Fractional Binding in Vector Symbolic Representations for Efficient Mutual Information Exploration

P. Michael Furlong
Centre for Theoretical Neuroscience
University of Waterloo
Waterloo, Canada
michael.furlong@uwaterloo.ca

Terrence C. Stewart
University of Waterloo Collaboration Centre
National Research Council of Canada
Waterloo, Canada
terrence.stewart@nrc-cnrc.gc.ca

Chris Eliasmith
Centre for Theoretical Neuroscience
University of Waterloo
Waterloo, Canada
celiasmith@uwaterloo.ca

***Abstract*—Mutual information (MI) is a standard objective function for driving exploration. The use of Gaussian processes to compute information gain is limited by time and memory complexity that grows with the number of observations collected. We present an efficient implementation of MI-driven exploration by combining vector symbolic architectures with Bayesian Linear Regression. We demonstrate equivalent regret performance to a GP-based approach with memory and time complexity that is constant in the number of samples collected, as opposed to t^2 and t^3, respectively, enabling long-term exploration.**

***Index Terms*—mutual information sampling, Bayesian optimization, vector symbolic architecture, fractional binding**

I. Introduction

Mutual information (MI) is a standard objective function for quantifying curiosity when exploring [1], [2]. In this paper we use Bayesian Optimization as a framework for curiosity, and present an algorithm for MI-driven exploration that has time and memory requirements that are constant in the number of observations, improving on the t^3 (time) and t^2 (memory) requirements of Gaussian Process approaches to computing MI.

A common approach to informative sampling is to compute information gain using Gaussian Processes (GPs), *e.g.* [3]–[6]. However, computing the variance, necessary to compute MI, requires inverting a matrix that grows with the square of the number of sampled data points, t. Unbounded growth of memory, and the concomitant increase in time to evaluate sampling locations, is not compatible with long-term operations using systems with limited memory capacity.

To overcome the limitations of GPs, researchers have improved occupancy grid methods [7] with efficient algorithms for computing information gain [8], [9]. The complexity of these approaches tend to grow linearly in the number of grid cells. However, occupancy grids have constraints that GPs do not - there is a fixed resolution, and only points in the grid are modelled. These shortcomings can be ameliorated with irregular and adaptive representations (e.g., triangulated meshes or KD-trees), but they require additional machinery to represent the function domain, and increased memory to represent larger areas.

We present an algorithm that provides the benefits of both approaches. It has memory and time requirements that are constant with respect to the number of observations, and is linear in the number of candidate sampling locations, but is still defined for all points in the function domain. We achieved this by combining the concept of *fractional binding* in Vector Symbolic Architectures (VSAs) with Bayesian Linear Regression (BLR), to model uncertainty over the function domain.

VSAs are used in modelling cognitive processes [10]–[14]. Symbols are represented as vectors, and cognition is conducted through operations on those vectors. Binding is a key operation, where a new symbol, C, is created by binding two existing symbols, A and B, denoted $C = A \circledast B$, typically representing a slot-filler relationship between A and B.

Cognitive Semantic Pointers are a neurally implemented VSA for which the binding operator is circular convolution[1]. Integer quantities, atomic symbols (e.g., words), and structured representations (e.g. sentences), can be represented in Semantic Pointers through binding. To represent integer quantities, binding is iterated an integer number of times, denoted $S^k = S \circledast \ldots \circledast S$, where $k \in \mathbb{N}$ and $S \in \mathbb{R}^d$ is a fixed semantic pointer of dimension d, that we call an "axis pointer" or "axis vector".

Spatial Semantic Pointers (SSPs) extend Semantic Pointers to represent real numbers through the process of fractional binding [13], [15]. Fractional binding is implemented with the Fourier transform $S^x = \mathcal{F}^{-1}\{\mathcal{F}\{S\}^x\}$, where $x \in \mathbb{R}$ is the real value that is encoded through element-wise exponentiation of the Fourier transform of the axis pointer, S. Using SSPs allows us to make a connection between biological representation and information-theoretic models of curiosity. SSPs link to biology through modelling grid and place cells [16], [17], representations linked to an organism's location [18]. Further, the organization of spatial relationships may be used in other brain areas [19].

SSPs connect to information theoretic exploration through

[1]Plate [13] originally suggested circular convolution for a purely algebraic VSA.

the kernel induced by the dot product over SSP vectors. SSPs induce a sinc kernel function [20], and sinc kernels have been found to be efficient kernels for kernel density estimators [21], [22]. Vector representations that induce kernels can be used to make memory- and time-efficient kernel density estimators, as in the EXPoSE algorithm [23]. But where Schneider *et al.* [23] made a KDE, we are combining SSPs with Bayesian Linear Regression to approximate Gaussian Process regression.

Other approaches combine vector representations with BLR to improve computational efficiency. ALPaCA uses uncertainty over the vector space for meta-learning [24], [25]. Perrone *et al.* [26] project data into a vector space, for more computationally efficient Bayesian Optimization. However, these techniques require learning a projection from input data into a vector space. The advantage of SSPs is that the representation doesn't need to be learned, it can be designed, further improving efficiency.

In this paper we compare the performance of our algorithm, the Spatial Semantic Pointer Mutual Information Bayesian optimization (SSP-MI), to the Gaussian Process Mutual Information Bayesian optimization (GP-MI) algorithm developed in [6]. We empirically show that the regret achieved by the algorithm is at least as good as the GP algorithm, with time and memory complexity that is constant in the number of samples collected, t, as opposed to $\mathcal{O}(t^3)$ and $\mathcal{O}(t^2)$ for the GP-based method. The constant time and memory requirements of SSP-MI means that it is feasible to deploy this algorithm in limited hardware for long-duration operations.

II. Approach

Our exploration algorithm uses the Bayesian Optimization framework of Contal *et al.* [6], given in Algorithm 1. The objective is to find the sampling location, x^*, that maximizes a function $f(\cdot)$. The function domain is sampled to provide a set of candidate function sampling points, $\mathcal{X}$. The algorithm computes an acquisition function, given by $\mu_t(x) + \sqrt{\gamma_t + \sigma_t^2(x)} - \sqrt{\gamma_t}$, where $\mu_t(x)$ is the current estimate of $f(x)$, $\sigma_t^2(x)$ is the predicted variance of $\mu_t(x)$, and γ_t accumulates the predicted variance of previously observed locations. The highest scoring candidate sample location is selected for follow-up observations, which are used to update the algorithm that predicts $\mu_t(x)$ and $\sigma_t^2(x)$.

In the baseline algorithm, GP-MI, $\mu_t(x)$ and $\sigma_t^2(x)$ are provided by a Gaussian Process regression using a radial basis kernel function, operating on the raw input vectors, $x \in \mathcal{X}$. In our algorithm $\mu_t(x)$ and $\sigma_t^2(x)$ are provided by a BLR over the SSP representation of the points in $\mathcal{X}$. GP-MI was implemented using the GPy library [27]. For GP-MI, mean and variance are computed per [28, §6.4]:

$$\mu_t(x) = \mathbf{k}_{t-1}^T \Sigma_{t-1} \mathbf{y}_{t-1} \tag{1}$$

$$\sigma_t^2(x) = k(x,x) - \mathbf{k}_{t-1}^T \Sigma_{t-1} \mathbf{k}_{t-1} \tag{2}$$

where $\mathbf{k}_t = (k(x,x_1), \ldots, k(x,x_t))$ is the vector of kernel function evaluations between the test point x and all points in the data set collected up to observation $t \in \{0, \ldots, T\}$.

Algorithm 1 Mutual information Bayesian optimization

procedure MIBO($budget, f(\cdot), \mathcal{X}$)
  $\gamma \leftarrow 0$
  while $t < budget$ **do**
    $\mu_t(x), \sigma_t^2(x) \leftarrow agent.query(x)\ \forall x \in \mathcal{X}$
    $\phi_t(x) \leftarrow \sqrt{\gamma_t + \sigma^2(x)} - \sqrt{\gamma_t}$
    $x_t \leftarrow \arg\max_{x \in \mathcal{X}} \mu(x) + \phi_t(x)$
    $y_t \leftarrow f(x_t)$ ▷ Observe $f(\cdot)$ at x_t
    $agent.update(x_t, y_t)$
    $\gamma_t \leftarrow \gamma_t + \sigma^2(x)$
    $t \leftarrow t + 1$
  end while
end procedure

$\mathbf{y}_t = (y_1, \ldots, y_t)$ is the vector of previously collected function value observations, $y_t = f(x_t)$. After the observation (x_t, y_t), is collected, Σ_{t-1}, $\mathbf{k}_{t-1}$ and $\mathbf{y}_{t-1}$ are updated.

For SSP-MI candidate sample locations are transformed into SSPs, which are row vectors denoted $\psi(x)$. To fractionally bind vector-valued variables, we select a random axis pointer, S_i, for each dimension of the vector as in [15]. We then fractionally bind each axis pointer using the corresponding vector element of x, and then bind all of the vectors representing the individual axes:

$$\psi(\vec{x}) = \mathop{\circledast}_{i=1}^{n} S_i^{x_i/l} \tag{3}$$

l is a length scale parameter. In this work we use the same length scale parameter for all vector elements, although using one length scale per dimension would be reasonable. The length scale parameter is optimized by minimizing the L_2 error in the predicted observations:

$$\arg\min_{l \in \mathbb{R}^+} \sum_{i}^{t} \|(y_i - \mathbf{m}_t^T \psi(\vec{x}_i/l))\|_2 \tag{4}$$

The choice to minimize the prediction error instead of maximizing the log likelihood of the observations was arbitrary. The BLR parameters were updated online, per [28, §3.3]:

$$\Sigma_t^{-1} = \Sigma_{t-1}^{-1} + \beta \psi(x_t)^T \psi(x_t) \tag{5}$$

$$\mathbf{m}_t = \Sigma_t \Sigma_{t-1}^{-1} \mathbf{m}_{t-1} + \Sigma_t \beta \psi(x_t) y_t \tag{6}$$

The predicted mean and variance were computed:

$$\mu_t(x) = \mathbf{m}_{t-1}^T \psi(x) \tag{7}$$

$$\sigma_t^2(x) = \frac{1}{\beta} + \psi(x) \Sigma_{t-1} \psi(x)^T \tag{8}$$

Both algorithms are initialized with ten observations that are used to optimize hyperparameters. Initialization points were selected by randomly shuffling the candidate locations and using the first 10 points. For both algorithms the hyperparameters were only optimized on the initial 10 samples, and not modified afterwards.

A. Experiment

We tested the algorithms on the Himmelblau, Branin-Hoo, and Goldstein-Price functions, which were used as benchmarks in [6]. We scaled the functions to make the problem a maximization, to ensure GP hyperparameter fitting converged, and to get similar regret numbers to those reported in [6]. The functions were evaluated, without noise, over a restricted domain, with points spaced evenly along each axis to form a 100×100 grid. Agents were given a budget of 250 samples. The domain and scale factors are given in Table I.

Function	Domain	Scale
Himmelblau	$[-5, 5] \times [-5, 5]$	$-\frac{1}{100}$
Branin-Hoo	$[-5, 10] \times [0, 15]$	-1
Goldstein-Price	$[-2, 2] \times [-2, 2]$	$-\frac{1}{10^5}$

TABLE I: The tested functions, the evaluation domains, and the scale factors applied.

Algorithm performance was measured with regret averaged over samples taken. The regret at each time point is the difference between the function value at sampling location, x, and the maximum value of the function over the candidate sampling locations, $x^* = \arg\max_{x \in \mathcal{X}} f(x)$. The regret at time t, is $R_t = \frac{1}{t} \sum_{i=1}^{t} f(x^*) - f(x_i)$. We also recorded the running total of the time it took to predict $\mu_t(x)$ and $\sigma_t^2(x)$ over the set of candidate sampling points including, for SSP-MI, the one-time encoding of the points in $\mathcal{X}$ as SSPs.

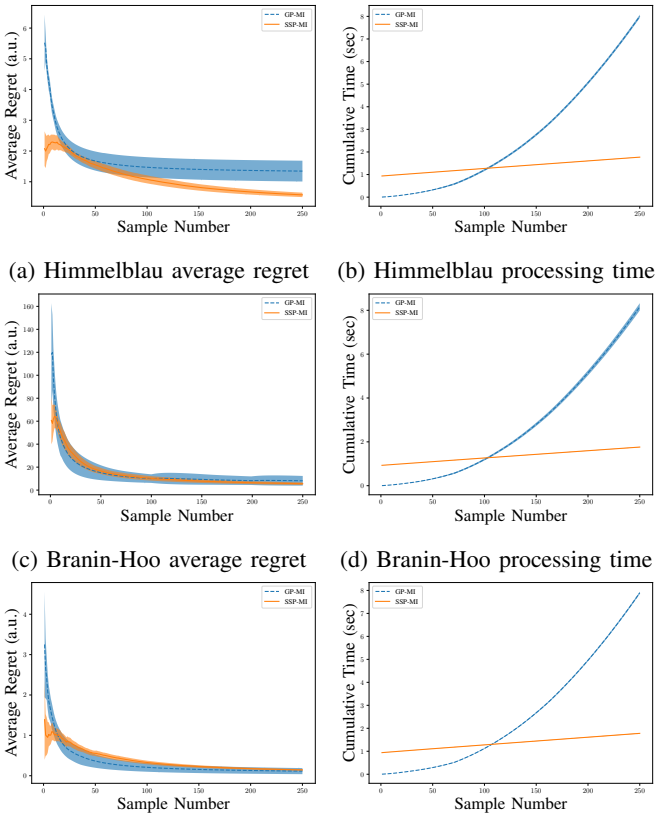
III. Results

Fig. 1 shows the evolution of the algorithms' regret on the left, and the cumulative time spent evaluating sampling locations on the right. Shaded regions represent a 95% confidence interval for the $N = 30$ trials.

Except for some initial samples, the algorithms' average regret is largely indistinguishable. Table II reports the difference in the means and in the standard deviation of R_t of the algorithms. Positive values mean the SSP-MI algorithm has lower regret, standard deviation. A Bayesian hypothesis test at samples 125 and 250, finds the performance of the SSP-MI algorithm is either better than or statistically indistinguishable from the GP-MI algorithm with 95% probability. Where there is a statistical difference, the effect size (Cohen's d) is moderate to large. For regret performance, under the tested scenarios, there is no reason to choose GP-MI over SSP-MI.

The benefits SSP-MI become apparent in the time to select sampling locations. At each iteration the algorithm recomputes the acquisition function for the candidate sampling locations. For GP-MI the time grows as a function of the number of samples collected. For SSP-MI the time to compute the acquisition function is constant in the number of samples collected, hence the observed linear trend in Fig. 1.

Of note is the large constant offset for the initial processing time. This is due to the time it takes to encode the candidate sampling locations as SSPs, $\{\psi(x) : x \in \mathcal{X}\}$. These values were cached so the encoding was only performed once.



(a) Himmelblau average regret (b) Himmelblau processing time

(c) Branin-Hoo average regret (d) Branin-Hoo processing time

(e) Goldstein-Price average regret (f) Goldstein-Price processing time

Fig. 1: Graphs on the left show the average regret, R_t, and graphs on the right the total accumulated processing time. Shaded regions are 95% confidence intervals for $N = 30$ trials. In all cases the R_t for SSP-MI is either the same as or statistically significantly better than the GP-MI algorithm. SSP-MI shows a substantial improvement in performance with respect to accumulated processing time.

IV. Discussion

We have demonstrated, using biologically plausible representations, MI-driven exploration that has fixed limits on memory and computation time while still being defined over continuous spaces. Combining Spatial Semantic Pointers and Bayesian Linear Regression enables operations with limited memory space and long-term operation in arbitrary spaces.

Our empirical regret performance is either statistically indistinguishable, or better than the baseline GP approach on three standard optimization targets. The time to evaluate candidate sampling locations is constant in the number of samples collected, unlike the GP-based approaches, and has a memory requirement $O(d^2)$, in the dimensionality of the SPP, compared to $O(t^2)$, in the number of observations, for GP-MI. Note also that our estimation of compute time in Fig. 1 conservatively includes the one-time encoding cost. While it takes over 100 samples to amortize this cost, the encoding could be done prior to the onset of operations, which

Function	t	$\mathbb{E}_{GP}[R_t] - \mathbb{E}_{SSP}[R_t]$		$SD_{GP}[R_t] - SD_{SSP}[R_t]$		Effect Size	
		μ	95% HDI	μ	95% HDI	μ	95% HDI
Himmelblau	125	1.01	[0.50, 1.61]	0.09	[-0.11, 0.34]	0.93	[0.44, 1.48]
	250	1.06	[0.51, 1.60]	0.09	[-0.10, 0.35]	0.98	[0.45, 1.48]
Branin-Hoo	125	2.42	[-0.07, 4.82]	3.54	[1.72, 5.74]	0.67	[0.02, 1.35]
	250	2.83	[0.94, 4.70]	3.03	[1.68, 4.47]	0.79	[0.25, 1.34]
Goldstein-Price	125	0.02	[-0.39, 0.41]	0.00	[-0.11, 0.11]	0.01	[-0.35, 0.42]
	250	0.02	[-0.37, 0.43]	0.00	[-0.12, 0.12]	0.02	[-0.37, 0.41]

TABLE II: The difference in regret measured at samples $t = 125$ and $t = 250$. SSP-MI regret is either better than or statistically indistinguishable from GP-MI, at the selected sample points. We report the differences in the average regret and the standard deviations, as well as the effect size (Cohen's d) for 30 trials. We also report 95% high density intervals (HDI). Positive values means the SSP algorithm has a lower regret or standard deviation. Results using an unpaired Bayesian hypothesis test [29].

would further favour SSP-MI. Like occupancy grid methods, evaluation time grows linearly with the number of candidate locations, but SSP-MI retains its definition over the continuum.

Our algorithm represents an initial proof-of-concept for curiosity guided exploration using vector representations. If SSPs are a unifying tool for modeling cognition, as in Eliasmith *et al.* [14], then our approach could also model curiosity in conceptual spaces. However, while we encoded data with SSPs, other vector encoding that induce kernels could be used.

There remain algorithmic refinements to explore. Hexagonal SSPs [17] could improve the efficiency of encoding candidate sample locations, and further integrate the work to neural models of spatial representation. Performance degradation in response to noise remains to be examined.

Because computing the acquisition function is efficient, it should be practical to find sample collection points, x, that maximize the acquisition function, instead of selecting from a finite set, avoiding regret due to arbitrary sampling of the function domain.

Further, we may be able to evaluate entire trajectories, not just individual sample locations. Single SSPs can represent trajectories and regions of a space, facts that may be exploitable to efficiently evaluate trajectories for informativeness and, through computing dot-products, feasibility in a configuration space. As our curiosity model is goal-driven, via the $\mu(x)$ term in the acquisition function, a variable weighting of expected reward and information gain could allow switching between task-driven exploration as well as something akin to play.

We have presented an efficient implementation of curiosity that may be of use in memory- and time-limited contexts. While preliminary, this work is a jumping-off point for efficient autonomous exploration.

ACKNOWLEDGEMENT

The authors would like to thank Nicole Sandra-Yaffa Dumont for discussions that helped improve this paper. This work was supported by CFI and OIT infrastructure funding as well as the Canada Research Chairs program, NSERC Discovery grant 261453, NUCC NRC File A-0028850.

REFERENCES

[1] D. V. Lindley, "On a measure of the information provided by an experiment," *The Annals of Mathematical Statistics*, pp. 986–1005, 1956.

[2] T. Loredo, "Bayesian adaptive exploration in a nutshell," *Statistical Problems in Particle Physics, Astrophysics, and Cosmology*, vol. 1, p. 162, 2003.

[3] A. Singh, A. Krause, C. Guestrin, W. J. Kaiser, and M. A. Batalin, "Efficient planning of informative paths for multiple robots," in *IJCAI*, vol. 7, 2007, pp. 2204–2211.

[4] D. R. Thompson and D. Wettergreen, "Intelligent maps for autonomous kilometer-scale science survey," 2008.

[5] K. Yang, S. Keat Gan, and S. Sukkarieh, "A gaussian process-based rrt planner for the exploration of an unknown and cluttered environment with a uav," *Advanced Robotics*, vol. 27, no. 6, pp. 431–443, 2013.

[6] E. Contal, V. Perchet, and N. Vayatis, "Gaussian process optimization with mutual information," in *International Conference on Machine Learning*. PMLR, 2014, pp. 253–261.

[7] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *IEEE/RSJ international conference on intelligent robots and systems*, vol. 1. IEEE, 2002, pp. 540–545.

[8] B. Charrow, S. Liu, V. Kumar, and N. Michael, "Information-theoretic mapping using cauchy-schwarz quadratic mutual information," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4791–4798.

[9] Z. Zhang, T. Henderson, S. Karaman, and V. Sze, "Fsmi: Fast computation of shannon mutual information for information-theoretic mapping," *The International Journal of Robotics Research*, vol. 39, no. 9, pp. 1155–1177, 2020.

[10] P. Kanerva, *Sparse distributed memory*. MIT press, 1988.

[11] ——, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive computation*, vol. 1, no. 2, pp. 139–159, 2009.

[12] T. Plate, "Holographic reduced representations: Convolution algebra for compositional distributed representations." in *IJCAI*, 1991, pp. 30–35.

[13] T. A. Plate, "Holographic reduced representations," *IEEE Transactions on Neural networks*, vol. 6, no. 3, pp. 623–641, 1995.

[14] C. Eliasmith, "How to build a brain: From function to implementation," *Synthese*, vol. 159, no. 3, pp. 373–388, 2007.

[15] B. Komer, "Biologically inspired spatial representation," 2020.

[16] E. P. Frady, P. Kanerva, and F. T. Sommer, "A framework for linking computations and rhythm-based timing patterns in neural firing, such as phase precession in hippocampal place cells," 2018.

[17] N. S.-Y. Dumont and C. Eliasmith, "Accurate representation for spatial cognition using grid cells," in *42nd Annual Meeting of the Cognitive Science Society. Toronto, ON: Cognitive Science Society*, 2020, pp. 2367–2373.

[18] E. I. Moser, E. Kropff, and M.-B. Moser, "Place cells, grid cells, and the brain's spatial representation system," *Annu. Rev. Neurosci.*, vol. 31, pp. 69–89, 2008.

[19] T. E. Behrens, T. H. Muller, J. C. Whittington, S. Mark, A. B. Baram, K. L. Stachenfeld, and Z. Kurth-Nelson, "What is a cognitive map? organizing knowledge for flexible behavior," *Neuron*, vol. 100, no. 2, pp. 490–509, 2018.

[20] A. R. Voelker, "A short letter on the dot product between rotated fourier transforms," *arXiv preprint arXiv:2007.13462*, 2020.

[21] I. K. Glad, N. L. Hjort, and N. G. Ushakov, "Correction of density estimators that are not densities," *Scandinavian Journal of Statistics*, vol. 30, no. 2, pp. 415–427, 2003.

[22] I. K. Glad, N. L. Hjort, and N. Ushakov, "Density estimation using the sinc kernel," *Preprint Statistics*, vol. 2, p. 2007, 2007.
[23] M. Schneider, W. Ertel, and G. Palm, "Expected similarity estimation for large scale anomaly detection," in *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1–8.
[24] J. Harrison, A. Sharma, and M. Pavone, "Meta-learning priors for efficient online bayesian regression," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2018, pp. 318–337.
[25] S. Banerjee, J. Harrison, P. M. Furlong, and M. Pavone, "Adaptive meta-learning for identification of rover-terrain dynamics," *arXiv preprint arXiv:2009.10191*, 2020.
[26] V. Perrone, R. Jenatton, M. Seeger, and C. Archambeau, "Multiple adaptive bayesian linear regression for scalable bayesian optimization with warm start," *arXiv preprint arXiv:1712.02902*, 2017.
[27] GPy, "GPy: A gaussian process framework in python," http://github.com/SheffieldML/GPy, since 2012.
[28] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
[29] J. K. Kruschke, "Bayesian estimation supersedes the t test." *Journal of Experimental Psychology: General*, vol. 142, no. 2, p. 573, 2013.