Probabilistic Cross-Validation Estimators for Gaussian Process Regression

Luca Martino, Valero Laparra, Gustau Camps-Valls
Image Processing Laboratory, Universitat de València (Spain)

***Abstract*—Gaussian Processes (GPs) are state-of-the-art tools for regression. Inference of GP hyperparameters is typically done by maximizing the marginal log-likelihood (ML). If the data truly follows the GP model, using the ML approach is optimal and computationally efficient. Unfortunately very often this is not case and suboptimal results are obtained in terms of prediction error. Alternative procedures such as cross-validation (CV) schemes are often employed instead, but they usually incur in high computational costs. We propose a probabilistic version of CV (PCV) based on two different model pieces in order to reduce the dependence on a specific model choice. PCV presents the benefits from both approaches, and allows us to find the solution for either the maximum a posteriori (MAP) or the Minimum Mean Square Error (MMSE) estimators. Experiments in controlled situations reveal that the PCV solution outperforms ML for both estimators, and that PCV-MMSE results outperforms other traditional approaches.**

Keywords: Probabilistic Cross Validation, Marginal Likelihood, MAP estimator, MMSE estimator, Gaussian Processes.

I. Introduction

> *"If someone puts a gun on my head and asks me to do model selection, I choose cross-validation."*
> Chih-Jen Lin and Olivier Chapelle

Gaussian processes (GPs) are Bayesian state-of-the-art tools for discriminative machine learning, i.e., regression [1], [2], classification [3] and dimensionality reduction [4]. GPs were first proposed in statistics by Tony O'Hagan [5] and they are well-known to the geostatistics community as *kriging*. However they did not become widely applied tools in machine learning until the early XXI century due to their high computational complexity [6]. In the last years, GPs have become one of the standard tools to approach regression from empirical data [6], and have received attention of many applied fields.

Essentially, a GP is a stochastic process whose marginals are distributed as a multivariate Gaussian density. If the observed data are truly generated by a GP, the use of the *marginal likelihood* (ML) induced by the GP model (a.k.a., Bayesian evidence) is the best procedure for inferring the hyperparameters. However, this perfect match between the data and the assumed model rarely occurs in practical applications. For this reason, despite the mathematically elegance of using the ML approach, other alternative procedures are often employed with similar success and adoption. Examples include *random sampling* [7], the Nelder-Mead method (aka downhill simplex) [8], Bayesian optimization [9]–[11], and many flavours of derivative-free optimization approaches such as stochastic local search, simulated annealing or evolutionary computation [12], [13].

Stepping backwards, perhaps the simplest approaches to hyperparameter selection are standard *cross-validation* (CV) grid-search procedures. Instead of taking care of the statistical description of the data and the hyperparameters, CV directly looks for the hyperparameters that minimize an arbitrary cost function of interest, e.g. the squared loss. The CV procedure needs to evaluate the cost for each possible combination of hyperparameter values and choose the set that minimizes the error in an out-of-sample validation test. Even though the CV optimization is independent from the previously assumed statistical model, it involves exhaustive evaluation of multiple sets of hyperparameters leading to high computational burden. CV is thus only applicable for a small number of hyperparameters.

In this work, first we introduce a Probabilistic Cross Validation (PCV) based on two different model stages in order to reduce the dependence on one specific choice of regression model. It also allows tuning a large number of hyperparameters by means of gradient-descent techniques and/or Monte Carlo methods [9], [12], [13]. The approach avoids exhaustive grid-search as in the standard CV approach. Other heuristic strategies are available, such as random search approaches [7]. However, they are more computationally demanding than the use of Monte Carlo methods [14]–[16], which search in a portion of the space according to the posterior distribution of the hyper-parameters given the observed data.

In PCV, it is possible to define both the Maximum a Posteriori (MAP) and Minimum Mean Square Error (MMSE) estimators in a similar way to ML. If a uniform prior density over the hyperparameters is considered, the PCV-MAP approach coincides with the solution of the classical CV procedure. We compare PCV and ML approaches by numerical simulations in controlled settings of Signal-to-Noise Ratio (SNR) and dataset cardinality. We also test MAP and MMSE estimators with both approaches. It does not escape our notice that other alternative estimators from robust statistics are available, such as the median estimator, L-estimators and M-estimators [17]. We leave these comparisons for future work though, and focus on the standard MAP and MMSE only.

The remainder of the paper is structured as follows. The needed background related to the GP regression model is given in Section II. The use of the marginal likelihood for tuning the hyperparameters is given in Section III. The PCV approach is presented in Section IV. Section V is devoted to the numerical simulations. Some conclusions are provided in Section VI.

ISBN 978-0-9928626-7-1 © EURASIP 2017

II. Gaussian Process (GP) Regression

GPs have been found wide adoption mainly for regression and function approximation. Let us consider a set of N pairs of observations or measurements $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$, perturbed by an additive independent noise. More specifically, we assume the following observation model,

$$y_i = f(\mathbf{x}_i) + e_i, \tag{1}$$

where $e_i \sim \mathcal{N}(0, \sigma^2)$, $f(\mathbf{x})$ is an unknown latent function and $\mathbf{x} \in \mathbb{R}^d$. In a GP approach [6], we assume that the vector $[f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$ where $\mathbf{K}_{i,j} := k(\mathbf{x}_i, \mathbf{x}_j)$ is a $N \times N$ covariance matrix and the kernel function $k(\mathbf{x}, \mathbf{x}')$ is, for instance,

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\lambda^2}\right). \tag{2}$$

Moreover, a GP prior over the latent function $f(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}'))$ means that each vector of values of f evaluated at different $\mathbf{x}$'s is Gaussian distributed with zero mean and covariance matrix obtained by $k(\mathbf{x}, \mathbf{x}')$. The hyper-parameters of the GP model are $\boldsymbol{\theta} = [\lambda, \sigma]$ where λ determines the length-scale of the kernel function and σ is the standard deviation of the additive Gaussian noise in the observation model in Eq. (1). The goal is to learn the latent function $f(\mathbf{x})$ given the received data points $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]$ and $\mathbf{y} = [y_1, \dots, y_N]^\top$. Given the previous assumptions, considering a generic test location $\mathbf{x}_*$, the posterior density of the random variable $f(\mathbf{x}_*)$ is Gaussian, $p(f(\mathbf{x}^*)|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) \sim \mathcal{N}\left(\mu_{\text{GP}}(\mathbf{x}_*), \sigma^2_{\text{GP}}(\mathbf{x}_*)\right)$, where

$$\begin{aligned}\mu_{\text{GP}}(\mathbf{x}_*) &= \widehat{f}(\mathbf{x}_*|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta}) \\ &= \mathbf{k}_*^\top(\mathbf{K} + \sigma^2\mathbf{I}_N)^{-1}\mathbf{y},\end{aligned}$$

and $\sigma^2_{\text{GP}}(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top(\mathbf{K} + \sigma^2\mathbf{I}_N)^{-1}\mathbf{k}_*$, $\mathbf{k}_* = [k(\mathbf{x}_*, \mathbf{x}_1), \dots, k(\mathbf{x}_*, \mathbf{x}_N)]^\top$ is a $N \times 1$ vector.

III. Learning by Marginal Likelihood (ML)

In this section, we describe two possible procedures for tuning the hyperparameters $\boldsymbol{\theta}$ considering the *marginal likelihood* obtained by the assumed GP model. Given the assumptions described in the previous section, we have the following marginal likelihood

$$p_{\text{ML}}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) \propto \exp\left(V(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})\right), \tag{3}$$

where

$$V(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = -\mathbf{y}^\top(\mathbf{K} + \sigma^2\mathbf{I}_N)^{-1}\mathbf{y} - \log\left[\det(\mathbf{K} + \sigma^2\mathbf{I}_N)\right].$$

For simplicity, we consider using proper or improper (whenever possible) a uniform prior density $p(\boldsymbol{\theta})$ defined for $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, and thus the posterior density of the hyperparameters becomes

$$\begin{aligned} p_{\text{ML}}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y}) &\propto p_{\text{ML}}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})p(\boldsymbol{\theta}), \qquad (4)\\ &\propto p_{\text{ML}}(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})\mathbb{I}(\boldsymbol{\theta}), \qquad (5)\end{aligned}$$

where $\mathbb{I}(\boldsymbol{\theta}) = 1$ if $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, i.e., where the uniform prior is defined, and $\mathbb{I}(\boldsymbol{\theta}) = 0$ otherwise, if $\boldsymbol{\theta} \notin \boldsymbol{\Theta}$. In this case, the most used approaches are the *Maximum a Posteriori* (MAP) estimator, defined as

$$\widehat{\theta}_{\text{MAP}} = \arg\max p_{\text{ML}}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y}), \tag{6}$$

or alternatively, we can compute (approximately, in general) the *Minimum Mean Square Error* (MMSE) estimator,

$$\widehat{\theta}_{\text{MMSE}} = \int_{\boldsymbol{\Theta}} \boldsymbol{\theta} p_{\text{ML}}(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y}) d\boldsymbol{\theta}. \tag{7}$$

Another classical approach in general machine learning methods is the *Cross-Validation* (CV) approach, which does not take into account the marginal likelihood. In the next section, we introduce a probabilistic version of the CV procedure to allow the definition of both MAP and MMSE estimators, in the same fashion as we have described above.

IV. Probabilistic Cross-Validation (PCV)

We introduce a probabilistic version of the classical CV approach (denoted as PCV), which employs a first regression model to find an approximation $\widehat{f}(\mathbf{x})$ (e.g., a GP regressor) and then consider the measurement equation $y = \widehat{f}(\mathbf{x}) + e$ where $e \sim \mathcal{N}(0, \sigma^2)$ as observation model, in order to tune the hyper-parameters. Note that PCV is different to the procedure described in [6, Chapter 5] where the marginal likelihood induced by the GP model is again used within a CV context. We expect that PCV is more robust with respect to a mismatch with the true data distribution and the chosen model. PCV can be easily applied for tuning a larger number of hyperparameters by means of gradient-descent techniques and/or Monte Carlo methods [11]–[13], [18], [19].

The underlying idea of the proposed method is to define the prediction error as a probabilistic function. Thus, we split the data in n disjoint subsets $\mathbf{X}^{(n)}$ as in the classical CV, and define the error for a particular subset as a probabilistic cross-validation distribution p_{CV} of the predictions given the data of the remaining datasets and parameters, $p_{\text{CV}}(y^{(n)}|y^{(n-1)}, \mathbf{X}^{(n-1)}, \theta)$. By doing so we can cast this distribution inside the GP framework and optimize the error using any optimization technique.

For instance, consider the case where we split the dataset $\mathcal{D}$ in two disjoint subsets such as that $N = N_1 + N_2$, $\mathcal{D} = \mathcal{D}^{(1)} \cup \mathcal{D}^{(2)}$, with $\mathcal{D}^{(1)} = \{\mathbf{x}_i^{(1)}, y_i^{(1)}\}_{i=1}^{N_1}$ and $\mathcal{D}^{(2)} = \{\mathbf{x}_i^{(2)}, y_i^{(2)}\}_{i=1}^{N_2}$. We also denote as $\mathbf{X}^{(k)} = [\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_{N_1}^{(k)}]$ and $\mathbf{y}^{(k)} = [y_1^{(k)}, \dots, y_{N_2}^{(k)}]$ for $k = 1, 2$. The two-fold PCV technique is formed by the following steps:

Step 1. Given the first subset $\mathcal{D}^{(1)}$, obtain an approximation $\widehat{f}(\mathbf{x}|\mathbf{y}^{(1)}, \mathbf{X}^{(1)}, \boldsymbol{\theta})$ using any regression procedure. In the rest of this work, we consider a GP model for a fair comparison with the ML procedure.

Step 2. Considering the following log-likelihood function

$$\log[p_{\text{CV}}(\mathbf{y}^{(2)}|\mathbf{y}^{(1)}, \mathbf{X}^{(1)}, \boldsymbol{\theta})] = \\ -\frac{\sum_{i=1}^{N_2}\left(y_i^{(2)} - \widehat{f}(\mathbf{x}_i^{(2)}|\mathbf{y}^{(1)}, \mathbf{X}^{(1)}, \boldsymbol{\theta})\right)^2}{2\sigma^2} - \frac{1}{2}\log(C_2\sigma^2) \tag{8}$$

where $C_2 = N_2(2\pi)^{N_2}$, find an estimator $\widehat{\boldsymbol{\theta}}^{(1)}$ of the hyper-parameters $\boldsymbol{\theta}$ assuming the observation model

$$y_i^{(2)} = \widehat{f}(\mathbf{x}_i^{(2)}|\mathbf{y}^{(1)}, \mathbf{X}^{(1)}, \boldsymbol{\theta}) + \epsilon_i, \tag{9}$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

Step 3. Repeat the procedure above switching $\mathcal{D}^{(1)}$ with $\mathcal{D}^{(2)}$ to obtain $\widehat{\boldsymbol{\theta}}^{(2)}$ and return $\widehat{\boldsymbol{\theta}} = \frac{1}{2}(\widehat{\boldsymbol{\theta}}^{(1)} + \widehat{\boldsymbol{\theta}}^{(2)})$.

The previous procedure can be extended considering $K \geq 2$ disjoint subsets $\mathcal{D}^{(k)} = \{\mathbf{x}_i^{(k)}, y_i^{(k)}\}_{i=1}^{N_k}$, $N = \sum_{k=1}^{K} N_k$, such that

$$\mathcal{D} = \mathcal{D}^{(1)} \cup \mathcal{D}^{(2)} \cup \cdots \cup \mathcal{D}^{(K)},$$

and denoting again $\mathbf{X}^{(k)} = [\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_N^{(k)}]$ and $\mathbf{y}^{(k)} = [y_1^{(k)}, \ldots, y_{N_k}^{(k)}]$ with $k = 1, \ldots, K$. Moreover, we define as

$$\{\mathbf{x}^{(\neg k)}, \mathbf{y}^{(\neg k)}\} = \bigcup_{i=1; i\neq k}^{K} \mathcal{D}^{(i)},$$

all the data points that do not belong to $\mathcal{D}^{(k)}$. At the k-th iteration, the log-likelihood function is

$$\log[p_{\text{CV}}(\mathbf{y}^{(\neg k)}|\mathbf{y}^{(k)}, \mathbf{X}^{(k)}, \boldsymbol{\theta})] = \log[p_{\text{CV}}(\mathbf{y}^{(\neg k)}|\mathcal{D}^{(k)}, \boldsymbol{\theta})] = \\ -\frac{\sum_{i=1}^{N} \left(y_i^{(\neg k)} - \widehat{f}(\mathbf{x}_i^{(\neg k)}|\mathbf{y}^{(k)}, \mathbf{X}^{(k)}, \boldsymbol{\theta})\right)^2}{2\sigma^2} - \log(C_{-k}\sigma^2),$$

where $C_{-k} = (N - N_k)(2\pi)^{N-N_k}$. At each iteration, one estimator $\widehat{\boldsymbol{\theta}}^{(k)}$ is obtained and, after K iterations, the final estimator is given by

$$\widehat{\boldsymbol{\theta}} = \frac{1}{K}\sum_{k=1}^{K} \widehat{\boldsymbol{\theta}}^{(k)}. \tag{10}$$

As for the marginal likelihood procedure, in PCV we have different possibilities for obtaining the estimators $\widehat{\boldsymbol{\theta}}^{(k)}$. Considering a prior $p(\boldsymbol{\theta})$ and building the corresponding posterior density, the MAP estimator is

$$\widehat{\theta}_{\text{MAP}}^{(k)} = \arg\max p_{\text{CV}}(\boldsymbol{\theta}|\mathcal{D}^{(k)}, \mathbf{y}^{(\neg k)}), \tag{11}$$

and the MMSE estimator is defined as

$$\widehat{\theta}_{\text{MMSE}}^{(k)} = \int_{\boldsymbol{\Theta}} \boldsymbol{\theta} p_{\text{CV}}(\boldsymbol{\theta}|\mathcal{D}^{(k)}, \mathbf{y}^{(\neg k)}) d\boldsymbol{\theta}. \tag{12}$$

Considering a uniform prior $p(\boldsymbol{\theta})$, the MAP estimator in Eq. (11) coincides with the standard CV solution. In the following section, we compare the performance PCV and ML procedures and the MAP and MMSE estimators.

Remark. Considering a uniform prior $p(\boldsymbol{\theta})$ over the hyper-parameter, the MAP estimator in Eq. (11) coincides with the solution of the standard CV solution.

A. Further observations

Like in the ML approach, the probabilistic version of CV allows the possibility of computing several other estimators. For instance, one could be interested in using the median $\widehat{\boldsymbol{\theta}}_{\text{MED}}$ of the posterior p_{CV} (or p_{ML}). This estimator $\widehat{\boldsymbol{\theta}}_{\text{MED}}$ is generally more robust with respect to the presence of outliers in the dataset [17], [20]. However, surprisingly this alternative is employed rarely in literature. Other possibilities often used in robust statistics could be applied, for instance, the so-called L-estimators or M-estimators [17]. In this preliminary research, we focus on the MAP and MMSE estimators and we leave further comparisons with other estimators as future work. We believe that these alternatives can have a positive impact in terms of the efficiency and robustness of the regression methods, especially when applied to structured data robust losses make a difference.

V. Experimental Results

Let us consider we observe $\mathcal{D}_{\text{all}} = \{x_i, y_i\}_{i=1}^{2N}$ generated by the following model

$$y_i = \sin(\omega x_i) + \epsilon_i, \tag{13}$$

with $\omega = 0.2\pi$, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. The system's outputs y_i are not a noisy version of a GP realization. However, note that a sinusoidal function $\sin(\omega x)$ can still be easily approximated by a GP model with kernel in Eq. (2). We compare the ML and PCV procedures computing both MAP and MMSE estimators, and averaging the results over 500 independent runs. The MAP estimators are obtained by stochastic optimization by a simulated annealing (SA) procedure [12], [13] and the MMSE are obtained by the use of Metropolis-Hastings (MH) method [13]–[15]. For the sake of a fair comparison, the same computational methods and with the same parameters are used for the ML or PCV approaches.

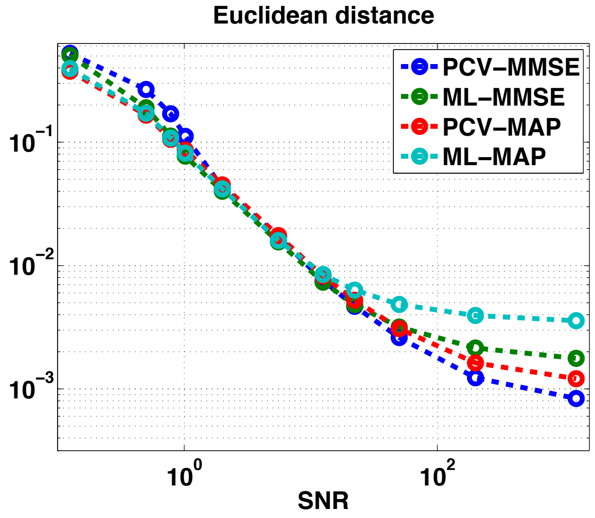

Figure 1. Euclidean distance $D(\widehat{f}, f)$ as function of the Signal-to-Noise Ratio (SNR), in a log-log plot (setting $N = 50$).

At each run of this experiment, the $2N$ pairs of data are generated according to the model and draw $x_i \sim \mathcal{U}([0, 20])$. Then, we permute randomly the pair of data in $\mathcal{D}_{\text{all}}$ and divide in two disjoint subsets of N pairs of data each one, $\mathcal{D}_{\text{traning}}$ and $\mathcal{D}_{\text{test}}$, i.e., we have $\mathcal{D}_{\text{all}} = \mathcal{D}_{\text{training}} \cup \mathcal{D}_{\text{test}}$. The first subset $\mathcal{D}_{\text{training}}$ is used for obtaining an estimation of the hyper-parameters $\boldsymbol{\theta}$ and the second one $\mathcal{D}_{\text{test}}$ is employed to compute the Mean Square Error (MSE) in prediction obtained

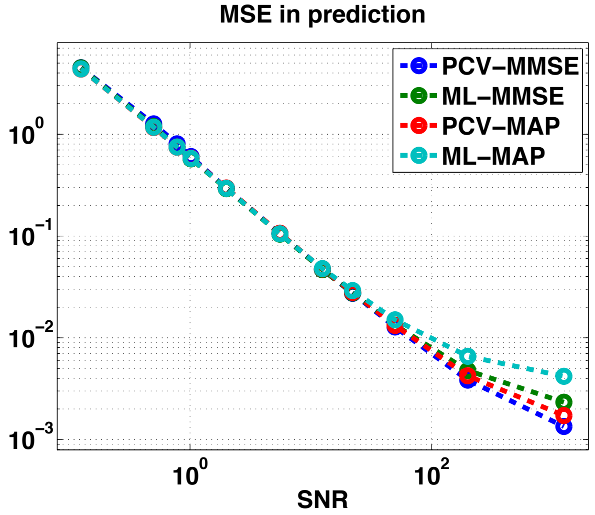

Figure 2. MSE in prediction considering $\mathcal{D}_{\text{test}}$ as function of the Signal-to-Noise Ratio (SNR), in a log-log plot (setting $N = 50$).

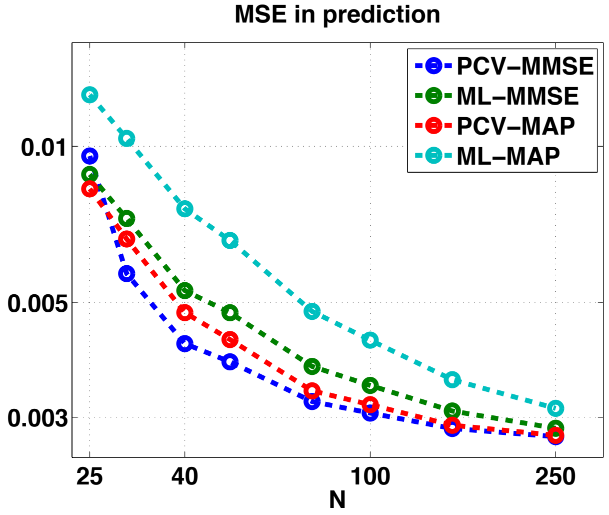

Figure 4. MSE in prediction considering $\mathcal{D}_{\text{test}}$ as function of the number of the data points N, in a log-log plot (setting $\sigma = 0.05$).

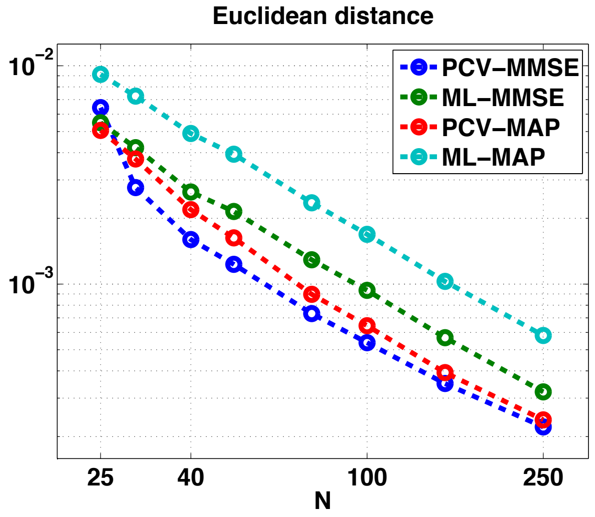

Figure 3. Euclidean distance $D(\widehat{f}, f)$ as function of the number of the data points N, in a log-log plot (setting $\sigma = 0.05$).

with different methods. For the PCV procedure, the first subset $\mathcal{D}_{\text{training}}$ is randomly split into $K = 2$ disjoint subsets $\mathcal{D}_{\text{training}} = \mathcal{D}^{(1)}_{\text{training}} \cup \mathcal{D}^{(2)}_{\text{training}}$, and proceeds as described in the previous section. At each run, we compute the MSE in the test phase considering $\mathcal{D}_{\text{test}}$ and also the Euclidean (L_2) distance $D(\widehat{f}, f)$ between the approximated regression function $\widehat{f}(x)$ given an estimator $\widehat{\boldsymbol{\theta}}$ obtained with the different techniques, and $f(x) = \sin(\omega x)$ with $x \in [0, 20]$, i.e., we approximate the integral below

$$D(\widehat{f}, f) = \int_0^{20} (\widehat{f}(x) - f(x))^2 dx.$$

We repeat the procedure above for different values of noise standard deviation σ (with $N = 50$) and different number of data N (with $\sigma = 0.05$). In Figures 1-2, we show the $D(\widehat{f}, f)$ and MSE as function of the Signal-to-Noise Ratio (SNR). In Figures 3-4, we show the results as function of the number of data points N. Observing the results, we can see that the performance are close for low SNR values and becomes more similar as N grows. The PCV approach is preferable for high SNR values. Furthermore, MMSE estimators are preferable for high SNR values whereas MAP estimators seem to have some small benefits for low SNR values. In general, the PCV-MAP approach outperforms the ML-MAP approach in all cases. PCV-MMSE seems to suffer small SNR values and small amount of data.

VI. Conclusions

We have presented a robust probabilistic cross-validation approach based on splitting the model definition in two different parts. It allows to find a set of parameters that minimizes directly the prediction error. We compared it with procedures involving ML in order to tune the hyperparameters of a GP regression model. In our experiments, we observed that PCV is preferable with more favourable SNR values. Our study suggests that ML-MMSE estimators should be preferred to ML-MAP estimators. With small SNR values, MAP estimators present some minimal benefits. As future work, we plan to compare other kind of estimators such as the robust alternatives called L and M-estimators. They can have a positive impact in terms of the robustness of the regression methods.

Acknowledgements

This work has been supported by the European Research Council (ERC) through the ERC Consolidator Grant SEDAL ERC-2014-CoG 647423, and the Spanish Ministry of Economy and Competitiveness (MINECO) through the projects TIN2015-64210-R and TEC2016-77741-R.

References

[1] C. K. I. Williams and C. E. Rasmussen, "Gaussian processes for regression," in *Neural Information Processing Systems, NIPS 8*. MIT Press, 1996, pp. 598–604.

[2] G. Camps-Valls, J. Verrelst, J. Munoz-Mari, V. Laparra, F. Mateo-Jimenez, and J. Gomez-Dans, "A survey on gaussian processes for earth-observation data analysis: A comprehensive investigation," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 58–78, June 2016.

[3] M. Kuss and C. Rasmussen, "Assessing approximate inference for binary Gaussian process classification," *Machine learning research*, vol. 6, pp. 1679–1704, Oct 2005.

[4] N. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *Machine learning research*, vol. 6, pp. 1783–1816, Nov 2005.

[5] A. O'Hagan and J. F. C. Kingman, "Curve fitting and optimal design for prediction," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 40, no. 1, pp. 1–42, 1978.

[6] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. New York: The MIT Press, 2006.

[7] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.

[8] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.

[9] M. U. Gutmann and J. Corander, "Bayesian optimization for likelihood-free inference of simulator-based statistical models," *Journal of Machine Learning Research*, vol. 16, pp. 4256–4302, 2015.

[10] J. Mockus, "Bayesian approach to global optimization," *Kluwer Academic Publishers, Dordrecht*, 1989.

[11] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," *Neural Information Processing Systems (NIPS) - arXiv:1206.2944*, pp. 1–9, 2012.

[12] S. K. Kirkpatrick, C. D. G. Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[13] L. Martino, V. Elvira, D. Luengo, J. Corander, and F. Louzada, "Orthogonal parallel MCMC methods for sampling and optimization," *Digital Signal Processing*, vol. 58, pp. 64–84, 2016.

[14] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer, 2004.

[15] L. Martino and V. Elvira, *Metropolis Sampling*. Wiley StatsRef: Statistics Reference Online, 2017.

[16] L. Martino and J. Read, "A multi-point Metropolis scheme with generic weight functions," *Statistics and Probability Letters*, vol. 82, no. 7, pp. 1445–1453, 2012.

[17] P. J. Huber, *Robust statistics*. Wiley-Interscience, New York, 2004.

[18] S. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, 1998.

[19] J. Read, L. Martino, and D. Luengo, "Efficient Monte Carlo optimization for multi-label classifier chains," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1– 5, 2013.

[20] T. P. Hettmansperger and J. W. McKean, *Robust nonparametric statistical methods*. Kendall's Library of Statistics. 5, 1998.