Constrained Bayesian Optimization for Automatic Chemical Design

Ryan-Rhys Griffiths

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy

Wolfson College

August 2017

Declaration

I, Ryan-Rhys Griffiths of Wolfson College, being a candidate for the M.Phil in Machine Learning, Speech and Language Technology, hereby declare that this thesis and the work described in it are my own work unaided except as may be specified below, and that the thesis does not contain material that has already been used to any substantial extent for a comparable purpose. This thesis contains 13,750 words.

Ryan-Rhys Griffiths
August 2017

Acknowledgements

I wish to thank my supervisor, José Miguel Hernández-Lobato, both for his high-level guidance and for his technical support.

Abstract

Automatic Chemical Design leverages recent advances in deep generative modelling to provide a framework for performing continuous optimization of molecular properties. Although the provision of a continuous representation for prospective lead drug candidates has opened the door to hitherto inaccessible tools of mathematical optimization, some challenges remain for the design process. One known pathology is the model's tendency to decode invalid molecular structures. The goal of this thesis is to test the hypothesis that the origin of this pathology is rooted in the current formulation of Bayesian optimization. Recasting the optimization procedure as a constrained Bayesian optimization problem, results in novel drug compounds produced by the model consistently ranking in the 100th percentile of the distribution over training set scores.

Contents

Front Matter

Acknowledgements . . . 3

Abstract . . . 4

Contents . . . 5

List of Figures . . . 8

List of Tables . . . 9

1 Introduction **10**

1.1 Motivation . . . 10

1.2 Automatic Chemical Design . . . 10

1.3 Outline of Thesis . . . 12

2 Background **13**

2.1 Bayesian Optimization . . . 13

2.2 Constrained Bayesian Optimization . . . 17

2.2.1 Modelling . . . 18

2.2.2 Data Collection Strategy . . . 29

3 Results **33**

3.1 Outline . . . 33

3.2 Branin-Hoo Function . . . 33

3.2.1 Implementation . . . 35

3.2.2 Results . . . 35

3.2.3 Discussion . . . 37

3.3 Background to Automatic Chemical Design . . . 38

3.4 Performance of Constrained Bayesian Optimization . . . 40

3.4.1 Implementation . . . 41

3.4.2 Diagnostic Experiments and Labelling Criteria . . . 41

3.4.3 Molecular Validity 43
3.4.4 Molecular Quality 46
3.5 Objective Design 47

4 Conclusion 50
4.1 Contributions 50
4.2 Future Work 50

References 51

List of Figures

1.1 An Illustration of the Automatic Chemical Design Pipeline. 11

2.1 A Demonstration of Bayesian Optimization [103]. 15

3.1 Constrained Bayesian optimization of the 2D Branin-Hoo Function. 34

3.2 Constrained Bayesian Optimization on the 2D Branin-Hoo Function with a Coupled Disk Constraint. **a)** Minima of the true Branin-Hoo function. **b)** Contour plot of the predictive mean of the sparse GP used to model the objective function. Lighter colours indicate lower values of the objective. **c)** Data points collected over 40 iterations of sequential Bayesian optimization. **d)** The disk constraint, where the feasible region is given in yellow. **e)** The predictive variance of the sparse GP used to model the objective function. **f)** The contour learned by the BNN giving the probability of constraint satisfaction. 36

3.3 Performance of Parallel Bayesian Optimization with EIC against Random Sampling. 37

3.4 Automatic Chemical Design as a Conglomerate of Smaller Models. 38

3.5 SMILES - A Text Representation for Molecules. 38

3.6 The Model as a Sequence of Representational Transformations. Step 1 shows a benzene molecule being represented in its text-based format as a SMILES string. Step 2 shows the SMILES string for benzene being encoded using a text encoder into a continuous-valued vector of latent points. Step 3 shows the optimization process occurring within the latent space of the variational autoencoder, whereupon some objective function of the latent points is optimized to yield a recommendation for the best point. Step 4 shows the optimal latent point being decoded to a SMILES string. Step 5 shows the now optimal SMILES string being decoded into an optimal molecule. 39

3.7 Experiments on 5 disjoint sets comprising 50 latent points each. VS Noise are training data points with approximately 1% noise added to their values, S Noise have 10% noise added to their values and B Noise have 50% noise added to their values. As an example, the percentage noise in the case of 1% noise is defined to be a value drawn from a uniform distribution where the lower end of the range is 0.01 times the maximum difference between the values of the training data along a given dimension. **a)** The percentage of times a given latent point from each set was decoded to a valid molecule, averaged over 50 latent points. **b)** The percentage of times a given latent point from each set was decoded to a methane molecule, averaged over 50 latent points. 42

3.8 Drug-like Molecules. 43

3.9 The percentage of latent points decoded to drug-like molecules. 20 iterations of Bayesian optimization with batches of 50 data points collected at each iteration (1000 latent points decoded in total). The standard error is given for 5 separate train/test set splits of 90/10. Drug-like in this instance specifies a requirement that the molecules must both be valid and have a length greater than 5 in the SMILES Representation. 44

3.10 The percentage of new drug-like molecules generated from 20 iterations of Bayesian optimization with batches of 50 data points collected at each iteration (1000 latent points decoded in total). The standard error is given for 5 separate train/test set splits of 90/10. 45

3.11 The best scores for new molecules generated from the baseline model (blue) and the model with constrained Bayesian Optimization (red). The vertical lines show the best scores averaged over 5 separate train/test set splits of 90/10. The histogram is presented such that only the top 10% of the training data scores are depicted. 46

List of Tables

3.1 Percentile of the Averaged New Molecule Score Relative to the Training Data. The Results of 5 Separate Train/Test Set Splits of 90/10 are Provided. 47

3.2 Percentile of the Best New Molecule Score Relative to the Training Data. The Results of 5 Separate Train/Test Set Splits of 90/10 are Provided. 47

Chapter 1

Introduction

1.1 Motivation

The goal of chemical design is to find novel molecular structures that display desirable properties. This search is complicated by the fact that chemical space is vast. From basic structural rules it is estimated that the space of pharmacologically active molecules satisfying Lipinski's rule of five [65] is upwards of 10^{60}, a novemdicillion [95, 56, 7]. Further to this, chemical space is discrete and so existing search methods in de novo drug design such as genetic algorithms cannot make use of continuous optimization techniques which may leverage geometrical cues such as gradients to optimize the objective [99, 107]. Recently however, advances in machine learning and in particular in the field of deep generative modelling have produced models capable of interconverting between discrete and continuous representations of traditionally challenging data formats [8]. The generation of realistic synthetic samples from these models has found applications in image [37], text [8], speech and music [87] generation. Closely following this work, Gómez-Bombarelli et al.[33] have developed a model capable of converting a discrete representation of molecules to and from a continuous representation. On conversion to the continuous representation, optimization that makes use of tools from continuous mathematics may be undertaken to search for molecules with desired properties. Optimized continuous representations may then be generated from the model, giving rise to the concept of Automatic Chemical Design.

1.2 Automatic Chemical Design

The operation of the Automatic Chemical Design pipeline is illustrated schematically in Figure 1.1. Starting with a discrete representation of some training molecules, denoted with

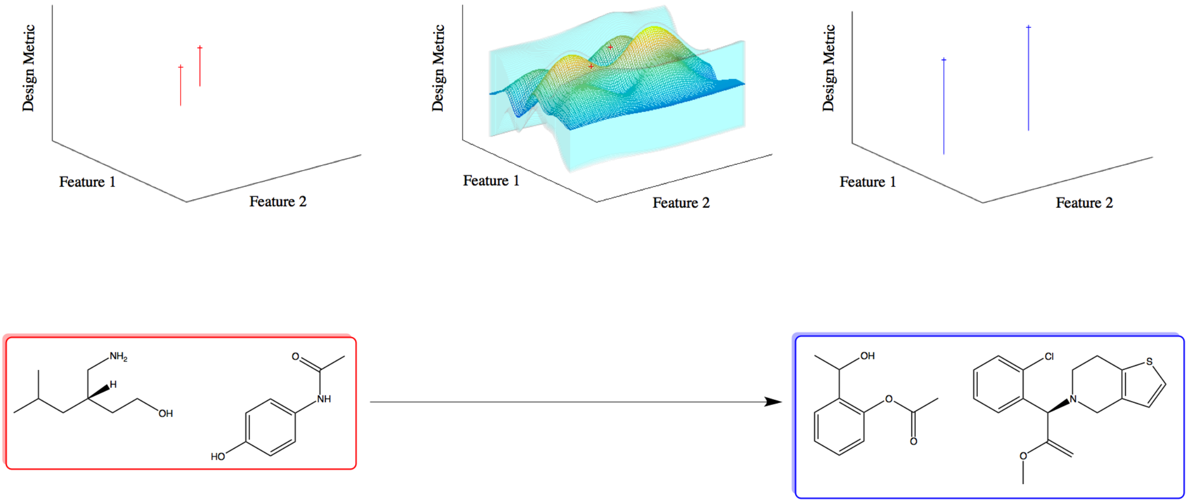


Fig. 1.1 An Illustration of the Automatic Chemical Design Pipeline.

feature 1 and feature 2, and a value for some design metric encoding drug-likeness, it is possible to construct a mathematical model of the space in-between these examples via Bayesian optimization. This statistical model is then used as the space to optimize over. The maxima of the statistical model are then decoded as optimized discrete molecules. This approach has applications in high throughput virtual screening of drug candidates [98, 90, 32] as a front end for more expensive molecular docking simulations or yet more expensive physical synthesis.

Although a strong proof of concept for the idea of performing continuous optimization in molecular space, the goal of de novo drug design using such a model suffers faces some obstacles to realization:

1. Dead regions in the continuous latent space: What to do when Bayesian optimization brings you far away from your data?

2. What to optimize? How do you encode the property of drug-likeness in a numerical metric? Is it even a good idea to attempt to do this?

These are the two questions this thesis seeks to answer. The principle contribution of the thesis will be to present a solution to the first question through the implementation of an algorithm for constrained Bayesian optimization. This will rein in the Bayesian optimization process such that it only selects points that lie within a certain region of the latent space where the probability of a successful decoding is high. The second question will be addressed from an applications perspective in terms of how best to go about using the model that addresses question 1.

1.3 Outline of Thesis

Chapter 2 will outline the theoretical background of both Bayesian optimization and constrained Bayesian optimization.

Chapter 3 will present the results of constrained Bayesian optimization first on a toy constrained optimization problem to demonstrate the validity of the algorithmic implementation and second, on the molecular data, paying particular attention to compare the validity and quality of the molecules produced by the baseline (unconstrained Bayesian optimization) and constrained implementations. The chapter will finish by proposing practical strategies for the application of constrained Bayesian optimization for Automatic Chemical Design.

Chapter 2

Background

2.1 Bayesian Optimization

Bayesian optimization [61, 82] is a method of solving the mathematical problem formulated as

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \tag{2.1}$$

where $\mathbf{x}^*$ is the global minimizer (or maximizer) of an unknown objective function f and $\mathcal{X}$ is the design space. In global optimization $\mathcal{X}$ tends to be a compact subset of $\mathbb{R}^d$ although the Bayesian optimization framework is sufficiently flexible to be applied to search spaces that feature categorical inputs such as the integers [27]. In practice, the problem is further characterised by the following three properties:

1. Black-box: The unknown objective function f lacks an analytical expression but may be evaluated point-wise at any arbitrary query point $\mathbf{x}$ in the domain.

2. Noisy evaluation: The evaluation of f produces outputs that are corrupted by zero-mean noise relative to the true value.

3. Expense: Querying $f(\mathbf{x})$ at any $\mathbf{x}$ takes a long time.

An intuitive example of this class of problem in machine learning is the optimization of hyperparameters for machine learning algorithms. The expensive black-box function in this case is the mapping between a set of hyperparameters and the test set performance of the algorithm. There is no analytical form for this mapping and hence the user must query some set of parameters corresponding to the input domain in order to access the value of the function

$f(x)$. The querying process entails re-training the algorithm on each query and is very time-intensive as a consequence. An intuitive example of this class of problem in chemistry is the optimization of a potential drug candidate. In this case the expensive black-box function is the mapping between the drug candidate and its efficacy as a medicine. In order to obtain an efficacy estimate $f(\mathbf{x})$ for a drug compound x in input space it is necessary to conduct the time-intensive querying process corresponding to physically synthesising the drug in the lab and assessing its properties experimentally.

It is useful to consider what strategy to adopt in this circumstance. The most obvious approach that may spring to mind in the case of machine learning is to start with some hyperparameter setting, evaluate it, and systematically increase it until you have found the optimal evaluation. Another approach would be to guess hyperparameter settings at random until you find an optimal solution. These approaches are known as grid search and random search respectively and are heavily influenced by property 3. in so far as they are prohibitively expensive when confronted with a time-intensive evaluation process.

Bayesian optimization presents an intelligent solution to performing this class of search. It consists of two components; a surrogate model of the black-box objective and an acquisition function. The surrogate model encodes the prior belief about how the objective function should behave as well as a model of the data-generation mechanism. Given that little information is typically known about the black-box objective function in advance, it is typically required that the surrogate model is probabilistic and can maintain a measure of uncertainty about how well it is doing in representing the true black-box objective. It is also necessary that the surrogate model be sufficiently flexible to represent the black-box function. This probabilistic model is sequentially updated on each iteration of the search when a new data point is obtained. The acquisition function must direct this search so as to explore areas of the design space where the model is uncertain but the acquisition function must also exploit regions where the model shows favourable values for the optimum. This principle is referred to as learning vs. earning in the financial domain [28]. The exploitation behaviour is necessary so that the algorithm will find a favourable solution within some finite number of black-box function evaluations. Considering property 3. above once again, it is necessary that the acquisition function be cheaper to evaluate relative to the black-box objective for this strategy to be efficient. As such, acquisition functions are often chosen so as to have analytical forms that are easy to evaluate or approximate.

An illustration of the Bayesian optimization process is given below.

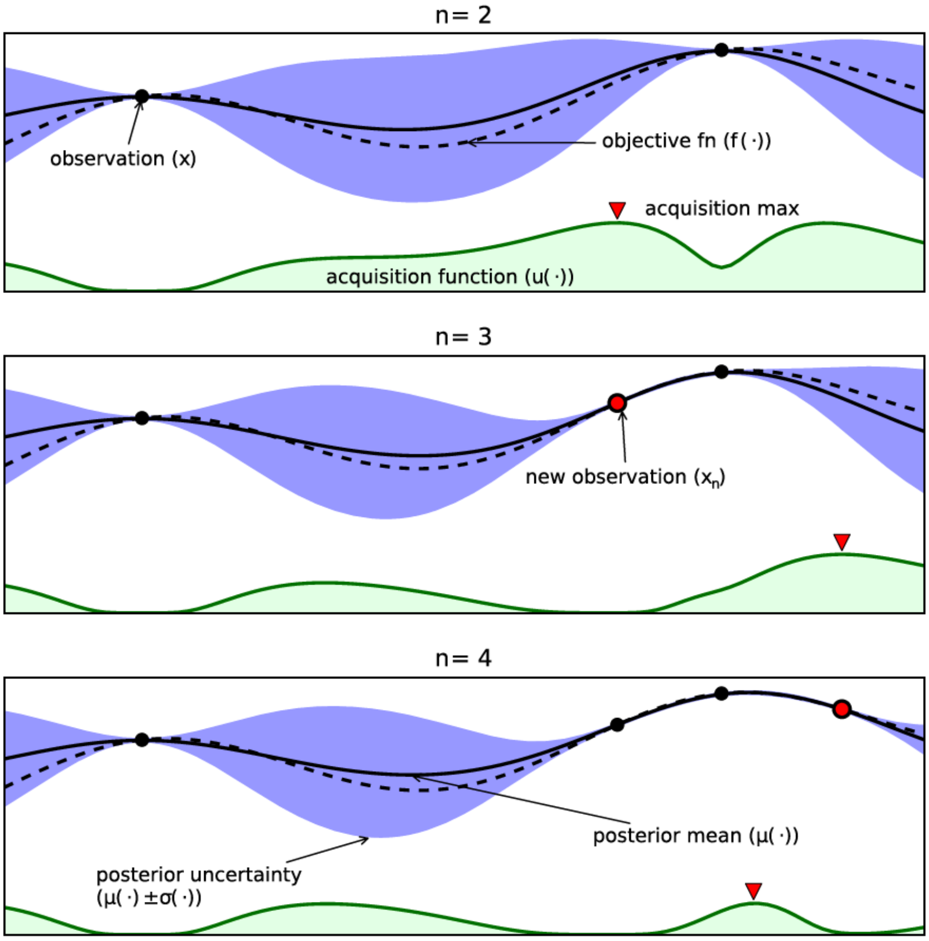


Fig. 2.1 A Demonstration of Bayesian Optimization [103].

One might imagine that the range of the machine learning parameter of interest lies on the x-axis and the y-axis represents test set performance. The black-box objective function is shown as the dotted line although it is important to note that this analytical form would not be available in practice. The surrogate model expresses uncertainty in regions where it has not seen any data so far. The acquisition function directs the search towards locations where the model has high uncertainty and high predictive mean for the maximization problem. In the chemistry setting one may imagine that the x-axis represents a set of drug candidates. One intuitively thinks of this set of drug candidates as being discrete but indeed [33] allow for a continuous representation that moulds the problem into a form suitable for continuous Bayesian optimization. The y-axis in this case would be the efficacy of the drug compound. Given a sufficiently good metric for the efficacy of a drug candidate, the procedure could lead to the identification of a strong candidate for clinical trials faster than grid search or random search.

In the specific application domain of automatic hyperparameter tuning discussed above, Bayesian optimization has found use in the tuning of deep belief networks [5], Markov Chain Monte Carlo (MCMC) methods [73, 38], Convolutional Neural Networks (CNNs) [112, 117] and choosing between algorithms offered by the WEKA and sckikit-learn machine learning libraries [120, 44]. The myriad applications beyond hyperparameter tuning include A/B testing [58, 101, 13], environmental monitoring and sensor networks [115, 26, 74], Recommender systems [64, 13, 123], Natural Language Processing (NLP) [126, 129] and Statistical Machine Translation (SMT) [4], Robotics and Reinforcement Learning [67, 78, 77, 17, 12], Combinatorial Optimization problems such as the tuning of mixed integer solvers [46, 127, 125] and nearest neighbour algorithms [75] and interactive interfaces for preference learning [23, 9]. More recently, Bayesian optimization has even spread into fields as disparate from Machine Learning as Combinatorial Chemistry [86] and wave energy flux prediction [15].

The list below gives a non-exhaustive overview of the open-source software packages available for Bayesian optimization, although the experiments in this thesis featured a custom implementation based on [42, 11, 33]:

- Hyperopt [5]. http://jaberg.github.com/hyperopt/
- SMAC [48]. http://www.cs.ubc.ca/labs/beta/Projects/SMAC/
- DiceOptim [96].
  http://cran.r-project.org/web/packages/DiceOptim/index.html
- Spearmint [112]. https://github.com/HIPS/Spearmint/
- pybo [44]. https://github.com/mwhoffman/pybo/
- MOE [14]. http://yelp.github.io/MOE/
- BayesOpt [76]. https://github.com/rmcantin/bayesopt/
- GPyOpt [34]. https://github.com/SheffieldML/GPyOpt/
- Auto-WEKA [120, 59]. http://www.cs.ubc.ca/labs/beta/Projects/autoweka/

For an in-depth treatment of Bayesian optimization, the reader is referred to the comprehensive works of [68, 10, 103].

2.2 Constrained Bayesian Optimization

An extra dimension of complexity may be added to the formulation of the Bayesian optimization problem if regions of the design space $\mathcal{X}$ are invalid. Examples of such occurrences are in the optimization of Machine Learning algorithm hyperparameters where certain hyperparameter configurations can cause divergences in training or models to run out of memory. An example in chemistry would be the case where a region of chemical space corresponding to a set of similar drug compounds whose efficacy you are trying to evaluate are synthetically inaccessible. This class of problem is known as constrained Bayesian optimization within the Bayesian optimization framework. The constraint in this case, is the fact that regions of the design space are invalid. Typically, the constraint function will be boolean-valued. When such a constraint is known a priori, it may be incorporated into the optimization of the acquisition function. The more interesting case arises when the constraints are a priori unknown. The objective and constraint function may be unknown for two reasons:

1. The constraint has not been observed over the full design space and so it is necessary to interpolate and extrapolate the function values of inputs that have not yet been evaluated.
2. The constraint function may itself be noisy; multiple queries to the constraint function at the same input can result in different function values.

The same principles also apply to the unknown black-box objective function. Problems that adhere to the two points above come under the general class of stochastic programming problems [105]. A natural goal for a statistical model of this kind of problem is to attempt to minimize the objective in expectation and to satisfy the constraints with high probability. Expressed formally the optimization problem is

$$\min_{\mathbf{x}} \mathbb{E}[f(\mathbf{x})] \text{ s.t. } \Pr(\mathcal{C}(\mathbf{x})) \geq 1-\delta \tag{2.2}$$

where $\mathcal{C}(\mathbf{x})$ is the boolean function representing the constraint condition and $1-\delta$ is some user-specified minimum confidence that the constraint is satisfied. These two features combined comprise a probabilistic constraint, the requirement that the constraint be satisfied with high probability. (2.2) describes the Bayesian optimization problem formulation in this thesis. The black-box objective function is noisy because a single latent point may decode to multiple molecules and hence obtain different values for drug-likeness. The constraint is that a given latent point must decode successfully a large fraction of the times decoding is attempted. There is no valid estimate of the value of this fraction because in practice only a finite number of decodings are attempted. In such a fashion, the constraint can be considered to be noisy. The formulation of constrained Bayesian optimization given above can be extended to the case of multiple constraints [29].

Other aspects to consider in this setting include whether the problem has a coupled or a decoupled structure. In the coupled setting the objective and the constraint are both evaluated at each iteration. If the problem has a decoupled structure, this can be exploited in the design of the search strategy as a form of multi-task Bayesian optimization [60, 130, 117]. A related aspect is the topic of cost-sensitive Bayesian optimization [112], where some regions of the design space may be more expensive to evaluate relative to others. For our purposes, the problem is coupled and all regions of the design space are assumed to be equally expensive to evaluate. Although in this work there is no use of data collection strategies for decoupled or cost-sensitive problems, references will be provided in the general discussion on data collection strategies. Now that the problem structure has been outlined, it is necessary to describe the two components of the solution:

1. The modelling of the black-box objective and the constraint.
2. The data collection strategy.

2.2.1 Modelling

The empirical performance of a statistical model for Bayesian optimization hinges on the ability to satisfy two antagonistic factors:

1. The ability to perform full Bayesian inference on the posterior over unknown objective functions.
2. The ability to scale to large datasets and/or large function evaluation budgets.

With regard to point 1., barring some technical difficulties in encoding prior information, Bayesian inference has been identified as an optimal calculus for dealing with uncertain or incomplete information [122, 16, 50, 72]. As such, it is natural that one would wish to perform full Bayesian inference on the posterior over functions in order to obtain well-calibrated uncertainty estimates. Uncertainty estimates are responsible for guiding exploration of the unknown black-box function and hence the omission of artefacts is desirable. Typically, Gaussian Processes (GPs) have been the model of choice for Bayesian optimization applications due to their ability to meet the criteria of point 1. This being said, GPs fall down heavily on point 2, displaying algorithmic complexity of $O(N^3)$ in the number of data points N. Given that the molecular dataset featured in this thesis features $N = 250,000$, the review will focus on two methods that satisfy point 2 at the expense of point 1, namely, Sparse Gaussian Processes to model the black-box objective function and Bayesian Neural Networks (BNNs) to model the black-box constraint function. The disadvantage of these models is that in scaling to large datasets, the ability to carry out full Bayesian inference is sacrificed and hence a large field of work is now devoted to performing approximate inference. The antecedent of the Sparse Gaussian Process, the Gaussian Process, will be discussed as a means of introducing the Sparse Gaussian Process. For sake of completeness it should be mentioned that other models have featured in work on Bayesian optimization such as Random Forests [48] and a Bayesian linear regression model built from features learned from deep neural networks [113]. The discussion will focus on the particular instantiations of these models that featured in the experimental work but will mention, for reference, related methods. The discussion will also aim to highlight the settings in which the different models might be chosen based on their relative advantages and disadvantages.

Gaussian Processes

A sparse Gaussian Process is used in this thesis to model the continuous-valued black-box objective function. In order to discuss the particular sparse Gaussian Process model implemented for the experiments, it will first be necessary to discuss the general full Gaussian Process and as a prerequisite its definition in GP-regression. The introduction will adopt the function space view of the Gaussian Process. In the context of Bayesian optimization, a Gaussian Process is used to describe the prior distribution over possible black-box objective functions. For the alternative weight-space view of Gaussian Processes the reader is referred to [94]. Formally:

Definition 2.2.1. A Gaussian Process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

In the context of Gaussian Process Regression, a single random variable consists of the value of the function $f(\mathbf{x})$ at location $\mathbf{x}$. The Gaussian Process is specified by a mean function

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \tag{2.3}$$

and a covariance function

$$k(\mathbf{x},\mathbf{x}') = \mathbb{E}[(f(\mathbf{x} - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \tag{2.4}$$

and is written as

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big). \tag{2.5}$$

In practice, the prior mean function is typically set to a constant and inferred from data. Henceforth, it will be taken to be $m(\mathbf{x}) = 0$. Formally, the covariance function specifies the covariance between pairs of random variables,

$$k(\mathbf{x}, \mathbf{x}') = \text{cov}\big(f(\mathbf{x}), f(\mathbf{x}')\big). \tag{2.6}$$

Choice of Kernel

In this case, the random variables corresponding to the pair of function outputs $f(\mathbf{x})$ and $f(\mathbf{x}')$ are written as a function of the pair of inputs $\mathbf{x}$ and $\mathbf{x}'$. Intuitively, the covariance function captures the notion of similarity in input space by expressing the notion that close inputs $\mathbf{x}$ will map to close targets $f(\mathbf{x})$. The covariance function is also known as a kernel owing to the definition of a kernel from integral operator theory as a general name for a function k mapping a pair of inputs $\mathbf{x} \in \mathcal{X}$, $\mathbf{x}' \in \mathcal{X}$ into $\mathbb{R}$. The covariance function will be termed kernel from here on in. Two popularly used kernels are the squared exponential kernel

$$\mathrm{k}_{\mathrm{SQE}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(\frac{-|\mathbf{x} - \mathbf{x}'|^2}{2\ell^2}\right) \tag{2.7}$$

and the Matern kernel

$$\mathrm{k}_{\mathrm{Matern}}(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\,|\mathbf{x} - \mathbf{x}'|}{\ell}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}\,|\mathbf{x} - \mathbf{x}'|}{\ell}\right) \tag{2.8}$$

where σ_f^2 is the amplitude parameter, ℓ is the lengthscale parameter, K_ν is a modified Bessel function of the second kind, Γ is the gamma function and ν is some non-negative parameter of the kernel function. Typical values for ν are $\frac{3}{2}$ and $\frac{5}{2}$ for machine learning applications [94]. Inspecting the form of the squared exponential covariance function, it can be seen that the covariance between a pair of inputs decays exponentially with their distance in the input space. The lengthscale parameter can be thought of as a coefficient for this exponential decay dictating the characteristic lengthscale of a given dimension in input space. Informally, at large distances, the inputs become decorrelated. The principle is similar in statistical physics, where correlation functions dictate how correlated the output of a process is through time. Henceforth, hyperparameters such as ℓ and ν will be suppressed in the notation as θ. A further characteristic of kernel functions is stationarity. Stationary kernel functions are functions of $\mathbf{x}-\mathbf{x}'$ and as such are invariant to translation in the input space. A stationary kernel may also be isotropic if it is a function of $|\mathbf{x}-\mathbf{x}'|$ whereupon the output becomes invariant to all rigid motions. Such stationary, isotropic kernels are also known as radial basis functions [94]. It is possible to specify an anisotropic kernel by specifying

$$\left|\mathbf{x}-\mathbf{x}'\right|^2 = (\mathbf{x}-\mathbf{x}')^\top M(\mathbf{x}-\mathbf{x}'). \tag{2.9}$$

for some positive semidefinite M. When $M = \text{diag}(\ell)^{-2}$ one recovers a kernel that implements automatic relevance determination (ARD) [84]. Informally, ARD dictates how important different dimensions are. At large values of the lengthscale for a given dimension, the output of the covariance function becomes invariant to the choice of input locations. This has the effect of removing the dimension from the inference. General $M's$ have been examined by [80] and [89].

Making Predictions

Returning to the application of Bayesian optimization, we are interested in the following realistic modelling situation:

$$y(\mathbf{x}) = f(\mathbf{x}) + \varepsilon\sigma_y \tag{2.10}$$

where

$$p(\varepsilon) = \mathcal{N}(0,1). \tag{2.11}$$

In this instance, we don't have access to function values, but rather only noisy observations thereof. Further to this, we assume that the noise is additive and iid. In this thesis, the main application of Bayesian optimization to the molecular data will be a problem of this kind, however in the toy problem considered, we will assume that we have access to the true function values when evaluating the black-box objective. This idealised scenario is realised in a few select applications such as computer simulations [97]. In the case of the modelling situation with noisy observations of the black-box objective we wish to place a Gaussian Process prior over the non-linear function $f(\mathbf{x})$ which governs the underlying generative process we wish to model:

$$p(f(\mathbf{x})|\theta) = \mathcal{GP}\big(0, K(\mathbf{x}, \mathbf{x}')\big) \tag{2.12}$$

where $K(\mathbf{x}, \mathbf{x}')$ is now a matrix such that $[K]_{ij} = k(\mathbf{x_i}, \mathbf{x_j})$ and θ represents the set of hyperparameters. The prior over observations will be given as

$$p(y(\mathbf{x})|\theta) = \mathcal{GP}\big(0, K(\mathbf{x}, \mathbf{x}') + I\sigma_y^2\big). \tag{2.13}$$

In this case the additive term $I\sigma_y^2$ follows from the independence assumption for the noise. Being in possession of some training data consisting of observed target values $\mathbf{y}$, we can arrive at the key predictive equations by writing down the joint distribution of the observed target values alongside the function values at the test locations under the prior

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X') + \sigma_y^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix}\right). \tag{2.14}$$

Using the relation

$$p(\mathbf{f}_*|\mathbf{y}) = \frac{p(\mathbf{f}_*, \mathbf{y})}{p(\mathbf{y})} \tag{2.15}$$

it becomes possible to condition the joint prior on the observations, restricting the joint prior to contain only the functions that agree with the observed data points. We arrive at

$$p(\mathbf{f}_*|X, \mathbf{y}, X_*) = \mathcal{N}\big(\bar{\mathbf{f}}_*, \mathrm{cov}(\mathbf{f}_*)\big) \tag{2.16}$$

for the predictive distribution, where

$$\bar{\mathbf{f}_*} = K(X_*, X)[K(X, X) + \sigma_y^2 I]^{-1}\mathbf{y} \tag{2.17}$$

is the predictive mean at the test locations X_* and

$$\mathrm{cov}(\mathbf{f}_*) = K(X_*,X_*) - K(X_*,X)[K(X,X) + \sigma_y^2 I]^{-1} K(X,X_*) \tag{2.18}$$

is the predictive uncertainty. Some interesting observations include the fact that the predictive mean $\bar{\mathbf{f}}_*$ is linear in the data $\mathbf{y}$ and that the predictive uncertainty can be thought of as comprising a term corresponding to the prior uncertainty $K(X_*,X_*)$, minus a term, $K(X_*,X)[K(X,X) + \sigma_y^2 I]^{-1} K(X,X_*)$ corresponding to the reduction in uncertainty acquired following the observation of the target values.

Model Selection

A further attractive feature of Gaussian Processes can be elucidated by examining them in the context of hierarchical Bayesian model selection [69]. The model hierarchy may be specified in three tiers corresponding to:

1. Model Parameters
2. Model Hyperparameters
3. Model Structures

As the Gaussian Process is a nonparametric approach, model parameters are typically obtained in the form of the posterior distribution over functions. Model hyperparameters include such entities as the lengthscales and amplitude parameters that were discussed in the context of kernels. An example of a model structure would be the specification of a parametric form of kernel. The Gaussian Process framework allows us to undertake model selection at the second tier through the optimisation of the marginal likelihood

$$\begin{aligned}\log p(\mathbf{y}|X,\theta) = & \underbrace{-\frac{1}{2}\mathbf{y}^\top (K_\theta(X,X') + \sigma_y^2 I)^{-1}\mathbf{y}}_{\text{encourages fit with data}} \\ & \underbrace{-\frac{1}{2}\log|K_\theta(X,X') + \sigma_y^2 I|}_{\text{controls model capacity}} - \frac{N}{2}\log(2\pi).\end{aligned} \tag{2.19}$$

The notation of $K_\theta(X,X')$ in this instance, indicates explicit dependence on the hyperparameters θ. and N denotes the number of observations. The optimization of the marginal likelihood (also termed the evidence [71]) encompasses the Occam factor [93] in so far as it favours models of intermediate complexity. This see-saw balancing effect is seen in the two terms in equation 19, one corresponding to a term that promotes a fit with the data and another term that penalises models that are too complex. Unlike with other methods, the marginal likelihood is typically analytic in the case of the Gaussian Process framework. The optimization of the marginal likelihood is usually performed in one of two ways. The first approach is based on optimizing (**??**) directly and results in a type II maximum likelihood estimate for the hyperparameters [94]. This approach can overfit if the data is sparse. The second approach involves performing full Bayesian inference at the hyperparameter level by marginalising out the hyperparameters at prediction time. This is undertaken by evaluating the integral

$$p(\mathbf{y}|X,H_i) = \int p(\mathbf{y}|X,\theta,H_i)\overbrace{p(\theta|H_i)}^{\text{hyperprior}}d\theta \tag{2.20}$$

and given an appropriate specification for the hyperprior can give rise to even more robust predictive uncertainty estimates relative to performing full Bayesian inference at the parameter level. Unfortunately this integral is often intractable in practice. Nonetheless approximate solution methods exist in the form of the Laplace approximation [70] which can work well in the case of peaked posteriors over hyperparameters, and methods such as slice sampling [85], which has been put to practical use for Bayesian optimization applications [112]. In the context of Bayesian optimization, Gaussian Processes possess the following advantages and disadvantages in their use as the surrogate model for the objective function.

Advantages

1. Availability of full Bayesian inference over the posterior to obtain a closed form posterior predictive distribution with well-calibrated uncertainty estimates
2. Availability of an analytic form for the marginal likelihood which enables Bayesian model selection at the hyperparameter level. Overfitting is hence less of an issue, especially if full Bayesian inference is available at the hyperparameter level.

Disadvantages

1. Carrying out full Bayesian inference is $O(N^3)$ in the number of observations due to the necessity of inverting the covariance matrix $[K(X,X)+\sigma_y^2 I]^{-1}$ which appears in the expressions for the predictive mean, predictive covariance and marginal likelihood. The covariance matrix here is given for the case of noisy observations. In practice, the

Cholesky decomposition may be computed once and stored resulting in $O(N^2)$ complexity for subsequent predictions given a fixed set of kernel hyperparameters. Given however, that the kernel hyperparameters must be recomputed at every iteration of Bayesian optimization this fact does not mitigate the $O(N^3)$ cost for large function evaluation budgets.

2. Non-Gaussian predictive distributions are not supported due to issues with intractability. In practice, one may want to use heavier-tailed non-Gaussian predictive distributions in order to be robust to outliers. Recent work has proposed a GP model based on the student T-distribution as an alternative [79].

3. A parametric form of kernel must be chosen although there have been some efforts to automate the process of doing this [21].

For the purposes of Bayesian optimization, it is point 1. under the disadvantages that is the most egregious property of GPs in practice. The $O(N^3)$ complexity is prohibitive both for large function evaluation budgets and large data sets and indeed, this has motivated the development of the sparse approximations which will be discussed in the following section. For alternative introductions to Gaussian Processes, the reader is referred to [1, 21, 94] in increasing order of technical detail.

Sparse Gaussian Processes

The computational requirements of full Gaussian Processes motivate the use of sparse approximations. Common to all sparse approximations is that only a subset of the latent variables are treated exactly while the rest are given some approximate but computationally cheaper treatment [92]. As such, sparse approximations are characterised by focussing inference on a set of quantities that represent approximately the posterior over functions [3]. Concretely, these approximations can take the form of pseudodata, spectral representations [63] or more abstract representations [62]. The discussion here will focus on the pseododata method given that a variant of this method was used in all experiments reported. The method of pseudopoints may be described as an attempt to summarise the original dataset witih $M < N$ data points. The M data points are known as pseudodata. The aim is to construct these M data points in such a way that the training process maintains a high degree of fidelity relative to the training process for the original dataset. The computational advantage of this approach is that $O(N^3)$ time complexity becomes $O(NM^2)$ and $O(N^2)$ memory complexity becomes $O(NM)$. The pseudopoint is "sparse" because it induces structure in the covariance matrix featuring zeros at various places. This sparse matrix structure then facilities efficient computation. It is possible to broadly categorise pseudopoint methods into two familes:

Approximate Model, Exact Inference The defining feature of this family of methods is an approximate generative model. The approximate generative model is chosen based on a divergence measure that either minimises the distance between models at modelling time or at a stage after the class of models has been restricted to a class in which inference is efficient. Examples in this family include the Deterministic Training Conditional (DTC)[102], the Fully Independent Training Conditional (FITC) [109] and the Partially Independent Training Conditional (PITC) [92].

Exact Model, Approximate Inference The defining feature of this family of methods is that the exact generative model is maintained and all "approximations" occur at inference time. At inference time, one measures the similarity to the exact posterior. Examples in this family include Variational Free Energy methods [121] and methods based on Expectation Propagation (EP) [110, 91, 11].

From the methods provided above, the two most popular are FITC and VFE [3], as unlike previous methods [108, 102, 92], they provide an approximation to the marginal likelihood which may be leveraged in conjunction with gradient-based optimization to learn the hyperparameters from data. The method employed in the experiments in this thesis was FITC and so it will now be outlined.

FITC Commensurate with FITC's interpretation as an approximate model in which exact inference is performed, the interpretation of modifying the prior of a full Gaussian Process to

$$p(f(\mathbf{x})|\theta) = \mathcal{GP}\big(0, Q_{\mathrm{ff}} + \mathrm{diag}[K_{\mathrm{ff}} - Q_{\mathrm{ff}}]\big) \tag{2.21}$$

will be used, where

$$K_{\mathrm{ff}} = K_{\theta}(X, X') \tag{2.22}$$

is a change of notation that is consistent with the notation of [3] and facilitates the description of

$$Q_{\mathrm{ff}} = K_{\mathrm{fu}} K_{\mathrm{uu}}^{-1} K_{\mathrm{uf}} \tag{2.23}$$

as a low-rank matrix introduced to reduce the size of the matrix inversion to M. Using this interpretation of the prior, the negative log marginal likelihood used to train the methods is given as

$$\mathcal{F} = \frac{N}{2}\log(2\pi) + \underbrace{\frac{1}{2}\log|Q_{\mathrm{ff}} + \mathrm{diag}[K_{\mathrm{ff}} - Q_{\mathrm{ff}}] + \sigma_y^2 I|}_{\text{controls model capacity}} \tag{2.24}$$

$$+ \underbrace{\frac{1}{2}\mathbf{y}^\top(Q_{\mathrm{ff}} + \mathrm{diag}[K_{\mathrm{ff}} - Q_{\mathrm{ff}}] + \sigma_y^2 I)^{-1}\mathbf{y}}_{\text{encourages fit with data}}. \tag{2.25}$$

$\mathcal{F}$ is the symbol used to denote the negative log marginal likelihood as opposed to the positive log marginal likelihood given for the full Gaussian Process in section 2.1.1. The expression is analogous in so far as it contains two terms corresponding to a data fit term and a complexity penalty, the difference being that the matrix inversion of $(Q_{\mathrm{ff}} + \mathrm{diag}[K_{\mathrm{ff}} - Q_{\mathrm{ff}}] + \sigma_y^2 I)^{-1}$ is now cheaper to compute. $(Q_{\mathrm{ff}} + \mathrm{diag}[K_{\mathrm{ff}} - Q_{\mathrm{ff}}] + \sigma_y^2 I)$ may be thought of as a covariance ellipse, data outside of which is penalised. In turn the complexity term $\frac{1}{2}\log|Q_{\mathrm{ff}} + \mathrm{diag}[K_{\mathrm{ff}} - Q_{\mathrm{ff}}] + \sigma_y^2 I|$ characterises the volume of possible datasets compatible with the data fit term. As in the full Gaussian Process scenario, the mechanism of Occam's razor penalises the model for being able to predict too many datasets. The recent empirical observations of [3] would appear to show that FITC has a tendency to underestimate the noise variance σy^2 by using the diagonal term $\mathrm{diag}(K_{\mathrm{ff}} - Q_{\mathrm{ff}})$ as heteroscedastic (input dependent) noise to account for differences between the sparse and full GP approaches. By placing inducing inputs near training data that lie close to the mean, the heteroscedastic noise term may be locally shrunk, giving rise to a smaller complexity penalty. If the noise variance σ_y^2 is altered in this manner, it loses its interpretation as the amount of uncertainty in the data that can't be explained [3]. This may cause issues for Bayesian optimization applications which rely on being able to draw a distinction between inherent noise and uncertainty in the surrogate model which may be reduced. The FITC approximation is used to model the black-box objective for all experiments reported here. The advantages and disadvantages of sparse GP methods in relation to Bayesian optimization may be summarised as follows:

Advantages The principle advantage sparse GPs possess over full GPs is scale. Typically full GPs are limited to datasets numbering the thousands of data points. The reduced time and memory requirements of sparse GPs make running inference on medium and large datasets computationally tractable.

Disadvantages The principle disadvantage of sparse GPs is the reduction in quality of uncertainty estimates that arises as a result of the approximation.

Bayesian Neural Networks

Bayesian Neural Networks (BNNs) have been a recent addition to the gamut of Bayesian optimization literature [113, 114, 57]. As in the case of sparse GPs, exact computation of the posterior is intractable with BNNs, rendering approximate inference a necessity. There are a number of approximate inference techniques currently being applied to BNNs, but one common characteristic is that the predictive distribution is approximated by averaging over samples $\{\mathcal{W}_k\}_{k=1}^{K}$, of the network weights

$$p(\mathbf{y}|\mathbf{x},\mathcal{D}) \approx \frac{1}{K}\sum_{k=1}^{K} p(\mathbf{y}|\mathbf{x},\mathcal{W}_k). \tag{2.26}$$

$\mathcal{D} = \{\mathbf{x_n}\}_{n=1}^{N}$ are the observations in this instance. Drawing analogies to Gaussian Process inference, the sampling over network weights equates to a sampling over functions. In all experiments reported here, we use the method of Black-box alpha divergence minimization [42] to undertake approximate inference. Black-box alpha divergence minimization is implemented by approximating the true posterior by the minimization of the αdivergence distance metric [106]. The original paper on the method [42] showed that by changing the value of α in this divergence, it is possible to interpolate between Variational Bayes [52] and EP [81] corresponding to the limits of $\alpha = 0$ and $\alpha = 1$ respectively. Intuitively, these limits may be thought of as interpolating between solutions that fit a single mode in the true posterior ($\alpha = 0$) or that aim to cover multiple modes in the true posterior ($\alpha = 1$) [42, 40]. Recent work, including that of the original paper, has shown empirically that a value of α of 0.5 produces good predictions [42, 18]. In this thesis, the Black-box alpha divergence minimization method is used in conjunction with a BNN that models the constraint for all experiments performed. In the context of Bayesian optimization the advantages and disadvantages of BNNs are as follows:

Advantages BNNs can scale to high-dimensional datasets and in such scenarios can be preferred to full GPs.

Disadvantages As in the case of sparse GPs, BNNs require approximate inference methods and hence produce uncertainty estimates of a lower calibre relative to full GPs.

2.2.2 Data Collection Strategy

The ideal scenario for Bayesian optimization would feature a loss function that described how optimal a sequence of queries to the black-box objective are. Such a loss function is available in the form of regret. Regret is defined as the difference between the reward obtained and the best possible reward obtainable at iteration n of the Bayesian optimization procedure. Regret may be expressed as cumulative regret,

$$R_N = \frac{1}{N} \sum_{n=0}^{N} r_n, \tag{2.27}$$

where r_n gives the instantaneous regret at iteration n. or as simple regret r_N which is the regret obtained at the final iteration of the Bayesian optimization procedure. Simple regret captures the goal of Bayesian optimization better in that penalisation is not incurred for poor function evaluations leading to the final recommendation. Intuitively, you are not punished for exploration. The usage of the regret metric features in the minimum expected risk framework, where the expected loss is minimized in order to produce an optimal set of queries. However, in practice, the true sequential risk, up to a full evaluation budget is typically intractable. Despite some attempts to approximate this ideal look-ahead policy for a finite number of future steps [88, 30, 35], most research focusses on the use of myopic heuristics known as acquisition functions. Taking the case of a minimization problem as an example, the general desiderata for an acquisition function is that the values should be low where the objective is potentially low. This corresponds to regions of the surrogate function where the uncertainty is high and the predictive mean is low. There are many examples of acquisition functions. Some of the more widely-used variants may be categorised as follows:

1. Improvement-based acquisition functions - Characterised by favouring points likely to improve upon an incumbent target. Examples include Probability of Improvement (PI) [61] as well as Expected Improvement (EI) [82, 51].

2. Optimistic acquisition functions - “Optimistic in the face of uncertainty”. Members of this class use a fixed probability best case scenario according to the model to decide on a query [103]. One example is the Upper Confidence Bound (UCB) acquisition function [115].

3. Information-based acquisition functions - These methods work with the posterior distribution over the unknown minimizer $\mathbf{x}*$. This posterior distribution being implicitly induced by the posterior over objective functions f. Examples in this class include Thompson

Sampling (TS) [119], the Informational Approach to Global Optimization (IAGO) [124], Entropy Search (ES) [39] and Predictive Entropy Search (PES) [41].

In addition, there has been work that claims that there is no single acquisition function that is optimal for every problem [43], giving rise to a field of literature concerned with the design of portfolios of acquisition functions [43, 45, 104]. The idea being, that at each iteration of Bayesian optimization, an ensemble of acquisition functions is used to decide upon the next query point.

This thesis makes use of a variant of EI known as Expected Improvement with Constraints that is applicable for constrained optimization problems. As such, EI will be introduced as a pre-requisite for the discussion of EIC.

Expected Improvement

Expected Improvement (EI) is the expected amount of improvement over a specified target η if the objective function is evaluated at $\mathbf{x}$. It is defined to be

$$\mathrm{EI}(\mathbf{x}) = \mathbb{E}[(\eta - y)_+] = \int_{-\infty}^{\infty} (\eta - y)_+ p(y|\mathbf{x}) dy \tag{2.28}$$

where $p(y|\mathbf{x})$ is the posterior predictive marginal density of the objective function evaluated at $\mathbf{x}$. $(\eta - y)_+ \equiv \max(0, \eta - y)$ is the improvement over the target η. Intuitively, one can think of the improvement as being a measure that only starts to accumulate value once you start integrating the predictive density below the line corresponding to $y = \eta$ for a given $\mathbf{x}$. The Expected Improvement heuristic encodes what this value is as a function of $\mathbf{x}$. Typically, the target η, also known as the incumbent, is set to be the current recommendation. In turn, for the unconstrained formulation of Bayesian optimization the recommendation may either be defined as the minimum observed value over previous observations [112] or as the minimum of the expected value of the objective [9]. For the constrained formulation of Bayesian optimization considered in this thesis, the current recommendation is defined to be the minimum expected value of the surrogate model such that a set of probabilistic constraints are satisfied. An extension to EI known as Expected Improvement with Constraints is one mean of achieving this.

Expected Improvement with Constraints

EIC may be thought of as EI that offers improvement only when a set of constraints are satisfied. EIC was first introduced by [100] and has received more recent attention from [111, 25]. It is defined as

$$\mathrm{EIC}(\mathbf{x}) = \mathrm{EI}(\mathbf{x})\,\mathrm{Pr}(\mathcal{C}(\mathbf{x}) \geq 0) \tag{2.29}$$

where $\mathrm{Pr}(\mathcal{C}(\mathbf{x}) \geq 0)$ denotes the probability that a boolean constraint $\mathcal{C}(\mathbf{x})$ is satisfied. The implicit η suppressed in $\mathrm{EI}(\mathbf{x})$ in equation 29, may be set in an analogous way to vanilla Expected Improvement as:

1. The best observation in which all constraints are observed to be satisfied [28].
2. The minimum of the posterior mean such that all constraints [28].

The latter approach is adopted for the experiments performed in this thesis. Additionally, if at the stage in the Bayesian optimization procedure where a feasible point has yet to be located, the form of acquisition function used is that defined by [28]

$$\mathrm{EIC}(\mathbf{x}) = \begin{cases} \mathrm{Pr}(\mathcal{C}(\mathbf{x}) \geq 0)\mathrm{EI}(\mathbf{x}), & \text{if } \exists \mathbf{x}, \mathcal{C}(\mathbf{x}) \\ \mathrm{Pr}(\mathcal{C}(\mathbf{x}) \geq 0), & \text{otherwise} \end{cases} . \tag{2.30}$$

The intuition behind (2.30) is that if the probabilistic constraint is violated everywhere, the acquisition function selects the point having the highest probability of being in the feasible region. The algorithm ignores the objective until it has located the feasible region. EIC is the acquisition function of choice in all experiments reported.

Parallel Bayesian Optimization

Traditional Bayesian optimization performs function evaluations in sequence. With the advent of computing clusters, one natural question is whether it is possible to submit multiple function evaluations in parallel to reduce wall-clock time [112, 47]. There are two known approaches for this:

1. Submit queries in a batch.
2. Submit queries asynchronously.

Approach 1 above, requires the optimization of a multipoints acquisition function in $q \times D$ dimensions. Such an acquisition function has been proposed by [30] but in general it is favourable to try and avoid the $q \times D$ dimensional optimization if possible. Naively selecting points for querying without evaluating them and re-training the model would result, with the exception of a sampling-based procedure like Thompson Sampling, in the same point repeatedly being chosen within an iteration. As such, approach 2 above maximises the acquisition function at each iteration to yield a suggestion $\hat{\mathbf{x}}$ given a set of pending queries $\{\mathbf{x}_\mathbf{j}^{\text{pend}}\}$ [100]. Intuitively, a point is selected as in the standard Bayesian optimization procedure, but instead of being passed through the expensive evaluation process it becomes a "pending query". The objective function value at this pending query, along with the effect it would have on the re-trained model, is then fantasised using some computationally cheap process. One means of enacting this approximate model update is by using the Kriging-Believer algorithm which uses the current estimate of the GP predictive mean for a query location as a proxy for the expensive black-box function evaluation of the query location. This is possible because conditioning on an observation equal to the posterior mean leaves the mean value of the function unchanged elsewhere and the model variance can be cheaply updated with a rank-one update to the Cholesky decomposition of $K(X_0, X_0)$, the old covariance matrix, to yield the new covariance matrix $K(X, X)$ [20]. The Kriging-Believer algorithm has been used for Bayesian optimization by [5] and [28] and will be used for all experiments in this thesis.

Chapter 3

Results

3.1 Outline

Having introduced the mathematical formulations of Bayesian optimization and constrained Bayesian optimization as paradigms of search in the face of uncertainty, the results of this chapter will investigate the hypothesis that constrained Bayesian optimization is a more natural paradigm for the problem of Automatic Chemical Design. The chapter will begin in section 3.2 with a validity test of the algorithm for constrained Bayesian optimization in the setting of a toy constrained optimization problem featuring the Branin-Hoo function. In section 3.3, some of the background necessary for understanding the application of Bayesian optimization to Automatic Chemical Design will be elaborated on. Section 3.4 will feature the principle results of the thesis in terms of answering whether constrained Bayesian optimization offers any improvement over unconstrained Bayesian optimization. Lastly, section 3.5 will investigate potential applications for Automatic Chemical Design with constrained Bayesian optimization.

3.2 Branin-Hoo Function

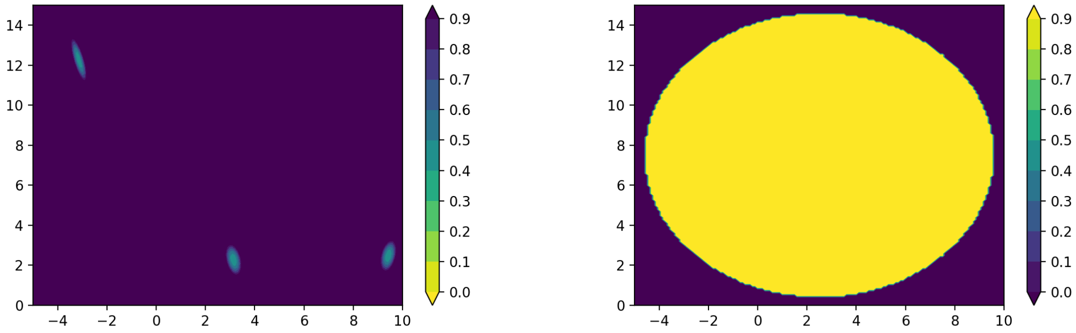
The Branin-Hoo function will act as a toy problem on which to test the validity of the custom algorithmic implementation for constrained Bayesian optimization. The optimization of the Branin-Hoo function has long been a benchmark for the comparison of global optimization algorithms [19] and more recently has become a benchmark for the comparison of Bayesian optimization algorithms [22]. The particular variant of the Branin-Hoo optimization that will be of interest here is the constrained formulation of the problem featured in [29]. The 2D Branin-Hoo function is given as

$$f(\mathbf{x}) = \left(x_2 - 5.1\frac{x_1^2}{4\pi^2} + 5\frac{x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10 \tag{3.1}$$

and has three global minima at the coordinates $(-\pi, 12.275), (\pi, 2.275)$ and $(9.42478, 2.475)$. In order to formulate the problem as a constrained optimization problem, a disk constraint on the region of feasible solutions,

$$(x_1 - 2.5)^2 + (x_2 - 7.5)^2 \leq 50, \tag{3.2}$$

is introduced. In contrast to the formulation of the problem in [29], the disk constraint is coupled in this scenario in the sense that the objective and the constraint will be evaluated jointly at each iteration of Bayesian optimization. In addition, the observations of the black-box objective function will be assumed to be non-noisy. The minima of the Branin-Hoo function as well as the disk constraint are illustrated in Figure 3.1.



(a) Minima Locations

(b) Disk Constraint

Fig. 3.1 Constrained Bayesian optimization of the 2D Branin-Hoo Function.

The disk constraint eliminates the upper-left and lower-right solutions, leaving a unique global minimum at $(\pi, 2.275)$. Given that the implementation of constrained Bayesian optimization considered here relies on the use of a sparse GP as the underlying statistical model of the black-box objective and as such is designed for scale as opposed to performance, the results will not be compared directly against those of Gelbart et al. [29] who use a full GP to model the objective. It will be sufficient to compare the performance of the algorithm against random sampling. Both the sequential Bayesian optimization algorithm and the parallel implementation utilising the Kriging-Believer algorithm will be tested.

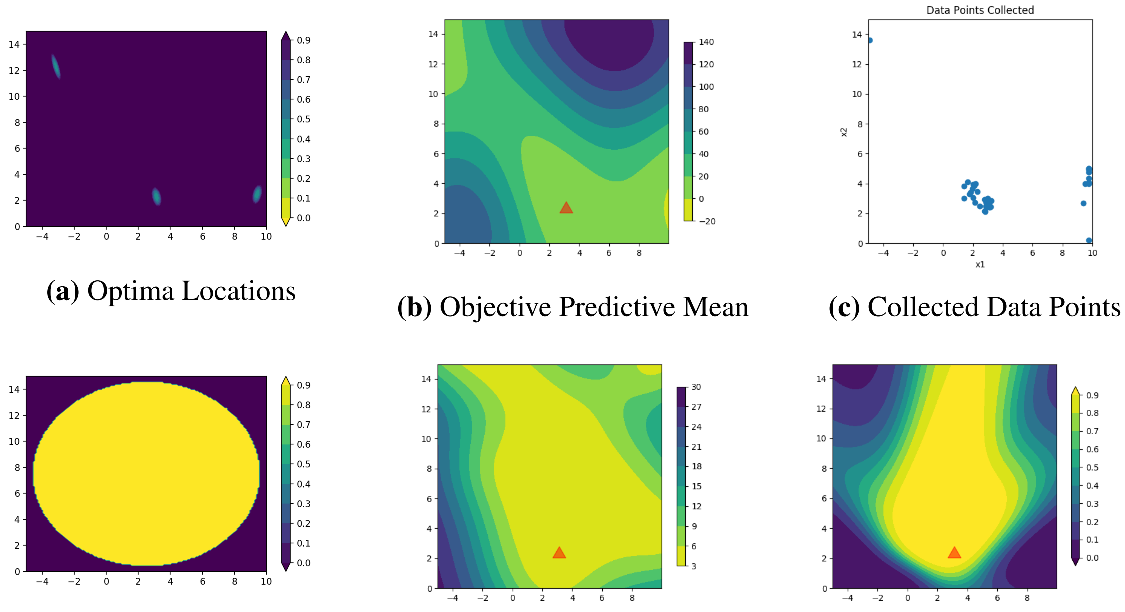
3.2.1 Implementation

A Sparse GP featuring the FITC approximation and based on the implementation of [11] is used to model the black-box objective function. The kernel choice is exponentiated quadratic with ARD lengthscales. The number of inducing points M was chosen to be 20 in the case of sequential Bayesian optimization, and 5 in the case of the parallel Bayesian optimization using the Kriging-Believer algorithm. The sparse GP is trained for 400 epochs using Adam [53] with the default parameters and a learning rate of 0.005. The minibatch size is chosen to be 5. The extent of jitter, a parameter that provides stabilization in the computation of the posterior predictive distribution such that the Cholesky decomposition will not fail [94], is chosen to be 0.00001. A Bayesian Neural Network, adapted from the MNIST digit classification network of [42] is trained using black-box alpha divergence minimization and models the constraint function.

The network has a single hidden layer with 50 hidden units, Gaussian activation functions and logistic output units. The mean parameters of q, the approximation to the true posterior, are obtained by sampling from a zero-mean Gaussian with variance $\frac{2}{d_{\text{in}}+d_{\text{out}}}$ according to the method of [31], where d_{in} is the dimension of the previous layer in the network and d_{out} is the dimension of the next layer in the network. The value of α is taken to be 0.5, minibatch sizes are taken to be 10 and 50 Monte Carlo samples are used to approximate the expectations with respect to q in each minibatch. The BNN adapted from [42] was implemented in the Theano library [2, 118]. The LBFGs method [66] was used to optimize the EIC acquisition function in all experiments.

3.2.2 Results

The results of the sequential constrained Bayesian optimization algorithm with EIC are shown in Figure 3.2. The algorithm was initialised with 50 labeled data points drawn uniformly at random from the grid depicted. 40 iterations of Bayesian optimization were carried out.



(a) Optima Locations

(b) Objective Predictive Mean

(c) Collected Data Points

(d) True Disk Constraint

(e) Objective Predictive Variance

(f) $\Pr(\mathcal{C}(\mathbf{x}) \geq 0)$

Fig. 3.2 Constrained Bayesian Optimization on the 2D Branin-Hoo Function with a Coupled Disk Constraint. **a)** Minima of the true Branin-Hoo function. **b)** Contour plot of the predictive mean of the sparse GP used to model the objective function. Lighter colours indicate lower values of the objective. **c)** Data points collected over 40 iterations of sequential Bayesian optimization. **d)** The disk constraint, where the feasible region is given in yellow. **e)** The predictive variance of the sparse GP used to model the objective function. **f)** The contour learned by the BNN giving the probability of constraint satisfaction.

The figures are provided for illustrative purposes only and show that the algorithm is correctly managing to collect data in the vicinity of the single feasible minimum. Figure 3.3 compares the performance of parallel Bayesian optimization using the Kriging-Believer algorithm against the results of random sampling. Both algorithms were initialised using 10 data points drawn uniformly at random from the grid on which the Branin-Hoo function is defined and were run for 10 iterations of Bayesian optimization. At each iteration a batch of 5 data points were collected for evaluation.

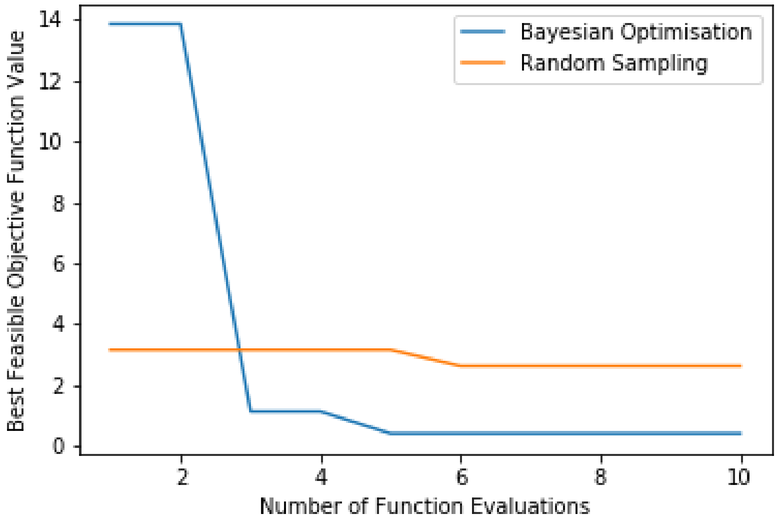


Fig. 3.3 Performance of Parallel Bayesian Optimization with EIC against Random Sampling.

After 10 iterations, the minimum feasible value of the objective function was 0.42 for parallel Bayesian optimization with EIC using the Kriging-Believer algorithm and 2.63 for random sampling. The true minimum feasible value is 0.40. It is worth noting that these results are not statistically significant and are merely provided as a means of demonstrating that the implementation is observing the constraint. It is very likely that the parallel constrained Bayesian optimization algorithm experienced a serendipitous initialisation on the single run shown.

3.2.3 Discussion

The results above act as a proof of concept that the implementation of constrained Bayesian optimization is behaving as expected in so far as it recognises the constraint in the problem and appears to do better than random sampling. Reiterating a point made previously, it would not be fully appropriate to compare the results of the current implementation to that of [29] as the sparse GP surrogate model of the objective is not expected to be competitive with a full GP in terms of its ability to represent the black-box function. It could be worth however, performing some investigation into how much worse the sparse GP performs relative to the full GP in the constrained setting. Another aspect that could be explored is the impact of the initialisation or initial design.It has recently been argued that different algorithms will vary in their performance depending on how much information about the design space there is available [83].

3.3 Background to Automatic Chemical Design

This section provides a brief background of some details of the model featured in [33] that may be needed in order to place the ensuing results in context. Figure 3.4 illustrates the composition of the Automatic Chemical Design model in terms of its constituents.

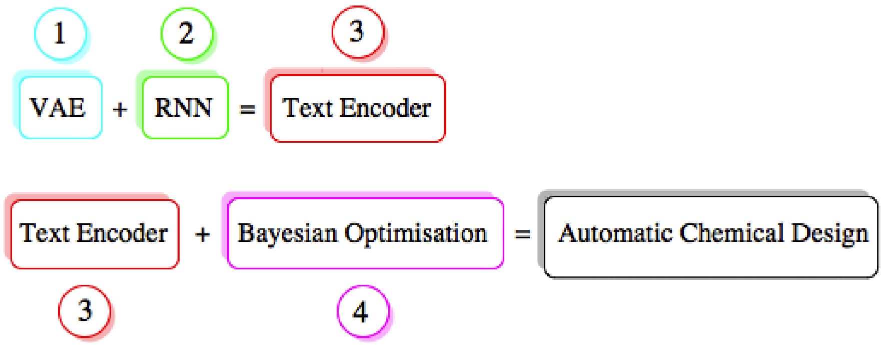


Fig. 3.4 Automatic Chemical Design as a Conglomerate of Smaller Models.

Implementing a VAE with RNN encoder-decoder networks gives rise to a text encoder, which has recently been used to encode sentences in a continuous space [8]. The ability to leverage the text encoder with a text-based representation of molecules known as simplified molecular-input line-entry system, or SMILES [128], gives rise to the possibility of representing a molecule as a continuous-valued vector. The continuous representation is then amenable as a target for continuous optimisation. Thus, by marrying the text encoder with Bayesian optimisation in its latent space, one arrives at the Automatic Chemical Design model. For further information on VAEs and RNNs the reader is referred to [55, 54] for VAEs, [116] for RNNs and [36] for both. An illustration of the SMILES representation for molecules is given for the molecule benzene in Figure 3.5.

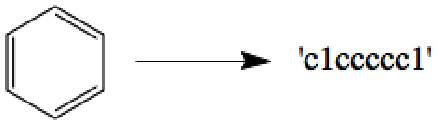


Fig. 3.5 SMILES - A Text Representation for Molecules.

Figure 3.6 offers another view of the Automatic Chemical Design model, illustrating the different representations that feature in the pipeline.

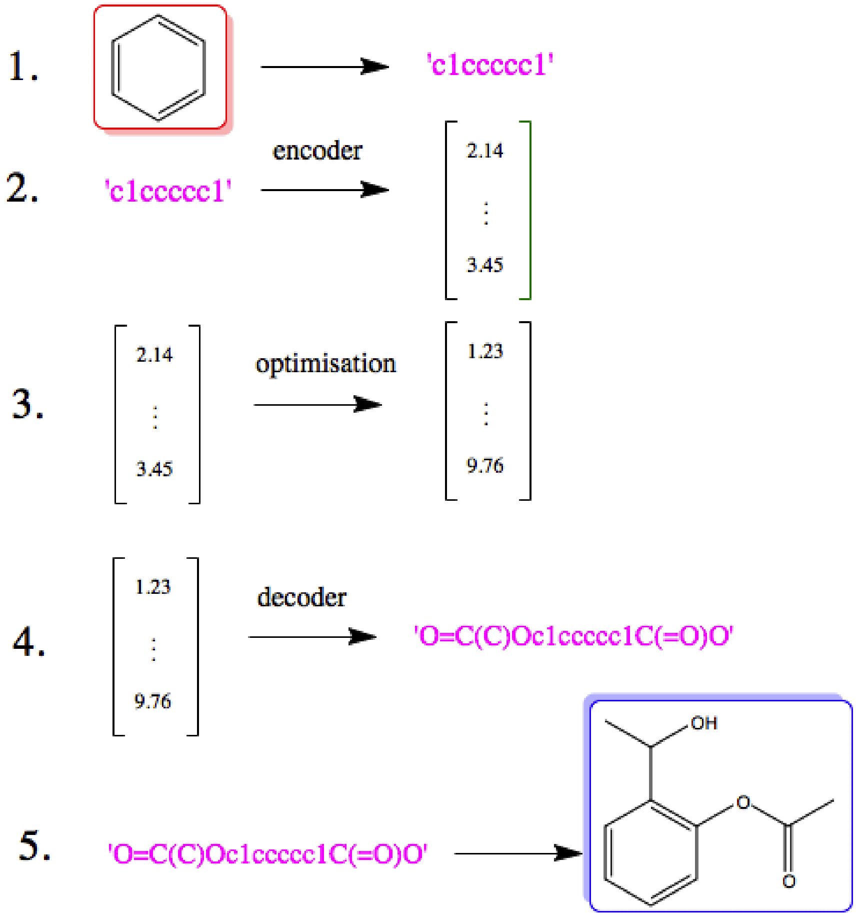


Fig. 3.6 The Model as a Sequence of Representational Transformations. Step 1 shows a benzene molecule being represented in its text-based format as a SMILES string. Step 2 shows the SMILES string for benzene being encoded using a text encoder into a continuous-valued vector of latent points. Step 3 shows the optimization process occurring within the latent space of the variational autoencoder, whereupon some objective function of the latent points is optimized to yield a recommendation for the best point. Step 4 shows the optimal latent point being decoded to a SMILES string. Step 5 shows the now optimal SMILES string being decoded into an optimal molecule.

A further question that requires answering is what the best applications would be for a model that can perform Automatic Chemical Design. The answer to this question is strongly linked to the properties of the objective function optimized in the latent space. The objective function optimized here and also in [33] is

$$J^{\mathrm{logP}}(m) = \mathrm{logP}(m) - \mathrm{SA}(m) - \text{ring-penalty}(m). \tag{3.3}$$

where m denotes a molecule, logP is the water-octanol partition coefficient, a molecular property whose value is an important estimate of drug-likeness, SA is the synthetic accessibility score of the molecule m [24] and ring-penalty is an ad-hoc penalty term introduced in order to prevent the model from generating molecules possessing ring sizes larger than 6. The scores corresponding to each term in this objective are normalised to have zero mean and unit standard deviation across the training data. It is important to note that the particulars of the quantities being optimized should be largely unimportant in assessing the efficacy of the Bayesian optimization procedure. As such, the objective function in (**??**) was used in all subsequent experiments as the true black-box objective function. A discussion will follow in section 3.5 about what the best types of objective function for an application of the Automatic Chemical Design model might be.

3.4 Performance of Constrained Bayesian Optimization

This section constitutes the formal test of the hypothesis from [33] that the root cause of the decoder's lack of efficiency in decoding latent points collected by Bayesian optimization arises as a result of the presence of large "dead regions" in the latent space from which it becomes very difficult to decode valid molecules. The section will proceed firstly by describing the implementation details of the Bayesian optimization procedure for the molecular dataset.

Secondly, the results of a set of diagnostic experiments are presented. The experiments are diagnostic in the sense that they are designed to investigate the hypothesis that unconstrained Bayesian optimization is collecting points in a dead region that lies far away from the training data in the latent space. Thirdly, the performance of constrained Bayesian optimization will be compared directly with that of the Bayesian optimization procedure of the original model (baseline model) in terms of the number of valid and drug-like molecules generated. The related aspect of the number of new molecules generated will also be discussed here. Lastly the quality of the molecules produced by constrained Bayesian optimization and the baseline model is compared, Quality in this case is measured by the objective function scores of new molecules proposed by the methods.

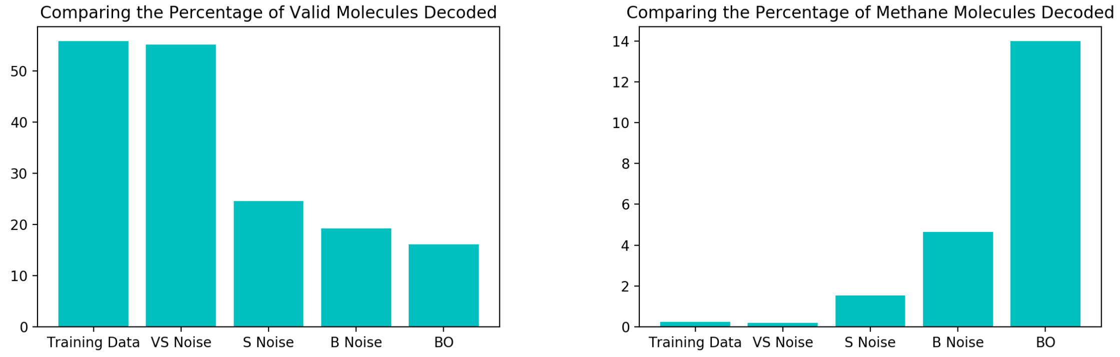
3.4.1 Implementation

The implementation details of the encoder-decoder network remain unchanged from [33]. The objective function is modelled by a sparse GP based on the FITC approximation in the case of both constrained and unconstrained Bayesian optimization. M, the number of inducing points is set to 500, the minibatch size is 5000, the number of training epochs is 50 and the learning rate is 0.005. If unmentioned, the remaining parameters remain the same as in the case of the implementation for the Branin-Hoo function. In the case of the constrained Bayesian optimization algorithm, the BNN is constructed with 2 hidden layers each 100 units wide with ReLU activation functions and a logistic output. Minibatch size is set to 1000 and the network is trained for 5 epochs with a learning rate of 0.0005.

Again, if unmentioned all remaining parameters are unchanged from the implementation for the Branin-Hoo experiments. In both the unconstrained and constrained Bayesian optimization procedures, 20 iterations of parallel Bayesian optimization were performed using the Kriging-Believer algorithm collecting data in batch sizes of size 50. Both the unconstrained and constrained approaches were initialised with $249,456$ training points corresponding to $249,456$ drug-like molecules drawn at random from the ZINC database [49]. The same training set of randomly-drawn molecules as [33] was used. The train/test split of this data was in the ratio of 90/10. In results where the mean and standard error is reported, each run corresponds to a different random train/test split in the ratio of 90/10. For constrained Bayesian optimization, the binary classification BNN was trained using a balanced training set of $234,880$ latent points with binary labels. The labelling procedure will be outlined in the following section.

3.4.2 Diagnostic Experiments and Labelling Criteria

A set of experiments were designed in order to investigate the hypothesis that points collected by Bayesian optimization lie far away from the training data in latent space and hence give rise to a large number of invalid decodings. Five sets of latent points were created. The first set consisted of 50 latent points from the training data. In the VS, or very small noise set, 1% noise was added to 50 randomly chosen training values along each dimension. The percentage of noise in this case is defined relative to the maximum difference between training set values along a given dimension. The noise itself is drawn from a uniform distribution that ranges from $[-\%\text{noise}, +\%\text{noise}]$. In the S, or small noise set 10% noise was added to the training values along each dimension. In the B, or big noise set 50% noise was added to the training values along each dimension. The final set, BO, consisted of 50 points collected by running the Bayesian optimization procedure of the original model. These latent points subsequently underwent 500 decode attempts and the resulting observations are summarised in Figure 3.7.



(**a**) Valid Molecules

(**b**) Methane Molecules

Fig. 3.7 Experiments on 5 disjoint sets comprising 50 latent points each. VS Noise are training data points with approximately 1% noise added to their values, S Noise have 10% noise added to their values and B Noise have 50% noise added to their values. As an example, the percentage noise in the case of 1% noise is defined to be a value drawn from a uniform distribution where the lower end of the range is 0.01 times the maximum difference between the values of the training data along a given dimension. **a)** The percentage of times a given latent point from each set was decoded to a valid molecule, averaged over 50 latent points. **b)** The percentage of times a given latent point from each set was decoded to a methane molecule, averaged over 50 latent points.

There would appear to be a noticeable decrease in the percentage of valid molecules decoded as one moves further away from the training data in latent space (adding more noise to the training values should be equivalent to moving further away from them). Consequently, the fact that the points collected by Bayesian optimization do the worst in terms of their percentage of valid decodings would suggest that these points lie farther away from the training data than than even the B Noise set. A further artefact of the decoder is that it would seem to over-generate methane molecules when far away from the data. One hypothesis for why this is the case is that methane is represented as 'C' in the SMILES syntax and is by far the most common character. Hence far away from the training data, combinations such as 'C' followed by a stop character may have high probability under the distribution over sequences learned by the decoder. Given that methane has far too low a molecular weight (or equivalently its SMILES is too short) to be a suitable drug candidate, a third plot is generated in Figure 3.8, whereupon the percentage of decoded molecules is given such that the molecules are both valid and have a tangible molecular weight. From here on in the definition of a tangible molecular weight will be interpreted somewhat arbitrarily as being a SMILES length of 5 or greater. Further to this, molecules that are both valid and have a SMILES length greater than 5 will be referred to as drug-like.

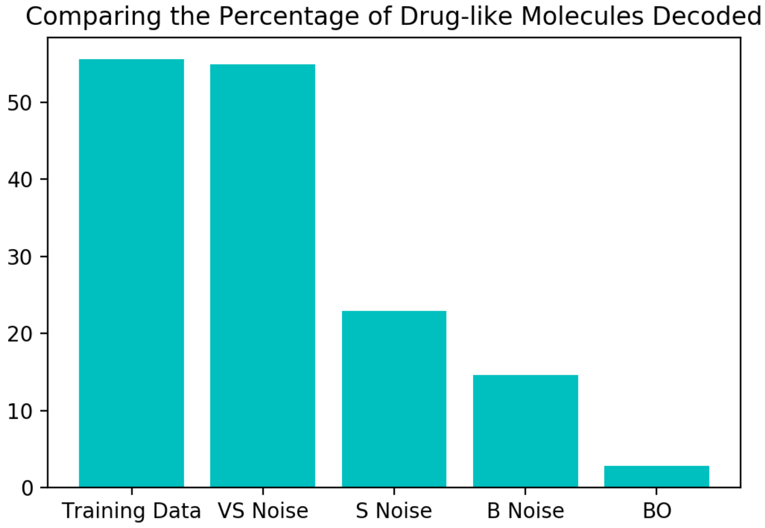


Fig. 3.8 Drug-like Molecules.

From the histogram in Figure 3.8, it may be observed that the points collected by Bayesian optimization produce an even smaller proportion of drug-like molecules than they did valid molecules relative to the training and noise-displaced training data. As a result of these diagnostic experiments, it was decided that the criteria for labelling latent points to initialise the binary classification constraint neural network would be the following: if the latent point decodes into drug-like molecules in more than 20% of decode attempts, it should be classified as a positive class and negative otherwise. The results of the constrained Bayesian optimization procedure trained using the aforementioned labelled data will be presented in the following section.

3.4.3 Molecular Validity

As mentioned in the section on implementation, the constraint function was initialised with 117440 positive class points and 117440 negative class points. For a latent point to be labelled positive at least 20% of its attempted decodings must be valid. The positive labelled data points were obtained by running the training data through the decoder and classifying those points that satisfied the criteria as positive. The negative class points, in contrast, were collected by decoding points sampled uniformly at random across the design space. The relative performance of constrained Bayesian optimization and unconstrained Bayesian optimization (baseline) [33] is compared in Figure 3.9.

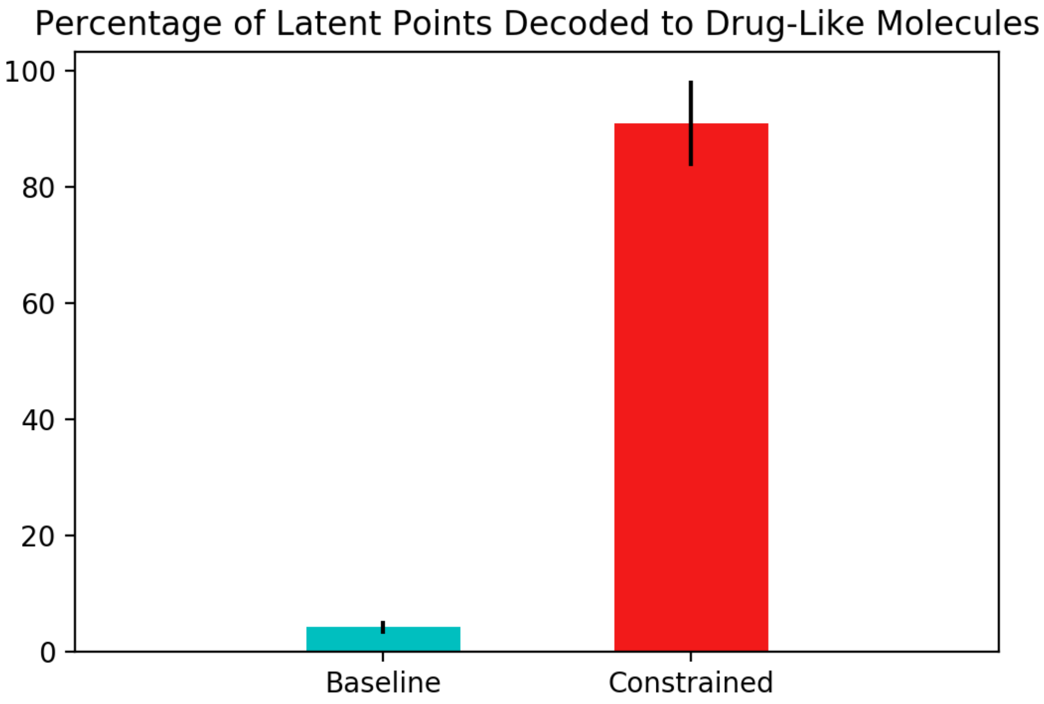


Fig. 3.9 The percentage of latent points decoded to drug-like molecules. 20 iterations of Bayesian optimization with batches of 50 data points collected at each iteration (1000 latent points decoded in total). The standard error is given for 5 separate train/test set splits of 90/10. Drug-like in this instance specifies a requirement that the molecules must both be valid and have a length greater than 5 in the SMILES Representation.

For this experiment, each latent point undergoes 100 decoding attempts and the most probable SMILES string is retained. This SMILES string is then categorised as being drug-like or not drug-like. The results suggest that greater than 80% of the latent points decoded by constrained Bayesian optimization produce drug-like molecules compared to less than 5% for unconstrained Bayesian optimization. However, one must also account for the fact that the constrained approach may be gaining an artificial advantage on this metric by decoding the training data. There would seem to be a fine line between exploring the latent space and wandering into a dead region. If the constrained region is too tight, then the decoding process will tend to produce molecules that have been seen in training, yet when there is no constraint, there is a danger that the optimisation process will decode points in dead regions of latent space. One means of controlling the trade-off between these two extremes would be to treat the labelling criteria as a parameter to be tuned. If the minimum percentage of decodings required to be ’drug-like’ is more lenient, the noose around the training data is loosened but one will be less successful at decoding due to the risk of entering a dead region of the latent space.

Conversely, if the noose is too tight, most of the molecules decoded are liable to be training data. One means of controlling the balance between these two factors is to look at the number of new drug-like molecules generated by the process. Constrained and unconstrained Bayesian optimization are compared on this metric in Figure 3.10.

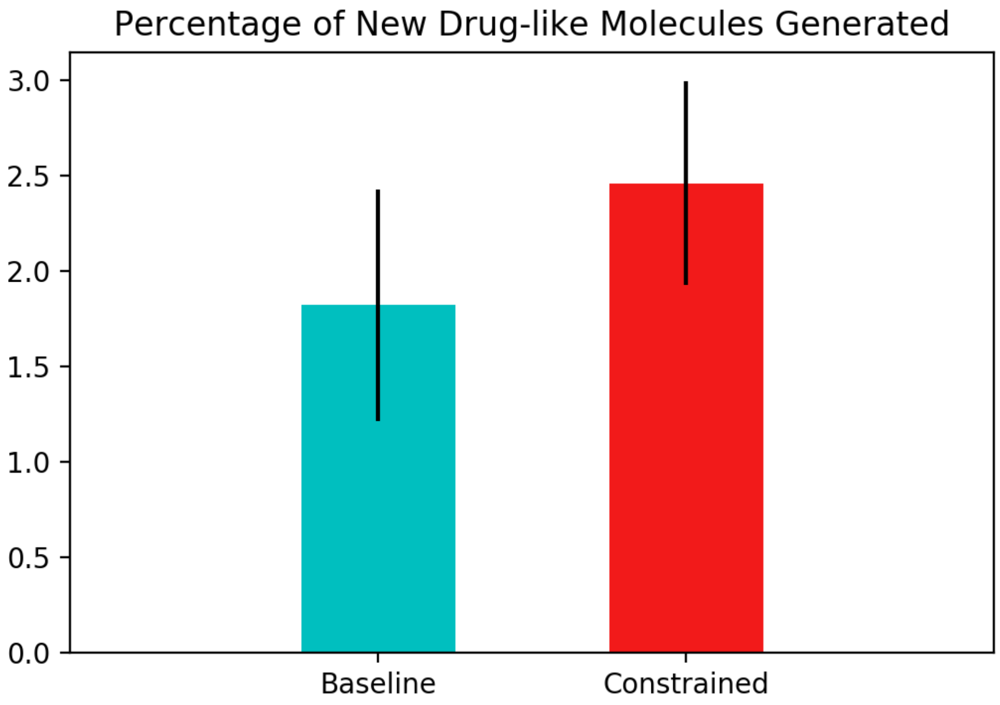


Fig. 3.10 The percentage of new drug-like molecules generated from 20 iterations of Bayesian optimization with batches of 50 data points collected at each iteration (1000 latent points decoded in total). The standard error is given for 5 separate train/test set splits of 90/10.

One may observe that constrained Bayesian optimization outperforms unconstrained Bayesian optimization in terms of generating new molecules, but not by a large margin. The number of new drug-like molecules however is a metric that is subject to some abuse due to the arbitrary definition of what it means to be drug-like. Following a manual inspection of the SMILES strings collected by the unconstrained optimization approach, it would appear as though there were many strings with lengths marginally larger than the cutoff point, which is suggestive of partially decoded molecules. As such, a fairer metric for comparison that is in-keeping with the ethos of Bayesian optimization should be the quality of the new molecules produced as judged by the scores from the black-box objective function. This comparison is made in the following section.

3.4.4 Molecular Quality

The histogram in Figure 3.11 represents the distribution of objective function scores seen in the training data. The blue and red lines are respectively the best molecules produced by unconstrained Bayesian optimization and constrained Bayesian optimization averaged over 5 random train/test splits.

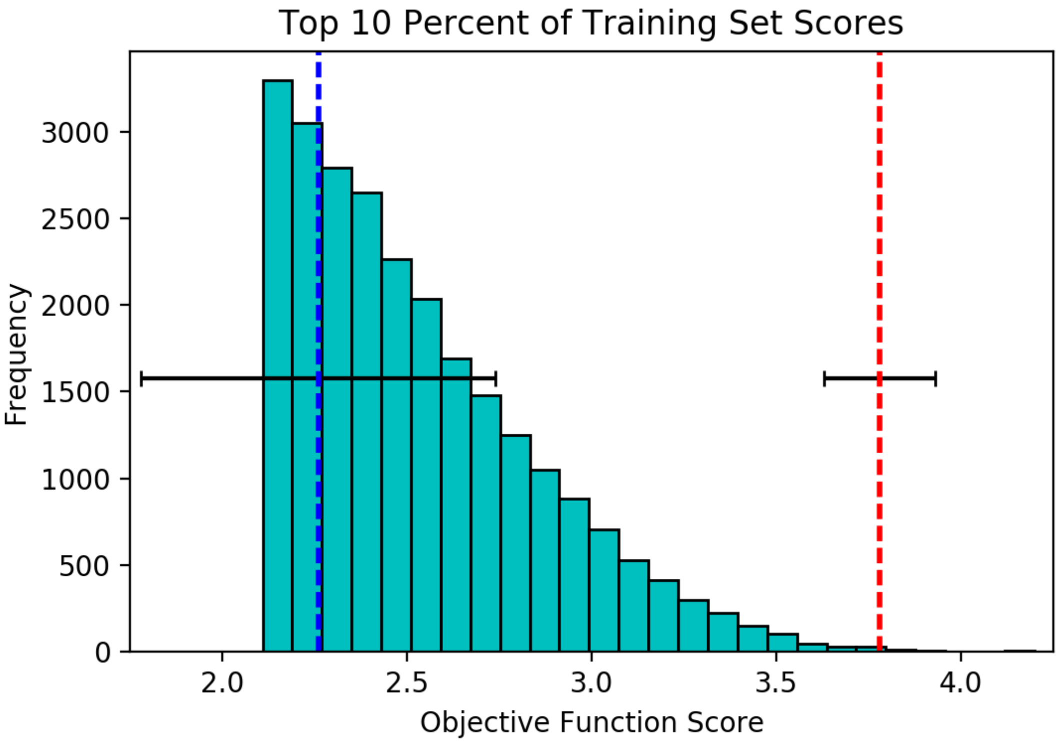


Fig. 3.11 The best scores for new molecules generated from the baseline model (blue) and the model with constrained Bayesian Optimization (red). The vertical lines show the best scores averaged over 5 separate train/test set splits of 90/10. The histogram is presented such that only the top 10% of the training data scores are depicted.

Table 3.1 gives the percentile that the averaged score of the new molecules found by each process occupies in the distribution over training set scores.

Run	Baseline	Constrained
1	49th	86th
2	51st	97th
3	12th	90th
4	37th	93rd
5	29th	94th

Table 3.1 Percentile of the Averaged New Molecule Score Relative to the Training Data. The Results of 5 Separate Train/Test Set Splits of 90/10 are Provided.

Returning briefly to the results of Figure 3.11, constrained Bayesian optimization would appear to be able to generate higher quality molecules relative to unconstrained Bayesian optimization and in slightly higher quantity. In order to ensure that the averaged results aren't vulnerable to outliers, the scores of the best new drug-like molecules are provided in Table 3.2.

Run	Baseline	Constrained
1	88th	100th
2	98th	100th
3	76th	100th
4	96th	100th
5	95th	100th

Table 3.2 Percentile of the Best New Molecule Score Relative to the Training Data. The Results of 5 Separate Train/Test Set Splits of 90/10 are Provided.

Over the 5 runs undertaken, the constrained optimization procedure in every run produced new drug-like molecules in the 100th percentile of the distribution over training set scores. Given a set of drug-like molecules, being able to generate new candidates that rank in the 100th percentile in the distribution over training set scores is an enticing prospect. There is however, one caveat in that it is first necessary to know what score to optimize. In drug discovery, the question of what constitutes a drug-like molecule is a surprisingly difficult question to think about [6]. The following section will address questions related to knowing what to optimize and will propose some areas of application for the current model.

3.5 Objective Design

The objective optimized in [33] is

$$J^{\mathrm{QED}}(m) = \mathrm{QED}(m) - \mathrm{SA}(m) - \text{ring-penalty}(m).. \tag{3.4}$$

Using Equation 3.4, [33] were able to generate new molecules with very high values under the objective relative to the training set and indeed this was a very important demonstration of the theoretical performance of Bayesian optimization within the Automatic Chemical Design model. The spirit of this section however, will be to show some conflicts that might occur between some commonly proposed metrics in the literature such as the synthetic accessibility metric [24] and the quantitative estimate of drug-likeness metric [6] that may occur in practice if the constrained Bayesian optimization model developed were to achieve widespread application. Writing the synthetic accessibility score explicitly as

$$\mathrm{SAscore} = \mathrm{fragmentScore} - \mathrm{complexityPenalty}, \tag{3.5}$$

and by noting that the complexityPenalty is composed of a sum over four terms corresponding to:

$$\mathrm{ringComplexityScore} = \log(\mathrm{nRingBridgeAtoms} + 1) + \log(\mathrm{nSpiroAtoms} + 1), \tag{3.6}$$

$$\mathrm{stereoComplexityScore} = \log(\mathrm{nStereoCenters} + 1), \tag{3.7}$$

$$\mathrm{macrocyclePenalty} = \log(\mathrm{nMacrocycles} + 1), \tag{3.8}$$

$$\mathrm{sizePenalty} = \mathrm{natoms}^{1.005} - \mathrm{natoms} \tag{3.9}$$

one may observe that the synthetic accessibility score incorporates a ring penalty and so there are contributions to ring penalties from two terms in Equation 3.4. Similar conflicts may arise in the joint optimization of the QED score and the SA score, where the SA score contains a size penalty and the QED score attempts to ensure that molecules remain under a certain molecular weight in accord with Lipinski's rule of five [65]. As can be seen above, it is a very difficult task to try to optimize for multiple properties given how often commonly used metrics tend to come into conflict. Additionally, the gold standard for validating metrics such as SA and QED is agreement with human medicinal chemists. As such, one proposal for the model of Automatic Chemical Design would be to make it available in the fashion described by [9]. In this manner human medicinal chemists could be used to help provide a prior over the likely or desirable parameter settings for the objective function that is being sought. One can envisage an iterative process occurring whereby a human chemist accepts or rejects the model's proposal based on chemical intuition, the model updates the parameters of the objective it is trying to optimize and proposes another molecule that is hopefully closer to what the chemist is looking for.

Chapter 4

Conclusion

4.1 Contributions

The reformulation of the search procedure in the Automatic Chemical Design model as a constrained Bayesian optimization problem has led to concrete improvements on two fronts:

1. Validity - The number of valid molecules produced by the constrained optimization procedure offers a marked improvement over the original model. Furthermore, it may be possible to treat the criteria for defining the constraint as a parameter that may be tuned to control the extent of exploration of the latent space.

2. Quality - For five independent train/test splits, the scores of the best molecules generated by the constrained optimization procedure consistently ranked in the 100th percentile of the distribution over training set scores. The ability to find new molecules that are competitive with the very best of the training set is a powerful demonstration of the model's capabilities. As a further point, the generality of the approach should be emphasised. This approach is liable to work for a large range of objectives encoding countless desirable molecular properties.

4.2 Future Work

Future work can entail experiments to investigate if further performance gains may be achieved from tuning the criteria for defining a constraint. Furthermore, it would be interesting to investigate the possibility of developing an interface whereby a human medicinal chemist could interact with the model in order to perform Bayesian optimization in a hierarchical fashion in the design of the objective function as well as in the design of molecules that optimize it.

References

[1] Bailey, K. (2016). Gaussian processes for dummies. http://katbailey.github.io/post/gaussian-processes-for-dummies/.

[2] Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D., and Bengio, Y. (2012). Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*.

[3] Bauer, M., van der Wilk, M., and Rasmussen, C. E. (2016). Understanding probabilistic sparse gaussian process approximations. In *Advances in Neural Information Processing Systems*, pages 1533–1541.

[4] Beck, D., de Gispert, A., Iglesias, G., Waite, A., and Byrne, B. (2016). Speed-constrained tuning for statistical machine translation using bayesian optimization. *arXiv preprint arXiv:1604.05073*.

[5] Bergstra, J. S., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 2546–2554. Curran Associates, Inc.

[6] Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S., and Hopkins, A. L. (2012). Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98.

[7] Bohacek, R. S., McMartin, C., and Guida, W. C. (1996). The art and practice of structure-based drug design: A molecular modeling perspective. *Medicinal research reviews*, 16(1):3–50.

[8] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

[9] Brochu, E., Brochu, T., and de Freitas, N. (2010a). A bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 103–112. Eurographics Association.

[10] Brochu, E., Cora, V. M., and De Freitas, N. (2010b). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

[11] Bui, T. D., Yan, J., and Turner, R. E. (2016). A unifying framework for sparse gaussian process approximation using power expectation propagation. *arXiv preprint arXiv:1605.07066*.

[12] Calandra, R., Seyfarth, A., Peters, J., and Deisenroth, M. P. (2016). Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence*, 76(1-2):5–23.

[13] Chapelle, O. and Li, L. (2011). An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*, pages 2249–2257.

[14] Clark, S., Liu, E., Frazier, P., Wang, J., Oktay, D., and Vesdapunt, N. (2014). Moe: A global, black box optimization engine for real world metric optimization. https://github.com/Yelp/MOE.

[15] Cornejo-Bueno, L., Garrido-Merchán, E. C., Hernández-Lobato, D., and Salcedo-Sanz, S. (2017). Bayesian optimization of a hybrid prediction system for optimal wave energy estimation problems. In *International Work-Conference on Artificial Neural Networks*, pages 648–660. Springer, Cham.

[16] Cox, R. T. (1961). *The algebra of probable inference*. Johns Hopkins Press Baltimore.

[17] Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. (2014). Robots that can adapt like animals. *arXiv preprint arXiv:1407.3501*.

[18] Depeweg, S., Hernández-Lobato, J. M., Doshi-Velez, F., and Udluft, S. (2016). Learning and policy search in stochastic dynamical systems with bayesian neural networks. *arXiv preprint arXiv:1605.07127*.

[19] Dixon, L. C. W. and Szegö, G. P. (1978). *Towards global optimisation*. North-Holland Amsterdam.

[20] Dongarra, J. J., Moler, C. B., Bunch, J. R., and Stewart, G. W. (1979). *LINPACK users' guide*. SIAM.

[21] Duvenaud, D. (2014). *Automatic Model Construction with Gaussian Processes*. PhD thesis, Computational and Biological Learning Laboratory, University of Cambridge.

[22] Eggensperger, K., Feurer, M., Hutter, F., Bergstra, J., Snoek, J., Hoos, H., and Leyton-Brown, K. (2013). Towards an empirical foundation for assessing bayesian optimization of hyperparameters. In *NIPS workshop on Bayesian Optimization in Theory and Practice*.

[23] Eric, B., Freitas, N. D., and Ghosh, A. (2008). Active preference learning with discrete choice data. In *Advances in neural information processing systems*, pages 409–416.

[24] Ertl, P. and Schuffenhauer, A. (2009). Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8.

[25] Gardner, J., Kusner, M., Weinberger, K. Q., Cunningham, J., and Xu, Z. (2014). Bayesian optimization with inequality constraints. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 937–945.

[26] Garnett, R., Osborne, M. A., and Roberts, S. J. (2010). Bayesian optimization for sensor set selection. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 209–219. ACM.

[27] Garrido-Merchán, E. C. and Hernández-Lobato, D. (2017). Dealing with Integer-valued Variables in Bayesian Optimization with Gaussian Processes. *ArXiv e-prints*.

[28] Gelbart, M. A. (2015). *Constrained Bayesian Optimization and Applications*. PhD thesis, Harvard University.

[29] Gelbart, M. A., Snoek, J., and Adams, R. P. (2014). Bayesian optimization with unknown constraints. *arXiv preprint arXiv:1403.5607*.

[30] Ginsbourger, D., Le Riche, R., and Carraro, L. (2010). Kriging is well-suited to parallelize optimization. *Computational Intelligence in Expensive Optimization Problems*, 2:131.

[31] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256.

[32] Gómez-Bombarelli, R., Aguilera-Iparraguirre, J., Hirzel, T. D., Duvenaud, D., Maclaurin, D., Blood-Forsythe, M. A., Chae, H. S., Einzinger, M., Ha, D.-G., Wu, T., et al. (2016a). Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Nature materials*, 15(10):1120–1127.

[33] Gómez-Bombarelli, R., Duvenaud, D., Hernández-Lobato, J. M., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. (2016b). Automatic chemical design using a data-driven continuous representation of molecules. *arXiv preprint arXiv:1610.02415*.

[34] González, J., Dai, Z., Hennig, P., and Lawrence, N. (2016). Batch bayesian optimization via local penalization. In *Artificial Intelligence and Statistics*, pages 648–657.

[35] Gonzalez, J., Osborne, M., and Lawrence, N. D. (2016). GLASSES: Relieving the myopia of Bayesian optimisation. In Gretton, A. and Robert, C., editors, *Proceedings of the Nineteenth International Workshop on Artificial Intelligence and Statistics*, volume 51, page 790, Cadiz, Spain. PMLR.

[36] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[37] Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., and Wierstra, D. (2015). Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*.

[38] Hamze, F., Wang, Z., and de Freitas, N. (2013). Self-avoiding random dynamics on integer complex systems. *ACM Transactions on Modelling and Computer Simulation*, 23(1):9:1–9:25.

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

[40] Hernández-Lobato, J. M. (2017). Bayesian neural networks with random inputs for model based reinforcement learning. https://medium.com/towards-data-science/bayesian-neural-networks-with-random-inputs-for-model-based-reinforcement-learning-36606a9399b4.

[41] Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. (2014). Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems*, pages 918–926.

[42] Hernández-Lobato, J. M., Li, Y., Rowland, M., Hernández-Lobato, D., Bui, T. D., and Turner, R. E. (2016). Black-box α-divergence minimization. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pages 1511–1520. JMLR. org.

[43] Hoffman, M., Brochu, E., and de Freitas, N. (2011). Portfolio allocation for bayesian optimization. In *Uncertainty in Artificial Intelligence (UAI)*. Corvallis ?Oregon.

[44] Hoffman, M., Shahriari, B., and Freitas, N. (2014). On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning. In *Artificial Intelligence and Statistics*, pages 365–374.

[45] Hoffman, M. W. (2013). *Decision making with inference and learning methods*. PhD thesis, University of British Columbia.

[46] Hutter, F., Hoos, H., and Leyton-Brown, K. (2010). Automated configuration of mixed integer programming solvers. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 186–202.

[47] Hutter, F., Hoos, H., and Leyton-Brown, K. (2012). Parallel algorithm configuration. *Learning and Intelligent Optimization*, pages 55–70.

[48] Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. *LION*, 5:507–523.

[49] Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. (2012). Zinc: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7):1757–1768.

[50] Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge university press.

[51] 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.

[52] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

[53] Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[54] Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589.

[55] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[56] Kirkpatrick, P. and Ellis, C. (2004). Chemical space. *Nature*, 432(7019):823.

[57] Klein, A., Falkner, S., Springenberg, J. T., and Hutter, F. (2016). Learning curve prediction with bayesian neural networks. In *International Conference on Learning Representations (ICLR2017)*.

[58] Kohavi, R., Longbotham, R., Sommerfield, D., and Henne, R. M. (2009). Controlled experiments on the web: survey and practical guide. *Data mining and knowledge discovery*, 18(1):140–181.

[59] Kotthoff, L., Thornton, C., Hoos, H. H., Hutter, F., and Leyton-Brown, K. (2016). Auto-weka 2.0: Automatic model selection and hyperparameter optimization in weka. *Journal of Machine Learning Research*, 17:1–5.

[60] Krause, A. and Ong, C. S. (2011). Contextual gaussian process bandit optimization. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 2447–2455. Curran Associates, Inc.

[61] Kushner, H. J. (1964). A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering*, 86(1):97+.

[62] Lázaro-Gredilla, M. and Figueiras-Vidal, A. (2009). Inter-domain gaussian processes for sparse inference using inducing features. In *Advances in Neural Information Processing Systems*, pages 1087–1095.

[63] Lázaro-Gredilla, M., Quiñonero-Candela, J., Rasmussen, C. E., and Figueiras-Vidal, A. R. (2010). Sparse spectrum gaussian process regression. *Journal of Machine Learning Research*, 11:1865–1881.

[64] Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM.

[65] Lipinski, C. A., Lombardo, F., Dominy, B. W., and Feeney, P. J. (1997). Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, 23(1-3):3–25.

[66] Liu, D. C. and Nocedal, J. (1989). On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1):503–528.

[67] Lizotte, D., Wang, T., Bowling, M., and Schuurmans, D. (2007). Automatic gait optimization with gaussian process regression. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 944–949, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[68] Lizotte, D. J. (2008). *Practical Bayesian Optimization*. PhD thesis, University of Alberta, Edmonton, Alta., Canada. AAINR46365.

[69] MacKay, D. J. (1992). Bayesian interpolation. *Neural computation*, 4(3):415–447.

[70] MacKay, D. J. (1999). Comparison of approximate methods for handling hyperparameters. *Neural computation*, 11(5):1035–1068.

[71] MacKay, D. J. C. (1991). *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology.

[72] MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press. Available from `http://www.inference.phy.cam.ac.uk/mackay/itila/`.

[73] Mahendran, N., Wang, Z., Hamze, F., and De Freitas, N. (2012). Adaptive mcmc with bayesian optimization. In *Artificial Intelligence and Statistics*, pages 751–760.

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

[75] Martinez, J., Little, J. J., and de Freitas, N. (2014). Bayesian optimization with an empirical hardness model for approximate nearest neighbour search. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 588–595. IEEE.

[76] Martinez-Cantin, R. (2014). Bayesopt: A bayesian optimization library for nonlinear optimization, experimental design and bandits. *The Journal of Machine Learning Research*, 15(1):3735–3739.

[77] Martinez-Cantin, R., de Freitas, N., Brochu, E., Castellanos, J., and Doucet, A. (2009). A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, 27(2):93–103.

[78] Martinez-Cantin, R., de Freitas, N., Doucet, A., and Castellanos, J. A. (2007). Active policy learning for robot planning and exploration under uncertainty. In *Robotics: Science and Systems*, pages 321–328.

[79] Martinez-Cantin, R., McCourt, M., and Tee, K. (2017). Robust bayesian optimization with student-t likelihood. *arXiv preprint arXiv:1707.05729*.

[80] Matérn, B. (2013). *Spatial variation*, volume 36. Springer Science & Business Media.

[81] Minka, T. P. (2001). Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc.

[82] Mockus J., T. V. and Žilinskas (1978). The application of bayesian methods for seeking the extremum. In Dixon, I. and Szego, G., editors, *Toward Global Optimization*. Elsevier, Amsterdam.

[83] Morar, M., Knowles, J., and Sampaio, S. (2017). *Initialization of Bayesian Optimization Viewed as Part of a Larger Algorithm Portfolio*.

[84] Neal, R. M. (1995). *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, Toronto, Ont., Canada, Canada. AAINN02676.

[85] Neal, R. M. (2003). Slice sampling. *Annals of statistics*, pages 705–741.

[86] Okamoto, Y. (2017). Applying bayesian approach to combinatorial problem in chemistry. *The Journal of Physical Chemistry A*, 121(17):3299–3304.

[87] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

[88] Osborne, M. A., Garnett, R., and Roberts, S. J. (2009). Gaussian processes for global optimization. In *3rd international conference on learning and intelligent optimization (LION3)*, pages 1–15.

[89] Poggio, T. and Girosi, F. (1990). Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497.

[90] Pyzer-Knapp, E. O., Suh, C., Gómez-Bombarelli, R., Aguilera-Iparraguirre, J., and Aspuru-Guzik, A. (2015). What is high-throughput virtual screening? a perspective from organic materials discovery. *Annual Review of Materials Research*, 45:195–216.

[91] Qi, Y. A., Abdel-Gawad, A. H., and Minka, T. P. (2010). Sparse-posterior gaussian processes for general likelihoods. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, pages 450–457, Arlington, Virginia, United States. AUAI Press.

[92] Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959.

[93] Rasmussen, C. E. and Ghahramani, Z. (2001). Occam's razor. In *Advances in neural information processing systems*, pages 294–300.

[94] Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

[95] Reymond, J.-L. and Awale, M. (2012). Exploring chemical space for drug discovery using the chemical universe database. *ACS chemical neuroscience*, 3(9):649–657.

[96] Roustant, O., Ginsbourger, D., and Deville, Y. (2012). Dicekriging, diceoptim: Two r packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software, Articles*, 51(1):1–55.

[97] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical science*, pages 409–423.

[98] Schneider, G. (2010). Virtual screening: an endless staircase? *Nature reviews. Drug discovery*, 9(4):273.

[99] Schneider, G. and Fechner, U. (2005). Computer-based de novo design of drug-like molecules. *Nature reviews. Drug discovery*, 4(8):649.

[100] Schonlau, M., Welch, W. J., and Jones, D. R. (1998). Global versus local search in constrained optimization of computer models. *Lecture Notes-Monograph Series*, pages 11–25.

[101] Scott, S. L. (2010). A modern bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6):639–658.

[102] Seeger, M., Williams, C., and Lawrence, N. (2003). Fast forward selection to speed up sparse gaussian process regression. In *Artificial Intelligence and Statistics 9*, number EPFL-CONF-161318.

[103] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. (2016). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

[104] Shahriari, B., Wang, Z., Hoffman, M. W., Bouchard-Côté, A., and de Freitas, N. (2014). An entropy search portfolio for bayesian optimization. *arXiv preprint arXiv:1406.4625*.

[105] Shapiro, A., Dentcheva, D., and Ruszczynski, A. (2014). *Lectures on Stochastic Programming: Modeling and Theory, Second Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

[106] Shun-ichi, A. (2012). *Differential-geometrical methods in statistics*, volume 28. Springer Science & Business Media.

[107] Sliwoski, G., Kothiwale, S., Meiler, J., and Lowe, E. W. (2014). Computational methods in drug discovery. *Pharmacological reviews*, 66(1):334–395.

[108] Smola, A. J. and Bartlett, P. L. (2001). Sparse greedy gaussian process regression. In *Advances in neural information processing systems*, pages 619–625.

[109] Snelson, E. and Ghahramani, Z. (2006). Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264.

[110] Snelson, E. L. (2008). *Flexible and efficient Gaussian process models for machine learning*. PhD thesis, University College London.

[111] Snoek, J. (2013). *Bayesian optimization and semiparametric models with applications to assistive technology*. PhD thesis, University of Toronto.

[112] Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959.

[113] Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., and Adams, R. (2015). Scalable bayesian optimization using deep neural networks. In *International Conference on Machine Learning*, pages 2171–2180.

[114] Springenberg, J. T., Klein, A., Falkner, S., and Hutter, F. (2016). Bayesian optimization with robust bayesian neural networks. In *Advances in Neural Information Processing Systems*, pages 4134–4142.

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

[116] Sutskever, I. (2013). *Training Recurrent Neural Networks*. PhD thesis, University of Toronto, Toronto, Ont., Canada, Canada. AAINS22066.

[117] Swersky, K., Snoek, J., and Adams, R. P. (2013). Multi-task bayesian optimization. In *Advances in neural information processing systems*, pages 2004–2012.

[118] Team, T. T. D., Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., et al. (2016). Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*.

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

[120] Thornton, C., Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2013). Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 847–855, New York, NY, USA. ACM.

[121] Titsias, M. K. (2009). Variational learning of inducing variables in sparse gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 567–574.

[122] Turner, R. E. (2010). *Statistical models for natural sounds*. PhD thesis, UCL (University College London).

[123] Vanchinathan, H. P., Nikolic, I., De Bona, F., and Krause, A. (2014). Explore-exploit in top-n recommender systems via gaussian processes. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 225–232. ACM.

[124] Villemonteix, J., Vazquez, E., and Walter, E. (2009). An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534.

[125] Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and de Feitas, N. (2016). Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387.

[126] Wang, Z., Shakibi, B., Jin, L., and Freitas, N. (2014). Bayesian multi-scale optimistic optimization. In *Artificial Intelligence and Statistics*, pages 1005–1014.

[127] Wang, Z., Zoghi, M., Hutter, F., Matheson, D., and De Freitas, N. (2013). Bayesian optimization in high dimensions via random embeddings. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 1778–1784. AAAI Press.

[128] Weininger, D. (1988). Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36.

[129] Yogatama, D. and Smith, N. A. (2015). Bayesian optimization of text representations. *arXiv preprint arXiv:1503.00693*.

[130] Zuluaga, M., Sergent, G., Krause, A., and Püschel, M. (2013). Active learning for multi-objective optimization. In *International Conference on Machine Learning*, pages 462–470.