Mon. Not. R. Astron. Soc. **000**, **??**–**??** (2015) Printed 26 February 2015 (MN LATEX style file v2.2)

Photometric Redshift Estimation

Author1,[1⋆] Author2,[2] etc.

[1] *Oxford Astrophysics, Department of Physics, Keble Road, Oxford, OX1 3RH, UK*
[2] *Other institution*

26 February 2015

ABSTRACT

In this study, a novel sparse regression model for photometric redshift estimation is presented which directly target the requirements for the Euclid Space Mission. Data from a synthesised survey was used to train and test the proposed models. We show that approaches which include careful data preparation and model design, a significant improvement can be achieved when compared with several off the shelf machine learning algorithms. Standard implementation of most regression algorithms has as objective the minimisation of the sum of squared errors. This induces a bias in the posterior mean of the output distribution which can be problematic. In this paper we directly optimise the Euclid mission requirement and compare this with other objective functions, such as minimising the maximum error and maximising the number of data points with a predictive error less than a threshold. The results are compared with other popular machine learning algorithms in the field such as ANNz, stableGP and SPGP. The proposed reached a $\Delta z = 0.0085(1+z)$ for a redshift range of $0.2 < z < 2$, exceeding the requirement for the Euclid mission of $\Delta z = 0.05(1+z)$ for the same redshift range.

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

1 INTRODUCTION

We introduce a novel sparse kernel regression model that greatly reduces the number of bases functions required to model the data. This os achieved this by allowing each kernel to have its own hyper-parameters, governing its shape. This is in contrast to the standard kernel-based model in which a set of global hyper-parameters are optimised. The complexity cost of such a kernel-based regression model is $O(n^3)$, where n is the number of bases functions. This cubic time complexity arise from the cost of inverting a n by n covariance matrix. In a basic Gaussian Process model (GP) **?**, seen as a kernel regression algorithm, we may regard the number of bases, n, as equal to the number of points in the training set. This renders such an approach unusable for many large-data applications where scalability is a major concern. Much of the work done to make GPs more scalable **?** is either to make the inverse computation faster or use smaller representative sample to compute the covariance. Examples of the former includes methods such as structuring the covariance matrix such that it is much easier to invert , using Toeplitz and Kronecker decomposition for example, or inverse approximation as an optimisation problem . To reduce the number of representative points, an $m << n$ subset of the training set can be selected which maximises the accuracy or the numerical stability of the inversion. Alternatively, one may search for "pseudo" points not necessarily present in the training set to use as bases for the covariance matrix such that it maximises the log marginal likelihood .

The focus in this paper is on sparse GP modelling where we extend the sparse pseudo-point GP using less, but more flexible kernels. Moreover, a weighting scheme is modelled as an integral part of the process to remove, or introduce, any systematic bias to the model. The results are demonstrated on photometric redshift estimation for the Euclid Space Mission. In particular, we use the weighting scheme to remove any distribution bias and introduce a linear bias to directly target the mission's requirement. The proposed approach reached a $\Delta z = 0.0085(1+z)$ on a simulated catalogue, far exceeding the mission's requirement of $\Delta z = 0.05(1+z)$ The paper is organised as follows, a brief introduction to Gaussian Processes for regression and sparse GPs are presented in sections 2 and 3, then the proposed approach is laid out in Section 4. We discuss other objectives that can be useful for other scientific goals in 5 then the data set description and experiments are provided in Sections 6 and 7 respectively. We summarise and conclude in Section 8.

2 GAUSSIAN PROCESSES

A Gaussian Process is a supervised non-linear regression algorithm that makes few explicit *parametric* assumptions about the nature of the function fit. For this reason, Gaussian Processes are seen as lying within the class of Bayesian non-parametric models. The main underlying assumption in a GP is that the joint probability of the input variable x and the output variable y is a multivariate Gaus-

⋆ E-mail: ibrahim.almosallam@some.ox.ac.uk

sian with mean $\mu = \begin{bmatrix}\mu_x \\ \mu_y\end{bmatrix}$ and covariance $\Sigma = \begin{bmatrix}\Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy}\end{bmatrix}$, where $\Sigma_{xy} = (x - \mu_x)(y - \mu_y)^T$. The input variables x is an n by d matrix, where n is the number of data points and d is the dimensionality of the input. Without lost of generality, the output variable y is assumed to be a vector of length n of target outputs but the same concept can be applied to multiple variable output.

$$p\left(x, y\right) \sim \mathcal{N}\left(\begin{bmatrix}\mu_x \\ \mu_y\end{bmatrix}, \begin{bmatrix}\Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy}\end{bmatrix}\right) \tag{1}$$

The mean and covariance of the conditional probability $p(y|x)$ therefore is Gaussian distributed as follows:

$$p(y|x) \sim \mathcal{N}\left(\mu_x + \Sigma_{yx}\Sigma_{xx}^{-1}\left(y - \mu_y\right), \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}\right) \tag{2}$$

The calculation can be simplified by subtracting the mean of the input and the output variables and assuming a prior mean $\mu_x = \mu_y = 0$ and Σ_{xy} redefined as xy^T. The conditional probability $p(y|x)$ can then be rewritten as:

$$p(y|x) \sim \mathcal{N}\left(\Sigma_{yx}\Sigma_{xx}^{-1}y, \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}\right) \tag{3}$$

For the rest of this paper, the prior mean is assumed to be zero unless otherwise stated. So far, it is assumed that no noise exists in our y observations. It can be shown that assuming some noise $\epsilon \sim \mathcal{N}\left(0, \sigma_n^2\right)$ on the output variable y, yields a the following:

$$p(y|x) \sim \mathcal{N}\left(\Sigma_{yx}\left(\Sigma_{xx} + I\sigma_n^2\right)^{-1}y, \Sigma_{yy} - \Sigma_{yx}\left(\Sigma_{xx} + I\sigma_n^2\right)^{-1}\Sigma_{xy} + \sigma_n^2\right) \tag{4}$$

For our current definition of the covariance matrix Σ, the predictive mean is equivalent to a linear regression model, in fact, one can reach the same conclusion by finding the linear regression model that minimizes the sum of squared errors. In other words, the model that minimizes the sum of squared errors also maximizes the probability of the data. For in depth discussion on Gaussian processes and its Bayesian interpretation, the reader is referred to [x].

Since the solution is entirely defined as inner products of the data points, one can utilize the so called "kernel trick" to learn nonleaner models by replacing the covariance matrix Σ with a covariance function K, where $K_{ij} = k(x_i, x_j)$. For a proper definition of the kernel function k, the matrix K will be a positive semi-definite matrix and therefore invertible. The concept behind the kernel trick is to compute the covariance matrix of high dimensional mapping of the input into a higher dimensional space without explicitly mapping them to that space. The choice of kernel is largely a modeling decision based on the definition of similarity for a given application. In this paper, the squared exponential kernel defined in (5) is used but the concepts introduced here applies to any other kernel function.

$$k(x_i, x_j) = \sigma^2 e^{-\frac{\sum_{k=1}^{d}(x_{ik} - x_{jk})^2}{2\lambda^2}} \tag{5}$$

The hyper-parameters of the squared exponential kernel σ^2 and λ^2 are called the height variance and characteristic length scale respectively. Together with the noise variance σ_n^2 define the set of hyper-parameters for the GP model. The optimal set of hyper-parameters are the set of values that maximizes the probability of the data given the model, which can be achieved by maximizing the log marginal likelihood defined in (6)

$$log\, p(y|x) = -\frac{1}{2}y^T\left(K + I\sigma_n^2\right)^{-1}y - \frac{1}{2}log\left|K + I\sigma_n^2\right| - \frac{n}{2}log(2\pi) \tag{6}$$

This can be achieved by using a gradient search based optimization by taking the derivative of the log marginal likelihood with respect to each hyper-parameter and following the direction of the gradient.

3 SPARSE GAUSSIAN PROCESS

Gaussian processes are often described as non-parametric regression models due to the analytical nature of its solution and its use of very limited number of hyper-parameters that live in the kernel. However, GP regression can be viewed as feature transformation methods $x \in \mathbb{R}^d :\rightarrow K \in \mathbb{R}^n$ parameterized by the data and the kernel function followed by a linear regression, or optimizing the following objective:

$$f(w) = \frac{1}{2}\left(Kw - y\right)^T\left(Kw - y\right) + \frac{1}{2}\sigma_n^2 w^T w \tag{7}$$

The feature transformation K is essentially describing each data point by how "similar" it is to every point in the training set where the similarity measure is defined by the kernel function. Obviously, if two training points are very similar, that will result in very correlated features and thus adding extra computational cost for very little or no added information. Selecting a subset of the training set that maximizes the preserved information is a research question addressed in [x], whereas in [x] the basis functions are treated as an optimization problem rather than a selection problem where their locations are treated as hyper-parameters. The new transformation is now $x \in \mathbb{R}^d :\rightarrow K \in \mathbb{R}^m$ where $m << n$ is the number of bases used. The transformation matrix K will therefore be a rectangular n by m matrix and the solution for w in (7) is calculated as follows:

$$w = \left(K^T K + I\sigma_n^2\right)^{-1}K^T y \tag{8}$$

Even though these models improve the computational cost greatly, very little is done to compensate for the reduction in modeling complexity. The selection method is always bounded by the full GP's accuracy. On the other hand, the sparse GP's ability to place the bases freely across the input space does compensate for the bases set size reduction as they can be better positioned to describe the distribution of the data. However, in both cases a global set of hyper-parameters is used for all basis functions therefore limiting the algorithm's capability to model patterns in the data with the number of basis and their locations. Moreover, the objective in (7) by definition minimizes the sum of squared errors, therefore for any non-uniformly distributed output, the optimization routine will bias the model towards the mean of the output distribution and to fit the region of space where there is more data.

In the next section, the proposed method is described which address the above issues by defining each basis with its own hyper-parameters to account for variable density and/or pattern across the input space and allows the algorithm to learn more complex models with a fewer number of basis functions. In addition, a weighting mechanism to remove any distribution bias from the model is directly incorporated into the objective.

4 PROPOSED APPROACH

In this paper, we extend the sparse GP approach by modeling each basis with its own set of hyper-parameters, namely the kernel function in (5) is redefined as follows:

$$k(x_i, p_j) = e^{-\frac{\|x_i - p_j\|^2}{2\lambda_j^2}} \tag{9}$$

Where $P = \{p_j\}_{j=1}^m$ is the set of bases coordinates in $\mathbb{R}^d$ and λ_j is the corresponding length scale for basis j. Note that the hyper-parameter σ has been dropped, as it interferes with the regularization objective. This can be seen from the final prediction equation $f(x_i, w) = \sum_{j=1}^m w_j \sigma_j^2 e^{-\|x_i - p_j\|^2/2\lambda_j^2}$, the weights are always multiplied by their associated σ. Therefore, the optimization process will always compensate for decreasing w_j^2 by increasing σ_j^2. Dropping the height variance ensures that the kernel functions do not grow beyond control and delegates learning the linear coefficients and regularization to the weights w_j. A gradient based optimization is used to find the set of hyper-parameters by computing the derivatives with respect to each variable to compute the search direction, specifically we used a Quasi-Newton optimization using the L-BFGS algorithm . The derivative with respect to each length scale and position are provided in equations (10a) and (10b) respectively.

$$\frac{\partial f(x, w)}{\partial \lambda_j} = \sum_{i=1}^n \left(f(x_i, w) - y_i\right) w_j K_{ij} \frac{\|x_i - p_j\|^2}{\lambda_j^3} \tag{10a}$$

$$\frac{\partial f(x, w)}{\partial p_{jk}} = \sum_{i=1}^n \left(f(x_i, w) - y_i\right) w_j K_{ij} \frac{(x_{ik} - p_{jk})}{\lambda_j^2} \tag{10b}$$

Finding the set of hyper-parameters that optimizes the solution, is in effect finding the set of radial bases defined by their positions p and radius λ that jointly describes the patterns across the input space, and by parameterizing them differently, the model is more capable to accommodate different regions in the space more specifically. The kernel in (9) can be further extended to, not only model each basis with its own radius λ_j, but also model each one with its own covariance defined by C_j. This enables the bases to have any arbitrary shaped ellipses given it more flexibility. The kernel in (9) can be extended as follows:

$$k(x_i, p_j) = e^{-\frac{1}{2}\left(x_i - p_j\right) C_j^{-1} \left(x_i - p_j\right)^T} \tag{11}$$

To make the optimization process faster and simpler, we define the following variables:

$$E = \left(f(x, w) - y\right) w^T \tag{12a}$$

$$C_j^{-1} = \Lambda \Lambda_j^T \tag{12b}$$

$$\Delta_j = \left(x - \underset{n}{1} p_j\right) \tag{12c}$$

$$V_j = \Delta_j \Lambda_j \tag{12d}$$

Where $\underset{n}{1}$ denotes a column vector of length n with all elements set to 1. Optimizing with respect to Λ_j directly ensures that the covariance matrix is positive semi-definite and makes it faster from a computational perspective, as the kernel functions for all the points with respect to a particular basis can be computed more efficiently as follows:

$$k(x, p_j) = e^{-\frac{1}{2}\left(V_j \circ V_j\right) \underset{d}{1}} \tag{13}$$

The symbol $\circ$ denotes the Hadamard product or the element-wise matrix multiplication. The exponent in (13) basically reads, square the elements of the matrix V_j then compute the sum of each row. This allows for a more efficient computation of the kernel functions for all points in a single matrix operation. The derivatives with respect to each Λ_j and p_j are shown in (14a) and (14b)

$$\frac{\partial f(x, w)}{\partial \Lambda_j} = -\left(\Delta_j^T \circ \left(\underset{d}{1} \left(K_{*,j} \circ E_{*,j}\right)^T\right)\right) V_j \tag{14a}$$

$$\frac{\partial f(x, w)}{\partial p_j} = \left(K_{*,j} \circ E_{*,j}\right)^T V_j \Lambda_j^T \tag{14b}$$

Where $K_{*,j}$ is the j'th column of matrix K, similarly for $E_{*,j}$. To illustrate the differences between the three approaches, an classification example is shown in **??** where all models were initialized with the same set of four bases. The class probability distribution clearly show the advantage of having bases functions with more flexibility.

4.1 Simple to Complex Modeling

5 OTHER OBJECTIVES

5.1 Minimizing the Maximum

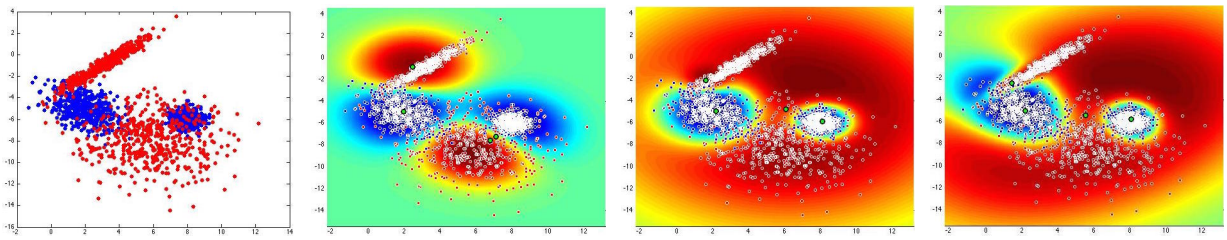
5.2 Maximizing the number of fitting samples

6 DATASET DESCRIPTION

7 EXPERIMENTS AND RESULTS

8 CONCLUSION

REFERENCES



(a) Dataset (b) Global Length Scale (c) Variable Length Scale (d) Variable Covariance

Figure 1. A comparisons between different sparse GP approaches