APPROACHES AND APPLICATIONS OF ARTIFICIAL INTELLIGENCE TO PROCESS OPTIMIZATION

A PREPRINT

Scott C. Riggs, Ph.D.
Founder, Find What Matters
San Francisco, CA
scr@findwhatmatters.ai

July 15, 2025

ABSTRACT

I explore the use of artificial intelligence (AI), specifically Bayesian optimization techniques across a variety of domains. I discuss the conditions under which machine and deep learning methods are most successful in discovering underlying relationships, as well as applications that are well-suited for Bayesian-AI treatment. I also discuss the importance of domain-specific knowledge and the need for data preprocessing and cleaning in order to achieve the best results.

Keywords Data science · Machine learning · Deep learning · AI · Artificial intelligence

1 Introduction

Artificial Intelligence (AI) is an interdisciplinary field that generally leverages the tools of statistics, mathematics, computer science, and subject-specific domain knowledge (among other tools) to reveal underlying relationships between control attributes (i.e., input features) and predictive outputs (i.e., output targets) in light of provided examples (i.e., based on data). Some of the conditions that are most fortuitous for the success of the data scientific method in discovering underlying relationships include (Tan et al., 2018):

- data sets that are rid of defects, such as transcription errors, measurement errors, outliers, missing values, and inconsistencies;
- data sets in which the experimental output measurement values are large in comparison to the stochastic fluctuation scale associated with observation (i.e., low noise / high reliability);
- data sets that contain sufficient observations to exhaustively span the relevant input space that will be analyzed and/or optimized;
- data sets that are representative of the underlying relationship (i.e., data sets that do not suffer from sampling bias);
- data sets comprised of experiments that are independent and identically distributed (i.e., there are no differences in the experimental environment under which each experiment is undertaken);
- underlying relationships that are constant in time (i.e., the underlying patterns are not dominated by time-dependent systematics associated with data collection time but rather reflect a genuine relationship between inputs and outputs);
- underlying relationships in which the outputs are determined by only a few input features (i.e., low dimensionality);
- underlying relationships in which the outputs are simple transformations of their determinative input features (e.g., linear functions of the control attributes);

- underlying relationships in which the outputs are simple transformations of transformed representations of their determinative input features (e.g., linear functions of the exponential of one or more control attributes); and
- underlying relationships in which the functional forms of the relationships are known *a priori* (e.g., from the laws of physics, chemistry, or biology; empirical relationships observed in the fields of sociology, psychology, or economics).

In practice, no AI application fulfills all of these conditions perfectly; the extent to which these conditions are violated ultimately affects the efficacy of the data scientific method in revealing the underlying relationships present in the data. Even in applications in which one or more conditions are strongly violated, however, the situation can often be salvaged by compensation along the other conditions (e.g., applications involving high dimensional, nonlinear relationships can still be revealed via increased numbers of low-noise observations).

Beyond the abstract conditions under which the data scientific method can be expected to be leveraged successfully, however, a separate matter is the kind of application that is *well-poised* for a data scientific treatment. Although it is difficult to be precise and exhaustive, some desiderata for promising AI applications include:

- **Leverageable domain knowledge specific to the particular application** Application-specific knowledge may support AI approaches at all stages of modeling and optimization, including during the course of data acquisition (e.g., guiding the selection of sampled data points, determining the extent of the sampling hypervolume, incorporating known correlations between inputs and/or outputs during the initial sampling stage), data preprocessing / data cleaning (e.g., identifying the presence of transcription errors, outliers, and experimental failure modes), feature engineering (e.g., identifying exhaustive and non-redundant control attributes; identifying relevant control conditions, including appropriate metadata), model selection (e.g., invoking a parametric functional form derived from laws of nature or observed scaling relations), model training (e.g., favoring and/or constraining models to reflect known causal relationships; enforcing that similar data points, i.e., data points that are "close" under a measure of input attribute distance and/or belong to the same cluster according to the underlying data generating process, are mapped to similar output target values), and optimization (e.g., specifying the quantitative form of an objective function in order to rank candidate data points according to their estimated utility under a model) (Childs and Washburn, 2019).
- **Availability of a high quality data set** As discussed above, the quality of a data set is multi-faceted and includes aspects such as the number of independent data points collected (in particular, in relation to the number of model parameters to be fit and the complexity of the underlying patterns that are to be revealed), the distribution of measured data points relative to a target optimized region, the capture of relevant control attributes, the absence of transcription errors and outliers, and the low-noise / high-reproducibility measurement of predictive output targets.
- **An underlying process that is well-approximated as a "black box" data generator** A black box data generator returns values sampled from a function with unknown structure; in particular, information about the underlying function is gained *exclusively* from sequential queries of the black box. Returned values from the black box may further be corrupted by stochastic noise, also of an unknown form. In situations in which data sampling is unconstrained (i.e., experimental data points can be collected inexpensively or a low-computational-cost / high-fidelity simulator exists), an empirically-driven AI approach may inefficiently optimize the underlying process in comparison to algorithms that explicitly leverage high frequency sampling (Droste et al., 2002). Other situations (e.g., ones in which experimental observations are expensive to obtain, ones in which the underlying data generating process is complex and difficult to simulate from first principles, ones in which the underlying relationships are nonlinear functions of many control attributes) are more conducive to AI approaches.

A specific class of applications that satisfies these desiderata is complex process optimization, where the goal is to achieve precise (e.g., atomic-level, cellular-level, etc.) control over the properties of a final material or product Skarlinski et al., 2024. Such processes often involve fabricating intricate multi-dimensional arrangements of constituent materials. This level of control typically requires a sequence of tightly-coupled physical and chemical transformation steps. These interactions are highly complex, often comprising the creation of reactive species, their transport to a substrate, a series of surface interactions like adsorption and chemical bonding, and the subsequent removal of product species, all while managing unwanted side effects like the re-deposition of byproducts Taylor et al., 2022.

These processes often involve a variety of complex kinetics and reaction mechanisms that span a vast range of length scales Chaves et al., 2024. The system's environment is frequently highly non-linear, with many components existing in non-equilibrium thermodynamic states Shaw et al., 2024. The system dynamics can be non-local, where energy imparted in one location affects interactions at remote locations, and the process outcomes can depend sensitively on

small changes in control parameters. A full characterization of such a process from first principles is often intractable, as it would require precise knowledge of the spatio-temporal distributions and energy states of all constituent species, their production and loss rates, the effective field distributions, and the rates of all surface chemical reactions under dynamic bombardment by various particles (Jones, 2025).

These fundamental complexities compound the challenge of deploying process optimization routines in a high-volume manufacturing environment, as the remedies for variation across nominally identical tools are difficult to uncover (i.e., the process transfer and matching problem) (Megat et al., 2023). While significant progress has been made in modeling such processes from first principles, these efforts inevitably involve simplifying assumptions that trade modeling fidelity for computational feasibility. As an alternative, black-box methods can be employed Hamilton et al., 2024. The efficacy of these efforts depends primarily on the availability of a high-quality central database and secondarily on strategies for digesting that data into leverageable predictive models. In many mature R&D-intensive fields, the requisite investment in data generation has likely already been made over decades of experimentation; future market value will be captured by entities who invest in the data engineering required to warehouse that data in an easily queryable, high-quality central database.

In this paper, a variety of strategies for enhancing the efficacy of AI approaches are discussed. While the strategies described are general, specific reference will sometimes be made to domain specific process optimization (e.g., materials discovery, small-molecule discovery, assay development, etc.) in order to ground the discussion in a concrete example that the authors believe is well-suited for AI techniques. Section 2 discusses strategies for conducting designs of experiment, with the intent of highlighting general aspects of generated data sets that hasten the optimization rate under a data-science-driven approach. Section 3 reviews strategies for using the acquired data (at any stage of the data acquisition process) in order to construct predictive models that can be leveraged for optimization. As will be emphasized, the modeling strategies employed assume that the underlying data generation process is a black box, and so the modeling strategies employed must have the capacity to fit to arbitrary relationships (i.e., the parametric model forms must be universal function approximators) without overfitting to latent noise of an unknown form. Finally, Section 4 presents strategies for using models in tandem with desired target objectives in order to recommend subsequent experiments; in particular, the discussion will focus on recommendation strategies that employ a suitable objective function in order to quantitatively compare candidate experiments, perform utility maximization (equivalently, objective function minimization) in order to propose candidate experiments, and iteratively update models based on results from those optimized candidates.

2 Sampling Strategies (i.e., Design of Experiments)

In order to employ AI strategies for optimization, one can begin by formulating the optimization as the solution to a suitable minimization problem. The mathematical formulation requires defining a suitable mathematical notation, so let us define:

- $\vec{\boldsymbol{x}}$ as each individual experiment; $\vec{\boldsymbol{x}}$ is a d-dimensional vector (i.e., there are d input control attributes), and so each experiment $\vec{\boldsymbol{x}}$ belongs to the d-dimensional vector space $\mathbb{R}^d$;
- $\mathcal{X}$ as the input feature space (i.e., $\mathcal{X}$ is the entire hypervolume of possible experiments); $\mathcal{X}$ is generally a subspace of $\mathbb{R}^d$;
- $\vec{\boldsymbol{y}}$ as the outcome of an individual experiment; $\vec{\boldsymbol{y}}$ is a k-dimensional vector (i.e., there are k outcomes of each experiment), and so the results of each experiment $\vec{\boldsymbol{y}}$ belong to the k-dimensional vector space $\mathbb{R}^k$;
- $\vec{\boldsymbol{y}}_{\text{target}}$ as the target outcome of the optimization process; $\vec{\boldsymbol{y}}_{\text{target}} \in \mathbb{R}^k$ is also a k-dimensional vector;
- $\vec{\boldsymbol{f}}_{\vec{\theta}}(\vec{\boldsymbol{x}})$ as a best-fit predictive model; $\vec{\boldsymbol{f}}_{\vec{\theta}}(\vec{\boldsymbol{x}})$ is a function that maps a candidate experiment $\vec{\boldsymbol{x}} \in \mathbb{R}^d$ to an estimated outcome $\vec{\boldsymbol{f}}_{\vec{\theta}}(\vec{\boldsymbol{x}}) \in \mathbb{R}^k$ and is parameterized by a set of p model parameters $\vec{\theta}$ (i.e., $\vec{\theta} \in \mathbb{R}^p$) that are determined based on the training data set; and
- $\mathcal{L}\left(\vec{\boldsymbol{x}}, \vec{\boldsymbol{f}}(\vec{\boldsymbol{x}}), \vec{\boldsymbol{y}}_{\text{target}}\right)$ as the objective function (also called a loss or a cost function); $\mathcal{L}\left(\vec{\boldsymbol{x}}, \vec{\boldsymbol{f}}(\vec{\boldsymbol{x}}), \vec{\boldsymbol{y}}_{\text{target}}\right)$ is a function that maps a candidate experiment $\vec{\boldsymbol{x}} \in \mathbb{R}^d$, an estimated outcome of that experiment $\vec{\boldsymbol{f}}_{\vec{\theta}}(\vec{\boldsymbol{x}}) \in \mathbb{R}^k$, and a desired optimization outcome $\vec{\boldsymbol{y}}_{\text{target}} \in \mathbb{R}^k$ to a scalar number that represents a quantitative estimate of how "bad" that candidate recipe is (i.e., better recipes have lower losses, and a recipe that meets all optimization targets has zero loss).

With this notation, the optimization problem can be formulated mathematically as

$$\vec{\boldsymbol{x}}^* = \underset{\vec{\boldsymbol{x}} \in \mathcal{X}}{\operatorname{argmin}} \mathcal{L}\left(\vec{\boldsymbol{x}}, \vec{\boldsymbol{f}}_{\vec{\theta}}(\vec{\boldsymbol{x}}), \vec{\boldsymbol{y}}_{\text{target}}\right); \tag{1}$$

in words, an optimized experiment $\vec{\boldsymbol{x}}^*$ is the set of control attributes (of all possible input combinations in $\mathcal{X}$) that minimizes the loss function according to the best-fit model $\vec{\boldsymbol{f}}_{\vec{\theta}}(\vec{\boldsymbol{x}})$ and a desired target outcome $\vec{\boldsymbol{y}}_{\text{target}}$.

The first step of solving this minimization problem involves initial data acquisition, requiring an initial sampling strategy. In the context of a specific application, the sampling strategy may be (and, traditionally, has been) within the purview of a domain expert (e.g., dictated by the informed reasoning of an expert process engineer); in such a context, the sampling strategy may be referred to as a design of experiments (DoE). If available, domain knowledge can and should be leveraged in the course of sampling strategy design; the primary mechanisms by which such domain knowledge can hasten a data-science-driven optimization is 1) by identifying and sampling within regions near to the optimization target region (which minimizes the degree of extrapolation required of the data-science-driven recommendations) and 2) by defining the extent of the viable search region (confining the input feature space more compactly filters out unviable regions from being modeled and proposed; however, the domain expert should be careful not to confine the viable space so compactly as to disregard unconventional but potentially fruitful search regions).

Beyond these application-specific observations, additional considerations about the sampling strategy that facilitate model learning and optimization include roughly equidistant spacing between data points, data sampling of a sufficient density to fully span the relevant input feature space, and multivariate sampling. These conditions ensure that distinct regions of input space are neither under- nor over-sampled, that the distance between the sampled region and the target optimization region is not too large (i.e., minimizes optimization extrapolation), and that models learn the marginal effect of each individual input attribute on the predictive outcomes. Data sets satisfying these conditions are well-studied in the field of quasi-random methods for numerical integration and are called low-discrepancy sets (so-called because such a sampling distribution minimizes the discrepancy with sampling from a uniform distribution over all input features) (Antonov and Saleev, 1979, Bratley et al., 1992, Faure, 1981, Halton, 1960, McKay et al., 1979, Sobol', 1967).

Another important aspect of the sampling strategy is that some input attributes matter more than others, but which inputs matter most (and their quantitative relative importances) are often unknown before optimization begins and may change depending on the portion of input space under consideration. In light of this, if the sampling strategy is not multivariate, the convergence rate of the optimization routine will be suppressed due to the curse of dimensionality (the amount of suppression will be exponential in the number of unimportant input attributes). Efficient sampling strategies must therefore space-fill over both the full input space and also over arbitrary subspaces of the full input space; examples of such space-filling methods are multivariate random sampling and quasi-random sampling (Bergstra and Bengio, 2012). Efficiency gains from multivariate random and quasi-random sampling methods are particularly large in the context of underlying relationships that, while embedded in a high dimensional input space, possess a low effective dimensionality (Caflisch et al., 1997, Wang et al., 2016).

A natural concern that may arise when contemplating data acquisition strategies regards the optimization convergence rate and its dependence on the sample size, particularly if experimental observations are costly and/or time consuming to collect. Unfortunately, the number of observations in isolation is inadequate for specifying the quality of a data set; characteristics including the proximity of the sampled region to the target region, the absence or presence of sampling bias, the absence or presence of significant measurement error, the complexity of the latent relations in the data, and the degree of multivariate input sampling are all at least as important as sample size in characterizing the quality of the data set and enabling predictive models to learn underlying data patterns for the purposes of optimization. Which initial data acquisition strategy accelerates the optimization convergence rate the most is application-dependent, depending further on additional modeling and optimizing strategies. The best context-independent statements are those from the preceding paragraphs, namely that sampling strategies should leverage domain knowledge, space-fill the relevant input feature space, sample the input control attributes multivariately, and yield a low-discrepancy data set; these aspects essentially lower-bound the worst case performances of data-driven approaches deployed in an arbitrary application. It should also be noted that a myopic focus on minimizing the number of experimental observations prior to deploying AI methods is at odds with the AI ethos—AI requires data! For this reason, applications where such historical data has *already* been collected are the best poised for leveraging the tools of AI. If a historical database is for some reason unavailable, approaches to optimization may then need to be adapted to rely more heavily on theoretical models driven from first principles and other kinds of domain knowledge (although the results from such adaptations are likely to be contingent upon the particular application and difficult to scale).

3 Modeling Strategies

Solving the optimization problem formulated in Equation (1) requires a best-fit predictive model $\vec{f}_{\vec{\theta}}(\vec{x})$, i.e., a nonlinear regression model (also referred to in the statistics literature as a response surface (Jones et al., 1998)) parameterized by model parameters $\vec{\theta}$ that are determined based on the training data set produced by the sampling strategy. While there are many options available for the parametric form of the model, some desiderata include:

- **Conformability to arbitrary underlying relationships in the data set** Given that the latent underlying relationships in the data are generally nonlinear and arbitrarily complex, the parametric model form (by virtue of suitable tuning of model fit parameters) should be a universal function approximator. Neural networks are one such model form (Cybenko, 1989, Hornik, 1991), but other options in this class include Gaussian process regressors (Rasmussen and Williams, 2006) and regression trees (Breiman et al., 2017). In addition to choosing a particular (or ensemble) of such universal approximators, training such a model generally involves making additional supplementary model choices (e.g., choosing a kernel for a Gaussian process regressor) and determining appropriate hyperparameters (e.g., via cross-validation (Hastie et al., 2001), grid search, or manual tuning).
- **Generalizability beyond training data set** Despite being conformable to arbitrary relationships contained within the data, predictive models should be resilient against fitting to latent noise contained in the training data (i.e., avoid overfitting). Overfitting is present when the predictive model learns patterns that are more reflective of the particularities of the training set than genuine causal relationships between input control attributes and target outputs; accordingly, overfitting can be identified when model training error is significantly smaller than model error on a holdout validation set. Techniques for minimizing the presence and adverse effects of overfitting during model training include cross-validation, regularization, and early stopping (Goodfellow et al., 2016).
- **Ability to compute calibrated probabilistic uncertainty estimates** In addition to making accurate predictions about the mean output target values conditioned on a particular set of input control values, the predictive model should also produce calibrated uncertainty estimates (i.e., prediction intervals at a specified probability coverage) associated with those predictions. Such uncertainty intervals are particularly important for navigating the exploration-exploitation trade off when leveraging the predictive models for optimization (Brochu et al., 2010, Shahriari et al., 2016). Standard methods for constructing such calibrated prediction intervals include the delta method (Seber and Wild, 1989) or bootstrapping (e.g., the pairs bootstrap method (Efron, 1979) or the wild bootstrap method (Wu, 1986)). Other techniques for prediction interval construction include variational inference (Fox and Roberts, 2011, MacKay, 2002) and deep ensembling (Lakshminarayanan et al., 2017, Wilson and Izmailov, 2020). There are also classes of models that directly learn both mean and variance output regression functions during training (Nix and Weigend, 1994, Russell and Reale, 2019), such that prediction intervals can be directly estimated at inference time.
- **Adaptability to known data structure** If the data possesses known structure (e.g., via *a priori* domain knowledge), it can hasten the convergence of the fitting routine to an accurate predictive model if those structures can be leveraged by incorporation in the parametric model form. For example, experiments in most scientific, engineering, and manufacturing industries are rarely as simple as setting a single array of control inputs and retrieving an experimental outcome; instead, a time-ordered sequence of control inputs spanning multiple steps may be required to complete the experiment. In this instance, model architectures that are specifically adapted for learning such sequentially-ordered patterns (e.g., recurrent neural networks (Rumelhart et al., 1986), long short term memory networks (Hochreiter and Schmidhuber, 1997), sequence-model-based autoencoders for learning a fixed-size embedding (Dai and Le, 2015, Goodfellow et al., 2016, Pei and Tax, 2018, Sagheer and Kotb, 2019), ordinary differential equation networks (Chen et al., 2018)) may improve the generalization performance of the predictive models fit to that data. As a second example, for data that possesses spatial structure, parametric model forms that can learn such spatial correlation (e.g., spatial econometric models (Anselin, 1988), including geographically weighted neural networks (Hagenauer and Helbich, 2021)) may be particularly well-suited.
- **Differentiability with respect to model parameters and with respect to input control features** Calculus-based methods are particularly efficient algorithms for model training and optimization; in both instances, a suitable objective function is defined and numerical optimization of that loss function is performed, typically via a gradient descent, conjugate gradient, or quasi-Newton method (Nocedal and Wright, 2006). Such methods involve differentiating the loss function with respect to model parameters (for model training) or inputs (for optimization), and so the base architecture itself must be differentiable with respect to these quantities. If a calculus-based optimization routine is not employed (e.g., derivative-free optimization (Conn et al., 2009)),

the differentiability requirement can be relaxed; however, physical processes are generally differentiable, and so failure to impose the differentiability constraint enables training data-driven models that evince unphysical (e.g., non-smooth) behavior.

While producing models with higher predictive accuracy presumably enables better recommendations during optimization, obtaining such high fidelity models is *not* the primary objective of a data-science-driven optimization routine. In particular, if the cost of obtaining such highly accurate predictive models is excessive (expensive) experimental observations, then a fixation on model predictive accuracy may actually increase the number of experiments before successful optimization (minimizing the number of such required experiments *is* the primary objective of a data-science-driven optimization process). Instead of focusing on model test error within the optimization loop, the data-science-driven approach trades high fidelity models for statistical forecasts; generating predictions from a statistical model (with calibrated prediction intervals) enables the rapid identification of fruitful regions of the input feature space in order to accelerate optimization.

As a concluding comment in this section, it should be noted that Equation (1) is not the only possible formulation of the optimization problem; in particular, there are formulations wherein a best-fit predictive model need not be learned at all. These methods generally fall into a class of kernel density estimation methods (Hastie et al., 2001, Parzen, 1962). For example, tree-structured Parzen estimators (Bergstra et al., 2011) are a type of non-parametric kernel density estimator that learns two density distributions (corresponding to partitions of the training data into better and worse observations); these distributions can then be used during optimization to suggest points closer to the distribution of better data observations and/or further away from the worse data observations without the need for a predictive model at all. A major downside of this approach, however, is that by explicitly eschewing attempts to learn a predictive model, no insight can be gleaned as to the marginal impact of select control inputs on desired target outputs (such Jacobian information requires explicit differentiation of a predictive model or approximations thereof). These marginal impacts (sometimes referred to as sensitivities) are often helpful for the purposes of model diagnostics and exploratory data analysis.

4 Optimizing Strategies

In the setting of Equation 1, optimization is equivalent to minimization of a suitable objective function. The objective function reduces experimental outcomes (or estimates of experimental outcomes under the model) to a scalar cost measure, with zero cost corresponding to an outcome that achieves all the optimization objectives. Examples of constituent terms within the objective function include a distance measure between predicted outcomes and target outputs, an uncertainty measure that penalizes (or favors) candidate inputs associated with uncertain predictions, or a monetary cost measure associated with each input candidate. In general, the objective function is presumed to be expensive to evaluate experimentally; it may further lack a known closed-form, may constitute a stochastic random variable (i.e., observations of the objective function may be corrupted by observation noise, itself deriving from an unknown stochastic noise distribution), and may possess unknown differentiability properties (and, in general, derivative information may only be estimated from sampled observations). Further, the minimization problem formulated in Equation 1 is generally non-convex, and so the optimization is generally an NP-hard problem (Jain and Kar, 2017).

Within this setting, Bayesian optimization encompasses a variety of iterative strategies for minimizing the cost function. In terms of the number of iterations (i.e., experimental function evaluations) required to find the minimum of the cost function, Bayesian optimization strategies are among the most efficient (Brochu et al., 2010, Jones, 2001, Jones et al., 1998, Mockus, 1994, Streltsov and Vakili, 1999), where the efficiency results from the incorporation (via Bayes' theorem) of information from prior beliefs and sequentially-updated observations to guide the sampling strategy, update the surrogate model, and trade off exploration and exploitation over the input feature space. Having already implemented an initial sampling and a modeling strategy, Bayesian optimization follows the principle of maximum expected utility (equivalently, the principle of minimum expected risk (Vapnik, 1991)) for proposing the next sample, which requires a choice of a suitable utility function (also referred to as an acquisition function) and a means of optimizing the expected value of this utility with respect to the posterior distribution of the cost function. Acquisition functions are defined such that high utility corresponds to low values of the cost function, either because 1) the estimated cost function value at a candidate input is low, 2) the uncertainty associated with the cost function estimation value at a candidate input is high (and with a substantial probability of obtaining a low value), or 3) both 1) and 2). Examples of acquisition functions discussed in the literature include:

- probability of improvement (Kushner, 1964), wherein input candidates are ranked based on the estimated probability that the mean outcome associated with those inputs will improve upon the best observed sample so far;

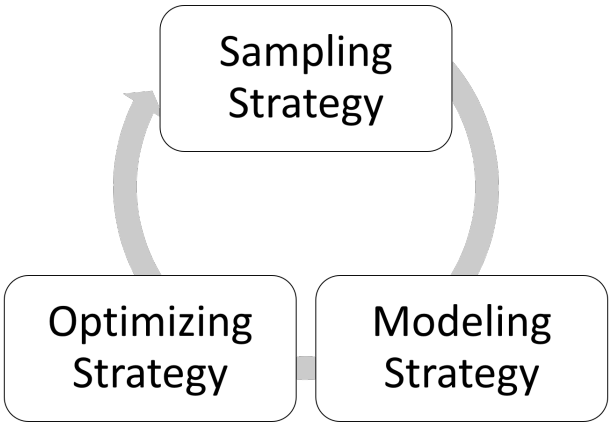

Figure 1: The sequential model-based optimization feedback loop. In the first step, an initial sampling strategy is employed to obtain a suitable initial data set, conforming to the conditions described in Section 2. In the second step, a modeling strategy is employed, which leverages the data set to produce a predictive model (Section 3). In the third step, an optimizing strategy is employed, which leverages the data set and the trained models to produce optimized input candidates. Experimental observations obtained from proposed inputs are then added to the data set and the loop run anew, terminating upon convergence.

- expected improvement (Mockus et al., 1978), wherein input candidates are ranked based on both the estimated probability that the mean outcome will improve on the best observed sample so far and the estimated magnitude of improvement of the mean outcome over the best observed sample so far;
- upper confidence bound (Auer, 2002, Auer et al., 2002, Kocsis and Szepesvári, 2006, Srinivas et al., 2010), wherein input candidates are ranked based on the probability that 68% of estimated outcomes for that input (i.e., outcomes within $\pm 1\sigma$ of the estimated mean outcome) will improve upon the best observed sample so far; and
- GP-Hedge (Hoffman et al., 2011), a portfolio strategy that utilizes multiple acquisition functions and adapts dynamically based on which acquisition functions propose better candidates throughout the optimization procedure.

In a general setting, there is no acquisition function that can be guaranteed to perform best in optimizing the cost function (Hoffman et al., 2011); which acquisition function will perform best in any particular optimization process is highly application specific, and the differences between optimization convergence rates under different acquisition functions are unlikely to be statistically significant. While the formulation in Equation 1 and most of the discussion has presumed, for simplicity, that a single input condition is proposed as an optimized candidate at each iteration of optimization, the approach can generally be extended to propose any number of input candidates at each iteration and to plan multiple steps ahead during optimization (Azimi et al., 2010, Brochu et al., 2010, Garnett et al., 2010). Once initial sampling, modeling, and optimizing strategies have been specified, the combination can feedback on itself within an iterative loop (sometimes referred to in the literature as sequential model-based optimization (Hutter et al., 2011)), as schematically depicted in Figure 1.

5 Conclusion

The variety of domains in which AI approaches have made dramatic advances (including computer vision (Krizhevsky et al., 2012, Szegedy et al., 2015), natural language processing (Sutskever et al., 2014, Vaswani et al., 2017), and reinforcement learning (Silver et al., 2017)) motivates further efforts to develop new AI capabilities and to apply them in increasingly diverse domains. The authors share in the collective enthusiasm for the potential impact of these tools. Despite this excitement, AI is *not* a panacea. For AI driven approaches to yield robust insights and accelerated optimizations, data must be available as the basis for the analysis, or else additional information must be brought to bear (e.g., *a priori* knowledge, simulation, knowledge engineering, physical or empirical laws). In order for AI techniques to generalize well, the data must be well-curated, test data must be similar to training data, and the underlying patterns themselves should be approximately stable (i.e., well-approximated as a stationary stochastic process). Even in situations in which data generation is imperfect, though, AI approaches can still guide aspects of the sampling, modeling, and optimizing strategies; indeed, such situations are especially likely to benefit from

AI techniques, particularly in comparison to *ad hoc* approaches that ignore aspects of the available data. While AI approaches can support analyses at any stage of the sequential model-based optimization feedback loop, their real power is harnessed in situations where domain knowledge can be synergistically matched with a high quality historical database. The authors are particularly enthusiastic about the sequential model-based optimization framework presented herein, noting its **universal** applicability across domains and its inherent **scalability** throughout the entire R&D and manufacturing life-cycle. This framework is equally adept at navigating the high-dimensional parameter space of initial, small-scale, high-throughput screening as it is at fine-tuning the process parameters of capital-intensive, large-scale production. At each stage, the process can be treated as an expensive black-box function, making it an ideal candidate for this data-efficient optimization strategy.

References

Luc Anselin. *Spatial Econometrics: Methods and Models*. Springer Netherlands, 1988. doi:10.1007/978-94-015-7799-1.

I. A. Antonov and V. M. Saleev. An Economic Method of Computing LP_τ-sequences. *USSR Computational Mathematics and Mathematical Physics*, 19:252–256, 1979. doi:10.1016/0041-5553(79)90085-5.

Peter Auer. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47:235–256, 2002. doi:10.1023/a:1013689704352.

Javad Azimi, Alan Fern, and Xiaoli Fern. Batch Bayesian Optimization Via Simulation Matching. In *Advances in Neural Information Processing Systems*, volume 23 of *NIPS'10*. Curran Associates Inc., 2010.

James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for Hyper-Parameter Optimization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pages 2546–2554, Red Hook, NY, USA, 2011. Curran Associates Inc. ISBN 9781618395993.

Paul Bratley, Bennett L. Fox, and Harald Niederreiter. Implementation and Tests of Low-Discrepancy Sequences. *ACM Transactions on Modeling and Computer Simulation*, 2:195–213, 1992. doi:10.1145/146382.146385.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification And Regression Trees*. Routledge, 2017. doi:10.1201/9781315139470.

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. *Computing Research Repository*, abs/1012.2599, 2010. URL `http://arxiv.org/abs/1012.2599`.

Russel E. Caflisch, William Morokoff, and Art Owen. Valuation of Mortgage-backed Securities Using Brownian Bridges to Reduce Effective Dimension. *Journal of Computational Finance*, 1:27–46, 1997. doi:10.21314/jcf.1997.005.

J. M. Z. Chaves, E. Wang, T. Tu, E. D. Vaishnav, B. Lee, S. S. Mahdavi, C. Semturs, D. Fleet, V. Natarajan, and S. Azizi. Tx-LLM: A Large Language Model for Therapeutics. *arXiv preprint arXiv:2406.06316*, 2024.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural Ordinary Differential Equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 6572–6583, Red Hook, NY, USA, 2018. Curran Associates Inc.

Christopher M. Childs and Newell R. Washburn. Embedding Domain Knowledge for Machine Learning of Complex Material Systems. *MRS Communications*, 9(3):806–820, 2019. doi:10.1557/mrc.2019.90.

Andrew R. Conn, Katya Scheinberg, and Luis N. Vicente. *Introduction to Derivative-Free Optimization*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2009. doi:10.1137/1.9780898718768.

G. Cybenko. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989. doi:10.1007/bf02551274.

Andrew M. Dai and Quoc V. Le. Semi-supervised Sequence Learning. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 3079–3087, Cambridge, MA, USA, 2015. MIT Press.

Stefan Droste, Thomas Jansen, and Ingo Wegener. Optimization with Randomized Search Heuristics—The (A)NFL Theorem, Realistic Scenarios, and Difficult Functions. *Theoretical Computer Science*, 287:131–144, 2002. doi:10.1016/S0304-3975(02)00094-4.

B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1–26, 1979. doi:10.1214/aos/1176344552.

Henri Faure. Discrépances de Suites Associées à un Système de Numération (en Dimension un). *Bulletin de la Société Mathématique de France*, 109:143–182, 1981. doi:10.24033/bsmf.1935.

Charles W. Fox and Stephen J. Roberts. A Tutorial on Variational Bayesian Inference. *Artificial Intelligence Review*, 38:85–95, 2011. doi:10.1007/s10462-011-9236-8.

R. Garnett, M. A. Osborne, and S. J. Roberts. Bayesian Optimization for Sensor Set Selection. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks - IPSN '10*. ACM Press, 2010. doi:10.1145/1791212.1791238.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL `http://www.deeplearningbook.org`.

Julian Hagenauer and Marco Helbich. A Geographically Weighted Artificial Neural Network. *International Journal of Geographical Information Science*, pages 1–21, 2021. doi:10.1080/13658816.2021.1871618.

J. H. Halton. On the Efficiency of Certain Quasi-Random Sequences of Points in Evaluating Multi-Dimensional Integrals. *Numerische Mathematik*, 2:84–90, 1960. doi:10.1007/BF01386213.

A. H. Hamilton, M. Roughan, and A. Kalenkova. Elo Ratings in the Presence of Intransitivity. *arXiv preprint arXiv:2412.14427*, 2024.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997. doi:10.1162/neco.1997.9.8.1735.

Matthew Hoffman, Eric Brochu, and Nando de Freitas. Portfolio Allocation for Bayesian Optimization. In *Uncertainty in Artificial Intelligence (UAI)*, pages 327–336, 2011.

Kurt Hornik. Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks*, 4:251–257, 1991. doi:10.1016/0893-6080(91)90009-T.

Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential Model-Based Optimization for General Algorithm Configuration. In *Lecture Notes in Computer Science*, pages 507–523. Springer Berlin Heidelberg, 2011. doi:10.1007/978-3-642-25566-3_40.

Prateek Jain and Purushottam Kar. Non-convex Optimization for Machine Learning. *Foundations and Trends in Machine Learning*, 10(3-4):142–363, 2017. doi:10.1561/2200000058.

Donald R. Jones. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization*, 21:345–2383, 2001. doi:10.1023/a:1012771025575.

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, 1998. doi:10.1023/a:1008306431147.

N. Jones. OpenAI's 'deep research' tool: is it useful for scientists? *Nature News*, 2025. doi:10.1038/d41586-025-00377-9.

Levente Kocsis and Csaba Szepesvári. Bandit Based Monte-Carlo Planning. In *Lecture Notes in Computer Science*, pages 282–293. Springer Berlin Heidelberg, 2006. doi:10.1007/11871842_29.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25 of *NIPS'12*. Curran Associates, Inc., 2012.

H. J. Kushner. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering*, 86:97–106, 1964. doi:10.1115/1.3653121.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing*, NIPS'17, page 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

David J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, USA, 2002. ISBN 0521642981.

M. D. McKay, R. J. Beckman, and W. J. Conover. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics*, 21:239, 1979. doi:10.2307/1268522.

S. Megat, N. Mora, J. Sanogo, O. Roman, A. Catanese, N. O. Alami, A. Freischmidt, X. Mingaj, H. De Calbiac, F. Muratet, and et al. Integrative genetic analysis illuminates ALS heritability and identifies risk genes. *Nature Communications*, 14:342, 2023.

J. Mockus, V. Tiešis, and A. Žilinskas. *The Application of Bayesian Methods for Seeking the Extremum*, volume 2, pages 117–129. Elsevier, 1978. ISBN 0-444-85171-2.

Jonas Mockus. Application of Bayesian Approach to Numerical Methods of Global and Stochastic Optimization. *Journal of Global Optimization*, 4:347–365, 1994. doi:10.1007/bf01099263.

David A. Nix and Andreas S. Weigend. Estimating the Mean and Variance of the Target Probability Distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 1, pages 55–60, 1994. doi:10.1109/ICNN.1994.374138.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer New York, 2006. doi:10.1007/978-0-387-40065-5.

Emanuel Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962. doi:10.1214/aoms/1177704472.

Wenjie Pei and David M. J. Tax. Unsupervised Learning of Sequence Representations by Autoencoders. *Computing Research Repository*, abs/1804.00946, 2018. URL `http://arxiv.org/abs/1804.00946`.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. ISBN 0-262-18253-X.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning Representations by Back-propagating Errors. *Nature*, 323:533–536, 1986. doi:10.1038/323533a0.

Rebecca L. Russell and Christopher P. Reale. Multivariate Uncertainty in Deep Learning. *Computing Research Repository*, abs/1910.14215, 2019. URL `http://arxiv.org/abs/1910.14215`.

Alaa Sagheer and Mostafa Kotb. Unsupervised Pre-training of a Deep LSTM-based Stacked Autoencoder for Multivariate Time Series Forecasting Problems. *Scientific Reports*, 9, 2019. doi:10.1038/s41598-019-55320-6.

G. A. F. Seber and C. J. Wild. *Nonlinear Regression*. John Wiley & Sons, Inc., 1989. doi:10.1002/0471725315.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando de Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104:148–175, 2016. doi:10.1109/JPROC.2015.2494218.

P. Shaw, B. Gurram, D. Belanger, A. Gane, M. L. Bileschi, L. J. Colwell, K. Toutanova, and A. P. Parikh. ProtEx: A Retrieval-Augmented Approach for Protein Function Prediction. *bioRxiv*, 2024.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the Game of Go Without Human Knowledge. *Nature*, 550:354–359, 2017. doi:10.1038/nature24270.

M. D. Skarlinski, S. Cox, J. M. Laurent, J. D. Braza, M. Hinks, M. J. Hammerling, M. Ponnapati, S. G. Rodriques, and A. D. White. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740*, 2024.

I. M. Sobol'. On the Distribution of Points in a Cube and the Approximate Evaluation of Integrals. *USSR Computational Mathematics and Mathematical Physics*, 7:86–112, 1967. doi:10.1016/0041-5553(67)90144-9.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1015–1022, 2010.

Simon Streltsov and Pirooz Vakili. A Non-myopic Utility Function for Statistical Global Optimization Algorithms. *Journal of Global Optimization*, 14:283–298, 1999. doi:10.1023/a:1008284229931.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. doi:10.1109/CVPR.2015.7298594.

Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. *Introduction to Data Mining*. Pearson, 2nd edition, 2018.

R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia, A. Poulton, V. Kerkez, and R. Stojnic. Galactica: A Large Language Model for Science. *arXiv preprint arXiv:2211.09085*, 2022.

V. Vapnik. Principles of Risk Minimization for Learning Theory. In *Advances in Neural Information Processing Systems*, volume 4 of *NIPS'91*, pages 831–838, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc. ISBN 1558602224.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Advances in Neural Information Processing Systems*, volume 30 of *NIPS'17*. Curran Associates, Inc., 2017.

Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando De Feitas. Bayesian Optimization in a Billion Dimensions via Random Embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016. doi:10.1613/jair.4806.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. *Advances in Neural Information Processing Systems*, 2020. ISSN 1049-5258.

C. F. J. Wu. Jackknife, Bootstrap and Other Resampling Methods in Regression Analysis. *The Annals of Statistics*, 14 (4):1261–1295, 1986. doi:10.1214/aos/1176350142.