Artificial Intelligence 243 (2017) 45–56

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Query efficient posterior estimation in scientific experiments via Bayesian active learning [☆]

Kirthevasan Kandasamy [*], Jeff Schneider, Barnabás Póczos

Carnegie Mellon University, Pittsburgh, PA, USA

ARTICLE INFO

Article history:
Received 10 January 2016
Received in revised form 22 September 2016
Accepted 27 November 2016
Available online 30 November 2016

Keywords:
Posterior estimation
Active learning
Gaussian processes

ABSTRACT

A common problem in disciplines of applied Statistics research such as Astrostatistics is of estimating the posterior distribution of relevant parameters. Typically, the likelihoods for such models are computed via expensive experiments such as cosmological simulations of the universe. An urgent challenge in these research domains is to develop methods that can estimate the posterior with few likelihood evaluations.

In this paper, we study active posterior estimation in a Bayesian setting when the likelihood is expensive to evaluate. Existing techniques for posterior estimation are based on generating samples representative of the posterior. Such methods do not consider efficiency in terms of likelihood evaluations. In order to be query efficient we treat posterior estimation in an active regression framework. We propose two myopic query strategies to choose where to evaluate the likelihood and implement them using Gaussian processes. Via experiments on a series of synthetic and real examples we demonstrate that our approach is significantly more query efficient than existing techniques and other heuristics for posterior estimation.

1. Introduction

Computing the posterior distribution of parameters given observations is a central problem in Bayesian statistics. We use the posterior distribution to make inferences about likely parameter values and estimate functionals we are interested in. For simple parametric models with conjugate priors, we may obtain the posterior in analytic form. In more complex models where the posterior is analytically intractable, we have to resort to approximation techniques. In some cases, we only have access to a black box which computes the likelihood for a given value of the parameters.

Our goal is an efficient way to estimate posterior densities when calls to this black box are expensive. This work is motivated by applications in computational physics and cosmology. Several cosmological phenomena are characterised by the cosmological parameters (e.g. Hubble constant, dark energy fraction). Given observations, we wish to make inferences about the parameters. Physicists have developed simulation-based probability models of the Universe which can be used to compute the likelihood of cosmological parameters for a given observation. Fig. 1 shows different scenarios to estimate/compute the likelihood. Many problems in scientific computing have a similar flavour. Expensive simulators in molecular mechanics, computational biology and neuroscience are used to model many scientific processes. Hence this work finds relevance in these fields as well.

[☆] This paper is an invited revision of a paper which first appeared at the International Joint Conference of Artificial Intelligence [Ijcai 2015].

[*] Corresponding author.
E-mail address: kandasamy@cs.cmu.edu (K. Kandasamy).

http://dx.doi.org/10.1016/j.artint.2016.11.002

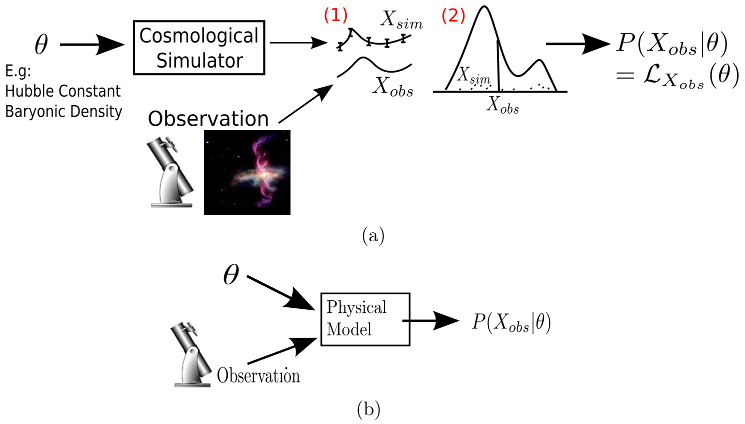


Fig. 1. Illustrations of cosmological experiments. (a): Given a set of values for the parameters θ the oracle produces several simulations $\mathbf{X_{sim}}$. The likelihood $P(\mathbf{X_{obs}}|\theta)$ can then be estimated, say, by (1) comparing $\mathbf{X_{sim}}$ to the $\mathbf{X_{obs}}$ via a statistical test or (2) constructing a density estimate using $\mathbf{X_{sim}}$ and then evaluating this estimate at $\mathbf{X_{obs}}$. (b): The oracle directly computes the likelihood using a physical model of the universe.

Related work

Practitioners have conventionally used sampling schemes [16] to approximate the posterior distributions. Rejection sampling and various MCMC methods are common choices. The advantage of MCMC approaches is their theoretical guarantees with large sample sets [24] and thus they are a good choice when likelihood evaluations are cheap. However, none of them is intended to be query efficient when evaluations are expensive. Some methods spend most of their computation evaluating point likelihoods and then discard the likelihood values after doing an acceptance test. This gives insight into the potential gains possible by retaining those likelihoods for use in regression. Despite such deficiencies, MCMC remains one of the most popular techniques for posterior estimation in experimental physics [6,21,13] and the other fields [14].

Approximate Bayesian computation (ABC) [17,18] is a method of last resort for estimating posteriors when a likelihood can not be computed. Unfortunately, it still requires the generation of simulated data, which is expensive in our setup, and it does not address efficient selection of parameter values to be tested at all. Nested Sampling [27] is a technique commonly used in Astrostatistics. Kernel Bayes' Rule [7] is a non-parametric method of computing a posterior based on the embedding of probabilities in an RKHS. Both these methods require sampling from a distribution and do not address the question of which samples to choose if generating them is expensive. The work in Bryan et al. [3] actively learns level sets of an expensive function and derives confidence sets from the results. Gotovos et al. [8] also actively learn level sets via a classification approach. Our work is more general since we estimate the entire posterior.

Our methods draw inspiration from Gaussian Process (GP) based active learning methods such as Bayesian optimisation (BO) [19], Bayesian quadrature (BQ) [20], active GP Regression (AGPR) [25] and several others [28,15,12,11,10]. These methods have a common modus operandi to determining the experiment θ_t at time step t: construct a utility function u_t based on the posterior GP conditioned on the queries so far; then maximise u_t to determine θ_t. $u_t(\theta)$ captures the value of performing an experiment at point θ. Maximising the typically multimodal u_t is itself a hard problem. Further, inference in GPs can also be quite expensive, especially when we have several queries. However, it is generally assumed that querying the function is more costly than the effort to determine the next experiment [2,28]. The key difference in such methods is essentially in the specification of u_t to determine the next experiment. In our work, we adopt this strategy. We present two utility functions for active posterior estimation.

Our contribution is to propose a query efficient method for estimating posterior densities when the likelihood is expensive to evaluate. We adopt a Bayesian active regression approach on the log likelihood using the samples it has already computed. We refer to this approach as Bayesian Active Posterior Estimation (BAPE). We propose two myopic query strategies on the uncertainty regression model for sample selection. Our implementation uses Gaussian processes [23] and we demonstrate the efficacy of the methods on multiple synthetic and real experiments.

The remainder of this manuscript is organised as follows. We begin with a brief review of GPs in Section 2.1. We present our methods in Sections 2.2, 2.3, 2.4 and compare them against non-active strategies and GP based active learning methods in Section 2.5. In Section 3 we discuss alternatives for empirical evaluation and in Section 4 we present experimental results on synthetic and real problems.

2. Bayesian active posterior estimation

2.1. Gaussian processes

A GP over a space Θ is a random process describing functions from Θ to $\mathbb{R}$. It is completely characterised by its mean function $\mu : \Theta \to \mathbb{R}$ and covariance kernel $k : \Theta \times \Theta \to \mathbb{R}$. The function values $f(\theta_1), \ldots, f(\theta_n)$ at any finite set of n points $\{\theta_1, \ldots, \theta_n\} \subset \Theta$ are distributed jointly Gaussian $\mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$. Here $\boldsymbol{\mu} \in \mathbb{R}^n$, $\boldsymbol{\mu}_i = \mu(\theta_i)$ and $\mathbf{K} \in \mathbb{R}^{n\times n}$ where $\mathrm{K}_{ij} = k(\theta_i, \theta_j)$.

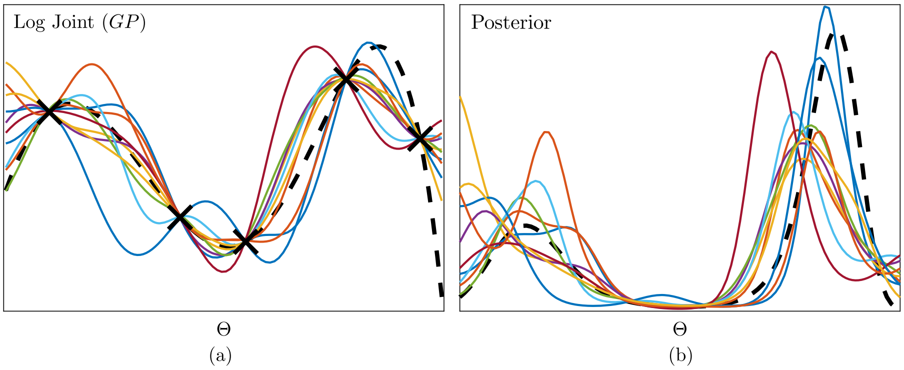


Fig. 2. (a) depicts the uncertainty for the log joint probability via samples g drawn from the GP. (b) illustrates the induced uncertainty model $F_{\theta|\mathbf{X_{obs}}}$ for the posterior via the exponentiated and normalised samples $f = \exp g / \int \exp g$.

Typically, a GP is specified by its prior mean μ_0 and kernel k_0. Given observations $\mathbf{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$ at points $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_n\}$ the posterior is also a GP with mean μ' and covariance k' given by,

$$\mu'(\theta) = \mu_0(x) + k_0(\boldsymbol{\theta}, \theta)^\top \mathbf{K}_0^{-1}(\mathbf{y} - \mu_0(\boldsymbol{\theta})),$$
$$k'(\theta, \theta') = k_0(\theta, \theta') - k_0(\boldsymbol{\theta}, \theta)^\top \mathbf{K}_0^{-1} k_0(\boldsymbol{\theta}, \theta').$$

The posterior variance at any $\theta \in \Theta$ is given by $k'(\theta, \theta)$. Here, we have overloaded notation to denote, $\mu_0(\boldsymbol{\theta}) = (\mu_0(\theta_1), \ldots, \mu_0(\theta_n))^\top \in \mathbb{R}^n$, $k_0(\boldsymbol{\theta}, \theta) = (k_0(\theta_1, \theta), \ldots, k_0(\theta_n, \theta))^\top \in \mathbb{R}^n$, and $\mathbf{K}_0 \in \mathbb{R}^{n \times n}$ whose (i, j)th element is $k_0(\theta_i, \theta_j)$. A popular choice for the covariance kernel is the squared exponential kernel $k_{\sigma_f,h}(\theta, \theta') = \sigma_f^2 \exp(-\|\theta - \theta'\|^2/(2h^2))$. When the observations of f are noisy, it is common to use a noise term in the prior kernel $k_0 = k_{\sigma,h} + \sigma_n^2 \delta(\theta - \theta')$ where δ is the Dirac-delta function. This corresponds to assuming white Gaussian noise with variance σ_n^2. For a more detailed review of GPs we recommend Rasmussen and Williams [23]. Following other GP based active learning methods [12,28], we use GPs primarily as an uncertainty model for an unknown function.

2.2. Active posterior estimation

Problem setting: We formally define our posterior distribution estimation problem in a Bayesian framework. We have a bounded continuous parameter space Θ for the unknown parameters (e.g. cosmological constants). Let $\mathbf{X_{obs}}$ denote our observations (e.g. signals from telescopes). For each $\theta \in \Theta$ we have the ability to query an oracle for the value of the likelihood $\mathcal{L}(\theta) = P(\mathbf{X_{obs}}|\theta)$. These queries are expensive and possibly noisy. Assuming a prior $P_\theta(\theta)$ on Θ, we have the posterior $P_{\theta|\mathbf{X_{obs}}}$.

$$P_{\theta|\mathbf{X_{obs}}}(\theta|\mathbf{X_{obs}}) = \frac{\mathcal{L}(\theta)P_\theta(\theta)}{\int_\Theta \mathcal{L}(\theta)P_\theta(\theta)} = \frac{\mathcal{L}(\theta)P_\theta(\theta)}{P(\mathbf{X_{obs}})} \tag{1}$$

Our goal is to obtain an estimate $\widehat{P}_{\theta|\mathbf{X_{obs}}}$ of $P_{\theta|\mathbf{X_{obs}}}$ using as few queries to the oracle.

Some smoothness assumptions on the problem are warranted to make the problem tractable. A standard in the Bayesian nonparametrics literature is to assume that the function of interest is a sample from a Gaussian Process. In what follows we shall model the log joint probability of the cosmological parameters and the observations via a GP.[1] This is keeping in line with Adams et al. [1] who use a similar prior for GP density sampling.

Uncertainty for the posterior via uncertainty for the log joint: Assume that we have queried the likelihood oracle at n points, and for each query point θ_i the oracle provided us with $\mathcal{L}_i \approx P(\mathbf{X_{obs}}|\theta_i)$ answers. Let $A_n = \{\theta_i, \mathcal{L}_i\}_{i=1}^n$ denote the set of these query value pairs. We build our GP using $B_n = \{\theta_i, \log(\mathcal{L}_i P_\theta(\theta_i))\}_{i=1}^n$ as the input output pairs. The GP automatically induces uncertainty on the log joint probability; let us denote the distribution of $\log P(\mathbf{X_{obs}}, \theta)$ values at any $\theta \in \Theta$ by $\tilde{\mathcal{L}}(\theta)$. Moreover, if g is a sample from this GP, then $f = \exp g / \int \exp g$ denotes a sample from the induced uncertainty model $F_{\theta|\mathbf{X_{obs}}}$ for the posterior $P_{\theta|\mathbf{X_{obs}}}$. A sample from $F_{\theta|\mathbf{X_{obs}}}$ is a distribution over Θ. This is illustrated in Fig. 2.

Finally, given any estimate $\widehat{\mathcal{P}}^{A_n}(\mathbf{X_{obs}}, \theta)$ of the log joint probability constructed using a set A_n of n parameter-likelihood pairs, the estimate of the posterior distribution is

$$\widehat{P}^{A_n}(\theta|\mathbf{X_{obs}}) = \frac{\exp \widehat{\mathcal{P}}^{A_n}(\mathbf{X_{obs}}, \theta)}{\int_\Theta \exp \widehat{\mathcal{P}}^{A_n}(\mathbf{X_{obs}}, \theta)}. \tag{2}$$

[1] We work in the log joint probability space since log smoothes out a function and is more conducive to be modelled as a GP. We also avoid issues such as non-negativity of $\widehat{P}_m^A(\theta|\mathbf{X_{obs}})$. Osborne et al. [20] also use a similar log-transform before applying a GP.

```
Algorithm 1 Bayesian active posterior estimation.
Given: Input space $\Theta$, GP prior $\mu_0$, $k_0$.
For $t = 1, 2, \ldots$ do
    1. $\theta_t = \mathrm{argmax}_{\theta_t \in \Theta}\, u_t(\theta)$
    2. $\mathcal{L}_t \leftarrow$ Query oracle at $\theta_t$.
    3. Obtain posterior conditioned on $(\theta_i, \mathcal{L}_i P_\theta(\theta_i))_{i=1}^t$
```

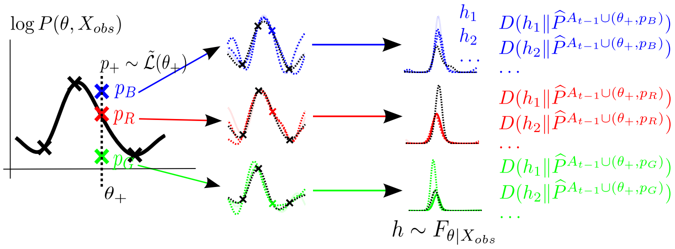

$\log P(\theta, X_{obs})$
$p_+ \sim \tilde{\mathcal{L}}(\theta_+)$
p_B
p_R
p_G
θ_+
h_1
h_2
$\ldots$
$D(h_1\|\widehat{P}^{A_{t-1}\cup(\theta_+,p_B)})$
$D(h_2\|\widehat{P}^{A_{t-1}\cup(\theta_+,p_B)})$
$\ldots$
$D(h_1\|\widehat{P}^{A_{t-1}\cup(\theta_+,p_R)})$
$D(h_2\|\widehat{P}^{A_{t-1}\cup(\theta_+,p_R)})$
$\ldots$
$D(h_1\|\widehat{P}^{A_{t-1}\cup(\theta_+,p_G)})$
$D(h_2\|\widehat{P}^{A_{t-1}\cup(\theta_+,p_G)})$
$\ldots$
$h \sim F_{\theta|X_{obs}}$

Fig. 3. An illustration of the NED utility. θ_+ is a candidate for the next evaluation. p_B, p_R, p_G denote values for $p_+ = \log P(\theta_+, \mathbf{X_{obs}})$ sampled from the GP. We add them as hallucinated points and rebuild our GP and generate samples (second step). These samples are exponentiated and normalised (third step) and then its KL divergence with the estimate is computed. (For interpretation of the colours in this figure, the reader is referred to the web version of this article.)

Bayesian active posterior estimation: We now describe the procedure to determine the point at which we should query the likelihood. At time step t, we have already queried at $t-1$ points and have the set A_{t-1} of query value pairs. Our goal is to select the point θ_t for the next experiment to evaluate the likelihood. We adopt a myopic strategy which picks the point that maximises a utility function u_t. u_t needs to capture a measure of divergence $D(\cdot\|\cdot)$ between the true and estimated distributions. A reasonable strategy would be to select θ_t to satisfy

$$\theta_t = \operatorname*{argmin}_{\theta_+ \in \Theta} D(\, P_{\theta|\mathbf{X_{obs}}} \,\|\, \widehat{P}^{A_{t-1}\cup\{(\theta_+, \mathcal{L}(\theta_+))\}}\,) \tag{3}$$

where $\widehat{P}^{A_{t-1}\cup\{(\theta_+, \mathcal{L}(\theta_+))\}}$ is our estimate of the posterior using $A_{t-1} \cup \{(\theta_+, \mathcal{L}(\theta_+))\}$. Obviously, this objective is not accessible in practice, since we know neither $P_{\theta|\mathbf{X_{obs}}}$ nor $\mathcal{L}(\theta_+)$. As surrogates to this ideal objective in Equation (3), in the following subsections we propose two utility functions for determining the next point: Negative Expected Divergence (NED) and Exponentiated Variance (EV). The first, NED adopts a Bayesian decision theoretic approach akin to the expected error reduction criterion used in active learning [26]. Here, we choose the point in Θ that yields the minimum expected divergence for the next estimate over the uncertainty model. Unfortunately, as we will see in Section 2.3, the NED utility is computationally expensive. Therefore, we propose a cheaper alternative, EV. In our experiments we found that both strategies performed equally well – so EV is computationally attractive. That said, some cosmological simulations are very expensive (taking several hours to a day) so NED is justified in such situations. We present our framework for BAPE using an appropriate utility function u_t in Algorithm 1.

2.3. Negative Expected Divergence (NED)

Equation (3) says that we should choose the point that results in the highest reduction in divergence *if we knew the likelihood and the true posterior at that point*. In NED, we choose the point with the highest expected reduction in divergence. For the next evaluation we choose the point that minimises the expected divergence between these models and the next estimate. Precisely,

$$u_t^{\mathrm{NED}}(\theta_+) = -\mathbb{E}_{p_+ \sim \tilde{\mathcal{L}}(\theta_+)}\, \mathbb{E}_{h \sim F_{\theta|\mathbf{X_{obs}}}}\, D(\, h \,\|\, \widehat{P}_{m+1}^{A\cup\{(\theta_+, p_+)\}}\,). \tag{4}$$

Here $p_+ \in \mathbb{R}$ is sampled from $\tilde{\mathcal{L}}(\theta_+)$, the uncertainty of the log joint probability at θ_+. The *density* h is sampled from $F_{\theta|\mathbf{X_{obs}}}$, the uncertainty model of the posterior obtained by adding (θ_+, p_+) to the set of already available points A_{t-1}. Both $\tilde{\mathcal{L}}(\theta_+)$ and $F_{\theta|\mathbf{X_{obs}}}$ are induced from the log joint GP as explained before. $\widehat{P}_{m+1}^{A\cup\{(\theta_+, p_+)\}}$ denotes the estimate of the posterior obtained by re-training the GP with (θ_+, p_+) as the tth point along with the t points already available. The first expectation above captures our uncertainty over $\log P(\theta_+, \mathbf{X_{obs}})$ while the second captures our remaining uncertainty over $P_{\theta|\mathbf{X_{obs}}}$ after observing $\mathcal{L}(\theta_+)$. Equation (4) says that you should minimise the expected divergence by looking one step ahead.

We have illustrated NED in Fig. 3. Assume we are considering the point θ_+ for the next evaluation. Our GP over the log joint probability gives us uncertainty for $\log P(\theta_+, \mathbf{X_{obs}})$ – depicted by p_B, p_R, p_G in blue, green and red respectively. For p_B, we add (θ_+, p_B) as a hallucinated point to the $t-1$ points we already have and obtain an estimate of the posterior $\widehat{P}^{A_{t-1}\cup(\theta_+, p_B)}$. Next, we rebuild our GP using these t points. We draw samples from the new GP and exponentiate and normalise them to obtain samples h_i from the uncertainty model for the posterior $F_{\theta|\mathbf{X_{obs}}}$. Then we compute the divergence between h_i and $\widehat{P}^{A_{t-1}\cup(\theta_+, p_B)}$. We repeat this for the blue and green points and average all the divergences. The next evaluation point will be that with the lowest expected one step ahead divergence.

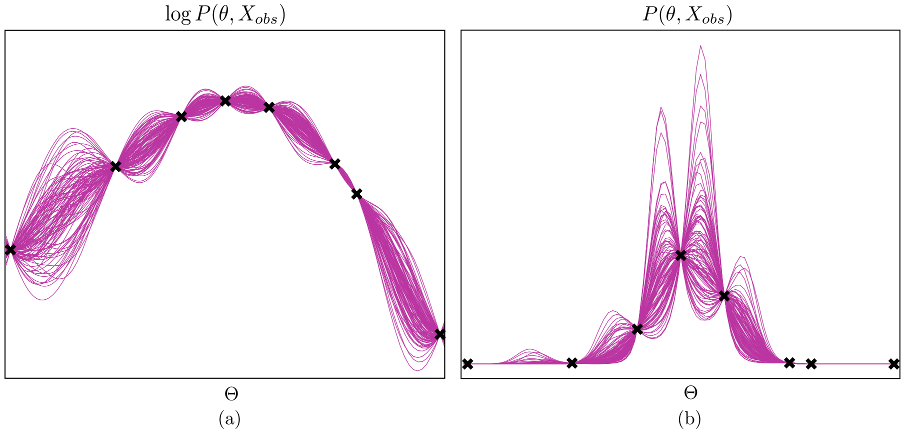


Fig. 4. (a): Samples drawn from the GP in the log joint probability space. (b): The same samples after exponentiation. High variance in the low likelihood regions are squashed and low variances in the high likelihood regions are blown up. This is the key insight that inspires our methods and the EV utility in particular.

The expectations in the NED utility above are computationally intractable. They can only be approximated empirically by drawing samples and require numerical integration (as Fig. 3 suggests). For these reasons we propose an alternate utility function below. In our experiments we found that both EV and NED performed equally well.

2.4. Exponentiated Variance (EV)

A common active learning heuristic is to choose the point that you are most uncertain about for the next experiment. As before we use a GP on the log joint probability. At any given point in this GP we have an associated posterior variance of the GP. However, this variance corresponds to the uncertainty of the *log* joint probability whereas our objective is in learning the joint probability – which is a multiplicative factor away from the posterior. See Fig. 4. Therefore, unlike in usual GP active learning methods [25], the variance of interest here is in the exponentiated GP. By observing that an exponentiated Gaussian is a log Normal distribution, the EV utility function is given by

$$u_t^{\mathrm{EV}}(\theta_+) = \exp(2\mu_t(\theta_+) + \sigma_t^2(\theta_+))(\exp(\sigma_t^2(\theta_+)) - 1) \tag{5}$$

Here μ_t, σ_t^2 are the posterior mean and variances of the GP at time t. The $\exp(2\mu(\theta_+))$ will squash high variances in the low likelihood regions and amplify low variances in the high likelihood regions. The expression for $u_t^{\mathrm{EV}}(\theta_+)$ corresponds precisely to the variance of $P(\theta, \mathbf{X_{obs}})$ according to the uncertainty model induced by the GP. We choose the point *maximising* the above variance to determine the next query location.

2.5. Discussion

We first argue that an active, i.e. an adaptive sequential, strategy will be useful for posterior estimation. In particular, the work of Castro et al. [4] demonstrates that active learning does not perform significantly better than passive strategies to estimate a uniformly smooth function uniformly well. However, in our case we wish to learn the function well at high log probability regions as they predominantly determine the shape of the posterior. To illustrate this we have shown a synthetically created log joint probability and the corresponding posterior in the first column of Fig. 5. The high likelihood regions largely affect the shape of the posterior since the variations in the low likelihood region are squashed after exponentiation. In the second column we queried the likelihood at uniformly spaced points and obtained estimates of the log joint probability and the posterior in green. In the third we have the same estimates in magenta, except that we used more queries at high likelihood regions. While the green estimate for the *log joint* may be uniformly better than the magenta estimate, it is the opposite for *posterior*. This is because, after exponentiation, the small errors in the high log probability regions have been inflated after exponentiation for the green estimate, whereas for the magenta estimate the large errors in the low probability regions have been diminished. Hence an active strategy, which uses more queries at the high log probability regions is likely to do better than a passive strategy. NED and EV do precisely this by attaching more emphasis to the uncertainties in the high log probability regions.

Next, it is important to distinguish our objective in this work from similar active learning literature in the GP framework. In BO, the objective is to find the maximum of a function. This means that once the active learner realises that it has found the mode of a function it has less incentive to explore around as it would not improve the current maximum values. For instance, consider the log joint probability in Fig. 6(a) and the joint probability in Fig. 6(b). We have shown the points where we have already queried at as brown crosses and the red circles (x) and (y) show possible candidates for the next query. The shaded regions represent the uncertainty due to three standard deviations in the GP. In BO, the active learner would

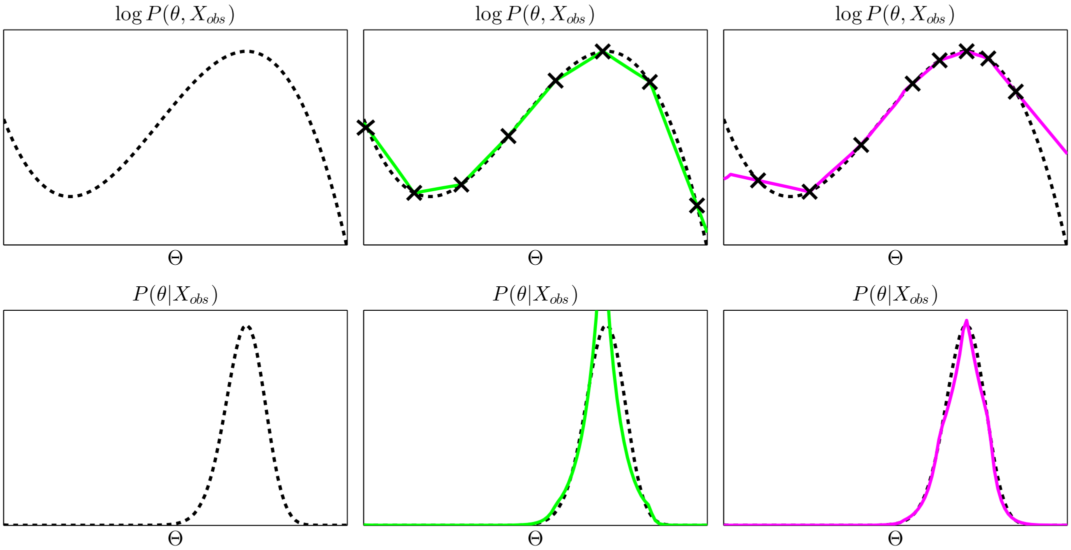


Fig. 5. The first column shows the log joint probability and the corresponding posterior. In the second column we have estimates of the log joint and the posterior for uniformly spaced points. In the third column we have the same except that more points were chosen in high likelihood regions. (For interpretation of the colours in this figure, the reader is referred to the web version of this article.)

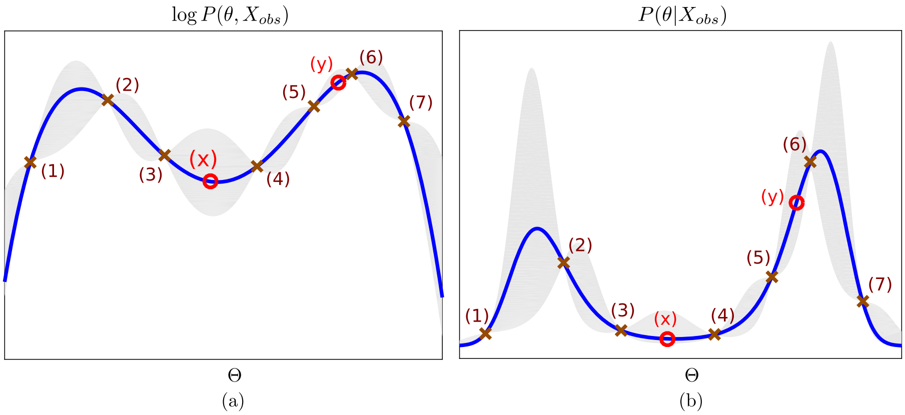


Fig. 6. (a) and (b) are the true log joint probability and joint probability in blue. Assume that we have already queried at the brown crosses and let the red circles (x) and (y) be candidates for the next query. In BAPE we would be interested in querying (y) but not (x). In AGPR we would be interested in both (x) and (y) whereas in BO we would be keen in neither. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)

not be interested in (y) as, by virtue of points (5), (6) and (7) it knows that (y) is not likely to be higher than (6). On the other hand, in BAPE we are keen on (y) as knowing it with precision will significantly affect our estimate of the posterior (Fig. 6(b)). In particular to know the posterior well we will need to query at the neighbourhood of modes and the heavy tails of a distribution. A BO utility is not interested in such queries. On the other extreme, in AGPR the objective is to learn the function uniformly well. This means in the same figures, AGPR will query point (x). However, given sufficient smoothness, we know that the joint probability will be very low there after exponentiation due to points (3) and (4). Therefore, the BAPE active learner will not be as interested in (x) as AGPR. Observe that the uncertainty at (x) is large in the log joint probability space in comparison to the uncertainty elsewhere; however, in the probability space this is smaller than the uncertainty at the high probability regions. As Fig. 5 indicates, while we model the log joint probability as a GP we are more interested in the uncertainty model of the posterior/joint probability. Finally, as a special case for BQ, Osborne et al. [20] consider evaluating the model evidence – i.e. the integral under the conditional. Their utility function uses approximations tailored to estimating the integral well. Note that our goal of estimating the posterior well is more difficult than estimating an integral under the conditional as the former implies the latter but not vice versa.

3. Other algorithms for comparison

We list and describe some potential alternatives for posterior estimation which we use in our empirical evaluation.

1. MCMC – density estimation (MCMC-DE): We implement MCMC with a Metropolis Hastings (MH) chain and use kernel density estimation (KDE) on the accepted points to estimate the posterior. When comparing MCMC against NED/EV we consider *the total number of queries* and not just those accepted. There are several variants of the MH proposal scheme and several tuning parameters. Comparing to all of them is nontrivial. We use MH in its basic form using a fixed Gaussian proposal distribution. Practitioners usually adjust the proposal based on the acceptance rate. Here, we chose the proposal manually by trying different values and picking the one that performed best within the queries used. Note that this comparison is advantageous to MCMC. In one experiment we test with Emcee [6], a popular package for Affine Invariant MCMC which automatically fine tunes the proposal bandwidth based on acceptance rate [6].

2. MCMC – regression (MCMC-R): Here, as in MCMC-DE we use a MH Chain to generate the samples. However, this time we regress on the queries (not samples) to estimate the posterior. We include this procedure since MCMC can be viewed as a heuristic to explore the parameter space in high likelihood regions. We show that a principled query strategy outperforms this heuristic.

3. Approximate Bayesian computing (ABC): There are several variants of ABC [18,22]. We compare with a basic form given in [17]. At each iteration, we randomly sample θ from the prior and then sample an observation $\mathbf{X_{sim}}$ from the likelihood. If $d(\mathbf{X_{sim}}, \mathbf{X_{obs}}) < \epsilon$ we add θ to our collection. Here $d(\cdot, \cdot)$ is some metric on a sufficient statistic of the observation and $\epsilon > 0$ is a prespecified threshold. We perform a KDE on the collected samples to estimate the posterior. The performance of ABC depends on ϵ: As for MCMC-DE we choose ϵ by experimenting with different values and choosing the value which gives the best performance within the queries used. We compare with total number of parameter values proposed and not just those retained. We compare with ABC only in experiments where it is possible to sample from the likelihood (in addition to evaluating the likelihood).

4. Uniform random samples (RAND): Here, we evaluate the likelihood at points chosen uniformly on Θ and then regress on these points.

5. Active Gaussian process regression (AGPR): & 6. Bayesian optimisation (BO-EI): On our synthetic problems we also compare with the GP based active learning methods discussed in Section 2.5. For BO, we use the Expected Improvement [2]. We choose points using the above criterion and then regress on these points.

4. Experiments

We perform experiments on a series of low and high dimensional synthetic and real astrophysical experiments.

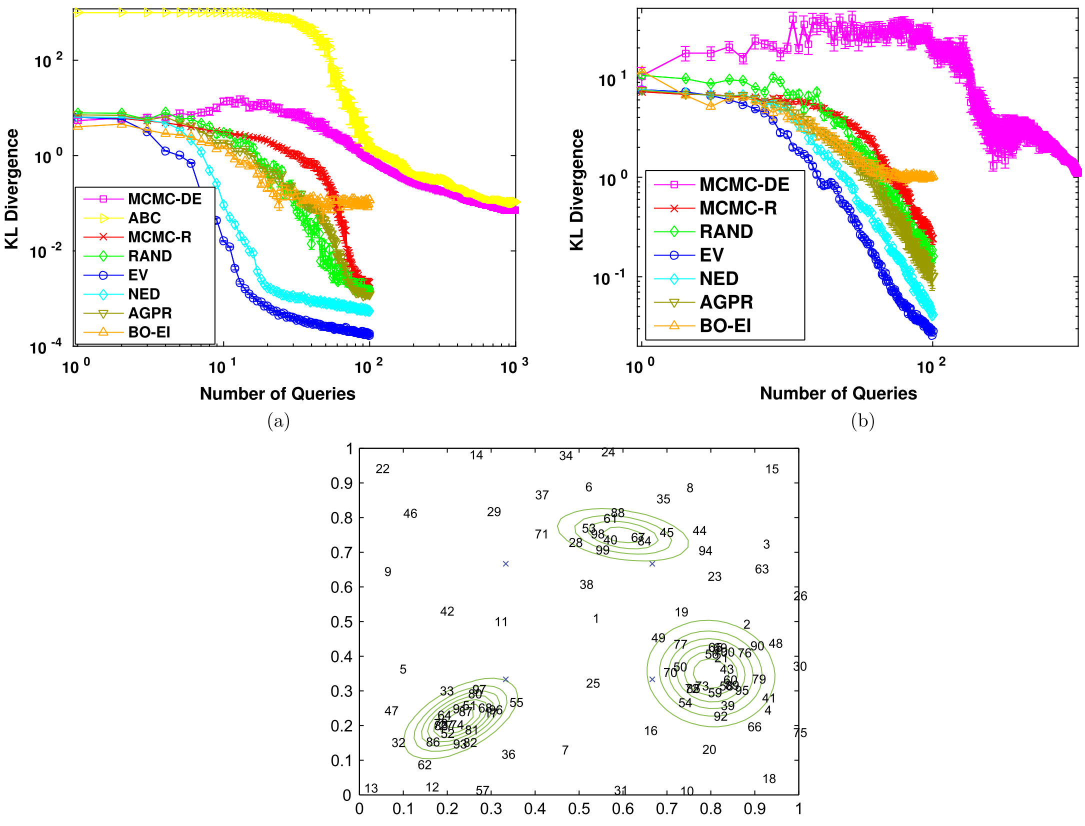
In our experiments the NED, EV utilities were maximised by evaluating them on a grid of size 10^3–10^8 depending on the dimensionality and then choosing the point with the maximum value. For numerical integration in NED, we use the trapezoidal rule. Further, since the inner expectation in Equation (4) is expensive we approximate it using a one sample estimate. NED is only tested on low dimensional problems since empirical approximation and numerical integration is computationally expensive in high dimensions. In our experiments, EV slightly outperforms NED probably since the EV utility can be evaluated exactly while NED can only approximated.

We use a squared exponential kernel in all our experiments. The bandwidth for the kernel was set to be $5n^{-1/d}$ where n is the total number of queries and d is the dimension. This was following several kernel methods (such as kernel regression) which use a bandwidth on the order $O(n^{\frac{-c_1}{c_2+d}})$ [9]. The constant 5 was hand tuned by experimenting with a series of independent synthetic experiments. The other GP hyper-parameters, σ_f^2 and σ_n^2 were set via cross validation every 20 iterations. When we tried setting the bandwidth via cross validation too we found that it had a tendency too choose a larger than required bandwidth in the early iterations and then get stuck without decreasing. The consequence of this behaviour is that our method might not sufficiently explore the space and hence miss out on certain regions of the likelihood. Such a phenomenon has also been observed in Bayesian Optimisation and hence the bandwidth is decreased artificially as a precautionary measure against insufficient exploration [30].

4.1. Low dimensional synthetic experiments

To illustrate our methods we have two simple yet instructive experiments. In the first, the parameters space is $\Theta = (0, 1)$ equipped with a Beta$(1.2, 1)$ prior. We draw θ from the prior, and then draw 500 samples from a Bernoulli $(\theta^2 + (1-\theta)^2)$ distribution: i.e. $\mathbf{X_{obs}} \in \{0, 1\}^{500}$. The ambiguity on the true value of θ creates a bimodal posterior. Fig. 7(a) compares NED/EV against the other methods as a function of the number of queries. For ABC, we rejected if $\left(\sum_i \mathbf{X}^{(i)}_{\mathbf{obs}} - \sum_i \mathbf{X}^{(i)}_{\mathbf{sim}}\right)/\sum_i \mathbf{X}^{(i)}_{\mathbf{obs}} > 0.02$.

The second experiment is a 2D problem with $\Theta = (0, 1)^2$. We artificially created a 3-modal log-joint posterior shown by green contours in Fig. 7(c). Fig. 7(b) compares all methods. Fig. 7(c) shows the points chosen by the NED query strategy in order. We have learned the high log joint probability regions well at the expense of being uncertain at low log joint probability areas. However, this does not affect the posterior significantly as they are very small after exponentiation. ABC does not apply here since we artificially constructed the log posterior and cannot sample from the likelihood.



(c)

Fig. 7. (a), (b): Comparison of NED/EV against MCMC-DE, ABC, MCMC-R and RAND for the 1D and 2D synthetic experiments respectively. The x-axis is the number of queries and the y-axis is the KL divergence between the truth and the estimate. All figures were obtained by averaging over 60 trials. (c): The 100 points chosen in order by NED for the 2D experiment. The green contours are the true posterior. Initially the algorithm explores the space before focusing on high probability regions. (For interpretation of the references to colour in this figure, the reader is referred to the web version of this article.)

Our methods outperform existing methods and other heuristics by orders of magnitude on these simple experiments. Both MCMC-DE and ABC require a large number of samples before being competitive with the methods using regression. This corroborates an earlier remark that using the evaluated likelihood values in the estimate can be useful when the queries are expensive. Note that the KL divergence for BO gets stuck without decreasing further. This is because after a certain stage, most evaluations are centred near the maximum. As a consequence, the heavy tails and other modes are not explored properly.

4.2. Higher dimensional synthetic experiments

We test in $d = 5$ and 15 dimensions. We construct an artificial log likelihood so that the resulting posterior is mixture of 2 Gaussians centred at $\mathbf{0}$ and $\mathbf{1}$. Both Gaussians had covariance $\sigma^2 I_d$ where $\sigma = 0.5\sqrt{d}$. We evaluate performance by the ability to estimate certain linear functionals. The exact value of these functionals can be evaluated analytically since we know the true posterior. We use a uniform prior. Our log-likelihood, functionals and their true values are

$$\ell(\theta) = \log\left(0.5\mathcal{N}\left(\theta;\ \mathbf{0},\ \frac{d}{4}I_d\right) + 0.5\mathcal{N}\left(\theta;\ \mathbf{1},\ \frac{d}{4}I_d\right)\right)$$

$$T_1 = \mathbb{E}\left[\sum_{i=1}^{d} X_i\right] = \frac{d}{2}, \qquad T_3 = \mathbb{E}\left[\sum_{i=1}^{d-2} X_i^2 X_{i+1}\right] = \frac{d-1}{2}(1+\sigma^2),$$

$$T_2 = \mathbb{E}\left[\sum_{i=1}^{d} X_i^2\right] = \frac{d}{2}(1+2\sigma^2), \quad T_4 = \mathbb{E}\left[\sum_{i=1}^{d-2} X_i X_{i+1} X_{i+2}\right] = \frac{d-2}{2}$$

For MCMC-DE, we draw samples $Z_1, Z_2, \ldots$ from the true likelihood. To estimate $T_i = \mathbb{E}[\phi_i(X)]$ we use the empirical estimator $\widehat{T}_i = 1/N \sum_k \phi_i(Z_k)$. Here $\phi_1 = \sum_{i=1}^{d} X_i$ for T_1 etc. For MCMC-DE we experimented with Gaussian proposal distributions. For EV, MCMC-R and RAND we first use the queried points to obtain an estimate of the log-likelihood by regressing on the

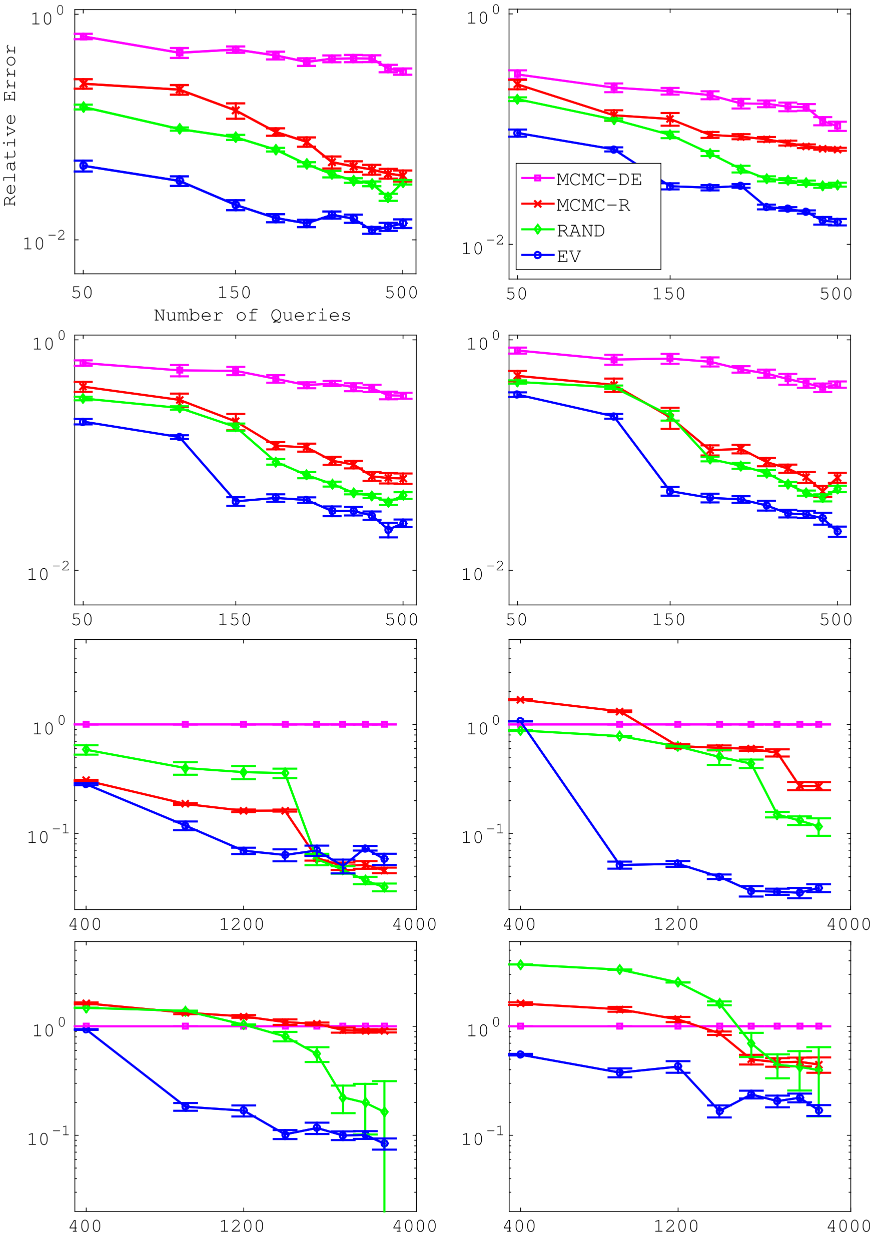


Fig. 8. The first row is for the functionals T_1, T_2 in $d = 5$ dimensions and the second for is for the functionals T_3, T_4. The last two rows are the same four functionals for $d = 15$. The x-axis is the number of queries and the y-axis is $|\widehat{T}_i - T_i|/|T_i|$. We use 500 queries for $d = 5$ and 3200 queries for $d = 15$. All figures were obtained by averaging over 30 trials.

likelihood values as explained before. Then we run an MCMC chain on this *estimate* to collect samples and use the empirical estimator for the functionals. Note that evaluating the estimate, unlike the likelihood, is cheap. We did not try NED since numerical integration is intractable in high dimensions. ABC does not apply in this experiment as we cannot sample from the log likelihood. For the proposal distributions for MCMC-DE and MCMC-R methods we used a Gaussian with standard deviations $\{0.25\sigma, 0.5\sigma, \sigma, 2\sigma, 4\sigma\}$ and report the one that performed best within the allotted queries. When applying MCMC on the regression estimates in EV, MCMC-R and RAND we used a Gaussian proposal with standard deviation σ. The results are shown in Fig. 8. They demonstrate the superiority of our query strategy over the alternatives.

4.3. Type Ia supernovae

We use supernovae data for inference on 3 cosmological parameters: Hubble Constant ($H_0 \in (60, 80)$, Dark Matter Fraction $\Omega_M \in (0, 1)$ and Dark Energy Fraction $\Omega_\Lambda \in (0, 1)$. The likelihood for the experiment is given by the Robertson–Walker

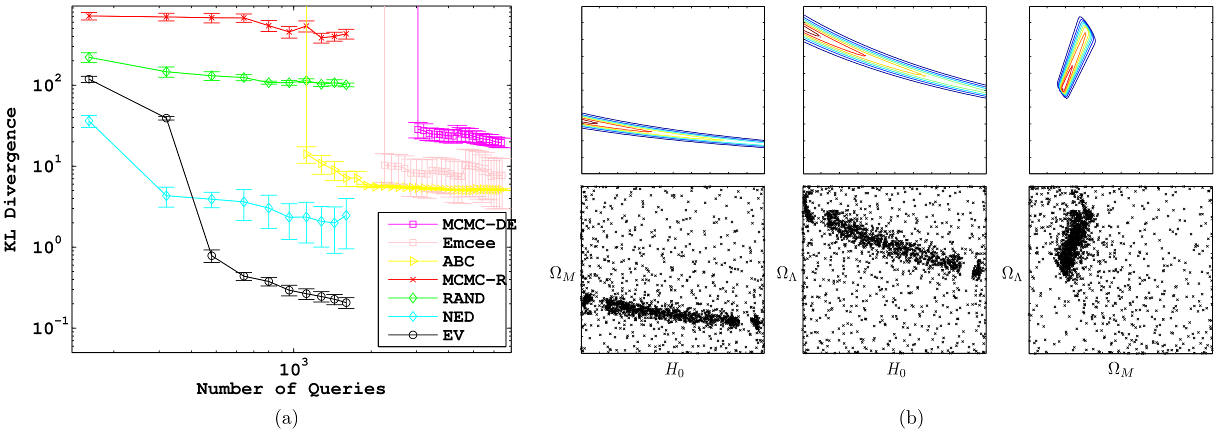


Fig. 9. (a): Comparison of NED/EV against MCMC-DE, ABC, Emcee, MCMC-R and RAND on the Type Ia Supernovae dataset. For all regression methods we show results for up to 1600 queries and up to 4 times as many for MCMC and ABC. For evaluation, KL was approximated via numeric integration on a $(100)^3$ grid. Note that MCMC and ABC require several queries before a nontrivial KL with the truth is obtained. All curves were obtained by averaging over 30 runs. (b): Projections of the points selected by EV (bottom row) and the marginal distributions (top row).

metric which models the distance to a supernova given the parameters and the observed red-shift. The dataset is taken from [5]. The parameter space is taken to be $\Theta = (0,1)^3$ (For H_0 we map it to $(60, 80)$ using an appropriate linear transform). We test NED/EV against MCMC-DE, ABC, MCMC-R, RAND and Emcee, a popular python package for affine invariant MCMC. For ABC, sampling from the likelihood is as expensive as computing the likelihood. Fig. 9(a) compares all methods. Fig. 9(b) shows the points queried by EV and the marginals of the true posterior. As expected, most of EV's queries are concentrated around the modes and heavy tails of the posterior. The KL for RAND decreases slowly since it accumulates points at the high likelihood region very slowly. MCMC-R performs poorly since it has only explored part of the high likelihood region. For NED/EV after an initial exploration phase, the error shoots down.

The likelihood evaluations in this experiment are quite cheap, taking only a fraction of a second for each query. Determining the next point is more expensive in EV than methods such as MCMC, ABC and RAND due to matrix inversion in GPs. However, we found that the latter methods required up to 20–30 times the number of likelihood evaluations to be competitive with EV in this experiment. Therefore, despite the fact that the EV query strategy is expensive it performs better than other methods on wall clock time. This illustrates that principled adaptive query strategies can reap great dividends in posterior estimation.

4.4. *Luminous red galaxies*

Here we used data on Luminous Red Galaxies (LRGs) for inference on 8 cosmological parameters: spatial curvature $\Omega_k \in (-1, 0.9)$, dark energy fraction $\Omega_\Lambda \in (0, 1)$, cold dark matter density $\omega_c \in (0, 1.2)$, baryonic density $\omega_B \in (0.001, 0.25)$, scalar spectral index $n_s \in (0.5, 1.7)$, scalar fluctuation amplitude $A_s \in (0.65, 0.75)$, running of spectral index $\alpha \in (-0.1, 0.1)$ and galaxy bias $b \in (0, 3)$. The likelihood is obtained via the Galaxy Power spectrum which measures the distribution of temperature fluctuations as a function of scale. We use software and data from [29]. Our parameter space is taken to be $(0,1)^8$ by appropriately linear transforming the range of the variables. Each query takes about 4–5 seconds. In EV, determining the next point takes about 0.5–1 seconds with ≈ 2000 points and about 10–15 seconds with ≈ 10000 points. In this regime, where the cost of the likelihood evaluation is more expensive or comparable to the cost of determining the next point in EV, we significantly outperforms other methods on wall clock time. We do not compare with NED due to the difficulty of high dimensional numerical integration. We do not compare with ABC since the software only permits evaluation of the likelihood but not sampling.

Fig. 10 shows points queried by MCMC, RAND and EV projected on the first 2 dimensions. MCMC has several high likelihood points but its queries are focused on a small region of the space. RAND does not have many points at high likelihood regions. EV has explored the space fairly well and at the same time has several queries at high likelihood regions. As numerical integration in 8 dimensions is difficult, we cannot obtain ground truth for this experiment. Therefore, we perform the following simple test. We queried 250,000 points uniformly at random from the parameter space to form a test set. We then run EV, MCMC-R and RAND for up to 12,000 queries to collect points and estimate the posterior. Performance is evaluated by the mean squared reconstruction error of the *exponentiated* log joint probabilities (joint probabilities). Fig. 11 shows the results. The error for RAND and MCMC-R stay the same throughout since the problem is difficult and they did not have sufficient number of high likelihood points throughout the space.

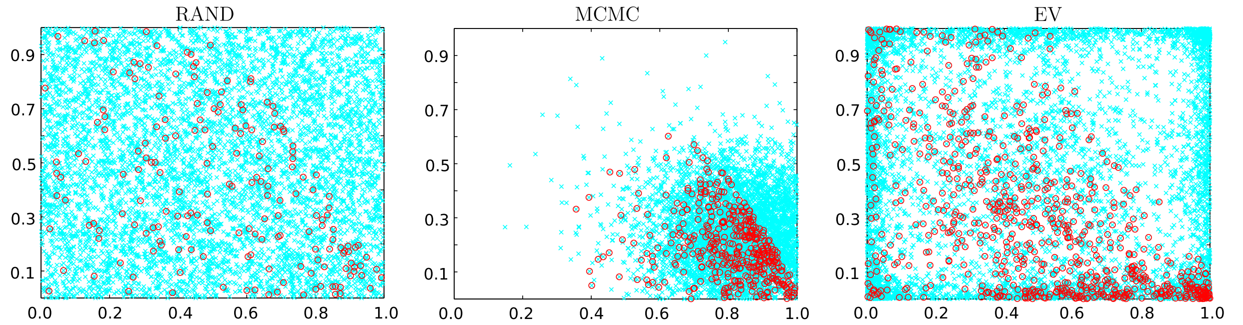


Fig. 10. The projections of the first 6000 points queried by RAND MCMC, and EV respectively on to the first 2 dimensions in cyan. The points shown in red are queries at high likelihood ($\log P > -50$) points. (For interpretation of the colours in this figure, the reader is referred to the web version of this article.)

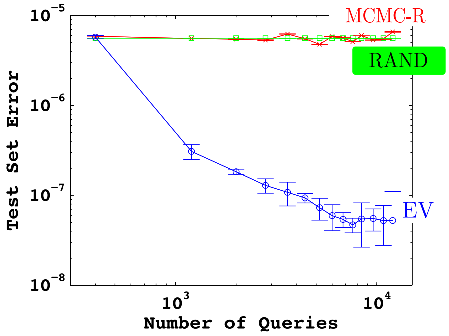


Fig. 11. Comparison of EV against MCMC-R and RAND. We use up to 12000 queries for all methods. The y-axis is the mean squared reconstruction error. The curves were obtained by averaging over 16 runs.

5. Conclusions

We proposed a framework for query efficient posterior estimation for expensive blackbox likelihood evaluations. Our methods use GPs and are based on popular ideas in Bayesian active learning. We demonstrate that our methods outperform natural alternatives in practice.

Note that in Machine Learning it is uncommon to treat posterior estimation in a regression setting. This is probably since the estimate will depend on the intricacies of the regression algorithm. Thus if likelihood evaluations are inexpensive, MCMC seems like a natural choice due to its theoretical guarantees in the large sample regime. However, our work demonstrates that when likelihood evaluations are expensive, such as in scientific simulations, treating posterior estimation in an active regression framework enables us to be significantly query efficient.

The proposed methods do not scale very well with dimension, which is a common problem with nonparametric methods. Going forward we wish to tackle active posterior estimation in several dozens of dimensions. A possible avenue would be to use ideas from some recent progress on high dimensional Bayesian Optimisation using additive models [11].

Acknowledgement

This research was partly funded by DOE grant DESC0011114. An abridged version of this work appeared in Kandasamy et al. [10].

References

[1] Ryan Prescott Adams, Iain Murray, David J.C. MacKay, The Gaussian process density sampler, in: NIPS, 2008.
[2] Eric Brochu, Vlad M. Cora, Nando de Freitas, A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, CoRR (2010).
[3] Brent Bryan, Jeff Schneider, Robert Nichol, Christopher Miller, Christopher Genovese, Larry Wasserman, Active learning for identifying function threshold boundaries, in: Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2006.
[4] Rui Castro, Rebecca Willett, Robert Nowak, Faster rates in regression via active learning, in: NIPS, 2005.
[5] T.M. Davis, et al., Scrutinizing exotic cosmological models using ESSENCE supernova data combined with other cosmological probes, Astrophys. J. (2007) 716–725.
[6] Daniel Foreman-Mackey, David W. Hogg, Dustin Lang, Jonathan Goodman, emcee: The MCMC Hammer, January 2013.
[7] Kenji Fukumizu, Le Song, Arthur Gretton, Kernel Bayes' rule: Bayesian inference with positive definite kernels, J. Mach. Learn. Res. (2014).
[8] Alkis Gotovos, Nathalie Casati, Gregory Hitz, Andreas Krause, Active learning for level set estimation, in: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013, Beijing, China, August 3–9, 2013, 2013.

[9] László Györfi, Micael Kohler, Adam Krzyzak, Harro Walk, A Distribution Free Theory of Nonparametric Regression, Springer Series in Statistics, 2002.

[10] Kirthevasan Kandasamy, Jeff Schneider, Barnabás Póczos, Bayesian active learning for posterior estimation, in: International Joint Conference on Artificial Intelligence, 2015.

[11] Kirthevasan Kandasamy, Jeff Schneider, Barnabás Póczos, High dimensional Bayesian optimisation and bandits via additive models, in: International Conference on Machine Learning, 2015.

[12] Andreas Krause, Ajit Singh, Carlos Guestrin, Near-optimal sensor placements in gaussian processes: theory, efficient algorithms and empirical studies, J. Mach. Learn. Res. (2008).

[13] David Landau, Kurt Binder, A Guide to Monte Carlo Simulations in Statistical Physics, Cambridge University Press, 2005.

[14] Jun S. Liu, Monte Carlo Strategies in Scientific Computing, Springer, ISBN 0387952306, 2001.

[15] Yifei Ma, Roman Garnett, Jeff Schneider, Active area search via Bayesian quadrature, in: International Conference on Artificial Intelligence and Statistics, 2014.

[16] David J.C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.

[17] Jean-Michel Marin, Pierre Pudlo, Christian P. Robert, Robin J. Ryder, Approximate Bayesian computational methods, Stat. Comput. (2012).

[18] Paul Marjoram, John Molitor, Vincent Plagnol, Simon Tavaré, Markov chain Monte Carlo without likelihoods, Proc. Natl. Acad. Sci. USA (2003).

[19] J.B. Mockus, L.J. Mockus, Bayesian approach to global optimization and application to multiobjective and constrained problems, J. Optim. Theory Appl. (1991).

[20] M. Osborne, D. Duvenaud, R. Garnett, C. Rasmussen, S. Roberts, Z. Ghahramani, Active learning of model evidence using Bayesian quadrature, in: Neural Information Processing Systems, 2012.

[21] David Parkinson, Pia Mukherjee, Andrew R. Liddle, A Bayesian model selection analysis of WMAP3, Phys. Rev. D, Part. Fields 73 (2006) 123523.

[22] Gareth W. Peters, Y. Fan, Scott A. Sisson, On sequential Monte Carlo, partial rejection control and approximate Bayesian computation, Stat. Comput. (2012).

[23] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, Adaptative Computation and Machine Learning Series, University Press Group Limited, 2006.

[24] Christian P. Robert, George Casella, Monte Carlo Statistical Methods, Springer Texts in Statistics, Springer-Verlag New York, Inc., 2005.

[25] Sambu Seo, Marko Wallat, Thore Graepel, Klaus Obermayer, Gaussian process regression: active data selection and test point rejection, in: International Joint Conference on Neural Networks, 2000.

[26] Burr Settles, Active Learning Literature Survey, Technical report, University of Wisconsin-Madison, 2010.

[27] John Skilling, Nested sampling for general Bayesian computation, Bayesian Anal. (2006).

[28] Niranjan Srinivas, Andreas Krause, Sham Kakade, Matthias Seeger, Gaussian process optimization in the bandit setting: no regret and experimental design, in: International Conference on Machine Learning, 2010.

[29] M. Tegmark, et al., Cosmological constraints from the SDSS luminous red galaxies, Phys. Rev. (December 2006).

[30] Ziyu Wang, Masrour Zoghi, Frank Hutter, David Matheson, Nando de Freitas, Bayesian optimization in high dimensions via random embeddings, in: International Joint Conference on Artificial Intelligence, 2013.