Bayesian optimization for the design of tax and transfer experiments

Dennis Oscar Hein

Wednesday 30$^{\text{th}}$ June, 2021

Abstract

This paper considers the setting in which a Bayesian policy maker adaptively designs experiments to maximize ex-ante expected social welfare. This problem can be formulated as optimizing some expensive to evaluate black-box function, from which only noise corrupted observations are available, and thus falls within the domain of Bayesian optimization. An additional complication is that utility is unobservable and hence the social welfare function cannot be queried at arbitrary points in the search space. To circumvent this issue, this paper draws on insights from previous research which allows us to map observable experimental output into social welfare. We then approximate draws from the posterior distribution of social welfare using random Fourier features and negotiate the exploration-exploitation trade-off using Thompson sampling. We evaluate the suggested algorithm by running calibrated simulations of two policy problems: optimal subsidy rates in health insurance and optimal linear income taxation. Results in this paper indicate that the additional complication of mapping observable experimental output into social welfare has a negligible effect on convergence rates.

Keywords: Experimental design, Bayesian optimization, Thompson sampling, Gaussian process priors, Random Fourier features, Optimal policy

Acknowledgements

I would like to thank Dr. Maximilian Kasy for excellent supervision and support throughout this project. Special thanks for allowing me to utilize code developed when working with him as research assistant on a related project.

Contents

1 Introduction

This paper considers the setting in which a Bayesian policy maker adaptively designs experiments to maximize ex-ante expected social welfare. We can formulate the policy maker's problem as optimizing some expensive to evaluate black-box function, from which only noise corrupted observations are available. Queries to the black-box might be expensive computationally, or carry some monetary cost (e.g., cost to implement the experiment) (Frazier, 2018). Hence, sampling is to be minimized. Common approaches to this problem include the multi-armed bandit paradigm (see e.g., Russo and Van Roy (2016); Bubeck and Cesa-Bianchi (2012)), in which the objective is to maximize the cumulative rewards by negotiating the trade-off between exploitation and exploration, and experimental design (see e.g., Chaloner and Verdinelli (1995)), where the objective is to learn the unknown function globally in as few iterations as possible[1] (Srinivas et al., 2012). Since we are interested here in function optimization, we only wish to sample in areas of the search space until they can confidently be ruled out as suboptimal (Shahriari et al., 2016). The main intuition is that it might be wasteful to learn the function globally and we should instead concentrate sampling in areas that are likely to contain the global optima. More generally, algorithms can, and should, be adjusted for the particular objectives of the experiment, some recent examples include Kasy and Sautmann (2021) and Teytelboym et al. (2020). In this paper, we consider the multi-armed bandit problem where the prior distribution over the reward function is a Gaussian process. Gaussian processes have a wide variety of different application in machine learning tasks (see e.g., Rasmussen and Williams (2006)). In addition, Gaussian processes are intimately connected with, the perhaps more familiar, kernel methods based on reproducing kernel Hilbert spaces, as discussed extensively in Kanagawa et al. (2018). This multi-armed bandit setup is called Gaussian process optimization (see e.g., Srinivas et al. (2012); Desautels et al. (2014)), or, perhaps more colloquially, Bayesian optimization.[2] Bayesian optimization is a popular technique for optimizing expensive to evaluate, black-box,

[1] Despite the fact that the multi-armed bandits and experimental design approaches have very different objectives, theoretical analysis of popular Bayesian optimization algorithms show that these approaches are formally intimately connected. This is due to the fact that regret bounds are usually formed in terms of information gain (Srinivas et al., 2012; Kandasamy et al., 2018). More generally, information theoretic regret bounds are common in the multi-armed bandit literature (see e.g., Russo and Van Roy (2016)). For a textbook treatment of information theory see e.g., Cover and Thomas (2006).

[2] Note however, that Bayesian optimization is actually more general and it includes other surrogate models than Gaussian processes (see e.g., Shahriari et al. (2016)). We will, nevertheless, use these terms interchangeably throughout.

objective functions in the machine learning literature (see e.g., Shahriari et al. (2016); Frazier (2018)). It has recently received a lot of attention as it is a very sample efficient approach to this general problem. Some applications of Bayesian optimization include robotics (Lizotte et al., 2007), automated machine learning (Snoek et al., 2012), reinforcement learning (Brochu et al., 2010), sensor networks (Garnett et al., 2010), chemical engineering (Hernández-Lobato et al., 2017), game theory (Picheny et al., 2019; Aprem and Roberts, 2018), and economics (Aprem and Roberts, 2021).

The main issue faced in this paper, compared to standard Bayesian optimization, is that the social welfare function is measured in utility—an unobservable quantity. This implies that we cannot query the objective function at arbitrary points in the domain and thus the minimum requirement for Bayesian optimization is not satisfied. To circumvent this issue, we will use the setup in Kasy (2018) which combines insights from the literature on optimal policy theory and machine learning using Gaussian process priors to solve a general class of policy problems. Kasy (2018) assumes that the policy maker is Bayesian in the sense that she uses the available data to form a posterior expectation of social welfare. The optimal policy is then simply the one which maximizes this posterior expectation. Leveraging the envelope theorem (Milgrom and Segal, 2002), in a similar manner as in the sufficient statistics literature (see e.g., Chetty (2009)), Kasy (2018) shows how we can map observable responses into social welfare. Crucially, this map will be linear and thus placing a Gaussian process prior on the observable response function implies a Gaussian process for the social welfare function—the reward function of interest. Since we know the posterior distribution of social welfare, we can apply some search heuristic, or acquisition function, such as Thompson sampling (see e.g., Thompson (1933); Russo and Van Roy (2016); Russo et al. (2017); Kandasamy et al. (2018)) to the policy maker's problem. However, there is an additional complication since our search space is continuous, which implies that samples from this posterior are infinite dimensional objects.[3] One way to sidestep this issue is to sample from an analytic approximation of the posterior using Random Fourier Features.[4] This approximation is based on a result from harmonic analysis called Bochner's theorem (Bochner, 1959) which allows us to write the kernel as the expectation of a Fourier basis with respect to its corresponding spectral density (Shahriari et al., 2016).

Leveraging insights from Kasy (2018), Bayesian optimization using Thompson sampling, and

[3] See section 2.1.3 or discussion in Hernández-Lobato et al. (2014).

[4] This is a commonly used approximation technique for high dimensional kernel methods originally proposed by Rahimi and Recht (2007).

Random Fourier features, this paper extends the setup in Kasy (2018) to the setting of optimal policy using adaptive experimental design. We evaluate our suggested approach by running calibrated simulations of two policy problems: optimal subsidy rates in health insurance based on data from the RAND health insurance experiment and analysis in Kasy (2018), and optimal linear income taxation building on Mirrlees (1971), Diamond (1998), and Saez (2001).

This paper has numerous contributions. First, for empirical researchers, this paper extends the setup in Kasy (2018) to the setting of optimal policy using adaptive experimental design. Although the applications in this paper are health insurance and income taxation, the results here could easily be extended to a wide range of other policy issues that can be formulated in a similar way.[5] Second, we add to the literature on Bayesian optimization with an algorithm that optimizes an unobservable objective function that is estimated using a Gaussian process priors framework. In particular, we consider the case when the objective function is a linear operator applied to some underlying observable response function which is modelled using a Gaussian process. Hence, at each query there is an additional mapping which will transform any existing noise. Results in this paper indicate that the additional complication of mapping observable experimental output into social welfare has a negligible effect on convergence rates. Third, we show that by appropriately transforming the feature map utilized in random Fourier features, we can obtain an analytic approximation of the posterior of the unobservable objective function. This result extends to any setting where the objective is a linear operator applied to some observable response function.

The rest of the paper is organized as follows. In section 2 we will cover the main background to this paper. The aim of this section is to convey key concepts required to understand the content of this paper. In section 3 we will present our algorithmic approach to the design of tax and transfer experiments. In this section we will also cover the main setup used to combine optimal policy theory with machine learning as suggested in Kasy (2018). This is followed by calibrated simulations in section 4 where we evaluate the proposed method. Finally, in section 5 we will conclude and discuss some avenues of future research. Code used for implementing these methods is available at `https://github.com/dennishein/BOforthedesignoftaxandtransferexperiments`.

[5] In particular, it extends to the class of policy problems treated in Kasy (2018).

2 Preliminaries

2.1 Bayesian optimization

2.1.1 Introduction

Bayesian optimization is a global optimization technique used to optimize expensive to evaluate, black-box, functions. In particular, it is a machine learning based optimization technique that places a probabilistic surrogate[6] model on the objective function. The most common choice for surrogate model is by far a Gaussian process (see e.g., Rasmussen and Williams (2006) for a textbook treatment). An acquisition function, based on this surrogate model, is subsequently used to navigate the search space—trading off exploration (i.e. sample in areas of high uncertainty) and exploitation (i.e. sample where the posterior mean is high[7]). More formally, Bayesian optimization is concerned with the global optimization problem

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) \tag{1}$$

where f is the objective function and $\mathcal{X}$ some design space of interest. Most commonly, $\mathcal{X} \subseteq \mathbb{R}^d$ but other choices are also possible (Shahriari et al., 2016). The objective function is treated as a black-box in the sense that it can only be evaluated via queries that provide noisy output on the form $y_i \sim \mathcal{N}(f(\boldsymbol{x}_i), \sigma^2)$. In other words, we observe f only through noisy, albeit unbiased, point-wise observations y (Shahriari et al., 2016). Bayesian optimization solves (1) in a sequential manner. Let $\mathcal{D}_n := \{(\boldsymbol{x}_i, y_i))\}_{i=1}^n$ denote the available data after n iterations. The next query point $\boldsymbol{x}_{n+1}$ is then selected by maximizing some acquisition function $\alpha(\boldsymbol{x}; \mathcal{D}_n)$. The algorithm runs for N queries and returns a final recommendation $\boldsymbol{x}^*$ which is the algorithm's best approximation of the global optimizer. A generic version of Bayesian optimization, based on Frazier (2018) and Shahriari et al. (2016), is presented in algorithm 1. Bayesian optimization has recently received a lot of attention as it is a very sample efficient approach to the problem in (1). Some examples of applications include robotics (Lizotte et al., 2007), automated machine learning (Snoek et al., 2012), reinforcement learning (Brochu et al., 2010), sensor networks (Garnett et al., 2010), chemical engineering (Hernández-Lobato et al., 2017), game theory (Picheny et al., 2019; Aprem and Roberts, 2018), and economics (Aprem and Roberts, 2021).

[6] See e.g., Gramacy (2020) for a textbook treatment of surrogate modelling in the applied sciences.

[7] We treat the global maximization problem.

The standard performance metrics for Bayesian optimization is the same as in the multi-armed bandit literature, namely regret. Suppose that $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$. Then, for some choice $\boldsymbol{x}_n \in \mathcal{X}$, we incur instantaneous regret

$$r_n := f(\boldsymbol{x}^*) - f(\boldsymbol{x}_n). \tag{2}$$

The cumulative regret, R_N, after N iterations is then

$$R_N := \sum_{n=1}^{N} r_n$$

and the average regret is

$$\bar{R}_N := \frac{R_N}{N}.$$

An algorithm is said to have no-regret if $\bar{R}_N \to 0$ as $N \to \infty$. Since $r_n \geq 0, \bar{R}_N \to 0$ implies that $r_n \to 0$ as well; hence, there is an n such that $f(\boldsymbol{x}_n)$ is arbitrarily close to $f(\boldsymbol{x}^*)$ (Berkenkamp et al., 2019). In other words, the algorithm converges. We note that neither r_n nor R_N is revealed by the algorithm since we only have noisy observations $y_n = f(\boldsymbol{x}_n) + \epsilon$.

Algorithm 1:

```
Place surrogate model on $f$;
Query $f$ at $n_0$ points using initial space-filling design. Set $n = n_0$;
for $n = n_0 + 1, n_0 + 2, ..., N + n_0$ do
    Select next sampling point $\boldsymbol{x}_{n+1} = \arg\max_{\boldsymbol{x}} \alpha(\boldsymbol{x}; \mathcal{D}_n)$;
    Query objective $y_{n+1}$;
    Augment data $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (\boldsymbol{x}_{n+1}, y_{n+1})\}$;
    Update surrogate model;
end
return $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}(f(\boldsymbol{x})|\mathcal{D}_{N+n_0})$
```

2.1.2 Surrogate models

Although many different surrogate models are possible, the most prominent approach by far is to place a Gaussian process prior on f. A Gaussian process can be thought of as a generalization of the multivariate Gaussian distribution. In particular, a Gaussian process is defined as follows:

Definition 2.1. *(Rasmussen and Williams, 2006) A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

Let $\mathcal{GP}(\mu, k)$ denote a Gaussian process with mean function $\mu : \mathcal{X} \to \mathbb{R}$ and a positive-definitive kernel function, or covariance function, $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The kernel function encompasses smoothness assumptions on samples from the Gaussian process. The mean function, on the other hand, provides the possibility of offsets (Shahriari et al., 2016). In practice, the mean function is usually set to some constant—most commonly zero.[8] We take this approach as well and place a zero mean Gaussian process prior on f. Let $\boldsymbol{x} \in \mathcal{X}$ be some test point and $\mathcal{D}_n := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ some finite set of data then

$$f(\boldsymbol{x})|\mathcal{D}_n \sim \mathcal{N}(\mu_n(\boldsymbol{x}), K_n(\boldsymbol{x}))$$

with

$$\begin{aligned}\mu_n(\boldsymbol{x}) &= \underbrace{\boldsymbol{k}_n(\boldsymbol{x})^\top}_{1\times n} \underbrace{(\boldsymbol{K}_n + \sigma^2 I)^{-1}}_{n\times n} \underbrace{\boldsymbol{y}}_{n\times 1} \\ K_n(\boldsymbol{x}) &= \underbrace{k(\boldsymbol{x}, \boldsymbol{x})}_{1\times 1} - \underbrace{\boldsymbol{k}_n(\boldsymbol{x})^\top}_{1\times n} \underbrace{(\boldsymbol{K}_n + \sigma^2 I)^{-1}}_{n\times n} \underbrace{\boldsymbol{k}_n(\boldsymbol{x})}_{n\times 1}\end{aligned}$$

where $\boldsymbol{f} := [f(\boldsymbol{x}_1), ..., f(\boldsymbol{x}_n)]$, $\boldsymbol{y} := [y_1, ..., y_n]$, $(\boldsymbol{K}_n)_{i,j} := k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $\boldsymbol{k}_n(\boldsymbol{x}) := [k(\boldsymbol{x}, \boldsymbol{x}_1), ..., k(\boldsymbol{x}, \boldsymbol{x}_n)]$ is a $n \times 1$ vector of cross covariance terms. We can of course predict q points simultaneously, in which case

$$\begin{aligned}\mu_n(\boldsymbol{x}) &= \underbrace{\boldsymbol{k}_n(\boldsymbol{x})^\top}_{q\times n} \underbrace{(\boldsymbol{K}_n + \sigma^2 I)^{-1}}_{n\times n} \underbrace{\boldsymbol{y}}_{n\times 1} \\ K_n(\boldsymbol{x}) &= \underbrace{k(\boldsymbol{x}, \boldsymbol{x})}_{q\times q} - \underbrace{\boldsymbol{k}_n(\boldsymbol{x})^\top}_{q\times n} \underbrace{(\boldsymbol{K}_n + \sigma^2 I)^{-1}}_{n\times n} \underbrace{\boldsymbol{k}_n(\boldsymbol{x})}_{n\times q}.\end{aligned}$$

The choice of kernel function and its hyperparameters is crucial (Shahriari et al., 2016). For this paper, it suffices to consider only stationary kernels.[9] Let $\boldsymbol{\Lambda} := \mathrm{diag}(l_1^2, ..., l_d^2)$ and $\boldsymbol{\tau} := \boldsymbol{x}_1 - \boldsymbol{x}_2$ then the automatic relevance determination (ARD) kernel is[10]

$$k(\boldsymbol{\tau}) = \sigma_0^2 e^{-\frac{1}{2}\boldsymbol{\tau}^\top \boldsymbol{\Lambda}^{-1} \boldsymbol{\tau}}$$

[8] This is w.l.o.g. (Rasmussen and Williams, 2006).

[9] This is mainly due to simplicity and due to the fact that stationary kernels are by far the most common in the literature. In addition, as we will see below, Bochner's theorem, the key ingredient of random Fourier features, requires a stationary kernel. However, note that the kernel used in Kasy (2018) is not stationary.

[10] For a stationary kernel we have that $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = k(\boldsymbol{x}_1 - \boldsymbol{x}_2, \boldsymbol{0}) = k(\boldsymbol{\tau}, \boldsymbol{0}) =: k(\boldsymbol{\tau})$ for $\boldsymbol{\tau} := \boldsymbol{x}_1 - \boldsymbol{x}_2$.

where σ_0^2 is the prior variance. This is one of the most commonly used kernels in the literature. It is also called the squared exponential or radial basis function (RBF) kernel. Some authors reserve the term ARD kernel for the case when there is a learnable length scale parameter l_k for each dimension. Obviously, for the one dimensional case the RBF and ARD kernels are exactly the same. The smoothness assumption embedded in the RBF kernel is considerable and it is not always appropriate (see e.g., discussion in Snoek et al. (2012)). Another large class of commonly used stationary kernels is the Matérn kernel, which is parameterized by a smoothness parameter ν (Shahriari et al., 2016). The Matérn kernel is scaled such that we obtain the RBF kernel for the special case when $\nu \to \infty$ (Rasmussen and Williams, 2006). For simplicity, we will use the RBF kernel throughout. Using Gaussian process priors is additionally convenient since it yields an analytical expression for the marginal likelihood. In particular, the log marginal likelihood is simply

$$\log p(\boldsymbol{y}|\boldsymbol{x}_{1:n}, \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{K}_n + \sigma^2 I)^{-1}\boldsymbol{y} - \frac{1}{2}\log|\boldsymbol{K}_n + \sigma^2 I| - \frac{n}{2}\log(2\pi) \tag{3}$$

where $\boldsymbol{x}_{1:n} := \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$, $\boldsymbol{\theta} := [l_1, ..., l_d, \sigma_0^2, \sigma^2]$ and $|\cdot|$ denotes the determinant. The three terms of the log marginal likelihood all have intuitive interpretations. The first term measures how well the model fits the data by a Mahalanobis distance, the second term is a complexity penalty, and the final term is just a normalization constant (Shahriari et al., 2016). The gradient of (3) with respect to $\boldsymbol{\theta}$ for the special case of the ARD kernel is derived in appendix A.1. We can find the hyperparameters numerically using e.g., a quasi-Newton approach such as L-BFGS (see e.g., Nocedal and Wright (2006) for a textbook treatment).

2.1.3 Acquisition functions

For some given loss function—usually in the form of cumulative or simple regret—we could in theory solve (1) using a minimum expected risk framework (Shahriari et al., 2016). Nevertheless, in practice this is computationally intractable. To circumvent this issue, many effective heuristics have been developed. These heuristics are referred to as acquisition functions in the Bayesian optimization literature. In this section, we will review some commonly used acquisition functions. Following Shahriari et al. (2016), we split these into three different categories: improvement-based, optimistic, and information-based. We also note that several of these acquisition functions can, and have been, generalized to the batch setting where we in each iteration select Q query points.

Improvement-based Two improvement-based acquisition functions are probability of improvement (PI) (Kushner, 1964) and expected improvement (EI) (Mockus, 1975). PI simply measures the probability that the next point $\boldsymbol{x}$ leads to an improvement upon incumbent $y^* := \max_{i=1,\dots,n} y_i$. This is adaptive approach is used in practice since we cannot observe $f(\boldsymbol{x}^*)$ (Shahriari et al., 2016). Recall that $f(\boldsymbol{x})|\mathcal{D}_n \sim \mathcal{N}(\mu_n(\boldsymbol{x}), K_n(\boldsymbol{x}))$. Hence,

$$\alpha_{\text{PI}}(\boldsymbol{x}; \mathcal{D}_n) := \mathbb{P}[f(\boldsymbol{x}) > y^*] = \Phi\left(\frac{\mu_n(\boldsymbol{x}) - y^*}{\sqrt{K_n(\boldsymbol{x})}}\right)$$

where $\Phi(\cdot)$ is the standard normal CDF. If we move beyond the PI and also consider the amount of improvement upon incumbent, then we arrive at EI. Let I denote the improvement function, s.t.

$$I(\boldsymbol{x}, f(\boldsymbol{x}), \boldsymbol{\theta}) = (f(\boldsymbol{x}) - y^*)\mathbb{I}(f(\boldsymbol{x}) > y^*)$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. We can again exploit that $f(\boldsymbol{x})|\mathcal{D}_n$ is normally distributed to obtain an analytic expression, for $K_n(\boldsymbol{x}) > 0$,

$$\begin{aligned} \alpha_{\text{EI}}(\boldsymbol{x}; \mathcal{D}_n) :&= \mathbb{E}[I(\boldsymbol{x}, f(\boldsymbol{x}), \boldsymbol{\theta})] \\ &= (\mu_n(\boldsymbol{x}) - y^*)\Phi\left(\frac{\mu_n(\boldsymbol{x}) - y^*}{\sqrt{K_n(\boldsymbol{x})}}\right) + \sqrt{K_n(\boldsymbol{x})}\varphi\left(\frac{\mu_n(\boldsymbol{x}) - y^*}{\sqrt{K_n(\boldsymbol{x})}}\right) \end{aligned}$$

where $\varphi(\cdot)$ is the standard normal PDF (Shahriari et al., 2016). Convergence rates for EI were provided in Bull (2011).

Optimistic Another commonly used heuristic is that of the upper confidence bound (UCB). This dates back to the seminal work of Lai and Robbins (1985) in the multi-armed bandit setting. The key principle behind the UCB is optimism in the face of uncertainty. An UCB algorithm for Bayesian optimization with provable regret bounds was introduced in Srinivas et al. (2012). As with the improvement-based acquisition functions, UCB is based on the posterior $f(\boldsymbol{x})|\mathcal{D}_n \sim \mathcal{N}(\mu_n(\boldsymbol{x}), K_n(\boldsymbol{x}))$. In particular,[11]

$$\alpha_{\text{UCB}}(\boldsymbol{x}; \mathcal{D}_n) := \mu_n(\boldsymbol{x}) + \beta_n\sqrt{K_n(\boldsymbol{x})}$$

[11] We formulate this as in Shahriari et al. (2016).

where the hyperparameter β_n is set optimally according to results in Srinivas et al. (2012). Recall that the next query point is selected as

$$\boldsymbol{x}_{n+1} = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \alpha(\boldsymbol{x}; \mathcal{D}_n)$$

which, in the case of UCB, is

$$\boldsymbol{x}_{n+1} = \arg\max_{\boldsymbol{x} \in \mathcal{X}} \mu_n(\boldsymbol{x}) + \beta_n \sqrt{K_n(\boldsymbol{x})}.$$

This clearly illustrates how the UCB negotiates the exploitation-exploration trade-off. Exploitation is sampling where the posterior mean $\mu_n(\boldsymbol{x})$ is high and exploration is sampling where there is high posterior uncertainty (i.e. where the posterior variance $K_n(\boldsymbol{x})$ is high).

Information-based Information based acquisition functions include Thompson sampling and predictive entropy search (PES) (Hernández-Lobato et al., 2014; Shah and Ghahramani, 2015). The perhaps simplest of these is Thompson sampling. Although Thompson sampling is not a new idea, dating back to the seminal work of Thompson (1933), it has over the last decades gained renewed attention as an effective heuristic for multi-armed bandit problems (see e.g., Russo et al. (2017) for a tutorial treatment) and Bayesian optimization (see e.g., Shahriari et al. (2016); Kandasamy et al. (2018)). At each iteration the next query point is selected based on the posterior probability that it is optimal. For instance, suppose $\mathcal{X}$ is restricted to a finite number of points m, then the latent function f would be an m dimensional vector. This would be the case for multi-armed bandit problems and Bayesian optimization where we have discretized the search space. Thompson sampling is then implemented through the following generative procedure: (i) draw a sample from the posterior $f|\mathcal{D}_n$ and (ii) return index with highest value (Hernández-Lobato et al., 2014). In the case of a Gaussian process prior, sample

$$f^{(n)}(\boldsymbol{x}) \sim \mathcal{N}(\mu_n(\boldsymbol{x}), K_n(\boldsymbol{x}))$$

and select the next query point as

$$\boldsymbol{x}_{n+1} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} f^{(n)}(\boldsymbol{x}).$$

For a continuous domain $\mathcal{X}$, f would be an infinite dimensional object. To circumvent this issue, we can sample from an analytic approximation of the posterior based on random Fourier features (Hernández-Lobato et al., 2014). In particular, as shown in the next section, we can approximate a sample from the posterior of f by the linear model

$$f^{(n)}(\boldsymbol{x}) = \phi^{(n)}(\boldsymbol{x})^\top \boldsymbol{\omega}^{(n)}$$

where $\boldsymbol{W}_i \sim p(\boldsymbol{w})$ and $\boldsymbol{b}_i \sim \mathcal{U}(0, 2\pi)$. Here $p(\boldsymbol{w})$ is the so-called spectral density and $\phi(\cdot)$ is the feature map suggested in Rahimi and Recht (2007). The next query point is then simply

$$\boldsymbol{x}_{n+1} = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathcal{X}} \phi^{(n)}(\boldsymbol{x})^\top \boldsymbol{\omega}^{(n)}.$$

This is the acquisition function that will be used in this paper.

2.2 Random Fourier features

In this section we generalize k s.t. it maps into $\mathbb{C}$ instead of $\mathbb{R}$, i.e. $k : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$. Suppose that k is a stationary kernel s.t. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = k(\boldsymbol{x}_1 - \boldsymbol{x}_2, \boldsymbol{0}) = k(\boldsymbol{\tau}, \boldsymbol{0}) =: k(\boldsymbol{\tau})$ for $\boldsymbol{\tau} := \boldsymbol{x}_1 - \boldsymbol{x}_2$. The foundation of random Fourier features, first introduced in Rahimi and Recht (2007), is a result from harmonic analysis due to Bochner (Bochner, 1959). We quote the statement of the theorem from Rasmussen and Williams (2006):

Theorem 2.1. *(Bochner's theorem (Rasmussen and Williams, 2006)) A complex-valued function k on $\mathbb{R}^d$ is the covariance function of a weakly stationary mean square continuous complex-valued random process on $\mathbb{R}^d$ if and only if it can be represented as*

$$k(\boldsymbol{\tau}) = \int_{\mathbb{R}^d} e^{2\pi i \boldsymbol{w}^\top \boldsymbol{\tau}} d\mu(\boldsymbol{w})$$

where μ is a positive finite measure.

Let $s(\boldsymbol{w})$ denote the density of μ, which is called the spectral density of k. If the spectral density exists, then the covariance function k and spectral density s are Fourier duals

$$k(\boldsymbol{\tau}) = \int s(\boldsymbol{w}) e^{2\pi i \boldsymbol{w}^\top \boldsymbol{\tau}} d\boldsymbol{w} \tag{4}$$

$$s(\boldsymbol{w}) = \int k(\boldsymbol{\tau}) e^{-2\pi i \boldsymbol{w}^\top \boldsymbol{\tau}} d\boldsymbol{\tau}. \tag{5}$$

This result is called the Weiner-Khintchine theorem (Rasmussen and Williams, 2006). (4) and (5) can alternatively be written as

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \int e^{-i\boldsymbol{w}^\top(\boldsymbol{x}_1 - \boldsymbol{x}_2)} s(\boldsymbol{w}) d\boldsymbol{w} \tag{6}$$

$$s(\boldsymbol{w}) = \frac{1}{(2\pi)^d} \int e^{i\boldsymbol{w}^\top \boldsymbol{\tau}} k(\boldsymbol{\tau}, \boldsymbol{0}) d\boldsymbol{\tau}. \tag{7}$$

This form of the Fourier transform is more common in the physics literature[12] and will be more convenient for our purposes. Note that the spectral density is positive and bounded. Hence, we can normalize it to obtain a proper probability density. In particular, we ensure this by letting $p(\boldsymbol{w}) := s(\boldsymbol{w})/\alpha$ where $\alpha := \int s(\boldsymbol{w}) d\boldsymbol{w}$. Hence, we can write the kernel as the expectation of the Fourier basis with respect to its spectral density $p(\boldsymbol{w})$ (Shahriari et al., 2016)

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \alpha \mathbb{E}_{p(\boldsymbol{w})} e^{-i\boldsymbol{w}^\top(\boldsymbol{x}_1 - \boldsymbol{x}_2)}.$$

Moreover, following Hernández-Lobato et al. (2014), we note further that[13]

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \alpha \mathbb{E}_{p(\boldsymbol{w})} e^{-i\boldsymbol{w}^\top(\boldsymbol{x}_1 - \boldsymbol{x}_2)} = 2\alpha \mathbb{E}_{p(\boldsymbol{w}, b)} \cos(\boldsymbol{w}^\top \boldsymbol{x}_1 + b) \cos(\boldsymbol{w}^\top \boldsymbol{x}_2 + b). \tag{8}$$

Averaging over m samples we get

$$\begin{aligned} k(\boldsymbol{x}_1, \boldsymbol{x}_2) &= \frac{2\alpha}{m} \mathbb{E}_{p(\boldsymbol{w}, b)} \cos(\boldsymbol{W}\boldsymbol{x}_1 + \boldsymbol{b})^\top \cos(\boldsymbol{W}\boldsymbol{x}_2 + \boldsymbol{b}) \\ &= \mathbb{E}_{p(\boldsymbol{w}, b)} \phi(\boldsymbol{x}_1)^\top \phi(\boldsymbol{x}_2) \\ &\approx \phi(\boldsymbol{x}_1)^\top \phi(\boldsymbol{x}_2) \end{aligned}$$

where $\phi(\boldsymbol{x}) := \sqrt{2\alpha/m} \cos(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$ and $\boldsymbol{W}$ and $\boldsymbol{b}$ are m stacked samples from $p(\boldsymbol{w})$ and $\mathcal{U}(0, 2\pi)$. In other words, we approximate the expectation via Monte Carlo estimation using the m draws from the spectral density (Shahriari et al., 2016). We now proceed, as in Hernández-Lobato et al. (2014), to show heuristically the equivalence between a Bayesian linear model using random features ϕ and a Gaussian process with kernel k. Consider the linear model

$$f(\boldsymbol{x}) = \phi(\boldsymbol{x})^\top \boldsymbol{\omega}, \boldsymbol{\omega} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}).$$

[12] See `https://mathworld.wolfram.com/FourierTransform.html` for a brief discussion.

[13] We show this in appendix A.2.

Conditional on data $\mathcal{D}_n := \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ we have that $\boldsymbol{\omega}|\mathcal{D}_n \sim \mathcal{N}(\boldsymbol{m}_n, \boldsymbol{V}_n)$ with

$$\boldsymbol{m}_n = (\Phi_n^\top \Phi_n + \sigma^2 I)^{-1} \Phi_n^\top \boldsymbol{y} \tag{9}$$

$$\boldsymbol{V}_n = (\Phi_n^\top \Phi_n + \sigma^2 I)^{-1} \sigma^2 \tag{10}$$

where $\Phi_n^\top$ is an $m \times n$ matrix and $\boldsymbol{y}$ an $n \times 1$ column vector s.t. $\Phi_n = [\phi(\boldsymbol{x}_1), ..., \phi(\boldsymbol{x}_n)]$, and $\boldsymbol{y}_i \sim \mathcal{N}(f(\boldsymbol{x}_i), \sigma^2)$, respectively (Hernández-Lobato et al., 2014). We can also write out the posterior predictive for some $\boldsymbol{x} \in \mathcal{X}$ which is Gaussian with mean and variance

$$\begin{aligned}
\mu_n(\boldsymbol{x}) &= \phi(\boldsymbol{x})^\top (\Phi_n^\top \Phi_n + \sigma^2 I)^{-1} \Phi_n^\top \boldsymbol{y} && (11)\\
&= \phi(\boldsymbol{x})^\top \boldsymbol{m}_n \\
K_n(\boldsymbol{x}) &= \phi(\boldsymbol{x})^\top (\Phi_n^\top \Phi_n + \sigma^2 I)^{-1} \phi(\boldsymbol{x}) \sigma^2 && (12)\\
&= \phi(\boldsymbol{x})^\top \boldsymbol{V}_n \phi(\boldsymbol{x}).
\end{aligned}$$

Since $\Phi_n^\top$ is an $m \times n$ matrix, $(\Phi_n^\top \Phi_n + \sigma^2 I)^{-1}$ requires the inversion of a $m \times m$ matrix. In the case where $n < m$ it would be preferable to invert an $n \times n$ matrix. We can achieve this by rewriting (11) and (12). First, we rewrite the posterior mean as a function of the inverse of an $n \times n$ matrix by noting that[14]

$$(\Phi_n^\top \Phi_n + \sigma^2 I)^{-1} \Phi_n^\top = \Phi_n^\top (\Phi_n \Phi_n^\top + \sigma^2 I)^{-1}. \tag{13}$$

Hence, we can rewrite (11) as

$$\mu_n(\boldsymbol{x}) = \phi(\boldsymbol{x})^\top \Phi_n^\top (\Phi_n \Phi_n^\top + \sigma^2 I)^{-1} \boldsymbol{y}.$$

To obtain $K_n(\boldsymbol{x})$ as a function of the inverse of a $n \times n$ matrix we apply the Woodbury formula (see appendix A.6) to $(\Phi_n^\top \Phi_n + \sigma^2 I)^{-1}$

$$\begin{aligned}
(\Phi_n^\top \Phi_n + \sigma^2 I)^{-1} &= \sigma^{-2} I + \sigma^{-2} \Phi_n^\top (\Phi_n \Phi_n^\top \sigma^{-2} + I)^{-1} \Phi_n \sigma^{-2} \\
&= \sigma^{-2} I + \sigma^{-2} \Phi_n^\top (\Phi_n \Phi_n^\top + \sigma^2 I)^{-1} \Phi_n. && (14)
\end{aligned}$$

[14] We show this in appendix A.2.

Plugging (14) into (12) yields the desired result

$$K_n(\boldsymbol{x}) = \phi(\boldsymbol{x})^\top \phi(\boldsymbol{x}) - \phi(\boldsymbol{x})^\top \Phi_n^\top (\Phi_n \Phi_n^\top + \sigma^2 I)^{-1} \Phi_n \phi(\boldsymbol{x}).$$

Now $(\Phi_n \Phi_n^\top + \sigma^2 I)^{-1}$ is $n \times n$ which results in considerable computational savings if $n << m$. Moreover, recall that $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \mathbb{E}_{p(\boldsymbol{w},b)} \phi(\boldsymbol{x}_1)^\top \phi(\boldsymbol{x}_2) \approx \phi(\boldsymbol{x}_1)^\top \phi(\boldsymbol{x}_2)$. Using this result we can see that

$$\begin{aligned} \mu_n(\boldsymbol{x}) &= \underbrace{\phi(\boldsymbol{x})^\top \Phi_n^\top}_{\approx \boldsymbol{k}_n(\boldsymbol{x})^\top} \underbrace{(\Phi_n \Phi_n^\top + \sigma^2 I)^{-1}}_{\approx (\boldsymbol{K}_n + \sigma^2 I)^{-1}} \boldsymbol{y} \\ K_n(\boldsymbol{x}) &= \underbrace{\phi(\boldsymbol{x})^\top \phi(\boldsymbol{x})}_{\approx k(\boldsymbol{x},\boldsymbol{x})} - \underbrace{\phi(\boldsymbol{x})^\top \Phi_n^\top}_{\approx \boldsymbol{k}_n(\boldsymbol{x})^\top} \underbrace{(\Phi_n \Phi_n^\top + \sigma^2 I)^{-1}}_{\approx (\boldsymbol{K}_n + \sigma^2 I)^{-1}} \underbrace{\Phi_n \phi(\boldsymbol{x})}_{\approx \boldsymbol{k}_n(\boldsymbol{x})}. \end{aligned}$$

Taking expectations of these inner products between features would yield an exact result (Hernández-Lobato et al., 2014). Hence, we can approximate a sample from the posterior of f by the linear model $f(\boldsymbol{x}) = \phi(\boldsymbol{x})^\top \boldsymbol{\omega}$. In particular, we draw m samples $\boldsymbol{W}_i \sim p(\boldsymbol{w}), \boldsymbol{b}_i \sim \mathcal{U}(0, 2\pi)$ and a sample $\boldsymbol{\omega}^{(n)}$ from $\boldsymbol{\omega} | \mathcal{D}_n$ by[15]

$$\boldsymbol{\omega}^{(n)} = \boldsymbol{m}_n^{(n)} + \mathrm{chol}(\boldsymbol{V}_n^{(n)}) \boldsymbol{\epsilon}^{(n)}, \quad \boldsymbol{\epsilon}^{(n)} \sim \mathcal{N}(\boldsymbol{0}, I).$$

An approximate sample from the posterior of f is then

$$f^{(n)}(\boldsymbol{x}) = \phi^{(n)}(\boldsymbol{x})^\top \boldsymbol{\omega}^{(n)}.$$

Note that we are required to sample from $p(\boldsymbol{w})$. This normalized spectral density will depend on the particular choice of kernel as per Bochner's theorem. We derive it for the ARD kernel in appendix A.2.

[15] For $n < m$ we do this step efficiently as described in the appendix of Seeger (2008).

3 Adaptive experimental design

3.1 Setup

This section is a summary of the in setup Kasy (2018). As in that paper, we present the more general approach in the context of optimal coinsurance rates in health insurance.[16] Suppose that individual i has healthcare expenditure y_i and that t_i denotes the fraction of the expenditure covered by the insurance (i.e. the subsidy rate). In other words, the coinsurance rate is $1 - t_i$. Given a change in the subsidy rate, the individual i may adjust their expenditure. This response is captured by the structural equation

$$y_i = g(t_i, \epsilon_i) \tag{15}$$

where ϵ_i is a random variable capturing any unobserved heterogeneity. ϵ_i is also assumed to be invariant to counterfactual policies. For $t_i = t$ and taking the expectation over the distribution ϵ_i yields the average structural function

$$m(t) = \mathbb{E}[g(t, \epsilon_i)] \tag{16}$$

which is assumed to be differentiable. We can interpret (16) as the average healthcare expenditure if all individuals i face policy t (Kasy, 2018). As in the sufficient statistics literature[17], Kasy (2018) notes that a marginal change in policy t will affect health care expenditure in two ways: mechanically and due to behavioral responses. Consider first the insurance provider's per person expenditure for policy level t which is

$$tm(t). \tag{17}$$

Given a marginal change dt in t the mechanical effect on (17) is $m(t)dt$. This effect is just accounting and does not require estimation of a causal effect (Kasy, 2018). The behavioral effect of dt on (17) is $tm'(t)dt$, where $m'(\cdot)$ denotes the derivative of $m(\cdot)$. In contrast, this quantity does require estimation of a causal effect (Kasy, 2018). Now consider the welfare of the insured. Given a marginal change dt there is again a mechanical effect $m(t)dt$. However, due to

[16] The setup in Kasy (2018) treats a wide class of policy problems. We focus on this particular example for the sake of clarity and since we will apply our algorithmic approach to this example in section 4.1.

[17] For a survey treatment see Chetty (2009).

so-called envelope theorem (Milgrom and Segal, 2002), the behavioral responses do not affect private welfare under standard utilitarian assumptions[18] (Kasy, 2018). The trade-off between the policy maker's two objectives is mediated by the parameter $\lambda > 1$ which represents both the marginal value of an additional dollar transferred to the sick relative to expenditure and private risk aversion (Kasy, 2018). λ is, for simplicity, assumed to be known. Adding up these effects, weighted by λ, yields

$$du = \left(\lambda m(t) - m(t) - t\frac{\partial}{\partial t}m(t)\right) dt$$

which can be rewritten as

$$u'(t) = \lambda m(t) - \frac{\partial}{\partial t}\left(tm(t)\right). \tag{18}$$

Integrating (18), and normalizing such that $u(0) = 0$, results in the social welfare function

$$u(t) = \lambda \int_0^t m(x)dx - tm(t) \tag{19}$$

which, for $\lambda = 1$, is a variant of the Harberger triangle (Kasy, 2018). This is the objective function of interest for the policy maker. Now, suppose that we can observe i.i.d. draws (y_i, t_i) and that t_i is randomly assigned such that $t_i \perp\!\!\!\perp \epsilon_i \; \forall i$. This implies

$$\mathbb{E}[y_i|t_i = t] = \mathbb{E}[g(t, \epsilon_i)|t_i = t] = \mathbb{E}[g(t, \epsilon_i)] = m(t).$$

Furthermore we assume that

$$y_i|t_i = t \sim \mathcal{N}(m(t), \sigma^2).$$

Kasy (2018) places a Gaussian process prior on $m(\cdot)$

$$m(\cdot) \sim \mathcal{GP}(\mu(\cdot), k(\cdot, \cdot))$$

where $\mathbb{E}[m(t)] = \mu(t)$ and $\mathrm{cov}(m(t_1), m(t_2)) = k(t_1, t_2)$ are the mean function and kernel, or covariance function, respectively. We will assume that $\mu(t) = 0 \; \forall t$.[19] Let $\boldsymbol{y} := [y_1, ..., y_n], \boldsymbol{t} := [t_1, ..., t_n]$ and $\mathcal{D}_n := \{(t_i, y_i)\}_{i=1}^n$ then

$$m(t)|\mathcal{D}_n \sim \mathcal{N}(\mu_n(t), K_n(t))$$

[18] This is discussed further in the appendix of Kasy (2018) and in Chetty (2009).

[19] Recall from above that this is w.l.o.g. (Rasmussen and Williams, 2006). This assumption is also invoked in Kasy (2018).

with

$$\mu_n(t) = \mathbb{E}[m(t)|\boldsymbol{y}, \boldsymbol{t}] = \boldsymbol{k}_n(t)^\top (\boldsymbol{K}_n + \sigma^2 I)^{-1} \boldsymbol{y} \tag{20}$$

$$K_n(t) = \text{var}(m(t)|\boldsymbol{y}, \boldsymbol{t}) = k(t,t) - \boldsymbol{k}_n(t)^\top (\boldsymbol{K}_n + \sigma^2 I)^{-1} \boldsymbol{k}_n(t)$$

where $\boldsymbol{k}_n(t) := [k(t,t_1), ..., k(t,t_n)]$ and $(\boldsymbol{K}_n)_{i,j} = k(t_i, t_j)$. As mentioned above, the objective function of interest is the social welfare function. The key observation in Kasy (2018) is that since $u(\cdot)$ is a linear transformation of $m(\cdot)$, we have that $u(\cdot)$ is also a Gaussian process where

$$\upsilon(t) = \lambda \int_0^t \mu(x) dx - t\mu(t) = 0$$

and

$$d(t_1, t_2) = \lambda \int_0^{t_1} k(x, t_2) dx - t_1 k(t_1, t_2). \tag{21}$$

Hence,

$$u(t)|\mathcal{D}_n \sim \mathcal{N}(\upsilon_n(t), D_n(t))$$

with

$$\upsilon_n(t) = \mathbb{E}[u(t)|\boldsymbol{y}, \boldsymbol{t}] = \boldsymbol{d}_n(t)^\top (\boldsymbol{K}_n + \sigma^2 I)^{-1} \boldsymbol{y} \tag{22}$$

$$D_n(t) = \text{var}(u(t)|\boldsymbol{y}, \boldsymbol{t}) = d(t,t) - \boldsymbol{d}_n(t)^\top (\boldsymbol{K}_n + \sigma^2 I)^{-1} \boldsymbol{d}_n(t)$$

where $\boldsymbol{d}_n(t) := \text{cov}(u(t), \boldsymbol{y}|\boldsymbol{t}) = [d(t,t_1), ..., d(t,t_n)]$ and

$$d(t,t) = \text{var}(u(t)|\boldsymbol{t}) = \lambda^2 \int_0^t \int_0^t k(x_1, x_2) dx_1 dx_2 - 2\lambda t \int_0^2 k(x,t) dx + t^2 k(t,t).$$

The posterior expected welfare (22) connects the optimal policy and Gaussian process prior setups (Kasy, 2018). A closed form expression of (21) for the ARD kernel is obtained in appendix A.3. Since the policy maker is assumed to be Bayesian, the optimal policy will be the t which maximizes the posterior expected social welfare (Kasy, 2018). In other words,

$$t^* = t^*(\boldsymbol{y}, \boldsymbol{t}) = \arg\max_t \upsilon_n(t)$$

where $\upsilon_n(t)$ is given in (22). The first order condition yields a simple expression and the optimal t can be found numerically by e.g., Newton-Raphson. For an application of this framework using data from the RAND health insurance experiment, a discussion on how this techniques compares the sufficient statistics approach, and several extensions, see appendix B.1 and Kasy (2018).

3.2 Algorithm

In this section, the optimal policy setup using Gaussian process priors from the preceding section, Bayesian optimization from section 2.1, and approximate samples using random Fourier features from section 2.2, will come together in an algorithm for optimal policy using adaptive experimental design. Recall that our objective is to find

$$t^* = \underset{t \in [0,1]}{\arg\max}\, u(t) \tag{23}$$

where $u(\cdot)$ is the social welfare function and $t \in [0, 1]$ is the tax. In contrast to standard Bayesian optimization, where we would just place a Gaussian process prior on $u(\cdot)$ directly and obtain the posterior expectation using unbiased noisy observations of $u(\cdot)$, we cannot query $u(\cdot)$ directly since utility is an unobservable quantity. This is where we can leverage the framework developed in Kasy (2018) which allows us to map the observable response function $m(\cdot)$ into social welfare by (19). Recall from section 2.2 that we can obtain an approximate sample from the posterior of $m(\cdot)$ using the linear model

$$m(t) = \phi(t)^\top \boldsymbol{\omega},\ \boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

for $\phi(t) := \sqrt{2\alpha/m}\cos(\boldsymbol{W}t + \boldsymbol{b})$ where $\boldsymbol{W}$ and $\boldsymbol{b}$ are m stacked samples from $p(\boldsymbol{w})$, the normalized spectral density of $k(\cdot,\cdot)$, and $\mathcal{U}(0, 2\pi)$. We can now leverage (19) to obtain approximate samples from the posterior of $u(\cdot)$, i.e. the posterior social welfare. In particular, due to the fact that $u(\cdot)$ is a linear transformation of $m(\cdot)$, we can simply apply the transformation in (19) to the feature map ϕ and obtain an approximate sample from the posterior of $u(\cdot)$ by

$$u(t) = \phi_u(t)^\top \boldsymbol{\omega},\ \boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

where

$$\phi_u(t) := \lambda \int_0^t \phi(x)dx - t\phi(t) \tag{24}$$

is the transformed feature map. Through similar argumentation as in section 2.2 we can see that

$$\phi_u(t)^\top \Phi_n^\top (\Phi_n \Phi_n^\top + \sigma^2 I)^{-1} \boldsymbol{y} \approx \boldsymbol{d}_n(t)^\top (\boldsymbol{K}_n + \sigma^2 I)^{-1} \boldsymbol{y} =: \upsilon_n(t)$$

$$\phi_u(t)^\top \phi_u(t) - \phi_u(t)^\top \Phi_n^\top (\Phi_n \Phi_n^\top + \sigma^2 I)^{-1} \Phi_n \phi_u(t) \approx d(t,t) - \boldsymbol{d}_n(t)^\top (\boldsymbol{K}_n + \sigma^2 I)^{-1} \boldsymbol{d}_n(t) =: D_n(t)$$

where $\Phi_n = [\phi(t_1), ..., \phi(t_n)]$. We derive a closed form expression of (24) in appendix A.4. Our approach to solving (23) is presented in algorithm 2. Algorithm 2 is efficient in the case when $n > m$. We present this version as it is very intuitive. However, note that for all simulations in this paper we will have $n < m$ and hence we sample from $\boldsymbol{\omega}|\mathcal{D}_n$ as described in the appendix of Seeger (2008). We assume, for simplicity, that N is divisible by the batch size Q. One striking aspect of this algorithm is that there is no reference to $\upsilon_n(\cdot)$ until the absolute final step. The reason for this is that we only need unbiased observations of $m(\cdot)$ to produce an estimate of $u(\cdot)$ by its posterior mean as in (22).

Algorithm 2:

```
Obtain $n_0$ points using initial space-filling design;
Query $m(t_i)$ by $y_i = g(t_i, \epsilon_i)$ for $i = 1, ..., n_0$;
Let $\mathcal{D}_{n_0} = \{t_i, y_i\}_{i=1}^{n_0}$;
for $n = n_0, n_0 + Q, \ldots, N + n_0 - Q$ do
    $\mathcal{Q}_0 = \emptyset$;
    for $q = 1, \ldots, Q$ do
        Draw $m$ samples $\boldsymbol{W}_i \sim p(\boldsymbol{w})$, $\boldsymbol{b}_i \sim \mathcal{U}(0, 2\pi)$;
        Let $\phi^{(q)}(t) = \sqrt{2\alpha/m}\cos(\boldsymbol{W}t + \boldsymbol{b})$ and $\phi_u^{(q)}(t)$ as in (24);
        Sample $\boldsymbol{\omega}^{(q)}$ from $\boldsymbol{\omega}|\mathcal{D}_n \sim \mathcal{N}(\boldsymbol{m}_n, \boldsymbol{V}_n)$ where
         $\boldsymbol{m}_n = (\Phi_n^\top \Phi_n + \sigma^2 I)^{-1} \Phi_n^\top \boldsymbol{y}$, $\boldsymbol{V}_n = (\Phi_n^\top \Phi_n + \sigma^2 I)^{-1} \sigma^2$, and
         $\Phi_n = [\phi(t_1), ..., \phi(t_n)]$;
        Select next query point $t_q = \arg\max_{t \in [0,1]} \phi_u^{(q)}(t)^\top \boldsymbol{w}^{(q)}$;
        Query $m(t_q)$ by $y_q = g(t_q, \epsilon_q)$;
        Augment data $\mathcal{Q}_q = \{\mathcal{Q}_{q-1}, (t_q, y_q)\}$;
    end
    Augment data $\mathcal{D}_{n+Q} = \{\mathcal{D}_n, \mathcal{Q}_Q\}$;
end
return $t^* = \arg\max_{t \in [0,1]} \upsilon_{N+n_0}(t)$, where $\upsilon_n(t)$ is defined in (22)
```

4 Simulations

4.1 Optimal subsidy rates in health insurance

4.1.1 Setup

In this section we will consider the problem of setting optimal subsidy rates in health insurance. For the coinsurance rate $(1-t)$ we have that t is the subsidy rate, which we can think of as a negative tax. This is the policy problem treated in Kasy (2018). However, he considers the case of setting the optimal subsidy rate after the data has been collected using data from the RAND health insurance experiment. Here, we consider the case when the policy maker is adaptively designing experiments to find the optimal subsidy rate. In order to simulate this setting, we will provide estimates similar to that of Kasy (2018) for the posterior expected average health care expenditure and social welfare using the RAND health insurance experiment data. We subsequently treat this as the ground truth, and simulate experimental output by sampling residuals using the wild bootstrap introduced in Wu (1986).

4.1.2 Generating samples

First of all, we want to reproduce the results from Kasy (2018). However, the kernel used in Kasy (2018) is non-stationary, whereas in this paper we consider only stationary kernels since this is a key hypothesis in Bochner's theorem, which is crucial for the approximate samples using random Fourier features. In addition, Kasy (2018) is conditioning on a set of covariates, whereas we here take an unconditional approach for simplicity. A third difference is that we are using random Fourier features to produce the posterior expected average health care expenditure and social welfare. Note, however, that the approximation error is fairly small and thus this will have a negligible effect. Hence, using (22) and the RAND health insurance experiment data will only approximately reproduce the results in Kasy (2018). Nevertheless, this is of secondary importance as we only strive to demonstrate that the proposed method is working using realistic experimental data. In addition, the difference from Kasy (2018) is very small, as can be seen in figure 1 which plots (20) and (22) using the RAND health insurance experiment data. The original RAND health insurance experiment had six different subsidy rates $t \in \{0.05, 0.5, 0.75, 1\}$ and two more complicated plans. Following Kasy (2018), for clarity of exposition, we only

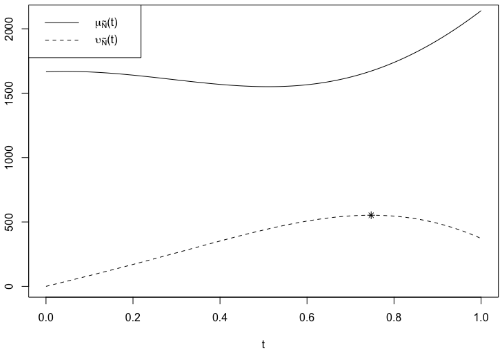


Figure 1: This is a partial, and approximate, reproduction of figure 1 in Kasy (2018). It is only an approximate reproduction since we are using a different kernel, consider the unconditional setting and use random Fourier features. $\mu_{\tilde{N}}(\cdot)$ and $\upsilon_{\tilde{N}}(\cdot)$ are defined as in (20) and (22) respectively for $\tilde{N} = 14776$ using the RAND health insurance experiment data. We use $m = 1000$ features. The hyperparameters are the same as used in Kasy (2018), namely $(\ell, \sigma_0^2, \sigma^2) = (1, 1, 1)$. The optimum posterior expected social welfare is at $*$ which is obtained at $t^* \approx 0.74$. Note that this is different from the estimate obtained in Kasy (2018) which is $t^* = 0.82$.

consider the first four plans.[20] To generate experimental responses, we will use the wild bootstrap originally proposed by Wu (1986). In particular, let $\hat{r}_i$ denote the residuals obtained as $\hat{r}_i := y_i - \mu_{\tilde{N}}(t_i)$ for $i = 1, ...\tilde{N}$, where $\tilde{N} = 14776$ is the number of observations in the RAND health insurance experiment data and $\mu_n(t)$ is defined in (20). Note that t_i only has four points of support in the data that we consider. We then generate a sample at $t_q \in [0, 1]$ as

$$y_q^* = \mu_{\tilde{N}}(t_q) + \hat{r}_q v_q \tag{25}$$

where $\hat{r}_q$ is drawn uniformly from the set of residuals and v_q is 1 with probability $\frac{1}{2}$ and -1 with probability $\frac{1}{2}$. Hence, (25) takes the place of $y_q = g(t_q, \epsilon_q)$ in algorithm 2.

4.1.3 Simulations

The observations produced using the wild bootstrap as described above are extremely noisy. This is expected, however, as individuals are subjected to widely different health shocks. In particular, only a few unlucky individuals will be subjected to severe health shocks and thus incur high health care expenditure. This is also the reason why we use the wild bootstrap instead of directly sampling residuals. Through extensive experimentation, it was shown to be extremely

[20] We use the data from `https://github.com/maxkasy/optimaltaxationusingML`.

unstable to set the hyperparameters using maximum marginal likelihood in this setting. That tuning the hyperparameters may be difficult is a known issue in the literature and a common suggestion is to take a fully Bayesian approach. However, this requires selecting hyperpriors[21] and therefore we decided instead to just fix the hyperparameters whilst running algorithm 2. This is still a perfectly sensible Bayesian approach to the policy maker's problem, however it is unlikely to work as well as when the hyperparameters are tuned to fit the given data. We fixed the hyperparameters by eye-balling an initial sample produced by the wild bootstrap approach described above at $(\ell, \sigma_0^2, \sigma^2) = (0.2, 1, 1)$. Following Kasy (2018), we let $\lambda = 1.5$. We run the algorithm for various batch sizes Q, ranging from $Q = 1$ (i.e. sequential) to $Q = 250$ (i.e. only two batches). The algorithm runs for $N = 500$ iterations with an initial sample $n_0 = 50$ obtained by standard latin hypercubes. Note that in this setting, 550 total observations is quite little.[22] However, we stop at 500 iterations due to the cubic complexity of algorithm 2 which slows things down considerably for moderately large n. We take an alternative approach in appendix B.2 where we make use of a considerably larger sample budget. We use $m = 1000$ draws from the spectral density, or features, for the random Fourier features approximation. Averaging over 50 trials yields the mean average regret in figure 2. These results indicate that the algorithm is robust to batch sizes even in the case of extremely noisy observations and fixed hyperparameters. Nevertheless, we have observed through extensive experimentation that performance for a relatively small N is fairly sensitive to batch size in this setting. We will consistently see convergence rates as in figure 2 for $Q = 1$. But for larger Q, average regret might remain constant, or even increase, for the first Q observations before the first update. In severe cases, a couple of updates of the posterior will be required prior to finding a reasonable estimate of the posterior expected social welfare and thus a good estimate of the global optima. In particular, this problem is mitigated as the total number of updates of the posterior increases, i.e. as Q/N becomes small where Q is the batch size and N is the number of iterations. In addition, as we will see in section 4.2, this is not a problem in the relatively lower noise setting where we can tune the hyperparameters to fit the given data. Hence, results in this setting are somewhat sensitive to the particular residuals drawn.[23]

[21] We tried this and other approaches. Nevertheless, this issue persisted.

[22] Compare to the actual RAND health insurance experiment where they had 14776 observations.

[23] Robustness improvements had to be left for future research due to time constraints.

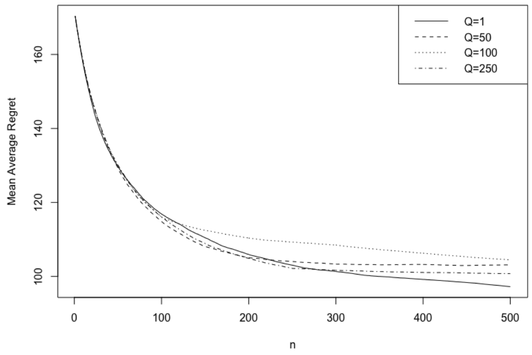


Figure 2: Simulations based on setup in section 4.1 where we generate experimental output using the wild bootstrap. We run for $N = 500$ iterations with an initial sample size of $n_0 = 50$. Each simulation is repeated for 50 trials and then averaged to produce the mean average regret. Q refers to the batch size. We use $m = 1000$ features. Hyperparameters are fixed throughout at $(\ell, \sigma_0^2, \sigma^2) = (0.2, 1, 1)$.

4.2 Optimal linear income taxation

4.2.1 Setup

As previously mentioned, the setup in Kasy (2018) applies to a wide class of different policy problems. In particular, it covers the optimal linear income taxation problem that we will now consider. We consider linear taxation schedules for the sake of simplicity. On the one hand, we could think of this essentially as a toy problem. On the other hand, one could also think of this setting as optimal income taxation in the top tax bracket as in Saez (2001). The main difference here compared to Saez (2001) is that we do not calibrate the skill distribution to match empirical evidence on income, but rather just use the lognormal specification from the classical Mirrlees (1971) nonlinear income taxation model. y_i now denotes labor supply and (15) now captures the relationship of how consumers adjust their labor supply for some tax t_i. An analogous discussion to the health insurance case yields the social welfare function

$$u(t) = -\lambda \int_0^t m(x)dx + tm(t) \tag{26}$$

where $\lambda < 1$ represents how much the policy maker values an additional dollar of the consumer relative to public revenues (Kasy, 2018). Note that this is the same as (19) except of a change in sign. Except for this change in sign, and the different interpretation of λ, this setup is exactly as in section 3.1 and discussion there applies almost verbatim.

4.2.2 Generating samples

To simulate experimental responses we require an explicit expression for $y_i = g(t_i, \epsilon_i)$. In the scenario of optimal linear income taxation, y_i represents the labor supplied by individual i faced with some tax t_i. We suppose that utility is quasilinear so that there are no income effects.[24] In particular, we will use the "type I" utility from Saez (2001)

$$u(c, l) = \log\left(c - \frac{l^k}{1+k}\right).$$

This functional form was also used in the theoretical work of Diamond (1998). The compensated (and uncompensated) elasticity is $\xi^c = \xi^u = \xi = \frac{1}{k}$. We will consider two values for the compensated elasticity that are within the plausible empirical range: $\xi = 0.25$ and $\xi = 0.5$ (Saez, 2001). Following the classical Mirrlees (1971) model, we will assume that earnings depend on the skill level n. Wages are normalized such that the marginal product of labor is simply $n \geq 0$. The policy maker levies a proportional tax which is returned in lump-sum rebates. Consumption c is simply the after tax labor income plus the non labor income R

$$c = ln(1-t) + R.$$

The Marshallian (uncompensated) labor supply function is thus

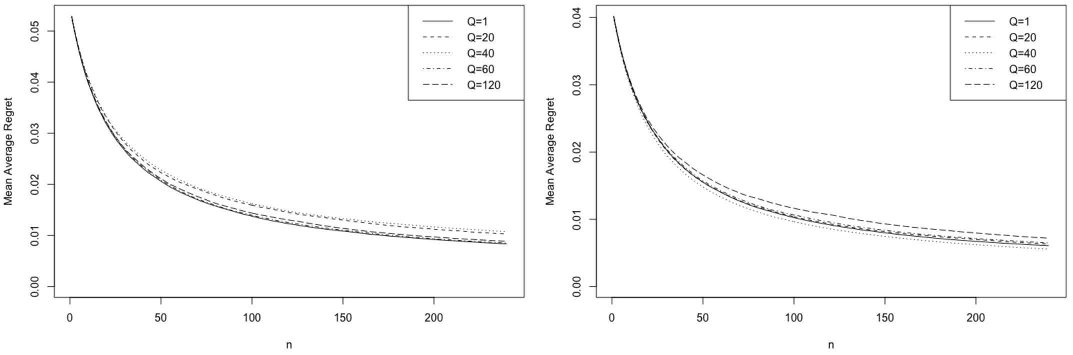
$$l(t) = (n(1-t))^{\frac{1}{k}}. \tag{27}$$

We will assume that $n \sim \text{lognormal}(\mu, \sigma)$. This assumption has been maintained in an abundance of papers following the seminal work of Mirrlees (1971). Hence, in iteration q, we will generate a sample y_q by drawing $n_q \sim \text{lognormal}(\mu, \sigma)$ and compute

$$y_q = g(t_q, \epsilon_q) = (n_q(1-t_q))^{1/k}.$$

For derivations of results in this section see appendix A.5.

[24] This is a very common assumption in the literature (see e.g., Diamond (1998); Saez (2001)).



(a)

(b)

Figure 3: Simulations based on setup in section 4.2 with $n \sim \text{lognormal}(-1, 0.39)$. In (a) $\xi = 0.25$ and in (b) $\xi = 0.5$. We run for $N = 240$ iterations with an initial sample size of $n_0 = 24$. Each simulation is repeated for 30 trials and then averaged to produce the mean average regret. Q refers to the batch size. We use $m = 1000$ draws for the random Fourier features. We set initial hyperparameters based on the first n_0 samples. These are subsequently updated after each batch for $Q > 1$ and every 10th iteration for $Q = 1$.

4.2.3 Simulations

As mentioned in the preceding section, we will consider the two cases $\xi = 0.25$ and $\xi = 0.5$. We will use the same lognormal skill distribution as in Mirrlees (1971). Namely, $n \sim \text{lognormal}(-1, 0.39)$. To demonstrate the robustness of Thompson sampling to batching, we will run several simulations with batch sizes ranging from $Q = 1$ to $Q = 120$. We run each simulation for $N = 240$ iterations with an initial sample of size $n_0 = 24$ obtained by standard latin hypercubes. We use $m = 1000$ draws for the random Fourier features approximation. For $Q = 1$ the hyperparameters will be updated each 10th iteration by maximizing the marginal likelihood. For $Q > 1$ the hyperparameters are simply updated at the end of each batch. We repeat each simulation for 30 trials and then take the average to obtain the mean average regret. We will set $\lambda = \frac{1}{2}$. Figure 3a and figure 3b show the results for $\xi = 0.25$ and $\xi = 0.5$, respectively. We can see that the Thompson sampling algorithm is very robust to batch sizes, as expected from the theoretical results in Kandasamy et al. (2018). We can note that in this setting we are running far more iterations than necessary to get a good approximation of the global optima. We do this mainly to illustrate how the algorithm is performing for relatively large batch sizes. Visually inspecting these results indicates that the additional mapping of experimental output into social welfare has a negligible effect on performance. In particular, problems faced in section 4.1 are far less salient in this setting.

5 Conclusion

In this paper we have extended the work in Kasy (2018) to the setting where a policy maker is adaptively designing experiments to maximize ex-ante expected social welfare. We do this by framing the problem as a multi-armed bandit, where the prior distribution over the reward function is a Gaussian process. This setting is commonly referred to as Bayesian optimization. Leveraging results from optimal policy theory using Gaussian process priors (Kasy, 2018), Bayesian optimization using Thompson sampling, and approximations based on random Fourier features, we proposed a solution to the policy maker's problem in algorithm 2. To evaluate the suggested approach, we ran calibrated simulations of two policy problems. First, we considered setting the optimal subsidy rate in health insurance. This experiment was simulated using the wild bootstrap and data from the RAND health insurance experiment. This setting resulted in extremely noisy observations and thus represented a very stringent test of the approach suggested in this paper. In addition, we considered the problem of optimal linear income taxation where we used empirically plausible elasticities and skill distribution as in the classical Mirrlees (1971) paper. We can think of this application as merely a toy problem to demonstrate robustness of the algorithm to batch sizes, or we could think of it as in Saez (2001) where we consider only the top tax bracket. Results from both of these applications are very encouraging and they indicate that Bayesian optimization is very robust to the additional complication of having to map observable experimental output into social welfare.

There are a lot of interesting avenues of future research as possible extensions to the work in this paper. First of all, as mentioned above, finding stable ways of setting the hyperparameters is a known issue in the the literature. In the social sciences, we are exposed to extremely noisy data, and it would be beneficial, for the sake of improving the applicability of the approach in this paper, to find a more stable way of setting the hyperparameters even when we have extremely noisy observations. It is clear from the applications in section 4.1 and 4.2 that the suggested approach performs better in the relatively lower noise setting where hyperparameters can be tuned to fit the data. In general, there is room for robustness improvements and further automation to expand the applicability of these results. Secondly, the unconditional approach taken in this paper is very limited and we should extend the results, as in Kasy (2018), to allow for the possibility of conditioning on relevant covariates. Third, an important task left for future research is to derive regret bounds for the suggested algorithm. Regret bounds for Thompson sampling have already been established (see e.g., Russo et al. (2017); Kandasamy et al. (2018)).

However, there are two additional complications in this paper than might affect these: the fact that we are using approximate draws from the posterior based on random Fourier features, and that we required this additional mapping from the observable response to the objective function. Nevertheless, we can note that the results in this paper suggest that the rate of convergence is negligibly impacted by these additional complications. Fourth, we are still considering one dimensional policy problems. An interesting and important extension is to generalize the results in this paper to the setting when we have a multidimensional policy vector. This generalization will introduce new problems both from an economic and a machine learning perspective. Fifth, we have used Thompson sampling as acquisition function as it is very effective yet simple to understand and implement. Nevertheless, there are other heuristics that are perhaps more suited for the problem at hand such as the knowledge gradient (see e.g., Frazier et al. (2009); Wu and Frazier (2016)). Note, however, that the choice of kernel is usually of higher importance than the choice of acquisition function (Shahriari et al., 2016). Hence, it might also be of interest to extend the approach in this paper to non-stationary kernels like the one used in Kasy (2018).

6 References

Aprem, A. and Roberts, S. J. (2018). A bayesian optimization approach to compute the nash equilibria of potential games using bandit feedback. *The Computer Journal*.

Aprem, A. and Roberts, S. J. (2021). Optimal pricing in black box producer-consumer stackelberg games using revealed preference feedback. *Neurocomputing (Amsterdam)*, 437:31–41.

Berkenkamp, F., Schoellig, A. P., and Krause, A. (2019). No-regret bayesian optimization with unknown hyperparameters. *Journal of Machine Learning Research*, 20(50):1–24.

Bochner, S. (1959). *Lectures on Fourier integrals : with an author's supplement on monotonic functions, Stieltjes integrals, and harmonic analysis.* Annals of mathematics studies ; no. 42. Princeton University Press, Princeton.

Brochu, E., Cora, V. M., and de Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *ArXiv e-prints:1012.2599*.

Bubeck, S. and Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122.

Bull, A. D. (2011). Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(88):2879–2904.

Chaloner, K. and Verdinelli, I. (1995). Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304.

Chetty, R. (2009). Sufficient statistics for welfare analysis: A bridge between structural and reduced-form methods. *Annual Review of Economics*, 1(1):451–488.

Cover, T. M. and Thomas, J. A. (2006). *Elements of information theory.* Wiley-Interscience, Hoboken, N.J., 2nd ed. edition.

Desautels, T., Krause, A., and Burdick, J. W. (2014). Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *Journal of Machine Learning Research*, 15(1):3873–3923.

Diamond, P. A. (1998). Optimal income taxation: An example with a u-shaped pattern of optimal marginal tax rates. *The American Economic Review*, 88(1):83–95.

Frazier, P. I. (2018). A tutorial on bayesian optimization. *ArXiv e-prints:1807.02811.*

Frazier, P. I., Powell, W., and Dayanik, S. (2009). The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, 21(4):599–613.

Garnett, R., Osborne, M. A., and Roberts, S. J. (2010). Bayesian optimization for sensor set selection. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '10, page 209–219, New York, NY, USA. Association for Computing Machinery.

Gramacy, R. B. (2020). *Surrogates: Gaussian Process Modeling, Design and Optimization for the Applied Sciences*. Chapman Hall/CRC, Boca Raton, Florida. `http://bobby.gramacy.com/surrogates/`.

Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. (2014). Predictive entropy search for efficient global optimization of black-box functions. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 27, pages 918–926. Curran Associates, Inc.

Hernández-Lobato, J. M., Requeima, J., Pyzer-Knapp, E. O., and Aspuru-Guzik, A. (2017). Parallel and distributed thompson sampling for large-scale accelerated exploration of chemical space. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1470–1479. JMLR.org.

Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B. (2018). Gaussian processes and kernel methods: A review on connections and equivalences. *ArXiv e-prints:1807.02582.*

Kandasamy, K., Krishnamurthy, A., Schneider, J., and Poczos, B. (2018). Parallelised bayesian optimisation via thompson sampling. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 133–142, Playa Blanca, Lanzarote, Canary Islands. PMLR.

Kasy, M. (2018). Optimal taxation and insurance using machine learning — sufficient statistics and beyond. *Journal of public economics*, 167:205–219.

Kasy, M. and Sautmann, A. (2021). Adaptive treatment assignment in experiments for policy choice. *Econometrica*, 89(1):113–132.

Kushner, H. J. (1964). A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering*, 86(1):97–106.

Lai, T. L. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4 – 22.

Lizotte, D. J., Wang, T., Bowling, M. H., and Schuurmans, D. (2007). Automatic gait optimization with gaussian process regression. In Veloso, M. M., editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 944–949.

Milgrom, P. and Segal, I. (2002). Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2):583–601.

Mirrlees, J. A. (1971). An exploration in the theory of optimum income taxation. *The Review of Economic Studies*, 38(2):175–208.

Mockus, J. (1975). On the bayes methods for seeking the extremal point. *IFAC Proceedings Volumes*, 8(1):428–431.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization.* Springer, New York, NY, USA, second edition.

Picheny, V., Binois, M., and Habbal, A. (2019). A bayesian optimization approach to find nash equilibria. *Journal of Global Optimization*, 73(1):171–192.

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. In *Proceedings of the 20th International Conference on Neural Information Processing Systems*, NIPS'07, page 1177–1184, Red Hook, NY, USA. Curran Associates Inc.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning.* MIT, Cambridge, Mass. ; London.

Russo, D. and Van Roy, B. (2016). An information-theoretic analysis of thompson sampling. *Journal of Machine Learning Research*, 17(68):1–30.

Russo, D., Van Roy, B., Kazerouni, A., Osband, I., and Wen, Z. (2017). A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning.*

Saez, E. (2001). Using elasticities to derive optimal income tax rates. *Review of Economic Studies*, 68(1):205–229.

Seeger, M. W. (2008). Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, 9(26):759–813.

Shah, A. and Ghahramani, Z. (2015). Parallel predictive entropy search for batch global optimization of expensive objective functions. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 3330–3338. Curran Associates, Inc.

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

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, page 2951–2959, Red Hook, NY, USA. Curran Associates Inc.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2012). Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265.

Teytelboym, A., Caria, A. S., Kasy, M., Gordon, G., Quinn, S., and Shami, S. (2020). An adaptive targeted field experiment: job search assistance for refugees in jordan. *Working paper.*

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

Wu, C. F. J. (1986). Jackknife, bootstrap and other resampling methods in regression analysis. *The Annals of Statistics*, 14(4):1261–1295.

Wu, J. and Frazier, P. I. (2016). The parallel knowledge gradient method for batch bayesian optimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 3134–3142, Red Hook, NY, USA. Curran Associates Inc.

Appendices

A Mathematical appendix

A.1 Bayesian optimization

The partial derivative w.r.t. θ_k of (3) is (Rasmussen and Williams, 2006)

$$\begin{aligned}\frac{\partial}{\partial\theta_k}\log p(\boldsymbol{y}|\boldsymbol{x}_{1:n},\boldsymbol{\theta}) &= \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{A}_n^{-1}\frac{\partial \boldsymbol{A}_n}{\partial\theta_k}\boldsymbol{A}_n^{-1}\boldsymbol{y} - \frac{1}{2}\operatorname{tr}\left(\boldsymbol{A}_n^{-1}\frac{\partial K}{\partial\theta_k}\right)\\ &= \frac{1}{2}\operatorname{tr}\left((\boldsymbol{\alpha}\boldsymbol{\alpha}^\top - K^{-1})\frac{\partial \boldsymbol{A}_n}{\partial\theta_k}\right) \qquad (28)\end{aligned}$$

where $\boldsymbol{\alpha} := \boldsymbol{A}_n^{-1}\boldsymbol{y}$ and $\boldsymbol{A}_n = \boldsymbol{K}_n + \sigma^2 I$. We further note that

$$\frac{\partial(\boldsymbol{A}_n)_{i,j}}{\partial\theta_k} = \frac{\partial(\boldsymbol{A}_n)_{i,j}}{\partial l_k} = (\boldsymbol{K}_n)_{i,j}\frac{(\boldsymbol{x}_{i,k}-\boldsymbol{x}_{j,k})^2}{l_k^3} \quad i,j=1,...,n, k=1,...,d$$

$$\frac{\partial \boldsymbol{A}_n}{\partial\theta_{d+1}} = \frac{\partial \boldsymbol{A}_n}{\partial\sigma_0^2} = \frac{1}{\sigma_0^2}\boldsymbol{K}_n$$

$$\frac{\partial \boldsymbol{A}_n}{\partial\theta_{d+2}} = \frac{\partial \boldsymbol{A}_n}{\partial\sigma^2} = I_n$$

where the the latter two are $n \times n$ matrices and the “gradient” of $\boldsymbol{A}_n$ w.r.t. to $\boldsymbol{\theta}$ is an order 3 tensor. Plugging this back into (28) and collecting the terms for $k = 1, ..., d+2$, into a $d+2$ dimensional vector yields the gradient of the marginal log likelihood w.r.t. to $\boldsymbol{\theta}$ using, $\nabla_{\boldsymbol{\theta}}\log p(\boldsymbol{y}|\boldsymbol{x}_{1:n},\boldsymbol{\theta})$.

A.2 Random Fourier features

In this section we expand on some of the results in section 2.2. We start with deriving (8). Following Hernández-Lobato et al. (2014) we note that, for a stationary kernel, (6) can be written as

$$\begin{aligned}
k(\boldsymbol{x}_1, \boldsymbol{x}_2) &= \alpha \mathbb{E}_{p(\boldsymbol{w})} e^{-i\boldsymbol{w}^\top (\boldsymbol{x}_1 - \boldsymbol{x}_2)} \\
&=_{(1)} \alpha \mathbb{E}_{p(\boldsymbol{w})} \left(\frac{1}{2} e^{-i\boldsymbol{w}^\top (\boldsymbol{x}_1 - \boldsymbol{x}_2)} + e^{i\boldsymbol{w}^\top (\boldsymbol{x}_1 - \boldsymbol{x}_2)} \right) \\
&=_{(2)} \alpha \mathbb{E}_{p(\boldsymbol{w})} \cos(\boldsymbol{w}^\top \boldsymbol{x}_1 - \boldsymbol{w}^\top \boldsymbol{x}_2) \\
&=_{(3)} \alpha \mathbb{E}_{p(\boldsymbol{w})} \cos(\boldsymbol{w}^\top \boldsymbol{x}_1 - \boldsymbol{w}^\top \boldsymbol{x}_2) + \alpha \mathbb{E}_{p(b)} \cos(\boldsymbol{w}^\top \boldsymbol{x}_1 + \boldsymbol{w}^\top \boldsymbol{x}_2 + 2b) \\
&=_{(4)} \alpha \mathbb{E}_{p(\boldsymbol{w},b)} \cos(\boldsymbol{w}^\top \boldsymbol{x}_1 + b - \boldsymbol{w}^\top \boldsymbol{x}_2 - b) + \cos(\boldsymbol{w}^\top \boldsymbol{x}_1 + b + \boldsymbol{w}^\top \boldsymbol{x}_2 + b) \\
&=_{(5)} 2\alpha \mathbb{E}_{p(\boldsymbol{w},b)} \cos(\boldsymbol{w}^\top \boldsymbol{x}_1 + b) \cos(\boldsymbol{w}^\top \boldsymbol{x}_2 + b)
\end{aligned}$$

where $=_{(1)}$ follows from the symmetry of $p(\boldsymbol{w})$ (Rasmussen and Williams, 2006), $=_{(2)}$ due to Euler's formula $e^{ix} = \cos(x) + i\sin(x)$, $=_{(3)}$ from the fact that $\int_0^{2\pi} \cos(a + 2b)db = 0, \forall a \in \mathbb{R}$, $=_{(4)}$ simple algebraic manipulation, $=_{(5)}$ due to the sum of angels formula $2\cos(x)\cos(y) = \cos(x - y) + \cos(x + y)$. To see (13), let $A := \Phi_n^\top \Phi_n + \sigma^2 I$

$$\begin{aligned}
\Phi_n^\top (\Phi_n \Phi_n^\top + \sigma^2 I) &= \Phi_n^\top \Phi_n \Phi_n^\top + \sigma^2 \Phi_n^\top \\
&= (\Phi_n^\top \Phi_n + \sigma^2 I) \Phi_n^\top \\
&= A \Phi_n^\top
\end{aligned}$$

Left multiplying by A^{-1} and right multiplying by $(\Phi_n \Phi_n^\top + \sigma^2 I)^{-1}$ yields

$$\begin{aligned}
A^{-1} \Phi_n^\top (\Phi_n \Phi_n^\top + \sigma^2 I)(\Phi_n \Phi_n^\top + \sigma^2 I)^{-1} &= A^{-1} A \Phi_n^\top (\Phi_n \Phi_n^\top + \sigma^2 I)^{-1} \\
A^{-1} \Phi_n^\top &= \Phi_n^\top (\Phi_n \Phi_n^\top + \sigma^2 I)^{-1} \\
(\Phi_n^\top \Phi_n + \sigma^2 I)^{-1} \Phi_n^\top &= \Phi_n^\top (\Phi_n \Phi_n^\top + \sigma^2 I)^{-1}
\end{aligned}$$

which is what we wanted to show. We now derive the normalized spectral density of k. Recall, the ARD kernel is

$$k(\boldsymbol{\tau}) = \sigma_0^2 e^{-\frac{1}{2} \boldsymbol{\tau}^\top \boldsymbol{\Lambda}^{-1} \boldsymbol{\tau}}$$

where $\mathbf{\Lambda} = \text{diag}(l_1^2, ..., l_d^2)$. From (7) we have that

$$\begin{aligned}
s(\boldsymbol{w}) &= \frac{1}{(2\pi)^d} \int e^{i\boldsymbol{w}^\top \boldsymbol{\tau}} k(\boldsymbol{\tau}, \mathbf{0}) d\boldsymbol{\tau} \\
&= \frac{\sigma_0^2}{(2\pi)^d} \int e^{i\boldsymbol{w}^\top \boldsymbol{\tau}} e^{-\frac{1}{2}\boldsymbol{\tau}^\top \mathbf{\Lambda}^{-1} \boldsymbol{\tau}} d\boldsymbol{\tau} \\
&=_{(1)} \frac{\sigma_0^2}{(2\pi)^d} \int e^{-\frac{1}{2}(\boldsymbol{\tau} - \mathbf{\Lambda}\boldsymbol{\omega} i)^\top \mathbf{\Lambda}^{-1} (\boldsymbol{\tau} - \mathbf{\Lambda}\boldsymbol{\omega} i) - \frac{1}{2}\boldsymbol{\omega}^\top \mathbf{\Lambda} \boldsymbol{\omega}} d\boldsymbol{\tau} \\
&= \sigma_0^2 e^{-\frac{1}{2}\boldsymbol{\omega}^\top \mathbf{\Lambda} \boldsymbol{\omega}} (2\pi)^{-d} \int e^{-\frac{1}{2}(\boldsymbol{\tau} - \mathbf{\Lambda}\boldsymbol{\omega} i)^\top \mathbf{\Lambda}^{-1} (\boldsymbol{\tau} - \mathbf{\Lambda}\boldsymbol{\omega} i)} d\boldsymbol{\tau} \\
&=_{(2)} \sigma_0^2 e^{-\frac{1}{2}\boldsymbol{\omega}^\top \mathbf{\Lambda} \boldsymbol{\omega}} (2\pi)^{-\frac{d}{2}} |\mathbf{\Lambda}|^{\frac{1}{2}} \int (2\pi)^{-\frac{d}{2}} |\mathbf{\Lambda}|^{-\frac{1}{2}} e^{-\frac{1}{2}(\boldsymbol{\tau} - \mathbf{\Lambda}\boldsymbol{\omega} i)^\top \mathbf{\Lambda}^{-1} (\boldsymbol{\tau} - \mathbf{\Lambda}\boldsymbol{\omega} i)} d\boldsymbol{\tau} \\
&=_{(3)} \sigma_0^2 e^{-\frac{1}{2}\boldsymbol{w}^\top \mathbf{\Lambda} \boldsymbol{w}} (2\pi)^{-\frac{d}{2}} |\mathbf{\Lambda}|^{\frac{1}{2}}
\end{aligned}$$

$=_{(1)}$ is using a "completion of the square" formula for matrices (see appendix A.6), $=_{(2)}$ is just algebraic manipulations, and $=_{(3)}$ the integral is equal to one (compare integrand to a multivariate Gaussian density). Hence,

$$\begin{aligned}
\alpha &= \int s(\boldsymbol{w}) d\boldsymbol{w} \\
&= \sigma_0^2 \int_{\mathbb{R}^d} e^{-\frac{1}{2}\boldsymbol{w}^\top \mathbf{\Lambda} \boldsymbol{w}} (2\pi)^{-\frac{d}{2}} |\mathbf{\Lambda}|^{\frac{1}{2}} d\boldsymbol{w} \\
&=_{(1)} \sigma_0
\end{aligned}$$

where $=_{(1)}$ is just due to the integrand being the density of a multivariate Gaussian, $\mathcal{N}(\mathbf{0}, \mathbf{\Lambda}^{-1})$. Finally,

$$p(\boldsymbol{w}) = \frac{s(\boldsymbol{w})}{\alpha} = e^{-\frac{1}{2}\boldsymbol{w}^\top \mathbf{\Lambda} \boldsymbol{w}} (2\pi)^{-\frac{d}{2}} |\mathbf{\Lambda}|^{\frac{1}{2}} d\boldsymbol{w}.$$

Thus, $p(\boldsymbol{w}) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}^{-1})$.

A.3 Setup

Let $\varphi(\cdot)$ and $\Phi(\cdot)$ denote the standard normal PDF and CDF, respectively. For the ARD kernel

$$k(t_1, t_2) = \sigma_0^2 \exp\left(-\frac{(t_1 - t_2)^2}{2\ell^2}\right)$$

we note that

$$\frac{\sigma_0^2}{\varphi(0)} \varphi\left(\frac{t_1 - t_2}{\ell}\right) = \sigma_0^2 \sqrt{2\pi} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(t_1 - t_2)^2}{2\ell^2}\right) =: k(t_1, t_2).$$

Hence,

$$\begin{aligned}\int_0^{t_1} k(x,t_2)dx &= \frac{\sigma_0^2}{\varphi(0)}\int_0^{t_1}\varphi\left(\frac{x-t_2}{\ell}\right)dx\\ &= \frac{\sigma_0^2}{\varphi(0)}\left[\ell\Phi\left(\frac{x-t_2}{\ell}\right)\right]_0^{t_1}\\ &= \frac{\ell\sigma_0^2}{\varphi(0)}\left[\Phi\left(\frac{t_1-t_2}{\ell}\right)-\Phi\left(\frac{-t_2}{\ell}\right)\right]\end{aligned}$$

and

$$\begin{aligned}d(t_1,t_2) &= \lambda\int_0^{t_1} k(x,t_2)dx - t_1 k(t_1,t_2)\\ &= \frac{\sigma_0^2}{\varphi(0)}\left[\lambda\ell\left(\Phi\left(\frac{t_1-t_2}{\ell}\right)-\Phi\left(\frac{-t_2}{\ell}\right)\right)-t_1\varphi\left(\frac{t_1-t_2}{\ell}\right)\right].\end{aligned}$$

A.4 Algorithm

For the case when $d=1$, i.e. $t\in\mathbb{R}$, we get a simple expression[25] for (24) since

$$\int_0^t \phi(x)dx = \sqrt{\frac{2\alpha}{m}}\int_0^t \cos(\boldsymbol{W}x+\boldsymbol{b})dx = \sqrt{\frac{2\alpha}{m}}\left(\sin(\boldsymbol{W}t+\boldsymbol{b})-\sin(\boldsymbol{b})\right)\odot\tilde{\boldsymbol{W}}$$

where $\odot$ denotes the Hadamard (point-wise) product and $\tilde{\boldsymbol{W}} := [\boldsymbol{W}_1^{-1},...,\boldsymbol{W}_m^{-1}]$ and $\boldsymbol{W}$ and $\boldsymbol{b}$ are both $m\times 1$ vectors. Thus,

$$\phi_u(t) = \sqrt{\frac{2\alpha}{m}}\Big(\lambda\left(\sin(\boldsymbol{W}t+\boldsymbol{b})-\sin(\boldsymbol{b})\right)\odot\tilde{\boldsymbol{W}} - t\cos(\boldsymbol{W}t+\boldsymbol{b})\Big). \tag{29}$$

A.5 Labour supply function

The consumer's problem is

$$\max_{c,l}\log\left(c-\frac{l^{1+k}}{1+k}\right)\quad\text{s.t.}\quad c = ln(1-t)+R$$

substituting the constraint into the objective and taking the first order condition yields

$$n(1-t) = l(t)^k$$

[25] The expression is conceptually simple for $d>1$ as well. However, we can then no longer use the Hadamard product and notation is thus no longer obvious.

and thus the Marshallian labor supply function is as in (27).

A.6 Linear algebra

The matrix inversion lemma, also called the Woodbury matrix identity, Woodbury-Sherman-Morrison formula, or simply the Woodbury formula, states that (Rasmussen and Williams, 2006)

$$(\boldsymbol{Z}+\boldsymbol{U}\boldsymbol{W}\boldsymbol{V}^\top)^{-1}=\boldsymbol{Z}^{-1}-\boldsymbol{Z}^{-1}\boldsymbol{U}(\boldsymbol{W}^{-1}+\boldsymbol{V}^T\boldsymbol{Z}^{-1}\boldsymbol{U})^{-1}\boldsymbol{V}^\top\boldsymbol{Z}^{-1}$$

provided that all the required inverses exist. For $\boldsymbol{A}$ symmetric and nonsingular, we have the following "completion of the square" formula[26]

$$-\frac{1}{2}\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}+\boldsymbol{m}^\top\boldsymbol{x}=-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{A}^{-1}\boldsymbol{m})^\top\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{A}^{-1}\boldsymbol{m})+\frac{1}{2}\boldsymbol{m}^\top\boldsymbol{A}^{-1}\boldsymbol{m}$$

Proof.

$$\begin{aligned}(\boldsymbol{x}-\boldsymbol{A}^{-1}\boldsymbol{m})^\top\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{A}^{-1}\boldsymbol{m})&=(\boldsymbol{x}^\top-\boldsymbol{m}^\top\boldsymbol{A}^{-1})\boldsymbol{A}(\boldsymbol{x}-\boldsymbol{A}^{-1}\boldsymbol{m})\\&=\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}-\boldsymbol{x}^\top\boldsymbol{m}-\boldsymbol{m}^\top\boldsymbol{x}+\boldsymbol{m}^\top\boldsymbol{A}^{-1}\boldsymbol{m}\\&=\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}-2\boldsymbol{m}^\top\boldsymbol{x}+\boldsymbol{m}^\top\boldsymbol{A}^{-1}\boldsymbol{m}\end{aligned}$$

Rearranging and dividing through by $\frac{1}{2}$ yields the desired result. □

B Appendix

B.1 Connection to sufficient statistics

This section briefly discuss how the setup in Kasy (2018) is related to the sufficient statistics approach. For a more thorough treatment see Kasy (2018). The first order condition of the policy maker's problem is

$$u'(t)=m(t)\left[\lambda-1-t\frac{m'(t)}{m(t)}\right]=m(t)\left[(\lambda-1)-\eta\frac{t}{t-1}\right]$$

[26] This derivation is based on `https://davidrosenberg.github.io/mlcourse/Notes/completing-the-square.pdf`.

where $\eta := -\frac{\partial m(t)}{\partial(1-t)}\frac{1-t}{m(t)}$ is the elasticity of health care expenditure w.r.t. the coinsurance rate $1-t$ (Kasy, 2018). Solving $u'(t^*) = 0$ for t^* yields

$$t^*(\eta) = \frac{1}{1+\frac{\eta}{\lambda-1}}. \tag{30}$$

Hence, the optimal tax is only a function of the relevant elasticity η and the parameter λ. The sufficient statistics approach is to estimate η by some $\hat{\eta}$ and substitute this into (30) to get the desired estimate $\hat{t}^*(\hat{\eta})$. The optimal policy from the approach in Kasy (2018) can be written in a similar manner

$$u'(t) = m(t)\left[\lambda - 1 - t\frac{m'(t)}{m(t)}\right] = m(t)\left[\lambda - 1 - \xi\frac{t}{1-t}\right]$$

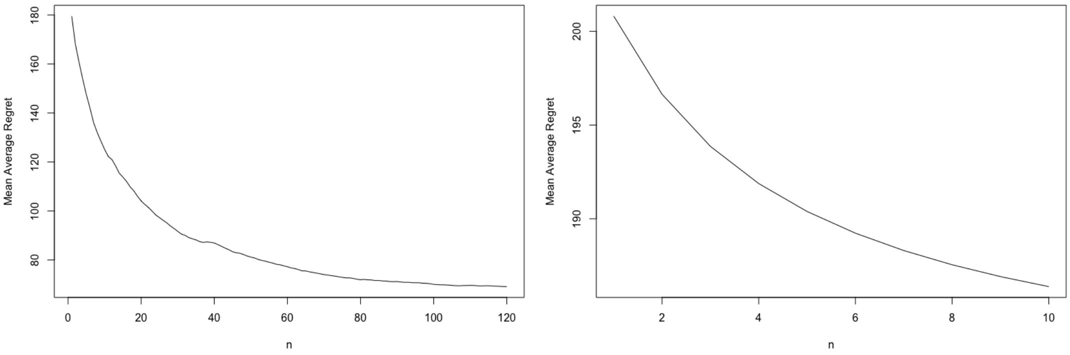
where $\xi := -\frac{\partial m(t)}{\partial(1-t)}\frac{1-t}{m(t)}$. Hence,

$$t^* = \frac{1}{1+\frac{\xi}{\lambda-1}} \tag{31}$$

and the estimate $\hat{t}^* = \arg\max_t \upsilon_n(t)$ is obtained by substituting $\hat{\xi} := -\frac{\hat{m}(\partial\hat{t}^*)}{\partial(1-\hat{t}^*)}\frac{1-\hat{t}^*}{\hat{m}(\hat{t}^*)}$ for ξ. Comparing (30) to (31) we note that the only difference is that we have replaced η with ξ. Nevertheless, this may yield quite different results in as discussed in Kasy (2018).

B.2 Alternative approach to extremely noisy observations

We consider here an alternative to the approach in section 4.1. Instead of batching as in algorithm 2 we could draw Q samples for each t and then take the average as the experimental output. Hence, we would essentially run a sequential version of algorithm 2 with output being the average over some Q draws. This will reduce the amount of noise and make it possible to set the hyperparameters using the standard approach. Note that a fairly large Q is required to reduce the noise sufficiently. However, as we make Q larger the total required sample size N grows quickly. We want to limit ourselves to a sample budget that is smaller than that of the RAND health insurance experiment. In particular, we will set the maximum N to be 13000. This can be achieved by different combinations of Q and N. Note that there is a trade-off between choosing Q and N. A larger Q will lead to more precise estimate of $u(\cdot)$ but it comes at the price of fewer iterations of the algorithm. For instance, one could use $n_0 = 10$, draw $Q = 100$ samples per batch, and run for $N = 120$ iterations. Alternatively, we could use $n_0 = 3$ initial samples, draw $Q = 1000$ samples per batch, and run the algorithm for $N = 10$ iterations. The output



(a)

(b)

Figure 4: Simulations based on setup in section 4.1. (a) Run for $N = 120$ iterations with $n_0 = 10$ initial samples. Each iterations is the average of $Q = 100$ draws. This is repeated and averaged over 30 trials to produce the mean average regret. Hyperparameters are updated every 10th iteration. (b) Run for $N = 10$ iterations with $n_0 = 3$ initial samples. Each iterations is the average of $Q = 1000$ draws. Hyperparameters are updated every iteration.

from these examples can be found in figure 4a and 4b, respectively. We run the algorithm for 30 trials and take the average output to get the mean average regret. Hyperparameters are again updated every 10th iteration for the $Q = 100$ case and for each iteration in the $Q = 1000$ case. We can see that we maintain a similar rate of decay of regret as observed in section 4.1. We use $m = 1000$ features for both simulations. Note, however, as the approach in section 4.1, it has been difficult to get this to work in a stable manner.