2016 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 13–16, 2016, SALERNO, ITALY

ON THE RELATIONSHIP BETWEEN ONLINE GAUSSIAN PROCESS REGRESSION AND KERNEL LEAST MEAN SQUARES ALGORITHMS

Steven Van Vaerenbergh[⋆]*, Jesus Fernandez-Bes*[† ‡]*, Víctor Elvira*[⋆]

[⋆] Dept. of Communications Engineering, University of Cantabria, Spain
[†] CIBER-BBN, Zaragoza, Spain,
[‡] BSICoS Group, I3A, IIS Aragón, University of Zaragoza, Zaragoza, Spain.
[⋆] Dept. of Signal Theory and Communications, Universidad Carlos III de Madrid, Spain

ABSTRACT

We study the relationship between online Gaussian process (GP) regression and kernel least mean squares (KLMS) algorithms. While the latter have no capacity of storing the entire posterior distribution during online learning, we discover that their operation corresponds to the assumption of a fixed posterior covariance that follows a simple parametric model. Interestingly, several well-known KLMS algorithms correspond to specific cases of this model. The probabilistic perspective allows us to understand how each of them handles uncertainty, which could explain some of their performance differences.

Index Terms— online learning, regression, Gaussian processes, kernel least-mean squares

1. INTRODUCTION

Gaussian Process (GP) regression is a state-of-the-art Bayesian technique for nonlinear regression [1]. Although GP models were proposed in the seventies [2], they did not become widely applied tools in machine learning until the last decade, mainly due to their computational complexity.

Through what is known as the "kernel trick", GP regression extends least squares to nonlinear estimation. By doing so, GP regression can be considered the natural Bayesian nonlinear extension of linear minimum mean square error estimation (MMSE) algorithms, which are central in signal processing [3]. Closely related to GPs are kernel methods [4], which have been successfully applied to several nonlinear signal processing problems, such as classification with support vector machines and kernel PCA for nonlinear dimensionality reduction. The main difference between Bayesian methods such as GPs and kernel methods is that the former provide a full probability distribution of the estimated variables, while the latter obtain only a point estimate.

Several kernel extensions of classical adaptive filters have been proposed in the literature (see for instance [5, 6] and the references therein). These algorithms, referred to as *kernel adaptive filtering* are mainly divided into two families, similar to the linear adaptive filtering literature: (i) kernel least-mean-squares (KLMS) algorithms [7, 8, 9], which are based on stochastic gradient minimization of the mean square error and have linear complexity per iteration w.r.t. the number of data points; (ii) and kernel recursive-least-squares (KRLS) algorithms [10], which recursively solve the least-squares problem, using quadratic complexity per iteration.

In [11], an online formulation of GP regression was obtained by deriving KRLS from a Bayesian point of view. An equivalent formulation for online GPs was presented in [12], though we will follow [11] as it offers a more intuitive choice for the variables and it provides a direct connection with KRLS algorithms. The online GP formulation from [11] adds two notable features to the KRLS literature: it allows the use of maximization techniques to set the hyperparameters without using cross-validation, and it provides an uncertainty measurement of the estimate.

KLMS algorithms are much more popular than KRLS, due to their low complexity. Interestingly though, as far as we know, there has not been a similar fully probabilistic interpretation of KLMS algorithms. Note that there exist some Bayesian interpretations of the LMS algorithm [13, 14], one of which considers kernels, albeit in a simplified setting [15].

In this work we provide a novel derivation of KLMS starting from a Bayesian model based on GPs. Using the sequential update rule of online GPs and a systematic approximation of its posterior covariance matrix, we are able to derive a KLMS formulation that generalizes the two main KLMS formulations, namely the KLMS algorithm [7] and the KNLMS algorithm [8]. The connection we establish with Gaussian processes sheds new light on the manner in which KLMS algorithms deal with uncertainty.

The work of S. Van Vaerenbergh is supported by the Spanish Ministry of Economy and Competitiveness, under projects PRISMA (TEC2014-57402-JIN) and RACHEL (TEC2013-47141-C4-3-R).

J. Fernandez-Bes' work was partially supported by projects TIN2013-41998-R, TEC2014-52289-R, and PRICAM S2013/ICE-2933.

The work of V. Elvira is supported by the Spanish Ministry of Economy and Competitiveness, under project TEC2013-41718-R.

978-1-5090-0746-2/16/$31.00

2. ONLINE GP REGRESSION

2.1. Gaussian process regression

Consider a set of N input-output pairs $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^D$ are D-dimensional input vectors and $y_i \in \mathbb{R}$ are scalar outputs. We assume that the observed data can be described by the following model,

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

in which f represents an unobservable *latent function* and $\varepsilon_i \sim \mathcal{N}(0, \sigma_n^2)$ is zero-mean Gaussian noise.

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [1]. To indicate that a random function $f(\mathbf{x})$ follows a Gaussian process we write it as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')).$$

All values of f at any locations $\mathbf{x}$ are jointly normally distributed, with $m(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$ representing the mean function and covariance function, respectively.

In a Bayesian regression setting, we are interested in inferring the predictive distribution of a new, unseen output y_* given the corresponding input $\mathbf{x}_*$ and the data $\mathcal{D}$. In particular, we take a Gaussian process as the prior over the latent function, and the vector of observations $[y_1, \ldots, y_n]^\top$ is related to the latent function through the likelihood function $p(\mathbf{y}|f)$. When the observations are contaminated with zero-mean Gaussian noise, as in Eq. (1), there exists a closed-form solution for the *posterior distribution* over functions, i.e. the distribution over the unknown function $f(x)$ after incorporating all the observed data. Specifically, the posterior of the function at any new location $\mathbf{x}_*$ is described by

$$p(f_*|\mathbf{x}_*, \mathcal{D}) = \mathcal{N}(\hat{f}_*, \hat{\sigma}_*^2).$$

When $m(\mathbf{x}) = 0$, which is a very common assumption, we obtain the following expressions [1]

$$\hat{f}_* = \mathbf{k}_*^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} \tag{2a}$$

$$\hat{\sigma}_*^2 = k_{**} - \mathbf{k}_*^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_*^\top \tag{2b}$$

where the covariances (or *kernel*) matrices $\mathbf{K}$ contain the elements $[\mathbf{K}]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and we have introduced the shorthand notations $\mathbf{k}_* = [k(\mathbf{x}_1, \mathbf{x}_*), \ldots, k(\mathbf{x}_N, \mathbf{x}_*)]^\top$ and $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$. The matrix inversion involved in Eqs. (2a) and (2b) leads to $\mathcal{O}(N^3)$ complexity.

2.2. Incremental GP updates

In an *online* scenario, the data pairs are made available on a one-at-a-time basis, i.e. $(\mathbf{x}_t, y_t)$ arrives at time t. Instead of recalculating the predictive distribution entirely once a new data pair $(\mathbf{x}_{t+1}, y_{t+1})$ arrives, i.e. by solving (2), it is more interesting to perform an incremental update. In this section we briefly review the sequential updates for online GP regression as presented in [11]. In order to avoid the unbounded growth of the involved matrices, the online learning process is typically coupled with a *sparsification* procedure.

At the t-th iteration of the online GP, the model contains the variables

$$\mathcal{M}_t = \{\mathcal{D}_t, \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t, \mathbf{Q}_t\},$$

where $\mathcal{D}_t$ is the observed data set that contains the data pairs $\{\mathbf{x}_i, y_i\}_{i=1}^t$; $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ are the mean and covariance matrices of the posterior $p(\mathbf{f}_t|\mathcal{D}_t) = \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$; and $\mathbf{Q}_t = \mathbf{K}_t^{-1}$ is the inverse covariance matrix corresponding to $\mathcal{D}_t$.

When a new data pair $(\mathbf{x}_{t+1}, y_{t+1})$ is obtained, the posterior distribution

$$p(\mathbf{f}_{t+1}|\mathcal{D}_{t+1}) = \mathcal{N}(\mathbf{f}_{t+1}|\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1})$$

is updated as

$$\boldsymbol{\mu}_{t+1} = \begin{bmatrix} \boldsymbol{\mu}_t \\ \hat{y}_{t+1} \end{bmatrix} + \frac{y_{t+1} - \hat{y}_{t+1}}{\hat{\sigma}_{yt+1}^2} \begin{bmatrix} \mathbf{h}_{t+1} \\ \hat{\sigma}_{ft+1}^2, \end{bmatrix}, \tag{3a}$$

$$\boldsymbol{\Sigma}_{t+1} = \begin{bmatrix} \boldsymbol{\Sigma}_t & \mathbf{h}_{t+1} \\ \mathbf{h}_{t+1}^\top & \hat{\sigma}_{ft+1}^2 \end{bmatrix} - \frac{1}{\hat{\sigma}_{yt+1}^2} \begin{bmatrix} \mathbf{h}_{t+1} \\ \hat{\sigma}_{ft+1}^2 \end{bmatrix} \begin{bmatrix} \mathbf{h}_{t+1} \\ \hat{\sigma}_{ft+1}^2 \end{bmatrix}^\top, \tag{3b}$$

where

$$\mathbf{q}_{t+1} = \mathbf{Q}_t \mathbf{k}_{t+1},$$
$$\mathbf{h}_{t+1} = \boldsymbol{\Sigma}_t \mathbf{q}_{t+1},$$

and the vector $\mathbf{k}_{t+1}$ has elements $[\mathbf{k}_{t+1}]_i = k(\mathbf{x}_i, \mathbf{x}_{t+1})$. Furthermore, the output variance is obtained as

$$\hat{\sigma}_{yt+1}^2 = \sigma_n^2 + \hat{\sigma}_{ft+1}^2,$$

and the variance of the latent function evaluations is

$$\begin{aligned} \hat{\sigma}_{ft+1}^2 &= k_{t+1} + \mathbf{k}_{t+1}^\top (\mathbf{Q}_t \boldsymbol{\Sigma}_t \mathbf{Q}_t - \mathbf{Q}_t) \mathbf{k}_{t+1} \\ &= \gamma_{t+1}^2 + \mathbf{q}_{t+1}^T \mathbf{h}_{t+1}. \end{aligned}$$

In order to further simplify equations, the variable

$$\gamma_{t+1}^2 = k_{t+1} - \mathbf{k}_{t+1}^T \mathbf{Q}_t \mathbf{k}_{t+1}$$

is introduced. The inverse kernel matrix $\mathbf{Q}_t$ can be updated efficiently through

$$\mathbf{Q}_{t+1} = \begin{bmatrix} \mathbf{Q}_t & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix} + \frac{1}{\gamma_{t+1}^2} \begin{bmatrix} \mathbf{q}_{t+1} \\ -1 \end{bmatrix} \begin{bmatrix} \mathbf{q}_{t+1} \\ -1 \end{bmatrix}^\top. \tag{4}$$

After applying Eqs. (3a), (3b), and (4), we obtain the updated model $\mathcal{M}_{t+1} = \{\mathcal{D}_{t+1}, \boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1}, \mathbf{Q}_{t+1}\}$.

At each step t of the learning process, the predictive distribution of a new observation y_{t+1} given all past data is a Gaussian $p(y_{t+1}|\mathcal{D}_t) = \mathcal{N}(y_{t+1}|\hat{y}_{t+1}, \hat{\sigma}_{yt+1}^2)$ with

$$\hat{y}_{t+1} = \mathbf{q}_{t+1}^\top \boldsymbol{\mu}_t = \mathbf{k}_{t+1}^\top \mathbf{Q}_t \boldsymbol{\mu}_t \tag{5a}$$

$$\hat{\sigma}_{yt+1}^2 = \sigma_n^2 + k_{t+1} + \mathbf{k}_{t+1}^\top (\mathbf{Q}_t \boldsymbol{\Sigma}_t \mathbf{Q}_t - \mathbf{Q}_t) \mathbf{k}_{t+1}. \tag{5b}$$

For more details we refer the reader to [11, 16].

3. KERNEL ADAPTIVE FILTERING AND KLMS

Kernel methods are a class of machine learning algorithms that are closely related to Gaussian processes. In many cases, kernel methods obtain the same solution as their GP counterpart. For instance, the most popular regression algorithm in kernel methods, kernel ridge regression (KRR) [17], obtains Eq. (2a) for predicting new outputs, which, in the kernel methods literature, is expressed as

$$\hat{f}_* = \boldsymbol{\alpha}^\top \mathbf{k}_* = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}_\star). \tag{6}$$

Vector $\boldsymbol{\alpha}$ contains the "kernel weights", which are found as $\boldsymbol{\alpha}^\top = (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1}\mathbf{y}$. Nonetheless, kernel methods do not follow a probabilistic Bayesian approach: their solution corresponds only to a point estimate, and they do not model the entire predictive distribution. This implies, among others, that kernel methods do not handle prediction uncertainty out-of-the-box. GP regression, in contrast, provides the predictive variance (2b) in addition to the predictive mean.

3.1. Kernel recursive least-squares

The kernel-methods counterpart of online GP regression is kernel recursive least-squares (KRLS, see for instance [10]), which obtains the KRR solution recursively. After receiving t data points, the kernel weights obtained by KRLS are those that solve the batch problem

$$\boldsymbol{\alpha}_t = (\mathbf{K}_t + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}_t. \tag{7}$$

The same weights can be obtained in online GP regression by computing

$$\boldsymbol{\alpha}_t = \mathbf{K}_t^{-1} \boldsymbol{\mu}_t = \mathbf{Q}_t \boldsymbol{\mu}_t, \tag{8}$$

which follows from Eq. (5a).

3.2. KLMS algorithms

Similar to online GP regression, updating the KRLS estimate in Eq. (7) with a new data point requires quadratic complexity. Kernel least-mean-squares (KLMS) algorithms alleviate this computational burden by performing stochastic gradient descent of the mean square error, resulting in linear complexity per time step [6].

KLMS algorithms can be categorized into two classes, depending on which kernel weights they update in each iteration to account for the prediction error[1]. We outline both approaches briefly in the remainder of this section.

[1] Both approaches are different approximations of the same update formulation in the *kernel feature space*; see [6] for details.

3.2.1. Type-I KLMS: concentrating the novelty

The first type of KLMS algorithm updates only one coefficient in order to compensate for the prediction error, at each time step. When the weight vector is allowed to grow, this update takes the form

$$\boldsymbol{\alpha}_{t+1}^{(I)} = \begin{bmatrix} \boldsymbol{\alpha}_t \\ \eta e_{t+1} \end{bmatrix}, \tag{9}$$

in which $e_{t+1} = y_{t+1} - \hat{y}_{t+1}$ is the instantaneous error.

Eq. (9) represents the basic update of what is known as the KLMS algorithm, proposed in [7]. In order to avoid the infinite growth of $\boldsymbol{\alpha}_t$, a more sophisticated version of this algorithm was presented in [9], known as Quantized Kernel Least Mean Square (QKLMS). When QKLMS receives a datum similar to a previously seen datum, for instance the i-th base it has stored, it does not expand $\boldsymbol{\alpha}_t$ but instead updates the corresponding weight α_i.

3.2.2. Type-II KLMS: spreading the novelty

A different strategy consists in updating all coefficients of $\boldsymbol{\alpha}_t$ in each iteration. This approach is followed for instance by the Kernel Normalized Least Mean Square (KNLMS) algorithm [8], whose update reads

$$\boldsymbol{\alpha}_{t+1}^{(II)} = \begin{bmatrix} \boldsymbol{\alpha}_t \\ 0 \end{bmatrix} + \eta \frac{e_{t+1}}{\epsilon + k_{t+1}^2 + \|\mathbf{k}_{t+1}\|^2} \begin{bmatrix} \mathbf{k}_{t+1} \\ k_{t+1} \end{bmatrix} \tag{10}$$

when $\boldsymbol{\alpha}_t$ is allowed to grow. Note that the rule (10) updates all coefficients in each iteration. In order to avoid unbounded growth, KNLMS follows a *coherence* criterion that promotes sparsity.

4. FROM ONLINE GP TO KLMS

The update of the KLMS kernel weights, $\boldsymbol{\alpha}_t$, can be obtained in terms of the online GP's predictive mean, $\boldsymbol{\mu}_t$, by elaborating $\boldsymbol{\alpha}_{t+1} = \mathbf{Q}_{t+1}\boldsymbol{\mu}_{t+1}$, which results in

$$\boldsymbol{\alpha}_{t+1} = \begin{bmatrix} \boldsymbol{\alpha}_t \\ 0 \end{bmatrix} + \frac{e_{t+1}}{\hat{\sigma}_{yt+1}^2} \begin{bmatrix} (\mathbf{Q}_t \boldsymbol{\Sigma}_t \mathbf{Q}_t - \mathbf{Q}_t)\mathbf{k}_{t+1} \\ 1 \end{bmatrix}. \tag{11}$$

Comparison of Eq. (11) with Eqs. (9) and (10) indicates that, in order to obtain a KLMS-like update rule, the covariance $\boldsymbol{\Sigma}_t$ is to be replaced by the following parametric model:

$$\boldsymbol{\Sigma}_t = \mathbf{K}_t \left(\beta \mathbf{K}_t + \mathbf{I}\right), \tag{12}$$

which implies

$$\mathbf{Q}_t \boldsymbol{\Sigma}_t \mathbf{Q}_t - \mathbf{Q}_t = \beta \mathbf{I}. \tag{13}$$

This substitution indicates that, instead of Eq. (2b), the following predictive variance is assumed

$$\hat{\sigma}_{ft+1}^2 = k_{t+1} + \beta \|\mathbf{k}_{t+1}\|^2. \tag{14}$$

The update of the predictive mean, Eq. (11), then simplifies to an expression that does not contain $\mathbf{\Sigma}_t$ nor $\mathbf{Q}_t$,

$$\boldsymbol{\alpha}_{t+1}^{\beta} = \begin{bmatrix} \boldsymbol{\alpha}_t \\ 0 \end{bmatrix} + \frac{e_{t+1}}{\sigma_n^2 + k_{t+1} + \beta\|\mathbf{k}_{t+1}\|^2} \begin{bmatrix} \beta\mathbf{k}_{t+1} \\ 1 \end{bmatrix}. \quad (15)$$

The update rule of Eq. (15) has linear complexity with respect to the number of processed data points, $t+1$.

4.1. Specific cases: $\beta = 0$ and $\beta = 1$

Setting $\beta = 0$ in Eq. (15) yields the update rule

$$\boldsymbol{\alpha}_{t+1}^{\beta=0} = \begin{bmatrix} \boldsymbol{\alpha}_t \\ \frac{1}{\sigma_n^2 + k_{t+1}} e_{t+1} \end{bmatrix}. \quad (16)$$

This rule is very similar to the KLMS update (9), and even identical when the learning rate is set to $\eta = 1/(\sigma_n^2 + k_{t+1})$. Note, furthermore, that $\beta = 0$ indicates that the posterior covariance from Eq. (12) reduces to

$$\mathbf{\Sigma}_t^{\beta=0} = \mathbf{K}_t. \quad (17)$$

In other words, for $\beta = 0$ we obtain a KLMS model that implies a fixed posterior covariance, equal to the prior covariance. The predictive variance, Eq. (14), simplifies to

$$\hat{\sigma}_{f\,t+1}^2|^{\beta=0} = k_{t+1}. \quad (18)$$

Under the adopted framework, this is the model that underlies algorithms such as KLMS from [7] and QKLMS from [9].

By setting $\beta = 1$ in Eq. (15), the update rule becomes

$$\boldsymbol{\alpha}_{t+1}^{\beta=1} = \begin{bmatrix} \boldsymbol{\alpha}_t \\ 0 \end{bmatrix} + \frac{e_{t+1}}{\sigma_n^2 + k_{t+1} + \|\mathbf{k}_{t+1}\|^2} \begin{bmatrix} \mathbf{k}_{t+1} \\ 1 \end{bmatrix} \quad (19)$$

which is very similar to the update of KNLMS, shown in Eq. (10). The implied posterior covariance then reads

$$\mathbf{\Sigma}_t^{\beta=1} = \mathbf{K}_t\mathbf{K}_t + \mathbf{K}_t, \quad (20)$$

and the predictive covariance from Eq. (14) now becomes

$$\hat{\sigma}_{f\,t+1}^2|^{\beta=1} = k_{t+1} + \mathbf{k}_{t+1}^\top\mathbf{k}_{t+1}. \quad (21)$$

Interestingly, this KLMS model implies a predictive covariance, for each point in space, that is larger than the prior covariance k_{t+1}. Furthermore, a closer inspection of the second term in Eq. (21) shows that the predictive variance grows as more training data is processed. According to the GP framework, this is the model that underlies type-II KLMS algorithms, in particular KNLMS from [8], for which $\beta = 1$, and in general any algorithm that corresponds to $\beta > 0$.

As seen through the adopted probabilistic perspective, the capability of Type-II KLMS algorithms to update all coefficients with each new data point is related to an increase in prediction uncertainty. These effects are illustrated in Fig. 1, which compares the interpreted predictive uncertainty for the different algorithms. GP regression exhibits the expected behavior, i.e. uncertainty shrinks around the observed data points. Type-I KLMS ($\beta = 0$) assumes a fixed predictive covariance, regardless of the number of observed data and the distances between them. The behavior of type-II KLMS (for instance with $\beta = 1$) is rather counterintuitive: its update procedure increases the predictive variance as more data points are observed, and this happens in the neighborhoods of those points. While an in-depth study is needed to extract solid conclusions from this observation, it is interesting to note that a similar behavior has been observed in the linear recursive least squares (RLS) algorithm with forgetting factor, see [11].

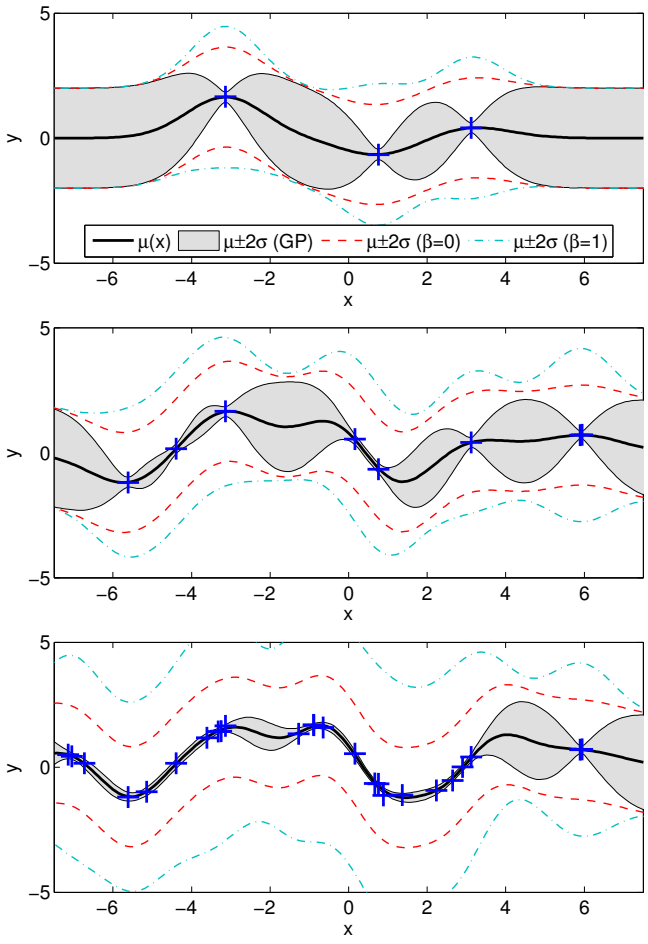

Fig. 1. Comparison of the predictive variance of three algorithms, for 3 data points (top plot, data is marked as blue crosses), 8 data points (middle), and 25 data points (bottom). The predictive mean of GP regression is indicated as the black curve that passes through the observations. The grey zone marks the GP mean plus/minus two standard deviations $\hat{\sigma}_y$, corresponding to the GP's 95% confidence interval. The dashed line marks the confidence interval for type-I KLMS ($\beta = 0$), and the dash-dot line marks the confidence interval for type-II KLMS with $\beta = 1$.

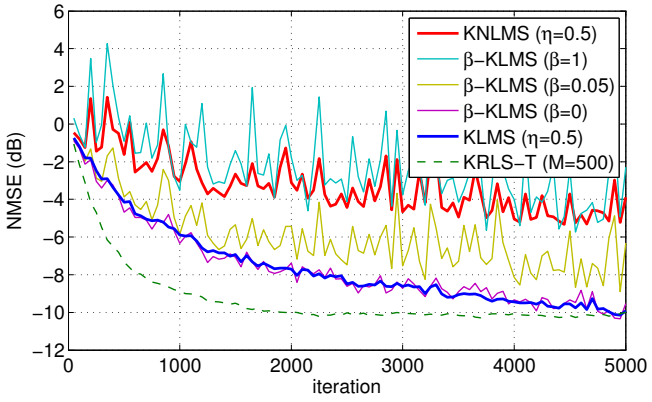

Fig. 2. Performance comparison for online prediction on the KIN40K benchmark regression problem.

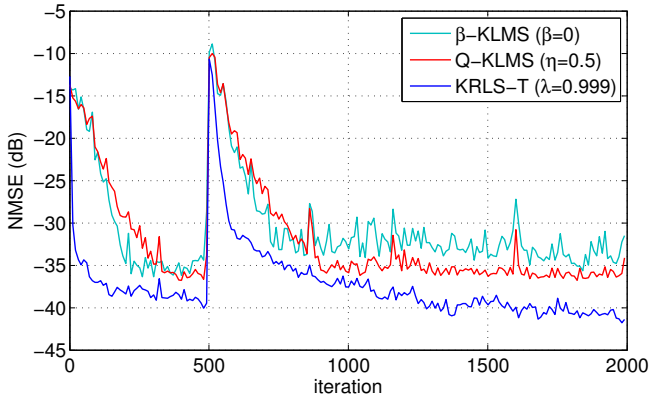

Fig. 3. Performance comparison for online prediction on a nonlinear channel with an abrupt change.

5. EXPERIMENTS

We illustrate the relationship between the discussed algorithms through a set of numerical experiments. We used the Matlab implementations found in the KAFBOX toolbox [18]. The code for these experiments is available at `http://gtas.unican.es/people/steven`.

5.1. Online regression on stationary data

In the first experiment, we wish to study the effect of different values of β in the KLMS algorithm from Eq. (15), which we will denote by β-KLMS. We evaluate this algorithm and three established kernel adaptive filtering algorithms on the stationary KIN40K benchmark.[2] This data set is obtained from the forward kinematics of an 8-link all-revolute robot arm, and it represents a very difficult regression problem. We randomly select 5000 data points for online training, and 5000 points for testing the regression.

The algorithms are considered in their evergrowing version here, in order to highlight only the influence of β. The KRLS-T algorithm, however, which implements the full online GP regression, is given a limited memory of 500 bases, for computational reasons. A Gaussian kernel was used for all algorithms, with parameters determined offline by standard GP regression, as detailed in [11].

The results are shown in Fig. 2. Each point of the learning curves corresponds to the test error on the entire test set. We observe that for $\beta = 1$, the β-KLMS algorithm obtains similar performance to KNLMS from [8]. For $\beta = 0$, the performance is almost identical to that of KLMS from [7].

[2]Available at `http://www.cs.toronto.edu/~delve/data/datasets.html`

5.2. Online prediction of time series with a model switch

In Fig. 3 we repeat a standard experiment from the kernel adaptive filtering literature. In particular, we consider a nonlinear system comprised of a linear channel followed by a nonlinearity, and the online regression task consists in predicting the next sample of the time series produced at its output. An abrupt change in the linear channel is triggered after 500 time steps, in order to test the algorithms' ability to reconverge. The experiment is repeated 5 times with random channel coefficients, and the average results are shown in Fig. 3. The same kernel was used for each algorithm.

In this experiment we observe that, as is usual in the LMS and KLMS literature, the β-KLMS model exhibits tracking behavior although it is based on a static data model. Furthermore, it does so without the need of including an additional parameter to control the update step size.

6. CONCLUSIONS

We studied the connections between online GP regression and KLMS, from a probabilistic perspective. We proposed a parametric model for fixing the posterior covariance that, when plugged into the update equations for online GP regression, yields the well-known equations of several KLMS algorithms. This approach allowed us to analyze the way in which KLMS algorithms implicitly handle uncertainty.

We furthermore categorized existing KLMS algorithms into two classes, depending on which coefficients they update during online operation: Type-I KLMS algorithms concentrate all novelty into a single coefficient, while type-II algorithms spread the novelty over many coefficients. According to the adopted probabilistic perspective, the former fixes its uncertainty regarding new data while the uncertainty of the latter grows as more data are processed.

Finally, while the proposed parametric model shows interesting features as a KLMS algorithm, it has several aspects,

such as the appropriate choice of the β parameter, that require a further analysis. Furthermore, the use of GPs as models for kernel adaptive filters could open the door to more sophisticated low-complexity algorithms, for instance by considering pseudo-inputs [19].

7. REFERENCES

[1] Carl Edward Rasmussen and Christopher K. I. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, 2005.

[2] Anthony O'Hagan and JFC Kingman, "Curve fitting and optimal design for prediction," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–42, 1978.

[3] Fernando Pérez-Cruz, Steven Van Vaerenbergh, Juan José Murillo-Fuentes, Miguel Lázaro-Gredilla, and Ignacio Santamaría, "Gaussian processes for nonlinear signal processing: An overview of recent advances," *IEEE Signal Processing Magazine*, vol. 30, pp. 40–50, July 2013.

[4] Bernhard Schölkopf and Alexander J Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*, MIT press, 2002.

[5] Weifeng Liu, José C. Príncipe, and Simon Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*, Wiley, 2010.

[6] Steven Van Vaerenbergh and Ignacio Santamaría, *Online Regression with Kernels*, chapter 21, pp. 477–501, Number Machine Learning & Pattern Recognition Series. Chapman and Hall/CRC, New York, 2014.

[7] Weifeng Liu, Puskal P. Pokharel, and José C. Príncipe, "The kernel least-mean-square algorithm," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 543–554, 2008.

[8] Cédric Richard, José Carlos M. Bermudez, and Paul Honeine, "Online prediction of time series data with kernels," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, Mar. 2009.

[9] Badong Chen, Songlin Zhao, Pingping Zhu, and José C. Príncipe, "Quantized kernel least mean square algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22–32, Jan. 2012.

[10] Yaakov Engel, Shie Mannor, and Ron Meir, "The kernel recursive least squares algorithm," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[11] Steven Van Vaerenbergh, Miguel Lázaro-Gredilla, and Ignacio Santamaría, "Kernel recursive least-squares tracker for time-varying regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1313–1326, Aug. 2012.

[12] Lehel Csató and Manfred Opper, "Sparse online Gaussian processes," *Neural Computation*, vol. 14, no. 3, pp. 641–668, 2002.

[13] Jesus Fernandez-Bes, Víctor Elvira, and Steven Van Vaerenbergh, "A probabilistic least-mean-squares filter," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2199–2203.

[14] Christian Huemmer, Roland Maas, and Walter Kellermann, "The NLMS algorithm with time-variant optimum stepsize derived from a Bayesian network perspective," *Signal Processing Letters, IEEE*, vol. 22, no. 11, pp. 1874–1878, 2015.

[15] Il Memming Park, Sohan Seth, and Steven Van Vaerenbergh, "Probabilistic kernel least mean squares algorithms," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 8272–8276.

[16] Miguel Lázaro-Gredilla, Steven Van Vaerenbergh, and Ignacio Santamaría, "A Bayesian approach to tracking with kernel recursive least-squares," in *2011 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, Sept. 2011, pp. 1 –6.

[17] Craig Saunders, Alexei Vinokourov, and John Shawetaylor, "String kernels, Fisher kernels and finite state automata," in *Advances in Neural Information Processing Systems*, 2002, pp. 633–640.

[18] Steven Van Vaerenbergh and Ignacio Santamaría, "A comparative study of kernel adaptive filtering algorithms," in *2013 IEEE Digital Signal Processing (DSP) Workshop and IEEE Signal Processing Education (SPE)*, Napa, CA, USA, July 2013, software available at `https://github.com/steven2358/kafbox`.

[19] James Hensman, Nicolo Fusi, and Neil D Lawrence, "Gaussian processes for big data," in *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI2013)*, A. Nicholson and P. Smyth, Eds. July 2013, AUAI Press.