Optimally weighted ensembles of surrogate models for sequential parameter optimization

Echtenbruck, M.M.

Chapter 3

Taxonomy

In this Chapter, an introduction to the development in the field of surrogate modeling towards ensembles and a more detailed overview of ensemble methods is given. The functionality of the different approaches is analyzed, and a taxonomy of ensemble methods is derived from the observations made. The variety of existing approaches is large and more than a few are specifically designed for one problem or one class of problems. The main focus of this overview is laid on methods that conform to the goal of this work. These are mainly regression models, that can also be applied to Designs of Experiment (DOE) of a smaller size. Additional approaches that do not fit this specification are introduced afterward. Finally, the insights are discussed, and a conclusion is drawn specifying the primary necessities for the envisioned ensemble method.

Models are used when the direct evaluation of the actual fitness function would be too expensive. However, the performance of the available models on different objective functions is strongly varying. All models have their strengths and weaknesses which enable them to model some features better than others. Thus, whenever a model is to be applied, the result is strongly depending on the choice of the model. To choose the right model, not only knowledge about the objective function is needed, but also about the strengths and weaknesses of the available models. This knowledge is not always available, which led to different approaches to facilitate this choice by providing tools and criteria to evaluate or choose a model, methods to automatically select the most appropriate models or even to combine several models into one stronger model. But the different approaches are so diverse that it is not easy to obtain a broad overview.

In the following the different approaches are examined closer, differences and

similarities between the approaches are pointed out, and possible advantages and disadvantages are discussed.

The general question is: Given, that more than only one model is available, how to obtain one prediction from n models.
There are two criteria that allow for the first classification of these approaches. We can differentiate between Model Selection and Model Mixtures or Model combination. Also, we can distinguish by regarding the number of model fitting processes, since there are solutions that only fit a single model to the data and solutions that fit all models to the data.

3.1 Overview of Previous Developments and the State of the Art

As stated before, the choice of the surrogate model can have a significant influence on the solution quality and performance of a surrogate model based optimization process. Burnham et al. even stated that the choice of the right surrogate model is the most crucial question in making statistical inferences [18]. But in order to make meaningful decisions on which surrogate model to select for a given problem, often expert knowledge is needed. This includes knowledge about the objective function and the characteristics of the surrogate model likewise.

However, if there is no preliminary knowledge about the objective function or the available surrogate models, the choice has to be taken nonetheless. This may be done by just choosing the surrogate model that performed well on past optimization tasks. It would even be possible to switch between surrogate models during the optimization process randomly or applying a round robin method to give all available surrogate models their fair share if more predictions are needed sequentially [76]. Other more sophisticated methods might interpret the problem as a multi-armed bandit problem [77]. A well-known strategy is SoftMax, which uses a probability vector where each element represents the probability for a corresponding model to be chosen. The probability vector is updated depending on the reward received for the chosen models [78]. However, these ad hoc rules do not rely on the data to help select the best model and therefore ignore the principle of parsimony[1]. The principle of parsimony [80], also known as Ockham's

[1] Newton wrote in one of his books: "We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances." [79, p.731]

razor, describes the idea that, when multiple hypotheses are available to explain a thing, one should select the one with the fewest possible assumptions.

To overcome this problem, it would be beneficial if the algorithm could learn all by itself which surrogate model type suits the problem best, based on the given data. This can be done by evaluating different models on available training data and using a statistical model selection approach to select the most promising surrogate model [76].

But how to handle the situation when there is more than one strong model in the set? In such circumstances, it might be beneficial to combine inference output across several models. In statistics and machine learning an *ensemble* is a prediction model from several models, aiming for better accuracy.
Different approaches for building ensembles of surrogates are known. Bagging [29] combines results from randomly generated training data partitions whereas Boosting [81] combines several weak learners to a strong one in a stochastic setting. Weighted averaging approaches combine model predictions by calculating the mean or the median of different predictions [26]. But also operations like calculating the minimum or the maximum over these predictions may be thinkable for some applications [82].

Since imprecise models should not deteriorate the overall result, a weighting scheme is introduced. In [83, 84, 85] every model's result for a single design point is weighted using some criterion, i.e., Akaike's Information Criterion (AIC). The sum over all models yields the final value assigned to the design point. A similar approach is blending or stacking [86], where the weights are chosen in an additional training step. Polikar [87] named further ensemble methods.

3.2 Single Evaluation Model Selection

The first class of solutions to handle a large set of models for generating one prediction only evaluates the single models that have been selected. The diagram in Figure 3.1 displays the general flow of this approach. Whenever a single model comes to action, it has to be selected from the set of available models using some criterion. If the user selects the model by hand, this may be done by guessing (if there is no a priori knowledge available) or the model may be selected by preference (which for instance could be based on good experiences made during previous applications) and so on.

However, also automated model selection methods were proposed as an efficient

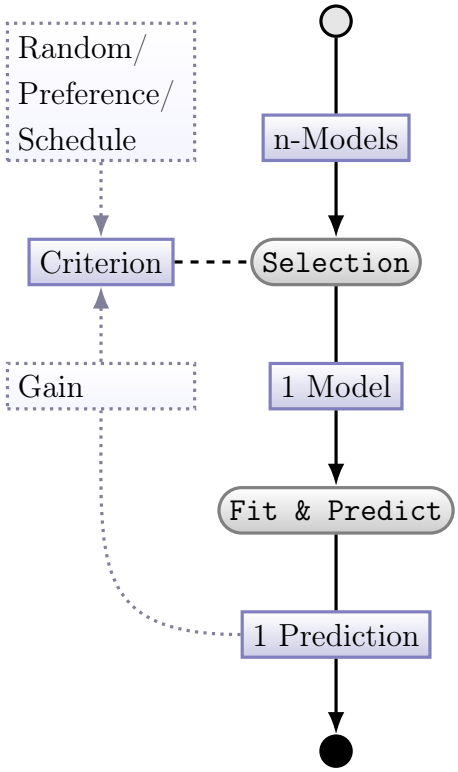

Figure 3.1: The Figure illustrates the general procedure in the case of *Single Evaluation Model Selection*. Given, that more than one model is available, one model has to be chosen with respect to some criterion. This model can then be fitted to the data and used for prediction. Knowledge gained in this step may be used in future decisions.

solution to the model selection problem in SPO as well as to the cold start problem in online modeling applications when there is not yet enough data available [88]. In the latter case, learning the choice of the right model when there is no data available yet corresponds to problems in the context of multi-armed bandits [77]. Methods proposed to approach the model selection problem, in this case, are often based on known solutions to the multi-armed bandit problem or use simple schedules.

The 'model scheduling' approaches provide a schedule that specifies the order in which to select a model. The most straightforward solution is to use a random order or to apply a round-robin strategy [76]. More advanced solutions to the multi-armed bandit problem also consider results from previous evaluations. ε-Greedy strategies choose the model that provided the best reward so far with a probability of ε [76]. Soft-Max approaches keep probabilities for every model

available, starting with an even distribution of probabilities. After every step, all probabilities are then adjusted according to the success of the last model [76]. To allow for a better distribution of the initial probabilities, an initialization step can be carried out when enough data is available [89]. The Bayesian Learning Automaton keeps track of a variable for each model that affects the probability for the corresponding model to be chosen, but unlike the Soft-Max approach, these variables are adjusted independently. Also, many other statistical criteria are used for model selection; an overview of some of them is given in [90].

All of these approaches benefit from the fact that in every step only a single model has to be fitted to the data. However, these ad hoc rules ignore the rules of parsimony and do not, or only marginally, rely on the data to help select the best model. The use of a schedule, or to chose even randomly, gives the same chance to adequate models and inadequate models likewise. The approaches that utilize the reward of the last model struggle with the same problem. The algorithm may settle down to a single model or a small subset of better performing models after some time. But the chances are that due to unfortunate choices for the reward evaluation and during the first steps, better performing models are put at a disadvantage.

3.3 Multi Evaluation Model Selection

A simple way to approach the drawbacks that come with the model selection methods presented in Section 3.2 is to evaluate all models on the data. This way the decision can be made based upon the models' predictions or their performances. Figure 3.2 displays the flow of such approaches. ε-Greedy approaches choose the model that performed best in the previous step, in terms of prediction error, with a probability of ε, with $\varepsilon \in [0, 1]$. With the complementary probability of $(1 - \varepsilon)$ one of the remaining models is chosen randomly [76].

This choice must not be taken based on knowledge from the last step but can also be based on the predictions itself. For such approaches, all models have to be fit to the data in an initial step. Baxter [91] proposed to choose between the predictions of two neural networks that were trained for different objectives by comparing their predictions to a predefined threshold.

Jacobs et al. [92] proposed to combine a set of neural networks by using one of the networks as a gating network that learns which model is the best choice for a given input.

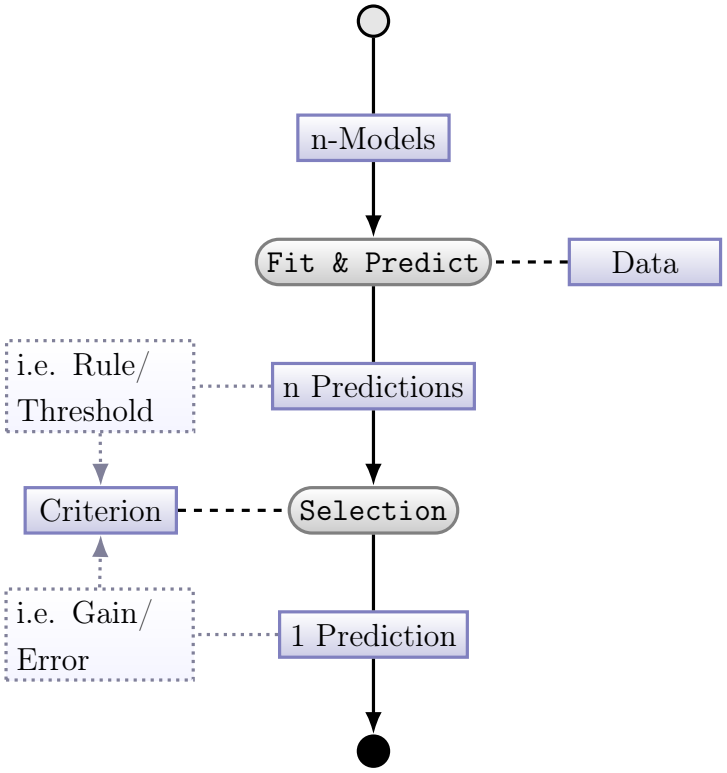

Figure 3.2: The Figure illustrates one approach to *Multi Evaluation Model Selection*, where the choice of the model is based on its prediction. For this purpose, all available models are fitted to the data. The model whose prediction satisfies the predefined criterion is chosen. Knowledge gained in this step may be used in future decisions.

Still, these approaches have limited insight into the performances of the models. Introducing an additional step to evaluate the models on the data allows for a deeper insight into the performances of the models on the data. By applying methods such as cross-validation, the prediction errors of the models can be estimated directly. Figure 3.3 displays the flow of such processes.

Friese et al. [76] proposed a method that does a leave-one-out cross-validation and an additional fitting step on the full data set to gain information about the models underlying uncertainty. This information is then combined with the models' error from the last prediction to obtain an indicator for the selection of the model.

An obvious drawback of such approaches is the number of fitting processes that have to be carried out to evaluate all models the one or the other way. For an

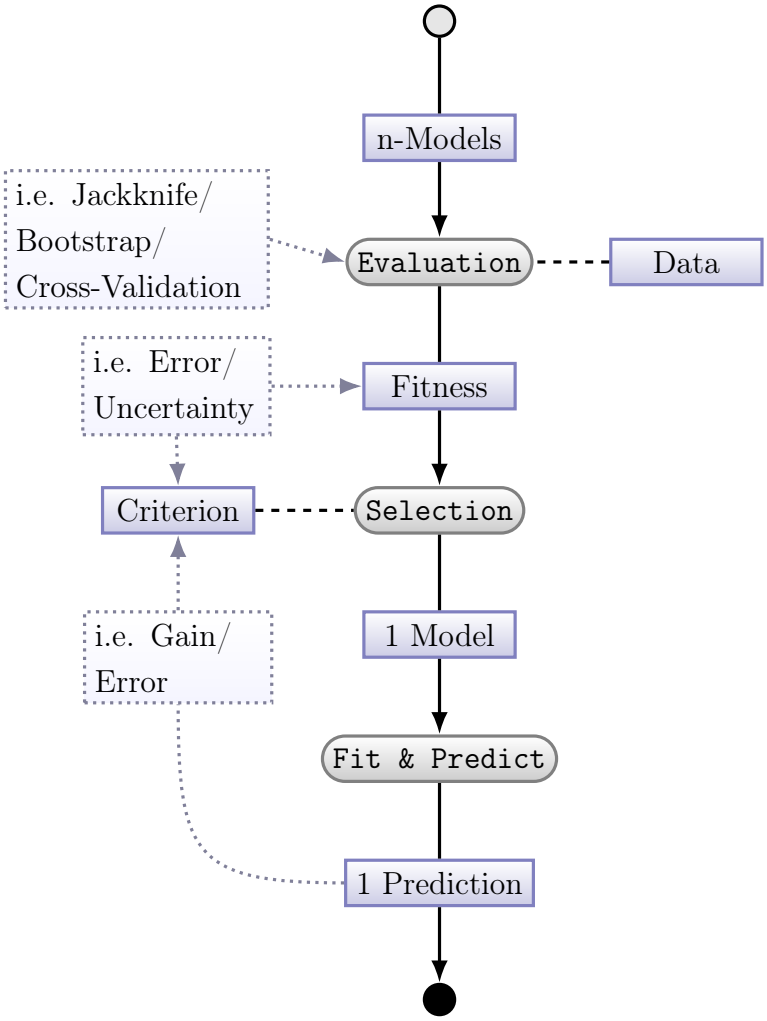

Figure 3.3: The Figure illustrates an approach to *Multi Evaluation Model Selection*, where the choice of the model is based on the general fit of the model on already known data. For this purpose, all available models are evaluated on the available data. The model that scores best during this evaluation is chosen. Knowledge gained in this step may be used in future decisions.

online or sequential process with constraints on the calculation time for the models this step might not be feasible. However, in real-world applications, the actual expenses originate from the evaluation of the objective function. The calculation times may be of small concern.

3.4 Model Combination

Model selection strategies, as introduced before, assign scores to the candidate models to allow for evaluation. In the case of single evaluation model selection, as introduced in Section 3.2, this score is based on performance values of previous steps, whereas in multi evaluation model selection, as introduced in Section 3.3, this score was obtained by a preliminary evaluation of the available models. Such evaluations might yield a clear winner, but they might also lead to the insight that several candidate models perform comparably well. Claeskens et al. [93] state, that in such circumstances, it may be beneficial combining the predictions of these strong models to obtain a more accurate or even better prediction. A straightforward approach to achieve this would be to fit all models to the data and then apply some rule that defines how to combine the various predictions to one.

Figure 3.4 displays the process flow of such methods.

With Bagging [29] multiple versions of a model are generated by fitting them to different partitions of the training data (further referred to as multi-data). Breiman et al. showed, that by averaging the predictions of these models a prediction of higher accuracy can be achieved. Random Forests [26] are another example of this approach where Bagging is used with trees. Their output is combined by majority voting, in the case of classification, or by averaging, in the case of regression.

Boosting [81, 94] is like an add-on to bagging that performs the fitting of the model on the multi-data sequentially. This way data that has not been learned adequately can be considered for the next fitting process with a higher probability. Models generated are combined by weighted averaging, with the weights derived from their prediction errors on the remaining data of the training set.

Bishop [95] combined a set of artificial neural networks to one committee of networks by calculating the weighted sum of their predictions. His method uses the error that the models make at the training points to define the models' weights. The only restriction on the weights is that they have to sum up to one.

Van Stein et al. [96] clustered the training data and learned an independent Kriging model on each data cluster. They introduced a variety of methods to cluster the data and proposed several approaches to retrieve a single prediction from this cluster. The original method [97] considers the Kriging models to be independent, and given this assumption computes an optimally weighted average. One of these approaches is to learn a decision tree on the data with its leaves

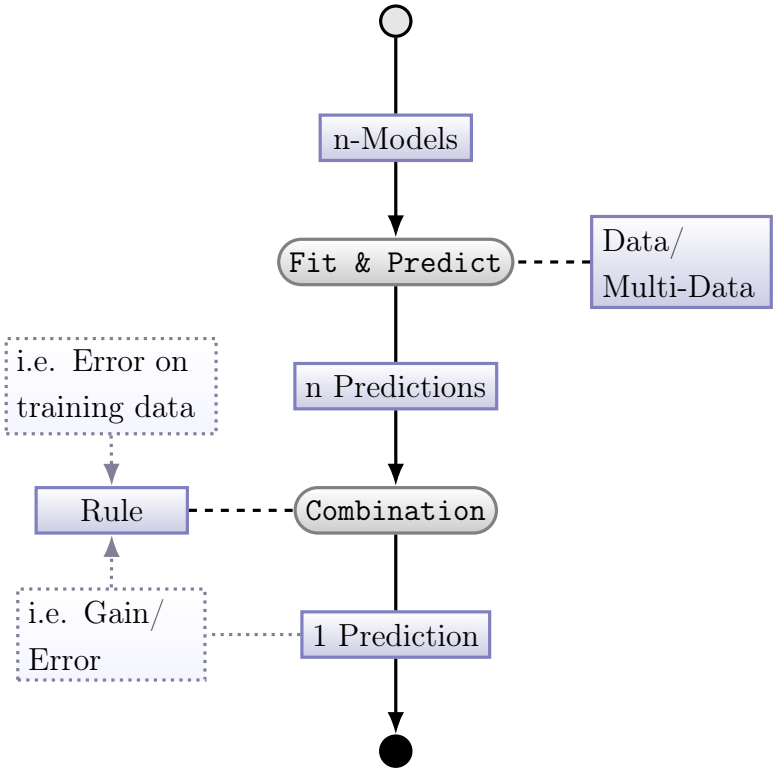

Figure 3.4: The Figure illustrates an approach to *Multi Evaluation Model Combination*, where the predictions of multiple models are combined using a predefined rule. For this purpose, all available models are fitted to the data, or partitions of the data. Knowledge gained from the prediction, or the single predicitons, may be used in future decisions.

being Kriging models. The probability for a data point to be assigned to one leaf of the tree is also used as the weight for the calculation of a weighted sum of the predictions.

Given a set of heterogeneous models is to be used, the models can be fitted to the complete data, variance in their predictions is already given through the heterogeneity of the models, and then be combined by averaging their predictions. However, working with heterogeneous models, this approach may run the risk that one model in the set performs considerably worse, which would bias the outcome of the averaged prediction. Using information (i.e., prediction error) from the last step would minimize this risk [76].

Zerpa et al. [98] proposed to combine heterogeneous models, that are able to provide information about their variance, by calculating a weighted sum of their

predictions. The information about the models' variance is used for the calculation of the models' weights.

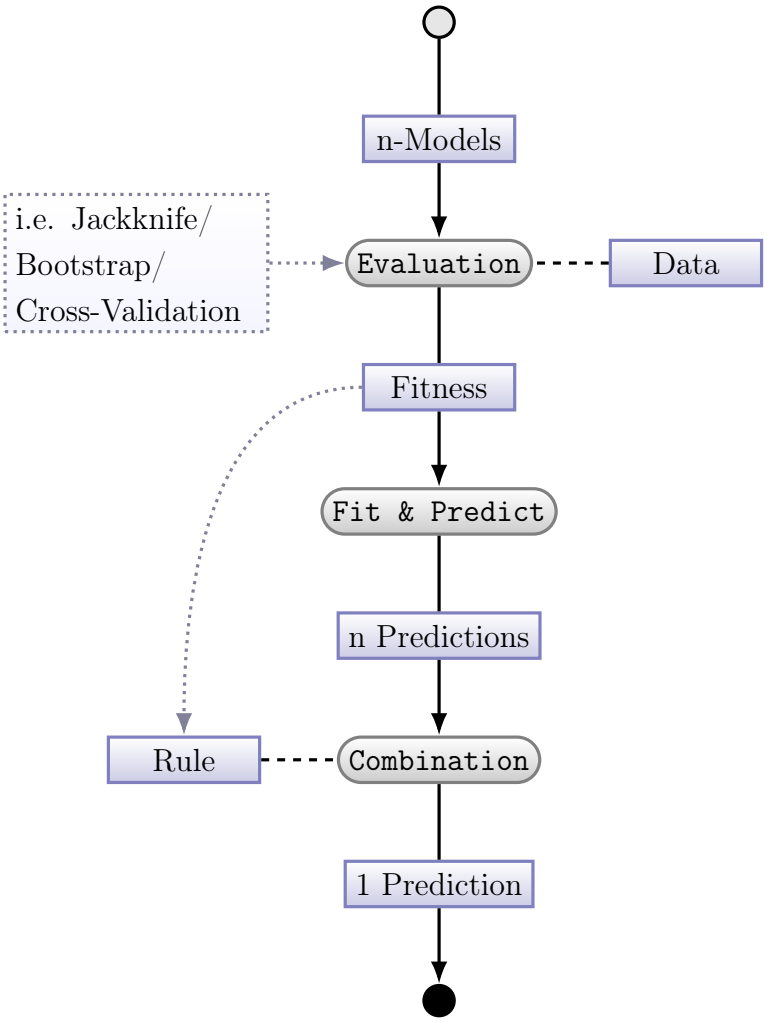

Figure 3.5: The Figure illustrates the most elaborate approach to *Multi Evaluation Model Combination* where the predictions of multiple models are combined according to their fitness on the available data. For this purpose, all available models are evaluated on the data. Each model's score directly affects how the models are combined.

Like with the Model Selection strategies here also an additional step to evaluate the models fit can be introduced (cf. Figure 3.5). Perrone et al. [99] combined a set of ten neural networks by computing a weighted average of the models' predictions. They performed a cross-validation step on the training data to determine

the prediction errors and calculated the optimal weights, in terms of minimal mean squared error[1] (MSE) based on these errors.

Goel et al. [100] combined three heterogeneous models by calculating the weighted sum of the models' predictions. To determine the weights, they used a formula that is based on the generalized mean square cross-validation error of the models. The formula uses two parameters α and β that have to be specified, and control if more trust is laid on the general average of all models predictions (larger α values and smaller negative β values) or if more weight is laid on the single model.

Acar et al. [101], proposed some adaptations of previously defined approaches of weighted sum ensembles for better generalizability and performance. Like in previous works they also required the weights to sum up to one as the only constraint on the weights. Building on the approach of Bishop et al. [95], they proposed to use k-fold cross-validation to allow for an evaluation of models that have per definition no error at the training points. For the works of Goel et al. [100] they suggested to select the parameters α and β to minimize the generalized mean squared error[2](GMSE) of the ensemble. Another approach they propose is a weighted sum ensemble that is evaluated by its root mean squared error[3] (RMSE) with respect to a predefined number N_V of evaluation points ($N_V = 2, 3$ or 5). They optimize the weights using a gradient-based search algorithm starting from the center point, annotating that the search space is not necessarily convex so that the solution possibly only presents a local minimum.

As mentioned before, an obvious drawback of these approaches is the number of fitting processes that have to be carried out. But, unlike in model selection processes, the additional information that might be retrieved from the other models is not necessarily discarded. Previous works show that further enhancement in accuracy can be achieved by combining several models of similar accuracy. However, also with weaker models being part of the set, the combination of models can be beneficial if the strengths and weaknesses are carefully considered.

[1] The MSE calculates the average of the squares of the errors between the observed value and its estimation. It is defined as $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$.

[2] The GMSE refers to the MSE applied in a leave-one-out cross-validation process.

[3] The RMSE calculates the root of the MSE. It is defined as $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$.

3.5 Further Ensemble Generating Strategies

The ideas regarded so far give an overview of the main strategies and ideas on the topic of ensemble building. Of course, this roundup does not claim to be complete. As said in the beginning, the main focus for this overview is laid on methods that conform with the goal of this work. However, there remain approaches to combine models and areas of application that have not been mentioned so far.

Torgo et al. [102] presented a method that uses a regression tree with several alternative independent models in the tree leaves and this way outperformed standard regression tree methods that only average the function value represented by a single leaf.

Dasarathy et al. [103] partitioned the feature space to use two or more classifiers in a pattern recognition system.

Van Stein et al. [104] proposed to improve the performance of Kriging on functions of higher complexity, in terms of dimension or number of known points, by partitioning the data set into smaller disjoint clusters of data, distinguishing between randomly generated clusters and location-based clustering with randomly chosen center points. Their approach trains individual Kriging models on each data cluster. For the prediction, a weighted average of the different models' predictions is calculated using the variance information for each cluster. They showed that using location-based clustering leads to better results than random clustering. The so-called Cluster Kriging generated this way outperforms Ordinary Kriging in terms of computation time as well as in terms of accuracy.

Ginsbourger et al. [105] proposed to use multiple kernels within Kriging. In this case, a mixture of the kernels is defined using a weighted sum. Friese et al. [106] applied the idea of ensemble modeling to time series analysis and forecasting. By adjusting the seasonality of the input data in a preprocessing step they enabled a larger set of models to be used on the data. For each forecasting step, they learned all available models and then used the AIC for the selection of the most promising model.

The possibilities of defining ensembles considering only the different options of combining and evaluating base models, as well as ensembles, seem vast. For evaluation, different methods (i.e., Bootstrapping, Cross-Validation) as also different quality measures or criteria are available (i.e., RMSE, MSE or AIC). Most approaches apply some form of weighted sum for the combination of multiple models. However, for different applications also different combination schemes may be beneficial. Kittler et al. [82] compared the operators product, sum, min,

max, median and majority voting to combine multiple models in a pattern recognition system.

$$
\begin{aligned}
\boxed{\textit{policy}} &= \textit{base policy} \\
&\mid \textit{policy} + \textit{policy} \\
&\mid \textit{policy} \times \textit{policy} \\
&\mid \textit{policy} - \textit{policy} \\
&\mid \text{mean}(\textit{policy}, \textit{policy}) \\
&\mid \min(\textit{policy}, \textit{policy}) \\
&\mid \text{decision}(\textit{point}, \textit{index}, \textit{threshold}, \textit{policy}, \textit{policy}) \\
\textit{base policy} &= \textit{base model}(\textit{point}) \\
&\mid \textit{constant scalar} \in \mathbb{R} \\
\textit{base model} &= \textit{fast base model} \mid \text{KF} \mid \text{MLP} \mid \text{SVM} \\
\textit{fast base model} &= \text{LM} \mid \text{MARS} \mid \text{RF} \\
\textit{point} &= x \mid \textit{constant vector} \\
\textit{index} &= 1 \mid 2 \mid \cdots \mid d \\
\textit{threshold} &= \textit{constant scalar} \in [-1, 1]
\end{aligned}
$$

Figure 3.6: Grammar given in Extended Backus-Naur Form defining the set of valid policy expressions. Terminal symbols are shown in a regular typeface, non-terminal symbols are shown in *italics*. The start symbol of the grammar is marked by a $\boxed{box}$. (based on [107])

Flasch et al. [107] proposed to use genetic programming to automatically build an ensemble, also allowing for multiple operators in an ensemble. For this purpose, a grammar of model ensemble expressions is defined and then searched using genetic programming.

Figure 3.6 shows the grammar that was used for the experiments presented. Using this grammar enables the method to build complex tree structures by combining multiple models using different combination schemes within an ensemble.

3.6 Conclusion

Recapitulating all approaches we considered in this chapter, it can be said that there are many ways to get from a set of models to a single prediction. Moreover, all approaches have their strengths and weaknesses, that have to be considered carefully to choose the most appropriate strategy for a given problem description.

Figure 3.7 combines the approaches considered in the Sections 3.2 through 3.4 in one diagram and gives an overview over the main decisions that have to be taken to specify an ensemble approach. The first of these questions is if a preliminary evaluation of all models on the data should be carried out. This evaluation is typically done by Jackknife, Bootstrap, Cross-Validation or any variations of these. The main characteristic that all of these approaches have in common is that the model is fitted to a part of the data and evaluated on the remaining data, at least once. The more computation time is spent in this step, the more information about the models fit can be gained here.

The next decision that has to be taken is if all models should be fitted to the data or if a single model is chosen for the next prediction. It has to be remembered that the preliminary evaluation step is optional. Thus if the choice of the model is taken now without preliminary evaluation of the models on the data the relation between this choice and the underlying data may be rather sparse or not existing at all. On the other hand, an exhaustive preliminary evaluation of all models allows for a reasonable selection of the best model that is well-founded on the underlying data.

Anyhow, if it is not desired to select a single model at this point, then all models have to be fitted. This step is also not depending on the preliminary evaluation step that might have been performed initially since the models in this step in general only have been trained on parts of the data.

The last decision that has to be taken after training all models to the complete data is if the prediction of one model is chosen or if the predictions of all models are combined into one, presumably more accurate prediction. This decision may depend on whether or not a preliminary fitness evaluation has been carried out or on particular conditions of the problem that has to be modeled. If the preliminary evaluation step has been carried out, it would be an utterly unnecessary step to fit all models to the data and then discard all but the best. Whereas when no preliminary evaluation has been carried out, it may depend on the task if it is best to select one prediction or combine all into one.

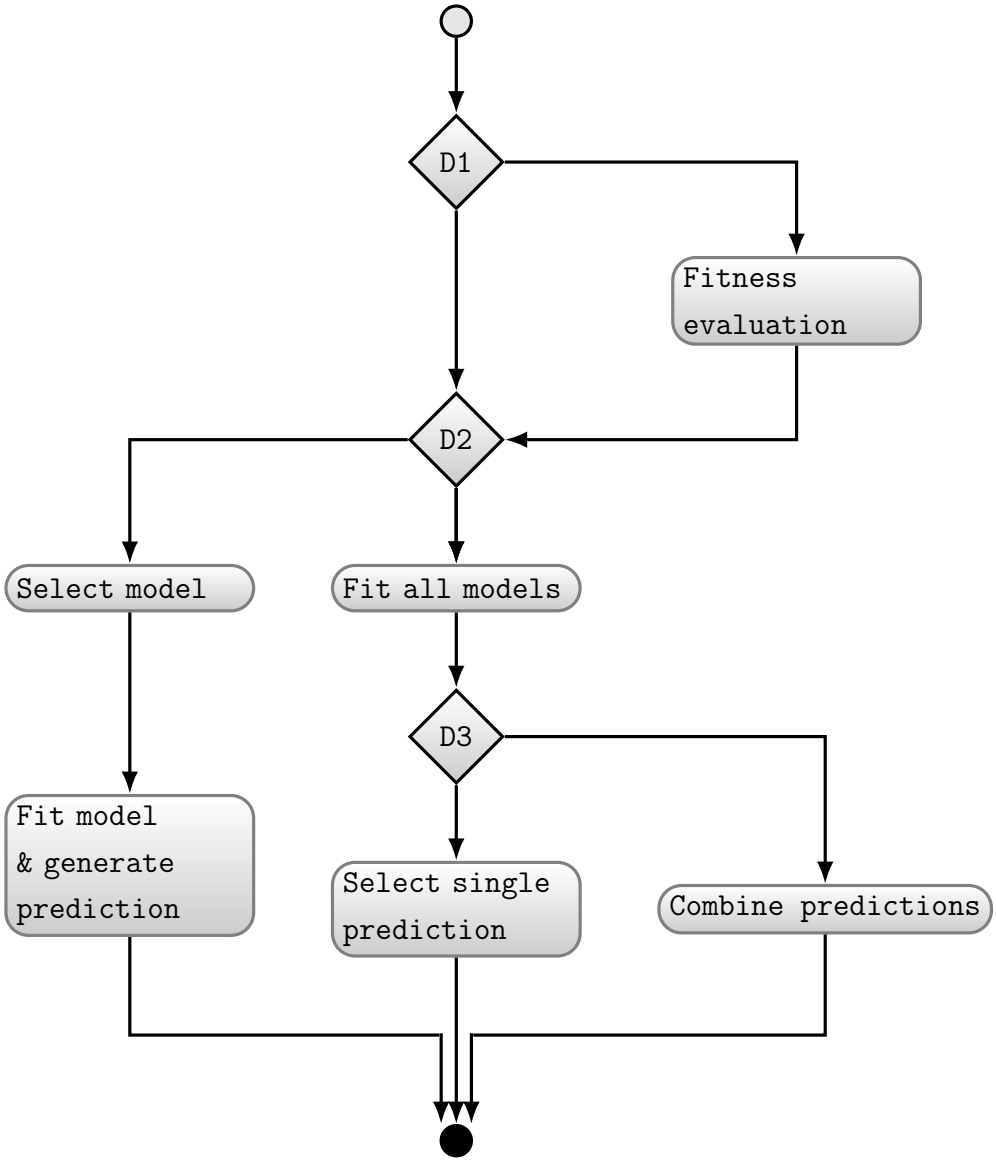

Figure 3.7: Overview over the decisions that have to be taken to define the way how to get a single prediction from a set of models.
D1: Do a fitness evaluation (i.e. cross-validation) on available models?
D2: Select single model or fit all available models?
D3: Select prediction of one model or combine available predictions?

The overall goal of this work is to create a strategy that works reliably and as accurately as possible on arbitrary objective functions well knowingly accepting that this is probably going to happen at the expense of the ensembles compu-

tation time. Thus a method is envisioned, that does an exhaustive preliminary evaluation of all models to gain the best insight into the models' performances, then trains all models on the data to enable the use of the complete knowledge of all models. Still, the method should follow the principle of parsimony and prefer a combination of predictions over a single prediction only if it is clearly beneficial for the overall accuracy. The same applies to the number of models used, it should not be a decision between a single model or a combination of all, but any number of models that seem to be best.