# Trading-off Data Fit and Complexity in Training Gaussian Processes with Multiple Kernels

Tinkle Chugh[1⋆], Alma Rahat[2], and Pramudita Satria Palar[3]

[1] University of Exeter, UK T.Chugh@exeter.ac.uk
[2] University of Plymouth, UK Alma.Rahat@plymouth.ac.uk
[3] Badung Institute of Technology, Indonesia pramsp@ftmd.itb.ac.id

**Abstract.** Gaussian processes (GPs) belong to a class of probabilistic techniques that have been successfully used in different domains of machine learning and optimization. They are popular because they provide uncertainties in predictions, which sets them apart from other modelling methods providing only point predictions. The uncertainty is particularly useful for decision making as we can gauge how reliable a prediction is. One of the fundamental challenges in using GPs is that the efficacy of a model is conferred by selecting an appropriate kernel and the associated hyperparameter values for a given problem. Furthermore, the training of GPs, that is optimizing the hyperparameters using a data set is traditionally performed using a cost function that is a weighted sum of data fit and model complexity, and the underlying trade-off is completely ignored. Addressing these challenges and shortcomings, in this article, we propose the following automated training scheme. Firstly, we use a weighted product of multiple kernels with a view to relieve the users from choosing an appropriate kernel for the problem at hand without any domain specific knowledge. Secondly, for the first time, we modify GP training by using a multi-objective optimizer to tune the hyperparameters and weights of multiple kernels and extract an approximation of the complete trade-off front between data-fit and model complexity. We then propose to use a novel solution selection strategy based on mean standardized log loss (MSLL) to select a solution from the estimated trade-off front and finalise training of a GP model. The results on three data sets and comparison with the standard approach clearly show the potential benefit of the proposed approach of using multi-objective optimization with multiple kernels.

**Keywords:** Machine learning · Kriging · Bayesian optimization · multi-objective optimization · model selection

## 1 Introduction

Gaussian processes (GPs) have been widely used in machine learning and optimization communities. Some of the problems where GPs have gained their

⋆ corresponding author

popularity are non-linear regression (also known as Kriging in geostatistics), classification [23] and Bayesian optimization [24]. The main advantage of using GPs is that they provide a predictive distribution instead of point predictions as in other models like neural networks and support vector regression. This uncertainty can be used in making the decisions [22, 20] and in selecting samples by optimizing an acquisition function in Bayesian optimization [15, 13].

Despite their wide applicability, little attention has been paid to the problem of selecting kernels and the hyperparameters. As mentioned in [23], multiple choices exist and it is not straightforward to select a kernel and its hyperparameters. It often requires prior knowledge about the underlying function that we are trying to model. To select the hyperparameters, the traditional approach is to maximize the marginal likelihood for a given kernel. A characteristic of this likelihood function is that it tries to balance between data-fit and complexity. For instance, the data-fit decreases monotonically with the length scale resulting in increasing the complexity of the model. A simple illustration of the model fit and complexity by varying the length scale when using a Gaussian kernel is shown in Fig. 1.

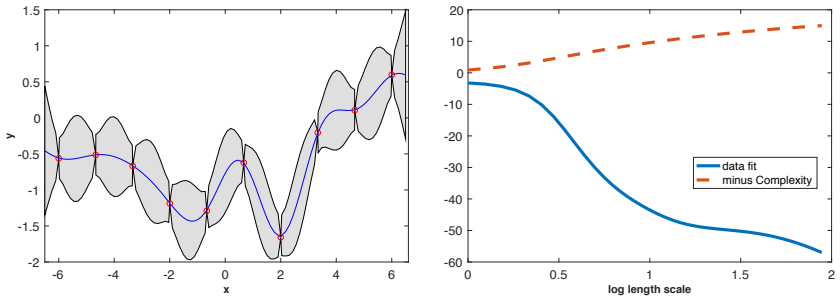



Fig. 1: Left plot: Data generated with a GP realization of length scale = 1, signal variance = 1 and noise variance = 0.1, Right plot: Data fit and minus complexity with length scale of the data set in the left plot.

As can be seen, the data fit decreases with the increase in complexity of models with different length scales. In selecting a kernel, several options like Gaussian (or RBF), exponential, linear, matern 5/2, matern 3/2 and periodic exist. In the literature, some studies are devoted to the concern of selecting a kernel. For example, in [17], a genetic programming approach was applied to find a composite kernel and in [8], different combinations like sum and product of kernels were used. In [21], a weighted sum of kernels was used in training of GPs. Recently, in [4], different kernels were studied in the context of Bayesian optimization and different correlations were observed between a kernel and other elements in Bayesian optimization.

In applying multi-objective optimization in machine learning, some studies exist in building models like neural networks [9, 12] and decision trees [10]. Approaches in these studies considered different objectives like bias and variance, model fit and complexity to find the number of hidden layers and number of nodes in neural networks and number and depth of trees in decision trees.

However, to the best of our knowledge, no study exist in using multi-objective optimization in GPs, despite the fact that the inherent property of the likelihood function when building the model is to balance between model fit and complexity. Therefore, in this work, by optimizing two objectives, maximizing model fit and minimizing complexity, we find the optimal hyperparameter values and weights for different kernels.

We use a non-dominated sorting genetic algorithm, NSGA-II [6] to find approximated Pareto optimal solutions, where each solution on the Pareto front represents a model with different accuracy, complexity, hyperparameter values and weights to different kernels. It should be noted that using a multi-objective optimization algorithm does not increase the computational complexity in optimizing the hyperparameter values as both the standard and the proposed approach aim to solve an optimization problem. Also, one is free to choose another multi-objective optimizer.

We then use MSLL [23] to select a model from the approximated Pareto front. The results of the proposed approach using multi-kernel and multi-objective (MKL-MO) is compared then with standard (with single kernel and maximizing the marginal likelihood) and single-kernel and multi-objective (SKL-MO) approaches. This comparison shows the effect of using multi-objective optimization and multiple kernels. We can summarize the main contributions of this paper as follows:

- We use weighted product of multiple kernels to relieve users from selecting one or more kernels for the problem at hand.
- We use multi-objective optimization to estimate the trade-off between data fit and complexity.
- We utilize the MSLL performance metric to select a model from the approximated Pareto front, and derive predictions from a GP model.

The rest of the article is structured as follows. In Section 2, we provide a brief description of GPs and different kernels used in this work. In section 3, we explain the proposed approach of using multi-objective optimization with single and multiple kernels. We conduct experiments and discuss the results in Section 4. Finally, we conclude and mention the future research directions in Section 5.

## 2 Gaussian Processes for Regression

A typical regression task is to model the relationship between some independent variables (or features) and a dependent variable. Consider a data set of M observations $\mathcal{D} = \{(\mathbf{x}_m, f(\mathbf{x}_m)) \mid m = 1, \ldots, M\}$, where $\mathbf{x} \in \mathbb{R}^n$ is a $n$-dimensional feature vector, and a function $f : \mathbf{x} \to \mathbb{R}$ produces a response (i.e. the dependent variable) based on $\mathbf{x}$. In the regression task, we are therefore interested in making predictions about $f_i = f(\mathbf{x}_i)$ for any arbitrary feature vector $\mathbf{x}_i$ given the data set $\mathcal{D}$.

As mentioned, GPs have grown in popularity for non-linear regression tasks in recent years. This is primarily due to its efficacy in providing a posterior

probability density indicating how confident the prediction is. Essentially, a GP is a collection of random variables such that any finite number of these have a joint Gaussian distribution [23]. This means that the posterior predictive density of the function $f(\mathbf{x})$ given some data set $\mathcal{D}$ and a feature vector $\mathbf{x}$ is normally distributed:

$$p(f \mid \mathbf{x}, \mathcal{D}, \boldsymbol{\theta}) = \mathcal{N}(f \mid \mu(\mathbf{x}), \sigma^2(\mathbf{x})), \tag{1}$$

where the mean and the variance of the prediction are given by:

$$\mu(\mathbf{x}) = \boldsymbol{\kappa}(\mathbf{x}, X, \boldsymbol{\theta})(K + \sigma_e^2 I)^{-1}\mathbf{f} \tag{2}$$

$$\sigma^2(\mathbf{x}) = \kappa(\mathbf{x}, \mathbf{x}, \boldsymbol{\theta}) - \boldsymbol{\kappa}(\mathbf{x}, X, \boldsymbol{\theta})^\top (K + \sigma_e^2 I)^{-1}\kappa(X, \mathbf{x}, \boldsymbol{\theta}) \tag{3}$$

Here $X \in \mathbb{R}^{M \times n}$ is the matrix of observed feature vectors and $\mathbf{f} \in \mathbb{R}^M$ is the corresponding vector of the responses $\mathbf{f} = (f_1, \ldots, f_M)^\top$; thus $D = \{(X, \mathbf{f})\}$. The covariance matrix $K \in \mathbb{R}^{M \times M}$ represents the covariance function $\kappa(\mathbf{x}', \mathbf{x}'', \boldsymbol{\theta})$ evaluated for each pair of observations $\mathbf{x}', \mathbf{x}'' \in X$ and $\boldsymbol{\kappa}(\mathbf{x}, X, \boldsymbol{\theta}) \in \mathbb{R}^M$ is the vector of covariances between an arbitrary $\mathbf{x}$ and each of the observations. The kernel hyperparameter vector $\boldsymbol{\theta} \in \mathbb{R}^k$ is a vector of parameters that controls the shape of the kernel. $\sigma_e^2$ is a homoscadastic Gaussian noise variance that encapsulates the potential error which may occur while measuring the responses $\mathbf{f}$. The overall hyperparameter vector is therefore $\mathbf{t} = (\boldsymbol{\theta}, \sigma_e^2)^\top$.

### 2.1 Kernels

A kernel (or covariance function) is usually defined as $\kappa(\mathbf{x}_i, \mathbf{x}_j, \boldsymbol{\theta})$, where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two feature vectors, and $\boldsymbol{\theta} \in \mathbb{R}^k$ is a vector of $k$ hyperparameters. In essence, the kernel captures the intuition that two feature vectors that are spatially closer should have similar response, and this relationship is defined by the hyperparameters $\boldsymbol{\theta}$. A kernel with its hyperparameters thus imposes a reproducing kernel Hilbert space for all possible functions that may be represented given data set $\mathcal{D}$.

To describe the relationship between responses $f_i$ and $f_j$ for a pair of feature vectors $x_i$ and $x_j$, typically we consider a distance measure in the feature space $r^2 = \sum_{v=1}^{n} \frac{(x_i[v] - x_j[v])^2}{l[v]^2}$ with a hyperparameter $l[v]$ that determines the length scale in the $v$th dimension. Here, $l[v]$ scales the $v$th dimension and thus controls the importance of the respective dimension in determining the response. In addition, an amplitude hyperparameter $\sigma_f$ that controls how much the function response may vary with distance in the feature space. Hence, the hyperparameter vector may be constructed as $\boldsymbol{\theta} = (\sigma_f, l[1], \ldots, l[n])^\top$. With this, we can define the following five popular kernels used in this paper [23].

**Radial basis function or Gaussian.** This is the most popular kernel with infinitely many derivatives, and therefore can produce very smooth function realisations. It may be expressed as:

$$\kappa(x_i, x_j, \boldsymbol{\theta}_r) = \sigma_f^2 \exp\left(-\frac{r^2}{2}\right). \tag{4}$$

**Exponential.** Closely related to the Gaussian kernel, but can produce rougher function realisations. It can be defined as:

$$\kappa(x_i, x_j, \boldsymbol{\theta}_e) = \sigma_f^2 \exp(-r). \tag{5}$$

**Matern.** A class of functions defined by:

$$\kappa(x_i, x_j, \boldsymbol{\theta}_\nu) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu} r)^{\nu+1} \beta_\nu, \tag{6}$$

where, $\nu$ is a smoothness parameter that is set to either $\frac{3}{2}$ for once differentiable functions or $\frac{5}{2}$ for twice differentiable functions in this paper and $\beta_\nu$ is the modified Bessel function. In the above kernels, the hyperparameter vector have the same attributes, i.e. $\boldsymbol{\theta}_r = \boldsymbol{\theta}_e = \boldsymbol{\theta}_m = (\sigma_f, l[1], \ldots, l[n])^\top$, and the number of hyperparameters is $k = n + 1$.

**Periodic.** To capture periodicity that may occur in a response, we may also consider the following periodic kernel [19]:

$$\kappa(x_i, x_j, \boldsymbol{\theta}_p) = \sigma_f^2 \exp\left[ -\frac{1}{2} \sum_{v=1}^{n} \left( \frac{\sin\left(\frac{\pi}{t[v]}(x_i[v] - x_j[v])\right)}{l[v]} \right)^2 \right], \tag{7}$$

where the additional hyperparameter $t[v]$ represents the distance between repetitions in the $v$th dimension. Thus, in this case, the hyperparameter vector is $\boldsymbol{\theta}_p = (\sigma_f, l[1], \ldots, l[n], t[1], \ldots, t[n])^\top$, and hence the number of hyperparameters is $k = 2n + 1$.

Clearly, the kernels above (and many others in the literature) can represent functions with varying degree of smoothness and periodicity. Nonetheless, a specific kernel on its own may not be appropriate for modelling all responses; no free lunch theorem applies here [27]. That is why choosing an appropriate kernel is an important step in training a GP model, and often requires domain specific knowledge.

## 3 Multi-objective Training of Gaussian Processes

As mentioned in the introduction, improving the data fit increases the model complexity, i.e. most complex model can fit the given data best. Therefore, data fit and complexity are conflicting objectives. This is because a very complex model may not generalise the training data well, and consequently perform poorly on unseen data set. We, therefore, want to control complexity such that it avoids over fitting without compromising performance on both training and validation data set in GP training.

Typically, training a GP model constitutes estimating the overall hyperparameter vector $\mathbf{t} = (\boldsymbol{\theta}, \sigma_e^2)^\top$ that brings together the kernel hyperparameters $\boldsymbol{\theta}$ and the Gaussian error noise variance $\sigma_e^2$ by maximising the marginal likelihood of the data:

$$\log p(\mathcal{D} \mid \mathbf{t}) = -\frac{1}{2}\mathbf{f}^\top (K + \sigma_e^2 I)^{-1}\mathbf{f} - \frac{1}{2}\log|K + \sigma_e^2 I| - \frac{M}{2}\log(2\pi) \tag{8}$$

$$= g_d(\mathcal{D}, \mathbf{t}) - g_c(\mathcal{D}, \mathbf{t}) + C, \tag{9}$$

where, $I \in \mathbb{R}^{M \times M}$ is an identity matrix. The first term is representing the *data fit* $g_d(\mathcal{D}, \mathbf{t})$ and the second term is representing *model complexity* $g_c(\mathcal{D}, \mathbf{t})$ [7, 23]. The last term is a *normalisation constant* $C$.

Clearly, the desire to strike a balance between data fit and complexity is evident from (8): when data fit $g_d(\mathcal{D}, \mathbf{t})$ is maximised and the model complexity $g_c(\mathcal{D}, \mathbf{t})$ is minimised simultaneously, it results in maximising the marginal likelihood. Intuitively, this means we improve data fit as much as possible while penalising the complexity at the same time. Interestingly, despite the recognition of the obvious conflict between the objectives (e.g. [23]), a multi-objective optimization approach has never been adopted. Instead, the training of a GP model is posed as a single objective optimization problem for locating suitable hyperparameters and error variance:

$$\mathbf{t}^* = \underset{\mathbf{t}}{\operatorname{argmax}} \quad \log p(\mathcal{D} \mid \mathbf{t}). \tag{10}$$

The estimated optimal solution for $\mathbf{t}^* = (\boldsymbol{\theta}^*, \sigma_e^{2*})^\top$ is then used in (2) and (3) to produce the posterior predictive distribution. In this work, we propose to deal with the conflicting objectives as a multi-objective optimization problem (MOP) of maximising both data fit and complexity penalty simultaneously:

$$\max_{\mathbf{t}} \quad g_d(\mathcal{D}, \mathbf{t}) = -\frac{1}{2}\mathbf{f}^\top (K + \sigma_e^2 I)^{-1}\mathbf{f}, \tag{11}$$

$$\min_{\mathbf{t}} \quad g_c(\mathcal{D}, \mathbf{t}) = \frac{1}{2}\log |K + \sigma_e^2 I|. \tag{12}$$

Generally, there is not a unique solution to this multi-objective problem, but a range of solutions $\mathbf{t}$ that trade-off between the data fit and complexity. The trade-off relationship is characterised by the notion of dominance [5]. A solution $\mathbf{t}$ is said to (weakly) dominate another shape $\mathbf{t}'$, denoted as $\mathbf{t} \prec \mathbf{t}'$, iff,

$$\begin{aligned} &g_d(\mathcal{D}, \mathbf{t}) > g_d(\mathbf{t}') \text{ and } g_c(\mathbf{t}) \leq g_c(\mathbf{t}') \\ \text{or } &g_d(\mathcal{D}, \mathbf{t}) \geq g_d(\mathbf{t}') \text{ and } g_c(\mathbf{t}) < g_c(\mathbf{t}'). \end{aligned} \tag{13}$$

The set of solutions that provide an optimal trade-off between the objectives is referred to as the Pareto set:

$$\mathcal{P} = \{\mathbf{t} \mid \mathbf{t}' \not\prec \mathbf{t}\ \forall \mathbf{t}, \mathbf{t}' \in \tau \wedge \mathbf{t} \neq \mathbf{t}'\}, \tag{14}$$

where $\tau$ is the space that consists of all permissible hyperparameter vectors $\mathbf{t}$. The image of the Pareto set $\mathcal{P}$ in the objective space is known as the Pareto front $\mathcal{F}$. It may not be possible to locate the exact Pareto set within a practical time limit, even if the objective functions were computationally cheap. Therefore, the goal of an effective optimization approach is to generate a good approximation of the Pareto set, denoted as $\mathcal{P}^* \subseteq \tau$, and the associated Pareto front, denoted as $\mathcal{F}^*$. In this paper, we used the popular NSGA-II optimizer to approximate the optimal trade-off front (and one is free to chose another multi-objective optimizer).

Clearly the maximum likelihood solution $\mathbf{t}^*$ in (10) is achieved by optimizing a weighted sum of the MOP in equations (11) and (12). In this case, both

objectives are equally weighted, i.e. they both have the same importance. It is well-known that the optimal solution of a weighted sum must reside in the Pareto set [5], and therefore $\mathbf{t}^* \in \mathcal{P}$. However, intuitively there is no reason to believe that for all problems data fit and model complexity are equally important, and this specific set of weights will outperform others. This is precisely why, in this paper, we attempt to estimate the optimal trade-off front, and decide on which solution to select based on the estimated performance.

### 3.1 Training with Multiple Kernels

Thus far, we introduced a single kernel, associated hyperparameters, and how to optimize these in a multi-objective manner to train a GP model. In this section, we present how we can combine multiple kernels so that a user does not have to select a kernel for a given problem.

There are various avenues to combine multiple kernels, for instance weighted sum or weighted product of kernels [8]. In this paper, we use a composite kernel consisting of $L$ kernels as a weighted product [25]:

$$\kappa_c(\mathbf{x}^i, \mathbf{x}^j, \boldsymbol{\Theta}) = \prod_{l=1}^{L} \omega_l \kappa_l(\mathbf{x}^i, \mathbf{x}^j, \boldsymbol{\theta}_l), \tag{15}$$

where $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_L)^\top$ is a weight vector with $\sum_l \omega_l = 1$, and composite hyperparameter vector $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L)^\top$. In this case, the overall hyperparameter vector becomes $\mathbf{t} = (\boldsymbol{\Theta}, \sigma_e^2, \boldsymbol{\omega})^\top$. With this, we now search over all possible $\mathbf{t}$ in equations (11) and (12). Note that it is straightforward to compute the covariance matrix $K$ using the kernel defined in equation (15).

In this paper, we used five kernels as described in Section 2.1. Thus we have $L = 5$, $|\boldsymbol{\omega}| = 5$ and $\boldsymbol{\Theta} = (\boldsymbol{\theta}_r, \boldsymbol{\theta}_e, \boldsymbol{\theta}_{m32}, \boldsymbol{\theta}_{m52}, \boldsymbol{\theta}_p)^\top$ with $|\boldsymbol{\Theta}| = 6n + 5$. Therefore the number of overall hyperparameters that we optimize is: $|\mathbf{t}| = 6n + 11$ (including parameters for noise variance).

### 3.2 Constructing a Model from The Estimated Pareto Front

As discussed, solving the MOP will result in a range of solutions for the overall hyperparameter vector $\mathbf{t} = (\boldsymbol{\theta}, \sigma_e^2)^\top$ for SKL-MO and $\mathbf{t} = (\boldsymbol{\Theta}, \sigma_e^2, \boldsymbol{\omega})^\top$ for MKL-MO, each of which is a potential GP model with a distinct posterior predictive distribution for $f(\cdot)$. It is, therefore, required to select one solution or combine multiple solutions to produce a single GP model for predictions. Different approaches may be adopted for this purpose: using ensemble of models [18, 11], selecting a model representing a knee point (or maximum trade-off) on the Pareto front [2] and using Bayesian information criterion [3].

In this paper, our goal is to shed light on the efficacy of SKL-MO and MKL-MO in comparison to the standard approach. To do so, intuitively, we want to estimate how good a model may be given a solution from the $\mathcal{P}^*$ and the data set.

| | Number of variables | Size of data set |
|---|---|---|
| Mauna $CO_2$ | 1 | 108 |
| Concrete | 8 | 100 |
| Sarcos | 21 | 100 |

Table 1: Number of variables and size of different data used

Hence, we use an performance metric called mean standardized log loss (MSLL) [23] which is defined as:

$$-\log p(\mu(x) \mid \mathcal{D}, x, \mathbf{t}) = \frac{1}{2}\log(2\pi\sigma^2(x)) + \frac{(\mu - f(x))^2}{2\sigma^2(x)} \tag{16}$$

The main benefit of using MSLL is that it is not sensitive to overall scale of the response variable values and considers both predicted values and their standard deviations.

In our approach, we split the data set into ten-folds leaving randomly chosen 90% for training and 10% for validation in each fold. For each fold, we perform multi-objective optimization to approximate the Pareto front, and select a solution with minimum MSLL value on the test set. This, of course, do not give us an idea on how to construct a model when we want to train on 100% of the data, but clearly shows which approach may yield better generalisation results. We expect to investigate this further in future.

## 4 Numerical experiments

This section provides the results and discussion of numerical experiments conducted on three popular data sets. First data set used was Mauna Loa monthly mean of $CO_2$ concentrations (in parts per million by volume (ppmv)) from 2010-2018 [14] [4] and is shown in Figure 2. The second data set used was the concrete data set [28] in which strength of the concrete depends on the concentration of cement, furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate used and the age of the concrete[5]. The third data set used was sarcos data set [26], in which 21 dimensions representing positions, velocities and accelerations map to the torque of the robot arm [6]. In this work, we used 100 uniformly distributed set of points in concrete and sarcos data sets. A summary of different data bases used with number of variables and size is provided in Table 1.

To show the potential of using multi-objective optimization and multiple kernels, we compared the proposed multi-kernel and multi-objective (MKL-MO) approach with standard and with single-kernel and multi-objective approach (SKL-MO) approaches. In both standard and SKL-MO approaches, we used the Gaussian kernel. Further, we used 10-fold cross validation and calculated the root mean square (rmse) values. In doing multi-objective optimization in SKL-MO and MKL-MO, we used NSGA-II algorithm. In using NSGA-II, we kept an

[4] available from: https://www.esrl.noaa.gov/gmd/ccgg/trends/data.html

[5] available from http://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength

[6] available from http://www.gaussianprocess.org/gpml/data/

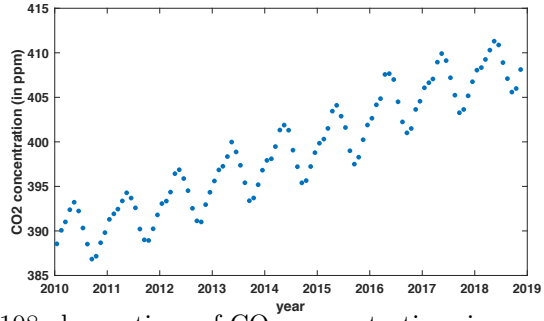



Fig. 2: The 108 observations of $CO_2$ concentrations in years 2010-2018

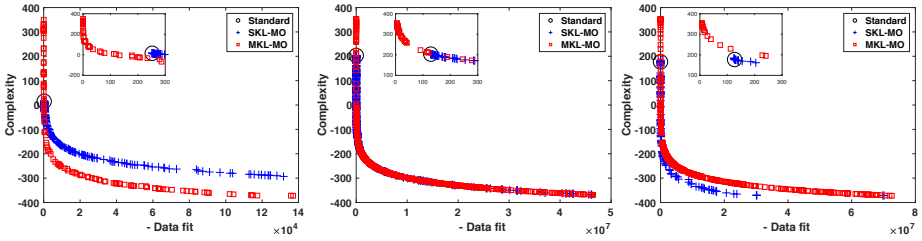
archive to store all solutions and nondominated solutions from the archive were used as the final solutions. The parameter values of different elements in NSGA-II were: population size: 50, number of generations: 50, crossover: simulated binary with 0.8 probability, mutation: polynomial with 1/number of variables.

The approximated Pareto optimal solutions (of a random fold among 10 folds) representing negative of data fit and complexity of different approaches on three data sets are shown in Figure 3. Each solution on the Pareto front has its own set of parameters i.e. kernel parameters, noise variance and weights. In solving standard approach, only one solution could be obtained which is represented with a circle in the figures. As both SKL-MO and MKL-MO solves a MOP, it is not surprising to get many solutions. However, one key observation from the results is that a much better distribution (or diversity) of solutions was obtained in MKL-MO when compared to SKL-MO approach. This is because the multi-objective optimization algorithm was able to explore in diverse regions with the help of multiple kernels. Finding a good distribution of solutions is one of the main features when solving a MOP and the proposed MKL-MO approach was able to achieve it.

Next, we selected a model with the least MSLL values from the $\mathcal{P}^*$ for SKL-MO and MKL-MO approaches, and calculated the rmse values. The box plots of the rmse values of all three different approaches on different data sets are shown in Figure 4, and the corresponding MSLL values are shown in Figure 5. To test whether one of the methods statistically significantly wins in all folds and problems, we performed Mann-Whitney-U test [16] as the folds were independently chosen. We also adjusted for multiple comparisons using Bonferroni correction [1]. The significance level was set to $\rho = 0.05$. The tests revealed that MKL-MO performed better than its competitors in concrete (rmse), mauna (rmse, MSLL), sarcos (rmse). Otherwise, we found no statistically significant results at the desired level. Visually, it is clear that MKL-MO outperforms other methods.

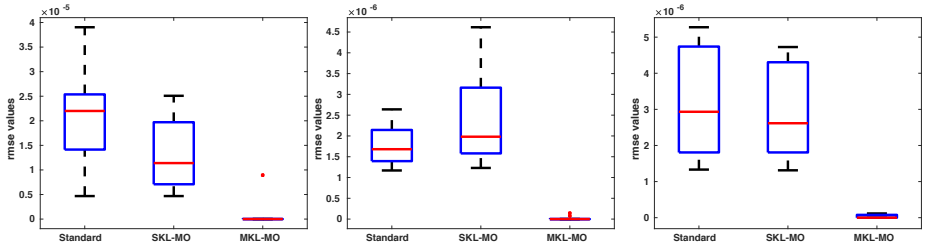
## 5 Conclusions

In this article, we focused on multi-objective optimization of two conflicting objectives, maximizing data fit and minimizing complexity when training a GP



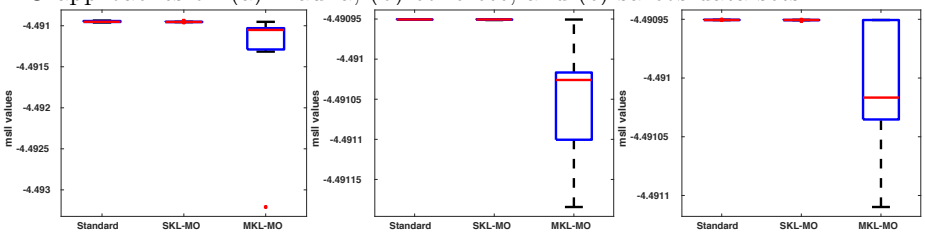

(a) (b) (c)

Fig. 3: Approximated Pareto optimal solutions representing -data fit and complexity of standard, SKL-MO and MKL-MO approaches on (a) Mauna, (b) concrete, and (c) sarcos data sets). A small part of the plot with low -data fit values is zoomed




(a) (b) (c)

Fig. 4: Root mean square error (rmse) values of standard, SKL-MO and MKL-MO approaches on (a) Mauna, (b) concrete, and (c) sarcos data sets




(a) (b) (c)

Fig. 5: Mean standardized log loss (MSLL) values of standard, SKL-MO and MKL-MO approaches on (a) Mauna, (b) concrete, and (c) sarcos data sets

model. In addition, we combined the multi-objective approach with multiple kernels to handle the challenges of selecting a particular kernel. For this, we used the weighted product of kernels where weights and the kernel parameters were calculated during the multi-objective optimization. The mean standardized log loss values were used in selecting a model from the approximated Pareto front after solving multi-objective optimization problem. The results on three different data sets and comparison with standard and single kernel-multi-objective approach clearly showed the potential of the proposed multi-kernel multi-objective approach. In future, we will investigate more methods of combining kernels and selecting a solution from the estimated Pareto front for a diverse set of data sets from practical applications with varying sizes.

## Acknowledgments

This research was partially supported by the Natural Environment Research Council, UK [grant number NE/P017436/1].

## References

1. Bender, R., Lange, S.: Adjusting for multiple testing: when and how? Journal of Clinical Epidemiology **54**(4), 343 – 349 (2001)
2. Branke, J., Deb, K., Dierolf, H., Osswald, M.: Finding knees in multi-objective optimization. In: Parallel Problem Solving from Nature - PPSN VIII. pp. 722–731. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
3. Burnham, K.P., Anderson, D.R.: Multimodel inference: Understanding AIC and BIC in model selection. Sociological Methods & Research **33**(2), 261–304 (2004)
4. Chugh, T., Rahat, A., Volz, V., Zaefferer, M.: Towards Better Integration of Surrogate Models and Optimizers, pp. 137–163. Springer International Publishing, Cham (2020)
5. Coello Coello, C.A., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer (2007)
6. Deb, K., Prarap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**, 182–197 (2002)
7. Duvenaud, D.: Automatic model construction with Gaussian processes. Ph.D. thesis, University of Cambridge (2014)
8. Duvenaud, D., Lloyd, J., Grosse, R., Tenenbaum, J., Zoubin, G.: Structure discovery in nonparametric regression through compositional kernel search. In: Proceedings of the 30th International Conference on Machine Learning. vol. 28, pp. 1166–1174. PMLR, Atlanta, Georgia, USA (2013)
9. Fieldsend, J.E., Singh, S.: Pareto evolutionary neural networks. IEEE Transactions on Neural Networks **16**(2), 338–354 (2005)
10. Fieldsend, J.E.: Optimizing Decision Trees Using Multi-objective Particle Swarm Optimization, pp. 93–114. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
11. Friese, M., Bartz-Beielstein, T., Bck, T., Naujoks, B., Emmerich, M.: Weighted ensembles in model-based global optimization. In: AIP Conference. vol. 2070, p. 020003 (2019)

12. Jin, Y., Sendhoff, B.: Pareto-based multiobjective machine learning: An overview and case studies. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **38**, 397–415 (2008)
13. Jones, D., Schonlau, M., Welch, W.: Efficient global optimization of expensive black-box functions. Journal of Global Optimization **13**, 455–492 (1998)
14. Keeling, C.D., Whorf, T.P.: Atmospheric CO2 records from sites in the sio air sampling network. in trends: A compendium of data on global change. carbon dioxide information analysis center, Oak Ridge National Laboratory, U.S.A. (2004)
15. Knowles, J.: ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation **10**, 50–66 (2006)
16. Knowles, J.D., Theile, L., Zitzler, E.: A tutorial on the performance assesment of stochastic multiobjective optimizers. Tech. Rep. TIK214, Computer Engineering and Networks Laboratory, ETH Zurich, Zurich, Switzerland (February 2006)
17. Kronberger, G., Kommenda, M.: Evolution of covariance functions for gaussian process regression using genetic programming. In: International Conference on Computer Aided Systems Theory. pp. 308–315. Springer (2013)
18. Lei, Y., Yang, H.: A gaussian process ensemble modeling method based on boosting algorithm. In: Proceedings of the 32nd Chinese Control Conference. pp. 1704–1707 (2013)
19. MacKay, D.J.: Introduction to gaussian processes. NATO ASI Series F Computer and Systems Sciences **168**, 133–166 (1998)
20. Mazumdar, A., Chugh, T., Miettinen, K., López-Ibáñez, M.: On dealing with uncertainties from kriging models in offline data-driven evolutionary multiobjective optimization. In: Evolutionary Multi-Criterion Optimization. pp. 463–474. Springer International Publishing, Cham (2019)
21. Palar, P.S., Shimoyama, K.: Kriging with composite kernel learning for surrogate modeling in computer experiments. In: AIAA Scitech 2019 Forum. pp. 2019–2209 (2019)
22. Rahat, A.A., Wang, C., Everson, R.M., Fieldsend, J.E.: Data-driven multiobjective optimisation of coal-fired boiler combustion systems. Applied Energy **229**, 446 – 458 (2018)
23. Rasmussen, C.E., Williams, C.K.I.: Gaussian processes for machine learning. The MIT Press (2006)
24. Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., de Freitas, N.: Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE **104**, 148–175 (2016)
25. Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B.: Large scale multiple kernel learning. Journal of Machine Learning Research **7**, 1531–1565 (2006)
26. Vijayakumar, S., Schaal, S.: Locally weighted projection regression: An O(n) algorithm for incremental real time learning in high dimensional space. In: in Proceedings of the Seventeenth International Conference on Machine Learning (ICML). pp. 1079–1086 (2000)
27. Wolpert, D.H.: The lack of a priori distinctions between learning algorithms. Neural computation **8**(7), 1341–1390 (1996)
28. Yeh, I.C.: Modeling of strength of high-performance concrete using artificial neural networks. Cement and Concrete Research **28**, 1797 – 1808 (1998)