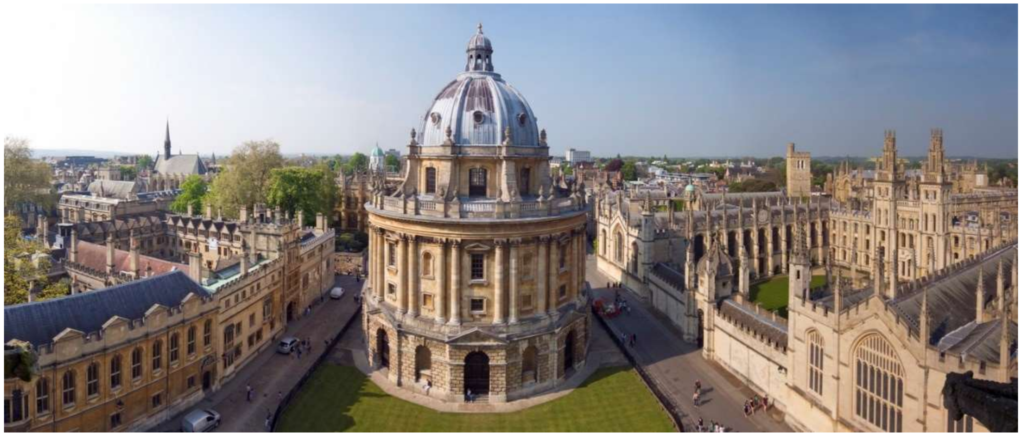
Part II: Advances in Bayesian Optimization

Sec 1: Batch Bayesian Optimization

Dr Vu Nguyen
vu@robots.ox.ac.uk
University of Oxford



Agenda

- Hyperparameter Tuning and Experimental Design as Black-Boxes
- Part I: Bayesian Optimization
- Part II: Recent Advances in Bayesian Optimization
  - Batch Bayesian Optimization
  - High dimensional Bayes Opt
  - Mixed Categorical-Continuous Bayes Opt
- Research Directions in Bayesian Optimization

Outline Part II.1: Batch Bayesian Optimization

- Introduction and Problem Statement
- Peak Suppression Approaches
  - Constant Liar
  - Batch Upper Confidence Bound (GP-BUCB)
  - Local Penalization
- Budgeted Batch Bayesian Optimization
- Thompson Sampling for Batch Bayes Opt
- Asynchronous Batch Bayes Opt

Batch Bayesian Optimization

- Evaluating multiple experiments take the same time as evaluating single experiment.

x_1 $\quad$ $y_1 = f(x_1)$

x_2 $\quad$ $y_2 = f(x_2)$

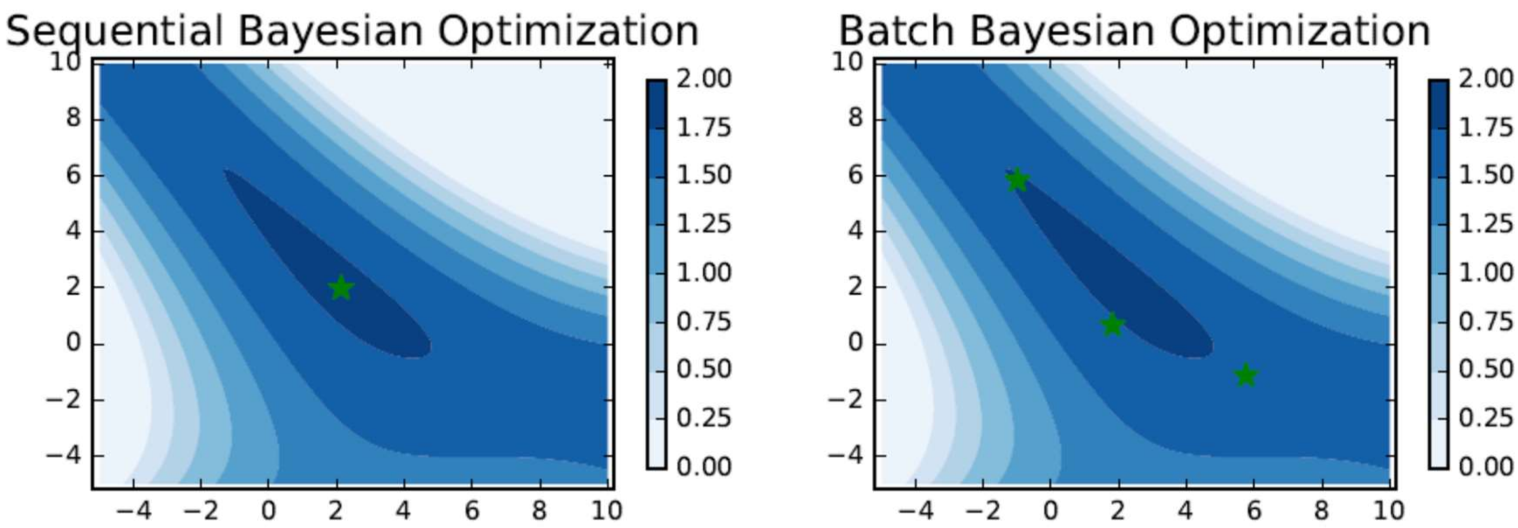
x_3 $\quad$ $y_3 = f(x_3)$

Evaluate **simultaneously**

Batch Bayesian Optimization

- When parallel experiments can be done, we suggest multiple points for evaluations at each iteration in parallel.

$$\boldsymbol{X}_t = [\boldsymbol{x}_{t1}, \boldsymbol{x}_{t2}, ..., \boldsymbol{x}_{tn_t}] = \underset{\boldsymbol{x} \in \mathcal{X}}{\text{argmax}} \alpha_t (\boldsymbol{x})$$



Batch Bayesian Optimization

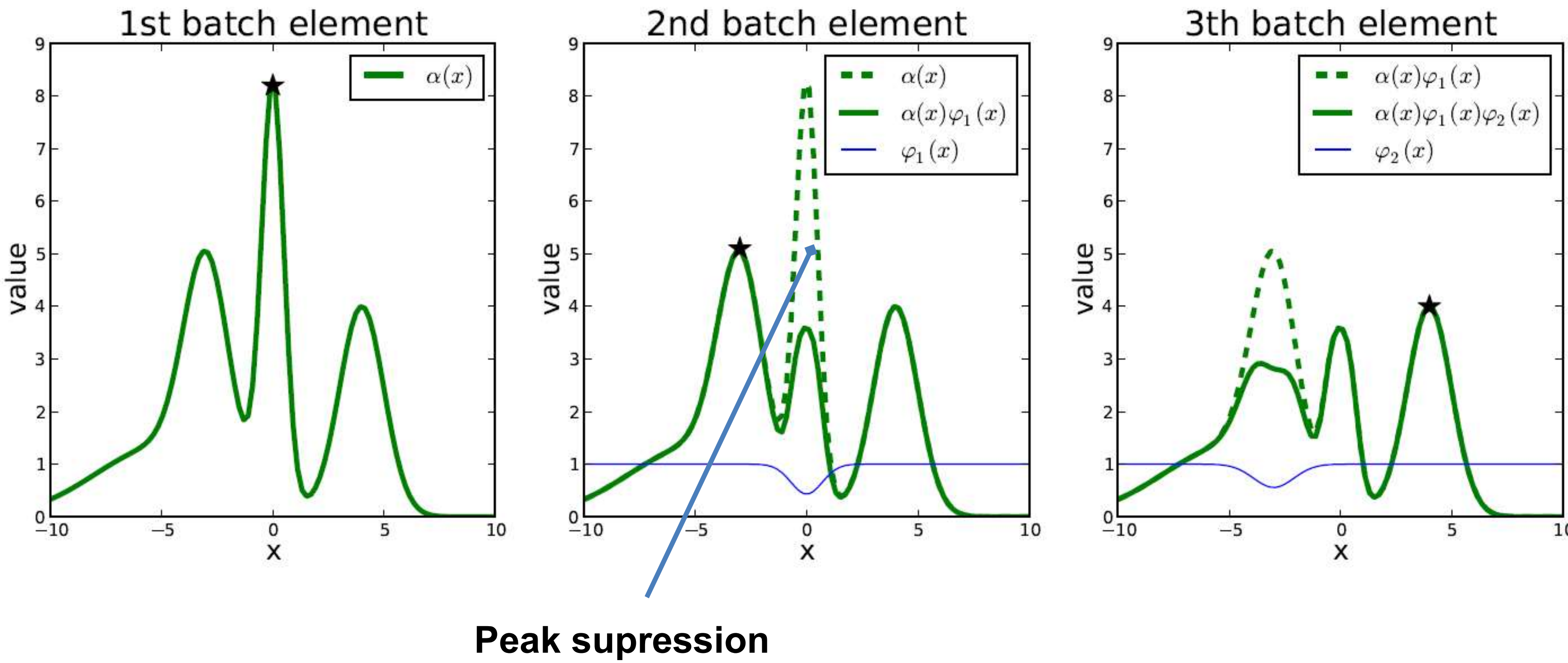
- Selects a batch of points for parallel evaluations at each iteration.
- Recent research:
  - GP-BUCB: [Desautels et al ICML'12]
  - Parallel PES: [Shah et al NIPS'15]
  - Local Penalization: [Gonzalez et al AISTATS'16]
  - Determinantal Point Process: [Kathuria et al NIPS'16]
  - Knowledge Gradient: [Wu et al NIPS'16]
  - Thompson sampling [Lobato et al ICML'17]
  - Asynchronous parallel Bayes opt [Kandasamy et al AISTATS 2018]
  - Asynchronous BO using improved local penalisation [Ahvin et al ICML19]

Outline Part II.1: Batch Bayesian Optimization

- Introduction and Problem Statement
- Peak Suppression Approaches
  - Batch Upper Confidence Bound (GP-BUCB)
  - Local Penalization
- Budgeted Batch Bayes Opt
- Thompson Sampling for Batch Bayes Opt

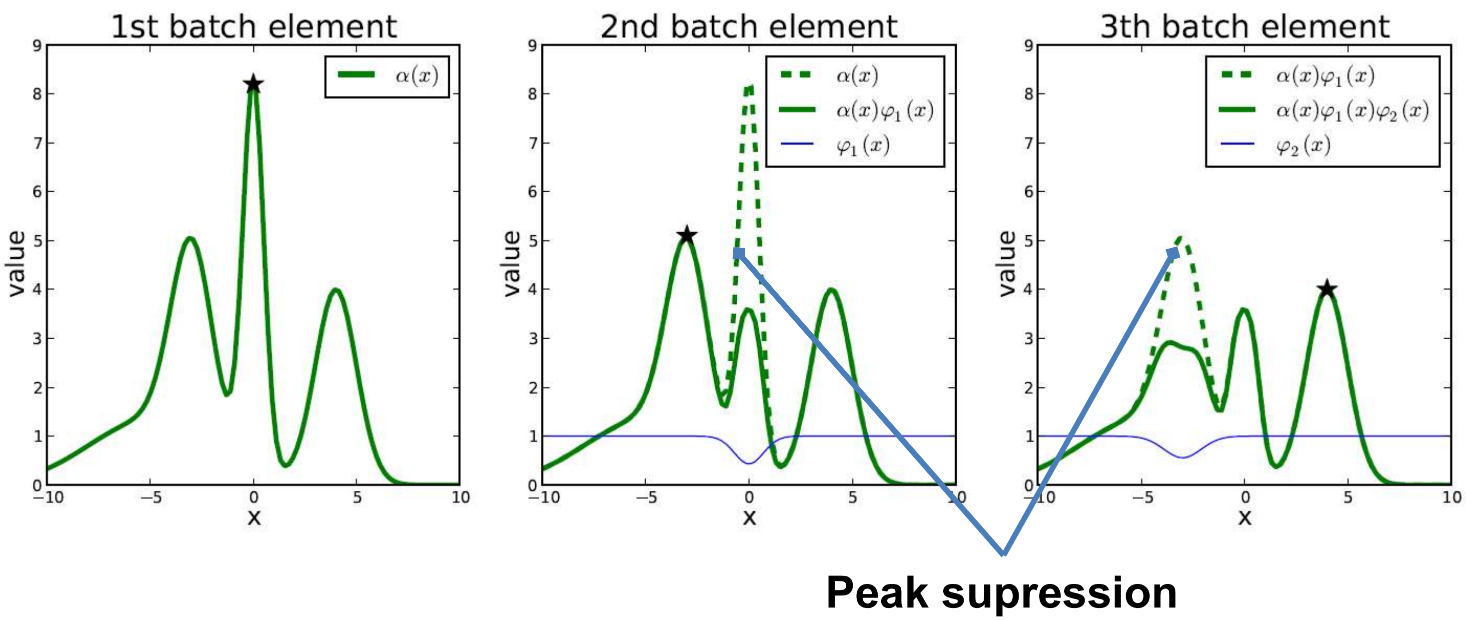
Peak Suppression Approaches

- Intuition: Identify the peaks of the acquisition functions as the batch of points for testing.



Peak Suppression Approaches

- Intuition: Sequentially select a peak, then suppress this peak and move to the next one.
- There are different ways to suppress the peaks.



Peak Suppression Approaches

- The acquisition function is a form of mean and variance.

$$\alpha_t^{GP-UCB}(x) = \mu_t(x) + \sqrt{\beta_t} \times \sigma_t(x)$$

- The peaks of the acquisition function will have high mean $\mu(x)$ and high variance $\sigma(x)$.
- We can suppress the peak by reducing its mean (and/or) variance.

Desautels, T et al 2014

Peak Suppression – GP-BUCB

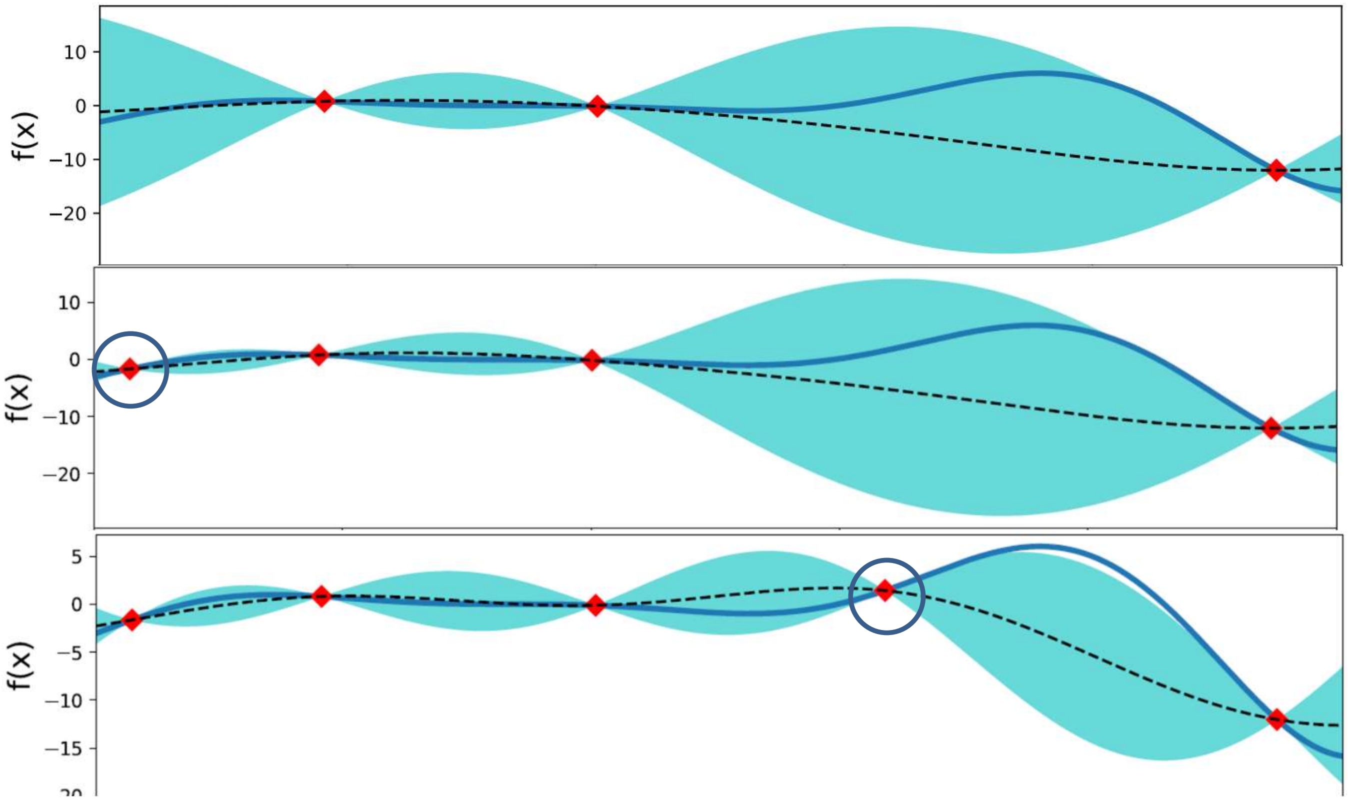
- GP-BUCB sequentially find a batch $X_t = [x_{t,1}, x_{t,2}, \dots, x_{t,B}]$ as follows
  1. Select the first element $x_{t,1}$ in a batch like the standard BO.
  2. Hallucinating the output $y_{t,1} = \mu(x_{t,1})$ by the GP predictive mean,
  3. Update the variance function by inserting $x_{t,1}$, $\sigma(.|x_{t,1})$
  4. Optimizing the UCB acquisition function to select the next point $x_{t,b}$
  5. Repeat the above steps (2 and 3) to find $x_{t,b+1}$.

- Without using the outcome y, by updating the variance function by inserting $x_{t,1}$, BUCB has suppressed the variance around $x_{t,1}$.

- The property of the variance in a Gaussian process used in BUCB:
  - the variance at the observed location is zero or close to zero (for noise setting).
  - the variance depends on the location x (but not the outcome y)

Desautels et al. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. JMLR, 2014

The Effect of Reducing Uncertainty at Selected Point

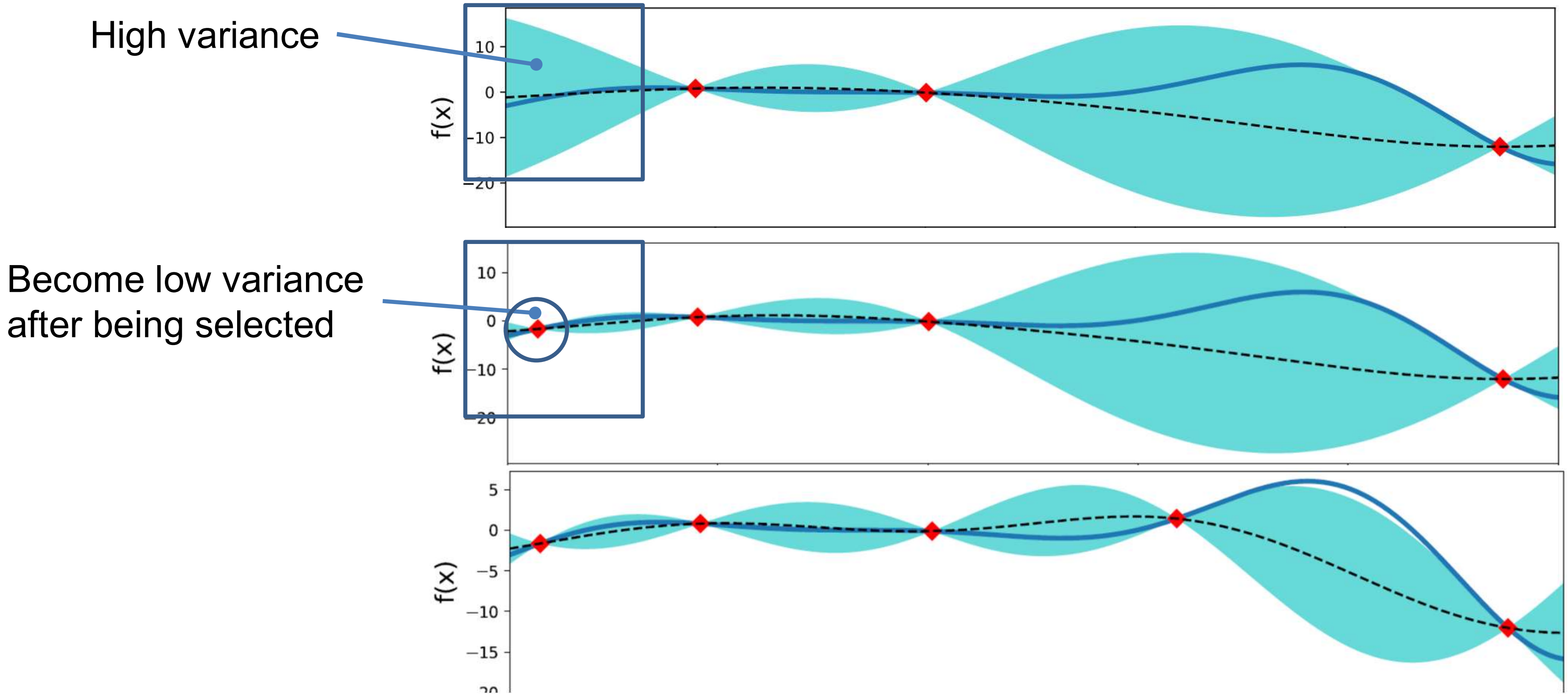


1. Get initial data
2. Fit a GP model to the data
3. Select x_{t+1}
4. Collect data $y_t = f(x_t)$

Repeat steps 2-4

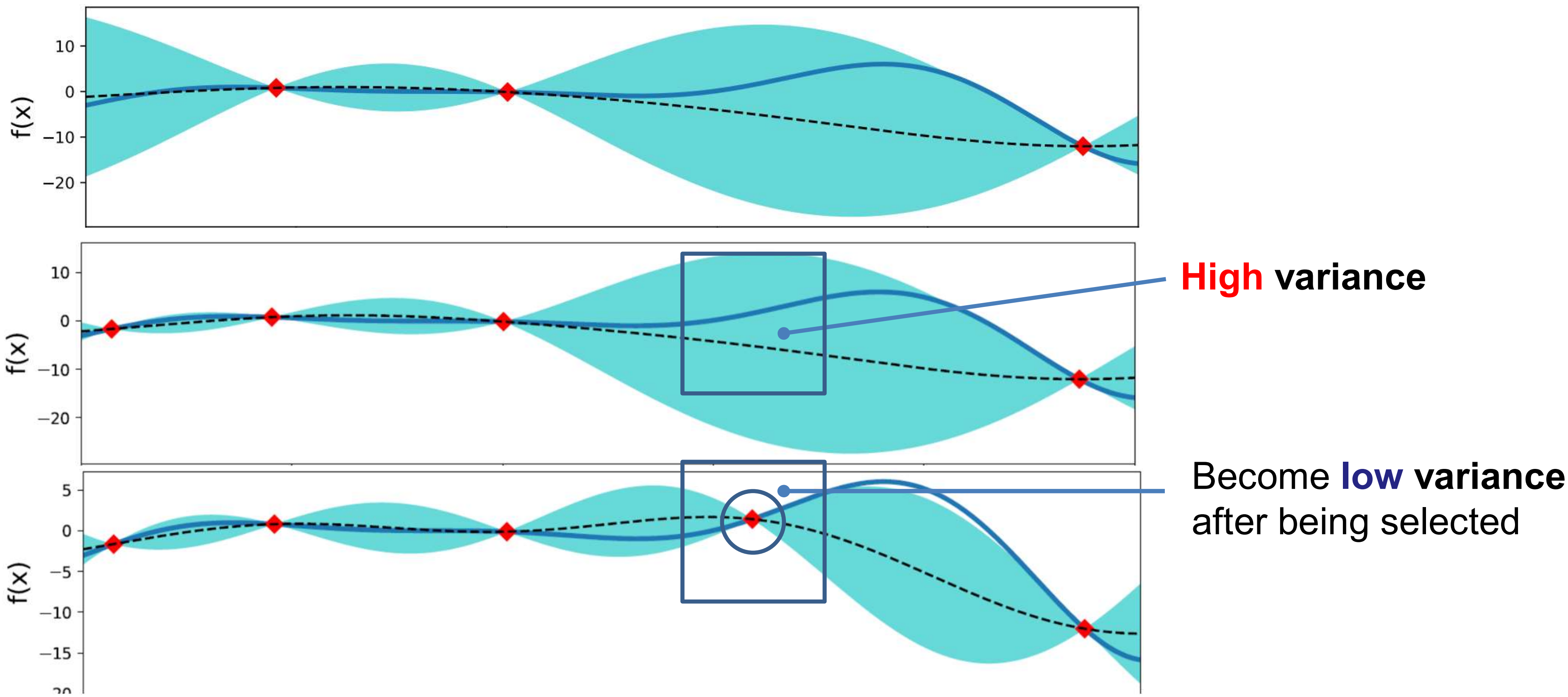
- Intuitively, BO selects a point which has high mean and high variance.

The Effect of Reducing Uncertainty at Selected Point



- The uncertainty at the selected points will be significantly reduced. The choice encodes the exploration.

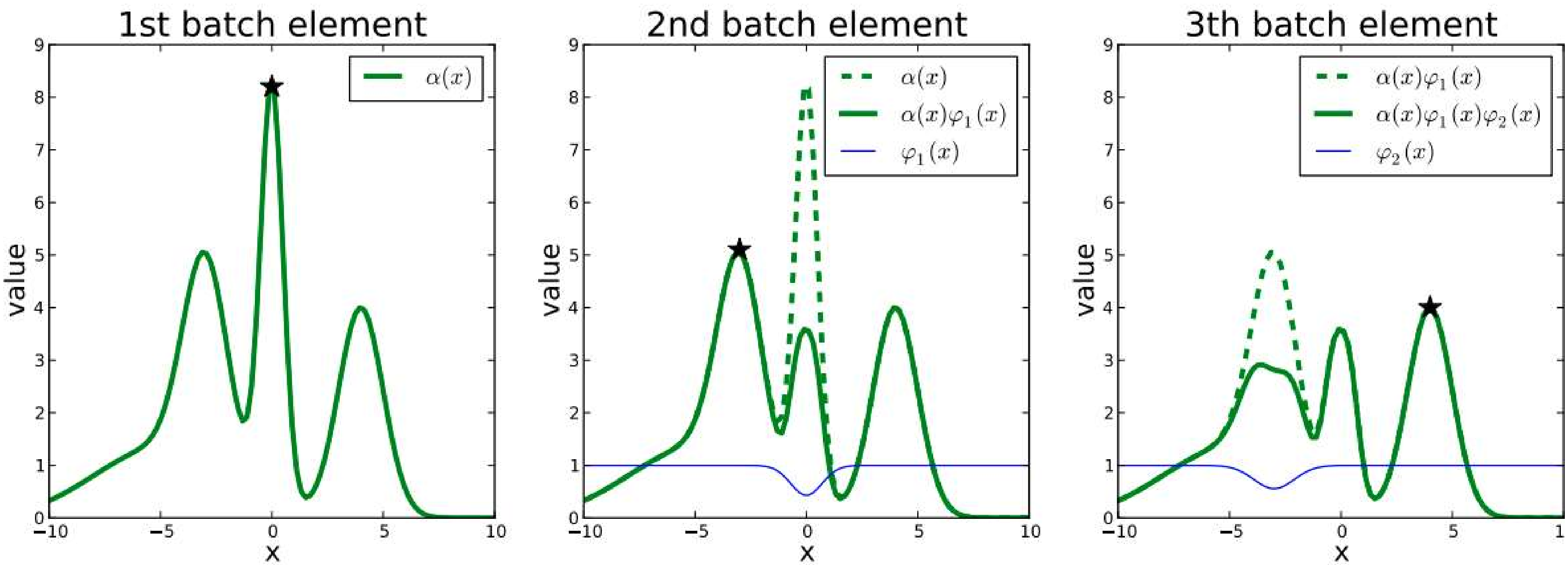
The Effect of Reducing Uncertainty at Selected Point



- The **uncertainty** at the selected points will be significantly reduced. Giving chance to explore at other locations

Peak Suppression – Local Penalization

- Illustration of peak suppression approach – Local Penalization



Gonzalez, J., et al. Batch bayesian optimization via local penalization. AISTATS 2016

Peak Suppression – Local Penalization

- Since the Lipschitz constant of the original function f is unknown.

$$L_{\nabla} = \max_{\mathbf{x} \in \mathcal{X}} \|\nabla f(\mathbf{x})\|$$

- We estimate Lipschitz Constant of the GP

$$\mu_{\nabla}(\mathbf{x}^*) = \partial \mathbf{K}_{n,*}(\mathbf{x}^*) \tilde{\mathbf{K}}_n^{-1} \mathbf{y}$$

where $(\partial \mathbf{K}_{n,*})_{i,l} = \frac{\partial \mathbf{k}_N(\mathbf{x}^*)}{\partial x^{(i)}}$

- Then, we choose $\hat{L}_{GP-LCA} = \max_{\mathcal{X}} \|\mu_{\nabla}(\mathbf{x}^*)\|$

Gonzalez, J., et al. Batch bayesian optimization via local penalization. AISTATS 2016

Peak Suppression – Local Penalization

- Define the penalized function $\varphi(x)$
  - indicates how much we want to exclude the nearby area around x.
  - smoothly reduce the value of the acquisition function in a neighborhood of x.

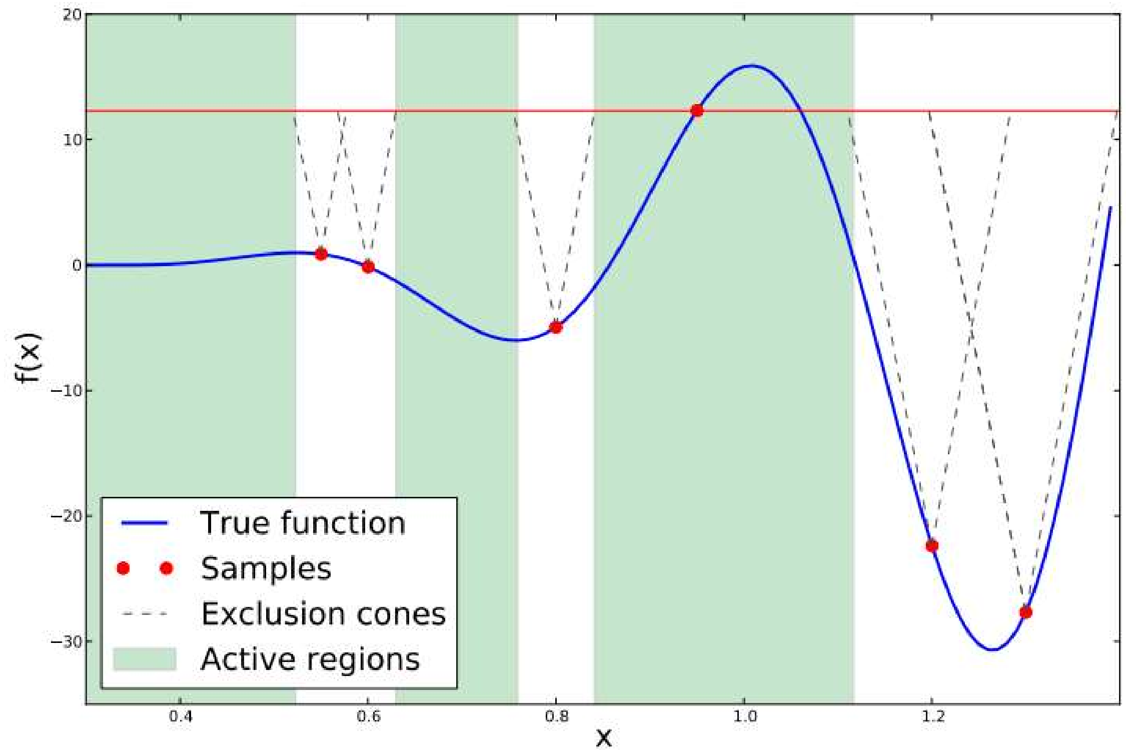
- Define a ball parameterized by a radius r_j

$$B_{r_j}(x_j) = \{x \in X : \left\| x_j - x \right\| \leq r_j\}$$

Gonzalez, J., et al. Batch bayesian optimization via local penalization. AISTATS 2016

Peak Suppression – Local Penalization

- Define a ball parameterized by a radius r_j where $r_j = \frac{M - f(x_j)}{L}$, M is the current best value $\max_i\{y_i\}$ and L is a Lipschitz constant.
- Intuition: the radius is large for the low value region where $f(x_j)$ is small and vice versa.



Drawback of Local Penalization

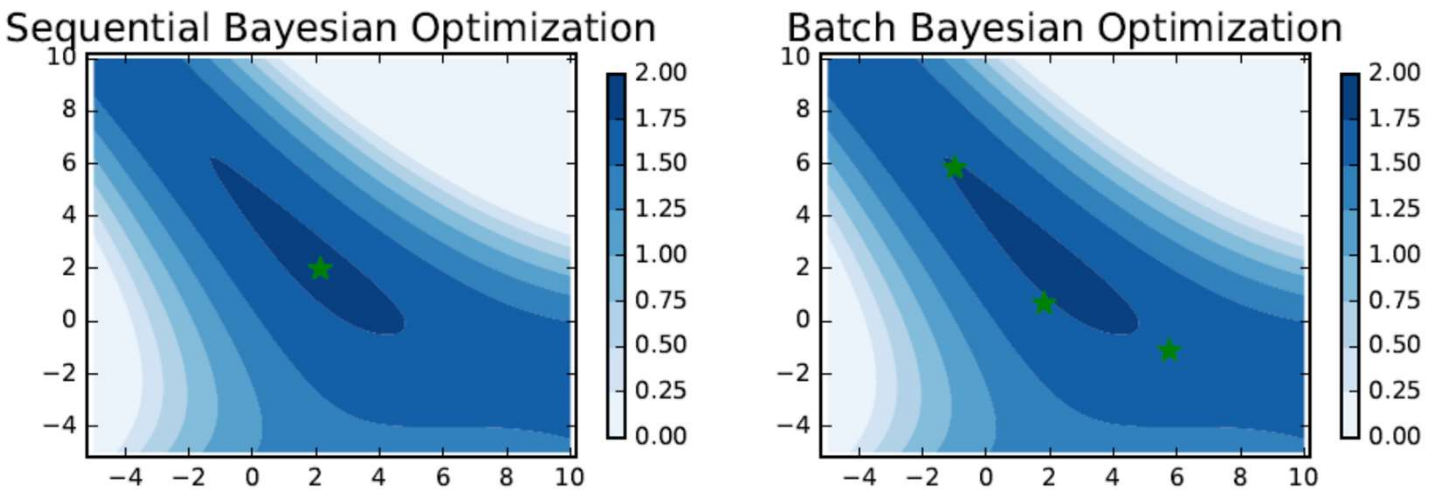
- They use a unique value of the Lipschitz constant L to represent for the whole function.
  - Some problems may not satisfy this condition, e.g. heteroscedastic functions.

- Estimating L in high dimension is still non-trivial.

Outline Part II.1: Batch Bayesian Optimization

- Introduction and Problem Statement
- Peak Suppression Approaches
  - Batch Upper Confidence Bound (GP-BUCB)
  - Local Penalization
- Budgeted Batch Bayesian Optimization
- Thompson Sampling for Batch Bayes Opt

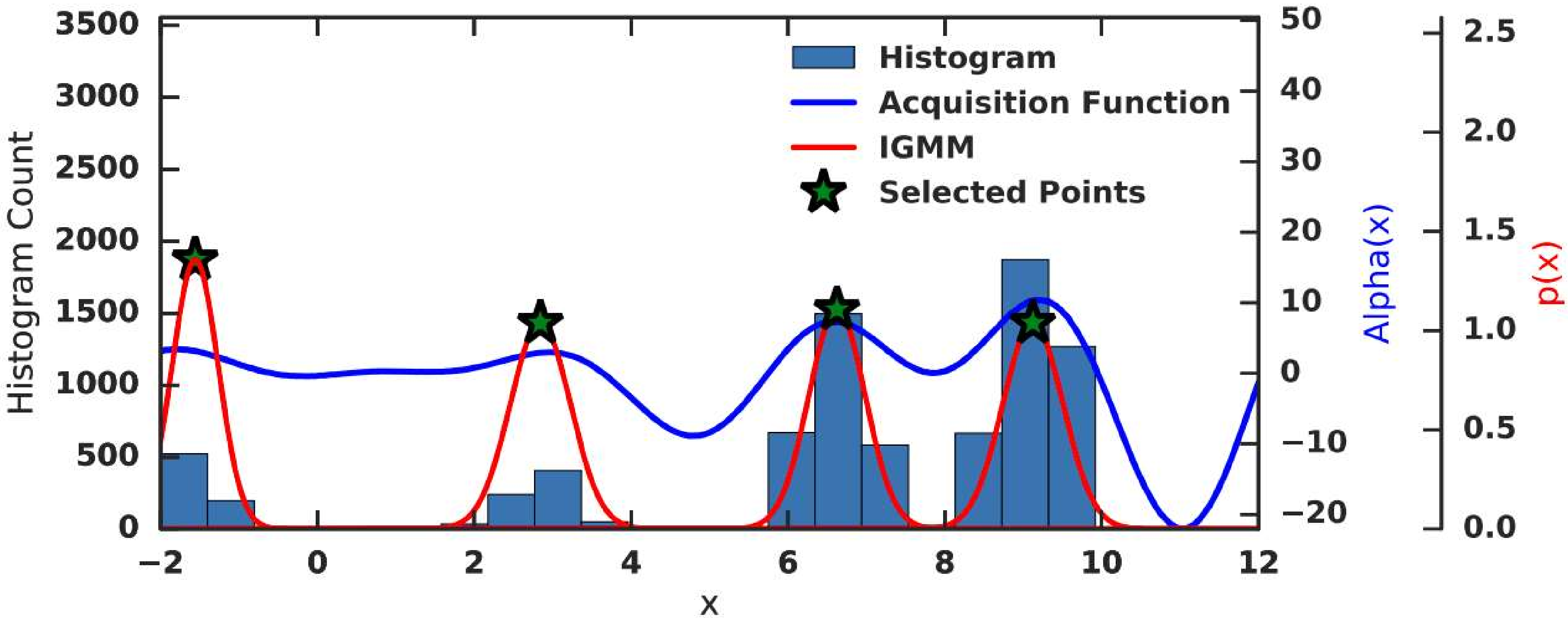
Batch Size in Batch Bayesian Optimization



- Existing approaches for batch BO require a fixed batch size of points:
  - Over-specify: wasting time and resources to evaluate redundant points.
  - Under-specify: missing important points that affect the performance.
- We aim to save the number of evaluations, but preserve the performance, by controlling the batch size in a principled way.

Acquisition Function as Multi-modal Functions

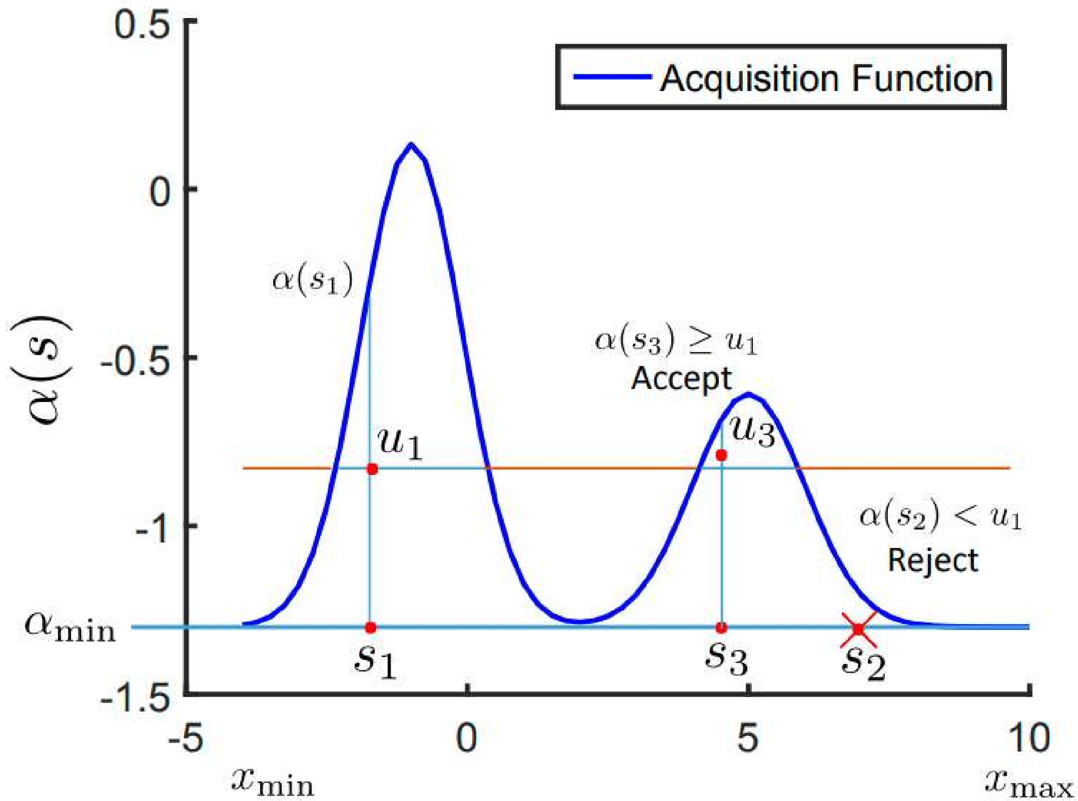
- The acq function is multi-modal with unknown number of peaks.
- Identifying the peaks (star) is equivalent to finding the mean locations in the infinite mixture of Gaussian (IGMM) (Red curve).



Vu Nguyen et al. Budgeted batch Bayesian optimization. In *ICDM 2016*,

Budgeted Batch Bayesian Optimization

- We propose the Generalized Batch Slice Sampling to draw samples under the acquisition function to learn the IGMM.

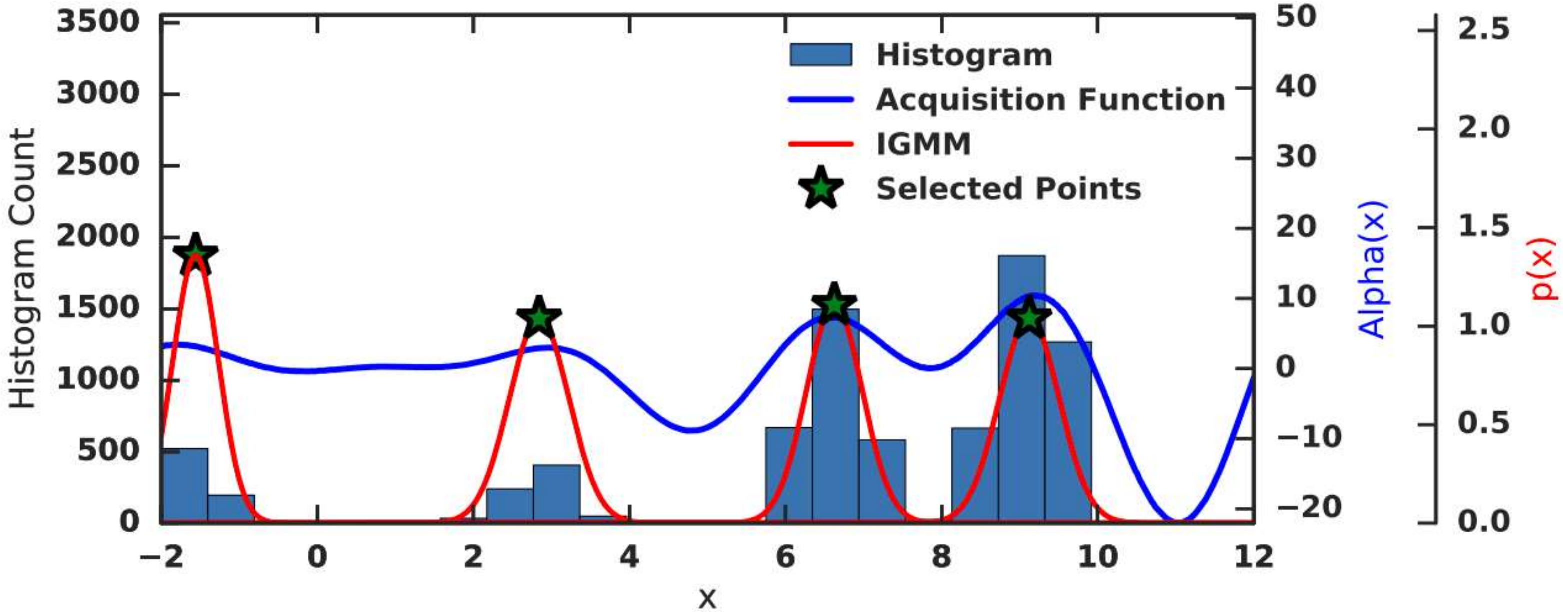


Vu Nguyen et al. Budgeted batch Bayesian optimization. In *ICDM 2016*,

Budgeted Batch Bayesian Optimization

1. Using Generalized Batch Slice Sampling to draw samples under the acquisition function (see the Histogram).
2. Fit the samples to the Infinite Gaussian Mixture Model.

- IGMM can detect the unknown number of peaks.



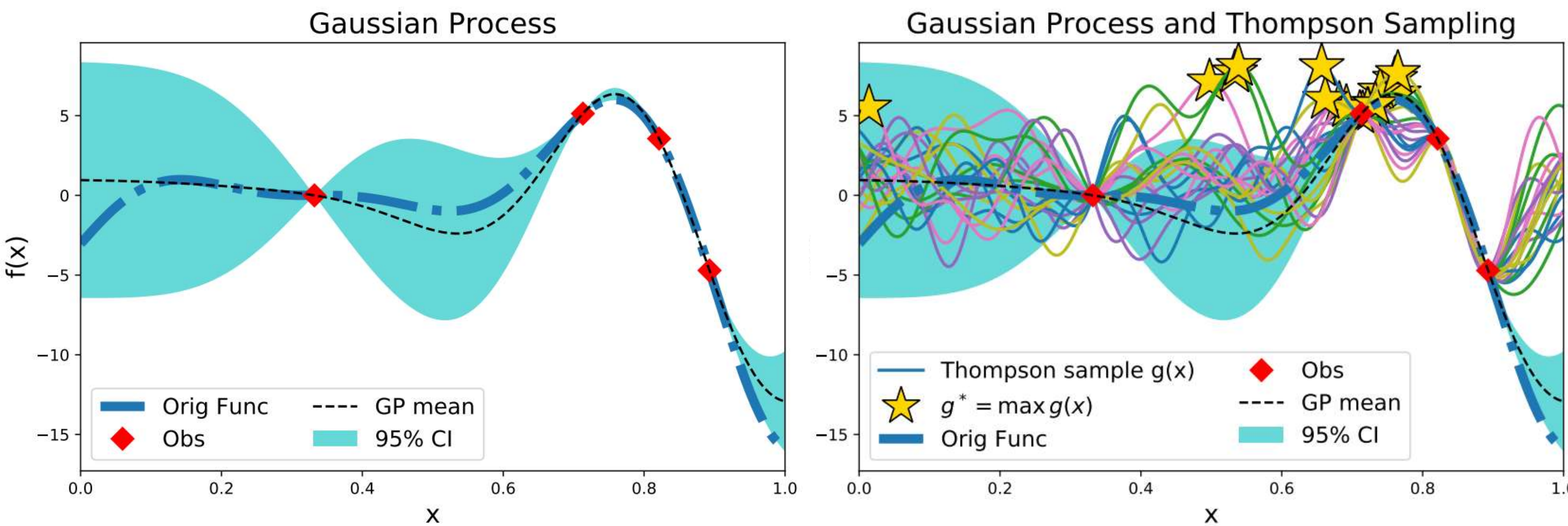
Vu Nguyen et al. Budgeted batch Bayesian optimization. In *ICDM 2016*,

Outline Part II.1: Batch Bayesian Optimization

- Introduction and Problem Statement
- Peak Suppression Approaches
  - Batch Upper Confidence Bound (GP-BUCB)
  - Local Penalization
- Budgeted Batch Bayesian Optimization
- Thompson Sampling for Batch Bayes Opt

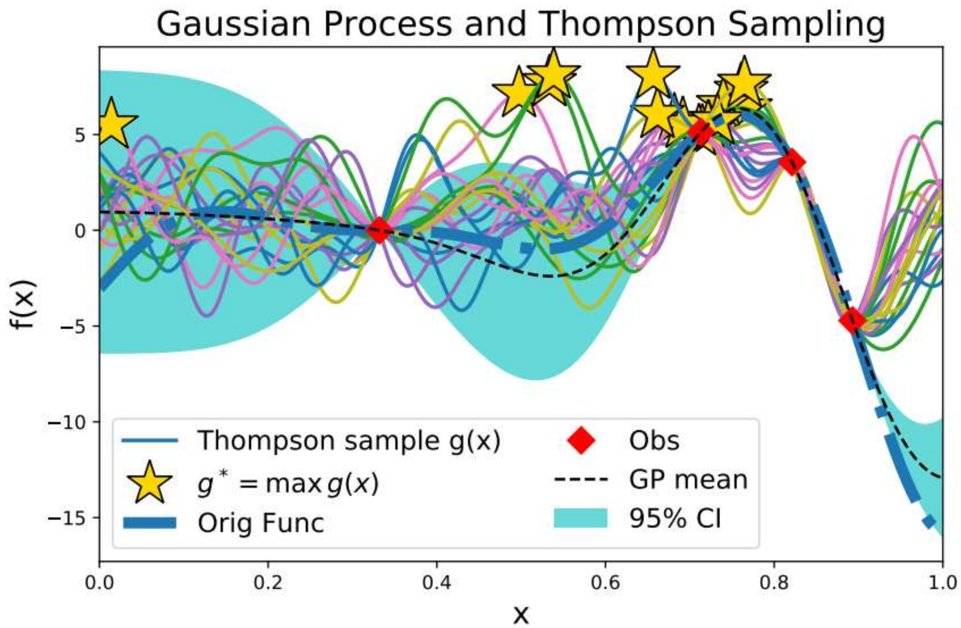
Thompson Sampling to Sample the Optimum Locations

- Thompson Sampling draws samples $g()$ from GP.
- Each yellow stars x^* is the maximizer of the sampled function $g()$
- We consider x^* as the perceived optimal samples



Thompson Sampling for Large Scale Batch BO

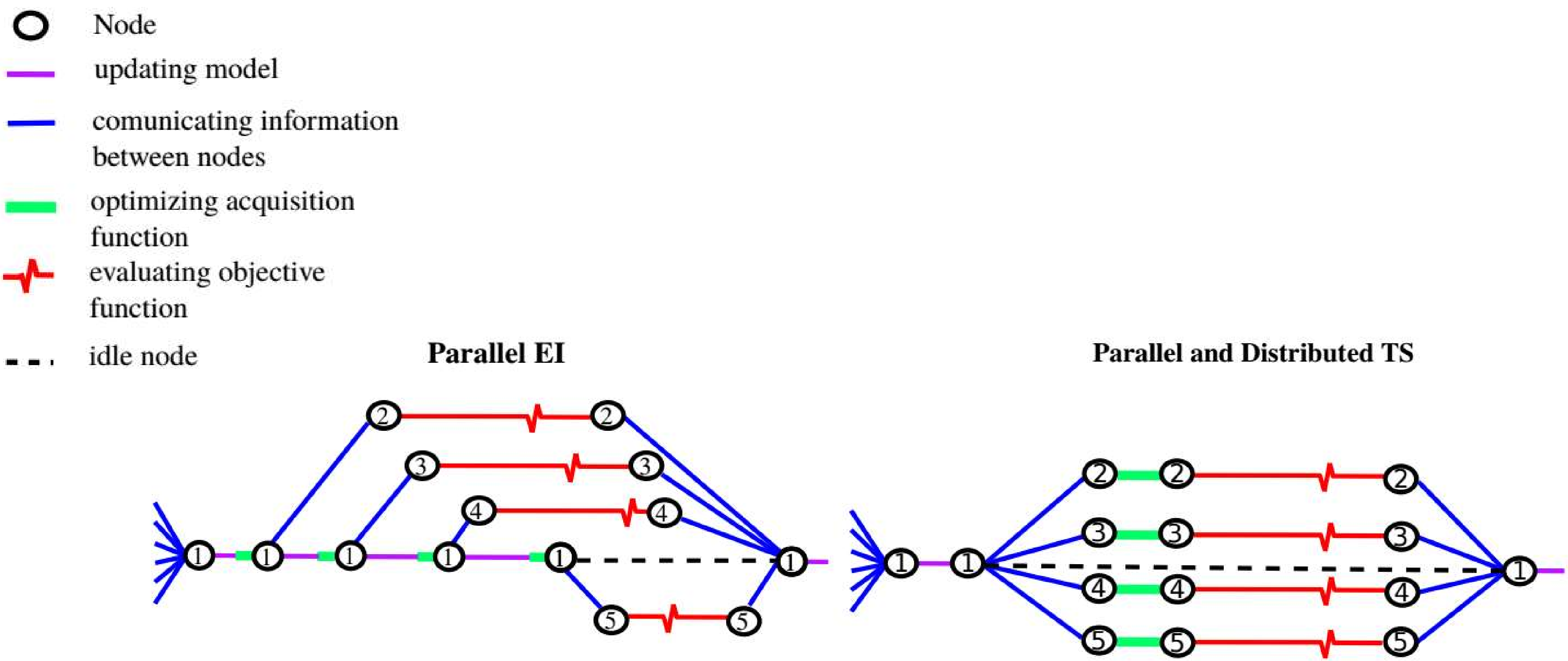
- For large batch size (e.g., $B \geq 100$), the penalization-based batch approaches may not be scalable.
- We can use different Thompson samples from a GP to draw multiple suggestions.



Hernández-Lobato, J. M., et al. ICML 2017

Thompson Sampling for Distributed BO

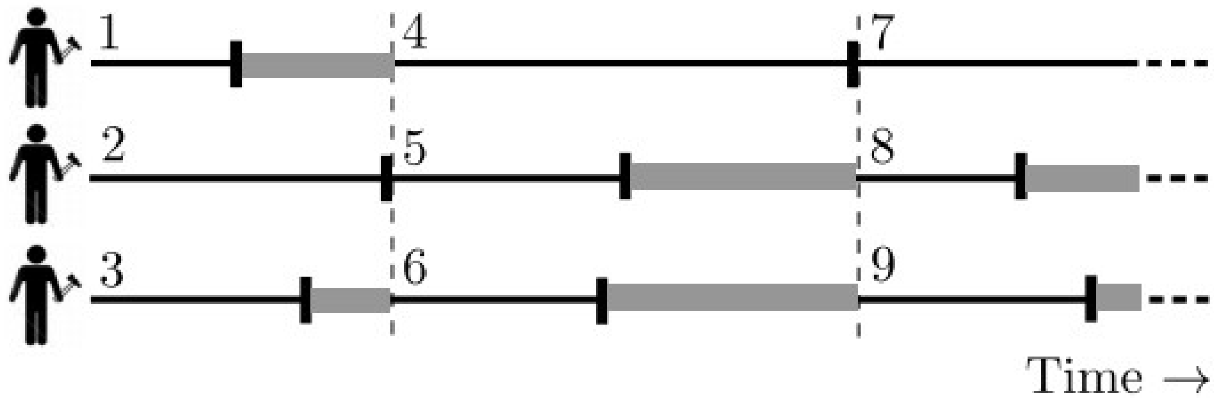
- Thompson Sampling for Parallel and Distributed BO



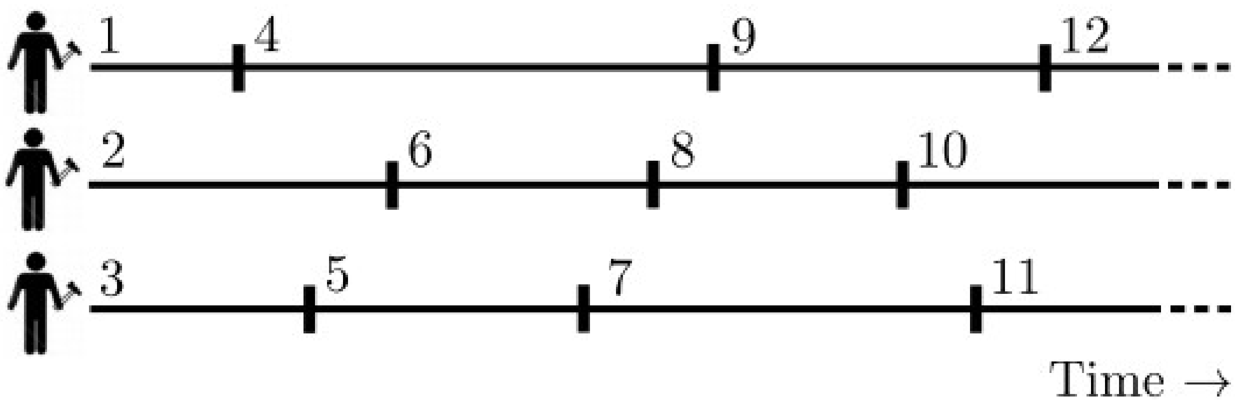
Hernández-Lobato, J. M., et al. "Parallel and Distributed Thompson Sampling for Large-scale Accelerated Exploration of Chemical Space.“ ICML 2017

TS for Asynchronous Optimisation

- Synchronous optimization



- Asynchronous optimization

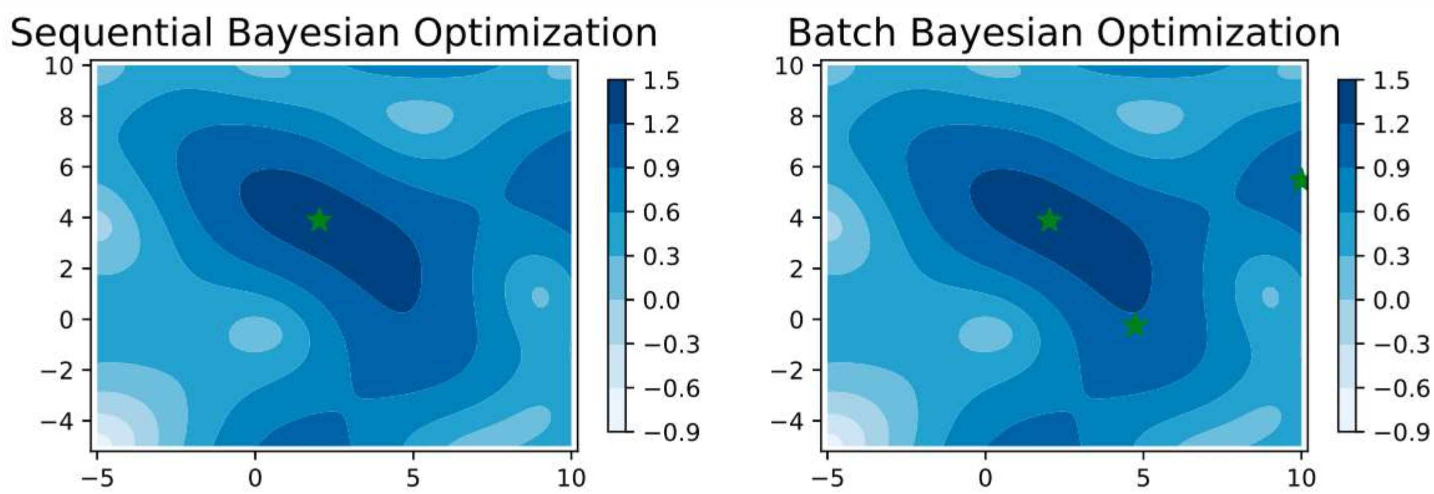


Kandasamy, K., et al,. Parallelised Bayesian Optimisation via Thompson Sampling. AISTATS, 2018.

Short Summary

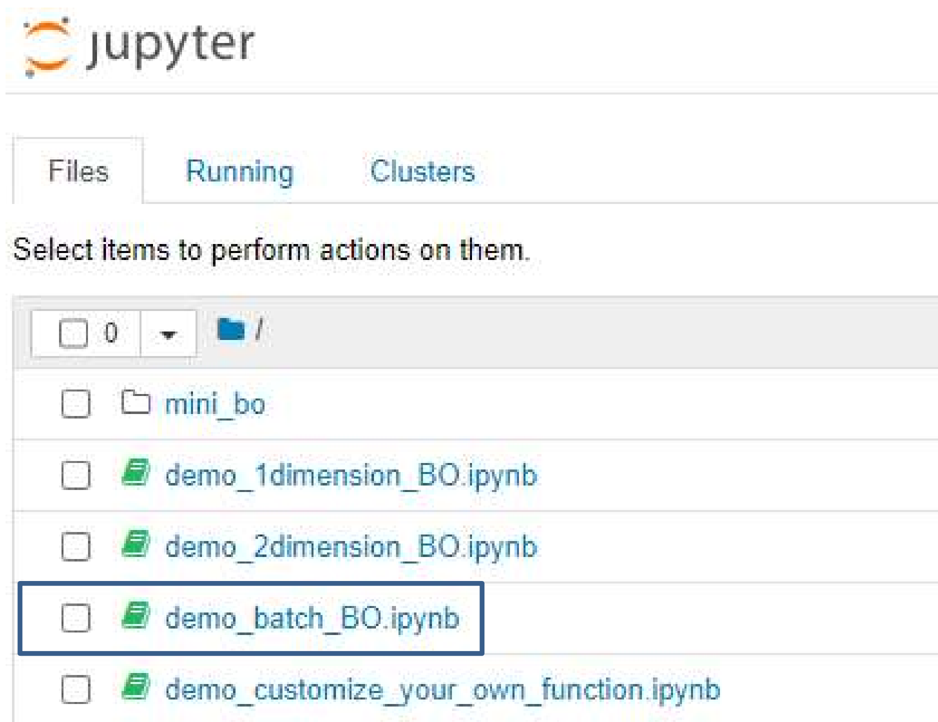
- Standard Bayes opt suggests one experiment to test at each iteration.

- Batch Bayes opt is desired to suggest multiple experiments and thus boost the optimization more efficient.



MiniBayesOpt for Batch Bayesian Optimization

- Code: vu-nguyen.org/BOTutorial_ACML20
- Github repository: MiniBayesOpt
- git+https://github.com/ntienvu/minibo



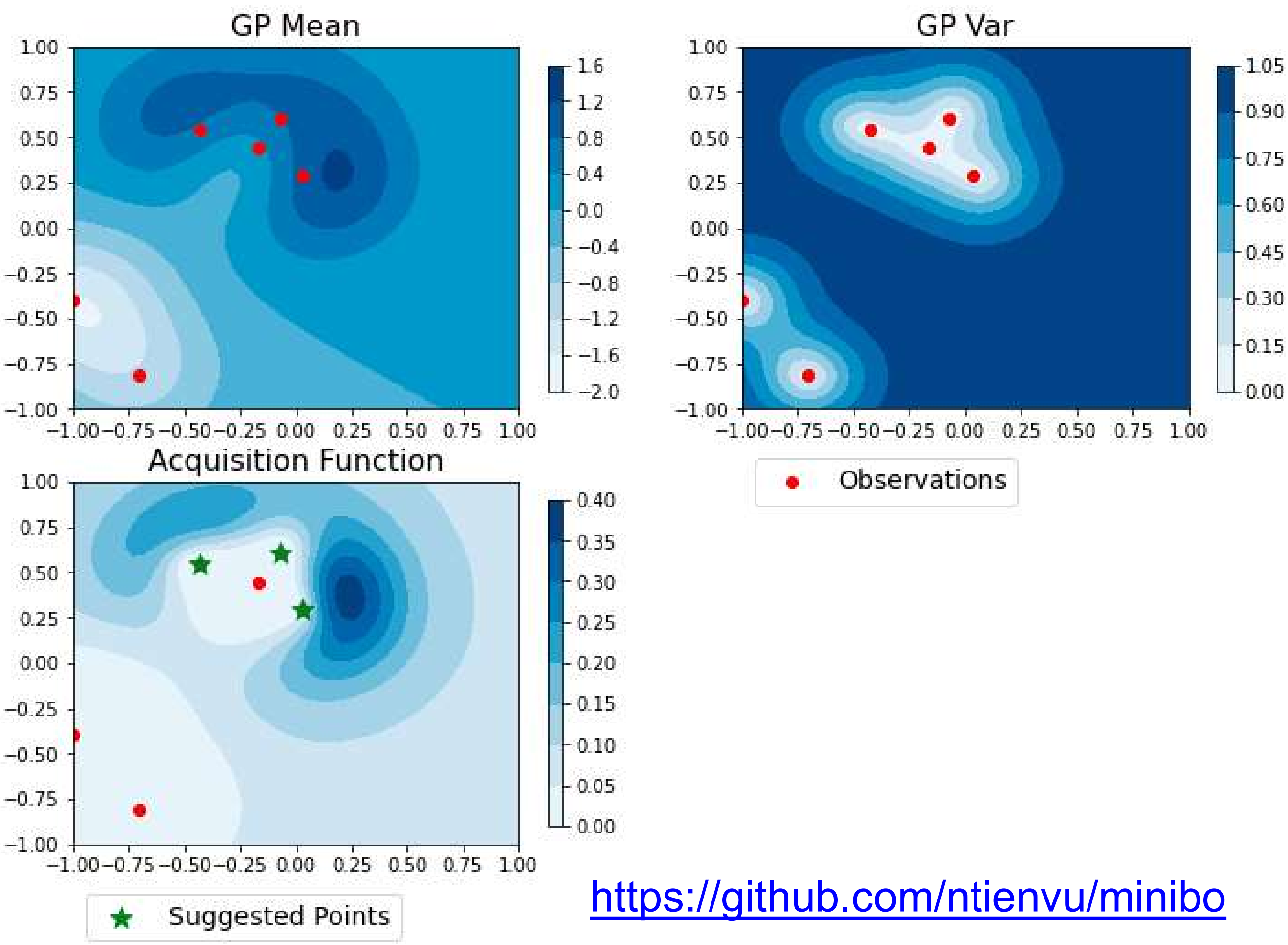
https://github.com/ntienvu/minibo

Using MiniBO for batch Bayes Opt

Select a batch of points B=3 (green stars)

```
In [5]: xt=bo.select_next_point(B=3)
        print("the next point is \n",xt)
        visualization.plot_bo_2d(bo)
```

Specifying $B \geq 1$



https://github.com/ntienvu/minibo

Agenda

- Parameter Tuning as Black-Box Function
- Part I: Bayesian Optimization
- Part II: Recent Advances in Bayesian Optimization
  - Batch Bayesian Optimization
  - High dimensional Bayes Opt
  - Mixed Categorical-Continuous Bayes Opt
- Research Directions in Bayesian Optimization

Challenges for High-dimensional Bayes Opt

- Standard BO often works on less than 10 dimensions.
- High dimension causes problem for optimization – the search space grows exponentially with the dimension.

High-dimensional Bayesian Optimization

- Optimize high-dimensional acquisition functions?

  Difficult

  - High-dimensional (d>10) acquisition functions feature only a few peaks and a large terrain of almost flat surface [Rana et al ICML 2017]

  - Global optimizers (DIRECT) fail to return an optimum within limited time and resource;

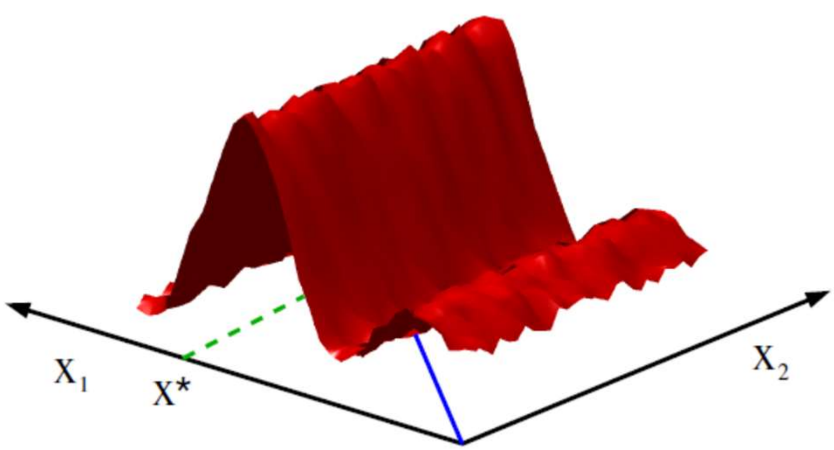
  - Gradient-dependent Local optimizers get stuck due to non-significant gradients in the flat surface of acquisition functions;

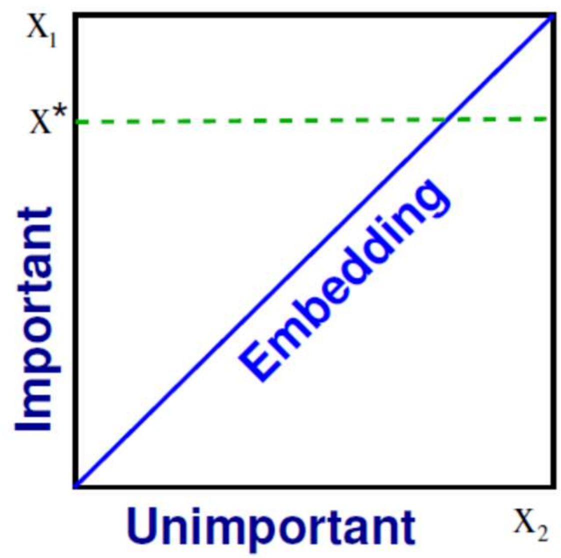
High dimensional Bayes Opt

- Random Embedding: Wang et al IJCAI 2013
- Additive GP: Kandasamy et al ICML 2015
- Drop-out: Li et al. IJCAI 2017
- TurBO: Eriksson et al NeurIPS 2019

Low-dimensional Embedding

- Project high-dimensional space to low-dimensional space
  - This 2D function only has d=1 effective dimension.
  - It is more efficient to search for the optimum along the 1-dimensional random embedding than in the original 2-dimensional space.

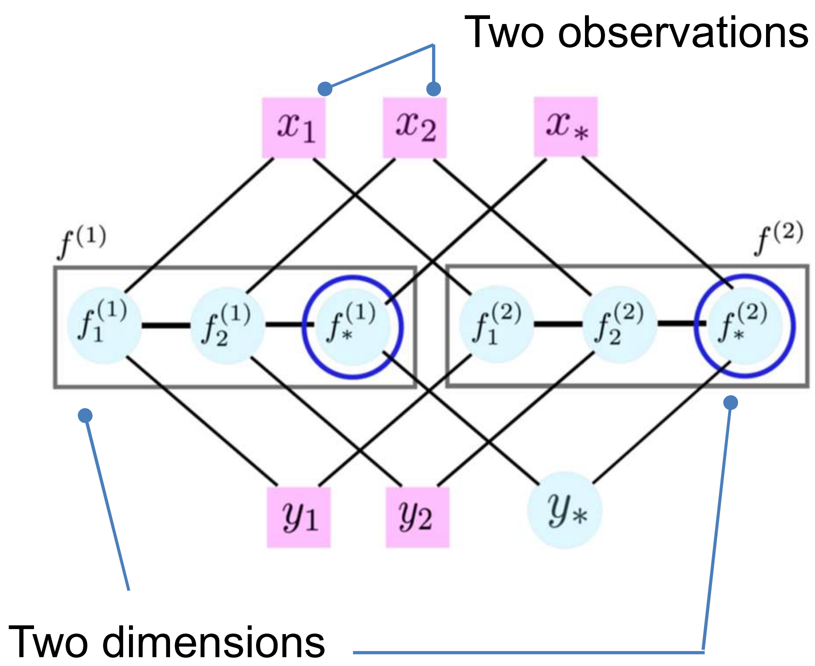




Wang Z et al. Bayesian optimization in high dimensions via random embeddings. IJCAI, 2013.

Additive Gaussian Process

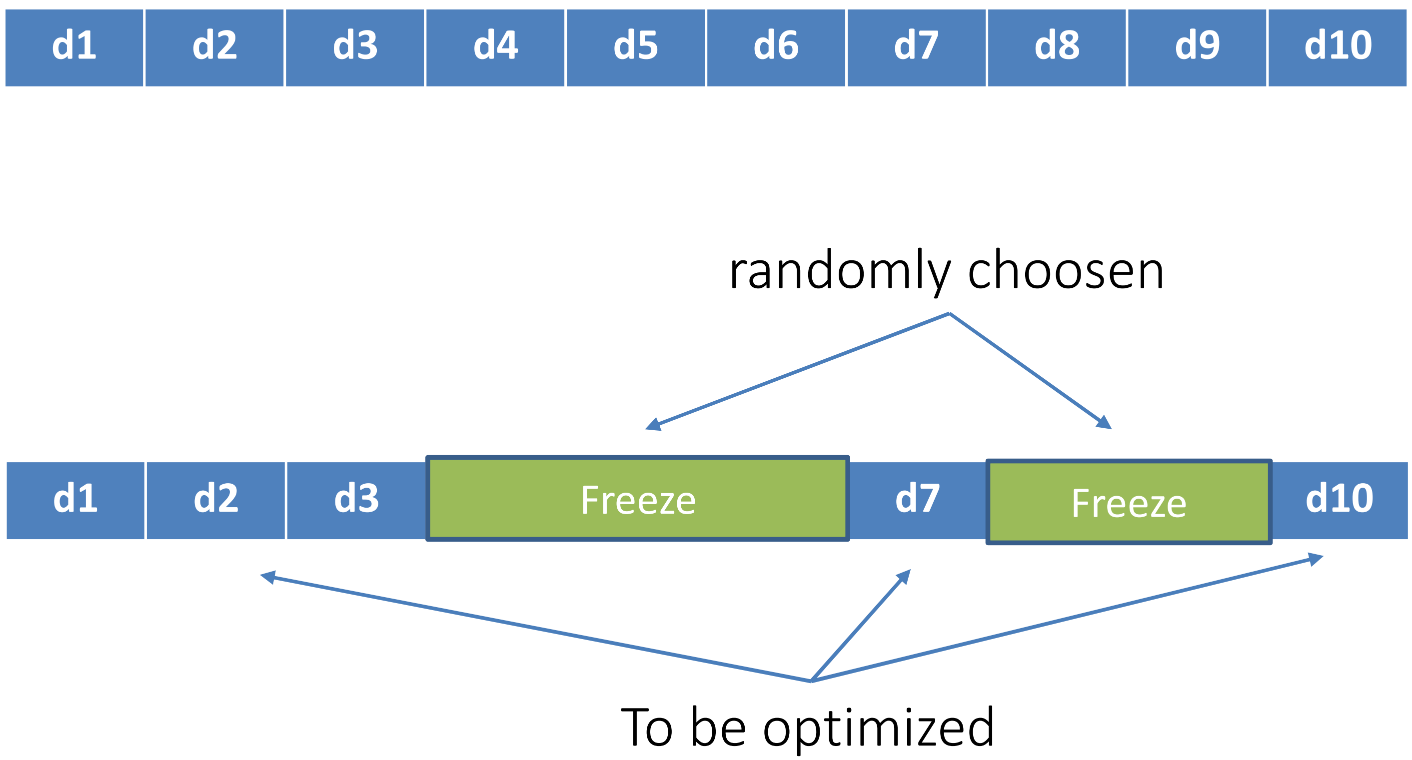
- Decompose high dimensions into disjointed groups
- Infer the individual GP posterior for each disjointed group.
- Define the additive acquisition function.



Kandasamy, K et al. High dimensional Bayesian optimisation and bandits via additive models. ICML, 2015.

High-dimensional Bayes Opt via Drop-out

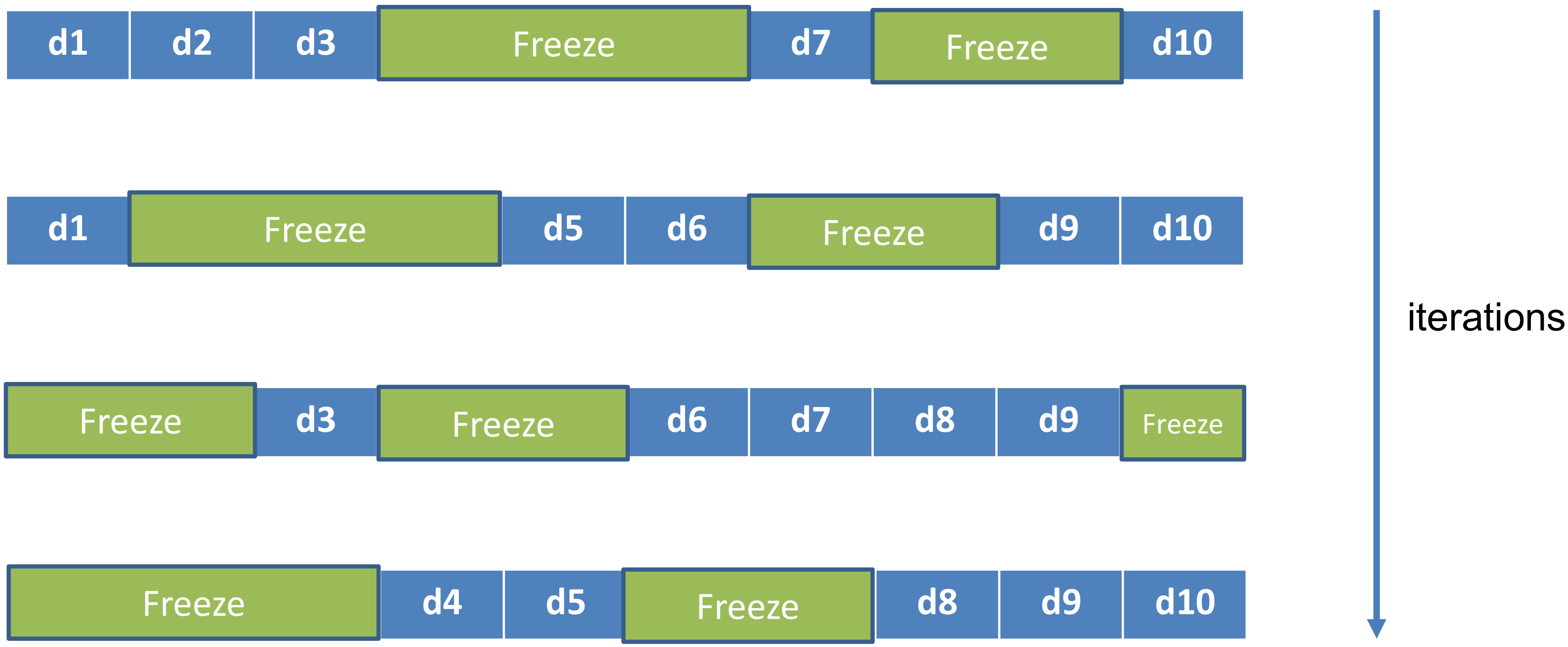
- Randomly select a smaller number of dimension for optimization.



Li C et al. High Dimensional BO Using Dropout. IJCAI, 2017

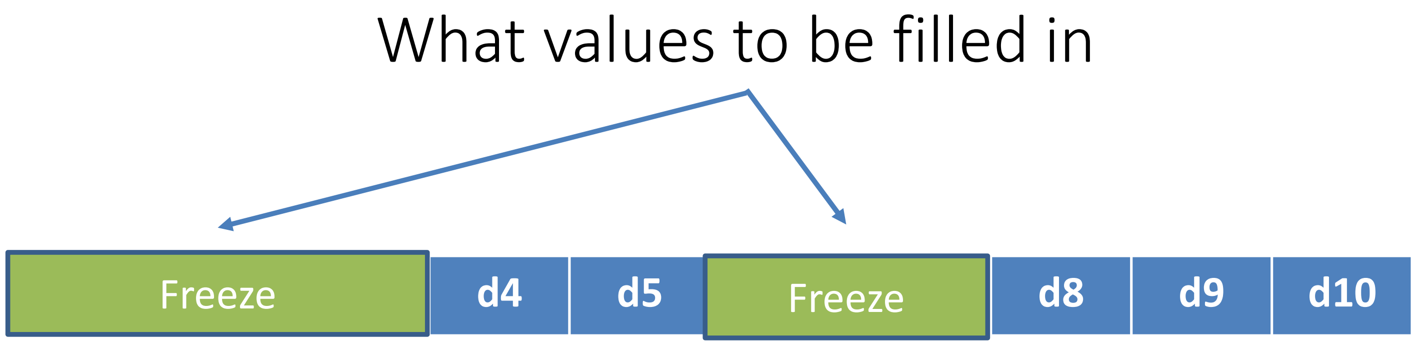
High-dimensional Bayes Opt via Drop-out

- Randomly select a smaller number of dimension for optimization.



- The number of effective dimensions to be optimized is small.

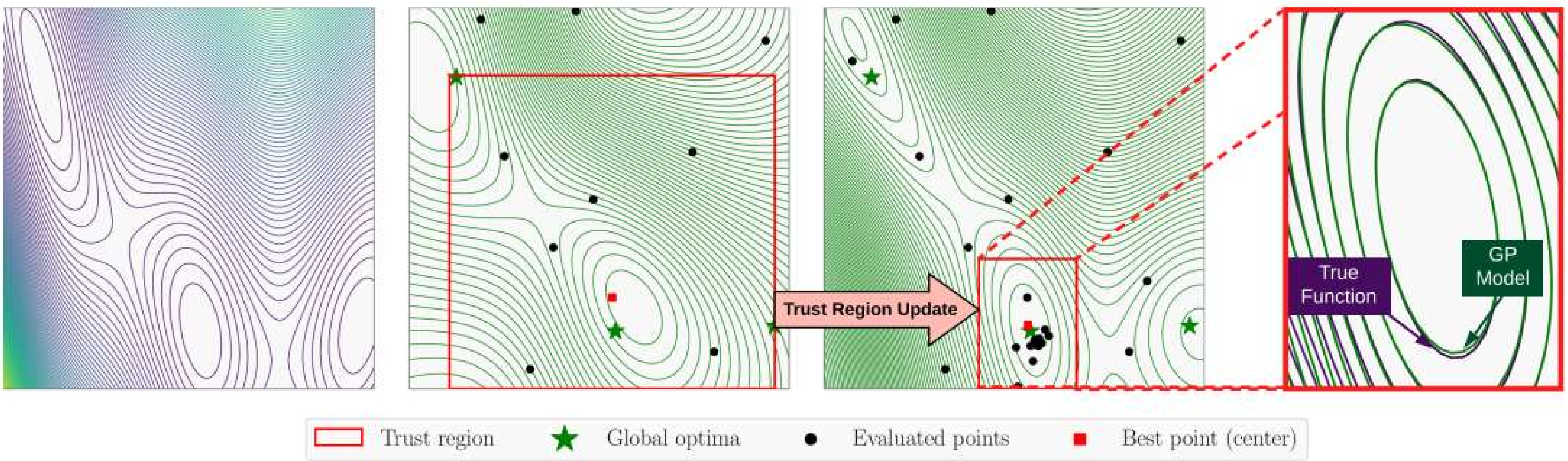
Fill-in strategies



- Dropout-Random: use a random value in the domain
- Dropout-Copy: copy the value of the best-performing variables.
- Dropout-Mix: Random with a probability p and Copy with a probability $1-p$

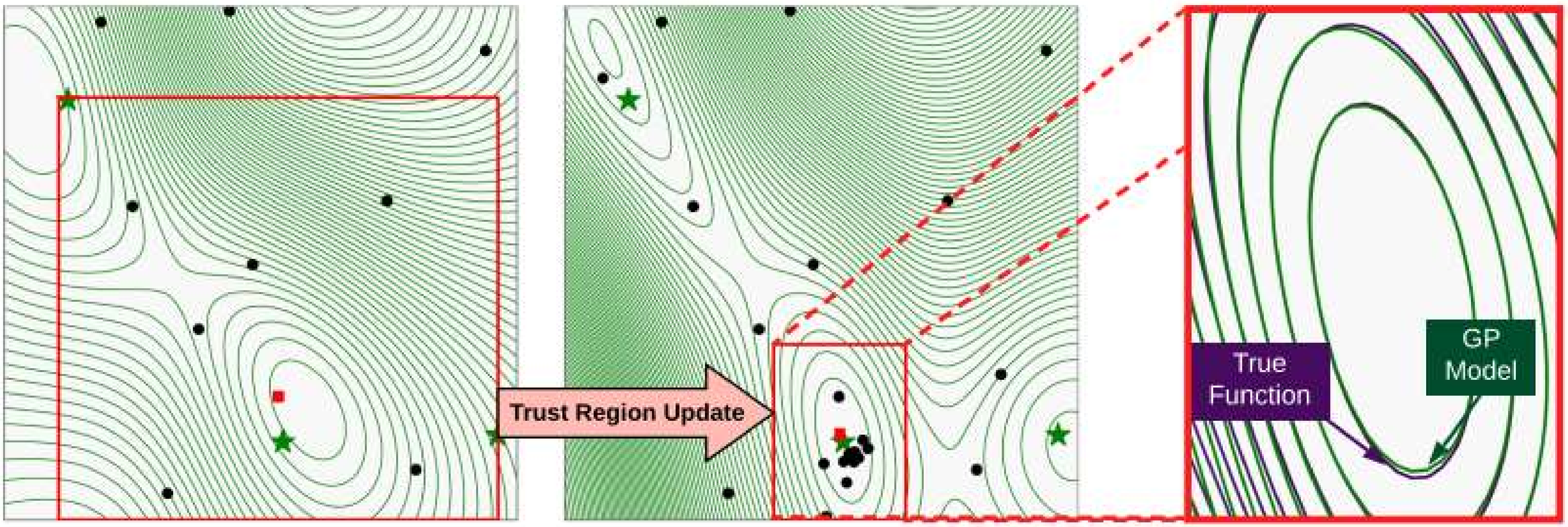
Trust Region Bayesian Optimization (TurBO)

- High-level idea:
  - Build the trust regions
  - Perform local optimization in each trust region
  - Repeat



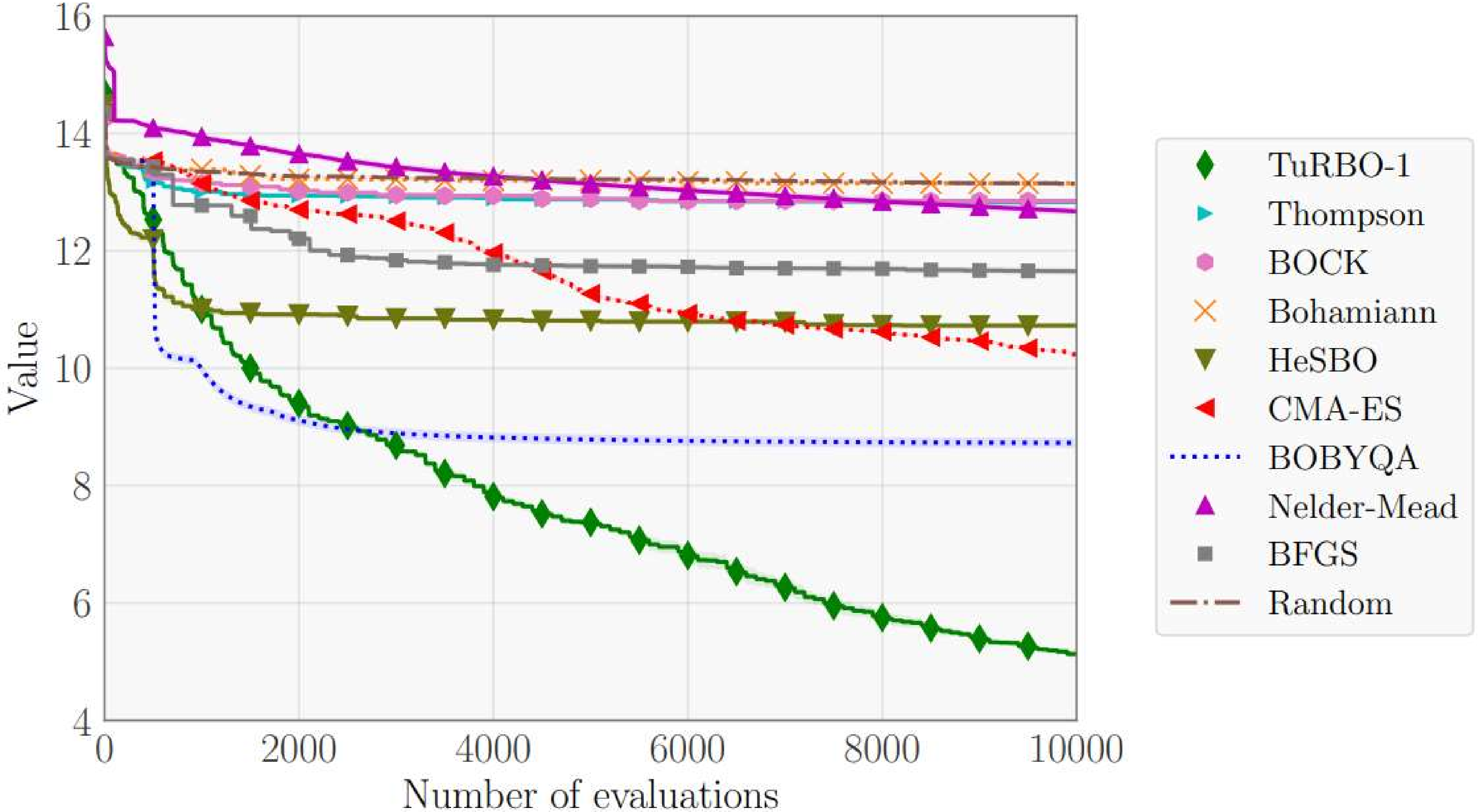
Eriksson, David, et al. "Scalable global optimization via local bayesian optimization." *NeurIPS*. 2019.

TuRBO



- Eriksson, David, et al. "Scalable global optimization via local bayesian optimization." *NeurIPS*. 2019.

TurBO in optimizing 200 dimension



Short Summary

- Bayesian optimization can work effectively up to 10 dimensions.
- In real-world scenarios, we may tackle the problems with large number of dimensions.
- Bayesian optimization research in high dimension is essential.

Agenda

- Hyperparameter Tuning and Experimental Design as Black-Boxes
- Part I: Bayesian Optimization
- Part II: Recent Advances in Bayesian Optimization
  - Batch Bayesian Optimization
  - High dimensional Bayes Opt
  - Mixed Categorical-Continuous Bayes Opt
- Research Directions in Bayesian Optimization

Bayes Opt Mixed Categorical – Continuous Input

- Tuning hyperparameters for deep neural network

continuous variables | **categorical variables**

- $y = f(\ x_1\ ,\ x_2\ ,\ x_3\ ,\ x_4\ )$

learning rate $\in [1e^{-6}, 1e^{-1}]$

weight decay $\in [1e^{-6}, 1e^{-1}]$

optimiser type $\in \{SGD, Adam, \ldots\}$

activation type $\in \{tanh, sigmoid, \ldots\}$

- Multiple categorical - each categorical has multiple options

Bayes Opt Mixed Categorical – Continuous Input

- Tuning hyperparameters for support vector machine

continuous variables | **categorical variables**

- $y = f(\; x_1 \;,\; x_2 \;|\; x_3 \;,\; x_4 \;)$

Penalty parameter $\in [0,10]$

Kernel parameter $\in [1e^{-6}, 1e^{-1}]$

kernel type $\in \{RBF, Poly, \dots\}$

Kernel coefficient $\in \{scale, auto, \dots\}$

- Multiple categorical - each categorical has multiple options

Bayes Opt Mixed Categorical – Continuous Input

- Tuning hyperparameters for XGBoost

continuous variables | **categorical variables**

- $y = f(\ x_1\ ,\ x_2\ ,\ x_3\ \ x_4\ ,\ x_5\ )$

learning rate $\in [0,1]$

subsample $\in [1e^{-3}, 1]$

regularisation $\in [0,5]$

booster type $\in \{gbtree, dart\}$

grow policies $\in \{depthwise, loss\}$

- Multiple categorical - each categorical has multiple options

Bayes Opt Mixed Categorical – Continuous Input

- One-hot encoding:
  - Red: [1,0,0]   Green: [0,1,0]   Blue: [0,01]

- Drawbacks:
  - Make the search space large.

  if C =4 categories, each has $V = 5$ choices=> 20 extra dimensions.
  - Non-continuous and non-differentiable space

- Challenging in optimizing mixed-type: categorical - continuous

Bayes Opt Mixed Categorical – Continuous Input

continuous | **categorical**

- $y = f(\underbrace{x_1, x_2}_{\text{continuous}}, \underbrace{x_3, x_4}_{\text{categorical}})$

learning rate $\in [1e^{-6}, 1e^{-1}]$

weight decay $\in [1e^{-6}, 1e^{-1}]$

optimiser type $\in \{SGD, Adam, \dots\}$

activation type $\in \{tanh, sigmoid, \dots\}$

Bayes Opt

Multi-armed Bandit

Agenda

- Hyperparameter Tuning and Experimental Design as Black-Boxes
- Part I: Bayesian Optimization
- Part II: Recent Advances in Bayesian Optimization
    - Batch Bayesian Optimization
    - Bayesian Optimization in Unknown Search Space
    - Mixed Categorical-Continuous Bayes Opt
        - Problem setting
        - Multi-armed bandits
        - Categorical-specific continuous optimization
        - Categorical-(non-)specific continuous optimization
- Research Directions in Bayesian Optimization

Multi-Armed Bandit Setting

- A fixed set of arms, each of which returns a reward.
- Observe the reward before the next pull.
- The goal is to pull and receive as high reward as possible.

A

B

C

Bandit “arms”

General Algorithm for Multi-Armed Bandit

- Input is a fixed set of action $[C]$
- Initialize the model θ
- For t=1.....T
  - Compute the probability p_c of selecting arm c from θ
  - Select an arm $c \in [C]$ using the probability p_c
  - Pull an arm c
  - Observe the reward $g_t(c)$ at the arm c
  - Update the model θ using the reward

There involves randomness here

General Algorithm for Multi-Armed Bandit

- Input is a fixed set of action $[C]$
- Initialize the model θ
- For t=1.....T
  - Compute the probability p_c of selecting arm c from θ
  - Select an arm $c \in [C]$ using the probability p_c
  - Pull an arm c
  - Observe the reward $g_t(c)$ at the arm c
  - Update the model θ using the reward

These steps are the key of MAB

UCB1 algorithm

Exploration-Exploitation

- Input is a fixed set of action $[C]$
- Initialize the model θ
- For t=1.....T

utility score

uncertainty

  - Compute the probability $p_c = \overline{g_c} + \sqrt{\frac{2 \log T}{n_c}}$
  - Select an arm $c \in [C]$ by maximizing p_c
  - Pull an arm c
  - Observe the reward $g_t(c)$ at the arm c
  - Update the model θ

EXP3 Algorithm

- EXP3 algorithm doesnot assume the underlying distribution over the process generating the reward.

Algorithm 1 Exp3 Algorithm for Categorical Selection.

Input: $\gamma \in [0,1]$, C #categorical choice, T #max iteration

1: Init $\omega_c = 1, \forall c = 1...C$
2: **for** $t = 1$ to T **do**
3: Compute the probability $p_t^c = (1-\gamma)\frac{\omega_c}{\sum_{c=1}^{C}\omega_c} + \frac{\gamma}{C}, \forall c = 1...C$ *This step is different*
4: Choose a categorical variable $h_t \in [1, ...C]$ at random according to distribution p_t.
5: Observe the reward $g_t(c) = f(h_t = c)$
6: Normalize $\hat{g}_t(c) = \frac{g_t(c)}{p_t^c}$
7: Update the weight $\omega_c = \omega_c \times \exp\left(\gamma \hat{g}_t(c)/C\right)$ *This step is different*
8: **end for**

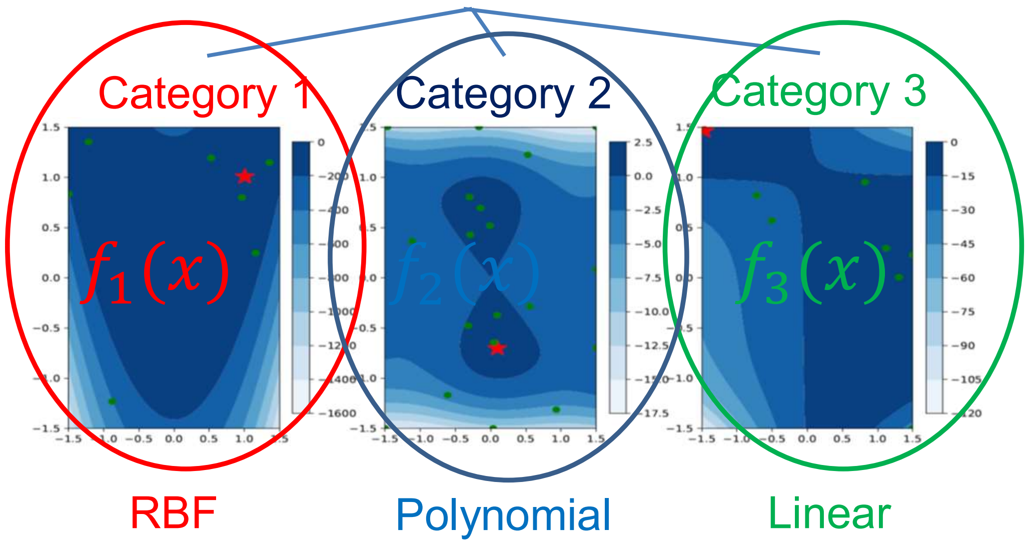
Output: $\mathcal{D}_T$

Auer P et al, “The non-stochastic multi-armed bandit problem” 2002

Two settings in mixed variables optimization

1. Continuous variable is specific to categorical variable

Kernel parameter is specific to the kernel type.

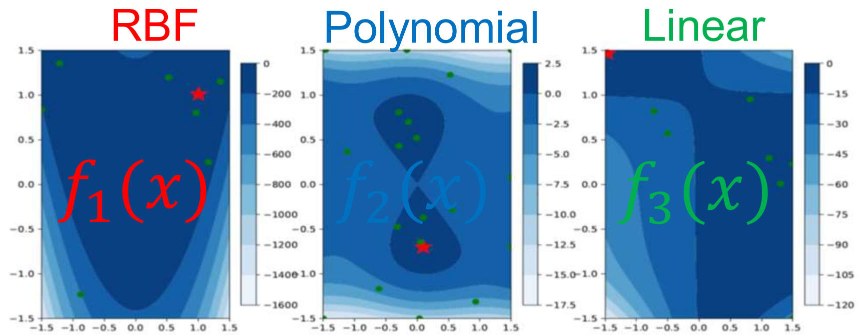


- Each categorical variable forms an independent function

$$f_c^* = \max_{x \epsilon X} f_c\,(x)$$

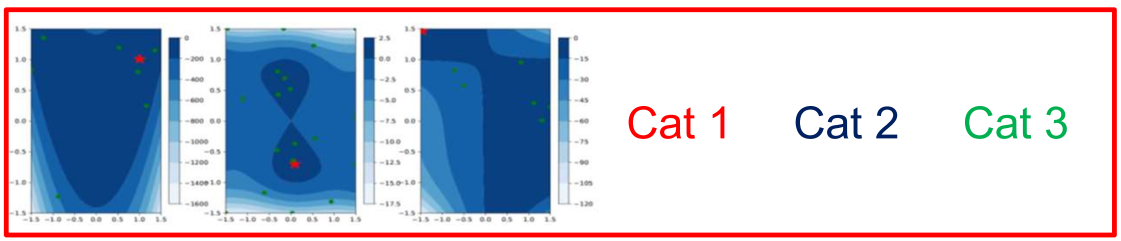
Two settings in mixed variables optimization

1. Continuous variable is specific to categorical variable



2. Continuous is not specific to categorical

Learning rate in DL is not necessarily specific to the activation type



continuous

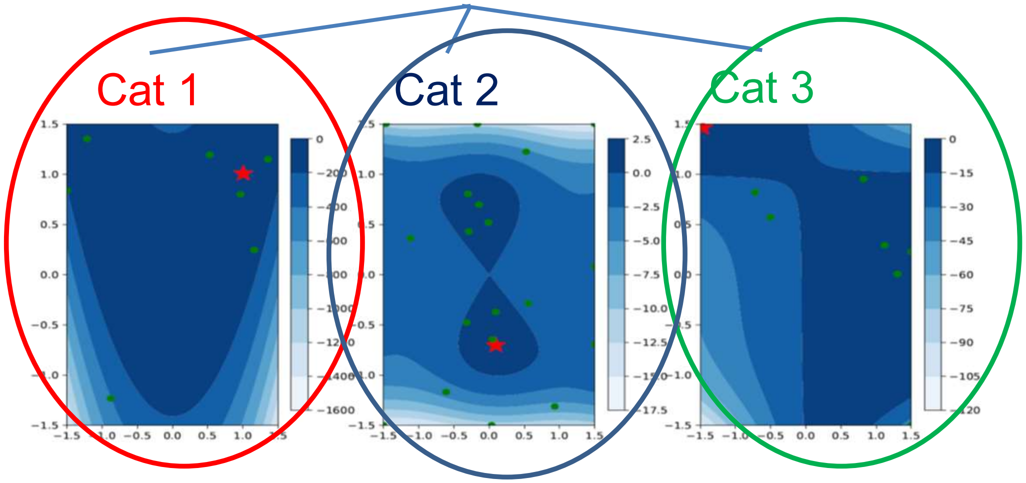
categorical

$$f(x, h)$$

continuous

Two settings in mixed variables optimization

1. Continuous variable is specific to categorical variable

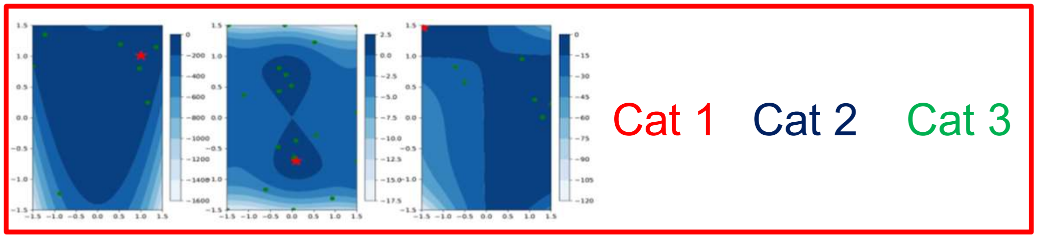


$f_1(x)$ $f_2(x)$ $f_3(x)$

- Three independent functions

Local

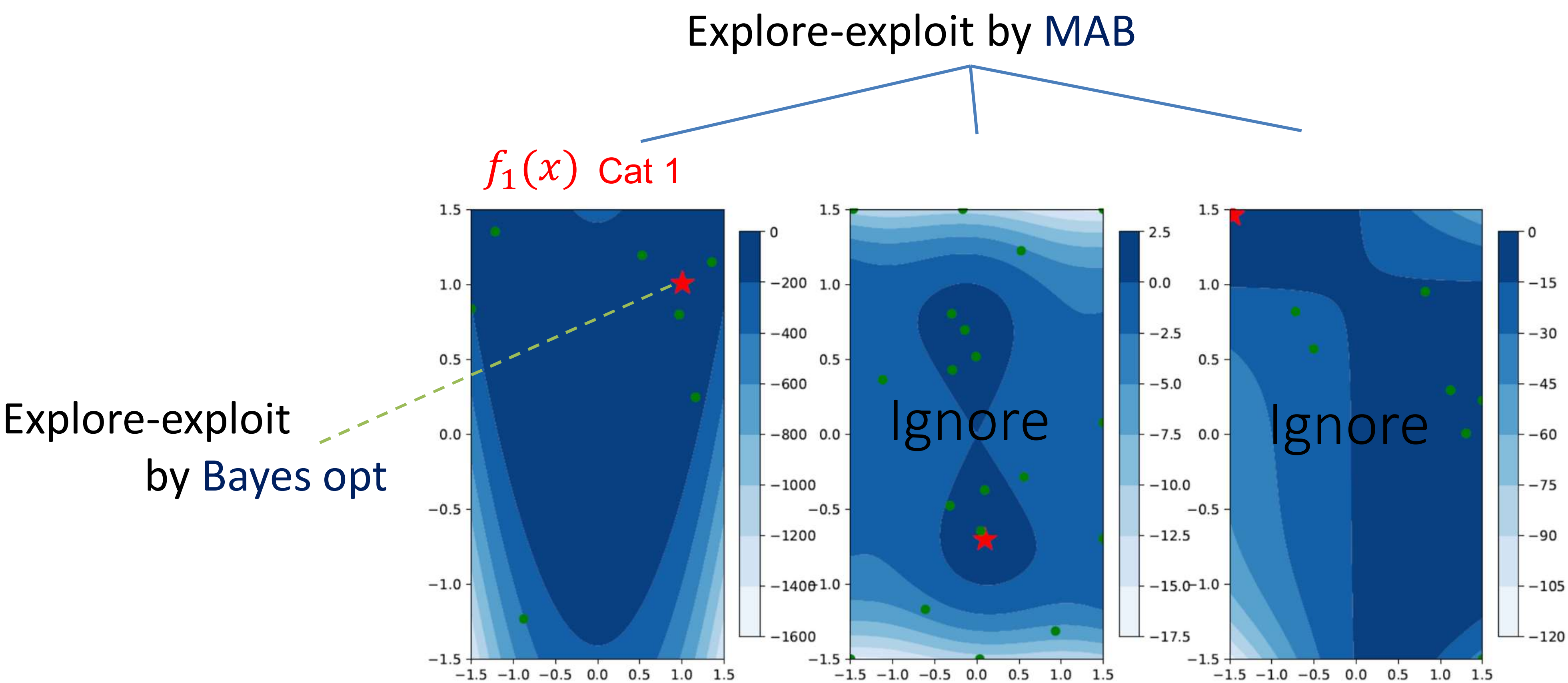
2. Continuous is sharing across categorical variables



- Single function $f(x, h)$

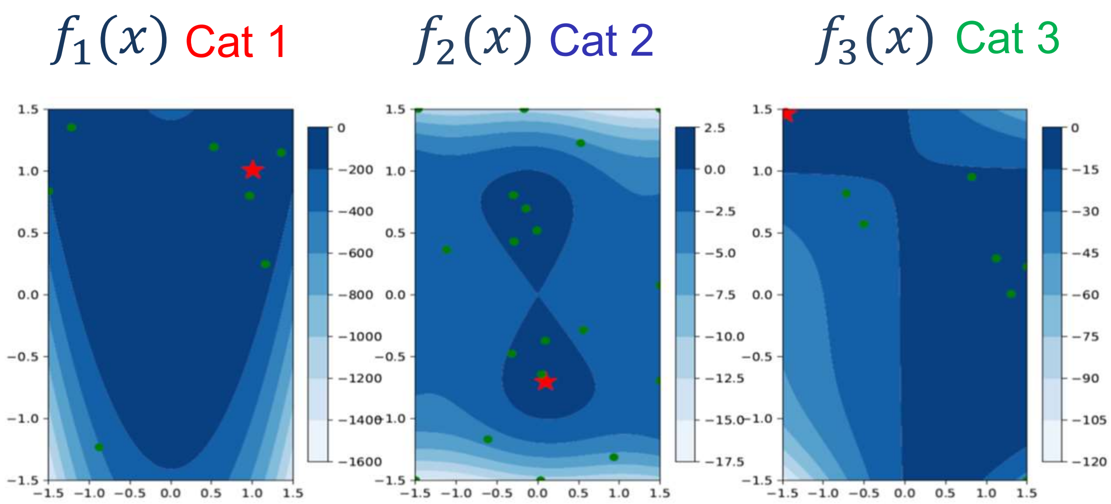
Global

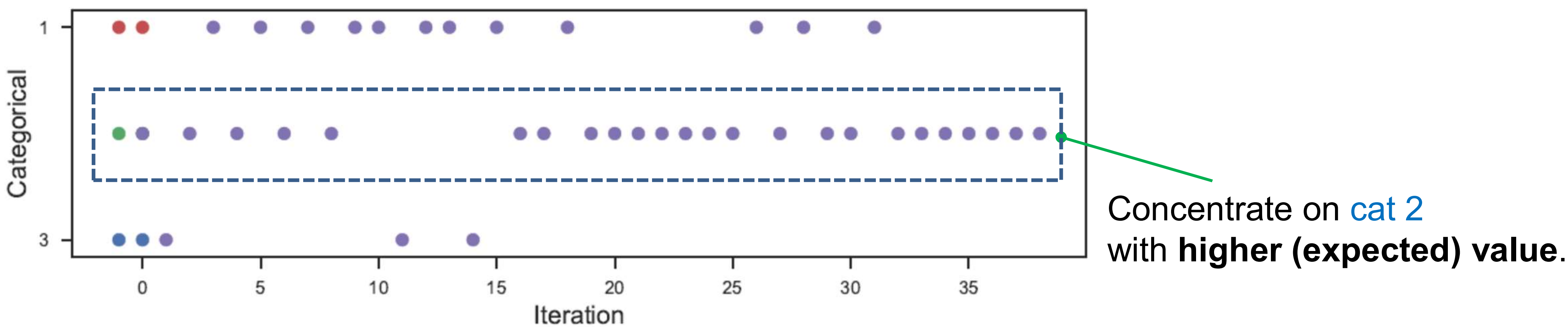
1. Categorical-specific



- Sequentially **pick a category *c*** using multi-armed bandit (MAB)
- Then optimize the continuous variables **given *c*** using Bayes opt

Visualization of the Algorithm





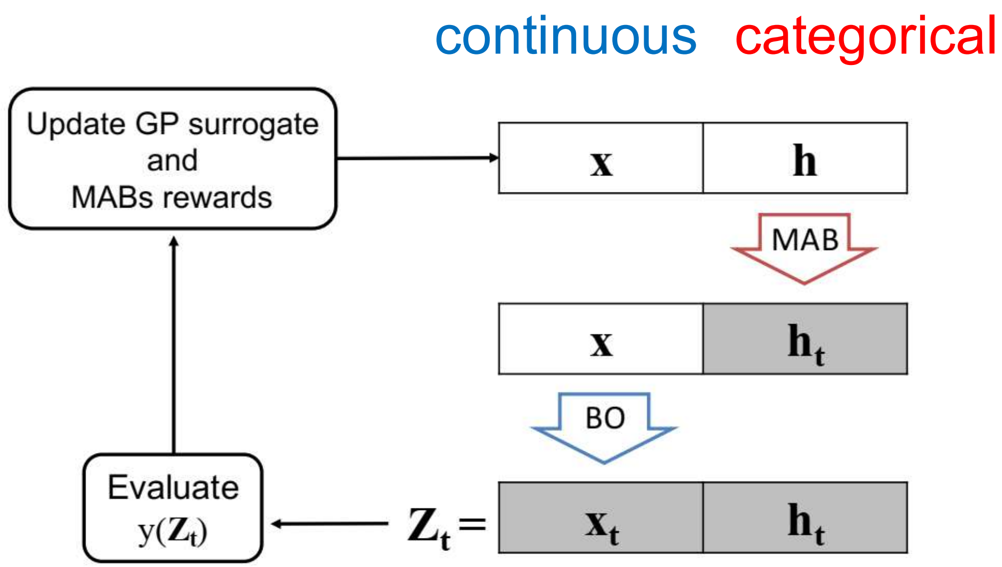
Agenda

- Hyperparameter Tuning and Experimental Design as Black-Boxes
- Part I: Bayesian Optimization
- Part II: Recent Advances in Bayesian Optimization
  - Batch Bayesian Optimization
  - Bayesian Optimization in Unknown Search Space
  - Mixed Categorical-Continuous Bayes Opt
    - Problem setting
    - Multi-armed bandits
    - Categorical-specific continuous optimization
    - Categorical-(non-)specific continuous optimization
- Research Directions in Bayesian Optimization

2. Continuous is sharing across categorical variables

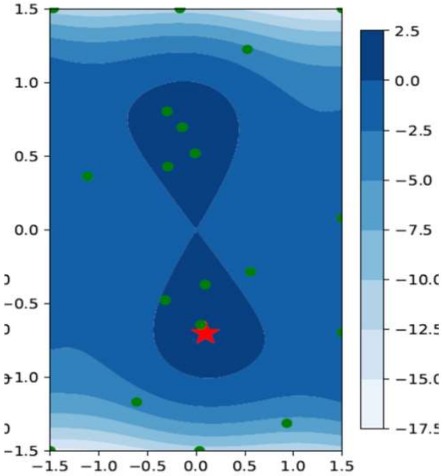
- MAB picks multiple categories: $\boldsymbol{h}$={SGD optimizer, tanh activation}
- Then optimize the continuous $\boldsymbol{x}$ given $\boldsymbol{h}$ in a single function

$$f(\boldsymbol{x}, \boldsymbol{h})$$



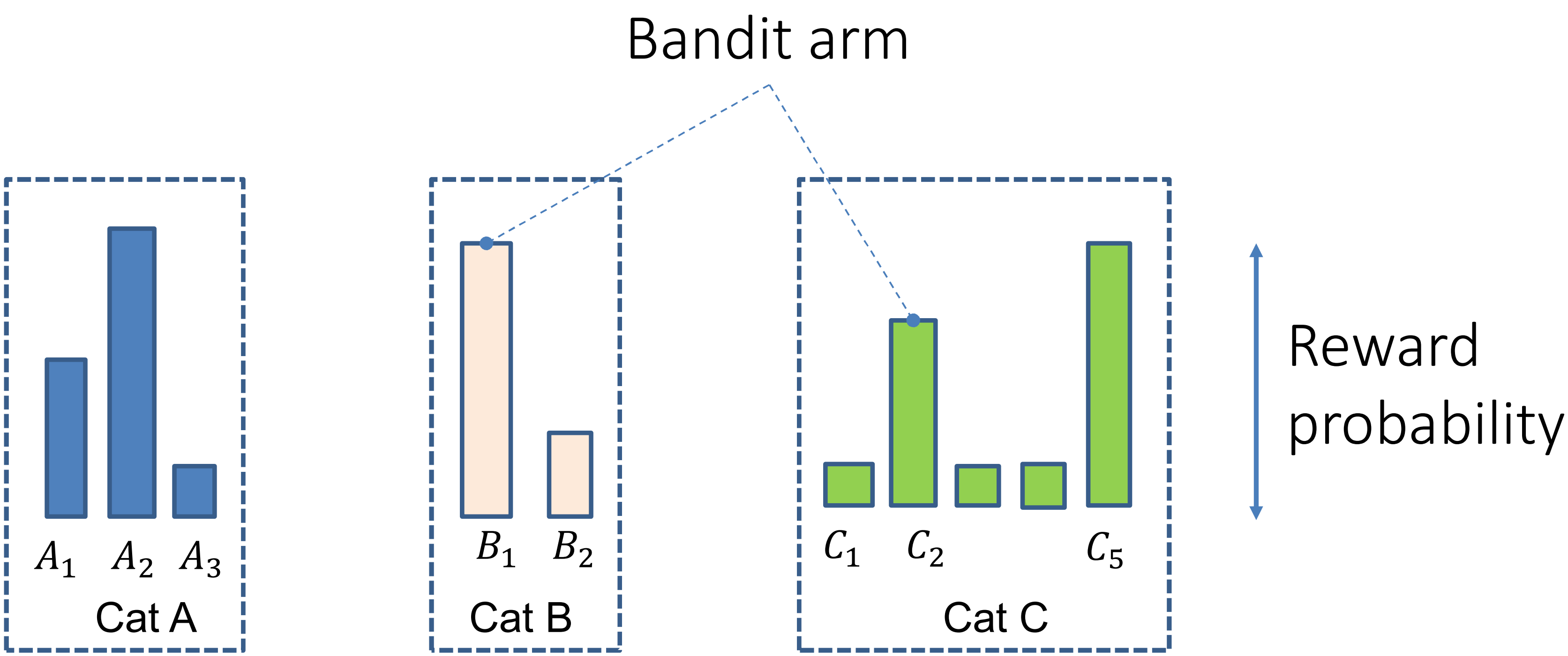
Given h, we optimize the continuous x

- Given the categorical $\boldsymbol{h}$, optimize continuous variable $\boldsymbol{x}$
- Consider the joint kernel for learning the surrogate model
  - Additive $k(\boldsymbol{x}, \boldsymbol{x}') + k(\boldsymbol{h}, \boldsymbol{h}')$
  - Multiplicative $k(\boldsymbol{x}, \boldsymbol{x}') \times k(\boldsymbol{h}, \boldsymbol{h}')$
  - $k(z, z') = (1-\lambda)k(\boldsymbol{x}, \boldsymbol{x}') \times k(\boldsymbol{h}, \boldsymbol{h}') + \lambda[k(\boldsymbol{x}, \boldsymbol{x}') + k(\boldsymbol{h}, \boldsymbol{h}')]$
  - λ can be estimated from the data.
- Optimizing in the continuous space



Given the feedback f([x, h]), we optimize h

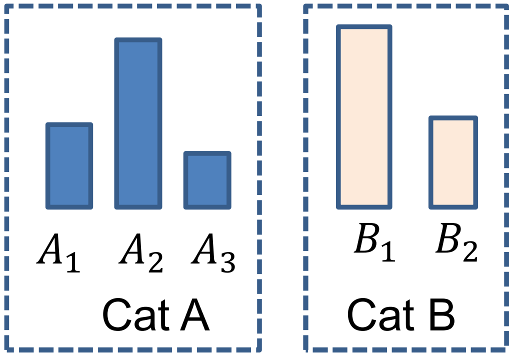
- Select $\boldsymbol{h}$ by EXP3 algorithm
- There is no assumption on the distribution of the reward



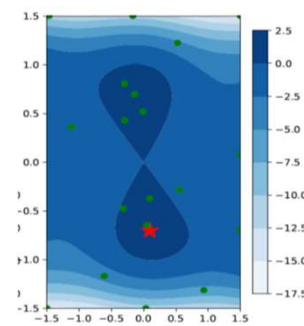
Multiple categories

Given the feedback f($[x, h]$), we optimize h

Optimize categorical $\boldsymbol{h}$



Optimize continuous $\boldsymbol{x}$ given $\boldsymbol{h}$



Observe the feedback $f([\boldsymbol{x}, \boldsymbol{h}])$

- Bandit feedback
- GP feedback

Bayes Opt Mixed Categorical – Continuous Input

continuous | **categorical**

- $y = f(x_1, x_2, x_3, x_4)$

x_1: learning rate $\in [1e^{-6}, 1e^{-1}]$

x_2: weight decay $\in [1e^{-6}, 1e^{-1}]$

x_3: optimiser type $\in \{SGD, Adam, \ldots\}$

x_4: activation type $\in \{tanh, sigmoid, \ldots\}$

Bayes Opt

Multi-armed Bandit

Continuous Categorical Bayesian Optimisation (CoCaBO)

Bayes opt over multiple continuous and categorical inputs. Ru Binxin et al 2020.

Alternative solutions

- Instead of using MAB, we can utilize decision tree for the categorical variable:

  Jenatton, R., Archambeau, C., González, J., & Seeger, M. Bayesian optimization with tree-structured dependencies. ICML, 2017.

- TPE:

  Bergstra, James S., Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. "Algorithms for hyper-parameter optimization." NeurIPS. 2011.

- SMAC:

  Hutter, Frank, Holger H. Hoos, and Kevin Leyton-Brown. "Sequential model-based optimization for general algorithm configuration." In *International conference on learning and intelligent optimization*, pp. 507-523. Springer, Berlin, Heidelberg, 2011.

Short Summary

- Bayesian optimization can work effectively up to 10 dimensions.

- In real-world scenarios, we may tackle the problems with large number of dimensions.

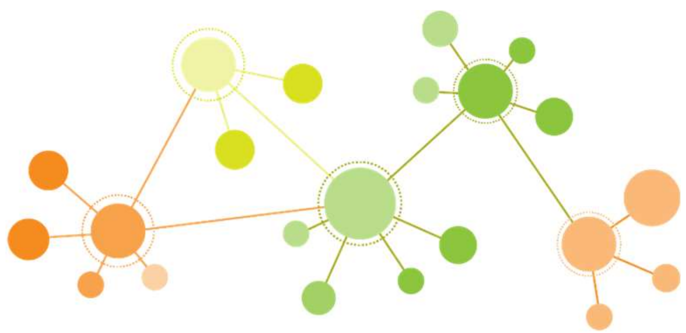
- Bayesian optimization research in high dimension is essential.

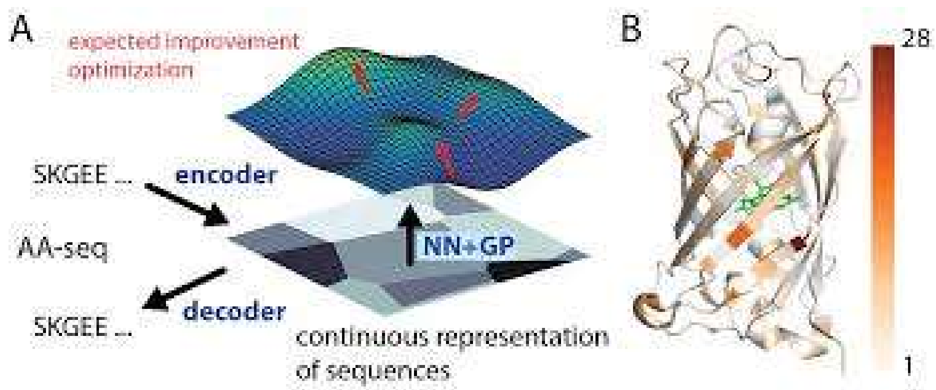
Agenda

- Hyperparameter Tuning and Experimental Design as Black-Boxes
- Part I: Bayesian Optimization
- Part II: Recent Advances in Bayesian Optimization
  - Batch Bayesian Optimization
  - High dimensional Bayes Opt
  - Mixed Categorical-Continuous Bayes Opt
- Research Directions in Bayesian Optimization

Opportunities for Future Research in BO

- Optimization in structured domains.
  - For example, how can we efficiently optimize over graphs, discrete sequences, trees, computer programs, etc.?

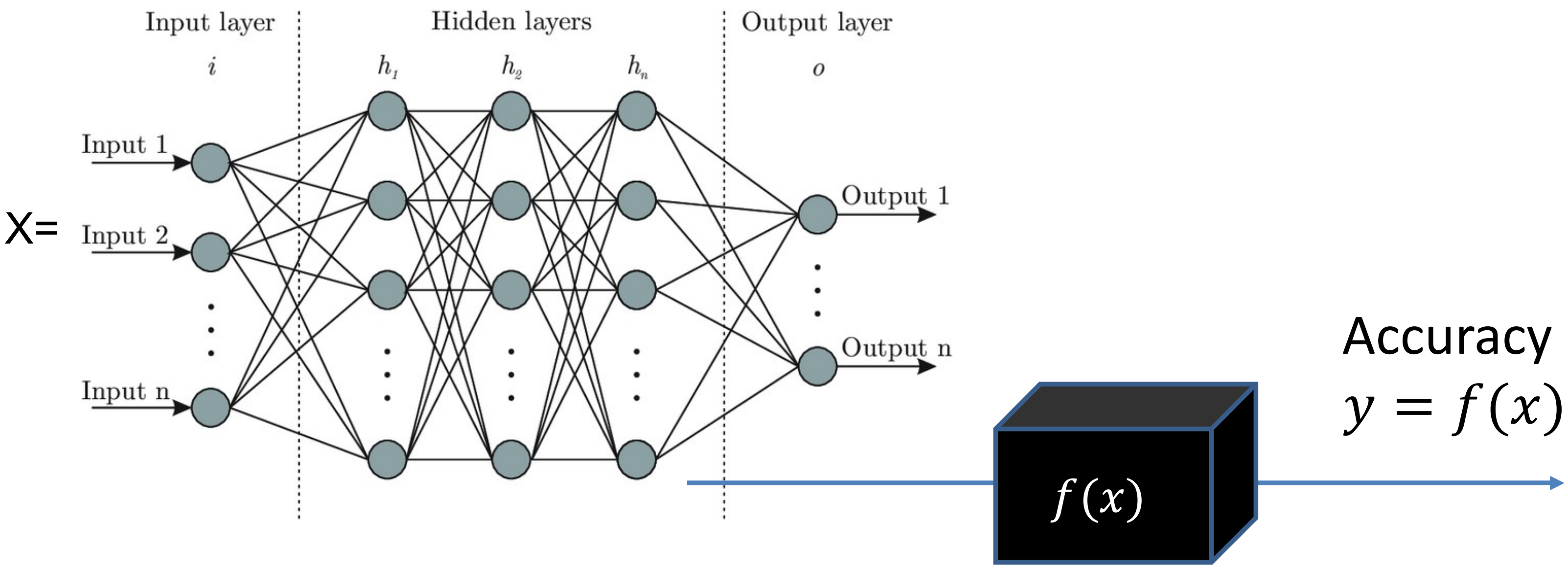




Eissman, Stephan, et al. "Bayesian optimization and attribute adjustment." *UAI*. 2018.

Opportunities for Future Research in BO

- Neural Architecture Search

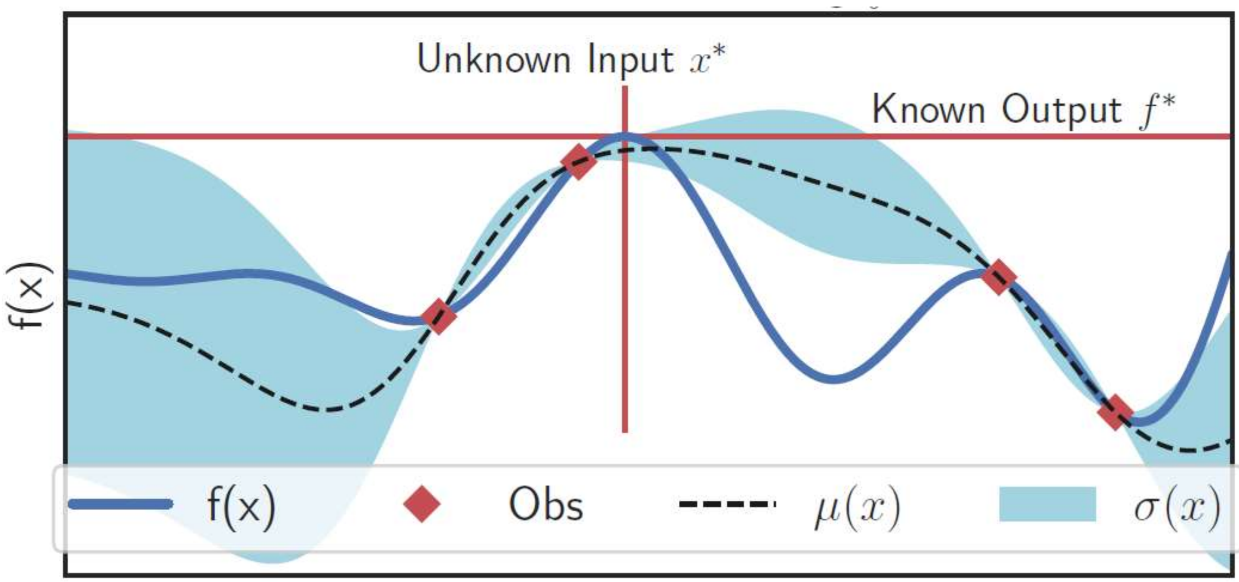


Finding the best architecture for the highest accuracy.

Kandasamy, K., at el NeurIPS 2018.

Opportunities for Future Research in BO

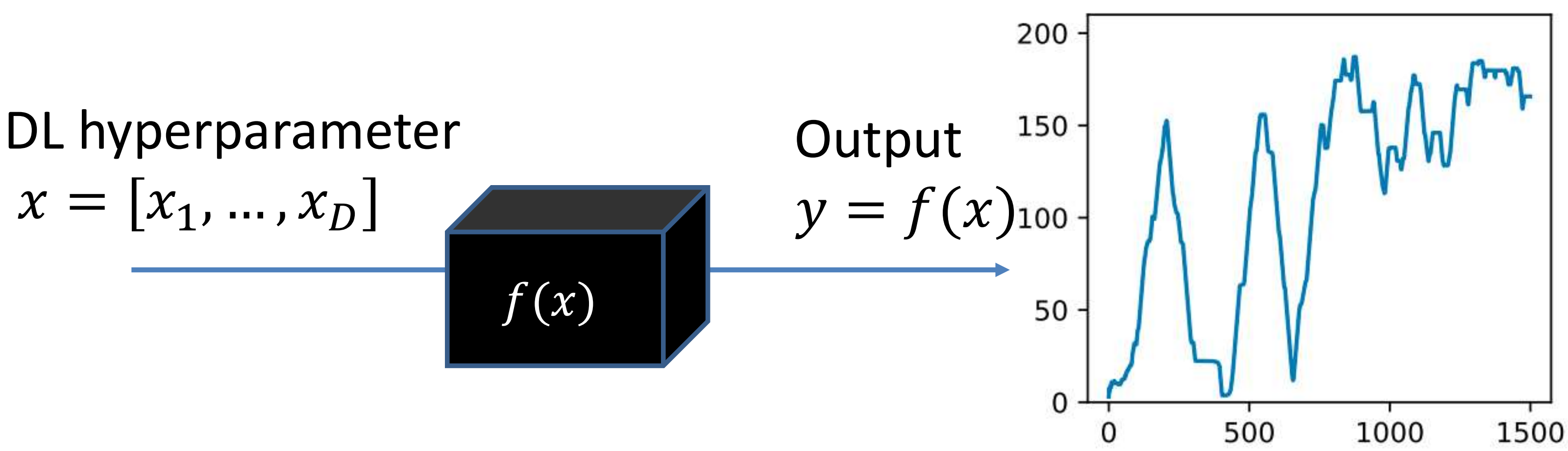
- Incorporating domain specific knowledge.
  - How can we easily encode and transfer available knowledge into BO methods in an easy and fast manner?
  - Knowing the optimum value of the function?



Vu Nguyen and Micheal Osborne. Knowing the what but not the where in Bayesian optimization. ICML 2020.

Opportunities for Future Research in BO

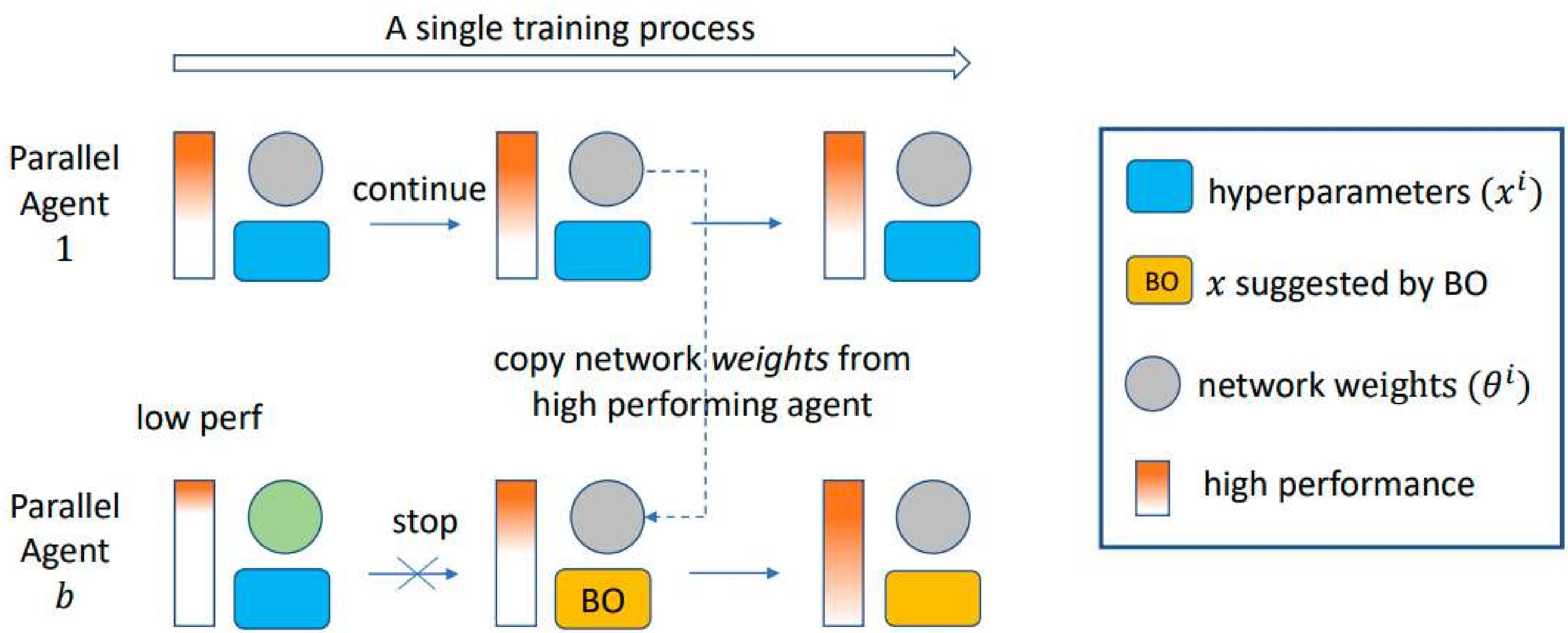
- Optimization with structured output response.
  - In training deep learning or deep reinforcement learning, the output is not only a single accuracy, but the whole training curve over epochs.



Vu Nguyen et al. "Bayesian optimization for iterative learning." *NeurIPS 2020*

Opportunities for Future Research in BO

- One-shot Bayesian optimization



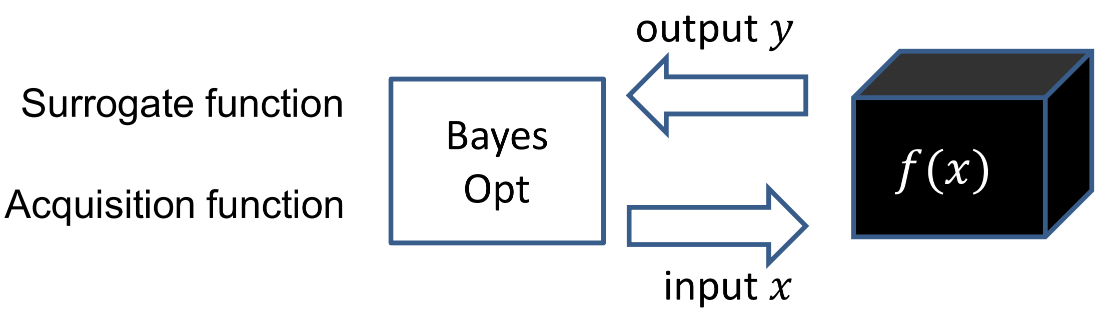
Parker-Holder, Jack, et al NeurIPS 2020.

Opportunities for Future Research in BO

- High dimension
  - Although previous researches have addressed high dimensional BO in different ways, the problem is still open.

- Mixed-type: categorical, continuous, discrete, binary variables
  - Categorical: Red, Green, Blue
  - Continuous: [0,1]
  - Discrete: 1,2,3,4,5
  - Binary: Yes, No

Research Summary in Bayesian Optimization



1. Neural architecture search

2. Mixed-type variable

3. Exploiting knowledge from the function

4. Multi-fidelity BO

5. Safe BO
Constraints BO

6. High dimensional

7. Theoretical Analysis

8. Parallel BO

9. Information theoretic
Multi-step look ahead

10. Gaussian process
Student-t process
Bayesian neural net

Take Home Messages

- Bayes opt is efficient for optimizing the black-box function.
- Existing works in Bayes opt are rich, this tutorial is by no means to handle every aspect of the field.
- Optimizing the black-box function with
  - Parallel optimization
  - High dimensional optimization
  - Mixed categorical-continuous optimization.

Final Remark

Reference

- Gonzalez, J., Dai, Z., Hennig, P., & Lawrence, N. D. Batch bayesian optimization via local penalization. AISTATS 2016.
- Ginsbourger, D., Le Riche, R., & Carraro, L. A multi-points criterion for deterministic parallel global optimization based on gaussian processes, 2008.
- Desautels, T., Krause, A., & Burdick, J. W. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. The Journal of Machine Learning Research, 15(1), 3873–3923, 2014.
- Contal, E., Buffoni, D., Robicquet, A., & Vayatis, N. Parallel gaussian process optimization with upper confidence bound and pure exploration. ECML/PKDD, 2013.
- V. Nguyen, S. Rana, S. K. Gupta, C. Li, S. Venkatesh. Budgeted Batch Bayesian Optimization. ICDM, 2016.
- Hernández-Lobato, J. M., J. Requeima, E. O. Pyzer-Knapp, and A. Aspuru-Guzik. "Parallel and Distributed Thompson Sampling for Large-scale Accelerated Exploration of Chemical Space.“ ICML, 2017
- Alvi, A., Ru, B., Calliess, J. P., Roberts, S., & Osborne, M. A. Asynchronous Batch Bayesian Optimisation with Improved Local Penalisation. ICML, 2019.
- Kandasamy, K., Krishnamurthy, A., Schneider, J., & Póczos, B,. Parallelised Bayesian Optimisation via Thompson Sampling. AISTATS, 2018.
- Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, and Nando De Freitas. Bayesian optimization in high dimensions via random embeddings. IJCAI, 2013.
- C. Li, K. Kandasamy, B. Poczos, and J. Schneider. High dimensional BO via restricted projection pursuit models. In AISTATS, 2016.
- S. Rana, C. Li, S. Gupta, V. Nguyen ,S. Venkatesh. High dimensional BO with elastic Gaussian process. ICML,2017.
- Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. ICML, 2010.

Reference

- C. Li, S. Gupta, S. Rana, V. Nguyen, S. Venkatesh, & A. Shilton. High Dimensional BO Using Dropout. IJCAI, 2017
- Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., & Poloczek, M. Scalable global optimization via local bayesian optimization. NeurIPS, 2019.
- Kandasamy, K., Schneider, J., & Póczos, B. High dimensional Bayesian optimisation and bandits via additive models. ICML, 2015.
- B. Ru, AS. Alvi, V. Nguyen, M. A. Osborne, SJ. Roberts. Bayesian Optimisation over Multiple Continuous and Categorical Inputs. ICML, 2020.
- S. Gopakumar, S. Gupta, S. Rana, V. Nguyen, S. Venkatesh. Algorithmic Assurance: An Active Approach to Algorithmic Testing using Bayesian Optimisation. NeurIPS, 2018
- Daxberger, E., Makarova, A., Turchetta, M., & Krause, A. Mixed-Variable Bayesian Optimization. IJCAI, 2020
- Auer P, Cesa-Bianchi N, Freund Y. & Schapire, RE. The non-stochastic multi-armed bandit problem. SIAM Journal of Computing.;32, 2002

Reference

- Jenatton, R., Archambeau, C., González, J., & Seeger, M. Bayesian optimization with tree-structured dependencies. ICML, 2017.
- Hutter, Frank, Holger H. Hoos, and Kevin Leyton-Brown. "Sequential model-based optimization for general algorithm configuration." In *International conference on learning and intelligent optimization*, pp. 507-523. Springer, Berlin, Heidelberg, 2011.
- Bergstra, James S., Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. "Algorithms for hyper-parameter optimization." NeurIPS, 2011.
- Eissman, Stephan, Daniel Levy, Rui Shu, Stefan Bartzsch, and Stefano Ermon. "Bayesian optimization and attribute adjustment." UAI, 2018.
- Vu Nguyen, Sebastian Schulze, and Michael A. Osborne. "Bayesian optimization for iterative learning." NeurIPS, 2020
- Parker-Holder, Jack, Vu Nguyen, and Stephen Roberts. "Provably Efficient Online Hyperparameter Optimization with Population-Based Bandits.“, NeurIPS 2020.
- Vu Nguyen and Micheal Osborne. Knowing the what but not the where in Bayesian optimization. ICML 2020.
- Kandasamy, K., Neiswanger, W., Schneider, J., Poczos, B. and Xing, E.P.. Neural architecture search with bayesian optimisation and optimal transport. NeurIPS 2018.

Question and Answer