# Gaussian Process Regression Networks

**Andrew Gordon Wilson** AGW38@CAM.AC.UK
**David A. Knowles** DAK33@CAM.AC.UK
**Zoubin Ghahramani** ZOUBIN@ENG.CAM.AC.UK
Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK

## Abstract

We introduce a new regression framework, Gaussian process regression networks (GPRN), which combines the structural properties of Bayesian neural networks with the nonparametric flexibility of Gaussian processes. GPRN accommodates input (predictor) dependent signal and noise correlations between multiple output (response) variables, input dependent length-scales and amplitudes, and heavy-tailed predictive distributions. We derive both elliptical slice sampling and variational Bayes inference procedures for GPRN. We apply GPRN as a multiple output regression and multivariate volatility model, demonstrating substantially improved performance over eight popular multiple output (multi-task) Gaussian process models and three multivariate volatility models on real datasets, including a 1000 dimensional gene expression dataset.

## 1. Introduction

"Learning representations by back-propagating errors" by Rumelhart et al. (1986) is a defining paper in machine learning history. This paper made neural networks popular for their ability to capture correlations between multiple outputs, and to discover hidden features in data, by using adaptive hidden basis functions that were shared across the outputs.

MacKay (1992) and Neal (1996) later showed that no matter how large or complex the neural network, one could avoid overfitting using a Bayesian formulation. Neal (1996) also argued that "limiting complexity is likely to conflict with our prior beliefs, and can therefore only be justified to the extent that it is necessary for computational reasons". Accordingly, Neal (1996) pursued the limit of large models, and found that Bayesian neural networks became Gaussian processes as the number of hidden units approached infinity, and conjectured that "there may be simpler ways to do inference in this case".

These simple inference techniques became the cornerstone of subsequent Gaussian process models (Rasmussen & Williams, 2006). These models assume a prior directly over functions, rather than parameters. By further assuming homoscedastic Gaussian noise, one can analytically infer a posterior distribution over these functions, given data. The properties of these functions – smoothness, periodicity, etc. – can easily be controlled by a Gaussian process covariance kernel. Gaussian process models have recently become popular for non-linear regression and classification (Rasmussen & Williams, 2006), and have impressive empirical performances (Rasmussen, 1996).

However, a neural network allowed for correlations between multiple outputs, through sharing adaptive hidden basis functions across the outputs. In the infinite limit of basis functions, these correlations vanished. Moreover, neural networks were envisaged as intelligent agents which discovered hidden features and representations in data, while Gaussian processes, though effective at regression and classification, are simply smoothing devices (MacKay, 1998).

Recently there has been an explosion of interest in extending the Gaussian process regression framework to account for *fixed* correlations between output variables (Alvarez & Lawrence, 2011; Yu et al., 2009; Bonilla et al., 2008; Teh et al., 2005; Boyle & Frean, 2004). These are often called 'multi-task' learning or 'multiple output' regression models. Capturing correlations between outputs (responses) can be used to make better predictions. Imagine we wish to predict cadmium concentrations in a region of the Swiss Jura, where geologists are interested in heavy metal concentrations. A standard Gaussian process regression model would only be able to use cadmium training measurements.

Appearing in *Proceedings of the 29$^{th}$ International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

With a multi-task method, we can also make use of correlated heavy metal measurements to enhance cadmium predictions (Goovaerts, 1997). We could further enhance predictions if we could use how these (signal) correlations change with geographical location.

There has similarly been great interest in extending Gaussian process (GP) regression to account for input dependent noise variances (Goldberg et al., 1998; Kersting et al., 2007; Adams & Stegle, 2008; Turner, 2010; Wilson & Ghahramani, 2010b;a; Lázaro-Gredilla & Titsias, 2011). Wilson & Ghahramani (2010a; 2011) and Fox & Dunson (2011) further extended the GP framework to accommodate input dependent noise correlations between multiple output (response) variables.

In this paper, we introduce a new regression framework, Gaussian Process Regression Networks (GPRN), which combines the structural properties of Bayesian neural networks with the nonparametric flexibility of Gaussian processes. This network is an adaptive mixture of Gaussian processes, which naturally accommodates input dependent signal and noise correlations between multiple output variables, input dependent length-scales and amplitudes, and heavy tailed predictive distributions, without expensive or numerically unstable computations. The GPRN framework extends and unifies the work of Journel & Huijbregts (1978), Neal (1996), Gelfand et al. (2004), Teh et al. (2005), Adams & Stegle (2008), Turner (2010), and Wilson & Ghahramani (2010b; 2011).

Throughout this text we assume we are given a dataset of input output pairs, $\mathcal{D} = \{(x_i, \boldsymbol{y}(x_i)) : i = 1, \ldots, N\}$, where $x \in \mathcal{X}$ is an input (predictor) variable belonging to an arbitrary set $\mathcal{X}$, and $\boldsymbol{y}(x)$ is the corresponding $p$ dimensional output; each element of $\boldsymbol{y}(x)$ is a one dimensional output (response) variable, for example the concentration of a single heavy metal at a geographical location $x$. We aim to predict $\boldsymbol{y}(x_*)|x_*, \mathcal{D}$ and $\Sigma(x_*) = \mathrm{cov}[\boldsymbol{y}(x_*)|x_*, \mathcal{D}]$ at a test input $x_*$, while accounting for input dependent signal and noise correlations between the elements of $\boldsymbol{y}(x)$.

We start by introducing the GPRN framework and discussing inference. We then further discuss related work, before comparing to eight multiple output GP models, on gene expression and geostatistics datasets, and three multivariate volatility models on several benchmark financial datasets. In the supplementary material (Wilson & Ghahramani, 2012) we further discuss theoretical aspects of GPRN, and review GP regression and notation (Rasmussen & Williams, 2006).

## 2. Gaussian Process Networks

We wish to model a $p$ dimensional function $\boldsymbol{y}(x)$, with signal and noise correlations that vary with $x$.

We model $\boldsymbol{y}(x)$ as

$$\boldsymbol{y}(x) = W(x)[\boldsymbol{f}(x) + \sigma_f \boldsymbol{\epsilon}] + \sigma_y \boldsymbol{z}\,, \tag{1}$$

where $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}(x)$ and $\boldsymbol{z} = \boldsymbol{z}(x)$ are respectively $\mathcal{N}(0, I_q)$ and $\mathcal{N}(0, I_p)$ white noise processes. $I_q$ and $I_p$ are $q \times q$ and $p \times p$ dimensional identity matrices. $W(x)$ is a $p \times q$ matrix of independent Gaussian processes such that $W(x)_{ij} \sim \mathcal{GP}(0, k_w)$, and $\boldsymbol{f}(x) = (f_1(x), \ldots, f_q(x))^\top$ is a $q \times 1$ vector of independent GPs with $f_i(x) \sim \mathcal{GP}(0, k_{f_i})$. The GPRN prior on $\boldsymbol{y}(x)$ is induced through GP priors in $W(x)$ and $\boldsymbol{f}(x)$, and the noise model is induced through $\boldsymbol{\epsilon}$ and $\boldsymbol{z}$.

We represent the *Gaussian process regression network* (GPRN)[1] of equation (1) in Figure 1. Each of the latent Gaussian processes in $\boldsymbol{f}(x)$ has additive Gaussian noise. Changing variables to include the noise $\sigma_f \boldsymbol{\epsilon}$, we let $\hat{f}_i(x) = f_i(x) + \sigma_f \epsilon \sim \mathcal{GP}(0, k_{\hat{f}_i})$, where

$$k_{\hat{f}_i}(x_a, x_w) = k_{f_i}(x_a, x_w) + \sigma_f^2 \delta_{aw}\,, \tag{2}$$

and $\delta_{aw}$ is the Kronecker delta. The latent *node functions* $\hat{\boldsymbol{f}}(x)$ are connected together to form the outputs $\boldsymbol{y}(x)$. The strengths of the connections change as a function of $x$; the weights themselves – the entries of $W(x)$ – are functions. Old connections can break and new connections can form. This is an *adaptive* network, where the signal and noise correlations between the components of $\boldsymbol{y}(x)$ vary with $x$. We label the length-scale hyperparameters for the kernels $k_w$ and $k_{f_i}$ as $\boldsymbol{\theta}_w$ and $\boldsymbol{\theta}_f$ respectively. We often assume that all the weight GPs share the same covariance kernel $k_w$, including hyperparameters. Roughly speaking, sharing length-scale hyperparameters amongst the weights means that, a priori, the strengths of the connections in Figure 1 vary with $x$ at the same rate.

To explicitly separate the adaptive signal and noise correlations, we re-write (1) as

$$\boldsymbol{y}(x) = \underbrace{W(x)\boldsymbol{f}(x)}_{\text{signal}} + \underbrace{\sigma_f W(x)\boldsymbol{\epsilon} + \sigma_y \boldsymbol{z}}_{\text{noise}}\,. \tag{3}$$

Given $W(x)$, each of the outputs $\boldsymbol{y}_i(x)$, $i = 1, \ldots, p$, is a Gaussian process with kernel

$$k_{y_i}(x_a, x_w) = \sum_{j=1}^{q} W_{ij}(x_a) k_{\hat{f}_j}(x_a, x_w) W_{ij}(x_w) + \delta_{aw}\sigma_y^2\,. \tag{4}$$

[1] Coincidentally, there is an unrelated paper called "Gaussian process networks" (Friedman & Nachman, 2000), which is about learning the structure of Bayesian networks – e.g. the direction of dependence between random variables.

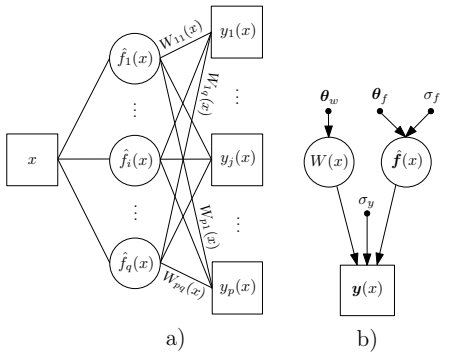


*Figure 1.* The Gaussian process regression network. Latent random variables and observables are respectively labelled with circles and squares, except for the weight functions in a). Hyperparameters are labelled with dots. a) This neural network style diagram shows the $q$ components of the vector $\hat{\boldsymbol{f}}$ (GPs with additive noise), and the $p$ components of the vector $\boldsymbol{y}$. The links in the graph, four of which are labelled, are latent random weight *functions*. Every quantity in this graph depends on the input $x$. This graph emphasises the adaptive nature of this network: links can change strength or even disappear as $x$ changes. b) A directed graphical model showing the generative procedure with relevant variables.

The components of $\boldsymbol{y}(x)$ are coupled through the matrix $W(x)$. Training the network involves conditioning $W(x)$ on the data $\mathcal{D}$, and so the predictive covariances of $\boldsymbol{y}(x_*)|\mathcal{D}$ are now influenced by the values of the observations, and not just distances between the test point $x_*$ and the observed points $x_1, \ldots, x_N$ as is the case for independent GPs.

We can view (4) as an adaptive kernel learned from the data. There are several other interesting features in equation (4): 1) the amplitude of the covariance function, $\sum_{j=1}^{q} W_{ij}(x)W_{ij}(x')$, is non-stationary (input dependent); 2) even if each of the kernels $k_{f_j}$ has different *stationary* length-scales, the mixture of the kernels $k_{f_j}$ is input dependent and so the effective overall length-scale is non-stationary; 3) the kernels $k_{f_j}$ may be entirely different: some may be periodic, others squared exponential, others Brownian motion, and so on. Therefore the overall covariance kernel may be continuously switching between regions of entirely different covariance structures.

In addition to modelling signal correlations, we can see from equation (3) that the GPRN is also a multivariate volatility model. The noise covariance is $\sigma_f^2 W(x)W(x)^\top + \sigma_y^2 I_p$. Since the entries of $W(x)$ are GPs, this noise model is an example of a *generalised Wishart process* (Wilson & Ghahramani, 2010a; 2011).

The number of nodes $q$ influences how the model accounts for signal and noise correlations. If $q$ is smaller than $p$, the dimension of $\boldsymbol{y}(x)$, the model performs dimensionality reduction and matrix factorization as part of the regression on $\boldsymbol{y}(x)$ and $\mathrm{cov}[\boldsymbol{y}(x)]$. However, we may want $q > p$, for instance if the output space were one dimensional ($p = 1$). In this case we would need $q > 1$ for nonstationary length-scales and covariance structures. For a given dataset, we can vary $q$ and select the value which gives the highest marginal likelihood on training data.

## 3. Inference

We have specified a prior $p(\boldsymbol{y}(x))$ at all points $x$ in the domain $\mathcal{X}$, and a noise model, so we can infer the posterior $p(\boldsymbol{y}(x)|\mathcal{D})$. The prior on $\boldsymbol{y}(x)$ is induced through the GP priors in $W(x)$ and $\boldsymbol{f}(x)$, and the parameters $\boldsymbol{\gamma} = \{\boldsymbol{\theta}_f, \boldsymbol{\theta}_w, \sigma_f, \sigma_y\}$. We perform inference directly over the GPs and parameters.

We explicitly re-write the prior over GPs in terms of $\boldsymbol{u} = (\hat{\mathbf{f}}, \mathbf{W})$, a vector composed of all the node and weight Gaussian process functions, evaluated at the training points $\{x_1, \ldots, x_N\}$. There are $q$ node functions and $p \times q$ weight functions. Therefore

$$p(\boldsymbol{u}|\sigma_f, \boldsymbol{\theta}_f, \boldsymbol{\theta}_w) = \mathcal{N}(0, C_B)\,, \tag{5}$$

where $C_B$ is an $Nq(p+1) \times Nq(p+1)$ block diagonal matrix, since the weight and node functions are a priori independent. We order the entries of $\boldsymbol{u}$ so that the first $q$ blocks are $N \times N$ covariance matrices $K_{\hat{f}_i}$ from the node kernels $k_{\hat{f}_i}$, and the last blocks are $N \times N$ covariance matrices $K_w$ from the weight kernel $k_w$.

From (1), the likelihood is

$$p(\mathcal{D}|\boldsymbol{u}, \sigma_y) = \prod_{i=1}^{N} \mathcal{N}(\boldsymbol{y}(x_i); W(x_i)\hat{\boldsymbol{f}}(x_i), \sigma_y^2 I_p)\,. \tag{6}$$

Applying Bayes' theorem,

$$p(\boldsymbol{u}|\mathcal{D}, \boldsymbol{\gamma}) \propto p(\mathcal{D}|\boldsymbol{u}, \sigma_y)p(\boldsymbol{u}|\sigma_f, \boldsymbol{\theta}_f, \boldsymbol{\theta_w})\,. \tag{7}$$

We sample from the posterior in (7) using elliptical slice sampling (ESS) (Murray et al., 2010), which is specifically designed to sample from posteriors with strongly correlated Gaussian priors. For comparison we approximate (7) using a message passing implementation of variational Bayes (VB). We also use VB to learn the hyperparameters $\boldsymbol{\gamma}|\mathcal{D}$. Details about our ESS and VB approaches are in Sections 1 and 2 of the supplementary material.

By incorporating noise on $\boldsymbol{f}$, the GP network accounts for input dependent noise correlations (as in (3)), without the need for costly or numerically unstable matrix decompositions during inference. The matrix $\sigma_y^2 I_p$

does not change with $x$ and requires only one $\mathcal{O}(1)$ operation to invert. In a more typical multivariate volatility model, one must decompose a $p \times p$ matrix $\Sigma(x)$ once for each datapoint $x_i$ ($N$ times in total), an $\mathcal{O}(Np^3)$ operation which is prone to numerical instability. In general, multivariate volatility models are intractable for $p > 5$ (Gouriéroux et al., 2009; Engle, 2002). Moreover, multi-task Gaussian process models typically have an $\mathcal{O}(N^3p^3)$ complexity (Alvarez & Lawrence, 2011). In Section 3 of the supplementary material (Wilson & Ghahramani, 2012) we show that, fixing the number of ESS or VB iterations, GPRN inference scales linearly with $p$, and further discuss theoretical properties of GPRN, like the heavy-tailed predictive distribution.

## 4. Related Work

Gaussian process regression networks are related to a large body of seemingly disparate work in machine learning, econometrics, geostatistics, physics, and probability theory.

In machine learning, the semiparametric latent factor model (SLFM) (Teh et al., 2005) was introduced to model multiple outputs with fixed signal correlations. SLFM specifies a linear mixing of latent Gaussian processes. The SLFM is similar to the linear model of coregionalisation (LMC) (Journel & Huijbregts, 1978) and intrinsic coregionalisation model (ICM) (Goovaerts, 1997) in geostatistics, but the SLFM incorporates important Gaussian process hyperparameters like length-scales, and methodology for learning these hyperparameters. In machine learning, the SLFM has also been developed as "Gaussian process factor analysis" (Yu et al., 2009), with an emphasis on time being the input (predictor) variable.

For changing correlations, the Wishart process (Bru, 1991) was first introduced in probability theory as a distribution over a collection of positive definite covariance matrices with Wishart marginals. It was defined as an outer product of autoregressive Gaussian processes restricted to a Brownian motion or Ornstein-Uhlenbeck covariance structure. In the geostatistics literature, Gelfand et al. (2004) applied a Wishart process as part of a linear coregionalisation model with spatially varying signal covariances, on a $p = 2$ dimensional real-estate example. Later Gouriéroux et al. (2009) returned to the Wishart process of Bru (1991) to model multivariate volatility, letting the noise covariance be specified as an outer product of AR(1) Gaussian processes, assuming that the covariance matrices $\Sigma(t) = \text{cov}(\boldsymbol{y}|t)$ are observables on an evenly spaced one dimensional grid. In machine learning, Wilson & Ghahramani (2010a; 2011) introduced the generalised Wishart process (GWP), which generalises the Wishart process of (Bru, 1991) to a process over arbitrary positive definite matrices (Wishart marginals are not required) with a flexible covariance structure, and using the GWP, extended the GP framework to account for input dependent noise correlations (multivariate volatility), without assuming the noise is observable, or that the input space is 1D, or on a grid.

Gaussian process regression networks act as both a multi-task and multivariate volatility model. The signal correlation model in GPRN differs from Gelfand et al. (2004) in that 1) the GPRN incorporates and estimates Gaussian process hyperparameters, like length-scales, effectively learning aspects of the covariance structure from data, 2) is tractable for $p > 3$, 3) is used as a latent factor model (where $q < p$), 4) can account for input dependent length-scales and covariance structures, and 5) incorporates an input dependent noise correlation model. Moreover, the VB and ESS inference procedures we present here are significantly more efficient than the Metropolis-Hastings proposals in Gelfand et al. (2004). Generally a noise model strongly influences a regression on the signal, even if the noise and signal models are a priori independent. In the GPRN prior of equation (3) the noise and signal correlations are explicitly related: through sharing $W(x)$, the signal and noise are encouraged to increase and decrease together. The noise model is an example of a GWP, although GPRN scales linearly and not cubically with $p$, per iteration of ESS or VB. If the GPRN is exposed solely to input dependent noise, the length-scales on the node functions $\boldsymbol{f}(x)$ will train to large values, turning the GPRN into solely a multivariate volatility model: all the modelling then takes place in $W(x)$. In other words, through learning Gaussian process hyperparameters, GPRN can automatically vary between a multi-task and multivariate volatility model. The hyperparameters in GPRN are also important for distinguishing between the behaviour of the weight and node functions. We may expect, for example, that the node functions will vary more quickly than the weight functions, so that the components of $\boldsymbol{y}(x)$ vary more quickly than the correlations between the components of $\boldsymbol{y}(x)$. The rate at which the node and weight functions vary is controlled by the Gaussian process length-scale hyperparameters, which are learned from data.

When $q = p = 1$, the GPRN resembles the nonstationary GP regression model of Adams & Stegle (2008). Likewise, when the weight functions are constants, the GPRN becomes the semiparametric latent factor model (SLFM) of Teh et al. (2005), except that the resulting GP regression network is less prone to overfitting through its use of full Bayesian inference. The

GPRN also somewhat resembles the natural sound model (MPAD) in Section 5.3 of Turner (2010), except in MPAD the analogue of the node functions are AR(2) Gaussian processes, and the "weight functions" are a priori correlated.

Ver Hoef & Barry (1998) in geostatistics and Boyle & Frean (2004) in machine learning proposed an alternative convolution GP model for multiple outputs (CMOGP) with fixed signal correlations, where each output at each $x \in \mathcal{X}$ is a mixture of latent Gaussian processes mixed across the whole input domain $\mathcal{X}$.

## 5. Experiments

We compare GPRN to multi-task learning and multivariate volatility models. We also compare between variational Bayes (VB) and elliptical slice sampling (ESS) inference within the GPRN framework. In the multi-task setting, there are $p$ dimensional observations $\boldsymbol{y}(x)$, and the goal is to use the correlations between the elements of $\boldsymbol{y}(x)$ to make better predictions of $\boldsymbol{y}(x_*)$, for a test input $x_*$, than if we were to treat the dimensions independently. A major difference between GPRN and alternative multi-task models is that the GPRN accounts for signal correlations that *change* with $x$, and input dependent noise correlations, rather than fixed correlations. We compare to multi-task GP models on gene expression and geostatistics datasets.

In the multi-task experiments, the GPRN accounts for both input dependent signal and noise covariance matrices. To specifically test GPRN's ability to model input dependent noise covariances (multivariate volatility), we compare predictions of $\text{cov}[\boldsymbol{y}(x)] = \Sigma(x)$ to those made by popular multivariate volatility models on benchmark financial datasets.

In all experiments, GPRN uses squared exponential covariance functions, with a length-scale shared across all node functions, and another length-scale shared across all weight functions. GPRN is robust to initialisation. We use an adversarial initialisation of $\mathcal{N}(0, 1)$ white noise for Gaussian process functions.

### 5.1. Gene Expression

Tomancak et al. (2002) measured gene expression levels every hour for 12 hours during Drosophila embryogenesis; they then repeated this experiment for an independent replica (a second independent time series). Gene expression is activated and deactivated by transcription factor proteins. We focus on genes which are thought to at least be regulated by the transcription factor `twi`, which influences mesoderm and muscle development in Drosophila (Zinzen et al., 2009). The assumption is that these gene expression levels are all correlated. We would like to use how these correlations change over time to make better predictions of time varying gene expression in the presence of transcription factors. In total there are 1621 genes (outputs) at $N = 12$ time points (inputs), on two independent replicas. For training, $p = 50$ random genes were selected from the first replica, and the corresponding 50 genes in the second replica were used for testing. We then repeated this experiment 10 times with a different set of genes each time, and averaged the results. We then repeated the whole experiment, but with $p = 1000$ genes. We used exactly the same training and testing sets as Alvarez & Lawrence (2011).

We use a smaller $p = 50$ dataset so that we are able to compare with popular alternative multi-task methods (LMC, CMOGP, SLFM), which have a complexity of $\mathcal{O}(N^3p^3)$ and would not scale to $p = 1000$ (Alvarez & Lawrence, 2011).[2] For $p = 1000$, we compare to the sparse convolved multiple output GP methods (CMOFITC, CMODTC, and CMOPITC) of Alvarez & Lawrence (2011). In both of these regressions, the GPRN is accounting for multivariate volatility; this is the first time a multivariate stochastic volatility model has been estimated for $p > 50$ (Chib et al., 2006). We assess performance using standardised mean square error (SMSE) and mean standardized log loss (MSLL), as defined in Rasmussen & Williams (2006) on page 23. Using the empirical mean and variance to fit the data would give an SMSE and MSLL of 1 and 0 respectively. The smaller the SMSE and more negative the MSLL the better.

The results are in Table 1, under the headings `GENE` (50D) and `GENE` (1000D). For `SET 2` we reverse training and testing replicas in `SET 1`. GPRN outperforms all of the other models, with between 46% and 68% of the SMSE, and similarly strong results on the MSLL error metric.[3] On both the 50 and 1000 dimensional datasets, the marginal likelihood for the network structure is sharply peaked at $q = 1$, as we might expect since there is likely one transcription factor `twi` controlling the expression levels of the genes in question.

Typical GPRN (VB) runtimes for the 50D and 1000D datasets were respectively 12 seconds and 330 seconds. These runtimes scale roughly linearly with dimension ($p$), which is what we expect. GPRN (VB) runs at about the same speed as the sparse CMOGP meth-

[2] We also implemented the SVLMC of Gelfand et al. (2004) but found it intractable on the gene expression and geostatistics datasets, and on a subset of data it gave worse results than the other methods we compare to. SVLMC is not applicable to the multivariate volatility datasets. The supplementary material has more details.

[3] Independent GPs severely overfit on `GENE`, giving an MSLL of $\infty$.

ods, and much faster than CMOGP, LMC and SLFM, which take days to run on the 1000D dataset. The GPRN (ESS) runtimes for the 50D and 1000D datasets were 40 seconds and 9000 seconds (2.5 hr), and required respectively 6000 and $10^4$ samples to reach convergence, as assessed by trace plots of sample likelihoods. In terms of both speed and accuracy GPRN (ESS) outperforms all methods except GPRN (VB). GPRN (ESS) does not mix as well in high dimensions, and the number of ESS iterations required to reach convergence noticeably grows with $p$. However, ESS is still tractable and performing relatively well in $p = 1000$ dimensions, in terms of speed and predictive accuracy. Runtimes are on a 2.3 GHz Intel i5 Duo Core processor.

### 5.2. Jura Geostatistics

Here we are interested in predicting concentrations of cadmium at 100 locations within a 14.5 km$^2$ region of the Swiss Jura. For training, we have access to measurements of cadmium at 259 neighbouring locations. We also have access to nickel and zinc concentrations at these 259 locations, as well as at the 100 locations we wish to predict cadmium. While a standard Gaussian process regression model would only be able to make use of the cadmium training measurements, a multi-task method can use the correlated nickel and zinc measurements to enhance predictions. With GPRN we can also make use of how the correlations between nickel, zinc, and cadmium change with location to further enhance predictions.

The network structure with the highest marginal likelihood has $q = 2$ latent node functions. The node and weight functions learnt using VB for this setting are shown in Figure 1 of the supplementary material (Wilson & Ghahramani, 2012). Since there are $p = 3$ output dimensions, the result $q < p$ suggests that heavy metal concentrations in the Swiss Jura are correlated. Indeed, using our model we can observe the *spatially varying* correlations between heavy metal concentrations, as shown for cadmium and zinc in Figure 2. Although the correlation between cadmium and zinc is generally positive (with values around 0.6), there is a region where the correlations noticeably decrease, perhaps corresponding to a geological structure. The quantitative results in Table 1 suggest that the ability of GPRN to learn these spatially varying correlations is beneficial for predicting cadmium concentrations.

We assess performance quantitatively using mean absolute error (MAE) between the predicted and true cadmium concentrations. We restart the experiment 10 times with different initialisations of the parameters, and average the MAE. The results are marked

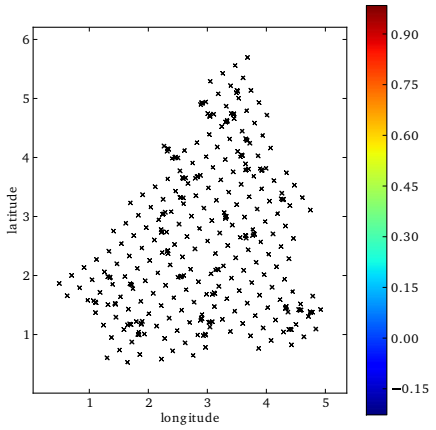


*Figure 2.* Spatially dependent correlation between cadmium and zinc learnt by the GPRN. Markers show the locations where measurements were made.

by JURA in Table 1. The experimental setup follows Goovaerts (1997) and Alvarez & Lawrence (2011). We found log transforming and normalising each dimension to have zero mean and unit variance to be beneficial due to the skewed distribution of the $y$-values (but we also include results on untransformed data, marked with *). All the multiple output methods give lower MAE than using an independent GP, and GPRN outperforms SLFM and the other methods.

For the JURA dataset, the improved performance of GPRN is at the cost of a slightly greater runtime. However, GPRN is accounting for input dependent signal and noise correlations, unlike the other methods. Moreover, the complexity of GPRN scales linearly with $p$ (per iteration), unlike the other methods which scale as $\mathcal{O}(N^3p^3)$. This is why GPRN runs relatively quickly on the 1000 dimensional gene expression dataset, for which the other methods are intractable. These data are available from http://www.ai-geostats.org/.

### 5.3. Multivariate Volatility

In the previous experiments the GPRN implicitly accounted for multivariate volatility (input dependent noise covariances) in making predictions of $\boldsymbol{y}(x_*)$. We now test the GPRN explicitly as a model of multivariate volatility, and assess predictions of $\Sigma(t) = \text{cov}[\boldsymbol{y}(t)]$. We make 200 historical predictions of $\Sigma(t)$ at observed time points, and 200 one day ahead forecasts. Historical predictions can be used, for example, to understand a past financial crisis. The forecasts are assessed using the log likelihood of new observations under the predicted covariance, denoted $\mathcal{L}$ Fore-

cast. We follow Wilson & Ghahramani (2010a), and predict $\Sigma(t)$ for returns on three currency exchanges (EXCHANGE) and five equity indices (EQUITY) processed as in Wilson & Ghahramani (2010a). These datasets are especially suited to MGARCH, the most popular multivariate volatility model, and have become a benchmark for assessing GARCH models (Poon & Granger, 2005; Hansen & Lunde, 2005; Brownlees et al., 2009; McCullough & Renfro, 1998; Brooks et al., 2001). We compare to full BEKK MGARCH (Engle & Kroner, 1995), the generalised Wishart process (Wilson & Ghahramani, 2010a), and the original Wishart process (Bru, 1991; Gouriéroux et al., 2009).

We see in Table 1 that GPRN (ESS) is often outperformed by GPRN (VB) on multivariate volatility sets, suggesting convergence difficulties with ESS. The high historical MSE for GPRN on EXCHANGE is essentially training error, and less meaningful than the encouraging step ahead forecast likelihoods; to harmonize with the econometrics literature, historical MSE for EXCHANGE is between the learnt covariance $\Sigma(x)$ and observed $\mathbf{y}(x)\mathbf{y}(x)^\top$. See Wilson & Ghahramani (2010a) for details. Overall, the GPRN shows promise as both a multi-task and multivariate volatility model, especially since the multivariate volatility datasets are suited to MGARCH. These data were obtained using Datastream (http://www.datastream.com/).

## 6. Discussion

A Gaussian process regression network (GPRN) has a simple and interpretable structure, and generalises many of the recent extensions to the Gaussian process regression framework. The model naturally accommodates input dependent signal and noise correlations between multiple output variables, heavy tailed predictive distributions, input dependent length-scales and amplitudes, and adaptive covariance functions. Furthermore, GPRN has scalable inference procedures, and strong empirical performance on several real datasets.

In the future, it would be enlightening to use GPRN with different types of adaptive covariance structures, particularly in the case where $p = 1$ and $q > 1$; in one dimensional output space it would be easy, for instance, to visualise a process gradually switching between brownian motion, periodic, and smooth covariance functions. It would also be interesting to apply this adaptive network to classification, or to use a GPRN where the weight functions depend on a different set of variables than the node functions. We hope the GPRN will inspire further research into adaptive networks, and further connections between different areas of machine learning and statistics.

*Table 1.* Comparative performance on all datasets.

| GENE (50D) | Average SMSE | Average MSLL |
|---|---|---|
| SET 1: | | |
| GPRN (VB) | $0.3356 \pm 0.0294$ | $\mathbf{-0.5945 \pm 0.0536}$ |
| GPRN (ESS) | $\mathbf{0.3236 \pm 0.0311}$ | $-0.5523 \pm 0.0478$ |
| LMC | $0.6069 \pm 0.0294$ | $-0.2687 \pm 0.0594$ |
| CMOGP | $0.4859 \pm 0.0387$ | $-0.3617 \pm 0.0511$ |
| SLFM | $0.6435 \pm 0.0657$ | $-0.2376 \pm 0.0456$ |
| SET 2: | | |
| GPRN (VB) | $0.3403 \pm 0.0339$ | $\mathbf{-0.6142 \pm 0.0557}$ |
| GPRN (ESS) | $\mathbf{0.3266 \pm 0.0321}$ | $-0.5683 \pm 0.0542$ |
| LMC | $0.6194 \pm 0.0447$ | $-0.2360 \pm 0.0696$ |
| CMOGP | $0.4615 \pm 0.0626$ | $-0.3811 \pm 0.0748$ |
| SLFM | $0.6264 \pm 0.0610$ | $-0.2528 \pm 0.0453$ |
| **GENE (1000D)** | **Average SMSE** | **Average MSLL** |
| SET 1: | | |
| GPRN (VB) | $\mathbf{0.3473 \pm 0.0062}$ | $\mathbf{-0.6209 \pm 0.0085}$ |
| GPRN (ESS) | $0.4520 \pm 0.0079$ | $-0.4712 \pm 0.0327$ |
| CMOFITC | $0.5469 \pm 0.0125$ | $-0.3124 \pm 0.0200$ |
| CMOPITC | $0.5537 \pm 0.0136$ | $-0.3162 \pm 0.0206$ |
| CMODTC | $0.5421 \pm 0.0085$ | $-0.2493 \pm 0.0183$ |
| SET 2: | | |
| GPRN (VB) | $\mathbf{0.3287 \pm 0.0050}$ | $\mathbf{-0.6430 \pm 0.0071}$ |
| GPRN (ESS) | $0.4140 \pm 0.0078$ | $-0.4787 \pm 0.0315$ |
| CMOFITC | $0.5565 \pm 0.0425$ | $-0.3024 \pm 0.0294$ |
| CMOPITC | $0.5713 \pm 0.0794$ | $-0.3128 \pm 0.0138$ |
| CMODTC | $0.5454 \pm 0.0173$ | $0.6499 \pm 0.7961$ |
| **JURA** | **Average MAE** | **Training (secs)** |
| GPRN (VB) | $\mathbf{0.4040 \pm 0.0006}$ | 1040 |
| GPRN* (VB) | $0.4525 \pm 0.0036$ | 1190 |
| SLFM (VB) | $0.4247 \pm 0.0004$ | 614 |
| SLFM* (VB) | $0.4679 \pm 0.0030$ | 810 |
| SLFM | $0.4578 \pm 0.0025$ | 792 |
| Co-kriging | 0.51 | |
| ICM | $0.4608 \pm 0.0025$ | 507 |
| CMOGP | $0.4552 \pm 0.0013$ | 784 |
| GP | $0.5739 \pm 0.0003$ | 74 |
| **EXCHANGE** | **Historical MSE** | **$\mathcal{L}$ Forecast** |
| GPRN (VB) | $3.83 \times 10^{-8}$ | **2073** |
| GPRN (ESS) | $6.120 \times 10^{-9}$ | 2012 |
| GWP | $\mathbf{3.88 \times 10^{-9}}$ | 2020 |
| WP | $\mathbf{3.88 \times 10^{-9}}$ | 1950 |
| MGARCH | $3.96 \times 10^{-9}$ | 2050 |
| **EQUITY** | **Historical MSE** | **$\mathcal{L}$ Forecast** |
| GPRN (VB) | $0.978 \times 10^{-9}$ | 2740 |
| GPRN (ESS) | $\mathbf{0.827 \times 10^{-9}}$ | 2630 |
| GWP | $2.80 \times 10^{-9}$ | **2930** |
| WP | $3.96 \times 10^{-9}$ | 1710 |
| MGARCH | $6.69 \times 10^{-9}$ | 2760 |

## References


Adams, R.P. and Stegle, O. Gaussian process product models for nonparametric nonstationarity. In *ICML*, 2008.

Alvarez, M.A. and Lawrence, N.D. Computationally efficient convolved multiple output gaussian processes. *JMLR*, 12:1425–1466, 2011.

Bonilla, E., Chai, K.M.A., and Williams, C. Multi-task Gaussian process prediction. In *NIPS*, 2008.

Boyle, P. and Frean, M. Dependent Gaussian processes. In *NIPS*, 2004.

Brooks, C., Burke, S.P., and Persand, G. Benchmarks and the accuracy of GARCH model estimation. *International Journal of Forecasting*, 17:45–56, 2001.

Brownlees, Christian T., Engle, Robert F., and Kelly, Bryan T. A practical guide to volatility forecasting through calm and storm, 2009. Available at SSRN: http://ssrn.com/abstract=1502915.

Bru, M.F. Wishart processes. *Journal of Theoretical Probability*, 4(4):725–751, 1991.

Chib, S., Nardari, F., and Shephard, N. Analysis of high dimensional multivariate stochastic volatility models. *Journal of Econometrics*, 134(2):341–371, 2006. ISSN 0304-4076.

Engle, R. New frontiers for ARCH models. *Journal of Applied Econometrics*, 17(5):425–446, 2002. ISSN 1099-1255.

Engle, R.F. and Kroner, K.F. Multivariate simultaneous generalized ARCH. *Econometric theory*, 11(01):122–150, 1995.

Fox, E. and Dunson, D. Bayesian nonparametric covariance regression. *Arxiv preprint arXiv:1101.2017*, 2011.

Friedman, N. and Nachman, I. Gaussian process networks. In *UAI*, pp. 211–219, 2000.

Gelfand, A.E., Schmidt, A.M., Banerjee, S., and Sirmans, CF. Nonstationary multivariate process modeling through spatially varying coregionalization. *Test*, 13(2): 263–312, 2004. ISSN 1133-0686.

Goldberg, Paul W., Williams, Christopher K.I., and Bishop, Christopher M. Regression with input-dependent noise: A Gaussian process treatment. In *NIPS*, 1998.

Goovaerts, P. *Geostatistics for natural resources evaluation*. Oxford University Press, USA, 1997.

Gouriéroux, C., Jasiak, J., and Sufana, R. The Wishart autoregressive process of multivariate stochastic volatility. *Journal of Econometrics*, 150(2):167–181, 2009.

Hansen, Peter Reinhard and Lunde, Asger. A forecast comparison of volatility models: Does anything beat a GARCH(1,1). *Journal of Applied Econometrics*, 20(7): 873–889, 2005.

Journel, AG. and Huijbregts, CJ. *Mining geostatistics*. Academic Press (London and New York), 1978.

Kersting, K., Plagemann, C., Pfaff, P., and Burgard, W. Most likely heteroscedastic gaussian process regression. In *ICML*, 2007.

Lázaro-Gredilla, M. and Titsias, M.K. Variational heteroscedastic gaussian process regression. In *ICML*, 2011.

MacKay, David J.C. Introduction to gaussian processes. In Christopher M. Bishop, editor (ed.), *Neural Networks and Machine Learning*, chapter 11, pp. 133–165. Springer-Verlag, 1998.

MacKay, D.J.C. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

McCullough, B.D. and Renfro, C.G. Benchmarks and software standards: A case study of GARCH procedures. *Journal of Economic and Social Measurement*, 25:59–71, 1998.

Murray, Iain, Adams, Ryan Prescott, and MacKay, David J.C. Elliptical Slice Sampling. *JMLR: W&CP*, 9:541–548, 2010.

Neal, R.M. *Bayesian learning for neural networks*. Springer Verlag, 1996. ISBN 0387947248.

Poon, Ser-Huang and Granger, Clive W.J. Practical issues in forecasting volatility. *Financial Analysts Journal*, 61 (1):45–56, 2005.

Rasmussen, Carl Edward. *Evaluation of Gaussian Processes and Other Methods for Non-linear Regression*. PhD thesis, Dept. of Computer Science, University of Toronto, 1996.

Rasmussen, Carl Edward and Williams, Christopher K.I. *Gaussian processes for Machine Learning*. The MIT Press, 2006.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Teh, Y.W., Seeger, M., and Jordan, M.I. Semiparametric latent factor models. In *AISTATS*, 2005.

Tomancak, P., Beaton, A., Weiszmann, R., Kwan, E., Shu, S., Lewis, S.E., et al. Systematic determination of patterns of gene expression during drosophila embryogenesis. *Genome Biol*, 3(12):0081–0088, 2002.

Turner, Richard E. *Statistical Models for Natural Sounds*. PhD thesis, University College London, 2010.

Ver Hoef, J.M. and Barry, R.P. Constructing and fitting models for cokriging and multivariable spatial prediction. *Journal of Statistical Planning and Inference*, 69 (2):275–294, 1998.

Wilson, Andrew G and Ghahramani, Zoubin. GPRN supplementary material. 2012. http://mlg.eng.cam.ac.uk/andrew/gprnsupp.pdf.

Wilson, Andrew Gordon and Ghahramani, Z. Generalised Wishart Processes. *Arxiv preprint arXiv:1101.0240*, 2010a.

Wilson, Andrew Gordon and Ghahramani, Z. Generalised Wishart Processes. In *UAI*, 2011.

Wilson, Andrew Gordon and Ghahramani, Zoubin. Copula processes. In *NIPS*, 2010b.

Yu, Byron M, Cunningham, John P, Santhanam, Gopal, Ryu, Stephen I, Shenoy, Krishna V, and Sahani, Maneesh. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In *NIPS*, 2009.

Zinzen, R.P., Girardot, C., Gagneur, J., Braun, M., and Furlong, E.E.M. Combinatorial binding predicts spatio-temporal cis-regulatory activity. *Nature*, 462(7269):65–70, 2009.