GPz: non-stationary sparse Gaussian processes for heteroscedastic uncertainty estimation in photometric redshifts

Ibrahim A. Almosallam,[1,2]★ Matt J. Jarvis[3,4] and Stephen J. Roberts[2]

[1]*King Abdulaziz City for Science and Technology, Riyadh 1142, Saudi Arabia*
[2]*Information Engineering, Parks Road, Oxford OX1 3PJ, UK*
[3]*Department of Physics, Oxford Astrophysics, Keble Road, Oxford OX1 3RH, UK*
[4]*Department of Physics, University of the Western Cape, Bellville 7535, South Africa*

ABSTRACT

The next generation of cosmology experiments will be required to use photometric redshifts rather than spectroscopic redshifts. Obtaining accurate and well-characterized photometric redshift distributions is therefore critical for *Euclid*, the Large Synoptic Survey Telescope and the Square Kilometre Array. However, determining accurate variance predictions alongside single point estimates is crucial, as they can be used to optimize the sample of galaxies for the specific experiment (e.g. weak lensing, baryon acoustic oscillations, supernovae), trading off between completeness and reliability in the galaxy sample. The various sources of uncertainty in measurements of the photometry and redshifts put a lower bound on the accuracy that any model can hope to achieve. The intrinsic uncertainty associated with estimates is often non-uniform and input-dependent, commonly known in statistics as heteroscedastic noise. However, existing approaches are susceptible to outliers and do not take into account variance induced by non-uniform data density and in most cases require manual tuning of many parameters. In this paper, we present a Bayesian machine learning approach that jointly optimizes the model with respect to both the predictive mean and variance we refer to as Gaussian processes for photometric redshifts (GPz). The predictive variance of the model takes into account both the variance due to data density and photometric noise. Using the Sloan Digital Sky Survey (SDSS) DR12 data, we show that our approach substantially outperforms other machine learning methods for photo-z estimation and their associated variance, such as TPZ and ANNZ2. We provide a MATLAB and PYTHON implementations that are available to download at https://github.com/OxfordML/GPz.

Key words: methods: data analysis – galaxies: distances and redshifts.

1 INTRODUCTION

Photometric redshift estimation largely falls into two main methodological classes, machine learning and template fitting. Machine learning methods (MLM), such as artificial neural networks (e.g. ANNZ; Firth, Lahav & Somerville 2003; Collister & Lahav 2004; Sadeh, Abdalla & Lahav 2015), nearest-neighbour (Ball et al. 2008), genetic algorithms (e.g. Hogan, Fairbairn & Seeburn 2015), self-organized maps (Geach 2012; Masters et al. 2015), random forest (TPZ; Carrasco Kind & Brunner 2013) and Gaussian processes (GPs; Way et al. 2009; Bonfield et al. 2010; Almosallam et al. 2016), use different statistical models to predict the most probable redshift

given the observed photometry, using a training sample where usually the spectroscopic redshift is known. Artificial neural networks (ANN) motivate the most commonly used MLM (Firth et al. 2003; Vanzella et al. 2004; Brescia et al. 2014; Sadeh et al. 2015). The parameters of the models often cannot be analytically inferred, so global and greedy optimization methods are used to estimate their parameters. In addition to providing a point estimate, MLM can provide the degree of uncertainty in their predictions (Roberts, Penny & Pillot 1996; Bishop 2006; Carrasco Kind & Brunner 2013; Bonnett et al. 2015; Rau et al. 2015). Template fitting methods, on the other hand, do not learn a model from a training sample but rather use templates of galaxy spectral energy distributions (SEDs) for different galaxy types that can be redshifted to fit the photometry. Some limitations of template fitting methods are whether the templates are representative of the galaxies observed at high redshift

★ E-mail: ialmosallam@kacst.edu.sa

and how emission lines affect the photometry. Some allow spectroscopic data to be used to adjust the zero-points on the photometry to compensate for any slight mismatch between SED templates and the observations. Examples of template fitting software include HYPERZ (Bolzonella, Miralles & Pelló 2000), ZEBRA (Feldmann et al. 2006), EAZY (Brammer, van Dokkum & Coppi 2008) and LE PHARE (Ilbert et al. 2006). There have been comprehensive evaluations of different photometric redshift estimation techniques (Hildebrandt et al. 2010; Abdalla et al. 2011; Sánchez et al. 2014; Bonnett et al. 2015).

In this paper, we extend our previous work (Almosallam et al. 2016) and complete the Bayesian picture of the sparse Gaussian model. In Almosallam et al. (2016), the noise variance was assumed to be constant and treated as an input parameter optimized using cross-validation. In the approach we propose here, the variance is an input-dependent function and is learned jointly with the mean function. The variance produced by the proposed approach is composed of two terms that captures different sources of uncertainty. The first term is the intrinsic uncertainty about the mean function due to data density, whereas the second term captures the uncertainty due to the intrinsic noise or the lack of precision/features in the training set. This provides additional utility to identify regions of input space where more data are required, versus areas where additional precision or information is required. Classical GPs, for example, only model the uncertainty about the mean function and assume that the noise uncertainty is constant.

Such a method is particularly useful for machine learning-based methods, as it is often the case that the training samples are incomplete due to the difficulty in obtaining complete spectroscopic redshift information. Moreover, imaging data are predominantly many magnitudes deeper than spectroscopic data, therefore quantifying the noise on the photometric redshift in terms of whether it is due to the density of training data available in a certain colour space, or due to a lack of sufficiently precise data within the colour space of interest could be crucial. In particular, it means that optimal spectroscopic survey strategies can be implemented to increase the photometric accuracy in the best way possible for a given experiment, i.e. obtaining more data in different colour space, or improving the quality of data in that colour space through additional imaging for example.

This paper is organized as follows; first a summary of related work is presented in Section 2 followed by an overview of sparse GPs in Section 3. In Section 4, we discuss how to expand the method to favour simpler, or sparser, models via automatic relevance determination. The extension to account for heteroscedastic noise is described in Section 5. The experimental setup is presented in Section 6, followed by results and analysis in Section 7. Finally, we provide concluding remarks in Section 8.

2 RELATED WORK

In recent related work, we have proposed sparse GPs for photometric redshift inference (Almosallam et al. 2016). GPs are very powerful probabilistic models for regression that are easy to implement. However, the quadratic storage cost and the cubic computational complexity required to train them is deemed impractical for many applications where scalability is a major concern; requiring far more efficient approximations. One of the most common approximations used for GPs is to reduce the computational cost required to invert the $n \times n$ covariance matrix (which gives GPs their computational complexity), where n is the number of samples in the training set. One can also take advantage of the structure of the covariance ma-

trix, if the recordings are evenly spaced in a time series problem, for example, then the covariance matrix will have a Toeplitz structure which can be inverted much faster (Zhang, Leithead & Leith 2005). Another approach is to decompose the covariance matrix as a sum of Kronecker products to simplify the computation of the inverse (Tsiligkaridis & Hero 2013). These properties do not always hold; however, the covariance matrix will always be a positive semidefinite matrix which one can exploit to compute a good approximation of the inverse by treating the problem as a system of linear equations and use the conjugate gradient method to solve it (Gibbs & MacKay 1997). However, the inverse needs to be computed several times during the optimization process of the internal parameters and providing an approximate inverse to the optimizer will cause it to be unstable.

A second class of approaches is to reduce the size of the covariance matrix by means of sparse approximations, instead of using the entire n samples in the training set, a set of $m \ll n$ samples are used to construct the covariance matrix. The samples can be pre-selected either randomly or in an unsupervised fashion such as in Foster et al. (2009), where the active set is selected to increase the stability of the computation. Alternatively, one may search for 'pseudo' points not necessarily present in the training set (and not necessarily even lying within the data range) to use as the active set such that the probability of the data being generated from the model is maximized (Snelson & Ghahramani 2006). This approach uses a richer likelihood that models input-dependent heteroscedastic noise. However, it assumes a specific form for the noise process and uses a global kernel definition. A comprehensive overview of sparse approximation methods is detailed in Candela & Rasmussen (2005). We provide a full formal description of GPs in Section A of the Appendix and the reader is advised to read Rasmussen & Williams (2006) for a complete review of GPs. Except for Snelson & Ghahramani (2006), none of the previously discussed methods account for variable noise, with variations in the posterior variance estimates providing an indication of the model's confidence about its mean function, not the noise, due to the underlying assumption that the observed data has constant white Gaussian noise. One method to learn heteroscedastic noise is to model both the mean and the noise functions as GPs. This is achieved by first holding the noise fixed and optimizing with respect to the mean, then holding the mean fixed and optimizing with respect to the noise and repeated until convergence (Kersting et al. 2007), this can be viewed as a group-coordinate ascent optimization. In this paper, we use basis function models (BFM), viewed as a sparse GP method, and provide novel methods to enhance the posterior variance accuracy.

3 SPARSE GPS

In this section, we describe sparse GPs as BFM, whose semiparametric form is defined via a set of weights. The underlying assumption in a BFM is that, given a set of inputs $\mathbf{X} = \{x_i\}_{i=1}^{n} \in \mathbb{R}^{n \times d}$ and a set of target outputs $y = \{y_i\}_{i=1}^{n} \in \mathbb{R}^{n}$, where n is the number of samples in the data set and d is the dimensionality of the input, that the observed target y_i is generated by a linear combination of m non-linear functions $\phi(x_i) = [\phi_1(x_i), \ldots, \phi_m(x_i)] \in \mathbb{R}^m$ of the input plus additive noise $\epsilon_i \sim \mathcal{N}(0, \beta^{-1})$:

$$y_i = \phi(x_i)\,w + \epsilon_i, \tag{1}$$

where w is a vector of length m of real-valued coefficients, or the parameters of the model. In the case of photometric redshift estimation, $\mathbf{X}$ are the photometric measurements and associated uncertainties of the filters, namely d inputs, and n training objects.

Throughout the rest of the paper, $\mathbf{X}[i, :]$ denotes the ith row of matrix $\mathbf{X}$, or x_i for short, whereas $\mathbf{X}[:, j]$ denotes the jth column, $\mathbf{X}[i, j]$ denotes the element at the ith row and the jth column in matrix $\mathbf{X}$, and similarly for other matrices. Note that the mean of the predictive distribution derived from a GP is a linear combination of n kernel functions. The BFM approach is to assume the form of the function to be a linear combination of $m \ll n$ basis functions and integrates out its parameters. In this paper, we choose the radial basis function (RBF) kernel as our basis function, defined as follows

$$\phi_j(x_i) = \exp\left(-\frac{1}{2}(x_i - p_j)^T \Gamma_j^T \Gamma_j (x_i - p_j)\right), \tag{2}$$

where we define $\mathbf{P} = \{p_i\}_{i=1}^m \in \mathbb{R}^{m \times d}$ to be the set of basis vectors associated with the basis functions and $\Gamma_j^T \Gamma_j$, $\Gamma_j \in \mathbb{R}^{d \times d}$, are bespoke precision matrices associated with each basis function. We refer to the model with such basis functions as Gaussian processes with variable covariances, or GPVC. The framework also allows for other types of covariance structures, the options include:

(i) GPVC: variable covariances, or a bespoke Γ_j for each basis function j.

(ii) GPGC: a global covariance, or a shared Γ for all basis functions.

(iii) GPVD: variable diagonal covariances, or a bespoke diagonal Γ_j for each basis function j.

(iv) GPGD: a global diagonal covariance, or a shared diagonal Γ for all basis functions.

(v) GPVL: variable length-scales, or a bespoke isotropic covariance for each basis function j; i.e. $\Gamma_j = \mathbf{I}\gamma_j$, where γ_j is a scaler.

(vi) GPGL: a global length-scale, or a shared isotropic covariance for all basis functions $\Gamma = \mathbf{I}\gamma$, where γ is a scaler.

We assume, for now, that our observations are noisy with a constant precision β and a mean of zero. This is obviously not true in reality as the photometric noise is dependent on the depth of the individual images in each band. Note that these assumptions are made to simplify our illustration and we relax these later in the paper. The likelihood is hence defined as follows

$$p(y|w) = \mathcal{N}(\Phi w, \beta^{-1}\mathbf{I}), \tag{3}$$

$$\Phi = \begin{bmatrix} \phi(x_1) \\ \vdots \\ \phi(x_n) \end{bmatrix}. \tag{4}$$

We now need to define a prior on w in order to proceed. We use a prior that promotes a smooth function, hence preferring the simplest explanation that fits the data. The smoothness assumption also transforms the objective from an ill-posed problem to a well-posed one, as there are an infinite number of functions that would fit the data. This can be achieved by requiring the weights in w to be independent and the norm as small as possible. This can be formulated probabilistically by taking $p(w) = \mathcal{N}(0, \alpha^{-1})$, where α is the prior precision of the parameters w. With a likelihood and a prior, we can derive the posterior as $p(w|y) = p(y|w)p(w)/p(y)$ from Bayes theorem, which can be shown to have the following normal distribution (Bishop 2006):

$$p(w|y) = \mathcal{N}(\bar{w}, \Sigma), \tag{5}$$

$$\bar{w} = \beta \Sigma^{-1} \Phi^T y, \tag{6}$$

$$\Sigma = \beta \Phi^T \Phi + \alpha \mathbf{I}. \tag{7}$$

The marginal likelihood, or the evidence function (Bishop 2006), can be derived by integrating out w as in equation (8).

$$p(y) = \int p(y|w) p(w) \, dw. \tag{8}$$

This can be expressed in terms of the mean $\bar{w}$ and the covariance Σ of the posterior distribution:

$$\ln p(y) = -\frac{\beta}{2} \|\Phi\bar{w} - y\|^2 + \frac{n}{2}\ln\beta - \frac{n}{2}\ln(2\pi)$$
$$-\frac{\alpha}{2}\bar{w}^T\bar{w} + \frac{m}{2}\ln\alpha - \frac{1}{2}\ln|\Sigma|. \tag{9}$$

The hyperparameters of the basis function, the precision β, the weight precision α and the pseudo points' locations $\mathbf{P}$ can now be optimized with respect to the log marginal likelihood defined in equation (9). Once the parameters have been inferred, the predictive distribution of an unseen test case x_* is distributed as follows (Bishop 2006)

$$p(y_*|y) = \mathcal{N}\left(\mu_*, \sigma_*^2\right), \tag{10}$$

$$\mu_* = \phi(x_*)\bar{w}, \tag{11}$$

$$\sigma_*^2 = \nu_* + \beta^{-1}, \tag{12}$$

$$\nu_* = \phi(x_*)\Sigma^{-1}\phi(x_*)^T. \tag{13}$$

Note that we are no longer restricted to Mercer kernels or a single basis function definition. The basis function can therefore be modelled using variable length-scales and variable covariances as in Almosallam et al. (2016), to capture different kinds of patterns that can arise in different regions of the input space. BFM can be shown to be a degenerate GP with an equivalent kernel function $\kappa(x_i, x_j) = \alpha^{-1}\phi(x_i)\phi(x_j)^T$ (Candela & Rasmussen 2005).

4 AUTOMATIC RELEVANCE DETERMINATION

In addition to achieving accurate predictions, we wish to minimize the number of basis functions, to produce a sparse model representation. Instead of adding an additional prior over the number of basis functions, we can achieve this goal by incorporating a sparsity-inducing prior on w. We use prior diagonal precision matrix $\mathbf{A} = \text{diag}(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^m$, or a precision parameter per weight. The modified prior is $p(w) = \mathcal{N}(0, \mathbf{A}^{-1})$ and the log marginal likelihood is simply extended as follows

$$\ln p(y) = -\frac{\beta}{2}\|\Phi\bar{w} - y\|^2 + \frac{n}{2}\ln\beta - \frac{n}{2}\ln 2\pi \tag{14}$$

$$-\frac{1}{2}\bar{w}^T\mathbf{A}\bar{w} + \frac{1}{2}\ln|\mathbf{A}| - \frac{1}{2}\ln|\Sigma|,$$
$$\text{where } \Sigma = \beta\Phi^T\Phi + \mathbf{A}. \tag{15}$$

By modelling each weight with its associated precision, we enable a natural shrinkage (or regularization). Take, for example, a specific precision α_i; note that maximizing $\frac{1}{2}\ln\alpha_i$ will minimize $-\frac{1}{2}\bar{w}_i^2\alpha_i$, unless $\bar{w}_i = 0$. The optimization routine will therefore drive as many of the weights to zero as possible, thus maintaining the least number of basis functions relevant to model the data. A similar approach was proposed by Tipping (2001) coined as the relevance vector machine, where the set of basis function locations $\mathbf{P}$ was set equal to the locations of the training samples $\mathbf{X}$ and held

fixed. Only the precision parameter β and the $\boldsymbol{\alpha}$ values were optimized to determine the relevant set of vectors from the training set. This approach is still computationally expensive and Tipping (2001) proposed an iterative workaround to add and remove vectors incrementally.

5 HETEROSCEDASTIC NOISE

The predictive variance in equation (12) has two components, the first term ν_* is the model variance and the second term β^{-1} is the noise uncertainty. The model variance thus depends on the data density of the training sample at x_*. Theoretically, this component of the model variance will go to zero as the size of the data set increases. This term hence models our underlying uncertainty about the mean function. The model becomes very confident about the posterior mean when presented with a large number of samples at x_*, or in photometric redshift terms in a particular region of colour–redshift space, at which point the predictive variance reduces to the intrinsic noise variance. The latter, at this point, is assumed to be white Gaussian noise with a fixed precision β.

In this section, we enhance the model's predictive variance estimation by modelling the noise variance as a function of input, or $\beta_i = f(x_i)$ to account for variable and input-dependent noise, i.e. heteroscedastic noise, as is the case for imaging using different surveys. We choose to model the function as a linear combination of basis functions via $\beta_i = \exp(\phi(x_i)u + b)$, where we choose the exponential form to ensure positivity of β_i. Note that if $u = 0$ and $b = \ln \beta$, the model reduces to the original assumption of a fixed precision β. We thus redefine the likelihood as follows

$$p(y|w) = \mathcal{N}(\Phi w, \mathbf{B}^{-1}), \tag{16}$$

where $\mathbf{B}$ is a $n \times n$ diagonal matrix where each element across the diagonal $\mathbf{B}[i, i] = \beta_i$. Following the same procedure, the posterior $p(w|y)$ is expressed as follows:

$$p(w|y) = \mathcal{N}(\bar{w}, \Sigma), \tag{17}$$

$$\bar{w} = \Sigma^{-1}\Phi^T \mathbf{B}y, \tag{18}$$

$$\Sigma = \Phi^T \mathbf{B}\Phi + \mathbf{A}, \tag{19}$$

and the updated log marginal likelihood becomes

$$\ln p(y) = -\frac{1}{2}\delta^T \mathbf{B}\delta + \frac{1}{2}\ln|\mathbf{B}| - \frac{n}{2}\ln 2\pi$$
$$\quad -\frac{1}{2}\bar{w}^T \mathbf{A}\bar{w} + \frac{1}{2}\ln|\mathbf{A}| - \frac{1}{2}\ln|\Sigma|, \tag{20}$$

where $\delta = \Phi\bar{w} - y$. Note that cost-sensitive learning (Almosallam et al. 2016), can be readily incorporated into our model by setting $\mathbf{B}[i, i] = \beta_i \omega_i$, where $\omega_i = (1 + z_i)^{-2}$, in which z_i is the spectroscopic redshift for source i. In addition, we also add a prior on u to favour the simplest precision function, namely that u is normally distributed with a mean of 0 and a diagonal precision matrix $\mathbf{N} = \mathrm{diag}(\boldsymbol{\eta})$, or $u \sim \mathcal{N}(0, \mathbf{N}^{-1})$, where $\boldsymbol{\eta} = \{\eta_i\}_{i=1}^m$. The final objective function to be optimized is thus the log marginal likelihood plus the log of the prior on u,

$$\ln p(y) = -\frac{1}{2}\delta^T \mathbf{B}\delta + \frac{1}{2}\ln|\mathbf{B}| - \frac{n}{2}\ln 2\pi$$
$$\quad -\frac{1}{2}\bar{w}^T \mathbf{A}\bar{w} + \frac{1}{2}\ln|\mathbf{A}| - \frac{1}{2}\ln|\Sigma|$$
$$\quad -\frac{1}{2}u^T \mathbf{N}u + \frac{1}{2}\ln|\mathbf{N}| - \frac{m}{2}\ln 2\pi. \tag{21}$$

The parameter $\boldsymbol{\eta}$ hence acts as an automatic relevance determination cost for the noise process, allowing the objective to dynamically select different sets of relevant basis functions for both the posterior mean and variance estimation. The probability of unseen test cases is normally distributed as follows

$$p(y_*|y) = \mathcal{N}(\mu_*, \sigma_*^2), \tag{22}$$

$$\mu_* = \phi(x_*)\bar{w}, \tag{23}$$

$$\sigma_*^2 = \nu_* + \beta_*^{-1}, \tag{24}$$

$$\beta_* = \exp(\phi(x_*)u + b), \tag{25}$$

where β_*^{-1} is the input-dependent noise uncertainty and ν_* is defined in equation (13). It is worth mentioning that in the parameter space, w, the problem is convex; and thus can be modelled using a single Gaussian distribution. In the hyperparameter space, however, the problem can be highly non-convex with many local minima. This adds an extra source of uncertainty about the model due to training that is dependent on the initial condition and the optimization procedure. This can be addressed using a committee of models, where each model is initialized differently, to fit a mixture of Gaussian distribution instead of a single one to better fit the true model distribution (Roberts et al. 1996; Penny & Roberts 1997).

We search for the optimal set of model parameters using a gradient-based optimization; hence, we require the derivatives of the log marginal likelihood with respect to each parameter. The gradient calculations of the log marginal likelihood, equation 5.6, with respect to the model's parameters are provided in Section C of the Appendix, for both the general case of any basis function and an efficient procedure for the six different configurations of RBFs. In this paper, the limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm is used to optimize the objective. This uses a quasi-Newton method to compute the search direction in each step by approximating the inverse of the Hessian matrix from the history of gradients in previous steps (Nocedal 1980). We use the MINFUNC optimization toolbox by Schmidt (2005).

In Fig. 1, we demonstrate the effect on a toy univariate example using a sparse GP with heteroscedastic noise and a full GP model with a squared exponential kernel. We used the GPML toolbox implementation (Rasmussen & Nickisch 2010) for the full GP model to offer a comparison to the variable length-scale basis function for the sparse GP (GPVL). Note that both models estimate a higher predictive variance in the absence of data (−6 to −3 on the x-axis). However, this is the only source of uncertainty that the full GP is able to estimate accurately; the constraint of a constant noise variance has the negative effect of both overestimating and underestimating the true variance in different regions. On the other hand, the noise variance estimation in the GPVL model is more accurate and leads to a more accurate determination of the total uncertainty about the mean function.

6 EXPERIMENTAL SETUP

6.1 Tested models

The focus of the method described in this paper is to generate input-dependent predictive distributions, we therefore only include photometric redshift algorithms from the literature that produce point estimates of the posterior mean (the expected value), as well as uncertainty predictions (typically a predictive variance) for each source, given its photometry. For comparison, we test our proposed approach against ANNZ2 (Sadeh et al. 2015), TPZ (Carrasco Kind &
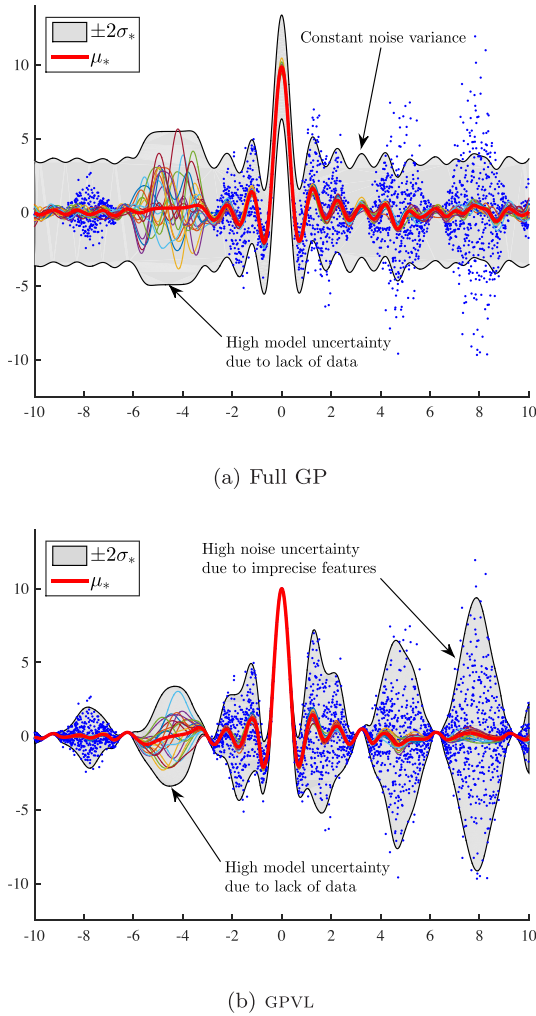
(a) Full GP

(b) GPVL

Figure 1. The mean, variance ($\pm 2\sigma_*$) and a sample of functions from the distribution produced by (a) a full GP model with a squared exponential kernel and (b) a GPVL model trained using 200 basis functions. The generative distribution of the target output $y(x) \sim \mathcal{N}(\mu(x), \sigma^2(x))$, where $\mu(x) = 10 \operatorname{sinc}(2x)$ and $\sigma(x) = (\frac{3\sin(x)}{1+\exp(-0.1x)} + 0.01)^2$.

Brunner 2013) and Sparse Pseudo-Input Gaussian Processes (SPGP) which also generate uncertainty predictions. ANNZ2 is an extension of ANNZ, a popular ANN-based code (Collister & Lahav 2004). ANNZ2 utilizes many MLM including ANNs, decision trees and k-nearest neighbours. ANNZ2 can be considered as a committee machine that combines the results of different models with various configurations, initializations and optimization techniques. For instance, the output of many ANNs with different number of layers, number of hidden units, input pre-processing, number of trees and sampling methods. TPZ is a random forest implementation that generates predictions by subdividing the data based on its features until a termination leaf is reached, determined using an information gain metric that measures the information quality of each feature and its ability to predict the desired output. The algorithm generates a number of trees, each trained on a sub-sample of features, which proves to be more effective and stable than a single tree trained on all features. SPGP is a sparse GP model that uses pseudo-inputs as the basis set to determine the covariance function of the GP (Snelson & Ghahramani 2006). The pseudo-inputs are treated as parameters of the model that are optimized to maximize the log marginal

likelihood. SPGP is similar to the GPGL model, except that the prior covariance of w is set to the covariance matrix of the pseudo-inputs, instead of setting it to $\mathbf{A}^{-1}$. The posterior variance is inferred using a stationary noise model in SPGP, whereas the posterior variance in GPGL is modelled as a separate function of the basis, hence allowing for non-stationarity and input sensitivity.

6.2 The data set

We train the models on the Sloan Digital Sky Survey's 12th Data Release (SDSS; Alam et al. 2015). We select galaxies where both the photometry and the spectroscopic redshifts are available. The total number of sources is 2 120 465, which contains 1 301 943 from the Baryon Oscillation Spectroscopic Survey, 817 657 from the SDSS-III survey, 826 from Segue-1 and 93 from Segue-2. The `modelMag` magnitudes for the u,g,r,i and z bands were used with their associated error estimates. We pre-process the associated uncertainties of the photometry by replacing them with their natural log to transform the domain of the features from the positive domain to the real domain. This has the advantage of having all the features share the same domain and allows for a fully unconstrained optimization. In addition, we use principle component analysis (Jolliffe 1986) to de-correlate the features, such that the data have a zero mean and an identity covariance, but retain all features with no dimensionality reduction. De-correlation speeds up the optimization processes and offers some numerical advantages. This approach, often referred to as 'sphering' or 'whitening' in the literature, is a common practice (Bishop 2006). We randomly sampled three sets of 100 000 sources each for training, validation and testing. The training set was used for learning the model, the validation set for model selection and the test set to report the results. The SQL statement used to create the data set is provided in Section D of the Appendix.

6.3 Metrics

Four metrics are considered to compare the results of the different methods. The mean log likelihood (MLL), the root mean squared error (RMSE), the fraction retained (FR) which provides a metric for the level of catastrophic outliers from the one-to-one relation, and the bias. These are defined as below:

$$\text{MLL} = \frac{1}{n}\sum_{i=1}^{n} -\frac{1}{2\sigma_i^2}(z_i - \acute{z}_i)^2 - \frac{1}{2}\ln\sigma_i^2 - \frac{1}{2}\ln 2\pi, \tag{26}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{z_i - \acute{z}_i}{1 + z_i}\right)^2}, \tag{27}$$

$$\text{FR}_e = \frac{100}{n}\left|\left\{i : \left|\frac{z_i - \acute{z}_i}{1 + z_i}\right| < e\right\}\right|, \tag{28}$$

$$\text{Bias} = \frac{1}{n}\sum_{i=1}^{n}\frac{z_i - \acute{z}_i}{1 + z_i}, \tag{29}$$

where z_i is the spectroscopic redshift for source i, $\acute{z}_i$ is the predicted photometric redshift, σ_i^2 is the predicted variance and e is the outlier threshold, i.e. $\text{FR}_{0.15}$ is the fraction of samples where $|(z_i - \acute{z}_i)/(1 + z_i)|$ is less than 0.15. The log likelihood is a natural way to evaluate the point estimate and the uncertainty prediction at the same time. The first term is the weighted sum of squared errors, where the weights are the predicted variance, prefers larger variance, whereas the second term punishes for large variances on
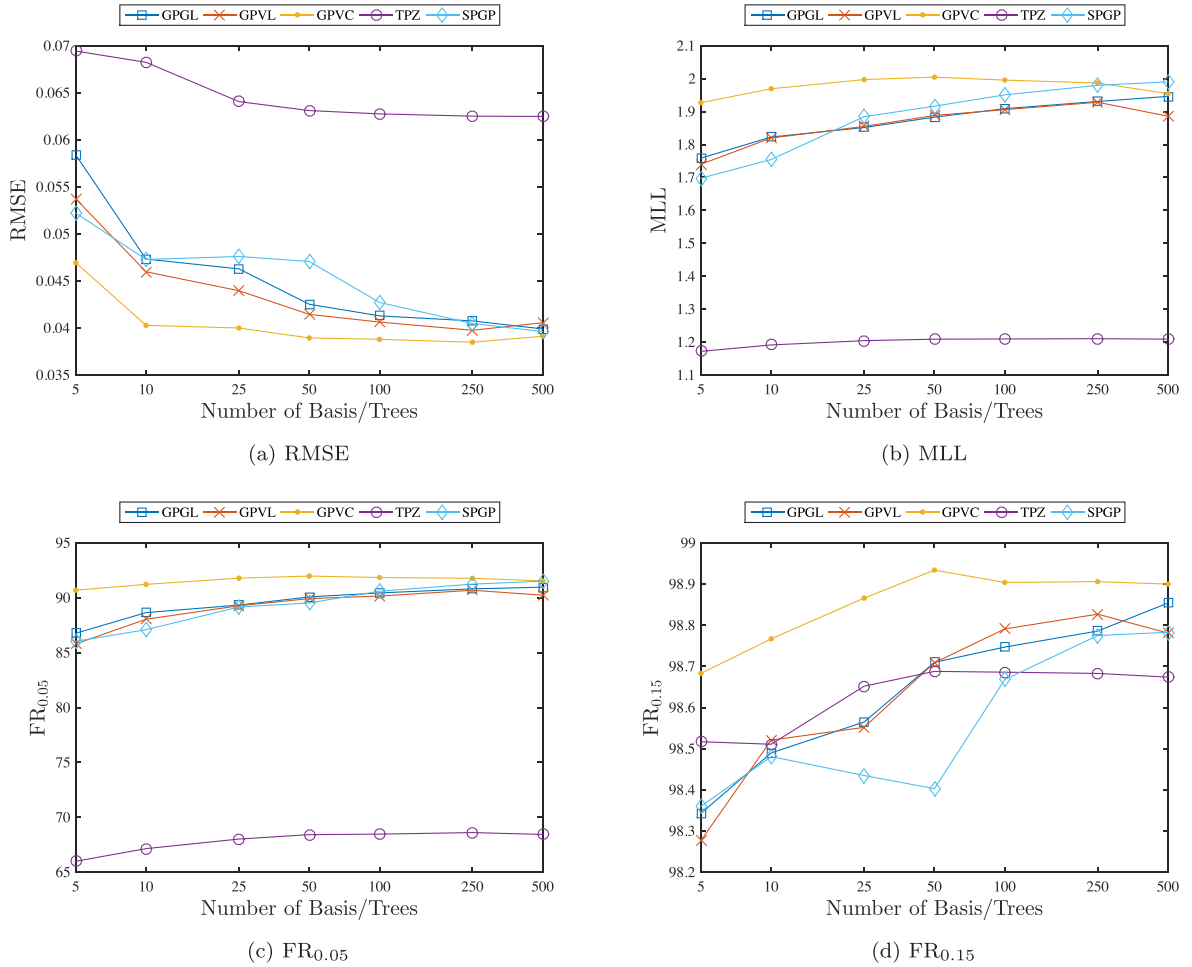
(a) RMSE

(b) MLL

(c) $FR_{0.05}$

(d) $FR_{0.15}$

Figure 2. The (a) RMSE, (b) MLL, (c) $FR_{0.05}$ and (c) $FR_{0.15}$ performance of each method on the test set using different numbers of basis/trees.

a log scale. The advantage of this form is that the optimal variance, if everything else were to set fixed, is exactly the squared error.

7 RESULTS AND ANALYSIS

In the following, we analyse the results from the various MLM within a number of tests. For the predictive mean, we use the `zmean1` prediction from TPZ and the `ANNZ_best` score from ANNZ2. We use the square of `err1` from TPZ and the square of `ANNZ_best_err` from ANNZ2 as the predictive variance.

7.1 Model complexity

In the first experiment, we analyse the relationship between the algorithms' complexity and their fit, as measured by the proposed metrics. For GPGL, GPVL, GPVC and SPGP, we vary the number of basis functions, whereas in TPZ, we vary the number of trees in the forest and fix the number of sub-features selected for each tree to the suggested value of $\sqrt{d} \simeq 4$ and keep the remaining configuration options as suggested, since the code is configured for SDSS-like surveys. We tested the models on 5, 10, 25, 50, 50, 100, 250 and 500 basis/trees. ANNZ2 is an aggregation of many models with various configurations so it is not included in this experiment. Fig. 2 shows the performance of the methods on the held-out test set as we vary the number of basis/trees. GPVC consistently outperforms the other methods in all metrics, reaching an RMSE $\sim$ 0.039, $FR_{0.05}$

Table 1. Performance measures for each algorithm trained using 100 basis functions for GPGL, GPVL, GPVC and SPGP, 100 trees for TPZ and 100 MLMs for ANNZ2 on the held-out test set. The best-performing algorithm is highlighted in bold font.

	RMSE	MLL	$FR_{0.15}$	$FR_{0.05}$
TPZ	0.0628	1.21	98.69 per cent	68.47 per cent
ANNZ2	0.0422	1.65	98.77 per cent	89.08 per cent
SPGP	0.0427	1.95	98.67 per cent	90.60 per cent
GPGL	0.0413	1.91	98.75 per cent	90.45 per cent
GPVL	0.0406	1.91	98.79 per cent	90.16 per cent
GPVC	**0.0388**	**2.00**	**98.90 per cent**	**91.85 per cent**

$\sim$91.9 per cent and $FR_{0.15} \sim$ 98.9 per cent, TPZ, on the other hand, is significantly worse in all metrics (RMSE $\sim$ 0.063; $FR_{0.05} \sim$ 68.5 per cent; $FR_{0.15} \sim$ 98.7 per cent).

7.2 Performance analysis

In the second experiment, we fix the number of basis/trees to 100, or at the point where they start to converge from the previous experiment, and generate predictions from ANNZ2 using the recommended randomized regression script. The number of learning models is set to 100 using both ANNs and Boosted Decision Trees; the remaining options are set to their default values as published. The performance measures are reported in Table 1, and for the general case of FR_e, we
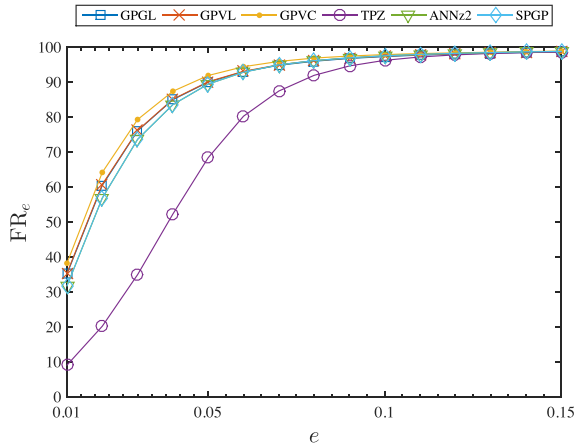
Figure 3. The FR$_e$ for different values of e for each method using 100 basis/trees/MLMs on the test set.

show in Fig. 3 the FR$_e$ score as we vary the value of the threshold e. The scatter plots for each method are colour coded by the predictive variance and shown in Fig. 4. We find that GPVC consistently outperforms all other GP methods and also ANNz2, although the margins are at the ∼1 per cent level. TPZ provides the poorest results by a

significant margin for low values of e, but asymptotes towards the FR values for the other codes at $e > 0.1$.

7.3 Rejection performance

As stated in Section 1, one of the critical aspects of using photometric redshifts in future cosmology experiments requires the understanding of the variance on the individual galaxy photometric redshift and on the distribution.

In this section, we analyse the quality of the models' uncertainty predictions by evaluating their rejection performance, namely their ability to infer which data are associated with high uncertainty; as we remove such samples, we would expect performance to improve. Fig. 5 shows the scores of the metrics as a function of the percentage of data selected based on the predictive variance generated by each method using 100 basis/trees/MLMs. TPZ is significantly worse than the other methods on all metrics, ANNz2 performs much better but still underperforms the GP-based methods. GPGL and GPVL perform equally well, but underperform SPGP slightly. GPVC consistently outperforms the other methods, on all metrics, for almost the entire range. Fig. 6 shows the relative change over GPVC as a reference for the plots in Fig. 5. GPVC shows a significant and consistent improvement over all methods, especially past 20 per cent of the data. For less than 20 per cent, SPGP is competitive to GPVC but is not consistently better. To quantify this, we compute the average
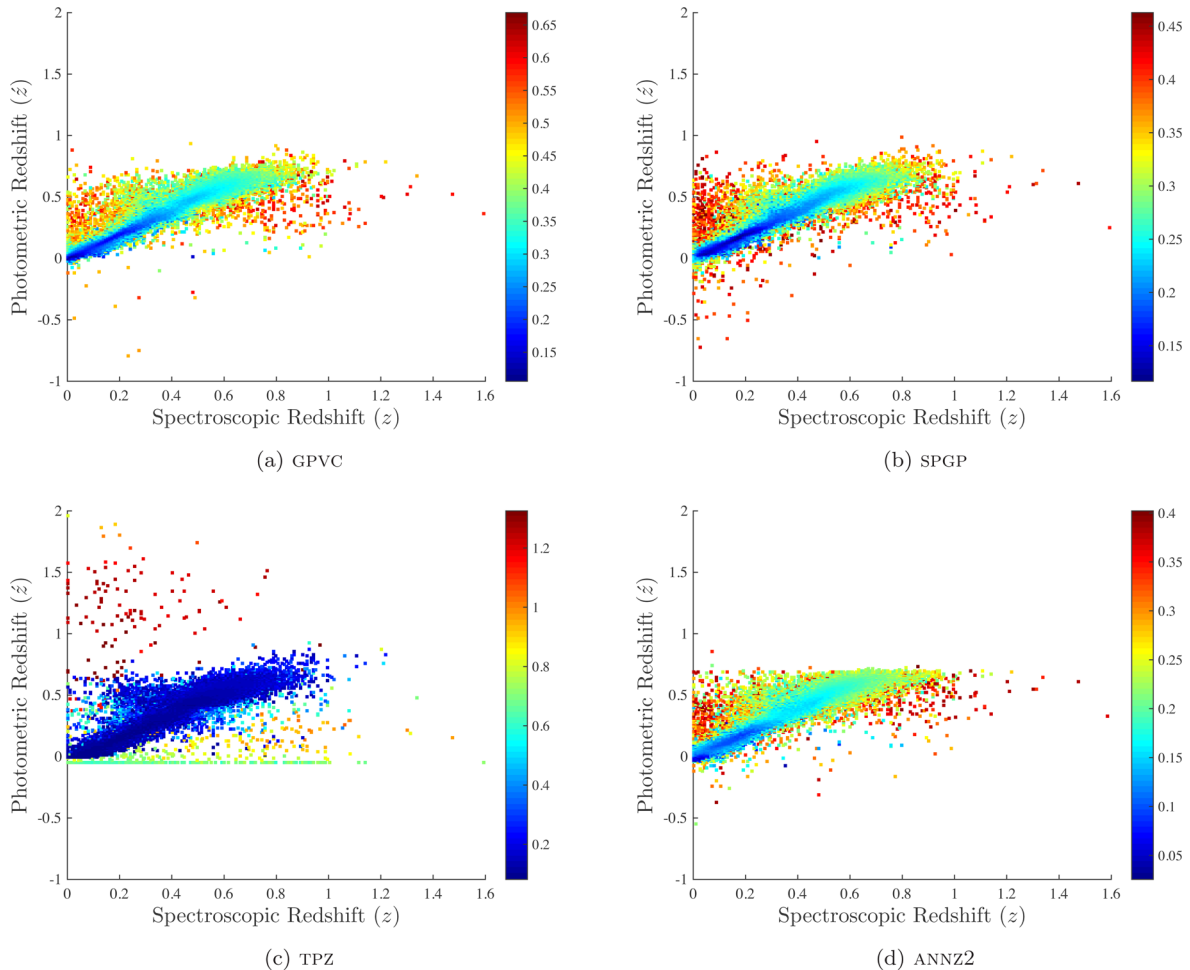

Figure 4. The scatter plots of the spectroscopic redshift z versus the predicted photometric redshift $ź$ on the test set for (a) GPVC, (b) SPGP, (c) TPZ and (d) ANNz2 using 100 basis/trees/MLMs. The predictive variance is colour coded, on a log scale, by the value of σ_* (equation 24).
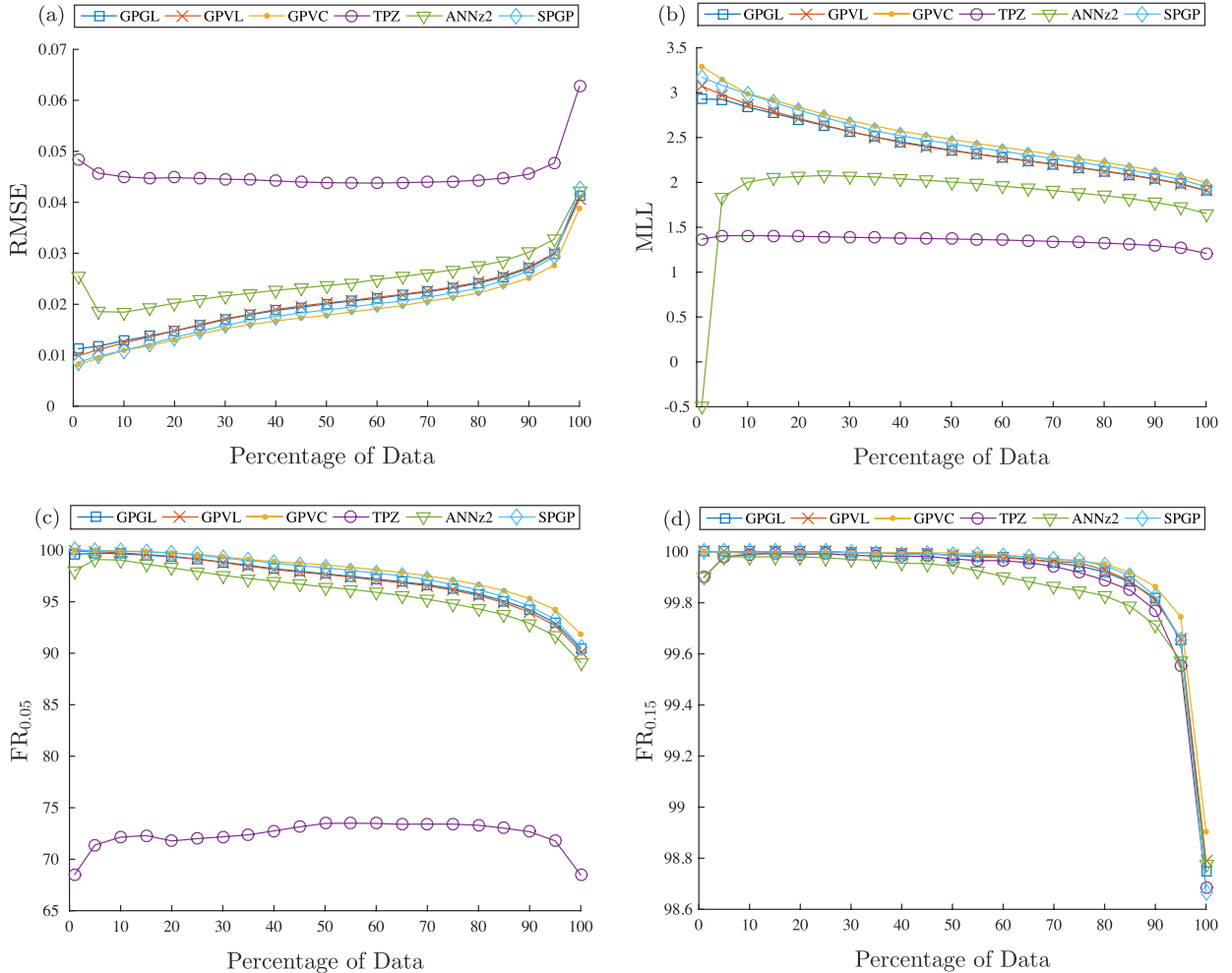
Figure 5. The (a) RMSE, (b) MLL, (c) $FR_{0.05}$ and (d) $FR_{0.15}$ scores as functions of the percentage of data selected based on the predictive variance generated by each method using 100 basis/trees/MLMs.

improvement that GPVC has over the other methods over the entire range; these results are reported in Table 2. The results show that GPVC provides performance improvement over all the other methods on all metrics. This therefore provides a robust basis for optimizing the sample selection of galaxies to use in various experiments, allowing the trade-off between number of galaxies included and their photometric-redshift accuracy.

7.4 Bias

We regard bias as a key metric for future experiments and science focus. The bias indicates how the photometric redshift systematically deviates from the true redshift as a function of the input and output. We report in Fig. 7 the bias (equation 29) as a function of the spectroscopic redshift (z), using different percentages of the data selected by each method's predictive variance grouped by uniformly spaced bins of width 0.1. Over the entire data range, TPZ shows the worst performance while the remaining methods perform equally well to a redshift of ~0.9. At higher redshifts, the performance of the GP-based methods and ANNZ2 vary with no clear winner. The figure shows that as we exclude more samples, all methods tend to be more certain about low-redshift ($z < 0.6$) samples. The methods we propose in this paper, however, are more stable and tend to im-

prove as we reject more data, whereas the bias scores for TPZ, ANNZ2 and SPGP, in some cases, degrade especially for high redshift.

7.5 Uncertainty analysis

As discussed in Section 5, the predictive variance produced by the proposed GPVC method is composed of two terms that model the uncertainty about the function due to data density inhomogeneity and the noise uncertainty. In this experiment, we analyse these two components of uncertainty separately using a GPVC model with 100 basis functions. Fig. 8 shows the model and noise uncertainties as functions of the spectroscopic redshift (z) using uniformly spaced bins of width 0.1. Both start to increase rapidly beyond $z \sim 0.5$. However, the overwhelming contribution to the overall uncertainty for high redshifts is due to the intrinsic noise rather than the scarcity of data. This indicates that the amount of data is sufficient for the model to be confident about its mean function and we have precise enough features for redshifts <0.5. For higher redshifts, the results indicate that obtaining more precise, or additional, features (e.g. near-infrared photometry) is a better investment than obtaining, or training on, more samples. This is not a surprising result given the data used, i.e. the spectroscopic training set and the test set are both sub-samples derived from the same overall SDSS galaxy sample. However, such a situation will not be the case for most
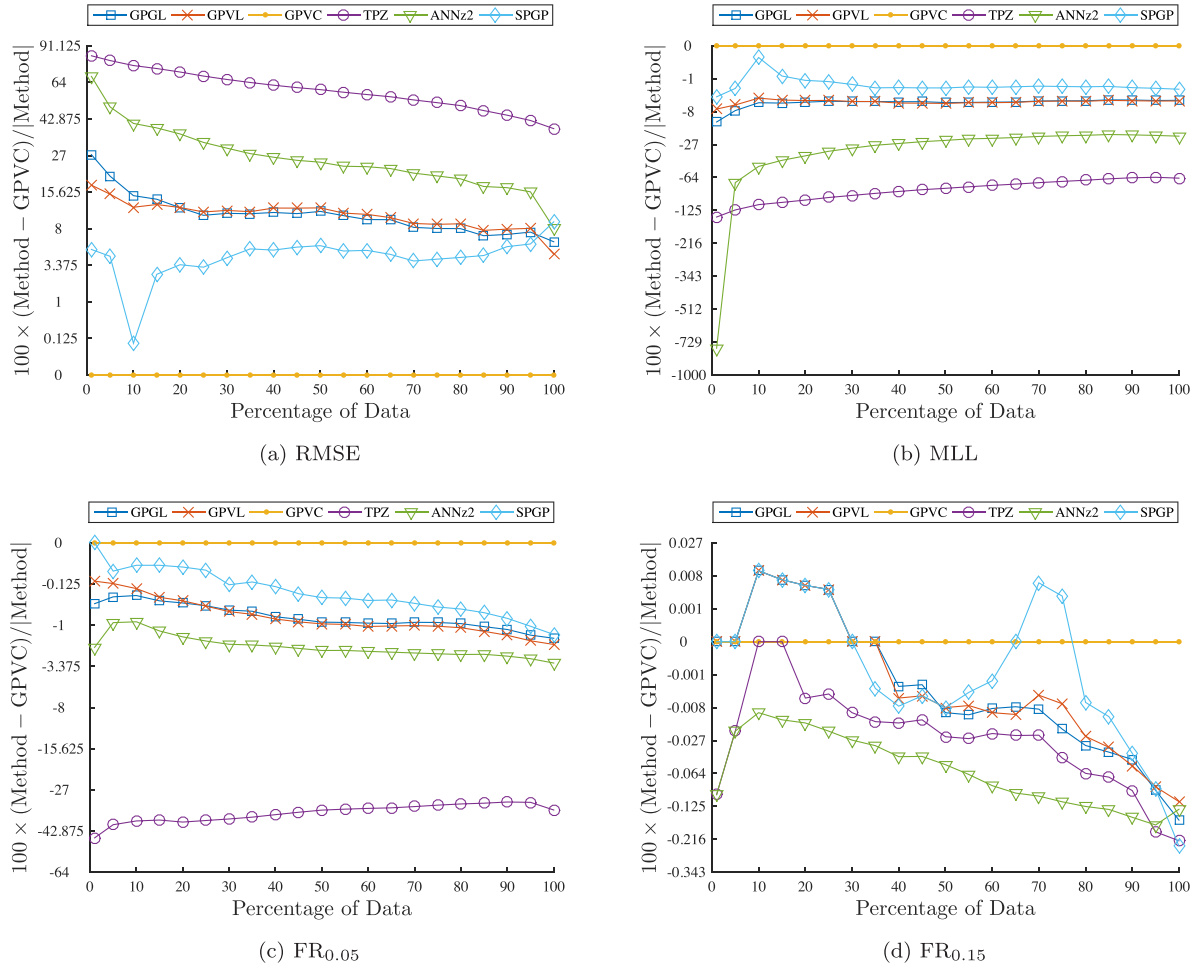
(a) RMSE

(b) MLL

(c) $FR_{0.05}$

(d) $FR_{0.15}$

Figure 6. The percentage of difference between GPVC and the other methods, computed as $100 \times (\text{Method} - \text{GPVC})/|\text{Method}|$, on (a) RMSE, (b) MLL, (c) $FR_{0.05}$ and (d) $FR_{0.15}$ as a function of the percentage of data selected based on the predictive variance generated by each method using 100 basis/trees/MLMs. The values are plotted on a cubic root y-axis to enhance visibility.

Table 2. The average relative improvement of GPVC over other tested methods on all metrics on the test set using 100 basis/trees/MLMs.

	RMSE improvement	MLL improvement	$FR_{0.15}$ improvement	$FR_{0.05}$ improvement
TPZ	59.87 per cent	85.76 per cent	0.0448 per cent	35.36 per cent
ANNZ2	27.44 per cent	58.04 per cent	0.0715 per cent	2.03 per cent
SPGP	4.29 per cent	1.91 per cent	0.0099 per cent	0.326 per cent
GPGL	10.89 per cent	5.69 per cent	0.0149 per cent	0.772 per cent
GPVL	10.80 per cent	5.02 per cent	0.0137 per cent	0.840 per cent

cosmological applications that require photometric redshifts, and having such separable noise terms will aid in determining the optimal approach to ensure that the requisite training samples are in place to address particular scientific problems, from galaxy evolution to various cosmology experiments.

7.6 Time complexity analysis

We provide in this section analysis of the theoretical and empirical time complexity of the methods tested in this paper. Table 3 shows the upper bound time complexity of each algorithm as a function of the number of input data samples n, features d (the dimensionality of the input data) and basis functions/trees/neurons m. If $m \geq d^2$, then the time complexity of GPGL, GPVL, GPVC and SPGP is equal to $O(nm^2)$, whereas the time complexities of TPZ and a single-layer neural network will remain the same. Thus, for $m < d\log(n)$, random forest trees have a higher upper bound than the other methods. The time complexity of random forests can be reduced to $O(n_s m d_s \mathcal{D})$, where n_s is the sub-sample size to grow each tree, d_s is the number of sub-sampled features used to grow each tree and $\mathcal{D}$ is the maximum depth allowed for each tree to grow. In practice, however, efficient implementation of the methods can significantly impact the actual running time. For example, effective computing of matrix operations ideally makes use of algorithms that are parallelizable and hence can be even further accelerated using graphical processing units. Using the same training data set of 100 000
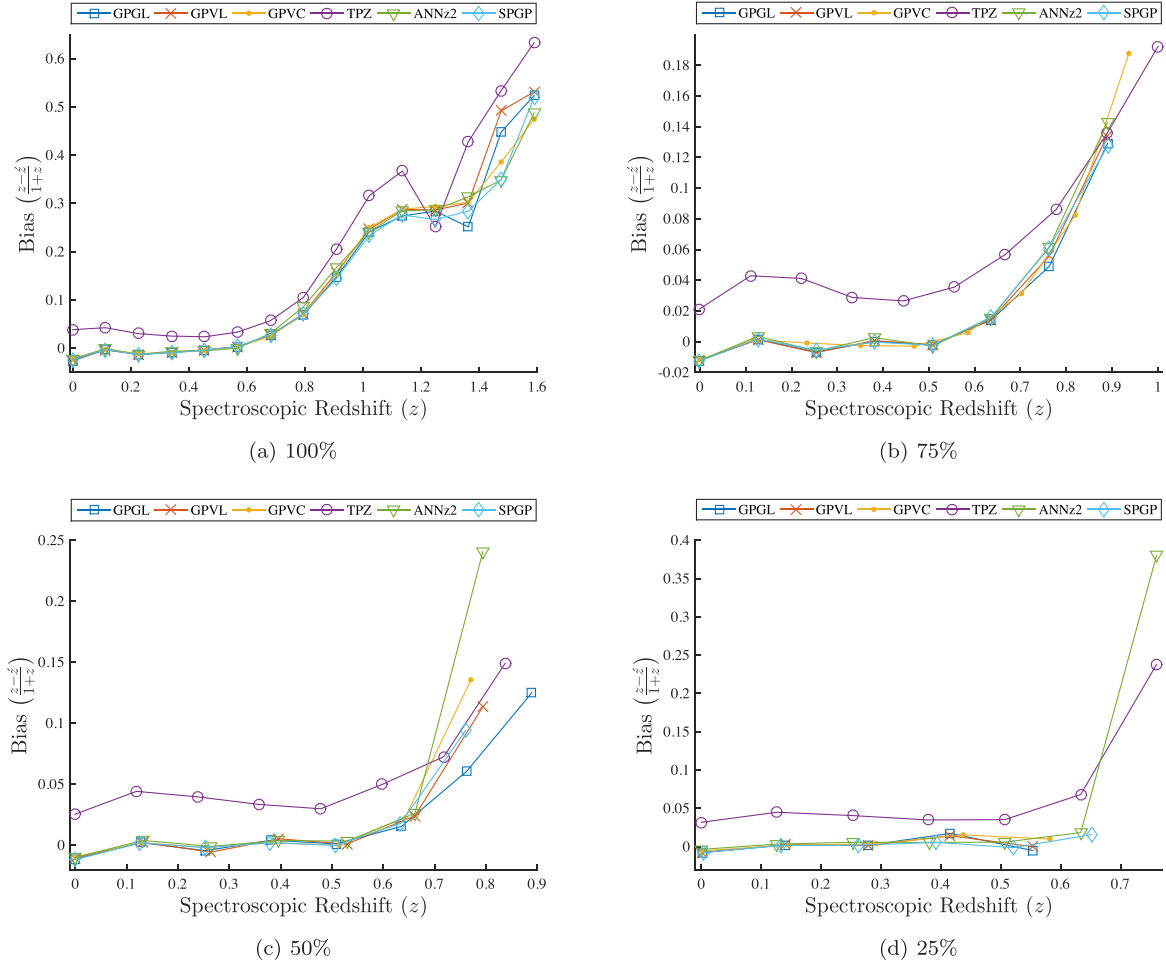
Figure 7. The RMSE as a function of the percentage of data selected based on the predictive variance generated by each method using 100 basis/trees/MLMs.
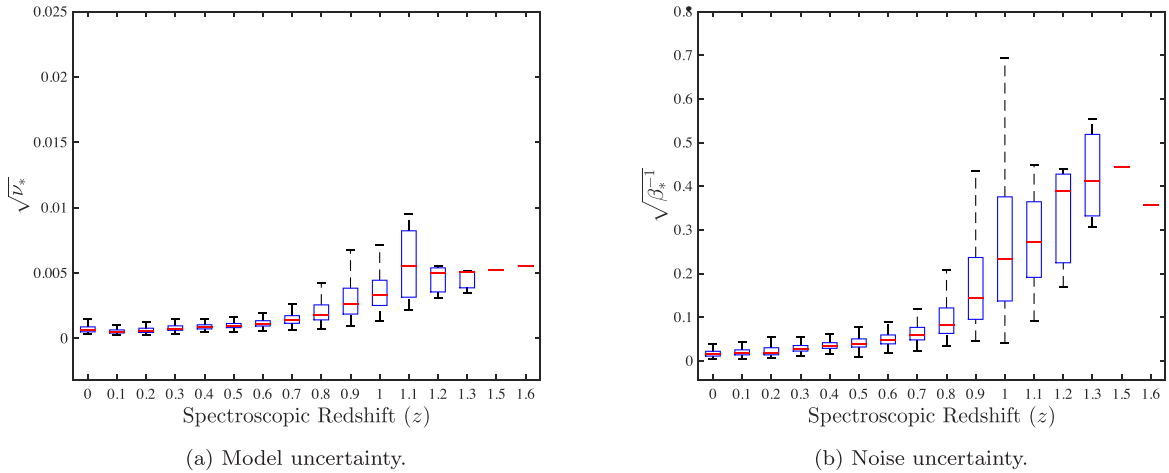

Figure 8. A box plots of the square root of (a) the model uncertainty (equation 13) and the (b) noise uncertainty (equation 25), produced by a GPVC model with 100 basis functions, versus the spectroscopic redshift showing median (bar), inter-quartile range (box) and range (whiskers).

samples, it took TPZ 1 h, 46 min and 42 s to train a forest of five trees; whereas ANNz2's random forest implementation required only 4 min and 34 s. On the other hand, the random forest implementation of MATLAB's statistical and machine learning toolbox, and PYTHON's

SKLEARN library required less than 3 s. Training a single-layer neural network with five hidden neurons using ANNz2 for 500 iterations required 20 min and 41 s, whereas SPGP, GPGL, GPVL and GPVC trained using five basis functions for the same number of iterations required

Table 3. The theoretical time complexity of each approach, where n is the number of samples, d is the number of features or the dimensionality of the input, m is the number of basis functions, trees in TPZ and hidden units in a single-layer ANN.

Method	Time complexity
ANN	$O(nmd)$
TPZ	$O(nmd\log(n))$
SPGP	$O(nmd + nm^2)$
GPGL, GPVL, GPGD and GPVD	$O(nmd + nm^2)$
GPGC and GPVC	$O(nmd^2 + nm^2)$

2 min and 5 s, 1 min and 28 s, 1 min and 28 s, and 2 min and 32 s, respectively.

8 CONCLUSIONS

We have produced and implemented an extension of the sparse GP framework presented in Almosallam et al. (2016) to incorporate separable terms for intrinsic noise in the data and the model uncertainty due to the finite data samples in the training set. These are combined to estimate the total variance on the predicted photometric redshifts.

We find that our algorithm outperforms other MLM tested in the literature across all metrics considered. In particular, we find that by including these terms, we are able to accurately determine the relative variance between photometric redshift in individual galaxies. This leads to the ability to reject parts of the data set in order to gain higher accuracy on the required metric, i.e. root mean square error, normalized median absolute deviation and/or the bias as a function of redshift. Moreover, the presented models provide a significant time improvement especially over TPZ and ANNZ2. The algorithm, which includes the cost-sensitive learning discussed in Almosallam et al. (2016), in addition to the separable noise terms presented in this paper, is available in MATLAB and PYTHON implementations from https://github.com/OxfordML/GPz.

In a subsequent paper, we will investigate how the algorithm can be used to define future imaging and spectroscopic surveys in order to provide the most efficient strategy for delivering photometric redshifts of the accuracy required to perform various cosmological experiments with future facilities, similar to the work of Masters et al. (2015) but with the added advantage of being able to separate data density issues from uncertainty due to photometric noise.

ACKNOWLEDGEMENTS

REFERENCES

Abdalla F. B., Banerji M., Lahav O., Rashkov V., 2011, MNRAS, 417, 1891
Alam S. et al., 2015, ApJS, 219, 12
Almosallam I. A., Lindsay S. N., Jarvis M. J., Roberts S. J., 2016, MNRAS, 455, 2387
Ball N. M., Brunner R. J., Myers A. D., Strand N. E., Alberts S. L., Tcheng D., 2008, ApJ, 683, 12
Bishop C. M., 2006, Pattern Recognition and Machine Learning. Springer, New York
Bolzonella M., Miralles J.-M., Pelló R., 2000, A&A, 363, 476
Bonfield D. G., Sun Y., Davey N., Jarvis M. J., Abdalla F. B., Banerji M., Adams R. G., 2010, MNRAS, 405, 987
Bonnett C. et al., 2015, preprint (arXiv:1507.05909)
Brammer G. B., van Dokkum P. G., Coppi P., 2008, ApJ, 686, 1503
Brescia M., Cavuoti S., Longo G., De Stefano V., 2014, A&A, 568, A126
Candela J. Q., Rasmussen C. E., 2005, J. Mach. Learn. Res., 6, 1939
Carrasco Kind M., Brunner R. J., 2013, MNRAS, 432, 1483
Collister A. A., Lahav O., 2004, PASP, 116, 345
Feldmann R. et al., 2006, MNRAS, 372, 565
Firth A. E., Lahav O., Somerville R. S., 2003, MNRAS, 339, 1195
Foster L. et al., 2009, J. Mach. Learn. Res., 10, 857
Geach J. E., 2012, MNRAS, 419, 2633
Gibbs M., MacKay D. J. C., 1997, Technical report, Efficient Implementation of Gaussian Processes. Cavendish Laboratory, Cambridge
Hildebrandt H. et al., 2010, A&A, 523, A31
Hogan R., Fairbairn M., Seeburn N., 2015, MNRAS, 449, 2040
Ilbert O. et al., 2006, A&A, 457, 841
Jolliffe I. T., 1986, Principal Component Analysis. Springer-Verlag, New York
Kersting K., Plagemann C., Pfaff P., Burgard W., 2007, Ghahramani Z. ed., Proceedings of the Twenty-Fourth International Conference (ICML 2007), Machine Learning. Corvallis, Oregon, p. 393
Masters D. et al., 2015, ApJ, 813, 53
Mercer J., 1909, Phil. Trans. R. Soc. A, 209, 415
Nocedal J., 1980, Math. Comput., 35, 773
Penny W. D., Roberts S. J., 1997, Technical report TR-97-1, Neural Network Predictions with Error Bars. Imperial College London, Department of Electrical and Electronic Engineering, London
Rasmussen C. E., Nickisch H., 2010, J. Mach. Learn. Res., 11, 3011
Rasmussen C., Williams C., 2006, Gaussian Processes for Machine Learning. University Press Group Limited. MIT Press, Cambridge, MA
Rau M. M., Seitz S., Brimioulle F., Frank E., Friedrich O., Gruen D., Hoyle B., 2015, MNRAS, 452, 3710
Roberts S. J., Penny W., Pillot D., 1996, Intelligent Sensors (Digest No: 1996/261), IEE Colloquium on. p. 10/1
Roberts S., Osborne M., Ebden M., Reece S., Gibson N., Aigrain S., 2013, Phil. Trans. R. Soc. A, 371, 20110550
Sadeh I., Abdalla F. B., Lahav O., 2015, preprint (arXiv:1507.00490)
Sánchez C. et al., 2014, MNRAS, 445, 1482
Schmidt M., 2005, minFunc: Unconstrained Differentiable Multivariate Optimization in Matlab, available at: http://www.cs.ubc.ca/schmidtm/Software/minFunc.html
Snelson E., Ghahramani Z., 2006, in Weiss Y., Schölkopf B., Platt J., eds, Advances in Neural Information Processing Systems 18. MIT Press, Cambridge, MA, p. 1257
Tipping M. E., 2001, J. Mach. Learn. Res., 1, 211
Tsiligkaridis T., Hero A., 2013, IEEE Trans. Signal Process., 61, 5347
Vanzella E. et al., 2004, A&A, 423, 761
Way M. J., Foster L. V., Gazis P. R., Srivastava A. N., 2009, ApJ, 706, 623
Zhang Y., Leithead W., Leith D., 2005, Time Series Gaussian Process Regression Based on Toeplitz Computation of O(N2) Operations and O(N)-level Storage. IEEE, Seville, Spain, p. 3711

APPENDIX A: GPS

A GP is a supervised non-linear regression algorithm lying within the class of Bayesian non-parametric models due to the few explicit parametric assumptions that it makes about the nature of the function fit. Given a set of input $\mathbf{X} = \{x_i\}_{i=1}^n \in \mathbb{R}^{n \times d}$ and a set of target outputs $y = \{y_i\}_{i=1}^n \in \mathbb{R}^n$, where n is the number of samples in the data set and d is the dimensionality of the input, the underlying assumption of a GP is that the observed target y_i is generated by a function of the input x_i plus additive noise ϵ_i:

$$y_i = f(x_i) + \epsilon_i, \tag{A1}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. It is assumed that y has a zero mean (this can readily be achieved without loss of generality) and univariate, although the derivation can be readily extended to the multivariable case. The likelihood, the probability of observing the targets given the function, is hence distributed as follows

$$p(y|f_x, \sigma^2) = \mathcal{N}(f_x, \sigma^2 \mathbf{I}), \tag{A2}$$

where $f_x = \{f(x_1), \ldots, f(x_n)\}$. A GP then proceeds by applying Bayes theorem to infer the sought-after distribution of the function f_x, given the observations

$$p(f_x|y, \mathbf{X}, \sigma^2) = \frac{p(y|f_x, \sigma^2) p(f_x|\mathbf{X})}{p(y|\mathbf{X}, \sigma^2)}. \tag{A3}$$

This requires us to define a prior, $p(f_x|\mathbf{X})$, over our space of functions. Most widely used priors assume local similarity in the data, i.e. closeby inputs are mapped to similar outputs. More formally, we assume a normally distributed prior with a mean of zero, to match the mean of the normalized target y, with a covariance function $\mathbf{K}$ to capture our prior belief of data locality, i.e. $p(f_x|\mathbf{X}) \sim \mathcal{N}(0, \mathbf{K})$. The covariance $\mathbf{K}$ is modelled as a function of the input, $\mathbf{K} = \kappa(\mathbf{X}, \mathbf{X})$. Each element at the ith row and the jth column of $\mathbf{K}$ is set equal to $\kappa(x_i, x_j)$, where κ is the covariance function. The function κ cannot be any arbitrary mapping, as it has to guarantee that $\mathbf{K}$ is a valid covariance matrix, i.e. symmetric and positive semidefinite. A class of functions referred to as Mercer kernels guarantees these structural constraints (Mercer 1909). An example of valid Mercer kernel is the squared exponential kernel, defined as follows

$$\kappa(x_i, x_j) = h^2 \exp\left(-\frac{1}{2\lambda^2}\|x_i - x_j\|^2\right), \tag{A4}$$

where h and λ are referred to as the height- and length-scale, respectively. The reader is referred to Rasmussen & Williams (2006) or Roberts et al. (2013) for in-depth discussion of covariance functions and kernels. With a likelihood $p(y|f_x)$ and a prior $p(f_x|\mathbf{X})$, the marginal likelihood $p(y|\mathbf{X})$ can be computed as follows (Rasmussen & Williams 2006):

$$p(y|\mathbf{X}, \sigma^2) = \int p(y|f_x, \mathbf{X}, \sigma^2) p(f_x|\mathbf{X}) \, df_x \tag{A5}$$

$$= \mathcal{N}(0, \mathbf{K} + \sigma^2 \mathbf{I}). \tag{A6}$$

The parameters of the kernel and the noise variance, collectively referred to as the hyperparameters of the model, are then optimized by maximizing the probability of the log of the marginal likelihood in equation (A5):

$$\ln p(y|\mathbf{X}, \sigma^2) = -\frac{1}{2} y^T (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} y$$
$$-\frac{1}{2} \ln|\mathbf{K} + \sigma^2 I| - \frac{n}{2} \ln(2\pi). \tag{A7}$$

Once the hyperparameters have been inferred, the probability of future predictions f_* for test cases $\mathbf{X}_*$ given the training set, the predictive distribution, can be inferred from the joint distribution of f_* and the observed targets y. If we assume that the joint distribution is a multivariate Gaussian, then the joint probability is distributed as follows

$$p(y, f_*|\mathbf{X}, \mathbf{X}_*, \sigma^2) = \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K}_{xx} + \sigma^2 \mathbf{I} & \mathbf{K}_{x*} \\ \mathbf{K}_{*x} & \mathbf{K}_{**} \end{bmatrix}\right), \tag{A8}$$

where $\mathbf{K}_{xx} = \kappa(\mathbf{X}, \mathbf{X})$, $\mathbf{K}_{x*} = \kappa(\mathbf{X}, \mathbf{X}_*)$, $\mathbf{K}_{*x} = \kappa(\mathbf{X}_*, \mathbf{X})$ and $\mathbf{K}_{**} = \kappa(\mathbf{X}_*, \mathbf{X}_*)$. The predictive distribution $p(f_*|y, \mathbf{X}, \mathbf{X}_*, \sigma^2)$ is therefore distributed normal with the following mean and variance:

$$\mu_* = \mathbf{K}_{*x}(\mathbf{K}_{xx} + \sigma^2 \mathbf{I})^{-1} y, \tag{A9}$$

$$\sigma_*^2 = \mathbf{K}_{**} - \mathbf{K}_{*x}(\mathbf{K}_{xx} + \sigma^2 \mathbf{I})^{-1}\mathbf{K}_{x*} + \sigma^2. \tag{A10}$$

APPENDIX B: THE RELATION BETWEEN SPARSE GPS AND ANNS

An ANNs for regression is a special case of BFM where the basis functions are sigmoid activations, i.e. $\phi_j(x_i) = \text{sigmoid}(x_i p_j^T + b_j)$, where p_j plays the role of the weights between the input and the hidden neuron j. The activations of the m hidden units for the n samples in a single-layer ANN is essentially the Φ matrix. The weight parameters w in an ANN regressor are the connections between the hidden units and the output layer. For the neurons' bias terms, they can be simply incorporated by augmenting the input vector and the basis response vector with an additional constant value of 1. Thus, a single-layer ANN with m hidden units is a BFM with m basis functions, set as the sigmoid function, and an additional basis function with a constant output of 1. A main distinction between them, however, is that the weight parameters w in ANNs are treated as parameters of the models to be optimized and are not integrated out. Moreover, the objective function to be optimized in ANNs is different. Unlike the log marginal likelihood in BFMs, the objective in ANNs is to minimize the regularized sum of squares:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\|\Phi w - y\|^2 + \frac{\lambda}{2} w^T w + \frac{\lambda}{2}\sum_{j=1}^{m} p_j^T p_j, \tag{B1}$$

where $\theta = \{w, p_1, \ldots p_m, b_1, \ldots, b_m\}$ is the set of free parameters to be optimized. We recognize the first two terms as the negative of two terms in the log marginal likelihood defined in equation (9), with $\lambda = \alpha/\beta$. Note that unlike the proposed approach where we model each weight with its bespoke precision parameter and each input with its own predictive variance, typical ANNs implicitly assume a constant noise width. Moreover, λ is typically treated as an input parameter tuned by cross-validation rather than a parameter of the model to be optimized. Another distinction is that ANNs also minimize the norm of the weights in the hidden layer as well as the output layer with no penalty on the bias terms. Moreover, the $\ln|\Sigma|$ term is missing from equation (B1), which is very crucial as it drives the optimization process towards reducing the uncertainty on the parameter w, thus producing more confident models with more accurate variance prediction.

APPENDIX C: OPTIMIZATION OF SPARSE GPS

To ensure that the $\boldsymbol{\alpha}$s and $\boldsymbol{\eta}$s are positive, we optimize with respect to the log of the parameters. We refer to the set of free parameters to be optimized as $\boldsymbol{\theta} = \{\mathbf{P}, \Gamma_1, \ldots, \Gamma_m, u, b, \ln \boldsymbol{\alpha}, \ln \boldsymbol{\eta}\}$. The derivative of the log marginal likelihood in equation (20) with respect to each parameter θ_i can be found by computing the following in the order:

$$\frac{\partial \Sigma}{\partial \theta_i} = \Phi^T \left(\frac{\partial \mathbf{B}}{\partial \theta_i}\Phi + 2\mathbf{B}\frac{\partial \Phi}{\partial \theta_i}\right) + \frac{\partial \mathbf{A}}{\partial \theta_i}, \tag{C1}$$

$$\frac{\partial \bar{w}}{\partial \theta_i} = \Sigma^{-1}\left(\Phi^T \frac{\partial \mathbf{B}}{\partial \theta_i} y + \frac{\partial \Phi^T}{\partial \theta_i}\mathbf{B}y - \frac{\partial \Sigma}{\partial \theta_i}\bar{w}\right), \tag{C2}$$

$$\frac{\partial \boldsymbol{\delta}}{\partial \theta_i} = \frac{\partial \Phi}{\partial \theta_i}\bar{w} + \Phi\frac{\partial \bar{w}}{\partial \theta_i}, \tag{C3}$$

738   *I. A. Almosallam, M. J. Jarvis and S. J. Roberts*

$$\frac{\partial \ln p(y)}{\partial \theta_i} = -\frac{1}{2}\delta^T\left(\frac{\partial \mathbf{B}}{\partial \theta_i}\delta + 2\mathbf{B}\frac{\partial \delta}{\partial \theta_i}\right)$$
$$-\frac{1}{2}\bar{w}^T\left(\frac{\partial \mathbf{A}}{\partial \theta_i}\bar{w} + 2\mathbf{A}\frac{\partial \bar{w}}{\partial \theta_i}\right)$$
$$-\frac{1}{2}u^T\left(\frac{\partial \mathbf{N}}{\partial \theta_i}u + 2\mathbf{N}\frac{\partial u}{\partial \theta_i}\right)$$
$$-\frac{1}{2}\text{trace}\left(\Sigma^{-1}\frac{\partial \Sigma}{\partial \theta_i}\right) + \frac{1}{2}\text{trace}\left(\mathbf{B}^{-1}\frac{\partial \mathbf{B}}{\partial \theta_i}\right)$$
$$+\frac{1}{2}\text{trace}\left(\mathbf{A}^{-1}\frac{\partial \mathbf{A}}{\partial \theta_i}\right) + \frac{1}{2}\text{trace}\left(\mathbf{N}^{-1}\frac{\partial \mathbf{N}}{\partial \theta_i}\right). \quad \text{(C4)}$$

The derivative computation provided in equation (C4) is the general form for computing the gradient for any basis function definition, the only difference is the definition of $\frac{\partial \Phi}{\partial \theta}$. However, if computed naively, the computation can be time consuming since the partial derivatives will be mostly zeros for any given parameter in θ and some of the same operations are repeated. In the next section, we provide a more efficient way to compute the gradient for RBF basis functions.

C1 Efficient optimization

For the case of RBF basis functions, we can compute the partial derivatives more efficiently by first defining $\Delta_j = \mathbf{X} - 1_n p_j$, where 1_n is a vector of length n consisting of all ones. We also first derive the partial derivatives with respect to $\bar{w}$, $\ln \beta$ and $\ln \Phi$:

$$\frac{\partial \bar{w}}{\partial \ln \alpha} = -\Sigma^{-1}\mathbf{A}\bar{w}, \quad \text{(C5)}$$

$$\frac{\partial \ln p(y)}{\partial \ln \beta} = -\frac{1}{2}\mathbf{B}\delta^2 - \frac{1}{2}\mathbf{B}(\Phi \circ (\Phi\Sigma^{-1}))1_m + \frac{1}{2}, \quad \text{(C6)}$$

$$\frac{\partial \ln p(y)}{\partial \ln \Phi} = \left(\frac{\partial \ln p(y)}{\partial \ln \beta}u^T - \mathbf{B}\delta\bar{w}^T - \mathbf{B}\Phi\Sigma^{-1}\right) \circ \Phi, \quad \text{(C7)}$$

where $\delta^p = \{\delta_i^p\}_{i=1}^m$ and similarly for other vectors. The symbol $\circ$ denotes the Hadamard product, i.e. element-wise matrix multipli-

cation. The partial derivatives, with respect to the parameters u, b, $\ln \alpha$ and $\ln \eta$, are as follows:

$$\frac{\partial \ln p(y)}{\partial u} = \Phi^T\frac{\partial \ln p(y)}{\partial \ln \beta} - \mathbf{N}u, \quad \text{(C8)}$$

$$\frac{\partial \ln p(y)}{\partial b} = \sum_{i=1}^n \frac{\partial \ln p(y)}{\partial \ln \beta_i}, \quad \text{(C9)}$$

$$\frac{\partial \ln p(y)}{\partial \ln \eta} = -\frac{1}{2}\mathbf{N}u^2 + \frac{1}{2}, \quad \text{(C10)}$$

$$\frac{\partial \ln p(y)}{\partial \ln \alpha} = -(\Phi^T\mathbf{B}\delta) \circ \frac{\partial \bar{w}}{\partial \ln \alpha} - \frac{1}{2}\mathbf{A}\bar{w}^2$$
$$-\mathbf{A}\bar{w} \circ \frac{\partial \bar{w}}{\partial \ln \alpha} - \frac{1}{2}\text{diag}(\mathbf{A}\Sigma^{-1}) + \frac{1}{2}. \quad \text{(C11)}$$

The partial derivatives, with respect to the parameters Γ_j and p_j of the pseudo points, can be computed as follows

$$\frac{\partial \ln p(y)}{\partial p_j} = \frac{\partial \ln p(y)}{\partial \ln \Phi}[:,j]^T \Delta_j \Gamma_j^T \Gamma_j, \quad \text{(C12)}$$

$$\frac{\partial \ln p(y)}{\partial \Gamma_j} = -\Gamma_j\left(\Delta_j \odot \frac{\partial \ln p(y)}{\partial \ln \Phi}[:,j]\right)^T \Delta_j, \quad \text{(C13)}$$

where $\mathbf{A} \odot v$ denotes a broadcast multiplication, i.e. an element-wise multiplication between the vector v and each column vector in $\mathbf{A}$. Note that if all basis are forced to share the same parameter Γ, then the partial derivative with respect to it is
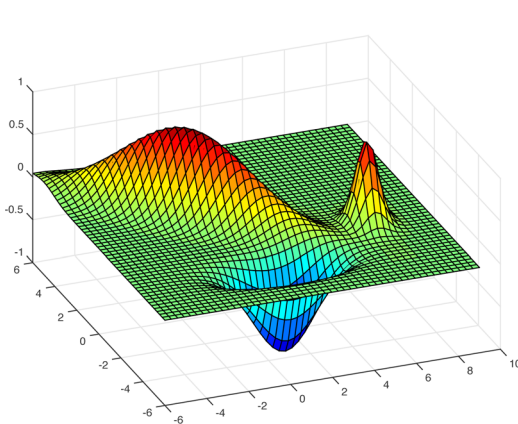
$$\frac{\partial \ln p(y)}{\partial \Gamma} = \sum_{j=1}^m \frac{\partial \ln p(y)}{\partial \Gamma_j}, \quad \text{(C14)}$$

we can also force Γ_j to be a diagonal covariance, in which case the partial derivatives with respect to each $\text{diag}(\Gamma_j)$:
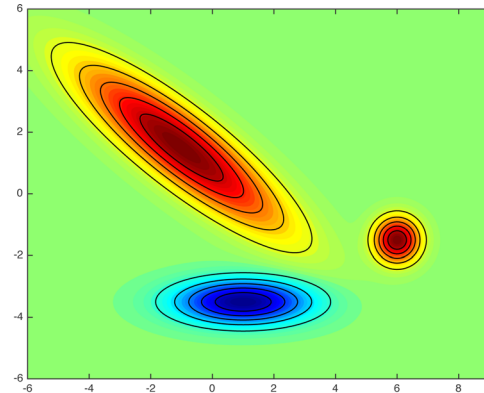
$$\frac{\partial \ln p(y)}{\partial \text{diag}(\Gamma_j)} = \text{diag}\left(\frac{\partial \ln p(y)}{\partial \Gamma_j}\right), \quad \text{(C15)}$$

similarly, the basis functions can be forced to share a global diagonal $\text{diag}(\Gamma)$:

$$\frac{\partial \ln p(y)}{\partial \text{diag}(\Gamma)} = \sum_{j=1}^m \text{diag}\left(\frac{\partial \ln p(y)}{\partial \Gamma_j}\right). \quad \text{(C16)}$$

(a) Target function

(b) Top view showing the true centres and covariances of the generating basis functions

Figure C1. A 2D synthetic example to illustrate the performance difference between GPVC, GPGC, GPVD, GPGD, GPVL and GPGL. The results are shown in Fig. C2.

MNRAS **462**, 726–739 (2016)

Downloaded from https://academic.oup.com/mnras/article-abstract/462/1/726/2589544
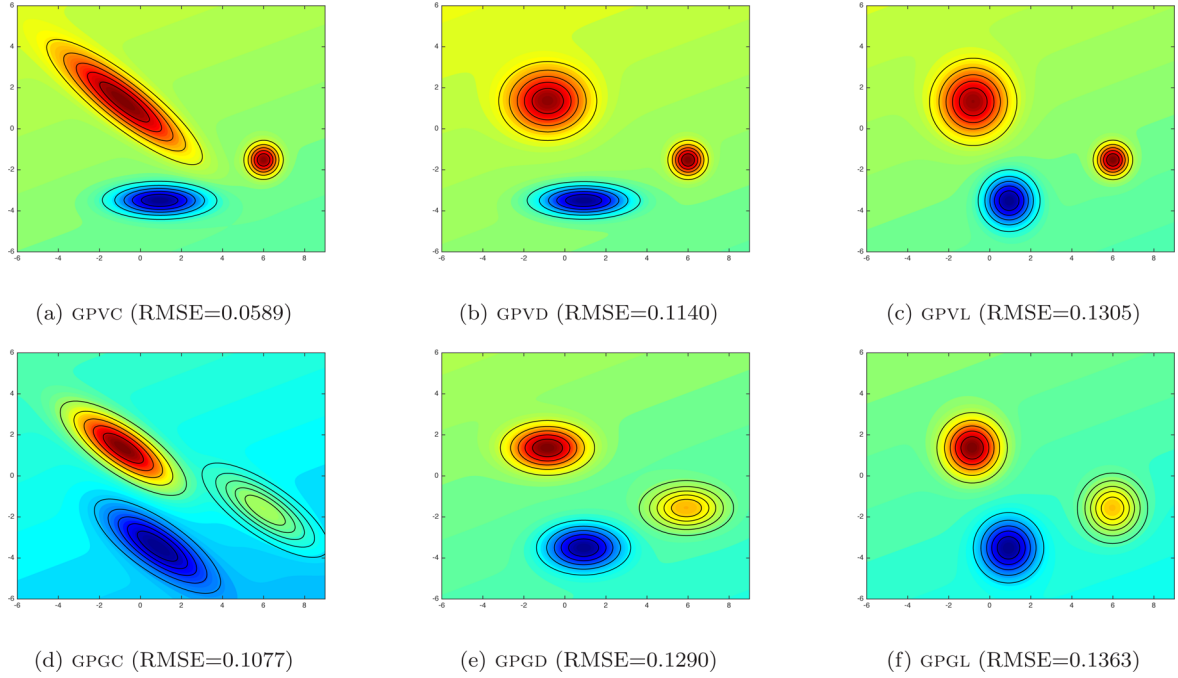by Said Business School user
on 15 January 2018

Figure C2. The results of running (a) GPVC, (b) GPVD, (c) GPVL, (d) GPGC, (e) GPGD and (f) GPGL using three basis functions on the synthetic 2D regression example in Fig. C1. The RMSE performance on a held out test set for each are reported in sub-captions.

In the case of variable length-scales, where Γ_j is a scalar value γ_j, the partial derivative with respect to each γ_j is

$$\frac{\partial \ln p(y)}{\partial \gamma_j} = \sum_{k=1}^{d} \frac{\partial \ln p(y)}{\partial \Gamma_j}[k,k], \qquad (C17)$$

and to force all basis functions to have a global length-scale γ, the partial derivative is computed as follows

$$\frac{\partial \ln p(y)}{\partial \gamma} = \sum_{j=1}^{m}\sum_{k=1}^{d} \frac{\partial \ln p(y)}{\partial \Gamma_j}[k,k]. \qquad (C18)$$

The framework thus allows for six different configurations, variable full covariances (GPVC) as in equation (C13), a global full covariance (GPGC) as in equation (C14), variable diagonal covariances (GPVD) as in equation (C15), a global diagonal covariance (GPGD) as in equation (C16), variable scalar length-scales (GPVL) as in equation (C17) and a global scalar length-scale (GPGL) as in equation (C18). The six configurations are all special cases of equation (C13); however, the computational cost can be greatly reduced by taking advantage of the simpler structures of the other configurations.

We illustrate the difference between the different configurations of the model, namely GPVC, GPGC, GPVD, GPGD, GPVL and GPGL, using a synthetic 2D example shown in Fig. C1. The target function to be modelled is a linear combination of three basis function with different centres and covariances $f(x, y) = \phi_1(x, y) + \phi_2(x, y) - \phi_3(x, y)$. The different configurations were trained on examples of x and y as inputs and $f(x, y)$ as the target output plus some additive noise, the results are shown in Fig. C2. It is not surprising that GPVC performed the best, as it has more flexibility in modelling the covariance of each basis function. The other configurations would require more basis functions compared to GPVC to achieve the same accuracy.

APPENDIX D: SQL STATEMENT

The following SQL statement was used to extract the data from the SDSS DR12 data base using the CasJobs service provided by SDSS.[1]

```
SELECT
p.objid,
p.modelMag_u, p.modelMag_g,
p.modelMag_r, p.modelMag_i,
p.modelMag_z, p.modelMagerr_u,
p.modelMagerr_g, p.modelMagerr_r,
p.modelMagerr_i, p.modelMagerr_z,
s.z as zspec, s.zErr as zspecErr
INTO
mydb.modelmag_data set
FROM
PhotoObjAll as p, SpecObj as s
WHERE
p.SpecObjID = s.SpecObjID AND
s.class = 'GALAXY' AND
s.zWarning = 0 AND
p.mode = 1 AND
dbo.fPhotoFlags('PEAKCENTER') != 0 AND
dbo.fPhotoFlags('NOTCHECKED') != 0 AND
dbo.fPhotoFlags('DEBLEND_NOPEAK') != 0 AND
dbo.fPhotoFlags('PSF_FLUX_INTERP') != 0 ; AND
dbo.fPhotoFlags('BAD_COUNTS_ERROR') != 0 AND
dbo.fPhotoFlags('INTERP_CENTER') != 0
```

[1] casjobs.sdss.org

This paper has been typeset from a TEX/LATEX file prepared by the author.