Optimal Learning for Nonlinear Parametric Belief Models

Xinyu He

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Electrical Engineering
Adviser: Professor Warren B. Powell

June 2017
Abstract

Many real-world optimization problems require making measurements to determine which choice works the best. Such measurements can be noisy and expensive, as might arise in simulations or laboratory experiments. Optimal learning addresses the problem of how to collect information efficiently. In particular, a class of Bayesian policies, known as the Knowledge Gradient (KG), has drawn extensive attention. KG is easy to implement for low-dimensional problems with simple belief models such as look-up table or linear models, but becomes computationally intractable with nonlinear parametric belief models, which arise frequently in multidimensional problems.

In this thesis, we study the optimal learning problem with nonlinear parametric belief models. We assume the function to optimize, which could be some performance, utility or output, can be globally or locally described by some parametric model, where the parameters are unknown. Our goal is to identify the optimal choice (where a choice is also called an experimental design or an alternative) after running a limited number of sequential measurements, which are noisy and expensive.

We first focus on problems with a global parametric form. We use a sampled set of parameters to solve the computational issue of KG, and introduce a resampling process to update the sampled set adaptively. We also develop an efficient implementation of the KG policy for continuous alternatives. We prove that our algorithm, which works for multidimensional and continuous alternative and parameter spaces, can asymptotically learn both the correct parameter and the optimal alternative. Next we study problems in which a globally accurate model is unavailable, but there exist some parametric models that can well describe the function in local regions. We propose a method to construct global approximations of the function and apply KG to quickly identify the optimal alternative. Experiments on both synthetic and real-world problems show the strong performance of our algorithm.
Acknowledgements

I would like to express my deepest gratitude to my advisor, Professor Warren B. Powell, for his continuous support and patient guidance. I am truly fortunate to have the opportunity to work with Professor Powell, whose enthusiasm, expertise, encouragement and trust have helped me tremendously throughout my research and will continue influencing me in the future.

I would also like to thank my committee members, Professor Sun-Yuan Kung and Professor Emmanuel Abbe, for their mentorship and advice, and thank Professor Paul Cuff and Professor Mengdi Wang for their time to read the thesis and all the insightful comments.

I am deeply grateful to everyone who has supported me during my years of study. It is my great honor to have worked with a group of most talented people. In particular, thanks to Si, Yingfei, Kris, Yan, Weidong, Javad, Tsevetan, Harvey, Daniel, Kobby, Raymond, Donghun, Joe, Lina, Juliana, Hugo, and many more. It has been a great pleasure working with all of you, and I hope our paths cross again one day.

I am also thankful to the fellow students and staff members in the Department of Electrical Engineering and the Department of Operations Research and Financial Engineering for their help and support. A special thanks to all my good friends at Princeton, for the best time we spent together.

Furthermore, I would like to thank my parents, for their constant and selfless love, understanding and support, without which I cannot imagine what my life would be like today. I would also like to thank my parents-in-law, for their unconditional love and encouragement.

Finally, my everlasting love and sincere gratitude go to my wife, Yangzhou, for her love and everything.
To my family.
Contents

Abstract . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . iii
Acknowledgements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . iv
List of Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . x
List of Figures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . xi

1 Introduction 1
1.1 Background and Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.2 Applications . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
1.3 Thesis Outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

2 Review of the Knowledge Gradient Policy 13
2.1 Ranking and Selection (R&S) Problem . . . . . . . . . . . . . . . . . . . . . 13
2.2 Knowledge Gradient (KG) Policy . . . . . . . . . . . . . . . . . . . . . . . . 14
2.3 Knowledge Gradient with Discrete Priors (KGDP) . . . . . . . . . . . . . . 15
2.4 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

3 Optimal Learning for Nonlinear Parametric Belief Models over Multidi-
   mensional Parameters 19
3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
3.2 Optimal Learning for Parameters and Alternatives . . . . . . . . . . . . . . 22
   3.2.1 Overview of the Resampling Scheme . . . . . . . . . . . . . . . . . . . . 23
   3.2.2 Criterion for Resampling: Mean Square Error (MSE) . . . . . . . . . 24
3.2.3 Probability Updating in Resampling ............................................. 25
3.2.4 Resampling Procedure ................................................................. 26
3.2.5 Evaluation Metrics ................................................................. 27
3.2.6 Detailed Flowchart ................................................................. 30
3.3 Discussion of a Two-curve Model ................................................. 30
3.4 Convergence of KGDP without Resampling .................................. 33
3.5 Convergence of KGDP with Resampling ....................................... 38
3.6 Empirical Results ................................................................. 40
  3.6.1 An Asymmetric Unimodal Benchmark Function ....................... 41
  3.6.2 Application in W/O/W Nanoemulsion Stability Study .............. 43
  3.6.3 Standard Test Functions ......................................................... 47
3.7 Conclusion ................................................................. 47
3.8 Appendix ................................................................. 49
  3.8.1 Proof of the Two-curve Model .................................................. 49
  3.8.2 Proof of Convergence of KGDP without Resampling ............... 57
  3.8.3 Proof of Convergence of KGDP with Resampling ..................... 68

4 Optimal Learning for Nonlinear Parametric Belief Models over Multidi-
  mensional Continuous Alternatives ............................................... 73
  4.1 Introduction ........................................................................... 74
  4.2 Knowledge Gradient for Parametric Models with Continuous Alternatives .................................................................................. 76
    4.2.1 Approximation of the Knowledge Gradient ......................... 76
    4.2.2 Calculation of \( \nu^{KG-s}(x) \) and \( \nabla \nu^{KG-s}(x) \) ............ 77
    4.2.3 Maximizing \( \nu^{KG-s,n}(x) \) .................................................. 84
    4.2.4 Overall Procedure .......................................................... 85
  4.3 Convergence of Continuous KG ............................................... 85
    4.3.1 Convergence of Continuous KG without Resampling ........... 87
    4.3.2 Convergence of Continuous KG with Resampling ............... 90
4.4 Empirical Results .................................................. 92
  4.4.1 The Number of Restarts of the Gradient Ascent .............. 93
  4.4.2 Additive Newsvendor Problem ................................ 96
  4.4.3 Nested Newsvendor Problem .................................. 97
4.5 Conclusion ......................................................... 100
4.6 Appendix .......................................................... 102
  4.6.1 A Heuristic to Find \{S_k\} and \{x^{(k)}\} ...................... 102
  4.6.2 Proof of Convergence of Continuous KG ...................... 104

5 Optimal Learning with Local Nonlinear Parametric Belief Models over Continuous Designs 116
  5.1 Introduction ..................................................... 117
  5.2 Review of a Local Approximation Technique: DC-RBF .......... 119
  5.3 Algorithm ......................................................... 121
    5.3.1 Sketch of the Basic Ideas .................................. 121
    5.3.2 Construction of Candidate Functions ......................... 122
    5.3.3 Calculation of Candidate Function Probabilities .......... 124
    5.3.4 Minor Update of Candidate Functions ...................... 125
    5.3.5 Major Update of Candidate Functions ...................... 126
    5.3.6 KGDP and continuous KGDP ................................. 132
    5.3.7 Overall Flowchart .......................................... 133
  5.4 Empirical Results ............................................... 134
    5.4.1 Function Approximation in a Two-dimensional Multimodal Problem . 136
    5.4.2 Multi-dimensional Unimodal Problem ....................... 137
    5.4.3 Multi-dimensional Multimodal Problem ..................... 139
    5.4.4 An Application: Carbon Nanotube Growth .................. 140
  5.5 Conclusion ...................................................... 141
  5.6 Appendix ........................................................ 143
5.6.1 Heuristics in Refitting Local Functions ................. 143
5.6.2 Generation of the True Function $f^*(x)$ for Carbon Nanotube Growth 144

6 Conclusions and Future Work 148

Bibliography 153
List of Tables

3.1 Comparison of different policies at $n = 20$ (with 20\% noise) . . . . . . . . . . 48

4.1 Comparison of different policies when $n = 20$ . . . . . . . . . . . . . . . . . . . . . . . 99
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Unaligned functions and KGDP-( f ), KGDP-( H ) scores in the two-curve model.</td>
<td>33</td>
</tr>
<tr>
<td>3.2</td>
<td>Aligned functions and KGDP-( f ), KGDP-( H ) scores in the two-curve model.</td>
<td>33</td>
</tr>
<tr>
<td>3.3</td>
<td>Illustration of resampling in ( \theta ) space (Iterations 1, 2, 3, 5, 7, 9) (5,7,9 are zoomed in).</td>
<td>42</td>
</tr>
<tr>
<td>3.4</td>
<td>OC% of different policies for three-dimensional ( \theta ) (5%, 20% and 50% noise from left to right).</td>
<td>43</td>
</tr>
<tr>
<td>3.5</td>
<td>OC% of different policies in three-dimensional nanoemulsion (50% noise).</td>
<td>44</td>
</tr>
<tr>
<td>3.6</td>
<td>Performance of resampling KGDP in finding ( \theta ) in three-dimensional nanoemulsion under 50% noise (note that ( \theta_i ) is the ( i )-th dimension of the vector ( \theta )).</td>
<td>46</td>
</tr>
<tr>
<td>3.7</td>
<td>Evolution of ( \bar{f}^n(x) ) (seven-dimensional ( \theta ), 20% noise).</td>
<td>46</td>
</tr>
<tr>
<td>3.8</td>
<td>OC% of different policies in seven-dimensional nanoemulsion (50% noise).</td>
<td>47</td>
</tr>
<tr>
<td>3.9</td>
<td>Two examples of the set of ( g(x,\omega) ) and the separating points ( {S_k} ).</td>
<td>51</td>
</tr>
<tr>
<td>3.10</td>
<td>An example in which KGDP-( f ) identifies ( x^* ) but not ( \theta^* ) as ( N \to \infty ).</td>
<td>67</td>
</tr>
<tr>
<td>4.1</td>
<td>Illustration of the approximation of KG in one-dimensional ( \mathcal{X} ).</td>
<td>78</td>
</tr>
<tr>
<td>4.2</td>
<td>Illustration of the approximation of KG in two-dimensional ( \mathcal{X} ).</td>
<td>78</td>
</tr>
<tr>
<td>4.3</td>
<td>Example of the parametric functions.</td>
<td>79</td>
</tr>
<tr>
<td>4.4</td>
<td>Illustration of ( {S_k} ) and ( {x^{(k)}} ).</td>
<td>80</td>
</tr>
<tr>
<td>4.5</td>
<td>Illustration of the gradient ascent process to maximize ( \nu^{KG,s}(x) ).</td>
<td>84</td>
</tr>
<tr>
<td>4.6</td>
<td>Maximum KG found vs the number of restarts.</td>
<td>94</td>
</tr>
</tbody>
</table>
4.7 Comparison of different number of restarts in terms of opportunity cost (quality of the solution $\hat{x}^*$) and statistical error (quality of estimation $\hat{\theta}^*$) (1st row: 20% noise, 2nd row: 50% noise) .................................................. 95

4.8 Comparison of policies for ten-dimensional (1st row) and twenty-dimensional (2nd row) $\mathcal{X}$, in terms of opportunity cost (quality of the solution $\hat{x}^*$) and statistical error (quality of estimation $\hat{\theta}^*$) ................................................................. 97

4.9 Comparison of policies for ten-dimensional $\mathcal{X}$ in terms of opportunity cost (quality of the solution $\hat{x}^*$) and statistical error (quality of estimation $\hat{\theta}^*$) (1st row: 20% noise, 2nd row: 50% noise) .................................................. 98

4.10 Comparison of continuous KG and KGDP with different number of alternatives in terms of opportunity cost (quality of the solution $\hat{x}^*$) and statistical error (quality of estimation $\hat{\theta}^*$) (1st row: 20% noise, 2nd row: 50% noise) . . . 100

5.1 Construction of candidate functions .................................................. 123

5.2 Some examples of fitted Beta distribution. Blue curves: $c \cdot (D_{T,l}^T)^{a-1}(1 - D_{T,l}^T)_{\beta-1}$; Red dots: $\{p^n_l, l = 1, ..., L\}$. ............................................................... 130

5.3 Clustering results after 35 measurements in a two-dimension problem. $x$ and $y$ axes indicate the coordinates of $x^0, ..., x^{n-1}$, and each color indicates a cluster. The first and third images run DC-RBF with different permutations of $x^0, ..., x^{n-1}$. The second and fourth images run K-means based on the first and the third images, respectively .................................................. 130

5.4 Evolution of $\bar{f}^n$ under the KG policy. The black dots in the right column show the points we have measured. The noise level is 20% ........................................................................ 137

5.5 OC% and MSE of the four-dimensional newsvendor problem, approximated by linear functions. Noise level: 20%. .................................................. 138

5.6 Results of the four-dimensional newsvendor problem, approximated by quadratic functions .................................................. 138

xii
5.7 Results of the six-dimensional newsvendor problem, approximated by linear functions. Noise level: 20%. .......................................................... 139
5.8 OC% of four-dimensional test function. Left: no noise; right: 20% noise. . . 140
5.9 OC% and MSE using the linear function $g_{lin}(x; \theta_j)$ locally, with 0 noise (left) and 20% noise (right). .......................................................... 141
5.10 Evolution of $\bar{f}$ (red curves in the first row) and candidate functions (thin curves in the second row) under KG policy in a one-dimensional problem under 20% noise. .......................................................... 144
5.11 OC% and MSE using the chemical reaction network $g_{crn}(x; \theta_j)$ locally, with 0 noise. .......................................................... 147
Chapter 1

Introduction

The area of optimal learning addresses the problem of how to collect information efficiently. Such a problem arises when people want to optimize some unknown function or understand some underlying system, which requires running a few measurements before determining the best choice. The problem becomes particularly challenging when the procedure of information collection, which we also refer to as a measurement or experiment, is expensive and noisy. The area of optimal learning has extensive applications in materials science, simulation optimization, health and medicine, revenue management, energy, transportation, and many other fields. For example, materials scientists may want to find the optimal experimental designs through a number of time-consuming laboratory experiments; business managers may need to set a price that maximizes the profit by conducting a number of trials. Some more detailed examples are given in Section 1.2.

In this thesis, we focus on problems in which the underlying process can be modeled by some parametric function $f(x; \theta)$, either globally or locally, where $\theta$ is some unknown multidimensional parameter vector, and $x$ is a multidimensional continuous controllable design vector, which we also call an alternative. Our goal is to find the optimal alternative that maximizes the output, and at the same time, have a good estimate of the unknown parameter $\theta$, through a limited number of noisy and expensive measurements. Such a parametric
formulation appears in many situations. For instance, many materials science processes can be well approximated by mathematical models containing unknown parameters; in revenue management, the profit may be modeled as a function of the supplies, demands, costs and prices. More generally, some more complicated system may be hard to model globally, but can be approximated locally by simpler models.

Our algorithms are mainly built upon a major class of Bayesian policies, known as the Knowledge Gradient (KG) policy (Frazier et al. 2008), which is a strategy to select which alternative to measure for the next experiment based on current information. KG is a one-step look-ahead policy that tries to maximize the value of information from the next measurement. By construction, it is myopically optimal, but has also been proved to be asymptotically optimal in many cases. KG is easy to implement for simple models such as look-up table or linear belief models, but becomes highly challenging for nonlinear parametric belief models; on the other hand, the nonlinear parametric formulation appears in a wide range of real-world applications, especially those with multidimensional designs. In this thesis, we develop the KG policy for nonlinear parametric belief models, in which both the alternative and the parameter spaces are multidimensional and continuous. We show the strong performance of our algorithm empirically, and establish theoretical results.

1.1 Background and Motivation


Similar optimization problems with learning unknown parameters also arise in communities such as oligopoly games (Bischi et al. 2007, 2010, 2008, Szidarovszky & Krawczyk 2004, Szidarovszky 2004), adaptive control (Ljung & Gunnarsson 1990) and iterative learning control (Uchiyama 1978, Moore 1993). However, most of these problems are unconstrained, and they mostly focus on converging to some stationary state in the presence of adversaries or feedbacks (and then optimize), rather than collecting information efficiently for the purpose of optimization.

The problem of optimizing a noisy and expensive function over a finite set of alternatives is often referred to as the ranking and selection (R&S) problem (Kim & Nelson 2006, Hong & Nelson 2009, Kim & Nelson 2007). Since the experiments in such problems are expensive (consider, for example, the laboratory experiments in materials science which might take days or weeks), we have to work with a limited budget and manage the “exploration vs exploitation” tradeoff (see Chapter 12 of Powell 2011), which requires a balance of exploiting the current optimal solutions and exploring with uncertain outcomes to learn the problem.

A closely related field that also deals with the challenge of exploration vs exploitation is the multi-armed bandit (MAB) community (see, e.g, Berry & Fristedt 1985, Gittins et al.
The MAB problems are different from R&S problems in that MAB considers maximizing the cumulative rewards of all experiments, while R&S only considers the final reward. Some simple heuristics to balance the exploration vs exploitation tradeoff include epsilon-greedy (Sutton & Barto 1998, also see Singh et al. 2000 for convergence analysis), interval estimation (Kaelbling 1993), and Chernoff interval estimation (Streeter & Smith 2006). These methods usually perform well on small problems but have scaling issues in higher dimensions. More sophisticated methods include Gittins indices (Gittins & Jones 1974, Gittins 1979, Gittins et al. 2011) (usually used for problems with cumulative rewards), upper confidence bound methods (Auer et al. 2002), Thompson sampling (Thompson 1933, see Chapelle & Li 2011, Agrawal & Goyal 2012 for recent work) and expected improvement (Jones et al. 1998, also see Gramacy & Lee 2011 for recent work). More broadly, the R&S problems are also related to problems in active learning (Settles 2009, Tong & Chang 2001, Olsson 2009), subset selection (Miller 2002, Ryzhov & Powell 2009), and simulation optimization (Hong & Nelson 2009).

There have been two major approaches to R&S problems: the frequentist approach and the Bayesian approach. The frequentist approach assumes that information only comes from observations and uses the distribution of the observed data to estimate unknown parameters (see, e.g., Hastie et al. 2009, Kim & Nelson 2006, Fu et al. 2007, Audibert & Bubeck 2010). The Bayesian approach begins with a probabilistic belief model about the true parameters which are updated as new observations are collected (see, e.g., Chen et al. 2000, Chick 2006, He et al. 2007, Harrison et al. 2012, Russo & Van Roy 2014, Peng et al. 2016). For the Bayesian approach, there are mainly two directions. One maximizes the posterior probability of correct selection, such as the Optimal Computing Budget Allocation (OCBA) (see, e.g., Chen 1995, Chen et al. 2000, He et al. 2007) and Thompson sampling (Thompson 1933), while the other maximizes the incremental information from a single experiment, also known as the Value of Information (VoI) procedure (see, e.g., Chick & Inoue 2001, Frazier et al.
Below we give a more detailed review of the VoI procedure, with special attention given to the Knowledge Gradient policy (Frazier et al. 2008).

Since every experiment is expensive or time consuming, it is important to maximize the value of each measurement. Gupta & Miescke (1996) proposes the idea of computing the marginal value of information from a single experiment. Based on this, Frazier et al. (2008) extends the idea using the Bayesian approach, and presents the Knowledge Gradient (KG) policy. It analyzes the case where all alternatives are independent, and also proves that KG is the only stationary policy that is both myopically optimal (i.e., it produces the best decision if the budget is $N = 1$) and asymptotically optimal (i.e., it will identify the optimal alternative if the budget is $N = \infty$). Frazier et al. (2009) adapts the knowledge gradient to handle correlated beliefs among a discrete set of alternatives. Both Frazier et al. (2008) and Frazier et al. (2009) use look-up table belief models, which become computationally expensive when the number of alternatives is large, as typically happens when an experimental choice involves multiple dimensions. Negoescu et al. (2011) is the first paper that studies KG using a parametric belief model. It assumes that the belief model is linear in some unknown parameters, and imposes the uncertainty of the function values onto the parameters. This strategy reduces the number of parameters to be estimated from the number of alternatives (as required by a look-up table belief model) to the dimensionality of a parameter vector. While the knowledge gradient is easy to compute for belief models that are linear in the parameters, nonlinear belief models are computationally intractable. Chen et al. (2014) studies the more general nonlinear parametric model, and proposes the Knowledge Gradient with Discrete Priors (KGDP) model, which overcomes the computational issues but requires that we represent the uncertainty about the unknown parameters using a finite number of samples, where one of the samples has to be the true parameter. KGDP is able to handle any nonlinear belief model, but the assumption that one of the candidates is correct (we also refer to this assumption as the truth from prior assumption) is too strong in most real world settings, especially when the parameters have four or more dimensions. Moreover,
Chen et al. (2014) also fails to give any theoretical proof of the convergence of KGDP given the truth from prior assumption.

Additionally, while KG has shown impressive performance in all these models, the limitation of finite alternatives becomes a constraint especially for higher dimensional problems. Scott et al. (2011) is the first attempt to extend KG to continuous problems. It models the problems using a look-up table belief model and proposes an algorithm based on Gaussian process regression using previous observations. However, the use of a non-parametric belief model limits the algorithm to problems of at most three or possibly four dimensions.

In this thesis, we first focus on problems that can be modeled by a nonlinear parametric function globally. Based on the idea of KGDP (Chen et al. 2014), we propose a resampling algorithm that can deal with a much larger parameter space. We then introduce a continuous KG policy that can be applied to a multidimensional continuous alternative space. In the end, we relax the assumption that a globally accurate model is known, and propose a KG policy using locally accurate models.

1.2 Applications

There are many real-world optimization problems involving collecting information sequentially from noisy measurements, with objective functions that can be described either globally or locally by some parametric functions. Such settings arise in a wide range of fields including material science, health and drug discovery, revenue management, simulation optimization, and engineering. We list a few applications below.

- **Materials Science**

  Materials scientists often need to find the optimal experimental configuration through a number of trials and errors, which may be expensive or time-consuming. For example, they may want to identify the best temperature, density, or concentration of some
inputs, in order to maximize some output or quality. Many times, the underlying process can be fully or partly described by mathematical models, and the optimal learning technique can help reduce their time and effort significantly. Some concrete examples are listed below.

- **Stability of Water-oil-water Nanoemulsion**

  This problem studies the stability of payload delivery in water-oil-water (W/O/W) double nanoemulsions, in which the payload molecules are delivered from the internal water droplet, surrounded by a larger droplet of oil, to the external water. The emulsion stability is often characterized over a time scale of hours, days and sometimes even longer. The underlying kinetics can be modeled by a group of differential equations. As a result, the stability can be modeled by a globally accurate nonlinear parametric model with multidimensional control variables (i.e, alternatives) and unknown parameters. This problem is analyzed using the KGDP model in Chen et al. (2014), and is further studied in Chapter 3.6.2.

- **Carbon Nanotube Growth**

  The carbon nanotube is one of the most impressive materials with thousands of applications (Baughman et al. 2002). The process of carbon nanotube growth takes input gases including $C_2H_4$, $H_2$ and $CO_2$, and involves multiple complicated reactions. The control variables include concentrations of the three gases and the temperature, which are all continuous variables. The goal is to maximize the concentration of the carbon nanotube at the end an experiment, which takes 300 seconds. The exact mechanism is not well understood and thus a globally accurate model does not exist yet, but the process can be described locally by a chemical reaction network model (see, e.g, Érdi & Tóth 1989, Angeli 2009), or even by simple linear models. This problem is studied in Chapter 5.4.4, and simulations demonstrate promising results using our algorithm.
- Nanostructured Devices

Wang et al. (2015) applies the knowledge gradient policy to an application on nanostructured devices, in which the goal is to maximize the output current of a photoactive device. In this problem, scientists need to identify the optimal density, size and the type of nanostructures by sequential experiments. Wang et al. (2015) uses a third-order polynomial approximation to model the output current. More complicated models on optical and electrical properties of nanostructured devices can be found, for example, in Djurišić & Leung (2006), Gehr & Boyd (1996), and Flory et al. (2011).

- Health and Medicine

- Drug Discovery

Drug development is usually a long and expensive process. Scientists need to find the best composition of a new drug through a series of experiments, where the control variables include the compounds, structures, as well as some physiochemical properties such as molecular weight and lipophilicity. Some computational and statistical tools have been exploited in drug discovery. For example, some linear parametric models can be found in Free & Wilson (1964) and Hansch & Fujita (1964); a good review and examples of more complicated models can be found in Grover et al. (2000a,b). In particular, Negoescu et al. (2011) applies optimal learning methods to the Free-Wilson model proposed in Free & Wilson (1964), which shows impressive results.

- Drug Dosage

Researchers want to find the optimal dosage of certain drugs to maximize their therapeutic effects against a condition while controlling the side effects. Each patient responds differently to medication, and thus the measurements are noisy. In the area of drug dosage-response estimation, people have proposed various
models to capture the relationship between dosage and response. Some examples can be found in Eichhorn & Zacks (1973) and Altshuler (1981).

- **Simulation Optimization**

The community of simulation optimization often faces the problem of finding the design parameters that produce the best performance. Some example problems might be finding the positions of ambulances to minimize the arrival time, determining the production amounts from each plant and the supply amounts to each local region to maximize the profit, or finding the configuration of some components (e.g, the size of a buffer) to maximize the performance of a machine (see more applications in, e.g, Fu et al. 2005, Banks 1998, Pasupathy & Henderson 2006). These simulators usually conduct Monte Carlo simulations, which are time-consuming. Moreover, it is worth mentioning that model-based methods are widely used in simulation optimization. For example, one important class of algorithms is based on meta-models, which fit local or global parametric models to optimize the unknown underlying function (see, e.g, Khuri & Mukhopadhyay 2010, Regis & Shoemaker 2005, Barton & Meckesheimer 2006).

- **Business**

  - **Newsvendor Problem**

The newsvendor problem studies how many newspapers should be purchased at the start of each day, so that the long-run daily profit is maximized with random demands. There have been models for both one-dimensional and multi-dimensional newsvendor problems. Specifically, the multi-dimensional model, which produces a wide range of asymmetric, unimodular functions, is studied as a benchmark function in this thesis.
• Energy

- Design of Fuel Cells

In the design of fuel cells, some examples of design variables that influence the performance of a fuel cell include the power density of the anode or cathode, the operating temperature and pressure, and the conductivity of bipolar plates. Scientists often need to do a number of experiments to find the optimal designs. Some parametric models on fuel cells can be found in Wang et al. (2003), Corrêa et al. (2004), Cheddie & Munroe (2005).

1.3 Thesis Outline

We summarize the organization of the thesis in this section. This thesis focuses on optimal learning with nonlinear parametric belief models. We first consider problems which can be modeled by some parametric function globally, and propose algorithms that can deal with multidimensional continuous parameter and alternative spaces. We prove in theory that our algorithms are asymptotically optimal in finding both the true parameter and the optimal alternative. We then move to the case in which no global model is known but some local models exist, and propose a local approximation method that works alongside the KG policy to solve the optimal learning problem.

Chapter 2

In this chapter, we review the formulation of ranking and selection problems, the concept of knowledge gradient, and the model of Knowledge Gradient with Discrete Priors (KGDP) (Chen et al. 2014), which is a simple model that deals with nonlinear belief functions. While KGDP is the first attempt to solve nonlinear parametric problems, it is limited to a small set of parameters and a finite number of alternatives with low dimensions. In this thesis, we
build algorithms upon the basic idea of KGDP, and propose methods that can solve a much broader range of real-world problems.

Chapter 3

We start with considering the problem of optimizing an expensive function $f(x; \theta)$ with a known parametric form but unknown parameter vector $\theta$, where $\theta$ can be multidimensional. We formulate the problem as a dual-objective optimization problem, in which we maximize the function in the alternative space and minimize the fitting error in the parametric space. We use a sampled set of parameters to overcome the computational intractability of the knowledge gradient, and introduce a resampling algorithm in the parameter space which allows the sampled set to adapt to new information, exploiting past experiments. We also propose a new metric to calculate the knowledge gradient, which focuses on reducing the uncertainty of parameters by maximizing the entropic loss. We then show theoretically that the resampling method converges asymptotically to the true parameters, while simultaneously maximizing our function. In the end, we show empirically that the process exhibits rapid convergence, yielding good results with a very small number of experiments.

Chapter 4

In Chapter 4, we consider the optimal learning problem of optimizing an expensive parametric function $f(x; \theta)$, where the $x$ space is multidimensional and continuous. We develop an efficient implementation of the KG policy, and describe a gradient-based algorithm for maximizing the knowledge gradient with respect to the alternative $x$. Our algorithm can handle $x$ and $\theta$ that are multidimensional continuous vectors. We prove that our algorithm asymptotically learns the correct parameter $\theta$ and the optimal alternative $x$. Experiments show that the algorithm produces fast convergence, even in higher dimensions.
Chapter 5

We consider the problem of optimizing an unknown function over a multidimensional continuous space, where a globally accurate model of the function is not available, but there exist some parametric models that can well approximate the function in local regions. We propose an algorithm in the optimal learning framework that learns the shape of the function and finds the optimal design with a limited number of measurements. We build up belief functions using some local approximation technique and use the Knowledge Gradient policy to decide where to measure. We evaluate our algorithm on a variety of problems, ranging from a series of test functions to a real-world application on carbon nanotube growth, all showing strong performances.

Chapter 6

In the last chapter, we summarize the work in this thesis and propose some ideas on future work. The ideas include both possible improvements of our current methods, and some thoughts on potential new algorithms, theories, and applications.
Chapter 2

Review of the Knowledge Gradient Policy

In this chapter, we first review the ranking and selection (R&S) problem and introduce the Knowledge Gradient (KG) policy. We then review a simple model, known as the Knowledge Gradient with Discrete Priors (KGDP) (Chen et al. 2014), which is the first attempt to apply the KG policy to problems with nonlinear parametric belief models. We end this chapter by discussing how our work starts from the original KGDP model and extends it to solving more complicated multidimensional problems.

2.1 Ranking and Selection (R&S) Problem

In the ranking and selection problem, we have a finite number of alternatives \( \mathcal{X} = \{x_1, x_2, ..., x_M\} \). Each alternative \( x \in \mathcal{X} \) is associated with a true utility \( \mu_x \), which is presumed unknown to us. Information can be collected by measuring any alternative, but the measurements are expensive and noisy. The goal is to determine the \( x \) that has the largest utility through a budget of \( N \) sequential measurements.

At time \( n \) (i.e., after \( n \) measurements), suppose we decide to query alternative \( x^n \) according to some policy, and our measurement is \( \hat{y}^{n+1} \), where the superscripts imply that \( \hat{y}^{n+1} \) will
be unknown until the \((n + 1)\)-th measurement. We assume the inherent noise \((W^n)\) in each measurement follows a normal distribution with zero mean and variance \(\sigma^2\), where \(\sigma\) is known to us. That is, for \(n = 0, 1, ..., N - 1\),

\[ \hat{y}^{n+1} = \mu_{x^n} + W^{n+1}, \]

where \(W^{n+1} \sim \mathcal{N}(0, \sigma^2)\), i.i.d.

For each \(x \in \mathcal{X}\), we use \(\hat{\mu}_x^n\) as our estimate of the true utility \(\mu_x\) after \(n\) experiments. In an offline R&S problem, we only care about how well our final decision performs after \(N\) measurements, regardless of the performance of our choices during the experiments. Hence, the goal is to select the \(x\) with the highest estimate after \(N\) measurements, i.e.,

\[ x^N = \arg\max_{x \in \mathcal{X}} \hat{\mu}_x^N. \]

### 2.2 Knowledge Gradient (KG) Policy

The Knowledge Gradient policy proposes a rule of selecting which alternative \(x\) to measure at each iteration. The principle of this policy is to maximize the expected value of information in the next measurement, which is called the knowledge gradient. Specifically, let \(S^n\) denote the state of knowledge at time \(n\), and let \(V^n(S^n)\) be the value of information if we are in state \(S^n\). In the R&S problem, we have \(V^n(S^n) = \max_{x' \in \mathcal{X}} \hat{\mu}_{x'}^n\). The transition from \(S^n\) to \(S^{n+1}\) occurs when we take a measurement at \(x^n = x\) and observe \(\hat{y}^{n+1}\), with

\[ V^{n+1}(S^{n+1}(x)) = \max_{x' \in \mathcal{X}} \hat{\mu}_{x'}^{n+1}(x). \]

At time \(n\), \(\hat{\mu}_{x'}^{n+1}(x)\) is a random variable since it depends on the noise \(W^{n+1}\). We would like to maximize our expected incremental information from the next measurement, which
we call the knowledge gradient. At time $n$, the knowledge gradient of $x$ is defined as:

$$
\nu^{KG,n}(x) = \mathbb{E} \left[ V^{n+1}(S^{n+1}(x)) - V^n(S^n) \mid S^n = s, x^n = x \right] \\
= \mathbb{E} \left[ \max_{x' \in \mathcal{X}} \mu^{n+1}_{x'}(x) \mid x^n = x \right] - \max_{x' \in \mathcal{X}} \mu^n_{x'},
$$

(2.1)

where $\mathbb{E}^n$ is the expectation conditioned on $S^n$.

In each iteration, the Knowledge Gradient policy measures the alternative with the largest knowledge gradient:

$$x^{KG,n} = \arg\max_{x \in \mathcal{X}} \nu^{KG,n}(x).$$

This is a myopic policy that maximizes the value of information from a single experiment (Frazier et al. 2008).

The knowledge gradient in Equation (2.1) is easy to calculate when the belief model has some simple form, such as when it can be modeled using a look-up table (Frazier et al. 2008, 2009), or it is linear with respect to the parameters (Negoescu et al. 2011), with the form $f(x; \theta) = \theta_0 + \theta_1 \phi_1(x) + \ldots + \theta_n \phi_n(x)$, where $(\theta_1, \ldots, \theta_n)$ is the $n$-dimensional parameter vector. We run into problems, however, if the function is nonlinear in the parameter vector $\theta$, which complicates the process of computing the knowledge gradient, since the $\mathbb{E}[\max_{x' \in \mathcal{X}} \mu^{n+1}_{x'}(x)]$ becomes computationally intractable. An approach to overcome the computational difficulty, first introduced in Chen et al. (2014), is to replace the original probabilistic belief model with a sampled belief model, a method called the Knowledge Gradient with Discrete Priors (KGDP).

### 2.3 Knowledge Gradient with Discrete Priors (KGDP)

The Knowledge Gradient with Discrete Priors (KGDP) (Chen et al. 2014) is designed to handle parametric belief models that are nonlinear in the parameters. Suppose the utility at any $x \in \mathcal{X}$ can be described by a parametric function $f(x; \theta)$, where $x$ is the alternative
and $\theta$ represents the unknown parameters. The expression of $f(x; \theta)$ is known to us except for the unknown $\theta$. Furthermore, let $\theta^*$ denote the true parameter. Our goal is to find the $x$ that maximizes the true function $f(x; \theta^*)$.

KGDP uses a sampled set of $\theta$’s, $\{\theta_1, ..., \theta_L\}$, as candidates, and assumes one of them is the true $\theta^*$. At time $n$, the probability of $\theta_i$ being the truth is denoted as $p^n_i$. The true function $f(x; \theta^*)$ is approximated as a weighted sum of $L$ candidates:

$$f(x; \theta^*) \approx \bar{f}^n(x) = \sum_{i=1}^{L} f(x; \theta_i)p^n_i.$$ 

As measurements are taken, the $\theta$’s are fixed, while after each iteration, the probabilities $p^n_i$’s get updated according to Bayes’ rule, which means that the belief is naturally conjugate. The state variable $S^n$ is defined as the probability vector, namely $S^n = (p^n_1, p^n_2, ..., p^n_L)$. The KGDP score of each alternative $x$ at time $n$ is given by:

$$\nu^{KGDP,n}(x) = \mathbb{E}[\max_{x'} \sum_{i=1}^{L} f_i(x')p^{n+1}_i | S^n = s, x^n = x] - \max_{x'} \sum_{i=1}^{L} f_i(x')p^n_i.$$ 

In each iteration, the KGDP scores of all alternatives are calculated, and the alternative with the highest KGDP score is selected to evaluate next, i.e,

$$x^n = \arg\max_{x \in \mathcal{X}} \nu^{KGDP,n}(x).$$ 

### 2.4 Discussion

KGDP provides a powerful approach to handling nonlinear parametric models by using a sampled belief for the unknown parameters. However, as a oversimplified model, KGDP has some major limitations, especially in that it can only deal with a small number of discrete $x$’s and $\theta$’s.

First, KGDP uses a small set of candidate parameters and requires one of them to be the truth (we also refer to this as the truth from prior assumption). This may be a reasonable approximation if $\theta$ has one or two dimensions, but is unlikely to hold in higher
dimensions unless we keep a very large sample set. However, the complexity of KGDP
grows quadratically as the number of candidate parameters increases. In the experiments
described in Chen et al. (2014), KGDP can handle at most tens of candidates. In higher
dimensional applications (more than four or five dimensions), a few dozen candidates are
unlikely to be close to the truth in all dimensions, a well-known result of the curse of
dimensionality. Meanwhile, even if computational complexity is not the concern, after we
take several measurements, it is likely that a large portion of $\theta$’s would have very small
probabilities, making it inefficient to keep a large set of $\theta$’s. To solve the complexity and
inefficiency issues of KGDP, we propose a resampling algorithm that can adaptively find
more promising parameter candidates, which is presented in Chapter 3.

Second, the KGDP model can only handle a finite set of alternatives, which introduces
a limitation similar to the truth from prior assumption in KGDP for $\theta$. Specifically, if the
alternatives ($x$) are continuous with more than four or five dimensions, a finite set of alter-
natives (numbering in the tens) is not going to do a good job in either capturing the optimal
solution or providing enough information in the measurements. We thus propose and analyze
a way to extend the original KGDP model, along with the resampling algorithm presented
in Chapter 3, to handle control vectors $x$ and parameters $\theta$ that are both multidimensional
and continuous. We introduce our continuous KG algorithm in Chapter 4 and show that our
algorithm demonstrates strong performances even in high dimensions.

Third, the original KGDP as well as our work in Chapter 3 and Chapter 4 requires
the existence of a parametric model that can largely describe the utility function globally.
However, in many problems, such a global parametric function may not exist, or even if it
exists, the expression is not easy to obtain. Many times, we only have a locally accurate
model instead of a global one, and the divergence in some regions between the truth and
the local model may lead to misleading results. Therefore, in Chapter 5, we propose an
algorithm to handle the model divergence, by splitting the alternative space into multiple
local regions and fitting each region with a parametric function.
Finally, Chen et al. (2014) has no theoretical results on the convergence of KGDP, even under the assumption that one candidate is indeed the truth (i.e., the truth from prior assumption). In Chapter 3, we provide a proof showing that KGDP with truth from prior is able to identify the optimal alternative if we take an infinite number of measurements. We also prove that our resampling algorithm in Chapter 3 is capable of asymptotically finding not only the optimal alternative but also the true parameter. We further show in Chapter 4 that the same result holds for our the continuous KG policy when coupled with the resampling algorithm.
Chapter 3

Optimal Learning for Nonlinear Parametric Belief Models over Multidimensional Parameters

As discussed in Chapter 2, the KGDP model in Chen et al. (2014) uses a fixed set of parameters as candidates and assumes one of them is true. This method works well with low dimensional (say one or two) parameters but cannot be applied to multidimensional cases. We study how to tackle the limitation on the parameter space in this chapter. We propose a resampling algorithm in the parameter space, which dynamically discovers more probable parameters and replaces the less probable ones. We also propose a new metric to calculate the knowledge gradient (KG) score, aiming at identifying the true parameter quickly. We then study some properties of the KG policy with parametric models and prove that our methods are asymptotically optimal, followed by experiments on a variety of problems all showing fast empirical convergence rates.

The material of this chapter was presented in INFORMS Optimization Society (IOS) Conference 2016.
3.1 Introduction

We consider the following problem: we have a function over a finite number of alternatives, where the expression of the function is known except for some unknown parameters. Our goal is both to learn the correct parameters and to find the alternative that maximizes the function. Information can be collected by measuring the function value at any chosen alternative. However, the measurements are noisy and expensive, and we only have a limited budget to evaluate these alternatives. After exhausting the budget, we have to provide the best estimate of the unknown parameters and identify the best alternative to maximize our function.

As reviewed and discussed in Chapter 2, the work of Knowledge Gradient with Discrete Priors (KGDP) (Chen et al. 2014) proposes a basic framework to handle nonlinear belief models. However, it requires that we represent the uncertainty about the unknown parameters using a finite number of samples, where one of the samples has to be the true parameter. This assumption, which is also referred to as the truth from prior assumption, is too strong in most real world settings, especially when the parameters have four or more dimensions. Moreover, Chen et al. (2014) also fails to give any theoretical proof of the convergence of KGDP given the truth from prior assumption.

In this chapter, we present a resampling algorithm in the parameter space that works with KGDP to find not only the best alternative but also the correct parameters without the assumption of one candidate being the truth. We start with several potential candidates, and use the information from the measurements to guide resampling and discover more probable candidates. Regarding the objective of finding the correct parameter, we propose a new metric for calculating the knowledge gradient that minimizes the entropy of the candidate parameters. We also prove the convergence of both the non-resampling (with truth from prior) and the resampling KGDP algorithms. In the experimental section (Section 3.6), we apply our algorithm to a real problem in materials science: optimizing the kinetic stability of an experimental problem involving the control of a water-oil-water (W/O/W) nano emulsion.
(Chen et al. 2014). Compared with Chen et al. (2014), our algorithm shows significant improvements.

In order to find the correct parameter, we need to deal with the parameter space (denoted as the $\theta$ space) besides the alternative space (denoted as the $x$ space). Here arises a dual problem: in the $x$ space, we solve a maximization problem to figure out which alternative maximizes some performance metric (strength, conductivity, reflexivity); in the $\theta$ space, we solve another optimization problem to locate the most promising candidates. The second optimization problem is solved via minimizing the mean square error (MSE).

To the best of our knowledge, this is the first optimal learning work that addresses the dual problems of function maximization and parameter identification for problems with parametric belief models (other literatures with similar dual-objective formulations usually assume that experiments are inexpensive, e.g, Jiang & Shanbhag 2016). Previous papers only concentrate on discovering the optimal alternative, but in many real world situations, scientists also care about getting accurate estimates of the parameters. For example, materials scientists may want to know both the tunable variables of an experiment (such as the temperature, the pressure, and the density), and some unknown and uncontrollable parameters, which can help them understand the intrinsic nature of the physical phenomena. Our work in this chapter provides a powerful solution to this issue.

This work makes the following major contributions:

- We present a resampling algorithm in the parameter space, which works with the optimal learning methods in the alternative space, to discover both the optimal alternative and the correct parameters.

- We propose a new metric to calculate the knowledge gradient, which focuses on reducing the uncertainty of parameters by maximizing entropic loss.
We prove the asymptotic optimality of both the nonresampling (with truth from prior) and the resampling algorithms using either traditional function oriented metric or our new entropy oriented metric.

We show experimentally that our resampling algorithm has impressive performance in both finding the optimal alternative and estimating the parameters.

The remainder of the chapter is organized as follows. In Section 3.2, we introduce the resampling algorithm as well as the entropy oriented metric for calculating the knowledge gradient. Section 3.3 provides some insights on how KGDP works by discussing a simple two-candidate-function case. Section 3.4 provides the asymptotic convergence proof of KGDP without resampling under truth from prior assumption, and Section 3.5 proves asymptotic convergence of KGDP with resampling. We show empirical results on both simulated and real-world problems in Section 3.6, and conclude in Section 3.7.

3.2 Optimal Learning for Parameters and Alternatives

Assume we have a function $f(x; \theta)$, where $x \in \mathcal{X} = \{x_1, \ldots, x_M\}$, and $\theta$ is the unknown parameter. We want to learn the true parameter $\theta^*$ and the optimal $x$ that maximizes the true function $f(x; \theta^*)$, through a budget of $N$ noisy and expensive measurements. In other words, we solve the dual-objective optimization problem:

$$\max_{x \in \mathcal{X}} f(x; \theta^*)$$

$$\min_{\theta} \sum_{x \in \mathcal{X}} |f(x; \theta) - f(x; \theta^*)|^2$$

This dual-objective optimization problem requires handling the $x$ space and the $\theta$ space simultaneously: in the outer loop, we work in the $x$ space to collect as much information as possible from each measurement; in the inner loop, we search in the $\theta$ space for more promising $\theta$’s. Note that while finding the best alternative and learning the best estimate of
the parameter vector each have their own metrics, they are not conflicting goals, since the best estimate of the parameters can lead to the best alternative.

In this section, we first describe our resampling algorithm in the parameter space, and then introduce two Knowledge Gradient type of policies for selecting where to measure in the alternative space. The two policies, corresponding to the two objectives, focus on maximizing the function and identifying the true parameter, respectively.

### 3.2.1 Overview of the Resampling Scheme

As in KGDP, we still keep a set of \( L \) candidates, but they change over time. Let \( \mathbb{L}^n = \{\theta_1^n, ..., \theta_L^n\} \) denote the candidate set at time \( n \), and \( \vec{p}^n = \{p_1^n, ..., p_L^n\} \) denote their probabilities. Assume that the true function \( f(x; \theta^*) \) can be approximated by

\[
f(x; \theta^*) \approx \bar{f}^n(x) = \sum_{i=1}^{L} f(x; \theta_i^n)p_i^n.
\]

The candidates are no longer fixed and resampling happens periodically, in which the least probable candidates in \( \mathbb{L}^n \) get replaced by more probable ones according to the whole history of experiments. We hope to have access to a much larger population of \( \theta \)'s while only using a small number \( L \) as representatives to calculate the KG scores and approximate the function.

We start by choosing a large sample of \( K \) (say \( K = 1,000,000 \)) realizations of \( \theta \)'s, and call the \( K \) samples the large pool, denoted as \( \mathbb{K} \). The idea is that with such a large base population, it will be very likely that one of the \( \theta \)'s in \( \mathbb{K} \) will be the truth (or close to it). Then, we choose a small sample set \( \mathbb{L}^0 \) from \( \mathbb{K} \) as our sampled prior. As the candidates’ probabilities \( \vec{p}^n \) change over time, under certain conditions resampling is triggered and we replace less probable \( \theta \)'s in \( \mathbb{L}^n \) with more probable \( \theta \)'s from \( \mathbb{K} \). We use mean square error (MSE) as the criterion to choose \( \theta \)'s in the resampling steps.
3.2.2 Criterion for Resampling: Mean Square Error (MSE)

We assume the measurement noise $W$ is Gaussian, i.e., $W \sim \mathcal{N}(0, \sigma^2)$, i.i.d, where $\sigma$ is known. But note that the Gaussian noise assumption is not restrictive and our model still works for other types of noise. We use the mean square error (MSE) as the criterion to choose new $\theta$’s in resampling, which is equivalent to the maximum likelihood estimate (MLE) in Gaussian noise settings. Suppose that after $n$ measurements, the history of experimental results is $\mathcal{F}^n = \sigma\{(x^0, \hat{y}^1), ..., (x^{n-1}, \hat{y}^n)\}$. The likelihood of each $\theta_i$ is given by:

$$L(\theta_i | \mathcal{F}^n) = \prod_{j=1}^{n} \exp \left( -\frac{[\hat{y}^j - f(x^{j-1}; \theta_i)]^2}{2\sigma^2} \right).$$

By taking the logarithm of $L(\theta_i | \mathcal{F}^n)$, we have the MSE for $\theta_i$ as

$$MSE(\theta_i | \mathcal{F}^n) = \frac{1}{n} \sum_{j=1}^{n} [\hat{y}^j - f(x^{j-1}; \theta_i)]^2.$$

The idea is that each time resampling is triggered, we calculate the MSE of all $\theta$’s in the large pool $\mathbb{K}$. Although $\mathbb{K}$ can be arbitrarily large, this calculation is considered much faster and more efficient compared to the expensive physical experiments. Then resampling is conducted in the sub-level set of the MSE function $MSE(\theta)$. Specifically, we define a small pool of a certain size $S$ which contains the $S$ $\theta$’s with the smallest MSE, namely

$$\mathbb{S}^n = \{ \theta \in \mathbb{K} | MSE(\theta | \mathcal{F}^n) \leq MSE_S \},$$

where $MSE_S$ denotes the value of the $S$-th smallest $MSE$. In other words, the small pool $\mathbb{S}^n$ contains the $S$ most probable $\theta$’s up to now. We resample from this small pool later.

Note that the size of the small pool $S$ controls the trade-off of exploration and exploitation in $\theta$ space. On the one hand, this method rules out the majority of unlikely samples and avoids exploring too large a space inefficiently; on the other hand, it supplies a small pool containing sufficiently promising samples to resample from, thus providing plenty of exploration compared with simply selecting the $L$ samples with the smallest MSE.
3.2.3 Probability Updating in Resampling

For candidate $\theta^n_i$, suppose its prior probability is $p^n_i$. Conditioned on $\theta^n_i$ being the truth, after the $(n+1)$-th measurement given by $\hat{y}^{n+1} \sim \mathcal{N}(f(x^n; \theta^n_i), \sigma^2)$, the likelihood of $\theta^n_i$ is given by

$$g_Y(\hat{y}^{n+1} | \theta^* = \theta^n_i) = \frac{1}{\sqrt{2\pi \sigma}} \exp \left( -\frac{[\hat{y}^{n+1} - f(x^n; \theta^n_i)]^2}{2\sigma^2} \right).$$

By Bayes' rule, the posterior probability is proportional to the prior probability times the likelihood:

$$p^{n+1}_i \propto g_Y(\hat{y}^{n+1} | \theta^* = \theta^n_i) p^n_i = \frac{1}{\sqrt{2\pi \sigma}} \exp \left( -\frac{[\hat{y}^{n+1} - f(x^n; \theta^n_i)]^2}{2\sigma^2} \right) p^n_i. \quad (3.1)$$

After normalization, the updating rule of $p_i$ is given by

$$p^{n+1}_i = \frac{\exp \left( -\frac{[\hat{y}^{n+1} - f(x^n; \theta^n_i)]^2}{2\sigma^2} \right) p^n_i}{\sum_{l=1}^{L} \exp \left( -\frac{[\hat{y}^{n+1} - f(x^n; \theta^n_l)]^2}{2\sigma^2} \right) p^n_l}. \quad (3.2)$$

After the $(n+1)$-th measurement, we check if resampling is triggered (the resampling conditions and procedure are discussed in Section 3.2.4). If resampling is not triggered, $p^{n+1}_i$ can be updated as Equation (3.2) for $i = 1, 2, ..., L$. Otherwise, we should calculate the likelihood of $\theta_i$ by all the previous measurements, given by

$$g_Y(\hat{y}^1, ..., \hat{y}^n | \theta^* = \theta^{n+1}_i) = \prod_{j=0}^{n} \exp \left( -\frac{[\hat{y}^{j+1} - f(x^j; \theta^{n+1}_i)]^2}{2\sigma^2} \right). \quad (3.3)$$

Since initially all parameters are associated with equal probability, if resampling happens at time $(n+1)$, $p^{n+1}_i$ is given by

$$p^{n+1}_i = \frac{\prod_{j=0}^{n} \exp \left( -\frac{[\hat{y}^{j+1} - f(x^j; \theta^{n+1}_i)]^2}{2\sigma^2} \right)}{\sum_{l=1}^{L} \prod_{j=0}^{n} \exp \left( -\frac{[\hat{y}^{j+1} - f(x^j; \theta^{n+1}_l)]^2}{2\sigma^2} \right).} \quad (3.4)$$

where $i$ and $l$ index the set of candidates after resampling.
3.2.4 Resampling Procedure

Resampling is triggered under either of the two conditions: 1) the same set of candidates have been used for \( n^{resamp} \) iterations; 2) more than \( L/2 \) candidates have probabilities lower than a predefined threshold \( \epsilon \) (note that \( \epsilon < \frac{1}{L} \)). The resampling process goes as follows.

1. When resampling is triggered, we first remove the candidates with low probabilities. Denote the set as \( L^r_{nm} \):

\[
L^r_{nm} = \{ \theta \in L^n | p^n(\theta) \leq \epsilon \}.
\]

Note that if \( p^n(\theta) > \epsilon \) for all \( \theta \in L^n \), we still select a small portion (say 1 or 2) of the least probable ones as \( L^r_{nm} \). In this way, we can avoid getting stuck in a set of unlikely candidates (for example, in an extreme case where we have \( L \) identical but wrong candidates, we get stuck if not dropping any candidate).

2. Then we calculate the MSE of each \( \theta \) in the large pool \( K \) by Equation (3.1), and select \( S \) ones with the smallest MSE to form the small pool.

3. Next, we calculate the likelihood of each \( \theta \) in the small pool \( S^n \) according to Equation (3.3). Since we assume a uniform prior distribution for \( \theta \) at time 0, the likelihood is proportional to the posterior probability density of \( \theta \) at time \( n \). Hence, we then use the likelihoods as weights, and do weighted sampling without replacement to select \( |L^r_{nm}| \) \( \theta \)’s and add them to the candidate set.

4. Once the candidates are updated, we update the probabilities of all the candidates using the whole measurement history according to Equation (3.4).

5. Finally, we check if the current set can satisfy the terminating condition. If more than \( \frac{3}{4} L \) candidates have probabilities larger than \( \epsilon \), we finish resampling. Otherwise, repeat the resampling process.

A detailed flowchart is given in Section 3.2.6.
3.2.5 Evaluation Metrics

Traditionally, the optimal learning literature has focused on optimizing some metric such as choosing the best control variables (i.e., $x$) to maximize the strength of a material or minimize the number of defects. When we use parametric belief models, we can achieve these goals by finding good estimates of the unknown parameters, and then optimizing a (deterministic) nonlinear model based on the estimates. Therefore, we introduce two Knowledge Gradient type of policies for choosing the next alternative to measure. The two policies, corresponding to our dual objectives, focus on maximizing the performance metric and learning the true parameter, respectively. The first one, initially given in Chen et al. (2014), is function value oriented and hence denoted as KGDP-$f$. The second one, denoted as KGDP-$H$, focuses on minimizing the entropy of the belief vector $\tilde{p}^n = \{p^n_1, ..., p^n_L\}$, which leads to a better estimate of $\theta^*$, from which we can then optimize the original function $f(x; \theta)$.

KGDP-$f$

We can measure the expected incremental function value as in Chen et al. (2014), where the formula of the KGDP-$f$ is originally given. At time $n$, we define the state of knowledge $S^n$ as the current set of candidate $\theta$’s and their probabilities, i.e, $S^n = (\mathbb{L}^n, \tilde{p}^n)$, and $V^n(S^n)$ as the current largest estimate, i.e, $\max_{x \in \mathcal{X}} \tilde{f}^n(x)$ (recall that $\tilde{f}^n(x)$ is our estimate of $f(x; \theta^*)$ at time $n$, given by $\tilde{f}^n(x) = \sum_{i=1}^L f(x; \theta^n_i)p^n_i$). Let $p^{n+1}(x)$ represent the posterior probability after measuring $x$. Let $f_i(x) = f(x; \theta^n_i)$, and KGDP-$f$ is calculated in Chen et al. (2014) as:

$$\nu^{KGDP-f,n}(x) = \mathbb{E}^n \left[ \max_{x'} \sum_{i=1}^L f_i(x')p^{n+1}_i(x) \right] - \max_{x'} \sum_{i=1}^L f_i(x')p^n_i.$$

Here we have two sources of uncertainty: the uncertainty on $\theta$ and they uncertainty on the noise $W$. Hence, $\mathbb{E}^n$ is the expectation over two parts, and we have
\[ \nu_{\text{KGDP}-f,n}(x) = \mathbb{E}_\theta^n \left\{ \mathbb{E}_\omega \left[ \max_{x'} \sum_{i=1}^L f_i(x') p_i^{n+1}(x) \mid \theta_j = \theta^n \right] \right\} - \max_{x'} \sum_{i=1}^L f_i(x') p_i^n \]

\[ = \sum_{j=1}^L \left[ \int \omega \max_{x'} \sum_{i=1}^L f_i(x') p_{ij}^{n+1}(x,\omega) g(\omega) d\omega \right] p_j^n - \max_{x'} \sum_{i=1}^L f_i(x') p_i^n, \tag{3.5} \]

where \( g(\omega) = \frac{1}{\sqrt{2\pi\sigma}} \exp \left( -\frac{\omega^2}{2\sigma^2} \right) \) is the density of \( W \), and \( p_{ij}^{n+1}(x,\omega) \) is the posterior probability of \( \theta_i^n \) given that \( \theta_j^n \) is the truth and the noise is \( \omega \). By Bayes rule, we have:

\[ p_{ij}^{n+1}(x,\omega) = \frac{\exp \left( -\frac{[f_j(x)-f_i(x)+\omega]^2}{2\sigma^2} \right) p_i^n}{\sum_{l=1}^L \exp \left( -\frac{[f_j(x)-f_l(x)+\omega]^2}{2\sigma^2} \right) p_l^n}. \tag{3.6} \]

In each integral, let \( f_j(x) + \omega = \hat{y} \), and the above equation can be simplified as

\[ \nu_{\text{KGDP}-f,n}(x) = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{+\infty} \max_{x'} \left[ \sum_{i=1}^L f_i(x') p_i^n \exp \left( -\frac{[\hat{y} - f_i(x)]^2}{2\sigma^2} \right) \right] d\hat{y} \]

\[ - \max_{x'} \sum_{i=1}^L f_i(x') p_i^n, \tag{3.7} \]

where \( i \) indexes the candidates at time \( n \), and \( \hat{y} \) can be interpreted as the measurement result. Compared with Equation (3.5) in Chen et al. (2014), Equation (3.7) is simpler for both calculation and theoretical proof.

In KGDP-f, our decision at time \( n \) is

\[ x^n = \arg\max_{x \in \mathcal{X}} \nu_{\text{KGDP}-f,n}(x). \]

**KGDP-H**

Entropy is a metric to measure the uncertainty of unknown data, which is widely used especially in information theory. The entropy of the candidates at time \( n \) is given by

\[ H(p_1^n, \ldots, p_L^n) = -\sum_{i=1}^L p_i^n \log p_i^n. \]
In KGDP-H, we measure the alternative that has the largest expected entropic loss. Define the state variable $S^n = (\mathbb{L}^n, \mathbb{p}^n)$, and let $V^n(S^n)$ be the negative entropy of the current candidate set. The KGDP-H score is given by

$$\nu_{KGDP-H,n}(x) = \mathbb{E}^n \left[ \sum_{i=1}^L p_{i}^{n+1}(x) \log p_{i}^{n+1}(x) \right] - \sum_{i=1}^L p_{i}^n \log p_{i}^n \tag{3.8}$$

where $g(\omega) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{\omega^2}{2\sigma^2} \right)$, and $p_{i}^{n+1}(x, \omega)$ is the posterior probability of $\theta_i$ conditioning on $\theta_j^n = \theta^*$, given in Equation (3.6).

Similar to Equation (3.7), we can simplify the equation above as:

$$\nu_{KGDP-H,n}(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} \sum_{i=1}^L p_{i}^n \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] \log p_{i}^{n+1}(x) d\hat{y} - \sum_{i=1}^L p_{i}^n \log p_{i}^n, \tag{3.9}$$

where $p_{i}^{n+1}(x)$ is given by Equation (3.2).

In KGDP-H, our decision at time $n$ is

$$x^n = \arg\max_{x \in \mathcal{X}} \nu_{KGDP-H,n}(x).$$

After we exhaust the budget of $N$ experiment, we give our estimates of the optimal alternative $\hat{x}^N$ and parameters $\hat{\theta}^N$ by:

$$\hat{x}^N = \arg\max_{x \in \mathcal{X}} \bar{f}^N(x) = \arg\max_{x \in \mathcal{X}} \sum_{i=1}^L p_i^N f(x; \theta_i^N), \quad \hat{\theta}^N = \arg\min_{\theta \in \mathcal{K}} \text{MSE}(\theta | \mathcal{F}^N).$$

29
3.2.6 Detailed Flowchart

The detailed flow of the whole procedure is shown in Algorithm 1. In particular, the function select_x(policy,...) uses the designated policy to select an alternative to measure, and the flowchart of our experiment decision select_x() is shown in Algorithm 2. In select_x(), besides the KGDP-f and KGDP-H policies, we also include three competing policies used in our experiments, namely (1) Pure Exploration, which chooses x_n randomly; (2) Pure Exploitation, which always chooses the current best alternative; and (3) Max Variance (Max-Var), which picks the alternative that has the largest variance under current belief.

3.3 Discussion of a Two-curve Model

To gain some insights into how the KGDP policy works, we first study the simplest case, where there are only two candidate functions. Let f_1(x) = f(x; θ_1) and f_2(x) = f(x; θ_2) be the two functions, and p_1^n and p_2^n be their probabilities at time n, respectively.

Proposition 3.3.1. In the two-curve case,

1. The KGDP-f score is an increasing function of the distance between the two functions. Hence, it always measures x^n ∈ argmax_{x∈X} |f_1(x) − f_2(x)|. The only exception is when f_1 and f_2 achieves their maximum values at the same x, i.e, argmax_x f_1(x) ∩ argmax_x f_2(x) ≠ ∅, in which case the KGDP-f score at any x is 0.

2. The KGDP-H score is an increasing function of the distance between the two functions. Hence, it always measures x^n = argmax_{x∈X} |f_1(x) − f_2(x)|.

Sketch of proof. (See Appendix 3.8.1 for full proof.) First, in a two-curve model, Equation (3.7) and Equation (3.9) can both be rewritten as functions of the distance, i.e, h(d) where d = |f_1(x) − f_2(x)|. Then, the basic idea is to take the derivative of h(d) with respect to d and show \( \frac{dh(d)}{dd} \geq 0 \) for d ≥ 0.
Algorithm 1 Flow of the resampling algorithm

**Input:** Budget: $N$; Alternatives: $\mathcal{X} = \{x_1, ..., x_M\}$; Noise: $\sigma$; Large pool: $\mathbb{K}$; Size of the small pool: $S$; Number of candidates: $L$; Resample stepsize: $n^{\text{resamp}}$; Threshold for probability: $\epsilon$; The policy to select $x^n$.

**Output:** Estimate of the optimal alternative: $\hat{x}^N$; Estimate of the true parameter: $\hat{\theta}^N$.

1. Choose $L$ samples out of $\mathbb{K}$ randomly, as $L^0$.
2. Set $p_1^0 = ... = p_L^0 = 1_L$.
3. Set $\text{count} = 0$ ($\text{count}$ is the number of iterations the current set of $\theta$'s has been used).
4. for $n = 0$ to $N - 1$ do
5.   $x^n = \text{select } x(\text{policy, } \mathcal{X}, p^n, L^n)$.
6.   Take a measurement: $\hat{y}^{n+1}$.
7.   Update each $p_i$ using:
$$p_i^{n+1} = \frac{\exp \left[ - \frac{(\hat{y}^{n+1} - f_i(x))^2}{2\sigma^2} \right]}{\sum_{l=1}^L \exp \left[ - \frac{(\hat{y}^{n+1} - f_l(x))^2}{2\sigma^2} \right]} p_l^n. $$
8.   Set $\text{count} = \text{count} + 1$, $L^{n+1} = L^n$.
9.   if $\text{count} = 0 \pmod{n^{\text{resamp}}}$ or $|\{\theta_i \in L^{n+1} | p_i^{n+1} \leq \epsilon\}| \geq L/2$ then
10.   Calculate MSE for all the $K$ samples.
11.   Construct $\mathcal{S}^{n+1}$ as the set of $S$ $\theta$'s with the smallest MSE.
12.   Calculate the likelihood of each $\theta$ in $\mathcal{S}^{n+1}$.
13.   if $\min_i (p_i^{n+1}) > \epsilon$ then
14.     Find the least two probable $\theta$'s in $\mathcal{L}^{n+1}$, set their probabilities as 0.
15.   end if
16.   while $|\{\theta_i \in L^{n+1} | p_i^{n+1} \leq \epsilon\}| > L/4$ do
17.     Construct $\mathcal{L}^{n+1}_{\text{rm}} = \{\theta_i \in L^{n+1} | p_i^{n+1} \leq \epsilon\}$, let $L^{n+1} = L^{n+1} \setminus \mathcal{L}^{n+1}_{\text{rm}}$.
18.     Select $|\mathcal{L}^{n+1}_{\text{rm}}|$ $\theta$'s from $\mathcal{S}^{n+1}$ by weighted sampling without replacement according to their likelihoods, add them to $\mathcal{L}^{n+1}$.
19.     Update posterior probabilities for all $\theta$'s in $\mathcal{L}^{n+1}$:
$$p_i^{n+1} = \frac{\prod_{j=0}^n \exp \left[ - \frac{(\hat{y}^{j+1} - f(x; \theta_i^{j+1}))^2}{2\sigma^2} \right]}{\sum_{l=1}^L \prod_{j=0}^n \exp \left[ - \frac{(\hat{y}^{j+1} - f(x; \theta_l^{j+1}))^2}{2\sigma^2} \right]}.$$ 
20.   end while
21.   Set $\text{count} = 0$.
22. end if
23. end for
24. return $\hat{x}^N = \arg \max_{x \in \mathcal{X}} \sum_{i=1}^L p_i^N f(x; \theta_i^N)$, $\hat{\theta}^N = \arg \min_{\theta \in \mathbb{K}} \text{MSE}(\theta | \mathcal{F}^N)$. 

31
Algorithm 2 Choose the next alternative $x^n$.

**Input:** Policy: Pure Exploration, Pure Exploitation, Max-Var, KGDP-$f$, KGDP-$H$; Alternatives: $\mathcal{X} = \{x_1, \ldots, x_M\}$; Probability vector $\vec{p}^n = \{p^n_1, \ldots, p^n_L\}$; $L$ candidates $\mathbb{L}^n = \{\theta^n_1, \ldots, \theta^n_L\}$.

**Output:** The alternative to measure next: $x^n$.

1: switch (policy)
2: case KGDP-$f$:
3: $x^n = \arg\max_{x \in \mathcal{X}} \nu_{KGDP-f,n}(x)$.
4: case KGDP-$H$:
5: $x^n = \arg\max_{x \in \mathcal{X}} \nu_{KGDP-H,n}(x)$.
6: case Pure Exploration:
7: $x^n = \text{rand}(\{x_1, \ldots, x_M\})$.
8: case Pure Exploitation:
9: $x^n = \arg\max_{x \in \mathcal{X}} \bar{f}^n(x)$.
10: case Max-Var:
11: $x^n = \arg\max_{x \in \mathcal{X}} \sum_{j=1}^L p_j [f(x; \theta^n_j) - \bar{f}^n(x)]^2$.
12: end switch
13: return $x^n$.

For KGDP-$H$, the derivative is easy to calculate, which can further be shown to be nonnegative. For KGDP-$f$, it is trickier due to the $\max$ embedded in the expectation. Note that in Equation (3.7), different $\hat{y}$’s can result in different $x'$’s in the $\max$ function. The trick is that we divide the $\hat{y}$ axis into a finite number of intervals, such that on each interval the same $x'$ is picked. Then we can take the derivative and show it is nonnegative, too.

In conclusion, when the two curves are not “aligned,” namely they achieve maxima at different $x$’s, as illustrated in Figure 3.1 (a), KGDP-$f$ and KGDP-$H$ are equivalent since they both measure the point where the functions differ the most. By intuition, this is also the easiest way to distinguish the two curves. In this case, the KGDP-$f$ and KGDP-$H$ scores against the distance are shown in Figure 3.1 (b). However, when the two functions are aligned as illustrated in Figure 3.2 (a), KGDP-$H$ still measures the point with the largest distance, while KGDP-$f$ reduces to pure exploration since all alternatives have KGDP-$f$ scores as 0, as shown in Figure 3.2 (b), and as a result measures randomly. Intuitively, this is due to the fact that the value of information depends on the ability to change a decision.
In the aligned case, the optimal \( x \) is trivially found, and measuring any point provides us with zero further information in terms of discovering a better alternative. This is proved rigorously in Lemma 3.4.3, which satisfies the equality condition for KGDP-\( f \) to be 0.

![Diagrams](https://example.com/diagrams.png)

**Figure 3.1:** Unaligned functions and KGDP-\( f \), KGDP-\( H \) scores in the two-curve model.

![Diagrams](https://example.com/diagrams.png)

**Figure 3.2:** Aligned functions and KGDP-\( f \), KGDP-\( H \) scores in the two-curve model.

### 3.4 Convergence of KGDP without Resampling

By construction, KGDP-\( f \) and KGDP-\( H \) are naturally the optimal myopic policies to find the optimal alternative and the true parameter (in terms of entropy), respectively. In this section, we show that KGDP-\( f \) and KGDP-\( H \) with truth from prior are asymptotically optimal in the non-resampling scenario, a result not shown in Chen et al. (2014) where KGDP is first introduced. In other words, we show that as the budget \( N \to \infty \), KGDP-\( f \)
will learn the optimal alternative almost surely, and that KGDP-H will find the correct parameters (and thus also the optimal alternative) almost surely.

We denote the finite set of alternatives as $X$. We define $(\Omega, \mathcal{F}, P)$ as the probability space, where $\Omega$ is the set of all possible measurement histories $\{(x^0, \hat{y}^1), \ldots, (x^{N-1}, \hat{y}^N)\}$, and $\mathcal{F}$ is the $\sigma$-algebra generated by the history. In this section, $N = \infty$.

To prove asymptotic optimality, we first show that if an alternative $x$ is measured infinitely often, we can learn its true function value almost surely. As $N$ goes to infinity, there will be a subset of alternatives that are evaluated infinitely often. We prove that under the KGDP-H policy with truth from prior, $f(x, \theta^*)$ is the only function that fits the true function values of this subset, while under KGDP-f, all functions that fit this subset achieve their maximums at the same $x$ (we refer to this scenario as the functions being aligned). To show this, we first reveal the nonnegativity of KGDP-f and KGDP-H scores, and the fact that any alternative measured infinitely often has its KGDP-f and KGDP-H scores converge to 0. Then we use proof by contradiction, and show that under the contrary assumption, there would exist $x$ which is measured finitely often while KGDP-f or KGDP-H scores do not converge to 0. This is contrary to the fact that either KGDP-f or KGDP-H policy chooses the alternative with the largest score.

We begin by showing that if we measure a point $x$ infinitely often, the correct function value at $x$ will be found almost surely:

**Lemma 3.4.1.** Let $N^n(x)$ be the number of measurements taken on $x$ when the total number of measurements is $n$. If $N^n(x) \to \infty$ as $n \to \infty$, then for $\forall l \in \{1, 2, \ldots, L\}$ such that $f(x; \theta_l) \neq f(x; \theta^*)$, $p^n_l \to 0$ almost surely, i.e, $\mathbb{P}(\lim_{n \to \infty} p^n_l = 0) = 1$.

The rigorous proof of Lemma 3.4.1 comes from the strong law of large numbers (shown in Appendix 3.8.2).

We denote the set of the almost sure events defined by Lemma 3.4.1 as $\Omega_0$. For any $\omega \in \Omega$, we denote the set of alternatives as $X_\infty(\omega)$ that are measured infinitely often. Hence,
for any $\omega \in \Omega_0$, we have $p^n(\theta)(\omega) \to 0$ for any $\theta$ such that there exists $x \in \mathcal{X}_\infty(\omega)$ such that $f(x; \theta) \neq f(x; \theta^*)$.

We hope that we can learn $x^*$ (and $\theta^*$) via $\mathcal{X}_\infty$. To achieve this goal, we first study some properties of the KGDP-$f$ and KGDP-$H$ scores.

**Lemma 3.4.2.** Suppose a subset of candidate functions intersect at a particular $x \in \mathcal{X}$, and without loss of generality, suppose they are $f(x; \theta_1) = \ldots = f(x; \theta_h)$. Then calculating $\nu^{KGDP-f,n}(x)$ using $(f_1, \ldots, f_n)$ with probabilities $(p^n_1, \ldots, p^n_L)$ is equivalent to calculating $\nu^{KGDP-f,n}(x)$ using $(f_r, f_{h+1}, \ldots, f_n)$ with probabilities $(p^n_r, p^n_{h+1}, \ldots, p^n_L)$, where $f_r(x') = \frac{\sum_{i=1}^{h} f(x') p^n_i}{\sum_{i=1}^{r} p^n_i}$ for all $x' \in \mathcal{X}$ and $p^n_r = \sum_{i=1}^{h} p^n_i$. We call $f_r$ the reduced model at $x$.

**Sketch of proof.** (See Appendix 3.8.2 for full proof.) Basically, if $f(x; \theta_1) = f(x; \theta_2)$, measuring $x$ would not change the ratio of their probabilities, i.e., $\frac{p^n_1}{p^n_2} = \frac{p^n_{h+1}}{p^n_r}$. Thus, the updating rule, namely Equation (3.1), still holds for $p^n_r$ and $p^n_{r+1}$. Also, the distribution of the measurement result $\hat{y}$ in the reduced model is the same as that in the full model. Therefore, $\nu^{KGDP-f,n}(x)$ is the same.

We then show that the value of information for both objectives is always nonnegative:

**Lemma 3.4.3.** For $\forall n \geq 0, \forall x \in \mathcal{X}$, the KGDP-$f$ score $\nu^{KGDP-f,n}(x) \geq 0$. Equality holds if and only if (1) there exists $x'$ such that $x' \in \text{argmax}_{x_0} f_r(x_0)$ for all reduced model $f_r$, at $x$ with $p^n_r > 0$, or (2) all functions with $p^n > 0$ have the same value at $x$.

**Lemma 3.4.4.** For $\forall n \geq 0, \forall x \in \mathcal{X}$, the KGDP-$H$ score $\nu^{KGDP-H,n}(x) \geq 0$. Equality holds if and only if all functions with $p^n > 0$ have the same value at $x$.

**Sketch of proof for Lemma 3.4.3 and Lemma 3.4.4.** (See Appendix 3.8.2 for full proof.) First, we show $\mathbb{E}^n[p^n_{r+1}(x)] = p^n_r$. Then, applying Jensen’s inequality will give us the nonnegativity of both KGDP-$f$ and KGDP-$H$.

According to Lemma 3.4.3 and Lemma 3.4.4, KGDP-$f$ equals 0 for a particular $x$ in only two cases: (1) either when the reduced functions at $x$ are aligned (i.e, share the same
as the maximum point), (2) or when the functions with nonzero probabilities have the same value at \(x\). Specifically, if all the candidate functions are aligned, \(\nu^{\text{KGDF}-f,n}(x) = 0\) for all \(x \in \mathcal{X}\). For KGDP-H, \(\nu^{\text{KGDP}-H,n}(x) = 0\) only when all functions with positive probabilities intersect at \(x\).

We then show that for any \(x\) measured infinitely often, its KGDP-\(f\) and KGDP-\(H\) scores also go to zero:

**Lemma 3.4.5.** For \(\forall \omega \in \Omega_0\) and \(x \in \mathcal{X}_\infty(\omega)\), we have \(\lim_{n \to \infty} \nu^{\text{KGDP}-f,n}(x)(\omega) = 0\) under the KGDP-\(f\) policy, and \(\lim_{n \to \infty} \nu^{\text{KGDP}-H,n}(x)(\omega) = 0\) under the KGDP-\(H\) policy.

**Sketch of proof.** (See Appendix 3.8.2 for full proof.) Intuitively, as we measure \(x\) infinitely often and learn the true value gradually, the condition for both KGDP-\(f\) and KGDP-\(H\) being 0 is satisfied in the limit (given by Lemma 3.4.3 and Lemma 3.4.4). The full proof, as shown in Appendix 3.8.2, reveals the fact that given a fixed \(x\), \(\nu^{\text{KGDP}-f}(x)\) and \(\nu^{\text{KGDP}-H}(x)\) are both uniformly continuous in the probability vector space \(\vec{p}^n = (p^n_1, ..., p^n_L)\). Combined with the convergence result of \(p^n_i\) shown by Lemma 3.4.1, we conclude \(\nu^{\text{KGDP}-f}(x)\) and \(\nu^{\text{KGDP}-H}(x)\) also converge to 0.

We now define two concepts on a subset of \(\mathcal{X}\):

**Definition 3.4.6.** We define an \(x\)-sufficient set \(\mathcal{X}_s \subset \mathcal{X}\) as follows: for any \(\theta\) such that \(\arg\max_{x \in \mathcal{X}} f(x; \theta) \neq x^*\), there exists \(x \in \mathcal{X}_s\) such that \(f(x; \theta) \neq f(x; \theta^*)\).

**Definition 3.4.7.** We define a \(\theta\)-sufficient set \(\mathcal{X}_s \subset \mathcal{X}\) as follows: for \(\forall \theta \neq \theta^*\), there exists \(x \in \mathcal{X}_s\) such that \(f(x; \theta) \neq f(x; \theta^*)\).

Basically, for any subset of alternatives, if it is sufficient to infer \(x^*\) by fitting from the true function values of the subset, we call it an \(x\)-sufficient set; if it is sufficient to infer \(\theta^*\) by fitting from the true function values of the subset, we call it a \(\theta\)-sufficient set. In other words, for any subset \(\mathcal{X}_s = \{x_1, ..., x_m\} \subset \mathcal{X}\), if all \(f(x; \theta^*)\)'s that fit the \(m\) points \((x_1, f(x_1; \theta^*)), ..., (x_m, f(x_m; \theta^*))\) achieve their maximum at \(x^*\), then \(\mathcal{X}_s\) is an \(x\)-sufficient set.
set; if \( f(x; \theta^*) \) is the only function that fits the \( m \) points \( (x_1, f(x_1; \theta^*)), \ldots, (x_m, f(x_m; \theta^*)) \), then \( \mathcal{X}_s \) is a \( \theta \)-sufficient set. Therefore, a \( \theta \)-sufficient set is always an \( x \)-sufficient set; if the functions are pairwise unaligned, an \( x \)-sufficient set is also a \( \theta \)-sufficient set.

We show that under the KGDP-\( f \) policy, we measure an \( x \)-sufficient set infinitely often almost surely, while under the KGDP-\( H \) policy, we measure a \( \theta \)-sufficient set infinitely often almost surely:

**Lemma 3.4.8.** For any \( \omega \in \Omega_0 \), the alternatives measured infinitely often under the KGDP-\( f \) policy constitute an \( x \)-sufficient set. Specially, if any two candidate functions are not aligned, the alternatives measured infinitely often under the KGDP-\( f \) policy constitute an \( \theta \)-sufficient set.

**Lemma 3.4.9.** For any \( \omega \in \Omega_0 \), the alternatives measured infinitely often under the KGDP-\( H \) policy constitute a \( \theta \)-sufficient set.

**Sketch of proof for Lemma 3.4.8 and Lemma 3.4.9.** (See full proof in Appendix 3.8.2.) For any \( \omega \in \Omega_0 \), assume \( h \) functions, \( f_1, f_2, \ldots, f_h \), fit the true values of \( \mathcal{X}_\infty(\omega) \). We first show that for any \( 1 \leq l \leq h \), \( \lim_{n \to \infty} p_l^n(\omega) > 0 \) exists, while for any \( l > h \), \( \lim_{n \to \infty} p_l^n(\omega) = 0 \). Then we assume the contrary for both Lemma 3.4.8 and Lemma 3.4.9. Take KGDP-\( H \) for example. We show there exists some \( x \notin \mathcal{X}_\infty(\omega) \), such that \( \lim_{n \to \infty} \nu_{KGDP-H,n}(x)(\omega) > 0 \) exists. However, we have \( \lim_{n \to \infty} \nu_{KGDP-H,n}(x)(\omega) = 0 \) for any \( x \in \mathcal{X}_\infty(\omega) \). This contradicts with the fact that the KGDP-\( H \) policy always measures the alternative with the largest KGDP-\( H \) score. The proof of Lemma 3.4.8 is similar.

Using the previous lemmas, we can conclude with the following asymptotic optimality results:

**Theorem 3.4.10.** The non-resampling KGDP-\( f \) policy with truth from prior is asymptotically optimal in finding the optimal alternative. Furthermore, if the candidate functions are pairwise unaligned, then KGDP-\( f \) is also asymptotically optimal in finding the correct parameter.
Remark 3.4.11. If some functions are aligned, the non-resampling KGDP-$f$ policy may not find the correct parameter. We provide an example in Appendix 3.8.2.

Theorem 3.4.12. The non-resampling KGDP-$H$ policy with truth from prior is asymptotically optimal in finding both the optimal alternative and the correct parameter.

The detailed proofs of Theorem 3.4.10 and Theorem 3.4.12 are given in Appendix 3.8.2.

3.5 Convergence of KGDP with Resampling

We now establish the important result that the resampling algorithm converges to the true optimal solution, in terms of both identifying the right parameter $\theta^*$ and finding the optimal design $x^*$.

As we move from the non-resampling case to the resampling one, the difference is that the candidates are no longer fixed, and $\theta^*$ is not guaranteed to be one of the candidates.

Note that the nonnegativity of the knowledge gradient established in Lemma 3.4.3 and Lemma 3.4.4 still holds in the resampling case, regardless of whether the candidates include the truth or not.

We denote the probability space in the resampling case as $(\Omega_R, \mathcal{F}_R, \mathbb{P}_R)$. But note that unlike the non-resampling case, we have two sources of randomness: the measurement noise and the resampling procedure.

Let $\mathbb{K}$ denote the large pool, i.e, the set of the $K \theta$’s. We can adapt the concept of a sufficient set by considering the entire set $\mathbb{K}$. For a subset of alternatives $\mathcal{X}_s \subset \mathcal{X}$, if $f(x; \theta^*)$ is the only function among all the $K$ functions that fits the true values of alternatives in $\mathcal{X}_s$, we call it a $\theta$-sufficient set.

Unlike the non-resampling case, since the candidates may keep changing, the limit of a particular $p^n_t$ may not exist. However, if we consider the whole $\mathbb{K}$ and assume each $\theta \in \mathbb{K}$
has the probability \( w^n(\theta) \), given by:

\[
    w^n(\theta) = \frac{\prod_{j=0}^{n-1} \exp \left[ -\frac{(\hat{y}^j + 1 - f(x; \theta))^2}{2\sigma^2} \right]}{\sum_{\theta' \in \mathcal{K}} \prod_{j=0}^{n-1} \exp \left[ -\frac{(\hat{y}^j + 1 - f(x; \theta'))^2}{2\sigma^2} \right]},
\]

then (the resampling version of) Lemma 3.4.1 still holds, demonstrating that conducting an infinite number of measurements on \( x \) reveals its true value almost surely. We define \( \Omega_1 \) as the set of almost sure events in \( \Omega_R \) for the resampling version of Lemma 3.4.1 to hold.

For any \( \omega \), let \( \mathcal{X}_\infty(\omega) \) denote the set of alternatives measured infinitely often. We have the following lemma for \( \mathcal{X}_\infty(\omega) \):

**Lemma 3.5.1.** There exists \( \Omega_2 \in \Omega_1 \), \( P(\Omega_2) = P(\Omega_1) = 1 \), such that for any \( \omega \in \Omega_2 \), the alternatives measured infinitely often (i.e, \( \mathcal{X}_\infty(\omega) \)) under the resampling KGDP-\( f \) or KGDP-\( H \) policy constitute a \( \theta \)-sufficient set.

Note that Lemma 3.5.1 is similar to Lemma 3.4.8 and Lemma 3.4.9, but here for KGDP-\( f \), we no longer require the candidate functions to be pairwise unaligned as a result of the resampling procedure. We give the proof sketch of Lemma 3.5.1 below.

**Sketch of proof for Lemma 3.5.1.** (See Appendix 3.8.3 for full proof.) Assume the contrary. Then there exists at least one \( \theta \neq \theta^* \), such that \( f(x; \theta) = f(x; \theta^*) \) for any \( x \in \mathcal{X}_\infty(\omega) \). Denote the set of such \( \theta \)'s as \( \mathcal{K}'(\omega) \). We can show that it happens almost surely that at least two distinct \( \theta \)'s out of \( \mathcal{K}' \cup \{ \theta^* \} \) are included in the candidate set \( \mathcal{L}^n \) infinitely often. At these times, we can further show that there exists an \( x \notin \mathcal{X}_\infty(\omega) \) and \( \epsilon > 0 \) such that

\[
    \nu_{\text{KGDP-}f,n}(x) > \epsilon \quad \text{and} \quad \nu_{\text{KGDP-}H,n} > \epsilon,
\]

while on the other hand, for any \( x \in \mathcal{X}_\infty(\omega) \),

\[
    \lim_{n \to \infty} \nu_{\text{KGDP-}f,n}(x) = 0 \quad \text{and} \quad \lim_{n \to \infty} \nu_{\text{KGDP-}H,n}(x) = 0.
\]

This is contradictory to the fact that any \( x \notin \mathcal{X}_\infty(\omega) \) is measured for a finite number of times. \( \square \)
Lemma 3.5.1 implies the following theorem, which is also our main result in this section:

**Theorem 3.5.2.** The resampling KGDP-\(f\) policy is asymptotically optimal in finding the optimal alternative and the correct parameter. The same holds for the resampling KGDP-\(H\) policy.

**Sketch of proof.** (See Appendix 3.8.3 for full proof.) The crucial step is to show that with probability 1, \(\theta^*\) appears in the candidate set \(\mathbb{L}^n\) for all but a finite number of times. To prove this, we first prove it happens almost surely that \(\theta^* \in \mathbb{L}^n\) infinitely often. Then according to the resampling procedure, once \(\theta^*\) is included in \(\mathbb{L}^n\), since \(w(\theta^*) \to 1\), \(\theta^*\) will never be dropped. Hence, we have \(\theta^* \in \mathbb{L}^n\) for all but a finite number of times almost surely. Therefore, \(\lim_{n \to \infty} \tilde{f}(x; \theta) = f(x; \theta^*)\) for any \(x\) and the proof is done.

\(\square\)

### 3.6 Empirical Results

In this section, we study 1) an asymmetric unimodal synthetic problem, 2) a material science problem, known as the nanoemulsion problem in a water-oil-water (W/O/W) model, and 3) several test functions to show the performance on a larger variety of domains. In each experiment, we sample a sufficiently large \(\mathbb{K}\) based on the dimensionality of \(\theta\) and set one \(\theta \in \mathbb{K}\) as the truth.

We mainly demonstrate results from the following aspects:

- The visualization of the resampling process, including the evolution of the candidates, the small pool, and the approximated function \(\tilde{f}^n(x)\);

- The performance of resampling KGDP-\(f\) and KGDP-\(H\), both in relation to each other and their absolute performance in terms of different metrics;

- The relative performance of resampling KGDP-\(f\) and KGDP-\(H\) policies versus competing policies;
• The comparison of resampling and non-resampling KGDP methods;

• Most importantly, the empirical rate of convergence under different noise and dimensionality settings.

3.6.1 An Asymmetric Unimodal Benchmark Function

We study a multidimensional benchmark function, which produces a wide range of asymmetric, unimodular functions, which we have found to be a challenging class of test problems. The function is given by

$$f(x_1, ..., x_k) = \sum_{i=1}^{k} \eta_{1,i} \mathbb{E} \left[ \min \left( x_i, \left( D - \sum_{j=1}^{i-1} x_j \right)^+ \right) \right] - \sum_{i=1}^{k} \eta_{2,i} x_i,$$

where $x_1, ..., x_k$ are the decision vector, $D$ is a uniformly distributed random variable, $\eta_{1,i}$ are fixed constants, and $\eta_{2,i}$ are unknown parameters. In the following experiments, we fix the variance of $D$, but set its mean as well as $\eta_{2,i}$’s as parameters ($\theta$), which results in a $(k + 1)$-dimensional problem.

Illustration of the Resampling Process

We first show the results of resampling KGDP-$f$ in two-dimensional cases, where we can plot the $\theta$ space.

Figure 3.3 shows the evolution of the small pool and the $L$ candidates in $\theta$ space in the first several iterations of a realization of the problem. In these images, the horizontal and vertical axes represent $\theta_1$ and $\theta_2$, respectively, with colors indicating the values of MSE. The red pentagon shows where the truth $\theta^*$ is located, while the white star indicates $\hat{\theta}^n$, namely the $\theta$ with the smallest MSE in the large pool. The region circled by the red line is the range of the small pool. The green squares indicate the locations of the candidates, and the sizes are proportional to their probabilities. Note that the images in the second row have been
scaled to show more detail. We can see that the small pool shrinks quickly towards around $\theta^*$, and that $\hat{\theta}^n$ together with the candidates converges to $\theta^*$ within only a few iterations.

Figure 3.3: Illustration of resampling in $\theta$ space (Iterations 1, 2, 3, 5, 7, 9) (5,7,9 are zoomed in).

Comparison of Different Policies to Choose $x^n$

We use the opportunity cost ($OC$) to evaluate the performance of various policies from the alternative perspective. $OC$ is defined as the difference between the true function values at our estimated optimal $x$ and the true optimal $x$, i.e.,

$$OC(n) = \max_{x \in \mathcal{X}} f(x; \theta^*) - f(\operatorname{argmax}_{x \in \mathcal{X}} \bar{f}^n(x); \theta^*).$$

We normalize $OC$ by the range of the function and denote this ratio as $OC\%$:

$$OC\%(n) = \frac{\max_{x \in \mathcal{X}} f(x; \theta^*) - f(\operatorname{argmax}_{x \in \mathcal{X}} \bar{f}^n(x); \theta^*)}{\max_{x \in \mathcal{X}} f(x; \theta^*) - \min_{x \in \mathcal{X}} f(x; \theta^*)}.$$
We define the noise level as the ratio of the noise standard deviation to the range of the true function, i.e., \( \frac{\sigma}{\max_x f(x; \theta^*) - \min_x f(x; \theta^*)} \). When we say the noise is 20%, we mean this ratio is 20%.

Figure 3.4 shows the comparison of the five policies in three dimensional settings. The subimages correspond to 5%, 20% and 50% of noise from left to right. We can see that KGDP-f and KGDP-H are among the best under different noise levels. We have similar results for two- and five-dimensional cases, too.

Figure 3.4: OC% of different policies for three-dimensional \( \theta \) (5%, 20% and 50% noise from left to right).

### 3.6.2 Application in W/O/W Nanoemulsion Stability Study

We then study a materials science problem known as the water-oil-water (W/O/W) nanoemulsion stability, which is also studied in Chen et al. (2014), and repeat its experiments using the original non-resampling KGDP and our resampling KGDP methods as a comparison.

The problem studies controlled payload delivery using water-oil-water (W/O/W) double nanoemulsions, which depicts the delivery of payload molecules from the internal water droplet, surrounded by a larger droplet of oil, to the external water. This process is performed via laser excitation and controlled by several experimental variables. We can model the stability of this process as a function \( f(x; \theta) \) that is nonlinear in \( \theta \) (as well as \( x \)). \( x \) is a five-dimensional control variable and includes variables such as the initial volume of the
external water, the initial volume fraction of the oil droplets, and the diameter of the oil droplets. \( \theta \) includes seven dimensions, representing the unknown parameters that appear in the formulation, such as the energy barrier for flocculation, the rate prefactor for coalescence, and droplet adsorption/desorption energy barrier difference. Our goal is to conduct a series of experiments to learn the optimal setting \((x)\) that achieves the best stability, and also to learn the correct unknown parameters \(\theta\). A more detailed introduction and formulation of this problem can be found in Chen et al. (2014).

In order to be consistent with Chen et al. (2014), we fix three dimensions of \(x\) and create a finite set of alternatives using the other two dimensions. For \(\theta\), we study 1) a three dimensional case and 2) a seven dimensional one in our experiments. Remember that the dimensionality of \(\theta\) is more important than the dimensionality of \(x\), since we always have finite samples of \(x\).

**Three Dimensional \(\theta\)**

Figure 3.5 shows the comparison of various policies in terms of \(OC\)% under 50% of noise. The left image demonstrates the comparison across the five policies in the resampling framework, while the right one compares the resampling method with non-resampling KGDP in Chen et al. (2014).

(a) Comparison of five resampling policies

(b) Resampling vs non-resampling KGDP

Figure 3.5: \(OC\)% of different policies in three-dimensional nanoemulsion (50% noise).
The left image indicates that Max-Var, KGDP-$f$ and KGDP-$H$ are among the best, showing very close performance. The right image compares: (1) the orange curve: non-resampling KGDP with truth from prior (i.e., one $\theta$ in the candidate set being exactly $\theta^*$), which is the ideal case of Chen et al. (2014); (2) the pink curve: non-resampling KGDP with truth not from prior, which is what usually happens in reality using Chen et al. (2014); (3) the black curve: resampling KGDP-$f$; and (4) the green curve: resampling KGDP-$H$. We can see that (2) gets stuck after a few iterations, while our resampling methods using either KGDP-$f$ or KGDP-$H$ are both able to find the more probable parameters and approach the ideal case gradually.

We have shown the performance of our resampling algorithm in finding the best alternative in terms of OC%. To show its performance in learning the parameters, we denote our best estimate of $\theta^*$ at time $n$ as $\hat{\theta}^n$, and calculate:

(1) the mean square error of $f(x; \theta^*)$ and $f(x; \hat{\theta}^n)$, defined as

$$f_{MSE} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \left| f(x; \theta^*) - f(x; \hat{\theta}^n) \right|^2,$$

(2) or the error of our estimator, i.e., $||\hat{\theta}^n - \theta^*||_2$, as well as the error of a particular dimension of $\theta$.

Note that in method (2), when calculating $||\hat{\theta}^n - \theta^*||_2$, we should normalize each dimension to the same scale. Moreover, in a higher dimensional problem, since the function value $f$ may have different sensitivities in different dimensions of $\theta$, while the entire vector converges in aggregate, some individual dimensions of $\theta$, in which $f$ is less sensitive, may converge more slowly than others. Hence, we also calculate the error of each dimension of $\theta$ in some problems.

We show the MSE and the error of $\theta$ in Figure 3.6, comparing the four cases in Figure 3.5 (b). In the first image, we calculate $f_{MSE}$, and plot $\frac{\sqrt{f_{MSE}}}{\max_{x} f(x; \theta^*) - \min_{x} f(x; \theta^*)}$, and while other three show $|\theta_1^* - \hat{\theta}_1^n|$, $|\theta_2^* - \hat{\theta}_2^n|$ and $|\theta_3^* - \hat{\theta}_3^n|$, respectively (here the subscript $i \in \{1, 2, 3\}$.
indexes the $i$-th dimension of the vector $\theta$). Note that in this example, the second dimension ($\theta_2$) seems less important to $f$ than $\theta_1$ and $\theta_3$, and thus converges much slower. We also have similar results under 20% noise.

Figure 3.6: Performance of resampling KGDP in finding $\theta$ in three-dimensional nanoemulsion under 50% noise (note that $\theta_i$ is the $i$-th dimension of the vector $\theta$).

**Seven Dimensional $\theta$**

In a seven-dimensional case, Figure 3.7 shows the process of $\bar{f}^n(x)$ approaching the true function, using resampling KGDP-$f$ under 20% noise. The four images in the second row are our approximations at $n = 1, 10, 20, 45$, respectively.

Figure 3.7: Evolution of $\bar{f}^n(x)$ (seven-dimensional $\theta$, 20% noise).

The result of OC% under 50% noise are shown in Figure 3.8. We have similar results regarding learning the parameters as in three-dimensional cases.
3.6.3 Standard Test Functions

Next, we show the performance of our algorithm in comparison with different policies on several standard test functions. We use various test functions in Jamil & Yang (2013) (regarding some numbers as the unknown parameters $\theta$). Table 3.1 shows the results after 20 measurements under 20% noise, comparing the quality of the estimated $x$ ($OC\%$) and the estimated $\theta$ ($f_{MSE}$ and $||\hat{\theta}^n - \theta^*||_2$). In the table, KGDP (No Resampling) uses the algorithm in Chen et al. (2014) without truth from prior, while all the other policies use our resampling method. As we can see, our resampling method coupled with the KGDP-$f$ or the KGDP-$H$ policy shows strong performance in finding the optimal $x$ and the true $\theta$.

3.7 Conclusion

Optimal learning for nonlinear parametric belief models has been an interesting but difficult problem. With a belief model nonlinear in the unknown parameters, the knowledge gradient becomes computationally intractable for a continuous parameter space. In Chen et al. (2014), the Knowledge Gradient with Discrete Priors (KGDP) model is proposed, which uses a sampled representation of the parameter space. This method works well when one of the sampled candidates is exactly the truth, but this is unlikely to happen in practice.
Table 3.1: Comparison of different policies at $n = 20$ (with 20% noise)

<table>
<thead>
<tr>
<th>Beale</th>
<th>KGDP (No Resampling)</th>
<th>KGDP-f</th>
<th>KGDP-H</th>
<th>max-Var</th>
<th>Pure Exploration</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(x; \theta) = (\theta_1 - x_1 + x_1 x_2)^2 + (\theta_2 - x_1 + x_1 x_2^3)^2 + (\theta_3 - x_1 + x_1 x_2^3)^2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OC$(%)$</td>
<td>0.2175</td>
<td>0.1079</td>
<td>0.1382</td>
<td>0.1384</td>
<td>0.4243</td>
</tr>
<tr>
<td>$\sqrt{\text{MSE}}(%)$</td>
<td>2.5455</td>
<td>1.6552</td>
<td>1.5146</td>
<td>1.5370</td>
<td>4.1623</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\hat{\theta} - \theta^*</td>
<td></td>
<td>_2$</td>
<td>0.4794</td>
</tr>
</tbody>
</table>

| Booth       | $f(x; \theta) = (x_1 + \theta_1 x_2 - \theta_2)^2 + (\theta_1 x_1 + x_2 - \theta_3)^2$ |
| OC$(\%)$   | 2.0550               | 0.6355 | 0.7472 | 0.8225  | 1.4101          |
| $\sqrt{\text{MSE}}(\%)$ | 6.8320               | 4.4899 | 4.1379 | 4.2250  | 5.9833          |
| $||\hat{\theta} - \theta^*||_2$ | 0.3355               | 0.2829 | 0.2703 | 0.2913  | 0.3684          |

| Rosenbrock  | $f(x; \theta) = (x_1 - \theta_1)^2 + \theta_2(x_2 - \theta_3 x_1)^2$ |
| OC$(\%)$   | 3.1154               | 0.1154 | 0.1296 | 0.1241  | 0.9624          |
| $\sqrt{\text{MSE}}(\%)$ | 8.0019               | 2.6978 | 2.9016 | 2.8983  | 5.7225          |
| $||\hat{\theta} - \theta^*||_2$ | 0.3895               | 0.3546 | 0.3507 | 0.3474  | 0.4024          |

| Sphere      | $f(x; \theta) = \sum_{i=1}^5 (x_i - \theta_i)^2$ |
| OC$(\%)$   | 10.7258              | 2.1549 | 2.6405 | 2.7943  | 5.1539          |
| $\sqrt{\text{MSE}}(\%)$ | 14.1262              | 7.5191 | 7.3693 | 7.5489  | 9.9403          |
| $||\hat{\theta} - \theta^*||_2$ | 0.5346               | 0.2710 | 0.2702 | 0.2713  | 0.3625          |

| Trid        | $f(x; \theta) = (x_1 - \theta_1)^2 + (x_2 - \theta_2)^2 - \theta_3 x_1 x_2$ |
| OC$(\%)$   | 4.1506               | 0.4628 | 0.5565 | 0.7863  | 1.8276          |
| $\sqrt{\text{MSE}}(\%)$ | 10.3437              | 5.2282 | 3.9352 | 4.1588  | 7.2341          |
| $||\hat{\theta} - \theta^*||_2$ | 0.3218               | 0.1466 | 0.1150 | 0.1378  | 0.2982          |

* $\sqrt{\text{MSE}}(\%) = \sqrt{\text{max}_{f(x; \theta^*)} - \text{min}_{f(x; \theta^*)}} \times 100\%$

In this chapter, we propose a resampling algorithm that cooperates with KGDP to solve this problem. We no longer require that one of the candidates is correct, but use the measurement history to guide resampling and discover more probable parameters. Within a small budget of noisy and expensive experiments, our algorithm is capable of both finding the optimal alternative that maximizes the value function, and narrow down the location of the unknown parameters. Experiments on a synthetic benchmark function, a real world materials science problem, and some standard test functions have shown strong performances of our algorithm.

We recognize the duality of this optimization problem and manipulate in both the alternative and the parameter spaces. Besides the idea of maximizing the value function (denoted as KGDP-f) as used in previous optimal learning literature, we also put forward an entropy-
oriented metric that works under the knowledge gradient framework, which aims directly at
lowering the uncertainty of the parameters (in terms of entropy). Experiments show strong
performances of both metrics.

We also prove the asymptotic optimality of KGDP-$f$ and KGDP-$H$ in both the non-
resampling with truth from prior case, and the resampling framework. That means, given
an infinite number of measurements, we are able to find both the optimal alternative and the
correct parameter. Hence, KGDP-$f$ and KGDP-$H$ are both myopically and asymptotically
optimal in learning the alternative and reducing parameter uncertainty, respectively.

## 3.8 Appendix

### 3.8.1 Proof of the Two-curve Model

**Proposition 3.3.1.** In the two-curve case,

1. The KGDP-$f$ score is an increasing function of the distance between the two
functions. Hence, it always measures $x^n \in \arg\max_{x \in X} |f_1(x) - f_2(x)|$. The only
exception is when $f_1$ and $f_2$ achieves their maximum values at the same $x$, i.e,
$\arg\max_x f_1(x) \cap \arg\max_x f_2(x) \neq \emptyset$, in which case the KGDP-$f$ score at any $x$ is 0.

2. The KGDP-$H$ score is an increasing function of the distance between the two functions.
Hence, it always measures $x^n = \arg\max_{x \in X} |f_1(x) - f_2(x)|$.

**Proof of Proposition 3.3.1 (1)**

In the KGDP-$f$ model, we have the following formula given by Equation (3.7):

$$
\nu_{KGDP-f,n}^{f,n}(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} \max_{x'} \left( \sum_{i=1}^{L} f_i(x') p_i^n \exp \left[ - \frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] \right) d\hat{y} - \max_{x'} \sum_{i=1}^{L} f_i(x') p_i^n.
$$

(3.10)
Let $h(x)$ denote the first part:

$$h(x) = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{+\infty} \max_{x'} \left( \sum_{i=1}^{L} f_i(x') p_i^n \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] \right) \, d\hat{y}.$$ 

In the two-curve model, $L = 2$. Let $f_2(x) - f_1(x) = d$ and $\hat{y} - f_1(x) = \omega$. Then

$$h(x) = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{+\infty} \max_{x'} \left( f_1(x') p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right] + f_2(x') p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right] \right) \, d\omega. \quad (3.11)$$

Hence, given $f_1$ and $f_2$, we can see that $h(x)$ only depends on $d = f_1(x) - f_2(x)$. Since $h(x)$ can be viewed as a function of $d$ (in which $x$ is hidden), we denote it as $h(d)$ instead from now on.

From Equation (3.11), $h(d) = h(-d)$. Therefore, to prove our theorem, we need to show that when $d \geq 0$, $h(d)$ is a nondecreasing function of $d$. We show this by taking the derivative of $h(d)$ with respect to $d$.

Let $g(x, \omega) = f_1(x) p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right] + f_2(x) p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right]$. Suppose $\mathcal{X}$ contains $M$ alternatives and $\mathcal{X} = \{x_1, ..., x_M\}$. Then

$$h(d) = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{+\infty} \max_{x \in \mathcal{X}} g(x, \omega) \, dx = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{+\infty} \max \left\{ \begin{array}{c} g(x_1, \omega) \\ g(x_2, \omega) \\ \vdots \\ g(x_M, \omega) \end{array} \right\} \, d\omega.$$ 

For different $\omega$'s, the $\max$ function chooses different $x$'s. In order to take the derivative, we divide the real axis of $\omega$ into several intervals such that on each interval, the same $x$ is picked for the $\max$ function.

Note that the number of such intervals is finite. This is due to the fact that for each fixed $x$, $g(x, \omega)$ is either unimodal or bimodal, as can be seen from its definition. Therefore, a finite number of $x$'s would result in a finite number of such intervals.
Suppose we have \( K \) such intervals.

If \( K = 1 \), then the same \( x \) is chosen for all \( \omega \)'s (which further implies \( \text{argmax}_x f_1(x) = \text{argmax}_x f_2(x) \)). Without loss of generality, assume it is \( x_1 \). Hence,

\[
h(d) = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{+\infty} \left( f_1(x_1)p_1^n \exp\left[-\frac{\omega^2}{2\sigma^2}\right] + f_2(x_1)p_2^n \exp\left[-\frac{(\omega-d)^2}{2\sigma^2}\right]\right) d\omega
\]

\[
= f_1(x_1)p_1^n + f_2(x_1)p_2^n.
\]

Therefore, \( h(d) = \max_x [f_1(x)p_1^n + f_2(x)p_2^n] \), which is a constant regardless of \( d \). Hence, by Equation (3.10), \( \nu^{KGDP,n}(x) = h(d) - \max_x [f_1(x)p_1^n + f_2(x)p_2^n] = 0, \forall x \).

If \( K > 1 \), let \( S_1, \ldots, S_{K-1} \) be the separating points on the \( \omega \) axis, and let \( S_0 = -\infty \) and \( S_K = +\infty \). Thus, \( I_1 = (-\infty, S_1), I_2 = [S_1, S_2), \ldots, I_K = [S_{K-1}, S_K) \) are the \( K \) intervals. On each interval \( I_k \) (\( 1 \leq k \leq K \)), denote \( x^{(k)} \) as the alternative for \( g(x, \omega) \) to achieve its maximum, i.e,

\[
x^{(k)} = \text{argmax}_x g(x, \omega), \forall \omega \in I_k.
\]

Two examples of the set of \( g(x, \omega) \)'s as well as the separating points are shown in Figure 3.9. Take Figure 3.9(a) as an example, on \( I_1 = (-\infty, S_1), x^{(1)} = x_1 \); on \( I_2 = (S_1, S_2), x^{(2)} = x_3 \); on \( I_3 = (S_2, +\infty), x^{(3)} = x_2 \).

![Figure 3.9](image-url)

(a) \( f_1(x) = (8, 1, 6), f_2(x) = (1, 8, 6), d = 1, p_1 = p_2 = 1/2. g(x_1, \omega), g(x_2, \omega) \) and \( g(x_3, \omega) \), all unimodal, correspond to the red, blue and green curves, respectively.

(b) \( f_1(x) = (10, 6, 2), f_2(x) = (2, 4, 8), d = 5, p_1 = p_2 = 1/2. g(x_1, \omega), g(x_2, \omega) \) and \( g(x_3, \omega) \), all bimodal, correspond to the blue, orange and green curves, respectively.

Figure 3.9: Two examples of the set of \( g(x, \omega) \) and the separating points \( \{S_k\} \).
At any separating point $S_k \in \{S_1, S_2, ..., S_{K-1}\}$, we have

$$g(x^{(k)}, S_k) = g(x^{(k+1)}, S_k), \forall 1 \leq k \leq K - 1. \quad (3.12)$$

We then rewrite $h(d)$ as the sum of $K$ integrals on different intervals, and take the derivative with respect to $d$. Our goal is to show that $\frac{\partial h(d)}{\partial d} \geq 0$, when $d \geq 0$.

$$\frac{\partial h(d)}{\partial d} = \frac{\partial}{\partial d} \left( \frac{1}{\sqrt{2\pi\sigma}} \sum_{k=1}^{K} \int_{I_k} g(x^{(k)}, \omega) d\omega \right)$$

$$= \frac{1}{\sqrt{2\pi\sigma}} \sum_{k=1}^{K} \frac{\partial}{\partial d} \int_{I_k} g(x^{(k)}, \omega) d\omega$$

$$= \frac{1}{\sqrt{2\pi\sigma}} \sum_{k=1}^{K} \left( -\frac{\partial S_{k-1}}{\partial d} g(x^{(k)}, S_{k-1}) + \frac{\partial S_k}{\partial d} g(x^{(k)}, S_k) + \int_{I_k} \frac{\partial g(x^{(k)}, \omega)}{\partial d} d\omega \right). \quad (3.14)$$

Because of Equation (3.12), the sum of the first two items in Equation (3.14) can be written as a telescoping sum, and the remaining items are $-\frac{\partial S_0}{\partial d} g(x^{(0)}, S_0) + \frac{\partial S_K}{\partial d} g(x^{(K)}, S_K) = 0$. That is:

$$\sum_{k=1}^{K} \left( -\frac{\partial S_{k-1}}{\partial d} g(x^{(k)}, S_{k-1}) + \frac{\partial S_k}{\partial d} g(x^{(k)}, S_k) \right)$$

$$= \sum_{k=1}^{K} \left( -\frac{\partial S_{k-1}}{\partial d} g(x^{(k-1)}, S_{k-1}) + \frac{\partial S_k}{\partial d} g(x^{(k)}, S_k) \right)$$

$$= -\frac{\partial S_0}{\partial d} g(x^{(0)}, S_0) + \frac{\partial S_1}{\partial d} g(x^{(1)}, S_1) - \frac{\partial S_1}{\partial d} g(x^{(1)}, S_1) + ... + \frac{\partial S_K}{\partial d} g(x^{(K)}, S_K)$$

$$= -\frac{\partial S_0}{\partial d} g(x^{(0)}, S_0) + \frac{\partial S_K}{\partial d} g(x^{(K)}, S_K)$$

$$= 0.$$
Therefore, we have

\[
\frac{\partial h(d)}{\partial d} = \frac{1}{\sqrt{2\pi}\sigma} \sum_{k=1}^{K} \int_{I_k} \frac{\partial g(x^{(k)}), \omega}{\partial d} d\omega
\]

\[
= \frac{1}{\sqrt{2\pi}\sigma} \sum_{k=1}^{K} \int_{S_{k-1}}^{S_k} f_2(x^{(k)}) p_2^n \exp \left[-\frac{(\omega - d)^2}{2\sigma^2}\right] \frac{1}{2\sigma^2} (\omega - d) d\omega
\]

\[
= \frac{1}{\sqrt{2\pi}\sigma} \frac{1}{4\sigma^2} \sum_{k=1}^{K} \left( f_2(x^{(k)}) p_2^n \exp \left[-\frac{(S_{k-1} - d)^2}{2\sigma^2}\right] - f_2(x^{(k)}) p_2^n \exp \left[-\frac{(S_k - d)^2}{2\sigma^2}\right] \right)
\]

\[
= \frac{1}{\sqrt{2\pi}\sigma} \frac{1}{2\sigma^2} \sum_{k=1}^{K-1} \left( -f_2(x^{(k)}) p_2^n \exp \left[-\frac{(S_k - d)^2}{2\sigma^2}\right] + f_2(x^{(k+1)}) p_2^n \exp \left[-\frac{(S_k - d)^2}{2\sigma^2}\right] \right)
\]

\[
= \frac{1}{\sqrt{2\pi}\sigma} \frac{1}{2\sigma^2} \sum_{k=1}^{K-1} \left( [f_2(x^{(k+1)}) - f_2(x^{(k)})] p_2^n \exp \left[-\frac{(S_k - d)^2}{2\sigma^2}\right] \right).
\]

(3.15)

We then show \( f_2(x^{(k+1)}) > f_2(x^{(k)}), \forall 1 \leq k \leq K - 1 \), which indicates \( \frac{\partial h(d)}{\partial d} > 0 \) and will complete the proof.

Note that solving \( g(x^{(k)}, \omega) = g(x^{(k+1)}, \omega) \) (with \( \omega \) as the unknown variable), the only solution is

\[
S_k = \frac{1}{d} \ln \frac{f_1(x^{(k)}) p_1^n - f_1(x^{(k+1)}) p_1^n}{f_2(x^{(k+1)}) p_2^n - f_2(x^{(k)}) p_2^n} + \frac{d}{2}.
\]

Hence, for \( \forall S_k \leq \omega \leq \infty \), \( g(x^{(k+1)}) > g(x^{(k)}) \), i.e,

\[
\frac{f_1(x^{(k+1)}) p_1^n \exp \left[-\frac{\omega^2}{2\sigma^2}\right]}{f_2(x^{(k+1)}) p_2^n} > \frac{f_1(x^{(k)}) p_1^n \exp \left[-\frac{\omega^2}{2\sigma^2}\right]}{f_2(x^{(k)}) p_2^n} + \frac{\omega - d}{2\sigma^2}.
\]

This is equivalent to:

\[
[f_1(x^{(k+1)}) - f_1(x^{(k)})] p_1^n \exp \left[-\frac{\omega^2}{2\sigma^2}\right] + [f_2(x^{(k+1)}) - f_2(x^{(k)})] p_2^n \exp \left[-\frac{\omega^2}{2\sigma^2}\right] > 0.
\]
This is further equivalent to:

$$\left[f_1(x^{(k+1)}) - f_1(x^{(k)})\right] p_1^n \exp\left(\frac{d^2}{2\sigma^2}\right) \exp\left(-\frac{\omega d}{\sigma^2}\right) + \left[f_2(x^{(k+1)}) - f_2(x^{(k)})\right] p_2^n > 0.$$

This is true for any $\omega \in (S_k, \infty)$. Specially, let $\omega \to \infty$, and we must have $f_2(x^{(k+1)}) > f_2(x^{(k)})$. Plug this result into Equation (3.15), and we can see $\frac{\partial h(d)}{\partial d} > 0$. This completes the proof.

A further result is that $x^{(K)} = \arg\max_x f_2(x)$. Similarly, $x^{(1)} = \arg\max_x f_1(x)$. Moreover, if $d = 0$, then $h(d) = \max_x [f_1(x)p_1^n + f_2(x)p_2^n]$; if $d \to \infty$, then $h(d) \to \max_x f_1(x)p_1^n + \max_x f_2(x)p_2^n$.

Regarding the case when the two functions are aligned, refer to the proof of Lemma 3.4.3.

□

Proof of Proposition 3.3.1 (2)

In the KGDP-H model, we have:

$$\nu^{KGDP-H, n} (x) = \mathbb{E} \left[p_1^{n+1} \log p_1^{n+1} + p_2^{n+1} \log p_2^{n+1} | S^n = s, x^n = x\right]$$
$$- (p_1^n \log p_1^n + p_2^n \log p_2^n),$$

where

$$p_1^{n+1} = \frac{p_1^n \exp\left[-\frac{(\hat{y} - f_1(x))^2}{2\sigma^2}\right]}{p_1^n \exp\left[-\frac{(\hat{y} - f_1(x))^2}{2\sigma^2}\right] + p_2^n \exp\left[-\frac{(\hat{y} - f_2(x))^2}{2\sigma^2}\right]},$$

$$p_2^{n+1} = \frac{p_2^n \exp\left[-\frac{(\hat{y} - f_2(x))^2}{2\sigma^2}\right]}{p_1^n \exp\left[-\frac{(\hat{y} - f_1(x))^2}{2\sigma^2}\right] + p_2^n \exp\left[-\frac{(\hat{y} - f_2(x))^2}{2\sigma^2}\right]},$$

with $\hat{y}$ as the measurement result, on which the expectation $\mathbb{E}$ is taken.
Let $d = f_2(x) - f_1(x)$, and $h(d) = \mathbb{E}^n \left[ p_1^{n+1} \log p_1^{n+1} + p_2^{n+1} \log p_2^{n+1} | S^n = s, x^n = x \right]$. We want to show that $h(d)$ is an increasing function of $|d|$.

$h(d)$

\[
= \mathbb{E}^n \left[ p_1^{n+1} \log p_1^{n+1} + p_2^{n+1} \log p_2^{n+1} | S^n = s, x^n = x \right]
\]

\[
= \int_{-\infty}^{\infty} \left[ p_1^{n+1} \log p_1^{n+1} + p_2^{n+1} \log p_2^{n+1} \right] \cdot \frac{1}{\sqrt{2\pi} \sigma} \left( p_1^n \exp \left[ -\frac{(\hat{y} - f_1(x))^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{(\hat{y} - f_2(x))^2}{2\sigma^2} \right] \right) d\hat{y}
\]

\[
= \frac{1}{\sqrt{2\pi} \sigma} \int_{-\infty}^{\infty} \left( p_1^n \exp \left[ -\frac{(\hat{y} - f_1(x))^2}{2\sigma^2} \right] \right) \log p_1^{n+1} + p_2^n \exp \left[ -\frac{(\hat{y} - f_2(x))^2}{2\sigma^2} \right] \log p_2^{n+1} d\hat{y}.
\]

Let $\omega = \hat{y} - f_1(x)$, hence $\hat{y} - f_2(x) = \omega - d$, and

\[
p_1^{n+1} = \frac{p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right]}{p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right]}, \quad p_2^{n+1} = \frac{p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right]}{p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right]}.
\]

Therefore,

\[
h(d) = \frac{1}{\sqrt{2\pi} \sigma} \int_{-\infty}^{\infty} \left( p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right] \right) \log p_1^{n+1} + p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right] \log p_2^{n+1} d\omega
\]

\[
= \frac{1}{\sqrt{2\pi} \sigma} \int_{-\infty}^{\infty} p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right]\left( \log p_1^n - \frac{\omega^2}{2\sigma^2} \right) d\omega
\]

\[
+ \frac{1}{\sqrt{2\pi} \sigma} \int_{-\infty}^{\infty} p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right] \log p_2^n - \frac{(\omega - d)^2}{2\sigma^2} d\omega
\]

\[
- \frac{1}{\sqrt{2\pi} \sigma} \int_{-\infty}^{\infty} \left( p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right] \right) \log \left( p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right] \right) d\omega
\]

\[
= p_1^n \log p_1^n + p_2^n \log p_2^n - 1
\]

\[
- \frac{1}{\sqrt{2\pi} \sigma} \int_{-\infty}^{\infty} \left( p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right] \right) \log \left( p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right] \right) d\omega. \quad (3.16)
\]

Let $\omega_1 = -\omega$, and we can see that $h(d) = h(-d)$. Hence, it suffices to show that when $d > 0$, $h(d)$ is an increasing function of $d$, namely $\frac{\partial h(d)}{\partial d} > 0$ when $d > 0$. 

55
In Equation (3.16), we take the derivative of \( h(d) \) with respect to \( d \),

\[
\sqrt{2\pi\sigma} \cdot \frac{\partial h(d)}{\partial d} = -\int_{-\infty}^{\infty} \frac{\partial}{\partial d} \left\{ \left( p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right] \right) \cdot \log \left( p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right] \right) \right\} d\omega
\]

\[
= -\int_{-\infty}^{\infty} p_2^n \exp \left[ \frac{(\omega - d)^2}{2\sigma^2} \right] \left( \frac{\omega - d}{\sigma^2} \right) \left\{ 1 + \log \left( p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right] \right) \right\} d\omega
\]

\[
= -\int_{-\infty}^{\infty} p_2^n \exp \left[ \frac{(\omega - d)^2}{2\sigma^2} \right] \left( \frac{\omega - d}{\sigma^2} \right) \log \left( p_1^n \exp \left[ -\frac{\omega^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{(\omega - d)^2}{2\sigma^2} \right] \right) d\omega
\]

\[
= -\int_{-\infty}^{\infty} p_2^n \exp \left[ \frac{t^2}{2\sigma^2} \right] \left( \frac{t}{\sigma^2} \right) \log \left( p_1^n \exp \left[ \frac{(t + d)^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{t^2}{2\sigma^2} \right] \right) dt
\]

\[
= -\int_{-\infty}^{\infty} p_2^n \exp \left[ \frac{t^2}{2\sigma^2} \right] \left( \frac{t}{\sigma^2} \right) \log \left( p_1^n \exp \left[ \frac{(t + d)^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{t^2}{2\sigma^2} \right] \right) dt
\]

\[
+ \int_{0}^{\infty} p_2^n \exp \left[ \frac{t^2}{2\sigma^2} \right] \left( \frac{t}{\sigma^2} \right) \log \left( p_1^n \exp \left[ \frac{(t - d)^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{t^2}{2\sigma^2} \right] \right) dt
\]

Since \( d > 0 \), for \( \forall t > 0 \), we have \((t - d)^2 < (t + d)^2\). This is equivalent to

\[
\exp \left[ -\frac{(t - d)^2}{2\sigma^2} \right] > \exp \left[ -\frac{(t + d)^2}{2\sigma^2} \right].
\]

Hence,

\[
\log \frac{p_1^n \exp \left[ \frac{(t - d)^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{t^2}{2\sigma^2} \right]}{p_1^n \exp \left[ \frac{(t + d)^2}{2\sigma^2} \right] + p_2^n \exp \left[ -\frac{t^2}{2\sigma^2} \right]} > 0.
\]

Therefore, we have \( \frac{\partial h(d)}{\partial d} > 0 \), when \( d > 0 \).

(Moreover, when \( d = 0 \), \( h(d) = p_1^n \log p_1^n + p_2^n \log p_2^n \), thus \( \nu^{KGDP-H,n}(x) = 0 \); when \( d \to \infty \), \( \nu^{KGDP-H,n}(x) = -p_1^n \log p_1^n - p_2^n \log p_2^n \), which is the entropy at time \( n \).) 

\[\square\]
3.8.2 Proof of Convergence of KGDP without Resampling

Lemma 3.4.1. Let \( N^n(x) \) be the number of measurements taken on \( x \) when the total number of measurements is \( n \). If \( N^n(x) \to \infty \) as \( n \to \infty \), then for \( \forall l \in \{1, 2, ..., L\} \) such that \( f(x; \theta_l) \neq f(x; \theta^*) \), \( p^n_l \to 0 \) almost surely, i.e, \( \mathbb{P}(\lim_{n \to \infty} p^n_l = 0) = 1 \).

Proof of Lemma 3.4.1

We prove the following claim first.

Claim 1: \( \forall m \in \{1, ..., M\} \), let \( p^n_1, p^n_2, ..., p^n_L \) be the probabilities of the \( L \) \( \theta \)'s after \( n \) measurements at \( x_m \) (i.e, we take \( n \) measurements only at \( x \) and nowhere else). For \( \forall l \in \{1, 2, ..., L\} \) such that \( f(x_m; \theta_l) \neq f(x_m; \theta^*) \), \( \mathbb{P}(\lim_{n \to \infty} p^n_l = 0) = 1 \).

Proof of Claim 1:

If we take \( n \) measurements at \( x_m \), suppose the results are \( \{\hat{y}^1, ..., \hat{y}^n\} \). Let \( f_l = f(x_m; \theta_l) \).

Then \( p_1, ..., p_L \) are given by:

\[
p^n_l = \frac{\prod_{j=1}^{n} \exp\left[\frac{-(\hat{y}^j-f_l)^2}{2\sigma^2}\right]}{\sum_{i=1}^{L} \prod_{j=1}^{n} \exp\left[\frac{-(\hat{y}^j-f_i)^2}{2\sigma^2}\right]}.
\]

Let \( f^* = f(x_m; \theta^*) \). Then \( \hat{y}^j \sim \mathcal{N}(f^*, \sigma^2) \). Let \( Z^j = \frac{\hat{y}^j - f^*}{\sigma} \), then \( \hat{y}^j \sim f^* + \sigma Z^j \), \( Z^j \sim \mathcal{N}(0, 1) \).

For any \( l \) such that \( f_l \neq f^* \),

\[
p^n_l \leq \frac{\prod_{j=1}^{n} \exp\left[\frac{-(\hat{y}^j-f_l)^2}{2\sigma^2}\right]}{\prod_{j=1}^{n} \exp\left[\frac{-(\hat{y}^j-f^*)^2}{2\sigma^2}\right]} = \exp\left[-\frac{\sum_{j=1}^{n} (f^* - f_l)^2}{2\sigma^2} + \frac{n(f^* - f_l)^2}{2\sigma^2}\right] = \exp\left[-\frac{\sum_{j=1}^{n} (f^* - f_l + \sigma Z^j)^2}{2\sigma^2} - \frac{(\sigma Z^j)^2}{2}\right]
\]

\[
= \exp\left(-\frac{\sum_{j=1}^{n} [(f^* - f_l + \sigma Z^j)^2 - (\sigma Z^j)^2]}{2\sigma^2}\right)
\]

\[
= \exp\left(-\frac{n(f^* - f_l)^2 + 2\sigma(f^* - f_l) \sum_{j=1}^{n} Z^j}{2\sigma^2}\right).
\]

(3.17)
For any $\epsilon > 0$, \(\exp\left( -\frac{n(f^*-f_i)^2+2\sigma(f^*-f_i)\sum_{j=1}^{n}Z^j}{2\sigma^2} \right) < \epsilon \Rightarrow p^n_i < \epsilon\), and therefore
\[
\lim_{n \to \infty} \exp\left( -\frac{n(f^*-f_i)^2+2\sigma(f^*-f_i)\sum_{j=1}^{n}Z^j}{2\sigma^2} \right) = 0 \subseteq \lim_{n \to \infty} p^n_i = 0.
\]
We show that
\[
\mathbb{P}\left[ \lim_{n \to \infty} \exp\left( -\frac{n(f^*-f_i)^2+2\sigma(f^*-f_i)\sum_{j=1}^{n}Z^j}{2\sigma^2} \right) = 0 \right] = 1,
\]
which implies the almost-sure convergence of $p^n_i$ to 0.

By the strong law of large numbers, $\frac{\sum_{j=1}^{n}Z^j}{n} \xrightarrow{a.s.} 0$. That is, there exists $\Omega_{x,l} \in \Omega$, such that for $\forall \omega \in \Omega_{x,l}$, $\lim_{n \to \infty} \frac{\sum_{j=1}^{n}Z^j(\omega)}{n} = 0$, and $\mathbb{P}(\Omega_{x,l}) = 1$. Our goal is to show that for $\forall \omega \in \Omega_{x,l}$,
\[
\lim_{n \to \infty} \exp\left( -\frac{n(f^*-f_i)^2+2\sigma(f^*-f_i)\sum_{j=1}^{n}Z^j(\omega)}{2\sigma^2} \right) = 0.
\]

Without loss of generality, we assume $f^* - f_i \geq 0$. For $\forall \omega \in \Omega_{x,l}$, there exists $N_1 \in \mathbb{N}$ such that for $\forall n > N_1$, $\left| \frac{\sum_{j=1}^{n}Z^j(\omega)}{n} \right| < \frac{f^* - f_i}{4\sigma}$.

Therefore, for any $n > N_1$, $n \left[ (f^* - f_i)^2 + 2\sigma(f^* - f_i)\sum_{j=1}^{n}Z^j(\omega) \right] > n \cdot \frac{(f^* - f_i)^2}{2}$. Hence, for $\forall n > N_1$,
\[
\exp\left( -\frac{n(f^*-f_i)^2+2\sigma(f^*-f_i)\sum_{j=1}^{n}Z^j(\omega)}{2\sigma^2} \right) < \exp\left( -\frac{n(f^*-f_i)^2}{4\sigma^2} \right). \tag{3.18}
\]

For $\forall \epsilon > 0$, there exists $N_2$ such that for $\forall n > N_2$, $\exp\left( -\frac{n(f^*-f_i)^2}{4\sigma^2} \right) < \epsilon$.

Take $N = \max(N_1, N_2)$, we know that for $\forall \omega \in \Omega_{x,l}$, $\forall \epsilon > 0$, there exists $N > 0$, such that for $\forall n > N$,
\[
\exp\left( -\frac{n(f^*-f_i)^2+2\sigma(f^*-f_i)\sum_{j=1}^{n}Z^j(\omega)}{2\sigma^2} \right) < \epsilon.
\]

Hence,
\[
\Omega_{x,l} \subseteq \{ \lim_{n \to \infty} \exp\left( -\frac{n(f^*-f_i)^2+2\sigma(f^*-f_i)\sum_{j=1}^{n}Z^j}{2\sigma^2} \right) \} \subseteq \{ \lim_{n \to \infty} p^n_i = 0 \}.
\]

Since $\mathbb{P}(\Omega_{x,l}) = 1$, we have $\mathbb{P}\{\lim_{n \to \infty} p^n_i = 0\} = 1$. \qed
Back to Lemma 3.4.1, define \( N^n(x, j) \) as the time when we take the \( j \)-th measurement on \( x \) \((1 \leq j \leq N^n(x))\). We have

\[
    p^n_t = \frac{\prod_{j=1}^{n} \exp\left[\frac{-(\hat{g}^j - f(x^j; \theta))}{2\sigma^2}\right]}{\sum_{i=1}^{L} \prod_{j=1}^{n} \exp\left[\frac{-(\hat{g}^j - f(x^j; \theta))^2}{2\sigma^2}\right]} \\
= \prod_{x \in \mathcal{X}} \prod_{j=1}^{n} \exp\left[\frac{-(\hat{g}^j - f(x^j; \theta))^2}{2\sigma^2}\right] \\
\overset{\text{def}}{=} \prod_{x \in \mathcal{X}} r(x)
\]

where the first item equals 1 and the third equals a constant number. For any \( \omega \in \bigcap_{x \in \mathcal{X}} \Omega_{x,t} \), the second item has limit as 0 by Claim 1. Since \( \mathbb{P}(\bigcap_{x \in \mathcal{X}} \Omega_{x,t}) = 1 - \mathbb{P}(\bigcup_{x \in \mathcal{X}} \Omega_{x,t}) > 1 - \sum_{x \in \mathcal{X}} \mathbb{P}(\Omega^c_{x,t}) = 1 \), we have \( p^n_t \overset{a.s.}{\longrightarrow} 0 \).

\[ \square \]

**Lemma 3.4.2.** Suppose several candidate functions, without loss of generality, say \( \theta_1, ..., \theta_h \in \mathbb{L} \) intersect at a particular \( x \in \mathcal{X} \), namely \( f(x; \theta_1) = ... = f(x; \theta_h) \). Then calculating \( \nu^{KGDP-f,n}(x) \) using \((f_1, ..., f_n)\) with probabilities \((p^n_1, ..., p^n_n)\) is equivalent to calculating \( \nu^{KGDP-f,n}(x) \) using \((f_r, f_{r+1}, ..., f_n)\) with probabilities \((p^n_r, p^n_{r+1}, ..., p^n_n)\), where \( f_r(x') = \frac{\sum_{i=1}^{h} f_i(x') p^n_i}{\sum_{i=1}^{h} p^n_i} \) for all \( x' \in \mathcal{X} \) and \( p^n_r = \sum_{i=1}^{h} p^n_i \). We call the latter the **reduced model** at \( x \).

**Proof of Lemma 3.4.2**

At time \( n \), we can write \( \bar{f}^n \) as

\[
\bar{f}^n = \sum_{i=1}^{L} p^n_i f_i = \left(\sum_{i=1}^{h} p^n_i\right) \frac{\sum_{i=1}^{h} p^n_i f_i}{\sum_{i=1}^{h} p^n_i} + \sum_{i=h+1}^{L} p^n_i f_i = p^n_r f_r + \sum_{i=h+1}^{L} p^n_i f_i.
\]

59
For any $1 \leq i, j \leq h$, according to Equation (3.1), we know $\frac{p^{n+1}_i}{p^{n}_j} = \frac{p^n_i}{p^n_j}$ since $f(x; \theta_i) = f(x; \theta_j)$. Let $\frac{p^{n+1}_1}{p^n_1} = \frac{p^{n+1}_2}{p^n_2} = \ldots = \frac{p^{n+1}_h}{p^n_h} = \lambda$. Hence, $p^{n+1}_r = \sum_{i=1}^{h} p^{n+1}_i = \lambda \sum_{i=1}^{h} p^n_i = \lambda p^n_r$, thus satisfying the updating equation as in Equation (3.1) if considering $f_r$ as a candidate function. As a result, $\bar{f}^{n+1}$ can be written as:

$$\bar{f}^{n+1} = \sum_{i=1}^{L} p^{n+1}_i f_i = \lambda \sum_{i=1}^{h} p^n_i f_i + \sum_{i=h+1}^{L} p^{n+1}_i f_i$$

$$= \lambda p^n_r f_r + \sum_{i=h+1}^{L} p^{n+1}_i f_i = p^{n+1}_r f_r + \sum_{i=h+1}^{L} p^{n+1}_i f_i.$$ 

Moreover, the distribution of the measurement result if considering the reduced model is the same as considering the original model. Therefore, for this $x$,

$$\nu^{KGDP-f,n}(x) = \mathbb{E}^n \left[ \max_{x'} \sum_{i=1}^{L} f_i(x') p^{n+1}_i(x) \right] - \max_{x'} \sum_{i=1}^{L} f_i(x') p^n_i$$

$$= \mathbb{E}^n \left[ \max_{x'} \left( f_r(x') p^{n+1}_r(x) + \sum_{i=h+1}^{L} f_i(x') p^{n+1}_i(x) \right) \right] - \max_{x'} \left( f_r^n(x') p^n_r + \sum_{i=h+1}^{L} f_i(x') p^n_i \right).$$

This finishes the proof.

Lemma 3.4.3. For $\forall n \geq 0, \forall x \in X$, the KGDP-$f$ score $\nu^{KGDP-f,n}(x) \geq 0$. Equality holds if and only if (1) there exists $x'$ such that $x' \in \arg\max_{x} f(x; \theta_i)$ for all $i$ such that $p^n_i > 0$, or (2) $p^n_i = 0$ if $f(x; \theta_i) \neq f(x; \theta^*)$.

Proof of Lemma 3.4.3

Suppose at time $n$, the probabilities of our prior candidates are $(p^n_1, p^n_2, \ldots, p^n_L)$. Since max is a convex function, by Jensen’s inequality,

$$\nu^{KGDP,n}(x) = \mathbb{E}^n \left[ \max_{x'} \sum_{i=1}^{L} f_i(x') p^{n+1}_i(x) | S^n = s, x^n = x \right] - \max_{x'} \sum_{i=1}^{L} f_i(x') p^n_i$$

$$\geq \max_{x'} \mathbb{E} \left[ \sum_{i=1}^{L} f_i(x') p^{n+1}_i(x) | S^n = s, x^n = x \right] - \max_{x'} \sum_{i=1}^{L} f_i(x') p^n_i.$$
We show that $\forall x, x', E \left[ \sum_{i=1}^{L} f_i(x') p_i^{n+1}(x) \big| S^n = s, x^n = x \right] = \sum_{i=1}^{L} f_i(x') p_i^n$. We know
\[
E \left[ \sum_{i=1}^{L} f_i(x') p_i^{n+1}(x) \big| S^n = s, x^n = x \right] = \sum_{i=1}^{L} f_i(x') E \left[ p_i^{n+1}(x) \big| S^n = s, x^n = x \right].
\]

Suppose the measurement at $x$ is $\hat{y}$, we have
\[
p_i^{n+1}(x) = \frac{p_i^n \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right]}{\sum_{j=1}^{L} p_j^n \exp \left[ -\frac{(\hat{y} - f_j(x))^2}{2\sigma^2} \right]}.
\]

Based on our prior belief, the distribution of $\hat{y}$ is:
\[
\hat{y} \sim \begin{cases} 
\mathcal{N}(f_1(x; \theta_1), \sigma^2), \text{with probability } p_1^n \\
\mathcal{N}(f_2(x; \theta_2), \sigma^2), \text{with probability } p_2^n \\
\vdots \\
\mathcal{N}(f_L(x; \theta_L), \sigma^2), \text{with probability } p_L^n
\end{cases}
\]

Therefore,
\[
E \left[ p_i^{n+1}(x) \right] = E \left( \frac{p_i^n \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right]}{\sum_{j=1}^{L} p_j^n \exp \left[ -\frac{(\hat{y} - f_j(x))^2}{2\sigma^2} \right]} \right)
\]
\[
= \int_{-\infty}^{\infty} \sum_{k=1}^{L} p_i^n \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] \frac{1}{\sqrt{2\pi}} \exp \left[ -\frac{(\hat{y} - f_k(x))^2}{2\sigma^2} \right] d\hat{y}
\]
\[
= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] \frac{\sum_{k=1}^{L} p_k^n \exp \left[ -\frac{(\hat{y} - f_k(x))^2}{2\sigma^2} \right]}{\sum_{j=1}^{L} p_j^n \exp \left[ -\frac{(\hat{y} - f_j(x))^2}{2\sigma^2} \right]} d\hat{y}
\]
\[
= p_i^n \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] d\hat{y}
\]
\[
= p_i^n.
\]

Hence, for any $x$ and $x'$, $E \left[ \sum_{i=1}^{L} f_i(x') p_i^{n+1}(x) \big| S^n = s, x^n = x \right] = \sum_{i=1}^{L} f_i(x') p_i^n$. 

61
For any convex function $f$, the general form of Jensen’s inequality is $\mathbb{E}[f(X)] \geq f(\mathbb{E}X)$. The equality of Jensen’s inequality requires that (1) $f$ is linear on $x$, or (2) $X$ is constant. (1) and (2) are equivalent to the two conditions stated in Lemma 3.4.3, respectively:

For (1), $\text{max}(X)$ is not a linear function in general, except when the reduced models are aligned (i.e., all reduced functions share the same maximum point $x$), as stated in Lemma 3.4.3, Condition 1.

For (2), this is equivalent to the following statement: $\forall x', \sum_{i=1}^{L} f(x'; \theta_i)p_{i}^{n+1}(x)$ is constant on all possible measurement outcomes $\hat{y}$ on $x$. This is further equivalent to the statement that the $p_{i}^{n+1}(x)$’s are constants. According to the updating rule given by Equation 3.2, this is equivalent to Condition 2 in Lemma 3.4.3.

□

**Lemma 3.4.4.** For $\forall n \geq 0, \forall x \in X$, the KGDP-H score $\nu_{KGDP-H,n}(x) \geq 0$. Equality holds if and only if $p_{i}^{n} = 0$ if $f(x; \theta_i) \neq f(x; \theta^*)$.

**Proof of Lemma 3.4.4**

Suppose at time $n$, the probabilities of our prior candidates are $(p_{1}^{n}, p_{2}^{n}, ..., p_{L}^{n})$. Since $x \log x$ is a convex function, by Jensen’s inequality,

$$\nu_{KGDP-H,n}(x) = \mathbb{E}^{n} \left[ \sum_{i=1}^{L} p_{i}^{n+1}(x) \log p_{i}^{n+1}(x) | S^{n} = s, x^{n} = x \right] - \sum_{i} p_{i}^{n} \log p_{i}^{n}$$

$$= \sum_{i=1}^{L} \mathbb{E}^{n} \left[ p_{i}^{n+1}(x) \log p_{i}^{n+1}(x) | S^{n} = s, x^{n} = x \right] - \sum_{i} p_{i}^{n} \log p_{i}^{n}$$

$$\geq \sum_{i=1}^{L} \mathbb{E}^{n} \left[ p_{i}^{n+1}(x) \right] \log \mathbb{E}^{n} \left[ p_{i}^{n+1}(x) \right] - \sum_{i} p_{i}^{n} \log p_{i}^{n}$$

$$= 0.$$
Since $x \log x$ is nonlinear, the only case for equality is that $p_i^{n+1}(x)$ is a constant, which means all functions with positive probability have the same value at $x$.

□

**Lemma 3.4.5.** For $\forall \omega \in \Omega_0$ and $x \in \mathcal{X}_\infty(\omega)$, we have $\lim_{n \to \infty} \nu^{KGDP-f,n}(x)(\omega) = 0$ under the KGDP-$f$ policy, and $\lim_{n \to \infty} \nu^{KGDP-H,n}(x)(\omega) = 0$ under the KGDP-$H$ policy.

**Proof of Lemma 3.4.5**

First, note that for a fixed $\omega$ and a fixed $x$, $\nu^{KGDP-f,n}(x)$ and $\nu^{KGDP-H,n}(x)$ are functions of $\vec{p} = (p_1^n, ..., p_L^n)$. In order to concentrate on $\vec{p}$, we pick a fixed $\omega \in \Omega_0$, a fixed $x \in \mathcal{X}_\infty(\omega)$, and then denote $\nu^{KGDP-f,n}(x)$ and $\nu^{KGDP-H,n}(x)$ as $\nu_f(\vec{p})$ and $\nu_H(\vec{p})$. For this fixed $x$, we assume there are $h \ (1 \leq h \leq L)$ functions equal to $f(x; \theta^*)$ at $x$. Without loss of generality, we assume they are $\theta_1, ..., \theta_h$. That is, $f(x; \theta_i) = f(x; \theta^*)$, for $\forall i \in [h]$.

Let $\vec{p}_0 = (p_1, ..., p_h, 0, 0, ..., 0)$, where $p_1 + ... + p_h = 1$. By Lemma 3.4.3 and Lemma 3.4.4, $\vec{p}_0$ meets the conditions for $\nu_f(\vec{p})$ and $\nu_H(\vec{p})$ to be 0. That is, $\nu_f(\vec{p}_0) = 0$, $\nu_H(\vec{p}_0) = 0$.

Obviously, $\nu_f(\vec{p})$ and $\nu_H(\vec{p})$ are continuous in $\vec{p}$. We take $\nu_f(\vec{p})$ as an example.

$$\sum_{i=1}^{L} f_i(x') p_i^n \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right]$$

is continuous. Therefore, we know that

$$\max_{x'} \left( \sum_{i=1}^{L} f_i(x') p_i^n \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] \right)$$

is also continuous. Hence,

$$\int_{-\infty}^{+\infty} \max_{x'} \left( \sum_{i=1}^{L} f_i(x') p_i^n \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] \right) d\hat{y}$$

is continuous, too. Then by Equation (3.7), $\nu_f(\vec{p})$ is continuous.

Since $\vec{p}$ is defined on a compact set, by the Heine-Cantor theorem, both $\nu_f(\vec{p})$ and $\nu_H(\vec{p})$ are uniformly continuous. That is, for $\forall \epsilon > 0$, there exists $\delta > 0$, such that $\forall \vec{p}_1 \neq \vec{p}_2$,

$|\vec{p}_1 - \vec{p}_2| < \delta, |\nu_f(\vec{p}_1) - \nu_f(\vec{p}_2)| < \epsilon, |\nu_H(\vec{p}_1) - \nu_H(\vec{p}_2)| < \epsilon$. 

63
For this $\omega \in \Omega_0$ and $x \in \mathcal{X}_\infty(\omega)$, since $\lim_{n \to \infty} p_i^n(\omega) = 0$, $h + 1 \leq i \leq L$, there exists $N$ such that for $\forall n > N$, $p_i^n(\omega) < \delta/(L + 1)$, $h + 1 \leq i \leq L$. Let $p_n^\gamma = (p_1^n, ..., p_{h-1}^n, p_h^n + \sum_{i=h+1}^L p_i^n, 0, ..., 0)$. Then $v_f(p_n^\gamma) = v_H(p_n^\gamma) = 0$, and $|p_n - p_n^\gamma| < \sqrt{(L - h) + (L - h)^2} \cdot \frac{\delta}{L+1} < \delta$.

Therefore, $\nu_f(p_n) < \epsilon$, $\nu_H(p_n) < \epsilon$, $\forall n > N$. Hence, for $\forall \omega \in \Omega_0$, and $x \in \mathcal{X}_\infty(\omega)$, $\lim_{n \to \infty} \nu^{KGD-f,n}(x) = 0$, $\lim_{n \to \infty} \nu^{KGD-H,n}(x) = 0$.

Lemma 3.4.8. For any $\omega \in \Omega_0$, the alternatives measured infinitely often under the KGDP-$f$ policy constitute an $x$-sufficient set. Specially, if any two candidate functions are not aligned, the alternatives measured infinitely often under the KGDP-$f$ policy constitute an $\theta$-sufficient set.

Lemma 3.4.9. For any $\omega \in \Omega_0$, the alternatives measured infinitely often under the KGDP-$H$ policy constitute a $\theta$-sufficient set.

Proof of Lemma 3.4.8 and Lemma 3.4.9

For any $\omega \in \Omega_0$, assume there are $h$ ($1 \leq h \leq L$) candidate functions that fit the true values of $\mathcal{X}_\infty(\omega)$. Without loss of generality, assume they are $\theta_1, ..., \theta_h$. So for any $l \in [h]$,

$$f(x; \theta_l) = f(x; \theta^*), \text{ for } \forall x \in \mathcal{X}_\infty(\omega).$$

By Lemma 3.4.1, for any $h + 1 \leq l \leq L$, $\lim_{n \to \infty} p_i^n(\omega) = 0$.

We now show that for any $1 \leq l \leq h$, $\lim_{n \to \infty} p_i^n(\omega) > 0$ exists.

For this $\omega$, let $L_1^n, L_2^n, ..., L_L^n$ be the likelihood of the $L$ $\theta$’s according to the first $n$ measurements. Hence,

$$L_l^n = \prod_{i=1}^n \exp\left(-\frac{[y_i - f(x_i^{i-1}; \theta_l)]^2}{2\sigma^2}\right).$$
Let $T$ be the last time that we measure any $x \notin X_{\infty}(\omega)$. After time $T$, for any $l \in \{1, 2, ..., h\}$,

$$p_l^n(\omega) = \frac{L_l^n}{\sum_{m=1}^{h} L_m^n + \sum_{m=h+1}^{L} L_m^n}$$

$$= \frac{L_l^n \prod_{i=T+1}^{n} \exp \left(-\frac{[\hat{y}^i-f(x^{i-1},\theta^*)]^2}{2\sigma^2}\right)}{\sum_{m=1}^{h} L_m^n \prod_{i=T+1}^{n} \exp \left(-\frac{[\hat{y}^i-f(x^{i-1},\theta^*)]^2}{2\sigma^2}\right) + \sum_{m=h+1}^{L} L_m^n \prod_{i=T+1}^{n} \exp \left(-\frac{[\hat{y}^i-f(x^{i-1},\theta_m^*)]^2}{2\sigma^2}\right)}.$$

In Proof of Lemma 3.4.1, we have shown that $\lim_{n \to \infty} \prod_{i=T+1}^{n} \exp \left(-\frac{[\hat{y}^i-f(x^{i-1},\theta_m^*)]^2}{2\sigma^2}\right) = 0$. Therefore,

$$\lim_{n \to \infty} p_l^n(\omega) = \frac{L_l^n}{\sum_{m=1}^{h} L_m^n} > 0 \text{ exists.}$$

We prove Lemma 3.4.9 first. We assume the contrary, i.e, $X_{\infty}$ under the KGDP-$H$ policy is not a $\theta$-sufficient set. By Lemma 3.4.5, for $\forall x \in X_{\infty}(\omega)$, we have $\lim_{n \to \infty} \nu^{KGDP-H,n}(x)(\omega) = 0$. According to the proof above, $\lim_{n \to \infty} p_l^n(\omega)$ exists for any $l$ and is strictly less than 1. Hence, there exist $x \in X_{\infty}(\omega)$ such that $f(x; \theta_1), ..., f(x; \theta_h)$ do not share the same value at this $x$, and thus $\lim_{n \to \infty} \nu^{KGDP-H,n}(x)(\omega) > 0$ exists. Since the KGDP-$H$ policy always chooses the $x$ with the largest KGDP-$H$ score, it will measure some $x \in X_{\infty}^c(\omega)$ after $T$, which is contradictory to our assumption.

For Lemma 3.4.8, we assume the contrary, too, which says that $X_{\infty}$ under the KGDP-$f$ policy is not an $x$-sufficient set. Similarly, by Lemma 3.4.5, for $\forall x \in X_{\infty}(\omega)$, $\lim_{n \to \infty} \nu^{KGDP-f,n}(x)(\omega) = 0$. According to the proof above, $\lim_{n \to \infty} p_l^n(\omega)$ exists for any $l$ and is strictly less than 1. Hence, if $X_{\infty}(\omega)$ is not an $x$-sufficient set, then there exists $1 \leq i \leq h$, such that $f(x; \theta_i)$ is not aligned with $f(x; \theta^*)$, and for any $x \in X_{\infty}(\omega)$, $f(x; \theta_i) = f(x; \theta^*)$. Therefore, either $x_1 = \arg\max_{x \in \mathcal{X}} f(x; \theta_i) \notin \cdots$
$X_\infty(\omega)$, or $x_2 = \arg\max_{x \in X} f(x; \theta^*) \not\in X_\infty(\omega)$. In this case, we would have either $\lim_{n \to \infty} \nu^{KGDP-f,n}(x_1)(\omega) > 0$, or $\lim_{n \to \infty} \nu^{KGDP-f,n}(x_2)(\omega) > 0$. Since the KGDP-$f$ policy always chooses the $x$ with the largest KGDP-$f$ score, it will measure $x_1 \not\in X_\infty(\omega)$ or $x_2 \not\in X_\infty(\omega)$ after $T$, which is contradictory to our assumption and thus $X_\infty(\omega)$ is an $x$-sufficient set.

Additionally, if the candidate functions are pairwise unaligned, then by definition, an $x$-sufficient set is also a $\theta$-sufficient set. This completes the proof for Lemma 3.4.8 and Lemma 3.4.9.

□

Theorem 3.4.10. The non-resampling KGDP-$f$ policy with truth from prior is asymptotically optimal in finding the optimal alternative. Furthermore, if the candidate functions are pairwise unaligned, then KGDP-$f$ is also asymptotically optimal in finding the correct parameter.

Proof of Theorem 3.4.10

Under the non-resampling KGDP-$f$, for any $\theta_l$ such that $\arg\max_{x \in X} f(x; \theta_l) \neq x^*$, Lemma 3.4.8 implies that for any $\omega \in \Omega_0$, there exists $x \in X_\infty(\omega)$ such that $f(x; \theta_l) \neq f(x; \theta^*)$. By Lemma 3.4.1, we have $\mathbb{P}(\lim_{n \to \infty} p^n_l = 0) = 1$. That is, $\mathbb{P}(\lim_{n \to \infty} \arg\max f(x) = x^*) = 1$.

Additionally, if the candidate functions are pairwise unaligned, then $\lim_{n \to \infty} p^n_l(\omega) = 0$ for any $\theta_l \neq \theta^*$ and $\omega \in \Omega_1$. Therefore, $\mathbb{P}(\lim_{n \to \infty} p^n(\theta^*) = 1) = 1$, and thus $\mathbb{P}({\bar{f}(x) = f(x; \theta^*)}) = 1$.

□

Remark 3.4.11. If some functions are aligned and at any $x \in X$, at least two functions intersect, then the non-resampling KGDP-$f$ policy may not find the correct parameter.

66
Example of Remark 3.4.11

Below we provide an example in which the KGDP-f policy measures an $x$-sufficient set but not a $\theta$-sufficient set, meaning that we can only identify $x^\ast$ but not $\theta^\ast$ in the limit.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure310.png}
\caption{An example in which KGDP-f identifies $x^\ast$ but not $\theta^\ast$ as $N \to \infty$.}
\end{figure}

As shown in Figure 3.10, there are three candidate functions $f_1, f_2, f_3$, and three alternatives $\mathcal{X} = \{x_1, x_2, x_3\}$. Assume $f_1$ is the truth. For $x_1$, $f_1$ and $f_2$ can be reduced to $f_{1,2}$, while for $x_2$, $f_2$ and $f_3$ can be reduced to $f_{2,3}$. If at some point, $p_3^n$ if much smaller than $p_2^n$ (as might result from measuring $x_1$), $f_{2,3}$ and $f_1$ will be aligned (note that “aligned” means that the candidate functions achieve maximums at the same alternative in the discrete set $\mathcal{X}$). Hence, $\nu_{KGDP-f,n}^x(x_2) = 0$ exactly, while for $x_1$, although $p_3^n$ goes to 0 in the limit, it is always greater than 0 at any $n$, and since $f_{1,2}$ and $f_3$ are not aligned, $\nu_{KGDP-f,n}^x(x_1) > 0$ for any $n$ (although its limit is 0). Therefore, KGDP-f would keep measuring $x_1$, and thus cannot distinguish between $f_1$ and $f_2$. 

$\square$
Theorem 3.4.12. The non-resampling KGDP-H policy with truth from prior is asymptotically optimal in finding both the optimal alternative and the correct parameter.

Proof of Theorem 3.4.12

Under the non-resampling KGDP-H policy, for any $\theta_1 \neq \theta^*$, Lemma 3.4.9 implies that for any $\omega \in \Omega_0$, there exists $x \in \mathcal{X}_\infty(\omega)$ such that $f(x; \theta_1) \neq f(x; \theta^*)$. By Lemma 3.4.1, we have $\mathbb{P}(\lim_{n \to \infty} p^n_l = 0) = 1$. That is, $\mathbb{P}(\lim_{n \to \infty} p^n(\theta^*) = 1) = 1$, and $\mathbb{P}(\tilde{f}(x) = f(x; \theta^*)) = 1$. □

3.8.3 Proof of Convergence of KGDP with Resampling

Lemma 3.5.1. There exists $\Omega_2 \in \Omega_1$, $\mathbb{P}(\Omega_2) = \mathbb{P}(\Omega_1) = 1$, such that for any $\omega \in \Omega_2$, the alternatives measured infinitely often (i.e, $\mathcal{X}_\infty(\omega)$) under the resampling KGDP-f or KGDP-H policy constitute a $\theta$-sufficient set.

Proof of Lemma 3.5.1

For any $\omega \in \Omega_1$, let $T$ be the last time that we measure any $x \notin \mathcal{X}_\infty(\omega)$.

Assume the contrary. Since $\mathcal{X}_\infty(\omega)$ is not a $\theta$-sufficient set, there exists at least one $\theta \in \mathbb{K}$, $\theta \neq \theta^*$ such that $f(x; \theta) = f(x; \theta^*)$ for all $x \in \mathcal{X}_\infty(\omega)$. Denote the set of such $\theta$’s as $\mathbb{K}'$.

Let $L^n(\theta)$ be the likelihood of any $\theta \in \mathbb{K}$ according to the first $n$ measurements. Similar to the proof of Lemma 3.4.8, if we consider each $\theta \in \mathbb{K}$ has a probability $w^n(\theta)$, then for $\forall \theta \in \mathbb{K}' \cup \{\theta^*\}$, $\lim_{n \to \infty} w^n(\theta) = \frac{L^n(\theta)}{\sum_{\theta' \in \mathbb{K}' \cup \{\theta^*\}} L^n(\theta')} > 0$; for all other $\theta$’s, by Lemma 3.4.1, $\lim_{n \to \infty} w^n(\theta) = 0$. $w^n(\theta)$ is actually proportional to the weight of $\theta$ when we do resampling. Hence, as $n$ gets larger, the $\theta$’s in $\mathbb{K}' \cup \{\theta^*\}$ always rank higher than others to enter the small pool (i.e, the sub-level set of MSE), from which we resample.

We refer to the event as $A_n$ that at least two $\theta$’s in $\mathbb{K}' \cup \{\theta^*\}$ are in the candidate set at time $n$. We claim that $A_n$ happens infinitely often almost surely (proved later). If this claim holds, then there exists $\Omega_2 \subset \Omega_1$, $\mathbb{P}(\Omega_2) = 1$, such that for any fixed $\omega \in \Omega_2$, we have
the subsequence \( \{t_n(\omega)\} \), such that at these times, the candidate set contains at least two \( \theta \)'s in \( K' \cup \{\theta^*\} \), denoted as \( \theta_1^n \) and \( \theta_2^n \). For any \( x \in \mathcal{X}_\infty(\omega) \), according to Lemma 3.4.5, \( \lim_{n \to \infty} \nu^{KGDP-f,n}(x)(\omega) = 0 \), \( \lim_{n \to \infty} \nu^{KGDP-H,n}(x)(\omega) = 0 \). On the other hand, for any \( x \) and any \( n \) such that \( f(x; \theta_1^n) \neq f(x; \theta_2^n) \), there exists an \( \epsilon > 0 \) such that \( \nu^{KGDP-H,t_n}(x)(\omega) > \epsilon \). Since \( t_n \) is an infinite sequence, this is contradictory to the fact that KGDP-H will not measure any \( x \notin \mathcal{X}_\infty(\omega) \) after time \( T \).

For KGDP-f, also consider any \( x \) at time \( t_n \) such that \( f(x; \theta_1^n) \neq f(x; \theta_2^n) \). If there also exists \( \epsilon > 0 \) such that \( \nu^{KGDP-f,t_n}(x)(\omega) > \epsilon \) infinitely often, then we are done. This happens when \( f(x; \theta_1^n) \) and \( f(x; \theta_2^n) \) are not aligned, which occurs almost surely as long as the functions \( \{f(x; \theta), \theta \in K' \cup \{\theta^*\}\} \) are not all aligned. If \( \nu^{KGDP-f,t_n}(x)(\omega) = 0 \) for any \( x \in \mathcal{X} \) for any \( t_n \) large enough, the probability of the following event: \( G = \{\omega : \exists N \text{ such that } \nu^{KGDP-f,t_n}(x)(\omega) = 0 \ \forall x \in \mathcal{X} \ \forall n > N, \text{ and } \mathcal{X}_\infty(\omega) \neq \mathcal{X}\} \) is 0, since when \( \nu^{KGDP-f,n}(x)(\omega) = 0 \), we would measure an \( x \) randomly. If neither of the above cases hold, then we must have \( \lim_{n \to \infty} \nu^{KGDP-f,t_n}(x)(\omega) = 0 \) but \( \nu^{KGDP-f,t_n}(x)(\omega) > 0 \) for \( t_n \) large enough. This could only happen if the functions \( \{f(x; \theta), \theta \in K' \cup \{\theta^*\}\} \) are all aligned, and at any sufficiently large \( t_n \), \( L_{t_n} \) always contains some \( \theta \notin K' \cup \{\theta^*\} \) selected during resampling. Similar to the proof as follows showing \( \mathbb{P}(B_0) = 0 \), where \( B_0 \) is a subcase of \( A_n \) happening infinitely often, we can see the precondition for \( H \) to happen has probability 0, and hence \( \mathbb{P}(H) = 0 \). Therefore, KGDP-f also measures a \( \theta \)-sufficient set infinitely often.

Now it suffices to show that \( A_n \) happens infinitely often almost surely. To show this, we just need to prove its opposite event happens with probability 0. Let \( B \) denote the opposite event, i.e, \( B = \{\omega \in \Omega_1 : \exists T(\omega), \text{ s.t. } \forall n > T(\omega), \mathbb{L}^n(\omega) \text{ contains at most one } \theta \in K'(\omega) \cup \{\theta^*\}\} \).
We first show $B_0$, a subset of $B$, has probability 0. Define $B_0 = \{ \omega \in \Omega_1 : \exists T(\omega), \text{s.t. } \forall n > T(\omega), \mathbb{L}^n(\omega) \text{ contains no } \theta \in \mathbb{K}^{'}(\omega) \cup \{ \theta^* \} \}$. Note that by Lemma 3.4.1, for any $\omega \in B_0$, $w^n(\theta)(\omega) \to 0$, $\forall \theta \notin \mathbb{K}^{'}(\omega) \cup \{ \theta^* \}$, and therefore, $\sum_{\theta \notin \mathbb{K}^{'}(\omega) \cup \{ \theta^* \}} w^n(\theta) \to 0$. Fix an arbitrarily small $\epsilon > 0$. For any $m \in \mathbb{Z}_{\geq 0}$, define $B_0^m = \{ \omega \in B_0 : \sum_{\theta \notin \mathbb{K}^{'}(\omega) \cup \{ \theta^* \}} w^{m-1}(\theta) \geq \epsilon, \text{ and } \sum_{\theta \notin \mathbb{K}^{'}(\omega) \cup \{ \theta^* \}} w^n(\theta) < \epsilon, \forall n \geq m \}$. Hence, $B_0^m$’s are mutually exclusive, and $\bigcup_{m \geq 0} B_0^m = B_0$. Furthermore, let $C_0^m = \{ \omega \in B_0 : \sum_{\theta \notin \mathbb{K}^{'}(\omega) \cup \{ \theta^* \}} w^n(\theta) < \epsilon, \forall n \geq m \}$, then $B_0^m \subset C_0^m$. By definition, under event $C_0^m$, there is no $\theta \in \mathbb{K}^{'} \cup \{ \theta^* \}$ being drawn in ANY resampling steps after $m$. Hence, $\mathbb{P}(C_0^m) \leq \lim_{n \to \infty} \epsilon^n = 0, \forall m \geq 0$. Therefore, $\mathbb{P}(B_0) = \sum_{m \geq 0} \mathbb{P}(B_0^m) \leq \sum_{m \geq 0} \mathbb{P}(C_0^m) = 0$.

We then show $B_1 = B \setminus B_0$ happens with probability 0. For any fixed $\omega \in B_1$, it happens for all but a finite number of times that $\mathbb{L}^n(\omega)$ contains exactly one $\theta \in \mathbb{K}^{'}(\omega) \cup \{ \theta^* \}$. According to our resampling procedure and the definition of $B_1$, there exists $n$ large enough, such that once a sufficiently probable $\theta \in \mathbb{K}^{'}(\omega) \cup \{ \theta^* \}$, say $\theta_1(\omega)$, is included in $\mathbb{L}^n$, it would never get removed (note that we allow duplicates in $\mathbb{L}^n$, and hence $\mathbb{L}^n$ may contain several copies of $\theta_1$). Furthermore, for $n$ large enough, when resampling finishes, at least $\frac{3}{4}L$ candidates are $\theta_1(\omega)$, and this would keep happening for all the resampling steps after $n$. Let $T_1(\omega)$ be the first time when such situation happens for a fixed $\omega$. For ALL resamplings after $T_1(\omega)$, by definition of $B$, only $\theta_1 \bigcup [\mathbb{K} \setminus (\mathbb{K}^{'} \cup \{ \theta^* \})] \bigcup \mathbb{K}_1$ can be drawn. Suppose at time $n$ (and not fixing $\omega$), we have at least $\frac{3}{4}L$ copies of $\theta_1$ in $\mathbb{L}$ and $\theta_1$’s likelihood is large enough. Now for the next resampling, the probability of not selecting any $\theta$ from a particular set $\mathbb{K}_1 \subset \mathbb{K}$ is no larger than $1 - \sum_{\theta \in \mathbb{K}_1} w(\theta)$.

For any fixed $\omega \in B_1$, define $q^n(\omega) = 1 - \sum_{\theta \notin \mathbb{K}^{'}(\omega) \cup \{ \theta^* \}} w^n(\theta)$. We know for any $\omega \in B_1 \subset \Omega_1$, $\lim_{n \to \infty} q^n(\omega) < 1$ exists. For any $\lambda \in (0, 1)$ such that $\lim_{n \to \infty} q^n(\omega) = \lambda$, define $T_2(\omega)$ as the time such that for any $n \geq T_2(\omega)$, $q^n(\omega) < \frac{1+\lambda}{2}$. Moreover, define $B_1^n(\lambda) = \{ \omega \in B_1 : \lim_{n \to \infty} q^n(\omega) = \lambda, \max \{ T_1(\omega), T_2(\omega) \} = m \}$. Then, $\mathbb{P}(B_1^n(\lambda)) \leq \lim_{n \to \infty} \left( \frac{1+\lambda}{2} \right)^n = 0$ for
any \( m > 0 \) and \( m \to \infty \). Therefore,

\[
\mathbb{P}(B_1) = \sum_{m \geq 0} \int_{0 < \lambda < 1} \mathbb{P}(B_1^m(\lambda)) \mathbb{d}\mathbb{P}\left(\{\omega \in \Omega_1 : \lim_{n \to \infty} q^n(\omega) = \lambda\}\right)
\]

\[
= \sum_{m \geq 0} \int_{0 < \lambda < 1} 0 \cdot \mathbb{d}\mathbb{P}\left(\{\omega \in \Omega_1 : \lim_{n \to \infty} q^n(\omega) = \lambda\}\right)
\]

\[
= 0.
\]

Therefore, \( \mathbb{P}(B) = \mathbb{P}(B_0 \cup B_1) = 0 \), and thus \( A_n \) happens infinitely often.

\[\square\]

**Theorem 3.5.2.** The resampling KGDP-\( f \) policy is asymptotically optimal in finding the optimal alternative and the correct parameter. The same holds for the resampling KGDP-\( H \) policy.

**Proof of Theorem 3.5.2**

We need to show that with probability 1, \( \mathbb{L}^n \) contains \( \theta^* \) all but a finite number of times (if this statement does not hold, then even as \( n \to \infty \), there are always times when \( \bar{f} \) is wrong and hence our estimate \( \hat{x}^n \) is wrong).

To prove the statement above, we only need to consider the times when resampling happens, since \( \mathbb{L}^n \) only changes during resampling. By Theorem 3.5.1, for any \( \omega \in \Omega_2 \), \( \mathcal{X}_\infty(\omega) \) is a \( \theta \)-sufficient set, and thus \( \lim_{n \to \infty} w^n(\theta^*)(\omega) = 1 \). Let \( C_n \) be the event that \( \theta^* \) is contained in the candidate set after \( n \)-th resampling finishes (note that in the notation of \( C_n \), \( n \) indexes the *resampling time*, rather than the measurement time).

Instead of showing \( C_n \) happens all but a finite number of times directly, we first show \( C_n \) happens infinitely often almost surely, using a similar logic to the proof of Theorem 3.5.1. For any \( \omega \), let \( s^n(\omega) \) be the \( n \)-th time that resampling happens. Let \( D \) be the opposite event of \( C_n \) happening infinitely often. Fix an (arbitrarily small) \( \epsilon > 0 \). For any \( m \in \mathbb{Z}_{\geq 0} \), define

\[
D^m = \{\omega \in D : w^{s_n}(\theta^*)(\omega) \leq 1 - \epsilon, \text{ and } w^{s_n}(\theta^*)(\omega) > 1 - \epsilon, \forall n \geq s_m(\omega)\}. \]

Hence, \( D^m \)'s
are mutually exclusive, and $\bigcup_{m \geq 0} D^m = D$. Furthermore, let $E^m = \{ \omega \in D : \omega^n(\theta^*)(\omega) > 1 - \epsilon, \forall n \geq s_m(\omega) \}$, then $D^m \subset E^m$. By definition, under event $E^m$, there is no $\theta \neq \theta^*$ being drawn in ANY resampling step after $m$. Hence, $P(E^m) \leq \prod_{n=m}^{\infty} \epsilon = 0$, $\forall m \geq 0$. Therefore, $P(D) = \sum_{m \geq 0} P(D^m) \leq \sum_{m \geq 0} P(E^m) = 0$, and hence $C_n$ happens infinitely often almost surely.

Let $\Omega_3$ denote the set of almost sure events that $C_n$ happens infinitely often. For any $\omega \in \Omega_3$, once $\theta^*$ is included in $L^n$ for $n$ large enough, by our resampling procedure, its probability is always greater than the threshold of probabilities $\epsilon$, and thus would never be removed. Hence, it happens almost surely that $\theta^*$ is included in $L^n$ all but a finite number of times.

This further means that $\forall \omega \in \Omega_3$, we have $\lim_{n \to \infty} \tilde{f}_n(x)(\omega) = \lim_{n \to \infty} \sum_{\theta \in L^n(\omega)} p^n(\theta)(\omega)f_i(x; \theta) = f(x; \theta^*)$. Therefore, with probability 1, both the KGDP-$f$ and KGDP-$H$ policies find the true $\theta^*$ and optimal $x^*$ as $n \to \infty$.

$\square$
Chapter 4

Optimal Learning for Nonlinear Parametric Belief Models over Multidimensional Continuous Alternatives

In the previous chapter, we propose a resampling algorithm that works efficiently with multi-dimensional parameter spaces, which lifts KGDP’s constraint of handling only a small set of candidate parameters. However, a similar constraint still exists for the alternative space. In particular, most previous knowledge gradient models, including the original KGDP model (Chen et al. 2014), optimize over a finite set of sampled alternatives, which works well with low-dimensional or categorical alternatives but may be problematic with multidimensional and numerical alternatives. In this chapter, we propose a gradient-based algorithm to deal with nonlinear parametric functions defined on multidimensional continuous alternative spaces. We also prove that our method is asymptotically optimal in theory, and demonstrate experimental results to show its strong performance in practice.

The material of this chapter was presented in INFORMS Annual Meeting 2016.
4.1 Introduction

We consider the setting where we have to learn a vector of parameters $\theta$ while simultaneously finding the best set of continuous controls $x$ for guiding an experimental process that is expensive to run, and which produces uncertain results. This might be finding the best temperature, pressure and concentrations for running a laboratory experiment to design a new material, the best dosages in a drug treatment regime, or the best physical configuration of a device such as an aerosol can (diameters, lengths, separations) to produce the best spray.

We assume that our physical process can be well approximated by a parametric function $f(x; \theta)$ that depends on our continuous controllable parameters $x$ (we call $x$ controls or alternatives), and a set of unknown parameters $\theta$. Our goal is to design a learning policy that produces the best estimate of the true parameter $\theta^*$ along with the $x$ that maximizes $f(x; \theta^*)$. Experiments are noisy and expensive, and the time and cost limit us to a budget of $N$ experiments.

In this chapter, we propose an algorithm based on a class of Bayesian policies, known as the Knowledge Gradient (KG) policy (see a detailed review of KG and related work on optimizing noisy functions in Chapter 1 and Chapter 2). The Knowledge Gradient policy maximizes the expected value of information from a single experiment, which is originally designed for a finite set of alternatives. KG has been developed for look-up table belief models (either independent (Gupta & Miescke 1996, Frazier et al. 2008) or correlated (Frazier et al. 2009)), linear belief models (Negescu et al. 2011) and nonlinear belief models (Chen et al. 2014). In all these models, KG has been shown to be the only stationary policy that is both myopically optimal and asymptotically optimal. Although KG has shown impressive performance in all these models, the limitation of finite alternatives becomes a constraint especially for higher dimensional problems, since we cannot discretize the alternative space satisfactorily as its dimensionality grows. Moreover, there are many real world applications where we need to make a decision from a continuous, multidimensional space. For example, in materials science, we may need to pick the best temperature, pressure, concentration and
size, which are all continuous controllable variables. Scott et al. (2011) is the first attempt to extend KG to continuous problems. It models problems using a look-up table belief model and proposes an algorithm based on Gaussian process regression using previous observations. However, the use of a nonparametric belief model limited it to at most three or possibly four dimensions.

In this chapter, we extend the knowledge gradient policy on parametric belief models to cases where $x$ is a vector which is both continuous and possibly high dimensional (we have applied our method to problems with 10 or 20 dimensions, without any indication that this is the limit). We require that we have access to a parametric belief model, and for this work, we assume that the model is globally accurate within a specified region. We further assume that the model is differentiable, and exploit this property in the design of our global search algorithm. There are many applications in science and engineering, especially in computational materials science, whose outputs do exhibit certain features of their inputs and can be structured with some parametric belief models, such as Chen et al. (2014), Negoescu et al. (2011).

Our work makes the following contributions: (1) We show how to adapt the knowledge gradient policy, previously developed primarily for discrete alternatives, to optimize unknown parametric models with continuous and multidimensional design variables. (2) We describe a gradient-based algorithm for maximizing the knowledge gradient with respect to the design vector. (3) We prove that our algorithm is able to find both the optimal alternative and the correct parameters almost surely as our budget goes to infinity. (4) We demonstrate experimentally that our method outperforms the knowledge gradient based on a sampled representation of the design vector (the standard methodology in the past).

The remainder of this chapter is organized as follows. We introduce our algorithm in Section 4.2, and prove it is asymptotically optimal in Section 4.3, which means it can find both the optimal alternative and identify the correct parameter as we take an infinite number
of experiments. Section 4.4 shows empirical results on two applications, and Section 4.5 summarizes our work in this chapter.

4.2 Knowledge Gradient for Parametric Models with Continuous Alternatives

We consider the problem:

$$\argmax_{x \in \mathcal{X}} f(x; \theta^*)$$,

where $f(x; \theta)$ is a function differentiable in $x$, $\mathcal{X}$ is a compact set of alternatives (or, decisions), and $\theta$ is a vector denoting the unknown parameters. Our goal is to find the optimal alternative $x^*$ and estimate the true parameter $\theta^*$ after a budget of $N$ measurements. We assume the measurements have Gaussian noise $W \sim \mathcal{N}(0, \sigma^2)$, i.i.d, where $\sigma$ is known.

In the optimal learning framework, we want to maximize the expected value of information, namely the knowledge gradient (KG), in each experiment. We propose an algorithm to handle continuous alternatives $x$. There are mainly three steps in our algorithm for finding the alternative to measure next which has the largest knowledge gradient score. We first propose an approximation of the KG score. We then develop a method to calculate the derivative of the KG score. Finally, we search for the global maximum of KG using a gradient-based search algorithm. This logic is then coupled with the resampling logic in Chapter 3 for handling large parameter spaces.

4.2.1 Approximation of the Knowledge Gradient

By definition, the formula of KG for continuous $x$ is given by:

$$\nu^{KG,n}(x) = \mathbb{E} \left( \max_{x' \in \mathcal{X}} \sum_{i=1}^{L} f(x'; \theta_i) p_{i}^{n+1}(x) | S^n = s, x^n = x \right) - \max_{x' \in \mathcal{X}} \sum_{i=1}^{L} f(x'; \theta_i) p_{i}^{n}.$$
Since $\mathcal{X}$ is continuous, $\nu^{KG,n}$ is computationally intractable. We thus choose to use a finite set of samples in $\mathcal{X}$ to approximate the knowledge gradient and transform the problem to the traditional settings in Knowledge Gradient with Discrete Priors (KGDP) (Chen et al. 2014). Specifically, let $\mathcal{X}_s$ be a finite subset of samples from $\mathcal{X}$, and define the approximation of KG as:

$$
\nu^{KG-s,n}(x) = \mathbb{E} \left( \max_{x' \in \mathcal{X}_s} \sum_{i=1}^{L} f(x'; \theta_i)p_i^{n+1}(x)|S^n = s, x^n = x \right) - \max_{x' \in \mathcal{X}_s} \sum_{i=1}^{L} f(x'; \theta_i)p_i^n,
$$

(4.1)

where the probability updating function is given by Bayes rule as:

$$
p_{i}^{n+1} = \mathbb{P}(\theta_i|\hat{y}^{n+1}) = \frac{g(\hat{y}^{n+1} | \theta_i) \cdot \mathbb{P}(\theta_i)}{g(\hat{y}^{n+1})} = \frac{1}{\sqrt{2\pi\sigma}} \exp \left[ - \frac{(\hat{y}^{n+1} - f(x; \theta_i))^2}{2\sigma^2} \right] \cdot p_i^n,
$$

(4.2)

where $g(\hat{y}^{n+1})$ is the probability density of the experiment outcome $\hat{y}^{n+1}$, given by

$$
g(\hat{y}^{n+1}) = \sum_{i=1}^{L} \frac{1}{\sqrt{2\pi\sigma}} \exp \left[ - \frac{(\hat{y}^{n+1} - f(x; \theta_i))^2}{2\sigma^2} \right] \cdot p_i^n.
$$

Therefore, instead of calculating the exact KG scores based on all points in $\mathcal{X}$, we calculate an approximation using the subset of points $\mathcal{X}_s$, and then search for the global maximum of the approximation as the alternative to measure next. To generate $\mathcal{X}_s$, we sample a number of points uniformly at random from $\mathcal{X}$. Empirically, we do not need a large sample to obtain a satisfactory approximation. Figures 4.1 and 4.2 illustrate the approximation in one- and two-dimensional settings, respectively.

**4.2.2 Calculation of $\nu^{KG-s}(x)$ and $\nabla
\nu^{KG-s}(x)$**

Although Equation (4.1) can be evaluated using numerical integration methods or Monte Carlo simulation, as in Chen et al. (2014) and Chapter 3, but there has been no formula to evaluate the gradient of KG. Here we propose a new method that can calculate both $\nu^{KG-s,n}(x)$ and its gradient at any $x$ in the continuous space $\mathcal{X}$. 
Since the expectation in Equation (4.1) is taken over $\hat{y}^{n+1}$, combining Equations (4.1) and (4.2), we can rewrite Equation (4.1) as

\begin{equation}
\nu^{KG-s,n}(x) = \int_{-\infty}^{\infty} \left[ \max_{x' \in X} \sum_{i=1}^{L} f(x'; \theta_i) \frac{1}{\sqrt{2\pi}\sigma} \exp \left[ -\frac{(\hat{y} - f(x'; \theta_i))^2}{2\sigma^2} \right] \cdot p_i^n \right] g(\hat{y}) d\hat{y} - \max_{x' \in X} \sum_{i=1}^{L} f(x'; \theta_i)p_i^n \quad (4.4)
\end{equation}

\begin{equation}
= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} \left\{ \max_{x' \in X} \sum_{i=1}^{L} f(x'; \theta_i)p_i^n \exp \left[ -\frac{(\hat{y} - f(x'; \theta_i))^2}{2\sigma^2} \right] \right\} d\hat{y} - \max_{x' \in X} \sum_{i=1}^{L} f(x'; \theta_i)p_i^n \quad (4.5)
\end{equation}

\begin{equation}
= \nu_1(x) - \nu_2. \quad (4.6)
\end{equation}

Let us denote the first term of Equation (4.5) as $\nu_1(x)$ and the second term, which is a constant independent of $x$, as $\nu_2$. Below we formulate our method of calculating $\nu^{KG-s,n}(x)$ and $\nabla_{\nu}^{KG-s}$. 

Figure 4.1: Illustration of the approximation of KG in one-dimensional $\mathcal{X}$.

Figure 4.2: Illustration of the approximation of KG in two-dimensional $\mathcal{X}$.
Calculation of the KG Score $\nu^{KG-s,n}(x)$

In this section, we propose a method to calculate $\nu^{KG-s,n}(x)$. Suppose $X_s = \{x_1, x_2, ..., x_M\}$. Let $h_{x'}(x', \hat{y}) = \sum_{i=1}^{L} f(x'; \theta_i)p_i^n \exp\left[-\frac{(\hat{y} - f(x'; \theta_i))^2}{2\sigma^2}\right]$. Note that $h_{x'}(x', \hat{y})$ is a function of $x$, but for the following discussion of $\nu^{KG-s,n}(x)$, let us fix $x$ and for simplicity write $h_{x'}(x', \hat{y})$ as $h(x', \hat{y})$. Then we have

$$\nu_1(x) = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{+\infty} \max\left\{ h(x_1, \hat{y}), h(x_2, \hat{y}), \ldots, h(x_M, \hat{y}) \right\} d\hat{y} = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{+\infty} \max h(X_s, \hat{y}) d\hat{y}.$$

Note that, for any fixed $m$, $1 \leq m \leq M$, $h(x_m, \hat{y})$ is a mixture of $L$ Gaussian functions, with the heights being $f(x'; \theta_i)p_i^n$ and the centers at $f(x; \theta_i)$, $1 \leq i \leq L$. Thus $h(x_m, \hat{y})$ has at most $L$ local maxima. Let us now consider the function $t(\hat{y}) = \max_{1 \leq m \leq M} h(x_m, \hat{y})$. Since for each $x_m$, $h(x_m, \hat{y})$ has only a finite number of local maxima, there are at most a finite number of intervals on the $\hat{y}$ axis such that $h(x_m, \hat{y}) = t(\hat{y})$ on these intervals. Therefore, the $\hat{y}$ axis can be divided into a finite number of intervals, such that in each interval, there is a unique $x'$ such that $x' = \text{argmax}_{1 \leq m \leq M} h(x_m, \hat{y})$.

The first image shows three functions $f_1(x), f_2(x)$ and $f_3(x)$, with $(p_1, p_2, p_3) = (\frac{1}{2}, \frac{1}{3}, \frac{1}{6})$. The second image shows that if we measure $x = 0.4$, the outcome $\hat{y}$ has a distribution of a mixture of 3 Gaussians, centered at $(f_1(0.4), f_2(0.4), f_3(0.4))$, respectively. The third image shows the sum of the 3 Gaussians, namely the probability density of $\hat{y}$.

Figure 4.3: Example of the parametric functions.
Assume we have $K$ such intervals on the $\hat{y}$ axis, in each of which there is a unique $x'$ such that $x' = \arg\max_{1 \leq m \leq M} h(x_m, \hat{y})$. If $K = 1$, there exists an $x'$ that always maximizes $\tilde{h}(\mathcal{X}_s, \hat{y})$ for any possible outcome $\hat{y}$. Without loss of generality, let us assume it is $x_1$. Therefore,

$$
\nu_1(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{+\infty} h(x_1, \hat{y}) d\hat{y} = \sum_{i=1}^{L} f(x_1; \theta_i)p_i^n.
$$

Note that $\nu_2 \geq \sum_{i=1}^{L} f(x_1; \theta_i)p_i^n$. Also by Proposition 4.3.2, we have $\nu_1(x) \geq \nu_2$. Hence, $\nu_1(x) = \nu_2$ and $\nu^{KG-s,n}(x) = 0$. Thus, $x$ is a local (also global) minimum of $\nu^{KG-s,n}(x)$, so $\nabla \nu^{KG-s,n}(x) = 0$.

If $K > 1$, we can divide the $\hat{y}$ axis into $K$ intervals, separated by $K - 1$ points $\{S_1, S_2, ..., S_{K-1}\}$. Figure 4.4 shows an example corresponding to the case in Figure 4.3. Define $S_0 = -\infty$ and $S_K = \infty$, and let $I_1 = (-S_0, S_1), I_2 = [S_1, S_2), ..., I_K = [S_{K-1}, S_K)$ be the $K$ intervals. On each interval $I_k$ ($1 \leq k \leq K$), denote $x^{(k)}$ as the alternative in $\mathcal{X}_s$ that

Consider the example in Figure 4.3, with $\mathcal{X}_s = \{1, 2, 3, 4\}$. Assume we measure at $x = 0.4$. Then $(f_1(0.4), f_2(0.4), f_3(0.4)) = (1, 2, 2.5)$. Hence, all $h(x, \hat{y})$’s are scaled Gaussian mixtures centered $(1, 2, 2.5)$. Take $h(x_4, \hat{y})$ for example. The heights of the 3 Gaussians are $f_1(4)p_1, f_2(4)p_2, f_3(4)p_3$, respectively. For $\hat{y} \in (-\infty, S_1)$, $h(x_4, \hat{y})$ is the largest so $x^{(1)} = x_4$. Similarly, $x^{(2)} = x_3$ and $x^{(3)} = x_2$.

Figure 4.4: Illustration of $\{S_k\}$ and $\{x^{(k)}\}$. 

80
maximizes $h(x, \hat{y})$, i.e.,

$$x^{(k)} = \underset{x \in X}{\text{argmax}} \ h(x, \hat{y}), \forall \hat{y} \in I_k.$$  \hfill (4.7)

By definition of $\{S_k\}$, we have

$$h(x^{(k)}, S_k) = h(x^{(k+1)}, S_k), \forall 1 \leq k \leq K - 1.$$  \hfill (4.8)

We can now write $\nu_1(x)$ as

$$\nu_1(x) = \frac{1}{\sqrt{2\pi\sigma}} \sum_{k=1}^{K} \int_{I_k} h(x^{(k)}, \hat{y}) d\hat{y}$$  \hfill (4.9)

$$= \frac{1}{\sqrt{2\pi\sigma}} \sum_{k=1}^{K} \sum_{i=1}^{L} f(x^{(k)}; \theta_i) p^n_i \exp \left[ -\frac{(\hat{y} - f(x; \theta_i))^2}{2\sigma^2} \right] d\hat{y}$$  \hfill (4.10)

$$= \sum_{k=1}^{K} \sum_{i=1}^{L} f(x^{(k)}; \theta_i) p^n_i \int_{I_k} \frac{1}{\sqrt{2\pi\sigma}} \exp \left[ -\frac{(\hat{y} - f(x; \theta_i))^2}{2\sigma^2} \right] d\hat{y}$$  \hfill (4.11)

$$= \sum_{k=1}^{K} \sum_{i=1}^{L} f(x^{(k)}; \theta_i) p^n_i \left[ \Phi \left( \frac{S_k - f(x; \theta_i)}{\sigma} \right) - \Phi \left( \frac{S_{k-1} - f(x; \theta_i)}{\sigma} \right) \right],$$  \hfill (4.12)

where $\Phi(z)$ is the cumulative density function of the standard normal random variable $Z$. Since $S_0 = -\infty$ and $S_K = \infty$, we have $\Phi \left( \frac{S_0 - f(x; \theta_i)}{\sigma} \right) = 0$ and $\Phi \left( \frac{S_K - f(x; \theta_i)}{\sigma} \right) = 1$ for any $i$. Therefore, we can further write $\nu^{KG-s,n}(x)$ as

$$\nu^{KG-s,n}(x)$$

$$= \sum_{i=1}^{L} \sum_{k=1}^{K} f(x^{(k)}; \theta_i) p^n_i \left[ \Phi \left( \frac{S_k - f(x; \theta_i)}{\sigma} \right) - \Phi \left( \frac{S_{k-1} - f(x; \theta_i)}{\sigma} \right) \right] - \max_{x' \in X_s} \sum_{i=1}^{L} f(x'; \theta_i) p^n_i$$

$$= \sum_{i=1}^{L} p^n_i \left\{ f(x^{(K)}; \theta_i) + \sum_{k=1}^{K-1} \Phi \left( \frac{S_k - f(x; \theta_i)}{\sigma} \right) \left[ f(x^{(k)}; \theta_i) - f(x^{(k+1)}; \theta_i) \right] \right\}$$

$$- \max_{x' \in X_s} \sum_{i=1}^{L} f(x'; \theta_i) p^n_i.$$  \hfill (4.13)
Equation (4.13) provides a new equation to calculate the KG score for any point \( x \in \mathcal{X} \). As long as we know the separating points \( \{S_k\} \) up to some precision, we can calculate \( \nu_{KG-s,n}^{x} \) to that precision.

**Calculation of \( \nabla \nu_{KG-s,n}^{x} \)**

We now compute the gradient of the KG score \( \nabla \nu_{KG-s,n}^{x} \). To do this, we take the derivative of \( \nu_{KG-s,n}^{x} \) using Equation (4.9). Remember that \( \{S_k\} \) and \( \{x^{(k)}\} \) are all functions of \( x \). If we change \( x \) by a small amount, \( S_k \)'s will also move by a small amount. If this amount is small enough, the \( \hat{y} \) axis can be divided into roughly the same number of intervals, each of which is associated with the same \( x^{(k)} \). Therefore, \( \nabla x x^{(k)} = 0 \).

Now we calculate the gradient \( \nabla_{x} \nu_{KG-s,n}^{x} \).

\[
\nabla_{x} \nu_{KG-s,n}^{x} = \nabla_{x} \frac{1}{\sqrt{2\pi\sigma}} \sum_{k=1}^{K} \int_{I_k} h(x^{(k)}, \hat{y})d\hat{y}
\]

(4.14)

\[
= \frac{1}{\sqrt{2\pi\sigma}} \sum_{k=1}^{K} \nabla_{x} \int_{I_k} h(x^{(k)}, \hat{y})d\hat{y}
\]

(4.15)

\[
= \frac{1}{\sqrt{2\pi\sigma}} \sum_{k=1}^{K} \left[ -h(x^{(k)}, S_{k-1}) \cdot (\nabla_{x} S_{k-1}) + h(x^{(k)}, S_{k}) \cdot (\nabla_{x} S_{k}) + \int_{I_k} \nabla_{x} h(x^{(k)}, \hat{y})d\hat{y} \right]
\]

(4.16)

\[
= \frac{1}{\sqrt{2\pi\sigma}} \sum_{k=1}^{K} \left[ -h(x^{(k)}, S_{k-1}) \cdot (\nabla_{x} S_{k-1}) + h(x^{(k)}, S_{k}) \cdot (\nabla_{x} S_{k}) \right]
\]

+ \frac{1}{\sqrt{2\pi\sigma}} \sum_{k=1}^{K-1} \int_{I_k} \nabla_{x} h(x^{(k)}, \hat{y})d\hat{y}
\]

(4.17)

\[
= \frac{1}{\sqrt{2\pi\sigma}} \sum_{k=1}^{K-1} \left\{ [h(x^{(k)}, S_{k}) - h(x^{(k+1)}, S_{k})] \cdot (\nabla_{x} S_{k}) \right\}
\]

+ \frac{1}{\sqrt{2\pi\sigma}} \sum_{k=1}^{K} \int_{I_k} \nabla_{x} h(x^{(k)}, \hat{y})d\hat{y}
\]

(4.18)

\[
= \frac{1}{\sqrt{2\pi\sigma}} \sum_{k=1}^{K} \int_{I_k} \nabla_{x} h(x^{(k)}, \hat{y})d\hat{y}.
\]

(4.19)
Equation (4.15) to Equation (4.16) is because of Leibniz’s rule. Equation (4.18) to Equation (4.19) is due to Equation (4.8). Similar to the derivation in Section 4.2.2, we can further calculate Equation (4.19) as

\[ \nabla_x \nu^{KG-s,n}(x) = \frac{1}{\sqrt{2\pi}\sigma} \sum_{k=1}^{K} \int_{I_k} \nabla_x h(x^{(k)}, \hat{y}) d\hat{y} \]

\[ = \frac{1}{\sqrt{2\pi}\sigma} \sum_{k=1}^{K} \int_{I_k} \nabla_x \left( \sum_{i=1}^{L} f_i(x^{(k)}) p^n_i \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] \right) d\hat{y} \]

\[ = \frac{1}{\sqrt{2\pi}\sigma} \sum_{k=1}^{K} \int_{I_k} \sum_{i=1}^{L} \left\{ f'_i(x^{(k)}) \nabla_x(x^{(k)}) \right\} p^n_i \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] d\hat{y} + f_i(x^{(k)}) p^n_i \nabla_x \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] d\hat{y}, \]

where \( f_i(x) = f(x; \theta_i) \) and \( f'_i(x) = \nabla_x f(x; \theta_i) \).

Since \( \nabla_x(x^{(k)}) = 0 \), we have

\[ \nabla_x \nu^{KG-s,n}(x) \]

\[ = \frac{1}{\sqrt{2\pi}\sigma} \sum_{k=1}^{K} \int_{I_k} \sum_{i=1}^{L} f_i(x^{(k)}) p^n_i \nabla_x \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] d\hat{y} \]

\[ = \frac{1}{\sqrt{2\pi}\sigma} \sum_{k=1}^{K} \int_{I_k} \sum_{i=1}^{L} f_i(x^{(k)}) p^n_i \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] \cdot \frac{\hat{y} - f_i(x)}{\sigma^2} \cdot f'_i(x) d\hat{y} \]

\[ = \frac{1}{\sqrt{2\pi}\sigma} \sum_{k=1}^{K} \sum_{i=1}^{L} p^n_i f_i(x^{(k)}) f'_i(x) \int_{I_k} \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] \cdot \frac{\hat{y} - f_i(x)}{\sigma^2} d\hat{y} \]

\[ = \frac{1}{\sqrt{2\pi}\sigma} \sum_{k=1}^{K} \sum_{i=1}^{L} p^n_i f_i(x^{(k)}) f'_i(x) \int_{I_k} \exp \left[ -\frac{(\hat{y} - f_i(x))^2}{2\sigma^2} \right] d(\hat{y} - f_i(x))^2 \]

\[ = \frac{1}{\sqrt{2\pi}\sigma} \sum_{k=1}^{K} \sum_{i=1}^{L} f'_i(x) p^n_i \sum_{k=1}^{K-1} f_i(x^{(k)}) \left\{ \exp \left[ -\frac{(S_{k-1} - f_i(x))^2}{2\sigma^2} \right] - \exp \left[ -\frac{(S_k - f_i(x))^2}{2\sigma^2} \right] \right\} \]

\[ = \frac{1}{\sqrt{2\pi}\sigma} \sum_{i=1}^{L} f'_i(x) p^n_i \sum_{k=1}^{K-1} \left[ f_i(x^{(k+1)}) - f_i(x^{(k)}) \right] \exp \left[ -\frac{(S_{k-1} - f_i(x))^2}{2\sigma^2} \right]. \tag{4.20} \]

Again, if we know \( \{S_k\} \) accurately enough, we would be able to calculate \( \nabla_x \nu^{KG-s,n}(x) \) to a certain precision according to Equation (4.20).
Calculation of \( \{S_k\} \) and \( \{x^{(k)}\} \)

By definition, \( S_k \)'s are solutions (in \( \hat{y} \) space) to a group of equations \( h(x_i, \hat{y}) = h(x_j, \hat{y}) \). Since \( h(x_m, \hat{y}) \) is a scaled mixture of Gaussian distributions, typically there are no closed form solutions. However, we can find \( \{S_k\} \) heuristically by discretizing the \( \hat{y} \) axis in certain regions of interest, and then use bisection search to achieve a certain precision. After \( S_k \)'s are determined, \( x^{(k)} \)'s can be found by Equation (4.7). For more detail, please refer to Appendix 4.6.1.

4.2.3 Maximizing \( \nu^{KG-s,n}(x) \)

Once we have the formula to calculate \( \nu^{KG-s,n}(x) \) and \( \nabla \nu^{KG-s,n}(x) \), we can use gradient ascent to find a local maximum of \( \nu^{KG-s,n}(x) \). The only remaining question is how to find the starting point of the gradient ascent. Our main goal is to avoid starting from some really bad point that might result in a very small local maximum. Our method of selection works as follows: We first randomly sample, say, 100 points, calculate their \( \nu^{KG-s,n}(x) \)'s, and take the maximum as one starting point and run the gradient ascent to get a local maximum. We repeat the above process for multiple times (say 10-20 times) and pick the largest point among all the local maxima. We call the number of repeating times the number of restarts of the gradient ascent algorithm.

An illustration of the gradient ascent in two-dimensional \( \mathcal{X} \) is shown in Figure 4.5.

![Illustration of the gradient ascent process to maximize \( \nu^{KG,s}(x) \).](image)

Figure 4.5: Illustration of the gradient ascent process to maximize \( \nu^{KG,s}(x) \).
Since the KG surface in parametric models is parameterized by \( \theta \) and hence is generally smooth with not too many local maximums, we usually do not need to repeat gradient ascent very often. We evaluate the effect of using different numbers of restarts of the gradient ascent in Section 4.4, where it shows that empirically, 10 restarts would be sufficient for the ten-dimensional problem defined in Section 4.4.

### 4.2.4 Overall Procedure

Now we have an algorithm to calculate and maximize \( \nu^{KG-s,n} \) at each step as an approximation of the exact solution \( \nu^{KG,n} \). We define the continuous KG policy as choosing the \( x \in \mathcal{X} \) that has the largest approximated KG score to measure next, i.e,

\[
x^n = \arg\max_{x \in \mathcal{X}} \nu^{KG-s,n}(x),
\]

while this maximization problem is solved heuristically by the algorithm above.

Moreover, we combine the continuous KG policy with the resampling scheme in \( \theta \) space as proposed in Chapter 3. The overall procedure, which solves the problem in multidimensional continuous \( x \) space and \( \theta \) space, is shown in Algorithm 3.

### 4.3 Convergence of Continuous KG

In this section, we prove that our continuous KG policy (coupled with the resampling policy) is asymptotically optimal. In other words, as the budget \( N \to \infty \), our algorithm is able to find both the correct parameter and the optimal alternative almost surely. We first discuss the convergence of the continuous KG policy when no resampling is involved given that we have the true parameter in prior. We show that our algorithm is able to find the optimal \( x \) in the sampled set \( \mathcal{X}_s \); in addition, if any two candidates \( \theta_i, \theta_j \) satisfy

\[
\arg\max_{x \in \mathcal{X}_s} f(x; \theta_i) \neq \arg\max_{x \in \mathcal{X}_s} f(x; \theta_j),
\]

our algorithm is also asymptotically optimal in finding the true parameter \( \theta^* \) and the optimal alternative \( x^* \) in \( \mathcal{X} \). We next show that with
Algorithm 3 Overall Procedure: multidimensional continuous $x$ and $\theta$

**Input:** Budget: $N$; Alternative space: $\mathcal{X}$; A large pool of parameters: $\mathbb{K}$; Number of Candidates: $L$; Noise: $\sigma$; Other resampling parameters.

**Output:** Estimate of the optimal alternative: $\hat{x}^*$, estimate of the true parameter: $\hat{\theta}^*$.

1: Initialization: Draw $L$ $\theta$’s from $\mathbb{K}$ to form $\mathbb{L}^0$; $p_1^0 = p_2^0 = \ldots = p_L^0 = \frac{1}{L}$.

2: for $n = 0$ to $N - 1$ do

3: Use the continuous KG policy to pick the alternative to measure next:

$$x^n = \arg\max_{x \in \mathcal{X}} \nu_{KG-s,n}^{x}(x).$$

4: Measure at $x^n$ and get result $\hat{y}^{n+1}$.

5: Update probabilities:

$$p_i^{n+1} = \frac{\exp\left[-\frac{(\hat{y}^{n+1} - f(x; \theta_i))}{2\sigma^2}\right] \cdot p_i^n}{\sum_{l=1}^L \exp\left[-\frac{(\hat{y}^{n+1} - f(x; \theta_l))}{2\sigma^2}\right] \cdot p_l^n}.$$  

6: while resampling is triggered do

7: Resample in $\theta$ space and get $\mathbb{L}^{n+1}$.

8: Recalculate probabilities (using all measurement histories):

$$p_i^{n+1} = \frac{\prod_{j=0}^n \exp\left[-\frac{(\hat{y}^{j+1} - f(x; \theta_i^{n+1}))}{2\sigma^2}\right]}{\sum_{l=1}^L \prod_{j=0}^n \exp\left[-\frac{(\hat{y}^{j+1} - f(x; \theta_l^{n+1}))}{2\sigma^2}\right]}.$$  

9: end while

10: end for

11: return $\hat{x}^* = \arg\max_{x \in \mathcal{X}} \sum_{i=1}^L p_i^N f(x; \theta_i^N)$, $\hat{\theta}^* = \arg\min_{\theta \in \mathbb{K}} MSE(\theta | \mathcal{F}^N)$.

resampling, our continuous KG algorithm is always asymptotically optimal in finding both the true parameter $\theta^*$ and the optimal alternative $x^*$ in $\mathcal{X}$.

We assume the following condition holds throughout the following proof:

**Assumption 4.3.1.** The gradient ascent algorithm in continuous KG can always find a local maximum $x^*$ and it is the global maximum within a sufficiently large neighborhood. That is, there exists a fixed number $r^* > 0$ such that the resulting $x^*$ is the global maximum in $B(x^*, r^*)$. 

86
Assumption 4.3.1 says that by gradient ascent we can always find a local maximum, and the distances between adjacent local maxima cannot be infinitely small.

Let $\mathcal{X}$ be the range of alternatives, which is a continuous space. For continuous KG without resampling, we have two sources of randomness: the measurement noise $W^n$ and the starting points in gradient ascent. For continuous KG with resampling, we have one more source of randomness: updating the parameter candidates set in the resampling process.

Note that the nonnegativity of the KG score with discrete priors still holds, as illustrated and proved in Chapter 3. We restate that result here:

**Proposition 4.3.2.** For $\forall n \geq 0$, $\forall x \in \mathcal{X}$, the KGDP-$f$ score $\nu^{KGDP-f,n}(x) \geq 0$. Equality holds if and only if (1) there exists $x'$ such that $x' \in \text{argmax}_{x_0} f_{r_i}(x_0)$ for all reduced model $f_{r_i}$ at $x$ with $p^n_{r_i} > 0$, or (2) all functions with $p^n > 0$ have the same value at $x$.

The definition of a reduced model at $x$ is also given in Chapter 3:

**Definition 4.3.3.** Suppose a subset of candidate functions intersect at a particular $x \in \mathcal{X}$, and without loss of generality, suppose they are $f(x; \theta_1) = \ldots = f(x; \theta_h)$. Then calculating $\nu^{KGDP-f,n}(x)$ using $(f_1, \ldots, f_n)$ with probabilities $(p^n_1, \ldots, p^n_L)$ is equivalent to calculating $\nu^{KGDP-f,n}(x)$ using $(f_{r}, f_{h+1}, \ldots, f_n)$ with probabilities $(p^n_r, p^n_{h+1}, \ldots, p^n_L)$, where $f_{r}(x') = \frac{\sum_{i=1}^{h} f_i(x') p^n_i}{\sum_{i=1}^{h} p^n_i}$ for all $x' \in \mathcal{X}$ and $p^n_r = \sum_{i=1}^{h} p^n_i$. We call $f_{r}$ the reduced model at $x$.

Furthermore, we call functions $f(x; \theta_i)$ and $f(x; \theta_j)$ are aligned if they achieve the maximum at the same $x \in \mathcal{X}$, i.e, $\text{argmax}_{x \in \mathcal{X}} f(x; \theta_i) = \text{argmax}_{x \in \mathcal{X}} f(x; \theta_j)$.

### 4.3.1 Convergence of Continuous KG without Resampling

We first prove that if we keep a fixed set of candidates $\mathbb{L}$, in which one of the candidates is exactly $\theta^*$, the continuous KG is able to find the optimal $x^* \in \mathcal{X}$ as our budget $N \to \infty$. Moreover, if we further assume the candidate functions are pairwise not aligned, then the continuous KG policy can also find the true $\theta^*$ almost surely as $N \to \infty$. 87
Intuitively, for any $f(x; \theta)$ ($\theta \neq \theta^*$), if we measure a sequence of points $\{x^n\}$ where the distances between $f(x^n; \theta)$ and $f(x^n; \theta^*)$ (i.e., $|f(x^n; \theta) - f(x^n; \theta^*)|$) are sufficiently large, we should see the probability of this $\theta$ go to zero as $n \to \infty$ and hence are able to conclude this $\theta$ is not the truth. If this happens for all but one $\theta$ in $\mathbb{L}$, we are done. Otherwise, for the subset of $\theta$’s whose probabilities do not go to zero, we argue that either (1) these functions are aligned, which means we have found the optimal $x^* \in X$, or (2) $|f(x^n; \theta) - f(x^n; \theta^*)|$ goes to zero as $n \to \infty$. In this latter case, all these $f(x^n; \theta)$’s are clustered in a small neighborhood of $f(x^n; \theta^*)$, and thus we learn almost nothing from measuring $x^n$. In other words, the KG score of $x^n$ goes to zero as $n \to \infty$. Then we argue that this happens only if there is always one $\theta$ whose probability is very close to 1, but this $\theta$ is not fixed. However, this is an event with probability 0 and thus we have contradiction.

We start by showing that if we measure a sequence of points such that $|f(x^n; \theta) - f(x^n; \theta^*)|$’s are sufficiently large, we are able to distinguish the correct parameter $\theta^*$ from the wrong ones:

**Proposition 4.3.4.** Let $x^n$ be the alternative we measure at time $n$. Let $\theta^*$ be the true $\theta$. For any $\theta_i, \theta_j \in \mathbb{L}$, let $r^n(\theta_i, \theta_j) = f(x^n; \theta_i) - f(x^n; \theta_j)$. For $\forall i \in [L]$, if $\sum_{n=0}^{\infty} [r^n(\theta_i, \theta^*)]^2 = \infty$, then

$$\lim_{n \to \infty} \frac{p^n(\theta_i)}{p^n(\theta^*)} = 0, \text{ almost surely.}$$

**Sketch of proof.** (See Appendix 4.6.2 for full proof.) We first show that $\frac{p^n(\theta_i)}{p^n(\theta^*)}$ is a Martingale process (regardless of where we measure). Then by the Martingale Convergence Theorem, there exists $X_\infty$ such that $\frac{p^n(\theta_i)}{p^n(\theta^*)} \to X_\infty$. Finally we show that if the sequence $\{x^0, x^1, \ldots, x^n, \ldots\}$ satisfies $\sum_{n=0}^{\infty} [r^n(\theta_i, \theta^*)]^2 = \infty$, then $X_\infty = 0$ almost surely. \hfill \Box

According to Proposition 4.3.4, we have the following corollary directly saying that if all $\theta$’s meet the property stated above, we know which one is $\theta^*$ almost surely (for simplicity, we denote $r^n(\theta_i, \theta^*)$ as $r^n_i$).
Corollary 4.3.5. If for every $\theta_i \in L$, $\sum_{n=0}^{\infty} (r_n^i)^2 = \infty$, we have $p^n(\theta^*) \to 1$ almost surely, i.e.,

$$\mathbb{P} \left( \lim_{n \to \infty} p^n(\theta^*) = 1 \right) = 1.$$

We also need the following two lemmas before proving the main theorem of the asymptotic optimality for the non-resampling case. Lemma 4.3.6 says that for any $x \in X$, the KG score $\nu^{KG-s,n}(x)$ has an upper bound that is proportional to the largest distance among $\{f(x; \theta_1), f(x; \theta_2), ..., f(x; \theta_L)\}$. Lemma 4.3.7 reveals another property of any sequence $\{r_n\}$ that satisfies $\sum_{n=0}^{\infty} r_n^2 < \infty$. Since both lemmas are pure calculus results, we skip the detailed proofs here. Interested readers could refer to Appendix 4.6.2.

Lemma 4.3.6. For any $x \in X$, let $d(x)$ denote the range of the $L$ function values at $x$, i.e, $d(x) = \max_{1 \leq i \leq L} f(x; \theta_i) - \min_{1 \leq i \leq L} f(x; \theta_i)$. We have

$$\nu^{KG-s,n}(x) \leq d(x) \cdot \frac{1}{\sqrt{2\pi}\sigma} \max_{x' \in X_s} \sum_{i=1}^{L} f(x'; \theta_i) p^n_i,$$

where $\frac{1}{\sqrt{2\pi}\sigma} \max_{x' \in X_s, 1 \leq i \leq L} f(x'; \theta_i)$ is a constant for a given $X_s$ and $\{\theta_i\}_{1 \leq i \leq L}$.

Lemma 4.3.7. For any positive real sequence $\{r_n\} \in \mathbb{R}_{>0}$, if $\sum_{n=0}^{\infty} r_n^2 < \infty$, then $\sum_{n=0}^{\infty} e^{\frac{c_1 r_n^2}{n}} < \infty$, where $c_1$ and $c_2$ are two constants with $c_1, c_2 \in \mathbb{R}_{>0}$.

Below is the main theorem of the asymptotic optimality for the non-resampling continuous KG algorithm.

Theorem 4.3.8. The non-resampling continuous KG algorithm with truth from prior is able to find the optimal $x \in X_s$ almost surely as we take an infinite number of measurements. Furthermore, if the candidate functions are pairwise not aligned, the non-resampling continuous KG algorithm with truth from prior is able to find both the optimal $x \in X$ and the true $\theta^*$. 89
Sketch of proof. (See Appendix 4.6.2 for full proof.) Assume the contrary is true, and let $\Omega_1$ be the set of event for the contrary to happen.

For any fixed $\omega \in \Omega_1$, we first show that $\lim_{n \to \infty} \nu^{KG-s,n}(x^n)(\omega) = 0$. For $\theta$’s whose probabilities go to 0, their contribution in $\nu^{KG-s,n}(x^n)$ goes to 0. For other $\theta$’s, since $\sum_{n=0}^{\infty}|r_n(\theta, \theta^*)|^2 < \infty$, $r_n(\theta, \theta^*)$ goes to 0 and hence the range function $d(x^n)$ goes to 0 as $n \to \infty$, where $d(x^n)$ is defined in Lemma 4.3.6, along with the fact that $\nu^{KG-s,n}(x^n)$ is upper-bounded by a constant times $d(x^n)$, according to Lemma 4.3.6.

Next, we argue that, based on Assumption 4.3.1 and the equality conditions in Proposition 4.3.2, $\lim_{n \to \infty} \nu^{KG-s,n}(x^n) = 0$ holds only if either (1) the functions that do not have probability 0 in the limit are aligned, or (2) the probability of one $\theta$ is very close to 1.

Finally, by discussing the two cases, we argue that in (1), the common optimal $x$ of the aligned functions is exactly $\text{argmax}_{x \in \mathcal{X}} f(x; \theta^*)$. In (2), the $\theta$ whose probability goes to 1 in the limit can only be $\theta^*$, and therefore, we are able to identify the true $\theta^*$ and the optimal $x^* \in \mathcal{X}$.

\[\Box\]

Remark 4.3.9. When some candidate functions are aligned, the non-resampling KG algorithm with truth from prior may not be able to find the true parameter $\theta^*$. Chapter 3 provides a counterexample in the discrete case.

4.3.2 Convergence of Continuous KG with Resampling

In this section, we prove that the resampling continuous KG policy is always able to find both the optimal $x^* \in \mathcal{X}$ and the true $\theta^*$ (assuming $\theta^* \in \mathcal{K}$) almost surely as $n \to \infty$, regardless of whether some candidate functions are aligned.

In the resampling case, Lemma 4.3.6 and Lemma 4.3.7 still hold without any change. Furthermore, if we think of the normalized likelihood of $\theta$’s in the entire set $\mathcal{K}$ as probabilities of $\theta$’s in $\mathcal{L}$, Proposition 4.3.4 also holds, with $\mathcal{L}$ replaced with $\mathcal{K}$ and probabilities $p_i$’s replaced...
with normalized likelihoods $L^n_i$, where $L^n_i$ is given by:

$$
L^n_i = \frac{\prod_{j=0}^{n-1} \exp \left[ -\frac{(\hat{y}^{i+1} - f(x^{i}, \theta^{i+1}))^2}{2\sigma^2}\right]}{\sum_{l=1}^{K} \prod_{j=0}^{n} \exp \left[ -\frac{(\hat{y}^{i+1} - f(x^{i}, \theta^{i+1}))^2}{2\sigma^2}\right]}.
$$

Based on this, we have the following proposition:

**Proposition 4.3.10.** Under the continuous KG policy, for $\forall \theta_i \neq \theta^* \ (1 \leq i \leq K)$, $\lim_{n \to \infty} L^n_i = 0$ almost surely, where $L^n_i$ is the normalized likelihood of $\theta_i$ at time $n$.

The proof of Proposition 4.3.10 is similar to that of Theorem 4.3.8, except that in the case when some candidate functions are aligned, we can only find the optimal $x^* \in \mathcal{X}$ in Theorem 4.3.8. Here we show that with the help of resampling, the true $\theta^*$ can also be identified together with the optimal $x^* \in \mathcal{X}$. Please refer to the detailed proof in Appendix 4.6.2.

However, unlike the non-resampling case, Proposition 4.3.10 cannot lead to our main theorem directly, in that we also need to show $\theta^*$ is always in $\mathbb{L}^n$ for $n$ large enough. We prove this is indeed true and thus have the final theorem:

**Theorem 4.3.11.** Under Assumption 4.3.1, the continuous KG policy with resampling is asymptotically optimal, i.e, it is able to find both the optimal $x^* \in \mathcal{X}$ and identify the true $\theta^*$ almost surely as we take an infinite number of measurements.

**Sketch of proof.** (See Appendix 4.6.2 for full proof.) Basically, we need to show that $\theta^*$ will be included in $\mathbb{L}^n$ for $n$ large enough almost surely. To prove this, we first show that $\theta^* \in \mathbb{L}^n$ infinitely often with probability 1. Then according to Proposition 4.3.10, since $\mathbb{L}^n(\theta^*) \to 1$, for $n$ large enough, once $\theta^*$ is included in $\mathbb{L}^n$, it would never get removed. Therefore, $\theta^* \in \mathbb{L}^n$ for all but a finite number of times almost surely, and thus $\hat{f}(x) \to f(x; \theta^*)$. \hfill $\square$
4.4 Empirical Results

In this section, we show the experimental results of applying our algorithm to two multidimensional benchmark functions, known as the additive newsvendor problem and the nested newsvendor problem, respectively. This problem class has been found to be particularly challenging since it encompasses a family of asymmetric, unimodular functions which can be varied from smooth to ones with difficult-to-approximate corners.

We mainly demonstrate results from the following aspects:

1. The number of restarts in gradient ascent we need to achieve satisfactory performance;
2. The comparison of continuous KG with other competing policies;
3. The empirical convergence rate of continuous KG under different dimensionality and noise settings;
4. Whether changing the setting of traditional resampling KGDP can achieve close performance to continuous KG.

In the following we define in detail a few terms used in our experiments.

- **Opportunity Cost (OC):** OC is defined as the difference between the value of the optimal $x$ and the value at the estimated optimal $x$ at time $n$; i.e.,

  $$OC(n) = \max_{x \in \mathcal{X}} f(x; \theta^*) - f(\arg\max_{x \in \mathcal{X}} \bar{f}^n(x); \theta^*),$$

  where $\bar{f}^n(x) = \sum_i p^n_i f(x; \theta_i)$. We normalize OC with the range of the true function, i.e.

  $$OC\%(n) = \frac{\max_{x \in \mathcal{X}} f(x; \theta^*) - f(\arg\max_{x \in \mathcal{X}} \bar{f}^n(x); \theta^*)}{\max_{x \in \mathcal{X}} f(x; \theta^*) - \min_{x \in \mathcal{X}} f(x; \theta^*)}.$$  

- **The Error of $\theta$:** Let $\hat{\theta}^n$ be our best estimate of $\theta^*$. We use the $L_2$ norm of the error, namely $||\hat{\theta}^n - \theta^*||_2$ as the metric for the quality of our estimate of $\theta^*$. 
• **The Noise Level**: This is a measurement of the noisiness of the experiments. Specifically, it is defined as the ratio of the standard deviation of the measurement noise to the range of the true function, i.e., \[ \frac{\sigma}{\max_x f(x;\theta^*) - \min_x f(x;\theta^*)} \]. When we say the noise is 20%, we mean this ratio is 20%.

We start this section by first discussing the number of restarts we use for the gradient ascent, and show that empirically a small number of restarts is enough to generate satisfactory results in terms of both finding the maximum of KG and minimizing OC. Then we compare our algorithm with two competing algorithms: (1) traditional KGDP with resampling (as introduced in Chapter 3), and (2) pure exploration, which picks an alternative from \( \mathcal{X} \) randomly. We note that pure exploration is a perfectly valid way for estimating parametric functions; in fact, we would argue that the vast majority of batch statistical learning is done with data that could be described as being collected at random. We evaluate these algorithms in terms of OC and the error of \( \theta \) in different dimensionality and noise settings. Experiments show that continuous KG demonstrates much stronger performance, especially in high dimensions or under high noise. We also explore the performances by changing the number of alternatives of KGDP and find that it does not help much in terms of catching up the performance of the continuous KG, which further demonstrates the advantage of the continuous KG algorithm.

### 4.4.1 The Number of Restarts of the Gradient Ascent

As presented in Section 4.2.3, in our continuous KG algorithm, we conduct gradient ascents to search for the alternative \( x \) that maximizes the KG score \( \nu_{KG-s,n}^*(x) \). Specifically, we start with 100 randomly sampled points, calculate their KG scores, and take the maximum as the starting point to run the gradient ascent algorithm. We repeat the above process for multiple times and select the largest point among all the obtained local maxima of KG. Below we refer to this algorithm as *start from 100*, and call the number of repeating times
the number of restarts. While the solution improves with more restarts, the marginal benefit from additional restarts declines quickly. We show experimentally that using *start from 100*, we obtain high quality solutions using as few as around 20 restarts for twenty-dimensional $\mathcal{X}$ problems.

We simulate a probability distribution of some additive newsvendor functions (see Section 4.4.2 for the formulation) and test the maximum KG our gradient ascent algorithm can find with different numbers of restarts. We repeat the simulation for 60 times; Figure 4.6 shows the average (of these simulations) of the maximum KG found compared with the average of the true maximum KG, for a ten-dimensional $\mathcal{X}$ problem and a twenty-dimensional $\mathcal{X}$ problem, respectively. As we can see from the graphs, 10 restarts under *start from 100* is sufficiently good in finding the maximum KG for a ten-dimensional problem, and 20-25 restarts is good for a twenty-dimensional problem. In each of our graphs, we also include a plot of the performance of the naive random start algorithm (denoted as *Naive Rdm Start* in Figure 4.6), which starts with one random point in each restart, instead of selecting one from 100 randomly sampled points. Note that increasing the number of potential starting points in each restart is much computationally easier than increasing the number of repeating times. This plot shows that starting with 100 points each time accelerates the algorithm and decreases the number of restarts needed for the gradient ascent algorithm.

![Figure 4.6: Maximum KG found vs the number of restarts.](image)

94
Besides comparing the effect of different numbers of restarts on the maximum KG obtained, we also compare how different numbers of restarts affect the final result of our continuous KG algorithm, measured using the opportunity cost and the error of the parameter $\theta$. Figure 4.7 shows the results with different numbers of restarts of our continuous KG algorithm, as well as a comparison with KGDP. This comparison is run on a nested newsvendor problem (see Section 4.4.3 for the formulation). We plot results with 5, 10 and 20 restarts in a ten-dimensional alternative space $X$ under 20% and 50% noise, respectively. Consistent with Figure 4.6, the results of 5 restarts here are in general worse than those with 10 or 20 restarts, but there is not much difference between 10 and 20. This implies two points. First, it confirms that the more restarts there are, the more likely it is to arrive at a global maximum of KG, and thus the better performance we can achieve. So in applications where we have strict requirements on the performance, we can achieve so by increasing the number of restarts. Second, the KG surface of problems with parametric belief models are

![Graphs showing OC% and Error of best $\theta$ (L2 norm) for different numbers of restarts and noise levels.]

Figure 4.7: Comparison of different number of restarts in terms of opportunity cost (quality of the solution $\hat{x}^*$) and statistical error (quality of estimation $\hat{\theta}^*$) (1st row: 20% noise, 2nd row: 50% noise).
much smoother than that with look-up table belief models, such that empirically, with 10 to 20 restarts, we are able to achieve sufficiently satisfactory results.

4.4.2 Additive Newsvendor Problem

The standard one-dimensional newsvendor function is given by

\[ f(x) = \eta_1 E[\min(D, x)] - \eta_2 x, \]

where \( x \) is the supply, \( \eta_1 \) is the price, \( \eta_2 \) is the cost, and \( D \) is the demand which follows a uniform distribution. The additive newsvendor function is an extension of the one-dimensional function in that it is the summation of several one-dimensional standard newsvendor functions, given by

\[ f(x_1, \ldots, x_k) = \sum_{j=1}^{k} \{ \eta_{1,j} E[\min(D_j, x_j)] - \eta_{2,j} x_j \}, \]

where \( x_1, \ldots, x_k \) is the \( k \)-dimensional decision vector and \( D_j \)'s are uniformly distributed with fixed variance. In our experimental settings, we let \( \eta_{1,j} \)'s be constants, while \( \eta_{2,j} \)'s as well as \( E[D_j] \)'s are unknown parameters. Therefore, we have the parameter \( \theta = (\eta_{2,1}, \ldots, \eta_{2,k}, E[D_1], \ldots, E[D_k]) \). For an additive newsvendor problem with a \( k \)-dimensional alternative space \( \mathcal{X} \), its parameter \( \theta \) has a dimensionality of \( 2k \).

The comparisons of the three algorithms (continuous KG with resampling, traditional KGDP, and pure exploration) in ten- and twenty-dimensional \( \mathcal{X} \) spaces (under 20\% noise) are shown in Figure 4.8. The \( x \)-axis is the number of measurements we take. In these experiments, continuous KG with resampling (denoted as “KGDP-C” in the graphs) uses three restarts for the gradient ascent, \( L = 20 \) as the size of the parameter candidate set \( \mathbb{L} \), and \( K = 100,000 \) and 1,000,000 as the sizes of the large parameter candidate pool \( \mathbb{K} \) for the ten- and twenty-dimensional problems, respectively. For KGDP, we construct the alternative set \( \mathcal{X} \)'s using 2500 random samples. As we can see from the graphs, continuous
Figure 4.8: Comparison of policies for ten-dimensional (1st row) and twenty-dimensional (2nd row) $\mathcal{X}$, in terms of opportunity cost (quality of the solution $\hat{x}^*$) and statistical error (quality of estimation $\hat{\theta}^*$).

KG outperforms the other two policies in terms of both OC and the error of $\theta$, especially in higher dimensions.

4.4.3 Nested Newsvendor Problem

The $k$-dimensional nested newsvendor function is given by

$$f(x_1, \ldots, x_k) = \sum_{i=1}^{k} \eta_{1,i} \mathbb{E} \left[ \min \left( x_i, (D - \sum_{j=1}^{i-1} x_j)^+ \right) \right] - \sum_{i=1}^{k} \eta_{2,i} x_i,$$

where $x_1, \ldots, x_k$ are the decision vector, $D$ is a uniformly distributed random variable. In the following experiments, we fix the variance of $D$ and all $\eta_{1,i}$’s, but let the mean of $D$ as well as $\eta_{2,i}$’s be unknown parameters. In other words, we have the parameter $\theta = (E[D], \eta_{2,1}, \ldots, \eta_{2,k})$, and therefore, a nested newsvendor problem with a $k$-dimensional alternative set $\mathcal{X}$ has
a \((k + 1)\)-dimensional parameter \(\theta\). The nested newsvendor problem creates a family of problems with more complex interactions in the \(X\)-space.

Figure 4.9 shows the comparison of the three algorithms in a ten-dimensional \(X\) problem under 20% and 50% noise, respectively. The \(x\)-axis is the number of measurements we take. For continuous KG with resampling (denoted as “KGDP-C” in the graphs), we use 20 restarts for the gradient ascent, \(L = 20\) as the size of the parameter candidate set \(\mathbb{L}\), and \(K = 100,000\) as the size of the large parameter candidate pool \(\mathbb{K}\). For KGDP, we construct the alternative set \(X_s\) using 1000 random samples. As shown in the graphs, continuous KG outperforms traditional KGDP and pure exploration in terms of both finding the optimal \(x^*\) and learning \(\theta^*\). In high noise settings, KGDP and pure exploration are very close in finding \(\theta^*\), while continuous KG performs much better.

Figure 4.9: Comparison of policies for ten-dimensional \(X\) in terms of opportunity cost (quality of the solution \(\hat{x}^*\)) and statistical error (quality of estimation \(\hat{\theta}^*\)) (1st row: 20% noise, 2nd row: 50% noise).
We have similar results for three- or six-dimensional problems. We list results of the three algorithms after 20 measurements in different dimensionality settings in Table 4.1. As we can see, continuous KG is consistently better than the other two policies.

Table 4.1: Comparison of different policies when $n = 20$

<table>
<thead>
<tr>
<th></th>
<th>Noise</th>
<th>Pure Exploration</th>
<th>SGDP</th>
<th>KGDP</th>
<th>KGDP-C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>OC%</td>
<td>$L_2$</td>
<td>OC%</td>
<td>$L_2$</td>
</tr>
<tr>
<td>3-dimensional $\mathcal{X}$</td>
<td>20%</td>
<td>0.0687</td>
<td>0.2472</td>
<td>0.0185</td>
<td>0.1727</td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>0.2051</td>
<td>0.4985</td>
<td>0.1298</td>
<td>0.4407</td>
</tr>
<tr>
<td>6-dimensional $\mathcal{X}$</td>
<td>20%</td>
<td>0.1652</td>
<td>0.6683</td>
<td>0.0722</td>
<td>0.6257</td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>0.2080</td>
<td>0.9391</td>
<td>0.1462</td>
<td>0.8755</td>
</tr>
<tr>
<td>10-dimensional $\mathcal{X}$</td>
<td>20%</td>
<td>0.1950</td>
<td>1.0399</td>
<td>0.1264</td>
<td>0.9548</td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>0.2953</td>
<td>1.2779</td>
<td>0.2202</td>
<td>1.2825</td>
</tr>
</tbody>
</table>

As we know, the performance of traditional KGDP improves with the number of alternatives. Theoretically, if we run the traditional KGDP algorithm with a very large number of alternatives, regardless of the computational complexity, the traditional KGDP’s performance should be close to that of the continuous KG algorithm. We test how much the traditional KGDP’s performance can catch up with that of the continuous KG as the number of alternatives increases. In Figure 4.10, we plot the performance of continuous KG along with that of the traditional KGDP with 500, 1000 and 2000 alternatives, respectively, in a ten-dimensional $\mathcal{X}$ with 20% noise. The $x$-axis is the number of measurements we take. The performance of KGDP does not improve much as we increase the number of alternatives from 500 to 2000. This makes sense because even with 2000 alternatives, the samples in the ten-dimensional space are still sparse. In other words, even as the number of alternatives in KGDP increases, our experiments show that the performance of KGDP is still not catching up with that of the continuous KG algorithm. Meanwhile, the complexity of traditional KGDP increases (at least) quadratically as the number of alternatives increases, while continuous KG does not have the limitation of handling just a finite number of alternatives and the number of function evaluations in a single gradient ascent is empirically relatively stable.
Figure 4.10: Comparison of continuous KG and KGDP with different number of alternatives in terms of opportunity cost (quality of the solution $\hat{x}^*$) and statistical error (quality of estimation $\hat{\theta}^*$) (1st row: 20% noise, 2nd row: 50% noise).

This further demonstrates the significant advantage of continuous KG, in terms of not only the outcomes but also the computational complexity, especially in high dimensional settings.

### 4.5 Conclusion

Optimal learning over a finite set of alternatives has been widely studied, while the extension to a continuous alternative space makes the problem significantly more difficult due to the computational intractability of the knowledge gradient. Scott et al. (2011) makes a first attempt to deal with the continuous alternative space. It focuses on problems with a look-up table belief model, but is limited to low (three or four) dimensional problems. Furthermore, the Knowledge Gradient with Discrete Priors (KGDP) model (Chen et al. 2014) proposes an approach to handle optimal learning problems with nonlinear parametric belief models, and the KG model with resampling (introduced in Chapter 3) lifts a major limitation of the KGDP model in which one of the few sampled candidates has to be exactly the
truth. However, both models can only handle the optimal learning problem on a finite set of alternatives.

In this chapter, we propose an algorithm that solves optimal learning problems with nonlinear parametric belief models over a multidimensional continuous space of alternatives. We establish a new method of calculating the KG score and present an algorithm to calculate the gradient of the KG score approximation. The former improves the accuracy of the KG score approximation, while the latter helps search for the most effective alternative to measure next. Our algorithm is built on a similar framework as the KG with resampling algorithm in terms of updating the probability of each parameter candidate and resampling the set of candidates from a large pool, but our algorithm uses newly proposed methods to approximate the KG score and its derivative so as to determine the best alternative to measure, out of a continuous alternative space.

Our algorithm is able to effectively find the true parameter and the optimal alternative over a continuous alternative space within a limited budget of noisy experiments. Experiments on two types of newsvendor problems, one with a nested function setting and the other with an additive function setting, show that our algorithm has a strong performance and the extension of handling continuous alternatives introduces a significant performance improvement to the original KGDP model, especially in high dimensions. We also prove that our algorithm is asymptotically optimal. In other words, as the number of measurements goes to infinity, our algorithm is capable of finding both the true parameter and the optimal alternative.
4.6 Appendix

4.6.1 A Heuristic to Find \( \{S_k\} \) and \( \{x^{(k)}\} \)

By definition, \( \{S_k\} \) is (the \( \hat{y} \) coordinates of) a subset of the intersections of \( h(x_m, \hat{y}) \), for \( m \in [M] \), as can be seen from Equation (4.8). Since \( h(x_m, \hat{y}) \) is a scaled mixture of Gaussian distributions, typically there does not exist analytical solutions to \( h(x_i, \hat{y}) = h(x_j, \hat{y}) \) (a special exception is the two-curve model in Chapter 3). Therefore, we choose to use a heuristic algorithm to find \( \{S_k\} \).

We do this by discretizing the \( \hat{y} \) axis. First, note that for any fixed \( \hat{y} \), we can find the \( x' \in X_s \) that maximizes the \( h \) function by evaluating all \( h(x_m, \hat{y}) \)'s explicitly, \( 1 \leq m \leq M \). In particular, for two special points, \( \hat{y} = S_0 = -\infty \) and \( \hat{y} = S_K = \infty \), we can get \( x = \text{argmax}_{x' \in X_s} h(x', \hat{y}) \) without calculating the \( h \) function; moreover, by (4.7), the corresponding \( x \)'s to \( S_0 \) and \( S_K \) are \( x^{(1)} \) and \( x^{(K)} \), respectively.

We fix an \( x \in X \) (instead of \( X_s \)), and consider calculating \( L^{KG-s,n}(x) \). For any \( x_m \in X_s \), \( h(x_m, \hat{y}) \) is a mixture of scaled Gaussian distributions centered at \( f(x, \theta_1), ..., f(x, \theta_L) \), respectively. Without loss of generality, we assume \( f(x; \theta_i) \neq f(x; \theta_j) \) for any \( 1 \leq i \neq j \leq L \). Let \( i_1 = \text{argmin}_{i \in [L]} f(x, \theta_i) \) and \( i_2 = \text{argmax}_{i \in [L]} f(x, \theta_i) \). We claim that \( x^{(1)} = \text{argmax}_{x' \in X_s} f(x', \theta_{i_1}) \) and \( x^{(K)} = \text{argmax}_{x' \in X_s} f(x', \theta_{i_2}) \), where \( x^{(1)} \) and \( x^{(K)} \) are defined in Equation (4.7). We only give the proof of \( x^{(1)} = \text{argmax}_{x' \in X_s} f(x', \theta_{i_1}) \) below, since the proof for \( x^{(K)} \) is similar.

Let \( \bar{x} = \text{argmax}_{x' \in X_s} f(x', \theta_{i_1}) \). Without loss of generality, we assume \( i_1 = 1 \). For any \( x' \in X_s \), \( x' \neq \bar{x} \), we have
\[
\frac{h(x', \hat{y})}{h(x, \hat{y})} = \lim_{\hat{y} \to -\infty} \frac{\sum_{i=1}^L f(x', \theta_i)p_i^n \exp \left[ -\frac{(\hat{y} - f(x, \theta_i))^2}{2\sigma^2} \right]}{\sum_{i=1}^L f(\bar{x}, \theta_i)p_i^n \exp \left[ -\frac{(\hat{y} - f(x, \theta_i))^2}{2\sigma^2} \right]}
\]

By assumption, \(i_1 = \argmin_{i \in [L]} f(x, \theta_i) = 1\), and thus, \(f(x, \theta_i) - f(x, \theta_1) > 0\) for any \(i \neq 1\). Hence, the second terms in both the numerator and the denominator go to 0 as \(\hat{y} \to -\infty\), so

\[
\lim_{\hat{y} \to -\infty} \frac{h(x', \hat{y})}{h(x, \hat{y})} = \frac{f(x', \theta_1)p_1^n}{f(\bar{x}, \theta_1)p_1^n} = \frac{f(x', \theta_1)}{f(\bar{x}, \theta_1)}.
\]

Since \(\bar{x} = \arg\max_{x' \in \mathcal{X}_s} f(x', \theta_{i_1})\) and \(i_1 = 1\), we have \(h(\bar{x}, \hat{y}) \geq h(x', \hat{y})\) for any \(x' \in \mathcal{X}_s\), i.e., \(\bar{x} = \arg\max_{x \in \mathcal{X}_s} h(x, \hat{y})\), as \(\hat{y} \to -\infty\). Thus by the definition of \(x^{(0)}\), we have

\[
x^{(0)} = \lim_{\hat{y} \to -\infty} \arg\max_{x \in \mathcal{X}_s} h(x, \hat{y}) = \bar{x} = \arg\max_{x' \in \mathcal{X}_s} f(x', \theta_{i_1}).
\]

Similarly, we have \(x^{(K)} = \arg\max_{x' \in \mathcal{X}_s} f(x', \theta_{i_2})\).

Therefore, to figure out the set \(\{x^{(k)}\}\), it suffices to find \(x^{(2)}\) through \(x^{(K-1)}\), while for the set \(\{S_k\}\), we need to find \(S_1\) through \(S_{K-1}\). We can do this by discretizing a large interval of the \(\hat{y}\) axis such that \((S_1, S_{K-1})\) is a subinterval of it. For normal distribution with mean \(\mu\) and standard deviation \(\sigma\), we only need to care about say the range \((\mu - 5\sigma, \mu + 5\sigma)\) in practice, which provides an error within \(\sim 10^{-6}\).

Note that \(\{x^{(k)}\}\) can be calculated once we have \(\{S_k\}\). To determine \(\{S_k\}\), below is a basic algorithm to start with. We first discretize \((f(x, \theta_{i_1}) - 5\sigma, f(x, \theta_{i_2}) + 5\sigma)\) into small intervals (for example, into intervals of size \(\sigma/20\)). Then the error of \(S_k\) is at most \(\sigma/20\) and based on Equation (4.13), the error of \(\nu^{KG-s,n}(x)\) is at most \(\phi(1/40) - \phi(-1/40) \approx 2\%\).
Suppose we get $S$ points $\{\hat{y}_s\} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_S\}$ after discretization. For any $\hat{y}_s$, we calculate $h(x_m, \hat{y}_s)$ for all $x_m$ and find $x(s) = \arg\max_{x_m \in \mathcal{X}} h(x_m, \hat{y}_s)$. For any two adjacent $y_s$ and $y_{s+1}$, if they result in two different alternatives, i.e., $x(s) \neq x(s+1)$, we know there must exist a separating point $S_k \in (x(s), x(s+1))$. We can further use the bisection search method to achieve higher accuracy. Note that empirically this is not a complicated algorithm, for the following two reasons. First, the range $(f(x, \theta_{i_1}), f(x, \theta_{i_2}))$ is empirically not large. This is because if $|f(x, \theta_{i_1}) - f(x, \theta_{i_2})|$ is large, after one measurement at $x$ we would immediately be able to rule out some candidates, thus narrowing down $|f(x, \theta_{i_1}) - f(x, \theta_{i_2})|$. Hence, using the resampling algorithm discussed in Chapter 3, after a few iterations, we can usually expect a small $(f(x, \theta_{i_1}), f(x, \theta_{i_2}))$ for most $x$’s. Second, we can start with a relatively coarse discretization of the interval, and let the bisection search method to refine it.

We can save more run time if we design the algorithm more carefully. First, we can drop the functions with very low probabilities, since they contribute little to the KG score (this can be seen via Equation (4.16)). Second, for each remaining candidate $\theta_{i'}$, we can focus on the interval $(f(x, \theta_{i'}) - 5\sigma, f(x, \theta_{i'}) + 5\sigma)$ and calculate their union, which is our region of interest. Compared with the basic algorithm stated above, we have the finalized algorithm as shown in Algorithm 4.

### 4.6.2 Proof of Convergence of Continuous KG

**Proposition 4.3.4.** Let $x^n$ be the alternative we measure at time $n$. Let $\theta^*$ be the true $\theta$.

For any $\theta_i, \theta_j \in \mathbb{L}$, let $r^n(\theta_i, \theta_j) = f(x^n; \theta_i) - f(x^n; \theta_j)$. For $\forall i \in [L]$, if $\sum_{n=0}^{\infty} [r^n(\theta_i, \theta^*)]^2 = \infty$, then

$$\lim_{n \to \infty} \frac{p^n(\theta_i)}{p^n(\theta^*)} = 0, \text{ almost surely.}$$
Algorithm 4 A heuristic algorithm to calculate \( \{S_k\} \) and \( \{x^{(k)}\} \)

**Input:** Alternative: \( x \); Candidates: \( \{\theta_i\} = \{\theta_1, \ldots, \theta_L\} \); Probabilities: \( \{p_i^n\} = \{p_1^n, \ldots, p_L^n\} \); Noise: \( \sigma \); Tolerance for error: \( \epsilon \); Step size for discretization: \( \text{Step} \) (set as \( \sigma/5 \) by default).

**Output:** Two sets: \( \{S_k\} \) and \( \{x^{(k)}\} \).

1. Region of interest: \( A = \emptyset \).
2. for \( i = 1 \) to \( L \) do
3. if \( p_i^n < \epsilon \) then
4. \( \{\theta_i\} = \{\theta_i\} \setminus \{\theta_i\}, \{p_i^n\} = \{p_i^n\} \setminus \{p_i^n\} \).
5. else
6. \( A = A \cup \{f_i(x) - 5\sigma, f_i(x) + 5\sigma\} \).
7. end if
8. end for
9. Discretize \( A \): \( A_D = \{\inf A : \text{Step} : \sup A\} \cap A \).
10. \( x_{vec} = \emptyset \) (length: \( |A_D| \)); \( S_I = \emptyset \).
11. for \( i = 1 \) to \( |A_D| \) do
12. \( x_{vec}(i) = \arg\max_{x \in X} h(x, A_D(i)) \).
13. if \( i \geq 2 \) and \( x_{vec}(i) \neq x_{vec}(i - 1) \) then
14. \( S_I = S_I \cup \{(A_D(i - 1), A_D(i))\} \) (note that \( S_I \) is a set of intervals).
15. end if
16. end for
17. for \( k = 1 \) to \( |S_I| \) do
18. Bisection search, function: \( f(y) = h(x_{vec}(i - 1), y) - h(x_{vec}(i), y) \), left point: \( S_I(k)(1) \), right point: \( S_I(k)(2) \), Tolerance for function error: \( \epsilon \). Output: \( y = S_k \).
19. end for
20. \( x^{(1)} = \arg\max_{x \in X} h(x, S_1 - \sigma) \), \( x^{(K)} = \arg\max_{x \in X} h(x, S_K + \sigma) \).
21. for \( k = 2 \) to \( K - 1 \) do
22. \( x^{(k)} = \arg\max_{x \in X} h(x, S_{k-1} + S_k \) / 2 \).
23. end for
24. return \( \{S_1, ..., S_{K-1}\}, \{x^{(1)}, ..., x^{(K)}\} \).

**Proof of Proposition 4.3.4**

For simplicity, let us denote \( r_i^n(\theta_i, \theta^*) \) as \( r_i^n \). According to the updating formula of \( p^n \), we have

\[
\begin{align*}
p^n(\theta_i) &= \frac{\prod_{m=1}^{n} \exp\left(-\frac{|\theta_i - f_i(x^{m-1})|^2}{2\sigma^2}\right)}{\prod_{m=1}^{n} \exp\left(-\frac{|\theta^* - f(x^{m-1})|^2}{2\sigma^2}\right)} \\
&= \prod_{m=1}^{n} \frac{\exp\left(-\frac{f_i(x^{m-1})^2 + 2\sigma f_i(x^{m-1}) + \sigma Z^m}{2\sigma^2}\right)}{\exp\left(-\frac{(Z^m)^2}{2}\right)} \\
&= \prod_{m=1}^{n} \exp\left(-\frac{(r_i^{m-1})^2 + 2\sigma r_i^{m-1} Z^m}{2\sigma^2}\right),
\end{align*}
\]
where $Z^m$ is a random variable with the standard normal distribution, $1 \leq m \leq n$.

Therefore,

\[
\mathbb{E} \left( \frac{p^{n+1}(\theta_i)}{p^n(\theta^*)} \big| \mathcal{F}^n \right) = \mathbb{E} \left( \frac{p^n(\theta_i)}{p^n(\theta^*)} \right) \cdot \int_{-\infty}^{\infty} \exp \left( \frac{-(r_i^n)^2 + 2\sigma r_i^n Z^{n+1}}{2\sigma^2} \right) \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{(Z^{n+1})^2}{2} \right) dZ^{n+1}
\]

\[
= \mathbb{E} \left( \frac{p^n(\theta_i)}{p^n(\theta^*)} \right) \cdot \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{(Z^{n+1})^2}{2} \right) dZ^{n+1}
\]

Moreover, since the $Z^m$'s are independent,

\[
\mathbb{E} \left[ \frac{p^n(\theta_i)}{p^n(\theta^*)} \right] = \prod_{m=1}^{n} \mathbb{E} \left[ \exp \left( -\frac{(r_i^{m-1})^2 + 2\sigma r_i^{m-1} Z^m}{2\sigma^2} \right) \right] = 1.
\]

Hence, $\left( \frac{p^n(\theta_i)}{p^n(\theta^*)} \right)_{n \geq 1}$ is bounded in $L^1$. So $\left( \frac{p^n(\theta_i)}{p^n(\theta^*)} \right)_{n \geq 0}$ is a Martingale process. According to the Martingale Convergence Theorem, there exists $X_\infty \in L^1$, such that $\frac{p^n(\theta_i)}{p^n(\theta^*)} \to X_\infty$ almost surely.

We claim $X_\infty = 0$. In fact, note that $\log \frac{p^n(\theta_i)}{p^n(\theta^*)} = -\frac{\sum_{m=1}^{n} (r_i^{m-1})^2 + 2\sigma \sum_{m=1}^{n} r_i^{m-1} Z^m}{2\sigma^2}$. For any $n$, $\log \frac{p^n(\theta_i)}{p^n(\theta^*)}$ is Gaussian with mean $-\frac{1}{2\sigma^2} \sum_{m=0}^{n-1} (r_i^{m})^2$ and variance $\frac{1}{\sigma^2} \sum_{m=0}^{n-1} (r_i^{m})^2$. Since $\sum_{m=0}^{\infty} (r_i^{m})^2 = \infty$, for any $\epsilon \in \mathbb{R}_{>0}$, $\mathbb{P}(\log \frac{p^n(\theta_i)}{p^n(\theta^*)} < \epsilon) \to 0$ as $n \to \infty$. So $\mathbb{P}(\log X_\infty = -\infty) = 1$, and thus $X_\infty = 0$ almost surely.

\[\square\]

**Lemma 4.3.6.** For any $x \in \mathcal{X}$, let $d(x)$ denote the range of the $L$ function values at $x$; i.e., $d(x) = \max_{1 \leq i \leq L} f(x; \theta_i) - \min_{1 \leq i \leq L} f(x; \theta_i)$. We have

\[
\nu^{KG-s,n}(x) \leq d(x) \cdot \frac{1}{\sqrt{2\pi}\sigma} \max_{x' \in \mathcal{X}} \sum_{i=1}^{L} f(x'; \theta_i) p^i_n,
\]

where $\frac{1}{\sqrt{2\pi}\sigma} \max_{x' \in \mathcal{X}, 1 \leq i \leq L} f(x'; \theta_i)$ is a constant for a given $\mathcal{X}$ and $\{\theta_i\}_{1 \leq i \leq L}$.
Proof of Lemma 4.3.6.

For a given $\mathcal{X}$ and $\{\theta_i\}_{1 \leq i \leq L}$, since $\mathcal{X}$ is a finite set, we can assume without loss of generality that $f(x'; \theta_i) > 0$ for $\forall x' \in \mathcal{X}$ and $1 \leq i \leq L$. Let $f_{\max} = \max_{1 \leq i \leq L} f(x; \theta_i)$ and $f_{\min} = \min_{1 \leq i \leq L} f(x; \theta_i)$. By Equation (4.5), we have:

$$\nu^{KG-s,n}(x) = \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{\infty} \left\{ \max_{x' \in \mathcal{X}} \sum_{i=1}^{L} f(x'; \theta_i) p_i^n \exp \left[ -\frac{(\hat{y} - f(x; \theta_i))^2}{2\sigma^2} \right] \right\} d\hat{y} - \max_{x' \in \mathcal{X}} \sum_{i=1}^{L} f(x'; \theta_i) p_i^n \exp \left[ -\frac{(\hat{y})^2}{2\sigma^2} \right] d\hat{y}$$

$- \max_{x' \in \mathcal{X}} \sum_{i=1}^{L} f(x'; \theta_i) p_i^n.$

For $\hat{y} \in (-\infty, f_{\min})$, $\exp \left[ -\frac{(\hat{y} - f(x; \theta_i))^2}{2\sigma^2} \right] \leq \exp \left[ -\frac{(\hat{y} - f_{\min})^2}{2\sigma^2} \right]$ for any $i$. Similarly, for $\hat{y} \in (f_{\max}, \infty)$, $\exp \left[ -\frac{(\hat{y} - f(x; \theta_i))^2}{2\sigma^2} \right] \leq \exp \left[ -\frac{(\hat{y} - f_{\max})^2}{2\sigma^2} \right]$ for any $i$. For $\hat{y} \in (-f_{\min}, f_{\max})$, $\exp \left[ -\frac{(\hat{y} - f(x; \theta_i))^2}{2\sigma^2} \right] \leq 1$. Therefore,

$$\nu^{KG-s,n}(x) \leq \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{f_{\min}} \left\{ \max_{x' \in \mathcal{X}} \sum_{i=1}^{L} f(x'; \theta_i) p_i^n \exp \left[ -\frac{(\hat{y} - f_{\min})^2}{2\sigma^2} \right] \right\} d\hat{y}$$

$+ \frac{1}{\sqrt{2\pi\sigma}} \int_{f_{\min}}^{f_{\max}} \left\{ \max_{x' \in \mathcal{X}} \sum_{i=1}^{L} f(x'; \theta_i) p_i^n \right\} d\hat{y}$

$+ \frac{1}{\sqrt{2\pi\sigma}} \int_{f_{\max}}^{\infty} \left\{ \max_{x' \in \mathcal{X}} \sum_{i=1}^{L} f(x'; \theta_i) p_i^n \exp \left[ -\frac{(\hat{y} - f_{\max})^2}{2\sigma^2} \right] \right\} d\hat{y} - \max_{x' \in \mathcal{X}} \sum_{i=1}^{L} f(x'; \theta_i) p_i^n$

$$= \left( \max_{x' \in \mathcal{X}} \sum_{i=1}^{L} f(x'; \theta_i) p_i^n \right) \cdot \left\{ \frac{1}{\sqrt{2\pi\sigma}} \left( \int_{-\infty}^{0} + \int_{0}^{\infty} \right) \exp \left( -\frac{\hat{y}^2}{2\sigma^2} \right) d\hat{y} + d(x) \cdot \frac{1}{\sqrt{2\pi\sigma}} - 1 \right\}$$

$= d(x) \cdot \frac{1}{\sqrt{2\pi\sigma}} \max_{x' \in \mathcal{X}} \sum_{i=1}^{L} f(x'; \theta_i) p_i^n.$

From the proof we can see that the first equality holds only when $d(x) = 0$, which means the $f(x; \theta_i)$’s have the same values.

$\square$
Lemma 4.3.7. For any positive real sequence \( \{r_n\} \in \mathbb{R}_{>0} \), if \( \sum_{n=0}^{\infty} r_n^2 < \infty \), then \( \sum_{n=0}^{\infty} e^{\frac{c_1}{r_n} - \frac{c_2}{r_n}} < \infty \), where \( c_1 \) and \( c_2 \) are two constants with \( c_1, c_2 \in \mathbb{R}_{>0} \).

Proof of Lemma 4.3.7

As \( \sum_{n=0}^{\infty} r_n^2 < \infty \) and \( r_n \in \mathbb{R}_{>0} \), \( \lim_{n \to \infty} r_n = 0 \). Thus there exists \( N_1 > 0 \) such that for \( \forall n > N_1, r_n < c_1 \). Therefore,

\[
\sum_{n=0}^{\infty} e^{\frac{c_1}{r_n} - \frac{c_2}{r_n}} = \sum_{n=0}^{N_1} e^{\frac{c_1}{r_n} - \frac{c_2}{r_n}} + \sum_{n>N_1} e^{\frac{c_1}{r_n} - \frac{c_2}{r_n}} < \sum_{n=0}^{N_1} e^{\frac{c_1}{r_n} - \frac{c_2}{r_n}} + \sum_{n>N_1} e^{-\frac{c_2}{r_n}}.
\]

Thus it suffices to show that \( \sum_{n>N_1} e^{-\frac{c_2}{r_n}} < \infty \). We will show that there exists \( N_2 > 0 \) such that for \( \forall n > N_2, e^{-\frac{c_2}{r_n}} < r_n^2 \). Since \( \sum_{n>N_2} r_n^2 < \infty \), by the comparison test, we have \( \sum_{n>N_2} e^{-\frac{c_2}{r_n}} < \infty \).

Note that \( \lim_{r \to 0} r \log \frac{1}{r} = 0 \). Additionally, since \( (r \log \frac{1}{r})' = \log \frac{1}{r} - 1 > 0 \) for \( r < \frac{1}{e} \), \( r \log \frac{1}{r} \) is an increasing function on \( r \in (0, \frac{1}{e}) \). Thus there exists \( N_2 > 0 \), such that for \( \forall n > N_2, r_n \log \frac{1}{r_n} < \frac{c_2}{2} \). That is, \( e^{-\frac{c_2}{r_n}} < r_n^2 \).

\( \square \)

Theorem 4.3.8. The non-resampling continuous KG algorithm with truth from prior is able to find the optimal \( x \in \mathcal{X} \) almost surely as we take an infinite number of measurements. Furthermore, if the candidate functions are pairwise not aligned, the non-resampling continuous KG algorithm with truth from prior is able to find both the optimal \( x \in \mathcal{X} \) and the true \( \theta^* \).

Proof of Theorem 4.3.8

We prove the theorem by contradiction. Assume the contrary is true, and let \( \Omega_1 \) be the set of events that the algorithm cannot find the optimal \( x \in \mathcal{X} \).

Fix an \( \omega \in \Omega_1 \). Since we cannot learn the optimal \( x \in \mathcal{X} \) for this \( \omega \), there exists at least one \( \theta \neq \theta^* \) such that \( \theta \)'s probability does not go to 0. Without loss of generality, assume
$p^n_i$ does not go to zero for $1 \leq i \leq k$, where $k \geq 1$, and the truth is $\theta^* = \theta_L$, while for $k + 1 \leq i \leq L - 1$, we have $\lim_{n \to \infty} p^n_i = 0$.

We claim $\lim_{n \to \infty} \nu^{KG-s,n}(x^n) = 0$. Fix some $\epsilon > 0$. On the one hand, for $\theta_{k+1}, \ldots, \theta_{L-1}$, their probabilities go to 0. We can thus discard them in KG with an error of at most $\epsilon$ for $n$ large enough, as a result of Equation (4.1). On the other hand, for $\theta_1, \ldots, \theta_k$ and $\theta^*$, since $p^n_i$ does not go to zero, by Corollary 4.3.5, $\sum_{n=0}^{\infty} (r^n_i)^2 < \infty$ for $1 \leq i \leq k$. Thus we have $\lim_{n \to \infty} \max_{1 \leq i \leq k} \left[ f(x^n; \theta_i) - \min_{1 \leq i \leq k} f(x^n; \theta_i) \right] = 0$. By a similar logic in the proof of Lemma 4.3.6, the KG score is at most $\epsilon$ for $n$ large enough. Therefore, for $\forall \epsilon > 0$, $\nu^{KG-s,n}(x^n) < \epsilon$ for $n$ large enough. Thus $\lim_{n \to \infty} \nu^{KG-s,n}(x^n) = 0$.

Note that as $\lim_{n \to 0} r^n_i = 0$ for $1 \leq i \leq k$, all $f(x; \theta_i)$’s must intercept at the same point. Without loss of generality, let us assume that there is a unique interception $x_0$. Then there must exist some $N > 0$, such that $x^n \in B(x_0, r^*/2)$ for all $n > N$, where $r^*$ is the fixed real number in Assumption 4.3.1.

Let $x_1$ be a fixed point in $B(x_0, r^*/2)$ and $x_1 \neq x_0$. Then by Assumption 4.3.1, $\nu^{KG-s,n}(x_1) \leq \nu^{KG-s,n}(x^n)$ for all $n > N$ and thus $\lim_{n \to \infty} \nu^{KG-s,n}(x_1) = 0$. Hence one of the equality conditions in Proposition 4.3.2 must hold in the limit. In other words, one of the two cases must hold: either (1) there exists $x'$ such that $x' \in \arg \max_{x \in \mathcal{X}_s} f(x; \theta)$ for all $\theta \in \{\theta_1, \ldots, \theta_k\}$, or (2) the probability of one $\theta \in \{\theta_1, \ldots, \theta_k, \theta^* = \theta_L\}$ is very close to 1.

Before we discuss these two cases, we propose the following claim, which is proved at the end of the proof of this theorem.

**Claim 4.6.1.** Assume $\theta_L$ is the true parameter. If there exists $j < L$ such that $\sum_{n=0}^{\infty} (r^n_j)^2 \leq \infty$, then $\mathbb{P}(\lim_{n \to} p^n_L = 0) = 0$. In other words, there exists $\epsilon > 0$, such that $p^n_L > \epsilon$ happens infinitely often almost surely.

Now we go back to the two possible cases. For (1), according to Claim 4.6.1, $p^n_L$ does not go to 0, and hence, $x'$ is also the maximum of $f(x; \theta^*)$ out of $\mathcal{X}_s$; otherwise, there exist $\epsilon, \epsilon_1 > 0$ such that $p^n_L > \epsilon$ infinitely often, and when $p^n_L > \epsilon$ for $n$ large enough, $\nu^{KG-s,n}(x_1) > \epsilon_1$, which contradicts to $\lim_{n \to \infty} \nu^{KG-s,n}(x_1) = 0$. Therefore, we find $x' = \theta^*$. Note that $p^n_L$ does not go to zero for $1 \leq i \leq k$, where $k \geq 1$, and the truth is $\theta^* = \theta_L$, while for $k + 1 \leq i \leq L - 1$, we have $\lim_{n \to \infty} p^n_i = 0$. We claim $\lim_{n \to \infty} \nu^{KG-s,n}(x^n) = 0$. Fix some $\epsilon > 0$. On the one hand, for $\theta_{k+1}, \ldots, \theta_{L-1}$, their probabilities go to 0. We can thus discard them in KG with an error of at most $\epsilon$ for $n$ large enough, as a result of Equation (4.1). On the other hand, for $\theta_1, \ldots, \theta_k$ and $\theta^*$, since $p^n_i$ does not go to zero, by Corollary 4.3.5, $\sum_{n=0}^{\infty} (r^n_i)^2 < \infty$ for $1 \leq i \leq k$. Thus we have $\lim_{n \to \infty} \max_{1 \leq i \leq k} [f(x^n; \theta_i) - \min_{1 \leq i \leq k} f(x^n; \theta_i)] = 0$. By a similar logic in the proof of Lemma 4.3.6, the KG score is at most $\epsilon$ for $n$ large enough. Therefore, for $\forall \epsilon > 0$, $\nu^{KG-s,n}(x^n) < \epsilon$ for $n$ large enough. Thus $\lim_{n \to \infty} \nu^{KG-s,n}(x^n) = 0$.

Note that as $\lim_{n \to 0} r^n_i = 0$ for $1 \leq i \leq k$, all $f(x; \theta_i)$’s must intercept at the same point. Without loss of generality, let us assume that there is a unique interception $x_0$. Then there must exist some $N > 0$, such that $x^n \in B(x_0, r^*/2)$ for all $n > N$, where $r^*$ is the fixed real number in Assumption 4.3.1.

Let $x_1$ be a fixed point in $B(x_0, r^*/2)$ and $x_1 \neq x_0$. Then by Assumption 4.3.1, $\nu^{KG-s,n}(x_1) \leq \nu^{KG-s,n}(x^n)$ for all $n > N$ and thus $\lim_{n \to \infty} \nu^{KG-s,n}(x_1) = 0$. Hence one of the equality conditions in Proposition 4.3.2 must hold in the limit. In other words, one of the two cases must hold: either (1) there exists $x'$ such that $x' \in \arg \max_{x \in \mathcal{X}_s} f(x; \theta)$ for all $\theta \in \{\theta_1, \ldots, \theta_k\}$, or (2) the probability of one $\theta \in \{\theta_1, \ldots, \theta_k, \theta^* = \theta_L\}$ is very close to 1.

Before we discuss these two cases, we propose the following claim, which is proved at the end of the proof of this theorem.

**Claim 4.6.1.** Assume $\theta_L$ is the true parameter. If there exists $j < L$ such that $\sum_{n=0}^{\infty} (r^n_j)^2 < \infty$, then $\mathbb{P}(\lim_{n \to} p^n_L = 0) = 0$. In other words, there exists $\epsilon > 0$, such that $p^n_L > \epsilon$ happens infinitely often almost surely.

Now we go back to the two possible cases. For (1), according to Claim 4.6.1, $p^n_L$ does not go to 0, and hence, $x'$ is also the maximum of $f(x; \theta^*)$ out of $\mathcal{X}_s$; otherwise, there exist $\epsilon, \epsilon_1 > 0$ such that $p^n_L > \epsilon$ infinitely often, and when $p^n_L > \epsilon$ for $n$ large enough, $\nu^{KG-s,n}(x_1) > \epsilon_1$, which contradicts to $\lim_{n \to \infty} \nu^{KG-s,n}(x_1) = 0$. Therefore, we find $x' = \theta^*$.
arg\max_{x \in \mathcal{X}} f(x; \theta^*) as \ n \to \infty. Furthermore, if the candidate functions are pairwise not aligned, then \ x' = \arg\max_{x \in \mathcal{X}} f(x; \theta^*) together with the fact that \ x' \in \arg\max_{x \in \mathcal{X}} f(x; \theta) for all \ \theta \in \{\theta_1, \ldots, \theta_k\} means that \ k = 0, or in other words, \ \lim_{n \to \infty} p_n^\theta = 1. Therefore, we find the true \ \theta^* successfully, and thus also the optimal \ x^* \in \mathcal{X}.

For (2), if the probability of a fixed \ \theta goes to 1, by Claim 4.6.1, it must be \ \theta^*, in which case we would find both \ \theta^* and \ x^* in \ \mathcal{X}. Otherwise, consider the case in which the most probable \ \theta, whose probability is very close to 1, changes over time. Specifically, let us define an event \ A_n as \ p_i^n > 1 - \epsilon and \ p_j^{n+1} > 1 - \epsilon, for some \ i \neq j, where \ \epsilon is some fixed very small positive real number. We must have this event happen infinitely many times. That is, \ \mathbb{P}(\cap_{m=0}^\infty \cup_{n=m}^\infty A_n) > 0.

We will show that \ \mathbb{P}(\cap_{m=0}^\infty \cup_{n=m}^\infty A_n) = 0, which will complete the proof of the theorem. Let us start with taking a look at what \ \mathbb{P}(A_n) is, for any given set of \ \{r_l^n, 1 \leq l \leq L - 1, n \geq 0, such that \ \sum_n (r_l^n)^2 < \infty, \ \forall 1 \leq l \leq k, and \ \sum_n (r_l^n)^2 = \infty, \ \forall k + 1 \leq l \leq L - 1\}. For simplicity, we denote the set as \ \{r_l^n\}_k. If the event \ A_n happens, say we have \ p_i^n > 1 - \epsilon, \ p_j^{n+1} > 1 - \epsilon, and \ i \neq j. Note that \ p_i^n < \epsilon and \ p_j^{n+1} < \epsilon. Let \ r = r^n(\theta_j, \theta_i) \geq 0 and \ \hat{y} = \hat{y}^{n+1}.

Without loss of generality, let us assume that \ f(x^n; \theta_i) = 0, f(x^n; \theta_j) = r. Then

\[
\frac{\epsilon}{1 - \epsilon} > \frac{p_i^{n+1}}{p_j^{n+1}} = \frac{p_i^n}{p_j^n} \cdot \frac{\exp\left(-\frac{[\hat{y} - f(x^n; \theta_i)]^2}{2\sigma^2}\right)}{\exp\left(-\frac{[\hat{y} - f(x^n; \theta_j)]^2}{2\sigma^2}\right)} = \frac{p_i^n}{p_j^n} \cdot \exp\left[\frac{(\hat{y} - r)^2 - \hat{y}^2}{2\sigma^2}\right] > \frac{1 - \epsilon}{\epsilon} \cdot \exp\left[\frac{(\hat{y} - r)^2 - \hat{y}^2}{2\sigma^2}\right].
\]

Thus, \ \hat{y} > \frac{r}{2} - \frac{2\sigma^2 \log \frac{1}{\epsilon}}{r}; that is \ \mathbb{P}(A_n | \{r_l^n\}_k) < \mathbb{P}(\hat{y} > \frac{r}{2} - \frac{2\sigma^2 \log \frac{1}{\epsilon}}{r} | \{r_l^n\}_k). Note that \ \hat{y} \sim \mathcal{N}(f(x^n; \theta^*), \sigma). The distribution of \ \hat{y} is independent of \ \{r_l^n\}_k, and thus \ \mathbb{P}(\hat{y} > \frac{r}{2} - \frac{2\sigma^2 \log \frac{1}{\epsilon}}{r} | \{r_l^n\}_k) = \mathbb{P}(\hat{y} > \frac{r}{2} - \frac{2\sigma^2 \log \frac{1}{\epsilon}}{r}). We claim that \ \mathbb{P}(A_n | \{r_l^n\}_k) is upper bounded by \ c_0 \exp\left(\frac{r(n)}{c_1} - \frac{c_0}{r(n)}\right) for \ n > N, where \ r(n) = \max_{l=1}^k |r_l^n|, and \ c_0, c_1, c_2, N are some constants with \ c_0, c_1, c_2 > 0.
By definition, \( r = r_j^n - r_i^n \). Also note that \( \hat{y} \sim \mathcal{N}(-r_i^n, \sigma) \). We have

\[
\mathbb{P}(\hat{y} > \frac{r}{2} - \frac{2\sigma^2 \log \frac{\epsilon}{r}}{r}) = \frac{1}{\sqrt{2\pi}} \int_{r_j^n + r_i^n}^{\infty} \frac{2\sigma}{r_j^n - r_i^n} \frac{1}{r} \exp \left( -\frac{t^2}{2} \right) dt \tag{4.21}
\]

\[
< \frac{1}{\sqrt{2\pi}} \int_{r_j^n + r_i^n}^{\infty} \frac{2\sigma}{r_j^n - r_i^n} \frac{1}{r} \exp \left( -\frac{t}{2} \right) dt \tag{4.22}
\]

\[
= \frac{2}{\sqrt{2\pi}} \exp \left( -\frac{r_j^n + r_i^n}{4\sigma} - \frac{\sigma \log \frac{\epsilon}{r}}{r_j^n - r_i^n} \right) \tag{4.23}
\]

\[
< \frac{2}{\sqrt{2\pi}} \exp \left( \frac{r(n)}{2\sigma} - \frac{\sigma \log \frac{1}{\epsilon}}{2r(n)} \right) \tag{4.24}
\]

where Equation (4.21) to Equation (4.22) holds when \( r_j^n + r_i^n - \frac{2\sigma^2 \log \frac{\epsilon}{r}}{r_j^n - r_i^n} > 1 \). Since by assumption, \( r_j^n - r_i^n = r \geq 0 \), we have \( r_j^n + r_i^n - \frac{2\sigma^2 \log \frac{\epsilon}{r}}{r_j^n - r_i^n} \geq -\frac{r(n)}{\sigma} - \frac{\sigma \log \frac{\epsilon}{r(n)}}{r(n)} \). Note that \( \lim_{r' \to 0^+} \left( -\frac{r'}{\sigma} - \frac{\sigma \log \frac{\epsilon}{r'}}{r'} \right) = \infty \), and since \( \sum_n (r_l^n)^2 < \infty \), we have \( \lim_{n \to \infty} |r_l^n| = 0 \) for all \( 1 \leq l \leq k \). Thus there exists \( N \) such that for all \( n > N \), \( -\frac{r(n)}{\sigma} - \frac{\sigma \log \frac{\epsilon}{r(n)}}{r(n)} > 1 \) holds.

Therefore, for \( n > N \),

\[
\mathbb{P}(A_n \mid \{r_l^n\}_k) < \mathbb{P} \left( \hat{y} > \frac{r}{2} - \frac{2\sigma^2 \log \frac{\epsilon}{r}}{r} \right) = c_0 \cdot \exp \left( \frac{r(n)}{c_1} - \frac{c_2}{r(n)} \right), \tag{4.25}
\]

where \( c_0 = \sqrt{\frac{2}{\pi}} \), \( c_1 = 2\sigma \), and \( c_2 = \frac{\sigma \log \frac{1}{\epsilon}}{2} > 0 \).

Note that as \( \sum_n (r_l^n)^2 < \infty \) for all \( 1 \leq l \leq k \), \( \sum_n (r(n))^2 < \infty \). By Lemma 4.3.7 and Equation (4.25),

\[
\sum_{n=0}^{\infty} \mathbb{P}(A_n \mid \{r_l^n\}_k) = \sum_{n=0}^{N} \mathbb{P}(A_n \mid \{r_l^n\}_k) + \sum_{n=N+1}^{\infty} \mathbb{P}(A_n \mid \{r_l^n\}_k)
\]

\[
< \sum_{n=0}^{N} \mathbb{P}(A_n \mid \{r_l^n\}_k) + \sum_{n=N+1}^{\infty} c_0 \cdot \exp \left( \frac{r(n)}{c_1} - \frac{c_2}{r(n)} \right) < \infty.
\]

Therefore, by the Borel-Cantelli lemma, \( \mathbb{P}(\cap_{n=0}^{\infty} \cup_{n=m}^{\infty} A_n \mid \{r_l^n\}_k) = 0 \), for any fixed set of \( \{r_l^n\}_k \). Note that this also holds if we generalize the definition of \( \{r_l^n\}_k \) to \( \{r_l^n\}_k = \)}
\( \{r^n_l, 1 \leq l \leq L - 1, n \geq 0, \text{ such that } |\{l : 1 \leq l \leq L - 1, \sum_n(r^n_l)^2 < \infty\}| = k\}. \) Finally,

\[
\mathbb{P}(\cap_{m=0}^{\infty} \cup_{n=m}^{\infty} A_n) = \sum_{j=1}^{L-1} \mathbb{P}(k = j) \mathbb{P}(\cap_{m=0}^{\infty} \cup_{n=m}^{\infty} A_n | k = j) = \sum_{j=1}^{L-1} \mathbb{P}(k = j) \int_{\{r^n_i\}_k} \mathbb{P}(\cap_{m=0}^{\infty} \cup_{n=m}^{\infty} A_n | \{r^n_i\}_k, k = j) d\mathbb{P}(\{r^n_i\}_k | k = j) = \sum_{j=1}^{L-1} \mathbb{P}(k = j) \int_{\{r^n_i\}_k} 0 \cdot d\mathbb{P}(\{r^n_i\}_k | k = j) = 0.
\]

Now it suffices to prove Claim 4.6.1, which would complete the proof of Theorem 4.3.8.

According to the proof of Proposition 4.3.4, for any \( 1 \leq i < L \),

\[
\log \frac{p^n(\theta_i)}{p^n(\theta^*)} = -\frac{\sum_{m=1}^{n}(r^m_i)^2 + 2\sigma \sum_{m=1}^{n} r^m_i Z^m}{2\sigma^2}.
\]

For any \( n \), \( \log \frac{p^n(\theta_i)}{p^n(\theta^*)} \) is Gaussian with mean \(-\frac{1}{2\sigma^2} \sum_{m=0}^{n-1}(r^m_i)^2 \) and variance \( \frac{1}{\sigma^2} \sum_{m=0}^{n-1}(r^m_i)^2 \).

If \( \sum_{n=0}^{\infty}(r^n_i)^2 < \infty \), \( \log \frac{p^n(\theta_i)}{p^n(\theta^*)} \) has limit \( Y_i \), which is Gaussian with mean \( \mu_i \) and variance \( \sigma_i^2 \).

Now assume the contrary, which means that \( p^n_L \rightarrow 0 \) as \( n \rightarrow 0 \) with nonzero probability.

Assume there are \( k \) \( \theta \)’s satisfying \( \sum_n[r^n(\theta, \theta^*)]^2 < \infty \), and without loss of generality, assume they are \( \theta_1, ..., \theta_k \). Consider \( Y = \max(Y_1, Y_2, \ldots, Y_k) \). Since \( Y_1, ..., Y_k \) all have finite means and finite variances, we know \( \lim_{M \rightarrow \infty} \mathbb{P}(Y > M) = 0 \). Since \( p^n_L \rightarrow 0 \) and \( p^n_1 + \ldots + p^n_k \rightarrow 1 \), we know \( Y \rightarrow \infty \). This happens with probability 0 and hence completes the proof.

\( \square \)

Proposition 4.3.10. Under the continuous KG policy, for \( \forall \theta_i \neq \theta^* \ (1 \leq i \leq K) \), \( \lim_{n \rightarrow \infty} \mathcal{L}^n_i = 0 \) almost surely, where \( \mathcal{L}^n_i \) is the normalized likelihood of \( \theta_i \) at time \( n \).
Proof of Proposition 4.3.10

Assume the contrary is true, and let $\Omega_1$ be the set of events for the contrary to happen.

We start by proving an analogy of Theorem 4.3.8. That is, as $n \to \infty$, we have
\[
\arg\max_{x \in \mathcal{X}} \sum_{i=1}^{K} \mathcal{L}_i^n f(x; \theta_i) = \arg\max_{x \in \mathcal{X}} f(x; \theta^*) \text{ almost surely.}
\]

For a fixed $\omega \in \Omega_1$, we still assume there are $k$ $\theta$'s, $\{\theta_1, \theta_2, ..., \theta_k\}$, whose probabilities do not go to 0, and the truth is $\theta_K$, while all other $\theta$'s have probability 0 in the limit. Similar to the proof of Theorem 4.3.8, the assumption above implies that: (a) $\lim_{n \to \infty} \nu^{KG-s,n}(x^n) = 0$; (b) $f(x; \theta_i)$'s intercept at the same point, for all $i \in [K]$ such that $\mathcal{L}_i^n$ does not converge to zero; (c) there exists an $x_1$ near the intersection (but different from the intersection) that $\lim_{n \to \infty} \nu^{KG-s,n}(x_1) = 0$; and (d) as $n \to \infty$, one of the two conditions must hold: (1) there exists $x'$ such that $x' \in \arg\max_{x \in \mathcal{X}} f(x; \theta)$ for all $\theta \in \{\theta_1, ..., \theta_k\}$, or (2) the probability of exactly one $\theta \in \{\theta_1, ..., \theta_k, \theta^* = \theta_K\}$ is very close to 1. From the proof of Theorem 4.3.8, we know we can learn $\theta^*$ and thus also $x^* \in \mathcal{X}$ if (2) happens. Below we show we can also learn $\theta^*$ for (1) in the resampling case.

For any $\omega$, let $s_1(\omega), s_2(\omega), ..., s_n(\omega), ...$ be the times when resampling happens. By assumption, $\mathcal{L}_i^n \to 0$ for $k + 1 \leq i < L$. Hence, for any $\epsilon_1 > 0$, there exists $N(\omega)$, such that for any $n(\omega) > N(\omega)$, $\sum_{i=k+1}^{L-1} \mathcal{L}_i^n < \epsilon_1$. Also, if $\sum_{i=k+1}^{L-1} \mathcal{L}_i^n < \epsilon_1$ before resampling happens, then the probability of not selecting any $\theta \in \{\theta_{k+1}, ..., \theta_{L-1}\}$ in resampling has a lower bound $\epsilon_2 > 0$, where $\epsilon_2$ is a function of $\epsilon_1$ (actually, if $\epsilon_1 \to 0$, $\epsilon_2 \to 1$). Let $A_n$ be the event that the candidate set $\mathcal{L}^{s_n}$ only contains $\{\theta_1, ..., \theta_k, \theta^*\}$ after resampling finishes at time $s_n$. Moreover, let $A$ denote the event that $A_n$ happens infinitely often. We show that $A$ happens with probability 1 under (1). If this holds, then at the times when $A_n$ happens, the KG score of any $x \in \mathcal{X}$ is 0 since these functions are aligned. Therefore, the continuous KG policy will measure an $x$ at random, which contradicts to the assumptions that $\sum_{n=0}^{\infty} (r_i)^2 < \infty$ for $1 \leq i \leq k$, and thus completes the proof.
To show $A$ happens almost surely under (1), we show its opposite event $B$ happens with probability 0. By definition, $B$ is the event that $A^c$ happens all but a finite number of times. For any $\omega \in B$, there exists $T(\omega)$, such that for any $s_n > T(\omega)$, $\sum_{i=k+1}^{L-1} L^s_n(\omega) < \epsilon_1$ and $L^s_n(\omega)$ contains $\theta \in \{\theta_{k+1}, ..., \theta_{L-1}\}$ (when resampling finishes). For each $\omega$, let $T_0(\omega)$ be the first time such that for any $n \geq T_0(\omega)$, the above event happens. Let $B_n = \{ \omega : T_0(\omega) = n \}$, $n = 1, 2, 3, ...$. We have $\mathbb{P}(B_n) \leq \lim_{m \to \infty} \prod_{i=n}^{n+m} (1 - \epsilon_2) = 0$. Since $B = \bigcup_{n \geq 1} B_n$ where $B_n$'s are mutually exclusive, we have $\mathbb{P}(B) = \sum_{n=1}^{\infty} B_n = \sum_{n=1}^{\infty} 0 = 0$. Therefore, $\mathbb{P}(A) = 1$ (conditioning on the fact that (1) happens) and this completes the proof.

\[ \square \]

**Theorem 4.3.11.** Under Assumption 4.3.1, the continuous KG policy with resampling is asymptotically optimal, i.e, it is able to find both the optimal $x^* \in X$ and identify the true $\theta^*$ almost surely as we take an infinite number of measurements.

**Proof of Theorem 4.3.11**

Based on Proposition 4.3.10, to prove Theorem 4.3.11, it suffices to show that with probability 1, $L^n$ contains $\theta^*$ for all but a finite number of times.

The proof of the statement above is basically identical to the proof of Theorem 3.5.2 in Chapter 3. We thus only give a brief overview of the idea here, while interested readers could refer to the detailed proof in Chapter 3.

To show $\theta^* \in L^n$ for all but a finite number of times, we only need to consider the times when resampling happens. We first show that $\theta^* \in L^n$ happens infinitely often. To show this, we prove its opposite event, namely $\theta^* \notin L^n$, happens for all but a finite number of times. According to Proposition 4.3.10, $L^n(\theta^*) \to 1$, so the probability of always not selecting $\theta^*$ is $\prod_{n \to \infty} \epsilon_3 = 0$, where $\epsilon_3$ is the upper bound of $\sum_{\theta \neq \theta^*} L^n(\theta)$ for $n$ large enough.
Therefore, $\theta^* \in \mathbb{L}^n$ infinitely often almost surely. For $n$ large enough, once $\theta^*$ is included in $\mathbb{L}^n$, since its probability is large, $\theta^*$ would never get removed. Hence, $\theta^* \in \mathbb{L}^n$ for all but a finite number of times with probability 1. Therefore, $\bar{f}(x) = f(x; \theta^*)$ almost surely, and we can find both $x^* \in \mathcal{X}$ and $\theta^*$ with probability 1.

$\square$
Chapter 5

Optimal Learning with Local
Nonlinear Parametric Belief Models
over Continuous Designs

In previous chapters, we solve the optimization problem of an unknown function in the optimal learning framework assuming we know the parametric form. However, in many situations, a parametric function that can describe the problem globally may not exist; instead, we may only have some simple, low-order functions that are locally accurate. In this chapter, we propose an algorithm that solves the problem when a globally accurate parametric function is not accessible, while a local model, which can well describe the shape of the true function in any local region, is available. We construct belief models using local approximations, and apply our continuous KG policy to the model. Our method provides a solution to a general class of problems with multidimensional continuous alternatives but without a global parametric expression.
5.1 Introduction

We consider the following problem: we have a smooth function over a multidimensional continuous space. The function itself is unknown to us, and there does not exist a model to describe it globally. But within any local region, the function can be well approximated by a parametric function, say \( g(x; \theta) \), which may be nonlinear in the parameter vector \( \theta \). Our goal is to find the global maximum of the true function. We can take a series of measurements to collect information at any point in the domain, but the measurements are noisy and expensive, as might arise in simulations, laboratory or field experiments. Therefore, we only have a budget of \( N \) measurements, and after exhausting the budget, we need to give our best estimate of the optimal \( x \) (where \( x \) is also called an alternative).

We first briefly review a class of Bayesian policies known as the knowledge gradient (KG), with special attention on KG with different belief models (see a detailed review in Chapter 2). KG is designed to maximize the value of information from each measurement, and hence well suited to problems where the cost of running experiments is high. By construction, KG is a myopically optimal policy, but has also been shown to be asymptotically optimal. KG has been developed for the three major classes of belief models: look-up table models, parametric models, and non-parametric models. For look-up table models, Frazier et al. (2008) proposes KG for independent alternatives, and Frazier et al. (2009) establishes KG for correlated alternatives. For parametric belief models, Negoescu et al. (2011) focuses on the case when the function is linear in the unknown parameters, while Chen et al. (2014) studies the case when the parametric function is nonlinear in unknown parameters. Some work on non-parametric models includes Mes et al. (2011), which proposes a hierarchical aggregation technique for KG, and Barut & Powell (2014), which estimates functions using kernel regression. All these methods are designed for discrete alternatives. For problems with continuous alternatives, Scott et al. (2011) uses Gaussian Process Regression based on a look-up table belief model, but is limited to only low dimensions; Chapter 4 proposes an algorithm when the true function can be modeled by a differentiable parametric function,
which works well in problems with even 10 or 20 dimensions, but requires a globally accurate parametric model for the true function.

For problems that do not have a globally accurate parametric belief model, an estimate of the global function is usually built using local approximation methods. There has been a rich literature on local approximation techniques (see Jamshidi & Powell 2016 for a comprehensive review). The locally linear model in Fan (1992), Fan & Gijbels (1996) builds linear models around each observation, but cannot provide a global mathematical formula for the regression function. Another class of algorithms build local models around local regions instead of each data point. Hannah et al. (2011) uses the Dirichlet process mixture model to identify local regions of the input space, and then fit linear models around each region. Moreover, radial basis functions (RBFs) Broomhead & Lowe (1988), Buhmann (2000) have received considerable attention due to their simplicity and generality. Jones et al. (1990) presents normalized RBF, which perform well for limited training data. Jamshidi & Powell (2016) proposes the Dirichlet Cloud Radial Basis Function (DC-RBF) model, which provides a compact representation over the entire alternative space and a fast method to update the approximation. Specifically, Cheng et al. (2015) combines the DC-RBF model with the knowledge gradient model for linear functions proposed in Negoeescu et al. (2011), and derives the KG policy for a local approximation model that fits linear functions around local regions. This is the first optimal learning work based on local approximation models. However, it can only handle the case when the local approximation is a linear function, and when the alternative space is discrete.

In this chapter, we address the problem when the alternative space is continuous and multidimensional, but a globally accurate model is not available, while any local region can be well described by a parametric function. Our work in this chapter makes the following contributions:

(1) We propose a method to build a belief model consisting of a variety of global approximations, and show how the model gets updated dynamically as we run experiments;
We further incorporate the optimal learning methods and develop the KG policy for our belief model over a continuous alternative space;

We evaluate our algorithm on a variety of problems, ranging from a series of test functions to a real-world application on carbon nanotube growth, all showing strong performance.

The remainder of this chapter is organized as follows. We review a local approximation method known as the DC-RBF model (Jamshidi & Powell 2016) in Section 5.2. In Section 5.3, we introduce our method to construct a belief model of the true function and how we use the Knowledge Gradient policy to select an alternative to measure. We show some empirical results on several different synthetic problems and a real application in Section 5.4, and conclude in Section 5.5.

5.2 Review of a Local Approximation Technique: DC-RBF

Before presenting our algorithm, we first review a related local approximation technique, known as the Dirichlet Cloud Radial Basis Function (DC-RBF) model (Jamshidi & Powell 2016). The DC-RBF model is a function approximation technique that employs radial basis functions (RBF) to link locally accurate linear functions to construct a global representation. The scheme groups the data into clusters, each corresponding to a local region and described by a locally accurate linear function. A nonlinear weighting system, defined by the RBF model, determines the contribution of each local model to the overall model output.

Suppose we have a predefined parameter, known as the radius and denoted as $D_T$, to control the size of each local region. The alternatives are grouped into clusters recursively. Initially, the first measured alternative, $x^0$, forms the only cluster itself. Assume at time $n$, the previously measured $n$ alternatives, $\{x^0, ..., x^{n-1}\}$, form $C^n$ clusters. For any cluster
Suppose it contains \( k_j \) data points, \( \{x_j^{(1)}, \ldots, x_j^{(k_j)}\} \), and its centroid \( c_j \) is defined as the arithmetic mean, given by \( c_j = \frac{x_j^{(1)} + \ldots + x_j^{(k_j)}}{k_j} \). When a newly measured alternative \( x^n \) arrives, \( x^n \) is assigned to a cluster according to the distance from \( x^n \) to \( c_j \). Specifically, let \( J \) be the cluster whose centroid has the smallest distance to \( x^n \), namely 
\[ J = \text{argmin}_{1 \leq j \leq C_n} D_j = \text{argmin}_{1 \leq j \leq C_n} \|x^n - c_j\| \]. If \( D_J < D_T \), \( x^n \) is added to cluster \( J \); otherwise, \( x^n \) forms a new cluster itself.

For each cluster \( j \), a linear model is fitted based on the \( k_j \) data points. Assume the linear model is given by \( g_j(x) = [x^T 1]\theta_j \). A width matrix, \( W_j \), is defined to indicate the influence of \( g_j(x) \). Specifically, regard \( x \) as a multivariate random variable with \( k_j \) observations, and let \( \Sigma_j \) be the empirical covariance matrix, given by 
\[ \Sigma_j = \frac{1}{k_j} \sum_{i=1}^{k_j} [x_i^{(i)}]^T - c_j c_j^T \]. 
Let \( W_j = \Sigma_j + \lambda I \), where \( \lambda \) is a positive constant and \( I \) is the identity matrix to avoid singularity. A global approximation of the true function is given by 
\[ f^n(x) = \sum_{j=1}^{C_n} w_j(x) \cdot [x^T 1]\theta_j \],
(5.1)
where \( w_j(x) \) is a normalized weight of \( g(x; \theta_j) \) at \( x \), given by 
\[ w_j(x) = \frac{\phi(||x - c_j||_{W_j})}{\sum_{k=1}^{C_n} \phi(||x - c_k||_{W_k})} \].
(5.2)
Here \( \phi(||x - c||_{W}) \) is the RBF function, and for the Gaussian kernel, we have \( \phi(||x - c||_{W}) = \exp(-||x - c||_{W}^2) \), where \( ||z||_W = \sqrt{z^T W^{-1} z} \).

The DC-RBF model provides a method for function approximation using locally linear models. However, its clustering result is often defective, since a data point is clustered only using the data prior to itself and thus the result heavily depends on the order of data. We refine the clustering procedure of DC-RBF in our algorithm, and construct global approximations similar to Equation (5.1) but using local nonlinear functions. Furthermore, we maintain a variety of radii instead of a fixed radius to construct a diversity of global approximations. We then apply the KGDP policy to select alternatives, and provide an algorithm that can handle a multidimensional continuous \( x \) space.
5.3 Algorithm

We consider the problem:

\[
\arg\max_{x \in \mathcal{X}} f^*(x),
\]

where \(f^*(x)\) is the underlying true function which is unknown to us, and \(\mathcal{X}\) is a compact set of alternatives. Our goal is to find the optimal alternative \(x^*\) after a budget of \(N\) sequential measurements, which are noisy and expensive. We assume that for any local region of proper size in \(\mathcal{X}\), \(f^*(x)\) can be well approximated by a differentiable function \(g(x; \theta)\), which is well-defined on \(\mathcal{X}\) and can be nonlinear in \(\theta\). In other words, for any \(x_0 \in \mathcal{X}\), there exists \(\theta^*\) such that \(|f^*(x) - g(x; \theta^*)| < \epsilon\) for any \(x \in B(x_0, r)\), with some \(r \in [D_{T,\text{min}}, D_{T,\text{max}}]\). Here we use \(D_{T,\text{min}}\) and \(D_{T,\text{max}}\) to indicate the range of reasonable radii of local regions.

At time \(n\), suppose we decide to measure an alternative \(x^n\) according to some policy, and the measurement result is \(\hat{y}^{n+1}\), where the superscripts imply that \(\hat{y}^{n+1}\) is unknown until after the \((n+1)\)-th measurement. Therefore, our measurements up to time \(n\) are \(\{(x^0, \hat{y}^1), ..., (x^{n-1}, \hat{y}^n)\}\). In this chapter, we assume the measurement noise \(W^n\) is Gaussian with 0 mean and variance \(\sigma^2\). Therefore, \(\hat{y}^{n+1} = f^*(x^n) + W^{n+1}\), where \(W^{n+1} \sim \mathcal{N}(0, \sigma^2)\), i.i.d.

Note that throughout the chapter, we assume the \(\mathcal{X}\) space has been properly scaled such that each dimension has roughly the same scale.

5.3.1 Sketch of the Basic Ideas

We start with an overview of some key elements of our algorithm:

1. We use a family of radii to construct a variety of clustering results, rather than a single radius as the tunable parameter.

2. For each cluster in each clustering result, we fit a local, low-order approximation \(g(x; \theta)\), using the resampling logic in Chapter 3.
3. For each clustering result, we use the normalized RBF (Jones et al. 1990, Jamshidi & Powell 2016) to stitch the local approximations together to generate a global approximation of the true function.

4. We keep a probability for each global approximation and update the probabilities as we run experiments.

5. If some of the probabilities are too low, the radii, the clustering results, and the local approximations would get updated to create new global approximations.

6. We apply the logic of the continuous KGDP introduced in Chapter 4 to our belief model to select an \( x \) to measure.

In the following, we discuss Items 1-3 in Section 5.3.2, Item 4 in Section 5.3.3 and Section 5.3.4, Item 5 in Section 5.3.5 and Item 6 in Section 5.3.6.

5.3.2 Construction of Candidate Functions

In this part, we introduce the general idea of constructing our belief model, while some implementation details are introduced in the following sections. We illustrate the construction of the belief model with the help of Figure 5.1. As shown in Figure 5.1, our belief model consists of a set of candidate functions, \( \{f_i^n, i = 1, \ldots, L\} \), specified by a family of radii, \( \{D_{n,i}, i = 1, \ldots, L\} \). Each \( f_i^n \) itself is a global approximation of the true function with a specific radius \( D_{n,i} \), while our overall estimate of the true function, \( \bar{f}^n \), is the weighted sum of the candidate functions.

At time 0, we start with \( L \) radii, \( \{D_{0,1}, \ldots, D_{0,L}\} \), evenly distributed in the range \([D_{T,min}, D_{T,max}]\). The radii may get updated as we run measurements, as discussed in Section 5.3.5. Suppose at time \( n \), the \( L \) radii are \( D_{n,1}, D_{n,2}, \ldots, D_{n,L} \), all within \([D_{T,min}, D_{T,max}]\). As shown in Figure 5.1, each \( D_{n,i} \) results in a clustering result \( i \), containing \( C_i^n \) clusters (the clustering method is introduced in Section 5.3.4 and Section 5.3.5). For each cluster \( j \) in
From left to right: (1) radii of local regions ($D^n_T$’s); (2) clustering results; (3) fitting $\theta$ for each cluster; (4) candidate functions; (5) estimate of the true function.

Figure 5.1: Construction of candidate functions.

clustering result $i$, we solve a statistical fitting problem, as introduced in Section 5.3.5, to get a local parameter, $\theta^n_{ij}$, such that $g(x; \theta^n_{ij})$ approximates the measurement results in the local region around cluster $j$. A candidate function $f^n_i(x)$ is defined as

$$f^n_i(x) = \sum_{j=1}^{C^n_i} w_{ij}(x) g(x; \theta^n_{ij}), \quad (5.3)$$

where $w_{ij}(x)$, similar to Equation (5.2), is constructed by the normalized RBF:

$$w_{ij}(x) = \frac{\phi(||x - c_{ij}\|| \omega_{ij})}{\sum_{k=1}^{C^n_i} \phi(||x - c_{ik}\|| \omega_{ik})}.$$

Note that $g(x; \theta)$ may not necessarily be linear. Moreover, since $w_{ij}(x)$ and $g(x; \theta^n_{ij})$ are both differentiable in $x$, so is $f^n_i(x)$, which makes it possible to compute derivatives with respect to $x$ as is required by the algorithm for KGDP with continuous alternatives in Chapter 4. In practice, we can use the finite difference as an approximation of the gradient of $f^n_i(x)$.
We maintain a probability \( p_i^n \) for each candidate function \( f_i^n(x) \), and use the expectation of the candidate functions as the estimate of the true function \( f^*(x) \), i.e.,

\[
f^*(x) \approx \bar{f}^n(x) = \sum_{i=1}^{L} p_i^n f_i^n(x).
\]

(5.4)

At time \( n \), our best estimate of the optimal alternative \( \hat{x}^n \) is given by \( \hat{x}^n = \arg\max_{x \in \mathcal{X}} \bar{f}^n(x) \).

### 5.3.3 Calculation of Candidate Function Probabilities

At time 0, we assign equal prior probabilities to all the \( L \) candidate functions, namely \( p_i^0 = \frac{1}{L} \) for all \( i \). We recalculate the \( p_i \)'s after each measurement according to Bayes’ rule.

In previous variations of KGDP (Chen et al. 2014, Chapter 3, Chapter 4), due to the assumption that our parametric model is globally accurate, the probabilities are calculated only considering the measurement noise \( \sigma \). However, in our problem, since the candidate functions \( f_i^n(x) \) can almost never be exactly the same as the true function, we should use a larger \( \sigma \) to compensate for the effect of model error when calculating probabilities (and when calculating the KG score, as discussed in Section 5.3.6). For instance, consider the case when there is no measurement noise. With a globally accurate parametric model, we might be able to identify the true function with as few as just one measurement. However, this is no longer a trivial problem in our case, since the noise-free measurements can only allow us to find the truth of a local region.

Let \( \tilde{\sigma}^2 > \sigma^2 \) be the adjusted variance used in calculating the probabilities and KG scores. In practice, \( \tilde{\sigma} \) should be set according to our estimate of the model discrepancy. Some typical values might be \( \tilde{\sigma} = 1.5\sigma \) or \( \tilde{\sigma} = \sigma + 0.1(f_{\text{max}} - f_{\text{min}}) \), where \( f_{\text{max}} - f_{\text{min}} \) is the estimated range of the true function. Recall that our experiments have noise \( W^n \sim \mathcal{N}(0, \sigma^2) \), and at time \( n \), the measurement results are \( \{(x^0, \hat{y}^1), \ldots, (x^{n-1}, \hat{y}^n)\} \). Hence, the likelihood of \( f_i^n \) is
given by

\[ L_i^n = \prod_{j=1}^{n} \exp \left( -\frac{[\hat{y}_j - f_i^n(x_j)]^2}{2\sigma^2} \right), \]  

(5.5)

We have \( p_i^n \propto p_i^0 \cdot L_i^n \) by Bayes’ rule. Therefore, \( p_i^n \) is given by

\[ p_i^n = \frac{L_i^n}{\sum_{j=1}^{L} L_j^n}. \]  

(5.6)

5.3.4 Minor Update of Candidate Functions

After each measurement, we get a new result \((x^n, \hat{y}^{n+1})\). We first update each clustering result by adding \( x^n \) in a lazy recursive way, as in DC-RBF. Specifically, for each \( i, 1 \leq i \leq L \), we calculate \( J_i = \arg\min_{1 \leq j \leq C_i^n} D_j = \arg\min_{1 \leq j \leq C_i^n} ||x^n - c_{ij}|| \). If \( D_{J_i} < D_{n,T,i} \), \( x^n \) is added to cluster \( J_i \) in cluster result \( i \); otherwise, \( x^n \) forms a new cluster itself in cluster result \( i \). We also update the corresponding centroids \( c_{ij} \) and width matrices \( W_{ij} \) accordingly. We call this recursive clustering procedure lazy clustering.

If \( x^n \) is added to an existing cluster for all clustering results in lazy clustering, we call such an update a minor update. In a minor update, only one \( c_{ij} \) and one \( W_{ij} \) in each clustering result get updated. As a result, the weights \( w_{ij} \) in Equation (5.3) are moderately changed, and hence the candidate functions are only slightly different from before updating. In other words, in Figure 5.1, the radii are unchanged, the clustering results are almost unchanged (with only \( x^n \) to an existing clustering), and the set of \( \theta_{ij} \)’s are unchanged, while only the fourth column, particularly \( w_{ij} \), is refreshed. We then update the probabilities \( \{p_i^{n+1}\} \) according to Equation (5.6) once a minor update is done.

However, if \( x^n \) creates new clusters in some clustering results during lazy clustering, or if some \( p_i^{n+1} \)’s after a minor update are too small, we turn to a major update, in which case we update some or all of the first three steps in Figure 5.1 for corresponding functions, namely (1) the radii, (2) the clustering results, and (3) the set of \( \theta_{ij} \)’s.
After each measurement, we always try a minor update first rather than do a major update directly, in order to keep the candidate functions more stable.

5.3.5 Major Update of Candidate Functions

At certain times, we may find that the probabilities of some candidate functions are too small, or that new clusters have been created during lazy clustering. In such cases, we do a major update of the corresponding candidate functions to better adapt to the data, by three possible operations: (1) resampling $D_{T,i}$’s, (2) re-clustering, and (3) refitting local $\theta_{ij}$’s, corresponding to the first three columns in Figure 5.1.

We define three conditions, each of which can trigger a major update of the candidate functions: (1) More than $L/2$ candidate functions have probabilities smaller than some threshold $\epsilon_p$; (2) Any clustering result has created new clusters; (3) We haven’t done major updates for $n^r$ (say $n^r = 5$) time steps, or in other words, we have used roughly the same candidates for $n^r$ times. In (1), at least all candidates with $p^n_i < \epsilon_p$ get updated; in (2), at least all candidates with new clusters get updated; in (3), we first set the $L/4$ smallest probabilities as 0 and thus at least $L/4$ of candidates get updated.

We define the termination condition for a major update to be at most a quarter of the candidates having $p^n_i < \epsilon_p$.

Every time a major update is triggered, for all candidate functions with low probabilities, we first update the local function $g(x; \theta_{ij})$ for each cluster, keeping the $D_{T,i}$’s and clustering results the same. We then recalculate the probabilities according to Equation (5.6). If the termination condition is satisfied, the major update finishes. Otherwise, we next update the radii and clustering results, followed by updating $\theta_{ij}$’s. We keep updating all the three elements until the termination condition is met. In practice, to make sure the update terminates after a reasonable amount of time, we keep track of the number of times each function fails to reduce the fitting error after a round of updating compared with before this round. If the failure count of a candidate function hits some threshold, we exclude the function when
checking the termination condition (refer to Algorithm 5, particularly failure_count, for more
detail).

We assume \( f^n_i \) is one of the candidate functions that needs a major update, and discuss
the three operations of a major update in detail below, in the order that they are conducted,
namely (1) refitting \( \theta_{ij} \)'s for local regions, (2) resampling \( D_{T,i} \), and (3) constructing new
clusterings. We include a flowchart of the major update in the end.

**Refitting the Local Functions**

When a major update is needed for \( f^n_i \), we first fix its radius \( D^n_{T,i} \) and its clustering result,
and try refitting \( \{ \theta^n_{ij}, j = 1, \ldots, C^n_i \} \). For each cluster, we search for the \( \theta \) that minimizes
the SSE (sum of squared errors), which is equivalent to the MLE (maximum likelihood estimate) fitting under Gaussian noise. Here we use a method similar to the resampling
method in Chapter 3. We keep a large pool of \( K \theta \)'s, assuming that \( K \) is large enough to
cover almost all reasonable \( \theta_{ij} \)'s that could arise when fitting a local region. Suppose cluster
\( j \) (1 \( \leq \) \( j \) \( \leq \) \( C^n_i \)) contains \( k_j \) data points: \( \{(x^{(m)}_j, \hat{y}^{(m)}_j), m = 1, \ldots, k_j \} \). For each \( \theta \) in the large
pool, we calculate the SSE between \( g(x; \theta) \) and the \( k_j \) measurements, given by:

\[
SSE(\theta) = \sum_{m=1}^{k_j} \left[ g(x^{(m)}_j; \theta) - \hat{y}^{(m)}_j \right]^2.
\]

(5.7)

We resample \( \theta \) from a sub-level set of the \( SSE(\theta) \). Specifically, let the \( S \theta \)'s with the
smallest \( SSE(\theta) \) form the small pool, denoted as \( S \), where \( L < S \ll K \). The new \( \theta_{ij} \) is
then sampled from the small pool \( S \). The probability of \( \theta \in S \) being drawn is given by its
normalized likelihood, namely

\[
p(\theta) = \frac{\exp \left( -\frac{\sum_{m=1}^{k_j} \left[ g(x^{(m)}_j; \theta) - \hat{y}^{(m)}_j \right]^2}{2\sigma^2} \right)}{\sum_{\theta' \in S} \exp \left( -\frac{\sum_{m=1}^{k_j} \left[ g(x^{(m)}_j; \theta') - \hat{y}^{(m)}_j \right]^2}{2\sigma^2} \right)} = \frac{\exp \left( -\frac{SSE(\theta)}{2\sigma^2} \right)}{\sum_{\theta' \in S} \exp \left( -\frac{SSE(\theta')}{2\sigma^2} \right)}.
\]

(5.8)
Note that we use the adjusted variance $\tilde{\sigma}^2$ in this formula. As illustrated in Chapter 3, the method of sampling from the small pool keeps a balance of exploration and exploitation in the $\theta$ space.

There are two heuristics in refitting local functions: (1) we can include some nearby data points outside of cluster $j$ when fitting $\theta_{ij}$, in order to make the candidate function $f_i^n$ smoother, and (2) if we have enough data points in a certain cluster, we can directly solve a MSE problem to obtain the optimal $\theta_{ij}$ instead of sampling it from a large pool. See Appendix 5.6.1 for details.

After updating $\{\theta_{ij}^n, j = 1, ..., C_i^n\}$, we check if the SSE between the new candidate function and the measurement results has decreased from before this update. We only replace the candidate function if the SSE has actually decreased. Additionally, every time a round of candidate function updating is finished, we recalculate the probabilities of all candidate functions using Equation (5.6) and check if the new probabilities satisfy the termination condition. If not, we repeat the major update. In such case, if we still have $p_i^n < \epsilon_p$ after updating $\{\theta_{ij}^n, j = 1, ..., C_i^n\}$, we need to update $D_{T,i}^n$ and the clustering result as well.

**Resampling a Radius**

As discussed above, if updating $\{\theta_{ij}^n, j = 1, ..., C_i^n\}$ cannot make $p_i^n > \epsilon_p$, we need to update $D_{T,i}^n$. First of all, it is worth pointing out that there is a tradeoff in the value of $D_{T,i}^n$. A smaller $D_{T,i}^n$ enables $g(x; \theta_{ij}^n)$ to match the measurement results better; on the other hand, it also results in more clusters, each containing fewer data, and thus makes the fitting problem undetermined. Our algorithm aims at sampling some new $D_{T,i}^n \in [D_{T,min}^n, D_{T,max}^n]$ with a balanced value. Roughly speaking, we look at the distribution of the current $L$ pairs $(D_{T,l}^n, p_l^n), 1 \leq l \leq L$, and resample a $D_{T,i}^n$ in $[D_{T,min}^n, D_{T,max}^n]$ such that the new $D_{T,i}^n$ falls closer to the $D_{T,l}^n$’s with higher probabilities.

Specifically, we first rescale all $D_{T,l}^n$’s from $[D_{T,min}^n, D_{T,max}^n]$ to $[0, 1]$ by letting $D_T' = \frac{D_T^n - D_{T,min}^n}{D_{T,max}^n - D_{T,min}^n}$. We then approximate the distribution of $(D_T', p_l^n)$ by a Beta distribution,
which provides a variety of distribution shapes on a limited interval with a small number of parameters. We assume there exists a constant $c$ such that for any $l = 1, 2, ..., L$, 

$$p^n_l = c \cdot (D'_{T,l})^{\alpha - 1}(1 - D'_{T,l})^{\beta - 1}.$$ 

Therefore, we can approximate $\alpha$, $\beta$, and $c$ by solving the following optimization problem of minimizing the sum of squared error:

$$\min_{\alpha > 0, \beta > 0, c > 0} \sum_{l=1}^{L} \left[ c \cdot (D'_{T,l})^{\alpha - 1}(1 - D'_{T,l})^{\beta - 1} - p^n_l \right]^2.$$ 

Note that for given $\alpha$ and $\beta$, the optimal $c$ can be uniquely determined by solving an ordinary least squares problem, with a closed-form equation given by

$$c = \frac{\sum_{l=1}^{L} (D'_{T,l})^{\alpha - 1}(1 - D'_{T,l})^{\beta - 1} p^n_l}{\sum_{l=1}^{L} [(D'_{T,l})^{\alpha - 1}(1 - D'_{T,l})^{\beta - 1}]^2},$$ 

so the optimization problem can be simplified to

$$\min_{\alpha > 0, \beta > 0} \sum_{l=1}^{L} \left[ c \cdot (D'_{T,l})^{\alpha - 1}(1 - D'_{T,l})^{\beta - 1} - p^n_l \right]^2,$$ 

where $c$ is given in Equation (5.9). Equation (5.10) can be solved by grid search. Figure 5.2 shows some results of fitting the Beta distribution. As shown in Figure 5.2, the probability densities of the left and right most parts are sometimes too low. We thus sample $D'_{T,i}$ from a mixture of the Beta distribution and uniform distribution, each with probability 0.5, and then transform back to get a new $D^n_{T,i}$, by $D^n_{T,i} = D_{T,min} + D'_{T,i} \cdot (D_{T,max} - D_{T,min})$.

**Reconstructing Clustering Results**

After a new $D^n_{T,i}$ is sampled, we also need to update the corresponding clustering result. We first permute the data. Let $(x^{(0)}, x^{(1)}, ..., x^{(n-1)})$ be a random permutation of the previously measured alternatives $(x^0, ..., x^{n-1})$. We add $(x^{(0)}, x^{(1)}, ..., x^{(n-1)})$ one by one according to the DC-RBF clustering procedure, as if they arrive in such an order, resulting in $C^n_i$ clusters.
with centroids \( \{c_{ij}, j = 1, \ldots, C_i^n\} \). We then run K-means to refine the clustering result. The number of clusters in K-means is set as \( C_i^n \), and the initial centers in K-means are set as the \( C_i^n \) centroids \( \{c_{ij}, j = 1, \ldots, C_i^n\} \).

Instead of running the DC-RBF clustering procedure, we use refinements like permutation and K-means because DC-RBF only looks at previous data when assigning data to clusters, and thus its result heavily depends on the order of the data. As a consequence, DC-RBF often generates suboptimal results, as shown in the first and third images in Figure 5.3. In comparison, after we run K-means based on DC-RBF clustering, as shown in the second and fourth images, the clustering results get much better.
Flowchart of Major Update of Candidate Function

We summarize the procedure of a major update of the candidate functions, as shown in Algorithm 5.

**Algorithm 5 Major Update of Candidate Functions**

- **Input:** Current candidates: \( \{f^n_i\} \); Probabilities: \( \{p^n_i\} \); Measurement history: \( \{(x^0, \hat{y}^1), ..., (x^{n-1}, \hat{y}^n)\} \).
- **Output:** Updated candidates: \( \{f^n_i\} \); Updated probabilities: \( \{p^n_i\} \).

1. **if** \(|\{i : p^n_i < \epsilon_p\}| \geq L/2\), or some candidates have added new clusters, or major update hasn’t been conducted for \(n^r\) iterations **then**
2. **while** \(\text{while\_flag} = 1\); \(\mathbb{L}_{rm-\text{old}} = \emptyset\); **failure\_count = (0, ..., 0) (zero vector of length \(L\)).**
3. **Set probabilities of the \(L/4\) least probable candidates as 0.
4. **while** \(\text{while\_flag}\) **do**
5. \(\mathbb{L}_{rm} = \{i : p^n_i < \epsilon_p\} \cup \{i : \text{clustering result } i \text{ has new clusters}\}\).
6. \(j = 1\).
7. **while** \(j \leq |\mathbb{L}_{rm}|\) **do**
8. \(i = \mathbb{L}_{rm}(j)\).
9. **if** \(i\) is in \(\mathbb{L}_{rm-\text{old}}\) **then**
10. Fit a Beta distribution and sample a new \(D^n_{T,i}\).
11. Permute \(\{x^0, ..., x^{n-1}\}\) and run DC-RBF clustering using the permutation.
12. Use K-means to refine the clustering result.
13. **end if**
14. **for** each cluster **do**
15. Resample \(\theta\) from the large pool (or fit \(\theta\) if overdetermined).
16. **end for**
17. **if** the new candidate has smaller SSE **then**
18. Replace the candidate \(f^n_i\) with the newly updated one; \(j = j + 1\).
19. **else**
20. \(\text{failure\_count}(i) = \text{failure\_count}(i) + 1\).
21. **if** \(\text{failure\_count}(i) \geq 100\) **then**
22. \(j = j + 1\).
23. **end if**
24. **end if**
25. **end while**
26. \(\mathbb{L}_{rm-\text{old}} = \mathbb{L}_{rm}\).
27. **end while**
28. Recalculate \(p^n_i\) for \(i = 1, ..., L\) by Equation (5.6).
29. **if** \(|\{i : p^n_i < \epsilon_p \text{ and } \text{failure\_count}(i) < 100\}| < L/4\) **then**
30. \(\text{while\_flag} = 0\).
31. **end if**
32. **end if**
33. **return** \(\{f^n_i(x)\}, \{p^n_i\}, i = 1, ..., L\).
5.3.6 KGDP and continuous KGDP

To further incorporate the KGDP logic, we need to define the recursive updating formula of $p_i^n$. Recall that after a minor update, we roughly have $f_{i}^{n+1}(x) \approx f_{i}^{n}(x)$, $\forall x \in X$. Then the likelihood of $f_{i}^{n+1}$ is given by

$$
\mathcal{L}_{i}^{n+1} = \prod_{j=1}^{n+1} \exp \left( -\frac{[\hat{y}_j - f_{i}^{n+1}(x_j-1)]^2}{2\sigma^2} \right) \approx \prod_{j=1}^{n+1} \exp \left( -\frac{[\hat{y}_j - f_{i}^{n}(x_j-1)]^2}{2\sigma^2} \right)
$$

$$
= \mathcal{L}_{i}^{n} \exp \left( -\frac{[\hat{y}_{n+1} - f_{i}^{n}(x_n)]^2}{2\sigma^2} \right).
$$

Therefore, $p_{i}^{n+1}$ can be approximately calculated by

$$
p_{i}^{n+1} = \frac{\mathcal{L}_{i}^{n+1}}{\sum_{j=1}^{L} \mathcal{L}_{j}^{n+1}} \approx \frac{\mathcal{L}_{i}^{n} \exp \left( -\frac{[\hat{y}_{n+1} - f_{i}^{n}(x_n)]^2}{2\sigma^2} \right)}{\sum_{j=1}^{L} \mathcal{L}_{j}^{n} \exp \left( -\frac{[\hat{y}_{n+1} - f_{j}^{n}(x_n)]^2}{2\sigma^2} \right)} = \frac{p_{i}^{n} \exp \left( -\frac{[\hat{y}_{n+1} - f_{i}^{n}(x_n)]^2}{2\sigma^2} \right)}{\sum_{j=1}^{L} p_{j}^{n} \exp \left( -\frac{[\hat{y}_{n+1} - f_{j}^{n}(x_n)]^2}{2\sigma^2} \right)}.
$$

(5.11)

Now with the candidate functions $f_i^n(x)$ defined in Equation (5.3), the recursive updating of $p_i^n$ defined in Equation (5.11), and the estimate of the true function $\hat{f}_i^n(x)$ defined in Equation (5.4), we can calculate the Knowledge Gradient score similar to Chen et al. (2014) on a discrete alternative set $X_s$. Specifically, as defined in Chen et al. (2014) and further analyzed in Chapter 3, $\forall x \in X_s$, we have

$$
u_{KGDP-f_i^n}(x) = \mathbb{E}^n \left[ \max_{x' \in X_s} \hat{f}_i^{n+1}(x') \right] - \max_{x' \in X_s} \hat{f}_i^n(x')
$$

$$
= \mathbb{E}_j \left\{ \mathbb{E}_\omega \left[ \max_{x' \in X_s} \sum_{i=1}^{L} f_i^n(x') p_i^{n+1}(x) \left| f_j^n = f^* \right. \right] \right\} - \max_{x' \in X_s} \sum_{i=1}^{L} f_i^n(x') p_i^n
$$

$$
= \sum_{j=1}^{L} \left[ \int \omega \max_{x' \in X_s} \sum_{i=1}^{L} f_i^n(x') p_i^{n+1}(x,\omega) g(\omega) d\omega \right] p_j^n - \max_{x' \in X_s} \sum_{i=1}^{L} f_i^n(x') p_i^n,
$$

(5.12)

where $g(\omega) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{\omega^2}{2\sigma^2} \right)$ is the density of $W$, and $p_{i,j}^{n+1}(x,\omega)$ is the posterior probability of $f_i^n$ given $f_j^n = f^*$ with measurement noise $\omega$. According to Equation (5.11), $p_{i,j}^{n+1}(x,\omega)$ is
given by:

\[ p_{ij}^{n+1}(x, \omega) = \frac{\exp \left( - \frac{[f_i^n(x)+\omega-f_j^n(x)]^2}{2\tilde{\sigma}^2} \right) p_i^n}{\sum_{l=1}^{L} \exp \left[ - \frac{[f_j^n(x)+\omega-f_l^n(x)]^2}{2\tilde{\sigma}^2} \right] p_l^n}. \]  \hspace{1cm} (5.13)

Furthermore, for problems with a continuous \( \mathcal{X} \) space, especially when \( \mathcal{X} \) is multidimensional, we can extend the discrete KGDP policy to the continuous case as described in Chapter 4. The exact definition of the KG score in the continuous case is

\[ \nu_{KG-cont,n}(x) = \mathbb{E}_n \left[ \max_{x' \in \mathcal{X}} f_j^{n+1}(x') \right] - \max_{x' \in \mathcal{X}} f_j^n(x'). \]

This is computationally intractable since \( \mathcal{X} \) is continuous. Therefore, Chapter 4 proposes an approximation by considering the improvement of information on a discrete set. Let \( \mathcal{X}_s \) be a random discrete subset of \( \mathcal{X} \). Then

\[ \nu_{KG-cont,n}(x) \approx \nu_{KG-s,n}(x) = \mathbb{E}_n \left[ \max_{x' \in \mathcal{X}_s} f_j^{n+1}(x') \right] - \max_{x' \in \mathcal{X}_s} f_j^n(x'). \]

Note that this approximation has the same expression as the case with discrete alternatives in Chen et al. (2014) and Chapter 3. However, instead of evaluating the KG score for each alternative, at each iteration, we can calculate the gradient of \( \nu_{KG-s,n}(x) \) using the algorithm presented in Chapter 4. We then run gradient ascent with multi-starts in the \( \mathcal{X} \) space to find the \( x \) that maximizes \( \nu_{KG-s,n}(x) \) and measure it for the next experiment.

### 5.3.7 Overall Flowchart

Algorithm 6 presents our overall algorithm, which uses local approximation to generate some candidate functions, uses a certain policy to select alternatives to measure, updates the candidate functions and their probabilities, and estimates the optimal alternative \( \hat{x}^N \) from the approximated true function. Besides KG and continuous KG, we provide another two policies for comparison in Section 5.4, including: (1) Pure Exploration, which selects an
x in the continuous space $\mathcal{X}$ randomly, and (2) Max-Var, which selects $x$ with the largest variance from a discrete space $\mathcal{X}_s$. The flowchart for selecting an alternative is given in Algorithm 7.

**Algorithm 6 Overall Flowchart**

**Input:** Budget for measurements: $N$; Alternative space: $\mathcal{X}$; The policy to select $x^n$.

**Output:** The estimated optimal alternative: $\hat{x}^N$.

1: Initialize $f_i^0(x)$ with random $\theta_i$’s. Set $p_1^0 = \ldots = p_L^0 = \frac{1}{L}$.
2: for $n = 0$ to $N - 1$ do
3: \hspace{1em} $x^n = \text{select}_x(\text{policy}, \mathcal{X}, \{f_i^n\}, \{p_i^n\})$.
4: \hspace{1em} Take a measurement at $x^n$, and the output is $\hat{y}^{n+1}$.
5: \hspace{1em} major_update_flag $= 0$.
6: \hspace{1em} for $i = 1, 2, \ldots, L$ do
7: \hspace{2em} Lazy clustering: add $x^n$ to clustering result $i$ in a recursive way.
8: \hspace{2em} if $x^n$ creates a new cluster in clustering result $i$ then
9: \hspace{3em} major_update_flag $= 1$.
10: \hspace{2em} end if
11: end for
12: if major_update_flag $= 0$ then
13: \hspace{1em} Recalculate probabilities $\{p_i^n\}$.
14: end if
15: if major_update_flag $= 1$ or $|\{i : p_i^n < \epsilon_r\}| > L/2$ then
16: \hspace{1em} Run a major update of candidate functions, as in Algorithm 5.
17: end if
18: end for
19: return $\hat{x}^N = \arg\max_{x \in \mathcal{X}} p_i^N f_i^N(x)$.

**5.4 Empirical Results**

We present some experimental results in this section. We first illustrate the function approximation in a low dimensional problem in Section 5.4.1. We then show the quality of our estimated optimal alternative in a multidimensional unimodal problem in Section 5.4.2, and a multimodal one in Section 5.4.3. We discuss how we apply our algorithm to a real world materials science problem on carbon nanotube growth in Section 5.4.4.
Algorithm 7 \textit{select}_x(policy, \mathcal{X}, \{f^n_i\}, \{p^n_i\})

\textbf{Input:} Policy: for discrete \(x\): KG, Max-Var, and for continuous \(x\): Continuous KG, Pure Exploration; The alternative space: \(\mathcal{X}\); the candidate functions: \(\{f^n_i(x), i = 1, ..., L\}\); the probabilities: \(\{p^n_i, i = 1, ..., L\}\).

\textbf{Output:} The alternative to measure next: \(x^n\).

1: \textbf{switch} (policy)
2: \textbf{case} KG:
3: \hspace{1em} \(x^n = \text{argmax}_{x \in \mathcal{X}} \nu^{KGDP,n}(x)\).
4: \textbf{case} Continuous KG:
5: \hspace{1em} \(x^n = \text{argmax}_{x \in \mathcal{X}} \nu^{KG-s,n}(x)\).
6: \textbf{case} Pure Exploration:
7: \hspace{1em} \(x^n = \text{rand}(\mathcal{X})\).
8: \textbf{case} Max-Var:
9: \hspace{1em} \(x^n = \text{argmax}_{x \in \mathcal{X}} \sum_{j=1}^L p^n_j [f^n_j(x) - \bar{f}^n(x)]^2\).
10: \textbf{end switch}
11: \textbf{return} \(x^n\).

To evaluate the quality of the estimated optimal alternative \(\hat{x}^n\), we define a metric called the opportunity cost (OC), which is the difference between the true function values at the true optimal alternative \(x^*\) and at our estimated optimal alternative \(x^n\), i.e.,

\[ OC(n) = f^*(x^*) - f^*(x^n) = \max_{x \in \mathcal{X}} f^*(x) - f^*(\text{argmax}_{x \in \mathcal{X}} \bar{f}^n(x)) . \]

We normalize \(OC\) by the range of the true function, i.e,

\[ OC\% (n) = \frac{OC(n)}{\max_{x \in \mathcal{X}} f^*(x) - \min_{x \in \mathcal{X}} f^*(x)} . \]

To evaluate our function approximation, we calculate the error between \(f^*(x)\) and \(\bar{f}^n(x)\).

The error is defined as the normalized mean squared error. We refer to this metric as \(MSE\) for simplicity, given by

\[ MSE(n) = \sqrt{\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} [f^*(x) - \bar{f}^n(x)]^2} \]

\[ \text{(5.14)} \]

Note that we calculate \(MSE\) mainly as a sanity check, since minimizing \(OC\) is our primary goal.
Moreover, we run the experiments using different noise levels which are defined as the ratio of the standard deviation of the measurement noise (i.e., $\sigma$) to the range of the true function, namely $\frac{\sigma}{\max_{x \in X} f^*(x) - \min_{x \in X} f^*(x)}$. When we say the noise is 10%, we mean this ratio is 10%.

5.4.1 Function Approximation in a Two-dimensional Multimodal Problem

We first present our function approximation in a two-dimensional problem. The true function is defined as

$$f^*(x) = 100x_1 \cdot x_2^3 \cdot \exp(-x_1^2 - 0.5x_2^2). \quad (5.15)$$

The shape of the true function is shown in the left column of Figure 5.4. As we can see, there are two local maxima and two local minima. We use the following quadratic function as the local approximation model:

$$g(x; \theta) = \theta_1 x_1^2 + \theta_2 x_2^2 + \theta_3 x_1 x_2 + \theta_4 x_1 + \theta_5 x_2 + \theta_6.$$

Note that for any local region, the quadratic function is not a perfect fit of the true function, but our goal is to capture the general shape of the true function and find the true $x^*$. Figure 5.4 shows the evolution of the approximation $\bar{f}^n$ at certain times under the KG policy with 20% noise. As we can see, after about 20 measurements, we are able to give a decent estimate of $x^*$; after 35 measurements, we are able to recognize that there exist two local maximums and two local minimums; after 50 measurements, our estimate gets very close to the truth.
5.4.2 Multi-dimensional Unimodal Problem

In this section, we show our optimal learning results on a multi-dimensional unimodal benchmark function, which produces a wide range of unimodal asymmetric functions. The function, known as the newsvendor problem, is given by:

\[
f^*(x) = \sum_{i=1}^{k} \eta_{1,i} \mathbb{E} \left[ \min \left( x_i, (D - \sum_{j=1}^{i-1} x_j)^+ \right) \right] - \sum_{i=1}^{k} \eta_{2,i} x_i.
\]

The newsvendor function exhibits a shape that cannot be well approximated by linear or quadratic functions.

In our experiment, we set \( k = 4 \), and thus have a four-dimensional function as the truth. We try two different local models: (1) the linear function, i.e., \( g_{\text{lin}}(x; \theta) = \sum_{i=1}^{k} \theta_i x_i + \)
\( \theta_{k+1} \), and (2) the quadratic function without the cross terms, i.e.,
\[
g_{\text{quad}}(x; \theta) = \sum_{i=1}^{k} \theta_i x_i^2 + \sum_{i=1}^{k} \theta_{k+i} x_i + \theta_{2k+1}.
\]

The results of OC and MSE using the linear function as the local approximation is presented in Figure 5.5. The noise level is 20%. As we can see, the KG and continuous KG policies have steady OC decrease over time as we take measurements and they outperform both Max-Var and Pure Exploration. For MSE, KG and Max-Var are among the best, with continuous KG being slightly worse.

Figure 5.5: OC% and MSE of the four-dimensional newsvendor problem, approximated by linear functions. Noise level: 20%.

Figure 5.6 shows the results of OC and MSE using quadratic functions as the local model, with no noise and 20% of noise, respectively. Note that with the parametric functions only being accurate in small local regions, the problem with zero noise is not trivial. With zero noise, we can see that the OC and MSE both decrease especially with continuous

From left to right: (1) OC% under no noise; (2) MSE under no noise; (3) OC% under 20% noise; (4) MSE under 20% noise.

Figure 5.6: Results of the four-dimensional newsvendor problem, approximated by quadratic functions.
KG. However, with 20% of noise, OC and MSE hardly drop even after 100 measurements, highlighting the challenge brought by the noise when there is no globally accurate model.

Note that from Figures 5.5 and 5.6, we see that using the linear local models achieves better results than using quadratic models under the same noise setting. This is not surprising since the newsvendor function is closer to being linear than quadratic in most of the regions. This result highlights the challenge of overfitting using a local model with too many parameters.

Additionally, the results of a six-dimensional newsvendor problem are shown in Figure 5.7.

Figure 5.7: Results of the six-dimensional newsvendor problem, approximated by linear functions. Noise level: 20%.

5.4.3 Multi-dimensional Multimodal Problem

In this section, we move to testing on a multi-dimensional multimodal function, which is an extension of the two-dimensional multimodal function in Section 5.4.1. The true function is defined as

\[
f^*(x) = 100x_1 \cdot x_2^3 \cdot \exp\left(-x_1^2 - 0.5x_2^2\right) + 50x_3 \cdot x_4^3 \cdot \exp\left(-0.5x_3^2 - x_4^2 + 0.5x_4\right).
\]

(5.16)

Since the two-dimensional function in Section 5.4.1 already has (at least) 4 local regions, the above \(f^*(x)\) has at least 16 local regions. We use the quadratic function without cross terms as the local model, given by \(g(x; \theta) = \sum_{i=1}^4 \theta_i x_i^2 + \sum_{i=1}^4 \theta_{4+i} x_i + \theta_9.\)
Figure 5.8 shows the OC under 0 noise and 20% noise, respectively. As we can see, in both cases, the continuous KG outperforms the other three policies and its OC steadily decreases over time, which proves the effectiveness of our algorithm with the continuous KG policy in finding the optimal alternative.

![Figure 5.8: OC% of four-dimensional test function. Left: no noise; right: 20% noise.](image)

### 5.4.4 An Application: Carbon Nanotube Growth

We now consider an application in optimizing carbon nanotube growth (CNT), using data obtained from the Air Force Research Laboratory’s ARES project (Nikolaev et al. 2016). ARES is an autonomous research system that autonomously synthesizes CNTs, characterizes growth via an *in-situ* Raman spectrometer, and plans the next experiment based on experimental results. Growth of CNT is performed by chemical vapor deposition in which feedstock gasses \( \text{C}_2\text{H}_4, \text{H}_2, \) and \( \text{CO}_2 \) are introduced at specified partial concentrations into a growth chamber that contains catalyst nanoparticles (Kumar & Ando 2010). A specific temperature, ranging between 600\( K \) and 1200\( K \) is maintained during CNT growth. Through a system of various gas and surface chemical reactions, CNTs nucleate and grow on the catalyst.

The myriad kinetic processes involved include gas phase and surface reactions (Puretzky et al. 2005), catalyst nanoparticle dynamics such as dewetting, ripening (Moseler et al. 2010), and poisoning (Schünemann et al. 2011, Puretzky et al. 2005), and phase change phenomena.
(Jansen et al. 2016). The exact nature of such processes is poorly understood, and as such, capturing a complete global picture of the dynamics, as a function of temperature and feedstock gas concentrations, is a difficult task. As a result, we need to use simpler models to approximate the growth of CNTs in local regions of the design space.

In this problem, our goal is to maximize the concentration of CNT after a synthesis process of 300 seconds. The concentrations of \( \text{C}_2\text{H}_4 \), \( \text{H}_2 \) and \( \text{CO}_2 \), together with the temperature, constitute a four-dimensional design space. For the purpose of evaluating our algorithm, we define the true function \( f^*(x) \) by fitting from the ARES data (see Appendix 5.6.2 for more detail on the generation of \( f^*(x) \)), where \( f^* \) is the CNT concentration at the end of a noise-free synthesis given a design vector \( x \). We approximate \( f^*(x) \) using a linear function \( g(x; \theta) \) as the local model, namely \( g(x; \theta) = \sum_{i=1}^{4} \theta_i x_i + \theta_5 \), which is a natural and general model to locally approximate many unknown smooth functions.

We run simulations under 0 and 20% of noise, respectively. The results of OC% and MSE of both cases are shown in Figure 5.9. As we can see, using the local linear models returns very promising results especially when we use the continuous KG policy, in which the OC% decreases to a relatively low level after 200 experiments.

![Figure 5.9: OC% and MSE using the linear function \( g_{lin}(x; \theta_j) \) locally, with 0 noise (left) and 20% noise (right).](image)

### 5.5 Conclusion

Optimal learning for nonlinear parametric belief models, especially in a multidimensional continuous alternative space, has been a challenging problem. Previous methods usually...
assume we have a parametric function $f(x; \theta)$ that can describe the true function globally, where the expression of $f$ is known but $\theta$ is unknown. In this category, Chen et al. (2014) starts from a discrete set of $x$’s and a small set of $\theta$’s, and develops the Knowledge Gradient with Discrete Priors (KGDP) model. Chapter 3 proposes a resampling algorithm in the $\theta$ space to adaptively learn more probable $\theta$’s, and thus extends KGDP to a much larger $\theta$ space. Based on this, Chapter 4 presents a gradient-based algorithm that works in a multidimensional continuous $x$ space. These methods show strong performance, even for 10 or 20 dimensional $x$ and $\theta$ spaces. However, they all require that the true function can be well described by a parametric function $f(x; \theta)$ globally. In many cases, the true function can be too complicated to model exactly, while a simpler model that can well describe a local region might exist.

In this chapter, we focus on the optimal learning problem in which a global parametric belief model is unavailable, but the true function at any local region can be well described by some nonlinear parametric model. We use DC-RBF (Jamshidi & Powell 2016) and K-means to cluster data and identify local regions, and apply normalized RBF to aggregate local functions as a possible global approximation of the true function. We use various global approximations as our candidate functions, each corresponding to a different clustering result and consisting of different locally fitted models, and then adopt the Knowledge Gradient policy for discrete priors proposed and studied in Chen et al. (2014), Chapter 3, and Chapter 4 to select the alternatives to measure, out of a multidimensional continuous alternative space. Experiments on some synthetic test functions and a real materials science application show the significant performance of our algorithm.

There are two points worth noting. First, our algorithm is still a non-parametric algorithm. Hence, as with other non-parametric algorithms, it may suffer from the curse of dimensionality. Specifically, if there are too many local regions especially in a high dimensional alternative space, and if the measurement budget is not large enough, our estimate of the true function, $\bar{f}$, might be very different from the ground-truth $f^*$. This can be
seen in the OC results of the first several measurements in Figure 5.9. Second, increasing the complexity of the local model does not necessarily help with the overall performance, as shown in Figure 5.5 and Figure 5.6. Actually, our method may favor a simpler local model especially when each local region does not contain sufficient data points. Over all, our algorithm provides a powerful tool for optimal learning problems with multidimensional continuous alternative spaces, and it works the most effectively when there are not too many local regions such that the local models can be well approximated with a limited budget.

5.6 Appendix

5.6.1 Heuristics in Refitting Local Functions

There are two heuristics in implementing the logic of refitting \( \theta_{ij} \). First, to make the candidate function \( f_i^n \) smooth, we can include some nearby data points of other clusters when calculating \( SSE(\theta) \), but assign smaller weights to these data. For instance, for cluster \( j \), let \( X'_j = \{ x \in \{ x^0, ..., x^n \} : x \notin \) cluster \( j, and \|x - c_{ij}\| \leq 1.5D_T \}, \) and \( \hat{y}'(x) \) be the measurement result associated with any \( x \in X'_j \). Let \( \sum_{i=1}^{k_j} \left[ g(x_{ij}^i; \theta) - \hat{y}_{ij}^i \right]^2 + \frac{1}{10} \sum_{x \in X'_j} [g(x; \theta) - \hat{y}'(x)]^2 \), and then calculate \( p(\theta) \) according to Equation (5.8). This heuristic is also useful when cluster \( j \) contains too few data points.

Another heuristic is that, if there are enough data points in a cluster such that the fitting problem is overdetermined, and if solving the fitting problem is efficient (for example, if \( g(x; \theta) \) is linear in \( \theta \), \( \theta \) can be found using linear regression), then we can calculate \( \theta_{ij} \) directly instead of sampling it from the large pool. However, calculating \( \theta \) directly in an underdetermined case is not encouraged, especially if the fitting algorithm always returns some \( \theta \) out of a small range. This is because our candidate functions are supposed to have various shapes and capture the distribution of possible truths. Intuitively, if a local region contains only a few data points but the candidate functions \( f_i^n(x) \) containing this region all use the same \( g(x; \theta^n_j) \), we may overlook the uncertainty in this region and thus may
not measure any point in this region. One example is given in Figure 5.10, which shows the results of a one-dimensional problem under the Knowledge Gradient policy proposed in Section 5.3.6. When we only have one measurement as in “n = 1” in Figure 5.10, the fitting results using our resampling logic include a variety of curves, and the uncertainty on the right side is the largest. Hence, the KG policy then measures an alternative on the right, as in “n = 2”. Next, KG measures some x in the middle as in “n = 3”. However, if we fit L curves using only x₀ in “n = 1” rather than resample from the small pool, the fitted L curves might be too close to reflect our uncertainty on the right.

Figure 5.10: Evolution of \( \bar{f} \) (red curves in the first row) and candidate functions (thin curves in the second row) under KG policy in a one-dimensional problem under 20% noise.

Note: The thick black curve is the truth, given by \( f^*(x) = (x + 0.5)^3 - 30x - 13 \), approximated by quadratic functions, i.e, \( g(x; \theta) = \theta_1 x^2 + \theta_2 x + \theta_3 \). The blue dots are where we take measurements. Note that the images in the first row have the same scale, while the last 3 images in the second row have the same scale.

5.6.2 Generation of the True Function \( f^*(x) \) for Carbon Nanotube Growth

In this section, we discuss how we generate the true function \( f^*(x) \) for the purpose of evaluating our algorithm. As mentioned in Section 5.4.4, the CNT growth is a complicated process such that, to the best of our knowledge, there does not exist a standard model to
describe it globally. Therefore, we construct a true function \( f^*(x) \) by capturing the locally dominant kinetic processes of CNT growth, with the help of chemical reaction networks (CRNs).\(^1\) Basically, we use the ARES data (Nikolaev et al. 2016) to train a group of CRNs locally, and combine them together as the true function \( f^*(x) \).

A chemical reaction network (CRN) specifies the \( R \) reactions between \( S \) chemical species using reactant and product stoichiometry matrices, \( A \) and \( B \), respectively. Both \( A \) and \( B \) are \( R \times S \), sparse matrices whose entries are small, positive integers. Given these matrices, mass action kinetics determine the time evolution of the species’ concentrations \( X_s(t) \), through the dynamical system

\[
\frac{dX_s}{dt} = \sum_{r=1}^{R} (B_{r,s} - A_{r,s})k_r \prod_{u=1}^{S} X_{u}^{A_{r,u}}, \tag{5.17}
\]

where the \( k_r \) is the reaction rate

\[
k_r = \nu_r \exp \left[ -\frac{\epsilon_r}{k_B T} \right],
\]

where \( \nu_r \) and \( \epsilon_r \) are kinetic parameters called the reaction rate prefactor and energy barrier, respectively, \( k_B \) is Boltzmann’s constant and \( T \) is temperature.

From ARES, we obtain several experimental observations \( \mathcal{O} = \{(x^m, \mathcal{O}^m_s(t_n))\} \) where \( x^m \) is the growth conditions (gas flow rates and temperature) for the \( m \)-th experiment, and \( \mathcal{O}^m_s(t_n) \) is the observed concentration of the \( s \) species at time \( t_n \) for the \( m \)-th experiment. From these observations, local kinetic models are fit by a two-stage optimization procedure. In the outer procedure, the stoichiometric parameters are optimized by simulated annealing to minimize the loss function

\[
L_{\text{outer}}(A, B \mid \mathcal{O}) = \min_{\epsilon, \nu} L_{\text{inner}}(A, B, \epsilon, \nu \mid \mathcal{O}),
\]

\(^1\)The formulation of the CRNs in this problem is provided by Kristofer-Roy G. Reyes.
through perturbations of the matrices $A, B$ and application of Metropolis criterion to accept or reject such perturbations. The inner optimization, i.e. the minimization of $L_{\text{inner}}$ searches for the optimal kinetic parameters for a fixed stoichiometry using gradient descent, where

$$L_{\text{inner}}(A, B, \epsilon, \nu \mid \mathcal{O}) = \sum_{m=1}^{M} \sum_{s=1}^{S'} \sum_{n=1}^{N} \frac{1}{N} \frac{X_{s}^{m}(t_{n}; A, B, \epsilon, \nu) - O_{s}^{m}(t_{n})}{O_{s}^{m}(t_{n})},$$

where $X_{s}^{m}(t_{n}; A, B, \epsilon, \nu)$ is the concentration of the $s$-th species at time $t_{n}$ predicted by the model (5.17) given stoichiometric and kinetic parameters, and growth conditions $x^{m}$. That is, we minimize the mean relative error between model and observations. We use relative error because observed concentrations between species and experiments can differ by several orders of magnitude. We note here that $S'$, the number of species actually observed in experiments, is less than $S$, the total number of species being modeled. That is, we are presuming the existence of several intermediate species that are results of internal reactions that are not captured during an experiment.

By selecting experimental observations clustered around a local region, and fitting CRNs to these observations using the procedure outlined above, we obtain a local model for CNT growth. We denote the local model given by CRN as $g_{\text{crn}}(x; \theta)$. To generate the true function $f^{*}(x)$, we divide the design space into 20 local regions, use the ARES data to fit $g_{\text{crn}}(x; \theta)$ for each region, and stitch them together according to Equation (5.3), namely

$$f^{*}(x) = \sum_{i=1}^{20} w_{i}(x) g_{\text{crn}}(x; \theta_{i}^{n}). \quad (5.18)$$

With the true function $f^{*}(x)$ defined in Equation (5.18), we can now use the optimal learning techniques proposed in this chapter to find the optimal design $x$. Apparently, CRN is a natural choice of the local model. We construct candidate functions using a number of CRNs, and the results of OC and MSE using CRN locally is shown in Figure 5.11.
Figure 5.11: OC% and MSE using the chemical reaction network $g_{crn}(x; \theta_j)$ locally, with 0 noise.

Note that the results in Figure 5.11 indicate poor performance of learning with CRNs. This is not surprising since a CRN has too many parameters (i.e, $\theta$) and as such, training a CRN is highly underdetermined especially with a small amount of data. As a result, we turn to a lower-order and more general model, namely the linear function as the local model, which produces much better results as shown in Figure 5.9.
Chapter 6

Conclusions and Future Work

Optimal learning with nonlinear parametric belief models has been an important but challenging problem. It is important in that if this problem is solved, its solution can be applied to a broad variety of applications, including when the design variables (i.e., alternatives) are multidimensional and continuous. It is challenging in that it is difficult to model the distribution of the unknown parameters and the unknown function values; as a result, the Knowledge Gradient policy, which is a major class of optimal learning policies, cannot be easily applied because of computational issues. As a first attempt, Chen et al. (2014) proposes a simple model called Knowledge Gradient with Discrete Priors (KGDP). However, it can only handle a finite number of alternatives and is limited to a small sampled set of parameters.

In this thesis, we focus on cracking the limitations of KGDP to make it applicable to real-world multidimensional problems. We first consider problems that can be modeled by a globally accurate parametric function. We propose a resampling algorithm in the parameter space, which adaptively discovers more probable parameters to replace the less probable ones. We also prove the KGDP policy equipped with the resampling algorithm is asymptotically optimal in learning both the true parameter and the optimal alternative. We then introduce a method to calculate the knowledge gradient score and its derivative in a contin-
uous alternative space, and use gradient ascent to maximize the KG score. The continuous KG policy, combined with the resampling algorithm, is also shown to be asymptotically optimal in identifying both the true parameter and the best alternative. So far, we have a solution to problems that have a global parametric model with multidimensional continuous alternatives and parameters.

However, in reality, the underlying system is often too complicated or not fully understood, such that the problem cannot be modeled globally accurately. Alternatively, some local and low-order models may exist, which can well describe the shape of the true function within a local region. We propose a method to construct global representations of the true function using local models, and apply our continuous KG policy to this model. Experiments on a series of problems show the strong performance of our algorithm.

To summarize, our work has been focused on developing optimal learning algorithms for problems with nonlinear parametric belief models. The problem may be defined on a multidimensional continuous alternative space, while the parametric model can be either globally or locally accurate, involving possibly multidimensional unknown parameters. We hope our work can provide some solutions to similar problems, and our methodology could inspire more creative work in the future.

We close by listing a few thoughts on possible future work. Some of the ideas below are direct follow-ups of our algorithms, while others are from a broader horizon.

- Resampling in a fully continuous \( \theta \) space

In our resampling algorithm in Chapter 3, we create a large pool of parameters to resample from for mainly two reasons. First, we hope to avoid solving non-convex maximum likelihood problems. Second, this implementation further enables us to create a small pool, which provides a good tradeoff between exploration and exploitation. Experiments show that our method works well for multidimensional (say less than 10-dimensional) problems, as long as the large pool is a good representation of the pa-
rameter space. However, for problems with even higher dimensions, implementations that support resampling from a continuous space might be needed. One possible direction is to design better search algorithm (e.g, gradient-based methods); another is to fit a continuous distribution for parameters for the purpose of sampling, and resample from the fitted distribution.

- **KG for continuous $x$**

  The first follow-up question of the continuous KG policy is how to sample a set of discrete alternatives (namely $\mathcal{X}_s$ in Equation (4.1)) to better approximate KG. We currently sample randomly, but if we get to know the structure of the parametric function well, especially where the maximum of $\bar{f}$ might be, we may be able to achieve a better approximation.

Another question is how to decide the starting points in the gradient ascent when maximizing KG. In Scott et al. (2011), a continuous KG policy is built based on look-up tables using Gaussian Process Regression. A nice property of this model is that the previously measured alternatives become local minima of KG, which provides some insights on where to start from (on the other hand, this model creates too many local minima for KG, making it difficult to locate a global maximum). In our work with parametric models, such a property does not always exist. However, it is still possible that similar properties might exist, especially for problems with special structures.

- **Local approximations**

  One possibility to improve our method is to explore other local approximation techniques, which hopefully would produce more accurate approximations with a better representation of the uncertainty.

  Second, as pointed out in Chapter 5, the local approximation method we propose is nonparametric in nature and thus suffer from the curse of dimensionality. Experiments
show that our method works for a six-dimensional problem, but it is still challenging to extend it to higher (say 10) dimensional problems.

Third, we currently do not have any theoretical results for the local approximation work in Chapter 5. It would be interesting to see if some convergence results can be established for the KG with local approximation (possibly with modifications of the current algorithm).

- Convergence rate of KG with parametric models

  The convergence rate of KG has always been a challenging problem. It would be interesting to see with parametric models, whether it is possible to have some theoretical results on the rate of convergence.

- Fast calculation of KG

  It takes some computational time to find the maximum of KG, especially for parametric models. For instance, maximizing KG over discrete alternatives requires calculating the KG score for all alternatives, while in the continuous case, the gradient ascent may take a number of steps to converge. Although this is almost negligible compared to the cost of a measurement, a faster version of KG is still useful, especially in that it might have the potential to solve more complex problems. Some possible methods include developing approximations of KG, or using approximations for the underlying function.

- Batch learning with parametric models

  Another question is how to select alternatives when multiple measurements can be conducted simultaneously. Wang et al. (2015) proposes a KG policy for the look-up table models in the batch settings, while Wu & Frazier (2016) develops a policy for KG with Gaussian Process Regression (Scott et al. 2011). However, no work has been done for KG with parametric models under a batch learning setting.
• New methods in local approximation or kernel regression

Our work in Chapter 5 might inspire some new ideas on local approximation techniques and kernel regression methods. That is, instead of using one approximation specified by a fixed tunable parameter, we may construct a variety of approximations specified by a family of parameters, and use their weighted sum as the approximation. Moreover, our method also provides an approach to representing the distribution of the truth, in that the multiple candidate functions can be viewed as samples from the posterior distribution of the truth.

• Model errors

Besides the local approximation method we introduce in Chapter 5, another way to handle problems without an accurate global form is to start with an approximate parametric model and then model the discrepancy. One possible method might be using simpler models, say Gaussian Process Regression (GPR) or linear belief models, to model the discrepancy. For example, in Poloczek et al. (2016), the authors use GPR to capture the model discrepancy, although the settings are different from ours.

• Active learning

The area of active learning also considers the problem of iteratively querying new data. It might be worth considering using global or local parametric functions to represent a contribution function, and using KG to select new measurements.

• Applications

Last but not least, I hope the methods in this thesis can be applied to and benefit more real-world applications. Specifically, the areas that have many potential applications include materials science, simulation optimization, and others as mentioned in Chapter 1. Moreover, some algorithms that require parameter tuning, for example the training of support vector machines, may also benefit from our work.
Bibliography


Cheddie, D. & Munroe, N. (2005), ‘Review and comparison of approaches to proton exchange membrane fuel cell modeling’.


Hastie, T., Tibshirani, R. & Friedman, J. (2009), The Elements of Statistical Learning: Data Mining, Inference, and Prediction, number 2.


URL: [http://dx.doi.org/10.1039/C6CP04719J](http://dx.doi.org/10.1039/C6CP04719J)


**URL:** [http://dx.doi.org/10.1021/nn102118y](http://dx.doi.org/10.1021/nn102118y)


URL: [http://dx.doi.org/10.1038/npjcompumats.2016.31](http://dx.doi.org/10.1038/npjcompumats.2016.31)


URL: [http://dx.doi.org/10.1007/s00339-005-3256-7](http://dx.doi.org/10.1007/s00339-005-3256-7)


