Multiple Recommendation for Bayesian optimization via Multi-Scale Search

Tinu Theckel Joy, Santu Rana, Sunil Gupta, Svetha Venkatesh
Centre for Pattern Recognition and Data Analytics, Deakin University, Australia
{ttheckel, santu.rana, sunil.gupta, svetha.venkatesh}@deakin.edu.au

Abstract

Bayesian optimization operates sequentially recommending single evaluation setting each time. Many practical applications, however, have the facility of evaluating a batch of multiple settings simultaneously. Current batch Bayesian methods are mostly heuristic based and none have considered heteroscedasticity of the unknown objective function. We base our method on extracting the most promising batch of recommendations by searching across different smoothness assumptions which is realized through different length-scales of the Gaussian process covariance function. Theoretical analysis suggests that the proposed batch method has tighter regret bound than a pure sequential approach. Further improvement is brought by introducing a novel multi-armed bandit (MAB) based length-scale selection procedure, resulting in a more computationally efficient algorithm. We evaluate our method by minimizing three heteroscedastic benchmarked test functions and tuning the hyperparameters of two machine learning algorithms.

1 Introduction

Bayesian optimization has evolved as an efficient framework for global optimization of expensive objective functions [Mockus, 1994, Brochu et al., 2010b]. It is particularly useful when users have access only to the noisy functional evaluations of a black-box function. Recently, it has found applications in a variety of domains including gait optimization of robots [Lizotte et al., 2007], design of user interfaces [Brochu et al., 2010a], environmental monitoring [Marchant and Ramos, 2012] and tuning the hyperparameters of machine learning algorithms [Snoek et al., 2012].

Bayesian optimization adopts a sequential strategy for optimizing the objective functions. The pure sequential strategy, however, has some practical limitations. With the availability of parallel infrastructures, experiments can be conducted in parallel. In wet-lab experiments, users have duplicated set-ups for performing experiments in parallel. In hyperparameter tuning of machine learning models, multiple cores/machines can be used for training and validating the models. Parallel search, when effectively utilized, can potentially save significant time by reducing the number of iterations.

Different strategies have been proposed to recommend a batch of multiple settings for experimentation. Azimi et al. [2012] proposed a multiple recommendation method by collecting the first sample in the batch by optimizing the acquisition function and selecting other samples in the batch by generating fake observations using the Gaussian process model. González et al. [2016] locally penalized the maximum of the acquisition function to collect the elements in the batch. *These methods have two drawbacks: they effectively choose multiple maxima of the same acquisition function as the points to be explored, and all methods assume that the underlying variation is uniform and that the scale is known. Moreover, none of the methods have shown an improved convergence property associated with the batch recommendation.*

Scales are unknown and prone to be mis-estimated from small numbers of observations, particularly at the start. Additionally, many real world functions are often heteroscedastic. *In the context of*

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

batch Bayesian optimization, developing a theoretically guaranteed algorithm that is also suitable for heteroscedastic functions is still an open problem. Addressing this, we propose our method, Multi Scale Multiple Recommendation (MSMR).

2 Batch Bayesian optimization using Multi-scale Multi-recommendation

In Bayesian optimization, underlying function is probabilistically modeled using a Gaussian process prior with a zero mean function and a covariance function. Without loss of generality, the covariance function we choose is the Squared Exponential kernel. The critical parameter that controls the type of functions a Gaussian process can suitably model is the kernel length-scale. Practically, it may not be possible to infer the appropriate length-scales from small observational data in the beginning. This motivates us to investigate the use of combining more than one Gaussian processes with different length-scales.

To begin, let us sample a large number of length-scales $\theta_{1:n}$ within the range $\theta \in [\theta_L, \theta_U]$. We build a GP for each length-scale and subsequently find the most promising location to sample for the next iteration by optimizing their individual acquisition functions. Let us say that, we can only conduct m evaluations per iteration. The set of all candidate sample locations is then reduced to the required batch size m by finding the most agreed sample locations among them.

$$\min_{x^*_{c_k} \in \mathcal{X}, x^*_i \in x^*_{c_k}} ||x^*_i - x^*_{c_k}|| \tag{1}$$

where $x^*_{c_k}$ is the medoid of the cluster k, where $k = 1, 2, ..m$ and i denotes the index of samples and it varies from $i = 1, 2..n$. This is a clustering problem which can be approximately solved by k-medoids [Park and Jun, 2009].

2.1 Theoretical Analysis of Proposed Method

The main line of reasoning is based on Theorem 1 of Wang and de Freitas [2014], where the authors have proved that the EI [Mockus et al., 1978] based Bayesian optimization converges irrespective to the choice of a fixed length-scale. The theorem guarantees sub-linear growth in the cumulative regret as long as the length-scale is within a pre-specified bound $[\theta_L, \theta_U]$. This prima facie shows that for a pure sequential method where solutions are obtained based on different length-scales at different iterations, the optimization is still guaranteed to have sub-linear growth in cumulative regret. Using the ideology, "information never hurts", we show that our batch Bayesian optimization method, in fact, have a superior convergence than a pure sequential method.

We can think of MSMR as a principal sequence of one observation per batch and auxiliary data containing the rest of the batch. Without loss of generality let us also assume that the principal sequence contains the observations with the lowest regret. Essentially, MSMR would be having consecutive system with less variance due to the presence of the additional observations, compared to a pure sequential approach. Lemma 1 guarantees that. Rest of the reasoning in Lemma 2 and finally Theorem 1 culminates from the result of Lemma 1.

We use all the assumptions made by Wang and de Freitas [2014] in stating our Theorem. Assume multiple kernel length-scales θ_i are selected such that $\theta_L \leq \theta_i \leq \theta_U$.

Algorithm 1 MSMR

```
Bounds on $\theta$-$(\theta_L, \theta_U)$,
Initial $\theta_{1:n} \sim \mathcal{U}[\theta_L, \theta_U]$
$m$ - Number of recommendations
for $t = 1, 2, ..T$ do
    $\theta^*_{1:\eta}$= MAB_Scale($\theta_{i=1:n}, \eta, \mu^{'}_t, \sigma^{2'}_t$)
    $x^*_{i=1:\eta} = \arg\max_x \text{EI}(x|GP(D, \theta^*_{i=1:\eta}))$
    $x^*_{i=1:m}$= k-medoids$(x^*_{i=1:\eta}, m)$
    Evaluate $y^*_{1:m} = f(x^*_{1:m})$ in parallel.
    $D \leftarrow D \cup \{x^*_{1:m}, y^*_{1:m}\}$
end for
```

Lemma 1. $\max_{x \in \mathcal{X}} \sigma^{'2}_{t-1}(x; \theta_L) < \max_{x \in \mathcal{X}} \sigma^2_{t-1}(x; \theta_L)$ *where* $t > 1$, $\sigma^{'2}_{t-1}(x; \theta_L)$ *denotes variance in MSMR method and* $\sigma^2_{t-1}(x; \theta_L)$ *denotes the variance for a pure sequential approach.*

Proof. Intuitively this happens as the iterations of MSMR beyond the first time would have more observations and thus have a reduced maximum variance. The detailed proofs are available in a supplementary material[1]. □

Lemma 2. *$\gamma'^{\theta_L} \leq \gamma^{\theta_L}$, where γ'^{θ_L} denotes maximum information gain for MSMR method and γ^{θ_L} for the pure sequential method.*

Proof. Follows from previous Lemma and Lemma 7 of Wang and de Freitas [2014] (see supplementary material[1] for details). □

Theorem 1. *The cumulative regret for MSMR achieves a tighter upper bound than pure sequential optimization i.e.,*

$$\beta_T^{'} \sqrt{T\gamma'^{\theta_L}} < \beta_T \sqrt{T\gamma^{\theta_L}} \tag{2}$$

where $\beta_T' = 2\log(\frac{T}{\sigma^2})\gamma_{T-1}'^{\theta_L} + \varLambda_T + \sqrt{\gamma_{T-1}'^{\theta_L}} + C_2\|f\|_{\mathcal{H}_{\theta^{\mathcal{U}}(\mathcal{X})}}$ is for MSMR, $\beta_T = 2\log(\frac{T}{\sigma^2})\gamma_{T-1}^{\theta_L} + \varLambda_T + \sqrt{\gamma_{T-1}^{\theta_L}} + C_2\|f\|_{\mathcal{H}_{\theta^{\mathcal{U}}(\mathcal{X})}}$ is for a pure sequential approach, $C_2 := \prod_{i=1}^{d} \frac{\theta_i^U}{\theta_i^L}$, $\theta^L \leq \theta_t \leq \theta^U$, $t \geq 1$, $f(.) \in \mathcal{H}_{\theta^{\mathcal{U}}}(\mathcal{X})$ and $\varLambda_T = \sqrt{8}\log(\frac{T}{\sigma^2})\log^{1/2}(4T^2\pi^2/6\delta)\sqrt{C_2}\|f\|_{\mathcal{H}_{\theta^{\mathcal{U}}}}$.

Proof. The proof follows from Lemma 2 and it is straightforward to see that $\beta_T' < \beta_T$. Refer supplementary material[1] for further details. □

The bounds on cumulative regret for pure sequential Bayesian optimization is derived in Theorem 1 of Wang and de Freitas [2014] as $R_T = \mathcal{O}\left(\beta_T \sqrt{T\gamma^{\theta_L}}\right)$ with probability at least $1-\delta$. From our theorem, we can see that regret bound for our batch method is tighter. Hence our method with multiple recommendations is expected to have a faster convergence.

2.2 Selection of Optimal Length-Scales using Multi-armed Bandit (MAB) Formulation

We propose a multi-armed bandit formulation to devise a scheme that that favors the length-scales that have resulted in a good value of the function in the past iteration whilst, we still allow occasional selection of the length-scales for which not much observations are available. The observed reward for different clusters is calculated as,

$$r_{c_k,s} = f(x_{c_k}^*) - f(x^+) \tag{3}$$

where $x_{c_k}^*$ denotes the cluster median, k denotes the cluster index, s denotes the number of samples belonging to cluster c and x^+ is the current best available observation. The reward r denotes the amount of improvement in the function values of the collected samples x^* over the current best observation x^+. The length-scales are selected such that it offers the best expected reward at each iteration. The goal of optimization is to maximize the cumulative expected reward:

$$\arg\max \mathbb{E} \sum r(x^*/\theta) \tag{4}$$

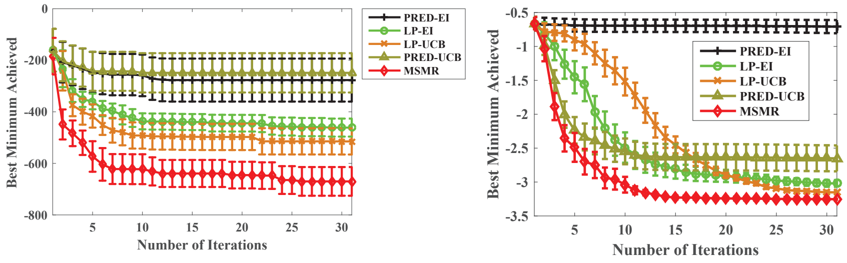
We assume that the rewards follow a Gaussian distribution $r \sim \mathcal{N}(\mu, \sigma^2)$. The optimization problem in Equation (4) can be solved using a strategy based on upper confidence bound, UCB [Auer, 2003] criteria. The overall algorithm for MSMR is presented in Algorithm 1.

3 Experiments

We conduct experiments on minimizing three benchmarked test functions for global optimization and later for tuning the hyperparameters of two machine learning algorithms. The baselines are EI and UCB variants of LP [González et al., 2016], PRED [Azimi et al., 2012]. For all the baselines, we used the implementations in GPyOpt[2].

[1] http://bit.ly/2f81DMI

[2] https://github.com/SheffieldML/GPyOpt



(a)

(b)

Figure 1: (a): Egg-Holder Function, (b): Hartman6 Function

Table 1: Average and the standard error of best minimum value for gSobol Function: #Dim denotes the dimension, #Rec denotes the number of recommendations in a batch

#Dim	**2**		**5**		**10**	
#Rec	**5**	**10**	**5**	**10**	**5**	**10**
EI	0.45(0.028)		66.75 (23.39)		2386.86 (722.24)	
UCB	0.44(0.065)		34.91 (9.01)		3287.32 (969.64)	
LP-EI	**0.3 (0.01)**	**0.27 (0.006)**	12.01 (3.38)	8.23 (1.68)	1166.48 (399.61)	1221.64 (742.76)
LP-UCB	**0.27 (0.01)**	**0.26 (0.003)**	18.03 (3.82)	17.04 (3.50)	1832.75 (996.34)	1155.76 (498.74)
PRED-EI	**0.28 (0.01)**	**0.27 (0.007)**	10.23 (2.54)	9.49 (2.14)	826.81 (278.31)	895.56 (312.42)
PRED-UCB	**0.29 (0.006)**	**0.27 (0.008)**	6.52 (1.87)	7.3482 (1.26)	218.85 (80.90)	56.53 (21.65)
MSMR	**0.26 (0.004)**	**0.26 (0.002)**	**1.14 (0.16)**	**2.24 (0.32)**	**8.56 (3.18)**	**17.98 (6.42)**

We evaluate different methods on the task of minimizing a highly heteroscedastic Egg-Holder function. The two dimensional Egg-Holder function has many local minima which makes it difficult to minimize. We run 30 iterations with a batch size of 5 per iteration. The results are averaged across 10 different initial settings. The average best found minimum value at each iteration with standard error is plotted in Figure 1a. Our method, MSMR, outperforms other baselines [Azimi et al., 2012, González et al., 2016]. MSMR is able to achieve a significant minimum value of the Egg-Holder function within 10 iterations. Similarly, experiment is repeated for another 6 dimensional multi-modal benchmarked test function, Hartman6. MSMR method outperforms all the other methods as plotted in Figure 1b. We also evaluate the performance of the different methods across varying dimensions and batch sizes on another test function, gSobol. In this experiment, EI and UCB denotes the pure sequential Bayesian optimization. Table 1 clearly shows that all the batch methods perform better than the pure sequential Bayesian optimization. As the dimension of the problem increases, MSMR performs better than all the other baselines [Azimi et al., 2012, González et al., 2016]. MSMR exhibits a stable performance even with higher batch size.

We tune four hyperparameters of Random Forest and six hyperparameters of Multi-Layered Perceptron (MLP) on two benchmarked real-world datasets. We run both the experiments for 30 iterations with 5 recommendations each iteration. All the methods are evaluated on the basis of the best model performance achieved within the allotted number of iterations. Average of the best achieved model performances with the standard errors across 5 different splits of train and test data are shown in Table 2. MSMR is able to find the best hyperparameter settings with minimum root mean squared error (RMSE) and test error.

Table 2: Average best model performance achieved with standard error

Methods	**MLP (% Test Error)**	**Random Forest (RMSE)**
	MNIST	**Protein Structure**
LP-EI	3.42 (0.12)	3.72(0.051)
LP-UCB	4.05 (0.33)	3.77(0.09)
PRED-EI	4.2 (0.55)	3.95(0.16)
PRED-UCB	4.16 (0.23)	3.78(0.11)
MSMR	**3.2 (0.12)**	**3.6(0.01)**

4 Conclusion

In this paper we proposed a novel theoretically guaranteed multiple recommendation algorithm for Bayesian optimization. A large set of promising sample locations is extracted from multiple Gaussian processes each having different smoothness assumption enforced by the choice of different kernel length-scales. This large set is then reduced to a smaller set to the size of the required batch size by finding the most agreeable sample locations. We further provide theoretical guarantee of the proposed method by deriving a tighter regret bound compared to the pure sequential approach. Further the efficiency of our method is improved by proposing a scheme based on multi-armed bandits to select a smaller subset of optimal length-scales at each iteration. Experiments demonstrate the superior performance of the proposed method compared to other batch methods.

References

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *The Journal of Machine Learning Research*, 3:397–422, 2003.

Javad Azimi, Ali Jalali, and Xiaoli Zhang-fern. Hybrid Batch Bayesian Optimization. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1215–1222, 2012.

Eric Brochu, Tyson Brochu, and Nando de Freitas. A Bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 103–112. Eurographics Association, 2010a.

Eric Brochu, Vlad M Cora, and Nando 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*, 2010b.

Javier González, Zhenwen Dai, Philipp Hennig, and Neil Lawrence. Batch Bayesian Optimization via Local Penalization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pages 648–657, 2016.

Daniel J Lizotte, Tao Wang, Michael H Bowling, and Dale Schuurmans. Automatic Gait Optimization with Gaussian Process Regression. In *IJCAI*, volume 7, pages 944–949, 2007.

Roman Marchant and Fabio Ramos. Bayesian optimisation for intelligent environmental monitoring. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 2242–2249. IEEE, 2012.

Jonas Mockus. Application of Bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4(4):347–365, 1994.

Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.

Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341, 2009.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.

Ziyu Wang and Nando de Freitas. Theoretical analysis of bayesian optimisation with unknown gaussian process hyper-parameters. *arXiv preprint arXiv:1406.7758*, 2014.