UNIVERSITY OF CAMBRIDGE

Batched High-dimensional Bayesian Optimisation via Structural Kernel Learning

Zi Wang, Chengtao Li, Stefanie Jegelka, Pushmeet Kohli

R244: Large Scale Data Processing and Optimization
Pedro Sousa

Background

- A black-box function has no explicit formula and its internal process is unknown. We only observe **inputs** and **outputs**.
- Bayesian Optimisation (BO) optimises black-box functions that are expensive to evaluate.
- **Problem**: BO struggles in high-dimensional spaces because of computational and statistical challenges.
- **General Solution**: To decompose the high-dimensional space into subsets of dimensions.
- **Struggles**: Not assuming the dimensions of the subspaces before decomposition.

Inferring Latent Space

1. **Additive Structure**:
   - Additive decomposition of the GP kernel.
   - Decompose dimensional space into *M* latent groups, each containing a subset of dimensions.
2. **Generative Model**:
   - Defines the prior over the latent group assignments.
   - Dirichlet distribution to draw mixing proportions + Multinomial distribution for assigning dimensions to groups.

Inferring Latent Space (Cont.)

3. **Learning the Decomposition**:

   - By inferring the posterior distribution over the latent space group assignments using Gibbs sampling.

   - Gibbs = Markov Chain Monte Carlo method

4. **Joint Optimisation**:

   - Additive structure is learnt simultaneously with the optimization of the objective function.

The decomposition with the highest data likelihood is selected.

Diverse Batch Sampling

Batched Bayesian Optimization selects a batch of observations, since evaluations are parallelizable.

- Diversity in Sampling:
  - Ensure *diversity* in the selected batch of points.
  - Acquisition function updated to consider *diversity* when selecting points to be evaluated.
  - *Determinantal Point Processes* (*DPPs*) help favor diverse sets of points, making it unlikely to select similar points in a batch. This is enforced at the *m*-group level, rather than the full kernel.

Good balance of exploration (diversity) and exploitation (predicted).

Evaluating Decomposition Learning (Exp. 1)

Recovering Decompositions:

Task: For *D* input dimensions, randomly sample decompositions with a minimum of 2 groups and a maximum of 3 dimensions per group.

Results: The more data is observed, the more accurate the learned decompositions are. However, the higher the dimensions, the more data we need.

Table 1. Empirical posterior of any two dimensions correctly being grouped together by Gibbs sampling.

D \ N	50	150	250	450
5	0.81 ± 0.28	0.91 ± 0.19	1.00 ± 0.03	1.00 ± 0.00
10	0.21 ± 0.13	0.54 ± 0.25	0.68 ± 0.25	0.93 ± 0.15
20	0.06 ± 0.06	0.11 ± 0.08	0.20 ± 0.12	0.71 ± 0.22
50	0.02 ± 0.03	0.02 ± 0.02	0.03 ± 0.03	0.06 ± 0.04
100	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.02 ± 0.02

Table 2. Empirical posterior of any two dimensions correctly being separated by Gibbs sampling.

D \ N	50	150	250	450
2	0.30 ± 0.46	0.30 ± 0.46	0.90 ± 0.30	1.00 ± 0.00
5	0.87 ± 0.17	0.80 ± 0.27	0.60 ± 0.32	0.50 ± 0.34
10	0.88 ± 0.05	0.89 ± 0.06	0.89 ± 0.07	0.94 ± 0.07
20	0.94 ± 0.02	0.94 ± 0.02	0.94 ± 0.02	0.97 ± 0.02
50	0.98 ± 0.00	0.98 ± 0.00	0.98 ± 0.01	0.98 ± 0.01
100	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00

Evaluating Decomposition Learning (Exp. 2)

Effectiveness in Bayesian Optimisation:

Task: Evaluate learnt decompositions in terms of cumulative and simple *regret*.

Results: *Gibbs sampling* outperforms simpler methods. For higher dimension, sometimes it's even better than *Known* because of intelligent exploration-exploitation balance.

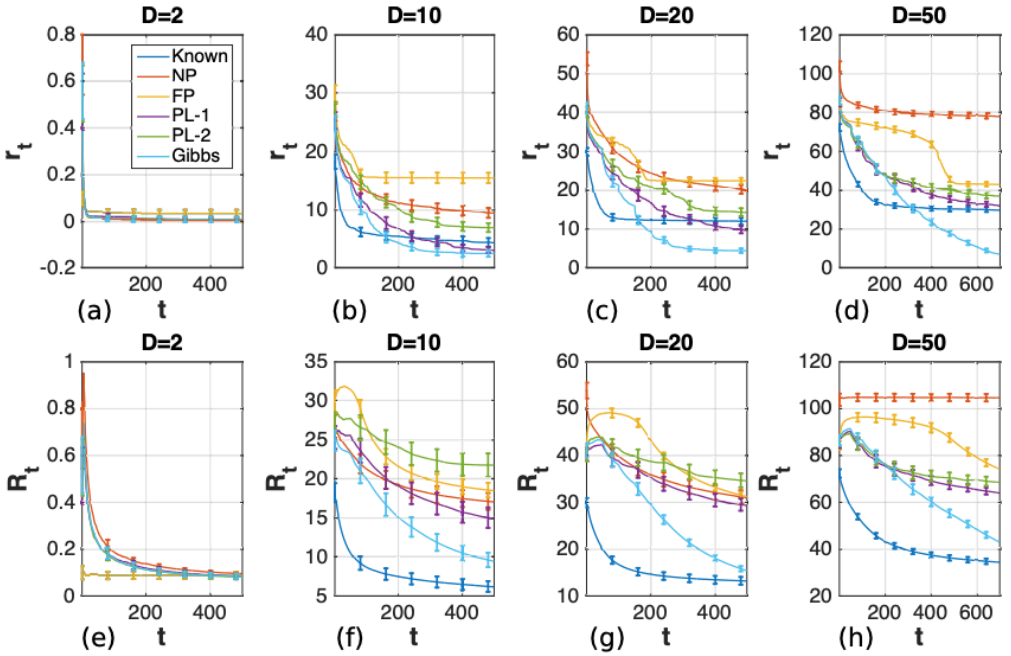

Figure 2. The simple regrets (r_t) and the averaged cumulative regrets (R_t) for setting input space decomposition with `Known`, `NP`, `FP`, `PL-1`, `PL-2`, and `Gibbs` on 2, 10, 20, 50 dimensional synthetic additive functions. `Gibbs` achieved comparable results to `Known`. Comparing `PL-1` and `PL-2` we can see that sampling more settings of decompositions did help to find a better decomposition. But a more principled way of learning the decomposition using `Gibbs` can achieve much better performance than `PL-1` and `PL-2`.

Evaluating Decomposition Learning (Exp. 3)

Real-world function:

Task: Two robot hands pushing objects towards a designated target location. The objective function is the distance between the target and the current object's position. The goal is to minimize this distance.

Results: *Gibbs sampling* outperforms all other alternatives, including Partial Learning (PL-1).

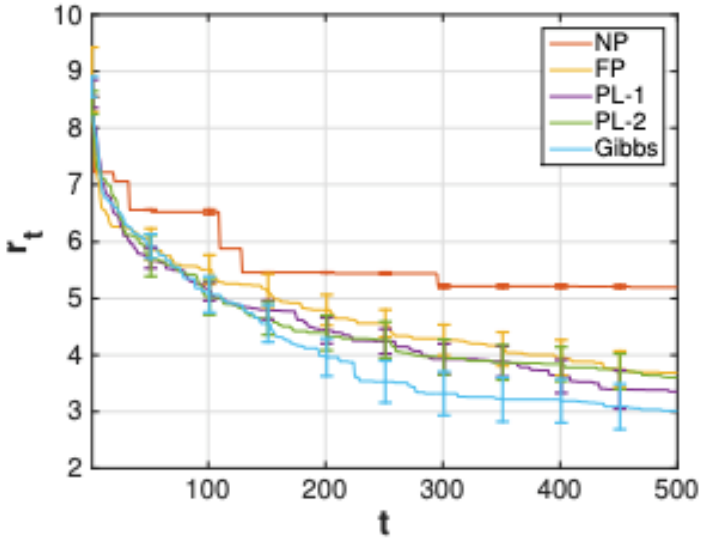

Figure 4. Simple regret of tuning the 14 parameters for a robot pushing task. Learning decompositions with `Gibbs` is more effective than partial learning (`PL-1`, `PL-2`), no partitions (`NP`), or fully partitioned (`FP`). Learning decompositions with `Gibbs` helps BO to find a better point for this tuning task.

Evaluating Diverse Batch Sampling (Exp. 1)

Effectiveness:

Task: Test the diverse batch sampling algorithms on the *Walker* function which returns the walking speed of a bipedal walker. 25 parameters and 40 points per dimension.

Results: A method that uses a selection by quality functions is the best performer. Rand, where batch points are chosen uniformly at random, is the worst.

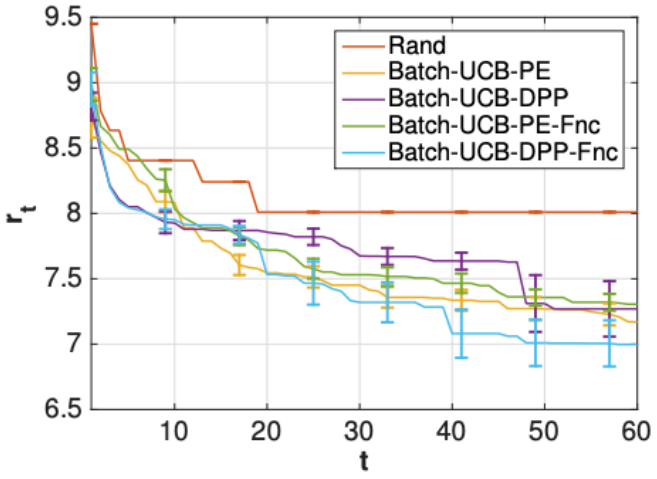

Figure 6. The simple regrets (r_t) of batch sampling methods on `Walker` data where $B = 5$. Four diverse batch sampling methods (`Batch-UCB-PE`, `Batch-UCB-DPP`, `Batch-UCB-PE-Fnc` and `Batch-UCB-DPP-Fnc`) outperform random sampling (`Rand`) by a large gap. `Batch-UCB-DPP-Fnc` performs the best among the four diverse batch sampling methods.

Pros

- Overall innovative method for handling high-dimensional spaces in Bayesian Optimization.
- Increase in efficiency by learning the additive kernel structure such that it adapts the function's decomposition based on the data. This contrasts previous work by Kandasamy et al. (2015) that used a "static" additive structure.
- Batch sampling is efficient for parallel evaluations.

Cons

- Computational Complexity:
  - Using Gibbs sampling may be computationally intensive, which could limit the method's applicability.
  - There are no reference in the experiments section to the computational cost of running this approach.
- The experiments also showed that this method is conditionate on the available data. The higher the dimension, the more data is needed.
- Lack of transparency on how certain benchmarks or comparisons are conducted. For example, how is the "Known" approach being optimized? What optimization technique is being used?

References

- Kandasamy, Kirthevasan, Schneider, Jeff, and Poczos, Barnabas. High dimensional Bayesian optimisation and bandits via additive models. In International Conference on Machine Learning (ICML), 2015.