Assessment of Surrogate Model Settings Using Landscape Analysis

Mikuláš Dvořák[1], Zbyněk Pitra[2,3], Martin Holeňa[3]

[1] Faculty of Information Technology, Czech Technical University, Thákurova 7, Prague, Czech Republic
[2] Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University, Trojanova 13, Prague, Czech Republic
[3] Institute of Computer Science, Czech Academy of Sciences, Pod vodárenskou věží 2, Prague, Czech Republic

Abstract: This work in progress concerns assessment of surrogate model settings for expensive black-box optimization. The assessment is performed in the context of Gaussian process models used in the Doubly Trained Surrogate (DTS) variant of the state-of-the-art black-box optimizer, the Covariance Matrix Adaptation Evolution Strategy (CMA-ES). This work focuses on the connection between Gaussian process surrogate model predictive accuracy and an essential model hyper-parameter – the covariance function. The performance of DTS-CMA-ES is related to the results of landscape analysis of the objective function. To this end various classification and regression methods are used, proposed in the traditional framework for algorithm selection by Rice. Several single-label classification, multi-label classification, and regression methods are experimentally evaluated on data from DTS-CMA-ES runs on the noiseless benchmark functions from the COCO platform for comparing continuous optimizers in black-box settings.

1 Introduction

Optimization is a field of mathematics that has been studied for centuries. Many problems can be reduced to a problem of finding global optima of a function. Gradient descent methods or analytical solutions are often used to solve these problems.

Expensive black-box optimization is addressing optimization problems in situations when a mathematical definition of the optimized objective is unknown and its evaluation costs valuable resources such as money or time.

The *Covariance Matrix Adaptation Evolution Strategy* (CMA-ES [4]) is a stochastic method suitable for optimization of black-box functions. A *surrogate model* is a regression model that can be used to approximate the unknown black-box function. Instead of evaluating the black-box function in every search point, the surrogate model is used to decrease the number of expensive evaluations based on already evaluated points. However, the combination of the CMA-ES with a surrogate model presents new challenges in tuning surrogate models to make the optimization more effective. Finally, *fitness landscape analysis* (FLA) is a technique that is trying to characterize the structure of a fitness landscape with measurable features. As these features are describing the structure of a fitness function, they could provide information based on which the most suitable surrogate model could be obtained.

This paper addresses the problem of how to select the most convenient surrogate model, in the context of various metrics quantifying the quality of the considered surrogate model, in every generation of the Doubly Trained Surrogate Covariance Matrix Adaptation Evolution Strategy (DTS-CMA-ES [14]). Later, this metric can be used, with a set of features from fitness landscape analysis, to train a classification model that selects the surrogate model for any black-box function. This idea is depicted in the Figure 1. An accurate model selection method could be used for the DTS-CMA-ES algorithm and could potentially speed up the optimization process. This work might provide valuable insight for such a goal.

To select a surrogate model, various classification strategies can be used, and by assessing their performance the most suitable classification model can be later utilized. The selection is described in the context of a framework for algorithm selection proposed by Rice in [18].

We have started from the research in [15], where authors used a classification tree for selection mapping. However, the accuracy of the classification tree was not very satisfactory. Therefore, we test more classification models.

This paper is structured as follows. In Section 2, an introduction to surrogate models for surrogate-assisted CMA-ES is presented. Section 3 discusses the design for algorithm selection utilizing fitness landscape analysis and Rice's framework. Finally, in Section 4, various classification and regression methods for selecting the most convenient surrogate model for DTS-CMA-ES are shown.

2 Surrogate Models in the Context of CMA Evolution Strategy

The CMA-ES is an algorithm for numerical black-box optimization. The algorithm can be simplified into a repetition of the following three steps:

(1) sample a new population of size λ by sampling from a multivariate normal distribution $\mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$,

(2) select the μ best offspring from the sampled population based on their respective function values,

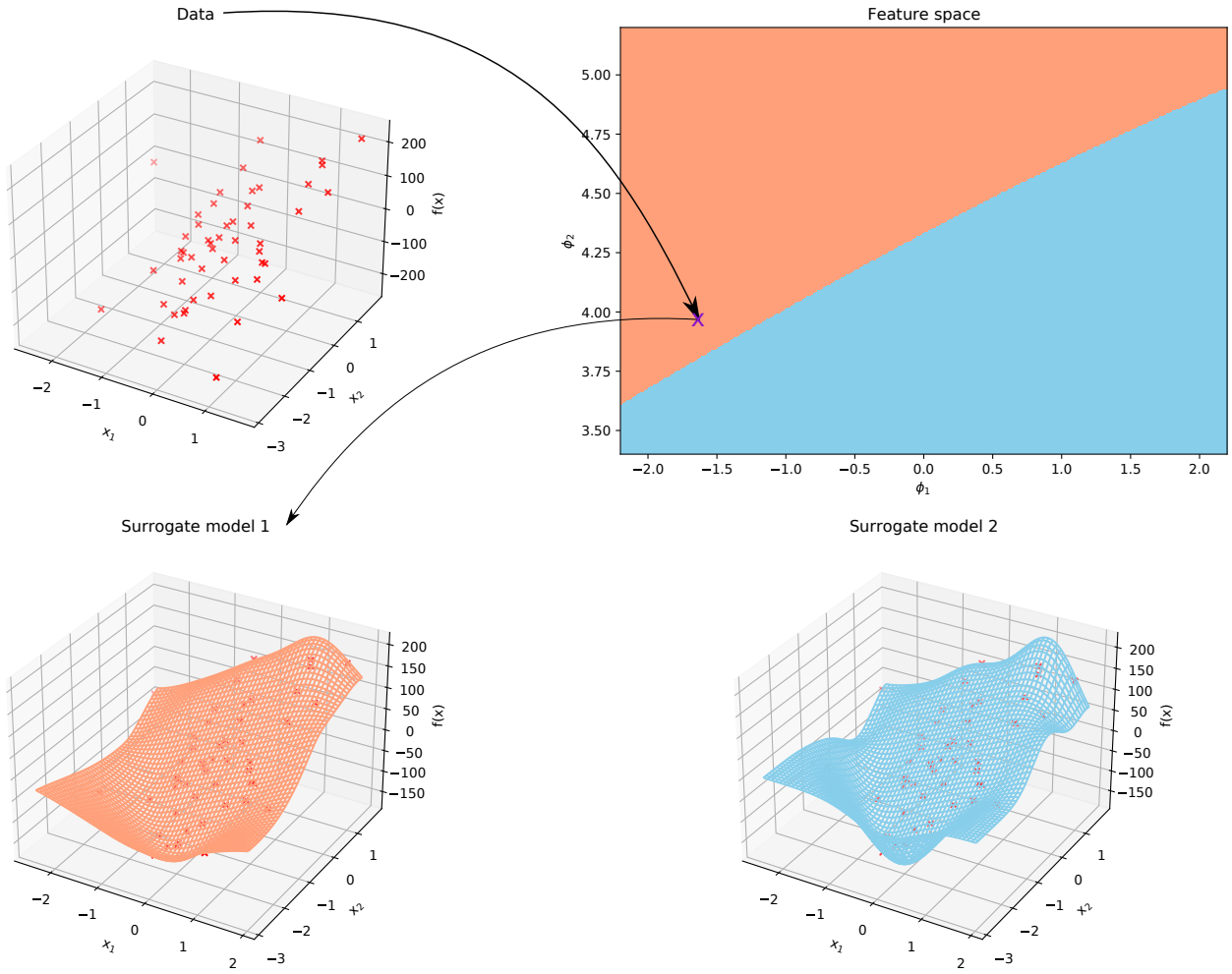

Figure 1: This figure is an artificial illustration of the relation between fitness landscape analysis and surrogate modeling. The top left graph represents the data that are modeled by the surrogate model. The top right graph shows how this data could be represented in the space described by features derived from the fitness landscape analysis. The decision boundary represents a trained classification model that should choose more accurate surrogate model in the fitness landscape analysis space. The bottom figures shows that two surrogate models could model the space in different way and hence one might be more accurate then other.

(3) update parameters of the multivariate distribution $\boldsymbol{m}$ and $\boldsymbol{\Sigma}$ with respect to the selected μ offspring.

In step (2), all λ offspring need to be evaluated in order to select the best μ offspring. A surrogate model that approximates the underlying black-box function can be used to decrease the number of needed expensive evaluations.

Surrogate modeling is a technique based on building regression models of the original function using the already evaluated data points. This technique originated from response surface modeling where the regression models are usually simple polynomial models. Response surface modeling was introduced by George E. P. Box and K. B. Wilson in 1951 [2].

DTS-CMA-ES is a version of the CMA-ES algorithm utilizing surrogate models. This algorithm uses regression models such as Gaussian process for their capability of predicting a whole distribution instead of just a value of the objective function. The covariance function of the used Gaussian process is a hyper-parameter, which we are trying to set by utilizing features from the FLA.

2.1 Gaussian process

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

Due to a joint Gaussian distribution, Gaussian process is described by its mean and covariance function. The mean function $m(\boldsymbol{x})$ and the covariance function $\kappa(\boldsymbol{x},\boldsymbol{x}')$ of a random variable $g(\boldsymbol{x})$ assigned to point $\boldsymbol{x}$ are defined

as

$$\begin{aligned} m(\boldsymbol{x}) &= \mathbb{E}\left[g(\boldsymbol{x})\right], \\ \kappa(\boldsymbol{x},\boldsymbol{x}') &= \mathbb{E}\left[(g(\boldsymbol{x})-m(\boldsymbol{x}))(g(\boldsymbol{x}')-m(\boldsymbol{x}'))^T\right] \end{aligned} \tag{1}$$

and the fact that m and κ define, respectively, the mean and covariance of the variables $g(\boldsymbol{x})$ forming the Gaussian process is sometimes denoted as

$$g(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), \kappa(\boldsymbol{x},\boldsymbol{x}')). \tag{2}$$

The posterior distribution can be inferred with rules for conditioning Gaussians as the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$, where

$$\begin{aligned} \boldsymbol{\mu}^* &= \boldsymbol{\mu}(\boldsymbol{X}^*) + \boldsymbol{K}^{*^T}\boldsymbol{K}^{-1}(\boldsymbol{f}-\boldsymbol{\mu}(\boldsymbol{X})), \\ \boldsymbol{\Sigma}^* &= \boldsymbol{K}^{**} - \boldsymbol{K}^{*^T}\boldsymbol{K}^{-1}\boldsymbol{K}^*, \end{aligned} \tag{3}$$

where $\boldsymbol{f}$ is a vector of measured responses, $\boldsymbol{X}$ is a matrix with inputs of known responses, $\boldsymbol{X}^*$ is a matrix with inputs of unknown responses, $\boldsymbol{K}_{ij} = \kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)$, $\boldsymbol{K}^*_{ij} = \kappa(\boldsymbol{x}_i,\boldsymbol{x}^*_j)$ and $\boldsymbol{K}^{**}_{ij} = \kappa(\boldsymbol{x}^*_i,\boldsymbol{x}^*_j)$ for the considered covariance function κ.

The covariance function κ must be a symmetric function of two vector inputs, and the matrix $\boldsymbol{K}$ by means of κ as described above must be for any number of points $\boldsymbol{x}_i$ positive semidefinite; each such function is called a *kernel*. The kernel defining a covariance function is as a hyper-parameter of a Gaussian process. There is a large variety of available kernels, in this work we consider the following ones.

Polynomial kernels are defined as follows:

$$\kappa(\boldsymbol{x},\boldsymbol{x}') = (\boldsymbol{x}^T\boldsymbol{x}' + \sigma_0^2)^p, \tag{4}$$

where $p \in \mathbb{N}$ and σ_0 is a constant term (bias).

For $p = 1$ the kernel is called *linear* (LIN) and for $p = 2$ the kernel is *quadratic* (Q).

A *Squared exponential* kernel (SE) is defined as

$$\kappa_{\mathrm{SE}}(\boldsymbol{x},\boldsymbol{x}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\ell^2}\right), \tag{5}$$

where ℓ is a *characteristic length-scale*, a hyper-parameter determining the relationship between the distance of vectors in the input space and correlations in the output space.

A *Rational quadratic* kernel (RQ) can be viewed as a generalization of the SE kernel. The RQ kernel is defined as

$$\kappa_{\mathrm{RQ}}(\boldsymbol{x},\boldsymbol{x}') = \sigma^2\left(1+\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\alpha\ell^2}\right)^{-\alpha}. \tag{6}$$

The hyper-parameter $\alpha > 0$ can be seen as a decomposition of the exponential function in the SE kernel.

Frequently, the following two kernels are used:

$$\kappa^{\frac{3}{2}}_{\mathrm{Mat}}(\boldsymbol{x},\boldsymbol{x}') = (1+a)\exp(-a), \text{ where } a = \frac{\sqrt{3}\|\boldsymbol{x}-\boldsymbol{x}'\|}{\ell}, \tag{7}$$

$$\kappa^{\frac{5}{2}}_{\mathrm{Mat}}(\boldsymbol{x},\boldsymbol{x}') = \left(1+a+\frac{\sqrt{5}a}{3\ell}\right)\exp(-a), \text{ where } a = \frac{\sqrt{5}\|\boldsymbol{x}-\boldsymbol{x}'\|}{\ell}. \tag{8}$$

These kernels are from the Matérn class [10].

Another kernel was introduced by Gibbs in [3]:

$$\kappa_{\mathrm{Gibbs}}(\boldsymbol{x},\boldsymbol{x}') = \prod_{i=1}^{D}\left(\frac{2\ell_i(\boldsymbol{x})\ell_i(\boldsymbol{x}')}{\ell_i^2(\boldsymbol{x})+\ell_i^2(\boldsymbol{x}')}\right)^{1/2} \exp\left(-\sum_{i=1}^{D}\frac{(x_i-x'_i)^2}{\ell_i^2(\boldsymbol{x})+\ell_i^2(\boldsymbol{x}')}\right), \tag{9}$$

where ℓ_i is a positive function which can be different for each i and D is the dimension of the vector $\boldsymbol{x}$. Making the hyper-parameter ℓ configurable in every dimension makes this kernel more flexible.

Also a neural network can be used as a kernel for GP. How to derive the following neural network kernel is discussed in [17].

$$\kappa_{\mathrm{NN}}(\boldsymbol{x},\boldsymbol{x}') = \frac{2}{\pi}\arcsin\left(\frac{2\tilde{\boldsymbol{x}}^T\boldsymbol{\Sigma}\tilde{\boldsymbol{x}}'}{\sqrt{(1+2\tilde{\boldsymbol{x}}^T\boldsymbol{\Sigma}\tilde{\boldsymbol{x}}')(1+2\tilde{\boldsymbol{x}}'^T\boldsymbol{\Sigma}\tilde{\boldsymbol{x}}')}}\right), \tag{10}$$

where $\tilde{\boldsymbol{x}}$ is $\boldsymbol{x}$ with an added bias component such that $\tilde{\boldsymbol{x}} = (1, x_1, \ldots, x_D)^T$ and $\boldsymbol{\Sigma}$ denotes a corresponding bias component.

A new kernel can be also created using addition. For instance the addition of a SE kernel to a Q kernel results in a new kernel defined as follows:

$$\kappa_{\mathrm{SE+Q}}(\boldsymbol{x},\boldsymbol{x}') = \sigma^2\exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\ell^2}\right) + (\boldsymbol{x}^T\boldsymbol{x}'+\sigma_0^2)^2. \tag{11}$$

3 Methodology

To clarify characteristics of the data used to train a surrogate model in the DTS-CMA-ES algorithm, we use the explanation from [15]: For each generation g of the DTS-CMA-ES algorithm, a set of surrogate models $\mathcal{M}$ are trained on a training set $\mathcal{T}$. The training set $\mathcal{T}$ is a subset of the archive $\mathcal{A}$ of all evaluated data points. Afterwards, a surrogate model $M \in \mathcal{M}$ is utilized to select new population $\mathcal{P}$. The question is how to select the most convenient surrogate model using the sets $\mathcal{A}, \mathcal{T}, \mathcal{P}$?

3.1 Framework for Model Setting Selection

One way to describe the surrogate model selection problem is to use a framework for algorithm selection proposed by Rice in [18]. This framework is designed with five main components and for problem of surrogate model selection can be briefly explained as:

Data space is a space of possible problems. In this case, data space contains sets of data points that are present in the DTS-CMA-ES runs.

Algorithm space is a space of possible surrogate models to solve a problem from the data space.

Feature space is a space of possible characterizations of the data space. We use feature sets from FLA and CMA-ES features described in Subsection 3.2.

Performance space is a space describing the performance of a particular algorithm for a particular problem.

Selection mapping is a function that gives a surrogate model M, for a particular vector of features ϕ, such that it minimizes models error ε.

The following diagram [13, 18] (Figure 2) illustrates the main parts of this framework and their relations.

The goal is to train a classifier, represented by the selection mapping, which could be later utilized to select the best covariance function of a Gaussian process for given data.

Data space In the DTS-CMA-ES, three sets of data points are used. The first one is an archive $\mathcal{A}$ containing all f-evaluated data points $\{\boldsymbol{x}_i, f(\boldsymbol{x}_i) \mid i = 1,\ldots,n\}$, where n is the number of f-evaluated points. The second one is the training set $\mathcal{T}$ containing f-evaluated data points which are a subset of $\mathcal{A}$ and are utilized for fitting a surrogate model in DTS-CMA-ES. The training set is selected to contain data points that near the currently searched space by the CMA-ES (see [1] for training set selection methods). The last set is a sampled population $\mathcal{P}$, for which the values of the black-box function are unknown. The population $\mathcal{P}$ is selected using the doubly trained evolution control that utilizes the predictions of the Gaussian process surrogate model. These sets are changing each generation. A more detailed explanation of how the sets $\mathcal{T}$ and $\mathcal{P}$ are selected can be found in [1, 14].

Model space The set of considered surrogate models consisted of Gaussian processes with various covariance functions.

Feature space The features are computed on datasets $\mathcal{A}, \mathcal{T}$, and $\mathcal{T} \cup \mathcal{P}$ for each generation in the run of the DTS-CMA-ES algorithm.

Performance space Performance can be measured with a variety of evaluation metrics and the question is which metric would be the most convenient for the surrogate model selection task. In [16], the authors used the Ranking Difference Error (RDE). However, error measures such as Mean Squared Error (MSE), Mean Absolute Error (MAE), or R^2 may be more convenient for the investigation of the relationships between model performance and fitness landscape features.

Selection mapping Utilizing the FLA features, we can construct a D-dimensional space Φ, where each dimension represents one FLA feature. In this space, we can create a classification model that will map a $\phi \in \Phi$ to the respective best performing covariance function of the Gaussian process learned from previous runs of the DTS-CMA-ES algorithm.

Selection mapping $S\colon \Phi \to \mathcal{M}$ is a component that maps landscape features $\phi \in \Phi$ to a model $M \in \mathcal{M}$ such that $S(\phi)$ maximizes the model performance.

3.2 Fitness Landscape Analysis

Fitness landscape analysis (FLA) aims to characterize the structure of a fitness function with measurable features. In the context of expensive black-box optimization, the feature calculation relies only on the already evaluated data points.

In [11], the authors discussed sets of low-level features that can be computed with various techniques. Some of them are not useful for the context of expensive black-box optimization because they require additional evaluations of the optimized black-box function.

Several of such feature sets have been suggested in the literature to support FLA, e.g. *Nearest-Better Clustering* [7], *Information Content of Fitness Sequences* [12], or *Dispersion* [9]. All of the mentioned feature sets were already used in the paper [16] and we briefly describe them in the following paragraphs.

***y*-Distribution** This set contains features based on the distribution of the fitness function values. In [11], the authors have presented three such features: *skewness*, *kurtosis*, and *number of peaks*.

Both the skewness and the kurtosis of a distribution are computed from central moments. The skewness tells us how asymmetric the distribution is and the kurtosis measures how much the distribution differs from the normal distribution in the sense of tailedness.

The last feature is an estimation of the number of peaks in the y-Distribution.

Levelset Levelset features are calculated from a dataset split into two classes based on a threshold in function values. As a split value, the median value or other quantile values have been studied in [11].

Linear, quadratic, and mixture discriminant analysis are used on the partitioned dataset to separate classes. The underlying idea is that for a right choice of the threshold value, a multimodal fitness landscape cannot be separated with linear or quadratic discriminant analysis. However,

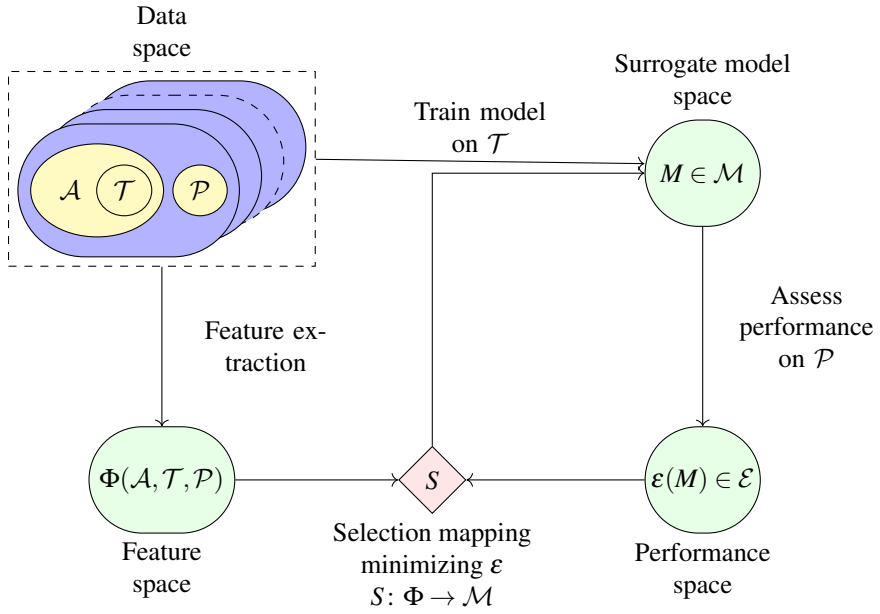

Figure 2: Modified Rice's framework for surrogate model selection in DTS-CMA-ES.

mixture discriminant analysis should have a better performance on a multimodal fitness landscape.

The features are defined as cross-validated misclassification errors for each type of discriminant analysis.

Meta-model Features from this class are acquired from fitting a linear and quadratic regression model.

The model performance, specifically the adjusted R^2 value of linear and quadratic models, has been used in [11] together with the minimum and the maximum of the absolute values of the linear model coefficients. For the quadratic model, the authors used the maximum absolute value divided by the minimum absolute value of the fitted model's coefficients.

Nearest-Better Clustering The features based on Nearest-Better Clustering (NBC) have been proposed in [7]. The presented five features should help to recognize funnel structures in the fitness landscape.

Dispersion Dispersion of a function measures how close together the sampled points are in the search space [9]. The dispersion features are derived from this idea. They average differences between dispersion values below a certain moving threshold value.

To estimate the dispersion, the authors of [9] sampled the space n times and took the best b points from which they averaged pairwise distances between them. This step was repeated for two different n values, and the final dispersion was computed by subtracting those results. That way a difference in dispersion is estimated.

Information Content of Fitness Sequences Information Content of Fitness Sequences (ICoFS) introduced in [12], measures how difficult is it to describe a given fitness function. For instance, a low information function would be a constant fitness function as opposed to a high information function such as some multimodal complicated fitness function.

This method uses neighboring values and compares their fitness values. The comparisons are later transformed into discrete information from which the features are computed.

CMA-ES features The authors of [16] proposed features related to the DTS-CMA-ES algorithm. They are computed from the CMA-ES settings, from the set of points $\boldsymbol{X} = \{\boldsymbol{x}_i \mid i = 1,\ldots,n\}$ for which the function value is known, and from DTS-CMA-ES parameters such as.

- The generation number g is an easy to obtain feature derived from an optimization run of DTS-CMA-ES.
- CMA-ES uses a step-size $\sigma^{(g)}$ for controlling the size of a distribution from which the CMA-ES samples new points. Therefore, the step-size can be also used as a feature.
- The evolution path $\boldsymbol{p}_c$ and the σ evolution path length features are derived from the evolution paths length used in the CMA-ES. These features encode how the path of the evolution process has changed in recent generations and measure how useful were previous steps for the optimization.
- An additional CMA-ES feature is derived from the number of restarts of the DTS-CMA-ES algorithm. This could indicate how difficult the problem is.

- Mahalanobis distance of the CMA-ES mean $\boldsymbol{m}^{(g)}$ to the mean of the empirical distribution of all points $\boldsymbol{X}$ is another feature described in [16]. This feature indicates the suitability of $\boldsymbol{X}$ for training a surrogate model.

- The CMA similarity likelihood feature is the log-likelihood of all points $\boldsymbol{X}$ with respect to the CMA-ES distribution. This may also represent a measure of set suitability for a surrogate model training.

4 Experimental Evaluation

Several experiments using the data obtained during the run of the DTS-CMA-ES on benchmark functions with different surrogate models were designed. From the error measures of used surrogate models, the best surrogate model can be selected as the one with the minimal error.

We used Gaussian processes as a surrogate model. In particular, the following covariance functions were used: κ_{LIN}, κ_{SE}, κ_{RQ}, κ_{SE}, $\kappa_{\text{Mat}}^{\frac{5}{2}}$, κ_{NN}, κ_{Gibbs}, and $\kappa_{\text{SE+Q}}$. The parameters of the kernel defining the covariance function are found by maximum-likelihood or leave-one-out cross-validation method [1].

In the feature space, we measured the following low-level feature sets: y-Distribution, Levelset, Meta-Model, Nearest-Better Clustering, Dispersion, Information Content, and CMA-ES features. These sets are described in greater detail in Subsection 3.2.

Experiments are compared using two accuracies. The exact accuracy measures exact matches of the classified kernel and the true best performing kernel. The loose accuracy is calculated from loose matches that considers as a correctly classified a prediction which falls into similarly best performing kernels. Kernels are considered similarly best performing if their error is in the 5% quantile of the considered kernels errors for a particular data point.

Experiments are compared to a baseline model that recommends the most frequent best performing kernel from the training set. To this end, various approaches can be used. With classification models we can classify the best performing kernels, or by utilizing the information about errors, we can apply regression or multi-label classifation models.

The following classifiers or their regression versions were trained: decision tree, random forest, support vector machine, and artificial neural network with two hidden dense layers (50 and 25 neurons respectively). Results are shown in Figures 3 and 4.

The baseline model was outperformed by almost every presented classification method. Single-label classification methods have the highest accuracy, but its accuracy is very similar to the multi-label classification methods. The advantage of the multi-label classification is that it provides more flexibility for tuning the settings and hence provides more room for improvement.

The differences between exact and loose accuracy vary between used error measures. For RDE, MSE, and MAE those differences are greater than for R^2 error. This might be a consequence of choosing the 5% quantile for similarly best performing kernels.

4.1 Used Data

The problems used for retrieving the data $\{\mathcal{A}^{(i)}, \mathcal{T}^{(i)}, \mathcal{P}^{(i)} \mid i = 1, \ldots, g\}$ in this paper were obtained from running the DTS-CMA-ES algorithm on the Black-Box Optimization Benchmarks from the COmparing Continuous Optimisers (COCO) platform, namely, problems in dimensions 2, 3, 5, 10, and 20 on instances 11-15 [5, 6]. The sets $\{\mathcal{A}^{(i)}, \mathcal{T}^{(i)}, \mathcal{P}^{(i)} \mid i = 1, \ldots, g\}$ were extracted for 25 uniformly selected generations for 8 considered surrogate models. The algorithm was terminated if one of the following two conditions was true:

(1) the target fitness value 10^{-8} is reached, or

(2) the number of evaluations of the optimized fitness function f is at least $250D$, where D is the dimension of the function f.

From the data, we calculate FLA features and errors derived from surrogate models predictions. The considered error measures are: RDE, MSE, MAE, and R^2.

The data generated for this paper were already used in [16]. Compared to [16], more metrics in the performance space and more classifiers are investigated. Features were calculated using the algorithm underlying the R-package flacco [8], reimplemented in the MATLAB language.

4.2 Single-Label Classification

A classifier $S_c \colon \Phi \to \mathcal{M}$ was trained on labels of the best performing models. To obtain a label for a data point, the minimal error value for each GP kernel was found and its kernel set as a label. It is not always clear which model should be selected as a label because multiple models can have equal errors. To address this ambiguity, multi-label classifiers are tested in the next subsection.

4.3 Multi-Label Classification

From the original dataset, not only the best, but all nearly best performing models are found and used as labels for S_c training. The trained classifier is then capable of predicting multiple labels for given landscape features. However, for a fair comparison with the single-label classification and with the regression approach, only one label has to be predicted. To this end, a regression model is utilized to predict the best performing model to select a single-label among labels predicted by the multi-label classifier. That regression model considers only labels predicted by the multi-label classifier and among them, the one best according to the regression model is selected as the final label for comparison.

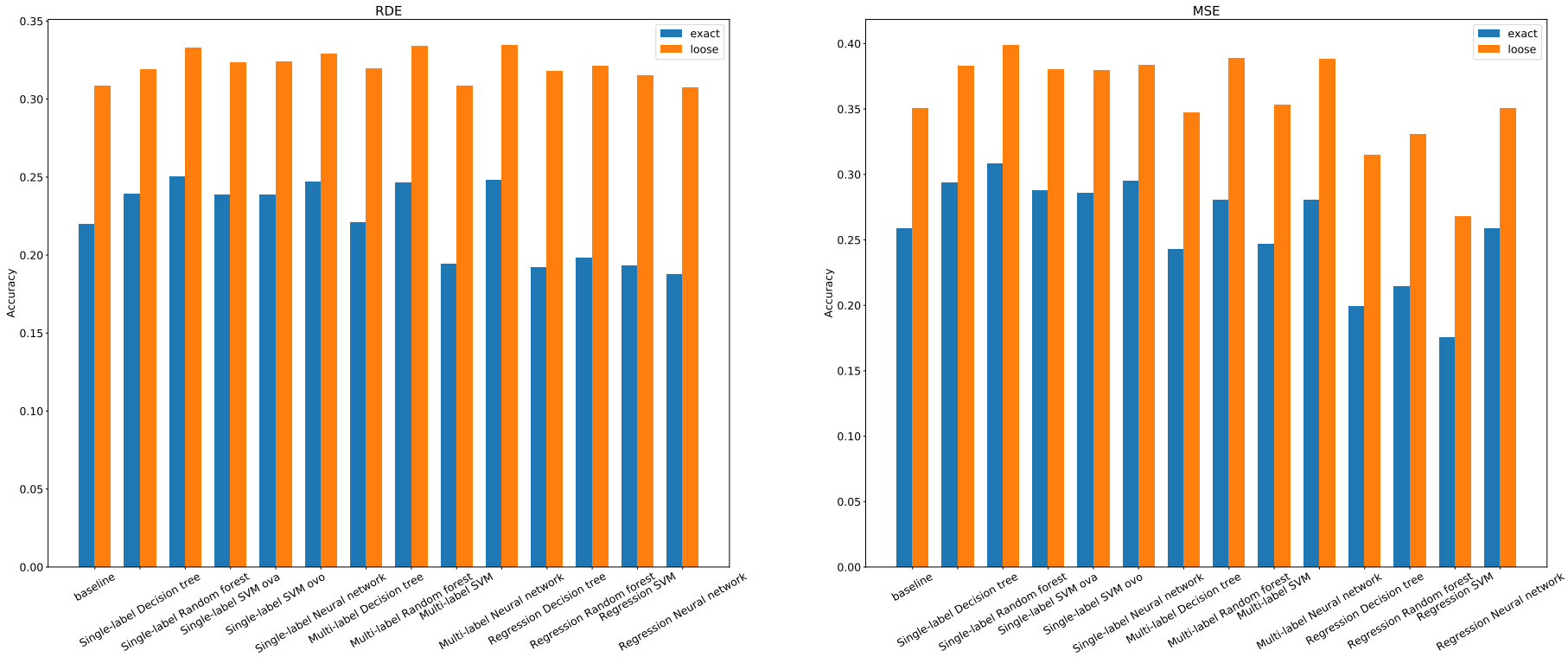

Figure 3: Exact accuracy and loose accuracy for each considered model trained with landscape features and predicting the best performing model w.r.t. Ranking Difference Error and Mean Squared Error. Hyper-parameters for each classifier were found using a 5-fold cross-validation and final accuracies were measured on the test set containing 20% of the original data set.

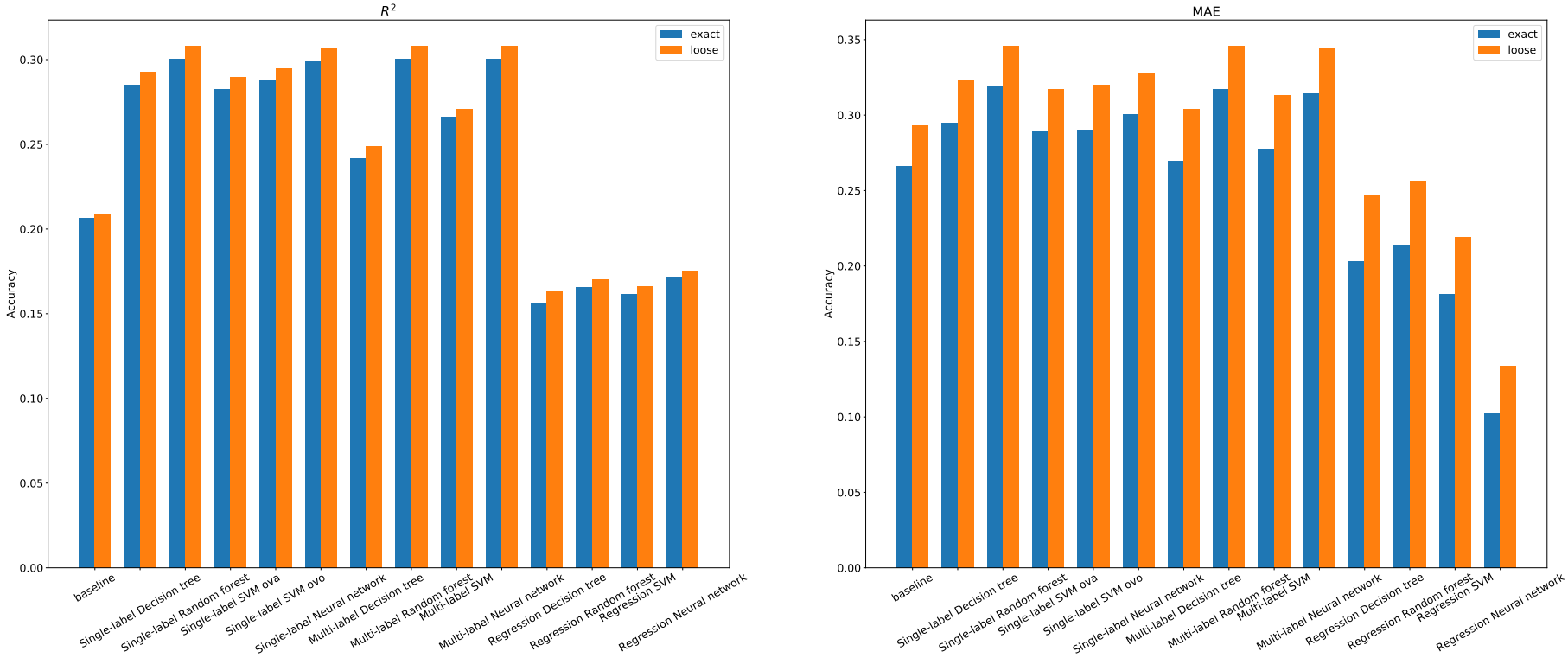

Figure 4: Exact accuracy and loose accuracy for each considered model trained with landscape features and predicting the best performing model w.r.t. R^2 and Mean Absolute Error. Hyper-parameters for each classifier were found using a 5-fold cross-validation and final accuracies were measured on the test set containing 20% of the original data set.

4.4 Regression

A regression model S_r: $\Phi \to \mathcal{E}^{|\mathcal{M}|}$ was trained to predict an error of a surrogate model for given landscape features. The S_r model yields errors from which a minimum is found and its corresponding surrogate model is selected.

Some regression models yield only one prediction. To this end, for each surrogate model one regression model is trained to predict the error and results are then combined.

5 Conclusion

A design of various methods for classifying the data from FLA to predict the most convenient surrogate model were presented. The baseline model was outperformed with almost every presented classification method. However, the differences between the highest accuracy scores and the baseline scores are very small. From the accuracy scores in Figures 3 and 4, it is clear that both the best classifiers and the best regression models are random forests for almost all considered approaches.

The accuracy scores suggests that the classifiers did not solve the problem of surrogate model selection for DTS-CMA-ES algorithm completely. However, this method might improve the performance of the DTS-CMA-ES algorithm because even a small improvement in accuracy might be useful.

Possible problems might be with an imbalance of classes in the training dataset and/or with similar performance of some surrogate models for some data points. The latter one was addressed with multi-label classification methods.

A further improvement of the presented methods could be achieved through improved fitness landscape analysis. This concerns on the one hand new suitable fitness landscape features, on the other hand feature selection that could reduce the feature space and improve the performance of employed classifiers and regression models.

Acknowledgement

The research reported in this paper has been supported by the Czech Science Foundation (GAČR) grant 18-18080S.

References

[1] Lukáš Bajer, Zbyněk Pitra, Jakub Repický, and Martin Holeňa. Gaussian process surrogate models for the CMA evolution strategy. *Evolutionary computation*, 27(4):665–697, 2019.

[2] G. E. P. Box and K. B. Wilson. On the Experimental Attainment of Optimum Conditions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 13(1):1–38, jan 1951.

[3] Mark N Gibbs. *Bayesian Gaussian processes for regression and classification*. PhD thesis, Citeseer, 1998.

[4] Nikolaus Hansen. The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation*, pages 75–102. Springer, 2006.

[5] Nikolaus Hansen, Anne Auger, Steffen Finck, and Raymond Ros. Real-Parameter Black-Box Optimization Benchmarking 2009: Noiseless Functions Definitions. Technical report, Citeseer, 2010.

[6] Nikolaus Hansen, Anne Auger, Steffen Finck, and Raymond Ros. Real-Parameter Black-Box Optimization Benchmarking 2012: Experimental Setup. Technical report, Citeseer, 2012.

[7] Pascal Kerschke, Mike Preuss, Simon Wessing, and Heike Trautmann. Detecting funnel structures by means of exploratory landscape analysis. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 265–272, 2015.

[8] Pascal Kerschke and Heike Trautmann. Comprehensive Feature-Based Landscape Analysis of Continuous and Constrained Optimization Problems Using the R-package flacco. In *Applications in Statistical Computing – From Music Data Analysis to Industrial Quality Improvement*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 93 – 123. Springer, 2019.

[9] Monte Lunacek and Darrell Whitley. The dispersion metric and the CMA evolution strategy. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation - GECCO 06*. ACM Press, 2006.

[10] Bertil Matérn. Spatial variation. Technical report, 1960.

[11] Olaf Mersmann, Bernd Bischl, Heike Trautmann, Mike Preuss, Claus Weihs, and Günter Rudolph. Exploratory landscape analysis. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 829–836, 2011.

[12] Mario A Muñoz, Michael Kirley, and Saman K Halgamuge. Exploratory landscape analysis of continuous space optimization problems using information content. *IEEE transactions on evolutionary computation*, 19(1):74–87, 2014.

[13] Mario A Muñoz, Yuan Sun, Michael Kirley, and Saman K Halgamuge. Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges. *Information Sciences*, 317:224–245, 2015.

[14] Zbyněk Pitra, Lukáš Bajer, and Martin Holeňa. Doubly trained evolution control for the surrogate CMA-ES. In *International Conference on Parallel Problem Solving from Nature*, pages 59–68. Springer, 2016.

[15] Zbyněk Pitra, Lukáš Bajer, and Martin Holeňa. Knowledge-based Selection of Gaussian Process Surrogates. In *Workshop & Tutorial on Interactive Adaptive Learning*, page 48, 2019.

[16] Zbyněk Pitra, Jakub Repický, and Martin Holeňa. Landscape analysis of gaussian process surrogates for the covariance matrix adaptation evolution strategy. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 691–699, 2019.

[17] Carl Rasmussen. *Gaussian processes for machine learning*. MIT Press, Cambridge, Mass, 2006.

[18] John R. Rice et al. The algorithm selection problem. *Advances in computers*, 15(65-118):5, 1976.