Bayesian Functional Optimisation with Shape Prior

Pratibha Vellanki,[1] **Santu Rana,**[1] **Sunil Gupta,**[1] **David Rubin de Celis Leal,**[2]
Alessandra Sutti,[2] **Murray Height,**[2] **Svetha Venkatesh**[1]

[1]Centre for Pattern Recognition and Data Analytics
Deakin University, Geelong, Australia
pratibha.vellanki, santu.rana, sunil.gupta, svetha.venkatesh@deakin.edu.au
[2]Institute for Frontier Materials, GTP Research, Deakin University, Geelong, Australia
d.rubindecelisleal, alessandra.sutti, murray.height@deakin.edu.au

Abstract

Real world experiments are expensive, and thus it is important to reach a target in a minimum number of experiments. Experimental processes often involve control variables that change over time. Such problems can be formulated as functional optimisation problem. We develop a novel Bayesian optimisation framework for such functional optimisation of expensive black-box processes. We represent the control function using Bernstein polynomial basis and optimise in the coefficient space. We derive the theory and practice required to dynamically adjust the order of the polynomial degree, and show how prior information about shape can be integrated. We demonstrate the effectiveness of our approach for short polymer fibre design and optimising learning rate schedules for deep networks.

Introduction

Functional optimisation arises when a time-varying system requires optimal control variable values to change with time. As an example consider optimising the learning rate schedule while training a neural network. The learning rate schedule can be expressed as a function of time and often there is knowledge about the shape of this function; traditionally, it is a decreasing function. Consider also recirculation processes, common in industries like drug production or plastic recycling. Recirculation involves the reintroduction of partially formed output product into the input of the system until target output is reached. Recirculation might require adjusting of input parameters as a function of time to keep the system optimal throughout. Often, industrial knowledge exists about the trend of this adjustment.

We propose a Bayesian functional optimisation algorithm for expensive processes that offers two main capabilities: a) allows detection of underspecification of complexity of the functional search space and adjusting for it in a dynamic fashion, and b) admits loose prior information on the shape of the function.

The closest work is that of (Vien, Zimmermann, and Toussaint 2018), where functions are represented in a functional RKHS, captured through functional kernels. Based on that a Gaussian process is constructed and subsequently, Bayesian optimisation is performed. However, their method does not

allow incorporating prior information or for adjustment of function complexity mid-optimisation. When Bayesian optimisation is used for optimisation of physical experiments loose prior knowledge may be available. But, only recently, the value of using such information in Baysian optimisation is being looked at (Li et al. 2018).

Our solution is primarily based on representing the control function on Bernstein polynomial basis (Bernstein 1912) and then optimising in the coefficient space. Bernstein polynomials basis follow the Stone-Weierstrass approximation theorem i.e. any function on a bounded subspace when represented on this basis system can be pointwise approximated to an arbitrary precision. Whilst the theorem is true for any polynomial basis system, Bernstein basis offers many unique properties, some of which are critical for achieving our goals. They are presented in more detail in proposed framework. Once the control functions are represented on a Bernstein basis with a suitable order, we can directly use the coefficient vector as the input subspace for the global optimisation of a function, which maps the coefficient vector to the outcome of the system.

Bernstein polynomials have been a popular choice in the field of aerospace for the optimisation of aerofoil geometry. For example, it has been used in (Kulfan and Bussoletti 2006) to convert a shape optimisation problem into a function optimisation problem. Often Computational Fluid Dynamic (CFD) tools are used to optimise such geometries (Samareh 2001). However, these methods are not designed to be sample-efficient, and hence, not feasible for expensive optimisation tasks. The proposed Bayesian functional optimisation algorithm addresses the optimisation of expensive experimental processes, two of which we present in this paper (physical recirculation systems, and optimisation of learning rate schedule for a large neural network model being trained on a large dataset).

Mathematically, if $g(t)$ is the control function that drives the system output (y) by a functional $h : g(t) \rightarrow y$ where $g \in B(R)$, the space of all bounded real-valued functions, then the original functional optimisation problem can be written as:

$$g^*(t) = \text{argmax}_{g(t) \in B(R)} h(g(t))$$

When we convert $g(t)$ onto the nth order Bernstein polynomial as $g(t) = \sum_{v=0}^{n} \alpha_v b_{v,n}(t)$, where $\boldsymbol{\alpha} = \{\alpha_v\}_{v=1}^{n}$ are
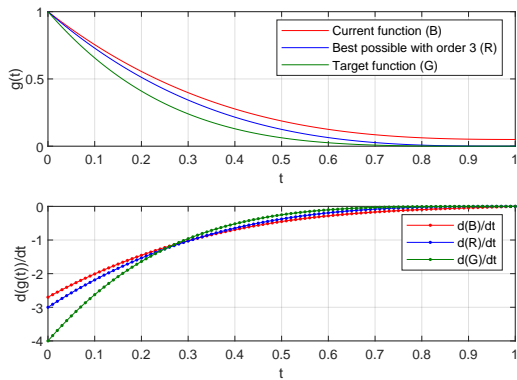
Figure 1: Intuition behind determining whether the current order of Bernstein suffices to capture the optimal function.

Recirculate the butanol and short polymer fiber mixture

Figure 2: Short polymer fibre production using recirculation.

the Bernstein coefficients and $b_{v,n}(t)$ are the base polynomials (more in framework), the above optimisation problem can thus be reformulated in a more familiar function optimisation problem as:

$$\boldsymbol{\alpha}^* = \operatorname{argmax}_{\alpha \in \mathcal{A}} f(\boldsymbol{\alpha})$$

where $f : \boldsymbol{\alpha} \to y$. In addition to a control function, we may also have some other control variables ($\boldsymbol{u}$) that need to be optimised. For example, in the case of neural network hyper-parameter tuning, $\boldsymbol{u}$ may represent network size parameters, whilst $\boldsymbol{\alpha}$ represents the learning rate schedule. In such cases, our formulation can be easily extended to $\{\boldsymbol{\alpha}^*, \boldsymbol{u}^*\} = \operatorname{argmax}_{\alpha \in \mathcal{A}, u \in \mathcal{U}} f(\boldsymbol{\alpha}, \boldsymbol{u})$. Putting them together, we will henceforth indicate $\boldsymbol{x} = \{\boldsymbol{\alpha}, \boldsymbol{u}\}$, and $\mathcal{X} = \mathcal{A} \cup \mathcal{U}$.

Next, following the properties of derivative of a function of Bernstein basis, and using results from (Chang et al. 2007), we show that prior information on shapes such as monotonicity or unimodality can be encoded by simply adding constraint functions on the values of $\boldsymbol{\alpha}$. We further propose a principled way to detect if the order of the polynomial is underspecified. The intuition is as follows: Fig 1 shows three functions at the top and their corresponding derivative at the bottom panel with matching colour. Let us assume that the current function we have with a 3rd order Bernstein polynomial is the red function. The maximum derivative of the red function happens at $t = 0$ and it is around -2.8. The target function is depicted via the colour green and its maximum derivative as seen from the bottom panel is -4 at $t = 0$. In the absolute sense, the derivative magnitude is higher than the current function that we have. However, we see that the derivative limit of a 3rd order Bernstein polynomial (blue function) is at -3, which in the magnitude sense is lower than the maximum derivative of the target function (green). Thus the optimal function (green) cannot be reached by a 3rd order Bernstein polynomial. The target function can only be reached if the order is increased. An underspecification of the order is thus detected if the derivative magnitude is close to the theoretical maxima possible within the current order of the polynomial basis based on
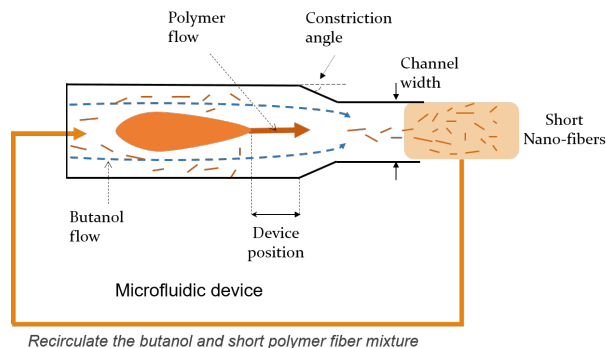
the ranges over $\boldsymbol{\alpha}$. When an underspecification is detected, we increment the order of the Bernstein basis. We derive a key lemma to compute the maximum of the derivative for a function realised using Bernstein polynomial basis of a fixed order. Next, using the existing results of order elevation for Bernstein basis, we show how to reuse observation obtained using lower order basis for the new higher order basis. In some cases, it may not be possible to detect order underspecification via derivative checking (if the function has more number of modes than can be modelled by the current order of the polynomial), and hence, we also increment order at a fixed interval up to a maximum specified order. At all times, we use all the observations using order elevation technique. Convergence is guaranteed as long as the function is realisable within the maximum order specified.

We apply our algorithm to two problems: the design and production of concentrated short polymer fibre solution using recirculation and learning rate schedule optimisation for neural network training. Interesting new materials like short polymer fibres can impart exotic properties to natural fabrics (Feng et al. 2003; Ma, Hill, and Rutledge 2008). Production involves injecting a liquid polymer into a high speed butanol flow through a specially designed apparatus (Fig 2) (Li et al. 2017). This turns the liquid polymer into short nano-scale fibres. The control variables include the apparatus geometry and flow rates which in turn determine the produced fibre quality. To increase the fibre concentration in the mixture produced, the same mixture (butanol+fibre) is recirculated through the apparatus, keeping the polymer flow uninterrupted. Since the recirculation process introduces dynamics in the constituents of the mixture, one may need to change the control variables to keep them optimal throughout this dynamic process. We apply our algorithm in maximising the quality of fibre yield for the already mentioned short polymer fibre production process. During recirculation we only change the butanol flow rate as a function of time, as others are not easy to change dynamically in the used setup. We used the experimenter's hunch that an increasing flow rate will result in the highest quality fibre. In our experiments, we found two profiles, one which is nearly constant, and the other which is increasing that both result in the highest quality possible, loosely validating the experimenter's hunch.

In neural network training, the learning rate schedule can

be modelled either as a long vector or as a function of epochs (Bengio 2012). The latter is attractive as the smoothness in the consecutive learning rate values implies smaller effective design space when considered as a function than with a full blown vector of the corresponding schedule. We applied our algorithm for learning rate schedule optimisation and found that an optimised learning rate schedule can even make SGD to perform better than both the method proposed by (Vien, Zimmermann, and Toussaint 2018) and a state of the art optimiser with automatic scheduling like Adam.

Bayesian optimisation

Bayesian optimisation is a global optimisation method for expensive black-box function (Jones, Schonlau, and Welch 1998; Mockus 1994). The optimisation problem:

$$\boldsymbol{x}^* = \text{argmax}_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$$

The function is usually modelled using a *Gaussian Process* (Rasmussen 2006) as the prior *i.e.*

$$f(\boldsymbol{x}) \quad \sim \quad \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')).$$

where $m(\boldsymbol{x})$ and $k(\boldsymbol{x}, \boldsymbol{x}')$ are the mean and the covariance functions of the Gaussian process (Brochu, Cora, and de Freitas 2010). Mean $m(\boldsymbol{x})$ can be assumed to be a zero function without any loss of generalisation. Popular covariance functions include squared exponential (SE) kernel, Matérn kernel, etc. The predictive mean and variance of the Gaussian process is a Gaussian distribution, which encapsulates epistemic uncertainty. Using an observation model of $y = f(\boldsymbol{x}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_{noise}^2)$, and denoting $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^t$, the predictive distribution can be derived as:

$$P(f_{t+1} \mid \mathcal{D}_{1:t}, \boldsymbol{x}) = \mathcal{N}(\mu_{t+1}(\boldsymbol{x}), \sigma_{t+1}^2(\boldsymbol{x}))$$

where, $\boldsymbol{k} = [k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_t)]$, the kernel matrix $[K_{ij}] = k(\boldsymbol{x}_i, \boldsymbol{x}_j) \forall i, j \in 1, \ldots, t$, and

$$\mu_{t+1}(\boldsymbol{x}) = \boldsymbol{k}^T [K + \sigma_{noise}^2 I]^{-1} y_{1:t}$$
$$\sigma_{t+1}(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}^T [K + \sigma_{noise}^2 I]^{-1} \boldsymbol{k}$$

Next, a surrogate utility function called *acquisition function* is constructed to find the next sample to evaluate. It balances two contrastive needs of sampling at the high mean location versus sampling at the high uncertainty location so that the global optima for $f(.)$ is reached in a fewer number of samples. Acquisition functions are either constructed based on improvement over the current best (e.g. Probability of Improvement, Expected Improvement (Kushner 1964; Mockus, Tiesis, and Zilinskas 1978)), or information based criteria (e.g. Entropy Search (Hennig and Schuler 2012). Predictive Entropy Search (Hernández-Lobato, Hoffman, and Ghahramani 2014)), or confidence based criteria (e.g. GP-UCB (Srinivas et al. 2010)). A GP-UCB acquisition function for the $(t+1)$ th iteration is:

$$a_{t+1}(\boldsymbol{x}) = \mu_t(\boldsymbol{x}) + \sqrt{\beta_{t+1}} \sigma_t(\boldsymbol{x})$$

where β_t is an increasing sequence of $O(\log t)$. An example sequence can be $\beta_t = 2 \log(t^{d/2+2} \pi^2 / 3\delta)$ where d represents the dimensionality of the data and $1 - \delta$ is the probability of convergence. A regret r_t is defined as the difference of the t'th function evaluation and the global maxima, i.e. $r_t = max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) - f(\boldsymbol{x}_t)$, and the cumulative regret is defined as $R_t = \sum_{t'=1}^t r_{t'}$. It can be shown that when Gaussian process is used with SE kernel then $R_t \sim O(\sqrt{t(\log t)^{d+1}})$, *i.e.* the cumulative regret only grows sublinearly and $Lim_{t \to \infty} R_t / t \to 0$, implying a 'no regret' algorithm. Generic Bayesian optimisation is a sequential algorithm with one recommendation per iteration. However, when at each iteration it is convenient to perform a batch of recommendation. it can be altered to batch Bayesian optimisation (Contal et al. 2013; Desautels, Krause, and Burdick 2014; Vellanki et al. 2017; Gupta et al. 2018).

Proposed framework

As previously mentioned, we model the control function $g(t)$ using the Bernstein polynomial basis. Instead of optimising the function $g(t)$ directly, we optimise its Bernstein coefficients $\{\alpha_v\}_{v=1:n}$. In this manner, we are able to convert our functional optimisation problem into a vector optimisation problem. In this section, first we present how the optimum control function can be found in the presence of basic shape information. We also discuss how the order of the polynomial can be adjusted based on the complexity of the control function being optimised.

Bernstein polynomial representation with shape constraints

An nth order Bernstein polynomial as a linear combination of its basis polynomials is represented as

$$g_n(t) = \sum_{v=0}^n \alpha_v b_{v,n}(t) \qquad (1)$$

where $b_{v,n}(t) = \binom{n}{v} t^v (1-t)^{n-v}$ are the Bernstein basis polynomials for order n defined on $[0, 1]$, $\binom{n}{v}$ is the binomial coefficient, and α_v are the Bernstein coefficients. In other words, the Bernstein polynomial is the weighted sum of the basis polynomials. We first present a lemma that guarantees universality of Bernstein polynomial basis.

Lemma 1. *(Bernstein 1912).* Any continuous function f defined on the closed interval $[0, 1]$ can be uniformly approximated by a Bernstein polynomial function $B_n(f)$. Let $B_n(f)(t) = \sum_{v=0}^n f\left(\frac{v}{n}\right) b_{v,n}(t)$. then as $n \to \infty$, $B_n(f)$ converges to the function f, i.e. $lim_{n \to \infty} B_n(f) = f$. □

Next, we present the lemma to control the shape of the function. Our interest lies in the elegant relationship between the Bernstein coefficients $\{\alpha_v\}_{v=0}^n$ and the shape of the Bernstein polynomial. In Theorem 1 and 2, we elaborate on the details of this relationship for the monotonic function and the unimodal function case. The following Lemma leads us towards Theorems 1 and 2.

Lemma 2. For a Bernstein polynomial $g_n(t) = \sum_{b=0}^n \alpha_v b_{v,n}(t)$, the derivative of the polynomial is given by $g_n'(t) = n \sum_{v=0}^{n-1} (\alpha_{v+1} - \alpha_v) b_{v,n-1}(t)$. In other words the derivative of the nth order Bernstein polynomial can be expressed through a linear combination of Bernstein base polynomials up to order $(n-1)$. *(Please refer to supplementary material for detailed proof)* □

Theorem 1. *(Monotonicity):* If $\alpha_{v+1} \geq \alpha_v$, then $g_n(t)$ is a monotonically increasing function. Similarly if $\alpha_{v+1} \leq \alpha_v$, then $g_n(t)$ is a monotonically decreasing function. *(Chang et al. 2007)*

Proof. From Lemma 2, consider the derivative of the Bernstein polynomial $g_n'(t) = n \sum_{v=0}^{n-1} (\alpha_{v+1} - \alpha_v) b_{v,n-1}(t)$. Here, the base polynomials $\{b_{v,n-1}(t)\}_{v=0}^{n-1}$ are always positive by definition. Therefore, if the difference $(\alpha_{v+1} - \alpha_v)$ is kept positive, the derivative $g_n'(t)$ remains positive implying $g_n(t)$ to be a monotonically increasing function. A similar argument can be made for $g_n(x)$ to be a monotonically decreasing function provided $\alpha_{v+1} < \alpha_v$. $\qquad\square$

Theorem 2. *(Unimodality)(Chang et al. 2007):* For $n \geq 3$. If $\alpha_0 = \alpha_1 = ... = \alpha_{l_1} < \alpha_{l_1+1} \leq \alpha_{l_1+2} \leq ... \leq \alpha_{l_2}$ and $\alpha_{l_2} \geq \alpha_{l_2+1} \geq ... \geq \alpha_{l_3} > \alpha_{l_3+1} = ... = \alpha_n$ for some $0 \leq l_1 < l_2 \leq l_3 \leq n$, then there exists some $s \in (0, \tau)]$ such that s is the unique maximum point of $g_n(t)$ and $g_n(t)$ is strictly increasing on $[0, s]$ and it is strictly decreasing on $[s, \tau]$.

Proof. Please refer to (Chang et al. 2007) for proof. The proof also uses Lemma 2 as a key ingredient. $\qquad\square$

These theorems are used to formulate the constraints in the next subsection. Chang et al. (Chang et al. 2007) have also provided theory for the cases other than unimodal concave. Our framework can be extended to all such cases where such a relationship between the coefficients and the shape of the polynomial has been established. Often the prior information about the trend is available from the domain experts. Specifically, in the design of short polymer fibre, it is known that fibres get shorter as they are recirculated. Hence, to get a narrow size-distribution we need to produce longer fibres initially, with a gradual reduction in length. This would require butanol flow to increase monotonically with time. On the contrary, for deep learning schedule tuning we use monotonically decreasing constraint. Our method can also be applied to applications where the function is either unimodal or even when there is no prior information available about the shape of the control function.

For illustration, let us consider the functions that are represented via third order Bernstein polynomial basis. The base polynomials in this case are $b_{0,3}(t) = (1-t)^3$, $b_{1,3}(t) = 3t(1-t)^2$, $b_{2,3}(t) = 3t^2(1-t)$, and $b_{3,3}(t) = t^3$. Fig 3 shows the sample functions from functions with only range constraint, as well as functions with monotonicty constraint and unimodality constraint.

Control function optimisation with shape constraints

We recall our optimisation problem as:
$$\boldsymbol{x}^* = \text{argmax}_{x \in \mathcal{X}} f(\boldsymbol{x})$$
where $\boldsymbol{x} = \{\boldsymbol{\alpha}, \boldsymbol{u}\}$, $\mathcal{X} = \mathcal{A} \cup \mathcal{U}$, $\boldsymbol{\alpha}$ is the Bernstein coefficient vector and $\boldsymbol{u}$ are the other fixed control variables. When prior information about the shape of the control function is available, then it boils down to usual constrained optimisation problem as:
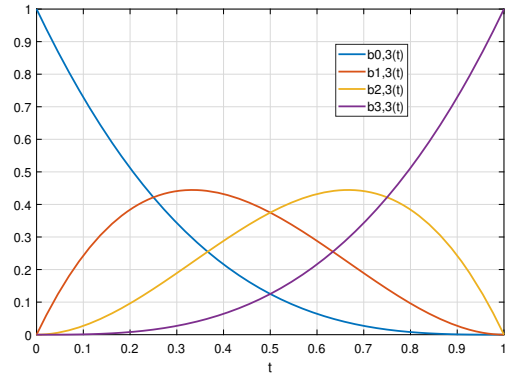
(a) Bernstein base polynomials for the third order Bernstein polynomial.
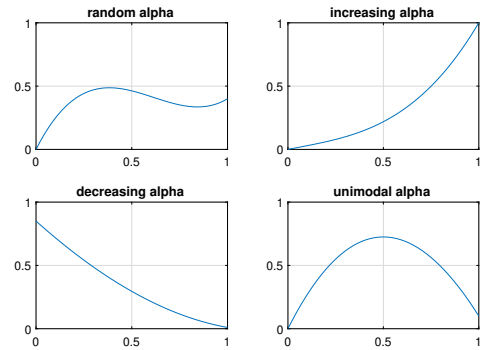
(b) Examples of Bernstein polynomial $g_3(t)$ based on how alphas are sampled.

Figure 3: Examples of a third order Bernstein polynomial.

$$\boldsymbol{x}^* = \text{argmax}_{x \in \mathcal{X}} f(\boldsymbol{x}) \qquad s.t.\ \mathbb{C} \geq 0$$

where $\mathbb{C}$ is a set of inequality constraints over the parameter space. Such constraints can be easily enforced during acquisition function optimisation. In the following, we present the $\mathbb{C}$ for monotonically increasing and decreasing control functions, and unimodal control functions based on the Theorems 1 and 2.

Controlling the range of the control function: By choosing the values of $\alpha_{0:n} \in [0, 1]$ the Bernstein polynomial is limited to the range $[0, 1]$ (this is possible due to Lemma 3, which will be described in the next subsection). For any application, optimised Bernstein polynomial can then be rescaled using the known range of input function.

Increasing control function: $\mathbb{C} = \{C_v\}_{v=0}^{n-1}$ where $C_v = \alpha_{v+1} - \alpha_v$. This is used in the design of recirculation control function in short polymer fibre production experiment as presented in experiments.

Decreasing control function: $\mathbb{C} = \{C_v\}_{v=0}^{n-1}$ where $C_v = \alpha_v - \alpha_{v+1}$. These constraints are used in modelling

Algorithm 1 Framework for control function optimisation.

Input: Observations $\mathcal{D}_{1:m} = \{\boldsymbol{x}_i, y_i\}_{i=1}^m$, where $y_i = f(g_n(t) \mid \boldsymbol{x}_i) + \varepsilon$ for an nth order Bernstein polynomial, $\boldsymbol{x}_i = \{\boldsymbol{\alpha}_i, \boldsymbol{u}_i\}$ for $\boldsymbol{\alpha}_i \in \mathbb{R}^{1 \times n}$. Fixed increment schedule ω. Constraints based on shape prior C.

for $m = 1 : MaxIteration$

 build $\mathcal{GP}$ on $\mathcal{D}_{1:m}$

 sample $\boldsymbol{x}_{m+1} = \text{argmax}_{x \in \mathcal{X}} \text{a}(\boldsymbol{x}_m \mid C, \mathcal{D})$

 evaluate $y_{m+1} = f(g_n(t) \mid \boldsymbol{x}_{m+1})$ (see eq: (1))

 obtain the current best sample $\{\boldsymbol{x}^+, y^+\}$

 compute the maximum difference d between any ...

 two coefficients in $\boldsymbol{\alpha}^+$

 if $d > 0.95$ or $m\, mod\, \omega = 0$

  update order $n = n + 1$

  re-evaluate $\{\boldsymbol{\alpha}_i\}_{i=1}^m$ for $\boldsymbol{\alpha}_i \in \mathbb{R}^{1 \times (n+1)}$

  update observations $\mathcal{D}_{1:m}$

 end if

 augment the data $\mathcal{D}_{1:m+1} = \mathcal{D}_{1:m} \cup \{\boldsymbol{x}_{m+1}, y_{m+1}\}$

end for

learning rate schedules for deep neural network optimisers as presented in experiments.

Unimodal control function: For unimodal control function we chose simplified constraints adapted from Theorem 2. For some $0 < l < n$, we define the constraints as $\mathbb{C} = \{C_v\}_{v=0}^{n-1}$ where $C_v = \alpha_{v+1} - \alpha_v \; \forall v = 0 : l$ and $C_v = \alpha_v - \alpha_{v+1} \; \forall v = l : n - 1$.

Dynamically adjusting order of Bernstein basis

As discussed earlier the intuition is illustrated in Fig 1. We compare the maximum computed derivative of the best control function with the maximum derivative possible with the same order polynomial. If it is close then we increase the polynomial order.

We first derive Lemma 3 that provides an easy way to compute the maximum of the derivative of a function based on the Bernstein coefficients.

Lemma 3. The derivative $g_n'(t)$ of the Bernstein polynomial of order n $g_n(t)$ is bounded by 'n' times the maxima of the basis polynomial with the highest coefficient.

Proof. It can be said about Bernstein polynomials that the range of the polynomial $g_n(t)$ is bounded by the values of the minimum and maximum values of the Bernstein coefficients $\{\alpha_v\}_{v=0:n}$. Then from Lemma 2 we can say that the derivative of the Bernstein polynomial $g_n'(t)$ is bounded by the values of the minimum and maximum of the values of the Bernstein coefficients $\{n(\alpha_{v+1} - \alpha_v)\}_{v=0:n-1}$. By the same token the theoretical maximum of the magnitude of the derivative is $(\alpha_{max} - \alpha_{min})$, assuming $\alpha \in [\alpha_{max}, \alpha_{min}]$. $\square$

Based on this it can be said that an n-th order Bernstein polynomial can only be used to sufficiently approximate a function if the derivate of the function is within the bounds as stated in Lemma 3. We should test whether the current function has already reached the limit or at least close to it. If it is true then we can decide that the current order is underspecified and we increment the order by one.

Next, we use the following lemma to reuse the past observations by transforming $\{\alpha_v\}_{v=0:n}$ vectors of length n to $\{\alpha_v'\}_{v=0:n+1}$ vectors of length $n + 1$.

Lemma 4. A Bernstein polynomial of order n and with Bernstein coefficients $\{\alpha_v\}_{v=0:n}$ can be represented using a Bernstein polynomial of order $n + 1$ with Bernstein coefficients $\{\alpha_v'\}_{v=0:n+1}$, such that $\alpha_v' = \frac{v}{n+1}\alpha_{v-1} + \left(1 - \frac{v}{n+1}\right)\alpha_v$. In other words, it is possible to raise the order of the Bernstein polynomial and recompute the Bernstein coefficients. *(Please refer to (Lorentz 2012))* $\square$

Lemma 3 can detect one type of signs of underspecification. To avoid underspecification altogether, we also increment the order at a regular interval until a maximum specified order is reached. The overall algorithm is presented in Algo 1. If the order of the Bernstein polynomial becomes very high then it is possible to use high-dimensional Bayesian optimisation (Rana et al. 2017; Oh, Gavves, and Welling 2018).
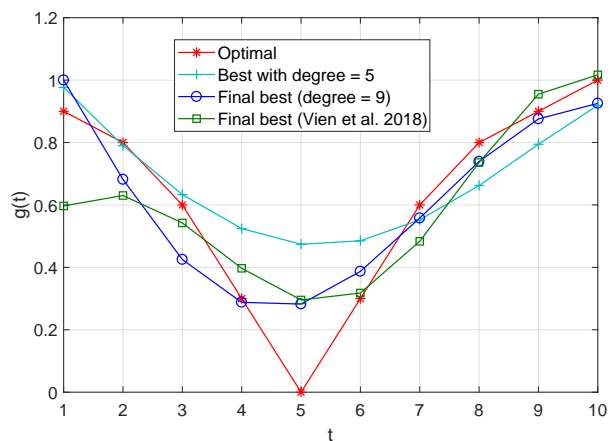
Experiments

We evaluate our proposed functional optimisation method on one synthetic and two real world experiments: optimisation of fibre yield in short polymer fibre production, and learning rate schedule optimisation for neural network training. For convenience, we refer to the proposed algorithm as BFO-SP. For all experiments, we start with a 5th order Bernstein polynomial basis, but limit the highest order to 10. The change of order is triggered due to hitting the derivative limit when it reaches 95% of the maximum derivative magnitude possible. The code will be made available upon request. Given a problem, the shape prior is an estimation of the field expert. For example, in the application of short polymer fibre design, our material science collaborators know from the fluid dynamics principles that the shape of the function needs to have an increasing flow profile for optimal production of fibres. It is this hunch that has been incorporated into the optimisation system. In a scenario where such a hunch is unavailable we would have used the general form of BFO without the shape prior constraint.

In the experiments proposed, the parameters are the coefficients of the Bernstein polynomials. Their values are optimised between 0 and 1. Then the function is reconstructed using these coefficients and Bernstein polynomials. This function is then rescaled to the true limits of the application.

In the experiments, BFO-SP is compared with the method proposed by (Vien, Zimmermann, and Toussaint 2018). We found that there is no significant difference in the computational cost of running BFO-SP against Vien et al.'s method. However, as Vien et al.'s method does not incorporate shape prior, it may take a higher number of iterations to converge.
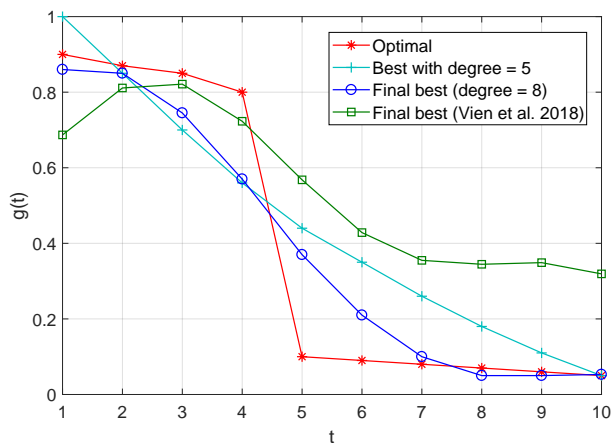
(a) With monotonicity constraint.

(b) With unimodality constraint.

Figure 4: Synthetic experiment.

Synthetic experiments

In the synthetic experiment, we construct two different 10 dimensional vectors, one which is monotonically decreasing and the other which is unimodal convex. These can be thought of schedule vectors, sampled on a fixed grid of size 10 from the optimal control functions that we would like to recover via our functional optimisation approach. For each, the utility of a trial control function is measured by first sampling the control function on the same grid and evaluating the resultant 10 dimensional vector through a Gaussian pdf function whose mean is the respective optimal vector.

We start with Bernstein polynomial of order 5 and then increase the order when either a) at the trigger as mentioned in Lemma 2, or b) at a fixed schedule of every 10 iterations. The results are shown in Fig 4 with a) showing the case for monotonicity and b) showing the case for unimodality. In the plots we show the optimal (in red), the best so far when the first order change is triggered by lemma 2 (in teal) and the best one after 20 iterations. For both the cases order changes have been triggered multiple times resulting in the final best with a considerable higher order (8 for monotonicity and 9
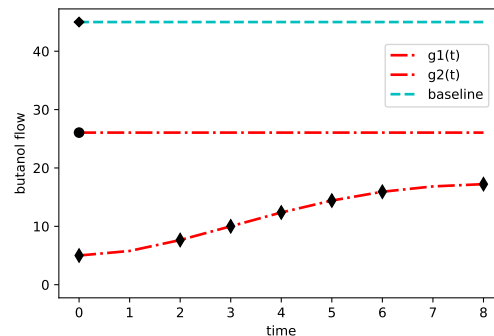

Figure 5: Short polymer fibre production: Best recirculation control functions $g(t)$ found compared to baseline. $g1(t)$ and $g2(t)$ gave highest score (9).

for unimodality). The final best ones appear much closer to the optimal one. The results are compared with the method proposed by (Vien, Zimmermann, and Toussaint 2018) (in green), where incorporating prior is not possible.

Short polymer fibre production

We optimise the butanol flow profile over the recirculation period to achieve a high quality yield of concentrated fibres. All other variables (device geometry and polymer flow) are fixed to the known best setting. The recirculation is run till a fixed time, limited by the maximum concentration achievable with the chosen polymer flow. At the end of each experiment a sample of fibre is looked at under a powerful optical microscope to inspect fibre length and diameter distribution. Quality score is given between 1-10, with 10 being the highest for fibre distribution with small variance. Experimenters had a hunch that an increasing flow profile will result in a higher quality yield, which we used as our shape prior. We used GP-UCB-PE with batch size of 6.

We have been able to reach a score of 9 out of a maximum score of 10, within 5 iterations. Fig 5 shows examples of butanol flow schedules for which high scores were recorded. Both a flat and increasing profile results in a score of 9, thus validating the experimenter's hunch. These are improvements over their current baseline with a fixed butanol flow (8). The markers on each function are time intervals at which the butanol flow is changed.

Learning rate schedule optimisation

For neural network training it has been observed that stochastic gradient descent (SGD) performs better if the learning rate is varied as a function of training duration (Bengio 2012). Specifically, it has been reported that starting with an adequately high learning rate and decreasing it over the training duration can significantly speed-up convergence. We optimise for a schedule of learning rate $f(\eta)$ for a couple of neural networks, each for the CFIR10 and MNIST datasets, while keeping the other parameters the same for all the experiments.

Dataset	Validation error			
	BFO-SP + SGD	SGD	Adam	BFO + SGD (Vien et al. 2018)
CFIR10	**18.81%**	20.30%	20.20%	22.2%
MNIST	**0.74%**	1.26%	0.86%	0.87%

Table 1: Comparison of prediction error of Bayesian optimisation of learning rate schedule against SGD and Adam with exponential decay for both CFIR10 and MNIST datasets.
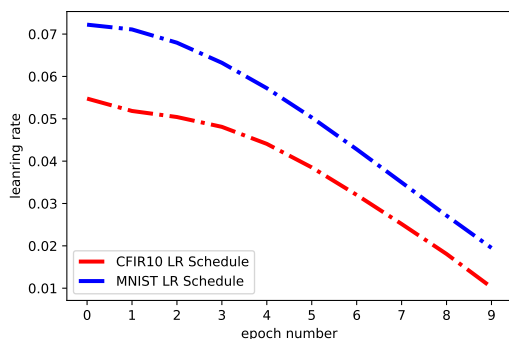

Figure 6: Learning rate schedules that resulted in the highest accuracy on CFIR10 and MNIST datasets using the BFO-SP with known prior - monotonicity constraint.
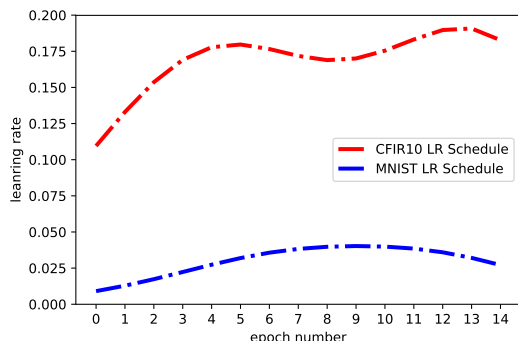

Figure 7: Learning rate schedules that resulted in the highest accuracy on CFIR10 and MNIST datasets using (Vien, Zimmermann, and Toussaint 2018)

For CFIR10 we use a network architecture that be summarised as $(Conv2D \rightarrow Dropout \rightarrow Conv2D \rightarrow Maxpooling2D) \times 3 \rightarrow Flatten \rightarrow (Dropout \rightarrow Dense) \times 3$. Whereas for MNIST the network architecture used is $(Conv2D \rightarrow Maxpooling2D \rightarrow Dropout \rightarrow Flatten \rightarrow Dense \rightarrow Dense)$. (Details about the network architecture are given in the supplementary material). Fig 6 shows the optimised learning schedules for CFIR10 and MNIST datasets using the proposed method. For Bayesian optimisation the range of learning rate was chosen between 0.2 and 0.0001. For Adam and SGD the starting learning rate used was 0.01, with 0.8 momentum for SGD and default values for hyper-parameters of Adam (Kinga and Adam 2015). We compare the performance of learning rate opti-

mised stochastic gradient descent (SGD) optimiser against a) SGD with an exponential decay and b) Adam. Tab 1 shows the performance comparison with the baselines after 20 Bayesian optimisation iteration. On both the datasets our method BFO-SP achieved higher performance than the baselines. The results are also compared with BFO by (Vien, Zimmermann, and Toussaint 2018) as shown in Fig 7. By admitting prior knowledge, our proposed method turns out to be the most efficient .

Conclusion

We present a novel approach for functional optimisation with Bayesian optimisation. We use Bernstein polynomials to model the control function and in turn optimise the Bernstein coefficients to learn the optimum function shape. Prior shape information (e.g. monotonicity, unimodality etc) is integrated and the polynomial order is dynamically adjusted during the optimisation process. We demonstrate the performance of our method by applying it for short polymer fibre production via a recirculating process, and for modelling learning rate schedule for deep learning networks. For fibre production we use monotonically increasing prior, whereas for deep networks we use monotonically decreasing shape prior. Our method outperform baselines in all the cases.

Currently, the method proposed is built to work for unimodal and increasing or decreasing shape priors. A natural progression is to extend this for other complex shape priors (eg. multimodal) as well. It would also be interesting to consider multi-variate Bernstein polynomial when multiple variables may form inter-related function of time. An integration of stable Bayesian optimisation (Dai Nguyen et al. 2017) will be important for industrial applications. It would be interesting to analyse how variation in the final schedule dictates variation in the coefficients of the Bernstein polynomials. The proposed method can be applied in a variety of domains including plastic recycling industry, drug manufacturing and even in alloy heat treatment optimisation where the usual process of constant temperature heat treatment can be replaced by a variable temperature treatment to achieve faster result with possibly lower energy usage.

Acknowledgement

References

Bengio, Y. 2012. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*. Springer. 437–478.

Bernstein, S. 1912. Demonstration of a theorem of weierstrass based on the calculus of probabilities. *Communications of the Kharkov Mathematical Society* Volume XIII(1912/13):1–2.

Brochu, E.; Cora, V. M.; and de Freitas, N. 2010. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv:1012.2599*.

Chang, I.-S.; Chien, L.-C.; Hsiung, C. A.; Wen, C.-C.; and Wu, Y.-J. 2007. Shape restricted regression with random bernstein polynomials. *Lecture Notes-Monograph Series* 187–202.

Contal, E.; Buffoni, D.; Robicquet, A.; and Vayatis, N. 2013. Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 225–240. Springer.

Dai Nguyen, T.; Gupta, S.; Rana, S.; and Venkatesh, S. 2017. Stable Bayesian optimization. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 578–591. Springer.

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

Feng, L.; Song, Y.; Zhai, J.; Liu, B.; Xu, J.; Jiang, L.; and Zhu, D. 2003. Creation of a superhydrophobic surface from an amphiphilic polymer. *Angewandte Chemie International Edition* 42(7):800–802.

Gupta, S.; Shilton, A.; Rana, S.; and Venkatesh, S. 2018. Exploiting strategy-space diversity for batch bayesian optimization. In Storkey, A., and Perez-Cruz, F., eds., *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, 538–547. Playa Blanca, Lanzarote, Canary Islands: PMLR.

Hennig, P., and Schuler, C. J. 2012. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research* 13:1809–1837.

Hernández-Lobato, J. M.; Hoffman, M. W.; and Ghahramani, Z. 2014. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in neural information processing systems*, 918–926.

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

Kinga, D., and Adam, J. B. 2015. A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, volume 5.

Kulfan, B., and Bussoletti, J. 2006. "Fundamental" parameteric geometry representations for aircraft component shapes. In *11th AIAA/ISSMO multidisciplinary analysis and optimization conference*, 6948.

Kushner, H. J. 1964. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering* 86(1):97–106.

Li, C.; de Celis Leal, D. R.; Rana, S.; Gupta, S.; Sutti, A.; Greenhill, S.; Slezak, T.; Height, M.; and Venkatesh, S. 2017. Rapid bayesian optimisation for synthesis of short polymer fiber materials. *Scientific reports* 7(1):5683.

Li, C.; Rana, S.; Gupta, S.; Nguyen, V.; Venkatesh, S.; Sutti, A.; Rubin, D.; Slezak, T.; Height, M.; Mohammed, M.; and Gibson, I. 2018. Accelerating experimental design by incorporating experimenter hunches. In *Data Mining (ICDM), 2018 IEEE International Conference on*, 1–8. IEEE.

Lorentz, G. G. 2012. *Bernstein polynomials*. American Mathematical Soc.

Ma, M.; Hill, R. M.; and Rutledge, G. C. 2008. A review of recent results on superhydrophobic materials based on micro-and nanofibers. *Journal of Adhesion Science and Technology* 22(15):1799–1817.

Mockus, J.; Tiesis, V.; and Zilinskas, A. 1978. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization* 2:117–129.

Mockus, J. 1994. Application of bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization* 4(4):347–365.

Oh, C.; Gavves, E.; and Welling, M. 2018. BOCK : Bayesian optimization with cylindrical kernels. In Dy, J., and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 3868–3877. PMLR.

Rana, S.; Li, C.; Gupta, S.; Nguyen, V.; and Venkatesh, S. 2017. High dimensional Bayesian optimization with elastic Gaussian process. In Precup, D., and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 2883–2891. PMLR.

Rasmussen, C. E. 2006. Gaussian processes for machine learning.

Samareh, J. A. 2001. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA journal* 39(5):877–884.

Srinivas, N.; Krause, A.; Kakade, S.; and Seeger, M. W. 2010. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 1015–1022.

Vellanki, P.; Rana, S.; Gupta, S.; Rubin, D.; Sutti, A.; Dorin, T.; Height, M.; Sanders, P.; and Venkatesh, S. 2017. Process-constrained batch bayesian optimisation. In *Advances in Neural Information Processing Systems*, 3417–3426.

Vien, N. A.; Zimmermann, H.; and Toussaint, M. 2018. Bayesian functional optimization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18).*, 4171–4178.