Addressing Practical Challenges of Bayesian Optimisation

by

Thanh Dai Nguyen
BSc. (Honours)

Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

Deakin University

October 2018
I am the author of the thesis entitled

**Addressing Practical Challenges of Bayesian Optimisation**

submitted for the degree of **Doctor of Philosophy**

This thesis may be made available for consultation, loan and limited copying in accordance with the Copyright Act 1968.

'I certify that I am the student named below and that the information provided in the form is correct'

**Full Name:** THANH DAI NGUYEN  
(Please Print)

Signed: 

Date: 19/10/2019
I certify the following about the thesis entitled (10 word maximum)

Addressing Practical Challenges of Bayesian Optimisation

submitted for the degree of Doctor of Philosophy

a. I am the creator of all or part of the whole work(s) (including content and layout) and that where reference is made to the work of others, due acknowledgment is given.

b. The work(s) are not in any way a violation or infringement of any copyright, trademark, patent, or other rights whatsoever of any person.

c. That if the work(s) have been commissioned, sponsored or supported by any organisation, I have fulfilled all of the obligations required by such contract or agreement.

d. That any material in the thesis which has been accepted for a degree or diploma by any university or institution is identified in the text.

e. All research integrity requirements have been complied with.

'I certify that I am the student named below and that the information provided in the form is correct'

Full Name: THANH DAI NGUYEN

Signed: Signature Redacted by Library

Date: 19/10/2018
Acknowledgements

With the deepest appreciation, I would like to thank my executive supervisor A/Prof. Sunil Gupta for his guidance, encouragement and support throughout my research. With his deep insight, wisdom and patience, he has taught me invaluable knowledge and research skills. This thesis would have never taken shape without his thorough supervision.

I wish to thank my co-supervisor A/Prof. Santu Rana for providing constructive feedback and making suggestions on my papers and this thesis. I would never forget our valuable discussions.

It is my pleasure to acknowledge my associate supervisor Prof. Svetha Venkatesh for giving me the chance to do research at Deakin university. She has gave me valuable guidance, suggestions and advice in both research and life.

I would like to thank all staff and students of PRaDA for providing me a friendly working environments. I will always remember our full-of-laughter lunchtime discussions.

Finally, this thesis is dedicated to my beloved family: my parents, my sisters, my brother and my wife who always support, encourage, and stand by my side when I needed them most.
Relevant publications

Part of this thesis has been published or documented elsewhere. The details of these publications are as follows:

Chapter 3:


Chapter 4:


Chapter 5:

# Contents

Acknowledgements iv

Relevant publications v

Abstract xix

Abbreviations xx

Notation xx

1 Introduction 1

1.1 Aims and approaches 3

1.2 Significance and contributions 4

1.3 Structure of the thesis 5

2 Related background 8

2.1 Experimental design 8

2.2 Global optimisation 9
2.3 Bayesian optimisation ............................................. 11
  2.3.1 Overview of Bayesian optimisation ......................... 12
  2.3.2 Applications of Bayesian optimisation ...................... 13
2.4 Surrogate models .................................................. 18
  2.4.1 Gaussian processes .......................................... 18
  2.4.2 Random forests .............................................. 25
  2.4.3 Deep neural networks ....................................... 26
2.5 Acquisition functions ............................................ 27
  2.5.1 Improvement-based policies ................................. 27
  2.5.2 Optimistic policies ......................................... 31
  2.5.3 Information-based policies ................................. 31
2.6 Active research areas ............................................ 33
  2.6.1 High-dimensional problems ................................. 33
  2.6.2 Constrained Bayesian optimisation ......................... 34
  2.6.3 Multi-objective Bayesian optimisation ..................... 36
  2.6.4 Transfer learning ........................................... 37
2.7 Open problems relevant to practical applications ............ 38
  2.7.1 Stability .................................................. 38
  2.7.2 Cascaded process optimisation ............................. 39
  2.7.3 Privacy .................................................. 39
2.8 Privacy preserving machine learning and data mining . . . . . . . . . 40

2.8.1 Data perturbation methods . . . . . . . . . . . . . . . . . . . . 40

2.8.2 Group anonymisation methods . . . . . . . . . . . . . . . . . . 42

2.8.3 Methods for distributed data . . . . . . . . . . . . . . . . . . . 42

2.8.4 Differential privacy . . . . . . . . . . . . . . . . . . . . . . . . 43

2.9 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

3 Stable Bayesian optimisation 47

3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

3.2 The proposed framework . . . . . . . . . . . . . . . . . . . . . . . . 50

3.2.1 Stability of Gaussian process prediction . . . . . . . . . . . . . 50

3.2.2 Stable Bayesian optimisation . . . . . . . . . . . . . . . . . . . 53

3.3 Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59

3.3.1 Baseline method and evaluation measures . . . . . . . . . . . . 60

3.3.2 Experiments with synthetic function . . . . . . . . . . . . . . . 60

3.3.3 Experiments with hyperparameter tuning problems . . . . . . 62

3.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66

4 Bayesian optimisation for cascaded processes 67

4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68

4.2 The proposed solution . . . . . . . . . . . . . . . . . . . . . . . . . 71
4.3 Experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 78
  4.3.1 Baseline method and evaluation measure . . . . . . . . . . . . 79
  4.3.2 Experiments with synthetic data . . . . . . . . . . . . . . . . 79
    4.3.2.1 Data generation . . . . . . . . . . . . . . . . . . . 79
    4.3.2.2 Experimental results . . . . . . . . . . . . . . . . . 80
  4.3.3 Experiments with tuning hyperparameters for data analytic
      pipelines . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 82
    4.3.3.1 Tuning hyperparameters for data analytic pipelines . 82
    4.3.3.2 Experimental results . . . . . . . . . . . . . . . . . . 82
  4.3.4 Experiments with alloy heat treatment optimisation . . . . . . 84
    4.3.4.1 Alloy heat treatment . . . . . . . . . . . . . . . . . 84
    4.3.4.2 Experimental results . . . . . . . . . . . . . . . . . 85
  4.3.5 Cost-efficient optimisation . . . . . . . . . . . . . . . . . . . . 86
  4.4 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86

5 A new privacy framework for Bayesian optimisation and other al-
  gorithms 88
  5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89
  5.2 The new privacy framework . . . . . . . . . . . . . . . . . . . . . . 92
  5.3 Privacy preserving Bayesian optimisation . . . . . . . . . . . . . . . 93
    5.3.1 The proposed algorithm . . . . . . . . . . . . . . . . . . . 93
List of Figures

2.1 Visualisation of the surrogate model and the acquisition function of Bayesian optimisation. In this illustration, a Gaussian process and its upper confidence bound (UCB) are used for the surrogate model and the acquisition function. The graph in red colour shows the true objective function. The dotted line represents the mean of Gaussian process posterior. The blue area indicates 95% confidence interval. The yellow triangle represents maximum value of the acquisition function, suggesting the next point for evaluation. .................. 14

2.1 Visualisation of the surrogate model and the acquisition function of Bayesian optimisation. In this illustration, a Gaussian process and its upper confidence bound (UCB) are used for the surrogate model and the acquisition function. The graph in red colour shows the true objective function. The dotted line represents the mean of Gaussian process posterior. The blue area indicates 95% confidence interval. The yellow triangle represents maximum value of the acquisition function, suggesting the next point for evaluation. .................. 15

2.2 An example of Gaussian process for a one dimensional function. Different shades of blue represent the predictive density at each input location. The colour lines illustrate some samples drawn from Gaussian posterior distribution. Fig. 2.2a shows a Gaussian process prior and its samples. The remaining plots show Gaussian process posteriors after observing few samples from a function. .................. 19
2.3 Example of various kernel profiles sampled from a Gaussian process posterior. Note that when $\nu$ increases, the function samples become smoother.

2.4 Examples of acquisition functions. The top plot is the Gaussian model with 3 observations. The remaining plots are the acquisition functions for the Gaussian process: PI - Probability Improvement (see Eq. (2.17)), EI - Expected Improvement (see Eq. (2.19)), UCB - Upper Confidence Bound (see Eq. (2.21)). The yellow triangle makers are the maximum of the acquisition functions.

3.1 Performance versus hyperparameters for a Support Vector Machine training as colour coded images: a) on a small validation set, and b) on a large test set. Spurious peaks of region 1 seen for the validation set vanish for the test set while the stable peak of region 2 still remains.

3.2 Predictive distributions of Gaussian approximation and Monte-Carlo approximation with respect to different values of noise levels: a) $\Sigma_x = 0.01$, b) $\Sigma_x = 0.02$, c) $\Sigma_x = 0.05$. In practice, parameter settings are limited to 1% perturbation and our approximation based on Gaussian distribution can handle it well.

3.3 a) The synthetic function with one stable peak and multiple spurious peaks b) The STABLE-UCB acquisition function and aleatoric variance after 30 iterations.

3.4 Performance of stable Bayesian optimisation and standard Bayesian optimisation with respect to number of iterations on Synthetic function. a) and c) shows that STABLE-UCB and STABLE-EI converge to 2.3 and 2.5 (stable peak) whereas BO-UCB and BO-EI converge to 3.7 (spurious peak). b) and d) shows that stable Bayesian optimisation reaches stable peak more often than the baseline.
3.5 Sampling behaviour of both STABLE-UCB and BO-UCB for hyperparameter tuning of SVM for Letter classification a) on validation dataset and b) on test dataset. The background portrays the performance function with respect to the hyperparameters. Spurious peaks (region 1) is evident for the validation dataset but vanished for the test set while stable region (region 2) still remains.

3.6 Performance of stable Bayesian optimisation and standard Bayesian optimisation using SVM with respect to number of iterations on Letter dataset. The performance of standard Bayesian optimisation, due to convergence to spurious peaks on the validation set (Fig 3.6a and Fig 3.6c), results in poor performance for the test set (Fig 3.6b and Fig 3.6d).

3.7 Performance of stable Bayesian optimisation and standard Bayesian optimisation using SVM with respect to number of iterations on Glass dataset. The performance of standard Bayesian optimisation, due to convergence to spurious peaks on the validation set (Fig 3.7a and Fig 3.7c), results in poor performance for the test set (Fig 3.7b and Fig 3.7d).

4.1 A diagram of a multi-stage cascaded process. Input (e.g. raw material) is transformed under the conditions set by control parameters. The output of each stage then becomes the input for the next stage. The output of the last stage is often the quantity that we are interested in optimising as a function of the controls.

4.2 Notations of input, output and control parameters of a cascaded process. For each stage $s$, $u_s^t$, $v_s^t$ and $w_s^t$ are the input, control parameters and output at iteration $t$; and $GP_s^t$ is the Gaussian process model built at iteration $t$. 
4.3 Experimental results of Synthetic-I (left column) and Synthetic-II (right column): Best found value so far as a function of iteration. In the left column, number of control parameters per stage is fixed at 4 but the number of stages are varied. In the right column, number of stages is fixed at 5, but the number of control parameters per stage is varied. 81

4.4 Performance of cascade Bayesian optimisation and joint Bayesian optimisation using SVM with respect to number of iterations on tuning hyperparameters of a data analytic pipeline (a) Letter dataset (b) Glass dataset. 83

4.5 Best hardness achieved as a function of number of iterations for both cascade BO (red) and joint Bayesian optimisation (blue), for different cost parameter $\lambda$. 85

4.6 Experimental result for Aluminium Scandium hardening: Hardness of alloy (red) and time (blue) vs cost parameter $\lambda$. 86

5.1 Privacy preserving Bayesian optimisation framework. The optimisation process is a three-step procedure. The first step is to evaluate the objective function at the input suggested by optimiser. The second step is to perturb the output by a noise to ensure the privacy. The third step is at the untrusted end where optimiser is used to suggest the next point for evaluation. 91

5.2 Using Error Preserving Privacy framework for machine learning algorithms. 94

5.3 The ratio $\frac{q_{n+1}}{\epsilon_n}$ versus $q_{n+1}$. We set $q_{n+1} = \alpha$ so that $\frac{q_{n+1}}{\epsilon_n} \geq \exp(-\epsilon)$. 98

5.4 Best value so far with respect to Bayesian optimisation iteration. The reported values are noise-free values of the function a) Branin 2D, b) Rosenbrock 4D, c) Hartmann 4D, d) Hartmann 6D. 101
5.5 Comparison between the best values achieved by the experimenter and the optimiser with respect to iterations on Hartman 4D function with different level of privacy: a) $\epsilon = 0.1$ (high privacy) and b) $\epsilon = 0.5$ (low privacy). The blue and the magenta show the best function value achieved so far from the experimenter’s perspective and optimiser’s perspective, respectively.

5.6 Optimisation results for two real datasets.

5.7 Results using Synthetic dataset with $N = 180, K = 3$. (a) Average perturbation in cluster centroid with respect to $\epsilon$, (b) NMI with respect to $\epsilon$, (c) Ratio of variance for estimation errors $E_{\text{inc}}$ and $E_{\text{exc}}$, (d) NMI for varying number of data points at $\epsilon = 0.1$.

5.8 Results using Seeds dataset with $n = 210, K = 3$, (a) Average perturbation in cluster centroid with respect to $\epsilon$, (b) NMI with respect to $\epsilon$, (c) Ratio of variance for estimation errors $E_{\text{inc}}$ and $E_{\text{exc}}$, (d) NMI for varying number of data points at $\epsilon = 0.1$.

5.9 Results using UKM dataset with $n = 258, K = 4$, (a) Average perturbation in cluster centroid with respect to $\epsilon$, (b) NMI with respect to $\epsilon$, (c) Ratio of variance for estimation errors $E_{\text{inc}}$ and $E_{\text{exc}}$, (d) NMI for varying number of data points at $\epsilon = 0.1$. 

......

102

104

112

114

115
List of Tables

5.1 Comparison with the baselines in terms of various metrics at $\epsilon = 0.1$. Average results over 30 random centroid initialisations are reported with the standard errors in parenthesis. The bold face indicates the best results among private algorithms. .......................... 116
List of Algorithms

2.1 Bayesian optimisation algorithm

3.1 The proposed stable Bayesian optimisation.

4.1 Cascade Bayesian optimisation algorithm

5.1 Error Preserving Private Bayesian Optimisation (EPP-BO)

5.2 Error Preserving Private K-means algorithm
Abstract

Bayesian optimisation is a powerful machine learning method that allows efficient optimisation of expensive black-box functions. Recently, Bayesian optimisation has been applied successfully in several applications like robotic, material design and automated machine learning. However, despite its recent advancement, applying Bayesian optimisation in practical settings still faces many challenges. This thesis focuses on addressing these practical challenges. The contributions of this thesis are new Bayesian optimisation algorithms for three practical problems: i) finding stable optimum of functions that may have several narrow peaks/valleys, ii) optimising multi-stage cascaded structure processes, and iii) developing a privacy-aware algorithm when it is not possible to reveal the optimum for untrusted optimisers.

In many real world systems, the optimisation surface contains multiple peaks/valleys with different widths. For such applications, if the optimisation has converged to a narrow peak/valley, the utility of the solution will be seriously limited due to the imprecise nature of the systems. We address this problem through a novel stable Bayesian optimisation framework. We construct two new acquisition functions that help Bayesian optimisation to avoid the convergence to narrow (sharp) peaks. We theoretically analyse our algorithm and guarantee that Bayesian optimisation using the proposed acquisition function favours stable (wide) peaks over the unstable (narrow) ones. We demonstrate the effectiveness of our proposed algorithm on both synthetic function optimisation and hyperparameter tuning for support vector machines.

Multi-stage cascaded structure processes are fairly common in manufacturing industry. In a cascaded process, the input of each stage is transformed under condi-
tions set by control parameters. The output then becomes the input for the next stage. To achieve high output quality at low cost, one needs to set the control parameters correctly. We propose a novel cascade Bayesian optimisation algorithm that effectively optimises cascaded structure processes. At the last stage, we use the standard Bayesian optimisation to find the input and control parameters that maximise its output quality. For the remaining stages, we formulate a novel optimisation problem. In a back-propagation manner, we solve this problem to find the input and control parameters of the remaining stages through the inversion of the Gaussian processes. We incorporate a cost-sensitive component to the formulation to discover cost-efficient solutions. We theoretically analyse and show the convergence rate our proposed algorithm. Experiments with a heat treatment testbed of an Al-Sc alloy and hyperparameter tuning of data analytic pipelines show the efficacy of our algorithm.

In industry, experimenters, who are domain experts of their fields, may not have mathematical optimisation expertise. To find the optimal design, they may require third-party optimisation services. Since the optimal design is the key to a business’s success, it often can not be disclosed to the third-party optimisers. To address the privacy concerns, we firstly propose a novel privacy preserving framework called Error Preserving Privacy (EPP) that provides strong privacy guarantee while ensuring high utility. Using EPP framework, we propose a privacy preserving Bayesian optimisation that helps the experimenters to find the optimal design without revealing it to the untrusted optimisers. We theoretically analyse the algorithm and derive the bound on the amount of noise required to guarantee the privacy. We demonstrate the applicability of our EPP framework by constructing a novel privacy preserving K-means clustering algorithm. Using both synthetic and real datasets, we show that the efficiency of our proposed Bayesian optimisation and K-means clustering algorithms is comparable to non-private algorithms and significantly better than the baseline.
Abbreviations

BO     Bayesian Optimisation
DNGO  Deep Networks for Global Optimisation
DP     Differential Privacy
EI     Expected Improvement
EPP    Error Preserving Privacy
GP     Gaussian Process
GP-UCB Gaussian Process - Upper Confidence Bound
PESC   Predictive Entropy Search with Constraints
PI     Probability of Improvement
PINQ   Privacy Integrated Queries
RBF    Radial Basis Function
REMBO  Random Embedding Bayesian Optimisation
SMAC   Sequential Model-based Algorithm Configuration
SMBO   Sequential Model-based Bayesian Optimisation
SVM    Support Vector Machine
UCB    Upper Confidence Bound
WEI    Weighted Expected Improvement
Notation

$\mathbb{R}$ the set of real numbers

$x$ vector $\mathbf{x}$

$x$ scalar $x$

$\mathcal{D}$ dataset

dia$g(w)$ a diagonal matrix containing the elements of vector $\mathbf{w}$

$\mathbb{E}$ expectation

$\text{Var}$ variance

$\mathcal{GP}$ Gaussian process

$K$ covariance matrix

$k(\mathbf{x}, \mathbf{x}')$ covariance function evaluated at $\mathbf{x}$ and $\mathbf{x}'$

$\mathcal{N}$ normal distribution

$\Phi(.)$ the standard normal cumulative distribution function

$\phi(.)$ the standard normal probability density function

$\alpha(\mathbf{x}; \mathcal{D}_t)$ acquisition function at $\mathbf{x}$ given a dataset $\mathcal{D}_t$

$\mathcal{X}$ search space

$\mathcal{O}(.)$ big O notation

$p(y|x)$ probability density of conditional random variable $y$ given $x$

$\mathbb{P}$ probability

$k^T$ the transpose of vector $\mathbf{k}$

$\sim$ distributed according to; example $x \sim \mathcal{N}(0, 1)$

$\mu(.)$ Gaussian process posterior mean

$\sigma^2(.)$ Gaussian process posterior variance

$H(\mathcal{X})$ entropy of random variable $\mathcal{X}$

Lap Laplace distribution

$\mathbf{x}_{1:t}$ set of $t$ vectors from $\mathbf{x}_1$ to $\mathbf{x}_t$

$y_{1:t}$ set of $t$ scalars from $y_1$ to $y_t$

$K^{-1}$ the inverse of matrix $K$
Chapter 1

Introduction

Experimental designs are crucial in research and development: scientists design experiments to discover and understand physical or social phenomena, widening the horizon of human knowledge; material researchers design new type of alloys that are strong and light; meteorologists design sensor networks to monitor the environment; pharmaceutical researchers design new type of drugs to fight diseases. In all of these applications, the objective is to find the optimal design to achieve the “best” possible outcome. The relation between the output and design variables can often be represented through a mathematical function. However, this function may not be expressed in a closed form and its evaluation is generally expensive. This make the task of finding the optimal design a nontrivial problem.

Bayesian optimisation has emerged as a powerful method for finding such optimal designs. It has seen successful applications in robotics (Lizotte et al., 2007; Calandra et al., 2014b; Martinez-Cantin et al., 2009), environmental monitoring (Srinivas et al., 2010), automated machine learning (Thornton et al., 2013; Hoffman et al., 2014), material design (Li et al., 2017), sensor networks (Garnett et al., 2010; Srinivas et al., 2010), reinforcement learning (Brochu et al., 2010b), combinatorial optimisation (Hutter et al., 2011; Wang et al., 2013) and interactive user-interfaces (Brochu et al., 2010a). Due to popularity of Bayesian optimisation in so many applications, there have been efforts to extend its reach to more complex but useful problems such as optimisation in high dimensional spaces (Chen et al., 2012; Wang et al., 2013), and adding the ability to transfer source (function) knowledge to target functions.
(Joy et al., 2016b; Shilton et al., 2017), multi-objective optimisation (Feliot et al., 2017), parallel optimisation (Azimi et al., 2010; Nguyen et al., 2016), etc.

In spite of recent progress, Bayesian optimisation methods still face several challenges in practice. In many real-world systems, the objective function contains multiple peaks with different widths. For such applications, the utility of the optimal design (solution) can be seriously limited if the optimisation has converged to a narrow peak. This is because of the imprecise nature of such systems. For example, in an alloy design process, a goal may be to find a mixing proportion of a set of elements that yields the highest value of strength. However, it is practically nearly impossible to mix the elements in exact proportion due to the impurities in raw materials. Thus, if the optimal proportion is at a narrow peak of the surface then the strength of the alloy may dramatically reduce depending on values of impurities. Therefore, there is a need for a stable Bayesian optimisation framework that is able to find stable solutions.

Another important situation is where design optimisation is highly desired in industrial processes. Industrial processes often have multi-stage cascaded structure. A cascaded structure process consists of several stages, each stage’s input is the output from the previous stage. The standard Bayesian optimisation approach considers the whole cascaded process as one single black-box function and does not take into account the cascaded structure. Thus the problem becomes optimising a high-dimensional target function, which is a difficult task. Therefore, there is a need for a Bayesian optimisation algorithm that can utilise the cascaded structure in the optimisation.

Yet another important challenge in using Bayesian optimisation for real-world applications is privacy. In industry, experimenters, who are domain experts of their specific fields, may not have the mathematical optimisation expertise and thus require third-party optimisation services to find the optimal design. Since the optimal design is the key to a business’s success, it often can not be disclosed to the third-party optimisers. One approach to address this problem is to use privacy preserving approaches for Bayesian optimisation. It is important to have a Bayesian optimisation algorithm that helps the experimenters to find the optimum of the objective function without disclosing the exact details of the optimal design while ensuring a minimum compromise on the optimality or the quality of the design.
1.1 Aims and approaches

The goal of this thesis is to expand the knowledge base by addressing several challenges in applying Bayesian optimisation for real-world applications. In particular, we aim to develop Bayesian optimisation algorithms that are able to achieve the following objectives:

1. Finding stable solutions (wide peaks) for optimisation problems instead of unstable solutions (narrow peaks) where possible.
2. Optimising processes that have multi-stage cascaded structure.
3. Developing a privacy-aware Bayesian optimisation technique when it is not possible to reveal the exact value of the optimal solution to third-party untrusted optimisers.

To achieve these aims, we leverage recent advances in machine learning, especially in Bayesian optimisation and Gaussian processes to implement the following approaches:

- To realise Aim 1, *i.e.* to find stable solutions in optimisation, we use a modified Gaussian process model that takes into account any perturbations in the input variables. The extra variance due to the perturbations is included in the formulation of two novel acquisition functions for Bayesian optimisation. We theoretically analyse the two new acquisition functions and prove that they are guaranteed to have higher values for “more stable” peaks. Thus the Bayesian optimisation algorithm using these acquisition functions have higher tendency to sample in the area around the stable peaks.

- To realise Aim 2, *i.e.* to take advantage of multi-stage cascaded structures, each stage of a cascaded process is modelled by an independent function through separate Gaussian processes. Using the Gaussian process of the final stage, we use the standard Bayesian optimisation to find the input and control parameters of the final stage that maximise its output quality. For the remaining stages, in a back-propagation manner, we formulate a novel optimisation
problem. We solve this problem to find the input and control parameters of the remaining stages through the inversion of the Gaussian process. We introduce costs associated with the control parameters in the optimisation formulation to discover cost-efficient solutions.

- To realise Aim 3, i.e. to address the privacy concerns in using Bayesian optimisation, we firstly propose a novel privacy preserving framework called Error Preserving Privacy (EPP) that provides strong privacy guarantee while ensuring high utility. Using EPP framework, we propose a privacy preserving Bayesian optimisation that helps an experimenter in an industry to find the optimum of an expensive black-box function without revealing the exact function values. The EPP framework helps to maintain high optimisation efficiency even under stringent privacy requirements. Under certain assumptions on the adversary’s model, we theoretically analyse the algorithm and derive the bound on the amount of noise required to guarantee the privacy. To demonstrate the applicability of EPP framework, we construct a novel privacy preserving K-means clustering algorithm.

### 1.2 Significance and contributions

This thesis is significant because: (i) it develops novel algorithms to address several practical challenges of Bayesian optimisation and (ii) it applies these algorithms to a wide range of practical problems in machine learning and industrial processes. In particular, our main contributions are:

- We provide a definition for stability of a peak. Based on our definition of stability, it is possible to measure the stability of a peak using a modified Gaussian process model with input perturbation. We propose two novel acquisition functions for Bayesian optimisation that actively seek the stable peaks of the objective function. We theoretically prove that under mild assumptions, when two peaks are of the same height, the proposed acquisition functions would always favour the more stable peak. We validate the effectiveness of our proposed method using both synthetic and real datasets.
• A cascade Bayesian optimisation framework that effectively optimises multi-stage cascaded processes. We formulate a novel optimisation problem to find the desired input quality and control parameters at intermediate stages of a cascaded process. We theoretically analyse the proposed solution and show that the convergence rate of our algorithm is sub-linear in the number of evaluations. We demonstrate the efficacy of our algorithm on synthetic data, tuning data analytic pipelines and alloy heat treatment optimisation.

• A new privacy preserving framework that provides strong privacy guarantees while ensuring high utility. We propose a novel privacy preserving Bayesian optimisation algorithm under the proposed privacy framework that helps experimenters in an industry to utilise the expertise of a third-party optimisation service without revealing the exact function values. We theoretically analyse the proposed algorithm and derive bounds on the amount of noise required to ensure the privacy. We validate the effectiveness of our method on both synthetic and real dataset. In addition, we demonstrate the applicability of our proposed privacy preserving framework by developing a novel privacy preserving K-means algorithm with high utility. We illustrate and validate the efficacy of the proposed K-means algorithm through experiments with both synthetic and real datasets.

1.3 Structure of the thesis

The remainder of this thesis is organised as follows:

• Chapter 2 provides the preliminary background and research works relevant to the problems considered in this thesis. We first briefly discuss about experimental design and the importance of optimal design problem. We then review several popular global optimisation methods. The chapter then focuses on Bayesian optimisation method and its applications, followed by a discussion about various choices for surrogate models and acquisition functions. In the subsequent part of chapter 2, we review active research areas of Bayesian optimisation, and then discuss about several open problems that are relevant to
1.3. Structure of the thesis

practical applications. The chapter ends by reviewing the literature in privacy preserving machine learning methods.

• Chapter 3 begins with the presentation of our first contribution of developing a stable Bayesian optimisation framework. We define the notion of stability and introduce a modified Gaussian process model for noisy inputs. The variance in the modified Gaussian process posterior distribution consists of epistemic and aleatoric components, among which the aleatoric variance is the one that affects the stability of the solutions. By incorporating the aleatoric variance information, we next propose two novel acquisition functions that help Bayesian optimisation to avoid the convergences to the sharp peaks. The chapter then provides theoretical analysis of the algorithm and guarantees that the proposed stable Bayesian optimisation algorithm prefers the stable peaks over unstable ones. Finally, the experimental results with both synthetic function optimisation and hyperparameter tuning show the effectiveness of our proposed method.

• Chapter 4 proposes a novel framework, called cascade Bayesian optimisation, that deals with the optimisation of multi-stage cascaded processes. The proposed cascade Bayesian optimisation algorithm extends the standard approach by considering the objective as a series of optimisation problems that are solved sequentially from the last stage to the first stage. The cost is also incorporated into the formulation for cost-sensitive problems. We then theoretically analyse the cascade Bayesian optimisation algorithm and show the convergence rate of the proposed method. Finally, experiments using a simulated testbed of Al-Sc heat treatment and a data analytic pipeline are conducted to validate the efficiency of our proposed algorithm.

• Chapter 5 addresses the privacy concerns in applying Bayesian optimisation for practical applications. We first propose a novel privacy preserving framework, called Error Preserving Privacy (EPP), that stops an adversary from inferring sensitive information. By focusing directly on the estimation errors, EPP is able to provide privacy guarantees while the perturbation required is lower compared to traditional privacy preserving approaches. Under EPP framework, we present a novel Bayesian optimisation algorithm that can allow experimenters from an industry to utilise the expertise from a third-party optimisation service in a privacy preserving manner. We next demonstrate
the effectiveness of our privacy preserving Bayesian optimisation algorithm on both benchmark functions and optimisation problems from real-world processes. In the remainder of this chapter, we demonstrate the applicability of EPP framework for other machine learning algorithms by presenting a new privacy preserving K-means clustering algorithm. By conducting experiments on both synthetic and real world datasets, we show that the efficiency of the proposed privacy preserving K-means algorithm is comparable to non-private algorithms and significantly better than the baseline.

• Chapter 6 summarises the main content of the thesis, and discusses about potential future research directions.
Chapter 2

Related background

In this chapter, we present preliminary background and discussion of related works to this thesis. This chapter starts with the discussion of experimental design and its importance in scientific research. We then present a brief literature review of global optimisation and popular global optimisation approaches. The next part of this chapter focuses on Bayesian optimisation literature and its applications, followed by the discussion about various choices for the surrogate model and acquisition function of Bayesian optimisation. We then review active research areas of Bayesian optimisation and discuss about open problems that are relevant to practical applications. We end this chapter by reviewing literature of privacy preserving in machine learning, which is one of the problems we aim to tackle in this thesis.

2.1 Experimental design

An experiment is a test or a series of tests that is designed to discover something about a particular process or system. In experiments, changes are purposely made to the input variables so that we may identify the reason for the changes observed in the output responses (Montgomery, 2017). This is different from observational study in which the main task is to observe and collect data without making any changes to the input.
Experiments play vital roles in science and engineering. They appear in many aspects of engineering such as new product design, manufacturing process, process improvement, etc. In many cases, the objective is to plan, conduct the experiments and analyse the observed data such that the useful conclusions are drawn.

Optimised experimental designs can result in immediate products improvement and innovation for a wide range of areas. For example, in material design, suppose a material engineer is interested in designing a new type of alloy. Here the objective is to determine the type and amount of materials to use to produce the highest hardness. The experimenter has to melt and mix different sets of elements and measure the hardness of the specimen. From the results of several such experiments, the engineer selects the combination that produces the highest hardness. Optimised experimental designs will help the material engineer to understand the process better and achieve desired results. Thus, design of experiments is crucial in science and technology.

In the design of experiments, finding the best design efficiently is important since it helps to reduce the cost of experimentation. One approach to find the best design is to model the objective (e.g. hardness) as a function of input variables (e.g. proportion of materials). Then optimisation techniques can be used to find the optima of the objective function. The objective functions are typically non-convex having several peaks or valleys. Finding optima of such functions requires global optimisation and is a non-trivial task. In the following section, we will review the global optimisation problem and popular global optimisation approaches.

\section{2.2 Global optimisation}

The problem of finding global optimum of a function can be written as follows:

\begin{equation}
\max_{x \in \mathcal{X}} f(x)
\end{equation}

where $f$ is the objective function and $\mathcal{X}$ is the domain. The target function $f$ could be as simple as speed of a car, or as complicated as the tastiness of cookies. The domain $\mathcal{X}$ could be amount of burning gasoline, or the proportion of ingredients of
2.2. Global optimisation

There has been a lot of research for global optimisation in these situations. In the following, we will review some global optimisation approaches that may be relevant to this thesis.

**Lipschitz methods** Lipschitz methods are based on the assumption that the objective function is Lipschitz-continuous, meaning there exists a constant $L \geq 0$ for which

$$|f(x_1) - f(x_2)| \leq L \|x_1 - x_2\|, \forall x_1, x_2 \in \mathcal{X}. \quad (2.2)$$

When more and more function values are observed, this assumption effectively gives a lower and upper bound on the function. One can use these bounds to define the next observations to be acquired. There have been many variations of Lipschitz methods (Shubert, 1972; Jones et al., 1993), but one common approach is to evaluate the objective function $f$ where its upper bound is highest. After getting this new observation, the upper bound is updated and maximised to find the next point to evaluate.

**Simulated annealing methods** These methods are based on the physical analogy of cooling crystal structures. In this approach, at iteration $t$, the function is evaluated at the next trial point $x_{t+1}$, and the assignment of the “current point” $x_{\text{curr}}$ depends on the acceptance distribution:

$$P(x_{t+1} \rightarrow x_{\text{curr}}) = \min \left\{ 1, \exp \left[ \frac{f(x_{t+1}) - f(x_{\text{curr}})}{\tau_t} \right] \right\} \quad (2.3)$$

where $\tau$ is the “temperature” parameter. From (2.3), we can see that if $f(x_{t+1}) \leq f(x_{\text{curr}})$, we will assign the newly proposed point $x_{t+1}$ to the current point $x_{\text{curr}}$ with some probability, and we will always move to the new point otherwise. The sequence $\{\tau_t\}$ is referred as cooling schedule, and decreases toward 0, making the algorithm escapes from local optima (Horst and Pardalos, 2013). This method can be applied for combinatorial optimisation problems (Kirkpatrick et al., 1983), or continuous problems (Bélisle et al., 1993).
Genetic algorithms  The term “genetic algorithms” is often referred to the class of algorithms that resembles the natural selection and reproduction processes. In these approaches, a candidate solution can be considered as an individual in a population. Each individuals can evolve over time depending on its adaptiveness to the environment. Candidate solutions are perturbed according to biologically-inspired rules, resulting new candidates. The poorly performed candidates are removed from the population. This process is repeated until no further improvement is made. Some examples of genetic algorithms can be seen in (Holland, 1992; Bethke, 1978; Liepins and Hilliard, 1989).

Multi-start local optimisation  When function form is known and its derivatives exist, local optimisation methods can work really well, and a natural approach is to use them for global optimisation. The simplest adaptation of local optimisation methods is using random restarts: one can uniformly draw points from search domain $\mathcal{X}$, and have a local optimiser started from each point. As the number of starting points approaches infinity, all local maxima may have been found, thus we can simply select the best one we have observed.

2.3 Bayesian optimisation

Thus far, we have reviewed several global optimisation approaches. These approaches work well for optimisation of functions that have either known mathematical forms or cheap to evaluate. However, in the case the objective functions are unknown black-box functions and expensive to evaluate, standard global optimisation approaches often have poor performance. In this section, we review Bayesian optimisation - an efficient global optimisation technique for expensive black-box functions. We will start by providing an overview of Bayesian optimisation, followed by a discussion of successful applications of Bayesian optimisation.
Bayesian optimisation (Kushner, 1964; Mockus et al., 1978; Mockus, 1994) is a well-known methodology to find the optimum of an unknown expensive black-box function $f$. This optimisation problem can be formally defined as:

$$x^* = \arg\max_{x \in \mathcal{X}} f(x)$$

(2.4)

where $\mathcal{X}$ is the domain of $x$. In global optimisation, $\mathcal{X}$ is often a compact subset of $\mathbb{R}^d$ but Bayesian optimisation can be applied for more unusual search spaces such as combinatorial search spaces. It is assumed that although $f$ is a black-box function without a closed-form expression, it can be evaluated at any point $x$ in the domain. This evaluation produces the output $y = f(x) + \epsilon$, where $\epsilon$ is the measurement noise. Although this is the minimum requirement for Bayesian optimisation framework, gradient information, if available, can be incorporated to the algorithm as well.

Algorithm 2.1 Bayesian optimisation algorithm

1: **Input:**
2:  Initial observation set $D_{t_0} = \{x_{1:t_0}, y_{1:t_0}\}$.
3:  Bounds for the search space $\mathcal{X}$.
4: **Output:** $\{x_t, y_t\}_{t=1}^{T}$
5: **for** $t = t_0, \ldots, T-1$
6:  Construct a surrogate model and an acquisition function $\alpha(x; D_t)$ (see Section 2.5 for details).
7:  Select $x_{t+1}$ by maximising acquisition function $\alpha$

$$x_{t+1} = \arg\max_{x \in \mathcal{X}} \alpha(x; D_t)$$

8:  Evaluate the target function to obtain $y_{t+1}$.
9:  Augment the observation set: $D_t = D_t \cup \{x_{t+1}, y_{t+1}\}$.
10: **end for**

Bayesian optimisation is a sequential search algorithm. Given a few input and output observations of the function $f$, Bayesian optimisation iteratively suggests the next sample for evaluation to find the optimum value of the function. In other words, at iteration $t$, Bayesian optimisation selects a location $x_{t+1}$ at which to query $f$ and get the observation. After a predefined number of queries $T$ (defined by evaluation budget), the algorithm stops and returns its best estimates of the
Bayesian optimisation framework consists of two key components. The first is a (probability) surrogate model that can be evaluated at any point. This surrogate model uses prior knowledge, such as smoothness, about the cost function and known datapoints to capture and update our beliefs about the function. There are plenty of choices for this function such as Gaussian process (Rasmussen and Williams, 2006), neural networks (Snoek et al., 2015), random forest (Brochu et al., 2010b), etc. The second component of a Bayesian optimisation algorithm is an acquisition function that suggests where to evaluate the function next. After observing the output of each function evaluation, the algorithm updates the prior and produces a more informative posterior distribution over objective functions. Using the acquisition function, the original problem becomes optimising a less expensive non-convex function. The acquisition function maintains a trade-off between exploration (where the surrogate model has high uncertainty about the function value) and exploitation (where the surrogate model has high predictive value). This technique is able to minimise the number of the function evaluations. The details of Bayesian optimisation algorithm are provided in Algorithm 2.1.

Fig. 2.1 is the illustration of using Bayesian optimisation for optimising a 1-D function. The surrogate model is the Gaussian process and the acquisition function is upper confidence bound (UCB, see (2.21)). As we can see from the figures, after 3 steps, there are 3 regions with high uncertainties including the region with global optimum. Bayesian optimisation continuously suggests points for evaluation and after 6 iterations, the global optimum is found.

### 2.3.2 Applications of Bayesian optimisation

Before going into details of two main components of Bayesian optimisation, we present an overview of various applications of Bayesian optimisation.
2.3. Bayesian optimisation

Figure 2.1: Visualisation of the surrogate model and the acquisition function of Bayesian optimisation. In this illustration, a Gaussian process and its upper confidence bound (UCB) are used for the surrogate model and the acquisition function. The graph in red colour shows the true objective function. The dotted line represents the mean of Gaussian process posterior. The blue area indicates 95% confidence interval. The yellow triangle represents maximum value of the acquisition function, suggesting the next point for evaluation.
2.3. Bayesian optimisation

Figure 2.1: Visualisation of the surrogate model and the acquisition function of Bayesian optimisation. In this illustration, a Gaussian process and its upper confidence bound (UCB) are used for the surrogate model and the acquisition function. The graph in red colour shows the true objective function. The dotted line represents the mean of Gaussian process posterior. The blue area indicates 95% confidence interval. The yellow triangle represents maximum value of the acquisition function, suggesting the next point for evaluation.
2.3. Bayesian optimisation

Robotics and reinforcement learning

Gait optimisation is a fundamental and challenging problem in robotics research. The tuning of gait parameters is time-consuming even for a simple robot. In Lizotte et al. (2007), the authors proposed a new approach to gait learning based on Gaussian process optimisation. By parameterising the robot’s gait, the optimisation for gait velocity and gait smoothness was done on the Sony AIBO ERS-7 robot (Lizotte et al., 2007). In another work, different gait optimisation methods were compared and the Bayesian optimisation approach was shown to perform considerably better than grid or random search (Calandra et al., 2014b). Similarly, Bayesian optimisation was successfully applied for mobile robot to find the optimal sensing path (Martinez-Cantin et al., 2009).

An example of using Bayesian optimisation for reinforcement learning was presented in Brochu et al. (2010b), where Bayesian optimisation was used in tuning the parameters of a neural network policy and learning value functions at higher levels of the hierarchy. In another work, Wilson et al. (2014) proposed an effective approach for applying Bayesian optimisation to reinforcement learning by exploiting the sequential trajectory information generated by reinforcement learning agents.

Environmental monitoring and sensor networks

In environmental monitoring, sensors networks are used to monitor quantities such as temperature, pollution level in soil, oceans, etc. Using sensor networks, we might find locations that have the highest temperature in a building by activating sensors in a spatial network and regressing on their measurements (Srinivas et al., 2010). Since the activating process costs battery power, in Srinivas et al. (2010), the authors use Bayesian optimisation to minimise number of sensors for activation. In another similar work, Bayesian optimisation was used in Garnett et al. (2010) to identify the best choice of sensor locations for meteorological purposes.

For moving robots, there is a cost for the robots to travel to specific locations. This cost is incorporated to a new acquisition function for Bayesian optimisation algorithm in (Marchant and Ramos, 2012, 2014).
2.3. Bayesian optimisation

**Active learning**

Many computer graphic and animation applications require the setting of tricky parameters. In those applications, the models are often complex and the parameters are intuitive for non-experts. In Brochu et al. (2010a), the authors proposed an approach to optimise the parameters of an animation system by showing the users examples of parameterised animations and asking for feedback. Similarly, Brochu et al. (2008) used preference gallery approach to solve material design process, in which users are required to indicate which materials they prefer from images rendered with different material properties. This approach is particularly useful for situations where human input is required for learning.

**Automatic machine learning and hyperparameter tuning**

The goal of automatic machine learning and hyperparameter tuning is to automatically select the best predictive model (e.g. neural networks, random forests, support vector machines, etc.) for a given problem; and/or choose the best set of hyperparameters that yields the highest performance with a given model. For big datasets, since the cross-validation process is time-consuming, it is really important to select the best model and its hyperparameters within the cross-validation budget.

There are many applications of Bayesian optimisation in tuning deep belief networks (Bergstra et al., 2011), convolution neural networks (Snoek et al., 2012), Markov chain Monte Carlo methods (Mahendran et al., 2012). In automatic model selection, several uses of Bayesian optimisation have been introduced in (Thornton et al., 2013; Hoffman et al., 2014).

**Material design**

The discovery and development of new materials require conducting experiments, which includes many different input variables. As the complexity increases, the variable exploration process is impractical using traditional combinatorial approaches. In Frazier and Wang (2016), the authors introduced an application of Bayesian op-
timisation in material design using expected improvement and knowledge gradient methods. In another work, Li et al. (2017) proposed an approach using Bayesian optimisation to optimise process development, incorporating multiple qualitative and quantitative objectives. The authors demonstrated an example of using the proposed method for a developmental experimental system to achieve material and process objectives. There have been several successful applications of Bayesian optimisation in optimising heat treatment process of an Al-Sc alloy (Gupta et al., 2018; Nguyen et al., 2016), optimising alloy composition (Li et al., 2018; Rana et al., 2017) and optimising a short polymer fibre production process (Vellanki et al., 2017).

### 2.4 Surrogate models

In this section, we discuss about surrogate models - the first component in Bayesian optimisation. We will start with an in-depth description of Gaussian processes - the most common statistical model for Bayesian optimisation, followed by a literature review of other surrogate models such as random forest and deep neural networks.

#### 2.4.1 Gaussian processes

Gaussian processes (Rasmussen and Williams, 2006) are a principled probabilistic method for modelling data. For a set of observed data, instead of learning a single ‘best-fit’ (point estimate), Gaussian processes provide a complete posterior distribution over possible functions. A multivariate probability distribution represents a belief over the values of finite dimensional random vectors. Extending from this concept, a stochastic process provides the distribution over functions. As its name suggests, a Gaussian process is a stochastic process such that every finite collection of its random variables are jointly Gaussian. Intuitively, one can think of a Gaussian process as a multivariate Gaussian distribution over an infinite dimensional vector. Fig. 2.2 illustrates an example of a Gaussian process prior and posterior with few observations.
2.4. Surrogate models

Figure 2.2: An example of Gaussian process for a one dimensional function. Different shades of blue represent the predictive density at each input location. The colour lines illustrate some samples drawn from Gaussian posterior distribution. Fig. 2.2a shows a Gaussian process prior and its samples. The remaining plots show Gaussian process posteriors after observing few samples from a function.
A Gaussian process is specified by its mean function

\[ m(x) = \mathbb{E}[f(x)] \]  
(2.5)

and its covariance function:

\[ k(x, x') = \text{Cov}[f(x), f(x')] \].  
(2.6)

A function \( f(x) \) drawn from a Gaussian process with mean \( m(x) \) and covariance function \( k(x, x') \) is denoted as follows:

\[ f(x) \sim \mathcal{GP}(m(x), k(x, x')). \]

Without any loss in generality, the mean function \( m(x) \) can be assumed to be zero everywhere. By doing so, the Gaussian processes depend only on the covariance function \( k(x, x') \). In the following, we will first discuss about modelling data and making predictions using Gaussian processes, followed by a discussion about popular choices for covariance functions and other properties of Gaussian processes.

**Modelling data and making predictions with Gaussian processes**

In this section, we look at how we can build a Gaussian process model from a set of observations of a function and find the posterior distribution on function values. Roughly speaking, this can be viewed as finding a set of functions allowed by prior that fit the observed data. Let \( f(x) \) be an unknown function that maps a \( d \)-dimensional input value \( x \) to a scalar output value \( f \). Assume that we have a set of \( t \) observations

\[ D_t = \{(x_i, y_i)\}, i = 1, 2, ..., t. \]

Let us denote \( X \) as the \( t \times d \) input matrix and \( y \) as a \( t \) element output vector. Assuming the mean function to be zero everywhere, the prior using Gaussian process \( \mathcal{GP}(0, k(x, x')) \) is written as follows:
2.4. Surrogate models

\[ p(f \mid X) = \mathcal{N}(0, K) \]

where \( K \) is the covariance matrix

\[
K = \begin{bmatrix}
k(x_1, x_1) & \cdots & k(x_1, x_t) \\
\vdots & \ddots & \vdots \\
k(x_t, x_1) & \cdots & k(x_t, x_t)
\end{bmatrix}.
\]

To find the posterior distribution on function values given the observed data, we need to quantify the relationship between the observed output \( y \) with the unknown function value \( f(x) \) at the same input location. We consider the noisy output case

\[ y = f(x) + \epsilon \]

where \( \epsilon \) is measurement noise. For tractability purpose, we assume that the noise is Gaussian and independent on each data point:

\[ \epsilon \sim \mathcal{N}(0, \sigma^2_\epsilon), \]

where \( \sigma^2_\epsilon \) is the noise variance. Note that every finite collection of variables in a Gaussian process is also Gaussian. Using this property, one can make predictions by considering the joint distribution of the old observations \( D_t \) and a new observation \( (x_{t+1}, y_{t+1}) \):

\[
\begin{bmatrix}
y_{1:t} \\
y_{t+1}
\end{bmatrix} \sim \mathcal{N}
\left(0, \begin{bmatrix}
K + \sigma^2_\epsilon I & k \\
k^T & k^T(x_{t+1}, x_{t+1})
\end{bmatrix}
\right)
\]

where \( k = [k(x_1, x_{t+1}), k(x_2, x_{t+1}), \ldots, k(x_t, x_{t+1})]^T \). Using Gaussian conditioning, the posterior distribution of the function value at \( x_{t+1} \) can be written as:

\[
p(y_{t+1} \mid y_{1:t}, x_{1:t+1}) = \mathcal{N}(\mu_t(x_{t+1}), \sigma^2_t(x_{t+1}))
\]

where

\[
\begin{align*}
\mu_t(x_{t+1}) &= k^T \left[ K + \sigma^2_\epsilon I \right]^{-1} y_{1:t} \\
\sigma^2_t(x_{t+1}) &= k(x_{t+1}, x_{t+1}) - k^T \left[ K + \sigma^2_\epsilon I \right]^{-1} k.
\end{align*}
\]
Covariance functions

In a Gaussian process predictor, a covariance function (or kernel) is the crucial component to define the characteristic of the Gaussian process, as it encodes the structure of the functions we wish to learn (Rasmussen and Williams, 2006). There are several kernels that can be used as covariance functions in Gaussian process such as linear kernel, polynomial kernel, squared exponential kernel, Matérn kernel, \( \gamma \)-exponential kernel, rational quadratic kernel, etc (Rasmussen and Williams, 2006; Duvenaud, 2014).

A common choice for covariance function is stationary kernel. It is based on a scaled distance \( \delta \) between input \( \mathbf{x} \) and \( \mathbf{x}' \):

\[
\delta^2 = (\mathbf{x} - \mathbf{x}')^T \mathbf{W}^{-1} (\mathbf{x} - \mathbf{x}')
\] (2.11)

where \( \mathbf{W} \) is a \( d \times d \) diagonal matrix, \( \mathbf{W}^{-1} = \text{diag}(\frac{1}{l_1^2}, \frac{1}{l_2^2}, ..., \frac{1}{l_d^2}) \). Using stationary kernels, the covariance between two function values \( f(\mathbf{x}) \) and \( f(\mathbf{x}') \) depends solely on the distance between \( \mathbf{x} \) and \( \mathbf{x}' \), not on the actual values of them. In addition, the covariance between two function values generally increases when their corresponding inputs are closer, making it suitable for continuous functions.

One frequently used covariance function is squared exponential kernel (a.k.a Radial Basis Function kernel), defined as follows:

\[
k_{SE}(\mathbf{x}, \mathbf{x}') = \exp(-\frac{\delta^2}{2})
\] (2.12)

Functions generated from squared exponential kernel are indefinitely mean-square differentiable, making it a popular choice as a covariance function for Gaussian processes.

When control over differentiability is required, the Matérn(\( \nu \)) family of covariance functions are usually used. The function \( f(\mathbf{x}) \) generated by this class of kernels is \( k \)-times mean-square differentiable if and only if \( \nu > k \). When \( \nu \) is half-integer (i.e. \( \nu = p + \frac{1}{2}, p \) is a non-negative integer) the Matérn covariance functions become...
especially simple. Expressions for some of Matérn kernels are given below:

\[
\begin{align*}
    k_{\nu=1/2}(x, x') &= \exp(-\delta), \\
    k_{\nu=3/2}(x, x') &= \left(1 + \sqrt{3}\delta\right) \exp\left(-\sqrt{3}\delta\right), \\
    k_{\nu=5/2}(x, x') &= \left(1 + \sqrt{5}\delta + \frac{1}{3} \left(\sqrt{5}\delta\right)^2\right) \exp(-\sqrt{5}\delta), \\
    k_{\nu=7/2}(x, x') &= \left(1 + \sqrt{7}\delta + \frac{2}{5} \left(\sqrt{7}\delta\right)^2 + \frac{1}{15} \left(\sqrt{7}\delta\right)^3\right) \exp(-\sqrt{7}\delta).
\end{align*}
\] (2.13) (2.14) (2.15) (2.16)

It can be shown that when \(\nu \to \infty\) the Matérn kernel becomes squared exponential. In practice, we don’t generally have the prior knowledge about the existence of higher order derivative; so it is probably very difficult to distinguish between values of \(\nu \geq 7/2\) (Rasmussen and Williams, 2006). Fig. 2.3 presents a visualisation of various kernel profiles and sample from posteriors.

**Properties of Gaussian processes**

There are several reasons for using Gaussian processes for function modelling:

- The posterior distribution can be computed exactly in closed form from kernel functions and some observations. This is rare among Bayesian non-parametric models.
- Taking advantages of the variety of covariance functions, Gaussian processes can express a wide range of functions. The covariance functions can also be combined in Gaussian processes to represent more complex structure of functions.
- Gaussian processes allows us to compute *marginal likelihood* of the data given a model (MacKay, 1992), helping to compare different models for model selection purpose (MacKay and Mac Kay, 2003; Rasmussen and Ghahramani, 2001; Duvenaud, 2014).
- Using Gaussian processes, overfitting problem is less of an issue since only few parameters need to be estimated (compare to neural networks for example).

However, Gaussian processes have certain drawbacks:
Figure 2.3: Example of various kernel profiles sampled from a Gaussian process posterior. Note that when \( \nu \) increases, the function samples become smoother.
2.4. Surrogate models

- The computational complexity of matrix inversion in Eq. (2.9) and Eq. (2.10) is $O(t^3)$, making exact computation of predictions slow for number of observations greater than a thousand. This problem can be overcome by using approximate inference methods (Hensman et al., 2013; Quiñonero-Candela and Rasmussen, 2005).

2.4.2 Random forests

Random forest is a standard machine learning technique for classification and regression problems (Breiman, 2001). Random forest uses decision trees trained from random subsamples of data and averages the predictions of weak learners to produce accurate response surface.

As a surrogate model, random forest regression has been proposed as an expressive and flexible statistical model for sequential model-based algorithm configuration (SMAC) (Hutter et al., 2011; Shahriari et al., 2016). Since decision tree is flexible with various data types, random forest can easily handle categorical data and conditional variables. In addition, because of the scalability and parallelism, random forest can be readily used for large evaluation budgets which is an advantage compare to the cubic complexity of exact Gaussian process. By subsampling of dimensions to fit decision rule at each iteration, using random forest as surrogate model also helps to deal with high dimensional search spaces (Hutter et al., 2011; Shahriari et al., 2016).

Using random forest as a surrogate has certain drawbacks. First of all, random forest does not provide the uncertainty in the estimate. Hutter et al. proposed to use empirical variance across the tree predictions as the uncertainty of posterior (Hutter et al., 2011). This heuristic technique might not be accurate, but it has been shown to work well in the context of SMAC. Although being a good predictor for the neighbourhood of training data, random forest is terrible in predicting data points that are far from training data (Shahriari et al., 2016). Indeed, the predictions for those data points are almost identical, resulting in poor prediction (Shahriari et al., 2016). In addition, using variance estimate for the posterior’s uncertainty results in extremely confident intervals. Although Gaussian process is also poor
2.4. Surrogate models

with extrapolation, it still produces uncertainty for posterior, making it suitable for exploitation and exploration trade-off. Finally, random forest’s response surface is discontinuous and non-differentiable so the acquisition functions using random forest are optimised by a combination of local and random search without using gradient-based methods (Shahriari et al., 2016).

2.4.3 Deep neural networks

Similar to random forest, Bayesian deep neural network is a surrogate model for Bayesian optimisation which provide scalability while maintaining the flexibility and characterisation of uncertainty (Snoek et al., 2015). In (Snoek et al., 2015), the authors proposed a Bayesian optimisation algorithm using deep neural networks called Deep Networks for Global Optimisation (DNGO). The aim of this work is to replace Gaussian process by Bayesian neural networks to retain most of desirable properties of Gaussian process such as flexibility and well-calibrated uncertainty (Snoek et al., 2015). However, since Bayesian neural networks are computationally expensive at large scale, the authors proposed to add a Bayesian linear regressor to the last hidden layer of a deep neural network. Only the output weights of the net are marginalised making the algorithm scale linearly with number of observations and cubically with number of dimensions in basis function (Snoek et al., 2015). For suitable design choice, the author showed that it is possible to create robust and effective Bayesian optimisation system that generalises across many problems (Snoek et al., 2015). The performance of DNGO is shown to be competitive with other GP-based approaches (Snoek et al., 2015).

In another line of work, Springenberg et al. (2016) used Bayesian neural networks to create a robust, scalable and parallel optimiser called Bayesian Optimisation with Hamiltonian Monte Carlo Artificial Neural Network (BOHAMIANN). This algorithm supports both single-task and multi-task optimisation and can be used for high-dimensional optimisation as well as parallel function evaluations. The authors also used scale adaptation technique to improve the robustness of Bayesian inference using stochastic gradient Hamiltonian Monte Carlo (Springenberg et al., 2016).
2.5 Acquisition functions

Thus far, we have discussed about popular probabilistic models used to represent our belief about objective function $f$ at iteration $t$. We have not described the mechanism for sequentially selecting the next point $x_{t+1}$ for function evaluation. Naturally, one can choose an arbitrarily point $x_{t+1}$ to be evaluated. However, it would be wasteful given information obtained from surrogate models. There is a rich literature that leverages the uncertainty of posterior models to guide the search. In this section, we will focus on the mechanism for selection of next query point $x_{t+1}$, often realised by constructing an acquisition function. Fig. 2.4 shows some examples of acquisition functions.

The role of the acquisition function is to recommend the next sample for function evaluation. After defining an acquisition function, the original optimisation problem is approached by maximising the acquisition function $\alpha$ as follows:

$$x_{t+1} = \arg \max_{x \in X} \alpha(x; D_t)$$

where $D_t$ denotes the data set consists of $t$ observations. Typically, the acquisition function is defined such that its high value potentially leads to a high value of objective function $f$. The trade-off between exploring highly uncertain regions or exploiting promising areas is also represented in the acquisition function. Choosing acquisition function is a nontrivial task. In this section, we will discuss several different acquisition functions that have been proposed in the literature. We will first present traditional improvement-based and optimistic approaches, followed by more recent information-based acquisition functions.

2.5.1 Improvement-based policies

Improvement-based acquisition functions suggest points that are likely to improve upon a specific target. Probability of improvement (PI) is an early improvement-based acquisition function in literature proposed by Kushner (1964). This acquisition function measures the probability that an input $x$ leads to a function value that is greater than the best function value discovered so far $f(x^+)$, where
2.5. Acquisition functions

Figure 2.4: Examples of acquisition functions. The top plot is the Gaussian model with 3 observations. The remaining plots are the acquisition functions for the Gaussian process: PI - Probability Improvement (see Eq. (2.17)), EI - Expected Improvement (see Eq. (2.19)), UCB - Upper Confidence Bound (see Eq. (2.21)). The yellow triangle makers are the maximum of the acquisition functions.
\( \mathbf{x}^+ = \arg\max_{x_i \in x_1:t} f(x_i) \). Since the posterior of the objective function \( f \) is Gaussian, the probability of improvement acquisition function can be analytically computed as follows:

\[
\alpha_{PI}(\mathbf{x}; D_t) = \mathbb{P} \left( f(\mathbf{x}) \geq f(\mathbf{x}^+) \right) = \Phi \left( \frac{\mu(\mathbf{x}) - f(\mathbf{x}^+)}{\sigma(\mathbf{x})} \right)
\]

where \( \Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-z^2/2} dz \) is the standard normal cumulative distribution function. This acquisition function prefers points that have a high probability of being greater than \( f(\mathbf{x}^+) \) rather than points that might return larger gain but having higher uncertainty. In other words, probability of improvement acquisition function is highly exploitative. One can overcome this issue by adding a trade-off parameter \( \xi \geq 0 \):

\[
\alpha_{PI}(\mathbf{x}; D_t) = \mathbb{P} \left( f(\mathbf{x}) \geq f(\mathbf{x}^+) + \xi \right) = \Phi \left( \frac{\mu(\mathbf{x}) - f(\mathbf{x}^+) - \xi}{\sigma(\mathbf{x})} \right).
\] (2.17)

As recommended by Kushner, \( \xi \) can be set fairly high early in the optimisation to encourage exploration and decreased toward zero as the optimisation continues. The empirical impact of different values of parameter \( \xi \) has been studied in several domains (Jones, 2001; Lizotte, 2008).

An alternative approach is to maximise the expected improvement (EI) that involves the improvement function defined by Mockus et al. (1978) as follows:

\[
I(\mathbf{x}) = \max \left\{ 0, f_{t+1}(\mathbf{x}) - f(\mathbf{x}^+) \right\}.
\] (2.18)

The probability of the improvement function \( I(\mathbf{x}) \) can be computed from the density function of the normal distribution \( \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x})) \) as:

\[
P(I(\mathbf{x})) = \frac{1}{\sqrt{2\pi}\sigma(\mathbf{x})} \exp \left[ -\frac{(\mu(\mathbf{x}) - f(\mathbf{x}^+) - I)^2}{2\sigma^2(\mathbf{x})} \right].
\]

Then the expected improvement acquisition function can be computed as follows:
2.5. Acquisition functions

\[ \alpha_{EI}(\mathbf{x}; D_t) = \mathbb{E}[I(\mathbf{x})] \]

\[ = \int_{I=0}^{I=\infty} \frac{1}{\sqrt{2\pi} \sigma(I)} \exp \left[ -\frac{(\mu(I) - f(I^+) - I)^2}{2\sigma^2(I)} \right] dI. \]

This integral can be evaluated analytically (Mockus et al., 1978; Jones et al., 1998), yielding the following results:

\[ \alpha_{EI}(\mathbf{x}; D_t) = \begin{cases} 
    z\sigma(\mathbf{x})\Phi(z) + \sigma(\mathbf{x})\phi(z) & \text{if } \sigma(\mathbf{x}) > 0 \\
    0 & \text{if } \sigma(\mathbf{x}) = 0 
\end{cases} \quad (2.19) \]

where \( z = \frac{\mu(\mathbf{x}) - f(\mathbf{x}^+)}{\sigma(\mathbf{x})} \), \( \Phi(z) \) is the standard normal cumulative distribution function and \( \phi(z) \) is the standard normal probability density function. The EI function has been well studied theoretically (Bull, 2011) and proven to be effective in practice (Snoek et al., 2012).

Unlike the expected improvement acquisition function, the knowledge gradient policy (Frazier et al., 2009; Scott et al., 2011) does not depend on the assumption that the evaluations are noise free and the final recommendation must belong to the set of previously sampled points. This make knowledge gradient acquisition function possible to explore a broader class of solutions. The knowledge gradient policy chooses the next point for evaluation by maximising the expected incremental value of a measurement without assuming that the point returned must be a previously sampled point (Wu and Frazier, 2016). Formally, the knowledge gradient acquisition function is defined as follows:

\[ \alpha_{KG}(\mathbf{x}; D_t) = \mathbb{E} \left[ \max_{\mathbf{z} \in \mathcal{X}} \mu(\mathbf{z}) | D_t \cup \{ \mathbf{x} \} \right] - \max_{\mathbf{z} \in \mathcal{X}} \mu(\mathbf{z}) | D_t \quad (2.20) \]

where \( \mathcal{X} \) is the discrete search domain and the expectation is taken with respect to the posterior distribution after \( t \) evaluations. If the functions evaluations are noise-free and the final recommendation is restricted to previously sampled points, the knowledge gradient policy will become EI acquisition function, making it the generalisation of expected improvement policy.
2.5.2 Optimistic policies

In this class of strategies, the principle is to be optimistic in the face of the uncertainty in the posterior. Indeed, using the upper confidence for every query point \( x \) translates to effectively using a fixed probability best case scenario according to the model (Shahriari et al., 2016). The idea of using upper confidence bound criterion for exploitation - exploration trade-off has been first introduced in (Lai and Robbins, 1985). The upper confidence bound acquisition function can be defined as follows:

\[
\alpha_{UCB}(x; D_t) = \mu_t(x) + \lambda_t \sigma_t(x) \tag{2.21}
\]

where \( \lambda_t \) is the parameter represents the trade-off between exploitation and exploration. Recently, Srinivas et al. (2010) proposed Gaussian process upper confidence bound (GP-UCB) as an optimistic acquisition function with provable cumulative regret bounds. Formally, Gaussian process upper confidence bound is defined as follows (Srinivas et al., 2010):

\[
\alpha_{GP-UCB}(x; D_t) = \mu_t(x) + \kappa_t \sigma_t(x) \tag{2.22}
\]

where \( \kappa_t \) is a positive parameter that balances exploitation and exploration. Maximising a GP-UCB acquisition function suggests the point where to evaluate the target function \( f \) next. Srinivas et al. (2010) proved that if \( \kappa_t = 2 \log \left( t^2 2\pi^2/3\delta \right) + 2d \log \left( t^2 dbr \sqrt{\log(4da/\delta)} \right) \), GP-UCB achieves an upper bound on the cumulative regret \( R_T = \sum_{t=1}^{T} (f(x^*) - f(x_t)) \) that has the order \( O \left( \sqrt{T \gamma_T} \right) \) for all \( T \geq 1 \), with the probability greater than or equal to \( 1 - \delta \), where \( \gamma_T \) is the maximum information gain after \( T \) iterations, the search space is \([0, r]^d\) with some \( r > 0 \) and \( a, b > 0 \) are some constants related to the function smoothness.

2.5.3 Information-based policies

Different from the above acquisition functions, information-based acquisition functions take advantage of the posterior distribution over the unknown maximiser \( x^* \). We denote this distribution as \( p(X^* \mid D_t) \) where \( X^* \) is the random variable that models the maximum location \( x^* \). Two popular policies in this class are Thomson
Thompson sampling (Thompson, 1933) is one of the earliest works to minimise regret in stochastic bandit problems. It is a member of family of randomised probability matching algorithms (Agrawal and Goyal, 2012). Thompson sampling randomly samples a reward function from posterior and chooses the arm with the highest simulated award. Recently, the multi-arm bandit community has produced several practical evaluations (Scott, 2010; Chapelle and Li, 2011; May and Leslie, 2011) and theoretical results (Kaufmann et al., 2012; Agrawal and Goyal, 2013; Russo and Van Roy, 2014) of Thompson sampling. In the continuous search spaces, Thompson sampling draws a function $f_t$ from Gaussian process posterior and optimises it to obtain next query point $x_{t+1}$. However, optimisation of $f_t$ using standard optimisation methods requires $f_t$ to be fixed so it can be evaluated at arbitrary points. In order to use Thompson sampling for continuous search spaces, we can draw an approximate sample from Gaussian process posterior that can be queried at any arbitrary points (Hernández-Lobato et al., 2014) using spectral sampling technique (Bochner, 1959). Formally, Thompson sampling acquisition function can be defined as follows:

$$\alpha_{TS}(\mathbf{x}; D_t) = f_t(\mathbf{x})$$

(2.23)

where $f_t$ is the function randomly drawn from Gaussian process posterior via spectral sampling.

Instead of drawing functions from posterior distribution of $\mathbf{x}^*$, the aim of entropy search acquisition functions is to reduce the uncertainty in the location of unknown maximiser $\mathbf{x}^*$. To achieve this goal, entropy search techniques select the point that produces the largest entropy reduction of the posterior distribution over $\mathbf{x}^*$, e.g. $p(\mathbf{X}^* | D_t)$ (Hennig and Schuler, 2012). Formally, the entropy search acquisition function can be defined as follows:

$$\alpha_{ES}(\mathbf{x}; D_t) = H(\mathbf{X}^* | D_t) - \mathbb{E}_{y|x; D_t}[H(\mathbf{X}^* | D_t \cup \{(x, y)\})].$$

(2.24)

where $H(\mathbf{X}^* | D_t)$ is the entropy of the posterior distribution $p(\mathbf{X}^* | D_t)$, the subtrahend is the expectation of the entropy of $\mathbf{X}^*$ after observing $(x, y)$ and the expectation is taken with respect to $p(y | x, D_t)$. Similar to Thompson sampling, the entropy search acquisition function is intractable in continuous spaces. The
entropy search acquisition function can be approximately computed by discretising the search space \( X \) and compute entropy via Monte Carlo sampling (Villemonteix et al., 2009); or by discretising \( X \) to obtain a smooth approximation of \( p(X^* \mid D_t) \) and its expected information gain (Hennig and Schuler, 2012).

Using the symmetric property of mutual information, predictive entropy search acquisition function does not require discretisation of the search spaces (Hernández-Lobato et al., 2014). The predictive entropy search acquisition function can be defined as follows:

\[
\alpha_{PES}(x; D_t) = H(y \mid D_t, x) - \mathbb{E}_{x^* \mid D_t} [H(y \mid D_t, x, x^*)].
\]  

(2.25)

Unlike entropy search, the predictive entropy search acquisition function based on the entropy of predictive distribution \( H(y \mid D_t, x) \) which can be easily approximated, rather than the entropy of distribution over \( x^* \) which is difficult to approximate (Hernández-Lobato et al., 2014). Empirically, this acquisition function has been shown to perform better than the entropy search (Hernández-Lobato et al., 2014).

2.6 Active research areas

In the previous sections, we have reviewed literature of Bayesian optimisation and its components. In this section, we present some of the active research areas in Bayesian optimisation.

2.6.1 High-dimensional problems

Despite having many advances in last few years, Bayesian optimisation still has limitation when applied in high dimensional search spaces. The main problem arises because Bayesian optimisation uses a surrogate model which has the same number of dimensions as the search space. Using high dimensional models, the acquisition functions can be extremely sharp which makes optimising these functions infeasible for standard optimisation packages. Solving high-dimensional problem is also difficult since the number of evaluations needed for good coverage of \( X \) increases
2.6. Active research areas

exponentially with number of dimensions.

In Chen et al. (2012), the authors proposed a method to tackle high-dimensional problem by assuming that the unknown function depends only on few relevant variables. This method consists of two stages: variable selection using sequential likelihood ratio tests and optimisation over active variables using GP-UCB. The sampling complexity bounds are provided, guaranteeing strong end-to-end performance of Gaussian process optimisation in high dimensional spaces.

With a similar assumption that the problems have “low effective dimensionality”, Wang et al. (2013) proposed random embedding Bayesian optimisation (REMBO) algorithm for high-dimensional data. In this work, the number of dimensions is reduced by projecting the original space to a lower dimension space. The performance of REMBO on high-dimensional space provides evidence that for many practical problems, the number of important dimensions is much lower than their extrinsic dimensionality (Bergstra and Bengio, 2012; Wang et al., 2013).

There have been several works for high dimensional Bayesian optimisation without assuming that the dimensions are correlated. Rana et al. (2017) proposed a high dimensional Bayesian optimisation algorithm with elastic Gaussian process by leveraging two underlying facts: a) gradient can be boosted by increasing the length-scales of the Gaussian process prior and b) the extrema of the acquisition functions are close if the difference in the length-scales is small. Li et al. (2018) proposed a novel method for high dimensional Bayesian optimisation using a drop-out strategy. In Oh et al. (2018), the authors proposed a Bayesian optimisation algorithm for high dimensional spaces that transforms the ball geometry of the search space with a cylindrical transformation.

2.6.2 Constrained Bayesian optimisation

Many real world optimisation problems have constraints that make certain regions of search space $\mathcal{X}$ infeasible. In Gelbart et al. (2014), a scenario was described in which a food company want to design a new cookie recipe that minimises the calories while keeping the tastiness similar to the current ones. In this scenario, the objective function is a function representing number of calories and the constraint is the
tastiness that must be greater than a certain threshold. The constrained optimisation problem has applications in many areas such as product design (e.g. designing a alloy), machine learning hyperparameter tuning (e.g. find the hyperparameters so that the prediction time is lower than a threshold), or real-time system (e.g. a speech recognition system on a device with limited resources).

The constraints can be incorporated directly into the optimisation of acquisition functions if they are known a priori. However, in many cases, they are unknown (i.e. a black-box). Several approaches address the unknown constraint problem by proposing new acquisition functions.

In Gramacy and Lee (2010), the authors proposed the integrated expected conditional improvement (IECI) acquisition function:

$$\alpha_{IECI}(x; \mathcal{D}) = \int_{\mathcal{X}} (\alpha_{EI}(u; \mathcal{D}) - \alpha_{EI}(u; \mathcal{D} \cup u | x)) g(u) du$$  \hspace{1cm} (2.26)

where $\alpha_{EI}$ is the expected improvement acquisition function (Eq. (2.19)) and $g(u)$ is a probability density function (Gramacy and Lee, 2010). In this approach, the constraints are handled through $g(u)$ by making $g(u)$ uniform for $u$ satisfying the constraints and zero otherwise; or by giving higher weight to $u$ if it has greater chance of satisfying the constraints.

In another approach, Gelbart et al. (2014) proposed the weighted expected improvement (WEI) acquisition function:

$$\alpha_{WEI}(x; \mathcal{D}) = \alpha_{EI}(x; \mathcal{D}) g(x; \mathcal{D})$$  \hspace{1cm} (2.27)

where $g(x; \mathcal{D})$ represents the probability of $x$ satisfying the constraints. This encourages WEI to favour regions with high probability of being valid.

Gardner et al. (2014) proposed a variant of WEI to deal with the constraint on function values. Instead of using the $g(x; \mathcal{D})$ as the probability function of $x$ satisfying the constraint, they used it as the posterior probability of function value at $x$ being smaller than a value $\lambda$ (i.e. $f(x) < \lambda \mid \mathcal{D}$) under the Gaussian process model of the function.

In Hernández-Lobato et al. (2015), the authors proposed an information-based ac-
quisition function called predictive entropy search with constraints (PESC). Unlike EI based acquisition functions, PESC can handle decoupled cases where function and constraints can be evaluated independently.

### 2.6.3 Multi-objective Bayesian optimisation

Multi-objective optimisation (or Pareto optimisation) is an important area of optimisation that involves optimising simultaneously more than one objective functions. For example, in a car purchase situation, one wants to buy a car that has the lowest cost. However, cost is not the only objective since the lowest cost cars may not be comfortable. To make a decision, the buyer have to optimise both price and comfort of the car, making it a multi-objective optimisation problem.

In Feliot et al. (2017), the authors proposed a Bayesian approach for multi-objective optimisation problem with non-linear constraints. A variant of expectation improvement acquisition function is proposed to deal with problems where no feasible solutions are available at the beginning. Since the new expected improvement criterion has no closed-form expression, it is computed and optimised using sequential Monte Carlo sampling techniques.

Hernández-Lobato et al. (2016) presented a information-based acquisition function named PESMO for multi-objective Bayesian optimisation:

\[
\alpha_{PESMO}(x; D_t) = H(X^* | D_t) - \mathbb{E}_y [H(X^* | D_t \cup \{(x, y)\})],
\]

where \(X^*\) is the Pareto set, \(y\) is the output of all Gaussian process models at \(x\) and the expectation is taken with respect to the posterior distribution for \(y\) using these Gaussian process models. Based on the predictive entropy search criterion, at each iteration, PESMO evaluates the objective functions at the input location that is expected to maximally reduce the entropy of posterior estimate of Pareto set. PESMO criterion can be seen as a sum of \(k\) individual acquisition functions corresponding to \(k\) objectives. This allows identification of most promising objective by maximising the individual acquisition functions.

In another study, Shah and Ghahramani (2016) argued that it is important to model
correlations amongst objectives in multi-objective optimisation problems. A novel approximation is designed to overcome the intractability of integrals, which leads to an analytic and differentiable approximation to the expected increase in Pareto hypervolume acquisition function. In a related work, Calandra et al. (2014a) showed that they can approximate the real Pareto front better in presence of measurement noise by computing arbitrarily dense and continuous Pareto front.

### 2.6.4 Transfer learning

In many cases, Bayesian optimisation has the cold start problem, in which a lot of points with low function values may be recommended before a good region is found. This makes the optimisation process excruciatingly long (Joy et al., 2016a). An approach to this problem is to utilise the knowledge acquired from any previous function optimisations for the optimisation of a new related function (target) via transfer learning.

Bardenet et al. (2013) proposed the first algorithm on transfer learning in Bayesian optimisation. It transfers the source knowledge via a ranking function that was assumed to be applicable for the target function as well. In a similar approach, Yogatama and Mann (2014) proposed an algorithm that transfers knowledge from previous experiments using deviations from the mean. Feurer et al. (2015) proposed a method for Sequential Model-based Bayesian Optimisation (SMBO) using a meta-learning initialisation approach.

In Joy et al. (2016b, 2018), the authors proposed a flexible transfer learning framework for Bayesian optimisation that can leverage the knowledge from an already completed optimisation task for the optimisation of a new (target) task. The source data are modelled as noisy observations of the target function. The author also show the convergence properties of proposed method for two acquisition functions (Joy et al., 2018).

Since the relatedness of the source data to the target task is unknown a priori, Ramachandran et al. (2018) proposed an approach for optimal source selection for transfer learning in Bayesian optimisation. This method is theoretically guaranteed to select the most related source, helping to improve the optimisation efficiency.
In (Shilton et al., 2017), the authors derived regret bounds for two transfer learning algorithms in Bayesian optimisation. The first algorithm models the differences between the source and the target as a noise process (Joy et al., 2016a). The second one directly models the difference between the source and the target and corrects the source data to match the target function (Shilton et al., 2017). The regret bounds for the two transfer learning algorithms were shown to be tighter than the bounds for Bayesian optimisation without transfer learning.

### 2.7 Open problems relevant to practical applications

#### 2.7.1 Stability

Stability is an important aspect since unstable results can reduce users’ trust in the system. In real world problems, the end result of optimisation can be dramatically affected depending on the stability of the solution (e.g. the optimum is at narrow or wide peak). For example, in alloy design, the goal is to find the mixing proportion of a set of elements with the highest physical property (e.g. strength). However, due to the impurities in the raw material, it is impossible to mix the elements at the desired proportion. If the desired proportion is at a narrow peak then the performance of the alloy would not be stable when made repeatedly.

There have been some works for stability in machine learning (Bousquet and Elisseeff, 2002; Elisseeff et al., 2005; Liu et al., 2010). Nogueira et al. (2016) proposed a novel algorithm called unscented Bayesian optimisation that considers the uncertainty in the input space to discover safe optima. In chapter 3, we incorporate the stability information of a function into acquisition functions and propose a stable Bayesian optimisation algorithm to tackle the stability problem.
2.7. Open problems relevant to practical applications

2.7.2 Cascaded process optimisation

Multi-stage cascaded process are fairly common in manufacturing industry. A cascaded structure process consists of several stages, each stage’s output is the input for the next stage. The objective is to optimise the output quality at the last stage. The standard Bayesian optimisation approach does not take into account the cascaded structure and considers the problem as optimising a single black-box function.

In a recent work, Jenatton et al. (2017) proposed a Bayesian optimisation algorithm for the domain that exhibits a known dependency structure. By leveraging the structure, the proposed algorithm is able to explore the search space more efficiently and the posterior inference scales better with the number of observations (Jenatton et al., 2017). However, using Bayesian optimisation for cascaded structure processes is still an open problem. In chapter 4, we incorporate the cascaded structure to improve optimisation of cascaded processes. By using the extra information of cascaded processes, the number of dimensions is reduced, making our method outperforms the standard Bayesian optimisation approach.

2.7.3 Privacy

Privacy preserving data mining is an active research area. There has been many developed techniques to protect privacy of data such as perturbing data (Agrawal and Srikant, 2000), anonymisation (Sweeney, 2002), etc. Differential privacy (Dwork, 2006) recently has emerged as a strong privacy preserving framework that protects data from adversary given auxiliary information. Several machine learning and data mining models using this framework have been explored such as logistic regression (Chaudhuri and Monteleoni, 2009), Gaussian process (Smith et al., 2016), decision tree learning (Jagannathan et al., 2009) and matrix factorisation (Hua et al., 2015).

A differentially private Bayesian optimisation algorithm was proposed in (Kusner et al., 2015). In this work, the optimisation result is released privately with the assumption that the third-party optimiser are trusted. However, in many cases, the optimiser can not be trusted (adversary optimiser). In chapter 5, we propose a privacy preserving Bayesian optimisation algorithm that protects the optimisation
solution from an untrusted optimiser.

2.8 Privacy preserving machine learning and data mining

One of the focuses of this thesis is to address the privacy problem when applying Bayesian optimisation in real world situations. In this section, we review the literature of privacy preserving data mining and machine learning. We first discuss about data perturbation methods that directly add noise to the database. Next we review the literature of group anonymisation methods and methods for distributed data. We end this section by reviewing differential privacy - popular framework that provides strong theoretical privacy guarantee.

2.8.1 Data perturbation methods

Data perturbation is a class of techniques that directly adds noise to data and then applies machine learning algorithm to infer statistical information about data while privacy of records is maintained (Agrawal and Srikant, 2000; Agrawal and Aggarwal, 2001). It can be described as follows. Consider a dataset $\mathcal{D}_N = \{x_1, ..., x_N\}$ that contains sensitive information such as medical records, financial transactions and social security numbers. For each $x_i \in \mathcal{D}_N$, we add an amount of noise $\eta_i$ randomly chosen from a probability distribution $F(\eta)$. Thus, we have a new dataset $\mathcal{D}_N' = \{x'_1, ..., x'_N\}$ such that $x'_i = x_i + \eta_i$. To protect privacy of data, the amount of noise must be large enough so that attackers find it hard to infer original records from the perturbed ones. However, the distribution of the perturbed data should be similar to that of the original dataset so that the dataset remains useful for data mining purpose.

Early works One of early data perturbation methods is proposed in (Agrawal and Srikant, 2000). The authors proposed a new method to build a decision tree classifier from perturbed data. The sensitive information in data can be perturbed
by “value-class membership” method or “value distortion” method (Agrawal and Srikant, 2000). To quantify the level of privacy, they used the measure based on how close the original value can be estimated using the perturbed value. They also proposed a novel reconstruction procedure that estimates the original distribution of data. One drawback of this paper is that the reconstruction algorithm does not guarantee to provide a reasonable estimate of the original distribution. To address this issue, Agrawal and Aggarwal (2001) introduced expectation maximisation algorithm which is a more effective distribution reconstruction algorithm in terms of information loss. It works nearly as good as using the original data when the number of data points is large. The authors also provided a new metric to measure the privacy achieved. They proved that the proposed algorithm converges to the maximum likelihood estimate of the data distribution.

**Drawbacks**  In real world, public information can be easily obtained from many sources. These presented algorithms above don’t concern about the presence of public information which may lead to privacy breach. These algorithms also perform poorly on high dimensional data and are susceptible to outliers data or clusters (Aggarwal, 2007) in the data.

**Addressing drawbacks**  $k$-randomisation method was proposed which includes public information and tackle the issue of performance on high dimensional data (Aggarwal, 2007). Another direction is using Randomised Response Technique to build a decision tree classifier using disguised data (Du and Zhan, 2003).

One of the weaknesses of data perturbation method is that privacy can be compromised for a datapoint that has special attribute although privacy can be maintained on average. Evfimievski et al. (2003) introduced a new algorithm that does not require knowledge of the distribution of data. They defined a novel metric for privacy preserving using perturbation and analysed the amount of perturbation needed in order to avoid privacy breaches.
2.8.2 Group anonymisation methods

Although data perturbation methods have an advantage of being easily implementable at data collection period, they also have a disadvantage of finding it difficult to mask outliers and privacy can be breached when public information is available, especially in high dimensional cases (Aggarwal and Philip, 2008).

To address the privacy concern when public information is available, Samarati and Sweeney (1998) introduced the concept of $k$-anonymity. A database is considered to satisfy $k$-anonymity if for each record there is at least $k - 1$ other records identical to it. In order to achieve $k$-anonymity, generalisation and suppression technique are introduced (Samarati, 2001). Meyerson and Williams (2004) showed that optimal $k$-anonymisation is a NP-hard problem. In (Bayardo and Agrawal, 2005), the authors proposed $k$-optimise algorithm that can provide an effective solution for $k$-anonymity problem. Several works for $k$-anonymity problem has been done in (LeFevre et al., 2005, 2006; Fung et al., 2005; Wang et al., 2004; Domingo-Ferrer and Mateo-Sanz, 2002; Aggarwal et al., 2006).

Although $k$-anonymity is an effective and popular technique in protecting identification of records, it is vulnerable to some kind of attacks such as homogeneity attack and background knowledge attack (Machanavajjhala et al., 2007). One can overcome this issue by using $l$-diversity technique (Machanavajjhala et al., 2007) or $t$-closeness model (Li et al., 2007).

2.8.3 Methods for distributed data

In many real world situations, analysts want to aggregate information over several datasets while privacy of individuals in datasets is protected. Those datasets could be horizontally partitioned or vertically partitioned.

For horizontally partitioned datasets, different parties hold different sets of records which share the same set of attributes. There are many algorithms proposed for horizontal partitions. Following the method for ID3 algorithm proposed in (Lindell and Pinkas, 2000), there are several methods proposed for clustering (Inan et al., 2007;
Privacy preserving methods are proposed for decision tree and association rule mining over vertically partitioned data in (Du and Zhan, 2002) and (Vaidya and Clifton, 2002) respectively. Zhang et al. (2005) introduced a new perturbation method which uses randomised response with partial hiding to produce disguised data from original one. Private boosting on bipartite dataset and multiparty dataset are introduced in (Gambs et al., 2007).

### 2.8.4 Differential privacy

In real world, attackers can easily obtain external knowledge (often called auxiliary information) from many sources. With the presence of auxiliary information, traditional methods sometimes can not protect privacy. To tackle this issue, Cynthia Dwork proposed a new definition of privacy called differential privacy (Dwork, 2006; Dwork et al., 2006). This framework provides a strong guarantee of privacy and has become one of the most popular privacy preserving framework today. A differentially private mechanism protects the database even when an adversary knows all but one instance in the database. The authors introduced a “privacy budget” \( \epsilon \) which captures privacy requirement of data analysts.

**Definition 2.1.** A randomised function \( \mathcal{A} \) gives \( \epsilon \)--differential privacy (\( \epsilon > 0 \)) if for all neighbouring datasets \( D_1, D_2 \) differing on at most one entry, and all \( \mathcal{Y} \subseteq \text{Range}(\mathcal{A}) \),

\[
\frac{\mathbb{P}[\mathcal{A}(D_1) \in \mathcal{Y}]}{\mathbb{P}[\mathcal{A}(D_2) \in \mathcal{Y}]} \leq \exp(\epsilon).
\] (2.29)

From the definition, it is clear that smaller value of \( \epsilon \) provides stronger privacy. Because \( D_1 \) and \( D_2 \) can be switched interchangeably, the following also holds:

\[
\exp(-\epsilon) \leq \frac{\mathbb{P}[\mathcal{A}(D_1) \in \mathcal{Y}]}{\mathbb{P}[\mathcal{A}(D_2) \in \mathcal{Y}]} \leq \exp(\epsilon).
\]
2.8. Privacy preserving machine learning and data mining

When $\epsilon$ is small enough, $\exp(\epsilon) \approx 1 + \epsilon$. Thus we approximately have:

$$1 - \epsilon \lesssim \frac{\Pr[A(D_1) \in \mathcal{Y}]}{\Pr[A(D_2) \in \mathcal{Y}]} \lesssim 1 + \epsilon.$$ 

Mechanisms for achieving differential privacy

**Laplace mechanism and sensitivity** Laplace mechanism is designed for numerical queries, which is the most common form of queries. For a function $f : \mathcal{D}^m \to \mathbb{R}$, the sensitivity of $f$ is defined as

$$S(f) = \max_{D_1, D_2} \| f(D_1) - f(D_2) \|$$

for all $D_1, D_2 \in \mathcal{D}^m$ differing on at most one element.

The sensitivity allows us to find out how much noise needs to be added to hide the participation of an individual in the database. One method is to use a noise following Laplace distribution. The Laplace distribution with mean 0 and scale $b$ has the following probability density function:

$$\text{Lap}(\eta|b) = \frac{1}{2b} \exp\left(-\frac{|\eta|}{b}\right).$$

We will denote $X \sim \text{Lap}(b)$ for a random variable following Laplace distribution with scale $b$. The Laplace mechanism simply adds noise drawn from Laplace distribution to the output of function $f$. The scale of the distribution depends on the sensitivity of function $f$ and the required leakage parameter $\epsilon$.

**Definition 2.2.** Given any function $f : \mathcal{D}^m \to \mathbb{R}$, the Laplace mechanism is defined as:

$$\mathcal{M}_L(\eta, f(\eta), \epsilon) = f(\eta) + (Y_1, Y_2, \ldots, Y_k)$$

where $Y_1, Y_2, \ldots, Y_k$ are random variables drawn from $\text{Lap}(S(f)/\epsilon)$. 
Exponential mechanism  Exponential mechanism is the mechanism that can deal with both numerical and categorical data. Given some arbitrary range $\mathcal{R}$, the exponential mechanism uses a “utility function” $u : \mathbb{R}^N \times \mathcal{R} \rightarrow \mathbb{R}$ to measure the goodness of an output for a database. The global sensitivity of a mechanism is defined as below:

$$\Delta u = \max_{r \in \mathcal{R}, D_1, D_2 : \|D_1 - D_2\|_1 \leq 1} |u(D_1, r) - u(D_2, r)|.$$ 

Definition 2.3. The exponential mechanism randomly outputs element $r \in \mathcal{R}$ with the probability proportional to $\exp\left(\frac{\epsilon u(D, r)}{2\Delta u}\right)$.

Remarks  The differential privacy is currently the most popular framework, mainly due to its strong privacy guarantee for databases even with the presence of auxiliary information. However, using differential privacy often requires huge amount of noise to be added to the algorithm, reducing performance of algorithms seriously. Optimisation performance is crucial in Bayesian optimisation. To address the challenge of using Bayesian optimisation in a private manner, in chapter 5, we propose a new privacy preserving framework that slightly relaxes the privacy guarantee of differential privacy, but the required noise is considerably lower. This makes the private Bayesian optimisation algorithm using our framework have significantly high performance, while still maintaining a high level of privacy. Our framework is widely applicable to other machine learning algorithms. We demonstrate the applicability of the new framework by applying it to K-mean clustering algorithm and show that the proposed algorithm performance is significantly better than the differential privacy counterpart.

2.9 Summary

We have reviewed preliminary background and discussed works related to this thesis. We first introduced the importance of experimental design in research and development, followed by a brief overview of global optimisation approaches. We then focused on Bayesian optimisation, an efficient technique for finding global optimum of expensive black-box functions. It has many successful applications in robotics,
environmental monitoring, material design, etc. We further discussed about various choices of surrogate models and acquisition functions for Bayesian optimisation. We then reviewed active research areas in Bayesian optimisation, followed by introducing several challenges in using Bayesian optimisation for practical settings: finding stable solutions, incorporating cascade structure and privacy concerns. We will address these challenges in chapter 3, 4 and 5, respectively.
Chapter 3

Stable Bayesian optimisation

In previous chapters, we have introduced the aims of this thesis and reviewed the literature of Bayesian optimisation. In this chapter, we address the problem of finding stable solutions for Bayesian optimisation.

In many optimisation problems, finding stable solutions which are robust to the perturbation in input variables is important. For example, in hyperparameter tuning, the objective is to search for the set of hyperparameters that corresponds to the best model performance on a validation set. However, in some cases, especially when the availability of data is low, the function representing the model performance on the validation may contain several spurious sharp (narrow) peaks. These spurious peaks are the result of noisy classifiers from limited training data. They are thus generally sharper than the desired peak of where best generalisation performance is achieved. Standard Bayesian optimisation can converge to one of those undesirable peaks if appropriate safeguard is not instituted. Similar problem occurs when designing alloys, if the strength of the optimum alloy is highly sensitive to the perturbation, casting alloy may be hard and expensive. In this chapter, we develop theories and framework to build a safeguard against converging to such sharp (narrow) peaks, especially when a more stable peak is present. We construct two new acquisition functions that help Bayesian optimisation to avoid the convergence to the sharp peaks. We conduct a theoretical analysis and guarantee that Bayesian optimisation using the proposed acquisition functions prefers stable peaks over unstable ones. Experiments with synthetic function optimisation and hyperparameter tuning for
Bayesian optimisation is a technique to sequentially optimise expensive black-box functions in a sample efficient manner. Recently, it has emerged as a *de-facto* method to tune complex machine learning algorithms (Thornton et al., 2013). In tuning, the goal is to train a classifier at the right complexity so that it neither overfits, nor underfits. Performance on a validation set is used as an indicator of the fitting, and it is expected to peak at the hyperparameters corresponding to the right complexity. Thus, to tune a machine learning algorithm, Bayesian optimisation is employed in the pursuit of the peak validation set performance. However, in some situations, especially when training or validation dataset is small, spurious peaks appear along the performance surface (e.g. Fig 3.1). These peaks tend to be distributed randomly over low performance region. They are characteristically different from the peak corresponding to the right complexity in two ways a) they tend to be narrow and b) they vanish when tested on a large test data, whereas the right peak remains stable. Due to the latter difference, a Bayesian optimisation method that does not explicitly avoid these spurious peaks may converge to one of them and can result in a badly tuned system with inexplicably low performance during real world deployment. To the best of our knowledge, we are the first to identify and analyse this issue of spurious peaks and its serious downside.

Existence of multiple peaks with different widths along an optimisation surface is prevalent in many real world systems. For some of them, the end result of optimisation can get dramatically affected depending on whether the optimisation has converged to a wide peak or a narrow peak. For example, in alloy design (Xue et al., 2016), one of the main goals is to find the mixing proportion of a set of elements with the highest physical property (e.g. strength, ductility, etc.). However, alloy making is an imprecise process. Due to the impurities in the raw material, the elements can never be mixed at the exact proportion. Therefore, if the desired proportion is at a narrow peak then the performance of the alloy would not be stable as even a small difference in impurities could result in dramatic loss in performance. Hence, being able to avoid narrow peaks in favour of more stable peaks is a critical factor of suc-
Figure 3.1: Performance versus hyperparameters for a Support Vector Machine training as colour coded images: a) on a small validation set, and b) on a large test set. Spurious peaks of region 1 seen for the validation set vanish for the test set while the stable peak of region 2 still remains.

To address the issues with spurious peaks, we propose two new acquisition functions for Bayesian optimisation that actively seek stable peaks of the objective function. Based on our definition of stability, we show that it is possible to measure the stability of a peak by subjecting the underlying Gaussian process model with input perturbation. When faced with input perturbation, the predictive distribution of the Gaussian process changes. At any peak the mean of the distribution goes lower, and the variance goes higher. But more importantly, for two peaks of same height, the narrower peak will have lesser mean and more variance than the other peak. Further, we show that the variance can effectively be decomposed as a sum of two parts: a) epistemic variance due to the limited number of samples, and b) aleatoric variance arising from the interaction between the curvature of the function with the input perturbation. The narrower a peak is, the higher the aleatoric variance will be around that peak. Therefore, aleatoric variance can be used as an effective measure of the instability of a peak. Two acquisition functions are proposed in line with the GP-UCB and EI that while exploiting the usual combination of mean and variance also penalise for instability. Theoretically, we prove that under mild
assumptions, when two peaks are of same height, the proposed acquisition functions would always favour the more stable peak. We compare our method with a standard Bayesian optimisation implementation on both synthetic function optimisation and real-world hyperparameter tuning. On synthetic function optimisation, we create a function that has both stable peaks and spurious peaks. Experiments with synthetic function show that our proposed method converges to stable peaks more often than the baseline. For real world application, we demonstrate tuning the hyperparameters of Support Vector Machine on two real world datasets. Experimental results clearly demonstrate that our proposed method converges to a stable peak whereas the standard Bayesian optimisation converges to an unstable peak, and hence the SVMs tuned by our method perform better on test sets.

3.2 The proposed framework

We present two new acquisition functions for Bayesian optimisation designed to maximise a black-box function with behaviour that the maxima from stable regions are preferred over the maxima from relatively unstable regions. We first discuss the notion of stability, then describe how a Gaussian process model gets modified in presence of any perturbation in the input variables. Next, we use the predictive distribution of the modified Gaussian process to formulate two novel acquisition functions: STABLE-UCB and STABLE-EI. We theoretically analyse the proposed acquisition functions and prove that they are guaranteed to take higher values in more stable regions and thus a Bayesian optimisation using these acquisition functions will have higher tendency to sample from more stable regions. Finally, we present an algorithm summarising the proposed stable Bayesian optimisation.

3.2.1 Stability of Gaussian process prediction

Given a set of observed data \( D_t = \{x_i, y_i\}_{i=1}^t \) where \( x_i \in \mathbb{R}^d \) and \( y_i = f(x_i) + \epsilon \), we use a Gaussian process to model the function \( f \). Using \( D_t \), for a new input \( x \), the
3.2. The proposed framework

The proposed framework of the corresponding output $f(x)$ is given as

$$P(f(x)|D_t, x) = \mathcal{N}(\mu_t(x), \sigma^2_t(x)),$$

where the predictive mean $\mu_t(x) = k^T K^{-1} y$ (see Eq. (2.9)) and the predictive variance $\sigma^2_t(x) = k(x, x) - k^T K^{-1} k$ (see Eq. (2.10)) with a notation $y = y_{1:t}$. We define $\beta = K^{-1} y$ to be used later.

The above predictive mean and variance are instrumental to Bayesian optimisation as they provide a way to estimate the function value at any point in the function support along with the model’s uncertainty. The model uncertainty, also called “epistemic uncertainty”, is used in the Bayesian optimisation to express our belief in the estimation and guides efficient exploration of the function while keeping a balance on exploitation. This strategy is an instance of a general concept in reinforcement learning known as exploitation-exploration trade-off.

Since our goal is to develop a stable Bayesian optimisation framework that prefers solutions insensitive to small perturbations in input, we start from asking the question how does the predictive mean and variance of the function value change if an input is slightly perturbed. A large shift in the mean and/or a large increase in the variance indicates a fast varying function and can be used to detect the unstable regions. If a test input is corrupted by a Gaussian noise, $\epsilon_x \sim \mathcal{N}(0, \Sigma_x)$ such that $u = x + \epsilon_x$, the predictive distribution after marginalised over the input corruption can be given as

$$p(y|D_t, x, \Sigma_x) = \int p(y|D_t, u)p(u|x, \Sigma_x)d\mathbf{u}. \quad (3.1)$$

This distribution, in general, is non-Gaussian. However, in (Girard and Murray-Smith, 2005), it is shown that a fairly close Gaussian approximation can be obtained. Let us use $\mu_t(x)$ and $\sigma^2_t(x)$ to denote the mean and variance of the Gaussian predictive distribution $p(y|D_t, x)$ in the perturbation-free case. Also use $m_t(x, \Sigma_x)$ and $v_t(x, \Sigma_x)$ to denote the mean and variance of predictive distribution $p(y|D_t, x, \Sigma_x)$, then with the Gaussian approximation, we can write

$$p(y|D_t, x, \Sigma_x) \approx \mathcal{N}(m_t(x, \Sigma_x), v_t(x, \Sigma_x)). \quad (3.2)$$

Computation of the predictive mean and variance in (3.2) may also become intract-
3.2. The proposed framework

able when using an arbitrary covariance function. Fortunately, they are tractable for popular covariance functions such as linear and square exponential. We demonstrate our framework using squared exponential covariance function. Nonetheless, our framework remains amenable to any valid covariance function and appropriate approximations arising due to an arbitrary covariance function can be easily incorporated. For the squared exponential covariance function, the predictive mean and variance are given as (Girard and Murray-Smith, 2005)

\[
m_t(x, \Sigma_x) = \sum_{i=1}^{t} \beta_i k(x, x_i)k_1(x, x_i) \tag{3.3}
\]

\[
u_t(x, \Sigma_x) = \sigma_n^2(x) + \sigma_{t,a}^2(x, \Sigma_x) \tag{3.4}
\]

where \(\sigma_t^2(x)\) is the variance as in the unperturbed case and the extra variance due to perturbation is given as

\[
\sigma_{t,a}^2(x, \Sigma_x) = \sum_{i,j=1}^{t} K_{ij}^{-1} k(x, x_i)k(x, x_j)(1 - k_2(x, \bar{x}_{ij}))) + \sum_{i,j=1}^{t} \beta_i \beta_j k(x, x_i)k(x, x_j)(k_2(x, \bar{x}_{ij}) - k_1(x, x_i)k_1(x, x_j)). \tag{3.5}
\]

In the above expression, we have used the definitions:

\[
k_1(x, x_i) = Q_{x|x}(W)\exp \left[ \frac{1}{2}(x - x_i)^T S_{x|x}(W)(x - x_i) \right]
\]

\[
k_2(x, \bar{x}_{ij}) = Q_{x|x}(W)\frac{W}{2}\exp \left[ \frac{(x - \bar{x}_{ij})^T S_{x|x}(W)(x - \bar{x}_{ij})}{2} \right]
\]

where

\[
\bar{x}_{ij} = \frac{x_i + x_j}{2},
\]

\[
Q_{x|x}(W) = \left| I + W^{-1}\Sigma_x \right|^{-1/2},
\]

\[
S_{x|x}(W) = W^{-1}(W^{-1} + \Sigma_x^{-1})^{-1}W^{-1},
\]
3.2. The proposed framework

and $W$ is defined in Chapter 2 (see Eq. (2.11)).

We compare the Gaussian approximation to the Monte-Carlo based approximation of the predictive distribution for noisy inputs. For the Monte-Carlo approach, we draw 1000 samples to approximate the true predictive distribution (we have verified that by adding further samples the change in the distribution is negligible). Fig. 3.2 shows the predictive distributions of Gaussian approximation and Monte-Carlo approximation for three noisy inputs with respect to different level of perturbations $\Sigma_x$. As seen from the figure, the predictive distributions of Gaussian approximation are comparable to those of Monte-Carlo approximation. For small amount of perturbation, e.g. $\Sigma_x = 0.01$, the two approximations are similar for all three noisy inputs. As the noise level increases, Gaussian approximation starts to differentiate from Monte-Carlo based approximation but still remain close. For high level of noise (e.g. $\Sigma_x = 0.05$), the two approximations are comparable for inputs in stable region (e.g. $x = -0.25, 0.25$) and slightly different for the input in unstable region (e.g. $x = 0.65$).

In the following, we utilise the above analysis to define two novel acquisition functions to propose a stable Bayesian optimisation framework.

### 3.2.2 Stable Bayesian optimisation

**Algorithm 3.1** The proposed stable Bayesian optimisation.

1: **Input:**
2: Initial observation set $D_{t_0} = \{x_{1:t_0}, y_{1:t_0}\}$.
3: Bounds for the search space $X$.
4: **Output:** $\{x_t, y_t\}_{t=1}^T$
5: **for** $t = t_{0+1}, \ldots, T$
6: Find optimiser $x_{t+1}$ of acquisition function (3.6) or (3.7), over $x$.
7: Evaluate the target function as $y_{t+1} = f(x_{t+1}) + \epsilon$.
8: Augment the observation set: $D_t = D_t \cup \{x_{t+1}, y_{t+1}\}$ and update the GP $\mathcal{I}_t$.
9: **end for**

Having a closed form expression for the predictive mean and variance as in (3.3) and (3.4) provides us the required tractability to formulate an acquisition function for “stable” Bayesian optimisation. In the expression for predictive variance in (3.4), we note that the variance $\nu_t(x, \Sigma_x)$ has two components: (1) the *epistemic* variance
3.2. The proposed framework

Figure 3.2: Predictive distributions of Gaussian approximation and Monte-Carlo approximation with respect to different values of noise levels: a) $\Sigma_x = 0.01$, b) $\Sigma_x = 0.02$, c) $\Sigma_x = 0.05$. In practice, parameter settings are limited to 1% perturbation and our approximation based on Gaussian distribution can handle it well.
(uncertainty) term $\sigma_t^2(x)$, arising due to our lack of understanding about the function value, mainly due to finite set of observations and (2) the *aleatoric* variance term $\sigma_{t,a}^2(x, \Sigma_x)$ (further detailed in (3.5)), arising due to the inherent variation in the function around $x$. We associate the notion of the stability to this aleatoric variance which takes higher values in regions where the function has rate of change. In the remainder of this section, we use this property to define two new acquisition functions that yields a stable Bayesian optimisation. The algorithm results in a solution where the function value is robust to small perturbations.

**STABLE-UCB acquisition function:**

Denoting the epistemic and aleatoric variances at time $t$ by $\sigma_t^2(x)$ and $\sigma_{t,a}^2(x, \Sigma_x)$ respectively, we define the STABLE-UCB acquisition function as:

$$ a_t(x, \Sigma_x) = m_t(x, \Sigma_x) + \kappa_t \sigma_t(x, \Sigma_x) - \lambda \sigma_{t,a}(x, \Sigma_x) \tag{3.6} $$

where $\kappa_t$ is a $t$-dependent weight that sets a balance between exploitation and exploration, and $\lambda > 0$ is a fixed weight that sets our penalty on the instability. In the above formulation, our intuition is to *penalise* the points where the function is varying fast with small change in $x$. In our implementation, to balance between epistemic and aleatoric variance, we set $\lambda$ equals to $\kappa_t$.

**STABLE-EI acquisition function:**

We also propose a new acquisition function bases on improvement. This is because improvement based acquisition function is extremely popular among practitioners. Similar to the STABLE-UCB acquisition function, we incorporate the aleatoric variance $\sigma_{t,a}^2(x, \Sigma_x)$ at time $t$ to improvement function:

$$ I_t(x, \Sigma_x) = \max \left\{0, f_{i+1}(x) - \omega \sigma_{t,a}(x, \Sigma_x) - f(x^+)\right\} $$

where $x^+ = \arg\max_i f(x_i)$ and $\omega$ is a weight that penalises points that in unstable region. In this formulation, our idea is to make the stable region have higher improvement value compared to the unstable region with the same level of predictive
3.2. The proposed framework

mean. Using the new definition of improvement, the Stable Expected Improvement acquisition function (STABLE-EI) can be computed as:

\[
b_t(x, \Sigma_x) = \mathbb{E}[I(x, \Sigma_x)] = \int_0^\infty \frac{I}{\sqrt{2\pi}\sigma_t(x, \Sigma_x)} \times \exp \left[ -\frac{(I - m_t(x, \Sigma_x) + \omega\sigma_{t,a}(x, \Sigma_x) + f(x^+))^2}{2\sigma_t^2(x, \Sigma_x)} \right] \, dI
\]

This function can be analytically evaluated as:

\[
b_t(x, \Sigma_x) = \begin{cases} 
\sigma_t(x, \Sigma_x)(z_t\Phi(z_t) + \phi(z_t)) & \text{if } \sigma_t(x, \Sigma_x) > 0 \\
0 & \text{if } \sigma_t(x, \Sigma_x) = 0 
\end{cases}
\]

(3.7)

where \(z_t = \frac{[m_t(x, \Sigma_x) - \omega\sigma_{t,a}(x, \Sigma_x) - f(x^+)]}{\sigma_t(x, \Sigma_x)}\).

Stable Bayesian optimisation maximises the acquisition function \(a_t(x, \Sigma_x)\) or \(b_t(x, \Sigma_x)\) to suggest the next function evaluation at each iteration. A step-by-step procedure of stable Bayesian optimisation is provided in Algorithm 3.1.

Theoretical analysis:

In this section, we analyse the proposed acquisition functions to provide theoretical guarantees that the acquisition function \(a_t(x, \Sigma_x)\) and \(b_t(x, \Sigma_x)\) indeed prefer less sharper peaks of the function \(f(x)\).

Definition 3.1. (Identical data topology): Any two points \(x, x'\) are said to have identical data topology if for each observation \(x_i\), there exists another observation \(x_i'\) such that \(||x - x_i|| = ||x' - x_i'||\).

A consequence of identical data topology is that for points \(x, x'\), any distance based kernels induce Gram matrices that are equal up to a permutation. With increasing set of observations, it is not difficult to achieve identical data topology approximately. In the following lemma and theorems, to have no favour to any peak, let us assume that there are sufficiently many observations around both \(x, x'\) so that the two points have identical data topology.
Lemma 3.1. If $x$, $x'$ are the two highest peaks in the support of function $f$ such that $|f(x) - f(x')| < \eta_0$ for small $\eta_0$, and $f$ locally varies faster around $x'$ compared to $x$ in a small $h_0$-neighbourhood, i.e. $|f(x + h) - f(x)| < f(x' + h) - f(x')| < 1$, $\forall h \in (-h_0, h_0)$, under certain mild assumptions, the following relations hold true:

$$m_t(x, \Sigma_x) \geq m_t(x', \Sigma_x)$$
$$\sigma_t(x, \Sigma_x) = \sigma_t(x', \Sigma_x)$$
$$\sigma_{t,a}(x, \Sigma_x) \leq \sigma_{t,a}(x', \Sigma_x).$$

Proof. To have no favour to any peak, let us assume that there are sufficiently many observations around both $x$, $x'$ so that the two points have identical data topology. Due to the mild assumption, we have a pair of observations $x_i$ and $x_{i'}$ such that $||x - x_i|| = ||x' - x_{i'}||$. This implies that the covariance values $k(x, x_i) = k(x', x_{i'})$ and $k_1(x, x_i) = k_1(x', x_{i'})$. By definition, $\beta = K^{-1}y$. Since the peak at $x'$ is sharper than the peak at $x$, meaning $y_{i'} \leq y_i$ and therefore $\beta_{i'} \leq \beta_i$. Hence, $\sum_{i=1}^{t} \beta_i k(x, x_i) k_1(x, x_i) \leq \sum_{i=1}^{t'} \beta_{i'} k(x', x_{i'}) k_1(x', x_{i'})$, or $m_t(x, \Sigma_x) \geq m_t(x', \Sigma_x)$.

Next, we also note that due to the identical data topology assumption around both peaks, we have equal epistemic uncertainties, i.e. $\sigma_t(x, \Sigma_x) = \sigma_t(x', \Sigma_x)$ by definition of $\sigma_t(x)$ in (2.10).

Finally, we show that $\sigma_{t,a}(x, \Sigma_x) \leq \sigma_{t,a}(x', \Sigma_x)$. For this, consider the aleatoric variance term in (3.5). As above, we have the following relations: $k(x, x_i) = k(x', x_{i'})$, $k_1(x, x_i) = k_1(x', x_{i'})$, $\beta_{i'} \leq \beta_i$ and additionally, $k_2(x, x_{i'}) = k_2(x', x_{i''})$. Using these relations, it is straightforward to show that $\sigma_{t,a}(x, \Sigma_x) \leq \sigma_{t,a}(x', \Sigma_x)$. \hfill $\square$

Next, we state and prove our key results for the newly proposed acquisition functions.

Theorem 3.1. (STABLE-UCB case): If $x$, $x'$ are the two highest peaks in the support of function $f$ such that $|f(x) - f(x')| < \eta_0$ for small $\eta_0$, and $f$ locally varies faster around $x'$ compared to $x$ in a small $h_0$-neighbourhood, i.e. $|f(x + h) - f(x)| < f(x' + h) - f(x')| < 1$, $\forall h \in (-h_0, h_0)$, the acquisition function in (3.6) satisfies the relation: $a_t(x, \Sigma_x) \geq a_t(x', \Sigma_x)$ under certain mild assumptions.

Proof. Let us assume that $x$ and $x'$ have identical data topology. Consider the difference between the acquisition function values at $x$, $x'$:
\[
\Delta a_t = [m_t(x, \Sigma_x) - m_t(x', \Sigma_x)] \\
+ [\kappa_t (\sigma_t(x, \Sigma_x) - \sigma_t(x', \Sigma_x))] \\
- [\lambda (\sigma_{t,a}(x, \Sigma_x) - \sigma_{t,a}(x', \Sigma_x))].
\]

Using the three separate inequalities from Lemma 3.1, we can prove that \(\Delta a_t \geq 0\), i.e. \(a_t(x, \Sigma_x) \geq a_t(x', \Sigma_x)\). \(\square\)

**Theorem 3.2.** *(STABLE-EI case):* If \(x, x'\) are the two highest peaks in the support of function \(f\) such that \(|f(x) - f(x')| < \eta_0\) for small \(\eta_0\), and \(f\) locally varies faster around \(x'\) compared to \(x\) in a small \(h_0\)-neighbourhood, i.e. \(\frac{|f(x+h) - f(x')|}{f(x+h) - f(x')} < 1\), \(\forall h \in (-h_0, h_0)\), the acquisition function in (3.7) satisfies the relation: \(b_t(x, \Sigma_x) \geq b_t(x', \Sigma_x)\) under certain mild assumptions.

**Proof.** Let us assume that \(x\) and \(x'\) have identical data topology. We define the difference between the acquisition function values at \(x\), \(x'\) as:

\[
\Delta b_t = \sigma_t(x, \Sigma_x) (z_t \Phi(z_t) + \phi(z_t)) \\
- \sigma_t(x', \Sigma_x) (z'_t \Phi(z'_t) + \phi(z'_t)).
\]

Our aim is to show that \(\Delta b_t \geq 0\), i.e. \(b_t(x, \Sigma_x) \geq b_t(x', \Sigma_x)\). From the definition of \(z_t\) in (3.7) and Lemma 3.1, we have

\[
z_t - z'_t = [m_t(x, \Sigma_x) - m_t(x', \Sigma_x)] \\
- \omega (\sigma_{t,a}(x, \Sigma_x) - \sigma_{t,a}(x', \Sigma_x))/\sigma_t(x, \Sigma_x) \geq 0.
\]

Let \(\tau(z) = z \Phi(z) + \phi(z)\). Since \(\tau(z)\) is non-decreasing, we have \(\tau(z_t) \geq \tau(z'_t)\). On the other hand, from the Lemma 3.1 we have equal epistemic uncertainties at \(x\) and \(x'\), i.e. \(\sigma_t(x, \Sigma_x) = \sigma_t(x', \Sigma_x)\). Therefore, we have \(\sigma_t(x, \Sigma_x) \tau(z_t) \geq \sigma_t(x', \Sigma_x) \tau(z'_t)\), or in other words \(b_t(x, \Sigma_x) \geq b_t(x', \Sigma_x)\). \(\square\)

**Remarks:** The above Theorem 3.1 and 3.2 cover an important case that when the peaks in both stable and unstable regions are approximately equal in height, a
Bayesian optimisation algorithm using the acquisition functions in (3.6) and (3.7) will prefer the peak from the stable region. In the case, when a peak of unstable region is higher than the peak of stable region, the two terms $m_t(x, \Sigma_x) - m_t(x', \Sigma_x)$ and $\sigma_{t,a}(x, \Sigma_x) - \sigma_{t,a}(x', \Sigma_x)$ would be acting against each other and their net difference will decide whether the algorithm suggests the point from the stable region or unstable region. Since the parameter $\lambda$ and $\omega$ are user specified, there exists sufficiently large values of them that always guarantee the suggestion of stable peak. In the case, when a peak of unstable region is lower than the peak of stable region, both standard and the stable Bayesian optimisation will select the stable peak.

**Computational complexity**  Since the difference between our stable Bayesian optimisation algorithm and the standard one is the acquisition function, we will focus our attention on the complexity analysis of acquisition function computation. In the standard Bayesian optimisation algorithm, the complexity of UCB and EI for $T$ observed datapoints is $O(T^3)$. In our proposed acquisition functions (3.6) and (3.7), the complexity of computing mean $m_T(x, \Sigma_x)$, epistemic variance $\sigma_T(x, \Sigma_x)$ and aleatoric variance $\sigma_{T,a}(x, \Sigma_x)$ for $T$ observations are all $O(T^3)$. Therefore, our proposed algorithm has the same computational complexity as the standard Bayesian optimisation algorithm.

### 3.3 Experiments

In this section, we experiment on a set of synthetic and real datasets to demonstrate the efficacy of our stable Bayesian optimisation. Experiments with synthetic dataset show the behaviour of our proposed method with a known and complex function with multiple sharp peaks and one stable peak. We also conduct experiments with several hyperparameter tuning problems to show the utility of our method for real world applications.
3.3.1 Baseline method and evaluation measures

We compare the stable Bayesian optimisation using proposed acquisition functions (STABLE-UCB and STABLE-EI) with standard Bayesian optimisation using UCB acquisition function (BO-UCB) and EI acquisition function (BO-EI), respectively. On synthetic data, we compare STABLE-UCB and STABLE-EI with the corresponding baseline in two aspects: ‘the maximum value found’ and ‘the number of times an algorithm visits around the highest stable peak’ with respect to number of iterations. On real data, we show the performance of stable Bayesian optimisation and standard Bayesian optimisation on both validation and test sets. We compare STABLE-UCB with standard UCB and STABLE-EI with standard EI for fair comparison.

3.3.2 Experiments with synthetic function

Data generation:

The synthetic function $f(x)$ is generated using a squared exponential kernel with two different parameters (See Fig 3.3a). The stable region is created using squared exponential kernel with length scale 0.2. The unstable region is generated using a squared exponential kernel with length scale 0.01 to simulate spurious peaks. The unstable region of $f(x)$ is $0.7 \leq x \leq 1.1$ and the rest is the stable region.

Experimental results:

We randomly initialise 2 observations for Bayesian optimisation. Fig 3.3b illustrates the value of STABLE-UCB acquisition function and aleatoric variance after 30 iterations. In the unstable region, the STABLE-UCB acquisition function used for stable Bayesian optimisation has smaller value than that in the stable region due to high aleatoric variance capturing instability. We observe similar results for the STABLE-EI acquisition function.

\footnote{The highest stable peak region is $0 \leq x \leq 0.125.$}
3.3. Experiments

Figure 3.3: a) The synthetic function with one stable peak and multiple spurious peaks b) The STABLE-UCB acquisition function and aleatoric variance after 30 iterations.

Figure 3.4: Performance of stable Bayesian optimisation and standard Bayesian optimisation with respect to number of iterations on Synthetic function. a) and c) shows that STABLE-UCB and STABLE-EI converge to 2.3 and 2.5 (stable peak) whereas BO-UCB and BO-EI converge to 3.7 (spurious peak). b) and d) shows that stable Bayesian optimisation reaches stable peak more often than the baseline.
Fig 3.4 depicts the result of the optimisation comparing the standard BO with the proposed stable variants. Fig 3.4a shows the result of ‘maximum value found’ using STABLE-UCB averaged over 50 different initialisations. After 50 iterations, the proposed STABLE-UCB converges to averaged maximum value at around 2.3 while BO-UCB converges to 3.7. This is because STABLE-UCB often converges to stable region unlike BO-UCB which converges to unstable region. The number of peaks visited in the stable region by STABLE-UCB and BO-UCB are compared in Fig 3.4b. In 20 iterations, the percentage of times STABLE-UCB and BO-UCB visit around the highest stable peak are 96% and 34%, respectively. In 60 iterations, more than 84% of times the proposed STABLE-UCB visits around the highest stable peak, whereas this number for BO-UCB is only at 6%, illustrating better stability behaviour of STABLE-UCB. The pattern is similar for STABLE-EI. Fig 3.4c illustrates the ‘maximum value found’ result using STABLE-EI. After 50 iterations, STABLE-EI converges to 2.5 while BO-EI converges to 3.7. After 60 iterations, BO-EI can not find the stable peak while STABLE-EI reaches the stable peak around 60% of times (Fig 3.4d).

### 3.3.3 Experiments with hyperparameter tuning problems

#### Dataset:

We use Letter and Glass classification dataset from UCI machine learning repository\(^2\). Letter dataset contains 20,000 datapoints about the image characteristic of 26 capital letters in the English alphabet. Since spurious peaks occur mostly when the training set or the validation set has limited number of datapoints, we sample only 200 datapoints from the Letter dataset. Glass dataset consists of 214 datapoints represented using 10-features related to glass properties. Both datasets are divided into training set, validation set and test set. The validation set accuracy will be used as the objective for optimisation and the test set accuracy will be used as the performance measure.

\(^2\)http://archive.ics.uci.edu/ml
3.3. Experiments

Figure 3.5: Sampling behaviour of both STABLE-UCB and BO-UCB for hyperparameter tuning of SVM for Letter classification a) on validation dataset and b) on test dataset. The background portrays the performance function with respect to the hyperparameters. Spurious peaks (region 1) is evident for the validation dataset but vanished for the test set while stable region (region 2) still remains.

Experimental results with Support Vector Machine:

Support Vector Machine (SVM) is a popular machine learning algorithm for classification problem. Two main hyperparameters in SVM using RBF kernel are $C$ and $\gamma$ that represent the misclassification trade-off and the RBF kernel parameter respectively. We apply both stable Bayesian optimisation and standard Bayesian optimisation for tuning $C$ and $\gamma$. In the experiments, the objective function $f(x)$ is the validation set accuracy and vector $x$ represents the hyperparameters $C$ and $\gamma$. Performance on test set is used to compare the performance of the proposed method and the baseline.

Fig 3.5 shows the converged peaks by our proposed STABLE-UCB and BO-UCB over 30 different initialisation. As seen from the figure, the number of times BO-UCB converges to spurious peaks is considerably higher than that of STABLE-UCB. This behaviour leads to the accuracy performance shown in Fig 3.6. Fig 3.6a shows the performance of two STABLE-UCB and BO-UCB on validation set. We note that this is a multi-class classification task, hence a random classifier would have a mean accuracy of only $1/26=0.0385$. After 120 iterations, STABLE-UCB’s best accuracy on the validation set is 0.35 whereas BO-UCB’s best is 0.375. However,
3.3. Experiments

Figure 3.6: Performance of stable Bayesian optimisation and standard Bayesian optimisation using SVM with respect to number of iterations on Letter dataset. The performance of standard Bayesian optimisation, due to convergence to spurious peaks on the validation set (Fig 3.6a and Fig 3.6c), results in poor performance for the test set (Fig 3.6b and Fig 3.6d).

as we move to the test set and compare the performance of the two methods using the hyperparameters optimised using the validation set, we find that STABLE-UCB performance is higher compared to BO-UCB (see Fig 3.6b). After 120 iterations, STABLE-UCB performance remains high at 0.44 whereas BO-UCB reaches only up to 0.41. Fig 3.6c shows the performance of STABLE-EI and BO-EI on validation set of Letter dataset. After 140 iterations, STABLE-EI reaches 0.35 whereas BO-EI reaches 0.36. However, on the test set, the performance of STABLE-EI remains at 0.43 whereas BO-EI’s best accuracy is 0.41 after 140 iterations (see Fig 3.6d).

We observed a similar behaviour of two algorithms for Glass dataset (Fig 3.7). On the validation set, although BO-UCB and BO-EI converge to a higher accuracy (both have accuracy = 0.58) than that of STABLE-UCB and STABLE-EI (both have accuracy = 0.56), stable Bayesian optimisation accuracy score stays above 0.56 compared to standard Bayesian optimisation’s accuracy of under 0.52 for the test set.
3.3. Experiments

Figure 3.7: Performance of stable Bayesian optimisation and standard Bayesian optimisation using SVM with respect to number of iterations on Glass dataset. The performance of standard Bayesian optimisation, due to convergence to spurious peaks on the validation set (Fig 3.7a and Fig 3.7c), results in poor performance for the test set (Fig 3.7b and Fig 3.7d).
Our experiments with SVM hyperparameter tuning demonstrate that spurious peaks are indeed abound in case of small training and validation sets. The proposed stable Bayesian optimisation is able to successfully reduce the convergence to such peaks.

3.4 Summary

We proposed a stable Bayesian optimisation framework to find stable solutions for Bayesian optimisation. We discussed the notion of stability and presented a modified Gaussian process model in presence of noisy inputs. We constructed two novel acquisition functions based on the epistemic and aleatoric variances of the modified Gaussian process estimates. The aleatoric variance becomes high in unstable region around spurious narrow peaks and thus offers a way to guide the function optimisation towards stable regions. We theoretically showed that our proposed acquisition functions favour peaks at stable regions over unstable ones. Through experiments with both synthetic function optimisation and hyperparameter tuning for SVM classifier, we demonstrated the utility of our proposed framework.
Chapter 4

Bayesian optimisation for cascaded processes

In the previous chapter, we addressed the problem of finding stable solutions for Bayesian optimisation. In this chapter, we focus on multi-stage cascaded process optimisation. Multi-stage cascaded processes are fairly common, especially in manufacturing industry. In each stage of a multi-stage cascaded process, the input is transformed under the condition set by control parameters. The output then becomes the input for the next stage. Setting the right control parameters at each stage is important to achieve high quality products at low cost. We extend the standard Bayesian optimisation approach to the cascaded process through formulating a series of optimisation problems that are solved sequentially from the final stage to the first stage. We analyse the convergence property and show the convergence rate of the new proposed method. Experiments performed on a simulated testbed of Al-Sc heat treatment and a data analytic pipeline showed a considerable efficiency gain over the standard Bayesian optimisation approach, that ignores the cascaded structure.
4.1 Introduction

Cascaded processes abound in real life. As an example, consider the strength of materials produced by cascaded heat treatment processes. From planes and cars to skyscrapers, there is a critical dependence on industrial alloys with specific properties such as strength, creep resistance, weldability and so on. Heat treatments are applied to alloys to achieve such target properties and involve a cascade of steps, where the temperature is maintained for several hours in one stage before proceeding to a new temperature for the next step. This “cascade” matches the underlying physical processes, effectively allowing the earlier stages of lower temperature to determine the nucleation (the number of nuclei), and then “growth” or “coarsening” that dominates later stages and eventually controls final hardness. The temperature and duration of each stage influence the final alloy properties. In general, most of the industrial manufacturing processes are actually cascade of many different stages (see Fig. 4.1 for a diagram of a cascaded process). Precursors or raw materials are transformed at each stage before being used as the input to the next stage. The parameters of each stage influence the final product quality, and cost. Searching for the right parameters that result in the highest quality product with the lowest cost can be a time-consuming process. Therefore, it is important to find them in as minimum number of trials as possible.

The word “cascade” and “multistage” have been used in prior literature in many different contexts - it can mean “iteration” in a Gaussian process, it can mean “multiple stages” where the outputs of previous stages are coupled. Very early work in engineering design methodology refers to the term multistage Bayesian surrogates - the multistage here refers to consecutive iterations of a batch Gaussian process (Osio
and Amon, 1996). Wang et al. (2015) proposed multi-stage hyperparameter optimisation using Bayesian optimisation that splits the data - each stage merely takes in increasing amounts of data. Multistage Gaussian process (Quinonero-Candela et al., 2002; Candela et al., 2003) cascades Gaussian processes to do multi-step ahead prediction. This is done by transmitting the mean and covariance of the previous GP to the next stage. This model differs from ours in a fundamental way: it does not allow new inputs at different cascaded stages which is critical in experimental and industrial processes as each stage may perform a different function. Our definition of cascaded Bayesian optimisation stems from industrial cascaded processes. Each intermediate stage must allow not only the output of the last stage to be coupled, but also allow the control parameters for this stage to be input. The target quality is the output of the final stage of the cascade. Intermediate stage measurements can be made in such processes. The idea that Bayesian optimisation can be applied to such cascaded experimental processes is new. To the best of our knowledge, no current solutions exist to incorporate the structure of cascade into Bayesian optimisation frameworks.

To take advantage of the structure in cascaded processes, we propose a cascade Bayesian optimisation framework. Each of the cascaded stage is modelled by an independent function through a separate Gaussian process. We assume that output product quality of each stage is measurable. Since we need to maximise the quality of the product from the final stage, we find the control parameters of the final stage and desired input material quality to the final stage by maximising an acquisition function based on only the final stage Gaussian process posterior. The stage feeding to the final stage now thus only needs to supply with its desired input material quality. This is also the case for all other intermediate stages. They need to produce the desired material quality feeding the next stage by controlling its parameters and asking for a desired input material quality (see Fig. 4.1 for a diagram of a cascaded process). We formulate a novel optimisation problem to find the desired input quality and the control parameters of the intermediate stages through the inversion of the Gaussian process posterior, exploiting also the epistemic uncertainties in the model. A sequence of optimisation problems is solved starting from the penultimate stage to the first stage. At the end of the cascade Bayesian optimisation procedure we obtain the control parameter values of all the stages that need to be set for the next trial. Additionally, we can introduce costs associated with the control
parameters into our optimisation formulation to discover cost-efficient solutions.

We validate our algorithm on synthetic data, tuning data analytic pipelines and alloy heat treatment optimisation. In synthetic data, we show that the algorithm is effective under diverse simulated conditions. In the first experiment, we fix the number of parameters per stage to 4 and vary the number of stages from 4 to 8. In the second experiment, the number of stages is kept fixed at 5, however, the number of parameters per stage is varied from 3 to 7. We compare our cascade Bayesian optimisation algorithm with the standard Bayesian optimisation approach that ignores the cascaded structure which collapses all the control parameters into a joint optimisation problem. We call this baseline method as joint Bayesian optimisation. We show that the cascade Bayesian optimisation is particularly effective when the number of stages is large or the number of parameters per stage is high. In our extreme example where the number of stages is 5 and the number of parameters per stage is 7, cascade Bayesian optimisation is able to perform 11 times better than the joint Bayesian optimisation.

For the experiments with data analytic pipelines, we illustrate that the algorithm is effective in tuning hyperparameters of machine learning systems. We consider a pipeline that consists of 2 stages: feature preprocessing and classification. In the first stage, we choose kernel principal component analysis with polynomial kernel as the feature preprocessing algorithm. The second stage classifier is support vector machine with radial basis function kernel. We compare our algorithm with the baseline in tuning the pipeline with two real world datasets. The experimental results clearly show that our cascade Bayesian optimisation algorithm is able to tune the pipeline to achieve higher accuracy compared to the joint Bayesian optimisation.

For alloy heat treatment experiments, we consider industry standard simulation testbed of heat treatment for Al-Sc alloy. The testbed is based on the classical numerical precipitation model (Wagner et al., 1991; Robson et al., 2003). This model is built on molecular kinetic theory and can derive nucleation and growth of specific alloy compositions as the temperature is varied over time. Multiple processing steps are required as the radius of particles in the earlier stage affects the micro-structural evolution of the next stage. We design a three-stage heat treatment to achieve high hardness. We show that on average the cascade Bayesian optimisation can achieve peak hardness in only 15 trials compared to around 20 trials required for joint
4.2. The proposed solution

Bayesian optimisation. Since, a single heat-treatment trial can take anywhere from a day to 3 days, a saving of 5 trials can mean saving of at least 5 days. We further show that when cost is taken into account it is possible to save 30% of heat-treatment time while maintaining a similar level of hardness (<2% drop).

4.2 The proposed solution

Consider a process that consists of multiple stages (indexed as $s = 1, 2, ..., S$). In this process, each stage takes the output of the previous stage as input. Each stage also has control parameters that affect its output. Let us use $u^s_t$, $v^s_t$ and $w^s_t$ to denote the input, control parameters and output of stage $s$ for observation $t$, where $t = 1, 2, ..., T$. Further, let us use $f^s_t$ to denote the underlying function at iteration $t$ of stage $s$ such that $w^s_t = f^s_t(x^s_t)$ where $x^s_t = [u^s_t, v^s_t]$ is combination of input $u^s_t$ and control parameters $v^s_t$. This function is modelled by a Gaussian process $\mathcal{GP}^s_t$. See Fig. 4.2 for details of notation used in this section. Since the output of a stage $s$ acts as the input to the next stage $s + 1$, we have $w^s = u^{s+1}$. Let $y = w^S$ be the output of the final stage. Given the input $u^1$ at the first stage, the optimisation problem becomes discovering the set of control parameters $V^* = [v^{1*}, v^{2*}, ..., v^{S*}]$ that yields the highest output $y^*$ from the final stage in the minimum number of explorations. We call this problem as Cascade Bayesian Optimisation.

We propose an algorithm that consists of two steps (see algorithm 4.1).

**Step-1: Tuning parameters at the last stage.**
The problem of finding the maximum output value from the last stage can be represented as follows:

\[ y^* = \max_{x^*_t} f^*_t(x^*_t) \]  \hspace{1cm} (4.1)

Bayesian optimisation is used to solve this optimisation problem. At iteration \( t_0 + 1 \), the Gaussian process at last stage \( GP_{t_0}^S \) has \( t_0 \) observations: \( x^S_1, x^S_2, ..., x^S_{t_0} \). By using properties of Gaussian process, we have \( w^S_{t_0 + 1} \) following a Gaussian distribution whose mean and variance can be represented as functions of \( x^S_{t_0 + 1} \):

\[
\begin{align*}
\mu^S_{t_0}(x^S_{t_0 + 1}) &= k^T K^{-1} y_{1:t_0} \\
[\sigma^S_{t_0}(x^S_{t_0 + 1})]^2 &= k(x^S_{t_0 + 1}, x^S_{t_0 + 1}) - k^T K^{-1} k
\end{align*}
\]  \hspace{1cm} (4.2)

where \( k(x, x') \) is the squared exponential kernel function (see Eq. (2.12)), \( K \) is the co-variance matrix (see Eq. (2.8)) and \( k = [k(x^S_1, x^S_{t_0 + 1}), k(x^S_2, x^S_{t_0 + 1}), ..., k(x^S_{t_0}, x^S_{t_0 + 1})] \).

We use expected improvement (EI) acquisition function (Eq. (2.19)) to suggest the next sample \( x^S_{t_0 + 1} \) for exploration. This is done by maximising the acquisition function as

\[ x^S_{t_0 + 1} = \arg\max_x (\mu^S_{t_0}(x) - f^S_{t_0}(x^*) + \sigma^S_{t_0}(x) \phi(z)) \]  \hspace{1cm} (4.4)

where \( z = \frac{\mu^S_{t_0}(x) - f^S_{t_0}(x^*)}{\sigma^S_{t_0}(x)} \) when \( \sigma^S_{t_0}(x) > 0 \) and \( z = 0 \) otherwise, and \( f^S_{t_0}(x^*) \) is the current best value. It is also possible to use GP-UCB (Eq. (2.22)) acquisition function instead of EI.

**Step-2: Tuning parameters at remaining stages.**

After finding \( x^S_{t_0 + 1} \) at the last stage, we search for the input and control parameters at previous stages \( [u^1_{t_0 + 1}, v^1_{t_0 + 1}], [u^2_{t_0 + 1}, v^2_{t_0 + 1}], ..., [u^{S-1}_{t_0 + 1}, v^{S-1}_{t_0 + 1}] \) that produce \( x^S_{t_0 + 1} \). In other words, we have to compute \( x^1_{t_0 + 1}, x^2_{t_0 + 1}, ..., x^{S-1}_{t_0 + 1} \) given \( GP_{t_0}^1, GP_{t_0}^2, ..., GP_{t_0}^{S-1} \) and \( u^s_{t_0 + 1} \). Here we propose an approach to compute \( x^s_{t_0 + 1} \) given \( GP_s^{t_0} \) and \( u^s_{t_0 + 1} \) for \( s = 1, 2, ..., S - 1 \). Because a Gaussian process is built at each stage, \( x^s_{t_0 + 1} \) can be computed given the output \( w^s_{t_0 + 1} \). An intuitive approach to solve for \( x^s_{t_0 + 1} \) is to minimise the error between the output of Gaussian process \( GP_{t_0}^s \) and the desired
Algorithm 4.1 Cascade Bayesian optimisation algorithm

Input: Input of the first stage $u^1$

1. Initialise $t_0$ control parameters $v^{s_1}, ..., v^{s_{t_0}}$ and calculate $w^{s_1}_t, w^{s_2}_t, ..., w^{s_{t_0}}_t$ for $s = 1, 2, ..., S$
2. Build $GP^s_{t_0}$, $s = 1, 2, ..., S$

repeat $t := t_0$

for $s = S - 1$ down to 1

8. compute $x^s_t$ by minimising Eq. (4.5)

endfor

11. Add new data points $x^1_t, x^2_t, ..., x^S_t$ to the Gaussian processes $GP^1_t, GP^2_t, ..., GP^S_t$.

until $y^*$ found or $t > T$

Output: $V^*, y^*$

output $w^{s_{t_0+1}}$ (we note that $w^{s_{t_0+1}} = u^{s_{t_0+1}}$):

$$\min_{x^{s_{t_0+1}}} \left\| \mu_{t_0}^s(x^{s_{t_0+1}}) - u^{s_{t_0+1}} \right\|$$

However, since Bayesian optimisation deals with costly functions and there are only limited data points, the prediction of the model can not be overly trusted. Thus, the uncertainty of Gaussian process should also be considered in the objective function. Therefore, we incorporate uncertainty in the optimisation as

$$\min_{x^{s_{t_0+1}}} \left\{ \frac{\left\| \mu_{t_0}^s(x^{s_{t_0+1}}) - u^{s_{t_0+1}} \right\|^2}{\sigma^S_{t_0}(x^{s_{t_0+1}})^2} + \kappa_2 \left\| \mu_{t_0}^s(x^{s_{t_0+1}}) - u^{s_{t_0+1}} \right\|^2 \right\}$$

subject to $E \left[ u^{s_{t_0+1}} \mid GP^s_{t_0}, x^{s_{t_0+1}} \right] = \mu_{t_0}^s(x^{s_{t_0+1}})$

where $\kappa_1$ and $\kappa_2$ are model parameters balancing the requirement between exploitation (error term) and exploration (uncertainty) of the objective function. Since the function inversion problem may be ill-posed meaning there may be many different inputs that lead to the same output, we use a criteria for choosing one input among
The proposed solution

4.2. The proposed solution

The final optimisation problem is defined as follows:

$$
\min_{x_{t_0 + 1}} \sum_{s=1}^{n} \frac{\|\mu^s_{t_0}(x^s_{t_0 + 1}) - u^s_{t_0 + 1}\|^2}{\sigma^s_{t_0}(x^s_{t_0 + 1})^2} + \sum_{s=1}^{n} \sum_{j=1}^{m} \frac{\|\mu^s_{t_0}(x^s_{t_0 + 1}) - u^s_{t_0 + 1}\|^2}{\sigma^s_{t_0}(x^s_{t_0 + 1})^2} + c(x^s_{t_0 + 1})
$$

subject to

$$
\mathbb{E}\left[u^s_{t_0 + 1} | GP_{t_0}, x^s_{t_0 + 1}\right] = \mu^s_{t_0}(x^s_{t_0 + 1})
$$

where $c(x^s_{t_0 + 1})$ is the cost function that is designed to choose among multiple input solutions. In our implementation, we use $c(x^s_{t_0 + 1}) = \|x^s_{t_0 + 1}\|^2$. At the first stage, since the input is fixed and only control parameters can vary, the inversion process will only estimate the control parameter $v_1$ instead of $x_1$.

The problem of finding maximum value of cascaded process now becomes optimising two function (4.4) and (4.5). Here we derive the derivative of objective function of (4.5) which is useful if the objective function is optimised using local optimisers. Although it is possible to use global optimisers for this optimisation, local optimisers are our only choice in high dimensional parameter space due to computational reasons. Let $\Psi(x)$ be the objective function of (4.5). Its derivative can be written as

$$
\frac{d\Psi(x)}{dx} = \kappa_1 \frac{d}{dx} \left[ (\mu(x) - u)^2 \frac{\sigma^2(x)}{\sigma^2(x)} \right] + \kappa_2 \frac{d}{dx} (\mu(x) - u)^2 \sigma^2(x) + \frac{d}{dx} c(x)
$$

where $A = 2\sigma^2(x)(\mu(x) - u)\frac{d\mu(x)}{dx}$ and $B = (\mu(x) - u)^2 \frac{d\sigma^2(x)}{dx}$. Finally, $A$ and $B$ can be rewritten as follows

$$
A = -2 \frac{d}{dx} \sigma^2(x) (\mu(x) - u) [K^{-1}y]^T C
$$
$$
B = -2 \frac{d}{dx} (\mu(x) - u)^2 k^T K^{-1} C
$$

where

$$
C = \begin{bmatrix}
k(x, x_1)(x - x_1)^T \\
k(x, x_2)(x - x_2)^T \\
\vdots \\
k(x, x_{t_0})(x - x_{t_0})^T
\end{bmatrix}
$$
Convergence analysis

In this section, we analyse the convergence of the cascade Bayesian optimisation algorithm. First, we present the error analysis of the Gaussian process inversions. Then, we derive the convergence rate of the proposed algorithm.

In the cascade Bayesian optimisation algorithm, due to the uncertainties in the function models, the inversions in the first $S-1$ stages introduce errors. At iteration $t$, instead of producing the required input $u^S_t$ for the last stage, the evaluation for the $S-1$-th stage gives $\tilde{u}^S_t = u^S_t + \epsilon^S_t$ where $\epsilon^S_t$ is the error of the output. Indeed, this error is the generalisation error of Gaussian processes due to limited number of datapoints. In machine learning, the generalisation error is usually used as a measure of how accurate machine learning algorithms perform for test data. Since the generalisation errors can not be exactly computed, we usually aim to find their bounds. The generalisation error bounds have been proposed for various machine learning algorithms in (Bousquet and Elisseeff, 2002). For regularised least square regression, the generalisation error bound has been derived (Bousquet and Elisseeff, 2002), and can be written as follows:

$$
\epsilon_g \leq \frac{4\rho^2 B^2}{\lambda t} + \left( \frac{8\rho^2 B^2}{\lambda} + 2B \right) \sqrt{\frac{\ln \frac{1}{\delta}}{2t}} = O\left( \frac{1}{\sqrt{t}} \right),
$$

with the probability $1 - \delta$, where $\rho$ is a constant representing kernel's scale factor, $B$ is the upper bound of the function value (assuming 0 is its lower bound), $\lambda$ is a regularisation parameter controlling model complexity and $t$ is the number of datapoints in the training dataset. Since a Gaussian process regression model can be seen as a regularised least squares regression model (Rasmussen and Williams, 2006), using the above result, we have the bound on the variance of $\epsilon^S_t$ as follows:

$$
\text{var}(\epsilon^S_t) \leq O\left( \frac{1}{t} \right).
$$

From the definition of $\tilde{u}^S_t$ and the bound on the variance of $\epsilon^S_t$, $\tilde{u}^S_t$ can be seen as a random variable with $\text{var}(\tilde{u}^S_t) \leq O\left( \frac{1}{t} \right)$. We consider Gaussian distribution for the subsequent analysis. Since the number of dimensions in $u^S_t$ is smaller than the number of dimensions in $x^S_t$ and we can set exact $w^S_t$, we have $\text{var}(\tilde{x}^S_t) \leq O\left( \frac{1}{t} \right)$. 


4.2. The proposed solution

Using the law of total variance, the variance of \( f^S_i(x^S_i) \) can be written as follows:

\[
\text{var}(f^S_i(x^S_i)) = \mathbb{E}_{x^S_i} \left[ \text{var}_{f^S_i} \left( f^S_i(x^S_i) \right) \right] + \text{var}_{x^S_i} \left[ f^S_i(x^S_i) \right]
\]

For simplicity in notation, we drop the subscript notations for variances and expectations in subsequent derivation.

To analyse the convergence of our proposed algorithm, we derive the upper bound on \( \text{var}(\mu^S_i(x^S_i)) \). Because \( \mu^S_i(.) \) is the mean function of a Gaussian process, we can assume \( \mu^S_i(.) \) to be a smooth function with a Lipschitz constant \( L \) (González et al., 2016). We present the following lemma for the upper bound of the variance of Lipschitz functions.

**Lemma 4.1.** Let \( X \) be a random variable and \( f \) be a function with Lipschitz constant \( L \), the following inequality holds:

\[
\text{var}(f(X)) \leq L^2 \text{var}(X).
\]

**Proof.** For any random variable \( Y \), we have:

\[
\text{var}(Y) = \mathbb{E}[Y^2] - (\mathbb{E}[Y])^2 \leq \mathbb{E}[Y^2]
\]

Replace \( Y = f(X) - f(\mathbb{E}[X]) \) we have:

\[
\text{var}(f(X)) = \text{var}(f(X) - f(\mathbb{E}[X])) \leq \mathbb{E} \left[ |f(X) - f(\mathbb{E}[X])|^2 \right] \quad (4.8)
\]

Since \( f \) is a function with Lipschitz constant \( L \), we have:

\[
|f(x) - f(y)| \leq L |x - y| \quad (4.9)
\]
4.2. The proposed solution

From Eq. (4.8) and Eq. (4.9) we have:

\[
\text{var}(f(X)) \leq \mathbb{E} \left[ |f(X) - f(\mathbb{E}[X])|^2 \right] \\
\leq \mathbb{E} \left[ L^2 |X - \mathbb{E}[X]|^2 \right] \\
\leq L^2 \text{var}(X).
\]

Applying the above lemma for \( \mu_t^S(x_t^S) \), we obtain the upper bound on \( \text{var}(\mu_t^S(x_t^S)) \) as follows:

\[
\text{var}(\mu_t^S(x_t^S)) \leq L^2 \text{var}(x_t^S) \leq L^2 \mathcal{O}(\frac{1}{t}). \tag{4.10}
\]

Now we are going to present a theorem for the convergence rate of the cascade Bayesian optimisation algorithm. Let \( r_t = f_t^S(x^*) - f_t^S(x_t) \) be the instantaneous regret at iteration \( t \) of the proposed algorithm, where \( x^* = \arg\max_x f_t^S(x) \). The cumulative regret \( R_T \) of the algorithm after \( T \) observations is the sum of instantaneous regrets: \( R_T = \sum_{t=1}^{T} r_t \). Our goal is to derive bounds on the average regret (i.e. \( \frac{R_T}{T} \)) since it translates to the convergence rate of our algorithm.

**Theorem 4.1.** The average regret (i.e. \( \frac{R_T}{T} \)) of the proposed algorithm has the following upper bound:

\[
\frac{R_T}{T} \leq \mathcal{O}(\beta_T \sqrt{\frac{T}{T}})
\]

where \( \beta_T \) is a parameter proportional to \( \log T \) defined in (Wang and de Freitas, 2014) and \( \gamma_T \) is the maximum information gain.

**Proof.** Using the results in (Wang and de Freitas, 2014) for the relation between the cumulative regret and variances, we have:

\[
\frac{R_T^2}{T^2} \leq \frac{1}{T} \sum_{t=1}^{T} r_t^2 \\
\leq \frac{1}{T} \mathcal{O}(\beta_T^2) \sum_{t=1}^{T} \text{var}(f_t^S(x_t^S)) \\
\leq \frac{1}{T} \mathcal{O}(\beta_T^2) \sum_{t=1}^{T} \left( \sigma_t^S(x_t^S)^2 + \text{var}(\mu_t^S(x_t^S)) \right). \tag{4.11}
\]
For the squared exponential kernel, we have the following bound on the sum of 
\[ \sigma_S^2(x_t^S)^2 \] (Srinivas et al., 2010; Wang and de Freitas, 2014):

\[
\sum_{t=1}^{T} \sigma_t^S(x_t^S)^2 \leq O(\gamma T) = O((\log T)^{d+1})
\]

(4.12)

where \( d \) is the number of dimensions.

Using the properties of harmonic series and Eq. (4.10), we have the following bound

\[
\sum_{t=1}^{T} \text{var}(\mu_t^S(x_t^S)) \leq \sum_{t=1}^{T} O\left(\frac{1}{t}\right) \leq O(\log T + 1).
\]

(4.13)

Thus, using Eq. (4.11), Eq. (4.12) and Eq. (4.13), the average regret of our proposed
cascade Bayesian optimisation has the following upper bound:

\[
\frac{R_T}{T} \leq \sqrt{\frac{1}{T} O(\beta_T^2) [O(\gamma T) + O(\log T + 1)]} = O\left(\beta_T \sqrt{\frac{\gamma T}{T}}\right).
\]

From this theorem, it can be seen that the average regret goes toward zero as
\( T \to \infty \), which indicates the convergence property of our proposed cascade Bayesian
optimisation algorithm. Importantly, the rate of convergence is sub-linear in the
number of function evaluations \( T \).

### 4.3 Experiments

In this section, we conduct a set of experiments using both synthetic and real data
to demonstrate the efficiency of our cascade Bayesian optimisation method. Experi-
ments with synthetic data are performed to illustrate the behaviour of our method in
different scenarios e.g. how does the efficiency scale with growing number of stages
in the cascade, how does the efficiency scale with growing number of parameters in
a stage. We conduct the experiments with data pipeline to show the efficacy of our propose algorithm in tuning hyperparameters of machine learning systems. We also apply our method for alloy heat treatment optimisation where we show the benefits of our method in two terms: the number of trial experiments to reach a desired alloy hardness and the total heating time required.

4.3.1 Baseline method and evaluation measure

To evaluate the effectiveness of our proposed model, we compare its performance with a baseline named *joint Bayesian optimisation*. When presented with an $n$-stage cascaded process optimisation, this algorithm does not take cascaded process into account. Instead it combines all the control parameters of all stages into a single input vector and uses Bayesian optimisation to optimise the overall underlying process in this combined space. The output of the last stage is used as target to be maximised. To the best of our knowledge, there are no other methods that can perform optimisation for cascaded processes in sequential design setting.

To measure the effectiveness of our proposed and the baseline methods, we plot the best output value reached from the last stage as a function of number of iterations.

4.3.2 Experiments with synthetic data

We evaluate the proposed methods with two synthetic datasets. The first synthetic dataset is created to demonstrate the behaviour of the proposed model and baseline method with increasing number of stages, whereas the second synthetic dataset is generated to illustrate performance of cascade Bayesian optimisation and joint Bayesian optimisation by varying number of control parameters for each stage.

4.3.2.1 Data generation

**Synthetic-I:** This synthetic dataset is designed to illustrate the performance of the proposed method with varying number of stages. We generate 4, 6 and 8 stage
processes that have 4 control parameters for each stage. The underlying function at each stage is the probability density function of a multivariate Gaussian distribution. The input for $s$-th stage is the combination of control parameter of $s$-th stage and the output of the $s-1$-th stage. Thus, our proposed method has to optimise in 5 dimensional space, whereas joint Bayesian optimisation’s input has 16, 24 and 32 dimensions respectively.

**Synthetic-II:** We set the underlying function of each stage as the probability density function of a multivariate Gaussian distribution. In this case, we keep the number of stages fixed to be 5 and vary the number of control parameters as 3, 5 and 7. This makes the number of dimensions for the input of cascade Bayesian optimisation as 4, 6 and 8, whilst the input of baseline method has 15, 25 and 35 dimensions respectively.

### 4.3.2.2 Experimental results

We generate 20 different initialisations for the experiments with synthetic data. For each experiment, the number of initial datapoints are proportional to the total number of control parameters. The results reported below are averaged over these 20 initialisation.

**Synthetic-I:** Fig. 4.3a, 4.3c and 4.3e (left column) show the experimental results for the Synthetic-I dataset where number of control parameters is fixed and number of stages is varied. Fig. 4.3a illustrates performance of cascade Bayesian optimisation and baseline method for 4 stages and 4 control parameters for each stage. Starting with the same set of points, our proposed algorithm performs better than the baseline method in terms of best-found-value within a given number of iterations. Cascade Bayesian optimisation reaches 0.5 after 38 iterations and reaches 0.7 after 48 iterations while joint Bayesian optimisation can only reach 0.5 after 100 iterations. Fig. 4.3c and 4.3e show similar results. Cascade Bayesian optimisation gains 0.9 after 100 iterations whereas the joint Bayesian optimisation reaches 0.4 and 0.3 when the number of stages is 6 and 8 respectively.

**Synthetic-II:** Fig. 4.3b, 4.3d and 4.3f (right column) show the experimental results for Synthetic-II dataset where number of stages is fixed and number of control
4.3. Experiments

Figure 4.3: Experimental results of Synthetic-I (left column) and Synthetic-II (right column): Best found value so far as a function of iteration. In the left column, number of control parameters per stage is fixed at 4 but the number of stages are varied. In the right column, number of stages is fixed at 5, but the number of control parameters per stage is varied.

parameters at each stage is varied. When the number of control parameters is 3, joint Bayesian optimisation has to solve the optimisation problem with 15 parameters while cascade Bayesian optimisation has to solve the optimisation problem with 4 dimensional input. The maximum found value of cascade Bayesian optimisation is almost doubled the one of the baseline in 100 iterations (Fig. 4.3b). As the number of control parameters increase, both algorithms perform worse. After 100 iterations, the best value found by cascade Bayesian optimisation is 0.9 while the best value found by the joint Bayesian optimisation is only 0.2 in the case the process has 5 control parameters for each stage (Fig. 4.3d). Fig. 4.3f shows the results of the dataset with 7 control parameters for each stage. Joint Bayesian optimisation has to find result in 35 dimensional space while cascade Bayesian optimisation has to
optimise in 8 dimensional space. As a result, joint Bayesian optimisation suffered heavily and can not find any value bigger than 0.05 after 100 iterations. On the contrary, the best value found by the cascade Bayesian optimisation is 0.65 after the same number of iterations.

4.3.3 Experiments with tuning hyperparameters for data analytic pipelines

4.3.3.1 Tuning hyperparameters for data analytic pipelines

In this experiment, we compare our proposed cascade Bayesian optimisation with the baseline in tuning hyperparameters for a data analytic pipeline. We consider a pipeline consisting of two steps: feature preprocessing and classification. In the feature preprocessing stage, the raw data input will be processed using kernel principal component analysis with polynomial kernel. We tune three hyperparameters of this stage: number of principal components, degree of the kernel and the coefficient of the dot product. After feature preprocessing stage, data will be transformed and become input for the classification stage. We also add the total variance captured by the selected principal components to the input for the next stage. At the classification stage, we use kernel support vector machine with radial basis function kernel as the classifier. This classifier has two hyperparameters $C$ and $\gamma$ that represent misclassification trade-off and parameter of radial basis function kernel respectively. The final output is classification accuracy. Our goal is to tune hyperparameters for this pipeline to achieve the highest possible accuracy. We use letter and glass classification dataset from UCI machine learning repository\(^1\) to test our hyperparameter tuning.

4.3.3.2 Experimental results

Fig. 4.4 shows the performance of cascade Bayesian optimisation and the joint Bayesian optimisation on tuning the pipeline averaged over 15 initialisation. For

\(^1\)http://archive.ics.uci.edu/ml
4.3. Experiments

Figure 4.4: Performance of cascade Bayesian optimisation and joint Bayesian optimisation using SVM with respect to number of iterations on tuning hyperparameters of a data analytic pipeline (a) Letter dataset (b) Glass dataset.
letter dataset (Fig. 4.4a), after 20 iteration, the system tuned by cascade Bayesian optimisation reaches 0.86 in accuracy whereas the one tuned by the joint Bayesian optimisation reaches 0.77. After 50 iterations, the performance of the proposed algorithm and the baseline are 0.89 and 0.83 respectively. We observe similar results for glass dataset (Fig. 4.4b). After 20 iterations, the performance of our proposed algorithm and the baseline are 0.63 and 0.60 respectively. After 50 iterations, the cascade Bayesian optimisation algorithm reaches 0.71 compare to 0.63 of the joint Bayesian optimisation.

4.3.4 Experiments with alloy heat treatment optimisation

4.3.4.1 Alloy heat treatment

This is based on a simulation model of a real world heat treatment process of an Al-Sc alloy. The underlying physics of alloy strengthening is based on nucleation and growth. Nucleation is the process of either a new “phase” formation or clusters of atoms or precipitates through a self-organising process. This process happens at lower temperatures over time. It is a stochastic process and thus difficult. The aim of the first step is to maximise nucleation, or the number of precipitates. The second step is growth. Through diffusion the initial precipitates grow and the requisite alloy property is achieved. We use the industrial standard precipitation model KWN (Wagner et al., 1991; Robson et al., 2003) model for the kinetics of nucleation and growth. This tracks the precipitation nucleation, growth and coarsening over discrete time steps. It does so using Gibbs-Thomson relationship equations and nucleation theory. The model has several phases. For each heat treatment temperature, it iterates and calculates the precipitation for each time-step which is then adjusted using the Gibbs-Thomson equation. The outputs include hardness and precipitate.

We consider a three stage heat treatment process. The input to first stage is the alloy composition, the temperature and time. The nucleation output of this stage is input to the second stage along with the temperature and time for the second stage. The input of final stage is the hardness of the alloy composition at second stage, temperature and time. The final output is hardness of the material. We seek to find the heat treatment that results in maximum hardness.
4.3.4.2 Experimental results

Fig. 4.5 shows the comparison of the cascade Bayesian optimisation (red) and baseline method (blue) vs iteration, for varying cost parameter $\lambda$. Since the initial observation are randomly set, the hardness values remained very low at four random initialisation. After that, the cascade Bayesian optimisation outperformed the joint Bayesian optimisation in terms of the speed at which it reaches higher hardness values. It takes only 6 iteration for cascade Bayesian optimisation to reach the hardness of 120 whereas the baseline method needs 10 to reach the same level of hardness. After 10 iteration, our proposed method gets the hardness of 140 while the joint Bayesian optimisation needs 15 iteration to get to a similar value.

Figure 4.5: Best hardness achieved as a function of number of iterations for both cascade BO (red) and joint Bayesian optimisation (blue), for different cost parameter $\lambda$.
4.3.5 Cost-efficient optimisation

As we mentioned earlier, the optimisation problem of finding the input given the output of a Gaussian process generally has multiple solutions, therefore, it is possible to choose a solution that meets a specific criteria. As energy costs are related to oven temperature and times, in the intermediate stages we encourage solution that minimises the norm of time and temperature vector. Let us denote $q_i$ as the time at $i$-stage of the process, then the total time taken for each process is computed as $\sum_{i=1}^{N} q_i$. Fig. 4.6 shows this trade-off between hardness and time with respect to inverse Gaussian parameter $\lambda$. As $\lambda$ increases from 0 to 0.1, the hardness drops slightly, however we could save almost 20% of time. When we set $\lambda = 10$, the hardness drops to 146 while the average time is 5 hours which saves 30% of time compare with $\lambda = 0$.

4.4 Summary

We proposed a novel cascade Bayesian optimisation method to tune the parameters of a multi-stage cascaded system, often found in industrial processes. Each stage is separately modelled and appropriately optimised such that the final stage output is maximised. A novel optimisation formulation is provided that exploits the
epistemic uncertainties of the underlying model. Additionally, the formulation is also made cost-sensitive to find cost-efficient solutions in the small number of trials. We showed that the convergence rate of the proposed algorithm is sub-linear in the number of evaluations $T$. In the experiment with hyperparameter tuning, the systems tuned by the proposed cascade Bayesian optimisation algorithm had higher accuracy compared to the one tuned by the standard Bayesian optimisation approach. On a simulated testbed of 3-stage heat-treatment for Al-Sc alloy, the cascade Bayesian optimisation showed superior performance over a naïve approach that ignores the structure and jointly optimises all control variables. When cost is taken into account, we were able to save 30% of the total time with only slight drop in hardness value.
Chapter 5

A new privacy framework for Bayesian optimisation and other algorithms

In previous chapters, we addressed two challenges in applying Bayesian optimisation for real world applications: finding stable solutions and incorporating cascade structure. In this chapter, we aim to address privacy concerns of using Bayesian optimisation in practical settings.

In industrial applications, Bayesian optimisation can significantly reduce time and cost of finding optimal industrial designs. However, often the experimenters in industries may not have the expertise of optimisation techniques and may require help from third-party optimisation services. If the third-party optimisation services are untrusted, there may be privacy concerns as the optimised design of an industrial process typically needs to be kept secret to retain its competitive advantages.

Until now, there is no privacy preserving Bayesian optimisation method that allows to retain privacy under an untrusted third-party or an adversarial optimiser. In addition, the existing privacy frameworks that have strong privacy guarantees (e.g. differential privacy) have low utility to be applicable in practice. We propose a novel privacy preserving framework called Error Preserving Privacy (EPP) that thwarts an adversary from inferring any sensitive information without significantly
degrading the utility. Under the EPP framework, we propose a novel Bayesian optimisation algorithm that can allow the experimenters from an industry to utilise the expertise of a third-party optimisation service in privacy preserving manner. Our proposed EPP framework is generic and can be applied to any machine learning algorithm. As an example to demonstrate the applicability of the EPP framework, we propose a new privacy preserving K-means clustering algorithm under this framework. Using both synthetic and real datasets, we demonstrate that the efficiency of Bayesian optimisation and K-means clustering algorithm using our framework is comparable to non-private algorithms and significantly better than the differential privacy counterparts.

5.1 Introduction

Since Bayesian optimisation is an efficient method for optimising expensive functions, its use to optimise the industrial processes can significantly reduce the time and cost. However, the experimenters in industries typically may not have the expertise of optimisation techniques and therefore require optimisation services from a third-party. This can cause privacy concerns as the optimised design of an industrial process typically needs to be kept secret to retain its competitive advantages. To better understand this concern, consider the two parties involved in the design optimisation process: an experimenter A and an optimiser B. At each iteration of the Bayesian optimisation, the optimiser B asks the experimenter A to perform experiments (or evaluate the function) at the suggested point. The experimenter A conducts the experiment, assesses the outcomes to score the function output and then returns the function value to the optimiser B. This interaction repeats until the optimum is found or the number of experiments exceeds a per-defined budget. Since both experimenter and optimiser have access to the exact knowledge of the optimum point, this algorithm does not offer any privacy. In industrial world, often the data is sensitive and the optimum of the objective function can not be revealed for various reasons, for instance, to keep competitive advantage. As an example, consider an alloy making company that needs to design an alloy with certain target properties. The task involves optimising the mixture proportions of permitted metals, which then needs to be kept secret for business. In such cases, the experimenter A from
the industry wants to avail the service of the optimiser B without disclosing the exact function values.

Privacy preserving data mining has become an active research area. There are different ways to achieve privacy such as perturbing data (Agrawal and Srikant, 2000), anonymisation (Sweeney, 2002), etc. These methods aim to retain useful statistical information about data while changing the database itself. Recently, differential privacy (Dwork, 2006) has emerged as a strong privacy preserving framework. It protects the data privacy even when an adversary has access to auxiliary information (see section 2.8 in chapter 2 for more details). Several machine learning and data mining models using this framework have been explored such as logistic regression (Chaudhuri and Monteleoni, 2009), decision tree learning (Jagannathan et al., 2009) and matrix factorisation (Hua et al., 2015). Although differential privacy provides a strong guarantee on privacy, it often perturbs the output of algorithms so much that their utility drops to unacceptable levels.

In this chapter, we propose a new privacy preserving framework called Error Preserving Privacy that provides strong guarantee on privacy while ensuring high utility. This framework can handle arbitrary amounts of auxiliary knowledge about the database, that is, even if an adversary has access to all but one data point, the framework still thwarts the adversary from inferring the useful statistical information. We achieve this by randomising the output of the algorithm. Indeed, our framework ensures that the error in the adversary’s estimation (of the quantity of interest) does not change significantly due to the participation of a data point in the database. By ensuring that the error in estimation by the adversary is almost invariant to the inclusion/exclusion of the data point in the database, the adversary is defeated. Our framework significantly departs from the differential privacy in the manner that in presence/absence of a data point, differential privacy preserves the likelihood of algorithm output while our framework preserves the error variance. By focusing directly on the estimation error, our framework is able to use significantly smaller perturbation in the algorithm output compared to the differential privacy.

Using Error Preserving Privacy framework, we propose a privacy preserving Bayesian optimisation algorithm that helps to find the optimum of an expensive black-box function without revealing the best point up to any optimisation iteration. The EPP helps to maintain high optimisation efficiency even under the stringent privacy
requirements. The proposed algorithm follows a three-step iterative procedure. The first step is evaluating the function value at the input (performed by experimenter) suggested by the optimiser. The second step is perturbing the function value by a noise that helps to protect the privacy of the true optimum (performed by experimenter). The third step is at the untrusted end where the perturbed point is included in the function model by the optimiser to suggest the next evaluation point. Fig. 5.1 depicts our optimisation setting. Under certain assumptions on adversary model, we perform a theoretical analysis and derive the amount of perturbation required to guarantee the privacy. We apply our algorithm to benchmark optimisation problems as well as optimisation problems from real-world industrial processes and demonstrate that the optimisation efficiency of our algorithm is comparable to the non-private Bayesian optimisation algorithm. We also suggest a differentially-private Bayesian optimisation baseline and show that the performance of EPP based Bayesian optimisation algorithm is significantly better than the differentially-private baseline.

The proposed EPP framework can also be applied to many other machine learning algorithms. To demonstrate the applicability of EPP framework, we construct a novel, privacy preserving K-means algorithm. The key idea is to perturb the cluster centroids before their release. Since bootstrap sampling offers randomness, the perturbation is realised by using bootstrap aggregation to compute the cluster centroids. We analyse our method theoretically, and derive bounds on the size of bootstrap ensemble to ensure the stipulated privacy. We consider two cases depending on if the cluster membership of a data point is known or unknown to the adversary. Using both synthetic and real datasets, we compare our algorithm against baselines
5.2. The new privacy framework

- the conventional K-means (non-private) and differentially private K-means. The results are remarkable - at high levels of privacy, the utility of our method is almost the same as the non-private K-means, and significantly better than its differential privacy counterpart. This is because for the same privacy level, we need to add significantly lower levels of noise compared to differential privacy - as example, the noise in our framework is almost 20 times lower for high privacy stipulated by leakage parameter $\epsilon$ less than 0.1.

5.2 The new privacy framework

In this section, we present a new privacy framework where our goal is to provide strong privacy guarantee on the useful statistics of a database while ensuring that utility of algorithms remain high. The proposed framework is capable of handling the arbitrary amount of auxiliary knowledge about the database in the sense that even if an adversary has access to all but one data point, the framework still thwarts an adversary from inferring the statistics.

Let $\mathcal{D}_n = \{x_1, x_2, ..., x_n\}, x_i \in \mathbb{R}^d$ be a dataset with $n$ data points and $g$ be a quantity of interest that needs to be protected. Inspired by the strong guarantees of differential privacy framework (Dwork, 2006, 2008), we proposed Error Preserving Privacy (EPP) - a new privacy framework that provides privacy guarantees for $g$ even in presence of auxiliary information. As in the differential privacy framework, our proposed framework controls the level of privacy using a pre-specified leakage parameter $\epsilon$. In particular, given an adversary model for estimating $g$, the errors in the adversary’s estimates of $g$ are guaranteed to be similar for any two datasets $\mathcal{D}_n$ and $\mathcal{D}_{n+1}$ differing by just one data point (say $x_{n+1}$). Thus, the extra information gained by the adversary by knowing $x_{n+1}$ brings negligible risks on the privacy of the quantity $g$ for small $\epsilon$. Let us assume that an adversary estimates the statistic $\hat{g}(\mathcal{D}_n)$ and $\hat{g}(\mathcal{D}_{n+1})$ using data $\mathcal{D}_n$ and $\mathcal{D}_{n+1}$ respectively. If we denote by $\mathcal{E}(\hat{g}(\mathcal{D}_n))$ the error of the adversary in estimating $g$ using data $\mathcal{D}_n$, i.e. $\mathcal{E}(\hat{g}(\mathcal{D}_n)) = \mathbb{E} \left[ (\hat{g}(\mathcal{D}_n) - g)^2 \right]$ and by $\mathcal{E}(\hat{g}(\mathcal{D}_{n+1}))$ the error of the adversary in estimating $g$ using data $\mathcal{D}_{n+1}$, i.e. $\mathcal{E}(\hat{g}(\mathcal{D}_{n+1})) = \mathbb{E} \left[ (\hat{g}(\mathcal{D}_{n+1}) - g)^2 \right]$, then the EPP framework ensures the following
5.3 Privacy preserving Bayesian optimisation

inequality:
\[
\frac{\mathcal{E}(\hat{g}(\mathcal{D}_{n+1}))}{\mathcal{E}(\hat{g}(\mathcal{D}_n))} \geq \exp(-\epsilon)
\]  
(5.1)

where \( \epsilon \geq 0 \) is a pre-specified privacy leakage parameter. In the above inequality, when the value of \( \epsilon \) is 0, the strongest level of privacy is offered. As the value of \( \epsilon \) is increased, the level of privacy drops.

**Intuition**

We can think of Error Preserving Privacy framework as an adversarial situation between Alice and Bob. Alice is a scientist who wants to release the useful statistical information from database. Bob is an adversary who wants to get sensitive information from database. Intuitively, EPP framework helps Alice to protect the privacy of one record in the database even if Bob has auxiliary information about the remaining records. By assuming the attacking method is known, Alice can derive the amount of noise required to stop Bob from making any better estimate from what he already has, and perturb the results accordingly to preserve the privacy of the records. Fig. 5.2 shows a diagram of using EPP framework for machine learning algorithms.

### 5.3 Privacy preserving Bayesian optimisation

In this section, using our Error Preserving Privacy framework, we propose a new privacy preserving Bayesian optimisation algorithm that protects the optimum of a function from an untrusted optimiser. Then we conduct the experiments on optimisation of synthetic benchmark functions and real world problems to demonstrate the effectiveness of our algorithm.

#### 5.3.1 The proposed algorithm

Let us imagine an industrial design optimisation task involving two parties: “an experimenter” and “an optimiser”. The optimiser is assumed to be untrusted. The “experimenter” wants to find the maximum of an objective function (the function
5.3. Privacy preserving Bayesian optimisation

Figure 5.2: Using Error Preserving Privacy framework for machine learning algorithms.
underlying the industrial process) $f$ and wants to utilise the services offered by the “optimiser”. Let $\mathcal{D}_n = \{(x_1, y_1), ..., (x_n, y_n)\}$ be the set of observations such that $y_i = f(x_i) + \varepsilon_i$ where $\varepsilon_i \sim \mathcal{N}(0, \sigma_e^2)$ is measurement noise. Further, let $\mathcal{D}_{n+1} = \mathcal{D}_n \cup (x_{n+1}, y_{n+1})$. Let $(x_{n+1}^+, y_{n+1}^+)$ be the “best point so far” in $\mathcal{D}_{n+1}$ such that $x_{n+1}^+ = x_i$ and $y_{n+1}^+ = y_i$ and $i = \arg \max_{j=1,\ldots,n+1} y_j$. Since the optimiser may be not trustworthy, the experimenter does not want to disclose the true optimum $(x_{n+1}^+, y_{n+1}^+)$ to the untrusted optimiser. Instead, the experimenter decides to share the experimental data in a privacy preserving manner, which in this case is achieved by perturbing the function value. The quantity of interest that needs to be protected at all times is the best point at any iteration. Our aim is therefore to share the data between the experimenter and the optimiser in such a way so that $(x_{n+1}^+, y_{n+1}^+)$ is ambiguous for the optimiser (assumed to be an adversary here) even if the optimiser has exact knowledge of data in $\mathcal{D}_n$. In the following, we refer to the optimiser as adversary.

We next develop a Bayesian optimisation algorithm maintaining this privacy under the EPP framework. Let $\hat{\omega}_n = \hat{y}_{n+1}^+ | \mathcal{D}_n$ and $\hat{\omega}_{n+1} = \hat{y}_{n+1}^+ | \mathcal{D}_{n+1}$ be the estimates of the adversary about the “best point so far” using $\mathcal{D}_n$ and $\mathcal{D}_{n+1}$ respectively. The EPP framework ensures the errors of the adversary’s estimates $\hat{\omega}_n$ and $\hat{\omega}_{n+1}$ are similar, which means that by acquiring $(x_{n+1}, y_{n+1})$, the adversary’s estimate of $y_{n+1}^+$ does not change significantly. This helps in hiding the true optimum $y_{n+1}^+$ and also provides the protection for the location of the maximum $x_{n+1}^+$. Formally, we denote by $E(\hat{\omega}_n)$ and $E(\hat{\omega}_{n+1})$ as the errors in the adversary’s estimate of $y_{n+1}^+$ using $\mathcal{D}_n$ and $\mathcal{D}_{n+1}$ respectively. For simplicity in notation, we refer to these quantities as $E_n$ and $E_{n+1}$.

In the absence of the privacy preserving scheme, the error of the adversary’s estimates using $\mathcal{D}_{n+1}$ may be significantly lower than the one using $\mathcal{D}_n$ as the adversary can simply find the maximum over all the observations. To ensure the privacy, we add a Gaussian distributed perturbation noise to the function output $y_{n+1} \leftarrow y_{n+1} + \nu_{n+1}$ where $\nu_{n+1} \sim \mathcal{N}(0, q_{n+1}^2)$, $q_{n+1}$ is the standard deviation of the noise. The following theorem characterises the amount of noise required to guarantee the EPP framework.

**Theorem 5.1.** The noise standard deviation $q_{n+1}$ obtained as solution of Eq. (5.15) ensures Error Preserving Privacy for the Bayesian optimisation algorithm.
5.3. Privacy preserving Bayesian optimisation

Proof. To prove the theorem, we need to derive the error in the adversary’s estimates: $E_n$ and $E_{n+1}$. After deriving these errors, we plug them in Eq. (5.1) to obtain an equation in $q_{n+1}$ (see Eq. (5.15)). The minimum value of $q_{n+1}$ that satisfies Eq. (5.15) ensures the EPP privacy guarantee.

For the adversary estimation model, we assume that given $D_{n+1}$, the adversary estimates the $\hat{y}_{n+1}^+$ as

$$\hat{y}_{n+1}^+ = \max(y_1, \ldots, y_n, \hat{y}_{n+1}|D_{n+1})$$

Similarly, given $D_n$, the adversary estimates the $\hat{y}_{n+1}^+$ as

$$\hat{y}_{n+1}^+ = \max(y_1, \ldots, y_n, \hat{y}_{n+1}|D_n)$$

where $\hat{y}_{n+1}^+|D_n$ is also estimated using a Gaussian process model.

**Computation of $E_n$:** As per the stated adversary model, $\hat{\omega}_n$ is given as

$$\hat{\omega}_n = \max(\hat{\theta}_n, y_n^+)$$

where $y_n^+ = \max(y_1, \ldots, y_n)$, $\hat{\theta}_n = \hat{y}_{n+1}|D_n$ is the adversary’s estimate of $y_{n+1}$ using $D_n$ and is Gaussian distributed. The mean and variance of this estimate can be computed as follows:

$$\mathbb{E}[\hat{\theta}_n] = \mathbb{E}[k_n^T K_n^{-1} y_{1:n}] = k_n^T K_n^{-1} f_{1:n}$$

$$\text{Var}[\hat{\theta}_n] = \mathbb{E} \left[ (\hat{\theta}_n)^2 \right] - \left( \mathbb{E}[\hat{\theta}_n] \right)^2 = \sigma^2 \|k_n^T K_n^{-1}\|_2^2$$

where $k_n$, $K_n$ and $\sigma$ are the quantities introduced in Chapter 2. Using the mean and variance of $\hat{\theta}_n$, we can compute the distribution of $\hat{\omega}_n$. Defining $\mu_n = k_n^T K_n^{-1} f_{1:n}$ and $\sigma_n = \sigma^2 \|k_n^T K_n^{-1}\|_2^2$, the probability density function of $\hat{\omega}_n$ can be written as:

$$p_{\hat{\omega}_n}(y) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma_n^2} \exp \left[ -\frac{(y-\mu_n)^2}{2\sigma_n^2} \right] & \text{if } y > y_n^+ \\ Pr(y_{n+1} \leq y|D_n) & \text{if } y = y_n^+ \\ 0 & \text{if } y < y_n^+ \end{cases}$$

The mean square error $E_n$ can be written as follows:
5.3. Privacy preserving Bayesian optimisation

\[ E_n = \mathbb{E} \left[ (\hat{\omega}_n - y_{n+1})^2 \right] \]
\[ = (\mathbb{E} [\hat{\omega}_n] - y_{n+1})^2 + \text{Var} [\hat{\omega}_n] \]  \hspace{1cm} (5.6)

Given the distribution function \( p_{\hat{\omega}_n}(y) \), we can compute the expectation and variance of \( \hat{\omega}_n \) as follows:

\[ \mathbb{E} [\hat{\omega}_n] = \sigma_n p + (y_n^+ - \mu_n) P + \mu_n \]  \hspace{1cm} (5.7)

\[ \text{Var} [\hat{\omega}_n] = \mathbb{E} \left[ (\hat{\omega}_n)^2 \right] - (\mathbb{E} [\hat{\omega}_n])^2 \]
\[ = (y_n^+ + \mu_n) \sigma_n p + \left( (y_n^+)^2 - \sigma_n^2 - \mu_n^2 \right) P + \mu_n^2 + \sigma_n^2 - (\mathbb{E} [\hat{\omega}_n])^2 \]  \hspace{1cm} (5.8)

where \( p = \phi \left( \frac{y_n^+ - \mu_n}{\sigma_n} \right) \) and \( P = \Phi \left( \frac{y_n^+ - \mu_n}{\sigma_n} \right) \). Using Eq. (5.7) and Eq. (5.8), we can finally compute \( E_n \).

**Computation of \( E_{n+1} \):** After the adversary (or optimiser) suggests \( x_{n+1} \), the experimenter conducts the experiment and returns a noisy value \( y_{n+1} \leftarrow y_{n+1} + \nu_{n+1} \) to ensure privacy, where \( \nu_{n+1} \sim \mathcal{N}(0, q_{n+1}^2) \). We assume that after receiving \( y_{n+1} \), the adversary uses maximum statistic to estimate \( y_{n+1}^+ \) using \( D_{n+1} \) as follows:

\[ \hat{\omega}_{n+1} = \max(\hat{\theta}_{n+1}, y_{n+1}^+) \]  \hspace{1cm} (5.9)

where \( \hat{\theta}_n = \hat{g}_{n+1} | D_{n+1} \) is the adversary’s estimate of \( y_{n+1} \) using \( D_{n+1} \). Similar to the previous derivation, the mean and variance of \( \hat{\theta}_{n+1} \) can be computed as follows:

\[ \mathbb{E} [\hat{\theta}_{n+1}] = k_{n+1}^T K_{n+1}^{-1} f_{1:n+1} \]  \hspace{1cm} (5.10)

\[ \text{Var} [\hat{\theta}_{n+1}] = \sigma_\epsilon^2 \left\| k_{n+1}^T K_{n+1}^{-1} \right\|_2^2 + \gamma_{n+1} \left( \sigma_\epsilon^2 + q_{n+1}^2 \right) \]  \hspace{1cm} (5.11)

where \( \gamma = k_{n+1}^T K_{n+1}^{-1} \) is a vector and \( \gamma_{n+1} \) is the \( n+1 \)-th element of \( \gamma \). Let us define \( \mu_{n+1} = \mathbb{E} [\hat{\theta}_{n+1}] \) and \( \sigma_{n+1} = \text{Var} [\hat{\theta}_{n+1}] \). The error \( E_{n+1} \) can be derived as:

\[ E_{n+1} = \mathbb{E} \left[ (\hat{\omega}_{n+1} - y_{n+1})^2 \right] \]
\[ = (\mathbb{E} [\hat{\omega}_{n+1}] - y_{n+1})^2 + \text{Var} [\hat{\omega}_{n+1}] \]  \hspace{1cm} (5.12)
Figure 5.3: The ratio $\frac{E_{n+1}}{E_n}$ versus $q_{n+1}$. We set $q_{n+1} = \alpha$ so that $\frac{E_{n+1}}{E_n} \geq \exp(-\epsilon)$.

where $p' = \phi \left( \frac{y_{n+1}^+ - \mu_{n+1}}{\sigma_{n+1}} \right)$, $P' = \Phi \left( \frac{y_{n+1}^+ - \mu_{n+1}}{\sigma_{n+1}} \right)$ and

$$\mathbb{E} [\hat{\omega}_{n+1}] = \sigma_{n+1} p' + (y_{n+1}^+ - \mu_{n+1}) P' + \mu_{n+1}$$

(5.13)

$$\text{Var} [\hat{\theta}_{n+1}] = \left( y_{n+1}^+ + \mu_{n+1} \right) \sigma_{n+1} p' + \left( (y_{n+1}^+)^2 - \sigma_{n+1}^2 - \mu_{n+1}^2 \right) P' + \mu_{n+1}^2 + \sigma_{n+1}^2 - (\mathbb{E} [\hat{\omega}_{n+1}])^2$$

(5.14)

**Computation of $q_{n+1}$:** Now we have computed the errors of the adversary estimates $\mathcal{E}_n$ and $\mathcal{E}_{n+1}$. To ensure privacy condition while maintaining high utility, we want to add a smallest noise possible that make the following inequality satisfied:

$$\frac{\mathcal{E}_{n+1}}{\mathcal{E}_n} = \frac{(\mathbb{E} [\hat{\omega}_{n+1}] - y_{n+1})^2 + \text{Var} [\hat{\omega}_{n+1}]}{(\mathbb{E} [\hat{\omega}_n] - y_{n+1})^2 + \text{Var} [\hat{\omega}_n]} \geq \exp(-\epsilon)$$

(5.15)

Eq. (5.15) can be solved by plugging Eq. (5.7), Eq. (5.8), Eq. (5.13) and Eq. (5.14). Our objective is to find a smallest value of $q_{n+1}$ that satisfies Eq. (5.15). We note that by adding more noise, the variance of $\hat{\omega}_{n+1}$ will increase and hence $\frac{\mathcal{E}_{n+1}}{\mathcal{E}_n}$ is an increasing function of $q_{n+1}$. Fig. 5.3 shows the ratio $\frac{\mathcal{E}_{n+1}}{\mathcal{E}_n}$ as a function of $q_{n+1}$. We can find the smallest value $\alpha$ of $q_{n+1}$ that satisfies (5.15) using binary search. Assigning $q_{n+1} \leftarrow \alpha$, we can then add a noise sample $v_{n+1}$ to $y_{n+1}$ and keep the “best point so far” private from the adversary. Our algorithm is summarised in Algorithm 5.1. We refer to this algorithm as **Error Preserving Private Bayesian Optimisation** (EPP-BO).
Algorithm 5.1 Error Preserving Private Bayesian Optimisation (EPP-BO)

1: Input:
2: Initial observation set $D_{n_0} = \{x_1:n_0, y_1:n_0\}$, search space $X$, privacy budget $\epsilon$.
3: Output: $\{x_n, y_n\}_{n=1}^T$
4: for $n = n_0+1, \ldots, T$
5: Evaluate target function $y_n = f(x_n)$
6: Find smallest $q_n$ that satisfies (5.15) using binary search.
7: Add a noise $\nu_n \sim \mathcal{N}(0, q_n^2)$ to $y_n$.
8: Return the output to the optimiser.
9: end for

Discussion of differentially private Bayesian optimisation

Kusner et al. (2015) proposed a Differentially Private Bayesian Optimisation algorithm. This algorithm was designed to offer privacy in a setting where the optimiser was considered trusted and the experimenter can share all the data with the optimiser. Since our privacy setting is different, we cannot use the algorithm proposed in (Kusner et al., 2015). In the following we suggest another differentially-private Bayesian optimisation algorithm, which we’ll use for comparison in our experiments.

DP-BO baseline: We suggest a privacy preserving Bayesian optimisation algorithm under differential privacy framework using Laplacian mechanism (Dwork, 2006). Laplacian mechanism adds a perturbation noise to the output of the algorithm to achieve the required privacy. The amount of noise depends on the sensitivity of the quantity that needs to be protected. Since by releasing $y_{n+1}$, the maximum possible change in $y_{n+1}^2$ is $S = \|y_{max} - y_{min}\|$ where $y_{max}$ and $y_{min}$ are the maximum and the minimum possible value of $y$ respectively. Using the sensitivity, we iteratively add a Laplacian noise to the function output before passing it to the optimiser: $y_{n+1} \leftarrow y_{n+1} + \nu_{n+1}^{DP}$ where $\nu_{n+1}^{DP} \sim \text{Lap}(S/\epsilon_{DP})$ and $\epsilon_{DP}$ is the privacy budget for differential privacy. We refer to this differentially private Bayesian optimisation algorithm as DP-BO. When the quantities $y_{max}$ and $y_{min}$ are not known exactly, it may be possible to estimate them using Lipschitz smoothness where possible, otherwise this algorithm may not be usable.
5.3.2 Experiments

We experiment the proposed privacy preserving Bayesian optimisation algorithm on several benchmark optimisation problems as well as optimisation problems from real-world industrial processes and demonstrate the optimisation efficiency of our algorithm by comparing it with various baselines. We use the following three baselines:

- **Non-private Bayesian Optimisation (Non-private BO):** This is the standard non-private version of Bayesian optimisation. We use this algorithm to show the ultimate utility of private Bayesian optimisation algorithm.

- **Random Search:** This algorithm is used as a lower bound baseline. Since random search provides privacy guarantee by its nature, a private algorithm must achieve higher optimisation efficiency than random search to justify the extra complexity.

- **Differentially Private Bayesian Optimisation (DP-BO):** This baseline algorithm is a Bayesian optimisation algorithm constructed under differential privacy framework as discussed in Section 5.3.1.

For all our experiments, we use GP-UCB as the acquisition function and squared exponential kernel for Gaussian process. For global optimisation of acquisition function, we use DIRECT algorithm. We set the privacy leakage parameter $\epsilon = 0.1$. All hyperparameters are kept same across all algorithms for fair comparison.

**Experiments with benchmark functions**

To demonstrate the proposed EPP-BO algorithm for a variety of functions in different number of input dimensions, we experiment with four popular benchmark functions: Branin 2D, Rosenbrock 4D, Hartmann 4D and Hartmann 6D. The optimisation results are averaged over 20 different initialisations. Fig 5.4 shows the performance of EPP-BO against the baselines on the benchmark functions. For all four benchmark functions, EPP-BO’s performance is close to the Non-private Bayesian Optimisation and clearly outperforms both Random Search and DP-BO by a significant margin. We also note that the performance of the Random Search
is better than that of DP-BO due to huge amount of perturbation required in DP-BO. In optimisation, the ultimate objective is to minimise the number of iterations to find the optimum. These experimental results using the benchmark functions demonstrate that EPP-BO is able to find the optimum faster than Random Search and DP-BO.

Fig. 5.5 illustrates the illusion our proposed EPP-BO creates for an adversary for Hartmann 4D function. We show two scenarios: high privacy scenario (using \( \epsilon = 0.1 \)) and low privacy scenario (using \( \epsilon = 0.5 \)). From any run of EPP-BO, two different graphs are extracted showing different views of the optimisation from experimenter or optimiser (adversary) perspectives. The graph in ‘magenta’ colour shows the best function value achieved so far from the optimiser’s perspective. The graph in ‘blue’ colour shows the best function value achieved so far from the experimenter’s perspective. Between these graphs, the locations of the best points do not necessarily coincide. We also note that when the privacy decreases, the optimiser gets closer to the true optimum since less noise is added.
5.3. Privacy preserving Bayesian optimisation

Figure 5.5: Comparison between the best values achieved by the experimenter and the optimiser with respect to iterations on Hartman 4D function with different level of privacy: a) $\epsilon = 0.1$ (high privacy) and b) $\epsilon = 0.5$ (low privacy). The blue and the magenta show the best function value achieved so far from the experimenter’s perspective and optimiser’s perspective, respectively.
5.3. Privacy preserving Bayesian optimisation

Experiments with real datasets

**Alloy heat treatment:** This dataset is a simulation model of an Al-Sc alloy heat treatment process. The strengthening process of an alloy involves nucleation and growth. During nucleation, new “phase” is formed through a self-organising process of clusters of atoms. This process happens at low temperature. Following nucleation step, the growth step archives the requisite alloy property through diffusion. The industrial standard precipitation KWN model (Wagner et al., 1991) is used for nucleation and growth. This model consists of multiple stages, each of them having two parameters: temperature and time. The output quality of the alloy heat treatment process is measured by the hardness of the alloy. Our objective is to find the best combination of time and temperature parameters that achieve the highest level of hardness in few iterations.

Fig. 5.6a demonstrates the results of our experiments on Alloy heat treatment dataset averaged over 20 different initialisations. After 30 iterations, the best hardness achieved by EPP-BO is comparable to Non-private BO and higher than the best results of both Random Search and DP-BO.

**Short polymer fibre production:** The short polymer fibre production dataset was collected in a collaboration with material scientists from Deakin University. For short polymer fibre production, a particular geometric manifold is used to mix polymer rich fluid with the flow of another solvent. This manifold has 5 different parameters: device position, constriction angle, channel width, polymer flow, and coagulant speed. The objective is to find the best manifold parameter set to maximise a combine utility measured by the length and diameter of the output polymer.

Fig. 5.6b shows the experimental results on Short polymer fibre production dataset averaged over 40 different random initialisations. After 20 iterations, the best utility achieved by our method is just under the best utility of Non-private BO and clearly higher than both DP-BO and Random Search.
5.3. Privacy preserving Bayesian optimisation

![Graph](image)

(a) Alloy heat treatment

![Graph](image)

(b) Short polymer fibre production

Figure 5.6: Optimisation results for two real datasets.
5.4 Application of EPP framework for other machine learning algorithms

The proposed EPP framework can be applied to many other machine learning algorithms. To use EPP framework for preserving privacy, one needs to know the model that the adversary uses to attack. Given this model, we can derive the expected errors in the adversary estimates. Using the estimates, we can find the perturbation required to satisfy (5.1), making the algorithm private. To demonstrate the applicability of EPP to machine learning algorithms, in this section, we use EPP to protect the privacy of a popular machine learning algorithm: K-means clustering.

5.4.1 EPP based K-means clustering

Given the dataset $D_n$, the K-means clustering algorithm aims to partition $D_n$ into $K$ disjoint sets $\{C_1, C_2, ..., C_K\}$ by minimising the following cost function:

$$
\min_{C_1,...,C_K} \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - m_k\|^2
$$

(5.16)

where $m_k$ is the centroid of cluster $C_k$. The most popular algorithm for K-means clustering is due to Lloyd (Lloyd, 1982). This algorithm first randomly picks $K$ data points and uses them to initialise the centroids $m_1, m_2, ..., m_K$. Using these centroids, the algorithm assigns a data point $x_i$ to cluster $C_k$ if $m_k$ is the nearest centroid. After this assignment, each centroid $m_k$ is re-computed by averaging all data points that belong to cluster $C_k$. The algorithm is iterated between these two steps until it converges or exceeds the maximum number of iterations.

We propose a new privacy preserving K-means algorithm that can cluster the data while maintaining the data privacy under our proposed EPP framework in (5.1). Let $D_n \setminus r$ be a dataset that has all the data points of $D_n$ except a data point $x_r$. Without loss of generality, let us assume that the adversary’s goal is to learn about $x_r$. Let $h(D_n)$ be the statistical query result of database $D_n$. Let $E_{inc}(\hat{x}_{rj})$ be the error in the adversary estimate of the $j$-attribute of $x_r$ using $h(D_n)$ (i.e. the query result of
the database that contains \( x_r \). Similarly, let us denote \( \mathcal{E}_{\text{exc}}(\hat{x}_{rj}) \) as the error in the adversary estimate of the \( j \)-attribute of \( x_r \) using \( h(D_{n\setminus r}) \) (i.e. the query result of database that does not contain \( x_r \)). Our algorithm aim to protect the privacy of a data point \( x_r \) even when an adversary has access to data points in \( D_{n\setminus r} \). In other words, using EPP framework, our privacy preserving K-means algorithm ensures the following inequality:

\[
\frac{\mathcal{E}_{\text{inc}}(\hat{x}_{rj})}{\mathcal{E}_{\text{exc}}(\hat{x}_{rj})} \geq \exp(-\epsilon) .
\]  

(5.17)

The key to achieving privacy is to use a randomisation in the answer of the query such that the inequality in (5.17) is satisfied. In doing so, our effort should be to use a mechanism for the randomisation that does not degrade the utility of the answer for intended tasks. Motivated by this idea, we use a mechanism that is based on bootstrap sampling (Efron and Tibshirani, 1994) of data points. The proposed mechanism not only offers the desired randomness but also retains the high utility of the original algorithm.

Similar to the Lloyd’s algorithm, our algorithm iterates between the two steps of data assignment to cluster centroids and centroid re-computation until no improvement can be made. However, in the last iteration of our algorithm, the centroids are estimated using bootstrap aggregation (bagging) (Efron and Tibshirani, 1994). For each cluster, it generates a bag of data points through bootstrap sampling, i.e. uniformly randomly sampling of data points with replacement. The number of data points in each bag remains same as that in the original cluster. For each bag, the centroid is estimated by averaging the data points. A total of \( B \) such bags are generated and the aggregate centroid is computed by averaging the centroid estimates of all \( B \) bags. A step-by-step summary of our proposed algorithm is provided in Algorithm 5.2.

In the following analysis, we present a theoretical analysis of our algorithm showing that as long as the number of bags \( B \) in the bootstrap aggregation are smaller than a certain upper bound, the privacy of the algorithm is maintained under the framework of (5.1). This means given the bootstrap-perturbed cluster centroids and the data points except \( x_r \), the adversary can not estimate \( x_r \) significantly better than an estimate made by using the centroids that were computed without \( x_r \). We refer to this model as Error Preserving Private K-means (EPP-KM).
The analysis of EPP K-means

Due to the randomness of bootstrapping, the adversary’s estimate of unknown data point \( x_r \) is perturbed. In this section, we theoretically analyse the proposed model in the light of the adversary estimation of the unknown point. In general, we have the two possible cases: ‘the adversary knows which cluster the unknown data point belongs to’ or ‘otherwise’.

**Case-1 (The adversary knows which cluster \( x_r \) belongs to):** Let us assume that the adversary knows that \( x_r \in C_k \). Let us denote by \( n_k \) the number of data points in the cluster \( C_k \) and let \( x_{ij} \) be the \( j \)-th attribute value of a data point \( x_i \in C_k \). Using the centroid \( m_k \) and other data points of \( C_k \), the best estimate of \( x_{rj} \) is given by:

\[
\hat{x}_{rj} = n_k \times m_{kj} - \sum_{x_i \in C_k \setminus x_r} x_{ij}.
\]

(5.18)

where \( m_{kj} \) is the \( j \)-th attribute of the centroid \( m_k \). When the \( m_{kj} \) is estimated using bagging, it is a random variable. We will show that this randomness is used to preserve the privacy of \( x_{rj} \). In (5.18), \( n_k \) and the sum of attributes are already known. Thus, the variance of the estimation error of \( \hat{x}_{rj} \) is given by:

\[
\mathcal{E}_{inc}(\hat{x}_{rj}|D_{n\setminus r}, m_k, z_r = k) = n_k^2 \text{Var}(m_{kj}|D_{n\setminus r}),
\]

(5.19)

where the cluster indicator variable \( z_r = k \) encodes the knowledge \( x_r \in C_k \). Because of the bagging ensemble used in our privacy preserving algorithm, \( m_{kj} \) is given by:

\[
m_{kj} = \frac{1}{B} \times \frac{1}{n_k} \times \sum_{x_r \in C_k} \alpha_r x_{rj},
\]

where \( \alpha_r \) denotes the number of times \( x_r \) is sampled in \( B \) bags of bootstrap during the computation of \( m_k \). Clearly, \( \alpha_r \) is a random variable following a binomial distribution with mean \( B \) and variance \( B(1 - \frac{1}{n_k}) \). Therefore, the conditional variance of \( m_{kj} \) is:

\[
\text{Var}(m_{kj}|D_{n\setminus r}) = \frac{\text{Var}(\alpha_r)}{B^2 n_k^2} \left( \sum_{x_r \in C_k} x_{rj}^2 \right) = \frac{1}{B n_k^2} (1 - \frac{1}{n_k}) \left( \sum_{x_r \in C_k} x_{rj}^2 \right).
\]

(5.20)
Plugging (5.20) in (5.19), we have
\[ \mathcal{E}_{\text{inc}}(\hat{x}_{rj}|D_{n\backslash r}, m_k, z_r = k) = \frac{1}{B} \left( 1 - \frac{1}{n_k} \right) \left( \sum_{x_r \in C_k} x_{rj}^2 \right). \]
To ensure that this estimation error variance follows the privacy framework in (5.17), the number of bootstrap bags \( B \) has to satisfy
\[ B \leq \frac{(1 - \frac{1}{n_k}) \left( \sum_{x_r \in C_k} x_{rj}^2 \right)}{\mathcal{E}_{\text{exc}}(\hat{x}_{rj}) \times \exp(-\epsilon)}. \] (5.21)

The above bound is applicable to protect the \( j \)-th attribute of the data point \( x_r \). Since the framework is required to protect all the attributes of all the data points in the cluster, the following needs to be satisfied
\[ B \leq \min_j \frac{(1 - \frac{1}{n_k}) \left( \sum_{x_r \in C_k} x_{rj}^2 \right)}{\mathcal{E}_{\text{exc}}(\hat{x}_{rj}) \times \exp(-\epsilon)}. \] (5.22)

We refer to this case as \textbf{EPP-KM (1)}.

\textbf{Case-2 (The adversary doesn't know which cluster \( x_r \) belongs to)}: In this case, the adversary does not have the information of the cluster membership of \( x_r \). The unavailability of this information creates a bias in his estimation. To see this, consider the adversary model in (5.18). Assuming that \( x_r \) truly belongs to cluster \( k' \), the expectation of the adversary estimate is given as
\[ \mathbb{E}(\hat{x}_{rj}) = \mathbb{E}_{z_r}(\mathbb{E}(\hat{x}_{rj} | z_r)) = \pi_{k'} x_{rj} \]
where \( z_r \) is a random variable and \( z_r = k \) implies that \( x_r \) belongs to cluster \( C_k \). We use \( \pi_{k'} \) to denote the probability that \( x_r \) belongs to the cluster \( C_k' \). The probability \( \pi_{k'} \) can be approximately estimated using the partition of data \( D_{n\backslash r} \). Clearly, the estimate \( \hat{x}_{rj} \), in this case, is biased as \( \mathbb{E}(\hat{x}_{rj}) \neq x_{rj} \). The variance of the error in the estimation can be derived by \textit{the law of total variance} as below:
\[
\mathcal{E}_{\text{inc}}(\hat{x}_{rj}|D_{n\backslash r}, m_{1:K}) = \mathbb{E}_{z_r} \left[ \text{Var}(\hat{x}_{rj} | z_r, D_{n\backslash r}, m_{1:K}) \right] + \text{Var}_{z_r} \left[ \mathbb{E}(\hat{x}_{rj} | z_r, D_{n\backslash r}, m_{1:K}) \right] \\
= \sum_{k=1}^{K} \left[ \pi_k \left( 1 - \frac{1}{n_k} \right) \left( \sum_{x_r \in C_k} x_{rj}^2 \right) \right] + \pi_{k'} \left( 1 - \pi_{k'} \right) x_{rj}^2
\]
5.4. Application of EPP framework for other machine learning algorithms

To satisfy the privacy framework in (5.17), the number of bootstrap bags \( B \) has to satisfy

\[
B \leq \frac{\sum_{k=1}^{K} \pi_k \left( 1 - \frac{1}{n_k} \right) \left( \sum_{x_r \in C_k} x_{rj}^2 \right)}{\mathcal{E}_{\text{exc}}(\hat{x}_{rj}) \times \exp(-\epsilon) - \pi_{k'} (1 - \pi_{k'}) x_{rj}^2} \tag{5.23}
\]

Once again, since the above bound should be applicable to protect all the attributes of all the data points in the cluster, the following needs to be satisfied

\[
B \leq \min_{j, r} \frac{\sum_{k=1}^{K} \pi_k \left( 1 - \frac{1}{n_k} \right) \left( \sum_{x_r \in C_k} x_{rj}^2 \right)}{\mathcal{E}_{\text{exc}}(\hat{x}_{rj}) \times \exp(-\epsilon) - \pi_{k'} (1 - \pi_{k'}) x_{rj}^2} \tag{5.24}
\]

We refer to this case as EPP-KM (2).

**Algorithm 5.2** Error Preserving Private K-means algorithm

**Input:** Dataset \( D = \{x_1, ..., x_N\}, x_i \in \mathbb{R}^d \), number of clusters \( K \).

**Output:** The bootstrap estimated cluster centroids: \( m_1, ..., m_K \).

**Initialisation:** Randomly initialise the cluster centroids \( m_1, ..., m_K \).

1: repeat
2: for each point \( x_i \) do
3: if \( x_i \) is the closest to \( m_k \) out of all centroids \( m_1, ..., m_K \) then
4: Assign \( x_i \) to \( C_k \)
5: end if
6: end for
7: for \( k = 1 \) to \( K \) do
8: Compute \( m_k \) by averaging all \( x_i \in C_k \)
9: end for
10: until clustering converges
11: for \( k = 1 \) to \( K \) do
12: Calculate the value of \( B \) using (5.22) or (5.24) depending on if the adversary knows the cluster membership of data points or not.
13: Compute \( m_k \) using aggregation of \( B \) bootstrap samples.
14: end for

5.4.2 Experiments

We experiment with a total of three clustering datasets: one synthetic and two real datasets. Experiments with the synthetic data illustrate the behaviour of our proposed model in a controlled setting. Experiments with the real datasets show the effectiveness of our model for clustering under privacy constraints.
Baselines methods: To evaluate the efficacy of our model, we compare its performance with the following baseline methods:

- **The Original K-means (Non-Private):** This algorithm is the standard K-means algorithm. We note that this method does not protect privacy of database. We refer to this method as KM.

- **Differentially Private K-means:** This algorithm is a variant of K-means that protects the privacy of database under the framework of differential privacy (Dwork et al., 2006). In this algorithm, the $j$-th element of $k$-th K-means centroid is made $\epsilon$-differential private by adding to it a noise $\eta_{kj}$ that follows a Laplacian distribution with mean zero and standard deviation $S_{kj}/\epsilon$ where $S_{kj}$ is the sensitivity of the $j$-th element of the $k$-th centroid. The sensitivity $S_{kj}$ with respect to the presence/absence of any data point is approximately $\frac{1}{n_k} \max_r x_{rj}$, where $n_k$ is the number of data points in the $k$-th cluster. We refer to this method as DP-KM.

Performance measures: We use four different metrics for performance evaluation: Normalised Mutual Information (NMI) (Manning et al., 2008), Rand Index (Manning et al., 2008) and Purity (Manning et al., 2008) to evaluate the clustering performance, and Average Perturbation (AP) of privacy-preserving models to evaluate how much noise a model adds to the cluster centroids before releasing them for end use. The first three measures are widely used in clustering literature. The last evaluation measure is a normalised version of mean absolute error (MAE). Given $K$ clusters with the original centroids $\{m_k\}_{k=1}^K$ and the perturbed centroids $\{m'_k\}_{k=1}^K$, the average perturbation is calculated as

$$AP = \frac{1}{K} \sum_k \frac{\|m_k - m'_k\|}{\|m_k\|}.$$

Experimental setting: For both synthetic and real data experiments, the clustering performance of each algorithm is studied with respect to varying privacy levels ($\epsilon$) and the number of data points in the database. For the experiments showing clustering performance with respect to $\epsilon$, we average the performance of each algorithm for 30 random centroid initialisations for each value of $\epsilon$. For the experiments showing clustering performance with respect to varying number of data points ($n$), we vary from 25% to 100% of the data set size at a step of 25%.
5.4. Application of EPP framework for other machine learning algorithms

average performance is reported over 40 different random subsamples of size $n$ and 20 random centroid initialisations. To demonstrate the privacy guarantee of the proposed model, we estimate every data point in the database using the perturbed means and the adversary model in Eq. (5.18). We report the ratio of the estimation errors made by the adversary under presence/absence of the data points in the database as per our EPP framework (see Eq. (5.17)).

Experiments with synthetic data

We generate a synthetic data with 3 clusters in a 2-dimensional space. The centroids of these clusters are at $[0,0]$, $[5,0]$ and $[4,4]$. For each cluster, we generate 60 random data points from a bi-variate Gaussian distribution with its mean at the cluster centroid and a standard deviation of 1 along each dimension. Our goal is to illustrate the behaviour of the proposed model in terms of its clustering utility and privacy guarantees.

Fig. 5.7 shows the experimental results for the synthetic dataset. Fig. 5.7a compares the two cases of the proposed model with DP-KM in terms of average perturbation. As seen from the figure, DP-KM has much higher amount of perturbation compared to both EPP-KM (1) and EPP-KM (2) when $\epsilon$ is small. Fig. 5.7b compares the proposed models with original K-means (KM) and DP-KM in terms of NMI score with respect to increasing values of $\epsilon$. The NMI score of KM is the highest. This is not surprising as this method does not perturb the centroids and thus does not offer any privacy. However, it is interesting to note that the NMI scores of EPP-KM methods are not very different from that of KM in spite of the strong privacy guarantees offered by EPP-KM. On the other hand, DP-KM performs poorly as its NMI scores are significantly lower compared to the other methods. This poor performance of DP-KM is evident from the high levels of perturbations made by this algorithm to the cluster centroids. In Fig. 5.7c, we demonstrate the privacy guarantee offered by EPP-KM models. As seen from the figure, the variance of the error in an adversary's estimation for any data point changes by a factor of only $\exp(-\epsilon)$ due to its participation in the database. We can see that for low values of $\epsilon$, e.g. when $\epsilon = 0.001$, the ratio of the error variance in the adversary's estimation is around 1, meaning that no extra reduction in uncertainty is achieved.
by the adversary. At the other values of $\epsilon$, the plot follows the EPP framework of Eq. (5.17). We also study the effect of the number of data points in the database on the clustering performance. Fig. 5.7d compares the NMI score of the proposed models with KM and DP-KM. For this experiment, the privacy parameter $\epsilon$ is fixed at 0.1. The performance of all the algorithms improve with the number of data points due to reduction in the perturbation. The NMI scores of EPP-KM variants are close to that of KM. Once again the performance of DP-KM variants are close to that of KM. Once again the performance of DP-KM is poor in the beginning as it needs high perturbations due to small cluster size.

![Graphs showing AP, NMI, Variance Ratio, and NMI for varying number of data points at $\epsilon = 0.1$.](image)

Figure 5.7: Results using Synthetic dataset with $N = 180, K = 3$. (a) Average perturbation in cluster centroid with respect to $\epsilon$, (b) NMI with respect to $\epsilon$, (c) Ratio of variance for estimation errors $\mathcal{E}_{\text{inc}}$ and $\mathcal{E}_{\text{exc}}$, (d) NMI for varying number of data points at $\epsilon = 0.1$.

**Experiments with real data**

We use the following datasets from UCI machine learning repository\(^1\):

\(^1\)available at URL https://archive.ics.uci.edu/ml/datasets.html
5.4. Application of EPP framework for other machine learning algorithms

- **Seeds dataset**: This dataset consists of 210 data points of three wheat types: *Kama*, *Rosa* and *Canadian*. Each data point has 7 geometric attributes of wheat kernels: area, perimeter, compactness, length of kernel, width of kernel, asymmetry coefficient, length of kernel groove. Our task is to use these attributes to cluster the data points in 3 different categories.

- **User Knowledge Modelling dataset (UKM)**: The dataset is about student’s knowledge level about a subject of Electrical DC Machines. There are 4 levels of knowledge: Very Low, Low, Middle, High. The UKM dataset has 258 data points and each data point has 5 attributes: STG, SCG, STR, LPR, PEG. Our task is to use these attributes to cluster the data points in 4 different categories.

The experimental results with the Seeds dataset and the UKM dataset are shown in Fig. 5.8 and Fig. 5.9 respectively. The results follow similar patterns as in the Synthetic dataset. As seen from Figs. 5.8a and 5.9a, the average perturbations used in the centroids by both the proposed EPP-KM variants are quite small. In contrast, the average perturbation by DP-KM is extremely high for small values of $\epsilon$. The NMI performance of the proposed EPP-KM models with respect to $\epsilon$ is approximately 0.7 and 0.3, which is close to that of KM (see Figs. 5.8b and 5.9b) while the performance of DP-KM is extremely poor at small values of $\epsilon$ and only improves at higher values of $\epsilon$. Similar to the Synthetic dataset, Figs. 5.8c and 5.9c demonstrate that the adversary gains almost no extra information about any data point at small values of $\epsilon$ (at strict privacy).
5.4. Application of EPP framework for other machine learning algorithms

Figure 5.8: Results using Seeds dataset with \( n = 210, K = 3 \), (a) Average perturbation in cluster centroid with respect to \( \epsilon \), (b) NMI with respect to \( \epsilon \), (c) Ratio of variance for estimation errors \( \mathcal{E}_{\text{inc}} \) and \( \mathcal{E}_{\text{exc}} \), (d) NMI for varying number of data points at \( \epsilon = 0.1 \).
5.4. Application of EPP framework for other machine learning algorithms

We also study the effect of the number of data points in the database on the clustering performance. From Figs. 5.8d and 5.9d we can see that the NMI scores of both EPP-KM variants are almost same as that of KM. On the contrary, the performance of DP-KM is quite poor as when using 25% fraction of data points, NMI score of DP-KM drops to as low as 0.54 and 0.17 for Seeds and UKM dataset respectively.

A more complete set of results showing other clustering measures, in particular, Purity and Rand Index are reported in Table 5.1. As seen from the Table, both EPP-KM variants consistently achieve high level of clustering performance in terms of all three evaluation metrics. At times, we observed that the performances of EPP-KM (2) were slightly better than even KM. After further investigation, we found that this happens due to the robustness of bootstrap sampling to outliers (Salibian-Barrera and Zamar, 2002).
5.4. Application of EPP framework for other machine learning algorithms

<table>
<thead>
<tr>
<th></th>
<th>Synthetic</th>
<th>Seeds</th>
<th>UKM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NMI</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KM</td>
<td>0.9152</td>
<td>0.7010</td>
<td>0.2778</td>
</tr>
<tr>
<td>(1)</td>
<td>(0.0128)</td>
<td>(0.0014)</td>
<td>(0.0107)</td>
</tr>
<tr>
<td>EPP-KM (1)</td>
<td>0.9160</td>
<td><strong>0.7017</strong></td>
<td>0.2781</td>
</tr>
<tr>
<td>(2)</td>
<td>(0.0128)</td>
<td>(0.0016)</td>
<td>(0.0109)</td>
</tr>
<tr>
<td>EPP-KM (2)</td>
<td><strong>0.9162</strong></td>
<td>0.7010</td>
<td><strong>0.2790</strong></td>
</tr>
<tr>
<td>(1)</td>
<td>(0.0128)</td>
<td>(0.0014)</td>
<td>(0.0106)</td>
</tr>
<tr>
<td>DP-KM</td>
<td>0.8514</td>
<td>0.6709</td>
<td>0.2534</td>
</tr>
<tr>
<td>(1)</td>
<td>(0.0192)</td>
<td>(0.0064)</td>
<td>(0.0108)</td>
</tr>
<tr>
<td><strong>Purity</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KM</td>
<td>0.9707</td>
<td>0.8933</td>
<td>0.5683</td>
</tr>
<tr>
<td>(1)</td>
<td>(0.0112)</td>
<td>(0.0004)</td>
<td>(0.0078)</td>
</tr>
<tr>
<td>EPP-KM (1)</td>
<td>0.9709</td>
<td><strong>0.8938</strong></td>
<td>0.5686</td>
</tr>
<tr>
<td>(2)</td>
<td>(0.0110)</td>
<td>(0.0004)</td>
<td>(0.0080)</td>
</tr>
<tr>
<td>EPP-KM (2)</td>
<td><strong>0.9711</strong></td>
<td>0.8933</td>
<td><strong>0.5691</strong></td>
</tr>
<tr>
<td>(1)</td>
<td>(0.0112)</td>
<td>(0.0004)</td>
<td>(0.0078)</td>
</tr>
<tr>
<td>DP-KM</td>
<td>0.9446</td>
<td>0.8759</td>
<td>0.5536</td>
</tr>
<tr>
<td>(1)</td>
<td>(0.0128)</td>
<td>(0.0047)</td>
<td>(0.0062)</td>
</tr>
<tr>
<td><strong>Rand Index</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KM</td>
<td>0.9669</td>
<td>0.8732</td>
<td>0.6819</td>
</tr>
<tr>
<td>(1)</td>
<td>(0.0093)</td>
<td>(0.0003)</td>
<td>(0.0033)</td>
</tr>
<tr>
<td>EPP-KM (1)</td>
<td>0.9672</td>
<td><strong>0.8736</strong></td>
<td>0.6820</td>
</tr>
<tr>
<td>(2)</td>
<td>(0.0090)</td>
<td>(0.0003)</td>
<td>(0.0033)</td>
</tr>
<tr>
<td>EPP-KM (2)</td>
<td><strong>0.9674</strong></td>
<td>0.8732</td>
<td><strong>0.6823</strong></td>
</tr>
<tr>
<td>(1)</td>
<td>(0.0093)</td>
<td>(0.0003)</td>
<td>(0.0033)</td>
</tr>
<tr>
<td>DP-KM</td>
<td>0.9368</td>
<td>0.8559</td>
<td>0.6642</td>
</tr>
<tr>
<td>(1)</td>
<td>(0.0115)</td>
<td>(0.0041)</td>
<td>(0.0046)</td>
</tr>
<tr>
<td><strong>Average Perturbation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EPP-KM (1)</td>
<td>0.0108</td>
<td>0.0131</td>
<td>0.0210</td>
</tr>
<tr>
<td>(2)</td>
<td>(0.0007)</td>
<td>(0.0008)</td>
<td>(0.0008)</td>
</tr>
<tr>
<td>EPP-KM (2)</td>
<td><strong>0.0075</strong></td>
<td><strong>0.0071</strong></td>
<td><strong>0.0183</strong></td>
</tr>
<tr>
<td>(1)</td>
<td>(0.0004)</td>
<td>(0.0004)</td>
<td>(0.0007)</td>
</tr>
<tr>
<td>DP-KM</td>
<td>0.3275</td>
<td>0.2811</td>
<td>0.7059</td>
</tr>
<tr>
<td>(1)</td>
<td>(0.0227)</td>
<td>(0.0176)</td>
<td>(0.0433)</td>
</tr>
</tbody>
</table>

Table 5.1: Comparison with the baselines in terms of various metrics at $\epsilon = 0.1$. Average results over 30 random centroid initialisations are reported with the standard errors in parenthesis. The bold face indicates the best results among private algorithms.
5.5 Summary

We proposed a novel privacy preserving data mining framework for Bayesian optimisation algorithm to protect the optimum of the objective function. The proposed framework provides strong privacy guarantees even when an adversary has access to auxiliary knowledge about the database. The proposed Bayesian optimisation algorithm appropriately perturbs the output of experiment to ensure the privacy of the optimum. We theoretically analyse our method and derive the minimum amount of noise required to ensure privacy under Error Privacy Preserving framework. The experimental results clearly show that our algorithm has higher utility compared to the baselines.

We also developed a K-means clustering algorithm under the privacy preserving framework. Our private K-means algorithm calculates cluster centroids using bootstrap aggregation, which introduces just enough perturbation to ensure that privacy of every data point is maintained. We theoretically analyse our method and derive bounds on the size of bootstrap ensemble that ensures the privacy under the proposed framework. The experimental results clearly show that our algorithm has high utility with strong privacy guarantees.
Chapter 6

Conclusion

6.1 Summary

In this thesis, we have addressed several challenges of applying Bayesian optimisation in practical settings. We have emphasised Bayesian optimisation as an efficient method for optimising expensive black-box functions, often found in real world applications. We reviewed Bayesian optimisation and current advancement in Bayesian optimisation research (chapter 2). The main contributions of this thesis are three-fold. First, we proposed a stable Bayesian optimisation framework (chapter 3). This framework is able to find stable solutions (i.e. wide peaks) in optimisation problems that may also have narrow peaks. We then presented a Bayesian optimisation algorithm for cascaded structure optimisation that is common in industrial processes (chapter 4). This algorithm can utilise the cascaded structure and find the optimal design efficiently by avoiding high dimensional optimisation. Finally, we proposed a privacy preserving Bayesian optimisation algorithm that can keep the optimum solution private from an untrusted optimiser (chapter 5). Usually, privacy requirements lead to a drop in optimisation efficiency. To retain optimisation efficiency, we developed a new privacy framework called EPP. We showed wider applicability of the EPP framework by using it to construct a privacy aware K-means clustering algorithm (chapter 5).

Chapter 3 contributed a stable Bayesian optimisation framework aimed at finding
stable solutions for Bayesian optimisation. In this chapter, we discussed the notion of stability and used a modified Gaussian process model to measure stability via perturbed inputs. For stable Bayesian optimisation, we proposed two novel acquisition functions: STABLE-EI and STABLE-UCB that were constructed based on the epistemic and aleatoric variances of the modified Gaussian process posterior. Since the aleatoric variance is high in an unstable region (i.e. region around spurious narrow peaks), we incorporated them into the proposed acquisition functions to guide the function optimisation towards stable regions. We theoretically analysed our proposed acquisition functions and showed that they favour stable regions over unstable ones. We demonstrated the utility of our proposed framework through experiments with both synthetic function optimisation and hyperparameter tuning for SVM classifiers.

In chapter 4, we proposed a novel cascade Bayesian optimisation method to tune parameters of a multi-stage cascaded system, often encountered in industrial processes. In the proposed method, each stage of the cascaded process is separately modelled and appropriately adjusted such that the final stage output is optimised. We proposed a novel optimisation formulation that exploits the epistemic uncertainties of the underlying model. In addition, we also made the formulation cost-sensitive to find the cost-effective solutions in small number of experiments. We showed that the convergence rate of proposed cascade Bayesian optimisation algorithm is sub-linear in the number of evaluations $T$. We experimented the proposed algorithm by optimising two cascaded processes: multi-stage heat treatment of Al-Sc alloy via a simulated testbed, and hyperparameter tuning for a data analytic pipeline. In the experiment with the simulated testbed of 3-stage heat-treatment for Al-Sc alloy, the cascade Bayesian optimisation showed superior performance over the naïve joint approach. In addition, when the process’s cost is taken into account, we were able to save 30% of the total time with only slight drop in hardness value. In the experiment with the data analytic pipeline hyperparameter tuning, the systems tuned by the proposed cascade Bayesian optimisation algorithm had higher accuracy compared to the one tuned by the standard Bayesian optimisation approach.

Chapter 5 contributed a novel privacy preserving data mining framework for Bayesian optimisation algorithm to protect the optimum of the objective function. The proposed framework, named Error Privacy Preserving, provides strong privacy guarantees even when an adversary has access to auxiliary knowledge about the database.
Our proposed privacy preserving Bayesian optimisation algorithm appropriately perturbs the experimental output with noise to maintain the privacy of the optimum. We theoretically analysed our method and derived the bound on the amount of noise required to ensure privacy under the proposed privacy framework. We compared our proposed algorithm to various baselines on several benchmark optimisation problems and optimisation problems from real-world industrial processes. The experimental results clearly showed that our privacy preserving Bayesian optimisation algorithm had higher utility compared to the baselines. To show the applicability of our privacy framework, we also developed a K-means clustering algorithm. The private K-means algorithm perturbs the centroids by using bootstrap aggregation, which introduces just enough perturbation to ensure the privacy of every datapoint. We theoretically analysed our method and derived bounds on the size of bootstrap ensemble to ensure privacy under EPP. The experimental results on various datasets showed that the privacy preserving K-means algorithm had high utility with strong privacy guarantees.

6.2 Future directions

Overall, this thesis have addressed several practical challenges in Bayesian optimisation. The studies undertaken in this thesis suggest some potential directions for further developments. We list them as below:

- In this thesis, we have mainly focused on the stability for standard Bayesian optimisation. Another promising direction can be a stability framework for batch Bayesian optimisation, in which the evaluations are performed parallel in a batch. It would be interesting to see the role of the aleatoric variance in the acquisition functions of batch Bayesian optimisation. Stable batch Bayesian optimisation algorithm will enable us to achieve stable solutions in situations where large parallel processing facilities are available.

- In this thesis, we addressed the problem of optimising cascaded structure processes. One possible direction can be using Bayesian optimisation for more complex structures, such as loop structure. For example, in short polymer fibre production, a mixture of fluid is put in a continuous loop. The objective
6.2. Future directions

is to maximise the quality of the output fibre. Since the material mixture’s quality can be measured several times during the process, it would be interesting to see how we can take advantage of the additional information for Bayesian optimisation.

- The proposed privacy preserving Bayesian optimisation algorithm uses Gaussian process as the surrogate model. It would be interesting to extend the proposed algorithm to other surrogate models e.g. random forest. Due to bootstrap aggregation, random forest algorithm adds certain amount of perturbations to the prediction (Rana et al., 2015). Thus, privacy preserving Bayesian optimisation using random forest as the surrogate model can reduce the amount of perturbation, increasing the optimisation performance. Since EPP has two opposing parties, another interesting direction would be analysing EPP from a game theory perspective.
Bibliography


Albert Donally Bethke. Genetic algorithms as function optimizers. 1978.


Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable


Every reasonable effort has been made to acknowledge the owners of copyright material. I would be pleased to hear from any copyright owner who has been omitted or incorrectly acknowledged.
This Agreement between Mr. Thanh Nguyen ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

<table>
<thead>
<tr>
<th>License Number</th>
<th>4404741159217</th>
</tr>
</thead>
<tbody>
<tr>
<td>License date</td>
<td>Aug 09, 2018</td>
</tr>
<tr>
<td>Licensed Content Publisher</td>
<td>Springer Nature</td>
</tr>
<tr>
<td>Licensed Content Publication</td>
<td>Springer eBook</td>
</tr>
<tr>
<td>Licensed Content Title</td>
<td>Stable Bayesian Optimization</td>
</tr>
<tr>
<td>Licensed Content Author</td>
<td>Thanh Dai Nguyen, Sunil Gupta, Santu Rana et al</td>
</tr>
<tr>
<td>Licensed Content Date</td>
<td>Jan 1, 2017</td>
</tr>
<tr>
<td>Type of Use</td>
<td>Thesis/Dissertation</td>
</tr>
<tr>
<td>Requestor type</td>
<td>academic/university or research institute</td>
</tr>
<tr>
<td>Format</td>
<td>print and electronic</td>
</tr>
<tr>
<td>Portion</td>
<td>full article/chapter</td>
</tr>
<tr>
<td>Will you be translating?</td>
<td>no</td>
</tr>
<tr>
<td>Circulation/distribution</td>
<td>&lt;501</td>
</tr>
<tr>
<td>Author of this Springer Nature content</td>
<td>yes</td>
</tr>
<tr>
<td>Title</td>
<td>Mr</td>
</tr>
<tr>
<td>Instructor name</td>
<td>Thanh Dai Nguyen</td>
</tr>
<tr>
<td>Institution name</td>
<td>Deakin University</td>
</tr>
<tr>
<td>Expected presentation date</td>
<td>Oct 2018</td>
</tr>
<tr>
<td>Requestor Location</td>
<td>Mr. Thanh Nguyen</td>
</tr>
<tr>
<td></td>
<td>4/1A Villamanta Street</td>
</tr>
<tr>
<td></td>
<td>Geelong West, VIC 3218</td>
</tr>
<tr>
<td></td>
<td>Australia</td>
</tr>
<tr>
<td></td>
<td>Attn: Mr. Thanh Nguyen</td>
</tr>
<tr>
<td>Billing Type</td>
<td>Invoice</td>
</tr>
<tr>
<td>Billing Address</td>
<td>Mr. Thanh Nguyen</td>
</tr>
<tr>
<td></td>
<td>4/1A Villamanta Street</td>
</tr>
<tr>
<td></td>
<td>Geelong West, Australia 3218</td>
</tr>
<tr>
<td></td>
<td>Attn: Mr. Thanh Nguyen</td>
</tr>
<tr>
<td>Total</td>
<td>0.00 AUD</td>
</tr>
</tbody>
</table>

**Springer Nature Terms and Conditions for RightsLink Permissions**

Springer Customer Service Centre GmbH (the Licensor) hereby grants you a non-exclusive, world-wide licence to reproduce the material and for the purpose and
requirements specified in the attached copy of your order form, and for no other use, subject to the conditions below:

1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of this material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

2. Where **print only** permission has been granted for a fee, separate permission must be obtained for any additional electronic re-use.

3. Permission granted **free of charge** for material in print is also usually granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.

4. A licence for ‘post on a website’ is valid for 12 months from the licence date. This licence does not cover use of full text articles on websites.

5. Where ‘**reuse in a dissertation/thesis**' has been selected the following terms apply:
   Print rights for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).

6. Permission granted for books and journals is granted for the lifetime of the first edition and does not apply to second and subsequent editions (except where the first edition permission was granted free of charge or for signatories to the STM Permissions Guidelines http://www.stm-assoc.org/copyright-legal-affairs/permissions/permissions-guidelines/), and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence.

7. Rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com/bookpermissions@springernature.com for these rights.

8. The Licensor’s permission must be acknowledged next to the licensed material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book’s homepage. Our required acknowledgement format is in the Appendix below.

9. Use of the material for incidental promotional use, minor editing privileges (this does not include cropping, adapting, omitting material or any other changes that affect the meaning, intention or moral rights of the author) and copies for the disabled are permitted under this licence.

10. Minor adaptations of single figures (changes of format, colour and style) do not require the Licensor’s approval. However, the adaptation should be credited as shown in Appendix below.

**Appendix — Acknowledgements:**

**For Journal Content:**
Reprinted by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)
For Advance Online Publication papers:
Reprinted by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].)

For Adaptations/Translations:
Adapted/Translated by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)

Note: For any republication from the British Journal of Cancer, the following credit line style applies:

Reprinted/adapted/translated by permission from [the Licensor]: on behalf of Cancer Research UK: : [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)

For Advance Online Publication papers:
Reprinted by permission from The [the Licensor]: on behalf of Cancer Research UK: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM])

For Book content:
Reprinted/adapted by permission from [the Licensor]: [Book Publisher (e.g. Palgrave Macmillan, Springer etc) [Book Title] by [Book author(s)] [COPYRIGHT] (year of publication)

Other Conditions:

Version 1.0

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.
This Agreement between Mr. Thanh Nguyen ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

<table>
<thead>
<tr>
<th>License Number</th>
<th>4404750377161</th>
</tr>
</thead>
<tbody>
<tr>
<td>License date</td>
<td>Aug 09, 2018</td>
</tr>
<tr>
<td>Licensed Content Publisher</td>
<td>Springer Nature</td>
</tr>
<tr>
<td>Licensed Content Publication</td>
<td>International Journal of Data Science and Analytics</td>
</tr>
<tr>
<td>Licensed Content Title</td>
<td>Stable Bayesian optimization</td>
</tr>
<tr>
<td>Licensed Content Author</td>
<td>Thanh Dai Nguyen, Sunil Gupta, Santu Rana et al</td>
</tr>
<tr>
<td>Licensed Content Date</td>
<td>Jan 1, 2018</td>
</tr>
<tr>
<td>Type of Use</td>
<td>Thesis/Dissertation</td>
</tr>
<tr>
<td>Requestor type</td>
<td>academic/university or research institute</td>
</tr>
<tr>
<td>Format</td>
<td>print and electronic</td>
</tr>
<tr>
<td>Portion</td>
<td>full article/chapter</td>
</tr>
<tr>
<td>Will you be translating?</td>
<td>no</td>
</tr>
<tr>
<td>Circulation/distribution</td>
<td>&lt;501</td>
</tr>
<tr>
<td>Author of this Springer Nature content</td>
<td>yes</td>
</tr>
<tr>
<td>Title</td>
<td>Mr</td>
</tr>
<tr>
<td>Instructor name</td>
<td>Thanh Dai Nguyen</td>
</tr>
<tr>
<td>Institution name</td>
<td>Deakin University</td>
</tr>
<tr>
<td>Expected presentation date</td>
<td>Oct 2018</td>
</tr>
<tr>
<td>Requestor Location</td>
<td>Mr. Thanh Nguyen</td>
</tr>
<tr>
<td></td>
<td>4/1A Villamanta Street</td>
</tr>
</tbody>
</table>

Geelong West, VIC 3218
Australia
Attn: Mr. Thanh Nguyen

Billing Type
Invoice

Billing Address
Mr. Thanh Nguyen
4/1A Villamanta Street

Geelong West, Australia 3218
Attn: Mr. Thanh Nguyen

Total
0.00 AUD

Terms and Conditions

Springer Nature Terms and Conditions for RightsLink Permissions
Springer Customer Service Centre GmbH (the Licenser) hereby grants you a non-exclusive, world-wide licence to reproduce the material and for the purpose and
requirements specified in the attached copy of your order form, and for no other use, subject to the conditions below:

1. The Licenser warrants that it has, to the best of its knowledge, the rights to license reuse of this material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

2. Where print only permission has been granted for a fee, separate permission must be obtained for any additional electronic re-use.

3. Permission granted free of charge for material in print is also usually granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.

4. A licence for ‘post on a website’ is valid for 12 months from the licence date. This licence does not cover use of full text articles on websites.

5. Where ‘reuse in a dissertation/thesis’ has been selected the following terms apply: Print rights for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).

6. Permission granted for books and journals is granted for the lifetime of the first edition and does not apply to second and subsequent editions (except where the first edition permission was granted free of charge or for signatories to the STM Permissions Guidelines http://www.stm-assoc.org/copyright-legal-affairs/permissions/permissions-guidelines/), and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence.

7. Rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com/bookpermissions@springernature.com for these rights.

8. The Licensor's permission must be acknowledged next to the licensed material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book’s homepage. Our required acknowledgement format is in the Appendix below.

9. Use of the material for incidental promotional use, minor editing privileges (this does not include cropping, adapting, omitting material or any other changes that affect the meaning, intention or moral rights of the author) and copies for the disabled are permitted under this licence.

10. Minor adaptations of single figures (changes of format, colour and style) do not require the Licensor's approval. However, the adaptation should be credited as shown in Appendix below.

Appendix — Acknowledgements:

For Journal Content:
Reprinted by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION] (Article name, Author(s) Name), [COPYRIGHT] (year of publication)
For **Advance Online Publication** papers:
Reprinted by permission from [the Licenser]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].)

For Adaptations/Translations:
Adapted/Translated by permission from [the Licenser]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)

**Note:** For any republication from the **British Journal of Cancer**, the following credit line style applies:
Reprinted/adapted/translated by permission from [the Licenser]: on behalf of Cancer Research UK: : [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)

For **Advance Online Publication** papers:
Reprinted by permission from The [the Licenser]: on behalf of Cancer Research UK: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj. [JOURNAL ACRONYM])

For **Book content:**
Reprinted/adapted by permission from [the Licenser]: [Book Publisher (e.g. Palgrave Macmillan, Springer etc) [Book Title] by [Book author(s)] [COPYRIGHT] (year of publication)

**Other Conditions:**

Version 1.0

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.
This Agreement between Mr. Thanh Nguyen ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number 4404750058523
License date Aug 09, 2018
Licensed Content Publisher Springer Nature
Licensed Content Publication Springer eBook
Licensed Content Title Cascade Bayesian Optimization
Licensed Content Author Thanh Dai Nguyen, Sunil Gupta, Santu Rana et al
Licensed Content Date Jan 1, 2016
Type of Use Thesis/Dissertation
Requestor type academic/university or research institute
Format print and electronic
Portion full article/chapter
Will you be translating? no
Circulation/distribution <501
Author of this Springer Nature content yes
Title Mr
Instructor name Thanh Dai Nguyen
Institution name Deakin University
Expected presentation date Oct 2018
Requestor Location Mr. Thanh Nguyen
4/1A Villamanta Street

Geelong West, VIC 3218
Australia
Attn: Mr. Thanh Nguyen

Billing Type Invoice
Billing Address Mr. Thanh Nguyen
4/1A Villamanta Street

Geelong West, Australia 3218
Attn: Mr. Thanh Nguyen

Total 0.00 AUD

Terms and Conditions

Springer Nature Terms and Conditions for RightsLink Permissions

Springer Customer Service Centre GmbH (the Licensors) hereby grants you a non-exclusive, world-wide licence to reproduce the material and for the purpose and
requirements specified in the attached copy of your order form, and for no other use, subject to the conditions below:

1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of this material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

   If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

2. Where print only permission has been granted for a fee, separate permission must be obtained for any additional electronic re-use.

3. Permission granted free of charge for material in print is also usually granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.

4. A licence for 'post on a website' is valid for 12 months from the licence date. This licence does not cover use of full text articles on websites.

5. Where 'reuse in a dissertation/thesis' has been selected the following terms apply:
   Print rights for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).

6. Permission granted for books and journals is granted for the lifetime of the first edition and does not apply to second and subsequent editions (except where the first edition permission was granted free of charge or for signatories to the STM Permissions Guidelines http://www.stm-assoc.org/copyright-legal-affairs/permissions/permissions-guidelines/), and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence.

7. Rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com/bookpermissions@springernature.com for these rights.

8. The Licensor's permission must be acknowledged next to the licensed material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book's homepage. Our required acknowledgement format is in the Appendix below.

9. Use of the material for incidental promotional use, minor editing privileges (this does not include cropping, adapting, omitting material or any other changes that affect the meaning, intention or moral rights of the author) and copies for the disabled are permitted under this licence.

10. Minor adaptations of single figures (changes of format, colour and style) do not require the Licensor's approval. However, the adaptation should be credited as shown in Appendix below.

Appendix — Acknowledgements:

For Journal Content:
Reprinted by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)
For Advance Online Publication papers:
Reprinted by permission from [the Licenser]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM]).]

For Adaptations/Translations:
Adapted/Translated by permission from [the Licenser]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)

Note: For any republication from the British Journal of Cancer, the following credit line style applies:

Reprinted/adapted/translated by permission from [the Licenser]: on behalf of Cancer Research UK: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)

For Advance Online Publication papers:
Reprinted by permission from The [the Licenser]: on behalf of Cancer Research UK: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM])

For Book content:
Reprinted/adapted by permission from [the Licenser]: [Book Publisher (e.g. Palgrave Macmillan, Springer etc) [Book Title] by [Book author(s)] [COPYRIGHT] (year of publication)

Other Conditions:

Version 1.0

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.
This Agreement between Mr. Thanh Nguyen ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number 4404750150372
License date Aug 09, 2018
Licensed Content Publisher Springer Nature
Licensed Content Publication Springer eBook
Licensed Content Title A Privacy Preserving Bayesian Optimization with High Efficiency
Licensed Content Author Thanh Dai Nguyen, Sunil Gupta, Santu Rana et al
Licensed Content Date Jan 1, 2018
Type of Use Thesis/Dissertation
Requestor type academic/university or research institute
Format print and electronic
Portion full article/chapter
Will you be translating? no
Circulation/distribution <501
Author of this Springer Nature content yes
Title Mr
Instructor name Thanh Dai Nguyen
Institution name Deakin University
Expected presentation date Oct 2018
Requestor Location Mr. Thanh Nguyen
4/1A Villamanta Street
Geelong West, VIC 3218
Australia
Attn: Mr. Thanh Nguyen
Billing Type Invoice
Billing Address Mr. Thanh Nguyen
4/1A Villamanta Street
Geelong West, Australia 3218
Attn: Mr. Thanh Nguyen
Total 0.00 AUD

Springer Nature Terms and Conditions for RightsLink Permissions
Springer Customer Service Centre GmbH (the Licensor) hereby grants you a non-exclusive, world-wide licence to reproduce the material and for the purpose and
requirements specified in the attached copy of your order form, and for no other use, subject to the conditions below:

1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of this material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

   If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

2. Where print only permission has been granted for a fee, separate permission must be obtained for any additional electronic re-use.

3. Permission granted free of charge for material in print is also usually granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.

4. A licence for 'post on a website' is valid for 12 months from the licence date. This licence does not cover use of full text articles on websites.

5. Where 'reuse in a dissertation/thesis' has been selected the following terms apply:

   Print rights for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).

6. Permission granted for books and journals is granted for the lifetime of the first edition and does not apply to second and subsequent editions (except where the first edition permission was granted free of charge or for signatories to the STM Permissions Guidelines http://www.stm-assoc.org/copyright-legal-affairs/permissions/permissions-guidelines/), and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence.

7. Rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com/bookpermissions@springernature.com for these rights.

8. The Licensor's permission must be acknowledged next to the licensed material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book’s homepage. Our required acknowledgement format is in the Appendix below.

9. Use of the material for incidental promotional use, minor editing privileges (this does not include cropping, adapting, omitting material or any other changes that affect the meaning, intention or moral rights of the author) and copies for the disabled are permitted under this licence.

10. Minor adaptations of single figures (changes of format, colour and style) do not require the Licensor's approval. However, the adaptation should be credited as shown in Appendix below.

Appendix — Acknowledgements:

For Journal Content:
Reprinted by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION] (Article name, Author(s) Name), [COPYRIGHT] (year of publication)
For **Advance Online Publication** papers:
Reprinted by permission from [the Licenser]: [Journal Publisher (e.g. Nature/Springer/Palgrave)]: [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM]).]

**For Adaptations/Translations:**
Adapted/Translated by permission from [the Licenser]: [Journal Publisher (e.g. Nature/Springer/Palgrave)]: [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)

**Note:** For any republication from the British Journal of Cancer, the following credit line style applies:
Reprinted/adapted/translated by permission from [the Licenser]: on behalf of Cancer Research UK: : [Journal Publisher (e.g. Nature/Springer/Palgrave)]: [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)

For **Advance Online Publication** papers:
Reprinted by permission from [the Licenser]: on behalf of Cancer Research UK: [Journal Publisher (e.g. Nature/Springer/Palgrave)]: [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj. [JOURNAL ACRONYM])

**For Book content:**
Reprinted/adapted by permission from [the Licenser]: [Book Publisher (e.g. Palgrave Macmillan, Springer etc)]: [Book Title] by [Book author(s)] [COPYRIGHT] (year of publication)

**Other Conditions:**

Version 1.0

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.
This Agreement between Mr. Thanh Nguyen ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

<table>
<thead>
<tr>
<th><strong>License Number</strong></th>
<th>4404750262536</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>License date</strong></td>
<td>Aug 09, 2018</td>
</tr>
<tr>
<td><strong>Licensed Content Publisher</strong></td>
<td>Springer Nature</td>
</tr>
<tr>
<td><strong>Licensed Content Publication</strong></td>
<td>Springer eBook</td>
</tr>
<tr>
<td><strong>Licensed Content Title</strong></td>
<td>Privacy Aware K-Means Clustering with High Utility</td>
</tr>
<tr>
<td><strong>Licensed Content Author</strong></td>
<td>Thanh Dai Nguyen, Sunil Gupta, Santu Rana et al</td>
</tr>
<tr>
<td><strong>Licensed Content Date</strong></td>
<td>Jan 1, 2016</td>
</tr>
<tr>
<td><strong>Type of Use</strong></td>
<td>Thesis/Dissertation</td>
</tr>
<tr>
<td><strong>Requestor type</strong></td>
<td>academic/university or research institute</td>
</tr>
<tr>
<td><strong>Format</strong></td>
<td>print and electronic</td>
</tr>
<tr>
<td><strong>Portion</strong></td>
<td>full article/chapter</td>
</tr>
<tr>
<td><strong>Will you be translating?</strong></td>
<td>no</td>
</tr>
<tr>
<td><strong>Circulation/distribution</strong></td>
<td>&lt;501</td>
</tr>
<tr>
<td><strong>Author of this Springer Nature content</strong></td>
<td>yes</td>
</tr>
<tr>
<td><strong>Title</strong></td>
<td>Mr</td>
</tr>
<tr>
<td><strong>Instructor name</strong></td>
<td>Thanh Dai Nguyen</td>
</tr>
<tr>
<td><strong>Institution name</strong></td>
<td>Deakin University</td>
</tr>
<tr>
<td><strong>Expected presentation date</strong></td>
<td>Oct 2018</td>
</tr>
</tbody>
</table>
| **Requestor Location**   | Mr. Thanh Nguyen  
                          4/1A Villamanta Street |

Geelong West, VIC 3218  
Australia  
Attn: Mr. Thanh Nguyen

<table>
<thead>
<tr>
<th><strong>Billing Type</strong></th>
<th>Invoice</th>
</tr>
</thead>
</table>
| **Billing Address** | Mr. Thanh Nguyen  
                          4/1A Villamanta Street |

Geelong West, Australia 3218  
Attn: Mr. Thanh Nguyen

| **Total** | 0.00 AUD |

**Springer Nature Terms and Conditions for RightsLink Permissions**  
Springer Customer Service Centre GmbH (the Licenser) hereby grants you a non-exclusive, world-wide licence to reproduce the material and for the purpose and
requirements specified in the attached copy of your order form, and for no other use, subject to the conditions below:

1. The Licenser warrants that it has, to the best of its knowledge, the rights to license reuse of this material. However, you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

2. Where print only permission has been granted for a fee, separate permission must be obtained for any additional electronic re-use.

3. Permission granted free of charge for material in print is also usually granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.

4. A licence for 'post on a website' is valid for 12 months from the licence date. This licence does not cover use of full text articles on websites.

5. Where 'reuse in a dissertation/thesis' has been selected the following terms apply:
   Print rights for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).

6. Permission granted for books and journals is granted for the lifetime of the first edition and does not apply to second and subsequent editions (except where the first edition permission was granted free of charge or for signatories to the STM Permissions Guidelines http://www.stm-assoc.org/copyright-legal-affairs/permissions/permissions-guidelines/), and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence.

7. Rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com/bookpermissions@springernature.com for these rights.

8. The Licensor's permission must be acknowledged next to the licensed material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book's homepage. Our required acknowledgement format is in the Appendix below.

9. Use of the material for incidental promotional use, minor editing privileges (this does not include cropping, adapting, omitting material or any other changes that affect the meaning, intention or moral rights of the author) and copies for the disabled are permitted under this licence.

10. Minor adaptations of single figures (changes of format, colour and style) do not require the Licensor's approval. However, the adaptation should be credited as shown in Appendix below.

Appendix — Acknowledgements:

For Journal Content:
Reprinted by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)
For **Advance Online Publication papers:**
Reprinted by permission from [the Licensee]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM]).]

**For Adaptations/Translations:**
Adapted/Translated by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)

**Note:** For any republication from the British Journal of Cancer, the following credit line style applies:
Reprinted/adapted/translated by permission from [the Licensor]: on behalf of Cancer Research UK: : [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)

For **Advance Online Publication papers:**
Reprinted by permission from The [the Licensor]: on behalf of Cancer Research UK: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM]).]

For **Book content:**
Reprinted/adapted by permission from [the Licensor]: [Book Publisher (e.g. Palgrave Macmillan, Springer etc) [Book Title] by [Book author(s)] [COPYRIGHT] (year of publication)

**Other Conditions:**

Version 1.0

**Questions?** customecare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.