A Distance-based Framework for Gaussian Processes over Probability Distributions

Maxim Dolgov
Robert Bosch GmbH
Corporate Research
maxim.dolgov@bosch.com

Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory (ISAS)
Institute for Anthropomatics and Robotics
Karlsruhe Institute of Technology (KIT), Germany
uwe.hanebeck@ieee.org

***Abstract*—Gaussian processes constitute a very powerful and well-understood method for non-parametric regression and classification. In the classical framework, the training data consists of deterministic vector-valued inputs and the corresponding (noisy) measurements whose joint distribution is assumed to be Gaussian. In many practical applications, however, the inputs are either noisy, i.e., each input is a vector-valued sample from an unknown probability distribution, or the probability distributions are the inputs. In this paper, we address Gaussian process regression with inputs given in form of probability distributions and propose a framework that is based on distances between such inputs. To this end, we review different admissible distance measures and provide a numerical example that demonstrates our framework.**

I. Introduction

The problem of learning a regression or classification function given a training dataset can be addressed by either a *parametric* or a *nonparametric* approach. In the parametric approach, the designer selects a function model, e.g., a linear function or a neural network, and optimizes a single set of parameters of the model such that the model fits the training data. In many cases, choosing the model before the data is available leads to poor performance. A natural approach in this case would be to increase the number of function parameters. However, this step bears the risk of overfitting, where the performance on the training data becomes very good but the results for the test data are very poor. In nonparametric learning, the designer still has to choose a model family. But instead of having a single fixed set of parameters, the number of parameters either grows with the data or there are infinitely many parameter sets with an assigned probability distribution [1]. The methods from the latter class are referred to as Bayesian.

Gaussian Processes (GPs) constitute a special class of Bayesian nonparametric methods where the probability distribution of the model parameters, and the probability distribution of the measurements given the inputs and the parameters are Gaussian [2]. Moreover, the mean and the covariance of the Gaussian of the outputs are functions of the inputs. However, the classical GP formulation does not provide a consistent way to address problems where the inputs are noisy, i.e., vector-valued samples drawn from unknown underlying probability distributions, or the inputs are the probability distributions themselves. Consider for example the following scenarios. We wish to learn a model of a temperature distribution on a plane based on noisy temperature measurements collected using a mobile robot. In the first scenario, the robot can measure its position, however, these position measurements are corrupted by noise. Thus, the inputs to the GP that is used to learn the temperature distribution are noisy vectors. Now, assume that the robot uses the position measurements or measurements from, e.g., an inertial measurement unit or landmark measurements in order to estimate its position. The position estimates given in form of probability distributions then serve as the inputs to the GP[1]. This scenario is depicted in Fig. 1.

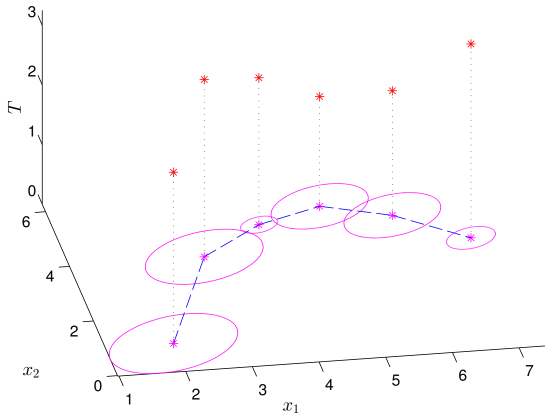


Fig. 1: Illustration of noisy temperature measurements (red stars) collected with a mobile robot. The true robot path is represented by the blue dashed line, while the magenta stars and ellipses depict the mean and the covariance of the estimated pdf of the robot position in the plane.

The distinction between the two described input classes (noisy inputs and probability distributions) is very important. In the case of noisy inputs, the input to the GP is still a vector although its value does not correspond to the value of the true input that generated the measurement. In the second case, where the input to the GP is a probability distribution, the input is a function, i.e., an infinite-dimensional quantity, and therefore not admissible to classical GP framework. Of course, one could argue that in case of probability distributions provided as inputs it is possible to use, e.g., the means as the inputs GP. However, by doing so, we lose information about the uncertainty. Furthermore, the mean may not be the appropriate input representation, e.g., if the probability distribution is multimodal.

[1] An alternative formulation of the problem could be to use the entire sequences of measurements from which the position estimates are inferred as the inputs to the GP. In this case, however, the GP needs to be able to deal with such input data, which is even more challenging than having probability distributions as inputs because the dimension of the inputs is not constant.

In this paper, we address GPs whose inputs are probability distributions and propose a framework that is based on distances between probability distributions. The remainder of the paper is organized as follows. First, we give a brief introduction to the classical GP framework with deterministic inputs, discuss approaches to GPs with probability distributions as inputs, and summarize the contribution of the paper. In Sec. II, we present a new framework for GPs that are defined over distances between probability distributions and review different admissible distance measures in Sec. III. A numerical example demonstrates the proposed framework in Sec. IV and Sec. V concludes the paper.

A. Gaussian Processes with Deterministic Inputs

In what follows, we give a brief introduction to GPs with deterministic inputs. For a much more thorough introduction, the interested reader is referred to [2].

Consider the measurement model

$$y = a(\underline{x}) + \boldsymbol{\nu} \ , \tag{1}$$

where $y \in \mathbb{R}$ denotes the measurement, $\underline{x} \in \mathbb{R}^n$ the input vector, and $\boldsymbol{\nu}$ is an independent and identically distributed Gaussian noise with zero mean and variance σ_ν^2. The measurement function $a(\cdot)$ is not known. However, we assume that given a finite set of inputs $\underline{x}_i$, $i = 1, 2, \dots, N$, $N \in \mathbb{N}$, the corresponding measurements y_i and every subset thereof are jointly normally distributed. Then, the measurements y_i are said to be generated by a GP with mean function $m(\underline{x}) = \mathrm{E}\left\{a(\underline{x})\right\}$ and covariance function $k(\underline{x}_i, \underline{x}_j) = \mathrm{E}\left\{(a(\underline{x}_i) - m(\underline{x}))(a(\underline{x}_j) - m(\underline{x}))^\top\right\}$ that depends on the parameters $\underline{w} \in \mathbb{R}^m$. We then write

$$[y_1 \quad \dots \quad y_N]^\top \sim \mathcal{GP}_{\underline{w}}(\underline{\mu}, \mathbf{\Sigma}) \ .$$

with

$$\underline{\mu} = \begin{bmatrix} m(\underline{x}_1) \\ m(\underline{x}_2) \\ \vdots \\ m(\underline{x}_N) \end{bmatrix} , \ \mathbf{\Sigma} = \begin{bmatrix} k(\underline{x}_1, \underline{x}_1) & k(\underline{x}_1, \underline{x}_2) & \dots & k(\underline{x}_1, \underline{x}_N) \\ k(\underline{x}_2, \underline{x}_1) & k(\underline{x}_2, \underline{x}_2) & \dots & k(\underline{x}_2, \underline{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\underline{x}_N, \underline{x}_1) & k(\underline{x}_N, \underline{x}_2) & \dots & k(\underline{x}_N, \underline{x}_N) \end{bmatrix} ,$$

where $\mathbf{\Sigma}$ is referred to as the kernel of the GP.

Given a training set $(\mathbf{X}, \underline{y})$, where $\mathbf{X} = \{\underline{x}_1, \dots, \underline{x}_N\}$ and $\underline{y} = \{y_{1,\dots,y_N}\}$, the goal is to predict the output y_* at a test input $\underline{x}_*$. For the probability density function (pdf) of y_*, the so-called predictive density of y_*, it holds

$$p(y_*|\underline{x}_*, \mathbf{X}, \underline{y}) = \int\limits_{\mathbb{R}^m} p(y_*|\underline{x}_*, \underline{w}) p(\underline{w}|\mathbf{X}, \underline{y}) \ \mathrm{d}\underline{w} \ ,$$

where $p(\underline{w}|\mathbf{X}, \underline{y})$ is the posterior (Gaussian) pdf of the parameters $\underline{w}$ given the training data $(\mathbf{X}, \underline{y})$ and the parameter prior $p_0(\underline{w})$. The parameters $\underline{w}$ of a GP are also referred to as the *hyperparameters*. If we do not have the prior $p_0(\underline{w})$, the hyperparameters $\underline{w}$ can be estimated by maximizing the likelihood $p(\underline{y}|\mathbf{X}, \underline{w})$ with respect to $\underline{w}$.

Since all involved pdfs are assumed to be Gaussian, the prediction of y_* at $\underline{x}_*$ is determined by the mean μ_* and covariance σ_*^2 that are given by

$$\begin{aligned} \mu_* &= k(\underline{x}_*, \mathbf{X})^\top \mathbf{\Sigma}^{-1} \underline{y} \ , \\ \sigma_*^2 &= k(\underline{x}_*, \underline{x}_*) - k(\underline{x}_*, \mathbf{X})^\top \mathbf{\Sigma}^{-1} k(\underline{x}_*, \mathbf{X}) \ . \end{aligned} \tag{2}$$

Please note that although (2) has a quite simple formulation, its evaluation can be computationally intense if the set of training data and/or the dimension of the inputs are large.

B. Gaussian Processes over Probability Distributions

So far, the input $\underline{x}_*$ was assumed to be deterministic. As motivated above, many real-world applications of GPs and also the nonparametric (Bayesian) methods require to be able to deal with inputs provided in form of probability distributions. A general treatment of such inputs in Bayesian nonparametric methods is discussed in [3].

In the context of GPs, the foundation for consideration of probability distributions as inputs was laid in [4] where it was applied to prediction of time series. In this work, the authors considered the case where the training inputs $\underline{x}$ were deterministic vectors and the test inputs were normal distributions. By doing so, the training of the GP can be performed by standard means, while the predictive density of the output y_* for an input provided in form of a probability distribution $\mathfrak{d}_* = p_*(\underline{x}) \in \mathbb{R}^n$ can be computed according to

$$p(y_*|\mathfrak{d}_*, \mathbf{X}, \underline{y}) = \int\limits_{\mathbb{R}^n} p(y_*|\underline{x}, \mathbf{X}, \underline{y}) p_*(x) \ \mathrm{d}\underline{x} \ , \tag{3}$$

where $p(y_*|\underline{x}, \mathbf{X}, \underline{y})$ is Gaussian as defined in Sec. I-A. The authors of [4] further argue that an analytical evaluation of the integral in (3) is generally not tractable because $p(y_*|\underline{x}, \underline{w})$ is a complicated function of $\underline{x}$. For this reason, they propose two approximation schemes. In the first scheme, only the mean and covariance of the predictive density are computed under the assumption that the probability distributions that are provided as inputs are Gaussian and the covariance function is the Squared Exponential (SE) covariance function [2]. For this case, an approximate expression for the mean and the variance of the predictive density is provided in [4]. The exact solution can be found in [5] (see also [6], [7], [8]). The second approximation scheme consists in solving (3) using a Monte-Carlo approach according to

$$p(y_*|\mathfrak{d}_*, \mathbf{X}, \underline{y}) \approx \frac{1}{T} \sum_{t=1}^{T} p(y_*|\underline{x}_t, \mathbf{X}, \underline{y}) \ , \tag{4}$$

where $\underline{x}_t$ are samples drawn from $\mathfrak{d}_*$. In order to remain in the framework of GPs, i.e., in order to have Gaussian predictive densities, only the mean and covariance of (4) are maintained.

Another application of the first approximation method is considered in [9], where it is applied to stabilization of a nonlinear system. Here, not only the test inputs but also the training inputs are assumed to be Gaussian pdfs, i.e., the training data now becomes $(\mathfrak{D}, \underline{y}) = \{(\mathfrak{d}_1, y_1), \dots, (\mathfrak{d}_N, y_N)\}$ and the predictive density of y_* given the probability distribution $\mathfrak{d}_* = p_*(\underline{x})$ is given by

$$p(y_*|\mathfrak{d}_*, \mathfrak{D}, \underline{y}) = \int\limits_{\mathbb{R}^n} p(y_*|\underline{x}, \mathfrak{D}, \underline{y}) p_*(\underline{x}) \ \mathrm{d}\underline{x} \ . \tag{5}$$

In [9], the problem of training the GP model and making predictions is addressed by introducing the mean covariance function

$$k(\mathfrak{d}_i, \mathfrak{d}_j) = \int\limits_{\mathbb{R}^n} \int\limits_{\mathbb{R}^n} k(\underline{x}_i, \underline{x}_j) p_i(\underline{x}_i) p_j(\underline{x}_j) \ \mathrm{d}\underline{x}_i \ \mathrm{d}\underline{x}_j \ . \tag{6}$$

For this reason, we will refer to GPs constructed using the approach from [9] as *mean-kernel* GPs. As a special case, the authors of [9] consider the SE covariance function and provide an analytical expression for (6) (recall that the input distributions are assumed to be Gaussian). However, it is pointed out that estimating the hyperparameters of the GP using the maximum likelihood approach is not trivial because of the many local maxima of the log-likelihood $\log p(\underline{y}|\mathbf{X}, \underline{w})$. To avoid this issue, the authors propose to apply the Maximum A Posteriori (MAP) approach by defining a (Gaussian) prior for the hyperparameters.

A different approach to training and predicting with Gaussian inputs was presented in [10]. The authors propose to use a Taylor expansion of the measurement model

$$y = a(\tilde{\underline{x}} + \underline{\nu}_x) + \nu_y$$

in $\tilde{\underline{x}}$, where $\underline{\nu}_x \in \mathbb{R}^n$ is the Gaussian input noise and $\nu_y \in \mathbb{R}$ the Gaussian measurement noise. Although the derivatives in the expansion are again GPs [11] and the first and the second derivatives can be calculated in closed form, an exact evaluation is computationally inefficient. For this reason, the authors propose to apply approximations. This allows to derive a linear model for the input noise. However, the training of the GP for this (approximate) linear model is still non-trivial and requires further approximations.

Finally, another method that allows to use GPs with Gaussian inputs and its application to predictions with localization uncertainty is considered in [12]. The authors address the problem by two approaches: the Monte-Carlo and the Laplace approximations. While the Monte-Carlo approximation is standard, the application of the Laplace approximation in the context of GPs is new. The Laplace approximation is used to compute an approximation of the integral in (5). However, the evaluation of the Laplace approximation [13] is computationally intense. Therefore, the authors propose further approximations.

C. Contribution

As we have seen, considering GPs with probability distributions as inputs is non-trivial and each existing approach has its disadvantages. First of all, the reviewed approaches consider only inputs that are Gaussian. But, this limitation can be avoided by using the Monte-Carlo approach and maintaining only the first and the second moments for the prediction, which implicitly approximates the output density with a Gaussian. Then, although the approach from [4] is efficient, it does not allow for probability distributions as training inputs, while learning of the GP parameters of the method from [9] is not simple even in the considered case of Gaussian distributions as training and test inputs. The approach from [10] has to make approximations both in the measurement model and the learning of the GP parameters. Finally, [12] has also to make approximations in order to solve (5) and an additional approximation to reduce the computational cost. For this reasons, we propose a novel framework for Gaussian Processes with inputs provided in form of probability distributions that defines the covariance function directly in the space of (arbitrary) probability distributions. Furthermore, our approach is not limited to a specific family of probability distributions such as Gaussians. Moreover, it even allows to incorporate both continuous and Dirac probability distributions, i.e., discrete distributions over a continuous domain, into the same GP. Finally, the Gaussian property of the predictive density is implicitly ensured by construction and does not involve approximations.

covariance function	**expression**
constant	σ_0^2
squared exponential	$\exp\left(-\frac{1}{2}\frac{\Delta^2}{l^2}\right)$
Matérn	$\frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}}{l}\Delta\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}}{l}\Delta\right)$
exponential	$\exp\left(-\frac{\Delta}{l}\right)$
γ-exponential	$\exp\left(-\left(\frac{\Delta}{l}\right)^{\gamma}\right)$
rational quadratic	$\left(1+\frac{\Delta^2}{2\alpha l^2}\right)^{-\alpha}$

TABLE I: Covariance functions defined in terms of the distance $\Delta = d(\mathfrak{d}_i, \mathfrak{d}_j)$ between the probability distributions $\mathfrak{d}_i$ and $\mathfrak{d}_j$.

II. Proposed Framework

In this section, we present the proposed framework. As outlined above, the main notion of our framework consists in defining the covariance function of the GP directly in the space of probability distributions. The proposed approach primarily applies to stationary covariance functions, i.e., covariance functions that are defined over the distances between the inputs. However, we also discuss how non-stationary covariance functions can be implemented within the proposed framework.

A. Stationary Covariance Functions

To address the issue of GP inputs provided in form of probability distributions, we propose to *use stationary covariance functions that take the distance between two probability distributions as the argument*. As a motivation for this approach, consider for example the classical isotropic SE covariance function

$$k(\underline{x}_i, \underline{x}_j) = \alpha^2 \exp\left(-\frac{1}{2}\frac{(\underline{x}_i - \underline{x}_j)^\top(\underline{x}_i - \underline{x}_j)}{l^2}\right)$$

defined for deterministic inputs $\underline{x}_i$ and $\underline{x}_j$, where $\alpha, l \in \mathbb{R}$ are the hyperparameters. Here, the quadratic term $(\underline{x}_i - \underline{x}_j)^\top(\underline{x}_i - \underline{x}_j)$ corresponds to the squared Euclidean distance between $\underline{x}_i$ and $\underline{x}_j$. Now, instead of the vector-valued inputs $\underline{x}_i$ and $\underline{x}_j$, let the inputs be probability distributions $\mathfrak{d}_i$ and $\mathfrak{d}_j$. Then, we can redefine the SE covariance function according to

$$k(\mathfrak{d}_i, \mathfrak{d}_j) = \alpha^2 \exp\left(-\frac{1}{2}\frac{d(\mathfrak{d}_i, \mathfrak{d}_j)^2}{l^2}\right) ,$$

where $d(\mathfrak{d}_i, \mathfrak{d}_j)$ denotes any admissible distance measure between the input densities $\mathfrak{d}_i$ and $\mathfrak{d}_j$. The proposed design approach can be applied to any stationary covariance function. Table I provides a small overview of selected stationary covariance functions from [2] redefined in terms of the distance between probability distributions. A discussion of a set of selected admissible distance measures between probability distributions is given given in Sec. III.

Please note that the choice of stationary covariance functions defined over distances of probability distribution as the main basis of the proposed framework is justifiable because the space of probability distributions is not ordered or partially ordered. Therefore, non-stationary covariance function with absolute positions in the underlying space may not be necessary. Nevertheless, we discuss a method how such covariance functions still can be incorporated in the proposed framework in Sec. II-B.

It is worth noting that our approach somewhat resembles the approach from [9] that uses the mean covariance function (6). However, in the method from [9] it is necessary to compute the integrals in (6) in each iteration step of the optimization problem that is solved in order to determine the hyperparameters. In the proposed approach however, the computation of the distances between the input densities is independent of the hyperparameters and only has to be performed once. Moreover, the estimation of the hyperparameters remains the same as in the classical GP framework.

B. Non-stationary Covariance Functions

A direct usage of non-stationary covariance functions is not possible within the presented framework. Hence, we propose the following workaround. According to [2], it is possible to construct new covariance functions from existing ones using operations such as addition, multiplication, convolution, tensor product, etc. For construction of GPs with probability distributions provided as inputs and non-stationary covariance functions, we thus propose to combine stationary covariance functions with non-stationary functions that operate, e.g., on the means or the modes of the probability distributions provided as inputs. For example, the covariance function constructed from a linear function and the SE function according to

$$k(\mathfrak{d}_i, \mathfrak{d}_j) = \mathrm{E}\left\{\underline{x}_i\right\}^\top \Sigma_d \mathrm{E}\left\{\underline{x}_j\right\} \exp\left(-\frac{1}{2}\frac{d(\mathfrak{d}_i, \mathfrak{d}_j)^2}{l^2}\right)$$

is admissible, where Σ_d is a matrix of hyperparameters and the expectations are computed with respect to $\mathfrak{d}_i$ and $\mathfrak{d}_j$. More possible candidates for construction of non-stationary covariance functions can be found in [2].

III. Distance Measures for Probability Distributions

In this section, we provide a small, not necessarily complete overview of distance measures between multivariate probability distributions that can be used in the proposed framework. In particular, we analyze the following distances

- total variation and L_P distance,
- Hellinger distance,
- Jensen–Shannon divergence,
- Wasserstein/OSPA distance, and
- modified Cramér–von Mises distance.

A more thorough review of distance measures for probability distributions can be found, e.g., in [14]. In what follows, we not only present the distances listed above but also point out the classes of probability distributions that can be compared using the presented distances, i.e., whether the measures can be used to compute the distance between two continuous distributions, two Dirac mixture distributions, or a continuous and a Dirac mixture probability distribution. A Dirac mixture or particle distribution with m components is given by

$$f(\underline{x}) = \sum_{i=1}^{m} w_i \delta(\underline{x} - \underline{x}_i) \ ,$$

where $0 < w_i \leq 1$ are the weights with $\sum_{i=1}^{m} w_i = 1$ and $\underline{x}_i$ are the positions of the Dirac components. Such densities are very important, e.g., in robotics and nonlinear filtering.

1) Total Variation and L_p Distance:

The total variation distance [14] of two continuous probability distributions f and g with respect to a measure μ is defined according to

$$d_p(f, g) = \left(\int_\Omega (f(\underline{x}) - g(\underline{x}))^p \ \mathrm{d}\mu(\underline{x})\right)^{\frac{1}{p}} \ .$$

If we set $\mathrm{d}\mu(\underline{x}) = \mathrm{d}\underline{x}$, we obtain the L_p distance between f and g. Both these distance measures are defined only for two continuous distributions.

2) Hellinger Distance:

The Hellinger distance between two continuous probability distributions f and g is given by

$$d(f, g) = \left(1 - \int \sqrt{f(\underline{x}) g(\underline{x})} \ \mathrm{d}\underline{x}\right)^2 \ .$$

For this distance, it holds $0 \leq d(f, g) \leq 1$. Therefore, it is less suitable for application in the proposed framework compared to other distances that are unbounded. Furthermore, there is no counterpart of the Hellinger distance for two Dirac distributions or a continuous and a Dirac distribution.

3) Jensen–Shannon Divergence:

The Jensen–Shannon divergence that was introduced in [15] is a symmetric version of the Kullback–Leibler divergence [16]. For the continuous probability distributions f and g, it can be computed according to

$$d(f, g) = \frac{1}{2} \int f(\underline{x}) \log \frac{f(\underline{x})}{p(\underline{x})} + g(\underline{x}) \log \frac{g(\underline{x})}{p(\underline{x})} \ \mathrm{d}\underline{x} \ ,$$

where $p(\underline{x}) = (f(\underline{x}) + g(\underline{x}))/2$. Comparison of two Dirac distributions or a continuous and a Dirac distributions is not possible with this measure.

4) Wasserstein/OSPA Distance:

For the Wasserstein distance [17] of two continuous probability distributions f and g, it holds

$$d_p(f, g) = \inf_h \left(\int d_e(\underline{x}, \underline{y})^p h(\underline{x}, \underline{y}) \ \mathrm{d}\underline{x} \ \mathrm{d}\underline{y}\right)^{\frac{1}{p}} \ ,$$

where $d_e(\underline{x}, \underline{y})$ is the Euclidean distance between the vectors $\underline{x}$ and $\underline{y}$, and $h(\underline{x}, \underline{y})$ is a joint distribution whose marginals are $f(\underline{x})$ and $g(\underline{y})$, i.e., it holds $\int h(\underline{x}, \underline{y}) \ \mathrm{d}y = f(\underline{x})$ and $\int h(\underline{x}, \underline{y}) \ \mathrm{d}\underline{x} = g(\underline{y})$.

The analog of the Wasserstein distance between continuous distributions for two Dirac distributions with equal numbers of components, also referred to as the Optimal MAss Transfer (OMAT) metric [18], is presented in [19]. For the special case of m equally weighted Diracs, it holds

$$d_p(f,g) = \left(\frac{1}{m} \min_{\pi \in \Pi} \sum_{i=1}^{m} d(\underline{x}_i, \underline{y}_{\pi_i})^p \right)^{\frac{1}{p}} \quad ,$$

where Π is the set of all possible assignments between the Diracs from the two distributions. An extension of the Wasserstein distance to Dirac distributions with different numbers of components is presented in [18], where it is referred to as the Optimal Sub-Pattern Assignment (OSPA) metric.

Please note that the Wasserstein metric is generally intractable for continuous distributions, because the infimum has to be taken over all possible joint distributions h. Furthermore, the OMAT distance may not be efficiently computable for Dirac distributions with large numbers of components because it requires the solution of a linear assignment problem.

5) Modified Cramér–von Mises Distance:

The distance metrics for probability distributions presented so far are either limited to the same probability distribution class and cannot be used to compute the distance between a continuous and a Dirac distribution, or are intractable (Wasserstein distance for continuous distributions). For this reason, we propose to use the modified Cramér–von Mises distance (mCvMd) [20], [21].

In order to present the mCvMd between two arbitrary n-dimensional probability distributions f and g, we first introduce the notion of the Localized Cumulative Distribution $F(\underline{m}, b)$ of f according to

$$F(\underline{m}, b) = \int_{\mathbb{R}^n} f(\underline{x}) k(\underline{x}, \underline{m}, b) \ \mathrm{d}\underline{x} \ ,$$

where $k(\underline{x}, \underline{m}, b)$ is a Radial Basis Function (RBF)

$$k(\underline{x}, \underline{m}, b) = \exp\left(-\frac{1}{2} \frac{(\underline{x} - \underline{m})^\top (\underline{x} - \underline{m})}{b^2} \right) \quad .$$

Then, the mCvMd of f and g can be calculated according to

$$d(f,g) = \left(\int_0^{b_{\max}} \int_{\mathbb{R}^n} w(b) (F(\underline{m}, b) - G(\underline{m}, b))^2 \ \mathrm{d}\underline{m} \ \mathrm{d}b \right)^{\frac{1}{2}}$$

where $G(\underline{m}, b)$ is the LCD of $g(\underline{x})$ and $w(b)$ is a weighting function with

$$w(b) = \begin{cases} \frac{1}{b^{n-1}} & \text{for } b \leq b_{\max} \ , \\ 0 & \text{otherwise} \ , \end{cases}$$

and $b_{\max}$ is a large positive constant.

According to [22], the mCvMd for two Dirac distributions f and g with samples at $\underline{x}_i^f$ and $\underline{x}_j^g$, weights w_i^f and w_j^g, $i = 1, \ldots, M$, $j = 1, \ldots, L$, and a large $b_{\max}$ evaluates to

$$d(f,g)^2 = \frac{\pi^{\frac{n}{2}}}{8} \left(D_f - 2D_{fg} + D_g + 2c_b D_E \right) \quad ,$$

with $c_b = \log\left(4b_{\max}^2\right) - \Gamma$ and

$$D_f = \sum_{i=1}^{M} \sum_{j=1}^{M} w_i^f w_j^f \operatorname{xlog}\left((\underline{x}_i^f - \underline{x}_j^f)^\top (\underline{x}_i^f - \underline{x}_j^f) \right) \ ,$$

$$D_{fg} = \sum_{i=1}^{M} \sum_{j=1}^{L} w_i^f w_j^g \operatorname{xlog}\left((\underline{x}_i^f - \underline{x}_j^g)^\top (\underline{x}_i^f - \underline{x}_j^g) \right) \ ,$$

$$D_g = \sum_{i=1}^{L} \sum_{j=1}^{L} w_i^g w_j^g \operatorname{xlog}\left((\underline{x}_i^g - \underline{x}_j^g)^\top (\underline{x}_i^g - \underline{x}_j^g) \right) \ ,$$

$$D_E = \left(\sum_{i=1}^{M} w_i^f \underline{x}_i^f - \sum_{j=1}^{L} w_j^g \underline{x}_j^g \right)^\top \left(\sum_{i=1}^{M} w_i^f \underline{x}_i^f - \sum_{j=1}^{L} w_j^g \underline{x}_j^g \right) ,$$

where

$$\operatorname{xlog}(x) = \begin{cases} 0 & \text{for } x = 0 \ , \\ x \log(x) & \text{otherwise} \ . \end{cases}$$

A more thorough discussion of the mCvMd can be found in [23].

IV. Numerical Example

$$\begin{aligned} \widetilde{v}_1(\underline{x}) &= \mathrm{E}\left\{ \underline{x}^\top \underline{x} \right\} \\ &= \mathrm{E}\left\{ \underline{x} \right\}^\top \mathrm{E}\left\{ \underline{x} \right\} + \mathrm{E}\left\{ (\underline{x} - \mathrm{E}\left\{ \underline{x} \right\})^\top (\underline{x} - \mathrm{E}\left\{ \underline{x} \right\}) \right\} \\ &= \mu_x^\top \mu_x + \sigma_x^2 \ , \end{aligned} \tag{7}$$

where μ_x is the mean of $\underline{x}$ and σ_x^2 its variance, and a slightly modified version of the Rosenbrock function[2]

$$\widetilde{v}_2(x_1, x_2) = a(x_1 + b)^2 + c(x_1^2 - x_2 + d)^2 \ . \tag{8}$$

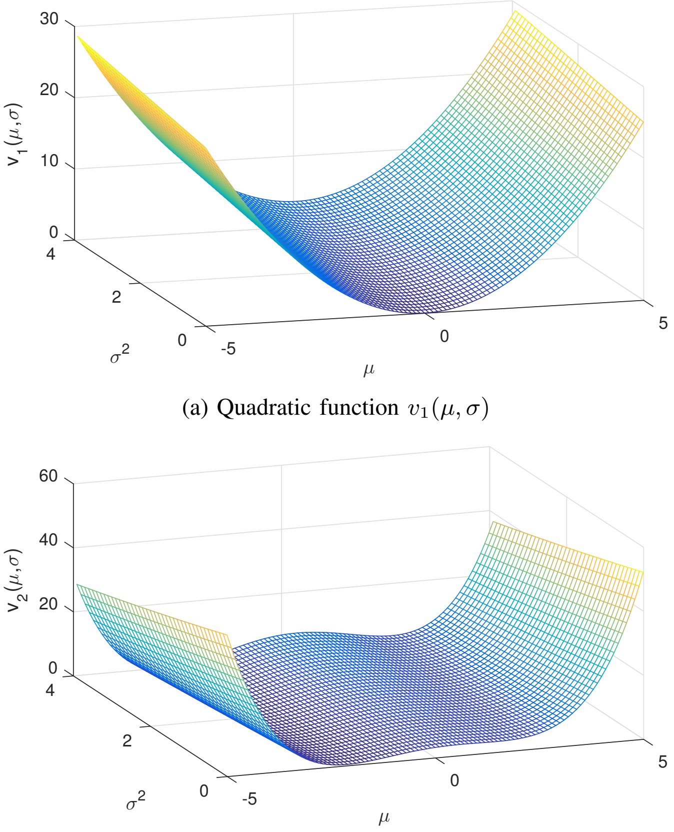
where we set $x_1 = \mu_x$, $x_2 = \sigma_x^2$, $a = c = 0.1$, $b = 4$, and $d = -4$. For the sake of visualization, we perform this demonstration over univariate Gaussians with means in the range $\mu \in [-5, 5]$ and covariances $\sigma^2 \in [0.1^2, 2^2]$. The two functions (7) and (8) then become

$$\begin{aligned} v_1(\mu, \sigma) &= \mu^2 + \sigma^2 \ , \\ v_2(\mu, \sigma) &= 0.1(\mu + 4)^2 + 0.1(\mu^2 - \sigma^2 - 4)^2 \ . \end{aligned}$$

The functions $v_1(\mu, \sigma)$ and $v_2(\mu, \sigma)$ are depicted in Fig. 2. In this section, we demonstrate the proposed framework applied to regression of two functions, a simple quadratic function

In particular, we proceed as follows. First, we draw 200 Gaussians for training from the considered range by drawing a mean and a variance and compute the outputs for each distribution using $v_1(\mu, \sigma)$ or $v_2(\mu, \sigma)$, respectively. The sampled Gaussians are depicted in Fig. 3, where each point represents a Gaussian with its corresponding mean and variance. For the simulation, we chose to represent the inputs to the GP as Dirac mixture distribution in order to include the GP constructed using the proposed framework with the Wasserstein distance. For this reason, we draw samples from the training Gaussians that are then used as inputs to the GP, i.e., each input is a set of samples drawn from one of the Gaussians that are used for training. The 10 samples from the Gaussians

[2] We added the parameter d.



(a) Quadratic function $v_1(\mu, \sigma)$

(b) Rosenbrock function $v_2(\mu, \sigma)$

Fig. 2: Functions $v_1(\mu, \sigma)$ and $v_2(\mu, \sigma)$ that are used in the numerical regression example.

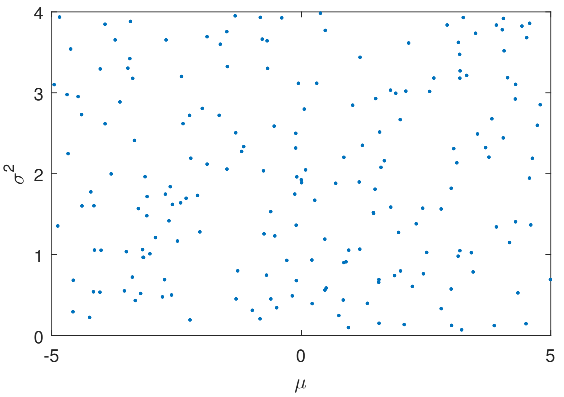


Fig. 3: Distributions used as training inputs. Each point represents a training Gaussian with mean μ and variance σ^2.

are drawn deterministically using the method from [21]. In summary, the training data for the GP consists of 200 sets à 10 samples and a function value for each sample set. We use this training data in order to train a GP that is based on the mean kernel [9] (here we used the original sampled mean and variances of the Gaussians), and two GPs designed according to the proposed framework, where one GP uses the mCvMd and the other the Wasserstein distance.

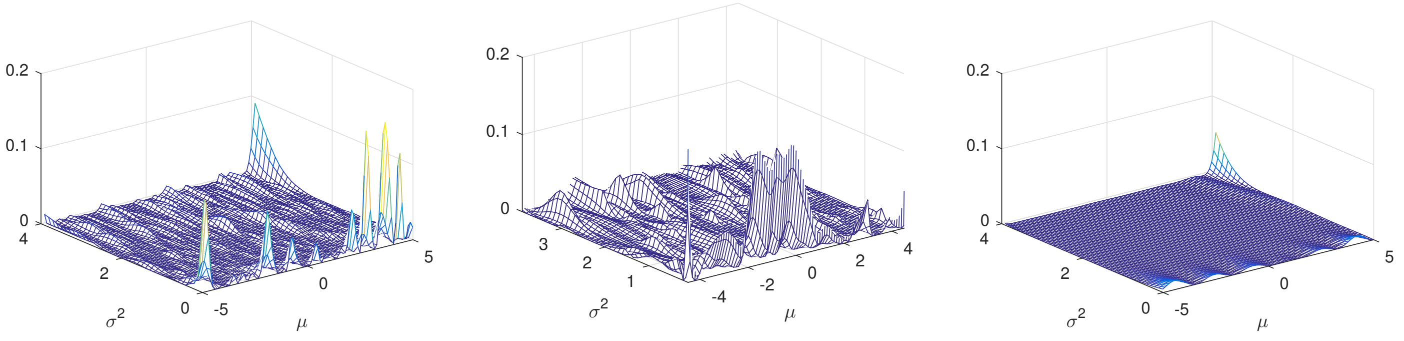
The simulation results can be seen in Figs. 4 and 5 that depict the quadratic errors of the regressions of the quadratic function $v_1(\mu, \sigma)$ and the Rosenbrock function $v_2(\mu, \sigma)$. Some of the figures have been cropped due to relatively large errors on the boundaries. In Fig. 4, it can be seen that the quadratic function $v_1(\mu, \sigma)$ can be approximated well by all three GPs, whereas the GP based on the mean kernel slightly outperforms the GP designed with the proposed framework that uses the mCvMd. The GP based on the Wasserstein distance performs worst but its performance is still comparable with the two other GPs.

In the scenario where we analyze the approximation of the Rosenbrock function $v_2(\mu, \sigma)$ (Fig. 5), the GP based on the mCvMd performs best and the GP based on the Wasserstein distance is only slightly worse. However, as in the regression of the quadratic function, the performance of the GP based on the Wasserstein distance is bad on the boundaries. For this reason, we suggest to use the mCvMd rather than the Wasserstein distance in the proposed framework. The mean-kernel GP [9] performs much worse than the GPs designed using the proposed distance-based framework. This is probably due to the wrong estimation of the hyperparameters that was performed using the maximum likelihood approach. From this issue, we may conclude that our framework is much more convenient in practice because the estimation of the hyperparameters is the same as in the classical GP framework with deterministic vector-valued inputs and therefore no prior is required. Furthermore, the estimation of the hyperparameters for the GP constructed according to our framework is faster because the distances between probability distributions that are provided as inputs are independent of the hyperparameters. On the other hand, the integrals in (6) have to be evaluated in each iteration step of the optimization algorithm that estimates the hyperparameters of the mean-kernel GP from [9].

A reference implementation of the proposed algorithm is available on GitHub [24].

V. Conclusion

In this paper, we proposed a framework for GPs, where the inputs are provided in form of probability distributions. The main notion of the proposed framework is to use stationary covariance functions that take the distances between the input probability distributions as arguments. We further discussed how it is possible to construct GPs with non-stationary kernels and compared several admissible distance measures. The proposed framework has the advantage that it can operate with arbitrary probability distributions if the appropriate distance measure is chosen. Moreover, we are able to construct GPs whose inputs contain continuous probability distributions and probability distributions represented using particles. In our numerical example, we compared our approach with an existing state-of-the-art method that is based on the notion of the mean kernel. The approximation quality of the GPs designed according to our framework was good for the considered quadratic and the Rosenbrock functions. Especially the GP based on the mCvMd performed well. In regression of the Rosenbrock function, our framework outperformed the mean-kernel approach from [9].

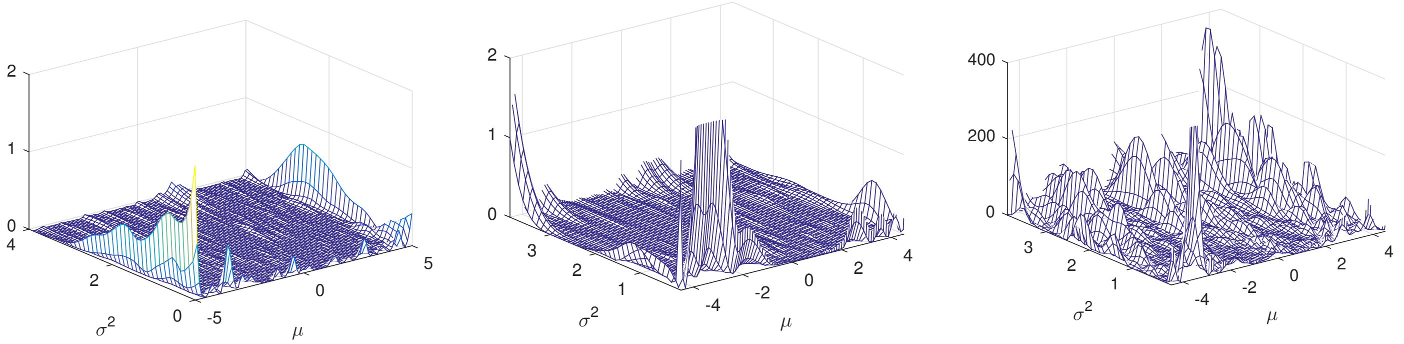


(a) proposed – mCvMd

(b) proposed – Wasserstein distance (cropped to $\mu \in [-4.5, 4.5]$, $\sigma^2 \in [.1, 3.9]$)

(c) mean kernel [9]

Fig. 4: Quadratic error of the GP regression of the quadratic function $v_1(\mu, \sigma)$.



(a) proposed – mCvMd

(b) proposed – Wasserstein distance (cropped to $\mu \in [-4.5, 4.5]$, $\sigma^2 \in [.1, 3.9]$)

(c) mean kernel [9] (cropped to $\mu \in [-4.5, 4.5]$, $\sigma^2 \in [.1, 3.9]$)

Fig. 5: Quadratic error of the GP regression of the Rosenbrock function $v_2(\mu, \sigma)$.

References

[1] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[2] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[3] P. Dellaportas and D. A. Stephens, “Bayesian Analysis of Errors-in-Variables Regression Models,” *Biometrics*, vol. 51, no. 3, pp. 1085–1095, 1995.

[4] A. Girard, J. Q. Candela, R. Murray-Smith, and C. E. Rasmussen, “Gaussian Process Priors with Uncertain Inputs – Application to Multiple-Step Ahead Time Series Forecasting,” in *Advances in Neural Information Processing Systems*, S. Thrun and K. Obermayer, Eds. Cambridge, MA: MIT Press, 2002, no. 15.

[5] J. Q. Candela, A. Girard, J. Larsen, and C. E. Rasmussen, “Propagation of Uncertainty in Bayesian Kernel Models - Application to Multiple-step ahead Forecasting,” in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*, Hong Kong, Apr. 2003.

[6] A. Girard, “Approximate Methods for Propagation of Uncertainty with Gaussian Process Models,” Ph.D. dissertation, University of Glasgow, 2004.

[7] J. Candela, “Learning with Uncertainty - Gaussian Processes and Relevance Vector Machines,” Ph.D. dissertation, Technical University of Denmark, 2004.

[8] A. Girard and R. Murray-Smith, “Gaussian Processes: Prediction at a Noisy Input and Application to Iterative Multiple-step ahead Forecasting of Time-Series,” in *Switching and Learning in Feedback Systems*. Springer, 2005, pp. 158–184.

[9] P. Dallaire, C. Besse, and B. Chaib-draa, “Learning Gaussian Process Models from Uncertain Data,” in *Proceedings of the 16th International Conference on Neural Information Processing (ICONIP 2009)*, Bangkok, Thailand, Dec. 2009.

[10] A. McHutchon and C. E. Rasmussen, “Gaussian Process Training with Input Noise,” in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 1341–1349.

[11] E. Solak, R. Murray-Smith, W. Leithead, D. Leith, and C. Rasmussen, “Derivative Observations in Gaussian Process Models of Dynamic Systems,” in *Advances in Neural Information Processing Systems 15*. Cambridge, MA, USA: MIT Press, Oct. 2003, pp. 1033–1040.

[12] M. Jadaliha, Y. Xu, J. Choi, N. S. Johnson, and W. Li, “Gaussian Process Regression for Sensor Networks Under Localization Uncertainty,” *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 223–237, 2013.

[13] L. Tierney and J. B. Kadane, “Accurate Approximations for Posterior Moments and Marginal Densities,” *Journal of the American Statistical Association*, vol. 81, no. 393, pp. 82–86, 1986.

[14] V. M. Zolotarev, “Probability Metrics (in Russian),” *Teoriya Veroyatnostei i ee Primeneniya*, vol. 28, no. 1, pp. 278–302, 1983.

[15] J. Lin, “Divergence Measures Based on the Shannon Entropy,” *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 2006.

[16] R. A. L. S. Kullback, “On information and sufficiency,” *Annals of Mathematical Statistics*, vol. 21, no. 1, pp. 79–86, 1951.

[17] L. N. Wasserstein, “Markov Processes Over Denumerable Products of Spaces, Describing Large Systems of Automata,” *Problems of Information Transmission*, vol. 5, no. 3, pp. 47–52, 1969.

[18] D. Schuhmacher, B. T. Vo, and B. N. Vo, “A Consistent Metric for Performance Evaluation of Multi-Object Filters,” *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.

[19] J. R. Hoffman and R. P. S. Mahler, “Multitarget Miss Distance and its Applications,” in *Proceedings of the Fifth International Conference on Information Fusion (Fusion 2002)*, Jul. 2002.

[20] U. D. Hanebeck and V. Klumpp, “Localized Cumulative Distributions and a Multivariate Generalization of the Cramér-von Mises Distance,” in *Proceedings of the 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008)*, Seoul, Republic of Korea, Aug. 2008.

[21] U. D. Hanebeck, M. F. Huber, and V. Klumpp, “Dirac Mixture Approximation of Multivariate Gaussian Densities,” in *Proceedings of the 2009 IEEE Conference on Decision and Control (CDC 2009)*, Shanghai, China, Dec. 2009.

[22] U. D. Hanebeck, “Optimal Reduction of Multivariate Dirac Mixture Densities,” *at - Automatisierungstechnik*, vol. 63, no. 4, 2015.

[23] I. Gilitschenski, “Deterministic Sampling for Nonlinear Dynamic State Estimation,” Ph.D. dissertation, Karlsruhe Institute of Technology, 2015.

[24] M. Dolgov, “Distance-based Gaussian Processes,” www://github.com/Mxttak/GP.