2017 American Control Conference
Sheraton Seattle Hotel
May 24–26, 2017, Seattle, USA

Lipschitz Optimisation for Lipschitz Interpolation*

Jan-Peter Calliess[1]

Abstract— **Techniques known as *Nonlinear Set Membership* prediction, *Kinky Inference* or *Lipschitz Interpolation* are fast and numerically robust approaches to nonparametric machine learning that have been proposed to be utilised in the context of system identification and learning-based control. They utilise *presupposed* Lipschitz properties in order to compute inferences over unobserved function values. Unfortunately, most of these approaches rely on exact knowledge about the input space metric as well as about the Lipschitz constant. Furthermore, existing techniques to estimate the Lipschitz constants from the data are not robust to noise or seem to be ad-hoc and typically are decoupled from the ultimate learning and prediction task. To overcome these limitations, we propose an approach for optimising parameters of the presupposed metrics by minimising validation set prediction errors. To avoid poor performance due to local minima, we propose to utilise Lipschitz properties of the optimisation objective to ensure global optimisation success. The resulting approach is a new flexible method for nonparametric black-box learning. We illustrate its competitiveness on a set of benchmark problems.**

I. Introduction

Supervised machine learning methods are algorithms for inductive inference. On the basis of a sample, they construct (learn) a computable model of a data generating process that facilitates inference over the underlying ground truth function and aims to predict its function values at unobserved inputs.

Among supervised learning methods, nonparametric algorithms tend to offer greater flexibility to learn rich function classes. Unfortunately, many classical techniques for nonparametric regression, such as the *Nadaraya-Watson estimator* [1], [2] or the *LOESS* method, [3] suffer from a practical limitation: their regression performance depends on the choice of hyperparameters. While in principle, it would be possible to tune these to the data (in manner similar in spirit to the one we propose in this work), to the best of our knowledge, currently there is little understanding on how to do so with a global optimiser that offers theoretical performance guarantees on the optimisation solution. This means that in practice, one is left to engineer these hyperparameters (or the settings of an optimiser) by manual tuning in order to ensure good performance on a particular learning problem. Of course, this stands in opposition to the motivation for utilising nonparametric learning, especially in system identification: which is to facilitate flexible and fully automated black-box learning that does not require manual intervention.

*I would like to thank Carl Rasmussen and Jan Maciejowski for helpful discussions and encouraging feedback. Also, funds via EPSRC NMZR/031 RG64733 are gratefully acknowledged.

[1]Jan-Peter Calliess is with the Engineering Department, University of Cambridge, UK. jpc73@cam.ac.uk

Perhaps one of the most popular nonparametric machine learning method is Bayesian inference with *Gaussian processes (GPs)* [4]. GPs offer a flexible and principled probabilistic method for nonparametric regression and have evolved into one of the chief work-horses for learning dynamic systems [5], [6], [7], [8], [9] in the research communities related to artificial intelligence and, more recently, also in control.

To address the problem of hyper-parameter determination, it is common practice to tune these to explain the data via the optimisation of the marginal log-likelihood [4]. While often successful on many data sets, the result can be highly sensitive to the choice of optimiser, initialisations, data sets and computational budget. Unfortunately, little theoretical understanding of the important interplay between these components in the resulting inference mechanism seems to exist.

In contrast to such approaches, our work builds on nonparametric regression techniques that harnesses Lipschitz regularity of the target function to provide bounds around the predictions. Applied to machine learning, the basic idea is that Lipschitz continuity constrains the set of possible function values of a target function at a query input, dependent on the distance between the query and the previously observed training examples. A prediction is then made by choosing a function value in the middle of the set of possible function values. This idea, at least going back to the era of "Russian mathematics" [10], has been redeveloped and advanced under different headlines including *Lipschitz Interpolation* [11], [12], *Nonlinear Set Membership (NSM)* interpolation methods [13] and *Kinky Inference (KI)* [14]. The presupposed Lipschitz constant as well as the assumed input space metric are crucial hyper-parameter choices of these methods that can drastically affect the predictive behaviour of trained inference rule. While a variety of Lipschitz constant estimators are known, they are designed independently from the prediction task and tend to be sensitive to noise. As an alternative, [13] proposed a method where the Lipschitz constant is estimated from a parametric model that is fitted to the data (e.g. a linear model or a neural network). However, it remains unclear in how far this Lipschitz constant will aide the predictive performance of the Lipschitz interpolation model.

We propose a different approach: Firstly, we consider the more general problem of optimising for parameters of a chosen pseudo-metric. (As we will see the Lipschitz constant determination problem can be cast as a special case of this). We then determine these free parameters by minimising an empirical estimate of the prediction error directly of the KI rule. When purposefully stating the error as an ℓ_1-loss, we can derive a Lipschitz constant for the optimisation

objective. This allows us to employ Lipschitz optimisation. In contrast, to other hyperparameter tuning approaches utilised in nonparametric machine learning, this global optimisation approach offers bounds on the optimisation success and can avoid falling into suboptimal solutions.

The result of this merger of Lipschitz optimisation for parameter optimisation and KI (based on the newly found parameters) is a reliable and fast nonparametric machine learning approach which we will refer to as *Parameter Optimised Kinky Inference (POKI)*.

Apart from the application to the automated determination of the Lipschitz constant in nonlinear set membership methods, we discuss other settings where our POKI approach uncovers inherent low-dimensionality (automated relevance determination) to learn good predictive models from data. For an extended preprint version of this paper, containing additional comparisons and explanatory material, the reader is referred to [15].

II. Kinky Inference and Lipschitz Interpolation

In this section, we will rehearse the class of learning rules sometimes referred to as *Kinky Inference*. They encompass a host of other methods such as Lipschitz Interpolation and Nonlinear Set Interpolation.

Setting. Let $f : \mathcal{X} \to \mathcal{Y}$ be a *target* or *ground-truth* function we desire to learn and let $\mathcal{X}$, $\mathcal{Y}$ be two spaces endowed with (pseudo-) metrics $\mathfrak{d} : \mathcal{X}^2 \to \mathbb{R}_{\geq 0}$, $\mathfrak{d}_{\mathcal{Y}} : \mathcal{Y}^2 \to \mathbb{R}_{\geq 0}$, respectively.

For simplicity, we restrict our exposition to real-valued targets with $\mathfrak{d}_{\mathcal{Y}}(y, y') = |y - y'|$.

Assume that, at time step n, we have access to a *sample* or *data set* $\mathcal{D}_n := \{(s_i, \tilde{f}_i) \,|\, i = 1, \ldots, N_n\}$ containing $N_n \in \mathbb{N}$ (possibly corrupted) sample values $\tilde{f}_i \in \mathcal{Y}$ of *target function* f at sample input $s_i \in \mathcal{X}$. The sampled function values are allowed to have *observational error* given by a (potentially stochastic) error function $\mathfrak{e} : \mathcal{X} \to \mathbb{R}^m_{\geq 0}$. That is, all we know is that $\tilde{f}_i = f(s_i) + \mathfrak{e}(s_i)$. For convenience, we may also write $\mathcal{D}_n = (\mathcal{X}_n, \mathcal{Y}_n)$ where $\mathcal{X}_n = \{s_i | i = 1, ..., N_n\} \subset \mathcal{X}$ is the collection of sample inputs and $\mathcal{Y}_n = \{\tilde{f}_i | i = 1, ..., N_n\} \subset \mathcal{Y}$ is the sequence of observed function values. It is our aim to learn target function f in the sense that utilise the available data $\mathcal{D}_n$ to infer *predictions* $\hat{\mathfrak{f}}_{\mathfrak{n}}(x)$ of $f(x)$ at unobserved *query inputs* $x \notin \mathcal{X}_n$. In our context, the evaluation of $\hat{\mathfrak{f}}_n$ is what we refer to as *(inductive) inference* or *prediction*.

The entire function $\hat{\mathfrak{f}}_{\mathfrak{n}}$ that is learned to facilitate predictions is referred to as the *predictor*.

To ground the inference, a priori assumptions are necessary. In our context, we will generally assume that the target can be arbitrarily well approximated by Lipschitz continuous function with some Lipschitz constant. Remember, relative to our chosen pseudo-metrics, a real-valued function ϕ is Lipschitz continuous (with Lipschitz constant $\ell \geq 0$) on domain $I \subset \mathcal{X}$ if $\mathfrak{d}_{\mathcal{Y}}(\phi(x), \phi(x')) \leq \ell\, \mathfrak{d}(x, x'), \forall x, x' \in I$. Note the metrics, may depend on a parameter θ. In this case, we highlight this dependency explicitly by writing $\mathfrak{d}(\cdot, \cdot; \theta)$ instead of $\mathfrak{d}(\cdot, \cdot)$. Of course, the Lipschitz parameter can be absorbed into the parameter of a chosen pseudo-metric.

The approximation success of $\hat{\mathfrak{f}}_{\mathfrak{n}}$ relative to a target can be measure by various metrics. In this work, we will be most interested in the *$\mathcal{L}_1$-prediction error* $\mathcal{E}_1(\hat{\mathfrak{f}}_{\mathfrak{n}}; f) := \left\| \hat{\mathfrak{f}}_{\mathfrak{n}} - f \right\|_1 = \int_{\mathcal{X}} \left| \hat{\mathfrak{f}}_{\mathfrak{n}}(x) - f(x) \right| \mathrm{d}x$. In practice, this error can be estimated by the *empirical prediction error* estimate

$$\hat{\mathcal{E}}_1(\hat{\mathfrak{f}}_{\mathfrak{n}}; f) := \frac{1}{|\mathcal{X}_{sample}|} \sum_{x \in \mathcal{X}_{sample}} \left| \hat{\mathfrak{f}}_{\mathfrak{n}}(x) - f(x) \right| \qquad (1)$$

where the *sample* $\mathcal{X}_{sample}$ is a finite set of sample inputs.

The average test set prediction error serves as a surrogate measure for the true prediction error; if the test set is chosen sufficiently dense (and the predictor and target are continuous) then $\hat{\mathcal{E}}_1 \approx \mathcal{E}_1$. In the case of i.i.d. samples, we can also construe Eq. 1 as a Monte-Carlo estimate of the $\mathcal{L}_1$-error.

In case we do not have access to a the target (or a noise-free sample), we might have to base our assessment of the method on the *empirical sample prediction error*

$$\tilde{\mathcal{E}}_1(\hat{\mathfrak{f}}_{\mathfrak{n}}; \tilde{f}) := \frac{1}{|\mathcal{X}_{sample}|} \sum_{x \in \mathcal{X}^{\text{test}}} \left| \hat{\mathfrak{f}}_{\mathfrak{n}}(x) - \tilde{f}(x) \right|. \qquad (2)$$

Learning rule. In this work we will expand on the basis of a special case of the class of kinky inference predictors [14] to perform learning as inference over unobserved function values. The prediction rule can be stated as follows:

Definition II.1 (Kinky inference (KI) rule (simplified))**.** Given access to a sample set $\mathcal{D}_n$ and an input space pseudo-metric $\mathfrak{d}(\cdot, \cdot; \theta(n)) : \mathcal{X}^2 \to \mathbb{R}$ parameterised by $\theta(n)$, we define the KI predictor by $\hat{\mathfrak{f}}_{\mathfrak{n}}\big(\cdot; \theta(n), \mathcal{D}_n\big) : \mathcal{X} \to \mathcal{Y}$ to perform inference over function values as per:

$$\hat{\mathfrak{f}}_{\mathfrak{n}}\big(x; \theta(n), \mathcal{D}_n\big) := \frac{1}{2}\mathfrak{u}_n(x; \theta(n)\big) + \frac{1}{2}\mathfrak{l}_n(x; \theta(n)\big).$$

Here, $\mathfrak{u}_n\big(\cdot; \theta(n)\big), \mathfrak{l}_n\big(\cdot; \theta(n)\big) : \mathcal{X} \to \mathbb{R}^m$ are called ceiling and floor functions, respectively. They are given by $\mathfrak{u}_n\big(x; \theta(n)\big) := \min_{i=1,\ldots,N_n} \tilde{f}_i + \mathfrak{d}(x, s_i; \theta(n))$ and $\mathfrak{l}_n\big(x; \theta(n)\big) := \max_{i=1,\ldots,N_n} \tilde{f}_i - \mathfrak{d}(x, s_i; \theta(n))$, respectively.

In the literature, various generalisations and special cases exist. For instance, Calliess [14] proposes a generalised framework called *Kinky Inference*, that also allows for the specification of additional parameters. These include functions that allow the incorporation of a priori knowledge about upper and lower bounds on the target as well as upper and lower bounds on observational noise.

A special case arises for the choice of $\mathfrak{d}(x, y; \theta(n)) = \ell(n)\,\|x - y\|$ which is referred to as *Lipschitz Interpolation* [12] or as *Nonlinear Set Interpolation* [13]. Here the parameter $\theta(n) = \ell(n)$ is the supposed Lipschitz constant of the target. Typically this constant is assumed to be either known a priori or estimated lazily from the data, e.g. [16] as follows:

$$\hat{\ell}(n) := \max_{i \neq j} \frac{\left| \tilde{f}_i - \tilde{f}_j \right|}{\|s_i - s_j\|}. \qquad (3)$$

Unfortunately, the latter estimate has the problem of being unbounded in the presence of observational noise. In the case of bounded noise, [14] proposed to utilise the alternative estimator $\ell(n) := \max_{i \neq j} \frac{\left|\tilde{f}_i - \tilde{f}_j\right| - 2\bar{\mathfrak{e}}}{\|s_i - s_j\|}$ where $\bar{\mathfrak{e}} := \sup_x |\mathfrak{e}(x)|$ is an upper bound on the (zero-mean) noise. While this prevents the estimates to blow up in the bounded noise case, generally it is not clear how to obtain $\bar{\mathfrak{e}}$. And, if $\bar{\mathfrak{e}}$ is chosen too conservatively large then the prediction quality is questionable. As an alternative approach, we will employ Lipschitz optimisation to find a parameter $\theta(n)$ that minimises empirical prediction error on a validation data set.

III. Parameter Optimised Kinky Inference (POKI)

Given some separate data sets $\mathcal{D}^{\text{cond}}$ and $\mathcal{D}^{\text{eval}}$, we will aim to choose a parameter of the pseudo-metric that minimises the empiricial sample prediction error. That is, we choose our parameter to be

$$\hat{\theta} := \text{argmin}_{\theta \in \Theta} \mathfrak{l}(\theta; \mathcal{D}^{\text{cond}}, \mathcal{D}^{\text{eval}}) \tag{4}$$

where Θ denotes a predefined parameter space and the loss quantifies the empirical sample prediction error quantified by the loss function $\mathfrak{l}(\cdot; \mathcal{D}^{\text{cond}}, \mathcal{D}^{\text{eval}}) : \mathcal{X} \to \mathbb{R}$ with

$$\mathfrak{l}(\theta; \mathcal{D}^{\text{cond}}, \mathcal{D}^{\text{eval}}) = \frac{1}{|\mathcal{X}^{\text{eval}}|} \sum_{x \in \mathcal{X}^{\text{eval}}} \left| \tilde{f}(x) - \hat{\mathfrak{f}}_{\mathfrak{n}}(x; \theta, \mathcal{D}^{\text{cond}}) \right|. \tag{5}$$

Here, we refer to $\mathcal{D}^{\text{eval}} = \left(\mathcal{X}^{\text{eval}}, \mathcal{Y}^{\text{eval}}\right)$ as the available *parameter evaluation data* and $\mathcal{D}^{\text{cond}}$ as the *conditioning data*.

To avoid overfitting and exact interpolation of the noise, we will generally ensure that the conditioning and evaluation data are different, i.e. at least $\mathcal{D}^{\text{eval}} \neq \mathcal{D}^{\text{cond}}$. Ideally, they should be drawn independently from each other. We will give an illustration of the utility of this approach below. For now, we will confine ourselves to describe the approach we propose:

Def. (Parameter Optimised Kinky Inference (POKI)): Assume at trial n, we are given access to a set of training examples $\mathcal{D}_n$ of size N_n.

We will undergo the following steps:

1) We randomly partition the data into two sub sets: a *conditioning data* set $\mathcal{D}_n^{\text{cond}}$ and a parameter *evaluation data* set $\mathcal{D}_n^{\text{eval}}$, $\mathcal{D}_n^{\text{tex}} = \mathcal{D}_n^{\text{cond}} \dot{\cup} \mathcal{D}_n^{\text{eval}}$. Unless explicitly stated otherwise, both sets will be made close to equal in size.
2) Utilising the conditioning and evaluation data sets, we compute the minimum-loss parameter estimate $\hat{\theta}_n := \text{argmin}_{\theta \in \Theta} \mathfrak{l}(\theta; \mathcal{D}_n^{\text{cond}}, \mathcal{D}_n^{\text{eval}})$ (cf. Eq. 4).
3) For prediction of future query inputs $q \in \mathcal{X}$, we use the KI prediction $\hat{\mathfrak{f}}_{\mathfrak{n}}(q; \hat{\theta}_n, \mathcal{D}_n)$ (cf. Def. II.1), utilising the learned newly identified parameter estimate $\hat{\theta}_n$ and conditioning on the full set of available data.

We refer to the resulting predictor $\hat{\mathfrak{f}}_{\mathfrak{n}}(\cdot; \hat{\theta}_n, \mathcal{D}_n^{\text{tex}})$ as a *Parameter Optimised Kinky Inference (POKI)* rule.

A. Lipschitz optimisation and parameter determination

In order to guarantee the quality of the parameter estimate $\hat{\theta}_n$ even in the presence of non- differentiabilities and local optima of the loss function, we propose to utilise Lipschitz optimisation methods. In one dimensional settings we could for instance utilise Shubert's method [17]. Various extensions to the multi-dimensional case exist including stochastic methods that offer probabilistic bounds that allow for guarantees with computational effort scaling linearly with the number of dimensions [18] as well as recent local approaches based on regret analysis [19]. In this work we use the simple approach sketched in Sec. 2.5 of [14] (also see [15]). While the effort to reduce the given worst-case bound scales exponentially in the number of dimensions, the approach has the advantage of being simple and giving deterministic bounds. However, we intend to try out the more advanced, scalable approaches to Lipschitz optimisation in the course of future work.

In our chosen Lipschitz optimisation method, if we know a Lipschitz constant $L(\Omega)$ of the objective function

$$\Omega : \begin{cases} \Theta \to \mathbb{R} \\ \theta \mapsto \mathfrak{l}(\theta; \mathcal{D}_n^{\text{cond}}, \mathcal{D}_n^{\text{eval}}) \end{cases}$$

then we can find a guaranteed global minimum up to a worst-case error bound that can be pre-specified in advance, thereby avoiding the pitfall of local minima.

To this end, we need to determine the Lipschitz constant of Ω, i.e. of the loss as a function of the parameter θ. We denote the Lipschitz constant of a function f by $L(f)$. For simplicity, we choose the metric induced by the maximum-norm to define Lipschitz continuity of Ω. Hence, for a given loss we desire to derive a nonnegative number $L(\Omega) \in \mathbb{R}_+$ such that: $\forall \theta, \theta' \in \Theta : |\Omega(\theta) - \Omega(\theta')| \leq L(\Omega) \ \|\theta - \theta'\|_\infty$.

Invoking the rules of Lipschitz arithmetic (see Lem. 2.5.6 in [14] or results in [20]) and utilising the definition of the kinky inference predictor, it is not hard to show that we have:

$$L(\Omega) \leq \max\{L\big(\mathfrak{d}(s_j, s_k; \cdot)\big) \,|\, j, k = 1, \ldots, N_n\} \tag{6}$$

where, for every pair s_j, s_k of inputs in the available data, $\mathfrak{d}(s_j, s_k; \cdot)\big)$ is a function of hyper-parameter θ. So, the determination of a bound on the desired Lipschitz constant depends on the conditioning data and the arithmetic definition the parameter θ within the chosen pseudo-metric $\mathfrak{d}$. For several cases of general interest, we will next provide some Lipschitz constant derivations. The best Lipschitz constant of a function ϕ will be denoted by $L(\phi)$.

- **Automated Lipschitz constant determination:** As mentioned above, we might seek to perform Lipschitz interpolation with automated Lipschitz constant determination. To this end, we might choose the metric $\mathfrak{d}(x, y; \theta_n) = \theta_n \|x - y\|$ where parameter θ acts as the assumed Lipschitz constant of the predictor. Clearly, we have $L(\mathfrak{d}(s_i, s_k; \cdot)) \leq |s_j - s_k|$. Thus, the Lipschitz constant of the optimisation objective Ω can be bounded by the diameter of the sample input set $\mathcal{X}_n$: $L(\Omega) \leq \max\{|s_j - s_k| \,|\, j, k = 1, \ldots, N_n\}$.

We refer to the resulting POKI inference rule as *POKI-LC* (where *LC* stands for Lipschitz constant). Illustrations of the performance of this approach in contrast other methods are provided in the experimental section.

- **Automated Relevance Determination (ARD).** A more general case of Lipschitz parameter determination is Automated Relevance Determination (ARD). Here, the goal is to find weights encoding the extent to which input space parameters are relevant for the prediction. This is interesting in high-dimensional input space where the input vectors might be features whose predictive role might be unclear a priori. To facilitate ARD, we might choose the metric $\mathfrak{d}(x, x'; \theta) = \max_{i=1,\ldots,d} \theta_i \, |x_i - x'_i|$. The weight θ_i quantifies the degree to which the ith input component should contribute to the prediction. Large weights suggest a strong degree of importance: i.e. small deviations of a query x_i from the closest example $s_{j,i}$ in the ith component will result in large uncertainty of the prediction. Conversely, if $\theta_i = 0$ effectively disables any influence of the ith input dimension in the prediction process.
  Analogously to the kernel literature, inferring θ from the data will be referred to as *automated relevance determination (ARD)* and the resulting POKI rule will be referred to as *POKI-ARD*.
  To facillitate Lipschitz optimisation of the parameter vector, we need to derive a Lipschitz constant bound. Again appealing to the rules of Lipschitz arithmetic, we see that $L(\,\mathfrak{d}(s_i, s_k; \cdot)) \leq \|s_i - s_k\|_\infty$ and thus, $L(\Omega) \leq \max\{\|s_j - s_k\|_\infty \mid j, k = 1, \ldots, N_n\}$. In other words, the Lipschitz constant of the loss is bounded from above by the diameter of the data relative to the standard maximum-norm.

IV. Experiments

In this section, we compare our approach to a number of well-established machine learning methods on artificial data. Comparisons on real data are provided in the long version of this article [15].

In order to have access to the ground-truth, we first tested the method on a sequence of artificial benchmark regression tasks. As the ground truth, we chose the target function $f : \mathcal{X} := [0, 1]^d \to \mathbb{R}, x \mapsto |-\cos(2\pi\, x_1)| + x_1$. Its values have a linear trend as well as a strongly nonlinear one, varying only with the first input component.

The data was obtained by artificially sampling uniformly $\mathcal{X}$ and superimposing the function values with i.i.d. zero-mean Gaussian noise with standard deviation $\frac{1}{4}$.

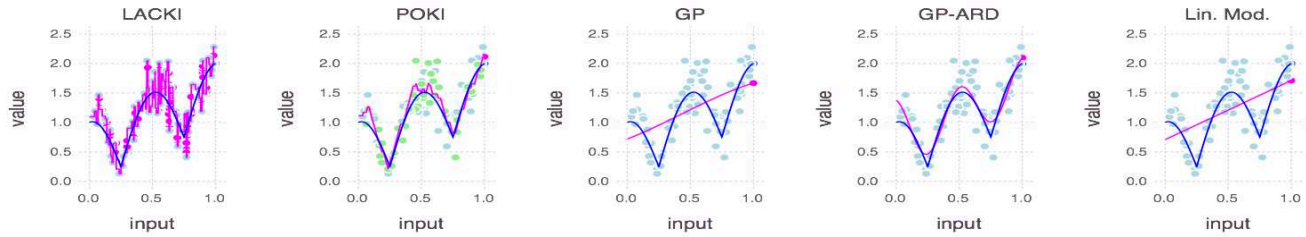
That is the sample was obtained from randomly selected inputs of the noisy data-generating "test function" $\tilde{f}(x) = f(x) + \nu(x)$ where the perturbations $\nu(x)$ were drawn i.i.d. from a distribution.

We trained predictors based on the following learning methods:

1) Linear regression model (**Lin. Mod.**) trained on the data with the least-squares method.
2) A Gaussian process (**GP**) provided with the correct observational noise level. The prior was based on a zero-mean function $\mu(\cdot) \equiv 0$ and a squared-exponential (SE) ARD-kernel $k(x, x') = \omega \exp\big(-\sum_{i=1}^{d} \frac{(x_i - x_i)^2}{\ell_i}\big)$ with relevance length scale parameters ℓ_i chosen uniformly to be $\ell_i = 0.5, \forall i$ and output-scale parameter chosen to be $\omega = 1$.
3) A kinky inference predictor (**LACKI**) with $\mathfrak{d}(x, x'; \theta) = \theta \, \|x - x'\|_\infty$ where the parameter was the Lipschitz constant $\theta = L(n)$ lazily estimated as per Eq. 3. This is equivalent to the LACKI method described in [21] with hyper-parameter choices $\underline{L} = 0$ and $\lambda = 0$.
4) Our parameter optimised kinky inference approach (**POKI-LC**) where again $\mathfrak{d}(x, x'; \theta) = \theta \, \|x - x'\|_\infty$.
5) Our parameter optimised kinky inference approach (**POKI-LC2**) trained as **POKI-LC** but where the parameter was optimised with Brent's method instead of Lipschitz optimisation.
6) Our parameter optimised kinky inference approach (**POKI-ARD**) where the metric was an ARD metric with relevance parameters optimised as described above.
7) A Gaussian process (**GP-ARD**) with a SE-ARD kernel as above, but whose relevance hyper-parameters were optimised with the conjugate-gradient approach to maximise the marginal log-likelihood [4].

Exp.1 : As a first illustration, we considered regression on one-dimensional input space ($d = 1$) based on a sample of 84 observations corrupted by zero-mean Gaussian noise with standard deviation $\frac{1}{4}$, i.e. $\nu(x) \overset{i.i.d.}{\sim} \mathcal{N}(0, \frac{1}{16})$. The results for a selection of the methods are depicted in Fig. 1(a). We can see that the linear regression method accurately fitted the linear trend in the data but does not explain the nonlinear variation. For the given hyper-parameter settings, the GP was clearly over-smoothing the data. This might well have to do with the choice of assumed observational noise and length scale. To obtain better results we have also tried GP hyper-parameter optimisation on the data (GP-ARD). The conjugate gradient optimiser managed to determine sensible hyper-parameters, thereby reducing the over-smoothing markedly. In contrast, the LACKI approach uncovered the nonlinearity well but overfitted to the noise in the data. Finally, our POKI-LC approach was able to smooth out the noise sufficiently well to offer a good fit to both the linear as well as to the non-linear trend in the target function. Being of comparable prediction accuracy as the solution found by GP-ARD, POKI's predictor was markedly less smooth than the GP's predictor with the chosen kernel and thereby, being able to fit the non-smooth parts of the target function better.

Exp.2 : We desired to gain an impression of the performance of our POKI methods with (i) increasing input space dimensionality and (ii) in the limit of increasing sample size. This time, for (ii), we chose uniformly distributed noise: $\nu(x) \overset{i.i.d.}{\sim} \text{Unif}(-0.5, 0.5)$. For an experiment with Gaussian



(a) Predictions.

Fig. 1. Exp.1. Predictions (magenta curve) of the target function $f : x \mapsto |-\cos(2\pi\, x_1)| + x_1$ (blue curve) on one-dimensional input space with various models trained on a random sample of 84 data points for some of our learning methods. The training data is plotted as cyan dots; for POKI we have plotted the conditioning data in green. Observe how the LACKI model overfits to the noisy data, while POKI manages to smooth out the noise and quite accurately predicts the ground-truth target function. Moreover, the GP with fixed hyper-parameters over-smoothes, while the GP with hyper-parameter optimisation (GP-ARD) manages to find hyper-parameters that lead to substantially better predictions.

noise, refer to the preprint version of this paper [15]. The results are depicted in Fig. 2 and Fig. 3, respectively. As performance metrics of interest, we considered the quality of the prediction and the computational effort required for training as well as for predicting with the trained models. The predictive accuracy was estimated by the empirical absolute prediction error means (cf. Eq. 1) estimated on the basis of a test set sample of 4000 inputs drawn i.i.d. from a uniform over the input space. The test sample was drawn independently from the training examples. Runtime measurements were based on code written in Julia 4.7 running on a single core of a 2.5 GHz i7 processor on a laptop with 16GB RAM. Our evaluation of the GPs was based on the implementation provided by the *GaussianProcesses.jl* library.

Examining the plots we note the following observations: Firstly, for all POKI methods, the prediction error seemed to vanish with increasing sample size. Secondly, the ARD methods were able to yield better prediction accuracy. We attribute this to their ability to uncover the inherently one-dimensional structure of the functional relationship. In comparison to the tested alternatives, POKI-ARD seemed to be more reliable in finding good parameters (and hence, to yield superior prediction accuracy) than its competitors. We attribute this to the use of the global Lipschitz optimiser. This benefit comes at the cost of increased training duration in the current implementation of the optimiser in the ARD case, when the dimensionality of the data increases. Addressing this issue is deferred to future work. In Exp.2 (ii) and (iii), it appeared that the Lipschitz optimisation-based hyper-parameter tuning seemed to be particularly beneficial for a small- to mid-range training data density (e.g. compare the prediction error plots of POKI-LC vs POKI-LC2 as well as POKI-ARD vs GP-ARD). Furthermore, in Exp.2 (i), it emerged that both GP-based predictors performed comparably poorly as the dimensionality of the input space grew (cf. Fig. 2).

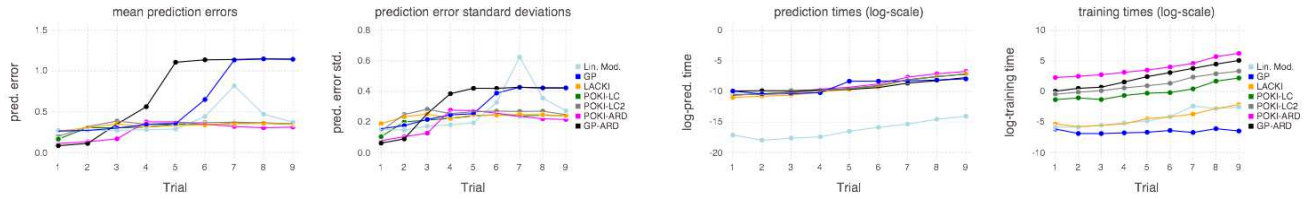
A. Conclusions

We developed POKI, a method for nonparametric regression that optimises parameters of a chosen pseudo-metric with the aim to maximise predictive performance. Our approach can be applied to automated determination of Lipschitz constants in nonlinear set membership methods thereby addressing an open problem in data-driven control in a more principled manner than previously proposed methods [13], [22]. In the long version of this paper [15], we have also presented additional examples where the automatically learned (hyper-) parameters were utilised for automated relevance determination and detected and leveraged periodicity in the data and where the Lipschitz optimisation approach yielded marked advantages over hyper-parameter tuning with a local optimiser.

We have found that our POKI methods work. They can compensate for noise in the data without the necessity to know the correct noise distributions. They are flexible to learn nonlinear, non-smooth functions. While the performance will not always be better than competing methods (such as GPs), an advantage of our approach is that the inference only involves basic computational steps that we would expect to be numerically robust and executable even on reduced instruction set micro-controllers. Furthermore, the particular mathematical properties of the Lipschitz interpolation rule allow us to easily determine Lipschitz constants of the empirical prediction errors. This in turn allows us to determine hyper-parameters by global Lipschitz optimisation, avoiding the pitfalls of local optima. Therefore, our approach can be more flexible and reliable to perform well in a large variety of problem domains and data sets without any manual tweaking. This is in contrast to many other methods (such as GPs or deep learning methods), where the predictive performance can be sensitive to a priori choices, e.g. of priors, hyper-parameters, optimisers and initialisations.

With the present choice of Lipschitz optimiser, problems involving high-dimensional hyper-parameters can render the determination of the approximately optimal hyper-parameter computationally intractable. Future work intends to explore the utilisation or development of alternative Lipschitz optimisation approaches that promise improved scalability (for a fixed error bound) in the dimensionality of the optimisation problem (e.g. [18]).

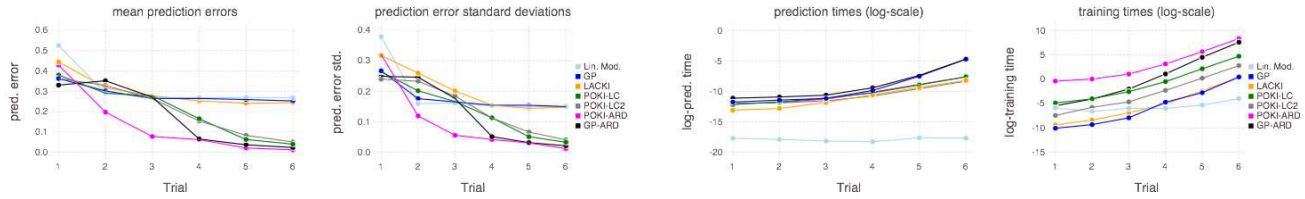
We observed that our approach can smooth out noise and hence, can yield accurate prediction of the ground truth (cf. e.g. Fig. 1(a) and Fig. 3) even in the presence



(a) Abs. pred. error statistics.

(b) Runtimes (log-sec.).

Fig. 2. Exp.2 (i). Fig. 2(a), left: Prediction error means ($\hat{\mathcal{E}}_1$) for the different trials. In trial i, the input space dimensionality was chosen to be $d = 2^i$. The training data size was set to a fixed value of 150 training examples during all trials. Fig. 3(b) depicts the logarithms of the pertaining records of runtimes (in seconds) for training the models (right) and the average prediction time for the test inputs (left). Not surprisingly, the training effort tends to increase with the number of parameters to be optimised.



(a) Abs. pred. error statistics.

(b) Runtimes (log-sec.).

Fig. 3. Exp. 2 (ii). Fig. 3(a) left: Prediction error means ($\hat{\mathcal{E}}_1$) for the different trials with increasing sample size. In all trials the input space dimensionality was chosen to be $d = 2$. In trial n, the sample size $|\mathcal{D}_n|$ was chosen to be $2^{d\,n}$. Fig. 3(b) depicts the logarithms of the pertaining records of the runtimes (in seconds) for training the models (right) and the average prediction time for the test inputs (left). Note how the prediction errors of most of the methods drops with increasing sample size. Successfully uncovering the low-dimensional structure, POKI-ARD tends to outperform or match the other methods in terms of $\hat{\mathcal{E}}_1$ error.

of additive observational noise. A theoretical challenge that remains to be investigated is the asymptotics of the true $\mathcal{L}_1$-prediction error in the limit of increasing number of noisy data points. We would hope that these results would allow us to derive probabilistic guarantees for a data-driven controller that combines our learning method with stochastic MPC approaches. This would provide an important extension to existing work of NSM-based MPC that had to rely on the knowledge of the Lipschitz constant [22], [14].

Finally, we would like to point out that Lipschitz interpolation rules (i.e. KI or NSM predictors) have a mathematical structure that made it particularly easy to calculate a Lipschitz bound on the empirical ℓ_1 prediction loss, facilitating global Lipschitz optimisation of the hyper-parameters. However, it might be worthwhile exploring in how far the Lipschitz optimisation approach can also be successfully employed in automated (hyper-) parameter tuning of other machine learning methods.

REFERENCES

[1] G. S. Watson, "Smooth regression analysis." *Sankhya: The Indian Journal of Statistics*, 1964.

[2] E. A. Nadaraya, "On estimating regression," *Theory of Probability and its Applications.*, 1964.

[3] W. S. Cleveland, "Robust locally weighted regression and smoothing scatterplots." *Journal of the American Statistical Association*, 1979.

[4] C. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[5] M. Deisenroth, C. E. Rasmussen, and J. Peters, "Gaussian process dynamic programming," *Neurocomputing*, 2009.

[6] M. Deisenroth and C. Rasmussen, "Pilco : A model-based and data-efficient approach to policy search," in *ICML*, 2011.

[7] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[8] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive processing*, 2011.

[9] M. P. Deisenroth, G. Neumann, and J. Peters, "A survey on policy search for robotics," *Foundations and Trends in Robotics*, 2013.

[10] A. Sukharev, "Optimal method of constructing best uniform approximation for functions of a certain class," *Comput. Math. and Math. Phys.*, 1978.

[11] Z. B. Zabinsky, R. L. Smith, and B. P. Kristinsdottir, "Optimal estimation of univariate black-box Lipschitz functions with upper and lower bounds." *Computers and Operations Research*, 2003.

[12] G. Beliakov, "Interpolation of Lipschitz functions," *Journal of Computational and Applied Mathematics*, 2006.

[13] M. Milanese and C. Novara, "Set membership identification of nonlinear systems," *Automatica*, 2004.

[14] J.-P. Calliess, "Conservative decision-making and inference in uncertain dynamical systems," Ph.D. dissertation, University of Oxford, 2014.

[15] J. Calliess, "Lipschitz Optimisation for Lipschitz Interpolation," *Arxiv preprint*, 2017.

[16] R. G. Strongin, "On the convergence of an algorithm for finding a global extremum," *Engineering in Cybernetics*, 1973.

[17] B. Shubert, "A sequential method seeking the global maximum of a function," *SIAM J. on Numerical Analysis*, vol. 9, 1972.

[18] B. P. Zhang, "Topics in lipschitz global optimisation." Ph.D. dissertation, University of Canterbury, 1995.

[19] R. Munos, "From Bandits to Monte-Carlo Tree Search: The Optimistic Principle Applied to Optimization and Planning ," *Foundations and Trends in Machine Learning*, 2014.

[20] N. Weaver, *Lipschitz Algebras*. World Scientific, 1999.

[21] J. Calliess, "Lazily Adapted Constant Kinky Inference for Nonparametric Regression and Model-Reference Adaptive Control," *Arxiv preprint arXiv:1701.00178*, 2016.

[22] M. Canale, L. Fagiano, and M. C. Signorile, "Nonlinear model predictive control from data: a set membership approach," *Int. J. Robust Nonlinear Control*, 2014.