Gaussian Processes for regression: a tutorial

José Melo
Faculty of Engineering, University of Porto
FEUP - Department of Electrical and Computer Engineering
Rua Dr. Roberto Frias, s/n 4200-465
Porto, PORTUGAL
jose.melo@fe.up.pt

Abstract

Gaussian processes are a powerful, non-parametric tool that can be be used in supervised learning, namely in regression but also in classification problems. The main advantages of this method are the ability of GPs to provide uncertainty estimates and to learn the noise and smoothness parameters from training data. The aim of this short tutorial is to provide the basic theoretical aspects of Gaussian Processes, as well as a brief practical overview on implementation.

The main motivation of this work was to develop a new approach to detect outliers on acoustic navigation algorithms for Autonomous Underwater Vehicles, capable of adjusting to different operation scenarios, since this is a major problem in the majority of Autonomous Underwater Vehicles. In the last part of the tutorial, a brief insight on this actual problem, and the solution proposed, that involves Gaussian Processes as a predictor, and some background subtraction techniques is described.

1. Introduction

In the machine learning context, supervised learning is concerned with inferring the values of one or more outputs, or response variables, for a given set of inputs that have not yet been observed, or predictor variables [4]. These predictions are based on the training samples of previously solved cases. Depending on whether the output is continuous or discrete, we talk about regression or classification problems, respectively. Traditional approaches to solve this kind of problem usually consist on parametric models, on which the behaviour of data is described by a previously defined model, and the parameters of this model are learned from the training data. By adjusting these parameters, it is possible to fit the model to the data. Once this is done, it should be straightforward to use the model to and predict the output if new inputs are provided. Both linear and non-linear regression techniques have been extensively used for this purpose, using different estimation techniques to fit the data, namely several different flavours of the least-squares algorithms, ridge regression, etc.

Despite all the advantages of these traditional regression techniques, in all of them it is necessary to make assumptions about the the smoothness of our model. While incorporating prior knowledge in the model that correctly describes the evolution of the data we have can be of great value, sometimes this information is just not available. And using a model that does not correctly characterizes the data is likely to lead to poor results.

A completely different approach is given by Gaussian Processes, by neglecting the parametric model viewpoint and instead define a prior probability distribution over all possible functions directly [3]. This paper has a strong focus on introducing the use of Gaussian Process in regression, and is is organised as follows. In Section 2 the basic principles of the Gaussian Processes are given. In section 3, prediction with Gaussian Processes is derived, and learning with Gaussian Processes is covered in section 4. In section 4 we describe the application and in section 5 the results obtained. In the end we present some conclusions and future work directions.

2. Gaussian Processes

Gaussian Processes (GPs) are powerful non-parametric technique with explicit uncertainty models, that finds its used mainly in regression and classification problems. The reason because they are non-parametric is because instead of trying to fit the parameters of a selected basis functions, instead GPs rather try to infer how all the measured data is correlated.

A GP is, by definition, a collection of random variables with the property that the joint distribution of any of its subset is joint Gaussian distribution. At this point, it is important to make a clear distinction between a Gaussian distribution and a Gaussian process.

A Gaussian distribution is a continuous probability distribution, informally known as the "bell shape curve", and fully specified by a mean and a covariance: $x \backsim \mathcal{N}(\mu, \sigma)$. Moreover, a uni-variate Gaussian distribution can be defined by the function:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

Gaussian Processess, on the other can, can be though of a generalization of the Gaussian probability distribution to infinitely many variables. A Gaussian process is a Gaussian random function, and is fully specified by a mean function $m(x)$ and covariance function $k(x, x')$:

$$f(x) \backsim \mathcal{GP}(m(x), k(x, x') \tag{2}$$

It is clear then the correspondence between GPs and Gaussian distributions. The representation given by (2) means that *"the function f is distributed as a GP with mean function m and covariance function k"* [13]. To define an individual GP, one needs to choose a form for the mean function, $m(x)$, and for the covariance function $k(x, x')$.

In most applications there is no prior knowledge about the mean functon, $m(x)$ of a given Gaussian Process. By simplicity, and because GPs are, by definition, a linear combination of random variables with Normal Distribution, this is commonly assumed to be zero [3]. If there is, however, enough information about the process we are modelling such that the mean function should be explicitly different than zero, this can be done in a very trivial way, without loss of the results presented below.

The covariance function, $k(x, x')$, can be in general any function that takes any two arguments, such that $k(x, x')$ generates a nonnegative definitive covariance matrix K. By choosing the covariance function, one is implicitly making underlying assumptions about certain aspects of the process being modelled, such as smoothness, periodicity, stationarity, among others. Obviously there are great set of possible covariance functions, but one that is most frequently used is the squared exponential covariance function:

$$k(x, x') = \sigma_f^2 exp(-\frac{1}{2l^2}|x - x'|^2) \tag{3}$$

This covariance function is also sometimes referred to as Radial Basis Function. It is easy to see that for equation 3 the covariance between any two inputs is really close to one if the inputs are close to each other, and decreases exponentially as the distance between the inputs increases. Here, σ_f and l are what we call the hyperparameters, mainly due to the resemblance to the hyperparameters of a Neural Network. In most cases, the choice of parameters can significantly influence the performance of the GP. It can be shown that using the squared exponential as a covariance function is equivalent to regression using infinitely many Gaussian shaped basis functions placed everywhere, and not just the training points [16].

The output of the Gaussian process model is a normal distribution, expressed in terms of mean and variance. The mean value represents the most likely output and the variance can be interpreted as the measure of its confidence.

3. Prediction with Gaussian Processes

Prediction problems are most of the time related with events occurring in a time-series . A typical example of a prediction problem can be stated in the following manner: given some observations $\{y_1, y_2, \ldots, y_N\}$ of a dependent variable, subject to noise, at certain time-instants $\{x_1, x_2, \ldots, x_N\}$, what is our best estimate of the dependent variable at a new time-instant x_{N+1}? In the Gaussian Process framework, the inputs would be the vector $X = \{x_1, x_2, \ldots, x_N\}$, and the test points would be the vector X_*, composed by all the points we want to predict and, in this case, only x_{N+1}.

If we are ready to make assumptions about the underlying model of the observed values follow, this problem is usually tackled by using traditional linear regression methods. However, if no assumptions are taken related to the distributions of the observations, then Gaussian Processes are likely to be a better choice when comparing to their parametric counterpart ones.

Lets consider we are in present of a set of observations y, on which each element is a sample from a Gaussian Distribution, representing the real value of the observation affected by some independent Gaussian noise ϵ with variance σ_n. We can then think on the observations as being the sum of a function plus an additive gaussian noise:

$$y = f(x) + \epsilon \tag{4}$$

Given this, the objective is now to predict f_*, expected value given the test input x_*. Recalling that a Gaussian Process is a set of random variables which have a consistent Gaussian distribution with mean zero, we can represent our problem as:

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \backsim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right). \tag{5}$$

Here, the different K matrix are built using any function $k(x, x')$ able to perform as a covariance function. In particular, as we are in the presence of observations corrupted with noise, the covariance between any two observations is given by:

$$cov(y_p, y_q) = k(x_p, x_q) + \sigma_n^2 \delta_{pq} \tag{6}$$

In equation 6 δ_{pq} is the Kronecker delta, which is a function of two variables equal to 1 if and only if both its inputs are equal, and 0 otherwise. By combining (3) and (6) we are now ready to build For the vector of inputs x, the covariance of the associated observations is given by equation 7 and by combinnit should be noted that the diagonal elements of K is $\sigma_f^2 + \sigma_n^2$

$$cov(y) = K(X, X) + \sigma_n^2 I \tag{7}$$

The prediction step consists in estimating the mean value and the variance for y_*. Considering equation 5, it is obvious that want is desired is to estimate the conditional distribution of y_* given y. An interesting result is that Remember that y and y_* are jointly Gaussian random vectors, then the conditional distribution of y_* given y is given by equation (8). For the simplicity of the notation, in (9) and (10), we used $k = K(x, x_*)$, $C_N = K(X, X) + \sigma_n^2 I$ and $k_{**} = K(X_*, X_*)$.

$$y_*|y \backsim \mathcal{N}(\bar{f}_*, cov(f_*)) \tag{8}$$

$$\bar{f}_* = k_*^T C_N^- 1y \tag{9}$$

$$cov(f_*) = k_{**} - k_*^T C_N^{-1} k_* \tag{10}$$

The mean value of the prediction, $\bar{f}_*$ in equation (9), gives us the our best estimate for y_*, and is also known as the matrix of regression coefficients. The variance, $cov(f_*)$, is the Schur complement, and is an indication of the uncertainty of our estimation. An important conclusion from these results is that the mean prediction $\bar{f}_*$ is a linear combination of the observations y. Another aspect to underline is that the variance, $cov(f_*)$ does not depend on the observations, but only in the inputs.

For reasons that will be more clear ahead, we should also at this point introduce the marginal likelihood, $p(y|X)$. By marginalization we mean that we are integrating over the function values f. The marginal likelihood is then integral of the likelihood times the prior:

$$p(y|X) = \int p(y|f, X)p(f|X)\, df \tag{11}$$

It can be seen in [16] that under the Gaussian process model, the prior is Gaussian, $f|X \backsim \mathcal{N}(0, K)$, and the likelihood is also a Gaussian, $y|f \backsim \mathcal{N}(f, \sigma_n^2 I)$. Using the logarithmic identify to simplify the calculations, the result of the integration over f, the log marginal likelihood is:

$$logp(y|X) = -\frac{1}{2} y^T C_N^{-1} y - \frac{1}{2} log|C_N| - \frac{n}{2} log 2\pi \tag{12}$$

This exact inference is possible because both the prior and the likelihood are Gaussian, otherwise the integral in (11) would likely be intractable. The three different terms in (12) play different roles in the likelihood. The first one is the only one involving the past observations y and, therefore, is the data-fit term. The second term, on the other hand, depends only on the covariance matrix, and works in an analogous way to the regularization terms in linear regression, adding a penalty as the complexity increases. The last term is only a normalizing constant, and doesn't play a very specific role in the marginalization of the likelihood. A careful analysis of the effects of the hyperparameters in the log marginal likelihood can be found in [16].

A small note about the computational aspects of computing the log likelihood, as given by (12). In fact, there are some complexity issues related with the inversion of C_N, which depending on the size of the data points, might be quite heavy. Moreover, if C_N is an ill-conditioned matrix, its inversion is not trivial. There is already some work devoted to solve this non-trivialities, and more details about this issues can be found in [16], [6] and [17].

4. Learning the Hyperparameters

Given a covariance function, it is straightforward to make predictions for new test points, as is only a matter of algebraic matrix manipulation. However, in practical applications it is unlikely to know which covariance function to use. Clearly, the reliability of our regression is then dependent on how well we select the parameters that the selected covariance function requires [17].

Let θ be the set of hyperparameters needed for a given covariance function. In particular lets consider the case in (13), where the squared (3) and (6) were combined to form a squared exponential function for the prediction of noisy observations; then $\theta = \{l, \sigma_f, \sigma_m\}$. The challenge now, and assuming that the covariance function is adequate for the data, is to choose a value for each of the hyperparameters, the free parameters ruling a covariance function.

$$k(x_p, x_q) = \sigma_f^2 exp(-\frac{1}{2l^2}(x_p - x_q)^2) + \sigma_n^2 \delta_{pq} \tag{13}$$

For the covariance function in (13), l is the length-scale, σ_f the signal noise and σ_n the noise variance. The length-scale characterizes the distance in input space before the function value can change significantly. Short length-scales mean that the predictive variance can grow rapidly away from the data points, and all the predictions are little correlated between each other. In the same way, we can think about σ_f as the vertical lengthscale. The noise that affects the process is supposed to be random, and so no correlation between different inputs is expected, and so the term σ_n is only present on the diagonals of the covariance matrix.

The trial-and-error approach for choosing the appropriate values for each parameters is obviously not adequate.

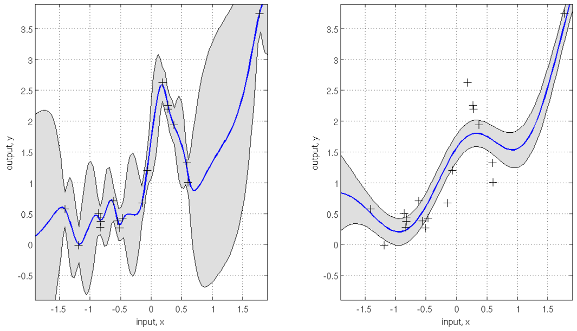

Figure 1. Example of the effect of optimizing the hyperparameters. On both plots the same Gaussian Process regression was done, but on the left ones the hyperparameters were not optimized, and on the right one they were.

Besides the obvious problems of this random approach, the covariance function can be as complex functions as needed and, therefore, the number of hyperparameters can be larger. What is needed is to find the set of parameters that optimize the marginal likelihood.

Our maximum a posteriori estimate of θ occurs when the marginal likelihood, now with the notation $p(y|X, \theta)$ to underline we interested in the hyperparameters. The problem of learning with Gaussian processes is exactly the problem of learning them. Care should be taken, as the minimization of $p(y|X, \theta)$ is a non-convex optimization task, so no guarantee of convergence is provided. To do such minimization is usually achieved through some standard gradient based technique, as long as the partial derivatives of the covariance matrix with respect to each one of the parameters are possible to get.

$$\frac{\partial}{\partial\theta_k}p(y|X,\theta) = \frac{1}{2}tr(C_N^{-1}\frac{\partial C_N}{\partial\theta_k}) + \frac{1}{2}y^T C_N^{-1}\frac{\partial C_N}{\theta_k}C_N^{-1}y \tag{14}$$

Equation (14) shows the analytical formulation to compute the different partial derivatives of the log marginal likelihood. Because of its simplicity, the Gradient Descent is a common technique to find the set of near-optimal hyperparameters that maximize the log likelihood. By iteratively combining this with the standard gradient descent method, synthesized by equation (15), until convergence given that the learning rate is appropriate.

$$\theta_k = \theta_k + w\frac{\partial}{\partial\theta_k}p(y|X,\theta) \tag{15}$$

Alternatives to the maximum likelihood estimation of the parameters, that was just described, is to use a cross-validation (CV), or generalized cross-validation algorithm. However, some previous works show that this approach is rather difficult whenever there is a large number of parameters to estimate [17]. More information about CV techniques to estimate values for the hyperparameters can be found in detail in [16].

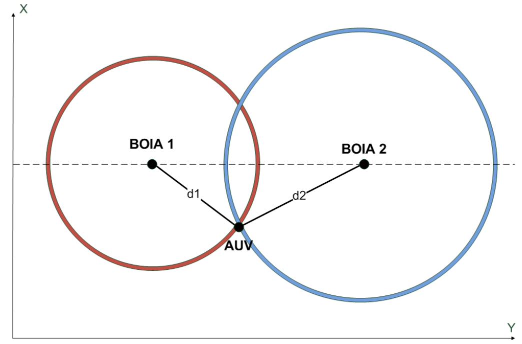

Figure 3. LBL acoustic positioning system schematic diagram

5. Application

The Ocean Systems Group (OSG), a study group within the Robotics unit of INESC TEC, has its main research efforts directed toward the development of small-sized autonomous robotic vehicles, both in underwater as on the surface. Currently, the main challenge for this kind of vehicles, and one of the main research-areas, is related with the improvement of the navigation algorithms that allow the vehicles to localize themselves within the environment.

While for the vehicles navigating on the surface can rely on the highly accurate GPS systems available, such is not possible for vehicles that move underwater, as GPS signals are not available in those environments. Therefore the majority of the Autonomous Underwater Vehicles (AUVs) rely on acoustic navitation algorithms and, in particular, on long baseline (LBL) acoustic positioning systems.

For this systems, prior to any mission, the vehicle is informed about the actual global coordinates of the the beacons that constitute the acoustic network used. Then, in order to know its exact localization at any given time, it has to interrogate each beacon, sending an acoustic signal with a specific frequency and waiting for the beacon reply. By timing this acoustic events, it is then possible to compute the actual distance of a given vehicle to each of the two beacons and, therefore, its real-time global coordinates.

The algorithm used to estimate distances d_1 and d_2 , as can be seen on figure 3, assumes that the AUV positions remains stationary between the interrogation of the beacon and the reception of the correspondent answer. It is also considered that the depths the AUV reaches while in mission are constant and quite small relative to the distances to both beacons and, thereby, we can assume only motion in the horizontal plane. The measures available are highly

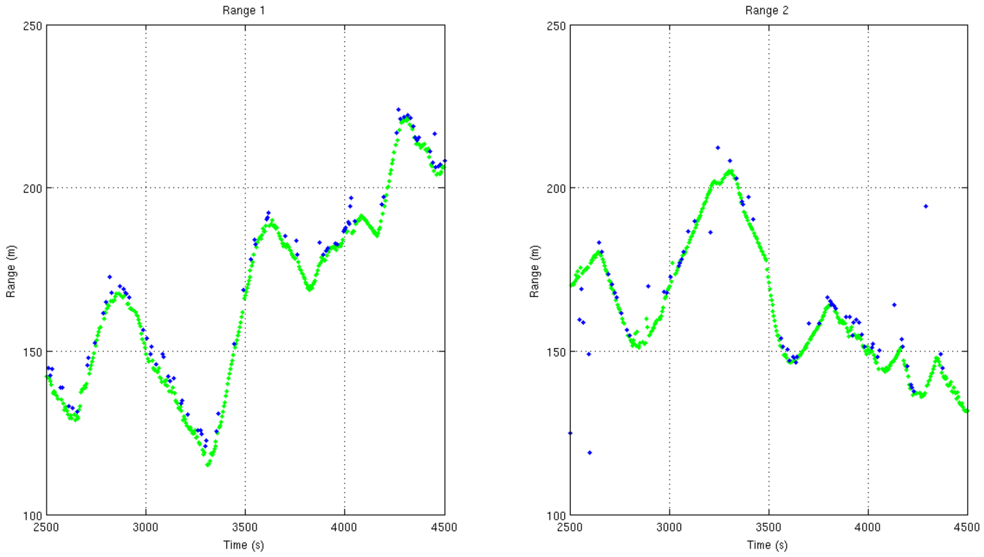

Figure 2. Example of range measurements acquired to a pair of buoys during a mission. In green we can see the measurements classified by an expert as good ones, while the ones in blue are classified as spurious and caused by reflections.

irregular and noisy, and require filtering. The technique currently in use by the OSG vehicles is based on a Kalman Filter (KF). The KF plays has a very important role in the estimation process, as not only it allows the elimination of spurious data measurements, but also to fuse navigation data coming from different sensors. An example of range measurements acquired during a standard mission can be seen in figure 2.

The filtering of the measures is done by evaluating the covariance of the error associated with the measurements as and comparing it to the design parameter γ. Even though this method achieves some reasonable results, is not very flexible as it relies solely on a single parameters, and doesn't allow to adapt to the different varying environmental conditions, such as temperature and salinity of the water, and that have a great affect in the measures. As a consequence, γ as to be wide enough, but this causes spurious messages to be accepted as if they were not spurious. In opposition to direct range measurements, these spurious measures are mostly caused by reflections of the acoustic signals both in the bottom of the sea, or in the surface.

The main motivation for this work came from a paper from Bingham and Seering [2], where both direct measurements and reflections were modelled, but in an off-line post processing environment. By using an Expectation-Maximization algorithm and a proper modelling of both the range, and the reflection, some very interesting results.

For this work, the objective was to filter the range measurements using some techniques used in background subtraction, a widely used approach for detecting moving objects from static cameras. After a careful review of some of the techniques used, it was decided to use a Running Gaussian average to online validate the range measurements. This model is based on ideally fitting a Gaussian probability density function on the last n pixels. In order to avoid fitting from scratch at each new frame time, a running average is computed instead [12].

$$\mu_t = (1-\rho)\mu_{t-1} + \rho X_t \tag{16}$$

$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \tag{17}$$

Following the paper by Stauffer and Grimson, who proposed some changes to the traditional running Gaussian approaches, the running averages mean and covariance are updated according to (16) and (17). X_t is the new observation to be validated and $\rho = \mathcal{N}(X_t|\mu_k, \sigma_k)$. A matched is defined whenever an observation is within 2.5 standard deviation of a distribution.

The range measurements are expected to vary throughout time, as the distance from the AUV to both beacons also varies according to the motion of the vehicle. In that sense, the mean of running Gaussian should also vary in the same way. To tackle this problem, what we propose is to predict the next range measurement, based on the past measurements taken as correct. To such prediction one can use whether a linear regression, or a Gaussian Process regression. Given the scope of this paper, which pretends to be a tutorial on the use of Gaussian Processes, in the next section results will be presented comparing both this approaches

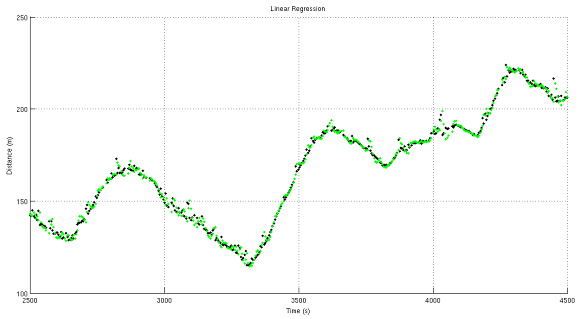

Figure 4. Linear Regression: results

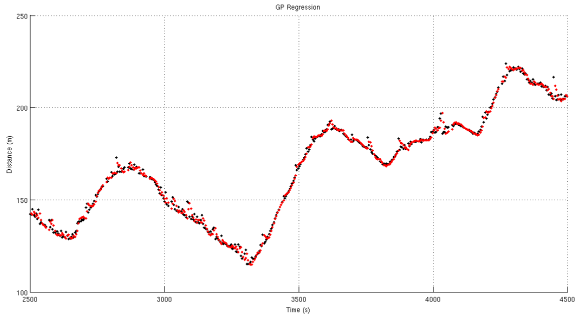

Figure 5. GP Regression: results

6. Results

In this section we will present the results that compare a standard linear regression, with a GP regression. All these results are comparable, in the sense that all of them relate to the same data, acquired during a mission performed in Augusto 2011, in the Douro River. The linear regression algorithms were implemented by the author, but for the regression with the Gaussian Processes, the framework developed for Matlab by Carl Edward Rasmussen and Christopher K. I. Williams, and freely available on the Internet was used.

On figure 4 we can see the result of the linear regression, for the basis are $\{1, x, x^2\}$, a regression using a quadratic function. Even though at a coarse level the regression is able to correctly follow actual ranges, we can see that specially at the inflexion points there is a lot of noise predictions, that don't match the actual behaviour of the vehicle.

On figure 5 we have the correspondent Gaussian Process based regression. Here in this case, it is also possible to understand that the regression predictions follow closely the actual range measures. A careful look will in fact realize that in this case, there is less noisy predictions in the inflexion points, with the regression fitting more closely to the data. This regression was made using a squared exponential function as in eq. (13) with the following hyperparameters: $l = 50$, $\sigma_f = 10$ and $\sigma_n = 1$;

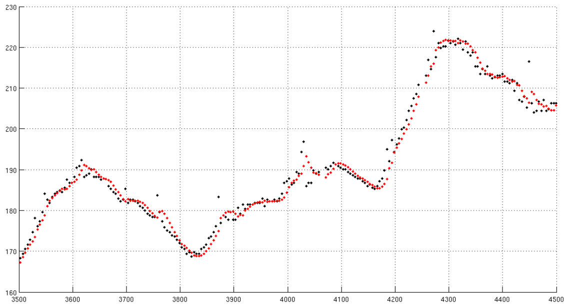

Figure 6. GP Regression: detail with improved parameters. It is possible to note some outliers being rejected.

As for the linear regression, there is not much to be improved, in the the regression with the Gaussian Process, we can still try to vary the hyperparameters, as described in the previous sections. In fact, the framework under used provides the methods necessary to obtain the derivatives of the log marginal likelihood, that would be of great help.

However, and despite some effort, the parameters couldn't be optimized, due to the lack of convergence. Recalling from the previous chapters, the minimization of the log marginal likelihood is not a convex optimization, and therefore the gradient descent methods that were used, with a large set of learning rates, are not guaranteed to succeed, as indeed happened. On the other side, and due to the intrinsic and very dynamic nature of the problem, where the vehicle in question is always moving with different attitudes towards each of the buoys, is is probably very difficult that the optimal parameters are the same throughout the whole mission. Instead, it is more likely that this parameters keep varying, and so, the gradient descent algorithm doesn't converge.

Even though, a simple trial and error approach lead to a small improve in the prediction, with its output more immune to spurious measures. With the hyperparameters set to $l = 50$, $\sigma_f = 10$ and $\sigma_n = 5$, where one can note the change on the signal noise from 1 to 5, we can achieve better results. With the detailed view, in figure 6 we can confirm that some outliers are being ignored.

7. Conclusion and Future Work

In this paper the regression with Gaussian Processes is covered. It was demonstrated that it can be a quite effective method if there is some prior knowledge about the covariance of the measures. In fact, this is of utmost importance, as choosing a wrong covariance function can lead to poor performance.

A note as well for the possibility that Gaussian process have for classification problems, where the approach is pretty similar as for prediction problems. The main differences are related with the fact that in classification problems, and due to the fact that an activation function is used, the integration of the prior times the posterior that leads to the likelihood is in fact intractable. So approximations algorithms must be employed, like the Laplace approximation or the Expectation Propagation.

To conclude, even though Gaussian Processes are, in a sense, close to some ARMA models or even the Kalman Filter, they provide a very efficient way to adapt to the data in a non-parametric way. Given that we choose the adequate mean function and covariance function, the problem of learning with Gaussian processes is exactly the problem of learning the hyperparameters of the covariance function.

References

[1] R. M. S. Almeida. Sistema inteligente de posicionamento acústico subaquático. Master's thesis, Faculdade de Engenharia da Universidade do Porto, 2010.

[2] B. Bingham and W. Seering. Hypothesis grids: improving long baseline navigation for autonomous underwater vehicles. *Oceanic Engineering, IEEE Journal of*, 31(1):209 – 218, jan. 2006. 5

[3] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 2

[4] J. S. Cardoso. *Machine Learning, Lecture Notes*. Faculty of Engineering, University of Porto, January 2012. 1

[5] M. Ebden. Gaussian processes for regression: A quick introduction. Available in http://www.robots.ox.ac.uk/ mebden/reports/GPtutorial.pdf, January 2011, August 2008.

[6] M. Gibbs and D. J. MacKay. Efficient implementation of gaussian processes. Technical report, 1997. 3

[7] B. Huhle, T. Schairer, A. Schilling, and W. Strasser. Learning to localize with gaussian process regression on omnidirectional image data. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 5208 – 5213, oct. 2010.

[8] B. Huhle, T. Schairer, A. Schilling, and W. Strasser. Learning to localize with gaussian process regression on omnidirectional image data. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 5208 – 5213, oct. 2010.

[9] J. Ko, D. Klein, D. Fox, and D. Haehnel. Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 742 –747, april 2007.

[10] J. Ko, D. Klein, D. Fox, and D. Haehnel. Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 742 –747, april 2007.

[11] J. Melo and A. Matos. Guidance and control of an asv in auv tracking operations. In *OCEANS 2008*, pages 1 –7, sept. 2008.

[12] M. Piccardi. Background subtraction techniques: a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3099 – 3104 vol.4, oct. 2004. 5

[13] C. E. Rasmussen. Gaussian processes in machine learning. Available in http://www.cs.ubc.ca/ hutter/earg/papers05/rasmussen_gps_in_ml.pdf, January 2011, January 2011. 2

[14] S. Srihari. Gaussian processes - lecture slides for machine learning and probabilistic graphical models. Technical report, Department of Computer Science and Engineering, University at Buffalo, 2011.

[15] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages 2 vol. (xxiii+637+663), 1999.

[16] C. E. R. . C. K. I. Williams. *Gaussian Processes for Machine Learning*. the MIT Press, 2006. 2, 3, 4

[17] C. K. I. Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. In *Learning and Inference in Graphical Models*, pages 599–621. Kluwer, 1998. 3, 4