Learning a Gaussian Process Model with Uncertain Inputs*

Agathe Girard
Dept. of Computing Science
University of Glasgow
Glasgow, UK
agathe@dcs.gla.ac.uk

Roderick Murray-Smith
Dept. of Computing Science
University of Glasgow
& Hamilton Institute
National University of Ireland
Maynooth, Ireland
rod@dcs.gla.ac.uk

Abstract

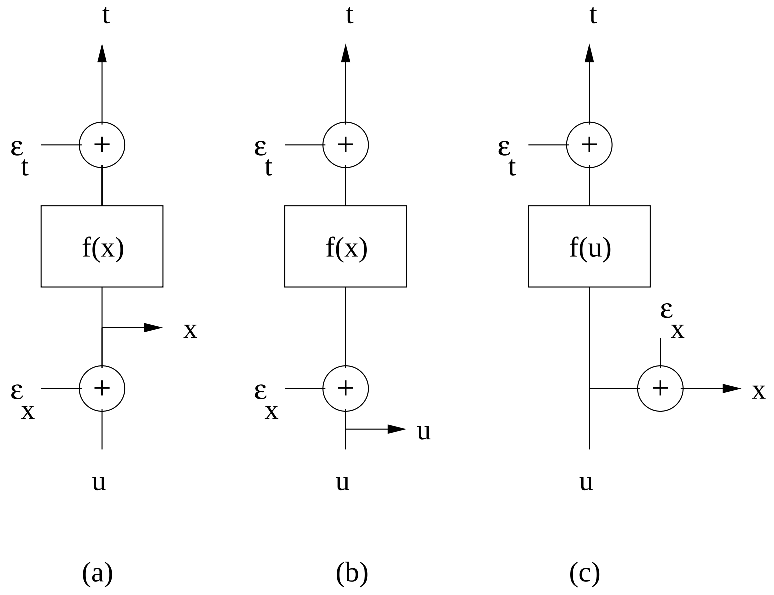
Learning with uncertain inputs is well-known to be a difficult task. In order to achieve this analytically using a Gaussian Process prior model, we expand the original process around the input mean (Delta method), assuming the random input is normally distributed. We thus derive a new process whose covariance function accounts for the randomness of the input. We illustrate the effectiveness of the proposed model on a simple static simulation example and on the modelling of a nonlinear noisy time-series.

1 Background

Solving the learning task with uncertain or missing inputs has been the scope of much research and the level of difficulty obviously depends on the type of model used. One can distinguish between different situations, depending on the nature of a particular application. Figure 1 summarizes the main different cases: (a) corresponds for instance to the modelling of a noisy time-series.[1] Case (b) is commonly encountered when the system of interest senses inputs imperfectly and (c) corresponds to clean inputs to the system, but corruption during sensing of the inputs for data collection. We can also imagine a blend of these, with both noisy channels from u to system, as in (a) & (b), and independent noise on observations of u, as in (c).

[*] Technical Report TR-2003-144, Department of Computing Science, University of Glasgow, June, 2003.

[1] When a state-space representation is used, in which the state is formed of delayed observed values.



(a) (b) (c)

Figure 1: Uncertain inputs arising in different situations. t is the target, noisy model output, x the noisy input such that $x = u + \epsilon_x$ with $\epsilon_x \sim \mathcal{N}(0, v_x)$, and the arrow indicates the variables used for training. Our model aims at improving the learning of systems like (a) and (b) where the target is a function of a random input x.

In the statistics community, such models dealing with uncertain inputs are known as *error-in-variables* models. In [1] these models are analyzed in the Bayesian framework and inference is made about the unknown x's (case (b) in Figure 1) and model parameters. In [2, 3], their solution consists of integrating over the unknown (uncertain) input, using an input distribution estimated directly from the data. Mixture models have also been used, along with the Expectation Maximization algorithm [4].

In this paper, we suggest a novel approach for the learning of systems of type (a) and (b). We introduce a *modified* Gaussian Process model with a *corrected* covariance function, accounting for the input noise variance.

2 Overview of the problem

We assume the following statistical model

$$t = f(\mathbf{x}) + \epsilon_t \tag{1}$$

where $\mathbf{x}$ is a D-dimensional input and ϵ_t the output, additive, Gaussian white noise such that $\epsilon_t \sim \mathcal{N}(0, v_t)$, where v_t is the unknown noise variance. Such a model implies that

$$E[t|\mathbf{x}] = f(\mathbf{x}) \; . \tag{2}$$

Now, let $\mathbf{x} = \mathbf{u} + \boldsymbol{\epsilon}_x$, or $\mathbf{x} \sim \mathcal{N}(\mathbf{u}, v_x I)$, where I is the $D \times D$ identity matrix and v_x is the input noise variance.[2] In this case, the expectation of t given the characteristics of $\mathbf{x}$ is obtained by integrating over the input distribution

$$E[t|\mathbf{u}, v_x] = \int f(\mathbf{x})p(\mathbf{x})d\mathbf{x}\ . \tag{3}$$

This integral can not be solved analytically without approximations for many forms of $f(\mathbf{x})$.

2.1 Analytical approximation using the Delta method

The function f of the random argument $\mathbf{x}$ can always be approximated by a second order Taylor expansion around the mean $\mathbf{u}$ of $\mathbf{x}$:

$$f(\mathbf{x}) = f(\mathbf{u}) + (\mathbf{x}-\mathbf{u})^T f'(\mathbf{u}) + \frac{1}{2}(\mathbf{x}-\mathbf{u})^T f''(\mathbf{u})(\mathbf{x}-\mathbf{u}) + O(||\mathbf{x}-\mathbf{u}||^3) \tag{4}$$

where $f'(\mathbf{u}) = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ and $f''(\mathbf{u}) = \frac{\partial^2 f(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}^T}$, evaluated at $\mathbf{x} = \mathbf{u}$.

Within this approximation,[3] we can now solve the integral (3). We have

$$\begin{aligned} E[t|\mathbf{u}, v_x] &\simeq \int \left[f(\mathbf{u}) + (\mathbf{x}-\mathbf{u})^T f'(\mathbf{u}) + \frac{1}{2}(\mathbf{x}-\mathbf{u})^T f''(\mathbf{u})(\mathbf{x}-\mathbf{u}) \right] p(\mathbf{x})d\mathbf{x} & (5) \\ &\simeq f(\mathbf{u}) + \frac{1}{2}\mathrm{Tr}[f''(\mathbf{u})v_x I] = f(\mathbf{u}) + \frac{v_x}{2}\mathrm{Tr}[f''(\mathbf{u})] & (6) \end{aligned}$$

where Tr denotes the trace.

Thus, the new generative model for our data is

$$\begin{cases} t = g(\mathbf{u}, v_x) + \epsilon_t \\ g(\mathbf{u}, v_x) = f(\mathbf{u}) + \frac{v_x}{2}\mathrm{Tr}[f''(\mathbf{u})]\ . \end{cases} \tag{7}$$

3 Gaussian Process modelling with noisy inputs

Let us recall that, in the case of inputs which are *certain*, the GP modelling framework consists in putting a normal prior on the space of admissible functions f. That is, for given $\mathbf{u}_1, \ldots, \mathbf{u}_n$, the model outputs $y_1 = f(\mathbf{u}_1), \ldots, y_n = f(\mathbf{u}_n)$ have a joint multivariate Gaussian distribution:

[2] Note that accounting for different variances and/or covariances between inputs in different dimensions would be straightforward; that would simply involve more parameters.

[3] All approximations being imperfect by nature, it is clear that the goodness of the expansion will depend on how nonlinear f is in the neighborhood of $\mathbf{u}$, as well as on how large v_x is.

$y_1, \ldots, y_n \sim \mathcal{N}(0, \mathbf{\Sigma})$, where $\Sigma_{ij} = \mathrm{cov}(y_i, y_j) = C(\mathbf{u}_i, \mathbf{u}_j)$. It is common to assume that the process is stationary, with zero-mean and a squared exponential covariance function

$$C(\mathbf{u}_i, \mathbf{u}_j) = v \exp\left[-\frac{1}{2} \sum_{d=1}^{D} w_d (u_i^d - u_j^d)^2 \right] \tag{8}$$

where v and $w_1, \ldots, w_D$ are the model's parameters (see [5] for more details).

3.1 Defining a new Gaussian Process

In the case of uncertain or random inputs, the new input/output relationship is given by (7), where the former function f, in the noise-free case, has been replaced by $g(\mathbf{u}, v_x) = f(\mathbf{u}) + \frac{v_x}{2}\mathrm{Tr}[f''(\mathbf{u})]$.

If we put a Gaussian prior on $f(\mathbf{u})$, we can derive the corresponding prior on its second derivative and then define the prior on the space of admissible functions $g(\mathbf{u}, v_x)$ which is viewed as the sum of the two correlated random functions, $f(\mathbf{u})$ and $\frac{v_x}{2}\mathrm{Tr}[f''(\mathbf{u})]$.

In the following, we use results from the theory of random functions [6]. Let us recall that if $X(r)$ and $Y(r)$ are two random functions of the same argument r, with expected values $m_x(r)$ and $m_y(r)$ and covariance functions $C_x(r, r')$ and $C_y(r, r')$ respectively, then the mean and covariance function of $Z(r) = X(r) + Y(r)$ are given by

$$m_z(r) = m_x(r) + m_y(r) \tag{9}$$

$$C_z(r, r') = C_x(r, r') + C_y(r, r') + C_{xy}(r, r') + C_{yx}(r, r') \tag{10}$$

in the case $X(r)$ and $Y(r)$ are correlated and $C_{xy}(r, r')$, $C_{yx}(r, r')$ are the cross-covariance functions.

We can now apply this to our function $g(.)$. Let us first derive the mean and covariance function of $g(\mathbf{u}, v_x)$ in the one-dimensional case and then extend these expressions to D dimensions.

Given that $f(u)$ has zero-mean and covariance function $C(u_i, u_j)$, as given by (8), its second derivative, $f''(u)$, has zero-mean and covariance function $\partial^4 C(u_i, u_j)/\partial u_i^2 \partial u_j^2$ [6]. It is then straightforward that $\frac{v_x}{2} f''(u)$ has zero-mean and covariance function $\frac{v_x^2}{4}\partial^4 C(u_i, u_j)/\partial u_i^2 \partial u_j^2$. Also, the cross-covariance function between $f(u)$ and $\frac{v_x}{2} f''(u)$ is given by $\frac{v_x}{2}\partial^2 C(u_i, u_j)/\partial u_i^2$ [6].

Therefore, using the fact we have $\frac{\partial^2 C(u_i, u_j)}{\partial u_i^2} = \frac{\partial^2 C(u_i, u_j)}{\partial u_j^2}$, in one dimension, $g(u, v_x) = f(u) + \frac{v_x}{2} f''(u)$ has zero-mean and covariance function

$$\mathrm{cov}[g(u_i, v_x), g(u_j, v_x)] = C(u_i, u_j) + \frac{v_x^2}{4}\frac{\partial^4 C(u_i, u_j)}{\partial u_i^2 \partial u_j^2} + v_x \frac{\partial^2 C(u_i, u_j)}{\partial u_i^2} \,. \tag{11}$$

In the case of D-dimensional inputs, we have

$$\begin{aligned}\mathrm{cov}[g(\mathbf{u}_i, v_x), g(\mathbf{u}_j, v_x)] = C(\mathbf{u}_i, \mathbf{u}_j) &+ \frac{v_x^2}{4}\mathrm{Tr}\left[\frac{\partial^2}{\partial \mathbf{u}_i \partial \mathbf{u}_i^T}\left[\frac{\partial^2 C(\mathbf{u}_i, \mathbf{u}_j)}{\partial \mathbf{u}_j \partial \mathbf{u}_j^T}\right]\right] \\ &+ v_x \mathrm{Tr}\left[\frac{\partial^2 C(\mathbf{u}_i, \mathbf{u}_j)}{\partial \mathbf{u}_i \partial \mathbf{u}_i^T}\right]\end{aligned} \tag{12}$$

where $\frac{\partial^2}{\partial \mathbf{u}_i \partial \mathbf{u}_i^T}\left[\frac{\partial^2 C(\mathbf{u}_i, \mathbf{u}_j)}{\partial \mathbf{u}_j \partial \mathbf{u}_j^T}\right]$ is a $D \times D$ matrix, each entry of which being a $D \times D$ matrix: the block (r, s) contains $\frac{\partial^2}{\partial \mathbf{u}_i^r \partial \mathbf{u}_i^s}\left[\frac{\partial^2 C(\mathbf{u}_i, \mathbf{u}_j)}{\partial \mathbf{u}_j \partial \mathbf{u}_j^T}\right]$.

So we see that the first term of the *corrected* covariance function corresponds to the noise-free case plus two correction terms weighted by the input noise variance, which might be either learnt or assumed to be known *a priori*. As we would expect, as v_x tends to zero, $\mathrm{cov}[g(\mathbf{u}_i, v_x), g(\mathbf{u}_j, v_x)] \rightarrow C(\mathbf{u}_i, \mathbf{u}_j)$ which corresponds to the *certain*, noise-free case.

3.2 Inference and prediction

Within this approximation, the likelihood of the data $\{t_1, \ldots, t_N\}$ is readily obtained. We have

$$\mathbf{t}|\mathbf{U} \sim \mathcal{N}(0, \mathbf{Q}) \quad \text{with} \quad Q_{ij} = \Sigma'_{ij} + v_t \delta_{ij} \tag{13}$$

where $\mathbf{t}$ is the $N \times 1$ vector of observed targets, $\mathbf{U}$ the $N \times D$ matrix of input means, Σ'_{ij} is given by (12) and $\delta_{ij} = 1$ when $i = j$, 0 otherwise. The parameters $\Theta = [w_1, \ldots, w_D, v, v_x, v_t]$ can then be learnt either in a Maximum Likelihood framework or in a Bayesian way, by assigning priors and computing their posterior distribution.

When using the *usual* GP, the predictive distribution of a model output corresponding to a new input $\mathbf{u}_*$, $p(f(\mathbf{u}_*)|\Theta, \{\mathbf{u}, \mathbf{t}\}, u_*)$, is Gaussian with mean and variance respectively given by

$$\begin{cases}\mu = \mathbf{k}^T Q^{-1} \mathbf{t} \\ \sigma^2 = k - \mathbf{k}^T Q^{-1} \mathbf{k}\end{cases} \tag{14}$$

where $\mathbf{k}$ is the vector of covariances between the test and the training inputs and k the covariance between the test input and itself. We have $Q_{ij} = \Sigma_{ij} + v_t \delta_{ij}$ and

$$\Sigma_{ij} = C(\mathbf{u}_i, \mathbf{u}_j),\ k_i = C(\mathbf{u}_*, \mathbf{u}_i),\ k = C(\mathbf{u}_*, \mathbf{u}_*) \tag{15}$$

for $i, j = 1, \ldots, N$ and with $C(., .)$ as given by (8).

With our new model, the prediction at a new (one-dimensional) noise-free input u_*, leads to a predictive mean and variance, again computed using (14) but with $Q_{ij} = \Sigma'_{ij} + v_t \delta_{ij}$, with Σ'_{ij} computed as (11), and

$$\begin{aligned}k_i &= C(u_*, u_i) + \frac{v_x}{2}\frac{\partial^2 C(u_*, u_i)}{\partial u_i^2} \\ k &= C(u_*, u_*)\end{aligned} \tag{16}$$

thus taking account of the randomness in the training inputs.

In [7] we derived the equations for the predictive mean and variance, for the *usual* GP, when predicting at a new random input. With this new model, the prediction at a random input is straightforward, simply by using the *corrected* covariance function to compute the covariances involving the test input. Assuming $x_* \sim \mathcal{N}(u_*, v_x)$, we have[4]

$$\begin{aligned} k_i &= C(u_i, u_*) + \frac{v_x^2}{4}\frac{\partial^4 C(u_i, u_*)}{\partial u_i^2 \partial u_*^2} + v_x \frac{\partial^2 C(u_i, u_*)}{\partial u_i^2} \\ k &= C(u_*, u_*) + \frac{v_x^2}{4}\frac{\partial^4 C(u_*, u_*)}{\partial u_*^2 \partial u_*^2} + v_x \frac{\partial^2 C(u_*, u_*)}{\partial u_*^2} \; . \end{aligned} \tag{17}$$

4 Illustrative examples

In the following, we assess the goodness of the predictions by computing the average squared error ($L1$) and the average minus log Gaussian predictive density (or minus log-likelihood of the predictions, $L2$).

4.1 Static case assuming v_x is known *a priori*

In this example, the underlying function is such that $y = 2x+3$ for $x < -1$, $y = 1$ for $x \in [-1, 0[$ and $y = \exp(x^2)$ for $x \geq 0$.

Assuming prior knowledge of the w, v and v_t parameters,[5] we consider the case (b) in Figure 1. Given $N = 10$ noise-free inputs and targets, that are known to be functions of the noise-free inputs corrupted with white noise with variance v_x, we compare the predictive means and variances at 300 noise-free test inputs computed when using the *usual* GP (i.e., using (15)) and when using the *corrected* GP assuming v_x is known (using (16)). Figure 2 (left) shows some of the training data used (circles), the noise-free inputs and corresponding targets (squares), the input distribution and the noisy inputs used to generate the training targets (lines) for $v_x = 0.1$. On the right hand side, the evolution of the losses $L1$ and $L2$ is plotted as v_x increases.[6]

As v_x increases, the predictive means computed when using the *corrected* GP become smoother, compared to those obtained using the *usual* GP. Also, the predictive variances in the *usual* case do not increase as v_x does, rendering the model overconfident. Figure 3 shows the predictive means and variance (left) when $v_x = 0.1$. Also shown, the percentage of model ouputs, function of the noisy input, falling in the 95% confidence interval for each test input (right).

[4] Note that we could easily consider a test input with a variance different from that of the training points, assuming that we had knowledge of it *a priori*.

[5] These parameters were actually found by Maximum Likelihood using 300 “clean” data pairs, i.e., noise-free inputs and corresponding outputs, corrupted by a white noise with variance $v_t = 10^{-4}$.

[6] In computing the losses, the mean predictions are compared to the function outputs.

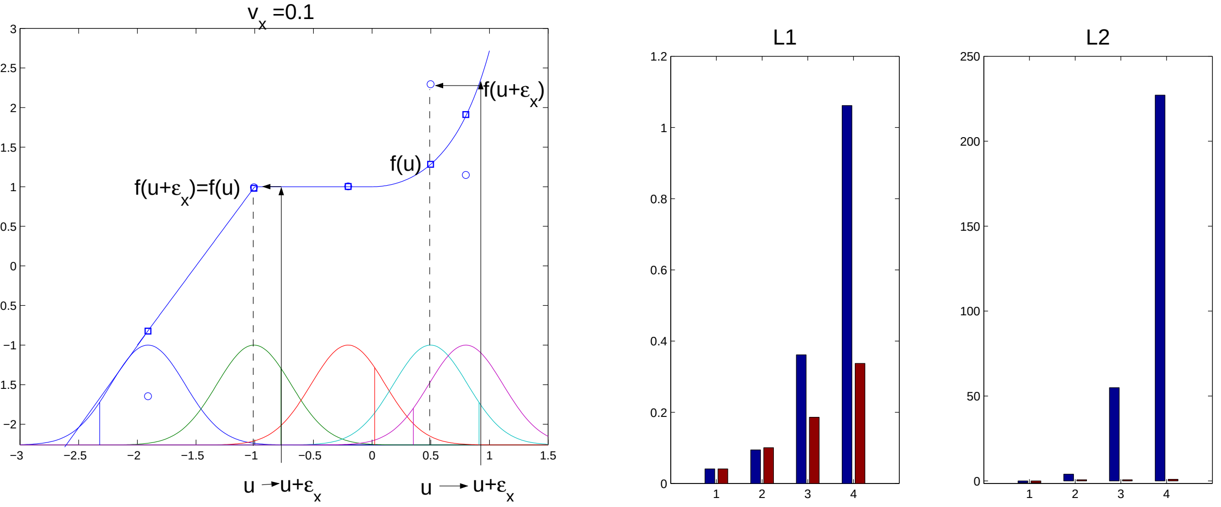


Figure 2: Left: The training data (circles) consist of noise-free inputs / targets corresponding to noisy inputs. Also shown are the noise-free inputs with their corresponding targets (squares) and the input distribution for $v_x = 0.1$. Note that in the region where the function varies, the output computed at the noisy input differs from that corresponding to the noise-free input, misleading a model which does not account for the input noise. Right: Average squared error (left) and average minus log Gaussian predictive density (right) as v_x increases: $v_x = 10^{-4}, 10^{-2}, 10^{-1}, 0.25$. Left bars: *usual* GP, right bars: *corrected* GP.

Now, using $N = 300$ data pairs, we compare the learning of the w, v and v_t parameters when using the *usual* GP, with covariance function given by (8) and the *corrected* one, using (11), again assuming v_x is known. Table 1 gives the Maximum Likelihood (ML) parameters found in each case, along with the losses computed after predicting at the N noise-free inputs.

From this experiment we can conclude that although ignoring the randomness in the input does not lead to a poor predictive mean, the predictive variances are small and the model is far too confident; thus leading to a large $L2$ (Figure 2, right). If we compare the parameters learnt using the noisy data compared to those obtained when learning with clean data (we had $w = 15.77$, $v = 1.2013$ and $v_t = 0.0001$), we see that the "extra" noise, induced by the random input, is "explained" by having a much smaller w (i.e., function varying more slowly) and larger v (controlling the vertical scale of variation) and v_t (estimate of the output noise variance). Also, the w and the v parameters that are learnt using the *corrected* GP are different from those learnt using the *usual* GP, showing the "correlations" that exist between w, v and v_x, thus leading to a multimodal likelihood function.

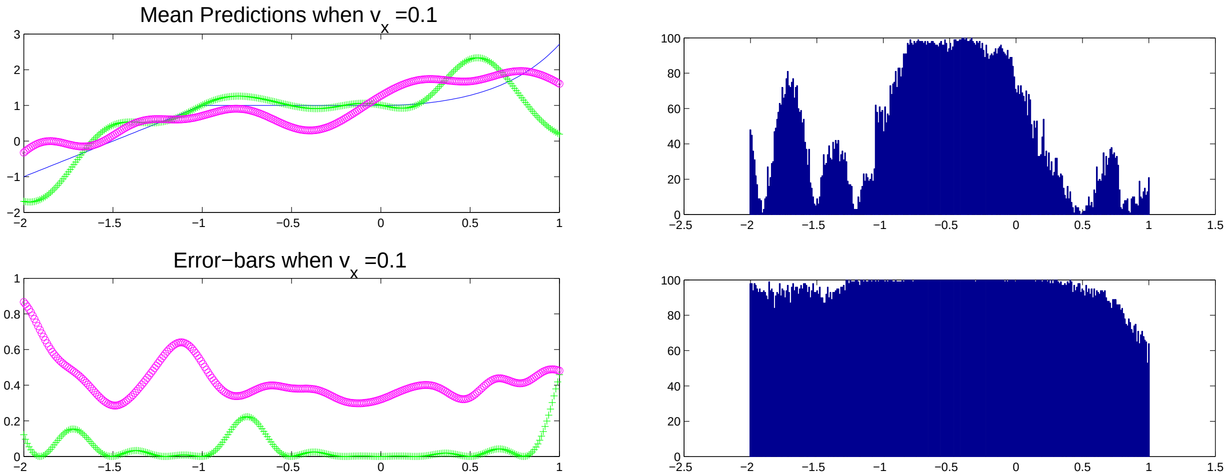


Figure 3: Left: Mean predictions (top) and 2σ error-bars (bottom) when computed using the *usual* GP (crosses) and the *corrected* one (circles), when $v_x = 0.1$. Right: Percentage of model outputs, function of the noisy test inputs, falling in the 95% confidence interval (i.e., within the $+/-2\sigma$ bounds) when using the *usual* GP (top) and the *corrected* one (bottom).

Table 1: Parameters found by ML and losses

	w	v	v_t	$L1$	$L2$
$v_x = 0.01$					
usual GP	1.9318	3.9031	0.0343	0.0022	-1.5258
corrected GP	2.0142	3.9853	0.0343	0.0016	-1.8381
$v_x = 0.1$					
usual GP	1.0131	12.4087	0.6471	0.0972	1.5704
corrected GP	0.9694	18.5297	0.6485	0.0422	-0.2122

4.2 Application to the modelling of the noisy logistic map

We now apply our approach to the modelling of the logistic map (as in [8]), corrupted with white noise with variance $v_t = 0.01$. Let $t_1, \dots, t_N$ be the time series. We assume a state-space model of the form $t_i = f(t_{i-1}) + \epsilon_t$, where the state is formed of one delayed output (case (a) Figure 1). In order to use our new model, we need to replace v_x by v_t in the relevant equations, since in this case the "input noise" is the same as the output noise.

We generate $N = 100$ points for training and another 100 points for testing. Again, we compare the training when ignoring the noise on the state (*usual* GP) and when using the *corrected* covariance function, assuming the noise variance is known, and when learning it, respectively. After optimization of the log-likelihood, we find

- *Usual* GP: $w = 18.3806$, $v = 0.3941$, $v_t = 0.0182$,
- *Corrected* GP, assuming v_t is known: $w = 33.0103$, $v = 0.6719$,
- *Corrected* GP, learning v_t: $w = 26.4954$, $v = 0.4808$, $v_t = 0.0181$

Figure 4 (left) shows the mean predictions with their $2\sigma + v_t$ error bars (with the appropriate v_t for each case) when having used the *usual* GP (top) and the *corrected* GP with v_t known (middle) and with v_t learnt (bottom). Note that the predictions were computed using (17), when using the *corrected* GP, and the equations derived in [7] when using the *usual* GP, since the test inputs are known to be random (with variance v_t).

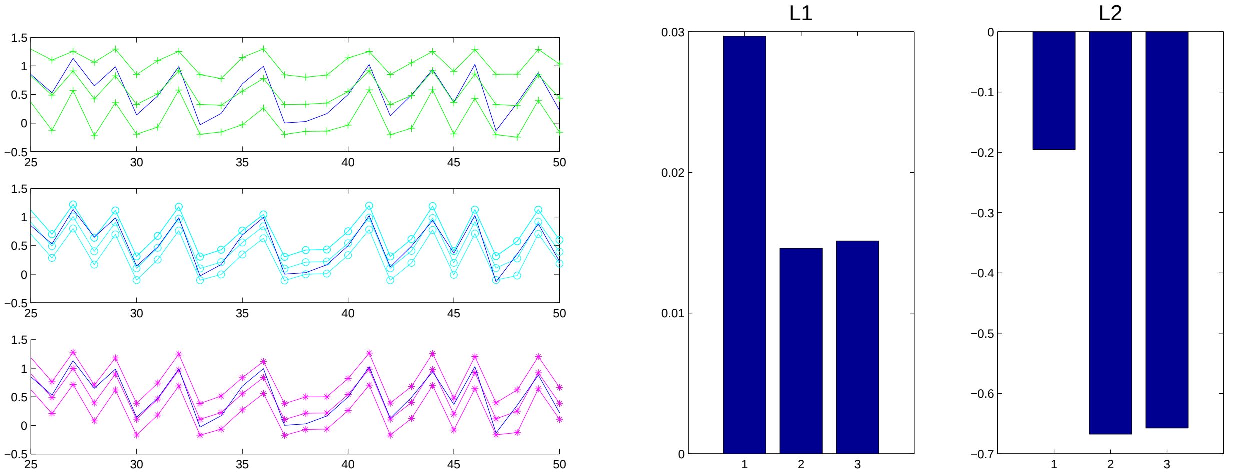


Figure 4: Left: Mean predictions with their $2\sigma + v_t$ error bars, along with the true time-series, in the *usual* case (top) and in the *corrected* case when v_t known (middle) and v_t learnt (bottom). Right: Losses, average over 100 points, for the *usual* case (left bar), the *corrected* case when v_t known (midlle) and v_t learnt (right).

From this experiment, it is clear that accounting for the noise in the input in the covariance function does improve the predictions. Again, we notice the “correlation” between w and the noise level when using the *corrected* covariance function.

5 Conclusions

We have presented a novel approach for the training of a Gaussian Process when the output is a function of a random input. The new *corrected* process is based on an approximation of the random function around the input mean. We have shown that this new model surpasses the usual GP in the case of modelling with noisy inputs in a simple static case, as well as when applied to the modelling of a nonlinear noisy time-series.

Both examples show that accounting for the randomness in the input improves the predictions, via a *corrected* covariance function. They also highlight the correlation between all the parameters, indicating the multimodal nature of the likelihood function, and the potential problems for maximum likelihood optimization. Use of numerical Bayesian approaches via MCMC, and putting priors on v_x, will ameliorate these issues.

Acknowledgements

Many thanks to Professor Mike Titterington and Jian Qing Shi for very useful discussions on the matter. The authors gratefully acknowledge the support of the *Multi-Agent Control* Research Training Network - EC TMR grant HPRN-CT-1999-00107 and RM-S is grateful for EPSRC grant *Modern statistical approaches to off-equilibrium modelling for nonlinear system control* GR/M76379/01.

References

[1] P. Dellaportas and D. A. Stephens. Bayesian analysis of error-in-variables regression models. *Biometrics*, 51, 1995.

[2] V. Tresp, S. Ahmad, and R. Neuneier. Training neural networks with deficient data. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 128–135. Morgan Kaufmann Publishers, Inc., 1994.

[3] V. Tresp and R. Hofmann. Missing and noisy data in nonlinear time-series prediction. In S. F. Girosi, J. Mahoul, E. Manolakos, and E. Wilson, editors, *Neural Networks for Signal Processing*. IEEE Signal Processing Society, 1995.

[4] Z. Ghahramani and M. I. Jordan. Supervised learning from incomplete data via the em approach. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6. San Mateo, CA: Morgan Kaufmann, 1994.

[5] C. E. Rasmussen. *Evaluation of Gaussian Processes and other methods for non-linear regresion*. PhD thesis, University of Toronto, 1996.

[6] V. S. Pugachev. *Theory of random functions and its application to control problems*. Pergamon Press, 1967.

[7] A. Girard, C. Rasmussen, J. Quinonero Candela, and R. Murray-Smith. Gaussian process priors with uncertain inputs – application to multiple-step ahead time series forecasting. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003.

[8] V. Tresp and R. Hofmann. Nonlinear time-series prediction with missing and noisy data. *Neural Computation*, 10(3):731–747, 1998.