Quantile Stein Variational Gradient Descent for Batch Bayesian Optimization

Chengyue Gong [1] **Jian Peng** [2] **Qiang Liu** [1]

Abstract

Batch Bayesian optimization has been shown to be an efficient and successful approach for black-box function optimization, especially when the evaluation of cost function is highly expensive but can be efficiently parallelized. In this paper, we introduce a novel variational framework for batch query optimization, based on the argument that the query batch should be selected to have both high diversity and good worst case performance. This motivates us to introduce a variational objective that combines a quantile-based risk measure (for worst case performance) and entropy regularization (for enforcing diversity). We derive a gradient-based particle optimization algorithm for solving our quantile-based variational objective, which generalizes Stein variational gradient descent (SVGD) by Liu & Wang (2016). We evaluate our method on a number of real-world applications, and show that it consistently outperforms other recent state-of-the-art batch Bayesian optimization methods.

1. Introduction

Bayesian optimization (BO) is an efficient method for optimizing non-convex and black-box functions. It has been recently well-studied in the machine learning community and starts to be applied to many applications in science and engineering (Fernandez et al., 2011; Turgeon et al., 2016). From searching hyper-parameter for machine learning algorithms (Oh et al., 2018), to designing electronic or chemical products automatically (Lyu et al., 2018; Griffiths, 2017; Gómez-Bombarelli et al., 2018), Bayesian optimization has already shown great success and potential for problems where automation is needed to replace tedious human labor.

The idea of Bayesian optimization is to treat the unknown, or expensive objective function as a random variable and leverage Bayesian inference, typically with a Gaussian process prior, to estimate the posterior distribution of the function value on new points in the design space, given the existing function evaluations collected. An acquisition function, such as the expected improvement (Jones et al., 1998) or the upper confidence bound (Srinivas et al., 2009), is then constructed based on the posterior estimation to determine a new point as query to evaluate the objective function. The choice of the new query points should balance the trade-off between exploration and exploitation, so that one can optimize the objective function using as few function evaluations as possible.

In this work, we focus on the setting of batch (or parallel) Bayesian optimization (Frazier et al., 2009; Frazier & Clark, 2012; Kathuria et al., 2016). In this setting, multiple query points are obtained simultaneously at each iteration, so that they can be evaluated in parallel. Because it is often easy to run multiple experiments or evaluation programs in parallel in practice, batch Bayesian optimization is an essential and practical way to accelerate Bayesian optimization for highly expensive experiments.

Unfortunately, searching multiple points jointly is much more challenging than searching for single query point, given that we search in a higher dimensional space with the same amount of information, and may yield a significant waste of computational resource if the joint selection is not performed carefully. We argue that the set of new query points should satisfy the following two desiderata, both arise due to the highly expensive evaluation cost.

I) **High Diversity**. The query points in the batch should be sufficiently different from each other, so that the evaluation is not redundant. This corresponds to exploration in optimization.

II) **Risk Aversion**. Every single query point, not just the average or mean of the whole batch, should have a sufficiently high expected function value or value of information, so that each evaluation deserves its cost.

Although there has been a significant number of recent work on batch Bayesian optimization (e.g. González et al., 2016; Desautels et al., 2014; Contal et al., 2013; Kathuria et al., 2016; Lyu et al., 2018), they are not designed to explicitly address these two desiderata *jointly*. The goal

[1]Department of Computer Science, UT Austin [2]University of Illinois at Urbana Champaign, IL. Correspondence to: Chengyue Gong <cygong@cs.utexas.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

of our work is to design a novel batch BO framework, to allow us to explicitly optimize both the diversity and risk aversion, and hence significantly improve the performance over existing approaches.

Our approach is based on a novel variational inference framework for query optimization, in which we transform the optimization of multiple query points to a variational inference of finding an optimal query distribution. We design the variational objective to explicitly account the two desiderata, by using a *quantile-based risk measure* for maximizing the worst acquisition function in the batch, and an *entropy regularization* for encouraging diversity.

Developing practical algorithm for optimizing our new variational objective is highly non-trivial mathematically. We develop a functional gradient descent approach by leveraging the basic idea from Stein variational gradient descent (Liu & Wang, 2016), which allows us to derive a simple yet highly efficient particle-based algorithm for our variational optimization. Our framework results in a substantial generalization of SVGD for quantile-based risk measures, which may find broader applications to other risk-sensitive settings as well. Empirically, we evaluate our new approach on a series of benchmark functions and several real-world applications, including automatic genomic selection and chemical design. The experiments demonstrate that our method achieves better or comparable performances when compared to other recent state-of-the-art batch Bayesian optimization methods.

2. Related Work

Batch Bayesian optimization has been an active research topic during the past years. Many algorithms have been developed to select the query points sequentially in a greedy fashion (e.g., Contal et al., 2013; González et al., 2016; Desautels et al., 2014; Wang et al., 2016; González et al., 2016), which may yield myopic selection due to the greedy selection strategy. Other methods, such as Nyikosa et al. (2015); Wu et al. (2017); Shah & Ghahramani (2015); Contal et al. (2013), are based on simultaneously optimizing the query points using a joint acquisition function. Diversity has also been considered in several very recent works in batch Bayesian optimization (e.g., Kathuria et al., 2016). However, our work is different in that we jointly consider both diversity and risk aversion as our design principle, and propose a novel variational query optimization framework that integrates quantile-based risk measures with SVGD.

Beyond batch query optimization, there has been a significant amount of literature on addressing various other issues in large-scale Bayesian optimization, which we do not have space to enumerate; examples include optimizing high-dimensional black-box functions (e.g., Wang et al., 2017; Oh et al., 2018; Rolland et al., 2018), and speeding up Gaussian process inference in big data settings (e.g., Wang et al., 2018; Klein et al., 2017; Balog et al., 2016).

3. Background

In this section, we introduce the background of batch Bayesian optimization and risk measures.

3.1. Bayesian Optimization (BO)

Our goal is to solve the optimization problem of

$$x^* = \arg\max_x f(x), \tag{1}$$

where f is a black-box function, whose analytic form is unknown and can be only evaluated point-wisely with a high computation or monetary cost.

In Bayesian optimization, the black box function f is treated as a random variable and Bayesian inference is used to infer its posterior distribution given existing evaluation results. Let $p(f)$ be a prior on the space of f, which is typically a Gaussian process (GP) (Rasmussen, 2004). Given a set of existing evaluation points $\mathcal{D} := \{x_i, f(x_i)\}$, we calculate posterior distribution $p(f \mid \mathcal{D}) \propto p(f)p(\mathcal{D} \mid f)$, which allows us to both estimate f and evaluate its uncertainty. In the case of GP, the posterior is also a GP and can be calculated in closed form.

Given the posterior information, an *acquisition function* (or *value of information*) is constructed to determines which point should be selected next:

$$x^{new} \leftarrow \arg\max_x \alpha(x \mid \mathcal{D}).$$

A canonical example of acquisition function is the upper confidence bound (UCB) (Auer, 2002; Srinivas et al., 2009):

$$\alpha(x \mid \mathcal{D}) := \mathbb{E}_f[f(x) \mid \mathcal{D}] \; + \; \eta\sqrt{\mathrm{var}_f[f(x) \mid \mathcal{D}]}, \tag{2}$$

where the expectation and variance are w.r.t. the posterior distribution of f, and η is a positive number that controls the magnitude of uncertainty to take into account for balancing exploitation and exploration. After getting x^{new}, its function value $f(x^{new})$ is evaluated and added to the dataset: $\mathcal{D} \leftarrow \mathcal{D} \cup \{x^{new}, f(x^{new})\}$. This procedure repeats until a stopping criterion is reached.

Besides UCB, many other common acquisition functions exist, including the probability of improvement (PI) (Duval & Silvia, 2002), expected improvement (EI) (Jones et al., 1998), entropy search (ES) (Shah & Ghahramani, 2015), knowledge gradient (KG) (Wu & Frazier, 2016; Wu et al., 2017; Negoescu et al., 2011), and others. See e.g., Shahriari et al. (2016) for an overview.

Batch Bayesian Optimization (BO) The standard BO above adds a single query point at each iteration. In many practical applications, however, it is useful to obtain a *batch* of query points $\{x_1^{new}, \ldots, x_n^{new}\}$ simultaneously, so that they can be evaluated in parallel to speed up the overall process. Unfortunately, searching multiple points jointly is substantially more challenging as we elaborated in the introduction. The goal of this work is to develop a novel batch BO framework that explicitly maximizes and balances the diversity and worst case acquisition value in the batch.

3.2. Quantile and Risk Measure

Quantile has been an important tool for developing risk-sensitive measures (e.g., Gilchrist, 2000; Mitra & Ji, 2010), especially in finance and economics. We leverage it for risk aversion in our batch BO framework. Let Z be a one-dimensional random variable on $\mathbb{R}$, whose cumulative probability function (CDF) is $F(t) = \Pr(Z \leq t)$. The quantile function $Q(\beta)$ of Z is defined to be the inverse function of F:

$$Q(\beta) := \inf\{t \colon F(t) \geq \beta\}.$$

In this work, we will assume F is continuous and invertible, for which we have $Q(\beta) = F^{-1}(\beta)$, or $F(Q(\beta)) = \beta$.

Like density and CDF, the quantile function fully characterizes a distribution. For example, the expectation can be represented as the expected value of the β-quantile when β is drawn from the uniform distribution, that is,

$$\mathbb{E}[Z] = \int_0^1 F^{-1}(\beta) d\beta.$$

Motivated by this, we consider the following *quantile-distorted expectation*, which is also known as spectral, or distortion risk measure (Acerbi, 2002; Balbás et al., 2009; Wirch & Hardy, 2001) in finance,

$$\mathbb{E}^{\omega}[Z] = \int_0^1 F^{-1}(\beta)\omega(\beta) d\beta, \tag{3}$$

in which each β-quantile $F^{-1}(\beta)$ is assigned with an importance weight, or distortion function $\omega(\beta)$, and hence contributes differently to the overall expectation. Clearly, $\mathbb{E}^{\omega}[z]$ reduces to the typical expectation when $\omega(\beta) \equiv 1$.

Different choice of ω can be viewed as representing different preference on risk. Assume Z is a monetary return of an financial investment, then $\mathbb{E}^{\omega}[Z]$ is a *risk-averse* measure when $\omega(\beta)$ is large for small β, which emphasizes the worst case return, and a *risk-seeking* measure when $\omega(\beta)$ is large for large β, emphasizing the best case returns. As an example, assume Z has a finite support $[z_-, z_+]$, then we have $\mathbb{E}^{\omega}[Z] = z_+$ if $\omega(\beta) = \delta[\beta = 1]$and $\mathbb{E}^{\omega}[Z] = z_-$ if $\omega(\beta) = \delta[\beta = 0]$. Here δ denotes the Dirac Delta function.

4. Quantile Stein Bayesian Optimization

We present our main framework in this section. We start by introducing a novel variational formulation of Batch Bayesian optimization, in which we design a novel variational acquisition function that leverages the quantile-distorted expectation to ensure the quality of the worst query points is maximized, and an entropy regularization to encourage diversity. We then optimize and solve our variational optimization by developing a quantile Stein variational gradient descent (Quantile SVGD) algorithm, based on developing a kernelized functional gradient descent procedure that extends the basic idea of SVGD by Liu & Wang (2016).

4.1. An Variational Framework for Batch BO

Let $[x_1, x_2, ..., x_n]$ be the batch of query points we are optimizing. Assume $x_i \in \mathbb{R}^d$. Because the query points are un-ordered, we can equivalently formulate them using their empirical distribution

$$\rho(x) = \sum_{i=1}^{n} \delta(x - x_i)/n.$$

The optimization can be then formulated as finding the optimal distribution ρ in the space of all distributions. This variational view allows us to naturally incorporate population-level characteristics into optimization, including both diversity and worst-case qualification.

Let $\alpha(x)$ be a standard acquisition function, such as the UCB in (2) and expected impovement (EI). We propose to find the optimal query distribution ρ by maximizing the following objective:

$$\max_{\rho} \left\{ L[\rho] := \mathbb{E}^{\omega}_{x\sim\rho}[\alpha(x)] + \tau H[\rho] \right\}, \tag{4}$$

where $\mathbb{E}^{\omega}[\cdot]$ denotes the quantile-distorted expectation in (3), and $H[\rho] := -\int \rho(x) \log \rho(x) dx$ denotes the entropy functional of ρ; τ is a positive regularization coefficient.

The two terms in $L[\rho]$ explicitly encourages diversity and risk aversion, respectively. The entropy regularization is incorporated to promote the distribution of ρ to spread out, so that the selected query points are different from each other. The quantile-distorted expectation is leveraged to control the minimum $\alpha(x)$ in the query set. We achieve this by defining the weighting function to be

$$\omega(\beta) = \beta^{-\lambda} \quad \text{with} \quad \lambda > 0,$$

so that the query points with lower $\alpha(x)$ contributes more to the overall objective. The parameters τ and λ controls diversity and risk aversion, respectively. We discuss the empirical choice of τ and λ in empirical studies (Section 5).

Algorithm 1 Quantile Stein Variational Gradient Descent

Goal: Find a particle distribution $\hat{\rho}(x) = \sum_{i=1}^n \delta(x-x_i)/n$ to approximately maximize the quantile variational objective function (4), with target function $\alpha(x)$, quantile-distortion function $\omega(\beta)$ and entropy regularization parameter τ.
Initialize a set of points ("particles") $\{x_i\}_{i=1}^n$; decide a scheme for step size ϵ.
while not converge **do**

$$x_i \leftarrow x_i + \frac{\epsilon}{n}\sum_{j=1}^{n}\left[\hat{\zeta}_j \nabla_{x_j}\alpha(x_j)k(x_i,x_j) + \tau\nabla_{x_j}k(x_i,x_j)\right], \quad \forall i = 1,\dots,n,$$

where $\hat{\zeta}_j = \omega(rank(x_j))$ and $rank(x_j) = \sum_{\ell=1}^n \mathbb{I}[\alpha(x_\ell) \le \alpha(x_j)]/n$.
end while

Algorithm 2 Quantile Stein Bayesian Optimization (QSBO)

Goal: Maximizing a black-box function $f(x)$.
Collect a set of initial data $\mathcal{D} = \{x_i, f(x_i)\}$. Decide an entropy regularization coefficient τ and a risk-aversion index λ.
for iteration t **do**
Construct a standard acquisition function $\alpha_t(x)$ based on $\mathcal{D}$ (e.g., using the UCB in (2)).
Find a query points $\{x_i^{new}\}_{i=1}^n$ using quantile SVGD (Algorithm 1) applied on α_t, with distortion function $\omega(\beta) = \beta^{-\lambda}$. Evaluate $f(x_i^{new})$ for $i = 1,\dots,n$ in parallel, and add them to dataset $\mathcal{D}$, that is, $\mathcal{D} = \mathcal{D} \cup \{x_i^{new},\ f(x_i^{new})\}_{i=1}^n$.
end for

4.2. Quantile Stein Variational Gradient Descent

We develop a simple particle-based algorithm for optimizing the quantile variational optimization in (4). Our algorithm is a generalization of Stein variational gradient descent, which corresponds to the special case of (4) when $\omega(\beta) \equiv 1$, and $\mathbb{E}^{\omega}_{x\sim\rho}[\cdot]$ reduces to the typical notion of expectation.

Assume the query points x_i are in $\mathbb{R}^d$. Similar to the derivation of SVGD, we start with a set of initial query points (or particles), and iteratively update the particles by gradient-like update steps:

$$\boldsymbol{T}(x_i) = x_i + \epsilon\boldsymbol{\phi}(x_i), \quad \forall i = 1,\dots,n,$$

where $\boldsymbol{\phi}\colon \mathbb{R}^d \to \mathbb{R}^d$ defines the update direction of the particles and ϵ is the step size that controls the perturbation magnitude. Denote by $\rho_{\boldsymbol{T}}$ the distribution of the updated particles $x' = \boldsymbol{T}(x)$ as $x \sim \rho$. In SVGD, the $\boldsymbol{\phi}$ is chose so that the difference between $L[\rho_{\boldsymbol{T}}]$ and $L[\rho]$ is maximized. This is explicitly formulated as an optimization problem:

$$\boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\phi}\in\mathcal{F}} G[\rho;\ \boldsymbol{\phi}], \qquad (5)$$
$$\text{and} \quad G[\rho;\ \boldsymbol{\phi}] = \frac{\mathrm{d}}{\mathrm{d}\epsilon}\Big(L[\rho_{\boldsymbol{T}}] - L[\rho]\Big)\Big|_{\epsilon=0},$$

where $\mathcal{F}$ is a set of candidate functions of $\boldsymbol{\phi}$, and $G[\rho;\ \boldsymbol{\phi}]$ denotes the increasing rate of $L[\rho]$ as we transform ρ with map $\boldsymbol{T}$, when using an infinitesimal step size ϵ.

The key ingredient of SVGD of Liu & Wang (2016) is a simple closed form solution of the optimal $\boldsymbol{\phi}^*$ above, but restricts to the special case of typical expectation ($\omega(\beta) \equiv 1$). Our key result below shows that we can obtain a surprisingly simple extension of the SVGD solution for the more general quantile-distorted expectation.

Theorem 1. *I) Let F_ρ be the CDF of $z = \alpha(x)$ as $x \sim \rho$. Assume F_ρ is continuous and invertible, and $\alpha(x)$ and $\boldsymbol{\phi}(x)$ are continuous differentiable. We have*

$$G[\rho;\boldsymbol{\phi}] = \mathbb{E}_{x\sim\rho}[\zeta(x)\nabla_x\alpha(x)^\top\boldsymbol{\phi}(x) \ +\ \tau\,\nabla_x^\top\boldsymbol{\phi}(x)]$$

where $\zeta(x) = \omega(F_\rho(\alpha(x)))$. Intuitively, $F_\rho(\alpha(x))$ represents the rank of $\alpha(x)$ in the population following distribution ρ, and hence $\zeta(x)$ denotes the weight assigned on particle x according to its rank of $\alpha(x)$.

II) Let $\mathcal{H}_0$ be a reproducing kernel Hilbert space (RKHS) of scalar-valued functions with a positive definite kernel $k(x,x')$, and $\mathcal{H} = \mathcal{H}_0\times\cdots\times\mathcal{H}_0$ be the set of vector-valued functions of form $\boldsymbol{\phi} = [\phi_1,\dots,\phi_d]$ with $\phi_i \in \mathcal{H}_0$ for $\forall i = 1,\dots,d$. Assume $\nabla_{x,x'}k(x,x')$ exists and is continuous. Define the optimization domain $\mathcal{F}$ in (5) *to be the unit ball of $\mathcal{H}$:*

$$\mathcal{F} = \{f \in \mathcal{H}\colon\ \ ||f||_{\mathcal{H}} \le 1\},$$

Then the solution $\boldsymbol{\phi}^*$ of (5) *satisfies*

$$\boldsymbol{\phi}^*(\cdot) \propto \mathbb{E}_{x\sim\rho}[\zeta(x)\nabla_x\alpha(x)k(x,\cdot) \ +\ \tau\nabla_x k(x,\cdot)]. \quad (6)$$

Theorem 1 is a generalization of the key result of Liu & Wang (2016) (see their Theorem 3.1 and Lemma 3.2), which corresponds to the case when the quantile-distortion function equals one ($\omega(\beta) \equiv 1$), and hence $\zeta(x) \equiv 1$. Therefore, the effect of the quantile distortion simply assigns a weight on each particle x, which can be empirically

estimated easily as we show below. Our extension is mathematically non-trival, for it requires an understanding of how quantile functional changes under variable transforms. See Appendix for the proof.

Quantile Stein Variational Gradient Descent Given the optimal ϕ^* in Theorem 1, we can develop a simple and efficient particle algorithm for optimizing (4). The idea is to start with an initial set of particles $\{x_i\}$, and iteratively update them via $x_i \leftarrow x_i + \epsilon \hat{\phi}^*(x_i)$, for all $i = 1, \ldots, n$, and $\hat{\phi}^*(x)$ is an empirical version of ϕ^* in (6),

$$\hat{\phi}^*(\cdot) = \frac{1}{n}\sum_{j=1}^{n}[\hat{\zeta}(x_j)\nabla_{x_j}\alpha(x_j)k(x_j,\cdot) \; + \; \tau\nabla_{x_j}k(x_j,\cdot)],$$

where $\mathbb{E}_\rho[\cdot]$ is replaced by averaging over the particles and $\hat{\zeta}(x)$ is an empirical approximation of $\zeta(x)$,

$$\hat{\zeta}(x) = \omega(\hat{F}(x)), \quad \hat{F}(x) := \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\left[\alpha(x_i) \le \alpha(x)\right].$$

$\hat{F}(x)$ is an approximation of CDF $F_\rho(x)$, which reflects the rank of x in the population according its value $\alpha(x)$. We take $\omega(\beta) = \beta^{-\lambda}$ with $\lambda > 0$ to encourage risk aversion.

See also Algorithm 1 for our main procedure. Like SVGD, quantile SVGD also leverages the gradient $\nabla\alpha(x)$ to push the particles to maximize $\alpha(x)$, and uses the kernel gradient $\nabla k(x, x')$, weighted by coefficient τ, to encourage the diversity of the particles. Our algorithm differs from the standard SVGD only on introducing the rank-based weight $\hat{\zeta}(x)$, which allows us to introduce risk aversion and better maximize the worst performance among the particles.

Quantile Stein Bayesian Optimization (QSBO) Applying quantile SVGD to batch Bayesian optimization yields our main algorithm, which we summarize in Algorithm 2. We start by collecting a set of initial data $\mathcal{D} = \{x_i, f(x_i)\}$. At each iteration t, we construct a standard acquisition function $\alpha_t(x)$, based on data $\mathcal{D}$ (e.g., using the UCB in (2)), and run quantile SVGD to obtain a set of new queries $\{x_i^{new}\}_{i=1}^n$. We then evaluate $f(x_i^{new})$ for all the query points in parallel, and add them to dataset $\mathcal{D}$, that is, $\mathcal{D} = \mathcal{D} \cup \{x_i^{new}, \; f(x_i^{new})\}_{i=1}^n$, and repeat the process until a stop criterion is reached. When combined with gradient free SVGD (Han & Liu, 2018), QSBO can also handle non-differentiable acquisition functions.

5. Experiments

We start with a toy example to demonstrate the effect of the risk aversion and entropy regularization in quantile SVGD, and then test our quantile Stein Bayesian optimization on a set of standard benchmark problems and two real-world problems, including an automatic genomic selection task and an automatic chemical design task.

Baselines We compare our method with five state-of-the-art batch Bayesian optimization methods, including MACE (Lyu et al., 2018), BUCB (Desautels et al., 2014), GP-UCB-PE (Contal et al., 2013), the local penalization method with UCB (LP-UCB) (González et al., 2016), and GP-UCB-DPP-SAMPLE (DPP) (Kathuria et al., 2016). GP-UCB-PE, BUCB and LP-UCB are three widely used batch BO algorithms based on UCB, while DPP and MACE aims to draw diverse particles, which partially align with our motivation. We use the open-source implementations of MACE[1], BUCB and LP-UCB[2], and implement GP-UCB-PE and DPP by ourselves.

For our QSBO, GP-UCB-PE and DPP, RBF kernel is used, and the Gaussian process hyperparameters are fitted by maximum likelihood. For the other algorithms, we use the default kernel in their implementions (e.g., Lyu et al., 2018; González et al., 2016). For all the experiments, the batch size of the new queries selected at each iteration is set to be 5 unless otherwise specified.

Settings of QSBO We use UCB in (2) as the the acquisition function $\alpha(x)$ in our method. The hyperparameter η is chosen by following the setting in Srinivas et al. (2009), with $\eta = \sqrt{\log t^{d/2+2}\pi^2/3/\delta}$, where t is the iteration step and δ is set as 0.05. We set the number of quantile SVGD steps at each iteration to be 600, with a constant learning rate of 0.1. The kernel in quantile SVGD is the standard RBF kernel and the optimizer is AdaDelta as suggested in the open-source code by Liu & Wang (2016)[3]. Empirically, we find it is useful to turn off the entropy regularization (setting $\tau = 0$) at the last 10% iterations of quantile SVGD (when the different particles have already been pushed into different modes), so that we can find the exact local optima without the impact from the entropy regularization term.

We investigated the choice of the entropy regularization coefficient τ and risk aversion coefficient λ, and found that the results are not sensitive to these parameters once they are the right region. We set $\tau = 5 \times 10^{-2}$ and $\lambda = 1$ in all the experiments unless specified otherwise, and recommend it as a general default setting when using UCB-based acquisition. More discussion is shown below.

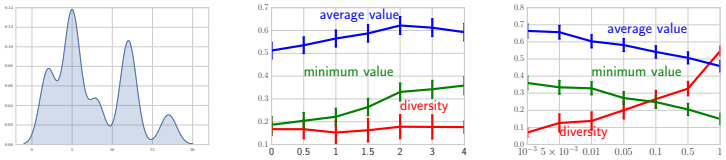
5.1. Illustration on a Toy Example

We run quantile SVGD with $\alpha(x)$ defined by a toy Gaussian mixture example shown in Figure 1 and use it to demonstrate the effect of quantile distortion and entropy

[1] code is at https://github.com/Alaya-in-Matrix/pyMACE

[2] code is at https://github.com/SheffieldML/GPyOpt

[3] code is at https://github.com/DartML/Stein-Variational-Gradient-Descent



(a) Target Function (b) Varying λ, fixed $\tau = 0.1$ (c) Varying τ, fixed $\lambda = 1$

Figure 1. (a) The target function $\alpha(x)$ on which we run quantile SVGD; it is the density of a mixture Gaussian distribution. (b)-(c) The diversity, minimum value $\min_i \alpha(x_i)$ and average value $\sum_i \alpha(x_i)/n$ of the points returned by quantile SVGD as we vary λ and τ, respectively. The y-axis in (b) and (c) is the value of the un-normalized probability density function.

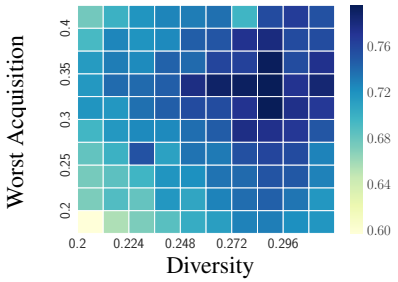


Figure 2. The final reward of QSBO (visualized by color map) for optimizing the function in Figure 1a, as a function of the diversity of the query points and the worst acquisition value in the query batches (both averaged across the iterations of BO).

regularization in quantile SVGD.

Quantitatively, we evaluate the diversity of the particles to be the average of the Euclidean distance of each particle to its nearest neighbor, which can be viewed as a non-parametric estimation of entropy (e.g., Singh et al., 2003). We evaluate the risk aversion by the worst value of the particles, which equals $\min_i \alpha(x_i)$. For reference, we also calculate the average value of the particles, $\sum_i \alpha(x_i)/n$.

Figure 1b shows the result when we vary risk aversion λ, with fixed entropy regularization ($\tau = 0.1$). We can see that increasing λ increases the worst case reward, but seems to have minor impact on the diversity; the average reward increases with λ when λ is small, but decreases as λ becomes too large and the worst case reward dominates. Figure 1c shows the result when we vary entropy regularization τ, with fixed $\lambda = 1$. We can see that diversity increases with τ as expected. We also find that the worst case reward decreases with diversity in this particular case; this is because the optimal region of the target function in Figure 1 is contained in a small region, and hence if the diversity force is too aggressive, the particles would be pushed away from the optimal region.

The Role of Diversity and Risk Aversion Our work is motivated by the observation that a good Batch BO algorithm should yield query points with high diversity and good worst performance. Here we verify this hypothesis, by running our QSBO with function in Figure 1a as the target black-box function, using different combinations of λ and τ. For each trial of QSBO, we calculate the diversity of the query points and the worst acquisition value in the query batches (both averaged across the iterations of BO). The heat map in Figure 2 shows that the final reward of QSBO (visualized by color) is positively correlated with both the average diversity and the worst acquisition values of the query points. This justifies our motivation and explains why QSBO may work well.

5.2. Benchmark Problems

We test our method and the baselines in ten benchmark problems from Bingham (2019), which have dimensions ranging from 2 to 15 (see Table 1 in Appendix for more information). We set the batch size to be 5 of all the Batch BO algorithms. The number of initial random sampling trials is set to be 50 for functions with dimensions high than 10, and 20 for lower dimensional problems. For all functions except the 10D and 15D functions, we set the number of total evaluated points to be 150. For the other functions, the total number of evaluation is set to be 300. The number of iteration of quantile SVGD in our QSBO is set to be 30 for the 2-dimension and 5-dimension functions, and 60 for all the other functions. The experiments were averaged on 20 random seeds.

The Results of the final negative rewards are listed in Table 1. It shows that our QSBO achieves the best result and outperforms the others by a large margin in most cases.

Hyper-parameters We study the choice of hyper-parameters in our method. We tested various other choices of λ and τ and show the results in Table 2-3 in appendix. We find that setting $\lambda = 1$ and $10^{-4} \leq \tau \leq 10^{-1}$ is suitable for most black-box functions. We also tested the case when $\lambda < 0$ (corresponding to risk seeking), and find it is worse than the case when $\lambda > 0$, confirming the importance of risk aversion. We experimented with other batch sizes (e.g., 10), and found the results are similar (see Table 6 in Appendix). We also tested QSBO when combining with other acquisition functions, e.g. EI, PI and did not find any single acquisition function dominates (see Table 5 in Appendix).

5.3. Automatic Genomic Selection

Genomic prediction is the task of searching for superior genotypes in a large number of accessions in germplasm

	GP-BUCB	GP-UCB-PE	LP-UCB	DPP	MACE	QSBO-UCB
Branin	2.96e-4±0.00	0.24±0.02	3.28e-4±0.00	9.63e-4±0.00	**2.85e-5±0.00**	5.14e-5±0.00
Eggholder	1.26e2±1.14e2	80.58±60.13	51.34±39.32	82.81±54.98	74.14±62.27	**46.86±44.92**
Dropwave	0.18±0.12	0.14±0.10	0.14±0.18	0.13±0.16	0.11±0.08	**0.07±0.06**
CrossInTray	9.05e-3±0.01	2.62e-2±0.00	6.83e-3±0.01	7.64e-3±0.01	3.78e-4±0.00	**1.35e-4±0.00**
gSobol5	2.91±0.59	17.91±5.32	1.85±2.08	2.34±0.52	1.14±0.37	**0.32±0.28**
gSobol10	8.79e2±7.93e2	4.58e3±3.82e3	1.04e2±3.13e2	1.07e3±7.82e2	48.92±31.97	**31.19±21.18**
gSobol15	7.84e4±5.27e4	1.89e4±1.09e4	2.34e3±3.91e3	5.28e3±4.25e3	6.39e2±4.22e2	**3.61e2±4.38e2**
Ackley5	3.86±1.27	18.02±0.63	3.71±1.93	3.74±0.97	2.36±0.37	**2.23±0.59**
Ackley10	4.12±0.42	18.98±0.27	3.87±3.05	4.23±3.92	3.01±0.63	**2.41±0.44**
Alpine2	83.45±23.09	1.34e2±17.33	75.92±23.83	73.39±24.17	**63.29±17.28**	73.01±19.49

Table 1. The negative rewards of different methods on the benchmark functions. Results are averaged over 20 runs.

	GP-BUCB	GP-UCB-PE	LP-UCB	DPP	MACE	QSBO-UCB
Brown Rice Surface Area	0.85±0.13	0.78±0.14	0.89±0.09	0.91±0.11	**0.95±0.04**	0.94±0.04
Brown Rice Seed Width	0.69±0.15	0.80±0.18	0.83±0.11	0.82±0.11	0.86±0.05	**0.89±0.03**
Brown Rice Seed Length	0.95±0.03	0.95±0.07	0.96±0.07	0.95±0.05	**0.98±0.03**	**0.98±0.02**
Plant Height	0.89±0.06	0.92±0.07	0.93±0.08	**0.95±0.02**	0.94±0.04	0.90±0.08
Flag Leaf Length	0.95±0.05	**0.97±0.02**	0.94±0.06	0.92±0.06	0.93±0.04	**0.97±0.02**
Protein Content	0.92±0.03	0.87±0.10	0.89±0.04	0.91±0.09	0.92±0.07	**0.94±0.08**
Flowering Time at Arkansas	0.84±0.09	0.81±0.17	0.84±0.09	0.82±0.07	0.85±0.05	**0.87±0.07**

Table 2. Rewards of different methods on the *Best Found Y* genomic selection tasks. For all the methods, we use a batch size of 10 and run 10 iterations, hence yielding a 100 evaluation in total. Results are averaged on 20 random seeds.

collections preserved in gene banks (Desta & Ortiz, 2014). We focus on pre-breeding genomic prediction using *rice dataset*, which analyzes the association between the 43K SNP markers and the phenotypic values of 34 traits of *Oryza sativa* (a species of Asian rice) (Zhao et al., 2011) [4]. BO has been applied to this problem (e.g., Tanaka & Iwata, 2018), but batch BO has not been considered. However, evaluation in this case is extremely costly in practice (which requires to actually plant the rice), parallel evaluation is highly favorable and allows us to speed up rice genotype selection by the order of years.

In rice dataset, we aim to find optimal rice genotype to optimize seven different traits (showed in different rows of Table 2). The search space is G^{32} where $G = \{T, C, G, A\}$, which embed into a continuous space of $\mathbb{R}^{32}$, where $\{T, C, G, A\}$ is mapped to $\{1, 2, 3, 4\}$. Although this looks a simple heuristic, we find it works well for this particular task. The dataset contains 43k genotypes whose traits have been pre-evaluated by researchers. For our simulation, we find the nearest neighbors of the query points as the true evaluate points at each iteration of BO; this is equivalent to running standard BO on a piece-wise constant function constructed by a nearest neighbor regression on the dataset.

[4]dataset at http://www.ricediversity.org/data/index.cfm

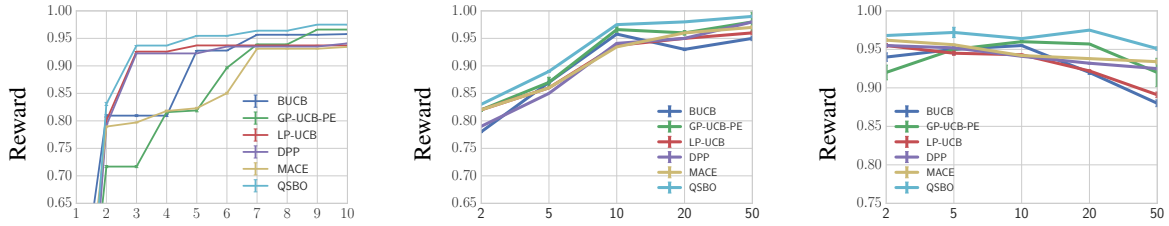
Results The results of various batch BO are shown in Table 2, all of which are the baselines reported in Tanaka & Iwata (2018). We find that our QSBO achieves the best result on five out of seven traits. As shown in Fig. 3, we find that our method converges faster than the other existing approaches, suggesting the practical value of our method for this problem. Furthermore, we also explore the performance of the experiments with different batch sizes and find that our proposed method outperforms others in most settings.

5.4. Automatic Chemical Design

The goal of automatic chemical design is to predict novel molecules that have certain desirable properties. We focus on a drug discovery problem following the setting in Gómez-Bombarelli et al. (2018); Griffiths (2017), which uses a per-trained variational autoencoder (VAE) developed in Gómez-Bombarelli et al. (2018) to map molecule structures to a 36 dimensional continuous latent space. BO is performed on the latent space, and molecule can be reconstructed by using the decoder of the VAE. The VAE is trained on the ZINC dataset[5].

Our goal is to find the best molecules that optimize three

[5]http://zinc15.docking.org/



(a) Learning Curve (b) Fixed # Iteration (b) Fixed total # of Evaluation

Figure 3. Further analysis on the *Flag Leaf Length* task from Table 2. (a) The learning curve of the optimizing procedure (iteration vs. reward), with batch size 10; see Appendix for the learning curves of the other tasks. (b) The reward of different methods as we vary the batch size, while keeping the total number of iteration fixed to be 10. (c) The reward of different methods as we vary the batch size, while keep the total number of evaluation points fixed to be 100 (so the iteration step equals 100/batch-size).

	GP-BUCB	GP-UCB-PE	LP-UCB	DPP	MACE	QSBO-UCB
QED	0.89±0.08	0.87±0.12	0.91±0.05	0.91±0.06	0.92±0.03	**0.93±0.03**
SAS	2.27±0.09	2.53±0.17	2.18± 0.06	2.29±0.08	2.16±0.04	**2.08±0.05**
LogP	0.51±0.07	0.76±0.23	0.50±0.11	0.47±0.07	0.41±0.06	**0.33±0.08**

Table 3. Rewards of different methods on the automatic chemical design tasks. For all methods, we use a batch size of 5 and run 20 iterations (and hence 100 evaluation points in total). Results are averaged on 20 runs.

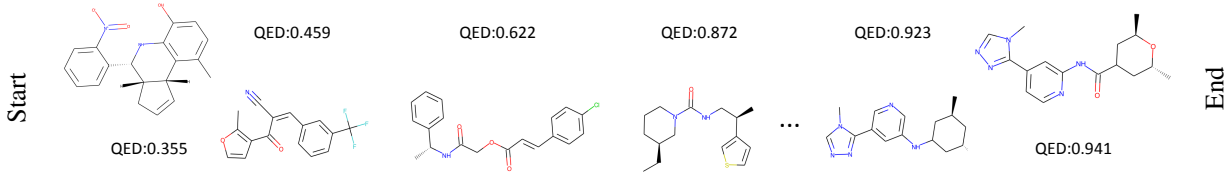


Figure 4. Illustration of the search process of our QSBO-UCB on the QED task.

different properties widely used for benchmarking purposes: water-octanal partition coefficient (LogP), synthetic accessibility score (SAS) and Qualitative Estimate of Drug-likeness (QED). The goal is to maximize QED, while minimizing SAS and LogP. The dataset used by Gómez-Bombarelli et al. (2018); Griffiths (2017) provides 65K candidate molecules that have been pre-evaluated. To conduct BO simulation, we evaluate the nearest neighbors of the query points returned by various methods. This is again effectively optimizing a piece-wise constant function constructed by performing a nearest neighbor regression on the dataset.

Results The results are reported in Table. 3, from which we can see that QSBO outperforms baselines for all the three objectives. The improvement is especially high for LogP. To visualize our optimization process, we plot in Figure 4 the molecule structures of the best evaluated molecules in the first four and last two iterations from our QSBO algorithm, along with their corresponding QED scores. We can observe that large modifications and improvements occur during the earlier stages, and then smooth and local modifications occur in the last two iterations.

6. Conclusion and Future Work

We propose a design principle of batch Bayesian optimization, which seek to jointly optimize and balance the diversity and risk aversion in batch query optimization. This allows us to derive quantile Stein Bayesian optimization, based on a novel quantile-based varitional framework which we solve using a simple and efficient quantile SVGD algorithm. Our results show that our new method consistently outperforms other recent strong baseline methods.

Our work opens a range of potential future works, including further improving our algorithm in high dimensional settings, and extending it to handle discrete search spaces in more principled ways. The quantile SVGD we developed is also of its own interest, and can be applied to various other challenging problems in risk sensitive areas.

Acknowledgement

This work is supported in part by NSF CRII 1830161, NSF CAREER 1846421, and a Tencent AI Lab Rhino-Bird Gift Fund. We would like to acknowledge Google Cloud for their support.

References

Acerbi, C. Spectral measures of risk: A coherent representation of subjective risk aversion. *Journal of Banking & Finance*, 26(7):1505–1518, 2002.

Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Balbás, A., Garrido, J., and Mayoral, S. Properties of distortion risk measures. *Methodology and Computing in Applied Probability*, 11(3):385, 2009.

Balog, M., Lakshminarayanan, B., Ghahramani, Z., Roy, D. M., and Teh, Y. W. The mondrian kernel. *arXiv preprint arXiv:1606.05241*, 2016.

Bingham, D. Virtual library of simulation experiments: Test functions and datasets. `https://www.sfu.ca/~ssurjano/index.html`, 2019. [Online; accessed 20-Jan-2019].

Contal, E., Buffoni, D., Robicquet, A., and Vayatis, N. Parallel gaussian process optimization with upper confidence bound and pure exploration. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part I*, pp. 225–240, 2013.

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

Desta, Z. A. and Ortiz, R. Genomic selection: genome-wide prediction in plant improvement. *Trends in plant science*, 19(9):592–601, 2014.

Duval, T. S. and Silvia, P. J. Self-awareness, probability of improvement, and the self-serving bias. *Journal of personality and social psychology*, 82(1):49, 2002.

Fernandez, M., Caballero, J., Fernandez, L., and Sarai, A. Genetic algorithm optimization in drug design qsar: Bayesian-regularized genetic neural networks (brgnn) and genetic algorithm-optimized support vectors machines (ga-svm). *Molecular diversity*, 15(1):269–289, 2011.

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

Frazier, P. I. and Clark, S. C. Parallel global optimization using an improved multi-points expected improvement criterion. In *INFORMS Optimization Society Conference, Miami FL*, volume 26, 2012.

Gilchrist, W. *Statistical modelling with quantile functions*. Chapman and Hall/CRC, 2000.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

González, J., Dai, Z., Hennig, P., and Lawrence, N. D. Batch bayesian optimization via local penalization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, pp. 648–657, 2016.

Griffiths, R.-R. Constrained bayesian optimization for automatic chemical design. *arXiv preprint arXiv:1709.05501*, 2017.

Han, J. and Liu, Q. Stein variational gradient descent without gradient. *International Conference on Machine Learning*, 2018.

Jones, D. R., Schonlau, M., and Welch, W. J. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

Kathuria, T., Deshpande, A., and Kohli, P. Batched gaussian process bandit optimization via determinantal point processes. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 4206–4214, 2016.

Klein, A., Falkner, S., Bartels, S., Hennig, P., and Hutter, F. Fast bayesian optimization of machine learning hyperparameters on large datasets. In *Artificial Intelligence and Statistics*, pp. 528–536, 2017.

Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose bayesian inference algorithm. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 2378–2386. Curran Associates, Inc., 2016.

Lyu, W., Yang, F., Yan, C., Zhou, D., and Zeng, X. Batch Bayesian optimization via multi-objective acquisition

ensemble for automated analog circuit design. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3312–3320, StockholmsmÃd'ssan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Mitra, S. and Ji, T. Risk measures in quantitative finance. *International Journal of Business Continuity and Risk Management*, 1(2):125–135, 2010.

Negoescu, D. M., Frazier, P. I., and Powell, W. B. The knowledge-gradient algorithm for sequencing experiments in drug discovery. *INFORMS Journal on Computing*, 23(3):346–363, 2011.

Nyikosa, F. M., Osborne, M. A., and Roberts, S. J. Adaptive bayesian optimisation for online portfolio selection. In *Workshop on Bayesian Optimization at NIPS*, volume 2015, 2015.

Oh, C., Gavves, E., and Welling, M. BOCK : Bayesian optimization with cylindrical kernels. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 3865–3874, 2018.

Rasmussen, C. E. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pp. 63–71. Springer, 2004.

Rolland, P., Scarlett, J., Bogunovic, I., and Cevher, V. High-dimensional bayesian optimization via additive models with overlapping groups. In *International Conference on Artificial Intelligence and Statistics*, pp. 298–307, 2018.

Shah, A. and Ghahramani, Z. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Advances in Neural Information Processing Systems*, pp. 3330–3338, 2015.

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

Singh, H., Misra, N., Hnizdo, V., Fedorowicz, A., and Demchuk, E. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23(3-4):301–321, 2003.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Tanaka, R. and Iwata, H. Bayesian optimization for genomic selection: a method for discovering the best genotype among a large number of candidates. *Theoretical and Applied Genetics*, 131(1):93–105, 2018.

Turgeon, M., Lustig, C., and Meck, W. H. Cognitive aging and time perception: roles of bayesian optimization and degeneracy. *Frontiers in aging neuroscience*, 8:102, 2016.

Wang, Z., Zhou, B., and Jegelka, S. Optimization as estimation with gaussian processes in bandit settings. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, pp. 1022–1031, 2016.

Wang, Z., Li, C., Jegelka, S., and Kohli, P. Batched high-dimensional bayesian optimization via structural kernel learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 3656–3664, 2017.

Wang, Z., Gehring, C., Kohli, P., and Jegelka, S. Batched large-scale bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, pp. 745–754, 2018.

Wirch, J. L. and Hardy, M. R. Distortion risk measures: coherence and stochastic dominance. In *International Congress on Insurance: Mathematics and Economics*, pp. 15–17, 2001.

Wu, J. and Frazier, P. The parallel knowledge gradient method for batch bayesian optimization. In *Advances in Neural Information Processing Systems*, pp. 3126–3134, 2016.

Wu, J., Poloczek, M., Wilson, A. G., and Frazier, P. Bayesian optimization with gradients. In *Advances in Neural Information Processing Systems*, pp. 5267–5278, 2017.

Zhao, K., Tung, C.-W., Eizenga, G. C., Wright, M. H., Ali, M. L., Price, A. H., Norton, G. J., Islam, M. R., Reynolds, A., Mezey, J., et al. Genome-wide association mapping reveals a rich genetic architecture of complex traits in oryza sativa. *Nature communications*, 2: 467, 2011.