Bayesian Optimization and Its Applications

Jungtaek Kim
jtkim@postech.ac.kr

Department of Computer Science and Engineering
POSTECH
77 Cheongam-ro, Pohang 37673
Republic of Korea

June 18, 2021

Table of Contents

Bayesian Optimization

Mathematical Optimization

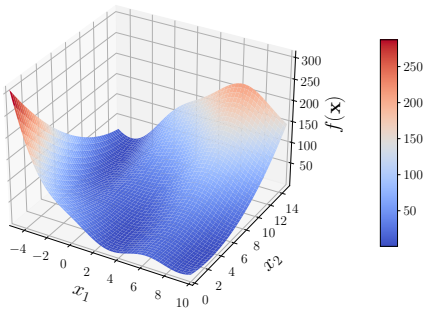


Figure 1: Branin function.

- Given an objective $f : \mathcal{A} \to \mathbb{R}$ where $\mathcal{A}$ is some set, it seeks minimum or maximum of the target function:

$$\mathbf{x}^* = \arg\min f(\mathbf{x}), \tag{1}$$

or

$$\mathbf{x}^* = \arg\max f(\mathbf{x}). \tag{2}$$

Mathematical Optimization

- To optimize an objective, we can select one of such strategies:
  - Gradient-based approaches;
  - Convex programming;
  - Metaheuristics.
- Each strategy has the advantage in the corresponding conditions of optimization problem.
- However, under certain circumstances, Bayesian optimization is the most effective method to solve some class of mathematical optimization problems.

Target Functions in Bayesian Optimization

- In general, a black-box function f,
  which has unknown functional forms or local geometric features such as saddle points, global optima, and local optima, is optimized,
  where a d-dimensional search space $\mathcal{X} \subset \mathbb{R}^d$ is convex and compact.

- Moreover, we can assume that the continuity of the objective function is unknown, a high-dimensional and mixed-variable feasible region is given,
  and the objective function is expensive to evaluate [Brochu et al., 2010, Shahriari et al., 2016, Frazier, 2018].

[Brochu et al., 2010] E. Brochu, V. M. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[Shahriari et al., 2016] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

[Frazier, 2018] P. I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Bayesian Optimization

- Bayesian optimization is a powerful strategy for finding an extremum of objective function,
  - where a closed-form expression for the objective function is not given,
  - but where a sample can be evaluated.
- Since we do not know a target function, it optimizes an acquisition function, instead of the target function.
- An acquisition function is defined with the factors that can control exploitation and exploration.

Bayesian Optimization

Algorithm 1 Bayesian Optimization

Input: Initial data $\mathcal{D}_{1:k} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{k}$ and a time budget T.
Output: The best candidate of global optimum $\mathbf{x}^\dagger$.

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Predict a function $\hat{f}(\mathbf{x} \,|\, \mathcal{D}_{1:k+t-1})$ considered as a surrogate of objective function.
3: Find a query $\mathbf{x}_{k+t}$ that maximizes an acquisition function:

$$\mathbf{x}_{k+t} = \arg\max_{\mathbf{x}} a(\mathbf{x} \,|\, \hat{f}, \mathcal{D}_{1:k+t-1}). \tag{3}$$

4: Evaluate $\mathbf{x}_{k+t}$ by a true objective function:

$$y_{k+t} = f(\mathbf{x}_{k+t}) + \epsilon_{k+t}, \tag{4}$$

where ϵ_{k+t} is a random observation noise.
5: Update historical data: $\mathcal{D}_{1:k+t} \leftarrow \mathcal{D}_{1:k+t-1} + \{(\mathbf{x}_t, y_t)\}$.
6: **end for**
7: **return** the best query $\mathbf{x}^\dagger$: $(\mathbf{x}^\dagger, y^\dagger) = \arg\min_{(\mathbf{x},y)\in\mathcal{D}_{1:k+T}} y$.

Surrogate Models

- A surrogate model estimates a true objective function, where historical observations are given.
- To balance a trade-off between exploration and exploitation, it predicts a function estimate and its uncertainty estimate over any query $\mathbf{x} \in \mathcal{X}$.
- Gaussian process regression, random forests regression [Hutter et al., 2011], and Bayesian neural network [Springenberg et al., 2016] have been used.

[Hutter et al., 2011] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the International Conference on Learning and Intelligent Optimization (LION)*, pages 507–523, Rome, Italy, 2011.

[Springenberg et al., 2016] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust Bayesian neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pages 4134–4142, Barcelona, Spain, 2016.

Gaussian Process

- A collection of random variables, any finite number of which have a joint Gaussian distribution [Rasmussen and Williams, 2006].
- Generally, Gaussian process (GP) is defined as

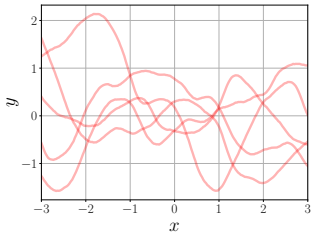
$$f \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{5}$$

where

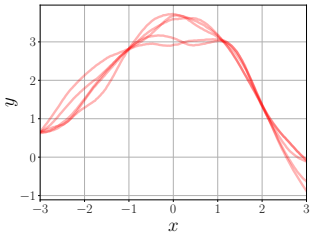
$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \tag{6}$$
$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]. \tag{7}$$

[Rasmussen and Williams, 2006] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

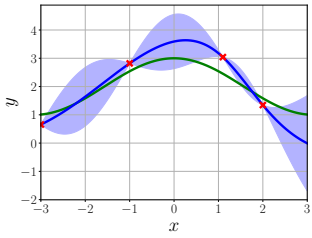
Gaussian Process Regression



(a) From prior function dist.



(b) From posterior function dist.



(c) Predictive dist.

Figure 2: Gaussian process regression for a function $\cos(x) + 2$ with an observation noise.

Gaussian Process Regression

- One of popular covariance functions, the squared-exponential covariance function in one dimension is defined as

$$k\left(x, x'\right) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}\left(x - x'\right)^2\right) + \sigma_n^2 \delta_{xx'}, \tag{8}$$

  where σ_f is a signal level, l is a length scale and σ_n is a noise level [Rasmussen and Williams, 2006].

- Posterior mean function $\mu(\cdot)$ and covariance function $\Sigma(\cdot)$:

$$\mu(\mathbf{X}^*) = K(\mathbf{X}^*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \boldsymbol{I})^{-1}\mathbf{y}, \tag{9}$$

$$\Sigma(\mathbf{X}^*) = K(\mathbf{X}^*, \mathbf{X}^*) - K(\mathbf{X}^*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \boldsymbol{I})^{-1} K(\mathbf{X}, \mathbf{X}^*). \tag{10}$$

[Rasmussen and Williams, 2006] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Gaussian Process Regression

- If non-zero mean prior is given, posterior mean and covariance functions:

$$\mu(\mathbf{X}^*) = K(\mathbf{X}^*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I})^{-1}(\mathbf{y} - \mu_p(\mathbf{X})) + \mu_p(\mathbf{X}), \tag{11}$$

$$\Sigma(\mathbf{X}^*) = K(\mathbf{X}^*, \mathbf{X}^*) + K(\mathbf{X}^*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I})^{-1} K(\mathbf{X}, \mathbf{X}^*), \tag{12}$$

  where $\mu_p(\cdot)$ is a prior mean function.

Acquisition Functions

- An acquisition function acquires the next sample to evaluate by a black-box function f.
- Traditionally, the probability of improvement (PI) [Kushner, 1964], the expected improvement (EI) [Močkus et al., 1978], and the GP upper confidence bound (GP-UCB) [Srinivas et al., 2010] have been used.
- Diverse acquisition functions such as knowledge gradient [Frazier et al., 2009], entropy search [Hennig and Schuler, 2012], and a clustering-guided GP-UCB [Kim and Choi, 2018b] have been proposed.

[Kushner, 1964] H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

[Močkus et al., 1978] J. Močkus, V. Tiesis, and A. Žilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2:117–129, 1978.

[Srinivas et al., 2010] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1015–1022, Haifa, Israel, 2010.

[Frazier et al., 2009] P. I. Frazier, W. B. Powell, and S. Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21(4):599–613, 2009.

[Hennig and Schuler, 2012] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13:1809–1837, 2012.

[Kim and Choi, 2018b] J. Kim and S. Choi. Clustering-guided GP-UCB for Bayesian optimization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2461–2465, Calgary, Alberta, Canada, 2018b.

Popular Acquisition Functions (Minimization Case)

- Suppose that $(\mathbf{x}^\dagger, y^\dagger) = \arg\min_{(\mathbf{x},y)\in\mathcal{D}} y$,

$$\mu(\mathbf{x}) \coloneqq \mu(\mathbf{x} \,|\, \mathcal{D}, \boldsymbol{\lambda}), \tag{13}$$

$$\sigma(\mathbf{x}) \coloneqq \sigma(\mathbf{x} \,|\, \mathcal{D}, \boldsymbol{\lambda}), \tag{14}$$

$$z = \begin{cases} \frac{f(\mathbf{x}^\dagger) - \mu(\mathbf{x})}{\sigma(\mathbf{x})}, & \text{if } \sigma(\mathbf{x}) > 0 \\ 0, & \text{if } \sigma(\mathbf{x}) = 0 \end{cases}. \tag{15}$$

- PI criterion [Kushner, 1964] is defined as

$$a_{\mathrm{PI}}(\mathbf{x} \,|\, \mathcal{D}, \boldsymbol{\lambda}) = \Phi(z), \tag{16}$$

where Φ is a cumulative distribution function of the standard normal distribution.

[Kushner, 1964] H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

Popular Acquisition Functions (Minimization Case)

- EI criterion [Močkus et al., 1978] is defined as

$$a_{\mathrm{EI}}(\mathbf{x} \,|\, \mathcal{D}, \boldsymbol{\lambda}) = \begin{cases} (f(\mathbf{x}^\dagger) - \mu(\mathbf{x}))\Phi(z) + \sigma(\mathbf{x})\phi(z), & \text{if} \quad \sigma(\mathbf{x}) > 0 \\ 0, & \text{if} \quad \sigma(\mathbf{x}) = 0 \end{cases}, \tag{17}$$

  where ϕ is a probability density function of the standard normal distribution.

- GP-UCB criterion [Srinivas et al., 2010] is defined as

$$a_{\mathrm{UCB}}(\mathbf{x} \,|\, \mathcal{D}, \boldsymbol{\lambda}) = -\mu(\mathbf{x}) + \beta\sigma(\mathbf{x}), \tag{18}$$

  where β is a trade-off hyperparameter.

[Močkus et al., 1978] J. Močkus, V. Tiesis, and A. Žilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2:117–129, 1978.

[Srinivas et al., 2010] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1015–1022, Haifa, Israel, 2010.

Acquisition Function Optimization

- We should find a global optimizer of acquisition function.
- But, in practice, either local optimizer or multi-started local optimizer can be a good option as a substitute of global optimizer.
- Analyses on these selections are provided in [Kim and Choi, 2020].

[Kim and Choi, 2020] J. Kim and S. Choi. On local optimizers of acquisition functions in Bayesian optimization. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, pages 675–690, Virtual, 2020.

On Local Optimizers of Acquisition Functions in Bayesian Optimization

Theorem 1 (Instantaneous regret difference between global and local optimizers)

Given $\delta_l \in [0, 1)$ and $\epsilon_l, \epsilon_1, \epsilon_2 > 0$, the regret difference for a local optimizer $\mathbf{x}_{t,l}$ at round t, $|r_{t,g} - r_{t,l}|$ is less than ϵ_l with a probability at least $1 - \delta_l$:

$$\mathbb{P}\big(|r_{t,g} - r_{t,l}| < \epsilon_l \big) \geq 1 - \delta_l, \tag{19}$$

where $\delta_l = \frac{\gamma}{\epsilon_1}(1 - \beta_g) + \frac{M}{\epsilon_2}$, $\epsilon_l = \epsilon_1 \epsilon_2$, $\gamma = \max_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}} \|\mathbf{x}_i - \mathbf{x}_j\|_2$ is the size of $\mathcal{X}$, β_g is the probability that a local optimizer of the acquisition function collapses with its global optimizer, and M is the Lipschitz constant.

On Local Optimizers of Acquisition Functions in Bayesian Optimization

Theorem 2 (Instantaneous regret difference between global and multi-started local optimizers)

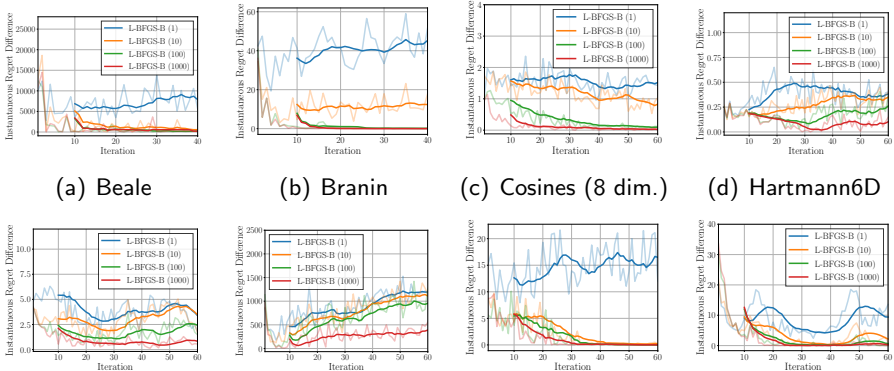
Given $\delta_m \in [0, 1)$ and $\epsilon_m, \epsilon_2, \epsilon_3 > 0$, a regret difference for a multi-started local optimizer $\mathbf{x}_{t,m}$, determined by starting from N initial points at round t, is less than ϵ_m with a probability at least $1 - \delta_m$:

$$\mathbb{P}\big(|r_{t,g} - r_{t,m}| < \epsilon_m \big) \geq 1 - \delta_m, \tag{20}$$

where $\delta_m = \frac{\gamma}{\epsilon_3} (1 - \beta_g)^N + \frac{M}{\epsilon_2}$, $\epsilon_m = \epsilon_2 \epsilon_3$, $\gamma = \max_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}} \|\mathbf{x}_i - \mathbf{x}_j\|_2$ is the size of $\mathcal{X}$, β_g is the probability that a local optimizer of the acquisition function collapses with its global optimizer, and M is the Lipschitz constant.

- By following our intuition, this bound is tighter than the bound provided in Theorem 1.

On Local Optimizers of Acquisition Functions in Bayesian Optimization



(a) Beale (b) Branin (c) Cosines (8 dim.) (d) Hartmann6D

(e) Holdertable (f) Rosenbrock (g) Six-Hump Camel (h) Sphere

Figure 3: Empirical results on Theorem 1 and Theorem 2.

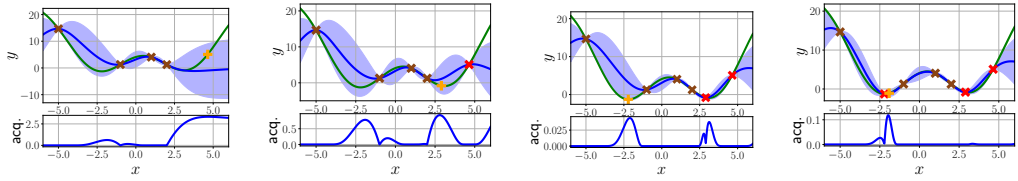
On Local Optimizers of Acquisition Functions in Bayesian Optimization

Table 1: Time (sec.) consumed in optimizing acquisition functions. L denotes L-BFGS-B.

	Beale	**Branin**	**Cosines (8 dim.)**	**Hart-mann6D**	**Holder-table**	**Rosen-brock**	**Six-Hump Camel**	**Sphere**
DIRECT	3.434	2.987	2.508	0.728	2.935	13.928	4.639	10.707
L (1)	0.010	0.004	0.023	0.026	0.017	0.005	0.010	0.030
L (10)	0.096	0.036	0.224	0.253	0.177	0.050	0.100	0.311
L (100)	0.977	0.363	2.224	2.533	1.760	0.504	0.969	3.048
L (1000)	9.720	3.633	22.306	25.305	17.629	5.049	9.682	30.764

- Multi-started local optimizer provides a more efficient approach than global optimizer, in terms of computational complexities.

Synthetic Example

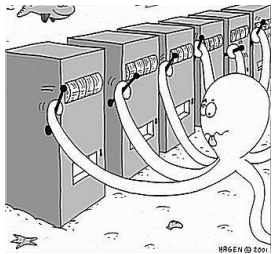


(a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4

Figure 4: Bayesian optimization example where a true objective function is $y = 4\cos(x) + 0.1x + 2\sin(x) + 0.4(x - 0.5)^2$ and EI is used as an acquisition function. It is implemented by BayesO [Kim and Choi, 2017].

[Kim and Choi, 2017] J. Kim and S. Choi. BayesO: A Bayesian optimization framework in Python. `https://bayeso.org`, 2017.

Relationship to Multi-Armed Bandit Problem



- Each machine returns a reward $\hat{r}_a \sim p_{\boldsymbol{\theta}_a}(r_a)$ where $a \in \{1, \ldots, K\}$.
- It minimizes a cumulative regret $T\mu^* - \sum_{t=1}^{T} \hat{r}_{a_t}$ where $\mu^* = \max_{a \in \{1,\ldots,K\}} \mu_a$.
- Bayesian optimization can be considered as infinite bandits with dependent arms.

Relationship to Thompson Sampling

- Thompson sampling is usually applied in multi-armed bandit problems.
- For the case of a beta-Bernoulli bandit, Thompson sampling is defined as

Algorithm 2 Thompson Sampling for a Beta-Bernoulli Bandit

for $t = 1, 2, \ldots, T$ **do**
  for $k = 1, \ldots, K$ **do**
    Sample $\hat{\theta}_k \sim \text{beta}(\alpha_k, \beta_k)$.
  end for
  $x_t \leftarrow \arg\max_k \hat{\theta}_k$.
  Apply x_t and observe r_t.
  $(\alpha_{x_t}, \beta_{x_t}) \leftarrow (\alpha_{x_t} + r_t, \beta_{x_t} + 1 - r_t)$.
end for

- After sampling the possibilities, it chooses a maximizer of those sampled values.

BayesO [Kim and Choi, 2017]

- Current version: 0.5.0
- Supported Python version: 3.6, 3.7, 3.8, 3.9 (tested by Travis CI)
- Web page: `https://bayeso.org`
- GitHub repo: `https://github.com/jungtaekkim/bayeso`
- Documentation: `https://bayeso.readthedocs.io`
- License: MIT license

[Kim and Choi, 2017] J. Kim and S. Choi. BayesO: A Bayesian optimization framework in Python. `https://bayeso.org`, 2017.

Applications of Bayesian Optimization

Automated Machine Learning

- It finds the optimal machine learning model without human intervention, by automatically conducting feature transformation, algorithm selection, and hyperparameter optimization [Hutter et al., 2019].
- Given a training dataset $\mathcal{D}_{\text{train}}$ and a validation dataset $\mathcal{D}_{\text{val}}$, the optimal hyperparameter vector $\boldsymbol{\lambda}^*$ for an automated machine learning system is found:

$$\boldsymbol{\lambda}^* = \text{AutoML}(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}, \Lambda), \tag{21}$$

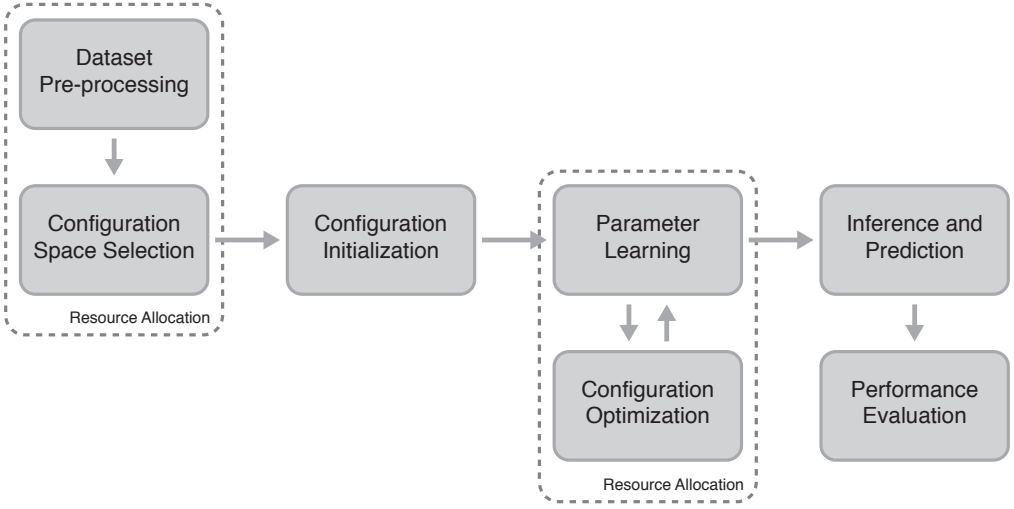
  where AutoML is an automated machine learning system.
- The approaches that take the third place in AutoML5 phase of AutoML Challenge [Kim et al., 2016] and the second place in AutoML Challenge 2018 [Kim and Choi, 2018a] have been presented.

[Hutter et al., 2019] F. Hutter, L. Kotthoff, and J. Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

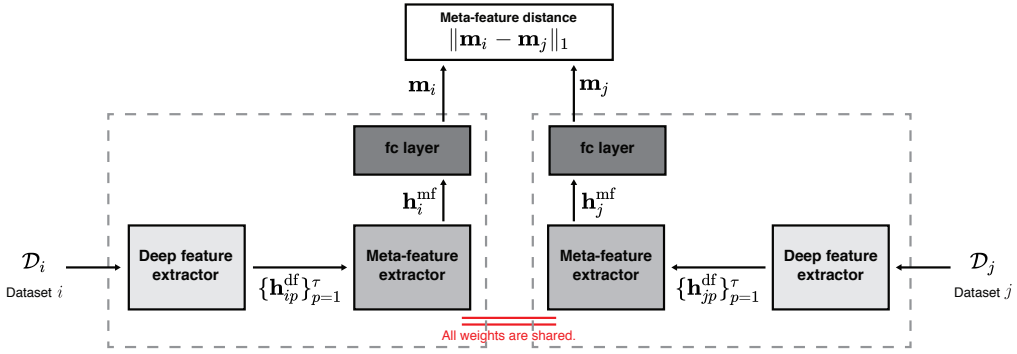
[Kim et al., 2016] J. Kim, J. Jeong, and S. Choi. AutoML Challenge: AutoML framework using random space partitioning optimizer. In *International Conference on Machine Learning Workshop on Automatic Machine Learning (AutoML)*, New York, New York, USA, 2016.

[Kim and Choi, 2018a] J. Kim and S. Choi. Automated machine learning for soft voting in an ensemble of tree-based classifiers. In *International Conference on Machine Learning Workshop on Automatic Machine Learning (AutoML)*, Stockholm, Sweden, 2018a.

Automated Machine Learning



Learning to Transfer Initializations for Bayesian Hyperparameter Optimization [Kim et al., 2017]



- It can measure the similarities between unseen dataset and historical datasets by learning to warm-start Bayesian hyperparameter optimization.

[Kim et al., 2017] J. Kim, S. Kim, and S. Choi. Learning to transfer initializations for Bayesian hyperparameter optimization. In *Neural Information Processing Systems Workshop on Bayesian Optimization (BayesOpt)*, Long Beach, California, USA, 2017.

Combinatorial 3D Shape Generation via Sequential Assembly

- 3D shape generation via sequential assembly mimics a human assembling process, by allocating a budget of primitives given [Kim et al., 2020].
- We solve a sequential problem with Bayesian optimization-based framework of combinatorial 3D shape generation, composed of a set of geometric primitives.
- To determine the position of the next primitive, two evaluation functions regarding occupiability and stability are defined.
- Occupiability encourages us to follow a target shape and stability helps to create a physically stable combination.
- A new combinatorial 3D shape dataset that consists of 14 classes and 406 instances is also introduced in this work.

[Kim et al., 2020b] J. Kim, H. Chung, J. Lee, M. Cho, and J. Park. Combinatorial 3D shape generation via sequential assembly. In *Neural Information Processing Systems Workshop on Machine Learning for Engineering Modeling, Simulation, and Design (ML4Eng)*, Virtual, 2020b.

Experimental Results

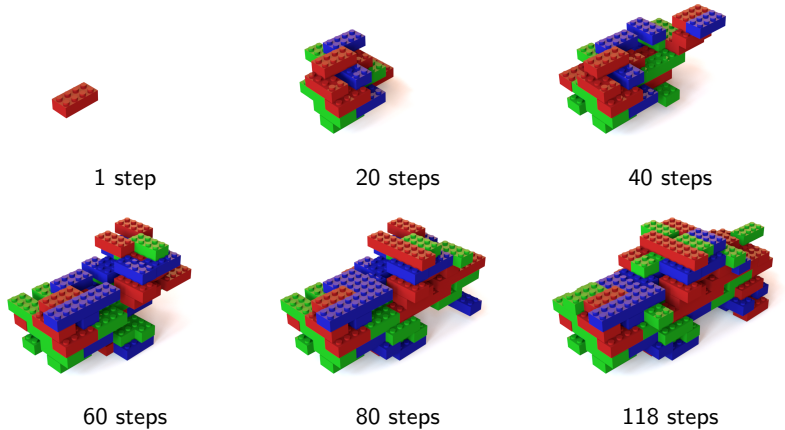
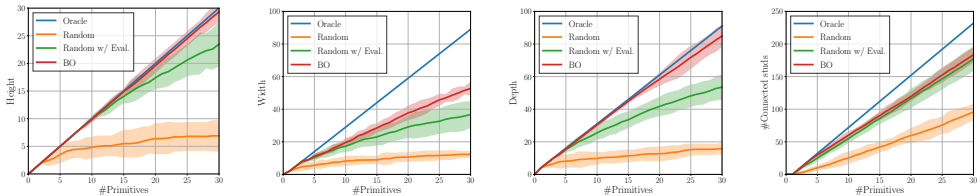


Figure 5: Generated assembling sequence that creates a *car* shape with 118 unit primitives.

Experimental Results

- We apply our framework in optimizing specific explicit functions.



(a) Height (b) Width (c) Depth (d) #Conn. studs

Figure 6: Quantitative results on maximizing explicit evaluation functions.

Combinatorial 3D Shape Dataset

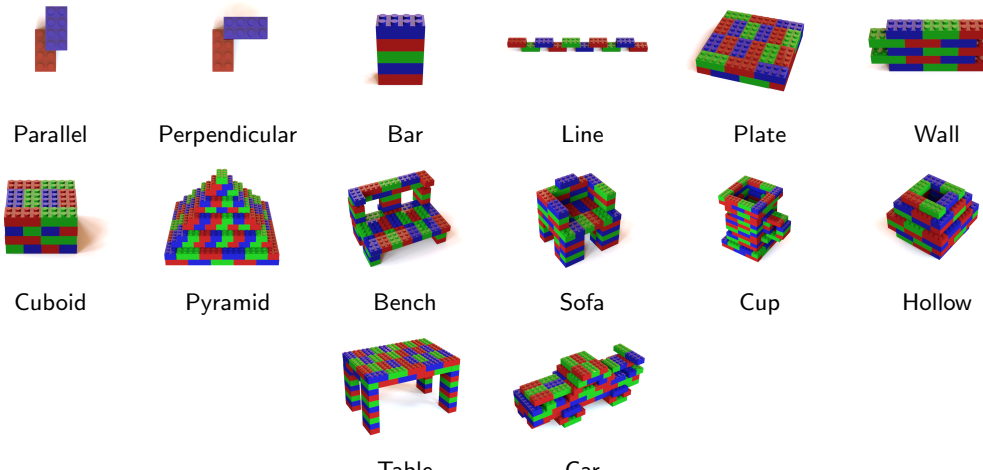


Figure 7: Selected examples from our dataset.

Takeaway

- Bayesian optimization is a powerful method to optimize a black-box function.
- Instead of methods based on heuristic or prior knowledge, it provides a structured approach to finding an optimal solution.
- Bayesian optimization is expanding into various real-world applications.

Thank you.

References I

E. Brochu, V. M. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

P. I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

P. I. Frazier, W. B. Powell, and S. Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21(4): 599–613, 2009.

P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13:1809–1837, 2012.

F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the International Conference on Learning and Intelligent Optimization (LION)*, pages 507–523, Rome, Italy, 2011.

F. Hutter, L. Kotthoff, and J. Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

J. Kim and S. Choi. BayesO: A Bayesian optimization framework in Python. `https://bayeso.org`, 2017.

J. Kim and S. Choi. Automated machine learning for soft voting in an ensemble of tree-based classifiers. In *International Conference on Machine Learning Workshop on Automatic Machine Learning (AutoML)*, Stockholm, Sweden, 2018a.

J. Kim and S. Choi. Clustering-guided GP-UCB for Bayesian optimization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2461–2465, Calgary, Alberta, Canada, 2018b.

J. Kim and S. Choi. On local optimizers of acquisition functions in Bayesian optimization. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, pages 675–690, Virtual, 2020.

J. Kim, J. Jeong, and S. Choi. AutoML Challenge: AutoML framework using random space partitioning optimizer. In *International Conference on Machine Learning Workshop on Automatic Machine Learning (AutoML)*, New York, New York, USA, 2016.

J. Kim, S. Kim, and S. Choi. Learning to transfer initializations for Bayesian hyperparameter optimization. In *Neural Information Processing Systems Workshop on Bayesian Optimization (BayesOpt)*, Long Beach, California, USA, 2017.

J. Kim, H. Chung, J. Lee, M. Cho, and J. Park. Combinatorial 3D shape generation via sequential assembly. In *Neural Information Processing Systems Workshop on Machine Learning for Engineering Modeling, Simulation, and Design (ML4Eng)*, Virtual, 2020.

H. J. Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

References II

J. Močkus, V. Tiesis, and A. Žilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2:117–129, 1978.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter. Bayesian optimization with robust Bayesian neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pages 4134–4142, Barcelona, Spain, 2016.

N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1015–1022, Haifa, Israel, 2010.