Bayesian Hyperparameter Optimization

The performance of a machine learning method depends on the used data set and the chosen hyperparameter settings for the model training. Finding the optimal hyperparameter settings is crucial for building the best possible model for the data at hand.

The figure below shows a simple 2-dimensional dataset for a regression problem. In this example Polynomial Regression is used to build a predictive model. The model complexity can be determined in advance via the polynomial degree.

……..

To evaluate the performance of the model for various hyperparameter settings a suitable loss function needs to be defined. An often used cost function for regression problems is the Mean Squared Error (MSE):

Loss function: Mean Squared Error

$$L(f, x, y) = (f(x) - y)^2$$

where: $f = A(D)$: function returned by algorithm A on the training set $D = z_1, \ldots, z_n$

y : observed values of the variable being predicted

Figure 1: Loss Function: Mean Squared Error (MSE) - Image by the author

Since the performance of the machine learning estimator depends not only on the hyperparameters but also on which part of the dataset was used as training and validation dataset. In order to obtain a generalised assessment of the performance of the algorithm used, the statistical procedure k-fold cross-validation (CV) is used in the following.

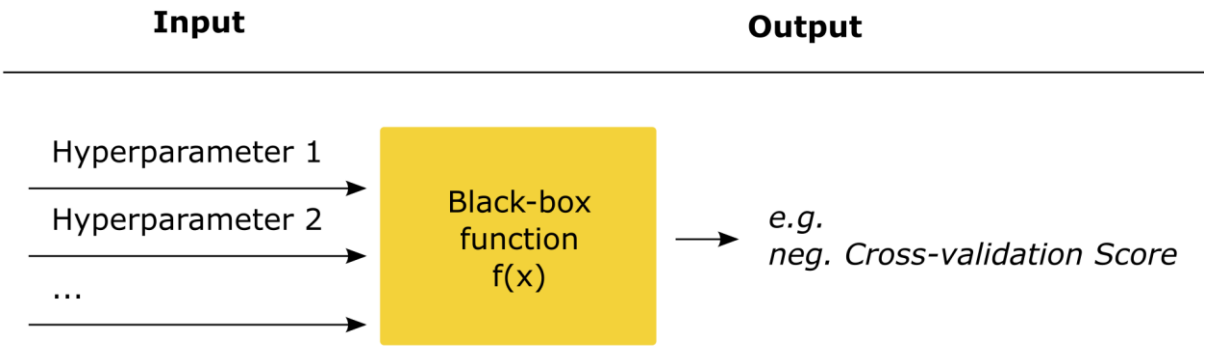
Therefor the data set is split in k subsets. Following k-1 subsets are used as training data set, one for validation. After the model was build, the MSE for the validation data set is calculated. This procedure is repeated until each subset has been used once as a validation data set. The CV score is then calculated as the average of the MSE values.

$$CV(D,\lambda) = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{m}\sum_{z_i \in T_k} L(A_\lambda(D_k), z_i)$$

where:
- K : number of subsets (data set D is chuncked into K disjoint subsets)
- $A(D_k)$: function returned by the algorithm A when trained on the *k*-th subset D_k
- m : size of subset $= n/K$
- λ : setting for hyperparameters $\lambda_1, \ldots, \lambda_t$

Figure 2: K-fold Cross Validation – Image by the author (inspired by [Sic18])

So the target of hyperparameter optimization in this case would be to find the optimal hyperparameter settings, where the Loss (e.g. the CV Score) is minimal. Since the analytical form of the function CV is not given, we speaking about a so called Black Box Function. (Back Box Optimization Problem)

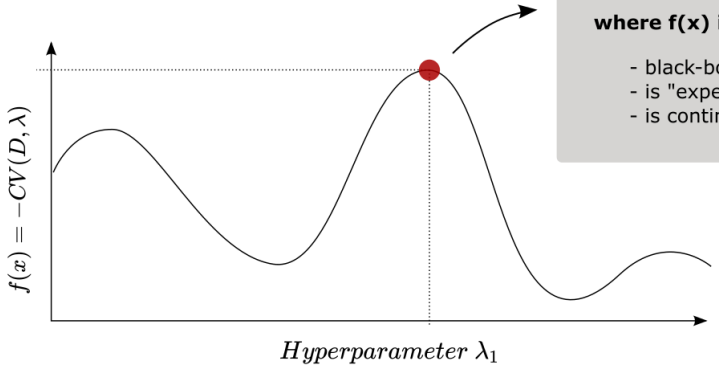


Optimization Problem

$$\max_{x \in A} f(x)$$

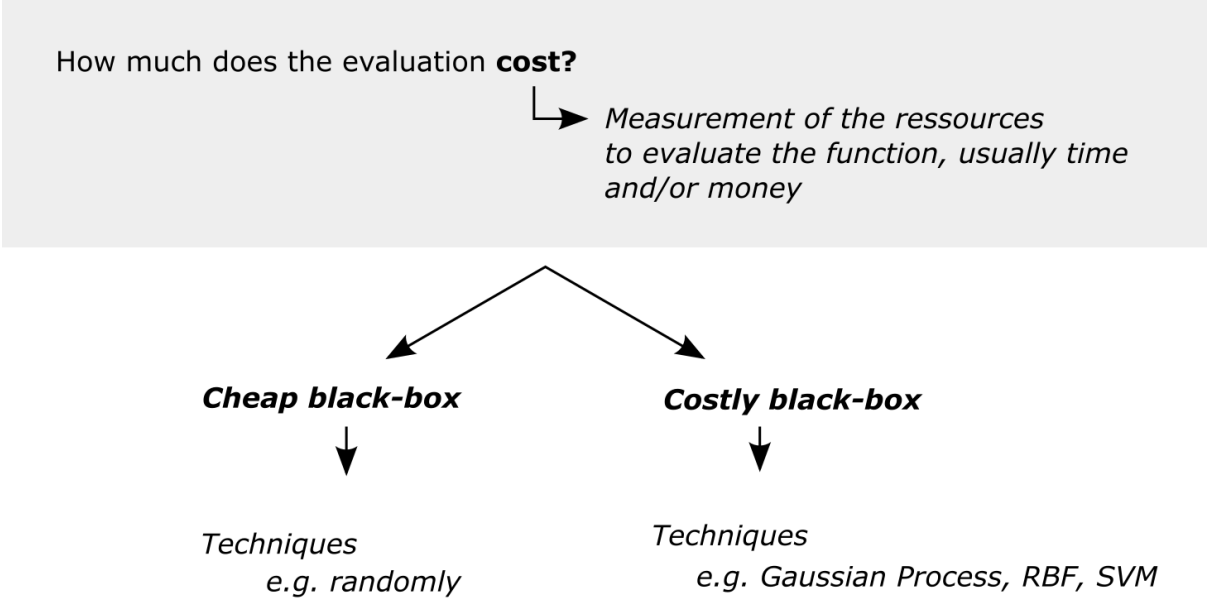
where f(x) is ...

- black-box function, with no closed form nor gradients
- is "expensive to avaluate"
- is continious



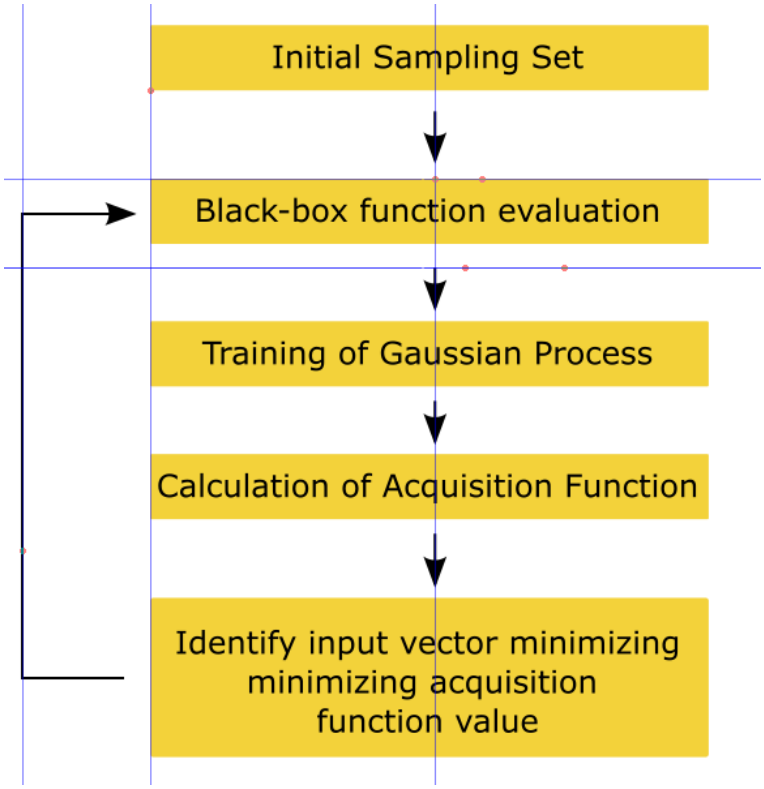
What we could do, is just calculate the value of f(x) in a for-loop to evaluate every position in the defined hyperparameter space. Afterwards we simply identify the Hyperparameter combination with the optimal value (also known as Grid Search).

Definitely a valid approach, at least for so called “cheap” black-box function, where the computation effort to calculate the CV values is low. But what if the evaluation of the function pretty costly (so the computational time and/or cost to calculate CV is high)? In this case it may makes sense to think about more “intelligent” ways to find the optimal value. [Cas13]



When speaking about costly black-box functions, the outcome of interest is expensive or time-consuming to calculate, a “cheaper” surrogate function could help. The Surrogate Function should approximates the black-box function f(x) [Cas13]. In total, the time needed to compute the needed sample values and the surrogate function, should be less time-consuming than calculating each point in the hyperparameter space. To model the surrogate function, a wide range of machine learning techniques is used, like Polynomial Regression, Support Vector Machine, Neuronal Nets and probably the most popular, the Gaussian Process (GP).

For the above regression problem, the following black-box function results. In order to be able to map the function with sufficient accuracy for the defined hyperparameter space, this range must be appropriately fine-granularly ebased. In this case we assume a predefined hyperparameter space (polynomial degree = 1 - 100). Since the polynomial degree can only assume integer values, there are 30 cross validations to be carried out for this delimited range.



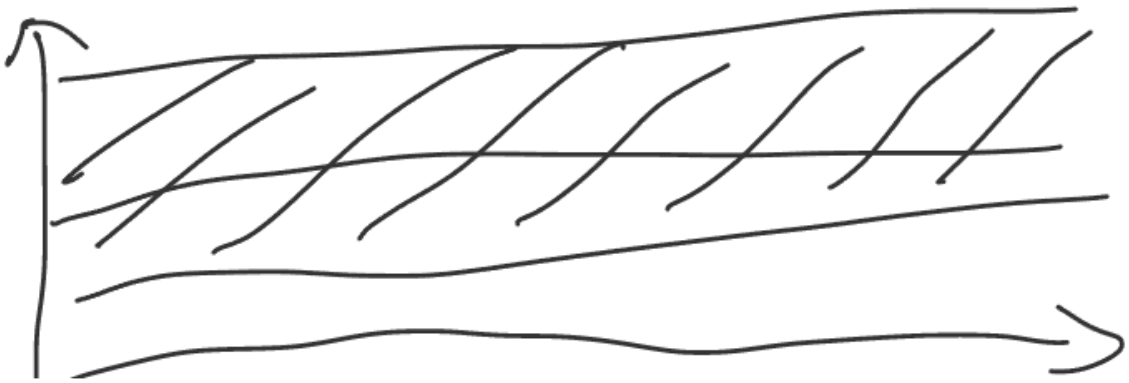
Surrogate Function

To reduce the number of necessary data points, we try to find a suitable surrogate function with few data points that approximates the actual course of our black-box function f(x) = CV(lambda). The best-known surrogate function in the context of hyperparameter optimisation is the Gaussian process, or more precisely the Gaussian process regression. A more detailed explanation of how the Gaussian Process Regression works can be found in "Gaussian Processes for Machine Learning" by Carl Edward Rasmussen and Christopher K. I. Williams, which is available for free at:
http://www.gaussianprocess.org/gpml/chapters/

You can also find an explanation of Gauss Process Regression in one of my recent articles:

https://towardsdatascience.com/7-of-the-most-commonly-used-regression-algorithms-and-how-to-choose-the-right-one-fc3c8890f9e3

In short, Gaussian process regression defines a priori Gaussian process that already includes prior knowledge of the true function. Since we usually have no knowledge about the true course of our black box function, a constant function with some covariance is usually freely chosen as the Priori Gauss.



By knowing individual data points of the true function, the possible course of the function is gradually narrowed down.

Acquisition Function

Die Surrogate Funktion wird nach jedem Berechnungsschritt erneut berechnet und dient als Grundlage für die Wahl des nächsten Berechnungsschrittes. Hierfür wird eine Aquisitions Funktion eingeführt. Die wohl populärste Aquisitions Funktion im Kontext der Hpyerparameter Optimieriung ist der Informationsgewinn.

References

[Cas13] Cassilo, Andrea. A Tutorial on Black–Box Optimization. https://www.lix.polytechnique.fr/~dambrosio/blackbox_material/Cassioli_1.pdf. 2013.

[Sci18] Sicotte, Xavier. Cross validation estimator. https://stats.stackexchange.com/questions/365224/cross-validation-and-confidence-interval-of-the-true-error/365231#365231 . 2018