Information-based Acquisition for General Models in Bayesian Optimization

Siddhartha Jain [*]
MIT
sj1@mit.edu

Nathan Hunt [*]
MIT
nhunt@mit.edu

David Gifford
MIT
gifford@mit.edu

Abstract

We introduce the Hilbert-Schmidt Independence Criterion (HSIC) Acquisition Function (HAF), an acquisition function for Bayesian optimization that uses HSIC to measure the statistical dependency to a distribution of interest. This enables extensions of information theoretic acquisition functions (e.g. entropy search variants) for more general models than just Gaussian Processes (GPs). HAF is also differentiable, so points can be acquired via gradient search on the input space. On a protein-DNA binding task we compare a particular instance of HAF with Thompson Sampling and Expected Reward. Though preliminary results are not impressive, we identify a major issue with the model used in this task and suggest a future direction to improve upon this work.

1 Introduction

Bayesian optimization (BO) is a popular class of techniques to optimize a function $f(x)$ which we are allowed to query only a limited number of times and for which the derivatives are unavailable. f is generally non-convex and observations are usually corrupted by noise as $y = f(x) + \epsilon$. Such functions are common in robotics, machine learning, vision, biology, and many other areas [3, 4, 21, 16, 17, 20].

In BO, we use a model that outputs a predictive distribution for each x to approximate f. The posterior distribution, after observing training data, guides the queries we make to f. In particular we select points $\mathbf{x}$ that maximize some *acquisition function* (AF) $\alpha(\mathbf{x})$. A large amount of research has gone into figuring out what is a good α. Some popular acquisition functions include expected improvement (EI) [12], and Gaussian Process Upper Confidence Bound (GP-UCB) [18].

Recently, acquisition functions based on *information theory* have gained prominence. These include Entropy Search (ES) [9], and its variants Predictive Entropy Search (PES) [10] and Max-value Entropy Search (MES) [22]. The ES family of acquisition functions enjoys a strong theoretical motivation as well as good empirical performance. In practice, ES acquisition functions are used only for GPs where closed forms for the mutual information are available. Many real world processes have non-Gaussian observation noise or distributions (ex. RNA-seq read counts in biology, financial markets [6], radio signal based distance estimation [14], etc.).

As the ES metrics require computing the mutual information between candidate points and the distribution of interest, one option is to try to directly estimate the MI using just samples. However this is known to be hard to do [15, 7]. While there has been recent effort to improve mutual information estimation with learned estimators [2], in the Bayesian optimization setting, it would require training an estimator for each candidate point for every acquisition. Furthermore, such an estimator wouldn't be *differentiable*, making it hard to acquire points via gradient descent.

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

[*] Equal contribution

We propose using HSIC as an alternative way of measuring the dependence between the predictive distribution of a candidate point and the distribution of interest [8]. As we only need samples for this estimation, it can be applied to general models with non-analytic distributions; the target distribution can also be non-analytic.

Our contributions are as follows: (1) We introduce the HAF which uses HSIC to determine the dependence between the predictive distribution of a candidate point and any distribution of interest which might be informative about the optimal point. (2) We test HAF with the max-value distribution as the target. As this is an approximation of MES, we term this HAF-MES.

2 Background

We briefly introduce HSIC and our evaluation methods. The acquisition functions to which we compare are described further in Appendix A.

2.1 Hilbert-Schmidt Independence Criterion (HSIC)

Suppose we have two (possibly multivariate) distributions $\mathcal{X}, \mathcal{Y}$ and we want to measure the dependence between them. A well-known way to measure it is using *distance covariance* which, intuitively, measures the covariance between the *distances* between pairs of samples from the joint distribution P_{XY} and the product of the marginal distributions P_X, P_Y. HSIC can simply be thought of as distance covariance except in a kernel space.

Formally, given a kernel k for $\mathcal{X}$ and a kernel l for $\mathcal{Y}$, the HSIC between $\mathcal{X}, \mathcal{Y}$ is defined as follows

$$\begin{aligned} HSIC(P_{XY}, k, l) = {} & \mathbb{E}_{x,x',y,y'}[k(x,x')l(y,y')] \\ & + \mathbb{E}_{x,x'}[k(x,x')]\mathbb{E}_{y,y'}[l(y,y')] - 2\mathbb{E}_{x,y}[\mathbb{E}_{x'}[k(x,x')]\mathbb{E}_{y'}[k(y,y')]] \end{aligned}$$

where (x, y) and (x', y') are independent pairs drawn from P_{XY}. Note that $HSIC(P_{XY}) = 0$ if and only if $\mathcal{X}, \mathcal{Y}$ are independent.

See Appendix B for the empirical HSIC estimator we use.

2.2 Evaluation criteria

At each acquisition t, we compute the ratio of $\hat{y}_t^*$, the best point acquired so far, to the optimal point y^* which we term the regret ratio complement: $r_t = \frac{\hat{y}_t^*}{y^*}$ (complement in that $1 - r_t$ is the ratio of the simple regret to y^*). We look at both which methods have the best (largest) value for r_{100} as well as which have the largest area under the regret ratio complement curve (AURRCC) which shows how well they do on average for all r_t from $t = 1$ to $t = 100$.

We test our method on several discrete optimization problems involving the binding of short (8 base pair) DNA sequences to different peptides from the protein binding microarray (PBM) dataset [1]. The goal is to find the DNA sequences with the highest affinity for the target. The 10 randomly selected proteins that we test on are ARX, HOXD13, VSX1, POU4F3, MSX2, SIX6, BCL6, FOXC1, ISX, and PAX6. For each protein (each separate dataset) we run BO 8 times, each time with a different seed. Because the input space is finite and fairly small (about 32,000 unique inputs), we maximize acquisition functions through exhaustive evaluation on the entire input space instead of, e.g., by gradient ascent.

3 HSIC Acquisition Function (HAF)

Let $\hat{f}_t$ be our probabilistic model the current step t. Suppose we have a distribution P_D^t which is informative about the optimal point. For example P_D^t could be the distribution of the optimal location (as in PES); the distribution of the optimal value (as in MES); or the distribution of the best few values (generalizing MES). Intuitively, we want to find the point $\mathbf{x}_t$ to query which would give us the most information about this distribution.

Formally, the HAF is

$$\alpha_{HAF}(\mathbf{x})_t = \operatorname*{argmax}_{\mathbf{x}} HSIC(DX, k, l)$$

where DX are samples $(d_1, x_1), \dots, (d_m, x_m)$ from the joint distribution of P_D^t and $P_{\mathbf{X}}^t$ where $P_{\mathbf{x}}^t$ is the predictive distribution of $\mathbf{x}$ at iteration t as given by the model $\hat{f}_t$. k, l are the kernels for g, x respectively. For this paper, we take both k, l to be a mixture of rational quadratic kernels; see Appendix C for more details.

4 Probabilistic Model

We use a deep ensemble as outlined in [13] (without the adversarial smoothing) with a very simple base model consisting of a single 100-node hidden layer with ReLU activation as in [11]. This is to allow our model to be trained more effectively even on only a few acquired points. Our ensemble consists of 250 members so that we can get 250 samples from the joint distribution P_{DX}.

After each acquisition, we reset each model in the ensemble to a random initialization again and then retrain on all of the acquired data. We always train for 50 epochs.

5 HAF-MES

We use the max-value distribution as D, the target distribution, for this work. Because this is analogous to MES, we refer to this specific instantiation of HAF as HAF-MES.

In Algorithm 1 we describe how we acquire points using HAF-MES as the acquisition function for an ensemble of neural networks.

Algorithm 1 HAF-MES

```
Input:
    $D = \{d_1, \dots, d_m\}$: the maximum values for each member of the ensemble, where $m$ is the size of the ensemble
    $X = \{x^1, \dots, x^n\}$: the mean prediction of each ensemble member on each input point, where $n$ is the size of the search space and each $x^i$ is a vector of size $m$
    $k, l$: the kernels for $D, X$ (both the same rational quadratic kernel mixture)
$B = \{\}$
repeat
    $S_B = \{(d_i, x_i^1, \dots, x_i^{|B|})\}_{i=1}^m$
    $x' = \text{argmax}_{x \in X}\, HSIC(S_{B \cup \{x\}}, k, l)$
    $B = B \cup \{x'\}$
until $|B| =$ batch size
```

Note that for all results in this paper, we only use a batch size of 1.

6 Results

For each peptide-binding task, we compute r_t for $1 \leqslant t \leqslant 100$; figure 1 shows the curves this generates for one peptide. To summarize this information, as mentioned earlier, we look at which acquisition function does the best at the end and which is the best on average (as measured by the area under this curve). Table 1 shows the average ranking of the three acquisition functions tested.

It's apparent that our method doesn't provide the better-informed exploration for which we would hope. We believe this is largely because the uncertainty estimates of our model are calibrated particularly poorly on the high value points. Thus the empirical max-value distribution which HAF-MES is trying to gain information about places negligible probability on the true max value, so the information we gain is not particularly useful. Additionally, at least for these models and datasets, there is low correlation between the HSIC of a point and the reduction in uncertainty about

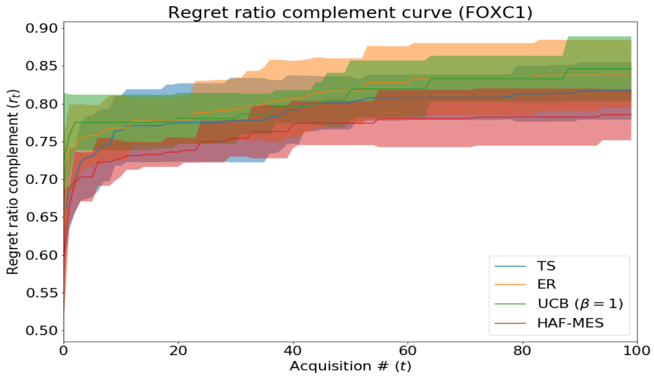

Figure 1: Regret ratio complement curves for three acquisition functions. The dark lines are mean values and the shaded regions an area of one standard deviation on either side of the mean. TS = Thompson sampling, ER = expected reward.

Table 1: Rankings of the acquisition functions (smaller is better) according to two metrics. "Last" is r_{100}. Mean values across the eight random seeds are used to compute the rankings.

Ac. Function	AURRCC-Rank	Last-Rank
ER	**2.2**	**2.2**
UCB ($\beta = 1$)	2.4	2.3
TS	**2.2**	2.5
HAF-MES (ours)	3.2	3.0

the max-value distribution from acquiring that point. Given the non-uniform costs of Bayesian optimization, where we care more about good predictions on high-value points than others, we hope that future work applying methods as in [5] to focus learning around high-utility points could improve the distributions around the high-value points and make HAF-MES more competitive.

7 Conclusion

We present HAF, a generic acquisition function which allows information-based acquisitions in models with non-analytic predictive distributions. We test HAF-MES, a specific instance of HAF that is analogous to MES, and discuss its shortcomings as presently implemented. We conclude with a direction for future work that we hope can improve the performance of HAF.

Acknowledgements

The authors would like to thank Dougal Sutherland and Jonas Mueller for many useful discussions about this project.

References

[1] Luis A Barrera, Anastasia Vedenko, Jesse V Kurland, Julia M Rogers, Stephen S Gisselbrecht, Elizabeth J Rossin, Jaie Woodard, Luca Mariani, Kian Hong Kock, Sachi Inukai, et al. Survey of variation in human transcription factors reveals prevalent dna binding changes. *Science*, 351(6280):1450–1454, 2016.

[2] Ishmael Belghazi, Sai Rajeswar, Aristide Baratin, R Devon Hjelm, and Aaron Courville. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

[3] 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*, 2010.

[4] Roberto Calandra, André Seyfarth, Jan Peters, and Marc Peter Deisenroth. An experimental comparison of bayesian optimization for bipedal locomotion. In *ICRA*, pages 1951–1958, 2014.

[5] Adam D Cobb, Stephen J Roberts, and Yarin Gal. Loss-calibrated approximate inference in bayesian neural networks. *arXiv preprint arXiv:1805.03901*, 2018.

[6] Rama Cont. Empirical properties of asset returns: stylized facts and statistical issues. 2001.

[7] Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Efficient estimation of mutual information for strongly dependent variables. In *Artificial Intelligence and Statistics*, pages 277–286, 2015.

[8] Arthur Gretton, Kenji Fukumizu, Choon H Teo, Le Song, Bernhard Schölkopf, and Alex J Smola. A kernel statistical test of independence. In *Advances in neural information processing systems*, pages 585–592, 2008.

[9] Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(Jun):1809–1837, 2012.

[10] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems*, pages 918–926, 2014.

[11] José Miguel Hernández-Lobato, James Requeima, Edward O Pyzer-Knapp, and Alán Aspuru-Guzik. Parallel and distributed thompson sampling for large-scale accelerated exploration of chemical space. *arXiv preprint arXiv:1706.01825*, 2017.

[12] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, Dec 1998.

[13] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[14] Henri Nurminen, Tohid Ardeshiri, Robert Piche, and Fredrik Gustafsson. Robust inference for state-space models with skewed measurement noise. *IEEE Signal Processing Letters*, 22(11):1898–1902, 2015.

[15] Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253, 2003.

[16] Philip A Romero, Andreas Krause, and Frances H Arnold. Navigating the protein fitness landscape with gaussian processes. *Proceedings of the National Academy of Sciences*, 110(3):E193–E201, 2013.

[17] 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.

[18] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[19] William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[20] Chris Thornton, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 847–855. ACM, 2013.

[21] Doniyor Ulmasov, Caroline Baroukh, Benoit Chachuat, Marc Peter Deisenroth, and Ruth Misener. Bayesian optimization with dimension scheduling: Application to biological systems. In *Computer Aided Chemical Engineering*, volume 38, pages 1051–1056. Elsevier, 2016.

[22] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. *arXiv preprint arXiv:1703.01968*, 2017.

[23] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016.

A Acquisition Functions

Let $\mathcal{D}$ be the set of datapoints observed so far. In general in Bayesian optimization, there is not a restriction that a point can only be acquired a single time. Because observations are noisy, it may be worth sampling an important point multiple times. In the specific setting of our test datasets, however, we only have a single label for each input. Thus, even though the underlying function is noisy, we only allow each point to be acquired once.

- **Thompson Sampling** (TS) [19]: $\alpha_{TS}(\mathbf{x}) = \operatorname{argmax}_{\mathbf{x}} \mathbb{E}[f(\mathbf{x})|\theta]$ where θ is a parameter set for your model drawn from the posterior distribution over the parameters ($\theta \sim p(\theta|\mathcal{D})$). For our specific model choice (ensemble of neural networks), sampling a set of parameters θ means sampling (uniformly at random, though one could also introduce a weighting scheme based on performance) a model from the ensemble.
- **Gaussian Process-Upper Confidence Bound** (GP-UCB) [18]: $\alpha_{UCB}(\mathbf{x}) = \operatorname{argmax}_{\mathbf{x}} \mu(f(\mathbf{x})) + \beta \cdot \sigma(f(\mathbf{x}))$ where μ, σ are the mean and standard deviation of the posterior predictive distribution and β is a hyperparameter. When $\beta = 0$, the acquisition function is also called *Expected Reward (ER).*
- **Max-value Entropy Search** (MES) [22]: $\alpha_{MES}(\mathbf{x}) = \operatorname{argmax}_{\mathbf{x}} I(\{\mathbf{x}, y\}; y^*|\mathcal{D})$ where I is the *mutual information* and y^* is the distribution of the *optimal value* under the current model.

B Empirical Estimates for HSIC

The empirical estimator of $HSIC(P_{XY})$ is

$$\begin{aligned} &HSIC(Z,k,l) = \\ &\quad \frac{1}{m^2}\sum_{i,j}^{m} k(x_i,x_j)l(y_i,y_j) + \frac{1}{m^4}\sum_{i,j,q,r}^{m} k(x_i,x_j)l(y_q,y_r) - 2\frac{1}{m^3}\sum_{i,j,q}^{m} k(x_i,x_j)l(y_i,y_q) \\ &\qquad = \frac{1}{m^2}\mathbf{tr}(KHLH) \qquad (1) \end{aligned}$$

where $Z = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ are a series of independent samples from P_{XY}, and m is the number of samples. $K, H, L \in \mathbb{R}^{m,m}$ where $K_{ij} = k(x_i, x_j), L_{ij} = l(y_i, y_j), H = I - \frac{1}{m}11^T$ and 1 is an $m \times 1$ vector of ones. K, L are the kernel distance matrices of X, Y, respectively, whereas H centers the data.

C Kernels

The choice of the kernel function can affect empirical performance, especially in extremely high dimensional settings, and it is not uncommon to learn a kernel (using a high capacity ML model like neural networks [23]) in such cases. However the dimensions we deal with in this paper are on the order of a few tens and thus we simply use a mixture of rational quadratic kernels for k which has been used successfully with kernel based statistical dependency measures in the past. The rational quadratic kernel is defined as

$$k_{RQ}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 + \frac{(\mathbf{x}-\mathbf{x}')^T(\mathbf{x}-\mathbf{x}')}{2al^2}\right)^{-a}.$$

The parameters we use are $\sigma = 1, l = 1$ and $a \in \{0.2, 0.5, 1, 2, 5\}$. We use the same kernel mixture for l as well. We did not perform any fine tuning of the kernel parameters and all our experiments in this paper use this kernel.