Hochschule Darmstadt

– Fachbereich Mathematik & Naturwissenschaften–

Extending Bayesian Optimization to non-Classical Problems Selected from Industry.

Abschlussarbeit zur Erlangung des akademischen Grades Master of Science (M.Sc) im Studiengang Data Science

vorgelegt von

Betül Cetinkaya

Matrikelnummer: 763552

Referent	:	Prof. Dr. Tobias Bedenk
Korreferent	:	Prof. Dr. Kilian Schwarz
Supervisor	:	Sascha Desch

Angemeldet am:	15.11.2021
Abgegeben am:	02.05.2022

DECLARATION

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kentlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, May 2, 2022

Betül Cetinkaya

ABSTRACT

In the industry, categorical, numeric discrete and non-controllable inputs occur which can not be applied to Bayesian Optimization. But in order to apply this method in industry, this thesis describes extending methods for modelling categorical, numeric discrete and non-controllable inputs. Therefore, one method for each non-conventional attribute was worked out in order to extend the Bayesian Optimization by them. Furthermore, the investigated extensions were combined to test their functionality in combination. This was done by applying the extensions to simulated functions. Additionally to noise-free data, some examples were applied to functions with noisy observations in order to show how both data are modelled. Because noise in industry is a known problem, this investigation is a relevant one. For numeric discrete as well as non-controllable inputs, effective solutions were found. However, the selected solution for categorical values performed also good, but depends strongly on initial weights which were set randomly. So this method needs to be improved. Finally, the Bayesian Optimization was extended by all three methods and applied to the three inputs categorical, numeric discrete and non-controllable. It was shown, that the problems the Bayesian Optimization has in dealing with the selected non-conventional data can perform good.

ZUSAMMENFASSUNG

In der Industrie kommen oftmals kategorielle, diskrete oder nicht- -steuerbare Einflussgrößen vor, mit denen Bayesianische Optimierungsmethoden nicht zurecht kommen. Da jedoch in der Industrie die Bayesianische Optimierungsmethoden angewendet werden soll, beschäftigt sich diese Arbeit mit Erweiterungsmethoden für diese. Die Erweiterungsmethoden sollen die Verfahren anpassen, um auch mit den unkonventionellen, für die Industrie relevanten Parametern arbeiten zu können. Hierfür wurden verschiedene Methoden zur Erweiterung der Bayesianischen Optimierung angewendet und miteinander kombiniert. Durch Simulationen wurde die Funktionsweise der Erweiterungen geprüft. Zusätzlich wurden einige Simulationsläufe mit Rauschen versehen, um das Verhalten der Verfahren auf Daten mit Rauschanteil zu analysieren. Da Rauschen aufgrund von Messungenauigkeiten in der Industrie ein bekanntes Problem ist, ist die Untersuchung des Verfahrens unter dem Einfluss von Rauschen von hohem Interesse. Es konnten Ansätze für numerisch diskret sowie nicht-steuerbare Einflussgrößen gefunden werden. Für kategorielle Werte wurde ein vielversprechender Ansatz untersucht, der jedoch noch Verbesserungspotential besitzt, da dieses Verfahren stark von initialen Gewichten abhängt, die zufällig gewählt werden. Schlussendlich wurden alle drei Ansätze miteinander verknüpft und es konnte gezeigt werden, dass die Probleme des Verfahrens durch Kombination aller drei Anpassungen verschwinden können.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

BO Bayesian Optimization

GP Gaussian Process

ACQ Acquisition Function

RBF Radial Basis Function

OHV One Hot Vector

C-UCB Context Upper Confidence Bound

MATHEMATICAL NOTATION

l	Length scale of the Radial Basis Function.
λ	Output scale of the Radial Basis Function.
ϵ_n	Random variable with Gaussian distribution representing the noise in the observed data.
$\sigma^2_{\epsilon_n}$	Variance of ϵ_n.
$f(x_i)$	Objective function value for x_i.
g	Function sample or realization of the Gaussian process.
x	Set of variables to be evaluated by the Bayesian Optimization.
$\mathcal{X}$	Domain of x.
N	Dimension of $\mathcal{X}$ or number of variables.
$\mathcal{X}_f$	finite subset of $\mathcal{X}$.
$\mathcal{X}_d$	Domain of discrete Variable within $\mathcal{X}$.
x_i	i-th value of x.
y_i	Noisy observation of $f(x_i)$.
$\mathcal{D}$	Set of observed train data.
$m(x_i)$	Mean function value of the Gaussian process for x_i.
$\Sigma(x_i, x_j)$	Variance function value of the Gaussian process for x_i, x_j.
$K(\cdot, \cdot)$	Covariance Function of the Gaussian Process.
μ_i	Expected value of the Gaussian Process at x_i.
σ_i	Standard deviation or uncertainty of the Gaussian process at x_i.
$\boldsymbol{\mu}$	Mean vector consisting of $\mathcal{X}_f$.
$\boldsymbol{\sigma}$	Standard deviation vector consisting of $\mathcal{X}_f$.
$d(x_i, x_j)$	Distance measure for the inputs x_i, x_j.
$d_e(x_i, x_j)$	Euclidean distance measure for the inputs x_i, x_j.
c	categorical variable.
c_i	i-th category of c.
$\alpha(x_i)$	Acquisition Function value of x_i.
β	Exploration parameter of the Upper Confidence Bound.
x^*	By α proposed sample regarding maximum acquisition.

1 MOTIVATION

The process of digitalization opens a tremendous number of opportunities in manufacturing industry, be it the resulting available data or the evoking field of Machine Learning [22]. In order to apply Machine Learning in industry, SCHULZ Systemtechnik GmbH plans to develop an optimization tool for industry. The optimized target can be cost, time or energy consumption during production. Especially energy consumption is very interesting in a time where sustainability and climate change should be taken seriously, see [10]. In order to optimize the target metric, the machine and process configuration parameters are evaluated by observing the corresponding target value. By doing so, finding the configuration which leads to the best target value is desired.

The selected method for this project is the Bayesian Optimization (BO). Investigating the suitability of this method for the selected task was not part of this thesis. However, according to [3] this method was used often for engineering systems and therefore applied in physical world. A major problem in applying an optimization method in industry is the cost of evaluation. When a proposed configuration is evaluated, this can lead to high costs, either because of the time involved or to obtain a poor target value that is nevertheless informative. In order to avoid waste of resource, the number of evaluations should be low and an intelligent sampling is important. BO is one method that takes a low evaluation number into account [3]. The first master thesis in this project was written by Nathan Wollek. He described the concept of applying BO in industrial plants and worked out diverse ideas and opportunities, like amortization calculation or visualization and comprehensibility[24].

This thesis shows methods which help the BO in modelling parameter properties which are called *non-classical*. Note that this term is not necessarily used in this context in literature.

BO basically consists of two components: a surrogate model and an Acquisition Function (ACQ). Using past evaluations, the surrogate model is build to extra- and interpolate the target metric in the entire configuration space to be optimized. It gives prediction about the unknown target metric and takes uncertainty of the prediction into account. Usually a Gaussian Process (GP) is used as surrogate model. Classically, the GP is applied to continuous inputs and can not handle other types of val-

ues, like numerical discrete. The ACQ on the other hand, uses the GP's values, to select the next promising machine configuration to be evaluated. In doing so, a problem of maximization or minimization is addressed. Classically, the ACQ is used for only controllable inputs. However, there are also non-controllable influences or disturbances which may have an influence on the target metric, like the outdoor temperature. When choosing the next evaluation, the strategy of exploration or exploitation can be applied. For example, if there are too few observations in the configuration space, exploration may be appropriate and vice versa. In this way, an intelligent sampling is performed. Consider a worker with expert knowledge in an industrial plant where the BO is applied and visualized. Interaction between worker and ACQ could help minimizing the number of observations, if the worker selects a useful subarea where to search next by applying his expert knowledge. Moreover, the BO can be used to visualize the influences of different configurations observed so far and expert knowledge can be preserved and transferred to others. This way, a human-supervised semi-automatic optimization can be performed in addition to automatic optimization, when the worker interacts with the ACQ, see [24].

Non-classical configuration properties were examined theoretically to define the main task of this thesis. Non-classical properties are those, which are either non-continuous, e.g. a level based parameter[1], or non-controllable, e.g. environmental temperature or humidity[2], in manufacturing processes. The BO has problems when being applied to non-continuous as well as non-controllable parameters. Therefore, three non-classical properties were selected and solution approaches were examined. The solution approach extend the BO in order to make it deal with the non-classical properties. Note that the GP classically models continuous target metrics. However, there are also other types of target metrics than continuous valued such as integer valued, e.g. number of discard. In this thesis, only continuous function values were considered. In table 1.1 non-classical properties are listed. The rows show the different properties of the parameters and the columns separate them into *controllable* and *non-controllable*. Examples are presented for most cells, e.g. a continuous and controllable configuration parameter is any machine parameter that takes continuous values (such as pressure) and is controllable by humans. An example for a continuous and non-controllable parameter is the outdoor temperature, which is observable regarding its current continuous value but is not subject to human influence. An example for categorical and non-controllable parameter is the restriction to a certain resource during production, e.g. a customer, who ordered a muesli product, wants it to be produced with rye. The properties *continuous*, *numerical discrete* and *categorical* re-

1 https://worldofinstrumentation.com/process-parameters-that-commonly-measured-in-industry/, last visited on 27.04.2022 at 14:23.

fer to the type of parameter values. The last two properties *delayed* and *trivial* describe parameters which have a delay between realization of configuration and reaching the configured value. An example for this is heating temperature. Consider one of a plant's stations heats the produced good. Then regulating the heating temperature takes some time until the desired heating value is reached. Trivial parameters on the other hand are those, which have no influence on the optimized target metric. For some cells, no examples could be found. Of course a trivial parameter, which is non-controllable, needs no consideration. For delayed and non-controllable parameters no example could be figured out. Note that the current outdoor temperature is observable and therefore not delayed.

property	**controllable**	**non-controllable**
continuous	speed of conveyor belt with continuous values, e.g. $1, 1.1, 1.01, ...[m/s]$	(constant) outdoor temperature
numerical discrete	# workers, # machines, level-based parameter values, e.g.: $0.0, 0.5, 1., ...$	# sick workers # broken machines
categorical	used material for conveyor belt, such as plastic or metall, diameter of a roller, e.g. to flatten flakes	restriction to a certain grain type when producing muesli, or to the available space (#machines can not infinitely increased)
delayed	heating/ cooling temperature	delayed arrived workers, dynamic change of outdoor temperature, e.g. sun rise, sudden and short rain in summer
trivial	energy supplier has no effect on discard	——

Table 1.1: List of examples for different input properties regarding controllable and non-controllable inputs.

In this thesis no data from physical world was used. Instead, simulated objective functions were applied and the BO was investigated on them. Furthermore, testing the investigated extensions here would be too expensive when being applied to real data. Simulations give far more flexibility when examining the extended BO. The selected parameter properties were selected from table 1.1 regarding their high number of examples found in theory. These are:

- numerical discrete & controllable
- categorical & controllable
- continuous & non-controllable.

The non-classical properties were underlined. As mentioned before, the classical GP can not model non-continuous values and the classical ACQ can not be applied to non-controllable inputs. The selected problems address both restrictions of them. The term *classical Bayesian Optimization* is referred to a BO containing a classical GP, which treats all inputs as continuous values, and a classical ACQ, which treats all inputs as controllable. The research question is:

How can the classical Bayesian Optimization be extended, to be applied to numerical discrete, categorical and continuous non-controllable inputs?

In order to investigate the research question, the Python library pyro[2] was used. In this library the GP as described in [23] is implemented. This implementation was used and further developed.

The structure of this thesis is as follows: First, in chapter 2 the theoretical basics are explained including basic terms. When describing the GP, [23] was used. Then, the ACQ is introduced and the usage of both when applying the BO. Furthermore, a transfer solution is introduced, which was used to model categorical values by the GP. The transfer solution is applied in the field of *Natural Language Processing*, where texts are analyzed based on the co-occurence of words or word sequences. Then, in chapter 3 the used methodology is described. and in chapter 4 the execution of this. Afterwards, the results are discussed in chapter 5 and the outlook described in chapter 6.

2 For the detailed documentation of pyro, see https://docs.pyro.ai/en/stable/, last visited on 26.04.2022 at 14:04.

2

THEORETICAL FOUNDATIONS

This chapter explains the theoretical background necessary for understanding the work described in this thesis. First, some basic terms are introduced in section 2.1. Afterwards, the Gaussian Process (GP) and Acquisition Function (ACQ) are described in sections 2.2 and 2.3 respectively. Then in section 2.4 the Bayesian Optimization (BO) is introduced. Finally, a method used as transfer solution for categorical values is introduced. With this method, the categorical values were mapped to a numerical space and afterwards applied to a distance measure. For a definition of distance measures, see [18]. The description for this is in section 2.5. Note that in this chapter only equations that have been referenced are numbered and the others are not.

2.1 BASIC TERMS

PARAMETER In this thesis, the model parameters of the Gaussian Process are called *parameter*. These are length scale l, output scale λ and the assumed noise variance $\hat{\sigma}_n^2$. In some literature the output scale is called signal variance and is symbolized as σ^2 [23]. For a better distinction between output scale and the assumed or true noise variance, the notation λ was adapted from [5].

Note that the output scale equals the variance for $x = x'$: $RBF(x, x') = \sigma^2$.

INPUT The simulations for machine and process configuration parameters are called *inputs* in this thesis, to distinguish them from the model parameters. Inputs include controllable and non-controllable influences which are considered and adjusted during the optimization process.

VARIABLE *Variables* are inputs that are controllable by humans. They can be continuous as well as numerical and categorical discrete. *Controllable input* is a synonym for *variable*.

CONTEXT *Context* is a type of inputs, just like variables. But other than variables, context include the inputs which are not controllable. *Non-controllable input* is a synonym for *context*.

CLASSICAL BAYESIAN OPTIMIZATION In this thesis, a Bayesian Optimization which can only handle classical inputs, is called *classical Bayesian Optimization*. When such a model is applied to non-classical inputs, varies problems occur.

NON-CLASSICAL INPUTS In this thesis, the term *non-classical* refers to numerical discrete, categorical and non-controllable inputs.

2.2 GAUSSIAN PROCESS

The GP was originally developed for geological applications, e.g. when an interpolation between spatial observations was needed[17]. An example for this is mining gold. If a gold miner finds gold at location x_0, digging next to this location results very likely in finding gold again. The further away the gold miner digs from x_0, the more the uncertainty of his success grows. Usually, the GP is used for approximation of an unknown objective function f. For this section [23] was used. If no reference is made, the information was taken from this book, and from the cited reference otherwise.

A GP is a multivariate normal distribution over functions G in a certain domain $\mathcal{X} \subset \mathbb{R}^N$, meaning one realization of it corresponds to a function g in domain $\mathcal{X}$.[17]

$$G \sim GP$$
$$g : \mathcal{X} \to \mathbb{R}$$

Definition 1 *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*[23]

At every single point $x_i \in \mathcal{X}$, the GP consists of a Gaussian distribution representing the expected value μ_i and variance σ_i^2 or standard deviation σ_i:[17]

$$GP(x_i) \sim N(\mu_i, \sigma_i^2).$$

In order to estimate the objective function f, μ_i represents the prediction at x_i and the variance corresponds to how certain the prediction is. The smaller the variance the higher the certainty about the prediction and vice versa. However, the mathematical definition of the GP does not consist of infinite Gaussian distributions over its domain, but of two functions used to calculate the mean and variance for each point[17, 23]:

$$G \sim GP\left(m(\cdot), \Sigma(\cdot,\cdot)\right).$$

m is the mean function and Σ the variance function:

$$m : \mathcal{X} \to \mathbb{R}$$
$$\Sigma : \mathcal{X}^2 \to \mathbb{R}.$$

The mean function represents the expected value of the GP at any given point x_i:

$$m(x_i) = \mathbb{E}[G(x_i)] = \mu_i.$$

In order to implement a mean vector programmatically, a finite subset was passed to the mean function to obtain the mean vector. Therefore, $\boldsymbol{\mu}$ is the mean vector calculated from a finite subset of the domain $\mathcal{X}_f \subset \mathcal{X}$ with $x_{fi} \in \mathcal{X}_f$:

$$\boldsymbol{\mu} = [m(x_{f1}), m(x_{f2}), ...] = [\mu_1, \mu_2, ...].$$

The variance function Σ defines the variation at a given point and $\boldsymbol{\sigma}$ is the corresponding standard deviation vector calculated from $\mathcal{X}_f$:

$$\boldsymbol{\sigma} = [\Sigma(x_{f1}), \Sigma(x_{f2}), ...] = [\sigma_1, \sigma_2, ...].$$

Both, m and Σ are calculated via K, the covariance function, which gives information about the similarity between two points $x, x' \in \mathcal{X}$[23]:

$$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}.$$

For a detailed description about how m and Σ are calculated and the functionality of Bayesian inference, see [5, 23]. The similarity between two points $x, x' \in \mathcal{X}$ is used to assume the corresponding function values $f(x)$,$f(x')$. The general assumption is: the smaller the distance between x and x', the more likely the similarity between $f(x)$ and $f(x')$:

$$d(x, x') = 0 \Rightarrow d(f(x), f(x')) = 0 \tag{2.1}$$

$$d(x, x') = \delta_0 \Rightarrow d(f(x), f(x')) \approx \delta_0 \tag{2.2}$$

with δ_0 as some small positive number near zero and d as a metric distance measure. The higher the distance, the more uncertain the assumption about the function values of x and x' become. Note that this does not leat to a proportional relation between $d(x, x')$ and $d(f(x), f(x'))$. Proportionality would mean, that if the distance between the inputs increases, the distance between their function values also grows. But inputs with high distance may result in similar function values with low difference. Only the uncertainty grows with increased input distance. The approximation symbol in equation 2.2 symbolizes this uncertainty.

In figure 2.1 20 samples (dotted lines) of a GP are shown with their corresponding mean (blue line) and standard deviation (blue area). The vertical black line marks the test point $x = 4$ where the Gaussian distribution is defined as:

$$f_{\text{prior}}(4) \sim \mathcal{N}(0, 1^2).$$

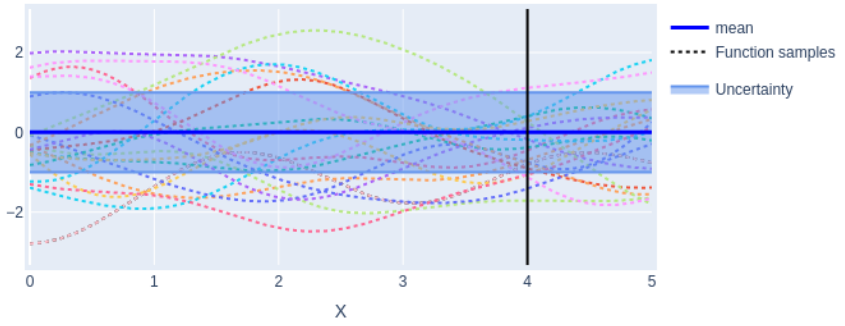


Figure 2.1: 20 function samples from prior of a Gaussian Process before observing data. The code for this figure was motivated by https://scikit-learn.org/stable/auto_examples/gaussian_process/plot_gpr_prior_posterior.html last visited on 21.03.2022 at 12:49.

A look at the other values for x shows, this predicted distribution is everywhere the same. The further away an observation is located from a test point, the more less this data influences the test point's prediction. In case no information is represented via influence, the GP predicts the corresponding test points by using the default mean value. The same behavior is observed when no information about f is given. Because in figure 2.1 no observation was made, the GP in this plot is called *prior distribution*, *prior* for short. After observing *new* data and updating the GP, the *posterior distribution*, *posterior* for short, is obtained[23].
In figure 2.2 the posterior of the GP is shown after observing the function value $f(x = 3.45) \approx 0.8$. It is noticeable that the Gaussian distribution at $x = 4$ in the posterior is different than in the prior before:

$$f_{\text{posterior}}(4) \sim \mathcal{N}(0.7, 0.4^2).$$

In the posterior, the uncertainty for the test point is reduced. Due to its distance to the observed sample, there is an influence of the observed information weighted by the distance to the observed point. Due to different causes, an observation can be noisy. An example for this kind of cause are measurement inaccuracies of input values and therefore an inaccurate observation of target values, see equation 2.3:

After observing one sample, the noise in the posterior is very small and may increase after more observations.

$$y_i = f(x_i) + \epsilon_n \tag{2.3}$$

with $\epsilon_n \sim \mathcal{N}(0, \sigma_n^2)$ being independent identically distributed for all $x_i \in \mathcal{X}$[23]. Note that $\hat{\sigma}_n^2$ is one of the parameters of the BO and estimates the true noise σ_n^2 in the data. The closer the test point is to the training point, the higher the influence of its covariance and the lower the uncertainty about the prediction at the test point. Note that due to

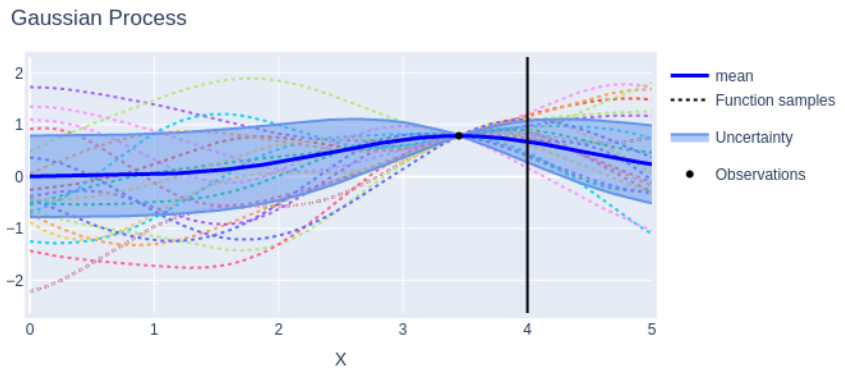


Figure 2.2: 20 function samples from posterior of a Gaussian Process after observing data. The code for this figure was motivated by https://scikit-learn.org/stable/auto_examples/gaussian_process/plot_gpr_prior_posterior.html last visited on 21.03.2022 at 12:49.

this noise in the physical world, the equation 2.1 has to be updated, see equation 2.4.

$$d(x, x') = 0 \Rightarrow d(f(x), f(x')) \approx 0. \tag{2.4}$$

When more than one training point is observed, their covariances have a weighted influence to the estimation about every unobserved test point in the domain, depending on their distances[5, 17, 23]. Due to [21], this is also called *Kernel Density Estimator*.

GP is a supervised learning method[23], where labeled data is used for generating the GP and predicting the objective function f. Labeled data consists of the input and its corresponding output. In manufacturing processes inaccurate measurements caused by "mechanical vibrations and electronic signals"[16] result in noisy data. The formula for noisy observations was shown in equation 2.3. Therefore, observed data or training data $\mathcal{D}$ is defined as: (x_i, y_i) (in figure 2.2 the training data is $\mathcal{D} = \{(3.45, 0.79)\}$). Furthermore, the GP is not based on gradient information of the objective function [5]. This has the advantage that also non-differentiable objective functions can be modeled.

Covariance Function

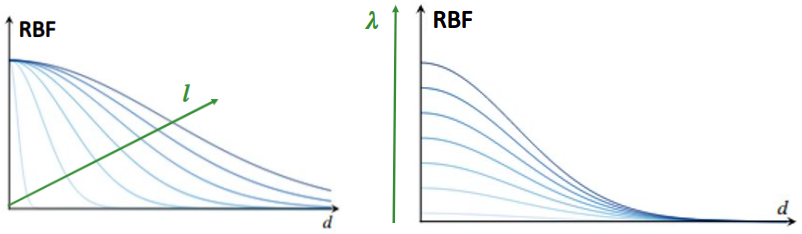
The Radial Basis Function (RBF) (also known as *Squared Exponential* covariance function) was used as covariance function[23]:

$$RBF(x, x') = \lambda^2 \exp\left(-\frac{d_e(x, x')}{2l^2}\right)$$

with $l > 0$ as the length scale, $\lambda > 0$. Therefore, the covariance function takes only positive values. In the simulations, d_e was used as distance measure and is defined as follows [12]:

$$d_e(x, x') = \|x - x'\| = \sqrt{\sum_{j=1}^{N} (x_j - x'_j)^2}.$$

N corresponds to the domain's dimension: $\mathcal{X} \subset \mathbb{R}^N$. The length scale l is also called *smoothing parameter*, *window size* or *bandwidth*, see [21]. It defines the smoothness of the weighted influences between observations and test points. Due to [23] the smoothness behaves as follows: the higher l, the smaller the distance radius in which the influence of a sample on its neighbors is still significant; the lower l, the wider this radius is. In figure 2.3a this behavior was illustrated. The output scale defines the strength of the influence between one observation and its neighbors: the higher λ, the higher the influence[5], see figure 2.3b. Therefore, the length scale defines the reach in which an influence is given and the output scale defines this influence's strength.



(a) (b)

Figure 2.3: Illustration of the Radial Basis Function as a function of the difference d with increasing length scales in panel (a) and increasing output scales in panel (b). Both figures were taken from [5] and slightly modificated.

The RBF-values depend on the euclidean distance $\|x-x'\|$ and decreases monotonically. Covariance functions which depend on the euclidean distance are isotropic and have identical influence in all directions. If in the real world, this assumption is false, an anisotropic covariance function can be applied; an example for an anisotropic kernel is one which has a different length scale parameter for each dimension.[5, 23]
The used kernel was an RBF with one length scale for all dimensions. The parameters l, λ and $\hat{\sigma}_n^2$ were estimated by maximizing the a posterior probability of the GP $p(l, \lambda, \hat{\sigma}_n^2 | \mathcal{D})$, which is sometimes referred

to as the *marginal likelihood* [13, 23]. In order to estimate the model parameters, a plausible belief about them can be introduced by defining a prior distribution. The prior is flat if no belief is given and the parameters are calculated via *maximum likelihood estimation* [5]. However, in the pyro implementation the parameters were trained by applying *Stochastical Variational Inference* where the ELBO-function is used as loss function for the gradient descent algorithm[1], see section 2.5.

In figure 2.4 the described structure of the GP is visualized.

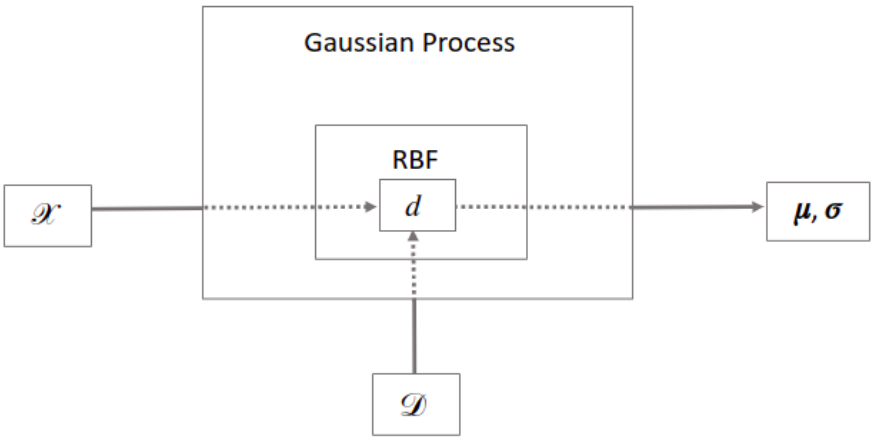


Figure 2.4: Structure of the Gaussian Process.

2.3 ACQUISITION FUNCTION

The ACQ provides suggestions where the next sample should be taken. In this thesis, only maximization problems are considered. Two strategies can usually be pursued: exploration and exploitation. Exploration involves sampling to obtain information in areas where few or no points have been observed. Exploitation corresponds to a sampling of the best observed function value observed so far. The highest function value is the best, because only maximization is considered. This behavior is also described as *greedy*[20]. The ACQ α takes the posterior GP's mean and uncertainty vectors $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ and assigns an acquisition value to each point in the domain $\mathcal{X}$, or due to a finite set $\mathcal{X}_f$:[5]

$$\alpha : \mathcal{X}_f | \boldsymbol{\mu}, \boldsymbol{\sigma} \Rightarrow \mathbb{R}$$

These values reflect the preference over locations in the domain. In case of maximization problem, a value x is preferred to another point x' whenever $\alpha(x) > \alpha(x')$. Therefore, the point x^* with the maximum acquisition value is suggested as the best one leading to either maximal

1 https://pyro.ai/examples/svi_part_i.html, last visited on 28.04.2022 at 13:24.

information gain in case of exploration or best known function value in case of exploitation[5]:

$$x^* = \arg\max_{x \in \mathcal{X}} \{\alpha(x)\}\,.$$

In case of a minimization problem, the ACQ is minimized. Alternatively, the ACQ is maximized after changing the sign of its values. Numerous definitions for the ACQ have been proposed, like Expected Improvement, Probability of Improvement and Thompson Sampling[5]. Because of its simple interpretability, the *Upper Confidence Bound*, *UCB* for short, was used in this work, see equation 2.5.

$$\mathrm{UCB}(x_i) = \mu_i + \beta\sigma_i \tag{2.5}$$

With $\beta \geq 0$ as exploration parameter: the higher β, the more exploration is done and vice versa. In case this parameter is set high, the uncertainties are stronger weighted. In case $\beta = 0$, only the mean values are considered. In figure 2.5 the posterior GP is shown with the corresponding ACQ. The red triangle in the ACQ plot marks the position of the next sample x^*.

Note that the interpretation of exploitation may differ dependent on the selected acquisition function.

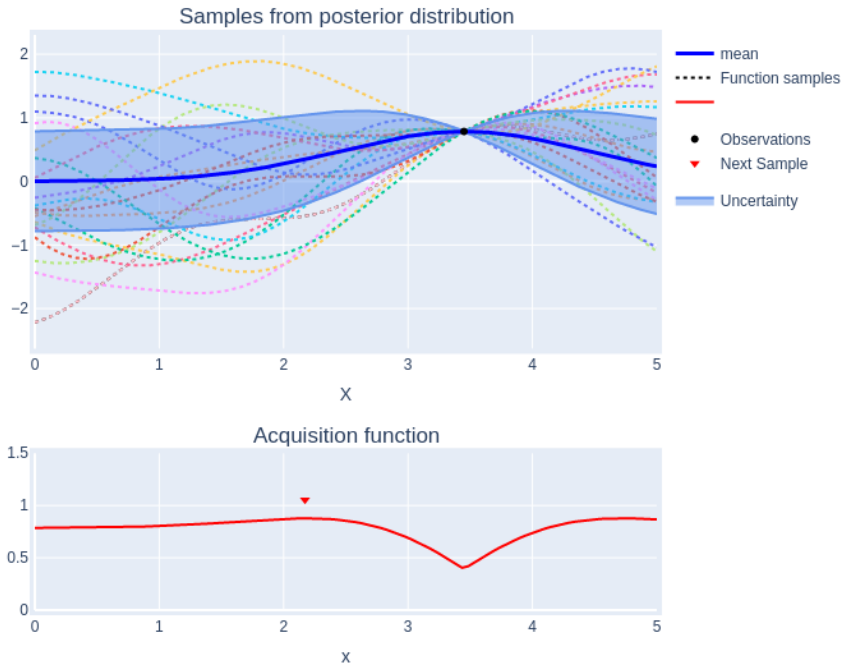


Figure 2.5: Illustration of Posterior Gaussian Process with the corresponding Acquisition Function.

The ACQ and the GP are cheap to evaluate, compared to the objective function. Both are generated computationally instead of applying ex-

pensive to evaluate data. This way, a difficult expensive problem is reduced to a series of simpler and inexpensive problems.[5]

In figure 2.6 the described structure of the ACQ is visualized.

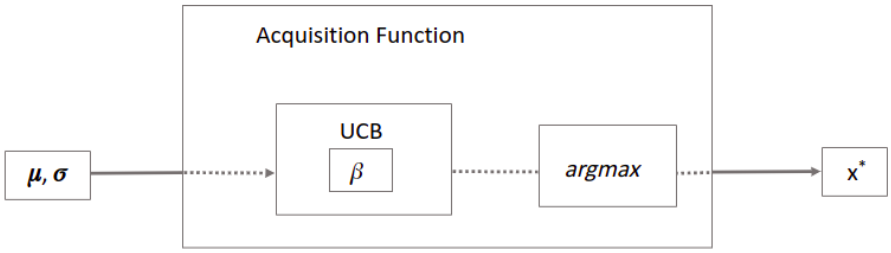


Figure 2.6: Structure of the Acquisition Function used.

2.4 BAYESIAN OPTIMIZATION

The aim of the Bayesian Optimization is not to approximate the objective but to find its optimum by approximating length scale.

The BO is a method which consists of the two components: surrogate model (here: Gaussian Process) and Acquisition Function. First, the prior of the GP is given. Its outputs $\boldsymbol{\mu}, \boldsymbol{\sigma}$ are used by the ACQ to select the next proposed sample x^*. Second, the function value $f(x^*) = y^*$ is observed and $\mathcal{D}$ gets updated due to this new observation. The GP is updated to the posterior, considering the new observation made. In the next iteration, the posterior GP is considered as prior and so on. The GP approximates important information of the physical world, where data is observed. In figure 2.7 this cycle is illustrated.

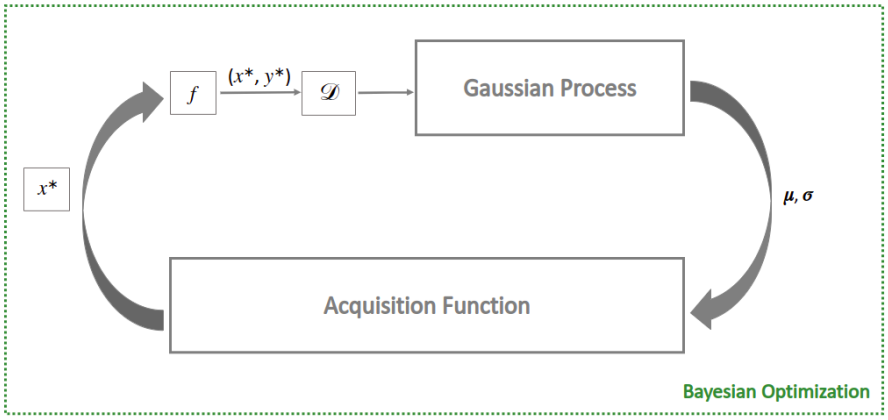


Figure 2.7: Structure of the Bayesian Optimization method.

2.5 WORD EMBEDDING

Embedding means mapping values from one space into another, called embedding space u. The main motivations for doing this are dimension reduction and metric representation.[7] In this work, Embedding was

used to transform categorical values into a continuous metric space, where similar categories are close to each other and vice versa[1], so that the euclidean distance can be applied. The transformation to the u-space was mainly motivated from *Word Embedding* used in *Natural Language Processing*. Therefore, this solution is a transfer solution.[1]

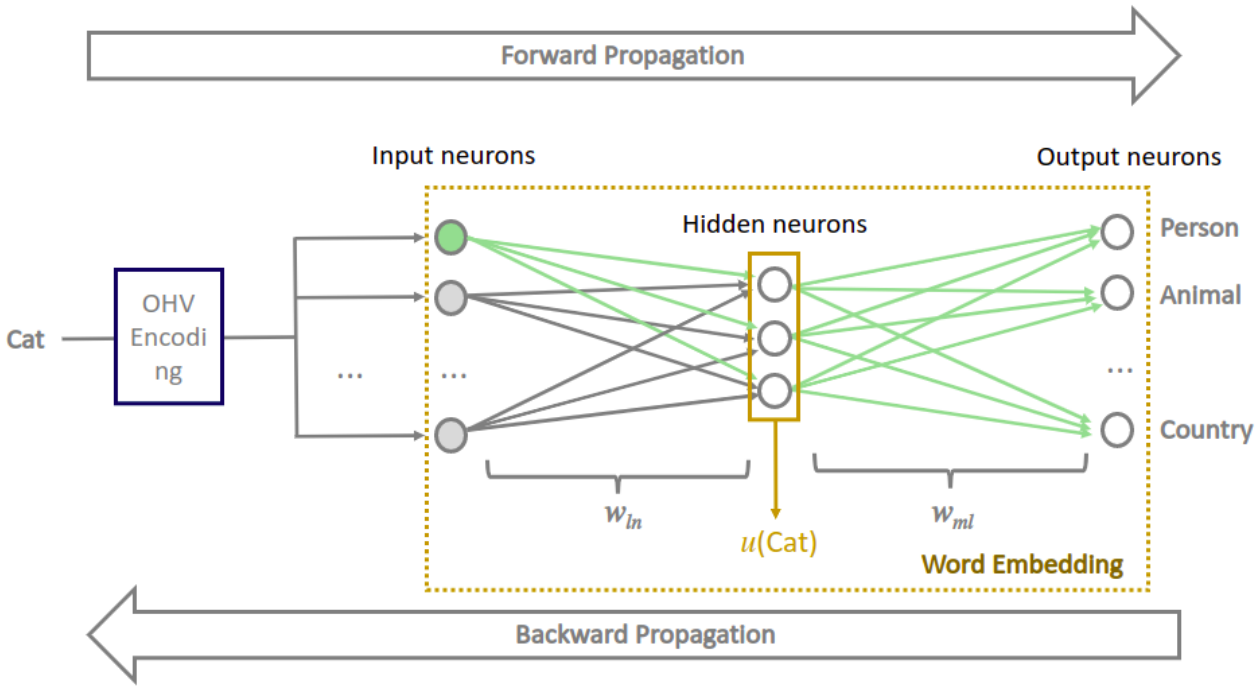


Figure 2.8: Conversion of a category to a One Hot Vector and the activation of the embedding's corresponding input neuron.

In Word Embedding, single words are represented with numerical vectors containing continuous values. These vectors are called Embedding Vectors and allow the application of the mathematical operations summation and subtraction. One popular examples for this is:[14]

$$WE(\text{king}) - WE(\text{man}) + WE(\text{woman}) \approx WE(\text{queen}).$$

With $WE(\cdot)$ as the function, which maps single words or One Hot Vector (OHV)s to the embedding space. Consider a network containing adaptive parameters $\boldsymbol{w}$ like in figure 2.8: a neural network is shown, with N input neurons $\left\{n_n^{(i)}\right\}_{n=1}^{N}$, L hidden neurons $\left\{n_l^{(h)}\right\}_{l=1}^{L}$ and M output neurons $\left\{n_m^{(o)}\right\}_{m=1}^{M}$ with $N >> L$. In figure 2.8, L equals 3. The input layer consists of the input neurons, the output layer consists of the output neurons and so does the hidden layer, which consists of the hidden neurons. After finishing the training, the hidden layer generates the embedding vector, which represents the inputs numerically[14].

In this presentation of Word Embedding, the inputs are words, like Cat, Dog or Airplane, and the outputs are so called *entities* which describe word categories, such as Person, Animal or Country. Therefore, the words with the same entity were mapped together. One example for a training sample could be $(Cat, Animal)$. After transforming the input, the sample may look like this: $(\text{OHV}(Cat), Animal)$. Every entity has its own output neuron, see figure 2.8. These give information about the word's assumed entity. For example, if a word is assumed to be an Animal, the second output neuron gets the highest value. The ideal output values y_1 for the input Cat could be $[0, 1, 0, ..., 0]$ with all output neurons producing the value 0 and only the second neuron producing 1.

The theoretical background about the neural network's training was taken from [9]. During training, the weights are updated due to the algorithm of *Back Propagation*. But first, a *Forward Propagation* is done.

Forward Propagation means, the inputs are passed through the network until reaching the output layer, see figure 2.8. For this, the inputs are converted into a numerical representation, usually a OHV. OHVs have elements $e \in \{0, 1\}$ with exactly one element being 1: $\sum_{n=1}^{N} e_n = 1$. This way, every input value is encoded with a unique vector:

$$\begin{aligned} \text{Cat} &= [1, 0, 0, ..., 0] \\ \text{Dog} &= [0, 1, 0, ..., 0] \\ &... \\ \text{Airplane} &= [0, 0, 0, ..., 1]. \end{aligned}$$

The number of elements in the OHV corresponds to N, the number of different inputs or words. These vectors are needed to apply them to the embedding. Since exactly one element always takes the value 1, exactly one input neuron is activated, namely the one that receives the value 1, see the green input neuron in figure 2.8. The other input neurons stay inactive (grey) due to the 0s achieved. The element 1 at the activated input neuron $n_1^{(i)}$ is multiplied with the weights w_{ln} on its path and passes a signal to each hidden neuron, see green arrows in w_{ln} in figure 2.8. The first hidden neuron achieves the value $1 \cdot w_{11}$, the second hidden neuron achieves the value $1 \cdot w_{21}$ and the third the value $1 \cdot w_{31}$. Afterwards, the hidden neurons apply these signals to their activation functions.

An activation function can lead to two possible states: inactive and active, and often has restricted output domain, e.g. $[0, 1]$. Thereby, low signal values are mapped to low output values by the activation function if a critical value is not reached and vice versa. Activation functions are often non-linear. If a linear mapping is desired, no activation function needs to be applied. Usual activation functions for binary prob-

lems are the sigmoid function and the tanh-function. For multi categorical output, the soft max is often used [15]. Next, the outputs of the hidden layer's activation functions are multiplied with the weights w_{ml} on their further path, see green arrows in w_{ml} in figure 2.8. After the Forward Propagation is finished, the network's outcome $\hat{y}_i$ is observed and the error compared to the true output value y_i, can be calculated. For this, a loss function E is used. Other names for the loss function are error function and cost function. The loss function often calculates the difference between predicted and true outcome of the input. Examples for this are the Mean Squared Error for regression problems and the cross entropy loss for multi categorical classification problems [4]. Now, the Back Propagation can be applied.

Back propagation runs in the opposite direction of Forward Propagation: from output to input layer, see figure 2.8. During Back Propagation, the gradient of the loss function E is calculated. The gradient $\frac{\partial E}{\partial \boldsymbol{w}}$ is a vector of all partial derivatives of the weights $w_i \in \boldsymbol{w}$ in the embedding and point in V-dimensional space in the direction of the strongest increase of $E(\boldsymbol{w})$, with K as the whole number of weights. For further information about batching options, see [8]. The minimum of the loss function is searched due to the gradient's negative direction. This algorithm is called *Gradient Descent*:

$$\Delta \boldsymbol{w} \sim -\eta \frac{\partial E}{\partial \boldsymbol{w}}.$$

η is the learning rate and regularizes the step size during this search. If η is defined to be too large, there is a risk of skipping the optimum due to too large steps. On the other hand, if the learning rate is too small, the search for the optimum progresses very slowly.

If the Back Propagation step is finished, the weights $w_i \in \boldsymbol{w}$ are updated due to:

$$w_i \leftarrow w_i + \Delta w_i.$$

After the predefined stop criterion for the training is reached, the training process is finished. A stop criterion is for example when the desired number of training iterations or a sufficient prediction precision is reached. Then, the output layer can be removed and the words can be represented by the hidden neuron, as symbolized in figure 2.8 with $u(\text{Cat})$ by applying Forward Propagation.

3

METHODOLOGY

In this chapter the methodology applied in chapter 4 is described. If the Bayesian Optimization (BO) shall be used for optimization tasks in industry, the problem that arises here is the inability of the Gaussian Process (GP) to model non-continuous inputs. Beside this, the second problem is, the Acquisition Function (ACQ)'s inability to handle non-controllable inputs. Therefore, the aim is to investigate extensions for the classical BO in order to find solutions for these problems. Non-continuous parameters are meant to be both, numerical or categorical. Here, no distinction was made between categories with a natural order, i.e. ordinal, and those without, i.e. nominal. Therefore, all categories are considered as nominal. The selected non-classical inputs are the following three:

1. numerical discrete, controllable
2. categorical discrete, controllable
3. continuous, non-controllable.

The properties leading to problems, when the classical BO is applied, are underlined. First, for all three of them, a property description was made. Second, the symptoms and causes of these problems where analyzed individually by applying the classical BO on them. In case of categorical variable, the classical BO was not investigated, because no reasonable information transfer can be made between categories, see 4.3. Therefore any method, that considers similarities between categories has a better performance than the classical BO. For categorical variables, first the symptoms and causes were explained. After defining the symptoms and cause of problem, the classical BO was extended by the corresponding solution approach, which should consider the investigated non-classical property. The solution approaches were worked out from literature or based on new assumptions. For numerical discrete variables, the paper [6] was examined, for continuous context the paper [11] and for the categorical variables, a transfer solution was investigated, which was taken from Word Embeddings, see section 2.5. The reason for investigating solution approaches individually was, that diverse papers show solutions for single problems. Therefore, first individual methods were worked out. The theories of [6] and [11] are described in sections 4.1 and 4.2. It was investigated whether the analyzed problems still occur after extending the BO. Last but not least, all three non-classical properties were combined in one objective function and the classical BO was extended by the three solution approaches and

applied to this objective function. The idea of combining them in the last step was to investigate the effectiveness of the solution approaches in combination.

The data used for this investigation was simulated. This has the advantage of considering different problems independent of a data set and therefore finding general solution approaches. Simulated data gives more flexibility than a data set which represents a certain plant. Moreover, the true value of noise, which is also estimated by the BO, is known and can therefore be compared with the estimation. But first, the simulation of the non-classical inputs are described. Numerical discrete values were rounded up or down, based on an algorithm, which maps a continuous value to an interval where every interval has a numeric representation, for more information see appendix A. Continuous context were called via a generator method in python. Generators return in every call another value. Therefore, the generator's current value is observable but not controllable, just like the context, see section 2.1. Last but not least, the categorical values were simulated by assigning a ground truth value from the objective's domain space, one for each category. The true values for categories were considered as unknown to the BO. For every single problem another objective function was defined, see appendix A. For numerical discrete variable, a 1D objective function was used during execution, see section 4.1. In case of context and categorical variable, a 2D objective function was applied for each. This was necessary to show the problem behavior and/ or the functionality of the solution approaches. The simulated objective functions were selected based on two properties:

1. Beside a global maximum, at least one local maximum must exist.
2. The function has to be smooth and not wiggly, so that the assumptions from equations 2.4 and 2.2 can be followed.

The term *wiggly* was adapted from [5] and describes high changes of function values in a small input domain. In order to compare the performance of classical and extended BO, the solution for the analyzed problems were investigated by analyzing several plots of the BOs. Furthermore, the model complexity regarding the covariance function's length scale l were compared. In chapter 4 noise-free data $\epsilon \sim \mathcal{N}(0, 0.0)$ was used and in the appendix noisy data, $\epsilon \sim \mathcal{N}(0, 0.5)$ or $\epsilon \sim \mathcal{N}(0, 0.8)$, was used. Therefore, the assumed noise in data was also compared to the true noise. The trade off parameter β was also listed. Because λ refers to the covariance function, it can not be interpreted very well. It only gives information about the direction of the covariance but none about its strength. Therefore, λ was not listed beside the other parameters.

4 EXECUTION

In this chapter, the Bayesian Optimization (BO) is applied to the selected non-continuous inputs and extended by the solution approaches. In sections 4.1, 4.2 and 4.3 the problems of the classical BO when being applied to *numeric discrete variables*, *categorical variables* and *continuous context* are analyzed. Then, the realization of the solution approach for the corresponding problem is described. The classical BO was extended for the solution approach and shown on plots. The plots include the 1st iteration with the initial samples and one iterated sample. The initial samples were set randomly, except for numeric discrete, there the samples were set manually to show the undesired behavior of classical BO. For some problems more iterations are shown to show different behaviors. Each of these sections begins with a description of the input property. Then, the symptoms and cause of the problem, classical BO has in dealing with the appropriate non-classical inputs, is explained. Finally, in section 4.4 all simple non-classical cases are combined to test the effectiveness of the combined solutions. Furthermore, noise-free and noisy data were examined for all applications. The initial samples are the same for classical, extended BO as well as noise-free and noisy data. With other words: The initial input samples in one section are identical. Further considerations are made in chapter 5, where results are discussed including the consideration of model parameters. The plots in this chapter only include noise-free data. In appendix B supplementary plots including noisy data are presented and discussed in chapter 5.

4.1 NUMERIC DISCRETE VARIABLE

An input is considered as *numeric discrete* if it has level-based numeric values. Numeric data are generated via measurement or counting, i.e. the total number of an entity [19]. This definition describes both, continuous and numeric discrete data. Examples for numeric data from industry are pressure and temperature [16] for continuous and total number of machines or workers for numeric discrete data, see chapter 1. In case a machine parameter has numeric discrete values, they are level-based and therefore non-continuous. One example for a numeric discrete machine parameter is an air conditioner with level-based values, e.g. with precision 0.5°C. In this case, a value of 24.25°C is not able to be realized. For simplicity, integer valued variables were considered when regarding numeric discrete inputs.

If a numeric discrete variable is modelled by a classical BO the Gaussian Process (GP) does not consider the level-based values. Distances between single input values are considered instead of calculating distances between the levels. In the following, the levels are described as intervals. Intervals have one integer value which represents the whole interval. This representation is applicable to metric distance measures such as the euclidean distance and was used for function observation. Due to [6], a sequence of repeatedly selecting the same interval during exploration is possible if a numeric discrete variable is modelled as continuous one. Multiple selections of one interval may be favorable, if the algorithm converges after finding the global optimum. However, during exploration phase, when there is much to discover, this behavior is undesired. Consider an interval I_a with the observed sample $x' \in I_a$ and the interval representation a. Then, evaluating the proposed sample which belongs also to this interval $x^* \in I_a$ results in observing the exact same function value as $f(x') = f(a)$ if the data are noise-free and a slightly different one if it is noisy. There is almost no information gain, except the noise variance. However, the target of optimization is to find the evaluation which leads to the best function value. Regarding evaluation costs, this undesired behavior is expensive and wastes resources such as time and money.

For this section, β was set to 4. In case this parameter was set too low, the following example could not be presented: in figure 4.1 a classical BO with mean prediction and prediction uncertainty (top: blue curve and blue area) is shown together with a level-based objective function (green line). The objective was defined as shown in A.1. The observations are marked as circles, orange for initial samples and black for the evaluated one. The evaluated sample was proposed during the last iteration. The Acquisition Function (ACQ) (bottom: red curve) has the maximum value at $x^* = 7.2$, which is proposed as next sample to evaluate (red triangle). This value belongs to the interval I_7, see equation A.2 for the definition of the intervals. The discreteness of the objective function is not considered by the GP and therefore not considered in the ACQ. The next proposed sample belongs to a yet unknown interval. The undesired behavior which occurs if a classical BO is applied to a numeric discrete variable, is not observable in figure 4.1. But after iterating 4 more times, a sample from a known interval is proposed even though the maximum observed function value does not belong to this, see figure 4.2. At the 5th iteration, the discreteness of the input variable is still not considered. The test point with the second highest acquisition value is located at $x^{*'} = 10$ and with the third highest value at $x^{*''} = 4.7$. The last one belongs to the interval I_5 where the global optimum is located but the point with the second highest acquisition value does not.

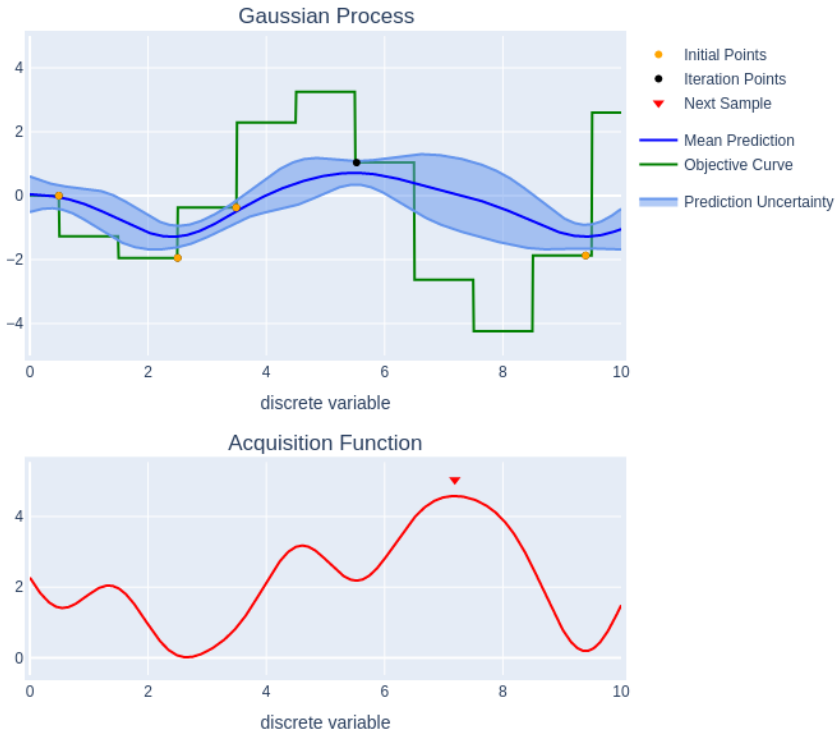


Figure 4.1: 1st iteration of classical Bayesian Optimization applied to numeric discrete variable. The mean prediction (blue line) and prediction uncertainty (blue area) of the Gaussian Process are shown in the upper figure. The objective function (green line) is also visualized. Initial (orange) and iterated samples (black) are marked as points. In the lower figure, the Acquisition Function (red line) is shown. The proposed sample is marked as red triangle.

Neither in figure 4.1 nor in 4.2 all observations are catched by the mean prediction. In order to select the model parameters considering the training points, the most simple model is selected and therefore an automated Occam's razor performed, see [23]. This is discussed in 5.1 together with the model parameters. The plots for the 1st and 5th iteration of the classical BO applied to noisy data are shown in the figures B.1 and B.2 respectively. In these plots, the mean predictions catch all observations and assumes low noise. This behavior is discussed in section 5.1. The difference of modelling noise-free and noisy data is, that the observations do not lie directly on the objective function.

In order to get the example shown in figure 4.2, the β parameter was set to 4 for all models applied to the numeric discrete variable.

As mentioned in section 2.2, the covariance function depends on the distance measure d which is classically applied to single input values. The problem here is that the discrete intervals should be considered. Suppose x, x' are values of a numeric discrete variable in domain $\mathcal{X}_d = \{I_1, I_2, ...\}$ with $x \in I_i$ and $x' \in I_j$. Then, the distance $d(x, x')$ should be zero if x and x' belong to the same interval and non-zero otherwise:

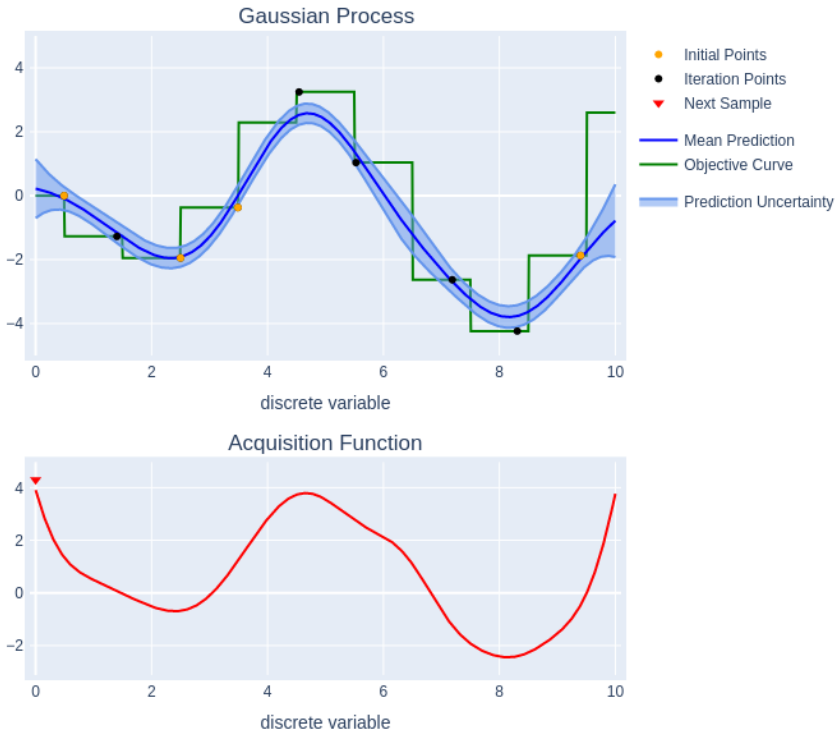


Figure 4.2: 5th iteration of classical Bayesian Optimization applied to numeric discrete variable. This plot shows a symptom that may occur if the discreteness of the input is not considered.

$$d(x, x') = 0 \iff I_i = I_j$$

$$d(x, x') \neq 0 \iff I_i \neq I_j.$$

In case of $I_i \neq I_j$, the distance $d(x, x')$ should be higher, the further away these intervals are due to the assumption based on similarity (see equations 2.4 and 2.2).

In order to consider the level-based values, the GP was modified. The input values were all mapped to intervals. These intervals were modelled by the GP. Each interval got a unique representing value. For this, a discrete transformer T was defined, which considers the levels of the discrete variable, see equation A.2. Afterwards, the corresponding representations were passed to the euclidean distance measure d_e. In this way, the distances between intervals were considered :

$$K(T(x), T(x')) \longrightarrow d_e(T(x), T(x')).$$

This approach was taken from [6]. In figure 4.3, the GP's structure was extended for the discrete Transformer T. Because now the GP considers the discrete property of the input, the ACQ does the same when

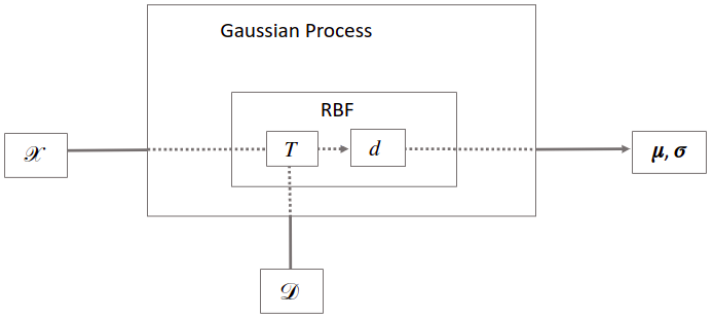


Figure 4.3: Structural Bayesian optimization extended by discrete transformer T.

calculating the acquisition values based on mean and standard deviation vectors, see section 2.3. The cause was fixed by doing so and thus the symptoms no longer occur. In figure 4.4, the 1st iteration of the extended BO is shown on noise-free data. The upper figure, again shows the GP (blue line and blue area) and the discrete objective function (green line). The initial samples (orange circles) and the iteration sample (black circle) are the same as the classical BO's 1st iteration, see figure 4.1. In the lower figure, the ACQ (red line) is illustrated together with the next proposition $x^* = 4.51 \in I_5$ (red triangle). Every interval has no more than one observation, so no resources were wasted. Furthermore, every interval has one mean and one uncertainty value. The unobserved intervals have high acquisition values and are therefore preferred to the observed ones, see section 2.3. The mean prediction barely misses some observations. For example in interval I_2, the blue and green line do not overlay. They differ slightly. And even though the data were noise-free, the observed intervals have a non-zero uncertainty. This means, the observed data are assumed to be noisy. The same was observed for classical BO at both iterations, see figure B.1 and B.2. For more information see the parameter discussion in section 5.1. In figure B.3 the 1st iteration of extended BO applied to noisy data is visualized. In this plot, the mean prediction at unobserved intervals is always near zero. This behavior is discussed in section 5.1.

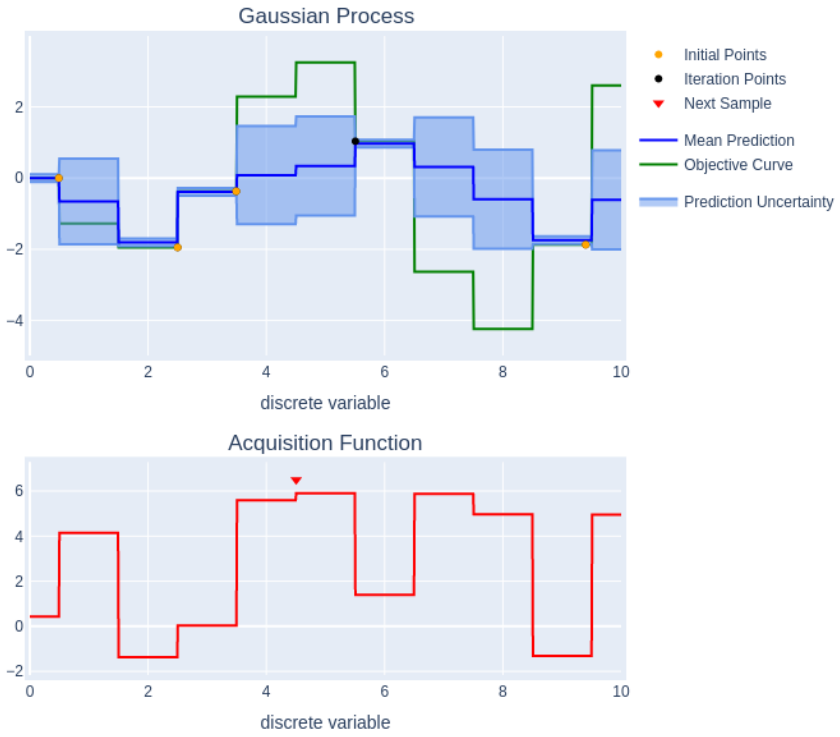


Figure 4.4: 1st iteration of extended Bayesian Optimization applied to numeric discrete variable. Here, the discreteness of the input is considered by Gaussian Process and Acquisition Function.

4.2 CONTINUOUS CONTEXT

An input is considered as *context* if it is observable but not subject to human influence. Its domain can be continuous, but also categorical or numeric discrete. The solution approach presented here considers continuous context. Examples for contexts are environment temperature (continuous), restriction of resources during production (categorical) and number of sick workers (numeric discrete). The numeric values of a continuous context is directly applicable to the distance measure used in the classical GP. The only difference between continuous context and continuous variable is the option of controllablility. If there is a context influencing the objective function, two symptoms may occur:

1. In figure 4.5 the 1st iteration of the classical BO applied to continuous variable is shown. Even though in the simulation a continuous context had also influence on the target metric, it was not considered. The 20 initial samples were set randomly. One sample was observed by evaluating the proposed sample, see right most point at $x = 5$. The model acts smooth while the uncertainty in the data is non-zero at observed points. The ACQ proposes the right most value, which is equal to the last observed one. The context was simulated to change its value

after each iteration and the influence on the target varies almost uncontrollably. The observed samples have despite small distances high variation in their function values. Consider the samples $x_1 \approx 2.66$ and $x_2 \approx 2.75$, the distance on the y-axis is high despite the small distance on the x-axis. This is in contradiction to the assumptions about the data (equations 2.2 and 2.4). In figure B.7 the first iteration of the classical BO applied to noisy continuous variable without considering the context is shown.

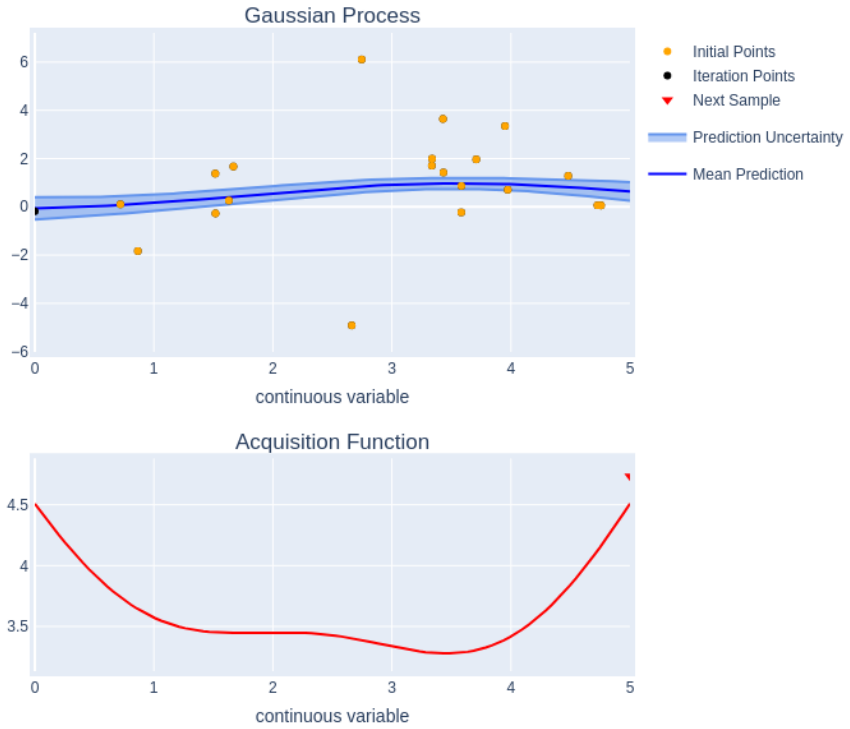


Figure 4.5: 1st iteration of classic Bayesian Optimization applied to continuous context ($\sigma_n^2 = 0.0$). In this plot, there is high variation in the observations, despite small input distances, see $x1 = 2.66$ and $x = 2.75$ because the continuous context was not considered.

2. If the context is considered by the GP the ACQ classically does not distinguish between controllable and uncontrollable inputs. There is no guarantee about the proposed sample being feasible. Suppose, the current context has the value 3.54. But the ACQ optimizes the next proposed point's values regarding the whole input domain and suggests $x^* = (2.3, 5.4)$ even though the value of 5.4 for the context is not feasible.

As mentioned before, the difference between contexts and variables is that the former, unlike the latter, is uncontrollable. Until now, every input modelled by the GP was controllable and therefore the proposed samples were all feasible. The acquisition values immediately

depend on the mean and standard deviation vectors generated by the GP. Both are taken as they are and no distinction into controllable or non-controllable is made. Therefore, this direct transfer of the mentioned vectors has to be modified in order to make the distinction and thus solve the problem. Due to the cause of the problem, in both, GP and ACQ, the approach from [11] was realized: an Acquisition Function called Context Upper Confidence Bound (C-UCB). The GP models both, variable and context. The resulting mean and standard deviation vectors were restricted regarding feasibility and afterwards passed to the ACQ. Feasibility here means, that the vectors were restricted to the current context value and therefore the search space for the ACQ consists of values which can be achieved via variable control. The context value itself remains unchanged when the next point is suggested.

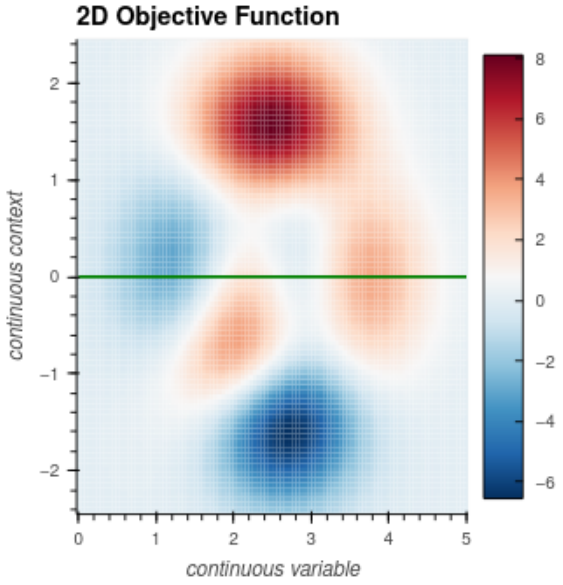
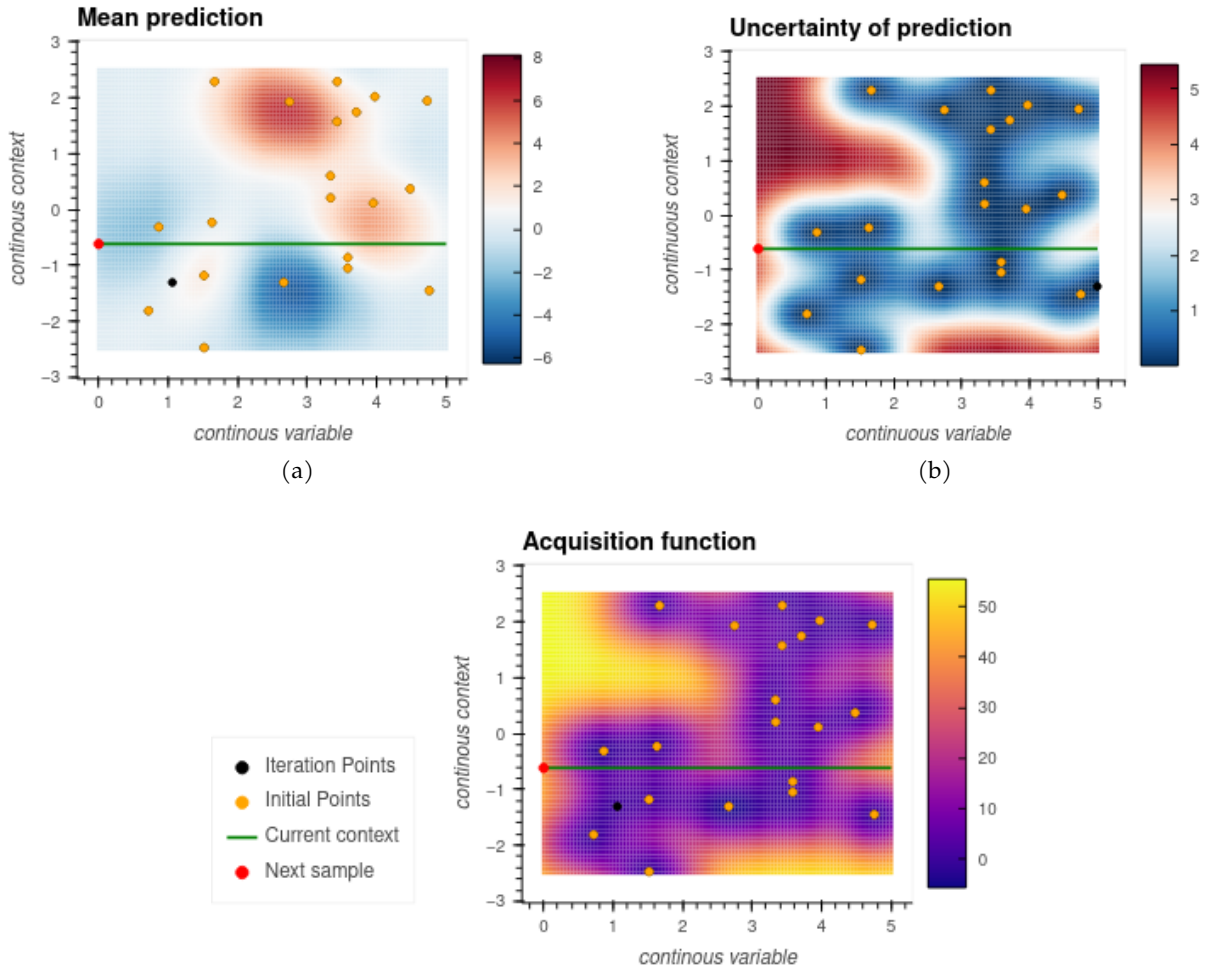


Figure 4.6: Heatmap of a 2D objective function with continuous variable on the x- and continuous context on the y-axis. The current context value is marked as as green line.

In case of 1D input including one context, this makes less sense. Therefore, a 2D objective function was defined in figure 4.6, see A.3. There is a heat map shown with a controllable continuous variable on the x-axis and continuous context on the y-axis. Note that in order to search the whole 2D domain, the β parameter was increased to 10. For lower values, close points are proposed and evaluated repeatedly but 10 leads to almost equally distributed samples for exploration. The function value is color coded. The current context value is shown as green horizontal line. The context varies its value automatically after each observation and was simulated in a stepwise sinusoidal manner: In the middle of the range, the derivative is high, near the edges it is low. The context values commutes from one end of the context's domain to the other. Every observation consists of two values: x- and y-value for controllable

and non-controllable inputs respectively. This objective function was also used for classical BO. In figure 4.7, the extended BO applied to this objective function is visualized at 1st iteration. The last observed context value is -0.62. Restricting the feasible values to this context value, the next point (red point) results in $x^* = (4.0, -0.62)$. The assumed noise variance is low near the observed points (black points), see figure 4.7b. Actually, after restricting the ACQ to the context value, it gets one dimensional, see figure 4.8. In figure B.8, the 1st iteration for noise data is shown.



(a) (b) (c)

Figure 4.7: 1st iteration of extended Bayesian Optimization applied to continuous variable and continuous context ($\sigma_n^2 = 0$). In panel (a) the 2D mean prediction of the Gaussian process is shown. Panel (b) presents the 2D prediction uncertainty and panel (c) the Acquisition Function. The current context value (green horizontal line -0.62) and the observations (orange and black) are also shown in all panels.

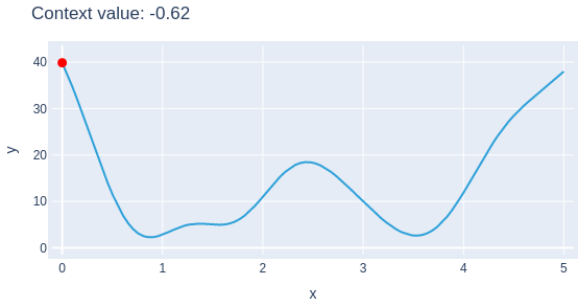


Figure 4.8: Acquisition function after restricting to the context value -0.62.

4.3 CATEGORICAL VARIABLE

An input is considered as *categorical* if it is non-numeric data, see section 4.1. Categorical entities are distinguished into ordinal and nominal. Ordinal categories have natural order, e.g. coearseness of sieve with values

1. rough : 0.8 cm
2. medium : 0.5 cm
3. fine : 0.3 cm.

Note that the actual values of $1m/s$, $2m/s$ and $5m/s$ are unknown to the user and therefore not directly applicable to the Gaussian process. Only the values 1,2,3 or slow, medium, fast and the property of non-equidistance are known.

The categories have a natural order regarding the fineness. The values 0.8 cm, 0.5 cm and 0.3 cm are fictive examples for diameter of the sieve holes. However, no information about the influence of this category on the target metric can be measured and therefore these values can not represented numerically. Nominal categories have no such order, e.g.

$$\textit{type of grain} \in \{\textit{wheat}, \textit{rye}, \textit{spelt}\}$$

as resources for producing muesli. In this thesis, no distinction was made between ordinal and nominal categorical inputs and both are simply referred to as nominal. Distances between categories can not be calculated with euclidean distance. The categories must be numerically represented. The numeric representations must be close to each other, if two categories behave similar regarding the objective and further away otherwise. For example, suppose the categories are encoded as One Hot Vector (OHV)s, see section 2.5. The distance between two unsimilar vectors has always the same value, in case of euclidean distance measure this value is $\sqrt{2}$. Only in case of identical vectors, the distance is 0:

$$\begin{aligned} d_e\left(\mathrm{OHV}(c_i), \mathrm{OHV}(c_j)\right) = \sqrt{2} &\iff c_i \neq c_j \\ d_e\left(\mathrm{OHV}(c_i), \mathrm{OHV}(c_j)\right) = 0 &\iff c_i = c_j. \end{aligned}$$

Therefore, no gradation is done if categories are converted to OHVs: vectors are either non-identical or identical. If two vectors are non-identical but similar, they are not recognized as such but only as unsimilar, even though this information is highly important for an efficient modelling. If n categories are considered to be independent of each other, just like in case of OHV, n GPs are generated (one for each category) resulting in high waste of resources and the lack of information transfer between categories. Another option is to manually assign a numeric representation r_i for each category c_i without knowing the true similarities between them. In this case, categories with similar representations $d(r_i, r_j) \approx 0$ are assumed to have similar function values $d(f(r_i), f(r_j)) \approx 0$ without any certainty about the correctness of the representations. Due to the lack of information transfer between categories, both options would not perform information transfer, therefore, the classical BO was not applied to categorical inputs. Only the extended BO was applied in order to examine whether the information transfer was considered. The categories need to be mapped to a metric space intelligently, with similar categories having similar numeric representations. Similar representations lead to small distances and vice versa. Suppose producing muesli from rye behaves similar to producing it from spelt and very unsimilar from producing it from wheat. Then, for both, rye and spelt, the numeric representation should be similar, e.g. 4.5 and 4.8. Conversely, their representation must be unsimilar to wheat, e.g. wheat: −2.2.

Similar to the solution approach for numeric discrete variables in section 4.1, the categorical variables were transformed before being applied to the covariance function's distance measure. For this, the categories were converted into numeric representations, as is the case in Word Embedding, see section 2.5. First, the values of the categorical variable $c_i \in c$ were converted to OHVs. Second, the OHVs were passed to an embedding, which was placed at the entry to the Radial Basis Function (RBF). Third, the covariance function including the embedding was trained by applying Back Propagation and Gradient Descent, see section 2.5. When initializing the GP, the training data consist of the initial points: the input data are the variable values and the output data are the corresponding observed function value of the unknown objective. In figure 4.9, this structure is illustrated. Beside the categorical variables, others, e.g. numeric discrete $x_d \in \mathcal{X}_d$ and continuous variables $x_i \in \mathcal{X} \backslash \{\mathcal{X}_d, c\}$, can be applied. T is the discrete transformer described in section 4.1 and was also included in this figure, to show the similar realization between embedding and discrete transformer. The neuron, which outputs the embedding representation $u(c_i)$ (orange) can be interpreted similar to the hidden layer's output in section 2.5. During training of the covariance function (blue dotted box), the weights w_{ln} were adjusted automatically to the used training data. The

embedding updates its weights w_{ln} whereby the RBF updates its parameters l, λ. The assumed noise variance $\hat{\sigma}_n^2$ is estimated after setting l and λ.

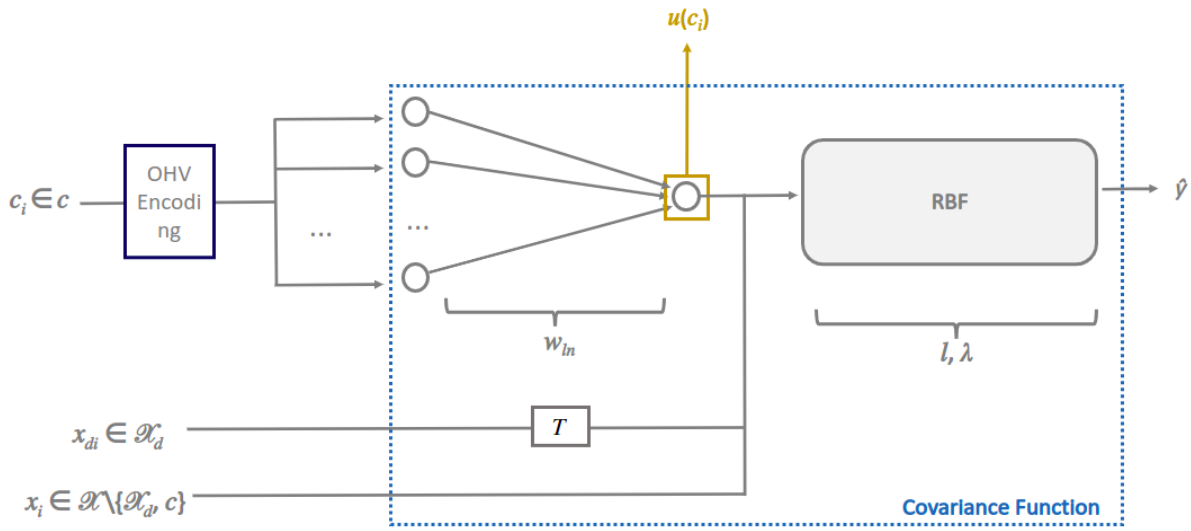


Figure 4.9: Structure of covariance function extended by embedding.

Due to the embedding being trained together with the RBF, the two components are not separable when being applied. This structure must be preserved during evaluation and further iterations. To avoid information loss, see section 2.5, no activation function was used in the embedding. The output u is mapped linearly into the metric space so the representations can drift as the distances increase. In case of 1D categorical input, all categories are assumed to be independent, due to the OHV representations being orthogonal to each other. Therefore, in 1D space, each category has to be observed at least once to being represented properly in metric space and no information transfer between the categories can be done. In case of 2D inputs including categorical and continuous variables, information could be transferred between the categories. When the continuous variable has a similar outcome for two categories, these two are assumed to be similar. Therefore, the following example is 2 dimensional.

In figure 4.10, the simulated objective function is illustrated. This is the same objective function used in section 4.2 for context. The colored horizontal lines represent the 5 categories, whereby categories 1 & 2 (green and red) were simulated to be similar, like the categories 3 & 4 (blue and violet). The fifth category (pink) was simulated unsimilar to the others. The categories were assigned the following values on the y-axis:

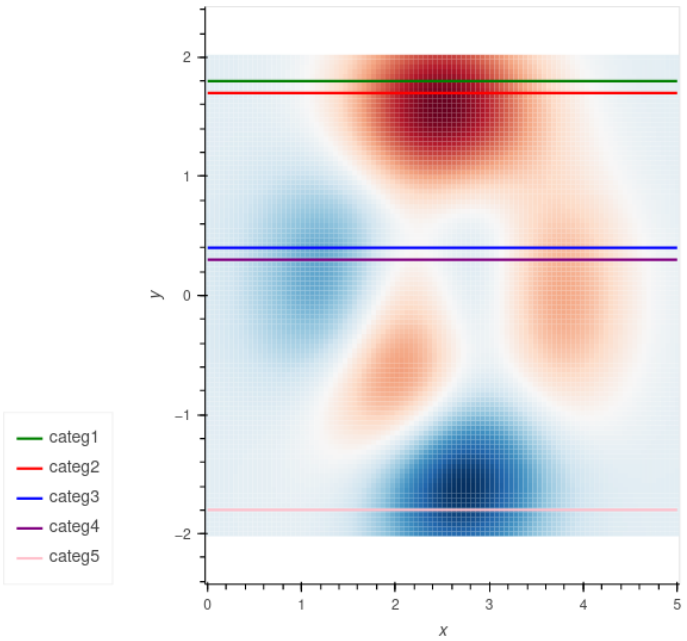


Figure 4.10: Heatmap of a 2D objective function with 5 categories, visualized by horizontal lines.

$$\begin{aligned} \text{category 1}: &\ 1.8 \\ \text{category 2}: &\ 1.7 \\ \text{category 3}: &\ 0.4 \\ \text{category 4}: &\ 0.3 \\ \text{category 5}: &\ -1.8. \end{aligned}$$

However, these values must not be trained by the embedding, it is only important that the relative similarities and unsimilarities are recognized. No more than 5 different values on the y-axis can be selected due to five categories. This means, the values between the categories are not feasible and can not be selected by the ACQ. This was realized by passing the OHVs. Therefore, the combinations between categorical and continuous values got an acquisition value and the one with the highest value was selected as proposed sample.
In figure 4.11 the 1st iteration of the extended BO is shown including mean prediction, uncertainty and acquisition values. The categories 1 and 2 are mapped close to each other. Category 3 was mapped between the two and category 4. And category 5 is placed further away from all of these. The ACQ suggests a sample in category 5 (see the grey diamond at the same height as category 5). The color code smoothly transitions between the categories. The data point at $(2.6, [0, 0, 0, 0, 1])$ has influence on a wide radius. These plots was generated by modelling two extended BOs: One which treats the y-axis as categorical as described and one that treats it as continuous. The last one was generate the shown the mean plot's color code in the whole domain. Because this is the 1st iteration, the similar categories may come closer

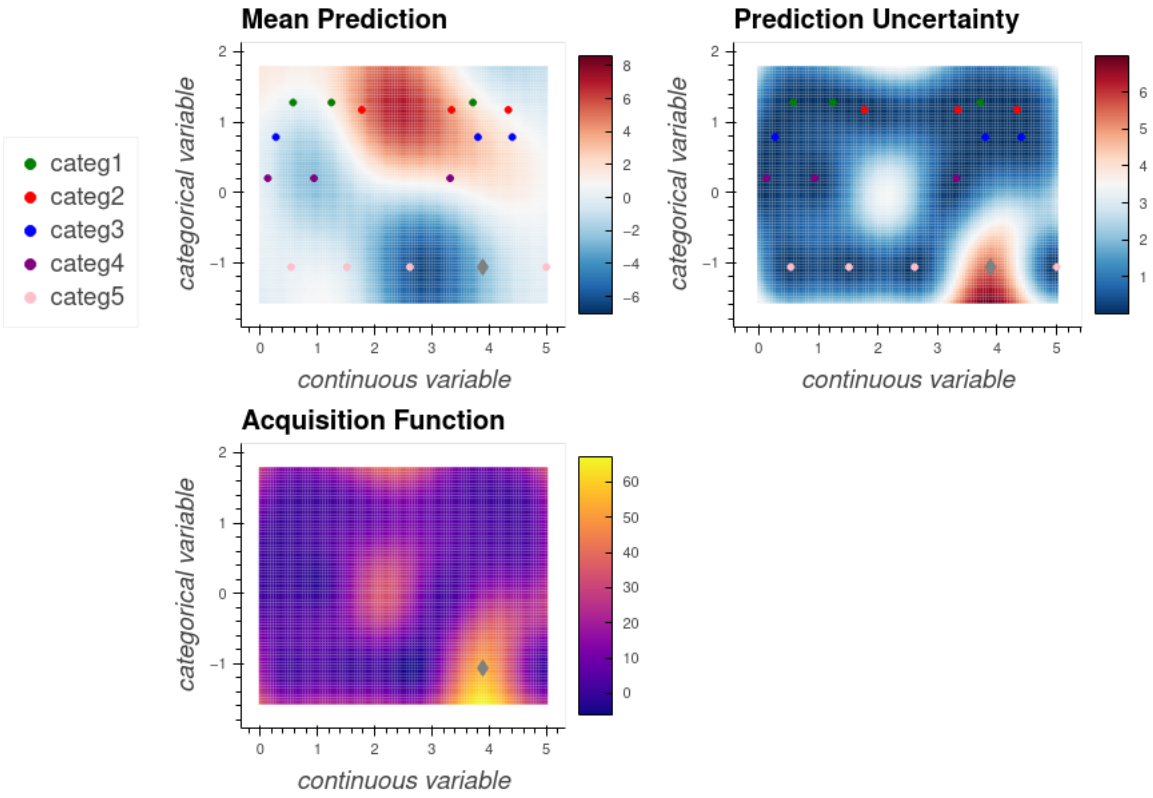


Figure 4.11: 1st iteration of extended Bayesian Optimization applied to continuous and categorical variables ($\sigma_n^2 = 0$). The 2D mean prediction and the 2D uncertainty plot are shown together with the category observations (colored points). The Acquisition Function is shown together with the next sample to evaluate (grey diamond).

with higher iterations. This was discussed in section 5.3. In figure 4.12 the 1st iteration of extended BO applied to noisy data is shown. The categories 1 (green), 3 (blue) and 5 (pink) does not affect any other category nor are they effected. The categories 2 (red) and 4 (violet) do effect each other even though they were simulated unsimilar: an undesired information transfer is performed. This behavior was further discussed in section 5.3.

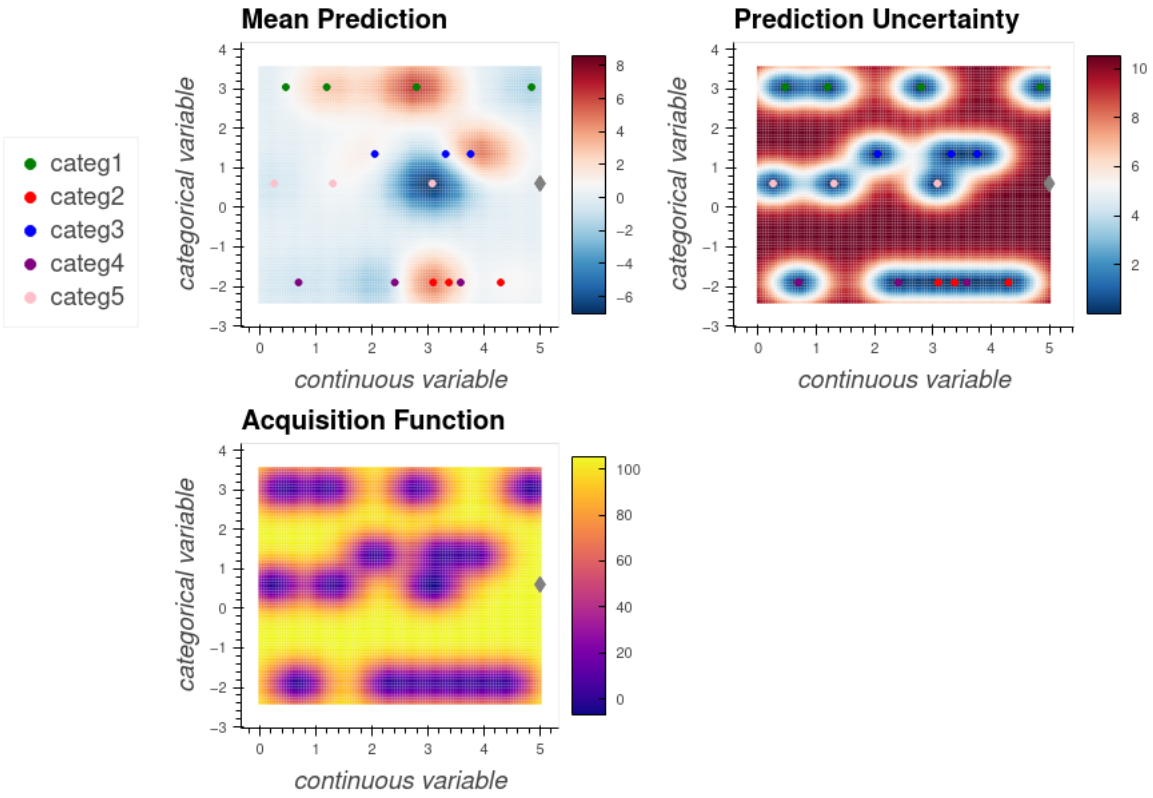
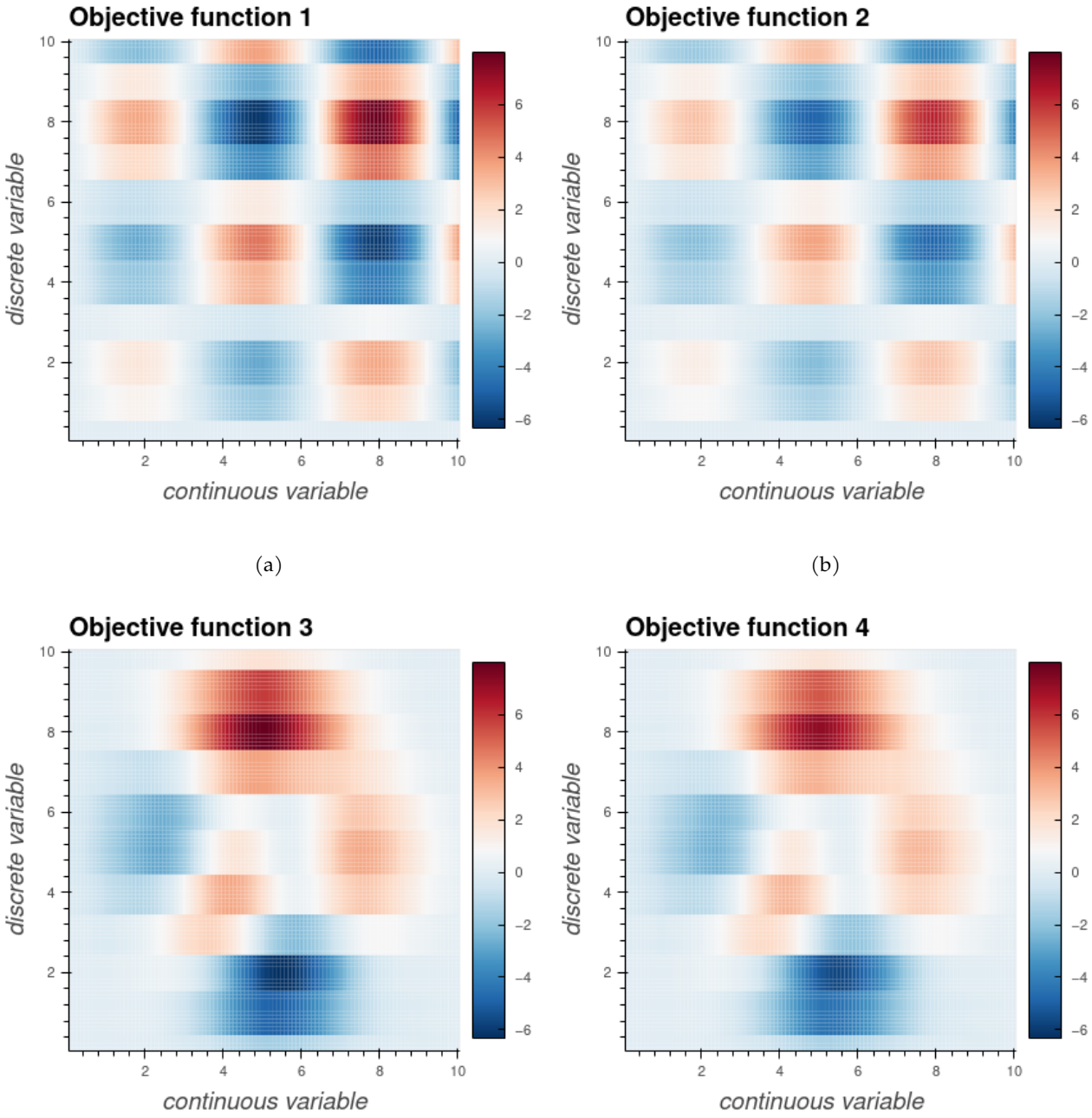


Figure 4.12: 1st iteration of extended Bayesian Optimization applied to continuous and categorical variables ($\sigma_n^2 = 0.8$). The 2D mean prediction and the 2D uncertainty plot are shown together with the category observations (colored points). The Acquisition Function is shown together with the next sample to evaluate (grey diamond).

4.4 COMBINED PROBLEMS

In this section, the three problems of numeric discrete, categorical discrete and context variable were combined. The first variable is a classical continuous variable, the second is a numeric discrete and, due to the simplicity of visualizing, the third is a categorical context variable. An example for categorical context is the restriction to a resource, see table 1.1. Suppose muesli has to be produced but the customer wants it to be made with wheat. So during production, an optimization, that takes the categorical value *type of grain* being equal to *wheat* into account, has to be done.

When applying the BO to these variables, all corresponding solution approaches were implemented. The motivation of doing this, was to investigate whether unknown problems occur in this combination. The investigated papers [6] and[11] consider single problems. The combination of different solution approaches using Bayesian optimization has not been done in this form before. In figure 4.13, four objective



(a) (b)

(c) (d)

Figure 4.13: 4 objective functions, one per context category. The different panels (a)-(d) show the objectives for categories 1-4 respectively. Note that categories 1 and 2 are slightly different regarding the function values, as well as categories 3 and 4. None of them are identical.

functions are shown with continuous variable on the x-axis and numeric discrete variable on the y-axis. These were used in this section to sample the evaluations from. Objectives 1 and 2 represent the true

course of categories 1 and 2 as well as the objectives 3 and 4 representing categories 3 and 4. However, no two objectives are identical but have slightly different scales regarding the function values, see the color codes. Therefore, categories 1 and 2 are relatively similar to each other but not identical as well as categories 3 and 4.
The BO was randomly initialized with 20 samples per category. The context was simulated as follows: First, for 4 iterations, the context was restricted to category 1. Then, for 4 iterations the context was restricted to category 2 and so on until all 4 categories were iterated 4 times. In figure 4.14 the plots for modelling the categories 1 and 2 during 1st iteration are shown including mean prediction and prediction uncertainty. The grey diamond in mean prediction 1 and uncertainty 1 symbolizes the proposed sample, see figures 4.14a and 4.14c. The corresponding ACQ is shown in figure 4.15. Furthermore, in figure 4.16 the mean prediction and uncertainty for categories 3 and 4 are visualized.

The numeric representations of categories 1-4 during 1st iteration are printed in the corresponding figures (see title of mean plots) and are as follows:

$$\begin{aligned} r_1 &: \ 1.1303 \\ r_2 &: \ 1.2073 \\ r_3 &: -1.5711 \\ r_4 &: -1.5561 \end{aligned}$$

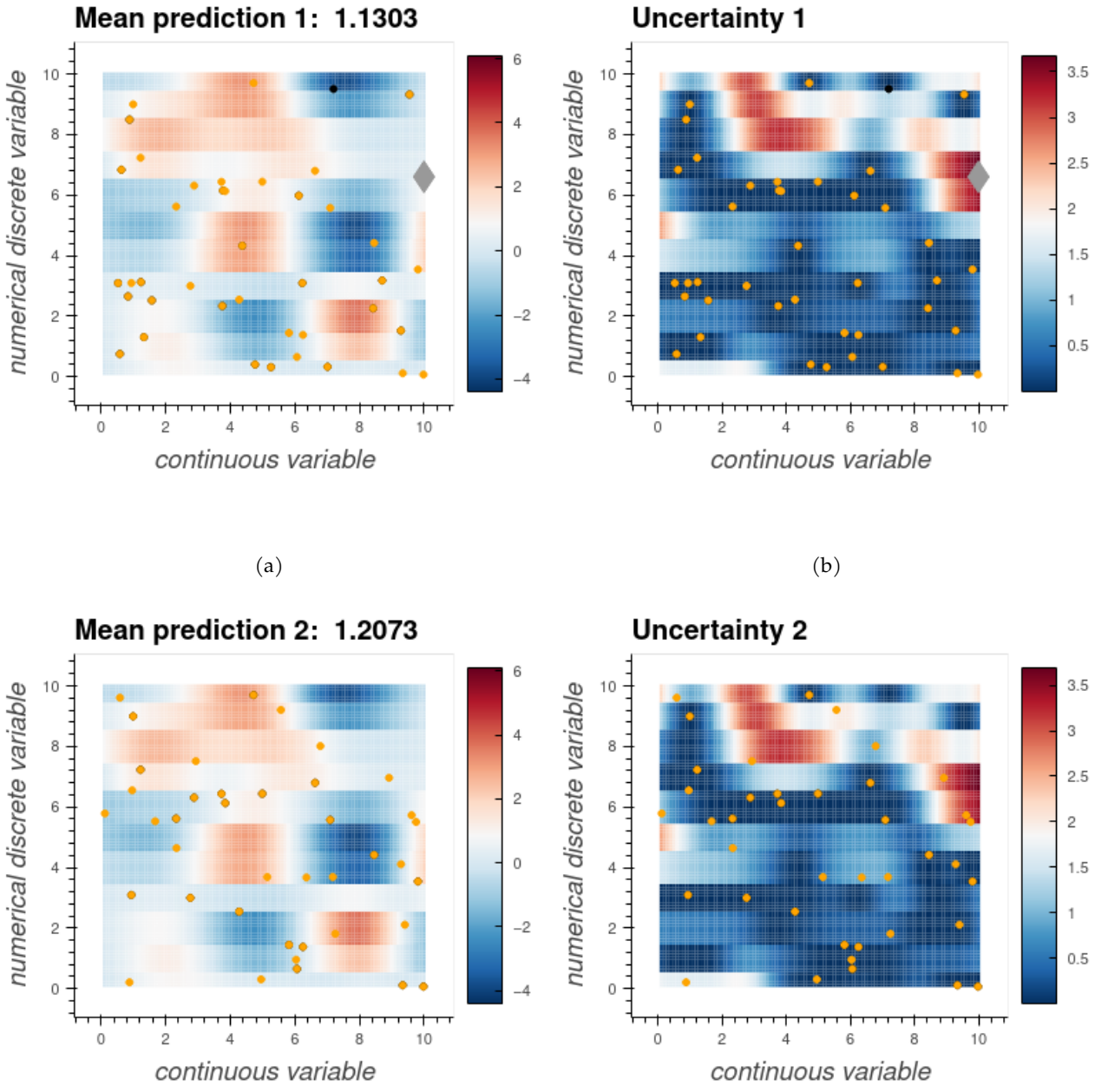
with r_i as the representation for category i. As already mentioned, the categories 1 and 2 were simulated similar to each other and unsimilar to the others. These two got positive representations. Conversely, the categories 3 and 4 were simulated to be similar to each other and unsimilar to the others. These two got negative representations. All similar categories were already represented close and further away from the unsimilar ones. Consider in figure 4.14d the uncertainty of category 2 at location $([8.5 - 10], I_8)$ with the interval $I_8 = [7.5, 8.5]$: the uncertainty of the prediction is low (bright blue in the middle of a white ares), even though there was no observation made. Looking at the uncertainty plot of category 1, see figure 4.14b, shows the reason why: at the location $x = (9.5, 9.3)$ an initial sample was observed. The information about this sample updated the mean prediction and its uncertainty of category 2 because of its close representation to category 1. Conversely, in the mean prediction for category 1 there is a red area located at $([3.5, 5], I_4 \cup I_4)$ (see figure 4.14a). In this area, no sample was observed. However, this information was transferred from category 2, which had 1 initial sample in this area. Compared to other areas, where neither category 1 nor category 2 were observed, the mean prediction for both is set to the default value of zero (bright blue). The corresponding uncertainties are high (red). A similar information transfer is observable between the categories 3 and 4. Consider the left upper and

lower corners in figure 4.16d where the uncertainty for category 4 is shown. There is low uncertainty despite no samples observed.

The 1st iteration of the extended BO applied to noisy data is shown in figure B.10 and B.11. The resulting representations here are:

$$\begin{aligned} r_1 &: -1.3399 \\ r_2 &: -3.3005 \\ r_3 &: \ 2.2173 \\ r_4 &: \ 2.1414. \end{aligned}$$

For a discussion of the results, see section 5.4.



(a) (b)

(c) (d)

Figure 4.14: 1st iteration of extended Bayesian Optimization applied to the 3 non-classical inputs ($\sigma_n^2 = 0$). The plots include mean prediction and uncertainty for categories 1 and 2. The continuous variable is shown on the x-axes and the numeric discrete one on the y-axes. The orange points are the initial points and the black ones were evaluated after iterating. The grey diamond is the next proposed sample by the Acquisition Function, see figure 4.15, and is only shown in the current context category's plots. The categories 3 and 4 are shown in figure 4.16.

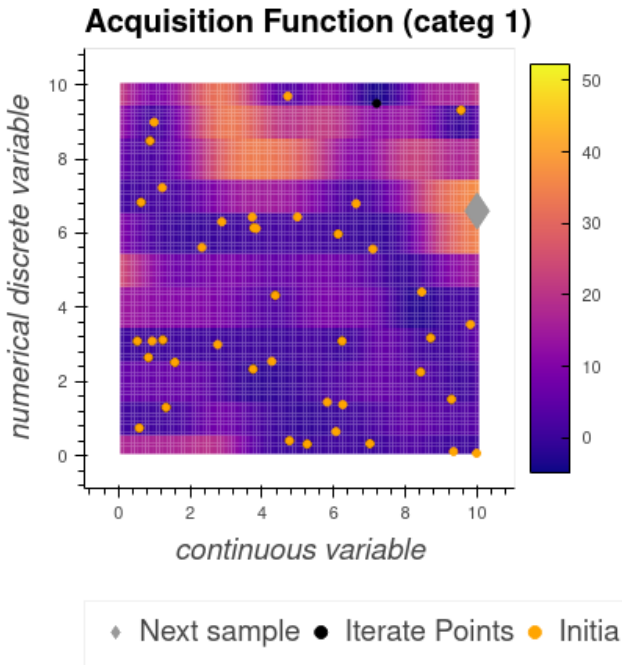
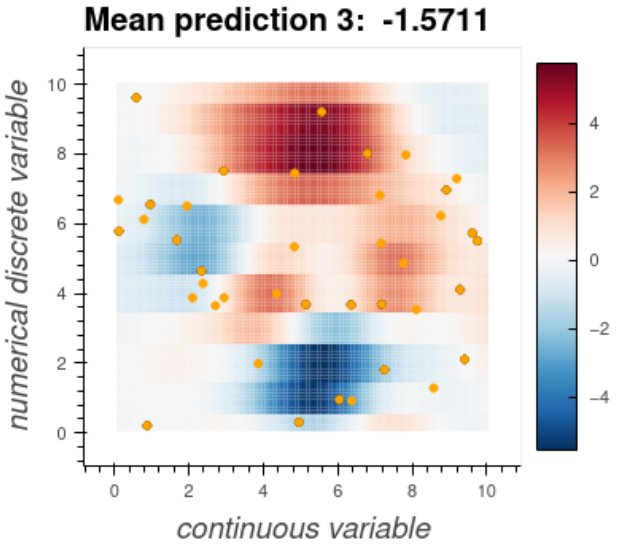
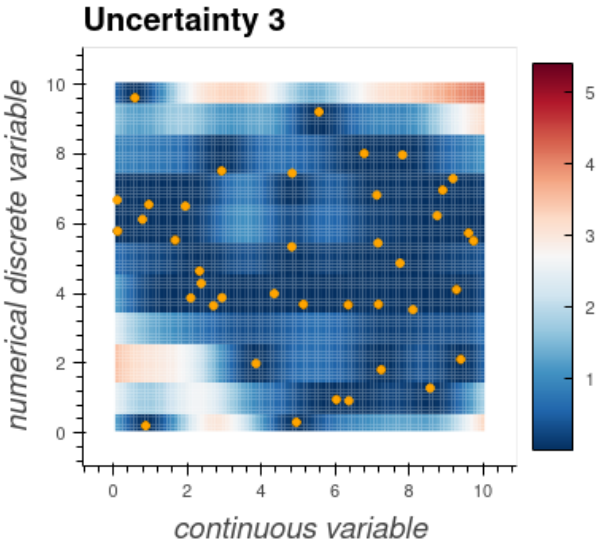


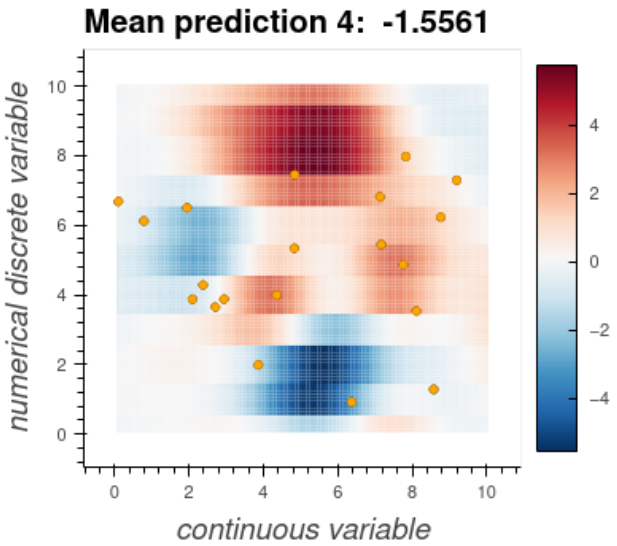
Figure 4.15: 1st iteration of the extended Bayesian Optimization's Acquisition Function ($\sigma_n^2 = 0$). The corresponding mean predictions and uncertainties are shown in figures 4.14 and 4.16. The context's value was restricted to category 1.



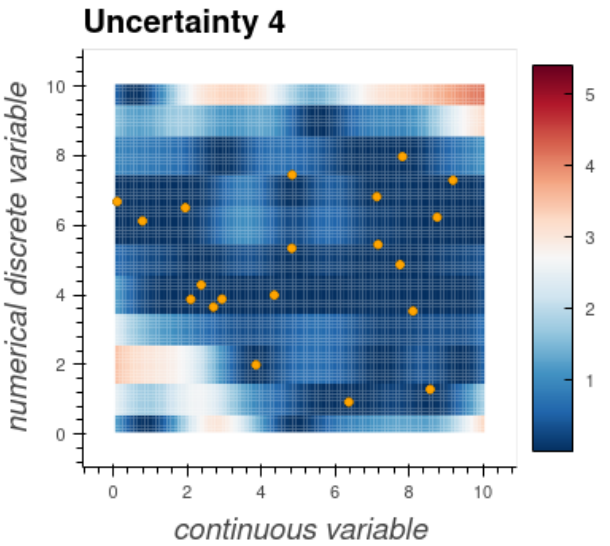
(a)



(b)



(c)



(d)

Figure 4.16: 1st iteration of extended Bayesian Optimization applied to all 3 problems ($\sigma_n^2 = 0$). The plots include mean prediction and uncertainty for categories 3 and 4. The continuous variable is shown on the x-axes and the numeric discrete one on the y-axes. The Acquisition Function and plots for categories 1 and 2 are shown in figures 4.15 and 4.14

5 RESULTS & DISCUSSION

In this chapter, the results of chapter 4 are discussed. The goal was to find methods for selected problems, which were motivated from industry. These problems were applying the Bayesian Optimization (BO) to variables with numeric discrete and categorical values as well as non-controllable inputs, which were called context. In order to achieve this goal, the classical BO was extended by three solution approaches each solving one problem. Finally, the three non-classical properties were combined in a 3D variable space and the BO was extended for all three solution approaches.

5.1 NUMERIC DISCRETE VARIABLE

The problem that occurred when the classical BO was applied to a numeric discrete variable, was the Gaussian Process (GP)'s inability to treat the values as discrete intervals. Therefore, the GP calculated the distance between individual points instead of intervals. In order to solve this problem, a discrete transformer T was placed in the covariance function. This transformer mapped the input values to discrete intervals and assigned a unique representation to them. Afterwards, these representations were passed to the distance measure d. T has to be defined application specific, dependent on the discreteness of the input. This solution was taken from [6].

However, the observed samples were assumed to be noisy, even though they were not (for classical BO see figures 4.1 and 4.2 and for extended the figure 4.4). Conversely, noisy observations were assumed to be noise-free (for classical BO see figures B.1, B.2 and for extended see figure B.3). Therefore, in the following the model parameters length scale and assumed noise variance are considered in more detail. The selection of these two parameters was based on the trade-off between them. Regarding [23], a low length scale leads to a quickly varying mean prediction which catches *all* observations. In this case, the assumed noise variance is near zero.
In the opposite, a high length scale leads to a smooth mean prediction which does *not* catch all observations and an increase of the assumed noise variance is the consequence. 15 iterations for all BO applied to numeric discrete variable were executed and the course of the length scales and assumed noise variances are shown in figure 5.1 and 5.2 respectively. After #iteration = 5, the extended BO in the upper plot has an almost constant l value (#iteration $\geq$ 5). From there, l was for ex-

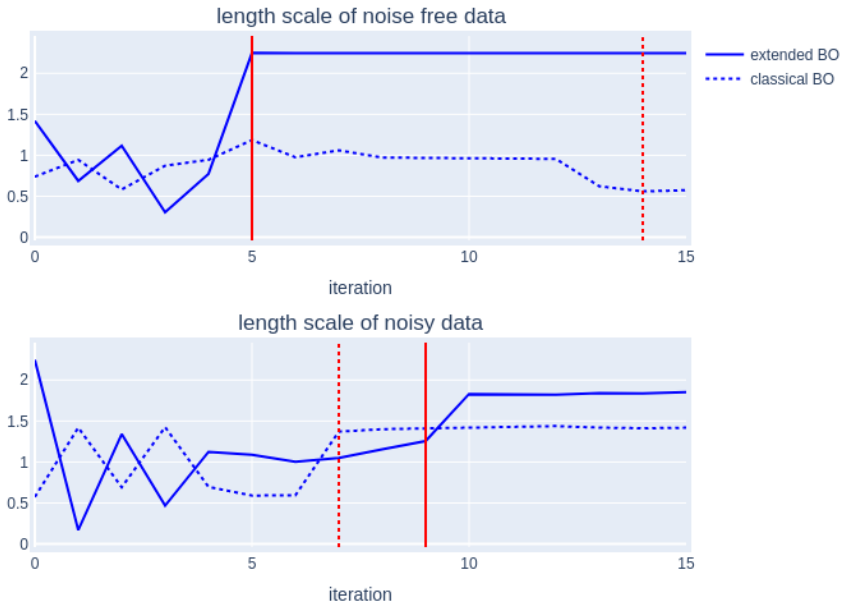


Figure 5.1: Length scales for 15 iterations ($\sigma_n^2 = 0$). The vertical red lines mark the iterations of interest for classical (dotted) and extended Bayesian Optimization (solid) at (5) and (14) for the upper and (7) and (9) for the lower plot.

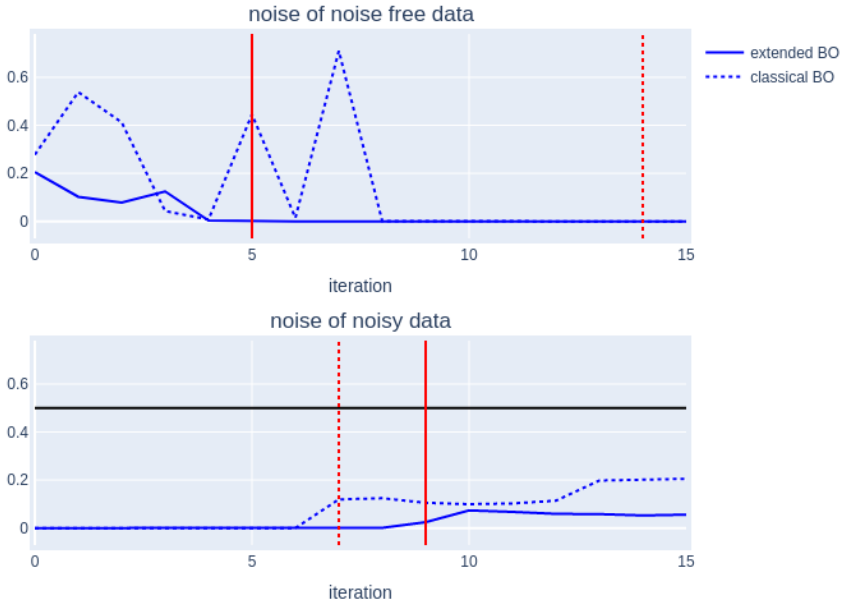


Figure 5.2: Assumed noise variances of 15 iterations. The vertical red lines mark the iterations of interest for classical (dotted) and extended Bayesian Optimization (solid) at (5) and (14) for the upper and (7) and (9) for the lower plot. The horizontal black line marks the true noise variance ($\sigma_n^2 = 0$).

tended BO always higher than for classical. However, the lower model

complexity was sufficient to catch all observations, see 5th in figure 5.4. The noise level for this model was assumed to be near zero, except in the beginning. However, $\sigma_n^2 = 0.5$ (black horizontal line in figure 5.2) was not reached by any model. The figures from section 4.1 were also taken from these iterations.

The iterations of interest were marked on these plots with a red solid vertical line for the extended BO and red dotted vertical line for the classical one. They were selected either to show the method did not converge (for classical BO applied to noise-free data (#iteration = 14)) or to show it did (all others). It was not systematically shown that all courses either converge or not. But in case, these plots show a pattern, which looks like convergence, corresponding iterations were examined in more detail. In figure 5.1, the convergence pattern for extended BO applied on noise-free data is visible, see #iteration = 5. The course of the length scale does not change for further iterations. In figures 5.3 and 5.4 the 4th and 5th iterations are shown.

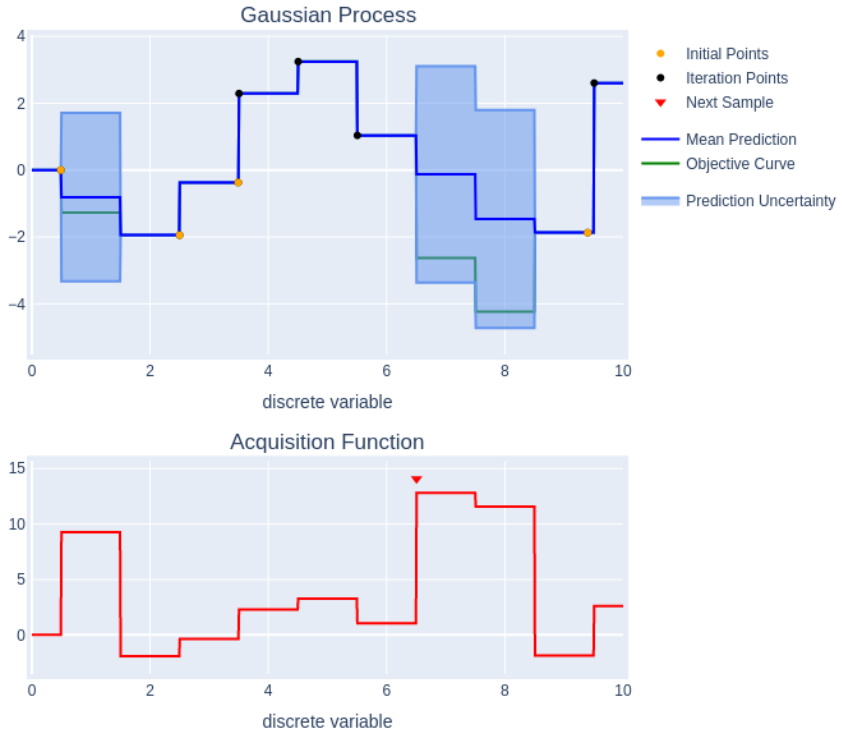


Figure 5.3: 4th iteration of extended Bayesian Optimization applied to numeric discrete variable ($\sigma_n^2 = 0$). The mean prediction (blue line) and prediction uncertainty (blue area) of the Gaussian Process are shown in the upper figure. The objective function (green line) is also visualized. Initial (orange) and iterated samples (black) are marked as points. In the lower figure, the Acquisition Function (red line) is shown. The proposed sample is marked as red triangle.

At the 4th iteration, the proposed sample ($x^* = 6.51 \in I_7$) is not the optimal one but belongs to an unobserved interval. However, the optimal function value was already observed. During the next iteration, see figure 5.4, the uncertainty about the mean prediction is near zero everywhere and the objective function was approximated almost perfectly. There is a slightly difference between mean prediction and true function value observable for I_2. The next proposed sample is the optimal one which was already observed. For further iterations, this sample was proposed and evaluated. Therefore, the extended BO applied to noise-free data converged. A look at figure B.4 shows that the classical BO (noise-free data) did not converge at 14th iteration: the proposed sample is not the highest observed so far. The remaining iterations are shown in figure B.6 (classic BO 7th iteration) and B.5 (extended BO 9th iteration). Both were converged.

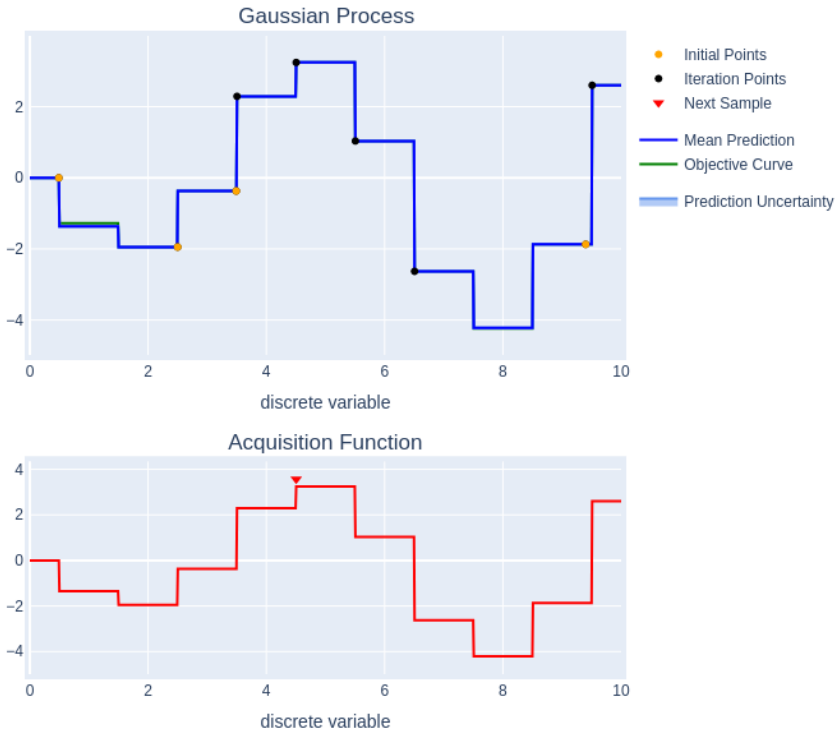


Figure 5.4: 5th iteration of extended Bayesian Optimization applied to numeric discrete variable after convergence ($\sigma_n^2 = 0$).

In figure B.3, the 1st iteration of extended BO is shown (noisy data). In this figure, the mean prediction for unobserved intervals are all zero. The length scale is, at this iteration, smaller than 1, see length scale course in figure 5.1. This parameter defines the radius of the influence for an observation. The smallest distance between two intervals is 1. Therefore, the minimal distance is higher than the length scale no ob-

servation from one interval has an effect to another. In this case, a default mean value is assigned, which is 0 for the used model[23].

5.2 CONTINUOUS CONTEXT

Two problems can occur when the classical BO is applied to a target metric, which is disturbed by a continuous context. Note that the context changes its current value automatically, see section 4.2.

1. The GP does not consider the context and models only the controllable inputs. In this case, there is high variation in data, because the context influence is not modelled. This example is shown in figure 4.5. The mean prediction there is smooth and the uncertainty is high at the observed points. The Acquisition Function (ACQ) proposes a sample which was already observed during last iteration. This can be caused by the uncertainty which does not shrink to zero at the observed points. This behavior is undesired regarding resource consumption (proposed sample was close to an observed one) and the modelling (high variation in data, smooth mean prediction).

2. The GP considers both, continuous context and variable. In this case, not all modelled inputs are controllable. The ACQ does not distinguish between those and proposes the next evaluation by adjust the values for context and variable. This can lead to not feasible propositions. The proposed sample and the evaluated one are not identical. A similar behavior is described in [6]: When the classical GP is applied to numeric discrete variable and the ACQ suggestes the next sample, this sample is rounded to an integer. Then, proposed samples are not identical with the realized evaluations. The GP considers only the evaluated value and the ACQ may repeatedly propose the same sample. This behavior again results in resource waste regarding time and money.

A similar reaction as in figure 4.5 when the classical BO was applied to a variable without considering the context, is described in [5]. In that example, the data contains heavy tailed, non-Gaussian noise which is falsely modelled as Gaussian noise. Then, there is high variance in the observed function values but different from the classical BO shown in figure 4.5, "the posterior GP is heavily effected by the outliers". However, the GP in figure 4.5 is smooth and less effected by the strong varying data.

The solution for this problem was taken from [11]. The vectors of the GP are not passed directly to the ACQ, instead restricted vectors are passed. This restriction was based on the current context's value and therefore only includes values of the controllable variables. This method is called Context Upper Confidence Bound (C-UCB). Both GPs of the extended

BOs had a similar mean predictions as the objective function regarding observed areas.

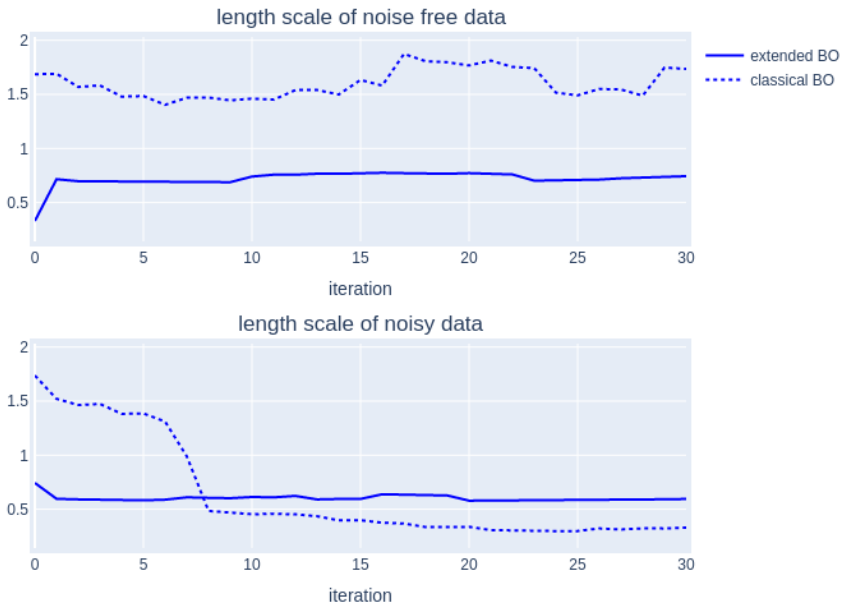


Figure 5.5: Length scales for 30 iterations of extended and classical Bayesian Optimization applied to numeric discrete variable ($\sigma_n^2 = 0$).

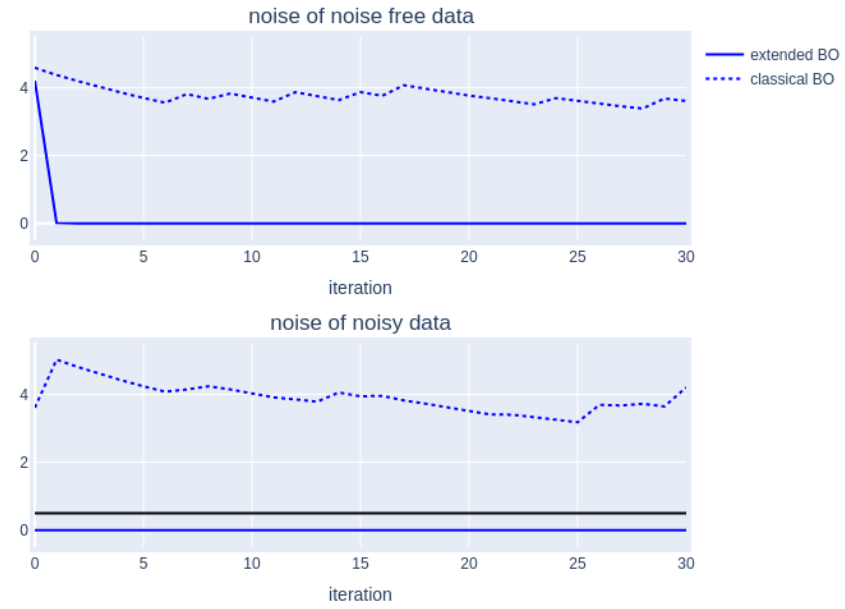
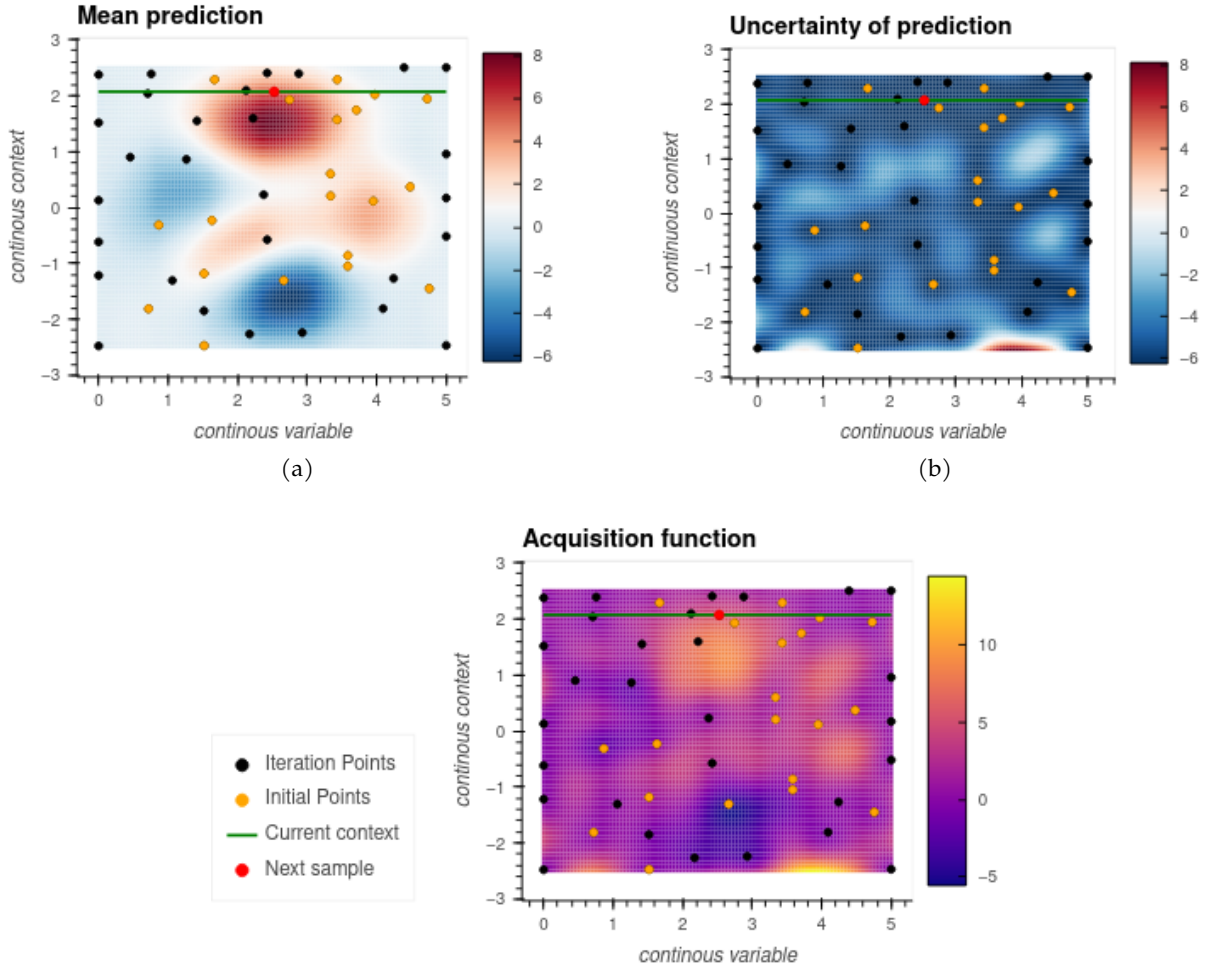


Figure 5.6: Assumed noise variances of 30 iterations. The horizontal black line marks the true noise variance ($\sigma_n^2 = 0.5$).

In figure 5.5 and 5.6 the model parameters length scale and assumed noise variance for 30 iterations are shown. The number of iterations were increased compared to section 5.1, because of the higher input di-

mension. The length scale of the extended BO was for noise-free data always lower than for classical BO. The assumed noise variance for the extension was lower for both, noise-free and noisy data and higher for the classical methods. This fits the trade-off between l and $\hat{\sigma}_n^2$, see section 5.1. For both, the true noise variance was not assumed correctly (see black horizontal line in figure 5.6). The fall down of the length scale for classical BO applied to noisy data show no interesting behavior regarding the performance of the model. The model complexity was increased but the assumed noise was still high. However, there is no convergence observable and therefore no interesting iterations were marked. The causes for the lack of convergence can be the low number of observations. Maybe after increasing the number of iterations or initial points, a convergence can be achieved. In order to show the fitting progress, the 30th iteration of extended BO for noise-free data is shown in figure 5.7. The iteration points are almost equally distributed over the whole input domain. The mean prediction is similar to the objective function, see figure 4.6. However, the observations are for high and low context values closer than in between. This behavior is caused by the sinusoidal simulation of the context where the deviation is lower at the edges and higher between them. The 30th iteration for noisy data is shown in the appendix in figure B.9.



(a) (b) (c)

Figure 5.7: 30th iteration of extended Bayesian Optimization applied to continuous variable and continuous context ($\sigma_n^2 = 0$). In panel (a) the 2D mean prediction of the Gaussian process is shown together with the current context value (green horizontal line 2.06) and the observations (orange and black points). Panel (b) presents the 2D prediction uncertainty and panel (c) the Acquisition Function.

5.3 CATEGORICAL VARIABLE

The problem that occurred when the classical BO was applied to a categorical variable, was the definition of distances or similarities. Categories are not numeric data and therefore not applicable to metric distance measures. They need to be numerically represented in order apply them to distance measures. The distances between variable values are important due to the assumption that small distances in input domain lead to small distances in output domain. The selected approach was motivated from Natural Language Processing, where words are represented numerically, see section 2.5. First, an embedding

(a simple linear net without activation function) was implemented at the entry of the covariance function. Second, at each iteration, the observations were used as training samples and the covariance function was trained together with the embedding. During the training process, the embedding's weights and the Radial Basis Function (RBF)'s length scale and output scale parameters were adjusted. After finishing training, the output neuron of the embedding gives information about the numeric representation for each category. Because embedding and RBF are trained together, they are not separable during application. The problem being solved here, was mapping the categories into numerical space. In figure 4.11 the 1st iteration of extended BO applied to noise-free data is shown. The representations of categories 1 and 2 are close. This is correct regarding the simulation, see figure 4.10. The category 3 is located between categories 1 & 2 and category 4. However, these four categories effect each other. This effect was not simulation. However, there are not much observation made in categories 3 and 4. In figure 4.12 the extended BO applied to noisy data is shown. The categories 1, 3 and 5 have no effect on any other category, see the color code. But the categories 2 and 4 were represented almost identically even though they were simulated unsimilar. Length scale and output scale for noise-free and noisy data are shown in figure 5.8 and 5.9. For noise-free, the length scale was higher than for noisy data. After the 9th iteration, the extended BO applied to noise-free data shows a convergence pattern. The vertical red line marks the iteration of interest. For noisy data, the iteration of interest are 9 ans 26.

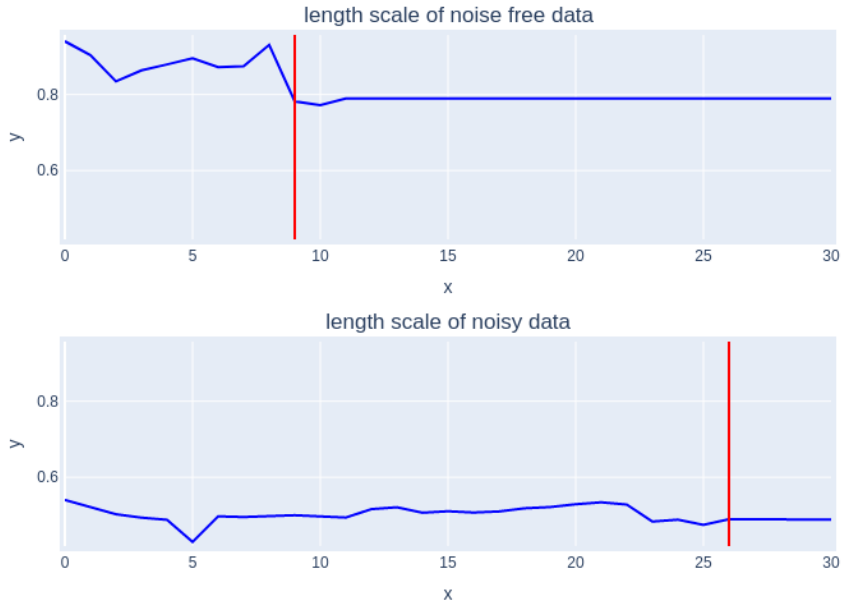


Figure 5.8: Length scales for 30 iterations. Red line marks the iteration of interest (9) and (26).

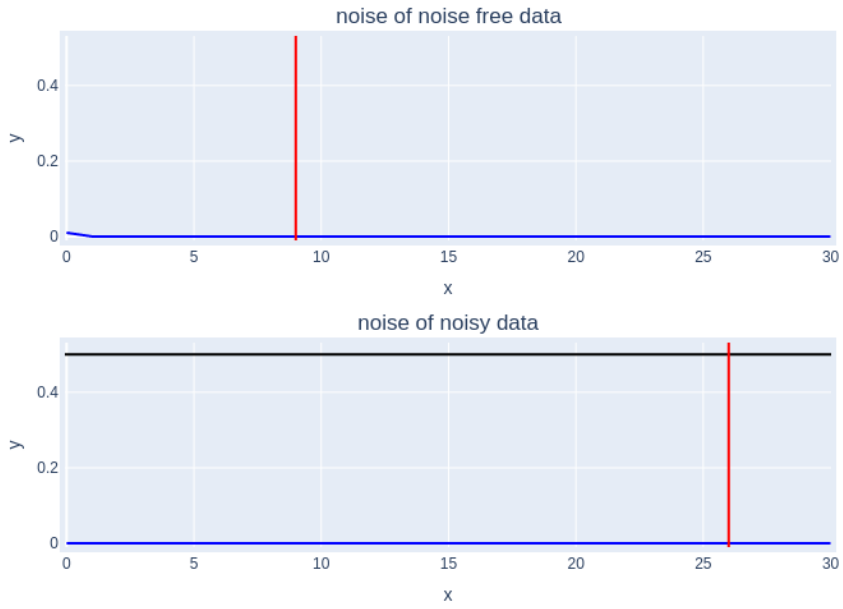


Figure 5.9: Assumed noise variances of 30 iterations. The horizontal black lines mark the true noise variance. Red line marks the iteration of interest (9) and (26).

In figure 5.10 the 8th iteration is shown. The proposed sample there is not the global optimum, see figure 4.10 and belongs to category 3 (blue). The distances between the categories are similar to the 1st iteration, see figure 4.12. During the next iteration the proposed sample is near the optimum, see figure 5.11. The categories are now represented slightly different. The categories 1 and 2 are still very close. But the categories 3 and 4 moved closer and are represented almost identically. Even though this was not simulated (they were slightly different scaled regarding the function values), their similarity was recognized. The category 5 has still high distance to the others. The proposed sample belongs to the far highest function value observed so far. For further iterations no other sample is proposed and the BO converged.

For noisy data the 26th iteration is the iteration of interest. Until then, the categories 4 and 2 were represented very similar, even though they were not, see figure 5.12. So there is no convergence observed. But after the 28th iteration, they were mapped further away, see figure 5.13. The correct representation was reached after a high number of iterations (28). The assumption about this is, that the categories 2 and 4 were not recognized as promising categories. They were assumed to lead to no high function values, which is important for maximization problems. After mapping these categories to different representations they did not effect each other anymore. The length scale plot gives information about the model complexity: the wrong model (noisy data) hat higher model complexity than the correct one (noise-free data).

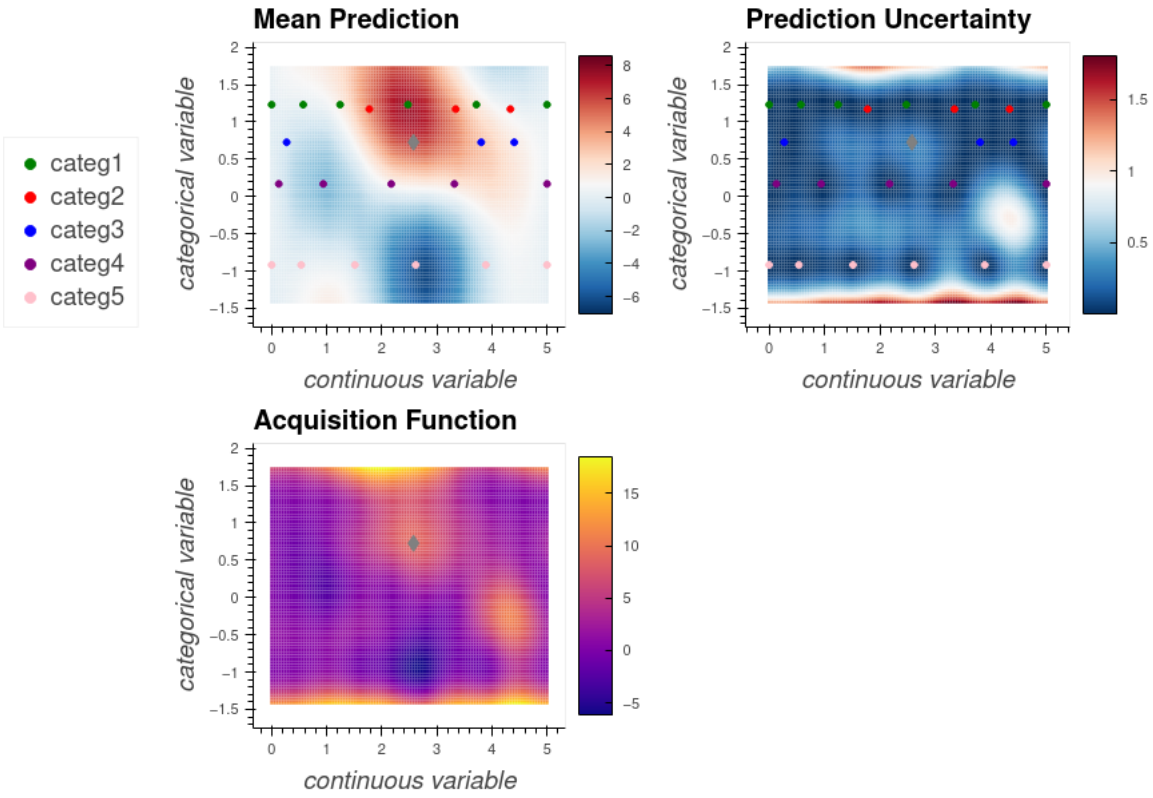


Figure 5.10: 8th iteration of extended Bayesian Optimization applied to continuous and categorical variables ($\sigma_n^2 = 0.0$): not converged yet. The 2D mean prediction and the 2D uncertainty plot are shown together with the category observations (colored points). The Acquisition Function is shown together with the next sample to evaluate (grey diamond).

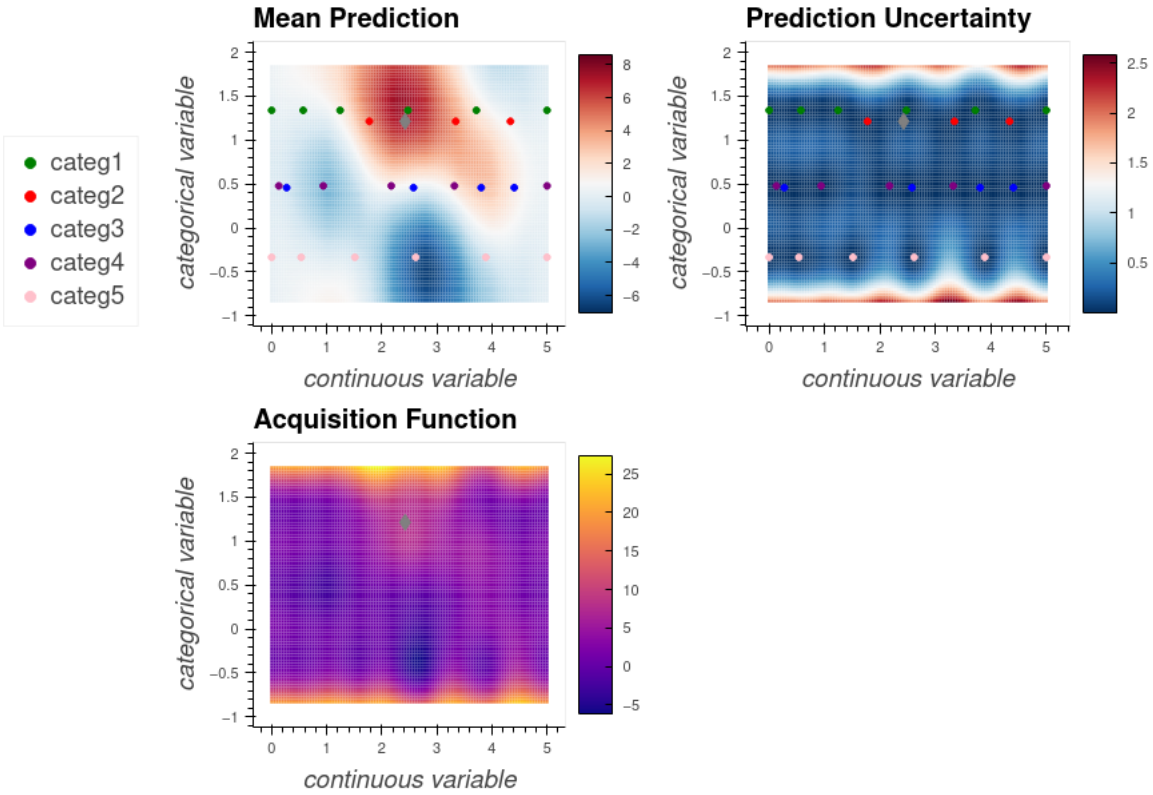


Figure 5.11: 9th iteration of extended Bayesian Optimization applied to continuous and categorical variables ($\sigma_n^2 = 0.0$): converged.

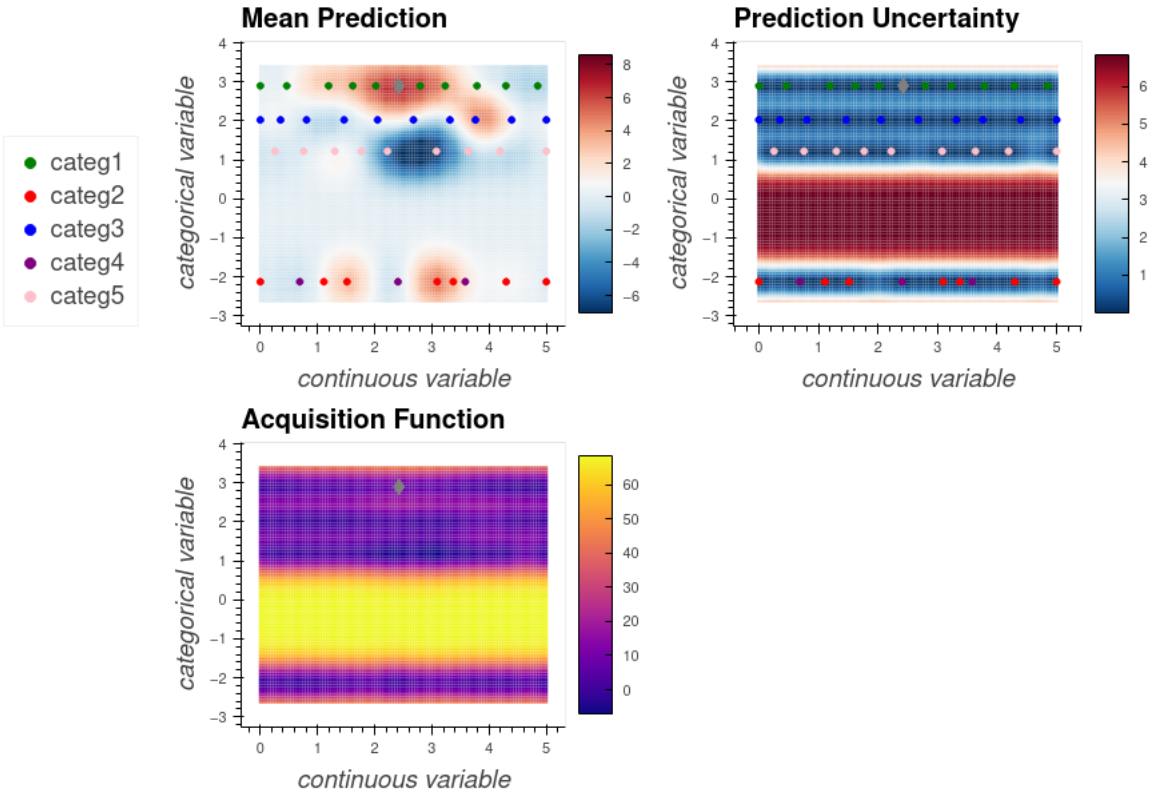


Figure 5.12: 26th iteration of extended Bayesian Optimization applied to continuous and categorical variables ($\sigma_n^2 = 0.8$).

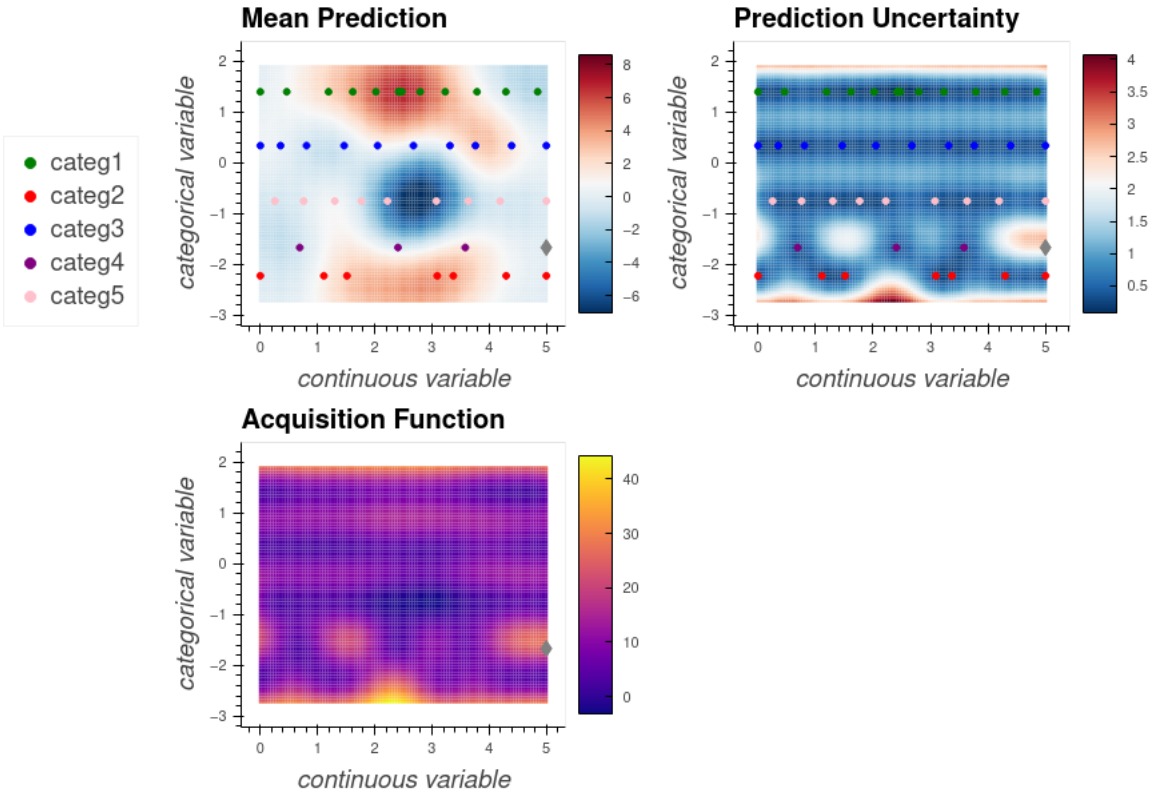


Figure 5.13: 28th iteration of extended Bayesian Optimization applied to continuous and categorical variables ($\sigma_n^2 = 0.8$).

5.4 COMBINED PROBLEMS

After implementing and testing the approaches for every single non-classical variable, all of them were combined in a 3D variable space. The variables were classic continuous and numeric discrete variable and categorical context. The categorical context had 4 possible categories, whereby the first and the second were simulated similar as well as the third and the fourth. The BO used here was initialized with 80 samples, 20 for each category. The number of initial samples was taken high compared to the other investigations, so that similarities between categories can be recognized in a 3D input space. The combination of the problems was taken to examine whether new problems occur. For noise-free data, the extended BO mapped similar categories close to each other during first iteration. For noisy data, only the similarities for categories 3 and 4 were recognized at 1st iteration. In the figures 5.14 and 5.15 the length scale and the noise for each iteration is shown. For this 3D example, 15 iterations were performed.

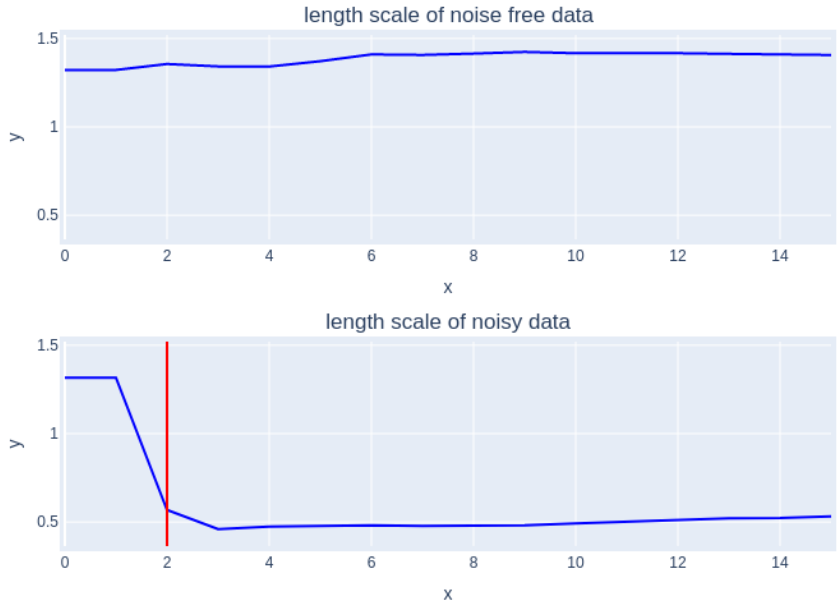


Figure 5.14: Length scales for 15 iterations. The red line marks the iteration of interest (2).

For noise-free data, the length scale is almost constant. During the 1st iteration, the similarities and unsimilarities were recognized correctly. However, for noisy data, there is a iteration of interest at 2nd iteration. In figure 5.17 and 5.18 the 2nd iteration for noisy data is shown and in figure 5.16. The similar categories 1 and 2 are mapped closer than at 1st iteration:

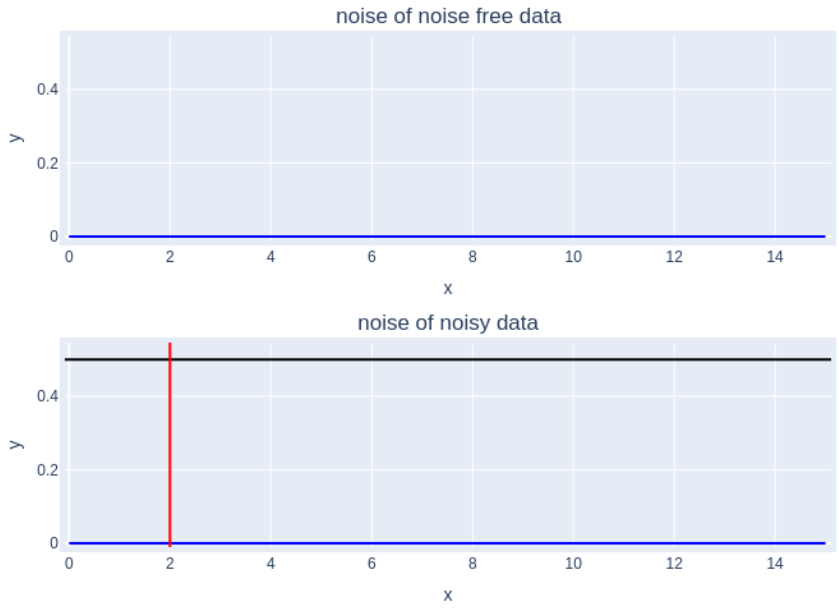


Figure 5.15: Assumed noise variances of 15 iterations. The horizontal black line marks the true noise variance. The red line marks the iteration of interest (2).

$$
\begin{aligned}
r_1 &: -2.2644 \\
r_2 &: -2.3761 \\
r_3 &: \ 2.2885 \\
r_4 &: \ 2.1703.
\end{aligned}
$$

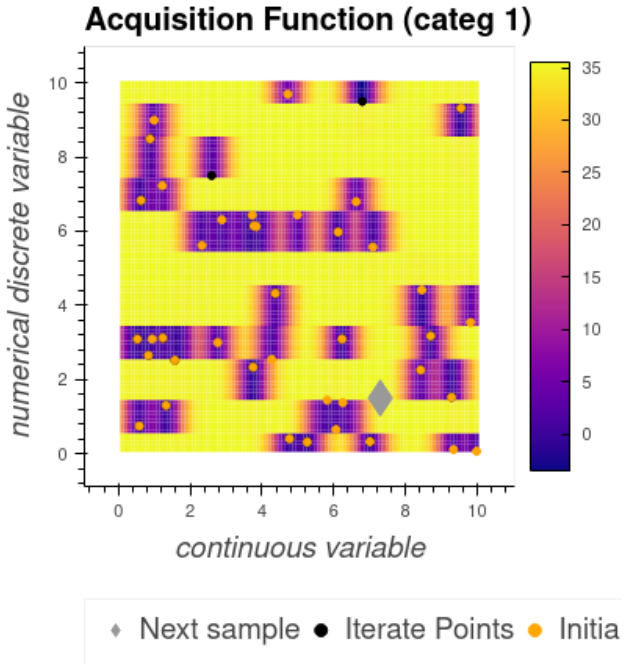
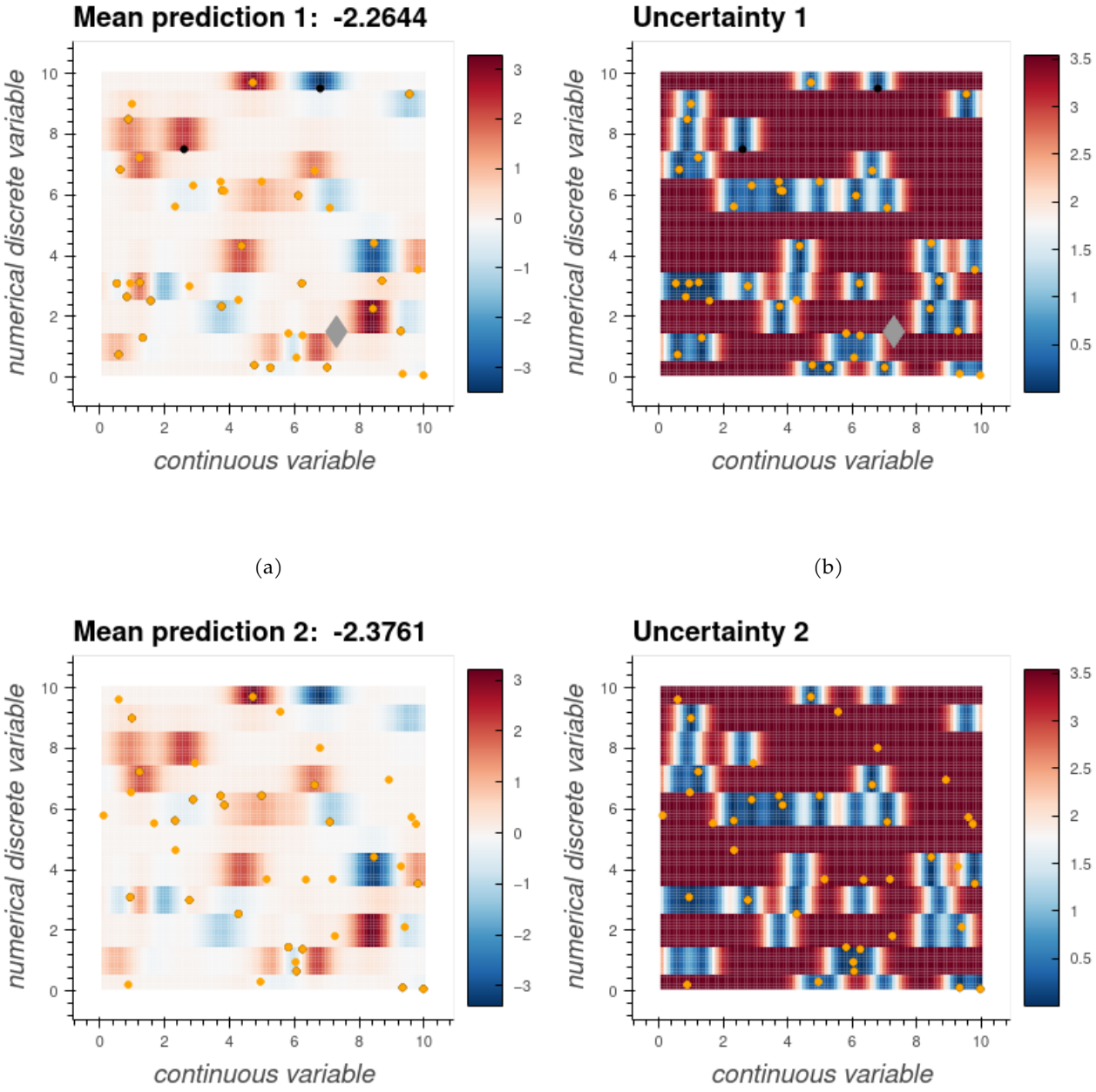


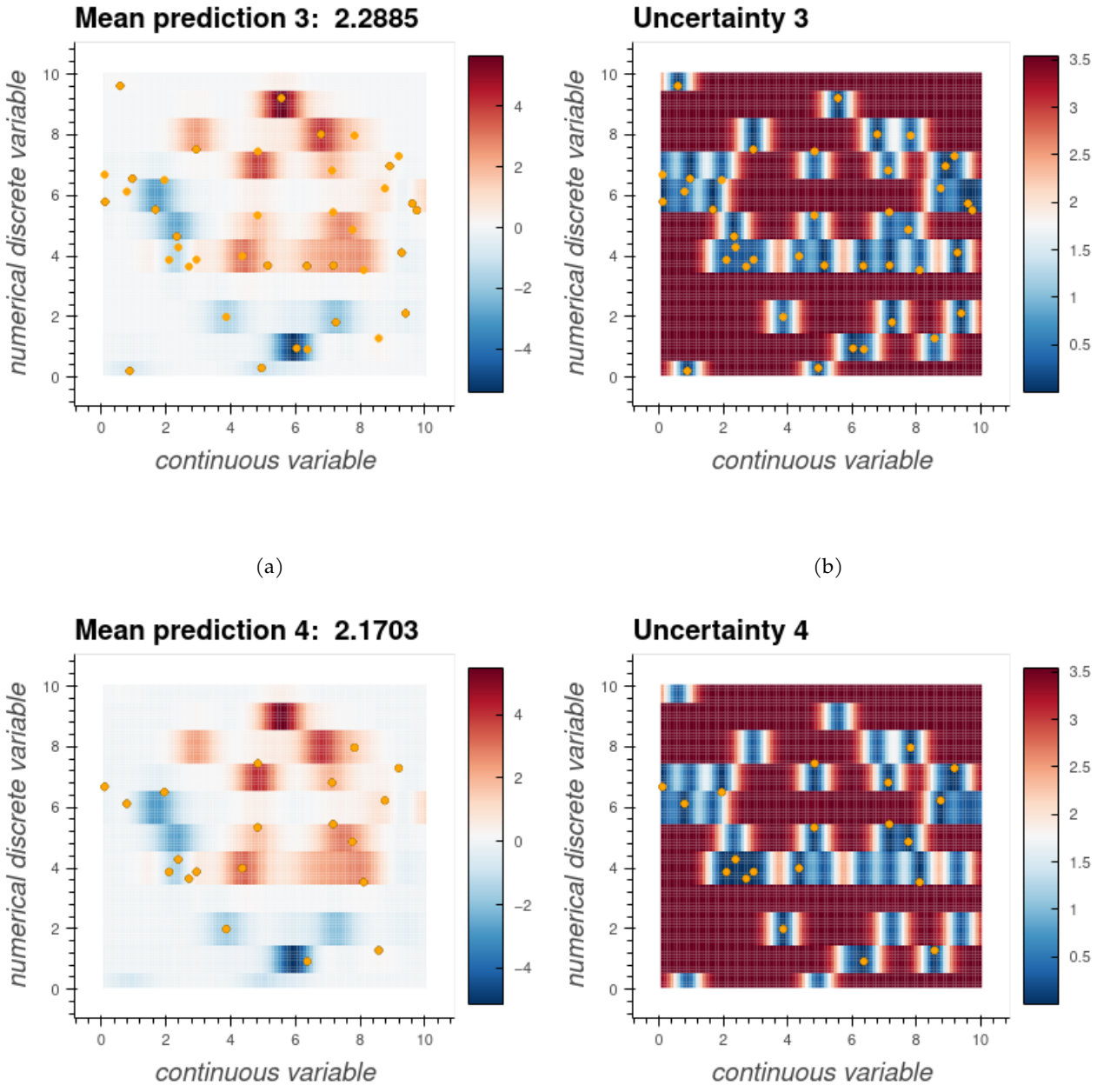
Figure 5.16: Acquisition function of 2nd iteration. The corresponding mean predictions and uncertainties for categories 1-4 are shown in figures 5.17 and 5.18



(a) (b)

(c) (d)

Figure 5.17: 2nd iteration of extended Bayesian Optimization applied to all 3 problems (noisy data). The plots include mean prediction and uncertainty for categories 1 and 2. The continuous variable is shown on the x-axes and the numeric discrete one is shown on the y-axes.



(a) (b)

(c) (d)

Figure 5.18: 2nd iteration of extended Bayesian Optimization applied to all 3 problems ($\sigma_n^2 = 0.5$). The plots include mean prediction and uncertainty for categories 3 and 4. The continuous variable is shown on the x-axes and the numeric discrete one is shown on the y-axes.

6

OUTLOOK

In this chapter, an outlook is given for further work.

In order to vanish the problem with categorical values, where the similarities are not recognized correctly, multiple starts of the training could be used. This way, different random initializations of the embedding are realized and therefore the orders of the categories vary. If then between two similar categories no unsimilar one is placed, the embedding has a very good representation of them after a sufficient number of samples. In doing so, the parameter length scale can be an indicator for a good or bad representation, dependent on the model complexity. If two unsimilar categories, which have an unsimilar course, were effecting each other, the length scale is assumed to be very low. The lower the length scale the higher the model complexity. In order to find a good fit by applying multiple starts, a low complexity could be considered when finding the best fit for the categories.

Furthermore, no distinction between nominal and ordinal categorical inputs was made. Considering ordinal categories, if there is a reasonable assumption that the similarities of the categories have the same order as the representations, this order should be taken into account when initializing the weights of the embedding. Of course, this assumption may not always be valid, because the similarities may have nothing to do with the categories' order. In case extreme categories (those which are furthest away) behave similar, this approach would not perform good. Therefore, the assumption about the similarities considering the natural order has to be verified.

Solution approaches for other non-classical inputs can be worked out, such as delayed inputs. Also, a performance analysis about the Bayesian Optimization (BO) regarding input dimension and model parameters β, initial number of samples, etc. can be done in order to get the best performance for diverse dimensions.

Another option is to automatically adjust the trade-off parameter β after each iteration. In the current implementation, this parameter is set once. But a dynamical adjusting could lead to higher performance and faster convergence. Furthermore, an investigation about the best performing Acquisition Function can be made.

BIBLIOGRAPHY

[1] Satya Almasian, Andreas Spitz, and Michael Gertz. "Word Embeddings for Entity-Annotated Texts." In: *Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14-18, 2019, Proceedings, Part I*. Springer, 2019, pp. 307–322.

[2] S Aparicio, MG Hernández, and JJ Anaya. "Influence of environmental conditions on concrete manufactured with recycled and steel slag aggregates at early ages and long term." In: *Construction and Building Materials* 249 (2020), p. 118739.

[3] Franz Brauße, Zurab Khasidashvili, and Konstantin Korovin. "Bayesian Optimisation with Formal Guarantees." In: *arXiv preprint arXiv:2106.06067* (2021).

[4] Ahmet Demirkaya, Jiasi Chen, and Samet Oymak. "Exploring the role of loss functions in multiclass classification." In: *2020 54th Annual Conference on Information Sciences and Systems (CISS)*. IEEE. 2020, pp. 1–5.

[5] Roman Garnett. *Bayesian Optimization*. in preparation. Cambridge University Press, 2022.

[6] Eduardo C Garrido-Merchán and Daniel Hernández-Lobato. "Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes." In: *Neurocomputing* 380 (2020), pp. 20–35.

[7] Cheng Guo and Felix Berkhahn. "Entity embeddings of categorical variables." In: *arXiv preprint arXiv:1604.06737* (2016).

[8] Saad Hikmat Haji and Adnan Mohsin Abdulazeez. "Comparison of optimization techniques based on gradient descent algorithm: A review." In: *PalArch's Journal of Archaeology of Egypt/Egyptology* 18.4 (2021), pp. 2715–2743.

[9] Simon Haydin. *Neural Networks and Learning Machines*. 3rd. Pearson Education, 2009.

[10] Jiri Klemes. *Sustainability in the process industry: integration and optimization*. McGraw-Hill Education, 2011.

[11] Andreas Krause and Cheng Ong. "Contextual gaussian process bandit optimization." In: *Advances in neural information processing systems* 24 (2011).

[12] T Soni Madhulatha. "An overview on clustering methods." In: *arXiv preprint arXiv:1205.1117* (2012).

[13] Andrew McHutchon and Carl Rasmussen. "Gaussian process training with input noise." In: *Advances in Neural Information Processing Systems* 24 (2011).

[14] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations." In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2013, pp. 746–751.

[15] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. "Activation functions: Comparison of trends in practice and research for deep learning." In: *arXiv preprint arXiv:1811.03378* (2018).

[16] Omogbai Oleghe. "A predictive noise correction methodology for manufacturing process datasets." In: *Journal of Big Data* 7.1 (2020), pp. 1–27.

[17] Tony Pourmohamad and Herbert K. H. Lee. *Bayesian Optimization with Application to Computer Experiments*. Springer, 2021.

[18] Lior Rokach and Oded Maimon. "Clustering methods." In: *Data mining and knowledge discovery handbook*. Springer, 2005, pp. 321–352.

[19] Deborah J Rumsey. *Statistics essentials for dummies*. John Wiley & Sons, 2010.

[20] Eric Schulz, Maarten Speekenbrink, and Andreas Krause. "A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions." In: *Journal of Mathematical Psychology* 85 (2018), pp. 1–16.

[21] Stanisław Węglarczyk. "Kernel density estimation and its application." In: *ITM Web of Conferences*. Vol. 23. EDP Sciences. 2018, p. 00037.

[22] Dorina Weichert, Patrick Link, Anke Stoll, Stefan Rüping, Steffen Ihlenfeldt, and Stefan Wrobel. "A review of machine learning for the optimization of production processes." In: *The International Journal of Advanced Manufacturing Technology* 104.5 (2019), pp. 1889–1902.

[23] Christopher K Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[24] Nathan Wollek. "Effiziente Parameteroptimierung von Industrieprozessen durch erklärbares Maschinenlernen." MA thesis. University of Osnabruck, Nov. 2020.

A

OBJECTIVE FUNCTIONS

The objective function[1] with one numerical discrete variable used in section 4.1 was defined as follows:

$$f(x) = \sin(T(x)) \cdot \sin(4) \cdot \sqrt{T(x) \cdot 4}, \tag{A.1}$$

with T as discrete transformer. T maps the input value to a closed interval if the discrete representation $x_d \in \mathcal{X}_d$ is even and to an open interval otherwise. T also describes the definitions of intervals I_i for numeric discrete variables:

$$T(x_i) = I_d = \begin{cases} [x_d - 0.5; x_d + 0.5], & \text{if } x_i \in [x_d - 0.5; x_d + 0.5] \\ & \text{and } x_d \text{ is even,} \\ (x_d - 0.5; x_d + 0.5), & \text{if } x_i \in (x_d - 0.5; x_d + 0.5) \\ & \text{and } x_d \text{ is odd} \end{cases} \tag{A.2}$$

In case of sampling with noise, ϵ_n was defined as follows:

$$\epsilon_n \sim \mathcal{N}(0, 0.5).$$

The objective function[2] with continuous variable and continuous context used in section 4.2 was defined as follows:

$$\begin{aligned} f(x_1, x_2) = 3 \cdot (-x_1 + 3.5)^2 \cdot \exp\left[-(x_1 - 2.5)^2 - (x_2 + 1)^2\right] \\ - 10(-0.5 + 0.2x_1 - (x_1 - 2.5)^3 - x_2^5) \\ \cdot \frac{1}{3} \exp\left[-(x_1 - 1.5)^2 - x_2^2\right]. \end{aligned} \tag{A.3}$$

The objective function for categories 1 and 2 in the 3D example used in section 4.4 were similar to the one described in equation A.1, but with the two inputs continuous and numerical discrete variables instead of

1 This function was motivated by http://clerc.maurice.free.fr/pso/Alpine/Alpine_Function.htm, last visited on 12th April 2022 at 13:23.

2 This function was motivated by https://www.math.uwaterloo.ca/~hwolkowi/henry/reports/talks.d/t09talks.d/09waterloomatlab.d/optimTipsWebinar/html/optimTipsTricksWalkthrough.html, last visited on 12th April 2022 at 15:45.

one variable and the constant 4. The formula is shown in equation A.4, T is again the discrete transformer.

$$f(x_1, x_2) = \sin(x_1) \cdot \sin(T(x_2)) \cdot \sqrt{x_1 \cdot T(x_2)}. \tag{A.4}$$

Note that these functions were used for simulating the objective functions. It may be necessary to compress these horizontally to gain the same results as shown.

B

SUPPLEMENTARY PLOTS

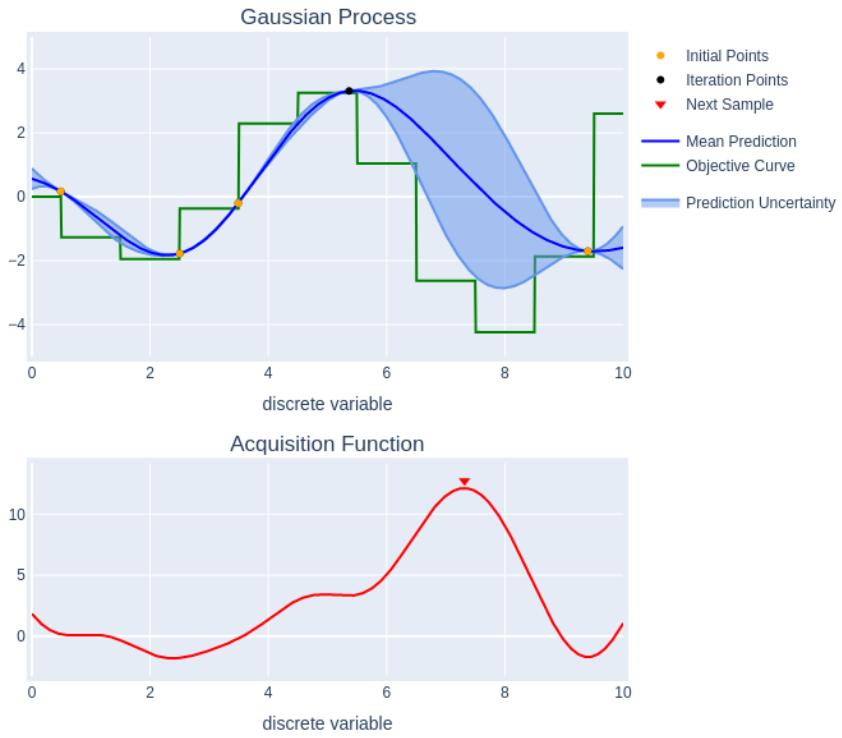


Figure B.1: 1st iteration of classical Bayesian Optimization applied to noisy numeric discrete variable. The mean prediction (blue line) and prediction uncertainty (blue area) of the Gaussian Process are shown in the upper figure. The objective function (green line) is also visualized. Initial (orange) and iterated samples (black) are marked as points. In the lower figure, the Acquisition Function (red line) is shown. The proposed sample is marked as red triangle.

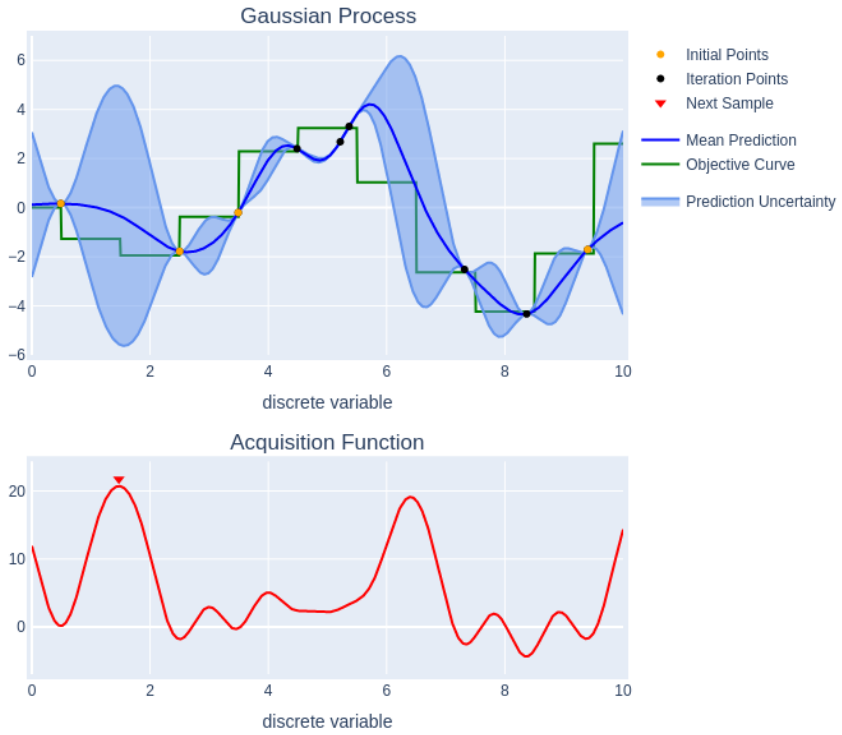


Figure B.2: 5th iteration of classical Bayesian Optimization applied to noisy numeric discrete objective.

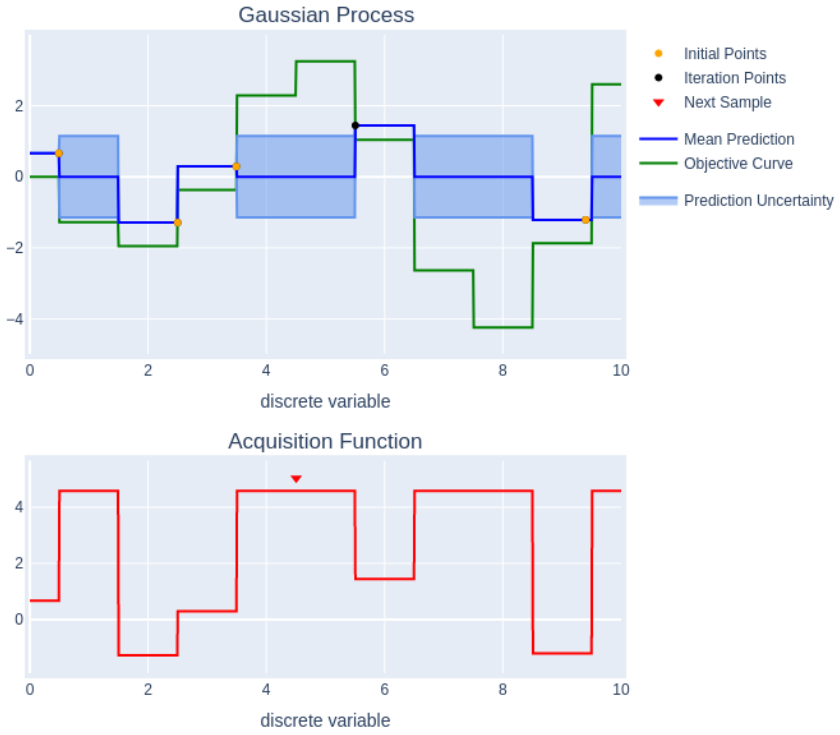


Figure B.3: 1st iteration of extended Bayesian Optimization applied to noisy numeric discrete objective. Note that the mean prediction at unobserved intervals has value near zero.

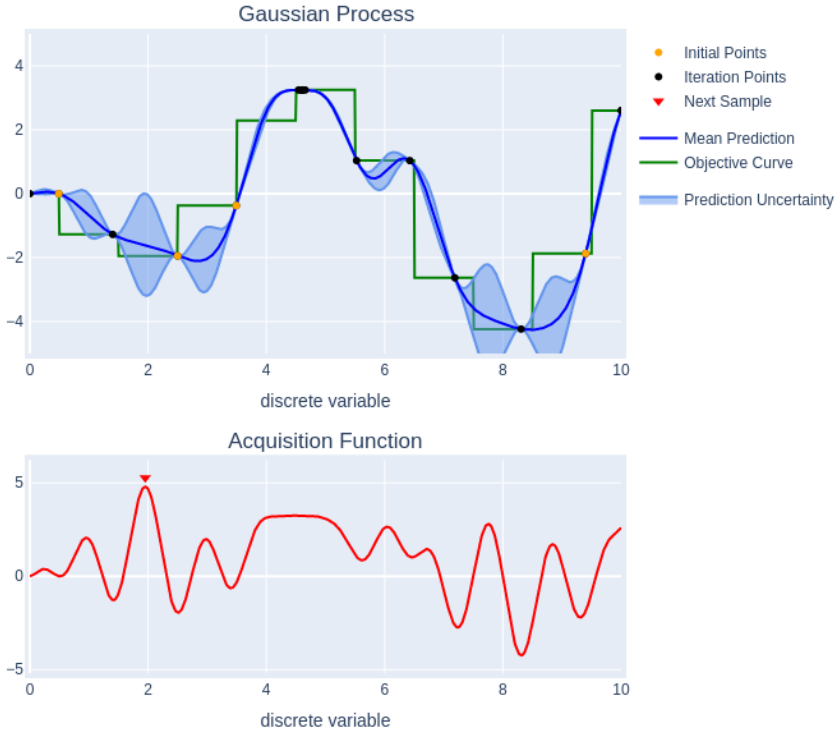


Figure B.4: 14th iteration of classical Bayesian Optimization applied to numeric discrete objective: not converged yet.

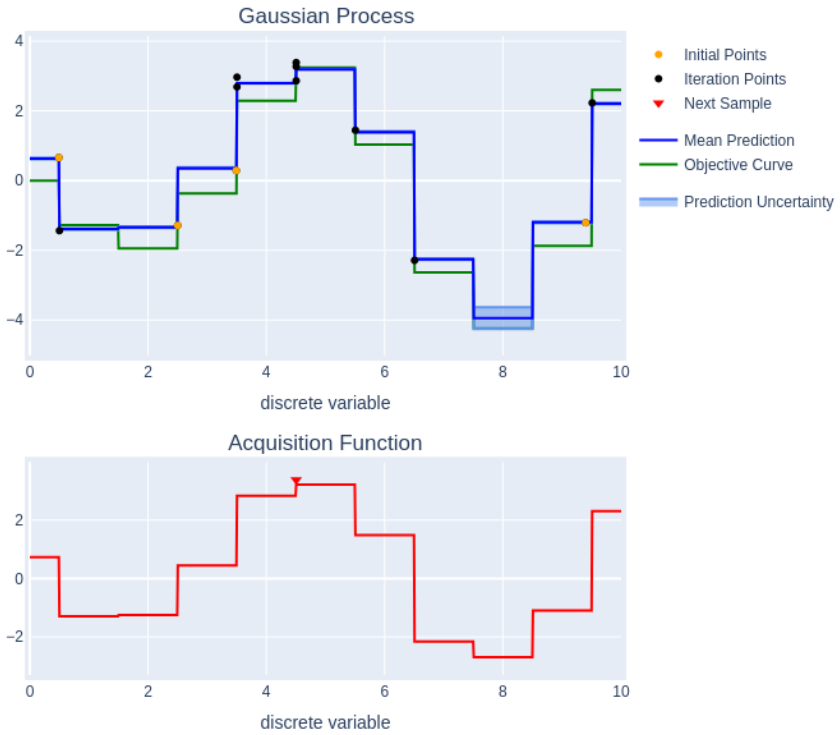


Figure B.5: 9th iteration of extended Bayesian Optimization applied to noisy numeric discrete objective: converged.

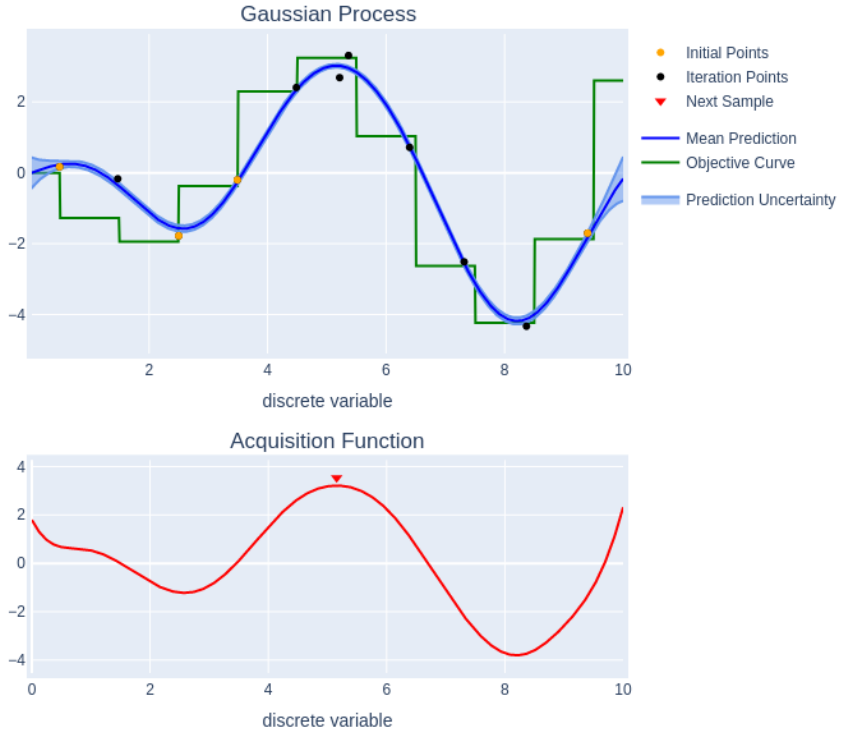


Figure B.6: 7th iteration of classical Bayesian Optimization applied to noisy numeric discrete objective: converged.

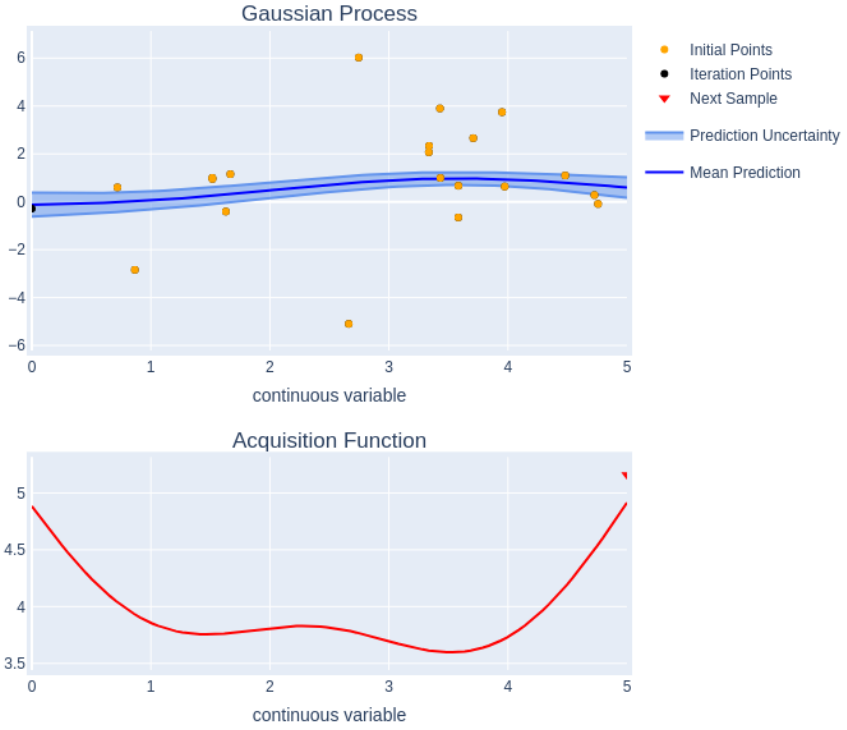
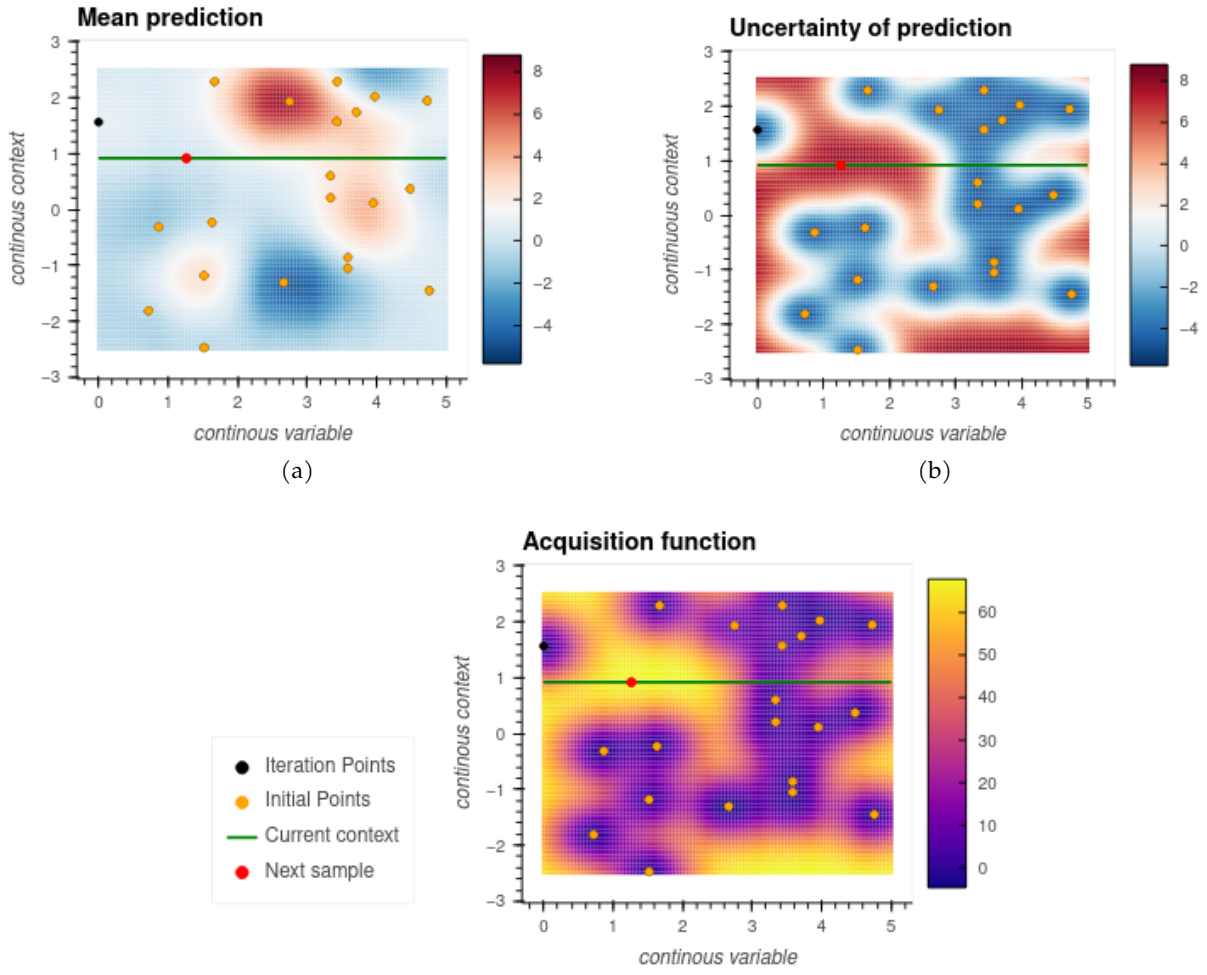
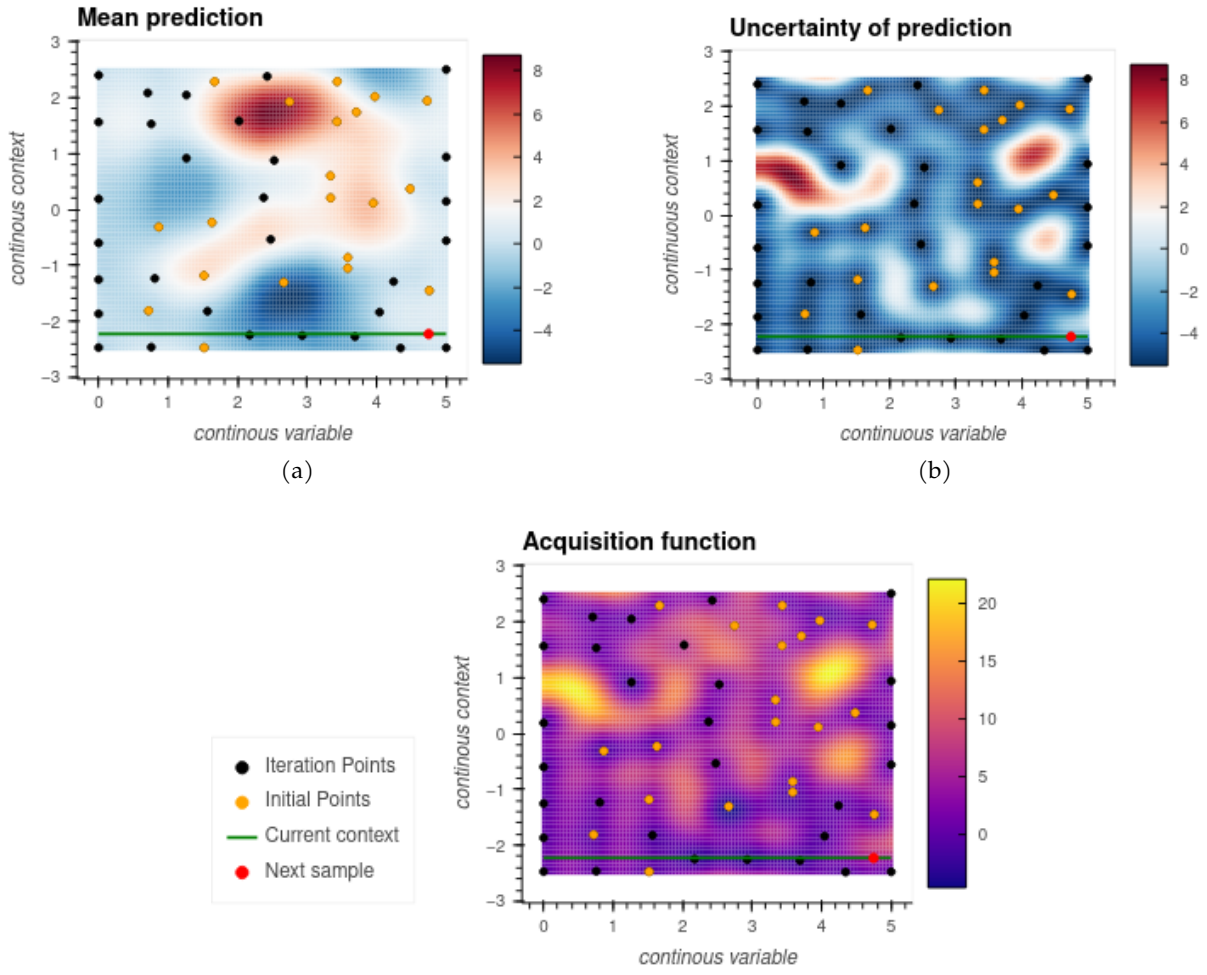


Figure B.7: 1st iteration of classical Bayesian Optimization applied to noisy objective with context ($\sigma_n^2 = 0.5$).



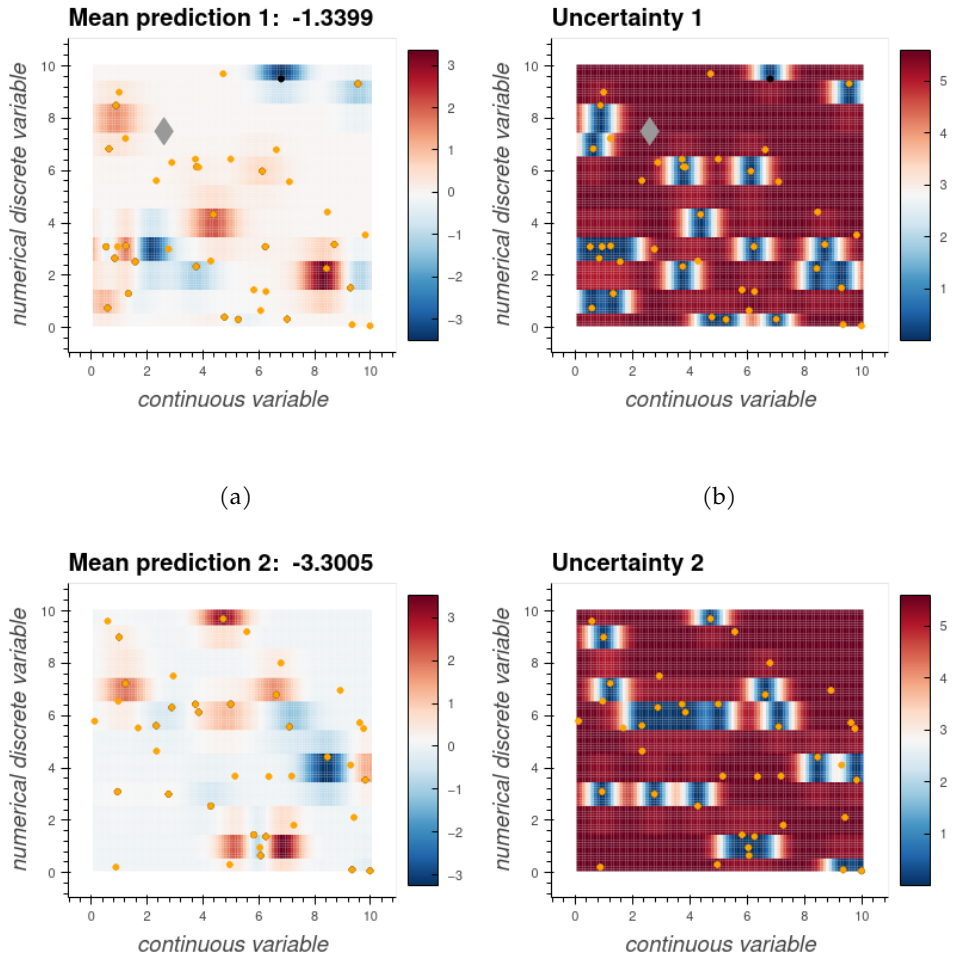
(a) (b) (c)

Figure B.8: 1st iteration of extended Bayesian Optimization applied to continuous variable and continuous context ($\sigma_n^2 = 0.5$). In panel (a) the 2D mean prediction of the Gaussian process is shown together with the current context value (green horizontal line 0.91) and the observations (orange and black). Panel (c) presents the 2D prediction uncertainty and panel (c) the Acquisition Function.



(a) (b) (c)

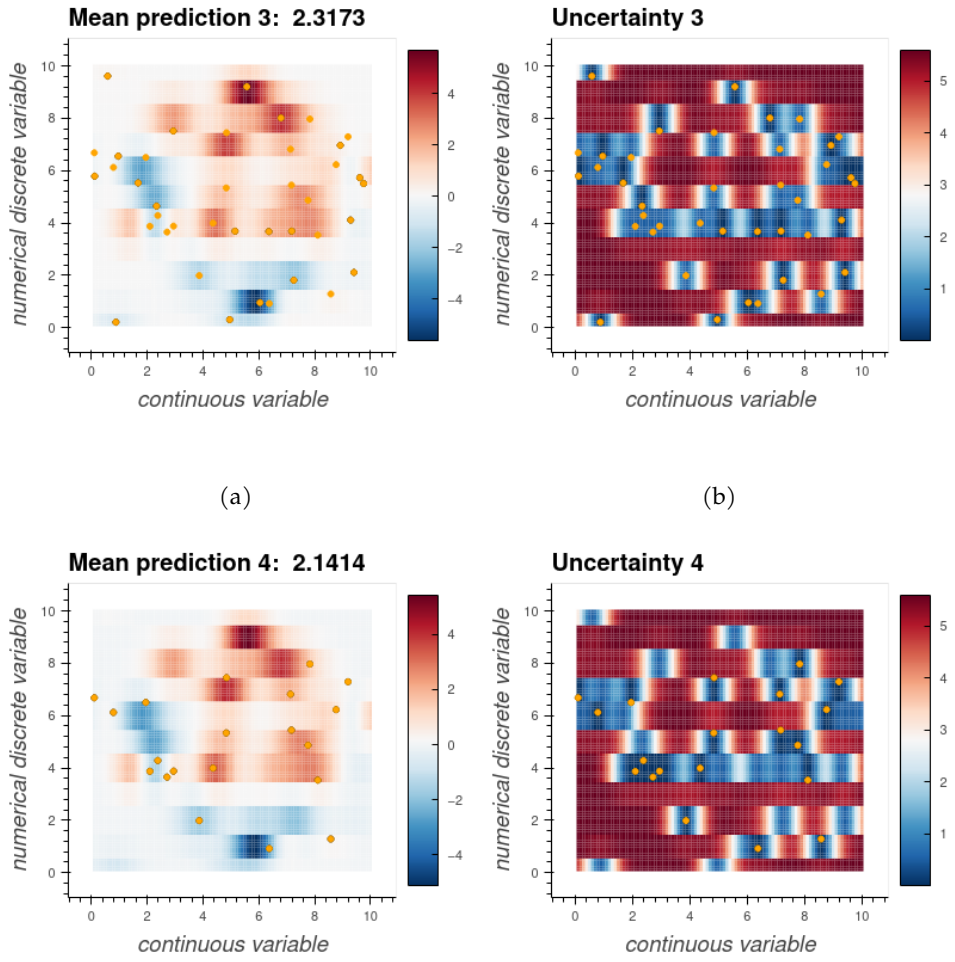
Figure B.9: 30th iteration of extended Bayesian Optimization applied to continuous variable and continuous context ($\sigma_n^2 = 0.5$). In panel (a) the 2D mean prediction of the Gaussian process is shown together with the current context value (green horizontal line -2.22) and the observations (orange and black points). Panel (b) presents the 2D prediction uncertainty and panel (c) the Acquisition Function.



(a) (b)

(c) (d)

Figure B.10: 1st iteration of extended Bayesian Optimization applied to all 3 problems ($\sigma_n^2 = 0.5$). The plots include mean prediction and uncertainty for categories 1 and 2. The continuous variable is shown on the x-axes and the numerical discrete one on the y axes.



(a) (b)

(c) (d)

Figure B.11: 1st iteration of extended Bayesian Optimization applied to all 3 problems ($\sigma_n^2 = 0.5$). The plots include mean prediction and uncertainty for categories 3 and 4. The continuous variable is shown on the x-axes and the numerical discrete one on the y axes.