Circular Regression Based on Gaussian Processes

Pablo Guerrero
Departamento de Ciencias de la Computación
Advanced Mining Technology Center
Universidad de Chile
Blanco Encalada 2120. Santiago. Chile.
Email: pguerrer@ing.uchile.cl

Javier Ruiz del Solar
Departamento de Ingeniería Eléctrica
Advanced Mining Technology Center
Universidad de Chile
Email: jruizd@ing.uchile.cl

Abstract—Circular data is very relevant in many fields such as Geostatistics, Mobile Robotics and Pose Estimation. However, some existing angular regression methods do not cope with arbitrary nonlinear functions properly. Moreover, some other regression methods that do cope with nonlinear functions, like Gaussian Processes, are not designed to work well with angular responses. This paper presents two novel methods for circular regression based on Gaussian Processes. The proposed methods were tested on both synthetic data from basic functions, and real data obtained from a computer vision application. In these experiments, both proposed methods showed superior performance to that of Gaussian Processes.

I. INTRODUCTION

Circular data [1], [2], [3], [4], [5], [6] corresponds to 2D angular data. Circular data is usually associated with directions or time. This kind of data is present in many interesting problems in various areas of knowledge. Just to mention a few of these areas, circular data is of interest in biology [7], geology [8], [9], and meteorology [10], [11]. Circular data is also relevant for computer vision [12] and image processing [13]. For instance, in [12], the orientation of a car in a picture is estimated from features extracted from the picture. The availability of reliable methods for circular regression is of key importance for performing adequate estimations in those domains.

On the one hand, most of the existing regression methods for dealing with circular data do not cope well with the nonlinearities of the problem. On the other hand, there are nonlinear and nonparametric regression methods, such as Gaussian Processes (GPs), which are not well designed to cope with circular data. In fact, as will be shown in the results section of this paper, GPs perform poorly when solving a regression problem with an angular response. The reason for this poor performance appears to be the fact that GPs always bring a linear combination of the training targets as the predicted mean. However, the linear combination of angles does not cope well with the discontinuities that are inherent to an angular space.

The two methods proposed in this document follow the same methodology. This methodology is intended to overcome the problem of the discontinuities by employing a better strategy for combining angles, which consists of following three steps: (i) getting an estimation of the sine and cosine of each angle, (ii) combining linearly the estimated sine and cosine values independently, and, (iii) using the arctangent function to get the final estimation of the angle from the estimated sine and cosine values. This approach is based on the projected normal distribution (See [4] for example).

The structure of the document is as follows: Related work on circular regression and on nonlinear regression is reviewed in Section II. Then, the nonlinear regression based on Gaussian Processes is outlined in Section III. The proposed methods for nonlinear angular regression are described in Section IV. In Section V, real and simulated experiments and their results are presented. Finally, some conclusions are drawn in Section VI.

II. RELATED WORK

The regression problem can be defined as the problem of learning a function \( f \) from finite paired realizations, \((x_i, y_i), i \in \{1, \ldots, N\}\), of a covariate \( x \) and a response \( y \). We are interested in the case in which \( y \) is circular. Given a test point \( x^* = (x^*_1, \ldots, x^*_m) \), a regression method estimates a prediction on the probability distribution of the related response \( y^* \). The predicted mean \( \mu^* = \mathbb{E}(y^*) \) is of particular interest. The circular regression problem has been studied in the last few decades using various approaches. For instance, an early solution for the circular regression problem proposed by [14], calculated \( \mu^* \) as a linear combination of the covariate components:

\[
\mu^* = \mu_0 + \sum_i \beta_i x_i^* \tag{1}
\]

These kinds of regression models are known as helical, [15] noted that these models had infinite maxima in the likelihood function. To overcome this problem for a single covariate, they proposed a specific model for the joint distribution of \( y \) and a linear variable, \( x \), with a completely specified marginal distribution function \( F(x) \). Then, they calculated the mean as:

\[
\mu^* = \mu_0 + 2\pi F(x^*) \tag{2}
\]

Following this idea, [3] introduced the use of a nonlinear link function, \( g \), that maps \( \mathbb{R} \) to \((−\pi, \pi)\) monotonically, and is applied to a linear combination of \( x^* \):

\[
\mu^* = \mu_0 + g\left(\sum_i \beta_i x_i^*\right) \tag{3}
\]
[16] pointed out that these methods are practically nonuseable in the general case due to several flaws including implausibility of fitted models, non-identifiability of parameters, and difficulties in the computation of the parameter estimates.

Finally, some Bayesian methods for circular regression have been proposed. For example, [4], proposed a model based on a projected normal distribution, where \((\cos y, \sin y)^T\) is the projection to the unit circle of the bivariate Gaussian \(z\):

\[
u = z/R, R = \|z\|\] (4)

Then, \(\mu_{*}^z = E(z^*)\) is calculated as:

\[
\mu_{*}^z = \sum_i \beta_i x_i^* \tag{5}
\]

To infer the parameters \(\beta_i\), a Bayesian procedure based on Gibbs sampling is proposed. As in [4], the method proposed in this paper uses the projected normal distribution to model the probability distribution of the response variable. However, differing from [4], the method proposed here uses Gaussian Processes (GPs) for estimating the mean and variance of the bivariate Gaussian \(z\).

There are also methods based on Gaussian Processes that allow the estimation of linear responses having circular covariates. For that purpose, periodic covariance functions have been used (see for instance [17]). Differing from those approaches, in this paper, circular responses are estimated using linear covariates.

III. GAUSSIAN PROCESSES FOR REGRESSION

Gaussian Processes (GPs) provide a non-parametric tool for non-linear regression and classification. An excellent summary that includes both theoretical and practical aspects and references to deeper theoretical insights can be found in [17]. Although we are only interested in the regression capabilities of GPs, they can also solve classification problems.

We are interested in the case when the observations in these samples have an associated noise, i.e., \(y_i = f(x_i) + \varepsilon_i\). GPs are able to solve this kind of regression problem at least when the noise \(\varepsilon_i\), sometimes called the observational noise, is assumed to be Gaussian with zero mean and arbitrary variance \(\sigma^2_{\varepsilon}\). Furthermore, for any arbitrary input test, \(x^*\), GPs are able to give a predictive mean and variance of the response \(y^*\).

A. Covariance Functions

Covariance functions are a key component of GPs for regression and classification. Covariance functions encode the information of the kind of functions that a GP can learn. They also restrict the possible measures of proximity that are necessary for the regression mechanism to operate. Usually covariance functions have parameters, called hyperparameters and denoted \(\theta\). There are several methods for estimating the hyperparameters from the training data. In general terms, the covariance function is defined as the covariance of the values of \(f\) at two points \(x\) and \(x^*\):

\[
k(x, x^*) = \text{Cov}(f(x), f(x^*)) \tag{6}
\]

There are several kinds of covariance functions examined in the literature, some of which are described in [17]. The selection of an adequate covariance function is crucial for the correct resolution of a determined problem. The most commonly used covariance function is the squared exponential covariance function:

\[
k_\theta(x, x') = \sigma^2_f \exp \left(-\frac{1}{2} (x - x')^T W (x - x') \right) \tag{7}
\]

where \(W\) is a diagonal matrix with scaling factors in its diagonal. Note that the subscript in \(k_\theta\) indicates the dependence of the covariance function on the hyperparameters \(\theta\).

B. Prediction

Given a covariance function and a vector \(\theta\) of values for its hyperparameters, a GP can predict the mean and variance of the function value for any arbitrary input point set. For convenience, given a set of input vectors, \(\{x_1, \ldots, x_m\}\), we will define its aggregated matrix column-wise as:

\[
\text{Agg} (\{x_1, \ldots, x_m\}) = [x_1 | \ldots | x_m] \tag{8}
\]

Then, let \(X\) denote the aggregated matrix of the training input set, \(\{x_i\}\), and \(y\) denote the transpose of the aggregated matrix of the training target set, \(\{y_i\}\). Note that since the response is one-dimensional, \(y\) is actually a vector. If we have a set of test inputs, \(\{x_i^*\}\), then let \(X^*\) denote the aggregated matrix of \(\{x_i^*\}\). Additionally, let us define \(f_i^* = f(x_i^*) \sim N(f_i^*, \text{Var}(f_i^*))\), which, of course, is a random variable.

Finally, let \(\mathbf{f}^*\) denote the aggregated matrix of \(\{f_i^*\}\), \(\mathbf{1}^*\) its mean, and \(\text{Cov}(\mathbf{f}^*)\) its covariance matrix.

Given two aggregated matrices \(X^*\) and \(X''^*\) of the input sets \(\{x_i^*\}\) and \(\{x_i''^*\}\), respectively, we can define the covariance matrix between \(X^*\) and \(X''^*\), \(K_\theta(X^*, X''^*)\), as the matrix whose components are defined by:

\[
K_\theta(X^*, X''^*)_{ij} = k_\theta(x_i^*, x_j''^*) \tag{9}
\]

Let us define \(K_X = K_\theta(X, X)\) and call it simply the covariance matrix. If there is only one test point \(x^*\), we can write \(K_\ast = K_\theta(X, x^*)\) and then the predictive mean and variance of the function value \(f^*\), for the input \(x^*\), are [17]:

\[
\mathbf{f}^* = K_n^\top K_n^{-1} y \tag{10}
\]

\[
\text{Var}(f^*) = k_\theta(x^*, x^*) - K_n^\top K_n^{-1} k_\ast \tag{11}
\]

Then, given that \(\varepsilon_i\) is assumed to have zero mean and variance \(\sigma^2_{\varepsilon}\), the response, \(y^*\), at input point \(x^*\) has mean \(\mathbf{f}^*\) and variance \(\text{Var}(f^*) + \sigma^2_{\varepsilon}\). Stated in equations,

\[
E(y^*) = K_n^\top K_n^{-1} y \tag{12}
\]

\[
\text{Var}(y^*) = k_\theta(x^*, x^*) - K_n^\top K_n^{-1} k_\ast + \sigma^2_{\varepsilon} \tag{13}
\]
C. Learning

As has been stated before, there are automatic methodologies to learn the hyperparameters from the training data. One of these methodologies pursues the maximization of the log marginal likelihood, which corresponds to (see [17] for derivation):

\[ \log p(y|X) = -\frac{1}{2} y^T K_n^{-1} y - \frac{1}{2} \log |K_n| - \frac{N}{2} \log 2\pi \] (14)

Then, the learning process consists of finding the hyperparameter vector \( \hat{\theta} \) that maximizes \( \log p(y|X) \):

\[ \hat{\theta} = \arg\max_{\theta} \log p(y|X) \] (15)

Note that the hyperparameter vector contains a set of parameters for the covariance function that influence \( \log p(y|X) \) through \( K_X \), and the noise variance, \( \sigma_n^2 \), that influences \( \log p(y|X) \) directly. A convex optimization algorithm may be employed for finding the maximum. A useful expression for the maximization algorithm is the marginal likelihood gradient:

\[ \frac{\partial \log p(y|X, \theta)}{\partial \theta_j} = \frac{1}{2} \text{tr} \left( (\alpha^\top - K_n^{-1}) \frac{\partial K_n}{\partial \theta_j} \right) \] (16)

where \( \alpha = K_n^{-1} y \) and \( \theta_j \) is the \( j \)-th element of \( \theta \), i.e., the \( j \)-th hyperparameter.

IV. REGRESSION WITH AN ANGULAR RESPONSE

In this section, two GP-based novel methods for circular regression are proposed.

A. SinCos-GP

The first method to be introduced solves the previously mentioned angle regression problem by separating it into two independent regression problems: one for the sine and the other for the cosine of the angle. For that reason, we call this method SinCos-GP. Two independent GPs are trained using the standard GP training procedure: one, \( GP^{\cos} \), for the sine of the angle, and the other, \( GP^{\sin} \), for the cosine. Then, two independent sets of hyperparameters must be learned:

\[ \theta_{\cos} = \arg\max_{\theta} \log p(y_{\cos}|X) \] (17)
\[ \theta_{\sin} = \arg\max_{\theta} \log p(y_{\sin}|X) \] (18)

where \( y_{\cos} \) and \( y_{\sin} \) are vectors whose elements are the sine and the cosine respectively of the elements of \( y \). In element-wise equations, \( [y_{\cos}]_i = \cos y_i \) and \( [y_{\sin}]_i = \sin y_i \).

Then, for each test input, SinCos-GP will predict the sine and the cosine of the angular response, \( y^* \), independently:

\[ \overline{y^*}_{\cos} = \mathbb{E}(\cos y^*) = k_{\cos}^\top [K_n^{\cos}]^{-1} y_{\cos} \] (19)
\[ \overline{y^*}_{\sin} = \mathbb{E}(\sin y^*) = k_{\sin}^\top [K_n^{\sin}]^{-1} y_{\sin} \] (20)

where

\[ K_n^{\cos} = K_X^{\cos} + (\sigma_n^{\cos})^2 I \] (21)
\[ K_X^{\cos} = K_{\theta_{\cos}}(X, X) \] (22)
\[ k_{\cos} = K_{\theta_{\cos}}(X, x^*) \] (23)

and

\[ K_n^{\sin} = K_X^{\sin} + (\sigma_n^{\sin})^2 I \] (24)
\[ K_X^{\sin} = K_{\theta_{\sin}}(X, X) \] (25)
\[ k_{\sin} = K_{\theta_{\sin}}(X, x^*) \] (26)

Finally, SinCos-GP approximates the mean of the angular response as the result of the arctangent function applied to the predicted sine and cosine:

\[ E(y^*) = \arctan(\overline{y^*}_{\sin}, \overline{y^*}_{\cos}) \] (27)

B. Angle-GP

Angle-GP is also a solution for the linear-circular regression problem that is based on SinCos-GP. In fact, the whole predictive part of Angle-GP is identical to that of SinCos-GP (See equations 19 to 27).

However, differing from SinCos-GP, Angle-GP assumes that the covariance of the sine and the cosine of the response are identical. For this purpose, the same hyperparameters are learned for both \( GP^{\cos} \) and \( GP^{\sin} \): \( \theta_{\cos} = \theta_{\sin} = \theta \).

For considering the influence of both the sine and the cosine in the learning process, the log marginal likelihood, \( \log \mathcal{L} = \log p(y_{\cos}, y_{\sin}|X) \), of both \( y_{\cos} \) and \( y_{\sin} \) is maximized. Assuming that \( y_{\cos} \) and \( y_{\sin} \) are independent:

\[ \log \mathcal{L} = \log p(y_{\cos}|X) + \log p(y_{\sin}|X) \]
\[ = -\frac{1}{2} y_{\cos}^\top K_n^{-1} y_{\cos} - \frac{1}{2} y_{\sin}^\top K_n^{-1} y_{\sin} \]
\[ - \log |K_n| - N \log 2\pi \] (28)

The hyperparameters are selected in order to maximize \( \log \mathcal{L} \):

\[ \hat{\theta} = \arg\max_{\theta} \log \mathcal{L} \] (29)

The gradient of \( \log \mathcal{L} \) may be useful for performing the former maximization. It can be calculated as:

\[ \frac{\partial \log \mathcal{L}}{\partial \theta_j} = \frac{1}{2} \text{tr} \left( (\alpha_{\cos} \alpha_{\cos}^\top + \alpha_{\sin} \alpha_{\sin}^\top - 2K_n^{-1}) \frac{\partial K_n}{\partial \theta_j} \right) \] (30)

where \( \alpha_{\cos} = K_n^{-1} y_{\cos} \) and \( \alpha_{\sin} = K_n^{-1} y_{\sin} \). In fact,
\[ \frac{\partial \log L}{\partial \theta_j} = -\frac{1}{2} y^\top \cos \frac{\partial K_n^{-1}}{\partial \theta_j} y \cos - \frac{1}{2} y^\top \sin \frac{\partial K_n^{-1}}{\partial \theta_j} y \sin = \frac{\partial \log |K_n|}{\partial \theta_j} \]
\[ = \frac{1}{2} \frac{y^\top \cos}{\theta_j} K_n^{-1} \frac{\partial K_n}{\theta_j} y \cos \]
\[ + \frac{1}{2} \frac{y^\top \sin}{\theta_j} K_n^{-1} \frac{\partial K_n}{\theta_j} y \sin - \text{tr} \left( K_n^{-1} \frac{\partial K_n}{\theta_j} \right) \]
\[ = \frac{1}{2} \text{tr} \left( \alpha_\cos \cos^\top + \alpha_\sin \sin^\top - 2 K_n^{-1} \frac{\partial K_n}{\theta_j} \right) \]

As in SinCos-GP, when a test input \( x^* \) is presented, Angle-GP uses \( GP^\cos \) and \( GP^\sin \) to predict the means of the cosine and sine of the angle independently, and then applies the arctangent function to them in order to get the predicted mean of the angular response.

V. RESULTS

The SinCos-GP and Angle-GP methods are tested using both synthetic data generated in MATLAB and real data obtained from a database of car images. The performances of Angle-GP and SinCos-GP are compared to that of a GP. As a measure of performance the root mean squared error (RMSE) is employed.

A. Simulated Experiment

In this experiment, each method was used to perform the regression of the following function:
\[ y = \arctan(x_2, x_1) + 45^\circ \quad (31) \]
for a two-dimensional covariate \( x = (x_1, x_2)^\top \). Regardless of the definition of the output interval of the arctangent function, the function was discontinuous at one point. The definition of the output interval selected for the purposes of this experiment was \((-180^\circ, 180^\circ]\). The size, \( N \), of the training set varied from 5 to 100 samples, increasing in increments of 5. For each training set size, 100 trials were performed, with different randomly sampled training sets. In each training sample, \( x_1 \) and \( x_2 \) were independently sampled in the interval \([-1, 1]\). In order to check the regression accuracy of the methods, a fixed set of test points were obtained from a fixed \( 20 \times 20 \) uniform grid in \([-1, 1] \times [-1, 1]\).

For each trial and for each method, the RMSE\(^1\) of the prediction was calculated in order to have a global comparison measure. Figure 1 shows the results in terms of comparative performance for the described regression task using: a GP, a SinCos-GP, and an Angle-GP.

From these results, it is clear that Angle-GP surpasses the performance of both the GP and the SinCos-GP. As the number of training samples grows, the three methods being compared improve their predictions and the accuracy of SinCos-GP comes closer to that of Angle-GP. However, the GP is not able to get a low RMSE even with a high number of training samples.

\(^1\)Note that the angular error cannot be obtained by a simple subtraction. Sometimes \( \pm 360^\circ \) must be added to the result of the subtraction in order to get an error in the \((-180^\circ, 180^\circ]\) interval.

In order to explore the behavior of the described methods in more detail, their prediction performances over different subintervals of the angular response space were measured. For this purpose, the average prediction error was calculated for different subsets of the test sets.

Each subset corresponds to a subinterval of the angular response space. Figure 2 shows the prediction RMSE through the whole experiment for each angle interval, with 5 and 100 training samples.

From Figure 2, it is possible to infer that the cause of the poor performance of the GP on the angle regression task is mainly due to the discontinuity in \(180^\circ\). No apparent correlation between the target angle and the performances of SinCos-GP and Angle-GP is shown. These figures also confirm the consistent superiority of Angle-GP over SinCos-GP, which can be explained by the selection of a more adequate log marginal likelihood in the sense that it reflects the symmetry of this particular problem.

B. Car Pose Estimation

The following experiment consists in estimating the orientations of cars in an image dataset [12]. The dataset consists of
Fig. 3. Examples of Cropped Pictures from the Car Database [12].

pictures taken at a car show. Each image contains a single car in its foreground. The bounding box and ground-truth orientation of each image is available.

There are 20 different car models rotating by 360 degrees with an image taken every 3 or 4 degrees. Figure 3 shows examples of images in the dataset.

The original dataset is split into two subsets, one for training (10 sequences) and one for testing (10 sequences). Since we are not concerned with the detection problem, the bounding box of the car in each image is provided to the system in advance and thus the image window containing the car is cropped so that the image just fits the car. In order to make the experiment more independent on the selection of the training set, 10 trials of the experiment were performed and in each of them, a different random selection of the training and testing subsets was made.

In order to extract a feature vector from each image, we follow a very similar procedure to the one used by the authors of the database [12]. The procedure employed is described as follows: For every image window, the DAISY descriptors [18] of each pixel are extracted. A subset of 117 900 DAISY descriptors (100 per each training image) is randomly sampled. Then, 100 clusters are extracted from the sampled subset using \( k \)-means. The resulting clusters are then used to classify every pixel, through its DAISY descriptor, in every image. Figure 4 shows an example of an image from the dataset and the resulting descriptor-class image. Then, a 4-level spatial pyramid of histograms [19] is created. A spatial pyramid considers frequencies of the different descriptor clusters not only in the whole image but also in different regions of it, yielding richer information about the positions in which these descriptors are found. Figure 5 illustrates a spatial pyramid with the same levels as the one mentioned above but with only 2 clusters of descriptors.

The resulting feature vector has 3 000 dimensions because for each of the 30 cells in the pyramid, there are 100 histograms (one per DAISY descriptor cluster). For the training stage, the training images were subsampled so that one of every ten was considered. Then, differing from [12], we used Principal Component Analysis (PCA) to reduce the dimensionality of the feature vector. Different numbers of principal components were chosen, from 5 to 100, in steps of 5.

The performance in terms of RMSE of the prediction was compared for each number of principal components. Figure 6 shows the RMSE of the rotation estimation as a function of the number of principal components used. Independent of the number of principal components, SinCos-GP and Angle-GP show a clearly lower RMSE than a GP. Additionally, the three tested methods were able to reduce
RMSE for each Number of Principal Components

Fig. 6. Mean Squared Error in the Rotation Angle Estimation for the Test Set.

their RMSE while the number of components increased up to its maximum number (120). With a few exceptions, Angle-GP has a slightly lower RMSE than SinCos-GP. However, all the tested methods have a high RMSE (over 60 degrees). This is probably due to the lack of representative training data (only 120 images per trial) in the experiment.

VI. CONCLUSION

Two novel methods for nonlinear and nonparametric regression of functions with a 2D angular response are presented, SinCos-GP and Angle-GP. Both methods are based on Gaussian Processes and replace the linear combination of targets by a linear combination of the sine and cosine of the angles. SinCos-GP and Angle-GP were compared to a GP in solving regression simulated and real circular regression problems. In both simulated and real data experiments, SinCos-GP and Angle-GP performed significantly better than a GP. The two experiments presented show the ability of the methods presented to fit problems of very different natures where the response is an angular variable.

Angle-GP seemed to work better than SinCos-GP in most of the tested situations. However, this superiority does not appear to be consistent enough to differentiate clearly the performance of both methods. Thus, it is advisable to evaluate both alternatives for each particular application. It seems a reasonable prediction that in those applications where some of the covariate components bring more useful information for the estimation of the sine, and others to the estimation of the cosine, SinCos-GP will show better results. Whereas in applications in which all the covariate components bring as much information for the estimation of the sine as for the cosine, it would be reasonable to think that Angle-GP would work better. For future work, it would be useful to develop a method for predicting the variance of the angular response. Moreover, the authors would like to investigate the extension of the proposed method to the case of 3D angular data. Additionally, it would be interesting to investigate the application of the ideas presented in this paper to the regression of angular variables in the presence of heteroscedasticity. Finally, some experiments should be carried out in the future in order to determine the sensitivity of the method to the training size in regression problems.

ACKNOWLEDGMENT

This work was partially funded by FONDECYT Postdoctoral Grant 3130729 and by FONDECYT Grant 1130153.

REFERENCES