Photometric Redshift Estimation

Authors

February 2015

Abstract

In this study, we present a novel sparse regression model for photometric redshift estimation are 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.

1 Introduction

We introduce a novel sparse kernel regression model that greatly reduces the number of basis functions required to model the data. We achieve 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 basis functions. This cubic time complexity arise from the cost of inverting a $n \times n$ 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, a $m << n$ 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 basis 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, related works are discussed in Section **??** and in Section **??** a brief introduction to Gaussian Processes for regression is presented then the proposed approach is laid out in Section **??**. We discuss other objectives that can be useful for other scientific goals in **??** then the data set description and experiments are provided in Sections **??** and **??** respectively. We summarise and conclude in Section **??**

2 Related Work

3 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 withing 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 Gaussian 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 $n \times d$ matrix, where n is the number of data point and d is the dimensionality of the input. Without lost of generality, the output variable y is assumed to be an $n \times 1$ vector of desired output

but the same method 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 $\mu = 0$ 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:

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

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 with the least 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 non-leaner 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 space 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 (**??**) is used but the concepts introduced here applies to any other kernel function.

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

The hyper-parameters of the squared exponential kernel σ_h^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 (**??**)

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

where n is the number of data points in the training set x. The hyper-parameters can be found using a gradient descent based optimization by taking the derivative of the log marginal likelihood with respect to each hyper-parameter and following the direction of the gradient.

4 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 $x \in R^d :\rightarrow K \in R^n$ parameterized by the data and the kernel function followed by a linear regression. or optimizing the following objective:

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

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 extra computation 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 the basis functions' locations are treated as hyper-parameters. The new transformation is now $x \in R^d :\rightarrow K \in R^m$ where $m << n$. The transformation matrix K is therefore a rectangular $n \times m$ matrix and the solution for w in (**??**) is calculated as follows:

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

Even though these models improve the computational cost greatly, very little is done to compensate for the loss of accuracy. In the sparse GP implementation, the ability to allow the points to move in space and not be restricted to points in the training set does some compensation. Furthermore, a single kernel function with a global set of hyper-parameters is used, which makes the assumeption that there is no pattern change across the input space. Moreover, the objective in (**??**) by definition minimizes the sum of squared errors, therefore for any non-uniformaly distributed output, the optimization routine will bias the model towards the mean of the output distribution and the region of space where there is more data.

In the next section, the proposed method is described which address all the above questions by defining each basis with its own hyper-parameters to address the problem of variable density and pattern across the input space and incorporates a weighting mechanism to remove any distribution bias from the model.

5 Proposed Approach

5.1 Simple to Complex Modeling

6 Other Objectives

6.1 Minimizing the Maximum

6.2 Maximizing the number of fitting samples

7 Dataset Description

8 Experiments and Results

9 Conclusion