13th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP13
Seoul, South Korea, May 26-30, 2019

Bayesian optimization in effective dimensions via kernel-based sensitivity indices

Adrien Spagnol
PhD student, Safran Tech, Modelling & Simulation, Rue des Jeunes Bois, Châteaufort, 78114 Magny-Les-Hameaux, France

Rodolphe Le Riche
CNRS senior researcher, LIMOS at École des Mines de Saint-Etienne, Saint-Etienne, France

Sébastien Da Veiga
Expert research engineer, Safran Tech, Modelling & Simulation, Rue des Jeunes Bois, Châteaufort, 78114 Magny-Les-Hameaux, France

ABSTRACT:

A determining factor to the utility of optimization algorithms is their cost. A strategy to contain this cost is to reduce the dimension of the search space by detecting the most important variables and optimizing over them only. Recently, sensitivity measures that rely on the Hilbert Schmidt Independence criterion (HSIC) adapted to optimization variables have been proposed. In this work, the HSIC sensitivities are used within a new Bayesian global optimization algorithm in order to reduce the dimension of the problem. At each iteration, the activation of optimization variables is challenged in a deterministic or probabilistic manner. Several strategies for filling in the variables that are dropped out are proposed. Numerical tests are carried out at low number of function evaluations that confirm the computational gains brought by the HSIC variable selection and point to the complementarity of the variable selection and fill-in strategies.

1. INTRODUCTION

We are concerned with the global optimization of a function $f : \mathscr{X} \to \mathbb{R}$ on $\mathscr{X} \subset \mathbb{R}^D$, where we try to solve

$$x^* = \arg\min_{x \in \mathscr{X}} f(x) \, . \tag{1}$$

f may be a costly black-box function which neither has a known closed-form expression, nor accessible derivatives, and which involves intensive computations.

We rely on Bayesian Optimization (BO) as a state-of-the-art approach for expensive problems Moćkus et al. (1978); Jones et al. (1998); Srinivas et al. (2009). How the efficiency of BO scales with dimension is an ongoing research issue, as the number of points required to achieve a sufficient quality in surrogate model grows with input dimensions and global optimization in high-dimensional spaces is difficult. Different solutions were proposed in the literature to reduce dimension during optimization steps. The idea of decomposing an optimization problem into subproblems of lower dimension is as old as numerical optimization and can be related, in convex optimization, to conjugate directions or block coordinate descent Auslender (1976). Shan and Wang (2010) propose to reduce the dimension of the problem at the start of the optimization using Sobol sensitivity indices, at the risk of missing important sub-spaces or even the global optima. In Wang et al. (2013), the Random EMbedding Bayesian Optimization (REMBO) method projects

the high dimensional variables onto a low dimensional space by random linear combinations of the variables. Yet, the effective dimension d_e must be specified and this method may not work when d_e is underestimated. Li et al. (2018) takes up a popular idea from the machine learning community called the *Dropout*, where the optimization is carried out, at each iteration, on a randomly selected subset of the variables. Salem et al. (2018) proposes a dimension reduction algorithm called "Split and Doubt" that performs at each iteration a selection of the optimized variables based on their correlation length and sets the other variables to improve the surrogate model. However the criterion to select the variables is global and potentially not adapted to optimization.

In this paper, we select the significant variables using sensitivity indices specifically designed for optimization. They gauge which inputs matter to reach low objective function values with a kernel-based dependency measure Spagnol et al. (2018). We introduce and compare new strategies for dropping out and filling in optimization variables. All of them are iterative, some are deterministic, others probabilistic.

2. Bayesian Optimization

Bayesion optimization is based on a prior distribution on f which reflects our belief about the behaviour of the function, completed by a posterior distribution on f that accounts for the function observations. BO uses this posterior to choose where to sample the following points through the maximization of an *acquisition function*. Maximizing the acquisition function has a low cost in the sense that it does not involve new calls to f. Classically, we adopt a Gaussian Process distribution prior Rasmussen and Williams (2006) over the function f. We denote the observations by $\mathbf{X} = \left(x^1, \ldots, x^N\right) \in \mathscr{X} \subset \mathbb{R}^D$ and the corresponding evaluation values by $\mathbf{y} = (y_1, \ldots, y_N)$, with $y_i = f\left(x^i\right)$ for $1 \leq i \leq N$. The GP is assumed to be centered and we have

$$\mathbf{Y} \sim \mathscr{N}(0, \mathbf{K})$$

where $\mathbf{K} = k(x_i, x_j')$, for $1 \leq i, j \leq N$, is the covariance matrix that relates one observation to another. As for the kernel k, popular choices are the squared exponential kernel and the Matèrn kernel, depending on our belief of how smooth the objective function might be.

Using the Gaussian prior, each new value can be predicted at any x and the resulting posterior distribution is also Gaussian Rasmussen and Williams (2006)

$$y(x)|\mathbf{y} \sim \mathscr{N}(\mu(x|\mathbf{y}), \sigma^2(x|\mathbf{y}))$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ are the predictive mean and variance given by

$$\begin{aligned} \mu(x) &= k(x,X)^T \mathbf{K}^{-1} \mathbf{y} \\ \sigma^2(x) &= k(x,x) - k(x,X)^T \mathbf{K}^{-1} k(x,X) \end{aligned}$$

Here, $k(x,X)$ is the vector $\left(k(x,x^1), \ldots, k(x,x^N)\right)$. At each iteration of BO, the two previous quantities are computed as one uses it to derive the *acquisition* function, in order to determine where to sample next points. In our work, we consider the Expected Improvement acquisition function, whose closed-form has been derived Moćkus et al. (1978); Jones et al. (1998)

$$a^{\mathrm{EI}}(x) = \mathbb{E}(\max(0, y(x) - f(x^+)|\mathbf{y})$$

with $x^+ = \arg\min_{x \in \mathscr{X}} f(x)$, the point corresponding to the best objective value found so far. The next point to be sampled in the optimization process is

$$x^{N+1} = \arg\max_{x \in \mathscr{X}} a^{\mathrm{EI}}(x).$$

The Expected Improvement aims at a trade-off between exploitation (areas with low predictive mean) and exploration (areas with high predictive variance). Many BO methods exist, differing in the choice of the prior or the acquisition function. For example, the Upper Confidence Bound is the acquisition function used in Srinivas et al. (2009) and is defined as $a^{\mathrm{UCB}}(x) = \mu(x) - \sqrt{\beta}\sigma(x)$, where $\sqrt{\beta}$ is the exploitation-exploration trade-off parameter. Because it is multimodal, the closed-form acquisition function is typically optimized using a global optimizer such as DIRECT Jones et al. (1993) or CMA-ES Hansen and Ostermeier (2001).

3. Dimension reduction and the Dropout technique

Motivated by the Dropout algorithm in neural networks Srivastava et al. (2014), Li et al. (2018) cope with the high dimensionality issue in Bayesian Optimization by, at each iteration, randomly choosing $d \leq D$ optimization variables and fixing the $D-d$ other variables. Let $\mathbb{d}$ be the indices of the d selected dimensions and $\mathbb{D}\backslash\mathbb{d}$ the left-out ones. The corresponding variables are $x_{\mathbb{d}}$ and $x_{\mathbb{D}\backslash\mathbb{d}}$. A d-dimensional noisy Gaussian Process is used to model $f([x_{\mathbb{d}}, x_{\mathbb{D}\backslash\mathbb{d}}])$, $\forall x_{\mathbb{D}\backslash\mathbb{d}}$ where the multiplicity of values for a given $x_{\mathbb{d}}$ due to the freedom in $x_{\mathbb{D}\backslash\mathbb{d}}$ is considered as noise. By doing so, a predictive mean $\mu(x_{\mathbb{d}})$ and a variance $\sigma(x_{\mathbb{d}})$ can be computed and the authors naturally resort to the d-dimensional UCB acquisition function. Furthermore, they provide three fill-in strategies for the left out $D-d$ dimensions:

1. *Dropout-Random*: randomly draw in the domain at each iteration, $x_{\mathbb{D}\backslash\mathbb{d}} \sim \mathscr{U}(\mathscr{X}_{\mathbb{D}\backslash\mathbb{d}})$.
2. *Dropout-Copy*: use the observations giving the best function value so far $x^{+,N} = \arg\min_{N' \leq N} f(x^{N'})$, $x_{\mathbb{D}\backslash\mathbb{d}} = (x^{+,N})_{\mathbb{D}\backslash\mathbb{d}}$.
3. *Dropout-Mix*: use a mixture of both methods above. For each component independently, choose a random value with probability p or copy a component of the best-so-far solution with probability $1-p$. The authors empirically tune the value of p and choose $p = 0.5$.

Intuitively, the Dropout-Random is interesting when away from the global optimum as we do not have any information about the location of the minimum value, hence random guesses are appropriate. The Dropout-Copy should be preferred to a random choice if the best-so-far point is close to the true minimum of the function. In Li et al. (2018), the Dropout-Mix gives the best results as it allows to avoid staying in a local optimum for too long. However, the main drawback of the method is the fully random aspect of the variable dropout as the authors noted in their conclusion. As an improvement to this random dropout, we propose to base the choice of the active variables on a preliminary step of sensitivity analysis.

4. Sensitivity analysis for optimization

4.1. HSIC sensitivities

We begin by introducing the Hilbert Schmidt Independence Criterion, as proposed by Gretton et al. (2005) as it is the basis of our sensitivity index. Let $\mathscr{X}$ be any topological space where a Borel measure can be defined and $\mathscr{H}$ a Hilbert space of $\mathbb{R}$-valued functions on $\mathscr{X}$. Assume that $k : \mathscr{X} \times \mathscr{X} \to \mathbb{R}$ is the unique positive definite kernel associated with the Reproducing Kernel Hilbert Space (RKHS) $\mathscr{H}$. Further theory about RKHS can be found in Aronszajn (1950). Note that the kernel used here is not that of the GP model described in Section 2.

We also define the kernel mean embedding $\mu_{\mathbb{P}_X} \in \mathscr{H}$ of the distribution $\mathbb{P}_X$ by $\mu_{\mathbb{P}_X} := \mathbb{E}_X[k(X,\cdot)] = \int_{\mathscr{X}} k(x,\cdot) d\mathbb{P}_X(x)$, provided $\mathbb{E}_X[k(X,X)] < \infty$, Smola et al. (2007). Fukumizu et al. (2007) show that if k is characteristic, i.e. the mapping from all distributions on $\mathscr{X}$ onto their kernel mean embedding is injective, meaning each distribution has a unique representation in the RKHS, then all its statistical features are preserved.

Kernel embeddings of probability measures provide a distance between distributions through the distance between their embeddings in the Hilbert space. Such a distance is called the Maximum Mean Discrepancy (MMD) Gretton et al. (2012), and its squared form is

$$\begin{aligned}\gamma_k^2(\mathbb{P}_X, \mathbb{P}_Y) &= \|\mu_{\mathbb{P}_X} - \mu_{\mathbb{P}_Y}\|^2_{\mathscr{H}} \\ &= \mathbb{E}_X\mathbb{E}_{X'}[k(X,X')] + \mathbb{E}_Y\mathbb{E}_{Y'}[k(Y,Y')] \\ &\quad - 2\mathbb{E}_X\mathbb{E}_Y[k(X,Y)]\end{aligned}$$

for X' and Y' independent copies of X and Y, s.t. $X, X' \sim \mathbb{P}_X$ and $Y, Y' \sim \mathbb{P}_Y$.

Now, assume a random variable $X \sim \mathbb{P}_X$ on $\mathscr{X}$ and a RKHS $\mathscr{H} : \mathscr{X} \to \mathbb{R}$ with a kernel k. Similarly, let $Y \sim \mathbb{P}_Y$ be a random variable on $\mathscr{Y}$ and $\mathscr{G} : \mathscr{Y} \to \mathbb{R}$ be a RKHS with a kernel l. X and Y have a joint distribution $\mathbb{P}_{XY}$, whose kernel mean embedding is $\mu_{\mathbb{P}_{XY}} = \mathbb{E}_{XY}[v((X,Y),\cdot)]$, where v is the kernel on the product space $\mathscr{X} \times \mathscr{Y}$, $v((X,Y),(X',Y')) = k(X,X')l(Y,Y')$. The Hilbert-Schmidt Independence Criterion (HSIC) of X and Y is the squared MMD γ_v^2 between $\mathbb{P}_{XY}$ and the prod-

uct of its marginals $\mathbb{P}_X\mathbb{P}_Y$

$$\begin{aligned}\mathrm{HSIC}(X,Y)_{\mathscr{H},\mathscr{G}} &= \gamma_v^2(\mathbb{P}_{XY},\mathbb{P}_X\mathbb{P}_Y)\\ &= \|\mu_{\mathbb{P}_{XY}} - \mu_{\mathbb{P}_X\mathbb{P}_Y}\|^2_{\mathscr{H}\otimes\mathscr{G}}\\ &= \mathbb{E}_{X,Y}\mathbb{E}_{X',Y'}k(X,X')l(Y,Y')\\ &\quad + \mathbb{E}_X\mathbb{E}_{X'}\mathbb{E}_Y\mathbb{E}_{Y'}k(X,X')l(Y,Y')\\ &\quad - 2\mathbb{E}_{X,Y}\mathbb{E}_{X'}\mathbb{E}_{Y'}k(X,X')l(Y,Y')\end{aligned} \tag{2}$$

The latter is also the squared Hilbert-Schmidt norm of the cross-covariance operator associated with $\mathbb{P}_{XY}$ between RKHSs Gretton et al. (2005). For characteristic kernels, $\mathrm{HSIC}(X,Y)_{\mathscr{H},\mathscr{G}}$ is zero if and only if X and Y are independent.

The HSIC has been applied to the sensitivity analysis framework several times, see for example Da Veiga (2015). We use

$$S^{\mathrm{HSIC}}(X_i) = \mathrm{HSIC}(X_i, 1_{X\in\mathscr{L}}) \tag{3}$$

with $1_{X\in\mathscr{L}}$ the indicator function and $\mathscr{L}$ is defined as a region of interest, e.g. locations where the objective function value is below a certain threshold. Spagnol et al. (2018) shows that this sensitivity index is proportional to the squared MMD between the kernel mean embedding of $\mathbb{P}_{X_i}$ and $\mathbb{P}_{X_i|X\in\mathscr{L}}$ and it reflects how important a variable is in order to reach $\mathscr{L}$.

Given $(\mathbf{X},\mathbf{Y}) = \{(x^1,y^1),\ldots,(x^N,y^N)\} \sim \mathbb{P}_{XY}$, an empirical estimator of HSIC can be computed in $O(N^2)$ by replacing the expectation in Eq. (2) by their corresponding empirical expectations on $(\mathbf{X},\mathbf{Y})$ Gretton et al. (2005).

4.2. Implementation in an optimization algorithm: metamodels and normalization

The computation of the HSIC sensitivities requires to define $\mathscr{L}$, which corresponds to the sublevel of interest, i.e. the level we want to reach. We set $\mathscr{L} = \{X\in\mathscr{X}, f(X)\leq q_\alpha\}$, with q_α the α%-quantile. To avoid evaluating f when computing q_α and the HSIC sensitivities, we rely on evaluations of the predictive mean of surrogate model $\mu(\cdot)$ instead of the true function:

$$\hat{S}^{\mathrm{HSIC}}(X_i) = \mathrm{HSIC}(X_i, 1_{X\in\hat{\mathscr{L}}}) \tag{4}$$

with $\hat{\mathscr{L}} = \{X\in\mathscr{X}, \mu(X) <= \hat{q}_\alpha\}$ and $\hat{q}_\alpha$ the quantile computed on the predictive mean.

Since the HSIC sensitivities are positive norms, we apply a simple normalization to the previous indices as

$$\hat{S}_n^{\mathrm{HSIC}}(X_i) = \hat{S}^{\mathrm{HSIC}}(X_i) / \sum_{j=1}^{D} \hat{S}^{\mathrm{HSIC}}(X_j) \tag{5}$$

This allows us to be able to compare one value of index with another for varying X_i's.

5. DROPOUT GUIDED BY HSIC SENSITIVITIES

As an improvement over the random selection of optimization variables, we guide the selection with the normalized HSIC sensitivities which are recomputed at each optimization step. Two strategies are proposed:

- *Probabilistic strategy*: $d < D$ inputs are drawn, each with a probability $p_i = \hat{S}_n^{\mathrm{HSIC}}(X_i)$. As for the value of d, we follow the recommendations of Li et al. (2018) and choose $d = 5$ as it gives good results empirically,
- *Deterministic strategy*: only the variables with $\hat{S}_n^{\mathrm{HSIC}}(X_i) \geq \tau$ are kept, where τ is a threshold of detection. In this paper, $\tau = 1/D$ because it is the normalized sensitivity all variables would have if they all ranked equal.

Both methods will favor variables with a high sensitivity index, hence those detected as important to reach locations where the predictive mean of the Gaussian Process is low, assuming the surrogate model is a good representation of the objective function. The main difference lies in the number of variables kept as the probabilistic method activates a constant number of variables, d, whereas the deterministic approach activates a varying number of variables. Unlike the deterministic strategy, the probabilistic method can draw variables with almost-zero sensitivity indices. Because all groups of variables have a non-zero probability of becoming active in the long run, the probabilistic strategy, when coupled with a global optimization algorithm, is globally convergent. On the contrary, the deterministic approach may fail to accurately converge to the optimum on functions for which some variables always have $\hat{S}^{\mathrm{HSIC}}(X_i)$ smaller than the selection threshold τ (e.g., a quadratic function with a high aspect ratio).

For the dropped out dimensions, we rely on the fill-in strategies as introduced in Li et al. (2018) and recalled in Section 3. We define an additional method called Dropout-Gauss, which samples values for the $\mathbb{D}\backslash\mathbb{d}$ dimensions along a multivariate normal distribution based on the λ best observation points, $\lambda = N/2$, defined by $x_{\mathbb{D}\backslash\mathbb{d}} \sim \mathcal{N}(\mu_\lambda, \mathbf{C}_\lambda)$, where

$$\mu_\lambda = \frac{1}{\lambda}\sum_{i=1}^{\lambda} x_{\mathbb{D}\backslash\mathbb{d}}^{i:N} ,$$

$$\mathbf{C}_\lambda = \frac{1}{\lambda-1}\sum_{i=1}^{\lambda}(x_{\mathbb{D}\backslash\mathbb{d}}^{i:N} - \mu_\lambda)(x_{\mathbb{D}\backslash\mathbb{d}}^{i:N} - \mu_\lambda)^T .$$

$x_{\mathbb{D}\backslash\mathbb{d}}^{i:N}$ observed points ranked from best to worst. All the methods are summarized in Algorithm 1.

Algorithm 1 Bayesian optimization with Dropout guided by HSIC sensitivity indices

Input: $\{\mathbf{X}, \mathbf{y} = f(\mathbf{X})\}$
while $N < N_{\max}$ **do**
  Construction of the surrogate model $\mu(x)$ and $\sigma^2(x)$
  for $i = 1, \ldots, D$ **do**
    Calculate $\hat{S}^{\text{HSIC}}(X_i)$ on the predictive mean $\mu(x)$ (Eq. (4))
  end for
  $\mathbb{d}_N \leftarrow$ deterministic or probabilistic variable selection (Sec. 5)
  $x_{\mathbb{D}\backslash\mathbb{d}_N} \leftarrow$ fill-in by Random or Copy or Mix or Gauss strategies (Sec. 3 and 5)
  $x_{\mathbb{d}_N} \leftarrow \arg\max_{x^{\mathbb{d}_N} \in \mathcal{X}^d} a^{\text{EI}}(x_{\mathbb{d}_N} | x_{\mathbb{D}\backslash\mathbb{d}_N})$
  $x^{N+1} \leftarrow x_{\mathbb{d}_N} \cup x_{\mathbb{D}\backslash\mathbb{d}_N}$
  $y_{N+1} = f(x^{N+1})$
  $N \leftarrow N + 1$
end while

6. NUMERICAL TESTS

6.1. Experimental procedure

We test the different versions of our algorithm on a small benchmark of functions whose main features are described in Table 1. The function are classical optimization test cases that we chose for the diverse difficulties they bring. All the functions are defined in $\mathbb{R}^d$ and $D-d$ dummy variables are added to increase the dimensionality of the problem to D.

Table 1: Test functions.

Name	d	D	**Main features**
Branin	2	25	Multiple global minima
Ackley	6	20	Many local minima
Borehole	8	25	Nonlinear, physically based
Rosenbrock	5	20	Unimodal, solution in a curved valley

The HSIC indices are computed following Eq. (4) with $\alpha = 10\%$. Comparisons of the versions of Algorithm 1 are based on 20 repeated runs for each function. Each repetition starts with a Latin Hypercube Sampling optimized with a maximin criterion. The optimization budget is 50 calls to the objective function. We compare our algorithms with the Dropout algorithm of Li et al. (2018) (with $d = 5$ like in the probabilistic selection) and an EGO procedure Jones et al. (1998) where all D variables are optimized.

In the same spirit as in Hansen et al. (2016), the performance of the optimizers is measured by the frequency at which the algorithms are successful at solving tasks of varying difficulties: we set 3 goals per function (easy, medium and hard to achieve) and count the number of successes at each iteration in the repeated trials of each version of the algorithm. Easy, medium and hard goals are defined as the 90%, 50% and 10% quantiles of the final results of all algorithms for each function. An example is given by the horizontal red lines in Figure 1 for the Borehole function. For consistency in the comparisons, all algorithm versions utilize the same surrogate model, created with the package DiceKriging in R with a Matèrn $5/2$ kernel and a CMA-ES optimizer Hansen and Ostermeier (2001).

6.2. Discussion

Before any comparison between algorithms performance, we test the variable selection and both deterministic and probabilistic selections are able to efficiently pick out determining variables over the iterations, as can be seen in Figure 2. The dummy variables are kept at a significantly, yet non null, rate, which is mostly due to the approximation errors of the surrogate model ($\mu(x)$).

Plots in Figure 3 show the rate of success of each

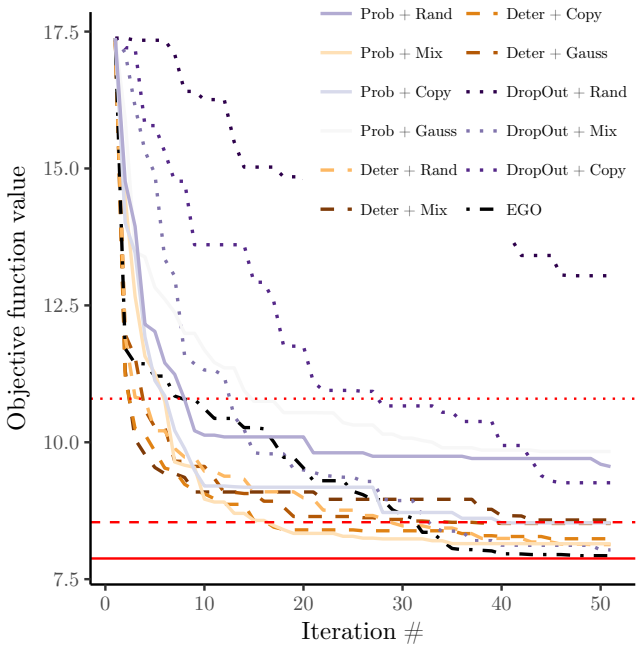

Figure 1: Median results of the different algorithms for the Borehole function. The red lines correspond to the easy, medium and hard goals (from top to bottom) for this test case.

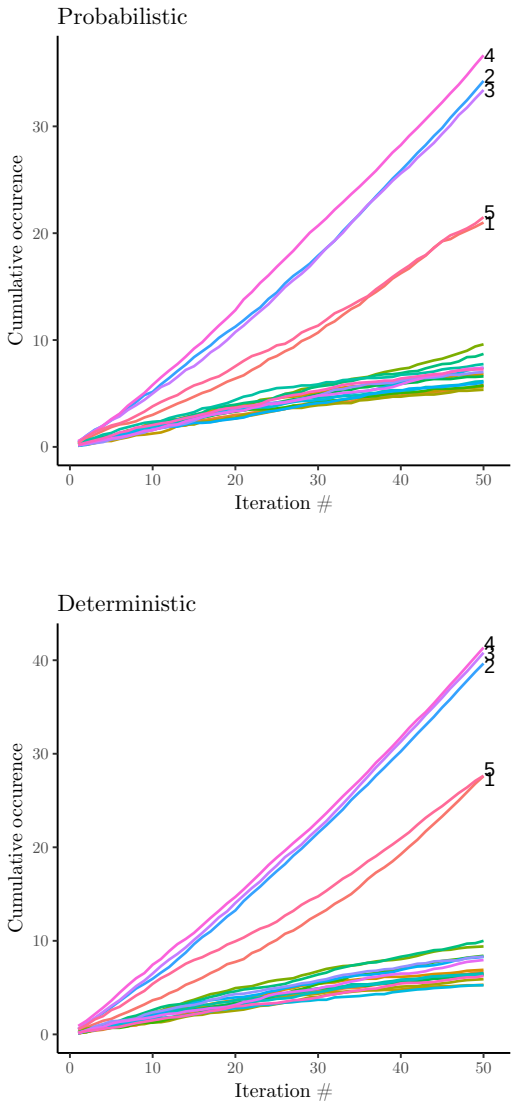

Figure 2: Average cumulative selection for each variable for the Probabilistic and Deterministic strategy with a Mix fill-in approach for the Rosenbrock function. The top 5 curves for each subplot correspond to the first five variables (the non-dummy ones).

algorithm averaged over all functions and all runs for the easy (A), medium (B) and hard (C) targets from left to right, respectively. When comparing all algorithms with EGO (no variable selection), it is seen that reducing the dimension allows visible gain for the medium and hard targets.

The fully random Dropout is consistently outperformed by the sensitivity guided versions. It confirms that better ways to choose the variables to be optimized over exist.

On the average, the probabilistic selection becomes better than the deterministic selection as the difficulty of the problems increases. This is especially visible with the deterministic selection and the Copy fill-in since it is not global: when the selected variables remain the same along the iterations and the dropped out variables have an impact on the function, this algorithm converges locally because it is unable to change the dropped out variables. In our tests, this is only visible with the hard targets as the medium and easy targets are attained with possible variable selection errors. Thus, for the easy and medium targets, coupling a deterministic dropout with a Copy fill-in performs well.

The Gaussian method for setting fixed variables behaves similarly to the Copy method. Despite its probabilistic formulation, it contributes to a premature convergence of the non-optimized variables towards the best observations. Therefore, it is appropriate for easy tasks and it is better coupled with the probabilistic dropout.

Finally, the algorithm composed of the probabilistic selection and the Mix fill-in outperforms all other algorithms for solving the medium and hard

problems. It is a good compromise between optimization, randomness and taking advantages of the best-so-far solutions. This shows that variable selection and fill-in strategies should be complementary.

7. CONCLUSIONS

This paper has studied improvements to the dimension reduction techniques that are adapted to Bayesian Optimization. At each iteration, HSIC sensitivity measures determine whether a variable is fixed or optimized. The overall algorithm combines a method to dropout variables and a method to fix their values. A new dropout in probability and a new Gaussian sampling for filling in variables have been described and compared to other pre-existing methods. Numerical comparisons averaging 4 functions and 3 levels of accuracy have shown that a good strategy is made of a probabilistic selection of the variables based on their HSIC sensitivity, coupled with a mixed random-copy of the best-so-far for filling in dropped out variables. Clear progress over random selection and fill-in are observed.

In this work, inputs were selected depending on their contribution in reaching a 10% quantile of performance on the surrogate model. A perspective is to consider other performance levels when selecting optimization variables, in particular levels going beyond already achieved performance.

8. REFERENCES

Aronszajn, N. (1950). "Theory of reproducing kernels." *Transactions of the American mathematical society*, 68(3), 337–404.

Auslender, A. (1976). *Optimisation: méthodes numériques*. Maîtrise de mathématiques et applications fondamentales. Masson.

Da Veiga, S. (2015). "Global sensitivity analysis with dependence measures." *Journal of Statistical Computation and Simulation*, 85(7), 1283–1305.

Fukumizu, K., Gretton, A., Sun, X., and Schölkopf, B. (2007). "Kernel measures of conditional dependence.." *NIPS*, Vol. 20, 489–496.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). "A kernel two-sample test." *Journal of Machine Learning Research*, 13(Mar), 723–773.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005). "Measuring statistical dependence with Hilbert-Schmidt norms." *International conference on algorithmic learning theory*, Springer, 63–77.

Hansen, N., Auger, A., Mersmann, O., Tusar, T., and Brockhoff, D. (2016). "COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. ArXiv e-prints, arXiv:1603.08785.

Hansen, N. and Ostermeier, A. (2001). "Completely derandomized self-adaptation in evolution strategies." *Evolutionary computation*, 9(2), 159–195.

Jones, D. R., Perttunen, C. D., and Stuckman, B. E. (1993). "Lipschitzian optimization without the lipschitz constant." *Journal of optimization Theory and Applications*, 79(1), 157–181.

Jones, D. R., Schonlau, M., and Welch, W. J. (1998). "Efficient global optimization of expensive black-box functions." *Journal of Global optimization*, 13(4), 455–492.

Li, C., Gupta, S., Rana, S., Nguyen, V., Venkatesh, S., and Shilton, A. (2018). "High dimensional bayesian optimization using dropout." *preprint arXiv:1802.05400*.

Moćkus, J., Tiesis, V., and Źilinskas, A. (1978). "The application of bayesian methods for seeking the extremum. vol. 2.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. The MIT Press, Cambridge, MA, USA, <http://mitpress.mit.edu/026218253X> (01).

Salem, M. B., Bachoc, F., Roustant, O., Gamboa, F., and Tomaso, L. (2018). "Sequential dimension reduction for learning features of expensive black-box functions.

Shan, S. and Wang, G. G. (2010). "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions." *Structural and Multidisciplinary Optimization*, 41(2), 219–241.

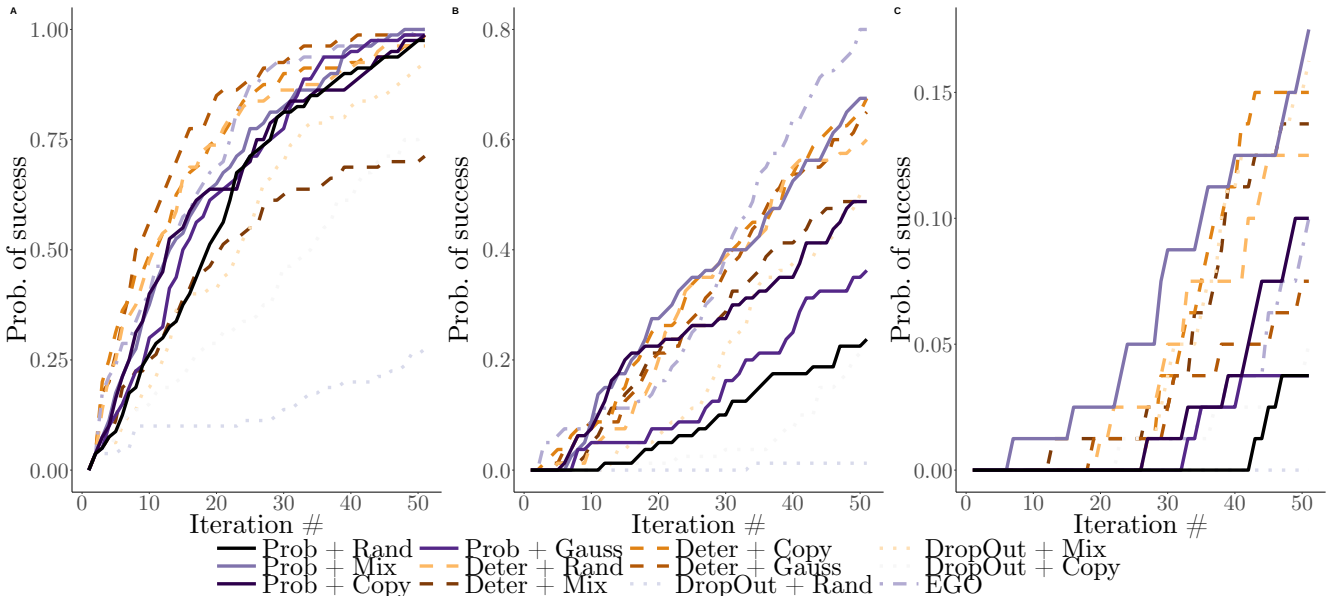

Figure 3: Summary of the average rate of success on all the benchmark functions of each algorithm for the easy (A), medium (B) and hard (C) goals. Solid lines are the probabilistic versions while the dashed lines are deterministic.

Smola, A., Gretton, A., Song, L., and Schölkopf, B. (2007). "A hilbert space embedding for distributions." *International Conference on Algorithmic Learning Theory*, Springer, 13–31.

Spagnol, A., Le Riche, R., and Da Veiga, S. (2018). "Global sensitivity analysis for optimization with variable selection." *arXiv preprint arXiv:1811.04646*.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2009). "Gaussian process optimization in the bandit setting: No regret and experimental design." *preprint arXiv:0912.3995*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). "Dropout: a simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research*, 15(1), 1929–1958.

Wang, Z., Zoghi, M., Hutter, F., Matheson, D., De Freitas, N., et al. (2013). "Bayesian optimization in high dimensions via random embeddings.." *IJCAI*, 1778–1784.