Bayesian Optimization with Approximate Set Kernels

Jungtaek Kim
jtkim@postech.ac.kr

A joint work with Michael McCourt, Tackgeun You,
Saehoon Kim, and Seungjin Choi

POSTECH
Pohang 37673, Republic of Korea
https://jungtaek.github.io

ECML-PKDD 2021, Journal Track

POSTECH POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY · SIGOPT · intel · kakaobrain · BARO AI ACADEMY

Table of Contents

Introduction

Introduction

- Bayesian optimization is an effective method to optimize an expensive black-box function.
- It has proven useful in several applications, including hyperparameter optimization [Snoek et al., 2012], neural architecture search [Kandasamy et al., 2018], and material design [Frazier and Wang, 2016].
- Unlike this standard Bayesian optimization formulation, we assume that a search region is $\mathcal{X}_{\text{set}} = \{\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \mid \mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d\}$ for a fixed positive integer m.
- Thus, for $\mathbf{X} \in \mathcal{X}_{\text{set}}$, f would take in a *set* containing m elements, all of length d, and return a noisy function value y:

$$y = f(\mathbf{X}) + \epsilon. \tag{1}$$

[Snoek et al., 2012] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *NeurIPS*, volume 25, pages 2951–2959, Lake Tahoe, Nevada, USA, 2012.

[Kandasamy et al., 2018] K. Kandasamy, W. Neiswanger, J. Schneider, B. Póczos, and E. P. Xing. Neural architecture search with Bayesian optimisation and optimal transport. In *NeurIPS*, volume 31, pages 2016–2025, Montreal, Quebec, Canada, 2018.

[Frazier and Wang, 2016] P. I. Frazier and J. Wang. Bayesian optimization for materials design. In *Information Science for Materials Discovery and Design*, pages 45–75. Springer, 2016.

Contributions

- To propose Bayesian optimization over sets, we adapt and augment a strategy proposed in Gätner et al. [2002] involving the creation of a specific *set kernel*.
- It can be directly used to build surrogate functions through GP regression, which can power the Bayesian optimization strategy.
- We develop a computationally efficient approximation to this set kernel.
- Another contribution is a constrained acquisition function optimization over set inputs.
- Finally, we conduct our method on various experimental circumstances such as clustering algorithm initialization and active nearest neighbor search for point clouds.

[Gätner et al., 2002] T. Gätner, P. A. Flach, A. Kowalczyk, and A. J. Smola. Multi-instance kernels. In *ICML*, pages 179–186, Sydney, Australia, 2002.

Background

Bayesian Optimization

- Bayesian optimization seeks to minimize an unknown function f which is expensive to evaluate, $\mathbf{x}^\star = \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x})$, where $\mathcal{X} \subset \mathbb{R}^d$ is a compact space.
- It is a sequential optimization strategy which, at each iteration, performs the following three computations:
  1. Using the n data presently available, $\{(\mathbf{x}_i, y_i)\}$ for $i \in [n]$, build a probabilistic surrogate model s_n meant to approximate f;
  2. Using the surrogate model s_n, compute an acquisition function a_n, which represents the utility of next acquiring data at some new point $\mathbf{x}$;
  3. Observe y_{n+1} from a true function f at the location $\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} a_n(\mathbf{x})$.
- In this paper, we use GP regression [Rasmussen and Williams, 2006] to produce the surrogate function s_n; from s_n, we use the Gaussian process upper confidence bound (GP-UCB) criterion [Srinivas et al., 2010].

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

[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 *ICML*, pages 1015–1022, Haifa, Israel, 2010.

Set Kernel

- A set of m vectors is denoted as $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, where $\mathbf{x}_i$ is in a compact space $\mathcal{X} \subset \mathbb{R}^d$.
- To build a GP surrogate, we require a prior belief of the covariance between elements in $\mathcal{X}_{\text{set}} = \{\{\mathbf{x}_1, \ldots, \mathbf{x}_m\} \mid \mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d\}$.
- Given an empirical approximation of the kernel mean $\boldsymbol{\mu}_{\mathbf{X}} \approx |\mathbf{X}|^{-1} \sum_{i=1}^{|\mathbf{X}|} \phi(\mathbf{x}_i)$, where ϕ is a feature map $\mathbb{R}^d \to \mathbb{R}^{d'}$ and d' is a dimensionality of projected space by ϕ, a set kernel [Gätner et al., 2002, Muandet et al., 2017] is defined as

$$k_{\text{set}}\left(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}\right) = \langle \boldsymbol{\mu}_{\mathbf{X}^{(1)}}, \boldsymbol{\mu}_{\mathbf{X}^{(2)}} \rangle = \frac{1}{|\mathbf{X}^{(1)}||\mathbf{X}^{(2)}|} \sum_{i=1}^{|\mathbf{X}^{(1)}|} \sum_{j=1}^{|\mathbf{X}^{(2)}|} k\left(\mathbf{x}_i^{(1)}, \mathbf{x}_j^{(2)}\right), \quad (2)$$

where $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$.

[Gätner et al., 2002] T. Gätner, P. A. Flach, A. Kowalczyk, and A. J. Smola. Multi-instance kernels. In *ICML*, pages 179–186, Sydney, Australia, 2002.

[Muandet et al., 2017] K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

Proposed Method

Approximation of the Set Kernel

- We define the matrix $\mathsf{K} \in \mathbb{R}^{n \times n}$ as

$$(\mathsf{K})_{ij} = k_{\text{set}}\left(\mathbf{X}^{(i)}, \mathbf{X}^{(j)}\right), \tag{3}$$

  for k_{set} defined with a chosen inner kernel k as in (2).

- Computing (3) requires pairwise comparisons between all sets present in $\mathfrak{X}$, which has computational complexity $\mathcal{O}(n^2 m^2 d)$.

- To alleviate this cost, we propose to approximate (2) with

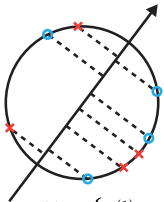
$$\tilde{k}_{\text{set}}\left(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}; \pi, \mathbf{w}, L\right) = k_{\text{set}}\left(\tilde{\mathbf{X}}^{(1)}, \tilde{\mathbf{X}}^{(2)}\right), \tag{4}$$

  where $\pi : [m] \to [m]$, $\mathbf{w} \in \mathbb{R}^d$ and $L \in \mathbb{Z}_+$ and $\tilde{\mathbf{X}}^{(i)}$ is a subset of $\mathbf{X}^{(i)}$ which is defined by those three quantities.

Approximation of the Set Kernel

1

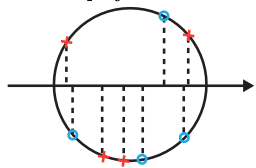
$\mathbf{w} \sim \mathcal{N}(0, \mathbf{I}_2)$



$$\mathbf{X}^{(1)} = \left\{\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_4^{(1)}\right\}$$
$$\mathbf{X}^{(2)} = \left\{\mathbf{x}_1^{(2)}, \ldots, \mathbf{x}_4^{(2)}\right\}$$

2

Random scalar projection



$$\left\{\mathbf{w}^\top \mathbf{x}_1^{(1)}, \ldots, \mathbf{w}^\top \mathbf{x}_4^{(1)}\right\}$$
$$\left\{\mathbf{w}^\top \mathbf{x}_1^{(2)}, \ldots, \mathbf{w}^\top \mathbf{x}_4^{(2)}\right\}$$

3

$$\tilde{\mathbf{X}}^{(1)} = \left\{\mathbf{x}_1^{(1)}, \mathbf{x}_3^{(1)}\right\}$$
$$\tilde{\mathbf{X}}^{(2)} = \left\{\mathbf{x}_2^{(2)}, \mathbf{x}_3^{(2)}\right\}$$

$$\tilde{k}_{\text{set}}\left(\tilde{\mathbf{X}}^{(1)}, \tilde{\mathbf{X}}^{(2)}\right)$$
$$= \frac{1}{4}\left(k(\mathbf{x}_1^{(1)}, \mathbf{x}_2^{(2)}) + k(\mathbf{x}_1^{(1)}, \mathbf{x}_3^{(2)})\right.$$
$$\left. + k(\mathbf{x}_3^{(1)}, \mathbf{x}_2^{(2)}) + k(\mathbf{x}_3^{(1)}, \mathbf{x}_3^{(2)})\right)$$

Properties of the Approximation

Property 1

The approximation satisfies pairwise symmetry.

Property 2

The "ordering" of the elements in the sets $\mathbf{X}^{(i)}, \mathbf{X}^{(j)}$ should not matter when computing $\tilde{k}_{set}$.

Property 3

Because k_{set} is positive-definite on these L-element sets, we know that $\tilde{k}_{set}$ is also positive-definite.

Property 4

Since the approximation method aims to choose subsets of input sets, the computational cost becomes lower than the original formulation.

Properties of the Approximation

Theorem 1

Suppose that we are given two sets $\mathbf{X}, \mathbf{Y} \in \mathcal{X}_{set}$ *and* $L \in \mathbb{Z}_+$. *Suppose, furthermore, that* $\mathbf{w}$ *and* π *can be generated randomly to form subsets* $\tilde{\mathbf{X}}$ *and* $\tilde{\mathbf{Y}}$. *The value of* $\tilde{k}_{set}(\mathbf{X}, \mathbf{Y}; \mathbf{w}, \pi, L)$ *is an unbiased estimator of the value of* $k_{set}(\mathbf{X}, \mathbf{Y})$.

Theorem 2

Under the same conditions as in Theorem 1, suppose that $k(\mathbf{x}, \mathbf{x}') \geq 0$ *for all* $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. *The variance of* $\tilde{k}_{set}(\mathbf{X}, \mathbf{Y}; \mathbf{w}, \pi, L)$ *is bounded by a function of* m, L *and* $k_{set}(\mathbf{X}, \mathbf{Y})$:

$$Var\left[\tilde{k}_{set}(\mathbf{X}, \mathbf{Y}; \mathbf{w}, \pi, L)\right] \leq \left(\frac{m^4}{L^4} - 1\right) k_{set}(\mathbf{X}, \mathbf{Y})^2. \tag{5}$$

Bayesian Optimization over Sets

Algorithm 1 Bayesian Optimization over Sets

Input: A domain $\mathcal{X}_{\text{set}}$, a function $f : \mathcal{X}_{\text{set}} \to \mathbb{R}$, a budget $T \in \mathbb{Z}_+$.

Output: Best acquired set $\mathbf{X}^\dagger$.

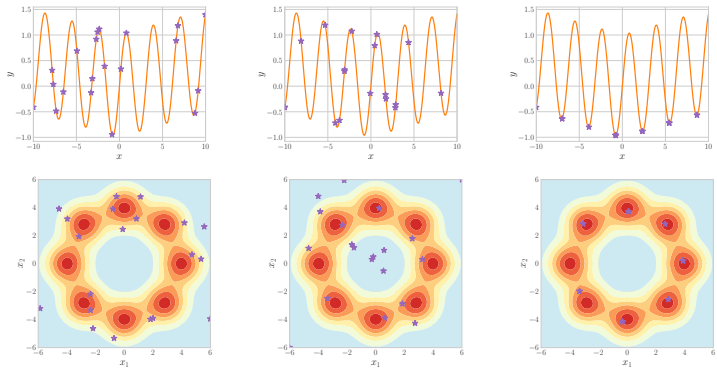
1: Choose an initial point $\mathbf{X}^{(1)}$ randomly from $\mathcal{X}_{\text{set}}$ and evaluate $y_1 = f(\mathbf{X}^{(1)}) + \epsilon_1$.
2: **for** k from 1 to $T - 1$ **do**
3: Fit the surrogate model s_k to all data $\{(\mathbf{X}^{(i)}, y_i)\}_{i=1}^{k}$.
4: Compute the acquisition function a_k from s_k.
5: Identify $\mathbf{X}^{(k+1)} = \arg\max_{\mathbf{X} \in \mathcal{X}_{\text{set}}} a_k(\mathbf{X})$.
6: Evaluate $y_{k+1} = f(\mathbf{X}^{(k+1)}) + \epsilon_{k+1}$.
7: **end for**
8: **return** $\mathbf{X}^\dagger = \mathbf{X}^{(i)}$ if $y_i = \max_{j \in [T]} y_j$

Experiments

Experiments

- We test our method on the following circumstances:
  1. Synthetic functions;
  2. Clustering algorithm initialization for synthetic datasets;
  3. Clustering algorithm initialization for real-world datasets;
  4. Active nearest neighbor search for point clouds.
- We define the application-agnostic baseline methods, Vector and Split:
  - Vector: A standard Bayesian optimization is performed over a md-dimensional space;
  - Split: Individual Bayesian optimization strategies are executed on the m components comprising $\mathcal{X}$.

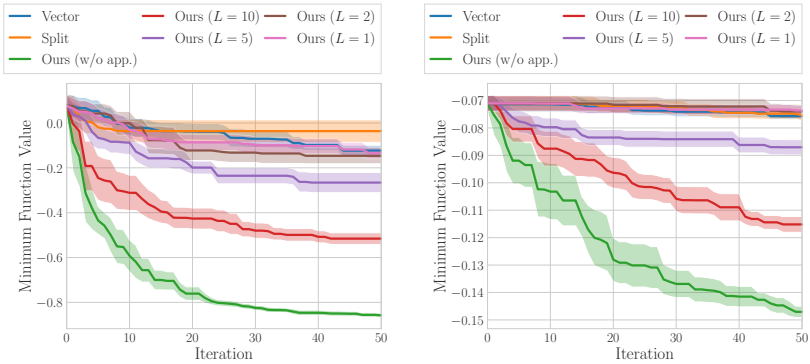
Experiments



(d) Vector (e) Split (f) Ours (w/o approx.)

Figure 1: Examples of one of the best acquisition results (i.e., purple stars indicate instances in the acquired set) via Vector, Split, and Ours (w/o approximation). For Synthetic 1 (first row) and Synthetic 2 (second row), m is set to 20.

Experiments

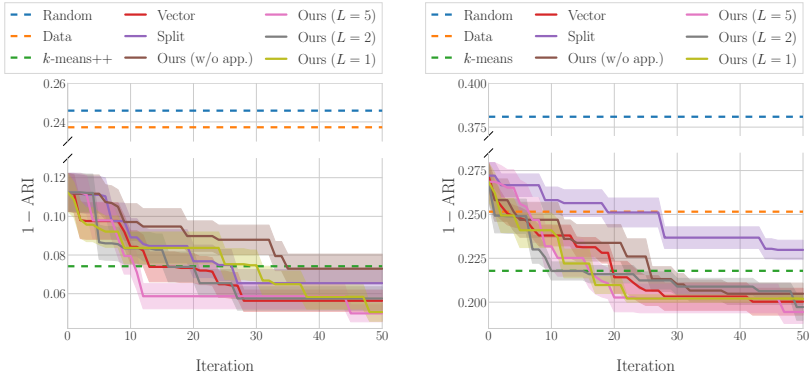


(a) Synthetic 1

(b) Synthetic 2

Figure 2: Results on optimizing two synthetic functions.

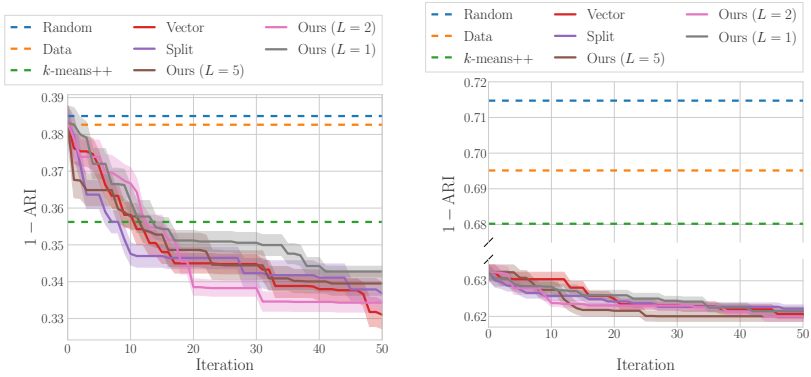
Experiments



(a) k-means clustering (b) Gaussian mixture model

Figure 3: Results on initializing clustering algorithms: k-means clustering and Gaussian mixture model for synthetic datasets.

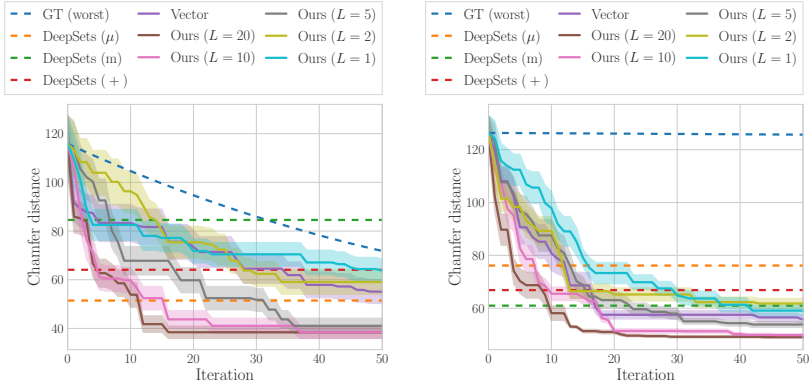
Experiments



(a) Handwritten Digits (b) NIPS Conference Papers

Figure 4: Results on initializing k-means clustering for Handwritten Digits and NIPS Conference Papers datasets.

Experiments

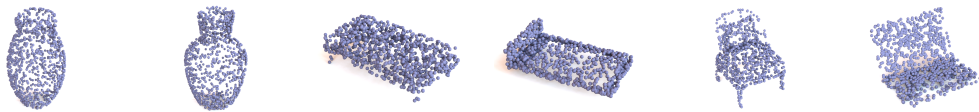


(a) 100 examples

(b) Entire examples

Figure 5: Nearest neighbor search results on ModelNet40 point clouds.

Experiments



(a) Query 1 (b) NN 1 (c) Query 2 (d) NN 2 (e) Query 3 (f) NN 3

Figure 6: Query and NN pairs are the query and its nearest neighbor examples found by our method.

Conclusion

Conclusion

- In this paper, we propose the Bayesian optimization method over sets, which takes a set as an input and produces a scalar output.
- Our method based on GP regression models a surrogate function using set-taking covariance functions, referred to as set kernel.
- We approximate the set kernel to the efficient positive-definite kernel that is an unbiased estimator of the original set kernel.
- Our experimental results demonstrate our method can be used in some novel applications for Bayesian optimization.

References I

P. I. Frazier and J. Wang. Bayesian optimization for materials design. In *Information Science for Materials Discovery and Design*, pages 45–75. Springer, 2016.

T. Gätner, P. A. Flach, A. Kowalczyk, and A. J. Smola. Multi-instance kernels. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 179–186, Sydney, Australia, 2002.

K. Kandasamy, W. Neiswanger, J. Schneider, B. Póczos, and E. P. Xing. Neural architecture search with Bayesian optimisation and optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 2016–2025, Montreal, Quebec, Canada, 2018.

K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

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

J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 25, pages 2951–2959, Lake Tahoe, Nevada, USA, 2012.

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.