Gaussian Processes for Global Optimization

Ludwig Winkler
Methods of Artificial Intelligence Group
Technical University of Berlin
coolbro@stanford.edu.com

Sami Ede
Methods of Artificial Intelligence Group
Technical University of Berlin
sami.ede@student.tu-berlin.de

Ansgar Dietrich
Methods of Artificial Intelligence Group
Technical University of Berlin
coolbro@mit.edu.com

Abstract

Gaussian processes are a powerful Bayesian inference method which are able to infer information globally from a limited set of observations. This becomes especially important for objective functions in optimization problems which are computationally expensive to evaluate. Gaussian processes are a natural fit for optimizing such functions due to their Bayesian methodology. We derive Gaussian processes from a multi-variate Gaussian distribution and show how Gaussian processes can be extended with additional features such as derivative observations. Ultimately we will apply Gaussian processes to Bayesian optimization and explain it's methodology. Lastly a series of practical application ranging from Kriging, to hyperparameter search and quantum chemistry are presented in which the advantages of Bayesian optimization are demonstrated on real-world applications.

1 Introduction

Many problems in science and engineering can be formulated as a mathematical optimization problem in which an optimal solution is sought, either locally or globally. The field of global optimization is the application of applied mathematics and numerical analysis towards finding the overall optimal solution in a set of candidate solutions. Local optimization is considered an easier problem, in which it suffices to find an optimum which is optimal with respect to its immediate vicinity. Such a local optimum is obviously a suboptimal solution and, while harder to find, global optima are more preferred.

Generally, optimization problems are formulated as finding the optimal solution which minimizes, respectively maximizes, a criterion, which is commonly referred to as the objective function. Further constraints on the the set of solutions can be formulated, such that only a subset of solutions are permissible as candidates for the optimum.

Optimization is commonly done in an iterative manner where the objective function is evaluated for multiple candidate solutions. Due to the iterative nature, it becomes desirable to evaluate this function as few times as possible over the course of the entire optimization, which becomes even more crucial when the evaluation of the objective function itself is costly. Therefore, it would be advantageous to infer information about the objective function beyond the evaluations themselves, which only provide punctual information.

Bayesian inference models provide such advantages since they compute predictive distributions instead of punctual evaluations. One class of Bayesian inference models are Gaussian processes (GP), which can be applied to model previous evaluations of the objective function as a multi-variate

Gaussian distribution. Given such a Gaussian distribution over the previous evaluations, information can be inferred over all candidate solutions in the feasible set at once.

2 Gaussian Processes

In most situations where observations have many small independent components, their distribution tends towards the Gaussian distribution. Compared to other probability distributions, the Gaussian distribution is tractable and it's parameters have intuitive meaning. The theory of the central limit theorem (CLT) makes the Gaussian distribution a versatile distribution which is used in numerous situations in science and engineering.

A convenient property of the Gaussian distribution for a random variable X is its complete characterization by its mean μ and variance Σ

$$\mu = \mathbb{E}[X] \tag{1}$$

$$\Sigma = \mathbb{E}[(X-\mu)^T(X-\mu)] \tag{2}$$

Mathematically, a multivariate Gaussian for a vector $\mathbf{x} \in \mathbb{R}^d$ is defined by its mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance function $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$:

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|^2}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right] \tag{3}$$

$$\propto \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right] \tag{4}$$

A useful property of the Gaussian distribution is that its shape is determined by its mean and covariance in the exponential term. This allows us to omitt the normalization constant and determine the relevant mean and covariance terms from the exponential term, as seen in (4).

Let $y = f(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^d$ and $y \in \mathbb{R}$ be the function which we want to estimate with a Gaussian Process. Furthermore, let $\mathcal{D} = (\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_i, y_i)\}_{i=0}^N$, with $\mathbf{X} \in \mathbb{R}^{N\times d}$ and $\mathbf{y} \in \mathbb{R}^N$, be our training observations of the function f. Lastly, let $\mathcal{D}_* = (\mathbf{X}_*, \mathbf{y}_*) = \{(\mathbf{x}_j, y_j)\}_{j=0}^{N_*}$, with $\mathbf{X}_* \in \mathbb{R}^{N_*\times d}$ and $\mathbf{y}_* \in \mathbb{R}^{N_*}$, be the test observations at which we want to compute the predictive distributions of $\mathbf{y}_* = f(\mathbf{X}_*)$ for the function f.

A Gaussian process is defined as a stochastic process, such that every finite collection of realizations $\mathbf{X} = \{\mathbf{x}_i\}_{i=0}^N, \mathbf{x}_i \in \mathbb{R}^d$ of the random variables $\boldsymbol{X} \sim \mathcal{N}(\,\cdot \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}), \boldsymbol{X} \in \mathbb{R}^d$ is a multivariate distribution. A constraint of Gaussian processes as they are used in machine learning, which can be relaxed in specific cases, is that they are assumed to have a zero mean. In order to compute a predictive distribution over $\mathbf{y}_*$ we initially construct the joint distribution over the training observations $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ and test observations $\mathcal{D}_* = (\mathbf{X}_*, \mathbf{y}_*)$:

$$p(\mathbf{y}_*, \mathbf{y}, \mathbf{X}_*, \mathbf{X}) = \frac{1}{\sqrt{(2\pi)^{N+N_*}|\mathbf{K}|^2}} \exp\left[-\frac{1}{2}\begin{bmatrix}\mathbf{y}\\ \mathbf{y}_*\end{bmatrix}^T \begin{bmatrix}K_{\mathbf{XX}} & K_{\mathbf{XX}_*}\\ K_{\mathbf{X}_*\mathbf{X}} & K_{\mathbf{X}_*\mathbf{X}_*}\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{y}\\ \mathbf{y}_*\end{bmatrix}\right] \tag{5}$$

$$\propto \exp\left[-\frac{1}{2}\begin{bmatrix}\mathbf{y}\\ \mathbf{y}_*\end{bmatrix}^T \begin{bmatrix}K_{\mathbf{XX}} & K_{\mathbf{XX}_*}\\ K_{\mathbf{X}_*\mathbf{X}} & K_{\mathbf{X}_*\mathbf{X}_*}\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{y}\\ \mathbf{y}_*\end{bmatrix}\right] \tag{6}$$

$$\propto \mathcal{N}\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{y}_*\end{bmatrix}\middle| \mathbf{0}, \mathbf{K}\right) \tag{7}$$

where the covariance matrix of the joint Gaussian distribution is given by

$$\mathbf{K} = \begin{bmatrix}K_{\mathbf{XX}} & K_{\mathbf{XX}_*}\\ K_{\mathbf{X}_*\mathbf{X}} & K_{\mathbf{X}_*\mathbf{X}_*}\end{bmatrix} = \begin{bmatrix}k(\mathbf{X},\mathbf{X}) & k(\mathbf{X},\mathbf{X}_*)\\ k(\mathbf{X}_*,\mathbf{X}) & k(\mathbf{X}_*,\mathbf{X}_*)\end{bmatrix} \tag{8}$$

and $k(x, x')$ is an kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that measures the similarity between two vectors $x, x' \in \mathcal{X}$. We can observe from (8) that the covariance between any two observations in the distribution is determined by the similarity through the kernel function $k(x, x')$, namely

$$\mathbb{C}[y, y'] = k(x, x') \tag{9}$$

An essential component of a GP is the kernel function with which the covariances is computed. Often the kernels are engineered to incorporate prior knowledge. A commonly used kernel is the squared exponential kernel

$$k(x, x' \,;\, \theta) = \alpha \exp\left[-\frac{||x - x'||^2}{2\sigma^2}\right], \quad \theta = \{\alpha, \sigma\} \tag{10}$$

where θ corresponds to the hyperparameters of the Gaussian process which can be independently optimized with respect to the observations $(\mathbf{X}, \mathbf{y})$.

Gaussian Processes can be readily extended to multiple dimensions by simply adjusting the kernel to incorporate multiple dimensions. The individual variances σ_i of the dimensions $\mathbb{R}^d$ in the exponential kernel can be independently adjusted, or optimized with the maximization of the marginal probability of the data. The expanded kernel for multidimensional input is defined as followed:

$$k(x, x';\, \theta) = \alpha \exp\left[-\frac{1}{2}(x - x')\Sigma^{-1}(x - x')\right], \quad \theta = \{\alpha, \Sigma\} \tag{11}$$

$$\Sigma = \text{diag}(\sigma_0^2, \sigma_1^2, \ldots, \sigma_d^2) \tag{12}$$

The block matrices $k(\mathbf{X}, \mathbf{X}) \in \mathbb{R}^{N \times N}, k(\mathbf{X}, \mathbf{X}_*) \in \mathbb{R}^{N \times N_*}, k(\mathbf{X}_*, \mathbf{X}) \in \mathbb{R}^{N_* \times N}$ and $k(\mathbf{X}_*, \mathbf{X}_*) \in \mathbb{R}^{N_* \times N_*}$ in (8) are the Gramian matrices of the training and test observations with respect to the kernel $k(x, x')$. Furthermore both $k(\mathbf{X}, \mathbf{X})$ and $k(\mathbf{X}_*, \mathbf{X}_*)$ are symmetric matrices and $k(\mathbf{X}, \mathbf{X}_*)$ and $k(\mathbf{X}_*, \mathbf{X})$ are each others mutually transposed.

Given the joint distribution $p(\mathbf{y}_*, \mathbf{y}, \mathbf{X}_*, \mathbf{X})$, the aim for modeling the training and test observations with a GP is to derive the posterior distribution $p(\mathbf{y}_* \,|\mathbf{y}, \mathbf{X}_*, \mathbf{X})$. In order to derive the mean and covariance function of the posterior distribution, the block matrix inversion lemma in equations (13 - 17) [1] is used to compute the inverse of the covariance matrix (8). For ease of reading and brevity the respective block matrices were replaced by more easily readible variables in the following identity:

$$\mathbf{K}^{-1} = \begin{bmatrix} K_{\mathbf{XX}} & K_{\mathbf{XX}_*} \\ K_{\mathbf{X}_*\mathbf{X}} & K_{\mathbf{X}_*\mathbf{X}_*} \end{bmatrix}^{-1} \tag{13}$$

$$= \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \tag{14}$$

$$= \begin{bmatrix} A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix} \tag{15}$$

$$= \begin{bmatrix} A^{-1} + A^{-1}B\Sigma^{-1}CA^{-1} & -A^{-1}B\Sigma^{-1} \\ -\Sigma^{-1}CA^{-1} & \Sigma^{-1} \end{bmatrix} \tag{16}$$

$$= \begin{bmatrix} P & Q \\ R & S \end{bmatrix} \tag{17}$$

$$\Sigma = D - CA^{-1}B = K_{\mathbf{X}_*\mathbf{X}_*} - K_{\mathbf{X}_*\mathbf{X}}{K_{\mathbf{XX}}}^{-1}K_{\mathbf{XX}_*} \tag{18}$$

Instead of computing the inverse of the entire matrix $\mathbf{K}$, which can be computationally expensive for large covariance matrices, the precision matrix $\mathbf{K}^{-1}$ can be computed block-wise with the block matrix inversion lemma. Given the precision matrix in block matrix notation, the inner product in the exponential term of the Gaussian distribution can be computed as a sum over the inner products with the independent block matrices:

$$p(\mathbf{y}_*, \mathbf{y}, \mathbf{X}_*, \mathbf{X}) \propto \exp\left[-\frac{1}{2}\begin{bmatrix}\mathbf{y} \\ \mathbf{y}_*\end{bmatrix}^T \begin{bmatrix} K_{\mathbf{XX}} & K_{\mathbf{XX}_*} \\ K_{\mathbf{X}_*\mathbf{X}} & K_{\mathbf{X}_*\mathbf{X}_*} \end{bmatrix}^{-1} \begin{bmatrix}\mathbf{y} \\ \mathbf{y}_*\end{bmatrix}\right] \tag{19}$$

$$= \exp\left[-\frac{1}{2}\begin{bmatrix}\mathbf{y} \\ \mathbf{y}_*\end{bmatrix}^T \begin{bmatrix} P & Q \\ R & S \end{bmatrix} \begin{bmatrix}\mathbf{y} \\ \mathbf{y}_*\end{bmatrix}\right] \tag{20}$$

$$= \exp\left[-\frac{1}{2}\left(\mathbf{y}^T P\mathbf{y} + \mathbf{y}^T Q\mathbf{y}_* + \mathbf{y}_*^T R\mathbf{y} + \mathbf{y}_*^T S\mathbf{y}_*\right)\right] \tag{21}$$

Since we are only interested in the posterior distribution $p(\mathbf{y}_* \,|\mathbf{y}, \mathbf{X}_*, \mathbf{X})$, terms which do not include $\mathbf{y}_*$ in (21) can be moved into the normalization term. The conditional distribution can thus be

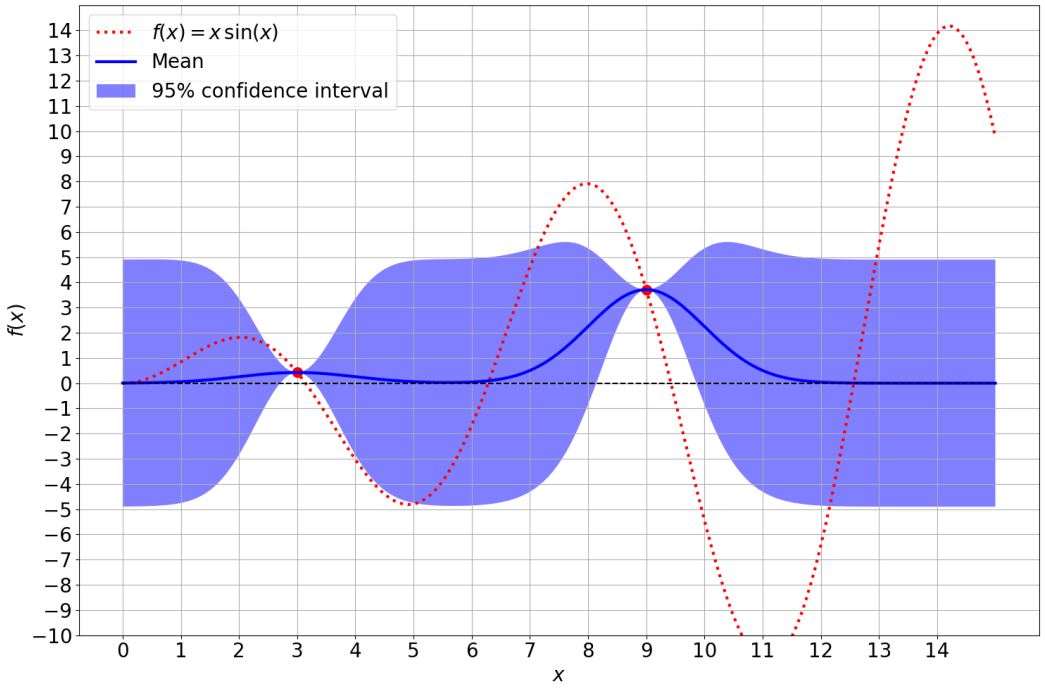


Figure 1: Depiction of a Gaussian Process with two prior obvervations. The red dotted line represents the ground truth, the objective function, while the solid blue line shows the predicted mean of the gaussian process for all datapoints. The blue area around the mean is the 95% confidence interval that is computed using the variance function.

simplified to

$$p(\mathbf{y}_* \mid \mathbf{y}, \mathbf{X}_*, \mathbf{X}) \propto \exp\left[-\frac{1}{2}\left(-\mathbf{y}^T Q \mathbf{y}_* - \mathbf{y}_*^T R \mathbf{y} + \mathbf{y}_*^T S \mathbf{y}_*\right)\right] \tag{22}$$

$$= \exp\left[-\frac{1}{2}\left(-\mathbf{y}^T A^{-1} B \Sigma^{-1} \mathbf{y}_* - \mathbf{y}_*^T \Sigma^{-1} C A^{-1} \mathbf{y} + \mathbf{y}_*^T \Sigma^{-1} \mathbf{y}_*\right)\right] \tag{23}$$

$$\propto \exp\left[-\frac{1}{2}\left(-2\mathbf{y}_*^T \Sigma^{-1} C A^{-1} \mathbf{y} + \mathbf{y}_*^T \Sigma^{-1} \mathbf{y}_*\right)\right] \tag{24}$$

$$\propto \exp\left[-\frac{1}{2}\left(-2\mathbf{y}_*^T \Sigma^{-1} K_{\mathbf{x}_*\mathbf{x}} K_{\mathbf{x}\mathbf{x}}{}^{-1} \mathbf{y} + \mathbf{y}_*^T \Sigma^{-1} \mathbf{y}_*\right)\right] \tag{25}$$

with the matrices Σ being a symmetric matrix by construction, and B and C being each other transposed, namely $C^T = B$, which gives rise to the identity:

$$(\mathbf{y}^T A^{-1} B \Sigma^{-1} \mathbf{y}_*)^T = \mathbf{y}_*^T (\Sigma^{-1})^T B^T (A^{-1})^T \mathbf{y} \tag{26}$$

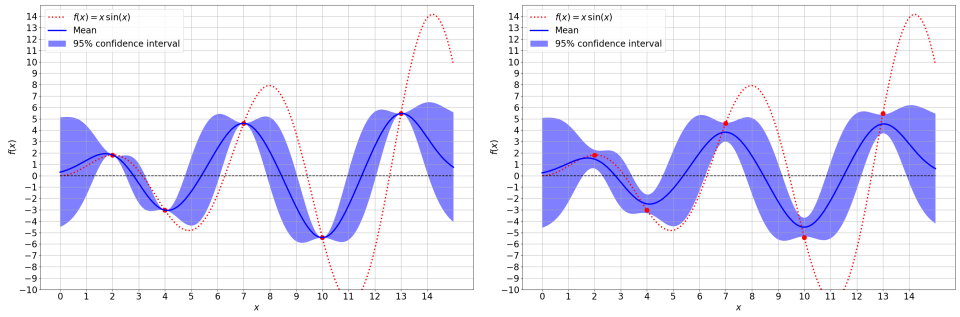
$$= \mathbf{y}_*^T \Sigma^{-1} C A^{-1} \mathbf{y} \tag{27}$$

Alternatively one would argue that the result of both inner products yields the same scalar value due to $B = C^T$. With the derivations above we obtain a posterior distribution $p(\mathbf{y}_* \mid \mathbf{y}, \mathbf{X}_*, \mathbf{X})$ with the mean and covariance function

$$\mu(\mathbf{y}_*) = K_{\mathbf{x}_*\mathbf{x}} K_{\mathbf{x}\mathbf{x}}{}^{-1} \mathbf{y} \tag{28}$$

$$\Sigma(\mathbf{y}_*) = K_{\mathbf{x}_*\mathbf{x}_*} - K_{\mathbf{x}_*\mathbf{x}} K_{\mathbf{x}\mathbf{x}}{}^{-1} K_{\mathbf{x}\mathbf{x}_*} \tag{29}$$

It should be noted that during plotting only the diagonal entries of the covariance matrix are of interest since the diagonal entries of the covariance matrix denote the variances at the evaluated points. Given



(a) Gaussian process without noisy observations.

(b) Gaussian process with noisy observations

(c) Gaussian process with noise only on the mean of the posterior distribution.

(d) Gaussian process with noise only on the variance of the posterior distribution

Figure 2: The influence of noisy observations on the posterior distribution of the Gaussian process. Especially (d) would be interesting in practice as it assumes the mean as undisturbed but includes the noise in the variance of the observation.

the computation of both the mean and variance of the posterior distribution we obtain a Gaussian distribution:

$$p(\mathbf{y}_*|\mathbf{y}, \mathbf{X}_*, \mathbf{X}) = \mathcal{N}\big(\underbrace{K_{\mathbf{X}_*\mathbf{X}}{K_{\mathbf{X}\mathbf{X}}}^{-1}\mathbf{y}}_{\mu}, \underbrace{K_{\mathbf{X}_*\mathbf{X}_*} - K_{\mathbf{X}_*\mathbf{X}}{K_{\mathbf{X}\mathbf{X}}}^{-1}K_{\mathbf{X}\mathbf{X}_*}}_{\Sigma}\big) \tag{30}$$

A visual depiction of a Gaussian process can be seen in Figure 1.

2.1 Noise

In the real world, the observations onto which the Gaussian process is fitted are often influenced and distorted by noise. This noise is modeled as a independent, identically distributed normal distribution around zero with an error variance σ_ε^2:

$$y = f(x) + \varepsilon, \qquad \text{i.i.d. } \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2) \tag{31}$$

The covariance matrix between the respective observations with noise is modified on the diagonal entries. The linear covariance operator can be applied independently to both the objective function evaluation and the noise, yet the noise variance can only be included for the diagonal entries of the covariance matrix. This is due to the assumption of independent, identical distributed noise, which is uncorrelated between observations.

$$\mathbb{C}[y, y'] = k(x, x') + \mathbb{1}_{y=y'}\, \mathbb{V}[\varepsilon] \tag{32}$$

$$= k(x, x') + \mathbb{1}_{y=y'}\, \sigma_\varepsilon^2 \tag{33}$$

This can be realized with the addition of the noise's variance to the diagonal entries of the covariance matrix of the observation kernel matrix $K_{\mathbf{X}\mathbf{X}}$:

$$\mathbf{K} = \begin{bmatrix} K_{\mathbf{X}\mathbf{X}} + \sigma_\varepsilon^2 \cdot I & K_{\mathbf{X}\mathbf{X}_*} \\ K_{\mathbf{X}_*\mathbf{X}} & K_{\mathbf{X}_*\mathbf{X}_*} \end{bmatrix} = \begin{bmatrix} k(\mathbf{X}, \mathbf{X}) + \sigma_\varepsilon \cdot I & k(\mathbf{X}, \mathbf{X}_*) \\ k(\mathbf{X}_*, \mathbf{X}) & k(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix} \tag{34}$$

where I is an identity matrix $I \in \mathbb{R}^{N \times N}$. While the noise itself decreases the precision with which we can fit the GP to the observations, it has convenient numerical properties. The Gramian block matrix $K_{\mathbf{XX}}$ has to be inverted during the computation of the mean and covariance function. Due to possible rank defincencies, $K_{\mathbf{XX}}$ can become singular which prohibits its inversion. Rank definencies in the covariance matrix can arise when two observations are numerically almost identical. Incorporating the noise variance into the covariance matrix can be thus regarded as a regularization of the Gaussian process. This opens the possibility of different regularization themes as both the mean and variance can be independently regularized with respect to the inverse of $K_{\mathbf{XX}}$ in $\mu(\mathbf{y}_*)$ and $\Sigma(\mathbf{y}_*)$. Such a setup is shown in Figure 2.

2.2 Model Selection

The optimization of hyperparameters in machine learning is a pivotal process which can influence the performance significantly. In this regard, Bayesian methods offer a substantial advantage over non-Bayesian methods as the optimal hyperparameters can be automatically recovered from the Bayesian model. For a supervised learning task, the objective is to maximize the likelihood probability of the targets $p(\mathcal{D})$.

A central aspect of Bayesian methods is the placement of a prior $p(\theta)$ over possible values of θ which encodes the prior belief what values of θ are regarded as probable. Instead of considering a single value for θ a probability distribution is used that assigns a different weighting to different values of θ. This is especially important in tasks with small datasets where the likelihood is sensitive to the variability in the data.

The prior can be marginalized to evaluate its influence on the data likelihood. The objective is therefore to find suitable distributions for θ which increase the likelihood of the data, ie.

$$p(\mathcal{D}) = \int p(\mathcal{D}, \theta)\; p(\theta)\; d\theta \tag{35}$$

In the case of Gaussian processes with the squared exponential kernel (10), the hyperparameters are $\theta = \{\alpha, \sigma\}$ for which we seek values that maximize the probability of the data, i.e.

$$\begin{aligned}
\max_{\theta}\; p(\mathcal{D}; \theta) &= \max_{\theta}\; p(\mathbf{y}, \mathbf{X}; \theta) && (36)\\
&= \max_{\theta}\; p(\mathbf{y}, \mathbf{X} \mid \theta)\; p(\theta) && (37)\\
&= \max_{\theta} \frac{1}{\sqrt{(2\pi)^N |K_{\mathbf{XX}}|^2}} \exp\left[-\frac{1}{2}\mathbf{y}^T {K_{\mathbf{XX}}}^{-1}\mathbf{y}\right] && (38)\\
&= \max_{\theta} \frac{1}{\sqrt{(2\pi)^N |k(\mathbf{X}, \mathbf{X}; \theta)|^2}} \exp\left[-\frac{1}{2}\mathbf{y}^T k(\mathbf{X}, \mathbf{X}; \theta)^{-1}\mathbf{y}\right] && (39)
\end{aligned}$$

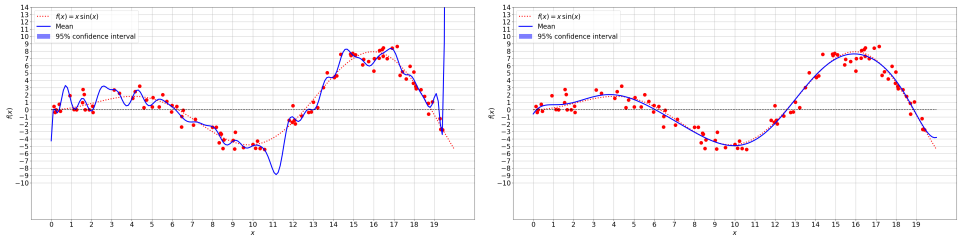
where the parameters θ determine the Gramian matrix $k(\mathbf{XX}; \theta)$. The maximization of the data likelihood in (39) is commonly reformulated as a minimzation of the negative log-likelihood. Working with the log-probability offers a higher numerical stability with respect to floating-point arithmetic of modern computers.

$$\begin{aligned}
\min_{\theta} -\log p(\mathcal{D}; \theta) &= \min_{\theta}\; \frac{N}{2}\log[2\pi] + \log[|K_{\mathbf{XX}}|] + \frac{1}{2}\mathbf{y}^T K_{\mathbf{XX}}^{-1}\mathbf{y} && (40)\\
&= \min_{\theta}\; \frac{N}{2}\log[2\pi] + \log[|k(\mathbf{XX}; \theta)|] + \frac{1}{2}\mathbf{y}^T k(\mathbf{X}, \mathbf{X}; \theta)^{-1}\mathbf{y} && (41)
\end{aligned}$$

The optimization of the log-likelihood can be done with regular optimization algorithms such as limited memory BFGS [2].

2.3 Derivative Information

Gaussian processes in their traditional definition are described as a Gaussian distribution over possibly infinite observations. A Gaussian process computes a predictive distribution for $\mathbf{y}_*$ such that predictions are close to observations in their vicinity. We can expand the Gaussian process by including derivative observations into the set of observations which enforces a similarity in the



(a) Gaussian process before hyperparameter optimization.

(b) Optimized hyperparameters using log-likelihood minimization. The kernel parameters have been fitted to the marginal data distribution.

Figure 3: Example of the model selection for a Gaussian process. for which the hyperparameters θ have been optimized.

gradients of the predictions with respect to observations in their vicinity:

$$\begin{bmatrix} \mathbf{y} \\ \nabla\mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathcal{N}\left(\cdot \,|\mathbf{0}, \mathbf{K}^{\nabla}\right) \tag{42}$$

The joint distribution over predictions, derivative observations and observations can be modeled as a Gaussian over all three types of observations:

$$p(\mathbf{y}_*, \nabla\mathbf{y}, \mathbf{y}, \mathbf{X}_*, \mathbf{X}) \propto \exp\left[-\frac{1}{2}\begin{bmatrix} \mathbf{y} \\ \nabla\mathbf{y} \\ \mathbf{y}_* \end{bmatrix}^T \left[\begin{array}{cc|c} K_{\mathbf{XX}} & K^{\nabla}_{\mathbf{XX}} & K_{\mathbf{XX}_*} \\ K^{\nabla T}_{\mathbf{XX}} & K^{\nabla\nabla}_{\mathbf{XX}} & K^{\nabla}_{\mathbf{XX}_*} \\ \hline K_{\mathbf{X}_*\mathbf{X}} & K^{\nabla}_{\mathbf{X}_*\mathbf{X}} & K_{\mathbf{X}_*\mathbf{X}_*} \end{array}\right]^{-1} \begin{bmatrix} \mathbf{y} \\ \nabla\mathbf{y} \\ \mathbf{y}_* \end{bmatrix}\right] \tag{43}$$

with the expanded covariance matrix which now includes similarity measures between predictions, observations and derivative observations:

$$\mathbf{K}^{\nabla} = \left[\begin{array}{c|c} K^{\nabla,\nabla\nabla}_{\mathbf{XX}} & K^{\nabla}_{\mathbf{XX}_*} \\ \hline K^{\nabla}_{\mathbf{X}_*\mathbf{X}} & K_{\mathbf{X}_*\mathbf{X}_*} \end{array}\right] \tag{44}$$

$$= \left[\begin{array}{cc|c} K_{\mathbf{XX}} & K^{\nabla}_{\mathbf{XX}} & K_{\mathbf{XX}_*} \\ K^{\nabla T}_{\mathbf{XX}} & K^{\nabla\nabla}_{\mathbf{XX}} & K^{\nabla}_{\mathbf{XX}_*} \\ \hline K_{\mathbf{X}_*\mathbf{X}} & K^{\nabla}_{\mathbf{X}_*\mathbf{X}} & K_{\mathbf{X}_*\mathbf{X}_*} \end{array}\right] \tag{45}$$

$$= \left[\begin{array}{cc|c} k_{\mathbf{y},\mathbf{y}}(\mathbf{X},\mathbf{X}) & k_{\mathbf{y},\nabla\mathbf{y}}(\mathbf{X},\mathbf{X}) & k_{\mathbf{y},\mathbf{y}_*}(\mathbf{X},\mathbf{X}_*) \\ k_{\nabla\mathbf{y},\mathbf{y}}(\mathbf{X},\mathbf{X}) & k_{\nabla\mathbf{y},\nabla\mathbf{y}}(\mathbf{X},\mathbf{X}) & k_{\nabla\mathbf{y},\mathbf{y}_*}(\mathbf{X},\mathbf{X}_*) \\ \hline k_{\mathbf{y}_*,\mathbf{y}}(\mathbf{X}_*,\mathbf{X}) & k_{\mathbf{y}_*,\nabla\mathbf{y}}(\mathbf{X}_*,\mathbf{X}) & k_{\mathbf{y}_*,\mathbf{y}_*}(\mathbf{X}_*,\mathbf{X}_*) \end{array}\right] \tag{46}$$

The posterior distribution including derivative observations can be derived from the joint distribution with the matrix inversion lemma in the same manner as seen above. The mean and covariance of the posterior distribution with derivative observations can be computed with the expanded kernel matrices:

$$p(\mathbf{y}_*|\nabla\mathbf{y}, \mathbf{y}, \mathbf{X}_*, \mathbf{X}) = \mathcal{N}\big(K^{\nabla}_{\mathbf{X}_*\mathbf{X}} K^{\nabla,\nabla\nabla\,-1}_{\mathbf{XX}}\mathbf{y}, K_{\mathbf{X}_*\mathbf{X}_*} - K^{\nabla}_{\mathbf{X}_*\mathbf{X}} K^{\nabla,\nabla\nabla\,-1}_{\mathbf{XX}} K^{\nabla}_{\mathbf{XX}_*}\big)$$

The Gramian block matrices in (46) between predictions, observations and derivative observations can be computed with updated kernels with incorporate the derivative observations [3]. More precisely, the covariance between two any entries in the observation respectively prediction vector are defined as

$$\mathbb{C}[y, y'] = k_{y,y'}(x, x') \tag{47}$$

$$\mathbb{C}[y, \nabla y'] = k_{y,\nabla y'}(x, x') \tag{48}$$

$$\mathbb{C}[\nabla y, y'] = k_{\nabla y,y'}(x, x') \tag{49}$$

$$\mathbb{C}[\nabla y, \nabla y'] = k_{\nabla y,\nabla y'}(x, x') \tag{50}$$

These updated kernels can be derived in a fairly straightforward manner since the covariance with the zero mean assumption is a linear operator [4]. In order to expand the Gaussian process with derivative observations we have to take the derivative of the kernel and expand the covariance matrix with the respective entries:

$$\mathbb{C}[y, y'] = \frac{1}{N}\sum_{i=0}^{N} y_i \cdot y_i' \tag{51}$$

$$= k(x, x') \tag{52}$$

$$\mathbb{C}[y, \nabla_{x'} y'] = \frac{1}{N}\sum_{i=0}^{N} y_i \cdot \nabla_{x'} y_i' \tag{53}$$

$$= \nabla_{x'} \frac{1}{N}\sum_{i=0}^{N} y_i \cdot y_i' \tag{54}$$

$$= \nabla_{x'} \mathbb{C}[y, y'] \tag{55}$$

$$= \nabla_{x'} k(x, x') \tag{56}$$

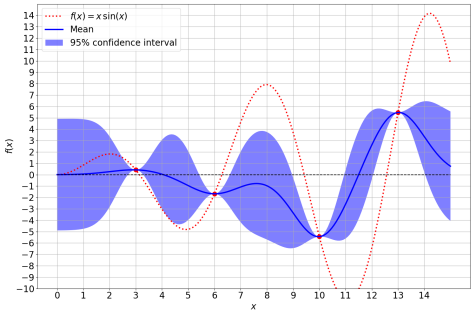
$$\mathbb{C}[\nabla_x y, \nabla_{x'} y'] = \frac{1}{N}\sum_{i=0}^{N} \nabla_x y_i \cdot \nabla_{x'} y_i' \tag{57}$$

$$= \nabla_x \nabla_{x'} \frac{1}{N}\sum_{i=0}^{N} y_i \cdot y_i' \tag{58}$$

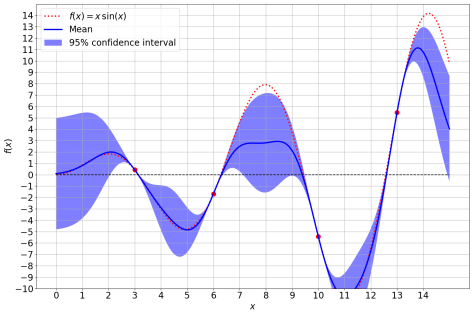
$$= \nabla_x \nabla_{x'} \mathbb{C}[y, y'] \tag{59}$$

$$= \nabla_x \nabla_{x'} k(x, x') \tag{60}$$

While derivative observations themselves are usually hard to come by for computationally expensive functions $f(x)$, derivative observations are of numerical advantage in cases where observations lie very close to each other. In these cases the inversion can become unstable or even impossible due to the rank definciency. Derivative observations pose a useful way to circumvent such rank definciencies for very close observations by combining two observations into one observation and a derivative observation [5]. An example of a Gaussian process with derivative observations is given in Figure 4.



(a) GP without derivative observations. Since the GP does not take the derivative of the function into concern and assumes a zero mean, the slope of the approximated function does not correspond to the slope of the objective function at the observations.



(b) The same GP with derivative observations. The true function $f(x)$ can now be modeled with a limited number of observations and derivative observations.

Figure 4: Example for a Gaussian process with additional derivative observations.

3 Bayesian Optimization

As stated above, many problem settings in engineering and science can be formulated as optimization problems of a criterion, commonly called an objective function, $\mathcal{F}(x)$ with respect to some argument x. The goal of any optimization is to find the global optimum of such a function $\mathcal{F}(x)$. For linear or convex optimization problems, this is usually feasible, yet optimization becomes difficult for non-linear objective functions. Bayesian optimization tries to tackle such non-linear objective functions by searching for a global optimum in a probabilistical manner.

3.1 Optimization

In computer science, mathematics and operations research, mathematical optimization aims to find the best value $x^* \in \mathcal{X}$ from a set of feasible values $\mathcal{X}$ with respect to an criterion or objective function $\mathcal{F}(x)$. Optimization problems can be formulated as either maximization or minimization problems of the objective function:

$$\mathcal{F}(x^*) = \min_{x \in \mathcal{X}} \mathcal{F}(x) = \max_{x \in \mathcal{X}} -\mathcal{F}(x) \tag{61}$$

where

$$x^* = \operatorname*{argmin}_{x \in \mathcal{X}} \mathcal{F}(x) = \operatorname*{argmax}_{x \in \mathcal{X}} -\mathcal{F}(x) \tag{62}$$

Since $\mathcal{F}(x)$ is often a complicated, non-linear function the solution is searched for in an iterative manner. Most optimization algorithms evaluate the objective function $\mathcal{F}(x)$ through a set of succesive queries $x_{1:n} = \{x_i\}_{i=1}^n \subset \mathcal{X}$ such that the information of the previous evaluations guide the next evaluation x_{n+1} through a utility function

$$x_{n+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \mathcal{U}(x \mid x_1, \ldots, x_n) \tag{63}$$

The information contained in the past evaluations $x_{1:n}$ is thus leveraged in a way to make the evaluation x_{n+1} as close as possible to the global optimum. The utility function $\mathcal{U}$ should balance the exploration of the set of feasible optima $\mathcal{X}$ while simultaneously exploiting existing information in $x_{1:n}$ to find the globally optimal solution x^*.

3.2 Bayesian Optimization with Gaussian Processes

In Bayesian optimization a Gaussian process is used to compute a probability distribution over the past evaluations $x_{1:n}$, which guides a subsequent sampling process. The sampling process uses an acquisition function $\Lambda(x \mid x_{1:n})$, which is a utility function on the posterior distribution computed by the Gaussian process. The acquisition function balances both the exploration as well as the exploitation of the unknown objective function $\mathcal{F}(x)$. The next evaluation is chosen such that it maximizes the acquisition function, i.e.

$$x_{n+1} = \operatorname*{argmax}_{x \in \mathcal{X}} \Lambda(x \mid x_{1:n}) \tag{64}$$

By computing posterior distributions over all predictions at once, Gaussian processes have a powerful property which enables them to search for an optimum globally. The posterior distributions allow Gaussian processes to balance both exploitation and exploration of the set of feasible solutions by incorporating their uncertainty into optimization task.

The acqusition function $\Lambda(x \mid x_{1:n})$ serves as an improvement criterion for the yet unevaluated feasible solutions. The improvement is computed relative to the optimal solution $x^+ \in x_{1:n}$ in the set of previous evaluations $x_{1:n}$,

$$x^+ = \operatorname*{argmax}_{x \in x_{1:n}} \mathcal{F}(x) \tag{65}$$

A popular acquisition functions is the upper/lower confidence bound [6], which scales the mean with respect to the previously best evaluation. It then considers a multiple of the standard deviation and adds it for maximization problems or subtracts it for minimization problems. The hyperparameter κ is usually selected as a small integer number, which can be intuitively selected due to its close

relationship to confidence values of the Gaussian distribution. Given the mean $\mu(x)$ and covariance function $\sigma(x)$, the upper confidence bound is computed with the hyperparameter κ via

$$\mathbb{UCB}[x] = \mu(x) + \kappa\sigma(x) - \mathcal{F}(x^+) \tag{66}$$

A different acquisition function is the expected improvement (EI) [7] which considers the expected value at a point x_{n+1} above the currently best value x^+. The expected improvement is the most Bayesian acquisition function as it incorporates the posterior in its entirety including the uncertainty.

$$\mathbb{EI}[x] = \int_{x^+}^{\infty} \left(\frac{f(x) - f(x^+)}{\sigma(x)} \right) \mathcal{N}(x|\mu(x), \sigma(x)) df(x) \tag{67}$$

$$= \int_{x^+}^{\infty} z(x)\, \mathcal{N}(x|\mu(x), \sigma(x)) df(x) \tag{68}$$

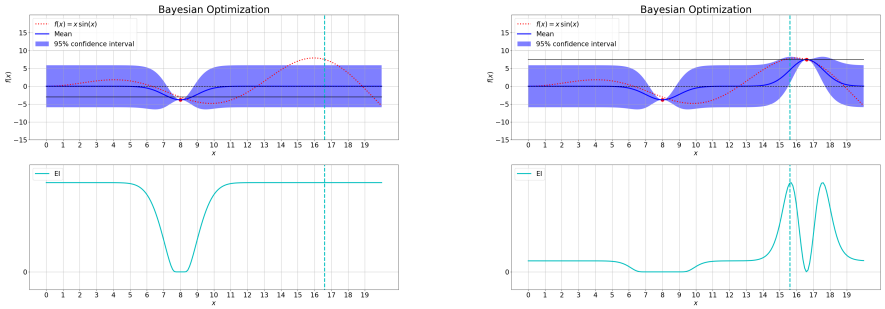
$$= \sigma(x)\left(z(x)\Phi\left(z(x)\right) + \mathcal{N}_{0,1}\left(z(x)\right)\right) \tag{69}$$

$$\tag{70}$$

where the term $z(x)$ represents the z-score for a specific value x in the yet unevaluated feasible set solutions:

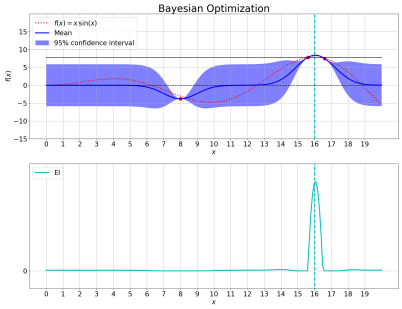
$$z(x) = \frac{f(x) - f(x^+)}{\sigma(x)} \tag{71}$$

While the UCB acquisition function has a more straightforward interpretation, it suffers from getting stuck in local minima. This is due to UCB using a fixed integer multiple κ of the variance instead of integrating over it. The EI acquisition utilizes the uncertainty in a fully Bayesian way and is able to explore the feasible set even after having found an optimum. This becomes evident in Figure 5 where it continues exploring the feasible set even after finding the global maximum at $x = 16$. It should be noted that the posterior variance in Figure 5 is plotted only for the 95% confidence intervall and the posterior distributions continue infinitely along the y-axis. For that reason EI was chosen as the standard acquisition function for our experiments in Section 4.

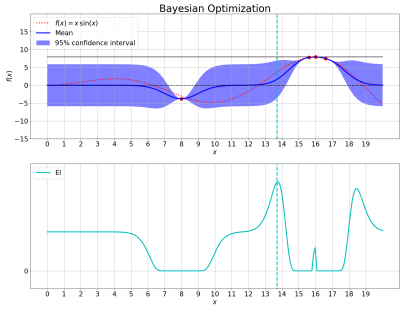


(a)

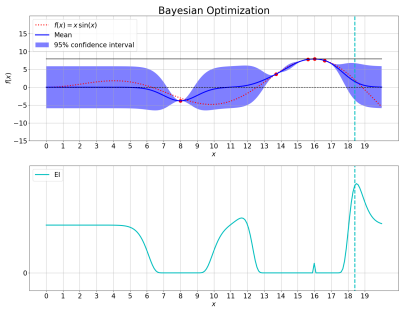
(b)



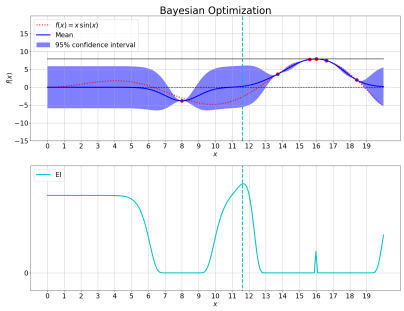
(c)



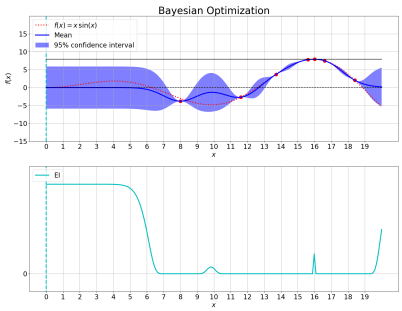
(d)



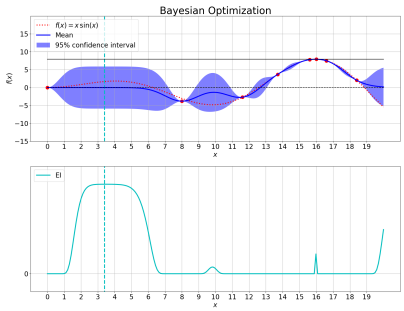
(e)



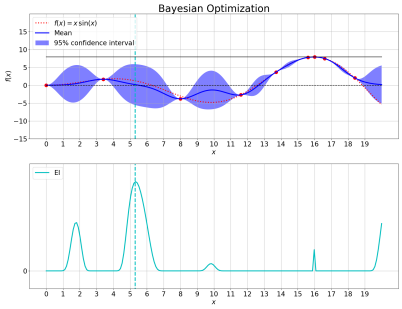
(f)



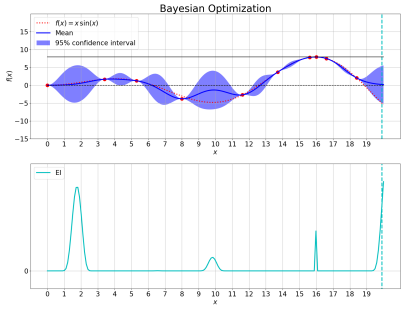
(g)



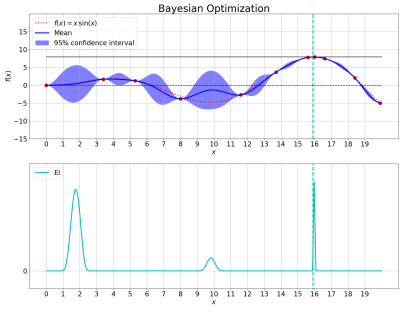
(h)



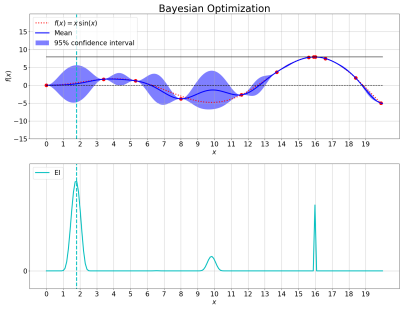
(i)



(j)



(k)



(l)

Figure 5: Bayesian optimization with the expected improvement acquisition function.

4 Experiments

In this section, we present the experiments we conducted to apply Gaussian Processes to real life problems. These entail Kriging – geospatial optimization –, an application of quantum chemistry, as well as the optimization of a Support Vector Machine.

4.1 Kriging

One the earliest, if not the earliest, application of Gaussian Processes to geostatistics is called "Kriging", after its inventor Danie Krige. In 1951, during his masters thesis, he tried to find a method to estimate the most likely distribution of gold at the Witwatersrand reef mine in South Africa, which holds the worlds most known gold reserves and to this day has produced about 50 % of gold ever mined. This estimation was to be done given samples from a relatively small amount of boreholes. Since the recovery of these samples is very costly, using heavy machinery, so typically, as few samples as possible are taken. Since this method interpolates the unknown values to find an estimation, it can also be categorized as a regression method and is also called Gaussian process regression.
To this end, we used a publicly available dataset[1] that contains the real world locations of about 21,500 gold samples obtained in a gold mine from the same type as the Witwatersrand reef mine that has been the inspiration for Kriging in the first place. While the gold values are, for obvious reasons, fictitious, the relative location of the boreholes are from a real gold mine.

4.1.1 Method

The dataset we used, was, as mentioned above, publicly available and is made up off three variables: a x- and a y-coordinate as well as the corresponding gold grade. To get a smoother distribution, the log values of the gold grade are taken and subsequently centered.
Because the gold grade is dependent on two variables, the x and the y coordinate, we extended the kernel function to two dimensions. Similarily to 3, the kernel is given as

$$k(x, x') = \alpha \exp\left[-\gamma ||x - x'||^2\right] \tag{72}$$

where α is the output variance and γ us the length scale of the kernel. The larger alpha, the further the GP interpolates away from its mean, the bigger the length scale, the smoother the approximation is. As is going to be visible in the working example, these two parameters are important for the success of the GP.
Additionally, since this method interpolates real-world-data, noise was added to the kernel. As stated in Section 2.1, not only is this done to model inconsistencies or errors in sensor data, it also numerically stabilizes the kernel matrix in respect to the inversion of $K_{\mathbf{XX}}$.

Since we could only obtain the gold grade values of the boreholes in the available data and no other ground truth to verify the functionality of the GP on the dataset, we modified the functionality in such a way that the estimation is not done on all arbitrary points in the space spanned by the coordinates, but only on the actually available boreholes. The process is started on one arbitrary datapoint, predictions are only done on coordinates no yet evaluated. Depending on the length scale, surrounding datapoints also have predicted values associated with them.

4.2 Quantum Chemistry

For another application of Gaussian process optimization we chose the subject of quantum chemistry. Quantum chemistry in general is the study of chemical properties on a very small scale of individual molecules and atoms. Molecules have a wide range of interesting chemical properties. One that is of particular interest is the atomization energy. The atomization energy of a molecule denotes the energy necessary to completely split up the molecule into individual atoms, i.e. to solve all existing atom bonds.

The atomization energy for a given molecule can be computed by solving the so called Schrödinger equation for that molecule. However, this process is computationally extremely expensive. For this reason, recently there has been a focus on the task of approximating this solution via machine

[1] http://www.kriging.com/datasets/samples.dat

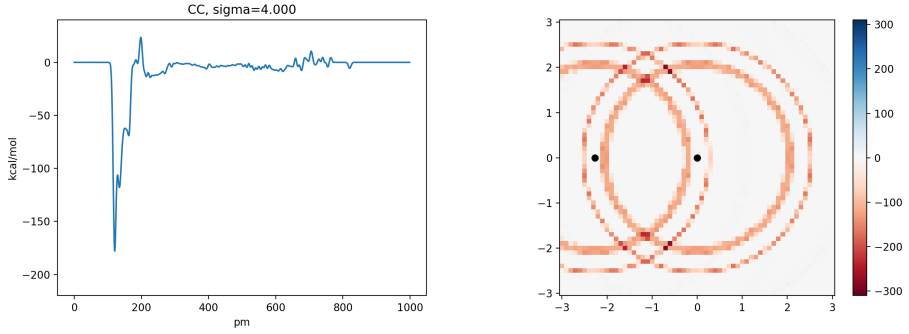
learning. These proposed algorithms take the types and 3d positions of the individual atoms as input and then output the estimated atomization energy in kcal/mol.

The first such approach was proposed in a paper by Rupp et al. (2012) and was able to estimate atomization energies on the QM7 dataset consisting of ca. 7.000 molecules with a mean absolute error of 10 kcal/mol. More recent approaches have used deep neural networks to further increase this accuracy.

The approach we are using is a simpler method using kernel ridge regression applied to the same QM7 dataset.

missing: short explanation of the kernel (introducing sample atoms) with formula

The application of this method depends on several parameters that have to be chosen. First, the regularization parameter λ. Second, the kernel width σ. Third, the number of sample atoms N. However, there is a relatively wide range of possible values for each of these parameters. As every execution of the algorithm still takes quite some time, it is infeasible to simply search for the best combination using grid search. Thus, this is an ideal candidate for the application of Gaussian process optimization.



(a) (b)

Figure 6: todo: description

4.2.1 Method

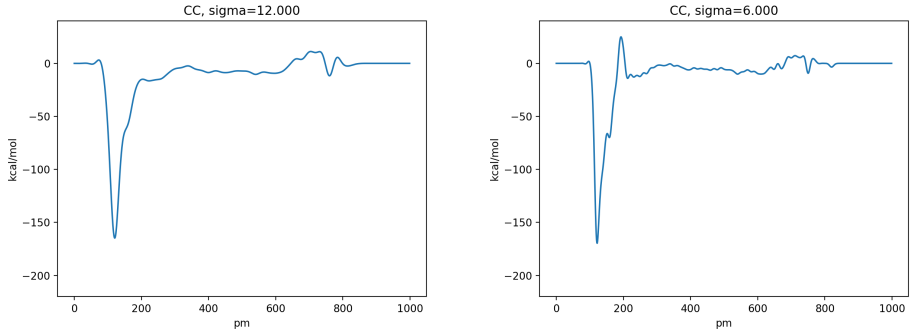
Our objective is to optimize the performance of the quantum chemistry algorithm as measured by the mean absolute error of the atomization energies estimation. This performance depends on the three parameters described above. However, as we know from experience that the algorithm is rather insensitive to changes in the regularization parameter, we decided to keep it fixed and only vary the remaining two parameters. Therefore, we end up with a 2-dimensional optimization over σ and N.

In order to create a more uniform input space, we normalized both σ and N according to the predetermined range within which we expected the optimal parameters. Similarly, we used prior experience with the performance of the kernel ridge regression to transform the mean absolute approximation error logarithmically, with 0 corresponding to average approximation performance and 2 being the best possible approximation (i.e. an error of 0 kcal/mol).

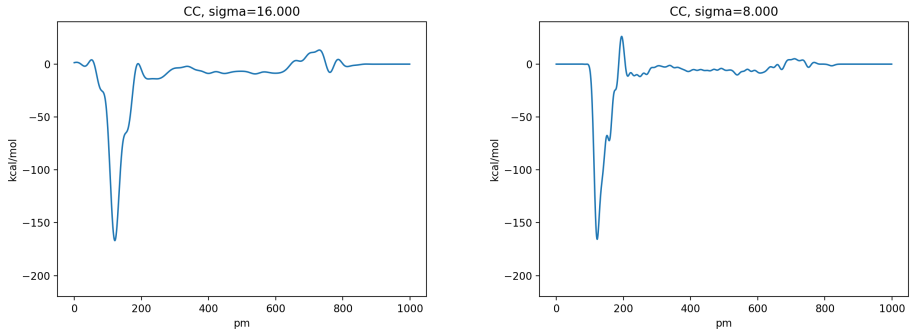
We used a Gaussian kernel for the Gaussian process. As the small number of evaluations was not enough to determine the kernel parameters via model selection, we chose appropriate values again relying on our prior knowledge of the problem.

4.2.2 Results

The optimization ran for 30 steps. The maximum found corresponded to a mean absolute approximation error of 3.5 kcal/mol, which is consistent with the best performance expected using the kernel ridge regression approach.



(a) (b)



(c) (d)

Figure 7: todo: description

Interesting to note is that this was the only experiment we ran that indeed had the function evaluations as the bottleneck. Thus, it was the most realistic application and was indeed able to reliably find the region of optimal parameters in a relatively short time.

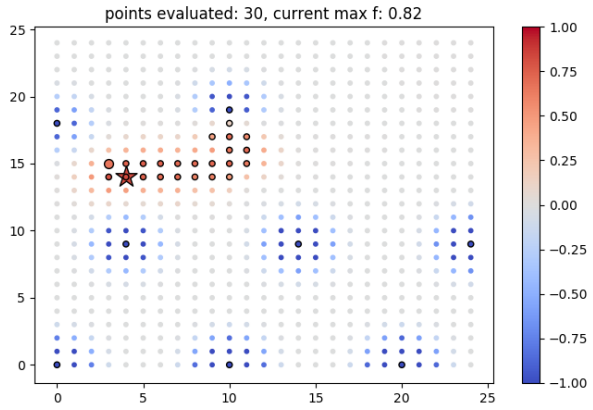


Figure 8: todo: better image, description

4.3 Text Categorization Optimization

The comprehension of language remains a demanding task for machine learning algorithms due to its intricacies and subtleties. While automatic speech recognition has made significant progress and is now almost on par with humans, language modelling remains a challenging topic. One application of language modelling by machine learning algorithms is text categorization. To that end, the 'SMS Guru' dataset was used and machine learning was used to categorize the questions in the dataset to specific knowledge areas.

Support vector machines (SVM) [8] a group of versatile and mathematically well defined machine learning algorithms using kernels that have been used in a wide variety of tasks. Due to the versatility of the Gaussian kernel (10), it is commonly used in SVMs as well. A key component of successfully training SVMs is the right choice of the kernel hyperparameters which have to be fine-tuned to the training data.

A commonly used approach to determine the right hyperparameters is a laborous grid-search. Especially in cases where the training of a single instance during the grid-search is computationally expensive, or the number of possible combinations become prohibitively large, grid-searches become infeasible. To alleviate this issue, Bayesian optimization with a fixed number of evaluations was used to determine the best hyperparameters for the text categorization with a SVM, the result of which can be seen in Figure 9.

The text data was cleaned by removing numerals and punctuation and embedded into a numerical representation with 'term frequency, inverse document frequency' (TFIDF) [9]. TFIDF balances the occurence of terms in the entire data set with the occurence of terms in single documents, thus increasing the significance of terms which indicate a specific document. Intuitively, a term which occurs in every document, ie.e. 'be', 'have', 'to', is not very indicative for a specific topic whereas a term like 'hemoglobin' and 'DNA' distinctively indicate a document on a medical topic. Finally 1.000 best features were selected with the χ^2-Test.

The results can be seen in Figure 9, where the Bayesian optimization manages to find the best parameters $\theta_* = \{C = 78, \sigma = 6.3\}$. These values were verified with a grid-search over 100 values for σ and 100 values for C, resulting in a grid-search over 10.000 values. The Bayesian optimization was able to consistently find the final optima very close to the true hyperparameters θ_*. The results were evaluated with 100 searches with a optima mean of $\mu_C = 77.8, \mu_\sigma = 6.32$ and a optima variance of $\sigma_C = 1.12, \sigma_\sigma = 0.32$. The close mean values and low variances indicate that even with just 20 steps of the Bayesian optimization, we are consistently able to find optima very close to the true optimum.

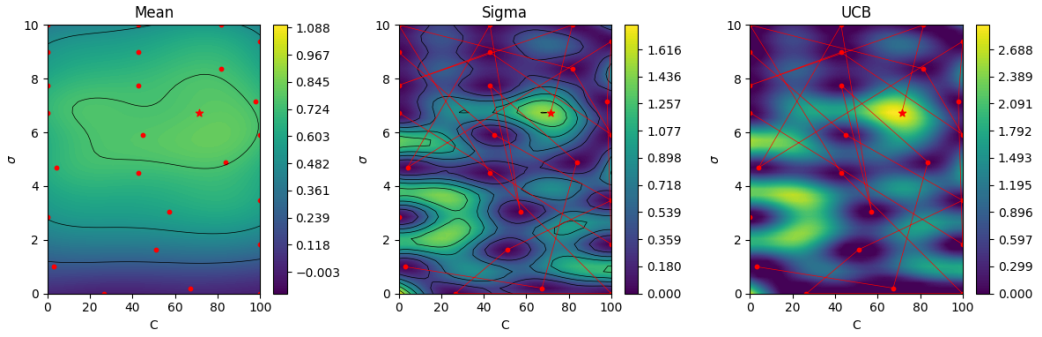


Figure 9: Mean and variance prediction of the hyperparameter search for a SVM trained on text categorization. THe UCB can be observed on the right.

References

[1] D. Tylavsky and G. Sohie, "Generalization of the matrix inversion lemma," *Proceedings of the IEEE*, vol. 74, pp. 1050–1052, July 1986.

[2] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[3] A. Wu, M. C. Aoi, and J. W. Pillow, "Exploiting gradients and hessians in bayesian optimization and bayesian quadrature," p. 20.

[4] E. Solak, R. Murray-smith, W. E. Leithead, D. J. Leith, and C. E. Rasmussen, "Derivative observations in gaussian process models of dynamic systems," p. 8.

[5] M. A. Osborne, R. Garnett, and S. J. Roberts, "Gaussian processes for global optimization," p. 15.

[6] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*, 2009.

[7] J. Močkus, "On bayesian methods for seeking the extremum," in *Optimization Techniques IFIP Technical Conference*, pp. 400–404, Springer, 1975.

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[9] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*, vol. 39. Cambridge University Press, 2008.