Semi-described and semi-supervised learning with Gaussian processes

Andreas Damianou
Dept. of Computer Science & SITraN
The University of Sheffield
Sheffield, UK

Neil D. Lawrence
Dept. of Computer Science & SITraN
The University of Sheffield
Sheffield, UK

Abstract

Propagating input uncertainty through non-linear Gaussian process (GP) mappings is intractable. This hinders the task of training GPs using uncertain and partially observed inputs. In this paper we refer to this task as "semi-described learning". We then introduce a GP framework that solves both, the semi-described and the semi-supervised learning problems (where missing values occur in the *outputs*). Auto-regressive state space simulation is also recognised as a special case of semi-described learning. To achieve our goal we develop variational methods for handling semi-described inputs in GPs, and couple them with algorithms that allow for imputing the missing values while treating the uncertainty in a principled, Bayesian manner. Extensive experiments on simulated and real-world data study the problems of iterative forecasting and regression/classification with missing values. The results suggest that the principled propagation of uncertainty stemming from our framework can significantly improve performance in these tasks.

1 INTRODUCTION

In many real-world applications missing values can occur in the data, for example when measurements come from unreliable sensors. Correctly accounting for the partially observed instances is important in order to exploit all available information and increase the strength of the inference model. The focus of this paper is on Gaussian process (GP) models that allow for Bayesian, non-parametric inference.

When the missing values occur in the outputs, the corresponding learning task is known as *semi-supervised learning*. For example, consider the task of learning to classify images where the labelled set is much smaller than the total set. Bootstrapping is a potential solution to this problem [Rosenberg et al., 2005], according to which a model

trained on fully observed data imputes the missing outputs. Previous work in semi-supervised GP learning involved the cluster assumption [Lawrence and Jordan, 2005] for classification. Here we consider an approach which uses the manifold assumption [Chapelle et al., 2006; Kingma et al., 2014] which assumes that the observed, complex data are really generated by a compressed, less-noisy latent space.

The other often encountered missing data problem has to do with unobserved *input* features (e.g. missing pixels in input images). In statistics, a popular approach is to impute missing inputs using a combination of different educated guesses [Rubin, 2004]. In machine learning, Ghahramani and Jordan [1994] learn the joint density of the input and output data and integrate over the missing values. For Gaussian process models the missing input case has received only little attention, due to the challenge of propagating the input uncertainty through the non-linear GP mapping. In this paper we introduce the notion of *semi-described learning* to generalise this scenario. Specifically, we define semi-described learning to be the task of learning from inputs that can have missing or uncertain values. Our approach to dealing with missing inputs in semi-described GP learning is, algorithmically, closer to data imputation methods. However, in contrast to past approaches, the missing values are imputed in a fully probabilistic manner by considering explicit distributions in the input space.

Our aim in this paper is to develop a general framework that solves the semi-supervised and semi-described GP learning. We also consider the related *forecasting regression* problem, which is seen as a pipeline where predictions are obtained iteratively in an auto-regressive manner, while propagating the uncertainty across the predictive sequence, as in [Girard et al., 2003; Quiñonero-Candela et al., 2003]. Here, we cast the auto-regressive GP learning as a particular type of semi-described learning. We seek to solve all tasks within a single coherent framework that preserves the fully Bayesian property of the GP methodology.

To achieve our goals we need three methodological tools. Firstly, we need approximations allowing us to consider and communicate uncertainty between the inputs and

the outputs of the non-linear GP model. For this, we build on the variational approach of Titsias and Lawrence [2010] which allows for approximately propagating densities throughout the nodes of GP-based directed graphical models. The resulting representation is particularly advantageous, because the whole input domain is now coherently associated with posterior distributions. We can then sample from the input space in a principled manner so as to populate small initial labelled sets in semi-supervised learning scenarios. In that way, we avoid heuristic self-training methods [Rosenberg et al., 2005] that rely on bootstrapping and present problems due to over-confidence. Previously suggested approaches for modelling input uncertainty in GPs also lack the feature of considering an explicit input distribution for both training and test instances. Specifically, [Girard et al., 2003; Quiñonero-Candela et al., 2003] consider the case of input uncertainty *only at test time*. Propagating the test input uncertainty through a non-linear GP results in a non-Gaussian predictive density, but Girard et al. [2003]; Quiñonero-Candela et al. [2003]; Quiñonero-Candela [2004] rely on moment matching to obtain the predictive mean and covariance. On the other hand, Oakley and O'Hagan [2002] do not derive analytic expressions but, rather, develop a scheme based on simulations. McHutchon and Rasmussen [2011] rely on local approximations inside the latent mapping function, rather than modelling the approximate posterior densities directly. Dallaire et al. [2009] do not propagate the uncertainty of the inputs all the way through the GP mapping but, rather, amend the kernel computations to account for the input uncertainty. [Quinonero-Cañdela and Roweis, 2003] can be seen as a special case of our developed framework, when the data imputation is performed using a standard GP-LVM [Lawrence, 2006]. Another advantage of our framework is that it allows us to consider different levels of input uncertainty per point and per dimension without, in principle, increasing the danger of under/overfitting, since input uncertainty is modelled through a set of *variational* rather than model parameters.

The second methodological tool needed to achieve our goals has to do with the need to incorporate partial or uncertain observations into the variational framework. For this, we develop a *variational constraint* mechanism which constrains the distribution of the input space given the observed noisy values. This approach is fast, and the whole framework can be incorporated into a parallel inference algorithm [Gal et al., 2014; Dai et al., 2014]. In contrast, Damianou et al. [2011] consider a separate process for modelling the input distribution. However, that approach cannot easily be extended for the data imputation purposes that concern us, since we cannot consider different uncertainty levels per input and per dimension and, additionally, computation scales cubicly with the number of datapoints, even within sparse GP frameworks. The constraints framework that we propose is interesting not only as an inference

tool but also as a modelling approach: if the inputs are constrained with the outputs, then we obtain the Bayesian version of the back-constraints framework of Lawrence and Quiñonero Candela [2006] and Ek et al. [2008]. However, in contrast to these approaches, the constraint defined here is a *variational* one, and operates upon a distribution, rather than single points. Zhu et al. [2012] also follow the idea of constraining the posterior distribution with rich side information, albeit for a completely different application. In contrast, Osborne and Roberts [2007] handle partially missing sensor inputs by modelling correlations in the input space through special covariance functions.

Thirdly, the variational methods developed here need to be encapsulated into algorithms that perform data imputation while correctly accounting for the introduced uncertainty. We develop such algorithms after showing how the considered applications can be cast as learning pipelines that rely on correct propagation of uncertainty between each stage.

In summary, our contributions in this paper are the following; firstly, by building on the Bayesian GP-LVM [Titsias and Lawrence, 2010] and developing a variational constraint mechanism we demonstrate how uncertain GP inputs can be explicitly represented as distributions in both training and test time. Secondly, we couple our variational methodology with algorithms that allow us to solve problems associated with partial or uncertain observations: semi-supervised learning, auto-regressive iterative forecasting and, finally, a newly studied type of GP learning which we refer to as "semi-described" learning. We solve these applications within a single framework, allowing for handling the uncertainty in semi-supervised and semi-described problems in a coherent way. The software accompanying this paper can be found at: http://git.io/A3TN. This paper extends our previous workshop paper [Damianou and Lawrence, 2014].

2 UNCERTAIN INPUTS REFORMULATION OF GP MODELS

Assume a dataset of input–output pairs stored by rows in matrices $\mathbf{X} \in \Re^{n \times q}$ and $\mathbf{Y} \in \Re^{n \times p}$ respectively. Throughout this paper we will denote rows of the above matrices as $\{\mathbf{y}_{i,:}, \mathbf{x}_{i,:}\}$ and columns (dimensions) as $\{\mathbf{y}_j, \mathbf{x}_j\}$, while single elements (e.g. $y_{i,j}$) will be denoted with a double subscript. We first outline the standard GP formulation, where all variables are fully observed. By assuming that outputs are corrupted by zero-mean Gaussian noise, denoted by ϵ_f, we obtain the following generative model:

$$y_{i,j} = \mathbf{f}_j(\mathbf{x}_{i,:}) + (\epsilon_f)_{i,j}, \quad (\epsilon_f)_{i,j} \sim \mathcal{N}\left(0, \beta^{-1}\right). \quad (1)$$

We place GP priors on the mapping $\mathbf{f}$, so that the function instantiations $\mathbf{F} = \{\mathbf{f}_j\}_{j=1}^{p}$ follow a Gaussian distribution $p(\mathbf{f}_j|\mathbf{X}) = \mathcal{N}\left(\mathbf{f}_j|\mathbf{0}, \mathbf{K}\right)$, where $\mathbf{K}$ is the covariance matrix obtained by evaluating the GP covariance function k_f on

the inputs $\mathbf{X}$. Therefore, the model likelihood $p(\mathbf{Y}|\mathbf{X})$ is:

$$\int_{\mathbf{F}} p(\mathbf{Y}|\mathbf{F})p(\mathbf{F}|\mathbf{X}) = \prod_{j=1}^{p} \mathcal{N}\left(\mathbf{y}_j|\mathbf{0}, \mathbf{K} + \beta^{-1}\mathbf{I}\right). \quad (2)$$

In the other end of the spectrum is the GP-LVM [Lawrence, 2006], where the inputs are fully unobserved (i.e. *latent*). This corresponds to the unsupervised GP setting. In the absence of observed inputs, the likelihood $p(\mathbf{Y}|\mathbf{X})$ takes the same form as in equation (2) but the inputs $\mathbf{X}$ now need to be recovered from the outputs $\mathbf{Y}$ through maximum likelihood. The Bayesian GP-LVM proceeds by additionally placing a Gaussian prior on the latent space, $p(\mathbf{X}) = \prod_{i=1}^{n} \mathcal{N}(\mathbf{x}_{i,:}|\mathbf{0}, \mathbf{I})$, and approximately integrating it out by constructing a variational lower bound $\mathcal{F}$, where

$$\mathcal{F} \leq \log p(\mathbf{Y}) = \log \int_{\mathbf{X}} p(\mathbf{Y}|\mathbf{X})p(\mathbf{X}),$$

and by introducing a variational distribution

$$q(\mathbf{X}) = \prod_{i=1}^{n} q(\mathbf{x}_{i,:}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{x}_{i,:}|\boldsymbol{\mu}_{i,:}, \mathbf{S}_{i,:}\right),$$

where $\mathbf{S}_{i,:}$ is a diagonal matrix, so that $\boldsymbol{\mu}_{i,:}, \mathrm{diag}(\mathbf{S}_{i,:}) \in \Re^q$. We can derive an expression for this variational bound,

$$\mathcal{F} = \langle \log p(\mathbf{Y}|\mathbf{X}) \rangle_{q(\mathbf{X})} - \mathrm{KL}\left(q(\mathbf{X}) \| p(\mathbf{X})\right), \quad (3)$$

where $\langle \cdot \rangle_{q(\mathbf{X})}$ denotes an expectation with respect to $q(\mathbf{X})$. Since $\mathbf{X}$ appears non-linearly inside $p(\mathbf{Y}|\mathbf{X})$ (in the inverse of the covariance matrix $\mathbf{K} + \beta^{-1}\mathbf{I}$), the first term of the above variational bound is intractable. However, we can follow [Titsias and Lawrence, 2010] to approximate the intractable expectation analytically.

In this paper we wish to define a general framework that operates in the whole range of the two aforementioned extrema, i.e. the fully observed and fully unobserved inputs case. The first step to obtaining such a framework is to allow for uncertainty in the inputs. We assume that the inputs $\mathbf{X}$ are not observed directly but, rather, we only have access to their noisy versions $\{\mathbf{z}_{i,:}\}_{i=1}^{n} = \mathbf{Z} \in \Re^{n \times q}$. The relationship between the noisy and true inputs is given by assuming Gaussian noise:

$$\mathbf{x}_{i,:} = \mathbf{z}_{i,:} + (\boldsymbol{\epsilon}_x)_{i,:}, \quad (\boldsymbol{\epsilon}_x)_{i,:} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_x\right), \quad (4)$$

so that $p(\mathbf{X}|\mathbf{Z}) = \prod_{i=1}^{n} \mathcal{N}(\mathbf{x}_{i,:}|\mathbf{z}_{i,:}, \boldsymbol{\Sigma}_x)$. Obviously, when this distribution collapses to a delta function we recover the standard GP case, and when $\mathbf{Z}$ is not given we recover the GP-LVM. The problem with the modelling assumption of equation (4) is that now we cannot use equation (1), since the inputs are not available. On the other hand, if we replace $\mathbf{x}_{i,:}$ in that equation with $\mathbf{z}_{i,:}$, then we effectively ignore the input noise. McHutchon and Rasmussen [2011] proceed by combining equations (1) and (4) to obtain the GP mapping $\mathbf{f}_j(\mathbf{x}_{i,:} - (\boldsymbol{\epsilon}_x)_{i,:})$ which is then

treated using local approximations. However, our aim in this paper is to consider an explicit input distribution. One way to achieve this is to treat the unobserved true inputs as latent variables to be estimated from the marginal likelihood $p(\mathbf{Y}|\mathbf{Z}) = \int_{\mathbf{X}} p(\mathbf{Y}|\mathbf{X})p(\mathbf{X}|\mathbf{Z})$. Following Damianou et al. [2011] we can obtain a variational lower bound $\mathcal{F} \leq \log p(\mathbf{Y}|\mathbf{Z})$, with:

$$\mathcal{F} = \langle \log p(\mathbf{Y}|\mathbf{X}) \rangle_{q(\mathbf{X})} - \mathrm{KL}\left(q(\mathbf{X}) \| p(\mathbf{X}|\mathbf{Z})\right). \quad (5)$$

This formulation corresponds to the graphical model of Figure 1(a). However, with this approach one needs to additionally estimate the noise parameters $\boldsymbol{\Sigma}_x$, which might be challenging given their large number and their interplay with the variational noise parameters $\{\mathbf{S}_{i,:}\}_{i=1}^{n}$. Therefore we considered an alternative solution which we found to result in better performance.
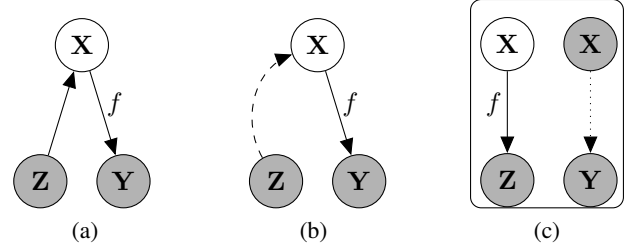

Figure 1: Incorporating uncertain inputs $\mathbf{Z}$ in GPs through an intermediate input space $\mathbf{X}$ by considering: (a) a Gaussian prior on $\mathbf{X}$, centered on $\mathbf{Z}$ and (b) a variational constraint (dashed line) on the approximate posterior. Figure (c) represents our two-stage approach to dealing with missing outputs for classification, where the dotted line represents a discriminative mapping.

2.1 VARIATIONAL CONSTRAINT

An alternative way of relating the true with the noisy inputs can be obtained by focusing on the *posterior* rather than the prior distribution. To start with, we re-express the variational lower bound of equation (5) as:

$$\log p(\mathbf{Y}|\mathbf{Z}) \geq \int_{\mathbf{X}} q(\mathbf{X}) \log \frac{p(\mathbf{Y}|\mathbf{Z})p(\mathbf{X}|\mathbf{Y}, \mathbf{Z})}{q(\mathbf{X})} = \mathcal{F}$$

from where we break the logarithm to obtain:

$$\mathcal{F} = \log p(\mathbf{Y}|\mathbf{Z}) - \mathrm{KL}\left(q(\mathbf{X}) \| p(\mathbf{X}|\mathbf{Y}, \mathbf{Z})\right).$$

We see that the lower bound becomes exact when the variational distribution $q(\mathbf{X})$ matches the true posterior distribution of the noise-free latent inputs given the observed inputs and outputs. To allow for this approximation we introduce a simple variational constraint which operates on the factorised distribution, which is now written as $q(\mathbf{X}|\mathbf{Z})$ to highlight its dependency on $\mathbf{Z}$. In the simplest case

where all inputs are observed but uncertain, the constraint just consists of replacing the variational means $\boldsymbol{\mu}_{i,:}$ of each factor $q(\mathbf{x}_{i,:})$ with the corresponding observed input $\mathbf{z}_{i,:}$. The variational parameters $\mathbf{S}_{i,:}$ then account for the uncertainty. Similarly to the back-constraint of Lawrence and Quiñonero Candela [2006]; Ek et al. [2008], our variational constraint does not constitute a probabilistic mapping. However, it allows us to encode the input noise directly in the approximate posterior without having to specify additional noise parameters or sacrifice scalability. Next, we elaborate on the exact form of the constraint.

In the general case, namely having inputs that are only partially observed, we can define a similar constraint which specifies a variational distribution as a mix of Gaussian and Dirac delta distributions. Notationally we consider data to be split into fully and partially observed subsets, e.g. $\mathbf{Z} = (\mathbf{Z}^{\mathcal{o}}, \mathbf{Z}^{u})$, where $\mathcal{o}$ and u denote fully and partially observed sets respectively. The features missing in $\mathbf{Z}^{u}$ can appear in different dimension(s) for each individual point $\mathbf{z}_{i,:}^{u}$, but for notational clarity u will index rows containing at least one missing dimension. In this case, the variational distribution is constrained to have the form

$$q(\mathbf{X}|\mathbf{Z}, \{\mathcal{o}, u\}) = q(\mathbf{X}^{\mathcal{o}}|\mathbf{Z}^{\mathcal{o}})\, q(\mathbf{X}^{u}|\mathbf{Z}^{u})$$
$$= \prod_{i \in \mathcal{o}} \mathcal{N}\left(\mathbf{x}_{i,:}^{\mathcal{o}}|\mathbf{z}_{i,:}^{\mathcal{o}}, \varepsilon\mathbf{I}\right) \prod_{i \in u} \mathcal{N}\left(\mathbf{x}_{i,:}^{u}|\boldsymbol{\mu}_{i,:}^{u}, \mathbf{S}_{i,:}^{u}\right), \quad (6)$$

where $\varepsilon \to 0$, so that the corresponding distributions approximate a Dirac delta. Notice that for a partially observed row $\mathbf{z}_{i,:}^{u}$, we can still replace an observed dimension j with its corresponding observation in the second set of factors of equation (6), i.e. $\mu_{i,j}^{u} = z_{i,j}^{u}$, so $q(\mathbf{X}^{u}|\mathbf{Z}^{u}) \neq q(\mathbf{X}^{u})$. Given the above, as well as a spherical Gaussian prior for $p(\mathbf{X})$, the required intractable density $\log p(\mathbf{Y}|\mathbf{Z})$ is approximated with a variational lower bound:

$$\mathcal{F} = \langle \log p(\mathbf{Y}|\mathbf{X}) \rangle_{q(\mathbf{X}|\mathbf{Z})} - \mathrm{KL}\left(q(\mathbf{X}|\mathbf{Z}) \,\|\, p(\mathbf{X})\right), \quad (7)$$

where for clarity we dropped the dependency on $\{\mathcal{o}, u\}$ from our expressions. Since the Dirac functions are approximated with sharply peaked Gaussians inside the posterior $q(\mathbf{X}|\mathbf{Z})$, the above variational bound can be computed in the same manner as the Bayesian GP-LVM bound of equation (3). Specifically, the KL term is tractable, since it only involves Gaussians.

As for the first term of equation (7), we follow the Bayesian GP-LVM methodology and we augment the probability space with m extra samples $\mathbf{U} = \{\mathbf{u}_i\}_{i=1}^{m}$ of the latent function f evaluated at a set of pseudo-inputs (known as "inducing points") $\mathbf{X}_u$, so that $\mathbf{U} \in \Re^{m \times p}$ and $\mathbf{X}_u \in \Re^{m \times q}$. Due to the consistency of GPs, $p(\mathbf{U}|\mathbf{X}_u)$ is a Gaussian distribution. From now on we omit dependence on $\mathbf{X}_u$ from our expressions. The likelihood then becomes:

$$p(\mathbf{Y}, \mathbf{F}, \mathbf{U}|\mathbf{X}) = p(\mathbf{Y}|\mathbf{F})p(\mathbf{F}|\mathbf{U}, \mathbf{X})p(\mathbf{U}).$$

Then, the marginal $p(\mathbf{Y}|\mathbf{X})$ can be obtained from Jensen's inequality after introducing a variational distribution $q(\mathbf{F}, \mathbf{U})$, so that $\hat{\mathcal{F}} \leq \log p(\mathbf{Y}|\mathbf{X})$, where:

$$\hat{\mathcal{F}} = \int_{\mathbf{F}, \mathbf{U}} q(\mathbf{F}, \mathbf{U}) \log \frac{p(\mathbf{Y}|\mathbf{F})p(\mathbf{F}|\mathbf{U}, \mathbf{X})p(\mathbf{U})}{q(\mathbf{F}, \mathbf{U})}. \quad (8)$$

Now the fist term of equation (7) is approxmated as $\langle p(\mathbf{Y}|\mathbf{X}) \rangle_{q(\mathbf{X}|\mathbf{Z})} \geq \langle \hat{\mathcal{F}} \rangle_{q(\mathbf{X}|\mathbf{Z})}$. However, this approximation is still intractable, since the problematic term $p(\mathbf{F}|\mathbf{U}, \mathbf{X})$ still appears inside $\hat{\mathcal{F}}$ and contains $\mathbf{X}$ in the inverse of the covariance matrix, thus rendering the expectation intractable. The trick of Titsias and Lawrence [2010] is to define a variational distribution of the form:

$$q(\mathbf{F}, \mathbf{U}) = p(\mathbf{F}|\mathbf{U}, \mathbf{X})q(\mathbf{U}). \quad (9)$$

Replacing equation (9) inside the bound of equation (8) results in the cancellation of $p(\mathbf{F}|\mathbf{U}, \mathbf{X})$, leaving us with a tractable (partial) bound, which takes the form:

$$\langle p(\mathbf{Y}|\mathbf{X}) \rangle_{q(\mathbf{X}|\mathbf{Z})} \geq \langle \hat{\mathcal{F}} \rangle_{q(\mathbf{X}|\mathbf{Z})} = -\mathrm{KL}\left(q(\mathbf{U}) \,\|\, p(\mathbf{U})\right)$$
$$+ \int_{\mathbf{X}, \mathbf{U}} \left[q(\mathbf{X}|\mathbf{Z})q(\mathbf{U}) \int_{\mathbf{F}} p(\mathbf{F}|\mathbf{U}, \mathbf{X}) \log p(\mathbf{Y}|\mathbf{F}) \right] \quad (10)$$

The augmentation trick decouples the latent function values given the inducing points, so that any uncertainty in the inputs can be propagated through the nested integral. After this operation, the inducing outputs $\mathbf{U}$ can be marginalised out. Therefore, the above integral is analytically tractable, since the nested integral is tractable and results in a Gaussian where $\mathbf{X}$ no longer appears in the inverse of the covariance matrix. The final lower bound to use as an objective function is thus obtained by using the partial bound of eq. (10) in place of the first term of equation (7), thus obtaining a new, final bound (more details in the Appendix):

$$\mathcal{F}_2 = \langle \hat{\mathcal{F}} \rangle_{q(\mathbf{X}|\mathbf{Z})} - \mathrm{KL}\left(q(\mathbf{X}|\mathbf{Z}) \,\|\, p(\mathbf{X})\right). \quad (11)$$

To summarise, the variational methodology seeks to approximate the true posterior with a variational distribution $q(\mathbf{F}, \mathbf{U}, \mathbf{X}) = q(\mathbf{F})q(\mathbf{U})q(\mathbf{X})$. To achieve this, $q(\mathbf{F})$ is constrained to take the exact form $p(\mathbf{F}|\mathbf{U}, \mathbf{X})$. This term is then "eliminated", giving us tractability, but its effect is re-introduced through the variational distribution (in the nested integral of eq. (10)). Contrast this with the variational constraint on $q(\mathbf{X})$: that approximate posterior factor is constrained according to $\mathbf{Z}$, so that the effect of $\mathbf{Z}$ is considered only through the $q(\mathbf{X}|\mathbf{Z})$ (eq. (11)). The above comparison gives insight in the conceptual similarity of the variational approach followed to obtain tractability and the one followed for handling partially observed inputs.

The variationally constrained model is shown in fig. 1(b). The total set of parameters to be optimised in the objective function $\mathcal{F}_2$ of equation (11) (e.g. using a gradient-based optimiser) are the model parameters $(\boldsymbol{\theta}_f, \beta)$, where $\boldsymbol{\theta}_f$ denotes the hyper-parameters of the covariance function k_f,

and the variational parameters $(\mathbf{X}_u, \{\boldsymbol{\mu}_{i,:}^{\mathcal{u}}, \mathbf{S}_{i,:}^{\mathcal{u}}\}_{i \in \mathcal{u}})$ $(q(\mathbf{U})$ can be optimally eliminated, see Appendix). Depending on the application and corresponding learning algorithm, certain dimensions of $\{\boldsymbol{\mu}_{i,:}^{\mathcal{u}}, \mathbf{S}_{i,:}^{\mathcal{u}}\}$ can be treated as observed. Such algorithms are discussed in the following sections.

3 GP LEARNING WITH MISSING VALUES

We formulate both the semi-described and semi-supervised learning as particular instances of learning a mapping function where the inputs are associated with uncertainty. In both cases, we devise a two-step strategy based on our uncertain inputs GP framework, which allows to efficiently take into account the partial information in the given datasets to improve the predictive performance. For brevity, we refer to the framework described in the previous section as a *variationally constrained GP*, from where a semi-described, an auto-regressive and a semi-supervised GP approach are obtained as special cases, given the algorithms that will be explained in this section.

3.1 SEMI-DESCRIBED LEARNING

We assume a set of observed outputs $\mathbf{Y}$ that correspond to fully observed inputs $\mathbf{Z}^{\mathcal{O}}$ and partially observed inputs $\mathbf{Z}^{\mathcal{u}}$, so that $\mathbf{Z} = (\mathbf{Z}^{\mathcal{O}}, \mathbf{Z}^{\mathcal{u}})$. To make the correspondence clearer, we also split the observed *outputs* according to the sets $\{\mathcal{O}, \mathcal{u}\}$, so that $\mathbf{Y} = (\mathbf{Y}^{\mathcal{O}}, \mathbf{Y}^{\mathcal{u}})$, but note that both output sets are fully observed. We are then interested in learning a regression function from $\mathbf{Z}$ to $\mathbf{Y}$ by using all available information. Since in the variationally constrained GP the inputs are replaced by distributions $q(\mathbf{X}^{\mathcal{O}}|\mathbf{Z}^{\mathcal{O}})$ and $q(\mathbf{X}^{\mathcal{u}}|\mathbf{Z}^{\mathcal{u}})$, the uncertainty over $\mathbf{Z}^{\mathcal{u}}$ can be taken into account naturally through this variational distribution. In this context, we formulate a data imputation-based approach which is inspired by self-training methods; nevertheless, it is more principled in the handling of uncertainty.

Specifically, the algorithm has two stages; in the first step, we use the fully observed data subset $(\mathbf{Z}^{\mathcal{O}}, \mathbf{Y}^{\mathcal{O}})$ to train an initial variationally constrained GP model which encapsulates the sharply peaked variational distribution $q(\mathbf{X}^{\mathcal{O}}|\mathbf{Z}^{\mathcal{O}})$ given in equation (6). Given this model, we can then use $\mathbf{Y}^{\mathcal{u}}$ to estimate the predictive posterior[1] $q(\mathbf{X}^{\mathcal{u}}|\mathbf{Z}^{\mathcal{u}})$ in the missing locations of $\mathbf{Z}^{\mathcal{u}}$ (for the observed locations we match the mean with the observations in a sharply peaked marginal, as for $\mathbf{Z}^{\mathcal{O}}$). Essentially, we replace the missing locations of the variational means $\boldsymbol{\mu}_{i,:}^{\mathcal{u}}$ and variances $\mathbf{S}_i^{\mathcal{u}}$ of $q(\mathbf{X}^{\mathcal{u}}|\mathbf{Z}^{\mathcal{u}})$ with the predictive mean and variance obtained through the "self-training" step. This selection for $\{\boldsymbol{\mu}_{i,:}^{\mathcal{u}}, \mathbf{S}_i^{\mathcal{u}}\}$ constitutes nevertheless only an initialisation. In

the next step, these parameters are further optimised together with the fully observed data. Specifically, after initializing $q(\mathbf{X}|\mathbf{Z}) = q(\mathbf{X}^{\mathcal{O}}, \mathbf{X}^{\mathcal{u}}|\mathbf{Z})$ as explained in step 1, we proceed to train a variationally constrained GP model on the full (extended) training set $((\mathbf{Z}^{\mathcal{O}}, \mathbf{Z}^{\mathcal{u}}), (\mathbf{Y}^{\mathcal{O}}, \mathbf{Y}^{\mathcal{u}}))$, which contains fully and partially observed inputs.

Algorithm 1 outlines the approach in more detail. This formulation defines a *semi-described GP* approach which naturally incorporates fully and partially observed examples by communicating the uncertainty throughout the relevant parts of the model in a principled way. Indeed, the predictive uncertainty obtained when imputing missing values in the first step of the pipeline is incorporated as input uncertainty in the second step of the pipeline. In extreme cases resulting in very non-confident predictions, for example in the presence of outliers, the corresponding locations will simply be ignored automatically due to the large uncertainty. This mechanism, together with the subsequent optimisation of the parameters of $q(\mathbf{X}^{\mathcal{u}}|\mathbf{Z}^{\mathcal{u}})$ in stage 2, guards against reinforcing bad predictions when imputing missing values based on a smaller training set. The model includes GP regression and the GP-LVM as special cases. In particular, in the limit of having no observed values our semi-described GP is equivalent to the GP-LVM and when there are no missing values it is equivalent to GP regression.

There are some similarities to traditional self-training [Rosenberg et al., 2005], but as there are no straightforward mechanisms to propagate uncertainty in that domain, they typically rely on boot-strapping followed by thresholding "bad" samples to prevent model over-confidence. In our framework, the predictions made by the initial model only constitute initialisations which are later optimised along with model parameters and, hence, we refer to this step as *partial* self-training. Further, the predictive uncertainty is not used as a hard measure of discarding unconfident predictions; instead, we allow all values to contribute according to an optimised uncertainty measure, that is, the input variances $\mathbf{S}_i$. Therefore, the way in which uncertainty is handled makes the self-training part of our algorithm principled compared to many bootstrap-based approaches.

DEMONSTRATION

We considered simulated and real-world data to demonstrate our semi-described GP algorithm. The simulated data were created by sampling inputs $\mathbf{Z}$ from a GP (which was unknown to the competing models) and then giving these samples as input to another unknown GP, to obtain corresponding outputs $\mathbf{Y}$. For the real-world data demonstration we considered a motion capture dataset taken from subject 35 in the CMU motion capture database. We selected a subset of walk and run motions of a human body represented as a set of 59 joint locations. We formulated a regression problem where the first 20 dimensions of the original data are used as targets and the remaining 39 as

---

[1] The predictive posterior for test data $\mathbf{Y}_*$ is obtained by maximising a variational lower bound similar to the training one (eq. (11)), but $\mathbf{X}$ and $\mathbf{Y}$ are now replaced with $(\mathbf{X}, \mathbf{X}_*)$ and $(\mathbf{Y}, \mathbf{Y}_*)$.

Algorithm 1 Semi-described learning with uncertain input GPs.

1: *Given*: Fully and partially observed inputs, $\mathbf{Z}^{\mathcal{O}}$ and $\mathbf{Z}^{\mathcal{U}}$ respectively, corresponding to fully observed outputs $\mathbf{Y}^{\mathcal{O}}$ and $\mathbf{Y}^{\mathcal{U}}$.

2: Construct $q(\mathbf{X}^{\mathcal{O}}|\mathbf{Z}^{\mathcal{O}}) = \prod_{i=1}^{n} \mathcal{N}\left(\mathbf{x}_{i,:}^{\mathcal{O}}|\mathbf{z}_{i,:}^{\mathcal{O}}, \varepsilon \mathbf{I}\right)$, where: $\varepsilon \to 0$

3: Fix $q(\mathbf{X}^{\mathcal{O}}|\mathbf{Z}^{\mathcal{O}})$ in the optimiser   *# (i.e. $q(\mathbf{X}^{\mathcal{O}}|\mathbf{Z}^{\mathcal{O}})$ has no free parameters)*

4: Train a variationally constrained GP model $\mathcal{M}^{\mathcal{O}}$ with inputs $q(\mathbf{X}^{\mathcal{O}}|\mathbf{Z}^{\mathcal{O}})$ and outputs $\mathbf{Y}^{\mathcal{O}}$

5: **for** $i = 1, \cdots, |\mathbf{Y}^{\mathcal{U}}|$ **do**

6:     Predict the distribution $\mathcal{N}\left(\mathbf{x}_{i,:}^{\mathcal{U}}|\hat{\boldsymbol{\mu}}_{i,:}^{\mathcal{U}}, \hat{\mathbf{S}}_{i}^{\mathcal{U}}\right) \approx p(\mathbf{x}_{i,:}^{\mathcal{U}}|\mathbf{y}_{i,:}^{\mathcal{U}}, \mathcal{M}^{\mathcal{O}})$ from the approximate posterior of model $\mathcal{M}^{\mathcal{O}}$.

7:     Initialise parameters $\{\boldsymbol{\mu}_{i,:}^{\mathcal{U}}, \mathbf{S}_{i}^{\mathcal{U}}\}$ of $q(\mathbf{x}_{i,:}^{\mathcal{U}}|\mathbf{z}_{i,:}^{\mathcal{U}}) = \mathcal{N}\left(\mathbf{x}_{i,:}^{\mathcal{U}}|\boldsymbol{\mu}_{i,:}^{\mathcal{U}}, \mathbf{S}_{i}^{\mathcal{U}}\right)$ as follows:

8:     **for** $j = 1, \cdots, q$ **do**

9:         **if** $z_{i,j}^{\mathcal{U}}$ is observed **then**

10:            $\mu_{i,j}^{\mathcal{U}} = z_{i,j}^{\mathcal{U}}$ and $(\mathbf{S}_{i}^{\mathcal{U}})_{j,j} = \varepsilon$, where: $\varepsilon \to 0$

11:            Fix $\mu_{i,j}^{\mathcal{U}}, (\mathbf{S}_{i}^{\mathcal{U}})_{j,j}$ in the optimiser   *# (i.e. they don't constitute parameters)*

12:         **else**

13:            $\mu_{i,j}^{\mathcal{U}} = \hat{\mu}_{i,j}^{\mathcal{U}}$ and $(\mathbf{S}_{i}^{\mathcal{U}})_{j,j} = (\hat{\mathbf{S}}_{i}^{\mathcal{U}})_{j,j}$

14: Train model $\mathcal{M}^{\mathcal{O},\mathcal{U}}$ with inputs $q(\mathbf{X}^{\{\mathcal{O},\mathcal{U}\}}|\mathbf{Z}^{\{\mathcal{O},\mathcal{U}\}})$ and outputs $(\mathbf{Y}^{\mathcal{O}}, \mathbf{Y}^{\mathcal{U}})$. The input distribution $q(\mathbf{X}^{\{\mathcal{O},\mathcal{U}\}}|\mathbf{Z}^{\{\mathcal{O},\mathcal{U}\}}) = q(\mathbf{X}^{\mathcal{O}}|\mathbf{Z}^{\mathcal{O}})q(\mathbf{X}^{\mathcal{U}}|\mathbf{Z}^{\mathcal{U}})$ is constructed in steps 2, 5-13 and further optimised in the non-fixed locations.

15: Model $\mathcal{M}^{\mathcal{O},\mathcal{U}}$ now constitutes the semi-described GP and can be used for all required prediction tasks.

---

inputs. That is, given a partial joint representation of the human body, the task is to infer the rest of the representation. For both datasets, simulated and motion capture, we selected a portion of the training inputs, denoted as $\mathbf{Z}^{\mathcal{U}}$, to have randomly missing features. The extended dataset $((\mathbf{Z}^{\mathcal{O}}, \mathbf{Z}^{\mathcal{U}}), (\mathbf{Y}^{\mathcal{O}}, \mathbf{Y}^{\mathcal{U}}))$ was used to train: a) our method, referred to as semi-described GP (SD-GP) b) multiple linear regression (MLR) c) regression by performing nearest neighbour (NN) search between the test and training instances, in the observed input locations d) performing data imputation using the standard GP-LVM. Not taking into account the predictive uncertainty during imputation was found to have catastrophic results in the simulated data, as the training set was not robust against bad predictions. Therefore, the "GP-LVM" variant was not considered in the real data experiment. We also considered: e) a standard GP which cannot handle missing inputs straightforwardly and so was trained only on the observed data $(\mathbf{Z}^{\mathcal{O}}, \mathbf{Y}^{\mathcal{O}})$. The goal was to reconstruct test outputs $\mathbf{Y}_*$ given fully observed test inputs $\mathbf{Z}_*$. For the simulated data we used the following sizes: $|\mathbf{Z}^{\mathcal{O}}| = 40$, $|\mathbf{Z}^{\mathcal{U}}| = 60$ and $|\mathbf{Z}_*| = 100$. The dimensionality of the inputs is $q = 15$ and of the outputs is $p = 5$. For the motion capture data we used $|\mathbf{Z}^{\mathcal{O}}| = 50$, $|\mathbf{Z}^{\mathcal{U}}| = 80$ and $|\mathbf{Z}_*| = 200$. In fig. 2 we plot the MSE obtained by the competing methods for a varying percentage offv missing features in $\mathbf{Z}^{\mathcal{U}}$. For the simulated data experiment, each of the points in the plot is an average of 4 runs which considered different random seeds. For clarity, the $y-$axis limit is fixed in figure 2, because some methods produced huge errors. The full picture is in figure 5 (Appendix). As can be seen in the figures, the semi-described GP is able to handle the extra data and make much better predictions, even if a very large portion is missing. Indeed, its performance starts to converge to that of a standard GP when there are 90% missing values in $\mathbf{Z}^{\mathcal{U}}$ and performs

identically to the standard GP when 100% of the values are missing. We found that when q is large compared to p and n, then the data imputation step can be problematic as the percentage of missing features in $\mathbf{Z}^{\mathcal{U}}$ approaches 100% i.e. the method is reliant on having some covariates available. Appendix D discusses this behaviour, but a more systematic investigation is left as future work.

3.2 AUTO-REGRESSIVE GAUSSIAN PROCESSES

Having a method which implicitly models the uncertainty in the inputs of a GP also allows for doing predictions in an autoregressive manner [Oakley and O'Hagan, 2002] while propagating the uncertainty through the predictive sequence [Girard et al., 2003; Quiñonero-Candela et al., 2003]. Specifically, assuming that the given data $\mathbf{Y}$ constitute a multivariate timeseries where the observed time vector $\mathbf{t}$ is equally spaced, and given a time-window of length τ, we can reformat $\mathbf{Y}$ into input-output collections of pairs $\hat{\mathbf{Z}}$ and $\hat{\mathbf{Y}}$ as follows: the first input to the model, $\hat{\mathbf{z}}_{1,:}$, will be given by the stacked vector $[\mathbf{y}_{1,:}, ..., \mathbf{y}_{\tau,:}]$ and the first output, $\hat{\mathbf{y}}_{1,:}$, will be given by $\mathbf{y}_{\tau+1,:}$ and similarly for the other data in $\hat{\mathbf{Z}}$ and $\hat{\mathbf{Y}}$, so that:

$$
\begin{aligned}
[\hat{\mathbf{z}}_{1,:}, \hat{\mathbf{z}}_{2,:}, ..., \hat{\mathbf{z}}_{n-\tau,:}] &= \\
&\left[[\mathbf{y}_{1,:}, \mathbf{y}_{2,:}, ..., \mathbf{y}_{\tau,:}], [\mathbf{y}_{2,:}, \mathbf{y}_{3,:}, ..., \mathbf{y}_{\tau+1,:}], ...\right], \\
[\hat{\mathbf{y}}_{1,:}, \hat{\mathbf{y}}_{2,:}, ..., \hat{\mathbf{y}}_{n-\tau,:}] &= [\mathbf{y}_{\tau+1,:}, \mathbf{y}_{\tau+2,:}, ..., \mathbf{y}_{n,:}].
\end{aligned}
$$

To perform extrapolation we first train the model on the modified dataset $(\hat{\mathbf{Z}}, \hat{\mathbf{Y}})$. By referring to the semi-described formulation described in Section 3.1, we assign all training inputs to the observed set $\mathcal{O}$. After training, we can perform iterative prediction to find a future sequence $\hat{\mathbf{Z}}_* := [\mathbf{y}_{n+1,:}, \mathbf{y}_{n+2,:}, ...]$ where, similarly to the approach taken by Girard et al. [2003], the predictive variance in each
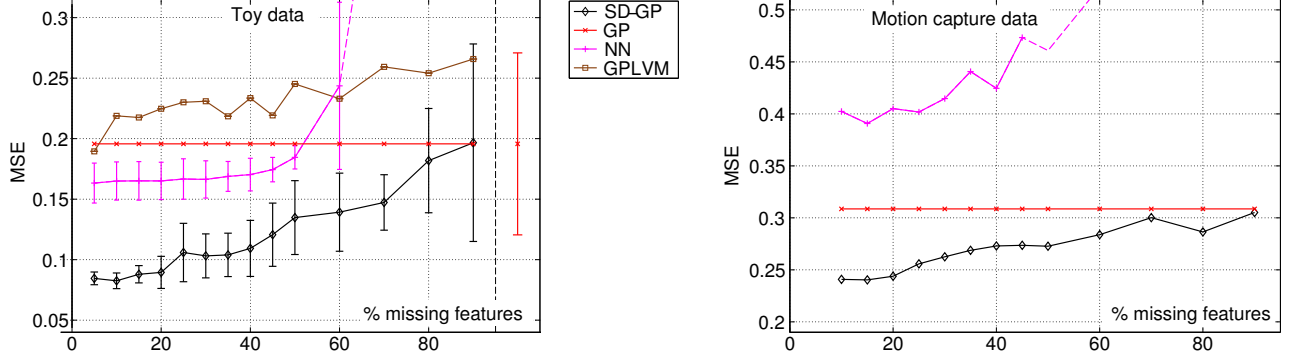
Figure 2: MSE for predictions obtained by different methods on semi-described learning. GP cannot handle partial observations, thus the uncertainty (2σ) is constant; for clarity, the errorbar is plotted separately on the right of the dashed vertical line (for nonsensical x values). The results for simulated data are obtained from 4 trials. For clarity, the limits on the $y-$axis are fixed, so when the errors become too big for certain methods they get off the chart. The errorbars for the GPLVM-based approach are also too large and not plotted. The full picture is given in figure 5 (Appendix).

step is accounted for and propagated in the subsequent predictions. The algorithm makes iterative 1-step predictions in the future; initially, the output $\hat{\mathbf{z}}_{1,*} \coloneqq \mathbf{y}_{n+1,:}$ will be predicted (given the training set) with predictive variance $\hat{\mathbf{S}}_{*;1}$. In the next step, the "observations" set will be augmented to include the distribution of predictions over $\mathbf{y}_{n+1,:}$, by defining $q(\mathbf{x}_{n+1,:}|\hat{\mathbf{z}}_{1,*}) = \mathcal{N}\left(\mathbf{x}_{n+1,:}|\hat{\mathbf{z}}_{*,1}, \hat{\mathbf{S}}_{*;1}\right)$, and so on. This simulation process can be seen as constructing a predictive sequence step by step, i.e. the newly inserted input points constitute parts of the (test) predictive sequence and not training points. Therefore, this procedure can be seen as an iterative version of semi-described learning.

Note that it is straightforward to extend this model by applying this auto-regressive mechanism in a latent space of a stacked model or, more generally, as a deep GP [Damianou and Lawrence, 2013]. By additionally introducing functions that map from this latent space nonlinearly to an observation space, we obtain a fully nonlinear state space model in the manner of Deisenroth et al. [2012]. For our model, uncertainty is encoded in both the states and the nonlinear transition functions. Correct propagation of uncertainty is vital in well calibrated models of future system behavior, and automatic determination of the structure of the model (e.g. the window size) can be informative in describing the order of the underlying dynamical system.

DEMONSTRATION: ITERATIVE FORECASTING

Here we demonstrate our framework in the simulation of a state space model. We consider the Mackey-Glass chaotic time series, a standard benchmark which was also considered by Girard et al. [2003]. The data is one-dimensional so that the timeseries can be represented as pairs of values $\{\mathbf{y}, \mathbf{t}\}, t = 1, 2, \cdots, n$ and simulates the process:

$$\frac{\mathrm{d}\zeta(t)}{\mathrm{d}t} = -b\zeta(t) + \alpha \frac{\zeta(t-T)}{1+\zeta(t-T)^{10}}, (\alpha, b, T) = (0.2, 0.1, 17).$$

Obviously the generating process is very non-linear, rendering this dataset challenging. We trained the autoregressive model on data from this series, where the modified dataset $\{\hat{\mathbf{z}}, \hat{\mathbf{y}}\}$ was created with $\tau = 18$ and we used the first $4\tau = 72$ points to train the model and predicted the subsequent 1110 points through iterative free simulation.

We compared our method with a "naive autoregressive" GP model where the input-output pairs were given by the autoregressive modification of the dataset $\{\hat{\mathbf{z}}, \hat{\mathbf{y}}\}$. For that model, the predictions are made iteratively and the predicted values after each predictive step are added to the "observation" set. However, this standard GP model has no straight forward way of incorporating/propagating the uncertainty and, therefore, the input uncertainty is zero for every step of the iterative predictions. We also compared against the method of Girard et al. [2003][2], denoted in the plots as "GP$_{\text{uncert}}$". Figure 3 shows the results for the last 310 steps (i.e. $t = 800$ onwards) of the full free simulation (1110$-$step ahead forecasting); figure 6 (Appendix) gives a more complete picture. As can be seen in the variances plot, both our method and GP$_{\text{uncert}}$ are more robust in handling the uncertainty throughout the predictions; the "naive" GP method underestimates the uncertainty. Consequently, as can be seen in figure 6, in the first few predictions all methods give the same answer. However, once the predictions of the "naive" method diverge a little by the true values, the error is carried on and amplified due to underestimating the uncertainty. On the other hand, GP$_{\text{uncert}}$ perhaps overestimates the uncertainty and, therefore, is more conservative in its predictions, resulting in higher errors. Quantification of the error is shown in Table 1 (Appendix).

---

[2]We implemented the basic moment matching approach, although in the original paper the authors use additional approximations, namely Taylor expansion around the predictive moments.
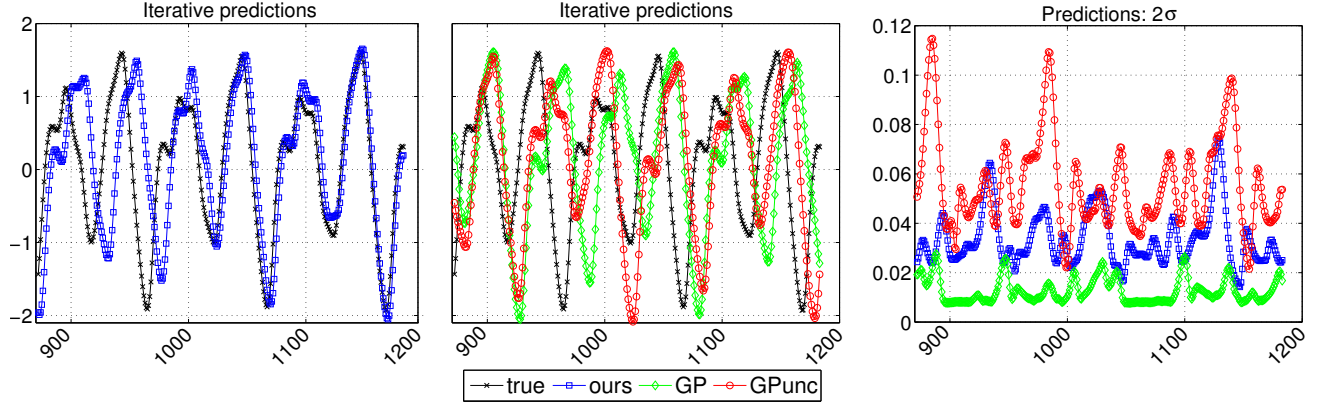
Figure 3: Chaotic timeseries: forecasting 1110 steps ahead by iterative prediction. The first 800 steps are not shown here, but figure 6 (Appendix) gives the complete picture. Comparing: a "naive autoregressive" GP which does not propagate (and hence underestimates) the uncertainties; the method of Girard et al. [2003], referred to as GP$_{\text{uncert}}$; and our approach, which closely tracks the true test sequence until the last steps of the extrapolation. The comparative depiction of the predictions is split into two plots (for clarity), left and center. The rightmost plot shows the predictive uncertainties (2σ). $x-$axis is the prediction step (t) and $y-$axis is the function value, $f(t)$.

3.3 SEMI-SUPERVISED LEARNING

In this section we study semi-supervised learning which, in contrast to semi-described learning, is for handling missing values in the outputs. This scenario is typically encountered in classification settings. Therefore, we introduce the sets $\{\mathcal{L}, \mathcal{M}\}$ that index respectively the *labelled* and *missing* (unlabelled) rows of the outputs (labels) $\mathbf{Y}$. Accordingly, the full dataset is split so that $\mathbf{Z} = (\mathbf{Z}^{\mathcal{L}}, \mathbf{Z}^{\mathcal{M}})$ and $\mathbf{Y} = (\mathbf{Y}^{\mathcal{L}}, \mathbf{Y}^{\mathcal{M}})$, where $\mathbf{Z}$ is now fully observed. The task is then to devise a method that improves classification performance by using both labelled and unlabelled data.

Inspired by Kingma et al. [2014] we define a semi-supervised GP framework where features are extracted from all available information and, subsequently, are given as inputs to a discriminative classifier. Specifically, using the whole input space $\mathbf{Z}$, we learn a low-dimensional latent space $\mathbf{X}$ through an approximate posterior $q(\mathbf{X}) \approx p(\mathbf{X}|\mathbf{Z})$. Obviously, this specific case where the input space is uncertain but totally unobserved (i.e. a latent space) just reduces to the Bayesian GP-LVM model. Notice that the posterior $q(\mathbf{X})$ is no longer constrained with $\mathbf{Z}$ but, rather, directly approximates $p(\mathbf{X}|\mathbf{Z})$, since we now have a forward *probabilistic* mapping from $\mathbf{X}$ to $\mathbf{Z}$ and $\mathbf{Z}$ is treated as a random variable with $p(\mathbf{Z}|\mathbf{X})$ being a Gaussian distribution, i.e. exactly the same setting used in the GP-LVM. Since there is one-to-one correspondence between $\mathbf{X}$, $\mathbf{Z}$ and $\mathbf{Y}$, we can notationally write $\mathbf{X} = (\mathbf{X}^{\mathcal{L}}, \mathbf{X}^{\mathcal{M}})$. Further, since we assume that $q(\mathbf{X})$ is factorised across datapoints, we can write $q(\mathbf{X}) = q(\mathbf{X}^{\mathcal{L}})q(\mathbf{X}^{\mathcal{M}})$.

In the second step of our semi-supervised algorithm, we train a discriminative classifier from $q(\mathbf{X}^{\mathcal{L}})$ to the observed labelled space, $\mathbf{Y}^{\mathcal{L}}$. The main idea is that, by including the inputs $\mathbf{Z}^{\mathcal{M}}$ in the first learning step, we manage to de-

fine a better latent embedding from which we can extract a more useful set of features for the discriminative classifier. Notice that what we would ideally use as input to the discriminative classifier is a whole distribution, rather than single point estimates. Therefore, we wish to take advantage of the associated uncertainty; specifically, we can populate the labelled set by sampling from the distribution $q(\mathbf{X}^{\mathcal{L}})$. For example, if a latent point $\mathbf{x}_{i,:}^{\mathcal{L}}$ corresponds to the input-output pair $(\mathbf{z}_{i,:}^{\mathcal{L}}, \mathbf{y}_{i,:}^{\mathcal{L}})$, then a sample from $q(\mathbf{x}_{i,:}^{\mathcal{L}})$ will be assigned the label $\mathbf{y}_{i,:}^{\mathcal{L}}$.

The two inference steps described above are graphically depicted in Figure 1c. This is exactly the same setting suggested by Kingma et al. [2014], but here we wish to investigate its applicability in a non-parametric, Gaussian process based framework. The very encouraging results reported below point towards the future direction of applying this technique in the framework of deep Gaussian processes [Damianou and Lawrence, 2013], so as to be able to compare to [Kingma et al., 2014] who considered deep, generative (but nevertheless parametric) models.

DEMONSTRATION

We evaluated our semi-supervised GP algorithm in two datasets: firstly, we considered 2000 examples from the USPS handwritten digit database [Hull, 1994]. These examples contained the digits $\{0, 2, 4, 6\}$ and were split so that 800 instances were used as a test set. From the remaining 1200 instances, we selected various portions to be labelled and the rest to be unlabelled. The experiment was repeated 8 times (each time involving different subsets due to different random seeds), so that we can include error-bars in our plots. Secondly, we considered the oil flow data [Bishop and James, 1993] that consist of 1000, 12 dimen-
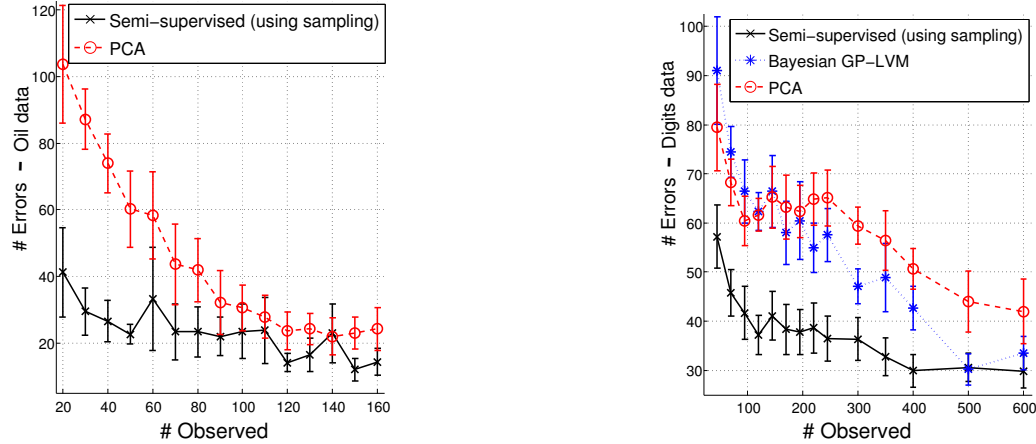
Figure 4: Plots of the number of incorrectly classified test points as a function of $|\mathbf{Z}^{\mathcal{L}}|$. Multiple trials were performed, but the resulting errorbars are shown at one standard deviation. In small training sets large errorbars are expected because, occasionally, very challenging instances/outliers can be included and result in high error rates (for all methods) that affect the overall standard deviation. The Bayesian GP-LVM baseline struggled with small training sets and performed very badly in the oil dataset; thus, it is not plotted for clarity.

sional observations belonging to three known classes corresponding to different phases of oil flow. In each of the 10 performed trials, 700 instances were used as a test set whereas the rest were split to different proportions of labelled/unlabelled sets. Multi-label data can also be handled by our method, but this case was not considered here.

Our method learned a low-dimensional embedding $q(\mathbf{X})$ from all available inputs, and a logistic regression classifier was then trained from the relevant parts of the embedding to the corresponding class space. We experimented with taking different numbers of samples from $q(\mathbf{X}^{\mathcal{L}})$ for populating the initial labelled set; the difference after increasing over 6 samples was minimal. Also, when using only the mean of $q(\mathbf{X}^{\mathcal{L}})$ (as opposed to using multiple samples) we obtained worse results (especially in the digits data), but this method still outperformed the baselines. We compared with training the classifier on features learned by (a) a standard Bayesian GP-LVM and (b) PCA, both applied in $\mathbf{Z}^{\mathcal{L}}$. Both of the baselines do not take $\mathbf{Z}^{\mathcal{M}}$ into account, nor do they populate small training sets using sampling. Figure 4 presents results suggesting that our approach manages to effectively take into account unlabelled data. The gain in performance is significant, and our method copes very well even when labelled data is extremely scarce. Notice that all methods would perform better if a more robust classifier was used, but logistic regression was a convenient choice for performing multiple trials fast. Therefore, our conclusions can be safely drawn from the obtained relative errors, since all methods were compared on equal footing.

4   DISCUSSION AND FUTURE WORK

We have defined semi-described learning as the scenario where missing and uncertain values occur in the inputs. We

considered semi-described problems to be part of a general class of missing value problems that also includes semi-supervised learning and auto-regressive future state simulation. A principled method for including input uncertainty and partial inputs in Gaussian process models was also introduced to solve these problems within a single, coherent framework. We explicitly represent this uncertainty as approximate posterior distributions which are variationally constrained. This allowed us to further define algorithms for casting the missing value problems as particular instances of learning pipelines which use our variationally constrained GP formulation as a building block. Our algorithms resulted in significant performance improvement in forecasting, regression and classification. We believe that our contribution paves the way for building powerful models for representation learning from real-world, heterogenous data. In particular, this can be achieved by combining our method with deep Gaussian process models [Damianou and Lawrence, 2013] that use relevance determination techniques [Damianou et al., 2012], so as to consolidate semi-described hierarchies of features that are gradually abstracted to concepts. We plan to investigate the application of these models in settings where control [Deisenroth et al., 2014] or robotic systems learn by simulating future states in an auto-regressive manner and by using incomplete data with miminal human intervention. Transfer learning is another promising direction for applying these models.

References

C. M. Bishop and G. D. James. Analysis of multiphase flows using dual-energy gamma densitometry and neural networks. *Nuclear Instruments and Methods in Physics Research*, A327:580–593, 1993.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-supervised Learning*. MIT Press, Cambridge, MA, 2006.

Z. Dai, A. Damianou, J. Hensman, and N. Lawrence. Gaussian process models with parallelization and GPU acceleration. *arXiv preprint arXiv:1410.4984*, 2014.

P. Dallaire, C. Besse, and B. Chaib-Draa. Learning Gaussian process models from uncertain data. In *Neural Information Processing*, pages 433–440. Springer, 2009.

A. Damianou and N. Lawrence. Deep Gaussian processes. In *Proceedings of the Sixteenth International Workshop on Artificial Intelligence and Statistics (AISTATS)*, pages 207–215. JMLR W&CP 31, 2013.

A. Damianou and N. Lawrence. Uncertainty propagation in Gaussian process pipelines. *NIPS workshop on modern non-parametrics*, 2014.

A. Damianou, M. Titsias, and N. D. Lawrence. Variational Gaussian process dynamical systems. In *Advances in Neural Information Processing Systems 24*, pages 2510–2518. 2011.

A. Damianou, C. Ek, M. Titsias, and N. Lawrence. Manifold relevance determination. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 145–152. Omnipress, 2012.

M. P. Deisenroth, R. D. Turner, M. F. Huber, U. D. Hanebeck, and C. E. Rasmussen. Robust filtering and smoothing with Gaussian processes. *Automatic Control, IEEE Transactions on*, 57(7):1865–1871, 2012.

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*, 99:1, 2014. ISSN 0162-8828.

C. H. Ek, J. Rihan, P. Torr, G. Rogez, and N. D. Lawrence. Ambiguity modeling in latent spaces. In A. Popescu-Belis and R. Stiefelhagen, editors, *Machine Learning for Multimodal Interaction (MLMI 2008)*, LNCS, pages 62–73. Springer-Verlag, 28–30 June 2008.

Y. Gal, M. van der Wilk, and C. E. Rasmussen. Distributed variational inference in sparse Gaussian process regression and latent variable models. *arXiv:1402.1389*, 2014.

Z. Ghahramani and M. I. Jordan. Learning from incomplete data. Technical Report CBCL 108, Massachusetts Institute of Technology, 1994.

A. Girard, C. E. Rasmussen, J. Quiñonero Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs—application to multiple-step ahead time series forecasting. In *Advances in Neural Information Processing Systems*, pages 529–536, 2003.

J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:550–554, 1994.

D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling. Semi-supervised learning with deep generative models. *CoRR*, abs/1406.5298, 2014.

N. D. Lawrence. The Gaussian process latent variable model. Technical Report CS-06-03, The University of Sheffield, Department of Computer Science, 2006.

N. D. Lawrence and M. I. Jordan. Semi-supervised learning via Gaussian processes. In L. Saul, Y. Weiss, and L. Bouttou, editors, *Advances in Neural Information Processing Systems*, volume 17, pages 753–760, Cambridge, MA, 2005. MIT Press.

N. D. Lawrence and J. Quiñonero Candela. Local distance preservation in the GP-LVM through back constraints. In W. Cohen and A. Moore, editors, *Proceedings of the International Conference in Machine Learning*, volume 23, pages 513–520. Omnipress, 2006. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143909.

A. McHutchon and C. E. Rasmussen. Gaussian process training with input noise. In *NIPS*, 2011.

J. Oakley and A. O'Hagan. Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika*, 89(4):769–784, 2002.

M. Osborne and S. J. Roberts. Gaussian processes for prediction. Technical report, Department of Engineering Science, University of Oxford, 2007.

J. Quiñonero-Candela. *Learning with uncertainty-Gaussian processes and relevance vector machines*. PhD thesis, Technical University of Denmark, 2004.

J. Quinonero-Cañdela and S. Roweis. Data imputation and robust training with Gaussian processes. *NIPS*, 2003.

J. Quiñonero-Candela, A. Girard, J. Larsen, and C. E. Rasmussen. Propagation of uncertainty in bayesian kernel models-application to multiple-step ahead forecasting. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 2, pages II–701. IEEE, 2003.

C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1.*, volume 1, pages 29–36, Jan 2005. doi: 10.1109/ACVMOT.2005.107.

D. B. Rubin. *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons, 2004.

M. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. *Journal of Machine Learning Research - Proceedings Track*, 9:844–851, 2010.

J. Zhu, A. Ahmed, and E. P. Xing. Medlda: maximum margin supervised topic models. *The Journal of Machine Learning Research*, 13(1):2237–2278, 2012.

A   APPENDIX: VARIATIONAL LOWER BOUND

In this appendix we give some more details on the computation of the variational lower bound for the variationally constrained model.

The augmented joint probability density (after introducing the inducing points) takes the form,

$$p(\mathbf{Y}, \mathbf{F}, \mathbf{U}, \mathbf{X}|\mathbf{X}_u)$$
$$= p(\mathbf{Y}|\mathbf{F})p(\mathbf{F}|\mathbf{U}, \mathbf{X}, \mathbf{X}_u)p(\mathbf{U}|\mathbf{X}_u)p(\mathbf{X})$$
$$= \left(\prod_{j=1}^{p} p(\mathbf{y}_j|\mathbf{f}_j)p(\mathbf{f}_j|\mathbf{u}_j, \mathbf{X}, \mathbf{X}_u)p(\mathbf{u}_j|\mathbf{X}_u) \right) p(\mathbf{X}).$$

In the r.h.s above, the observed inputs $\mathbf{Z}$ do not appear, exactly because we introduce them through the variational constraint, which does not constitute a probabilistic mapping. In the above equations we have

$$p(\mathbf{f}_j|\mathbf{u}_j, \mathbf{X}, \mathbf{X}_u) = \mathcal{N}\left(\mathbf{f}_j|\mathbf{a}_j, \boldsymbol{\Sigma}_f\right),$$

being the conditional GP prior with

$$\mathbf{a}_j = \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u}_j \quad \text{and} \quad \boldsymbol{\Sigma}_f = \mathbf{K} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}$$

and

$$p(\mathbf{u}_j|\mathbf{X}_u) = \mathcal{N}(\mathbf{u}_j|\mathbf{0}, \mathbf{K_{uu}}),$$

is the marginal GP prior over the inducing variables. In the above expressions, $\mathbf{K_{uu}}$ denotes the covariance matrix constructed by evaluating the covariance function on the inducing points, $\mathbf{K_{uf}}$ is the cross-covariance between the inducing and the latent points and $\mathbf{K_{fu}} = \mathbf{K_{uf}^\top}$.

In order to perform variational inference in this expanded probability model, we introduce the variational distributions $q(\mathbf{X}|\mathbf{Z})$ and $q(\mathbf{U})$, which are both taken to be Gaussian. For convenience, we drop the inducing points $\mathbf{X}_u$ from our expressions for the remainder of the Appendix, for convenience. We now have:

$$\log p(\mathbf{Y}|\mathbf{X}_u) =$$
$$\log \int_{\mathbf{U}, \mathbf{X}} p(\mathbf{U})p(\mathbf{X}) \int_{\mathbf{F}} p(\mathbf{Y}|\mathbf{F})p(\mathbf{F}|\mathbf{U}, \mathbf{X}).$$

By applying Jensen's inequality, we obtain a lower bound $\mathcal{F}(q(\mathbf{X}), q(\mathbf{U}))$ on the above marginal likelihood, where:

$$\mathcal{F}(q(\mathbf{X}|\mathbf{Z}), q(\mathbf{U})) =$$
$$\int_{\mathbf{U}, \mathbf{X}} q(\mathbf{U})q(\mathbf{X}|\mathbf{Z}) \log \frac{p(\mathbf{U})p(\mathbf{X})\int_{\mathbf{F}} p(\mathbf{Y}|\mathbf{F})p(\mathbf{F}|\mathbf{U}, \mathbf{X})}{q(\mathbf{U})q(\mathbf{X}|\mathbf{Z})}$$
$$= \int_{\mathbf{U}, \mathbf{X}} q(\mathbf{U})q(\mathbf{X}|\mathbf{Z}) \log \frac{p(\mathbf{U})\int_{\mathbf{F}} p(\mathbf{Y}|\mathbf{F})p(\mathbf{F}|\mathbf{U}, \mathbf{X})}{q(\mathbf{U})}$$
$$- \text{KL}\left(q(\mathbf{X}|\mathbf{Z}) \,\|\, p(\mathbf{X})\right)$$
$$:= \hat{\mathcal{F}} - \text{KL}\left(q(\mathbf{X}|\mathbf{Z}) \,\|\, p(\mathbf{X})\right).$$

At this point, our variational bound is similar to the one of equation (7), but the first term, here denoted as $\hat{\mathcal{F}}$, refers to the expanded probability space and, thus, involves the inducing inputs and the additional variational distribution $q(\mathbf{U})$. Since the second term (the KL term) is tractable (because it only involves Gaussian distributions), we are now going to focus on the $\hat{\mathcal{F}}$ term. By breaking the logarithm again, we can further write this term as:

$$\hat{\mathcal{F}} = \int_{\mathbf{U}, \mathbf{X}} q(\mathbf{U})q(\mathbf{X}|\mathbf{Z}) \log \left(\int_{\mathbf{F}} p(\mathbf{Y}|\mathbf{F})p(\mathbf{F}|\mathbf{U}, \mathbf{X}) \right)$$
$$- \text{KL}\left(q(\mathbf{U}) \,\|\, p(\mathbf{U})\right) \quad \text{(A.1)}.$$

We notice that we can make use of Jensen's inequality once more, because:

$$\log \left(\int_{\mathbf{F}} p(\mathbf{Y}|\mathbf{F})p(\mathbf{F}|\mathbf{U}, \mathbf{X}) \right) \geq \int_{\mathbf{F}} p(\mathbf{F}|\mathbf{U}, \mathbf{X}) \log p(\mathbf{Y}|\mathbf{F}).$$

This expectation is analytically tractable. Indeed, for a single dimension j, we can find this expectation as:

$$\int_{\mathbf{f}_j} p(\mathbf{f}_j|\mathbf{u}_j, \mathbf{X}) \log p(\mathbf{y}_j|\mathbf{f}_j) =$$
$$\log \mathcal{N}\left(\mathbf{y}_j|\mathbf{a}_j, \beta^{-1}\mathbf{I}\right) - \frac{\beta}{2}\text{tr}\left(\mathbf{K}\right)$$
$$+ \frac{\beta}{2}\text{tr}\left(\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\mathbf{K_{fu}}\right),$$

where $\mathbf{K}$ is the covariance matrix constructed by evaluating the covariance function on the training inputs $\mathbf{X}$. The full expression can be found by taking the appropriate product with respect to dimensions; indeed, since the joint probability factorises with respect to output dimensions j, then a bound to the logarithm of the marginal likelihood can be written as a sum over terms, where every term considers a single dimension j. Notice that to obtain this tractable bound we did not explicitly make the assumption of equation (9) about the form of the variational distribution. However, this assumption is still made implicitly and the equivalence of the two derivations is rather instructive with respect to the effect of a variational constraint.

We also notice that in the above expression, the covariance matrix $\mathbf{K}$ is no longer inverted. Therefore, by writting the term $\hat{\mathcal{F}}$ in this form, we manage to obtain an expression which allows the uncertainty in $\mathbf{X}$ to be propagated through the GP mapping.

It is possible to also obtain a "tighter" variational bound $\mathcal{F}(q(\mathbf{X}|\mathbf{Z})) \geq \mathcal{F}(q(\mathbf{U}), q(\mathbf{X}|\mathbf{Z}))$ which does not depend on $q(\mathbf{U})$. To do so, we need to "collect" all terms that contain $p(\mathbf{U})$ from equation (A.1) and find the stationary point with respect to the distribution $q(\mathbf{U})$ (by computing the gradient w.r.t $q(\mathbf{U})$ and setting it to zero). By doing so, we are then able to replace $q(\mathbf{U})$ with its optimal value back to the variational bound. Titsias and Lawrence [2010] further explain this trick.
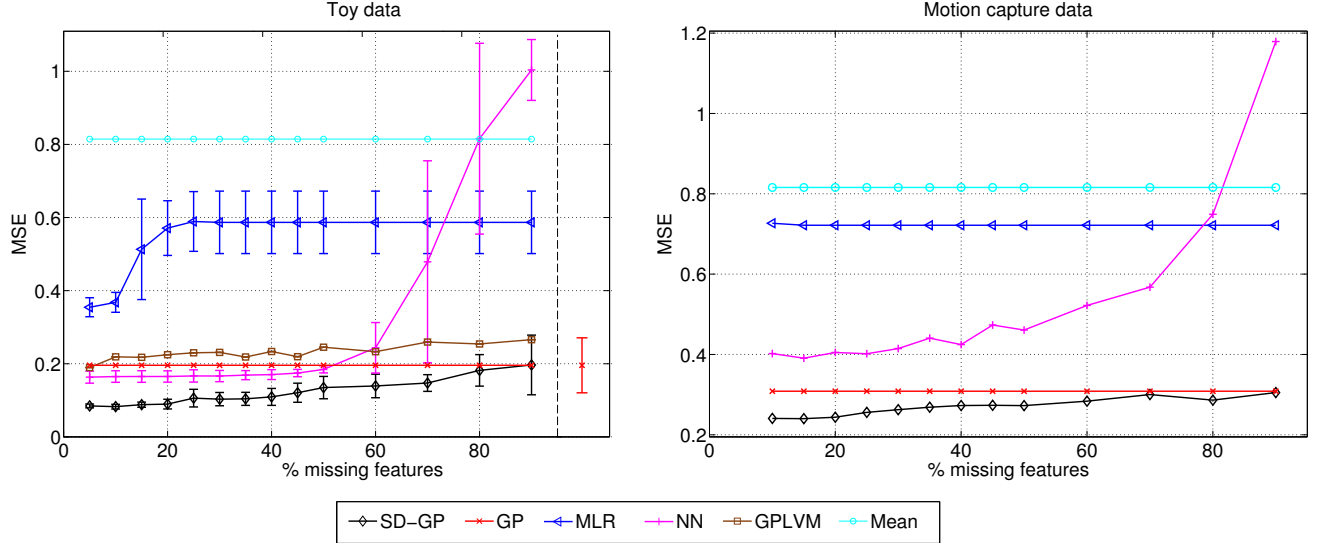
Figure 5: MSE for predictions obtained by different methods on semi-described learning (full version of figure 2). Comparing our method (SD-GP), the standard GP method, multiple linear regression (MLR), nearest neighbour regression on the input space (NN), the data-imputation method based on GP-LVM and the mean predictor (mean). The results for simulated data are obtained from 4 trials. The GP method cannot handle partial observations, thus the uncertainty (2σ) is constant; for clarity, the errorbar is plotted separately on the right of the dashed vertical line (for nonsensical x values). The GP-LVM method produced huge errorbars (about 3.5 times larger than thos of MLR), thus we don't plot them here, for clarity.

B APPENDIX: MORE DETAILS FOR THE SEMI-DESCRIBED LEARNING EXPERIMENT

In Section 3.1 we looked at performing predictions with Gaussian processes trained from partially observed inputs. Our method (semi-described GP or SD-GP) was compared to other approaches in figure 2, but the limit in the $y-$axis was fixed to a smaller value to show the comparison with the standard GP method more clearly. For the same reason, methods which produced very large errors were omitted. In this appendix we show the full figure from all the obtained results – figure 5.

The conclusion drawn from figure 5 is that our method is very efficient in taking into account the extra, partially observed input set $\mathbf{Z}^{\mathcal{U}}$. This is true even if this extra set only has a small proportion of features observed. On the other hand, nearest neighbour runs into difficulties when real data are considered and, even worse, produces huge errors when more than 60% of the features are missing in $\mathbf{Z}^{\mathcal{U}}$. Finally, the baseline which uses the standard GP-LVM as a means of imputing missing values produces bad results, in fact worse compared to if the extra set $\mathbf{Z}^{\mathcal{U}}$ is just ignored (i.e. the GP baseline). This is because the baseline using GP-LVM treats the input space as single point estimates; by not incorporating (and optimising jointly) the uncertainty for each input location, the model has no way of ignoring "bad" imputed values.

C APPENDIX: MORE DETAILS FOR THE AUTO-REGRESSIVE EXPERIMENT

This appendix refers to the auto-regressive Gaussian process model developed in Section 3.2. In figure 3 we showed the results from the last 310 steps of the iterative forecasting task. Here (figure 6) we show the rest of the predictive sequence, obtained for extrapolating up until 1110 steps. The corresponding quantification of the error is shown in Table 1.

Table 1: Mean squared and mean absolute error obtained when extrapolating in the chaotic time-series data. GP$_{\text{uncert}}$ refers to the basic (moment matching) method of Girard et al. [2003] and the "naive" autoregressive GP approach is the one which does not propagate uncertainties.

Method	MAE	MSE
ours	**0.529**	**0.550**
GP$_{\text{uncert}}$	0.700	0.914
"naive" GP approach	0.799	1.157

D APPENDIX: THE EFFECT OF q, p, n IN SEMI-DESCRIBED LEARNING

As mentioned in Section 3.1, we found that when q is large compared to p and n, then the data imputation step of our

algorithm can be problematic as the percentage of missing features in $\mathbf{Z}^{\mathcal{U}}$ approaches 100%. This is somehow a corner-case, but it still shows that the method is reliant on having some covariates available. To investigate further this issue we created simulated data as explained in Section 3.1, but this time multiple datasets were generated with different input and output dimensions, q and p respectively. In figure 7 we show the comparison of SD-GP and the standard GP (which ignores $\mathbf{Z}^{\mathcal{U}}$) for different selections of q, p and percentage of missing features in $\mathbf{Z}^{\mathcal{U}}$.
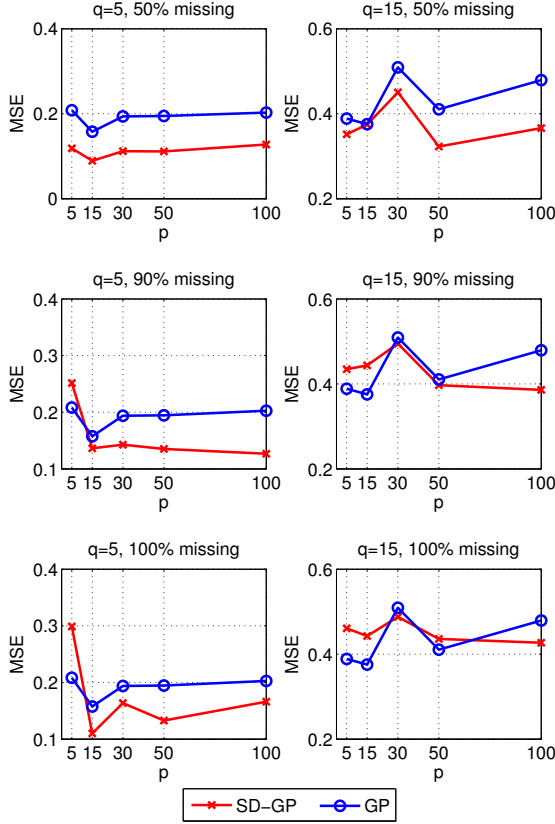

Figure 7: Comparison of our method (SD-GP) and the standard GP (which ignores $\mathbf{Z}^{\mathcal{U}}$) for different selections of q, p and percentage of missing features in $\mathbf{Z}^{\mathcal{U}}$.

The summary of this experiment is that:

- For the most usual scenarios, i.e. when the percentage of features missing is not too high, SD-GP performs very well, but as p and n become small compared to q, then the performance of the method seems to deteriorate.

- Even if 100% of the features are missing in $\mathbf{Z}^{\mathcal{U}}$, using our SD-GP can still be advantageous compared to using a standard GP. This is because SD-GP can utilise the extra information in the fully observed outputs, $\mathbf{Y}^{\mathcal{U}}$, which correspond to the fully missing set

$\mathbf{Z}^{\mathcal{U}}$. However, when the percentage of missing features is very large and the relative size of p and n is small compared to q, then the method can produce worst results compared to the standard GP.

To explain the challenge of handling missing values with SD-GP, consider that a separate variational parameter exists for every input, namely the parameters $\mu_{i,j}^{\mathcal{U}}, S_{i,j}^{\mathcal{U}}, i = 1, ...n, j = 1, ..., q$ of step 7 in Algorithm 1. In the extreme cases mentioned in the previous paragraph, the number of variational parameters remains large but the available covariates to learn from are too few. This renders the optimisation of the parameters very difficult.
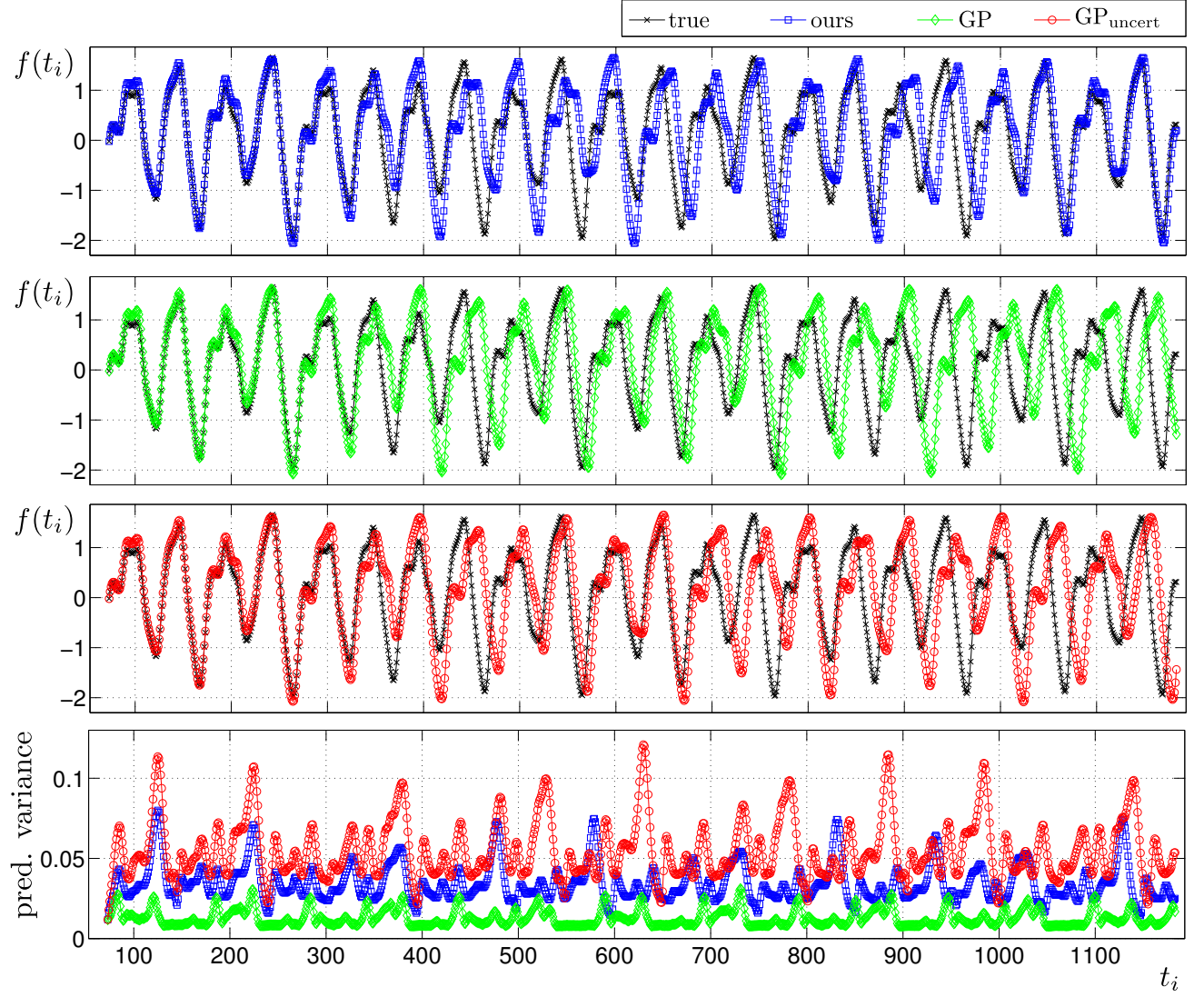
Figure 6: The full predictions obtained by the competing methods for the chaotic time-series data. The top 3 plots show the values obtained in each predictive step for each of the compared methods; the plot on the bottom shows the corresponding predictive uncertainties (2σ). GP$_{uncert}$ refers to the basic (moment matching) method of Girard et al. [2003] and the GP is the "naive" autoregressive GP which does not propagate uncertainties.