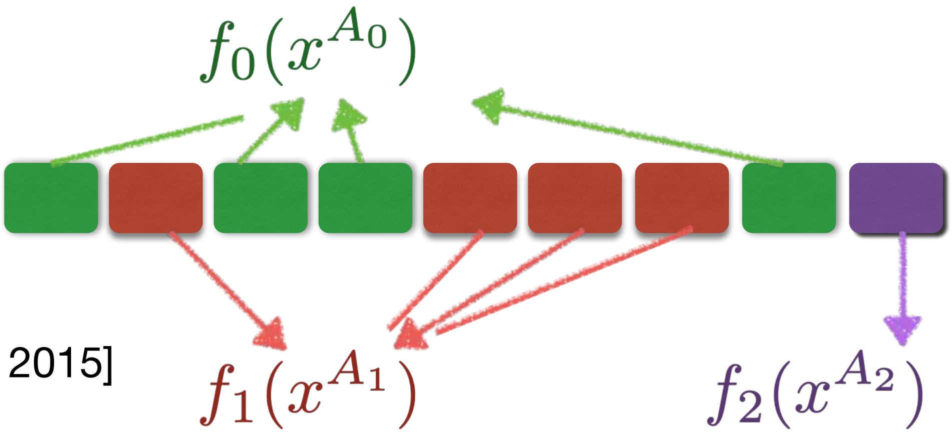
- estimating a nonlinear function in high input dimensions: need more observations
  statistically challenging

regret $r_T \approx O\left(\sqrt{\frac{(\log T)^{d+2}}{T}}\right)$

[Wang et al., 2013; Djolonga el al., 2013; Kandasamy et al., 2015]

Possible solution: additive Gaussian processes

$$f(x) = \sum_{m \in [M]} f_m(x^{A_m})$$



[Hastie&Tibshirani, 1990; Kandasamy et al., 2015]

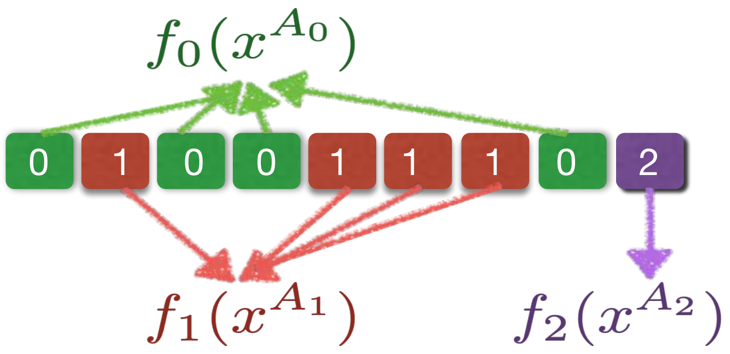
- optimize acquisition function block-wise
  computational efficiency
- lower-complexity functions
  statistical efficiency

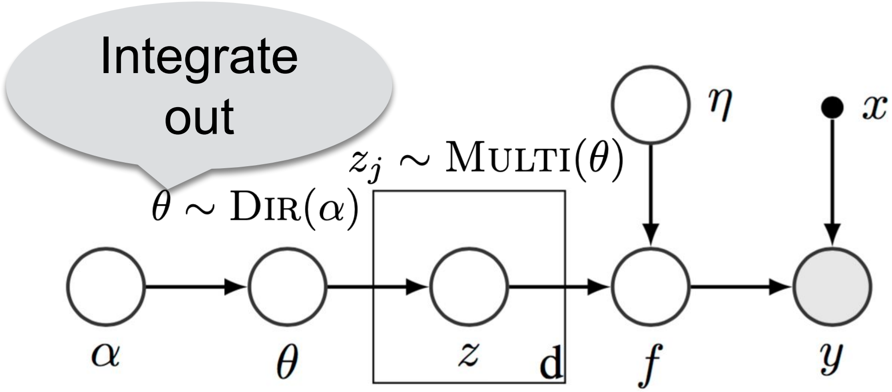
What is the additive structure?

Structural Kernel Learning (SKL)

[Wang et al., 2017]

$$f = f_0 + f_1 + f_2$$





Decomposition indicator:

z = [0 1 0 0 1 1 1 0 2]

Learn z!

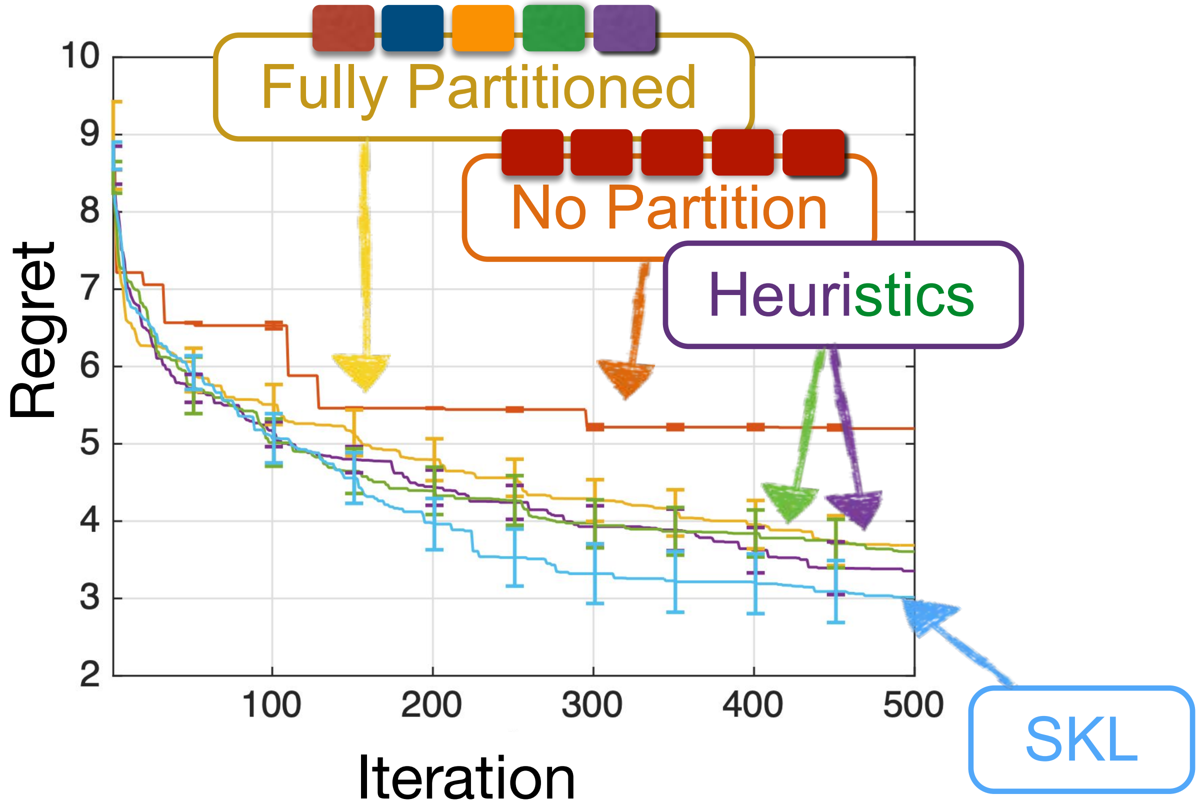
Learn posterior

$$p(z \mid D_n; \alpha)$$

via Gibbs sampling.

easy updates

Empirical results for Structural Kernel Learning (SKL)

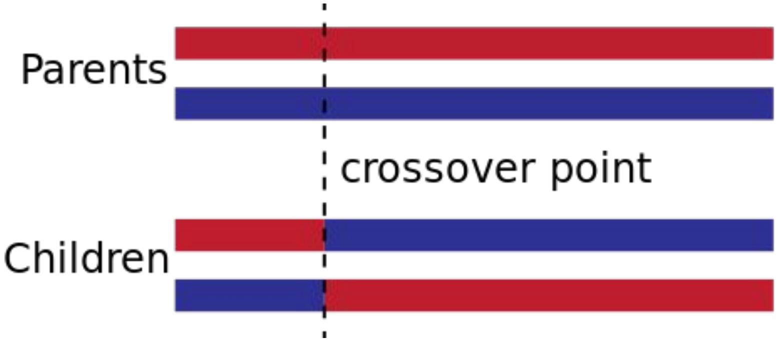


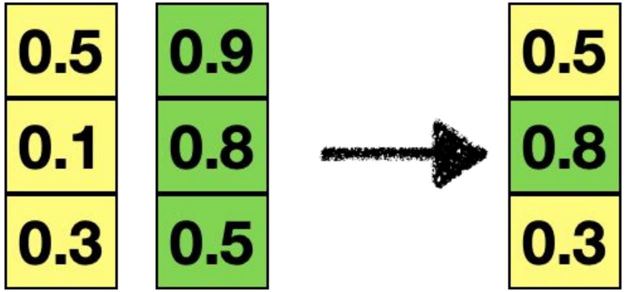
[Wang et al., 2018]

Connection to genetic algorithms?

Evolutionary/Genetic algorithms:

- maintain ensemble of promising points
- new points from exchanging coordinates of good points randomly





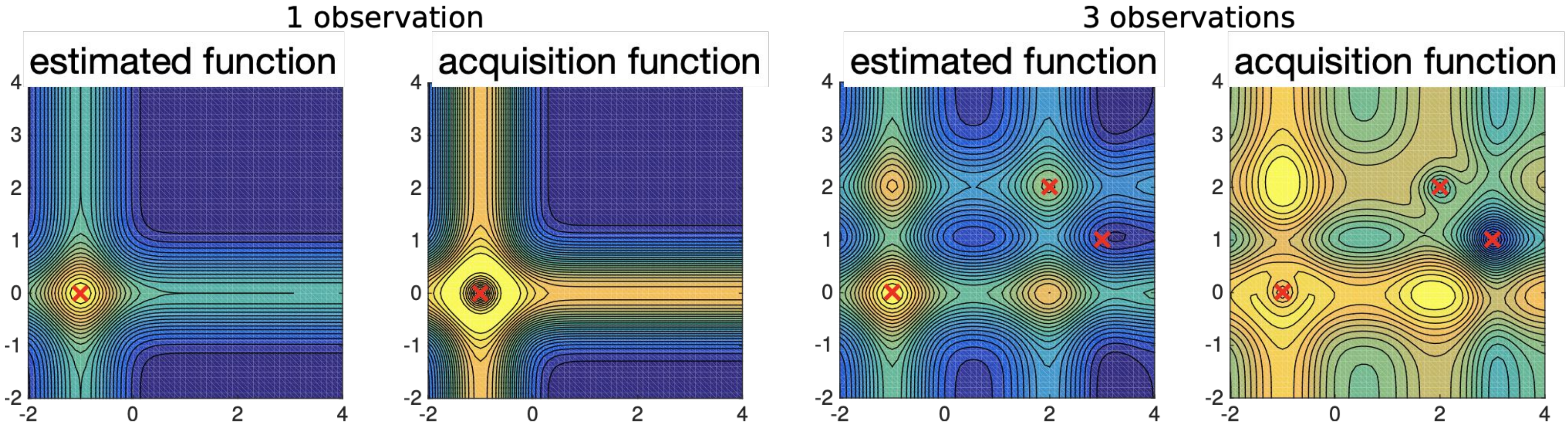
https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm)

Connection to genetic algorithms?

[Wang et al., 2018]

BayesOpt with additive GPs

toy example: 2D



Observed good points: [-1,0], [2,2]

Query points: [-1,2], [2,0]

Learned instead of completely random coordinate partition.

Other ideas to solve high-dim BayesOpt

- REMBO: low-dim embedding [Wang et al., JAIR 2016]
- BOCK: BO with cylindrical kernels [Oh et al., ICML 2018]
- Additive GPs with overlapping groups [Rolland et al., AISTATS 2018]
- ……

Joint problems:

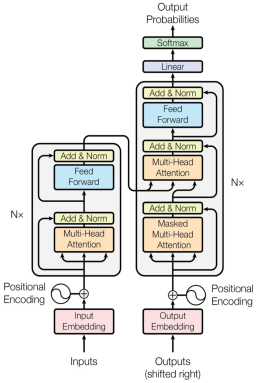
Assume special structures of high-dim functions but with little data, it is difficult to verify if the assumptions are true.

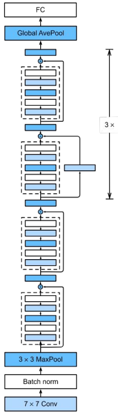
Challenges, open problems and some attempts

Parallel evaluations

BayesOpt with parallel compute resources

- GPUs running in parallel for hyperparameter tuning in deep learning;
- group of robots for offline learning of control parameter;
- parallel wet lab experiments for biology and chemistry applications; etc.





[https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04; https://d2l.ai/chapter_convolutional-modern/resnet.html; https://ai.googleblog.com/2016/03/deep-learning-for-robots-learning-from.html; https://en.wikipedia.org/wiki/Wet_lab]

Some ideas to propose a batch of queries

- Instead of optimizing one input over information gain, optimize Q inputs. [Shah&Ghahramani, 2015]

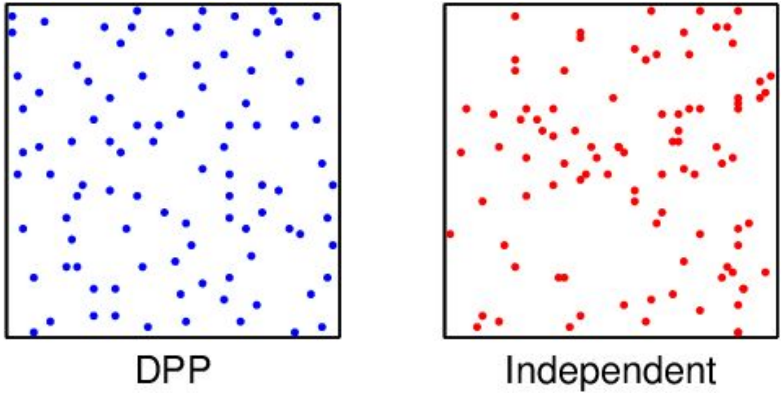
$$a_{\mathrm{PPES}}(\mathcal{S}_t|\mathcal{D}) = \mathrm{H}\big[p(\boldsymbol{x}^*|\mathcal{D})\big] - \mathbb{E}_{p\left(\{y_q\}_{q=1}^{Q}\middle|\mathcal{D},\mathcal{S}_t\right)}\Big[\mathrm{H}\big[p\big(\boldsymbol{x}^*|\mathcal{D}\cup\{\boldsymbol{x}_q, y_q\}_{q=1}^{Q}\big)\big]\Big]$$

- Choose a new point based on expected acquisition function under all possible outcomes of pending evaluations. [Snoek et al., 2012]

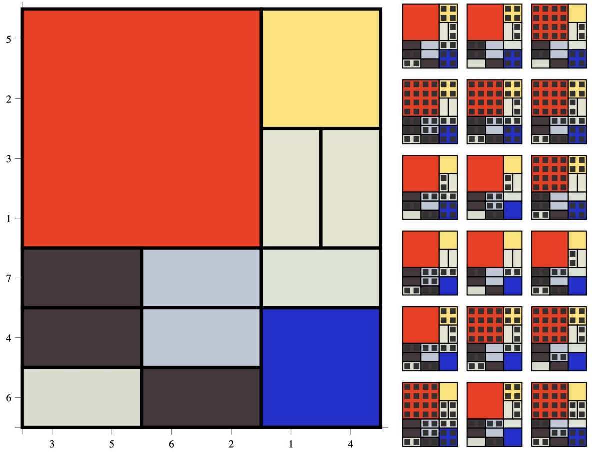
$$\hat{a}(\mathbf{x}\,;\,\{\mathbf{x}_n, y_n\}, \theta, \{\mathbf{x}_j\}) = \int_{\mathbb{R}^J} a(\mathbf{x}\,;\,\{\mathbf{x}_n, y_n\}, \theta, \{\mathbf{x}_j, y_j\})\, p(\{y_j\}_{j=1}^{J} \mid \{\mathbf{x}_j\}_{j=1}^{J}, \{\mathbf{x}_n, y_n\}_{n=1}^{N})\, \mathrm{d}y_1 \cdots \mathrm{d}y_J$$

Some ideas to propose a batch of queries

- Use determinantal point process (DPP) to generate a diverse set of queries. [Kathuria et al., 2016]
- Use a Mondrian process to propose one query per partition. [Wang et al., 2018]



[Kulesza&Taskar, 2011]



[Roy&Teh, 2009]

Potential issues with existing methods

- Computational cost is usually high.
- Not all adapt to asynchronous parallel BayesOpt settings.
- Difficult to debug especially in high-dimensional settings.
- Parallel BayesOpt typically co-occur with large scale high-dimensional problems, but a joint solution for these conditions is not yet satisfying.

Challenges, open problems and some attempts

Unknown priors

Bayesian optimization with an unknown prior

Estimate "prior" from data

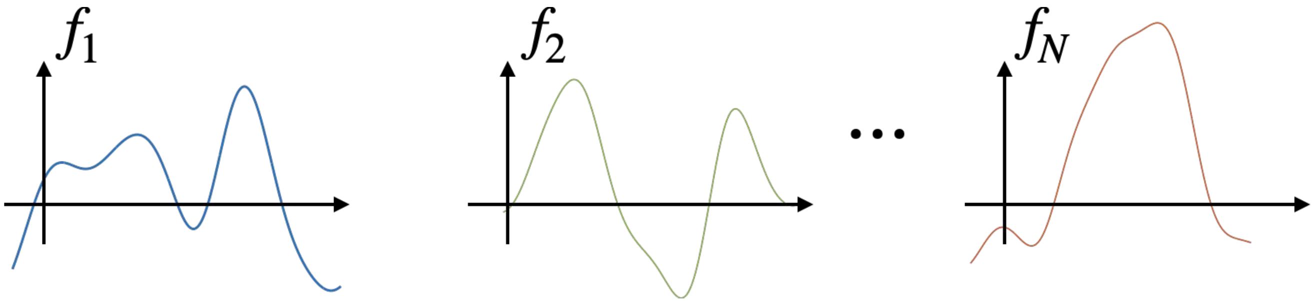
- maximum likelihood
- hierarchical Bayes

- Regret bounds exist only when prior is assumed given
- bad settings of priors make BO perform poorly and seem to be a bad approach

Bayesian optimization with an unknown prior

meta / multi-task / transfer learning

- Our idea: learn the "prior" from past experience with similar functions
- Assumption: we can collect data on functions sampled from the same prior $f_1, f_2, \cdots, f_N \sim GP(\mu, k)$



[Wang*&Kim*&Kaelbling, NeurIPS 2018]

How to learn the GP prior?

$$|\mathcal{X}| = M$$

Use finite input space to illustrate; extensions to continuous case requires more assumptions. [Wang et al., 2018 + ongoing work]

Prior estimation with meta training data $\{[(x_j, y_{ij})]_{j=1}^{M}\}_{i=1}^{N}$

Task 1	(x_1, y_11)	(x_2, y_12)	……	(x_M, y_1M)
Task 2	(x_1, y_21)	(x_2, y_22)	……	(x_M, y_2M)
……	……	……	……	……
Task N	(x_1, y_N1)	(x_2, y_N2)	……	(x_M, y_NM)
New Task	?	?	……	?

How to estimate the GP prior?

$$|\mathcal{X}| = M$$

Task 1	(x_1, y_11)	(x_2, y_12)	……	(x_M, y_1M)
Task 2	(x_1, y_			(x_M, y_2M)
……	……			……
Task N	(x_1, y_			(x_M, y_NM)
New Task	?	?	……	?

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix}$$

Unbiased prior estimator

$$\hat{\mu}(\mathcal{X}) = \frac{1}{N} Y^T 1_N \sim \mathcal{N}(\mu(\mathcal{X}), \frac{1}{N}(k(\mathcal{X}) + \sigma^2 \boldsymbol{I}))$$

$$\hat{k}(\mathcal{X}) = \frac{1}{N-1}(Y - 1_N \hat{\mu}(\mathcal{X})^T)^T (Y - 1_N \hat{\mu}(\mathcal{X})^T) \sim \mathcal{W}(\frac{1}{N-1}(k(\mathcal{X}) + \sigma^2 \boldsymbol{I}), N-1)$$

How to estimate the GP posterior?

$$|\mathcal{X}| = M$$

Task 1	(x_1, y_11)	(x_2, y_12)	……	(x_M, y_1M)
Task 2	(x_1, y_			(x_M, y_2M)
……	……			……
Task N	(x_1, y_			(x_M, y_NM)
New Task	?	?	……	?

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1M} \\ \cdots & & \cdots \\ y_{N1} & \cdots & y_{NM} \end{bmatrix}$$

Unbiased posterior estimator

$$\hat{\mu}_t(x) = \hat{\mu}(x) + \hat{k}(x, \boldsymbol{x}_t)\hat{k}(\boldsymbol{x}_t, \boldsymbol{x}_t)^{-1}(\boldsymbol{y}_t - \hat{\mu}(\boldsymbol{x}_t))$$

$$\hat{\sigma}_t^2(x, x') = \frac{N-1}{N-t-1}\left(\hat{k}(x, x') - \hat{k}(x, \boldsymbol{x}_t)\hat{k}(\boldsymbol{x}_t, \boldsymbol{x}_t)^{-1}\hat{k}(\boldsymbol{x}_t, x')\right)$$

Regret bound without the knowledge of the GP prior

regret: $r_T = \max_{x \in \mathcal{X}} f(x) - \max_{t \in [T]} f(x_t)$

Important assumptions:

- functions are sampled from the same Gaussian process
- enough number of functions in offline phase $N > T + 20$

Given T observations on the new function f, with probability $1 - \delta$,

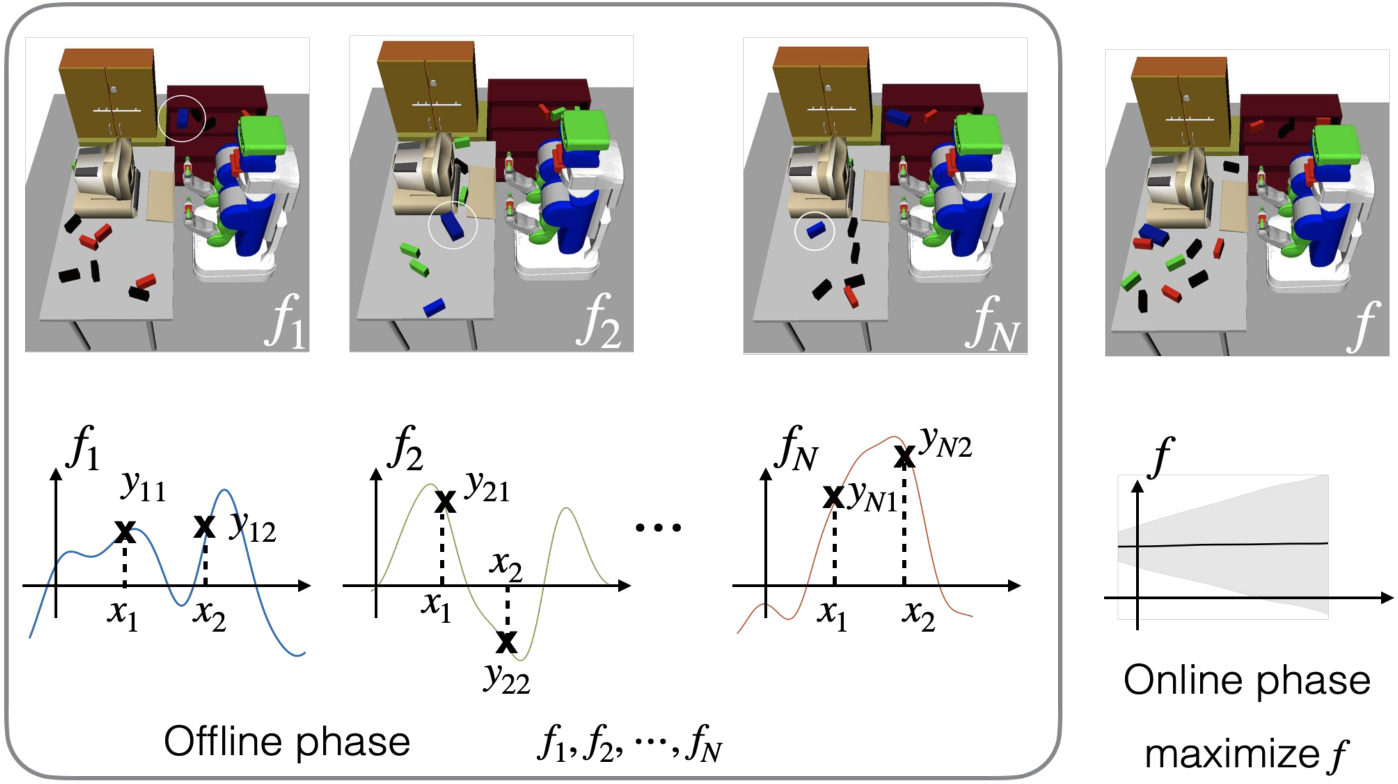
regret $\quad r_T \leq O\left(\left(\sqrt{\frac{1}{N-T}} + C\right)\left(\sqrt{\frac{\log T}{T}} + \sigma^2\right)\right) \to C\sigma$

constant depending on δ

observation noise

[Wang*&Kim*&Kaelbling, NeurIPS 2018]

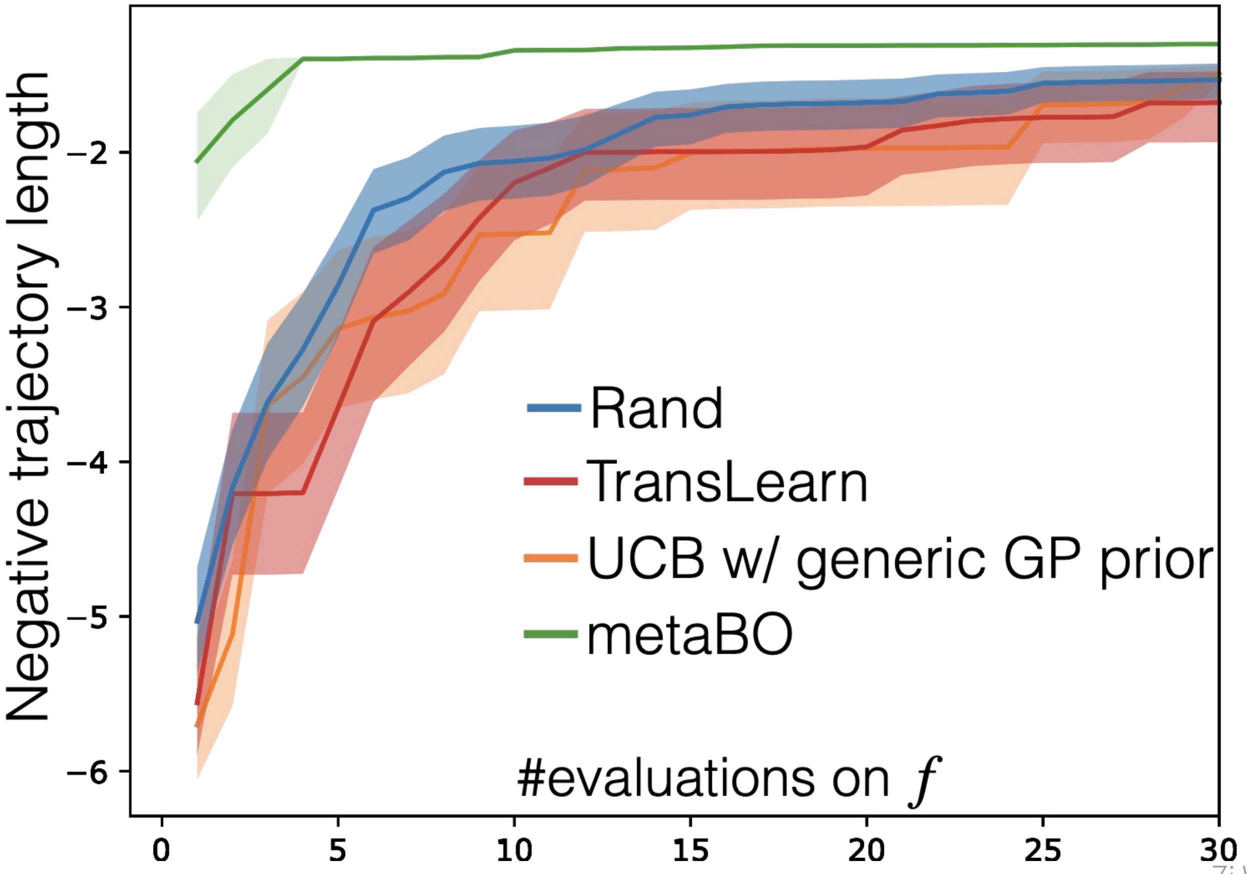
Empirical results on block picking and placing



[Wang*&Kim*&Kaelbling, NeurIPS 2018]

MetaBO gives better performance with fewer samples

[Wang et al., NeurIPS 2018]



Challenges, open problems and some attempts

Applications in robotics

Type of tasks:

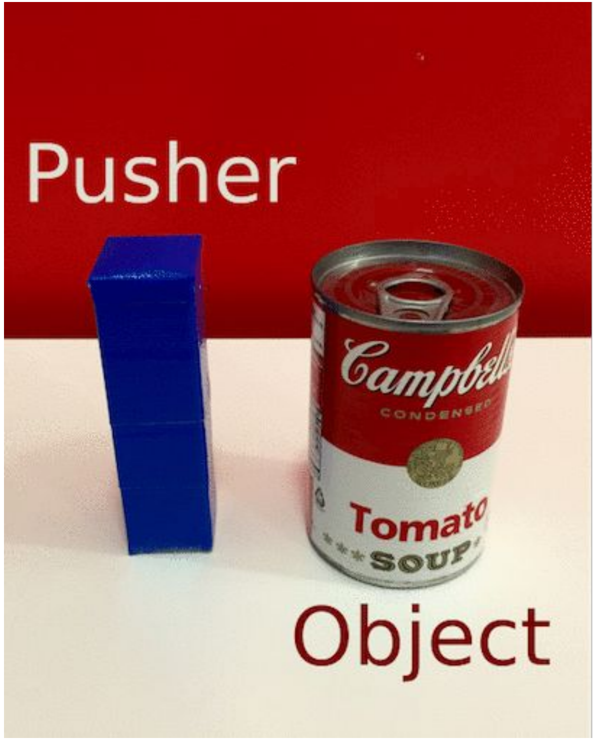
complex

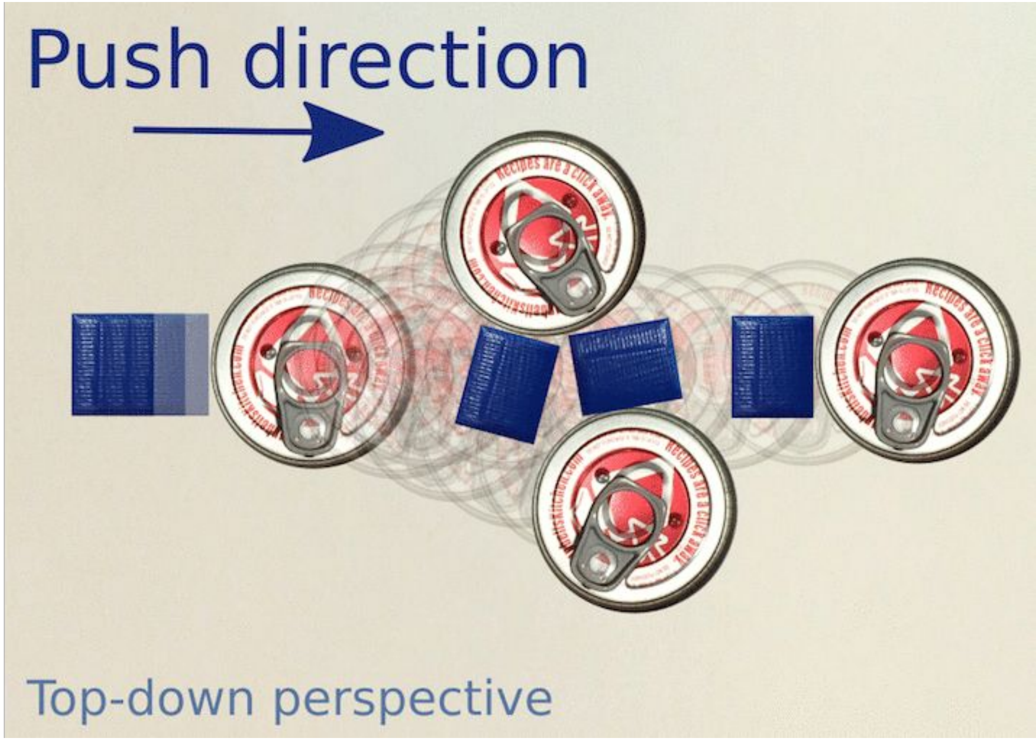
long-horizon

stochastic

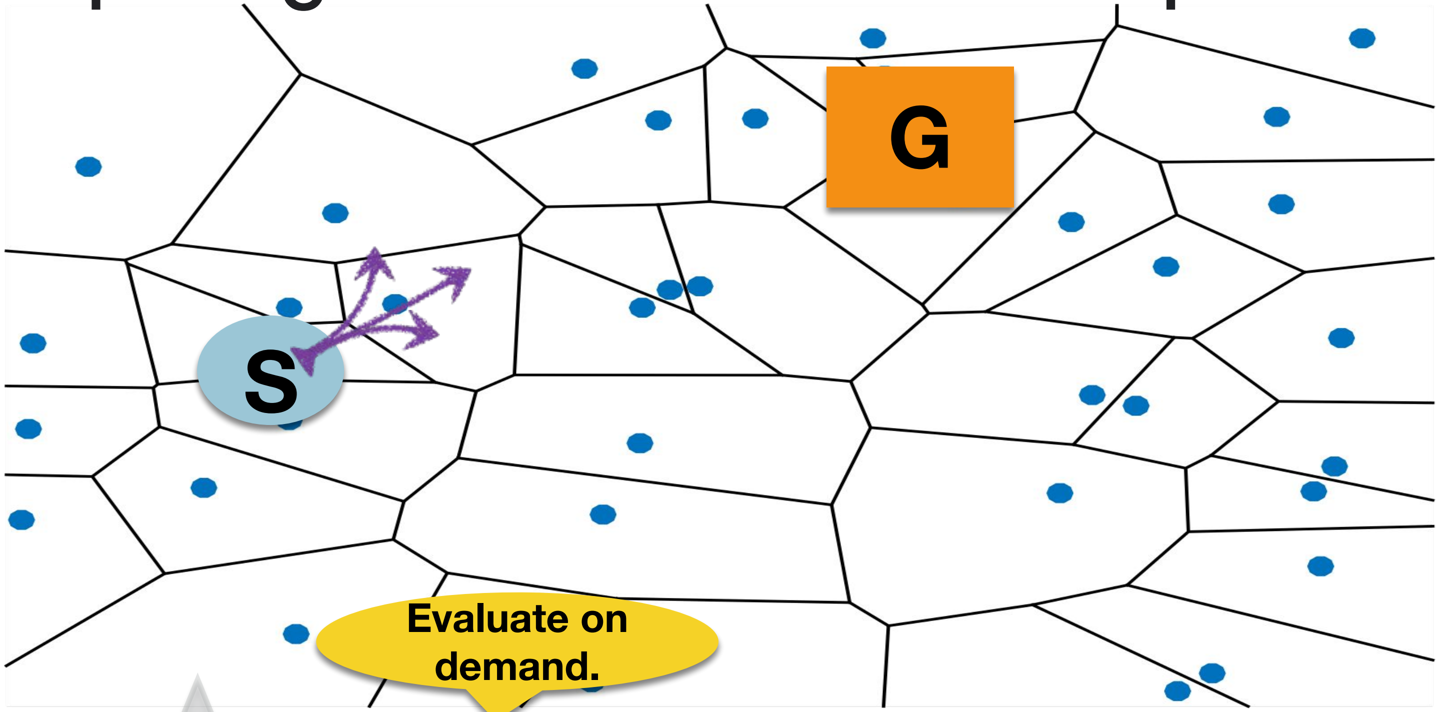
fully-observable

Learning a pushing skill with multi-modal dynamics





Computing the action values is expensive

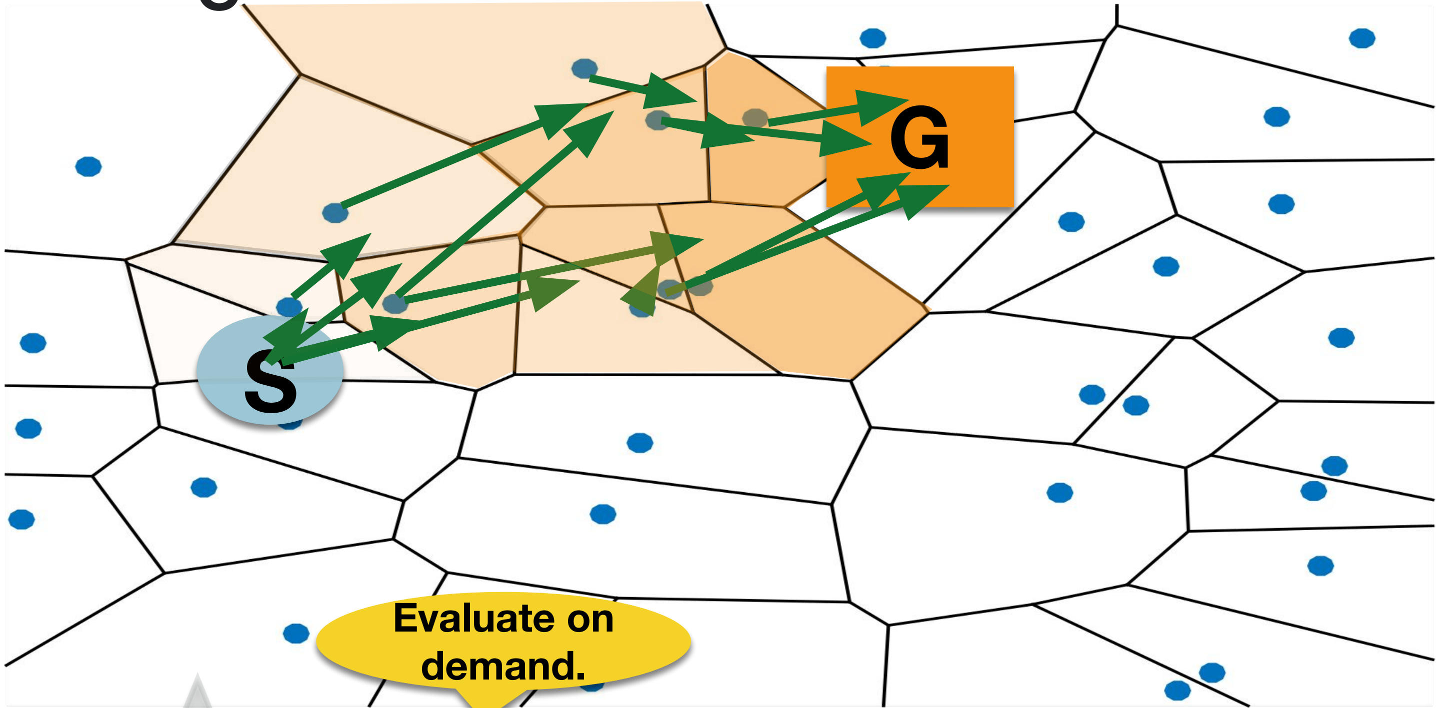


$$\text{ } = \arg\max_a \sum_{s' \in \tilde{S}} \boxed{P_{s'|s,a}(s' \mid s, a)} \left(R(s' \mid s, a) + \gamma^{\Delta t} V(s')\right)$$

Solve via BO

[Barto et al., 1995]

Focusing on relevant states with RTDP

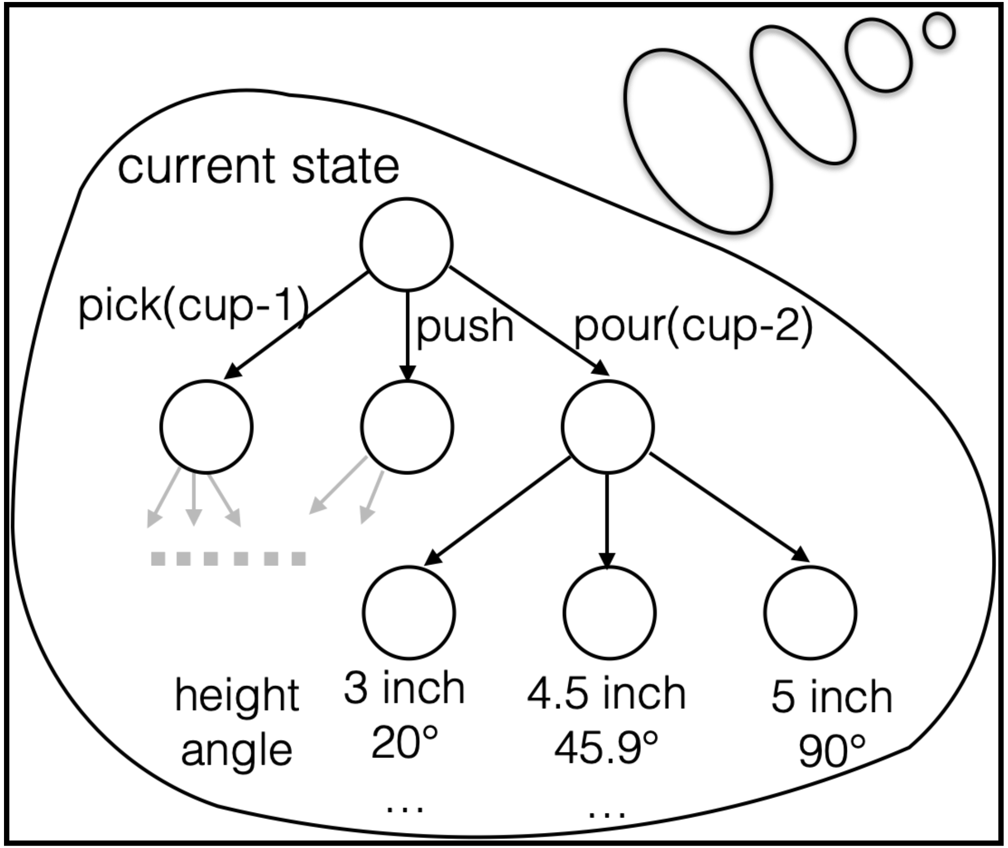


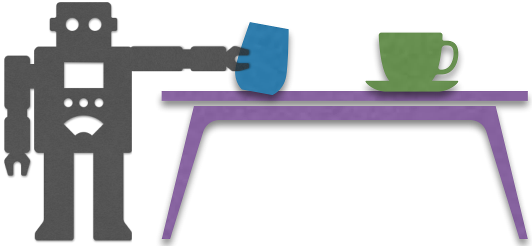
Evaluate on demand.

$$= \arg\max_{a} \sum_{s' \in \tilde{S}} \boxed{P_{s'|s,a}(s' \mid s, a)} \left(R(s' \mid s, a) + \gamma^{\Delta t} V(s')\right)$$

Solve via BO

How to plan with learned skills?

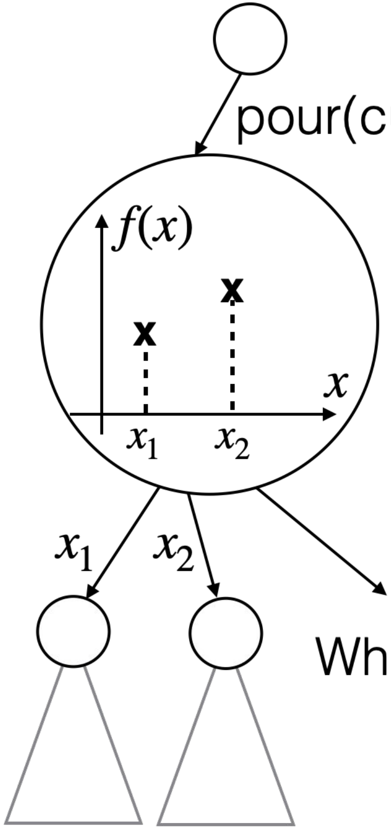




- Sample continuous skill parameters
- Tree search on both discrete and continuous variables

How to sample skill parameters for the planner?

pour(cup-1, cup-2)

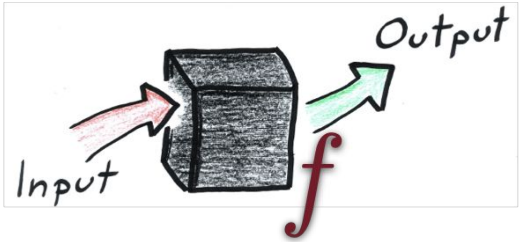


$$f(x_1) = 1.5$$

$$f(x_2) = 3.1$$

What x to try next?

Treat the problem of sampling skill parameters as a black-box function optimization problem

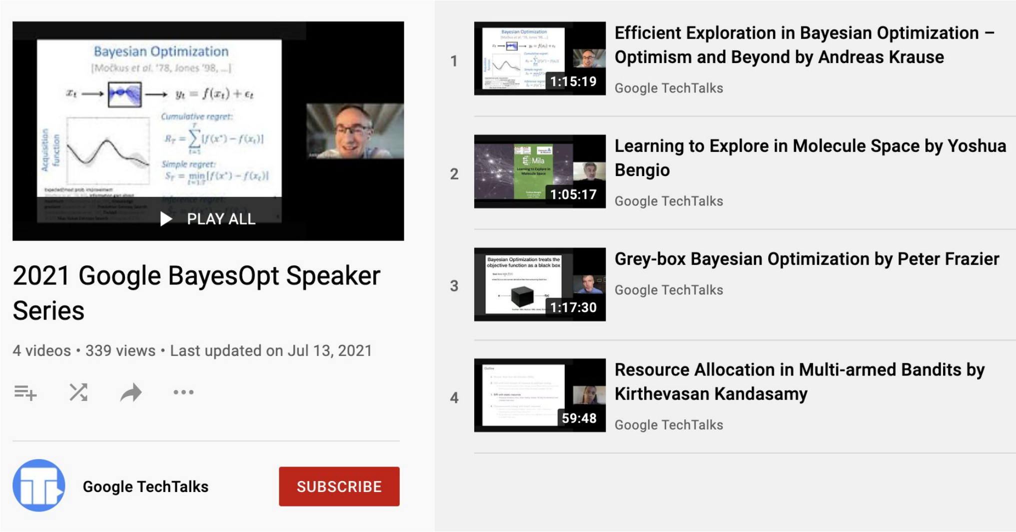


$$\underset{x \in \mathcal{X}}{\text{maximize}} \; f(x)$$

More talks on BayesOpt

2021 Google BayesOpt Speaker Series

now publicly available at Google TechTalks channel on YouTube



Google Research

Questions?