On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning

Matthew W. Hoffman
University of Cambridge

Bobak Shahriari
University of British Columbia

Nando de Freitas
Oxford University

Abstract

We address the problem of finding the maximizer of a nonlinear function that can only be evaluated, subject to noise, at a finite number of query locations. Further, we will assume that there is a constraint on the total number of permitted function evaluations. We introduce a Bayesian approach for this problem and show that it empirically outperforms both the existing frequentist counterpart and other Bayesian optimization methods. The Bayesian approach places emphasis on detailed modelling, including the modelling of correlations among the arms. As a result, it can perform well in situations where the number of arms is much larger than the number of allowed function evaluation, whereas the frequentist counterpart is inapplicable. This feature enables us to develop and deploy practical applications, such as automatic machine learning toolboxes. The paper presents comprehensive comparisons of the proposed approach with many Bayesian and bandit optimization techniques, the first comparison of many of these methods in the literature.

1 Introduction

This paper draws connections between Bayesian optimization approaches and best arm identification in the bandit setting. It focuses on problems where the number of permitted function evaluations is bounded. To solve these constrained optimization problems, we introduce a new algorithm: BayesGap. The paper shows, via comprehensive comparisons of a large number of bandit, experimental design and Bayesian optimization techniques, that this Bayesian best arm identification method outperforms other methods in this domain. Moreover, the Bayesian treatment enables us to attack problems where gap techniques relying on independent arms are inapplicable.

A typical example of this problem is that of automatic product testing [Kohavi et al., 2009, Scott, 2010], where common "products" correspond to configuration options for ads, websites, mobile applications, and online games. In this scenario, a company offers different product variations to a small subset of customers, with the goal of finding the most successful product for the entire customer base. The crucial problem is how best to query the smaller subset of users in order to find the best product with high probability. A second example, is that of automating machine learning algorithms. Here, the goal is to automatically select the best technique (boosting, random forests, support vector machines, neural networks, etc.) and its associated hyper-parameters for solving a machine learning task with a given dataset. For big datasets, cross-validation is very expensive and hence it is often important to find the best technique within a fixed budget of cross-validation tests (function evaluations).

In order to properly attack this problem there are three design aspects that must be considered. By taking advantage of *correlation* among different actions it is possible to learn more about a function than just its value at a specific query. This is particularly important when the number of actions greatly exceeds the *finite query budget* T. In this same vein, it is important to take into account that a recommendation must be made at time T in order to properly allocate actions and explore the space of possible optima. Finally, the fact that we are interested only in the value of the recommendation made at time T should be handled explicitly. In other words, we are only interested in finding the *best action* and are concerned with the rewards obtained during learning only insofar as they inform us about this optimum.

In this work, we introduce a Bayesian approach that meets the above design goals and show that it empiri-

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33.

cally outperforms the existing frequentist counterpart [Gabillon et al., 2012]. The Bayesian approach places emphasis on detailed modelling, including the modelling of correlations among the arms. As a result, it can perform well in situations where the number of arms is much larger than the number of allowed function evaluation, whereas the frequentist counterpart is inapplicable. The paper presents comprehensive comparisons of the proposed approach, Thompson sampling, classical Bayesian optimization techniques, more recent Bayesian bandit approaches, and state-of-the-art best arm identification methods. This is the first comparison of many of these methods in the literature and allows us to examine the relative merits of their different features. The paper also shows that one can easily obtain the same theoretical guarantees for the Bayesian approach that were previously derived in the frequentist setting [Gabillon et al., 2012].

1.1 Related work

Bayesian optimization has enjoyed success in a broad range of optimization tasks; see the work of Brochu et al. [2010b] for a broad overview. Recently, this approach has received a great deal of attention as a black-box technique for the optimization of hyperparameters [Snoek et al., 2012, Hutter et al., 2011, Wang et al., 2013b]. This type of optimization combines prior knowledge about the objective function with previous observations to estimate the posterior distribution over f. The posterior distribution, in turn, is used to construct an *acquisition function* that determines what the next query point a_t should be. Examples of acquisition functions include probability of improvement (PI), expected improvement (EI), Bayesian upper confidence bounds (UCB), and mixtures of these [Močkus, 1982, Jones, 2001, Srinivas et al., 2010, Hoffman et al., 2011]. One of the key strengths underlying the use of Bayesian optimization is the ability to capture complicated correlation structures via the posterior distribution.

Many approaches to bandits and Bayesian optimization focus on online learning (*e.g.*, minimizing cumulative regret) as opposed to optimization [Srinivas et al., 2010, Hoffman et al., 2011]. In the realm of optimizing deterministic functions, a few works have proven exponential rates of convergence for simple regret [de Freitas et al., 2012, Munos, 2011]. A stochastic variant of the work of Munos has been recently proposed by Valko et al. [2013]; this approach takes a tree-based structure for expanding areas of the optimization problem in question, but it requires one to evaluate each cell many times before expanding, and so may prove expensive in terms of the number of function evaluations.

The problem of optimization under budget constraints has received relatively little attention in the Bayesian optimization literature, though some approaches without strong theoretical guarantees have been proposed recently [Azimi et al., 2011, Hennig and Schuler, 2012, Snoek et al., 2011, Villemonteix et al., 2009]. In contrast, optimization under budget constraints has been studied in significant depth in the setting of multi-armed bandits [Bubeck et al., 2009, Audibert et al., 2010, Gabillon et al., 2011, 2012]; see also [Kaufmann and Kalyanakrishnan, 2013]. Here, a decision maker must repeatedly choose query points, often a finite number known as "arms", in order to observe their associated rewards [Cesa-Bianchi and Lugosi, 2006]. However, unlike most methods in Bayesian optimization the underlying value of each action is generally assumed to be independent from all other actions. That is, the correlation structure of the arms is often ignored.

2 Problem formulation

In order to attack the problem of Bayesian optimization from a bandit perspective we will consider a finite collection of arms $\mathcal{A} = \{1, \dots, K\}$ such that the immediate reward of pulling arm $k \in \mathcal{A}$ is characterized by a distribution ν_k with mean μ_k. From the Bayesian optimization perspective we can think of this as a collection of points $\{a_1, \dots, a_K\}$ where $\mu_k = f(a_k)$. Note that while we will assume the distributions ν_k are independent of past actions this *does not* mean that the means of each arm cannot share some underlying structure—only that the act of pulling arm k does not affect the future rewards of pulling this or any other arm. This distinction will be relevant later in this section.

The problem of identifying the best arm in this bandit problem can now be introduced as a sequential decision problem. At each round t the decision maker will select or "pull" an arm $a_t \in \mathcal{A}$ and observe an independent sample y_t drawn from the corresponding distribution ν_{a_t}. At the beginning of each round t, the decision maker must decide which arm to select based only on previous interactions, which we will denote with the tuple $(a_{1:t-1}, y_{1:t-1})$. For any arm k we can also introduce the expected immediate regret of selecting that arm as

$$R_k = \mu^* - \mu_k, \tag{1}$$

where μ^* denotes the expected value of the best arm. Note that while we are interested in finding the arm with the minimum regret, the exact value of this quantity is unknown to the learner.

In standard bandit problems the goal is generally to

minimize the cumulative sum of immediate regrets incurred by the arm selection process. Instead, in this work we consider the *pure exploration* setting [Bubeck et al., 2009, Audibert et al., 2010], which divides the sampling process into two phases: exploration and evaluation. The exploration phase consists of T rounds wherein a decision maker interacts with the bandit process by sampling arms. After these rounds, the decision maker must make a single arm recommendation $\Omega_T \in \mathcal{A}$. The performance of the decision maker is then judged *only* on the performance of this recommendation. The expected performance of this single recommendation is known as the *simple regret*, and we can write this quantity as R_{Ω_T}. Given a tolerance $\epsilon > 0$ we can also define the *probability of error* as the probability that $R_{\Omega_T} > \epsilon$. In this work, we will consider both the empirical probability that our regret exceeds some ϵ as well as the actual reward obtained. The related *fixed confidence* setting is one in which the learner continues selecting arms until some desired error probability is reached; however we will not address this setting in this work.

3 Bayesian bandits

We will now consider a bandit problem wherein the distribution of rewards for each arm is assumed to depend on unknown parameters $\theta \in \Theta$ that are shared between all arms. We will write the reward distribution for arm k as $\nu_k(\cdot|\theta)$. When considering the bandit problem from a Bayesian perspective, we will assume a prior density $\theta \sim \pi_0(\cdot)$ from which the parameters are drawn. Next, after $t-1$ rounds we can write the posterior density of these parameters as

$$\pi_t(\theta) \propto \pi_0(\theta) \prod_{n<t} \nu_{a_n}(y_n|\theta). \tag{2}$$

Here we can see the effect of choosing arm a_n at each time n: we obtain information about θ only indirectly by way of the likelihood of these parameters given reward observations y_n. Note that this also generalizes the *uncorrelated* arms setting. If the rewards for each arm k depend only on a parameter (or set of parameters) θ_k, then at time t the posterior for that parameter would only depend on those times in the past that we had pulled arm k.

We are, however, only partially interested in the posterior distribution of the parameters θ. Instead, we are primarily concerned with the expected reward for each arm under these parameters, which can be written as $\mu_k = \mathbb{E}[Y|\theta] = \int y\, \nu_k(y|\theta)\, dy$. The true value of θ is unknown, but we have access to the posterior distribution $\pi_t(\theta)$. This distribution induces a marginal distribution over μ_k, which we will write as $\rho_{kt}(\mu_k)$. The distribution $\rho_{kt}(\mu_k)$ can then be used to define upper and lower confidence bounds that hold with high probability and, hence, engineer acquisition functions that trade-off exploration and exploitation. We will derive an analytical expression for this distribution next.

We will assume that each arm k is associated with a feature vector $x_k \in \mathbb{R}^d$ and where the rewards for pulling arm k are normally distributed according to

$$\nu_k(y|\theta) = \mathcal{N}(y; x_k^T\theta, \sigma^2) \tag{3}$$

with variance σ^2 and unknown $\theta \in \mathbb{R}^d$. The rewards for each arm are independent conditioned on θ, but marginally dependent when this parameter is unknown. In particular the level of their dependence is given by the structure of the vectors x_k. By placing a prior $\theta \sim \mathcal{N}(0, \eta^2 I)$ over the entire parameter vector we can compute a posterior distribution over this unknown quantity. One can also easily place an inverse-Gamma prior on σ and compute the posterior analytically, but we will not describe this in order to keep the presentation simple.

The above linear observation model might seem restrictive. However, because we are only considering K actions (arms), it includes the Gaussian process (GP) setting. More precisely, let the matrix $G \in \mathbb{R}^{K\times K}$ be the covariance of a GP prior. Our experiments will detail two ways of constructing this covariance in practice. We can apply the following transformation to construct the design matrix $X = [x_1 \dots x_K]^T$:

$$X = VD^{\frac{1}{2}}, \text{ where } G = VDV^T.$$

The rows of X correspond to the vectors x_k necessary for the construction of the observation model in Equation (3). By restricting ourselves to finite actions spaces, we can also implement strategies such a Thompson sampling with GPs. For pragmatic reasons, many existing popular Bayesian optimization software tools consider either finite grids of actions, or some other such discretization of the query space [Hennig and Schuler, 2012, Swersky et al., 2013]. Although the question of how to best to discretize the space is a very interesting one, we defer it to future work and, in this work, only consider a finite, discreted subset of the action space.

We will now let $X_t = [x_{a_1} \dots x_{a_{t-1}}]^T$ denote the design matrix and $Y_t = [y_1 \dots y_{t-1}]^T$ the vector of observations at the beginning of round t. We can then write the posterior at time t as $\pi_t(\theta) = \mathcal{N}(\theta; \hat{\theta}_t, \hat{\Sigma}_t)$, where

$$\hat{\Sigma}_t^{-1} = \sigma^{-2} X_t^T X_t + \eta^{-2} I, \text{ and} \tag{4}$$

$$\hat{\theta}_t = \sigma^{-2} \hat{\Sigma}_t X_t^T Y_t. \tag{5}$$

From this formulation we can see that the expected reward associated with arm k is marginally normal

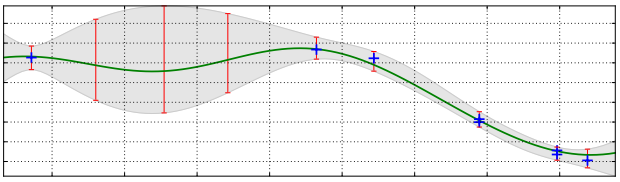

Figure 1: Example GP setting with finite arms. The full GP is plotted with observations and confidence intervals at each of $K = 9$ arms (mean and confidence intervals of $\rho_{kt}(\mu_k)$). Shown in green is a single sample from the GP.

$\rho_{kt}(\mu_k) = \mathcal{N}(\mu_k; \hat{\mu}_{kt}, \hat{\sigma}^2_{kt})$ with mean $\hat{\mu}_{kt} = x_k^T \hat{\theta}_t$ and variance $\hat{\sigma}^2_{kt} = x_k^T \hat{\Sigma}_t x_k$. Note also that the predictive distribution over rewards associated with the kth arm is normal as well, with mean $\hat{\mu}_{kt}$ and variance $\hat{\sigma}^2_{kt} + \sigma^2$. The previous derivations are textbook material; see for example Chapter 7 of [Murphy, 2012].

Figure 1 depicts an example of the mean and confidence intervals of $\rho_{kt}(\mu_k)$, as well as a single random sample. Here the features x_k were constructed by first forming the covariance matrix with an squared-exponential kernel $k(x, x') = \mathrm{e}^{-(x-x')^2}$ over the 1-dimensional domain. As with standard Bayesian optimization with GPs, the statistics of $\rho_{kt}(\mu_k)$ enable us to construct many different acquisition functions that trade-off exploration and exploitation. Thompson sampling in this setting also becomes straightforward, as we simply have to pick the maximum of the random sample from $\rho_{kt}(\mu_k)$, at one of the arms, as the next point to query.

4 Bayesian gap-based exploration

In this section we will introduce a gap-based solution to the Bayesian optimization problem, which we call BayesGap. This approach builds on the work of Gabillon et al. [2011, 2012], which we will refer to as UGap[1] and offers a principled way to incorporate correlation between different arms (whereas the earlier approach assumes all arms are independent).

At the beginning of round t we will assume that the decision maker is equipped with high-probability upper and lower bounds $U_k(t)$ and $L_k(t)$ on the unknown mean μ_k for each arm. While this approach can encompass more general bounds, for the Gaussian-arms setting that we consider in this work we can define these quantities in terms of the mean and standard deviation, i.e. $\hat{\mu}_{kt} \pm \beta \hat{\sigma}_{kt}$. These bounds also give rise to a confidence diameter $s_k(t) = U_k(t) - L_k(t) = 2\beta\hat{\sigma}_{kt}$.

Given bounds on the mean reward for each arm, we can then introduce the gap quantity

$$B_k(t) = \max_{i \neq k} U_i(t) - L_k(t), \tag{6}$$

which involves a comparison between the lower bound of arm k and the highest upper bound among all alternative arms. Ultimately this quantity provides an upper bound on the simple regret (see Lemma B1 in the supplementary material) and will be used to define the exploration strategy. However, rather than directly finding the arm minimizing this gap, we will consider the two arms

$$J(t) = \arg\min_{k \in \mathcal{A}} B_k(t) \text{ and}$$
$$j(t) = \arg\max_{k \neq J(t)} U_k(t).$$

We will then define the exploration strategy as

$$a_t = \arg\max_{k \in \{j(t), J(t)\}} s_k(t). \tag{7}$$

Intuitively this strategy will select either the arm minimizing our bound on the simple regret (i.e. $J(t)$) or the best "runner up" arm. Between these two, the arm with the highest uncertainty will be selected, i.e. the one expected to give us the most information. Next, we will define the recommendation strategy as

$$\Omega_T = J\big(\arg\min_{t \leq T} B_{J(t)}(t)\big), \tag{8}$$

i.e. the proposal arm $J(t)$ which minimizes the regret bound, over all times $t \leq T$. The reason behind this particular choice is subtle, but is necessary for the proof of the method's simple regret bound[2]. In Algorithm 1 we show the pseudo-code for BayesGap.

We now turn to the problem of which value of β to use. First, consider the quantity $\Delta_k = |\max_{i \neq k} \mu_i - \mu_k|$. For the best arm this coincides with a measure of the distance to the second-best arm, whereas for all other arms it is a measure of their sub-optimality. Given

Algorithm 1 BayesGap

for $t = 1, \ldots, T$ **do**
  set $J(t) = \arg\min_{k \in \mathcal{A}} B_k(t)$
  set $j(t) = \arg\max_{k \neq J(t)} U_k(t)$
  select arm $a_t = \arg\max_{k \in \{j(t), J(t)\}} s_k(t)$
  observe $y_t \sim \nu_{a_t}(\cdot)$
  update posterior $\hat{\mu}_{kt}$ and $\hat{\sigma}_{kt}$
  update bound on H_ϵ and re-compute β
  update posterior bounds $U_k(t)$ and $L_k(t)$
end for
return $\Omega_T = J\big(\arg\min_{t \leq T} B_{J(t)}(t)\big)$

[1] Technically UGapEb, denoting bounded horizon.

[2] See inequality (b) in the the supplementary material.

this quantity let $H_{k\epsilon} = \max(\frac{1}{2}(\Delta_k + \epsilon), \epsilon)$ be an arm-dependent hardness quantity; essentially our goal is to reduce the uncertainty in each arm to below this level, at which point with high probability we will identify the best arm. Now, given $H_\epsilon = \sum_k H_{k\epsilon}^{-2}$ we define our exploration constant as

$$\beta^2 = \big((T-K)/\sigma^2 + \kappa/\eta^2\big)/(4H_\epsilon) \tag{9}$$

where $\kappa = \sum_k \|x_k\|^{-2}$. We have chosen β such that with high probability we recover an ϵ-best arm, as detailed in the following theorem. This theorem relies on bounding the uncertainty for each arm by a function of the number of times that arm is pulled. Roughly speaking, if this bounding function is monotonically decreasing and if the bounds U_k and L_k hold with high probability we can then apply Theorem 2 to bound the simple regret of BayesGap[3].

Theorem 1. *Consider a K-armed Gaussian bandit problem, horizon T, and upper and lower bounds defined as above. For $\epsilon > 0$ and β defined as in Equation (9), the algorithm attains simple regret satisfying* $\Pr(R_{\Omega_T} \leq \epsilon) \geq 1 - KTe^{-\beta^2/2}$.

Proof. Using the definition of the posterior variance for arm k, we can write the confidence diameter as

$$\begin{aligned} s_k(t) &= 2\beta\sqrt{x_k^T \hat{\Sigma}_t x_k} \\ &= 2\beta\sqrt{\sigma^2 x_k^T \big(\textstyle\sum_i N_i(t-1)\, x_i x_i^T + \frac{\sigma^2}{\eta^2} I\big)^{-1} x_k} \\ &\leq 2\beta\sqrt{\sigma^2 x_k^T \big(N_k(t-1)\, x_k x_k^T + \frac{\sigma^2}{\eta^2} I\big)^{-1} x_k}. \end{aligned}$$

In the second equality we decomposed the Gram matrix $X_t^T X_t$ in terms of a sum of outer products over the fixed vectors x_i. In the final inequality we noted that by removing samples we can only increase the variance term, i.e. here we have essentially replaced $N_i(t-1)$ with 0 for $i \neq k$. We will let the result of this final inequality define an arm-dependent bound g_k. Letting $A = \frac{1}{N}\frac{\sigma^2}{\eta^2}$ we can simplify this quantity using the Sherman-Morrison formula as

$$\begin{aligned} g_k(N) &= 2\beta\sqrt{(\sigma^2/N) x_k^T \big(x_k x_k^T + AI\big)^{-1} x_k} \\ &= 2\beta\sqrt{\frac{\sigma^2}{N}\frac{\|x_k\|^2}{A}\Big(1 - \frac{\|x_k\|^2/A}{1 + \|x_k\|^2/A}\Big)} \\ &= 2\beta\sqrt{\frac{\sigma^2\|x_k\|^2}{\frac{\sigma^2}{\eta^2} + N\|x_k\|^2}}, \end{aligned}$$

which is monotonically decreasing in N. The inverse of this function can be solved for as

$$g_k^{-1}(s) = \frac{4(\beta\sigma)^2}{s^2} - \frac{\sigma^2}{\eta^2}\frac{1}{\|x_k\|^2}.$$

[3]The additional Theorem is in supplementary material and is a slight modification of that in [Gabillon et al., 2012].

By setting $\sum_k g_k^{-1}(H_{k\epsilon}) = T - K$ and solving for β we then obtain the definition of this term given in the statement of the proposition. Finally, by reference to Lemma B4 (supplementary material) we can see that for each k and t, the upper and lower bounds must hold with probability $1 - e^{-\beta/2}$. These last two statements satisfy the assumptions of Theorem 2 (supplementary material), thus concluding our proof. □

Here we should note that while we are using Bayesian methodology to drive the exploration of the bandit, we are analyzing this using frequentist regret bounds. This is a common practice when analyzing the regret of Bayesian bandit methods [Srinivas et al., 2010, Kaufmann et al., 2012a]. We should also point out that implicitly Theorem 2 assumes that each arm is pulled at least once regardless of its bound. However, in our setting we can avoid this in practice due to the correlation between arms.

One key thing to note is that the proof and derivation of β given above explicitly require the hardness quantity H_ϵ, which is unknown in most practical applications. Instead of requiring this quantity, our approach will be to adaptively estimate it. Intuitively, the quantity β controls how much exploration BayesGap does (note that β directly controls the width of the uncertainty $s_k(t)$). Further, β is inversely proportional to H_ϵ. As a result, in order to initially encourage more exploration we will lower bound the hardness quantity. In particular, we can do this by upper bounding each Δ_k by using conservative, posterior dependent upper and lower bounds on μ_k. In this work we use three posterior standard deviations away from the posterior mean, i.e. $\hat{\mu}_k(t) \pm 3\hat{\sigma}_{kt}$. (We emphasize that these are not the same as $L_k(t)$ and $U_k(t)$.) Then the upper bound on Δ_k is simply

$$\hat{\Delta}_k = \max_{j \neq k}(\hat{\mu}_j + 3\hat{\sigma}_j) - (\hat{\mu}_k - 3\hat{\sigma}_k).$$

From this point we can recompute H_ϵ and in turn recompute β (step 7 in the pseudocode). For all experiments we will use this adaptive method.

Comparison with UGap. The method in this section provides a Bayesian version of the UGap algorithm which modifies the bounds used in this earlier algorithm's arm selection step. By modifying step 6 of the BayesGap pseudo-code to use either Hoeffding or Bernstein bounds we can re-obtain the UGap algorithm. Note, however, that in doing so UGap assumes independent arms with bounded rewards.

We can now roughly compare UGap's probability of error, i.e. $O(KT\exp(-\frac{T-K}{H_\epsilon}))$, with that of BayesGap, $O(KT\exp(-\frac{T-K+\kappa\sigma^2/\eta^2}{H_\epsilon\sigma^2}))$. We can see that with minor differences, these bounds are of the same order.

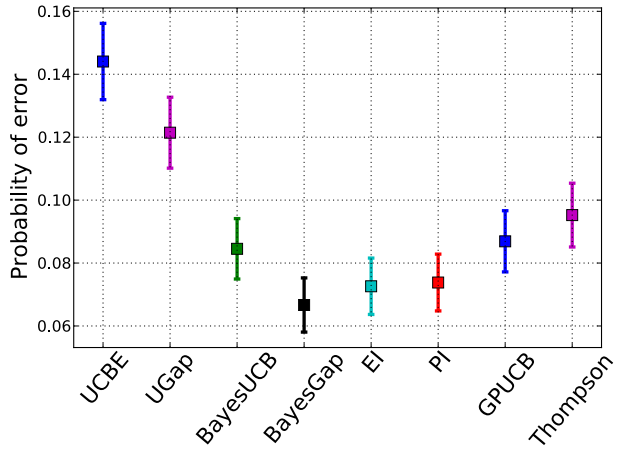

Figure 2: Probability of error on the optimization domain of traffic speed sensors (840 runs). For this real data set, BayesGap provides considerable improvements over the Bayesian cumulative regret alternatives and the frequentist simple regret counterparts.

First, we can ignore the additional σ^2 term as this quantity is primarily due to the distinction between bounded and Gaussian-distributed rewards. The η^2 term corresponds to the concentration of the prior, and we can see that the more concentrated the prior is (smaller η) the faster this rate is. Note, however, that the proof of BayesGap's simple regret relies on the true rewards for each arm being within the support of the prior, so one cannot increase the algorithm's performance by arbitrarily adjusting the prior. Finally, the κ term is related to the linear relationship between different arms. Additional theoretical results on improving these bounds remains for future work.

5 Experiments

In the following subsections, we benchmark the proposed algorithm against a wide variety of methods on two real-data applications. In Section 5.1, we revisit the traffic sensor network problem of Srinivas et al. [2010]. In Section 5.2, we consider the problem of automatic model selection and algorithm configuration.

5.1 Application to a traffic sensor network

In this experiment, we are given data taken from traffic speed sensors deployed along highway I-880 South in California. Traffic speeds were collected at $K = 357$ sensor locations for all working days between 6AM and 11AM for an entire month. Our task is to identify the single location with the highest expected speed, i.e. the least congested. This data was also used in the work of Srinivas et al. [2010].

Naturally, the readings from different sensors are correlated, however, this correlation is not necessarily only due to geographical location. Therefore specifying a similarity kernel over the space of traffic sensor locations alone would be overly restrictive. Following the approach of Srinivas et al. [2010], we construct the design matrix treating two-thirds of the available data as historical and use the remaining third to evaluate the policies. In more detail, The GP kernel matrix $G \in \mathbb{R}^{K\times K}$ is set to be empirical covariance matrix of measurements for each of the K sensor locations. As explained in Section 4, the corresponding design matrix is $X = VD^{\frac{1}{2}}$, where $G = VDV^T$.

Following Srinivas et al. [2010], we estimate the noise level σ of the observation model using this data. We consider the average empirical variance of each individual sensor (i.e. the signal variance corresponding to the diagonal of G) and set the noise variance σ^2 to 5% of this value; this corresponds to $\sigma^2 = 4.78$. We choose a broad prior with regularization coefficient $\eta = 20$.

In order to evaluate different bandit and Bayesian optimization algorithms, we use each of the remaining 840 sensor signals (the aforementioned third of the data) as the true mean vector μ for independent runs of the experiment. Note that using the model in this way enables us to evaluate the ground truth for each run (given by μ, but not observed by the algorithm), and estimate the actual probability that the policies return the best arm.

In this experiment, as well as in the next one, we estimate the hardness parameter H_ϵ using the adaptive procedure outlined at the end of Section 5.

We benchmark the proposed algorithm (BayesGap) against the following methods:

(1) UCBE: Introduced by Audibert et al. [2010]; this is a variant of the classical UCB policy of Auer et al. [2002] that replaces the $\log(t)$ exploration term of UCB with a constant of order $\log(T)$ for known horizon T.

(2) UGap: A gap-based exploration approach introduced by Gabillon et al. [2012].

(3) BayesUCB and **GPUCB:** Bayesian extensions of UCB which derive their confidence bounds from the posterior. Introduced by Kaufmann et al. [2012a] and Srinivas et al. [2010] respectively.

(4) Thompson sampling: A randomized, Bayesian index strategy wherein the kth arm is selected with probability given by a single-sample Monte Carlo approximation to the posterior probability that the arm is the maximizer [Chapelle and Li, 2012, Kaufmann et al., 2012b, Agrawal and Goyal, 2013].

(5) Probability of Improvement (PI): A clas-

sic Bayesian optimization method which selects points based on their probability of improving upon the current incumbent.

(6) Expected Improvement (EI): A Bayesian optimization, related to PI, which selects points based on the expected value of their improvement.

Note that techniques (1) and (2) above attack the problem of best arm identification and use bounds which encourage more aggressive exploration. However, they do not take correlation into account. On the other hand, techniques such ad (3) are designed for cumulative regret, but model the correlation among the arms. It might seem at first that we are comparing apples and oranges. However, the purpose of comparing these methods, even if their objectives are different, is to understand empirically what aspects of these algorithms matter the most in practical applications. There exist other approaches within the framework of Bayesian nonlinear experimental design, such as [Hennig and Schuler, 2012] for finding maxima and [Kueck et al., 2006, 2009], for learning functions, but we do not consider these here. These approaches involve many approximations, making them expensive and hard to deploy in practice.

The results, shown in Figure 2, are the probabilities of error for each strategy, using a time horizon of $T = 400$. (Here we used $\epsilon = 0$, but varying this quantity had little effect on the performance of each algorithm.) By looking at the results, we quickly learn that techniques that model correlation perform better than the techniques designed for best arm identification, even when they are being evaluated in a best arm identification task. The important conclusion is that one must always invest effort in modelling the correlation among the arms.

The results also show that BayesGap does better than alternatives in this domain. This is not surprising because BayesGap is the only competitor that addresses budgets, best arm identification and correlation simultaneously.

5.2 Automatic machine learning

There exist many machine learning toolboxes, such as `Weka` and `scikit-learn`. However, for a great many data practitioners interested in finding the best technique for a predictive task, it is often hard to understand what each technique in the toolbox does. Moreover, each technique can have many free hyperparameters that are not intuitive to most users.

Bayesian optimization techniques have already been proposed to automate machine learning approaches, such as MCMC inference [Mahendran et al., 2012, Hamze et al., 2013, Wang et al., 2013a], deep learning [Bergstra et al., 2011], preference learning [Brochu et al., 2007, 2010a], reinforcement learning and control [Martinez-Cantin et al., 2007, Lizotte et al., 2012], and more [Snoek et al., 2012]. In fact, methods to automate entire toolboxes (`Weka`) have appeared very recently [Thornton et al., 2013], and go back to old proposals for classifier selection [Maron and Moore, 1994].

Here, we will demonstrate BayesGap by automating regression with `scikit-learn`. Our focus will be on minimizing the cost of cross-validation in the domain of big data. In this setting, training and testing each model can take a prohibitive long time. If we are working under a finite budget, say if we only have three days before a conference deadline or the deployment of a product, we cannot afford to try all models in all cross-validation tests. However, it is possible to use techniques such as BayesGap and Thompson sampling to find the best model with high probability. In our setting, the action of "pulling an arm" will involve selecting a model, splitting the dataset randomly into training and test sets, training the model, and recording the test-set performance.

In this bandit domain, our arms will consist of five `scikit-learn` techniques and associated parameters selected on a discrete grid. We consider the following methods for regression: *Lasso (8 models)* with regularization parameters `alpha` = (0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5), *Random Forests (64 models)* where we vary the number of trees, `n_estimators`=(1,10,100,1000), the minimum number of training examples in a node to split `min_samples_split`=(1,3,5,7) and the minimum number of training examples in a leaf `min_samples_leaf`=(2,6,10,14), *linSVM (16 models)* where we vary the penalty parameter `C`= (0.001, 0.01, 0.1, 1) and the tolerance parameter `epsilon`=(0.0001, 0.001, 0.01, 0.1), *rbfSVM (64 models)* where we use the same grid as above for `C` and `epsilon`, and we add a third parameter which is the length scale γ of the RBF kernel used by the SVM `gamma` $= (0.025, 0.05, 0.1, 0.2)$, and *K-nearest neighbors (8 models)* where we vary number of neighbors `n_neighbors` $= (1, 3, 5, 7, 9, 11, 13, 15)$. The total number of models is 160. Within a class of regressors, we model correlation using a squared exponential kernel with unit length scale, i.e., $k(x, x') = \mathrm{e}^{-(x-x')^2}$. Using this kernel, we compute a kernel matrix G and construct the design matrix as before.

When an arm is pulled we select training and test sets that are each 10% of the size of the original, and ignore the remaining 80% for this particular arm pull. We then train the selected model on the training set, and test on the test set. This specific form of

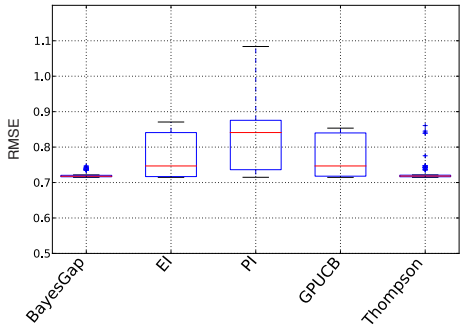

Figure 3: Boxplot of RMSE over 100 runs with a fixed budget of $T = 10$. EI, PI, and GPUCB get stuck in local minima. Note: lower is better.

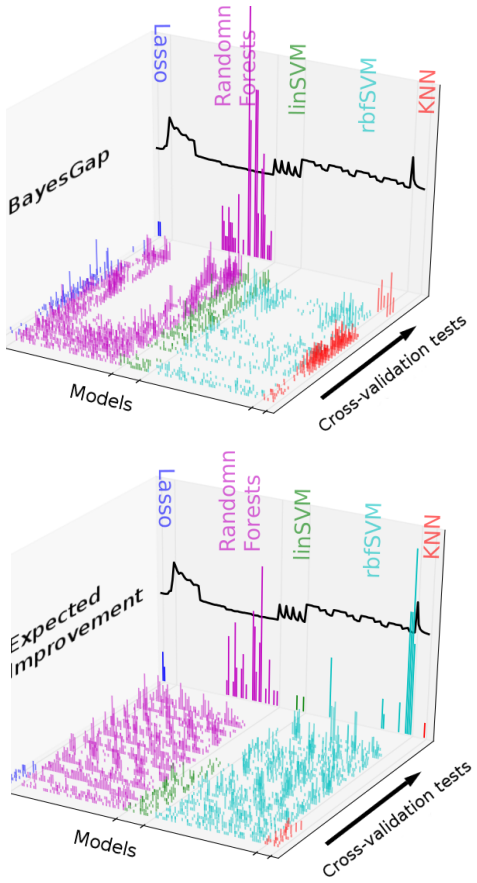

Figure 4: Allocations and recommendations of BayesGap (top) and EI (bottom) over 100 runs at a budget of $T = 40$ training and validation tests, and for 160 models (i.e., more arms than possible observations). Histograms along the floor of the plot show the arms pulled at each round while the histogram on the far wall shows the final arm recommendation over 100 different runs. The solid black line on the far wall shows the estimated "ground truth" RMSE for each model. Note that EI quite often gets stuck in a locally optimal rbfSVM.

cross-validation is similar to that of repeated learning-testing [Arlot and Celisse, 2010, Burman, 1989].

We use the `wine` dataset from the UCI Machine Learning Repository, where the task is to predict the quality score (between 0 and 10) of a wine given 11 attributes of its chemistry. We repeat the experiment 100 times. We report, for each method, an estimate of the RMSE for the recommended models on each run. Unlike in the previous section, we do not have the ground truth generalization error, and in this scenario it is difficult to estimate the actual "probability of error". Instead we report the RMSE, but remark that this is only a proxy for the error rate that we are interested in.

The performance of the final recommendations for each strategy and a fixed budget of $T = 10$ tests is shown in Figure 3. The results for other budgets are almost identical. *It must be emphasized that the number of allowed function evaluations (10 tests) is much smaller than the number of arms (160 models). Hence, frequentist approaches that require pulling all arms, e.g. UGap, are inapplicable in this domain.*

The results indicate that Thompson and BayesGap are the best choices for this domain. Figure 4 shows the individual arms pulled and recommended by BayesGap (above) and EI (bottom), over each of the 100 runs, as well as an estimate of the ground truth RMSE for each individual model. EI and PI often get trapped in local minima. Due to the randomization inherent to Thompson sampling, it explores more, but in a more uniform manner (possibly explaining its poor results in the previous experiment).

6 Conclusion

We proposed a Bayesian optimization method for best arm identification with a fixed budget. The method involves modelling of the correlation structure of the arms via Gaussian process kernels. As a result of combining all these elements, the proposed method outperformed techniques that do not model correlation or that are designed for different objectives (typically cumulative regret). This strategy opens up room for greater automation in practical domains with budget constraints, such as the automatic machine learning application described in this paper.

Although we focused on a Bayesian treatment of the UGap algorithm, the same approach could conceivably be applied to other techniques such as UCBE. As demonstrated by Srinivas et al. [2010] and in this paper, it is possible to easily show that the Bayesian bandits obtain similar bounds as the frequentist methods. However, in our case, we conjecture that much stronger bounds should be possible if we consider all the information brought in by the priors and measurement models.

References

S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In *ICML*, 2013.

S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

J.-Y. Audibert, S. Bubeck, and R. Munos. Best arm identification in multi-armed bandits. In *CoLT*, 2010.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

J. Azimi, A. Fern, and X. Fern. Budgeted optimization with concurrent stochastic-duration experiments. In *NIPS*, pages 1098–1106, 2011.

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *NIPS*, pages 2546–2554, 2011.

E. Brochu, N. de Freitas, and A. Ghosh. Active preference learning with discrete choice data. In *NIPS*, pages 409–416, 2007.

E. Brochu, T. Brochu, and N. de Freitas. A Bayesian interactive optimization approach to procedural animation design. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 103–112, 2010a.

E. Brochu, V. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions. Technical Report arXiv:1012.2599, 2010b.

S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in multi-armed bandits problems. In *International Conference on Algorithmic Learning Theory*, 2009.

P. Burman. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3):pp. 503–514, 1989.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, 2006.

O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In *NIPS*, 2012.

N. de Freitas, A. Smola, and M. Zoghi. Exponential Regret Bounds for Gaussian Process Bandits with Deterministic Observations. In *ICML*, 2012.

V. Gabillon, M. Ghavamzadeh, A. Lazaric, and S. Bubeck. Multi-bandit best arm identification. In *NIPS*, 2011.

V. Gabillon, M. Ghavamzadeh, and A. Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *NIPS*, 2012.

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

P. Hennig and C. Schuler. Entropy search for information-efficient global optimization. *JMLR*, 13:1809–1837, 2012.

M. W. Hoffman, E. Brochu, and N. de Freitas. Portfolio allocation for Bayesian optimization. In *UAI*, pages 327–336, 2011.

F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proceedings of LION-5*, page 507523, 2011.

D. Jones. A taxonomy of global optimization methods based on response surfaces. *J. of Global Optimization*, 21(4):345–383, 2001.

E. Kaufmann and S. Kalyanakrishnan. Information complexity in bandit subset selection. In *Conference on Learning Theory*, pages 228–251, 2013.

E. Kaufmann, O. Cappé, and A. Garivier. On Bayesian upper conf. bounds for bandit problems. In *AIStats*, 2012a.

E. Kaufmann, N. Korda, and R. Munos. Thompson sampling: an asymptotically optimal finite-time analysis. In *International Conference on Algorithmic Learning Theory*, 2012b.

R. Kohavi, R. Longbotham, D. Sommerfield, and R. Henne. Controlled experiments on the web: survey and practical guide. *Data Mining and Knowledge Discovery*, 18:140–181, 2009.

H. Kueck, N. de Freitas, and A. Doucet. SMC samplers for Bayesian optimal nonlinear design. In *IEEE Nonlinear Statistical Signal Processing Workshop*, pages 99–102, 2006.

H. Kueck, M. Hoffman, A. Doucet, and N. de Freitas. Inference and learning for active sensing, experimental design and control. In H. Araujo, A. Mendonca, A. Pinho, and M. Torres, editors, *Pattern Recognition and Image Analysis*, volume 5524, pages 1–10. Springer Berlin Heidelberg, 2009.

D. J. Lizotte, R. Greiner, and D. Schuurmans. An experimental methodology for response surface optimization methods. *Journal of Global Optimization*, 53(4):699–736, 2012.

N. Mahendran, Z. Wang, F. Hamze, and N. de Freitas. Adaptive MCMC with Bayesian optimization. *Journal of Machine Learning Research - Proceedings Track*, 22: 751–760, 2012.

O. Maron and A. W. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. In *NIPS*, pages 59–66, 1994.

R. Martinez-Cantin, N. de Freitas, A. Doucet, and J. A. Castellanos. Active policy learning for robot planning and exploration under uncertainty. 2007.

J. Močkus. The Bayesian approach to global optimization. In *Systems Modeling and Optimization*, volume 38, pages 473–481. Springer, 1982.

R. Munos. Optimistic optimization of a deterministic function without the knowledge of its smoothness. In *NIPS*, pages 783–791, 2011.

K. P. Murphy. *Machine learning: A probabilistic perspective*. Cambridge, MA, 2012.

S. Scott. A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry*, 26(6), 2010.

J. Snoek, H. Larochelle, and R. P. Adams. Opportunity cost in Bayesian optimization. In *Neural Information Processing Systems Workshop on Bayesian Optimization*, 2011.

J. Snoek, H. Larochelle, and R. Adams. Practical Bayesian optimization of machine learning algorithms. In *NIPS*, pages 2960–2968, 2012.

N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *ICML*, 2010.

K. Swersky, J. Snoek, and R. P. Adams. Multi-task Bayesian optimization. In *Neural Information Processing Systems*, 2013.

C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *KDD*, pages 847–855, 2013.

M. Valko, A. Carpentier, and R. Munos. Stochastic simultaneous optimistic optimization. In *ICML*, 2013.

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

Z. Wang, S. Mohamed, and N. de Freitas. Adaptive Hamiltonian and Riemann manifold Monte Carlo samplers. In *ICML*, 2013a.

Z. Wang, M. Zoghi, D. Matheson, F. Hutter, and N. de Freitas. Bayesian optimization in high dimensions via random embeddings. In *IJCAI*, 2013b.