Truncated Variance Reduction: A Unified Approach to Bayesian Optimization and Level-Set Estimation

Authored by:

Andreas Krause
Volkan Cevher
Ilija Bogunovic
Jonathan Scarlett

Abstract

We present a new algorithm, truncated variance reduction (TruVaR), that treats Bayesian optimization (BO) and level-set estimation (LSE) with Gaussian processes in a unified fashion. The algorithm greedily shrinks a sum of truncated variances within a set of potential maximizers (BO) or unclassified points (LSE), which is updated based on confidence bounds. TruVaR is effective in several important settings that are typically non-trivial to incorporate into myopic algorithms, including pointwise costs and heteroscedastic noise. We provide a general theoretical guarantee for TruVaR covering these aspects, and use it to recover and strengthen existing results on BO and LSE. Moreover, we provide a new result for a setting where one can select from a number of noise levels having associated costs. We demonstrate the effectiveness of the algorithm on both synthetic and real-world data sets.

1 Paper Body

Bayesian optimization (BO) [1] provides a powerful framework for automating design problems, and finds applications in robotics, environmental monitoring, and automated machine learning, just to name a few. One seeks to find the maximum of an unknown reward function that is expensive to evaluate, based on a sequence of suitably-chosen points and noisy observations. Numerous BO algorithms have been presented previously; see Section 1.1 for an overview. Level-set estimation (LSE) [2] is closely related to BO, with the added twist that instead of seeking a maximizer, one seeks to classify the domain into points that lie above or below a certain threshold. This is of considerable interest in applications such as environmental monitoring and sensor networks, allowing one to find all ?sufficiently good? points rather than the best point alone. While BO and LSE are

closely related, they are typically studied in isolation. In this paper, we provide a unified treatment of the two via a new algorithm, Truncated Variance Reduction (T RU VA R), which enjoys theoretical guarantees, good computational complexity, and the versatility to handle important settings such as pointwise costs, non-constant noise, and multi-task scenarios. The main result of this paper applies to the former two settings, and even the fixed-noise and unit-cost case, we refine existing bounds via a significantly improved dependence on the noise level. 1.1 Previous Work Three popular myopic techniques for Bayesian optimization are expected improvement (EI), probability of improvement (PI), and Gaussian process upper confidence bound (GP-UCB) [1, 3], each of which chooses the point maximizing an acquisition function depending directly on the current posterior mean and variance. In [4], the GP-UCB-PE algorithm was presented for BO, choosing the highest-variance point within a set of potential maximizers that is updated based on confidence bounds. Another relevant BO algorithm is BaMSOO [5], which also keeps track of potential maximizers, but instead chooses points based on a global optimization technique called simultaneous 1

online optimization (SOO). An algorithm for level-set estimation with GPs is given in [2], which keeps track of a set of unclassified points. These algorithms are computationally efficient and have various theoretical guarantees, but it is unclear how best to incorporate aspects such as pointwise costs and heteroscedastic noise [6]. The same is true for the Straddle heuristic for LSE [7]. Entropy search (ES) [8] and its predictive version [9] choose points to reduce the uncertainty of the location of the maximum, doing so via a one-step lookahead of the posterior rather than only the current posterior. While this is more computationally expensive, it also permits versatility with respect to costs [6], heteroscedastic noise [10], and multi-task scenarios [6]. A recent approach called minimum regret search (MRS) [11] also performs a look-ahead, but instead chooses points to minimize the regret. To our knowledge, no theoretical guarantees have been provided for these. The multi-armed bandit (MAB) [12] literature has developed alongside the BO literature, with the two often bearing similar concepts. The MAB literature is far too extensive to cover here, but we briefly mention some variants relevant to this paper. Extensive attention has been paid to the bestarm identification problem [13], and cost constraints have been incorporated in a variety of forms [14]. Moreover, the concept of ?zooming in? to the optimal point has been explored [15]. In general, the assumptions and analysis techniques in the MAB and BO literature are quite different. 1.2

Contributions

We present a unified analysis of Bayesian optimization and level-set estimation via a new algorithm Truncated Variance Reduction (T RU VA R). The algorithm works by keeping track of a set of potential maximizers (BO) or unclassified points (LSE), selecting points that shrink the uncertainty within that set up to a truncation threshold, and updating the set using confidence bounds. Similarly to ES and MRS, the algorithm performs a one-step lookahead that is highly beneficial in terms of versatility. However, unlike these previous works, our lookahead avoids the computationally expensive task of averaging over the

posterior distribution and the observations. Also in contrast with ES and MRS, we provide theoretical bounds for T RU VA R characterizing the cost required to achieve a certain accuracy in finding a near-optimal point (BO) or in classifying each point in the domain (LSE). By applying this to the standard BO setting, we not only recover existing results [2, 4], but we also strengthen them via a significantly improved dependence on the noise level, with better asymptotics in the small noise limit. Moreover, we provide a novel result for a setting in which the algorithm can choose the noise level, each coming with an associated cost. Finally, we compare our algorithm to previous works on several synthetic and real-world data sets, observing it to perform favorably in a variety of settings.

2

Problem Setup and Proposed Algorithm

Setup: We seek to sequentially optimize an unknown reward function f (x) over a finite domain D.1 At time t, we query a single point xt 2 D and observe a noisy sample yt = f (xt) + zt , where zt ? N (0, 2 (xt)) for some known noise function 2 (?) : D ! R+ . Thus, in general, some points may be noisier than others, in which case we have heteroscedastic noise [10]. We associate with each point a cost according to some known cost function c : D ! R+ . If both 2 (?) and c(?) are set to be constant, then we recover the standard homoscedastic and unit-cost setting. We model f (x) as a Gaussian process (GP) [16] having mean zero and kernel function k(x, x0), normalized so that k(x, x) = 1 for all x 2 D. The posterior distribution of f given the points and observations up to time t is again a GP, with the posterior mean and variance given by [10] ?t (x) = kt (x)T Kt + ?t

1

(1)

yt 1

= k(x, x) kt (x)T Kt + ?t kt (x), (2) ? ?t ? ? where kt (x) = k(xi , x) i=1 , Kt = k(xt , xt0) t,t0 , and ?t = diag(2 (x1), . . . , 2 (xt)). We also let t2 1—x (x) denote the posterior variance of x upon observing x along with x1 , ? ? ? , xt 1 . t (x)

1

2

Extensions to continuous domains are discussed in the supplementary material.

2

Confidence

Selected point

Target Max. lower bound

Potential maximizers

(a) t = 6

(b) t = 7

(c) t = 8

(d) t = 9

Figure 1: An illustration of the T RU VA R algorithm. In (a), (b), and (c), three points within the set of potential maximizers Mt are selected in order to

bring the confidence bounds to within the target range, and Mt shrinks during this process. In (d), the target confidence width shrinks as a result of the last selected point bringing the confidence within Mt to within the previous target. We consider both Bayesian optimization, which consists of finding a point whose function value is as high as possible, and level-set estimation, which consists of classifying the domain according into points that lie above or below a given threshold h. The precise performance criteria for these settings are given in Definition 3.1 below. Essentially, after spending a certain cost we report a point (BO) or a classification (LSE), but there is no preference on the values of f (xt) for the points xt chosen before coming to such a decision (in contrast with other notions such as cumulative regret). T RU VA R algorithm: Our algorithm is described in Algorithm 1, making use of the updates described in Algorithm 2. The algorithm keeps track of a sequence of unclassified points Mt , representing potential maximizers for BO or points close to h for LSE. This set is updated based on the confidence bounds depending on constants (i) . The algorithm proceeds in epochs, 1/2

where in the i-th epoch it seeks to bring the confidence (i) t (x) of points within Mt below a target value ?(i) . It does this by greedily minimizing the sum of truncated variances P 2 x2Mt 1 max{ (i) t 1—x (x), ?(i) } arising from choosing the point x, along with a normalization and division by c(x) to favor low-cost points. The truncation by ?(i) in this decision rule means that once the confidence of a point is below the current target value, there is no preference in making it any lower (until the target is decreased). Once the confidence of every point in Mt is less than a factor 1 + above the target value, the target confidence is reduced according to a multiplication by r 2 (0, 1). An illustration of the process is given in Figure 1, with details in the caption. For level-set estimation, we also keep track of the sets Ht and Lt , containing points believed to have function values above and below h, respectively. The constraint x 2 Mt 1 in (5)?(7) ensures that {Mt } is non-increasing with respect to inclusion, and Ht and Lt are non-decreasing. Algorithm 1 Truncated Variance Reduction (T RU VA R) Input: Domain D, GP prior (?0 , 0 , k), confidence bound parameters ¿ 0, r 2 (0, 1), { (i) }i 1 , ?(1) ¿ 0, and for LSE, level-set threshold h 1: Initialize the epoch number i = 1 and potential maximizers M(0) = D. 2: for t = 1, 2, . . . do 3: Choose P P 2 2 2 2 x2Mt 1 max{ (i) t 1 (x), ?(i) } x2Mt 1 max{ (i) t 1—x (x), ?(i) } xt = arg max . (3) c(x) x2D 4: 5: 6:

Observe the noisy function sample yt , and update according to Algorithm 2 to obtain Mt , ?t , t , lt and ut , as well as Ht and Lt in the case of LSE 1/2 while maxx2Mt (i) t (x) ? (1 +)?(i) do Increment i, set ?(i) = r ? ?(i 1) .

The choices of (i) , , and r are discussed in Section 4. As with previous works, the kernel is assumed known in our theoretical results, whereas in practice it is typically learned from training data [3]. Characterizing the effect of model mismatch or online hyperparameter updates is beyond the scope of this paper, but is an interesting direction for future work. 3

Algorithm 2 Parameter Updates for T RU VA R Input: Selected points and observations {xt0 }tt0 =1 ; {yt0 }tt0 =1 , previous sets Mt 1 , Ht 1 , Lt 1/2 parameter (i) , and for LSE, level-set threshold h. 1: Update ?t and t according

to (1)?(2), and form the upper and lower confidence bounds 1/2

2: For BO, set

or for LSE, set H t = Ht

Mt = x 2 Mt

1

1/2

ut (x) = ?t (x) + (i) t (x), ‘t (x) = ?t (x) (i) ? Mt = x 2 Mt 1 : ut (x) max ‘t (x) , x2Mt 1

[x 2 Mt

1

1

: ut (x)

: ‘t (x) ¿ h ,

t (x).

1

(4) (5)

h and ‘t (x) ? h

Lt = L t

1,

(6)

[x 2 Mt

1

: ut (x) ¡ h .

(7)

Some variants of our algorithm and theory are discussed in the supplementary material due to lack of space, including pure variance reduction, non-Bayesian settings [3], continuous domains [3], the batch setting [4], and implicit thresholds for level-set estimation [2].

3

Theoretical Bounds

In order to state our results for BO and LSE in a unified fashion, we define a notion of ?-accuracy for the two settings. That is, we define this term differently in the two scenarios, but then we provide theorems that simultaneously apply to both. All proofs are given in the supplementary material. Definition 3.1. After time step t of T RU VA R, we use the following terminology: ? For BO, the set Mt is ?-accurate if it contains all true maxima x? 2 arg maxx f (x), and all of its points satisfy f (x?) f (x) ? ?.

? For LSE, the triplet (Mt , Ht , Lt) is ?-accurate if all points in Ht satisfy f (x) ¿ h, all points in Lt satisfy f (x) ¡ h, and all points in Mt satisfy —f (x) h— ? 2? . Pt In both cases, the cumulative cost after time t is defined as Ct = t0 =1 c(xt0).

We use 2? in the LSE setting instead of ? since this creates a region of size ? where the function value lies, which is consistent with the BO setting. Our performance criterion for level-set estimation is slightly different from that of [2], but the two are closely related. 3.1

General Result

Preliminary definitions: Suppose that the { (i) } are chosen to ensure valid confidence bounds, i.e., lt (x) ? f (x) ? ut (x) with high probability; see Theorem 3.1 and its proof below for such choices. In this case, we have after the i-th epoch that all points are either already discarded (BO) or classified (LSE), or are known up to the confidence level (1 +)?(i) . For the points with such confidence, we have ut (x) lt (x) ? 2(1 +)?(i) , and hence ut (x) ? lt (x) + 2(1 +)?(i) ? f (x) + 2(1 +)?(i) ,

(8)

and similarly lt (x) f (x) 2(1 +)?(i) . This means that all points other than those within a gap of width 4(1 +)?(i) must have been discarded or classified: Mt ? x : f (x)

Mt ? x : —f (x)

f (x?)

4(1 +)?(i) =: M (i)

h— ? 2(1 +)?(i) =: M (i)

(LSE)

Since no points are discarded or classified initially, we define M (0) = D. 4

(BO)

(9) (10)

For a collection of points S = (x01 , . . . , x0—S—), possibly containing duplicates, we write the total cost P—S— as c(S) = i=1 c(x0i). Moreover, we denote the posterior variance upon observing the points up to time t 1 and the additional points in S by t 1—S (x). Therefore, c(x) = c({x}) and t 1—x (x) = t 1—{x} (x). The minimum cost (respectively, maximum cost) is denoted by cmin = minx2D c(x) (respectively, cmax = maxx2D c(x)). Finally, we introduce the quantity

n o C ? (?, M) = min c(S) : max 0—S (x) ? ? , (11) S x2M representing the minimum cost to achieve a posterior standard deviation of at most ? within M . Main result: In all of our results, we make the following assumption. Assumption 3.1. The kernel k(x, x0) is such that the variance reduction function t,x (S)

2 t (x)

=

2 t—S (x)

(12)

is submodular [17] for any time t, and any selected points (x1 , . . . , xt) and query point x. This assumption has been used in several previous works based on Gaussian processes, and sufficient conditions for its validity can be found in [18, Sec. 8]. We now state the following general guarantee. Theorem 3.1. Fix ? ¿ 0 and 2 (0, 1), and suppose there exist values {C(i) } and { (i) } such that ? ? ?(i) —M (i 1) — (i) C(i) C ? , M + cmax , (13) (i 1) log 2 2 1/2 ?(i) (i) and (i)

2 log

Then if T RU VA R is run with these choices of C? =

P

—D—

i0 ?i C(i0) 6 c2min
?
(14)
.
until the cumulative cost reaches X C(i) ,
(i)
i : 4(1+)?(i
then with probability at least 1
2 2
(15)
1) ¿?
, we have ?-accuracy.

While this theorem is somewhat abstract, it captures the fact that the algorithm improves when points having a lower cost and/or lower noise are available, since both of these lead to a smaller value of C ? (?, M); the former by directly incurring a smaller cost, and the latter by shrinking the variance more rapidly. Below, we apply this result to some important cases. 3.2

Results for Specific Settings

Homoscedastic and unit-cost setting: Define the maximum mutual information [3] 1 log det IT + 2 KT , (16) T = max x1 ,...,xT 2 and consider the case that 2 (x) = 2 and c(x) = 1. In the supplementary material, we provide a 1 theorem with a condition for ?-accuracy of the form T ?? C1 ?T2 T + 1 with C1 = log(1+ 2) , thus matching [2, 4] up to logarithmic factors. In the following, we present a refined version that has a significantly better dependence on the noise level, thus exemplifying that a more careful analysis of (13) can provide improvements over the standard bounding techniques. 2

2

? Corollary 3.1. Fix ? ¿ 0 and 2 (0, 1), define T = 2 log —D—T , and set ?(1) = 1 and r = 12 . 6 There exist choices of (i) (not depending on the time horizon T) such that we have ?-accuracy with probability at least 1 once the following condition holds: ? ? l 96(1 +)2 6(1 +)2 32(1 +)2 m 16(1 +)2 —D— T 2 T 2 + C + 2 log log , T T 1 T T 2 2 2 ?2 ? ?2 (17)

where C1 =
1 log(1+
2)
. This condition is of the form T 5
??
2
T
?2
T
+
C1
T 2
T
+1 .

The choices ?(1) = 1 and r = 12 are made for mathematical convenience, and a similar result follows for any other choices ?(1) ¿ 0 and r 2 (0, 1), possibly with different constant factors. As 2 ! 1 (i.e., high noise), both of the above-mentioned bounds have noise dependence O? (2), since log(1 + ? 1) = O(? 1) as ? ! 1. On the other hand, as 2 ! 0 (i.e., low noise), C1 is logarithmic, and Corollary 3.1 is significantly better provided that ? ? .

Choosing the noise and cost: Here we consider the setting that there is a domain of points D0 that the reward function depends on, and alongside each point we can choose a noise variance 2 (k) (k = 1, . . . , K). Hence, D = D0 ?{1, ? ? ? , K}. Lower noise variances incur a higher cost according to a cost function c(k). Corollary 3.2. For each k = 1, ? ? ? , K, let T ? (k) denote the smallest value of T such that (17) holds —D—T 2 c2 ? 2 with 2 (k) in place of 2 , and with T = 2 log 6 c2max . Then, under the preceding setting, there min exist choices of (i) (not depending on T) such that we have ?-accuracy with probability at least 1 once the cumulative cost reaches mink c(k)T ? (k). This result roughly states that we obtain a bound as good as that obtained by sticking to any fixed choice of noise level. In other words, every choice of noise (and corresponding cost) corresponds to a different version of a BO or LSE algorithm (e.g., [2, 4]), and our algorithm has a similar performance guarantee to the best among all of those. This is potentially useful in avoiding the need for running an algorithm once per noise level and then choosing the best-performing one. Moreover, we found numerically that beyond matching the best fixed noise strategy, we can strictly improve over it by mixing the noise levels; see Section 4.

4

Experimental Results

We evaluate our algorithm in both the level-set estimation and Bayesian optimization settings. Parameter choices: As with previous GP-based algorithms that use confidence bounds, our theoretical choice of (i) in T RU VA R is typically overly conservative. Therefore, instead of using (14) directly, we use a more aggressive variant with similar dependence on the domain size and time: 2 (i) = a log(—D—t(i)), where t(i) is the time at which the epoch starts, and a is a constant. Instead of the choice a = 2 dictated by (14), we set a = 0.5 for BO to avoid over-exploration. We found exploration to be slightly more beneficial for LSE, and hence set a = 1 for this setting. We found T RU VA R to be quite robust with respect to the choices of the remaining parameters, and simply set ?(1) = 1, r = 0.1, and = 0 in all experiments; while our theory assumes ¿ 0, in practice there is negligible difference between choosing zero and a small positive value. Level-set estimation: For the LSE experiments, we use a common classification rule in all al? t = {x : ?t (x) gorithms, classifying the points according to the posterior mean as H h} and ? Lt = {x : ?t (x) ¡ h}. The classification accuracy is measured by the F1 -score (i.e., the harmonic mean of precision and recall) with respect to the true super- and sub-level sets. We compare T RU VA R against the GP-based LSE algorithm [2], which we name via the authors? surnames as GCHK, as well as the state-of-the-art straddle (STR) heuristic [7] and the maximum variance rule (VAR) [2]. Descriptions can be found in the

supplementary material. GCHK includes 1/2 an exploration constant t , and we follow the recommendation in [2] of setting t = 3. Lake data (unit cost): We begin with a data set from the domain of environmental monitoring of inland waters, consisting of 2024 in situ measurements of chlorophyll concentration within a vertical transect plane, collected by an autonomous surface vessel in Lake Z?urich [19]. As in [2], our goal is to detect regions of high concentration. We evaluate each algorithm on a 50 ? 50 grid of points, with the corresponding values coming from the GP posterior that was derived using the original data (see Figure 2d). We use the Mat?ern-5/2 ARD kernel, setting its hyperparameters by maximizing the likelihood on the second (smaller) available dataset. The level-set threshold h is set to 1.5. In Figure 2a, we show the performance of the algorithms averaged over 100 different runs; here the randomness is only with respect to the starting point, as we are in the noiseless setting. We observe that in this unit-cost case, T RU VA R performs similarly to GCHK and STR. All three methods outperform VAR, which is good for global exploration but less suited to level-set estimation. 6

1
0.8
0.8
0.9
0.6 0.4 TruVaR GCHK STR VAR
0.2
F1 score
1
F1 score
F1 score
1
0.6 0.4 0.2
0
20
40
60 Time
80
100
TruVaR GCHK high noise GCHK medium noise GCHK small noise
0.5 0
120
0.7 0.6
TruVaR GCHK
0
0
0.8
0.5
1 Cost (?104)
1.5
2

0.1
0.15
?104
0.2 0.25 0.3 Cost (?104)
0.35
0.4
(a) Lake data, unit-cost
(b) Lake data, varying cost
(c) Synthetic data, varying noise
(d) Inferred concentration function
(e) Points chosen by GCHK
(f) Points chosen by T RU VA R
Figure 2: Experimental results for level-set estimation. 0.27
Averaged Regret
Median Regret
10-2
10-4
TruVaR EI GP-UCB
TruVaR EI GP-UCB ES MRS
100
Validation Error
TruVaR EI GP-UCB ES MRS
100
10-2
10-4
0.26
0.25
10-6 10-6 0
20
40
60
80
100
120
0.24 0
20
40
60
80
100
120
0
20
Time
Time
(a) Synthetic, median

(b) Synthetic, outlier-adjusted mean
40
60
80
100
Time
(c) SVM data

Figure 3: Experimental results for Bayesian optimization. Lake data (varying cost): Next, we modify the above setting by introducing pointwise costs that are a function of the previous sampled point x0 , namely, cx0 (x) = 0.25—x1 x01 — + 4(—x2 — + 1), where x1 is the vessel position and x2 is the depth. Although we did not permit such a dependence on x0 in our original setup, the algorithm itself remains unchanged. Our choice of cost penalizes the distance traveled —x1 x01 —, as well as the depth of the measurement —x2 —. Since incorporating costs into existing algorithms is non-trivial, we only compare against the original version of GCHK that ignores costs. In Figure 2b, we see that TruVaR significantly outperforms GCHK, achieving a higher F1 score for a significantly smaller cost. The intuition behind this can be seen in Figures 2e and 2f, where we show the points sampled by TruVaR and GCHK in one experiment run, connecting all pairs of consecutive points. GCHK is designed to pick few points, but since it ignores costs, the distance traveled is large. In contrast, by incorporating costs, T RU VA R tends to travel small distances, often even staying in the same x1 location to take measurements at multiple depths x2 . Synthetic data with multiple noise levels: In this experiment, we demonstrate Corollary 3.2 by considering the setting in which the algorithm can choose the sampling noise variance and incur the associated cost. We use a synthetic function sampled from a GP on a 50 ? 50 grid with an isotropic squared exponential kernel having length scale l = 0.1 and unit variance, and set h = 2.25. We use three different noise levels, 2 2 {10 6 , 10 3 , 0.05}, with corresponding costs {15, 10, 2}. We run GCHK separately for each of the three noise levels, while running T RU VA R as normal and allowing it to mix between the noise levels. The resulting F1 -scores are shown in Figure 2c. The best-performing version of GCHK changes throughout the time horizon, while T RU VA R is consistently better than all three. A discussion on how T RU VA R mixes between the noise levels can be found in the supplementary material. 7

Bayesian optimization. We now provide the results of two experiments for the BO setting. Synthetic data: We first conduct a similar experiment as that in [8, 11], generating 200 different test functions defined on [0, 1]2 . To generate a single test function, 200 points are chosen uniformly at random from [0, 1]2 , their function values are generated from a GP using an isotropic squared exponential kernel with length scale l = 0.1 and unit variance, and the resulting posterior mean forms the function on the whole domain [0, 1]2 . We subsequently assume that samples of this function are corrupted by Gaussian noise with 2 = 10 6 . The extension of T RU VA R to continuous domains is straightforward, and is explained in the supplementary material. For all algorithms

considered, we evaluate the performance according to the regret of a single reported point, namely, the one having the highest posterior mean. We compare the performance of T RU VA R against expected improvement (EI), GP-upper confidence bound (GP-UCB), entropy search (ES) and minimum regret search (MRS), whose acquisition functions are outlined in the supplementary material. We use publicly available code for ES and MRS [20]. The exploration parameter t in GP-UCB is set according to the recommendation in [3] of dividing the theoretical value by five, and the parameters for ES and MRS are set according to the recommendations given in [11, Section 5.1]. Figure 3a plots the median of the regret, and Figure 3b plots the mean after removing outliers (i.e., the best and worst 5% of the runs). In the earlier rounds, ES and MRS provide the best performance, while T RU VA R improves slowly due to exploration. However, the regret of T RU VA R subsequently drops rapidly, giving the best performance in the later rounds after ?zooming in? towards the maximum. GP-UCB generally performs well with the aggressive choice of t , despite previous works? experiments revealing it to perform poorly with the theoretical value. Hyperparameter tuning data: In this experiment, we use the SVM on grid dataset, previously used in [21]. A 25 ? 14 ? 4 grid of hyperparameter configurations resulting in 1400 data points was preevaluated, forming the search space. The goal is to find a configuration with small validation error. We use a Mat?ern-5/2 ARD kernel, and re-learn its hyperparameters by maximizing the likelihood after sampling every 3 points. Since the hyperparameters are not fixed in advance, we replace Mt 1 by D in (5) to avoid incorrectly ruling points out early on, allowing some removed points to be added again in later steps. Once the estimated hyperparameters stop to vary significantly, the size of the set of potential maximizers decreases almost monotonically. Since we consider the noiseless setting here, we measure performance using the simple regret, i.e., the best point found so far. We again average over 100 random starting points, and plot the resulting validation error in Figure 3c. Even in this noiseless and unit-cost setting that EI and GP-UCB are suited to, we find that T RU VA R performs slightly better, giving a better validation error with smaller error bars.

5

Conclusion

We highlight the following aspects in which T RU VA R is versatile: ? Unified optimization and level-set estimation: These are typically treated separately, whereas T RU VA R and its theoretical guarantees are essentially identical in both cases ? Actions with costs: T RU VA R naturally favors cost-effective points, as this is directly incorporated into the acquisition function. ? Heteroscedastic noise: T RU VA R chooses points that effectively shrink the variance of other points, thus directly taking advantage of situations in which some points are noisier than others. ? Choosing the noise level: We provided novel theoretical guarantees for the case that the algorithm can choose both a point and a noise level, cf., Corollary 3.2. Hence, T RU VA R directly handles several important aspects that are non-trivial to incorporate into myopic algorithms. Moreover, compared to other BO algorithms that perform a lookahead (e.g., ES and MRS), T RU VA R avoids the computationally expensive task

of averaging over the posterior and/or measurements, and comes with rigorous theoretical guarantees. Acknowledgment: This work was supported in part by the European Commission under Grant ERC Future Proof, SNF Sinergia project CRSII2-147633, SNF 200021-146750, and EPFL Fellows Horizon2020 grant 665667. 8

2 References

[1] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, ?Taking the human out of the loop: A review of Bayesian optimization,? Proc. IEEE, vol. 104, no. 1, pp. 148?175, 2016. [2] A. Gotovos, N. Casati, G. Hitz, and A. Krause, ?Active learning for level set estimation,? in Int. Joint. Conf. Art. Intel., 2013. [3] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, ?Information-theoretic regret bounds for Gaussian process optimization in the bandit setting,? IEEE Trans. Inf. Theory, vol. 58, no. 5, pp. 3250?3265, May 2012. [4] E. Contal, D. Buffoni, A. Robicquet, and N. Vayatis, Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg, 2013, ch. Parallel Gaussian Process Optimization with Upper Confidence Bound and Pure Exploration, pp. 225?240. [5] Z. Wang, B. Shakibi, L. Jin, and N. de Freitas, ?Bayesian multi-scale optimistic optimization,? http://arxiv.org/abs/1402.7005. [6] K. Swersky, J. Snoek, and R. P. Adams, ?Multi-task Bayesian optimization,? in Adv. Neur. Inf. Proc. Sys. (NIPS), 2013, pp. 2004?2012. [7] B. Bryan and J. G. Schneider, ?Actively learning level-sets of composite functions,? in Int. Conf. Mach. Learn. (ICML), 2008. [8] P. Hennig and C. J. Schuler, ?Entropy search for information-efficient global optimization,? J. Mach. Learn. Research, vol. 13, no. 1, pp. 1809?1837, 2012. [9] J. M. Hern?andez-Lobato, M. W. Hoffman, and Z. Ghahramani, ?Predictive entropy search for efficient global optimization of black-box functions,? in Adv. Neur. Inf. Proc. Sys. (NIPS), 2014, pp. 918?926. [10] P. W. Goldberg, C. K. Williams, and C. M. Bishop, ?Regression with input-dependent noise: A Gaussian process treatment,? Adv. Neur. Inf. Proc. Sys. (NIPS), vol. 10, pp. 493?499, 1997. [11] J. H. Metzen, ?Minimum regret search for single-and multi-task optimization,? in Int. Conf. Mach. Learn. (ICML), 2016. [12] S. Bubeck and N. Cesa-Bianchi, Regret Analysis of Stochastic and Nonstochastic Multi-Armed Bandit Problems, ser. Found. Trend. Mach. Learn. Now Publishers, 2012. [13] K. Jamieson and R. Nowak, ?Best-arm identification algorithms for multi-armed bandits in the fixed confidence setting,? in Ann. Conf. Inf. Sci. Sys. (CISS), 2014, pp. 1?6. [14] O. Madani, D. J. Lizotte, and R. Greiner, ?The budgeted multi-armed bandit problem,? in Learning Theory. Springer, 2004, pp. 643?645. [15] R. Kleinberg, A. Slivkins, and E. Upfal, ?Multi-armed bandits in metric spaces,? in Proc. ACM Symp. Theory Comp., 2008. [16] C. E. Rasmussen, ?Gaussian processes for machine learning.? MIT Press, 2006. [17] A. Krause and D. Golovin, ?Submodular function maximization,? Tractability: Practical Approaches to Hard Problems, vol. 3, 2012. [18] A. Das and D. Kempe, ?Algorithms for subset selection in linear regression,? in Proc. ACM Symp. Theory Comp. (STOC). ACM, 2008, pp.

45?54. [19] G. Hitz, F. Pomerleau, M.-E. Garneau, E. Pradalier, T. Posch, J. Pernthaler, and R. Y. Siegwart, ?Autonomous inland water monitoring: Design and application of a surface vessel,? IEEE Robot. Autom. Magazine, vol. 19, no. 1, pp. 62?72, 2012. [20] http://github.com/jmetzen/bayesian optimization (accessed 19/05/2016). [21] J. Snoek, H. Larochelle, and R. P. Adams, ?Practical Bayesian optimization of machine learning algorithms,? in Adv. Neur. Inf. Proc. Sys., 2012. [22] K. Swersky, J. Snoek, and R. P. Adams, ?Freeze-thaw Bayesian optimization,? 2014, http://arxiv.org/abs/1406.3896. [23] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, ?Lipschitzian optimization without the Lipschitz constant,? J. Opt. Theory Apps., vol. 79, no. 1, pp. 157?181, 1993. [24] A. Krause and C. Guestrin, ?A note on the budgeted maximization of submodular functions,? 2005, Technical Report. 9