GAUSSIAN PROCESSES FOR DIGITAL COMMUNICATIONS

Fernando Pérez-Cruz [*]

Gatsby Computational Neuroscience Unit
University College London
Queen's Square 17, WC1N 3AR, London UK
fernando@gatsby.ucl.ac.uk

Juan José Murillo-Fuentes [†]

Universidad de Sevilla
Dep. Teoría de la Señal y Comunicaciones
Paseo de los Descubrimientos sn, 41092
murillo@us.es

ABSTRACT

We present Gaussian Processes (GPs) for Digital communications. GPs can be used to construct analytical nonlinear regression functions, which can be suitable for digital communications in which linear solutions under perform. GPs can be cast as nonlinear MMSE and its hyperparameters can be easily learnt by maximum likelihood. We present some experimental results regarding multi-user detection in CDMA systems and show the GPs outperform linear and nonlinear state-of-the-art solutions.

1. INTRODUCTION

There is a vast literature in nonlinear methods for digital communications, as in many scenarios linear methods under perform. Neural Nets have been used for channel equalization [1] and Multi-User Detection (MUD) in CDMA systems [2, 3], to name a few. However, training times for these methods are long and unpredictable. Lately, Support Vectors Machines (SVMs), which were developed from well-founded learning theory results [4], have been used for channel equalization [5] and MUD [6]. All these methods need to solve an optimization problem to build the nonlinear predictor. The architecture of the Neural Net or the SVM hyperparameters have to be pre-specified, as they cannot be learnt for each problem because standard cross-validation techniques are not feasible in communication systems.

In this paper, we present a nonlinear estimation technique known as Gaussian Processes (GPs) for regression [7] as a novel detector for digital communications. GPs provide analytical answers to the estimation problem, if its covariance matrix is known. Hence, there is no need to solve an optimization problem. If the covariance matrix is not known, it can be learnt from training examples by maximum likelihood. Compared to the previous nonlinear tools, it does not need to pre-specify a structure/hyperparameters beforehand and therefore it can provide more accurate results as its hyperparameters are learnt for each instantiation of the problem. We propose to use this framework to solve the Multi-User Detection (MUD) problem in CDMA communication systems, which its optimal solution is known to be nonlinear [8].

2. GAUSSIAN PROCESSES FOR REGRESSION

Gaussian Processes (GPs) for regression is a Bayesian technique for nonlinear regression estimation. It assumes a zero-mean GP prior over the space of possible functions and a Gaussian likelihood model. The posterior can be analytically computed, it is a Gaussian density function, and the predictions given by the model are also Gaussians. Instead of presenting it from its GPs point of view, we present it as a Bayesian linear regression model[1]. We believe the latter is a simpler way to understand GPs for regression and it allows a straightforward comparison between the GP-MUD with the MMSE one.

Given a labelled training data set ($\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$, where the input $\mathbf{x}_i \in \mathbb{R}^{d\times 1}$ and the output $y_i \in \mathbb{R}$) and a new input location $\mathbf{x}^*$, we aim to predict the probability distribution for its output y^*, i.e. $p(y^*|\mathbf{x}^*, \mathcal{D})$. If we assume a Gaussian linear prediction model for y: $p(y|\mathbf{x}, \mathbf{w}) = \mathcal{N}(y; \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}), \sigma_\nu^2)$, where $\boldsymbol{\phi}(\cdot)$ defines a transformation of the input space, and a zero-mean Gaussian prior over $\mathbf{w}$, $p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \sigma_{\mathbf{w}}^2\mathbf{I})$, we can compute the posterior for the weight vector $\mathbf{w}$ using Bayes theorem:

$$p(\mathbf{w}|\mathcal{D}) = \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})} = \frac{1}{p(\mathbf{y}|\mathbf{X})}\prod_{i=1}^{n}\frac{\exp\left(-\frac{(y_i - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i))^2}{2\sigma_\nu^2}\right)}{\sqrt{2\pi\sigma_\nu^2}}\frac{\exp\left(-\frac{||\mathbf{w}||^2}{2\sigma_{\mathbf{w}}^2}\right)}{(2\pi\sigma_{\mathbf{w}}^2)^{d/2}} = \mathcal{N}(\mathbf{w}; \mu_{\mathbf{w}}, \Sigma_{\mathbf{w}}) \quad (1)$$

where $\mu_{\mathbf{w}} = \Sigma_{\mathbf{w}}\boldsymbol{\Phi}\mathbf{y}/\sigma_\nu^2$, $\Sigma_{\mathbf{w}}^{-1} = \boldsymbol{\Phi}^\top\boldsymbol{\Phi}/\sigma_\nu^2 + \mathbf{I}/\sigma_{\mathbf{w}}^2$, $\mathbf{y} = [y_1, \ldots, y_n]^\top$, $\boldsymbol{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), \ldots, \boldsymbol{\phi}(\mathbf{x}_n)]^\top$ and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$. Actually, the mean of the posterior can be computed as the maximum a posteriori (MAP) of (1), $\mu_{\mathbf{w}} = \operatorname{argmax}\{\log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) + \log p(\mathbf{w})\}$.

The prediction for y^* are obtained integrating out the posterior over $\mathbf{w}$ times its likelihood:

$$p(y^*|\mathbf{x}^*, \mathcal{D}) = \int p(y^*|\mathbf{x}^*, \mathbf{w})p(\mathbf{w}|\mathcal{D})d\mathbf{w} = \mathcal{N}(y; \mu_{y^*}, \sigma_{y^*}) \quad (2)$$

where

$$\mu_{y^*} = \boldsymbol{\phi}^\top(\mathbf{x}^*)\mu_{\mathbf{w}} = \mathbf{k}^\top\mathbf{C}^{-1}\mathbf{y} \quad (3)$$

$$\sigma_{y^*} = \boldsymbol{\phi}^\top(\mathbf{x}^*)\Sigma_{\mathbf{w}}\boldsymbol{\phi}(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) + \mathbf{k}^\top\mathbf{C}^{-1}\mathbf{k} \quad (4)$$

being $k(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}^\top(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_j)$, $(\mathbf{C})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) + \frac{\sigma_\nu^2}{\sigma_{\mathbf{w}}^2}\delta_{ij}$ and $\mathbf{k} = [k(\mathbf{x}^*, \mathbf{x}_1), \ldots, k(\mathbf{x}^*, \mathbf{x}_n)]$. The nontrivial steps needed to obtain (3) and (4) are detailed in [7]. The predicted value for y^* in (3) is the inner product of the MAP estimate of $\mathbf{w}$, $\mu_{\mathbf{w}}$, and the input vector, $\boldsymbol{\phi}(\mathbf{x}^*)$, described next.

2.1. Covariance Matrix

To get the estimation given by a GP model for regression, we only need to specify its covariance function $\mathbf{C}$. This matrix $\mathbf{C}$ represents

[*]Thanks Spanish MCYT funding, TIC2003-02602 and to EX2004-0698.
[†]Thanks Spanish MCYT agency for funding, TIC2003-03781.

[1]In [7], Williams introduces both views in a tutorial survey and shows their equivalency

the covariance matrix between the transformations $\boldsymbol{\phi}(\mathbf{x}_i)$ of the n training examples, which present a joint zero-mean Gaussian distribution (due to the GP prior over the space of functions). This covariance function plays the same role as the kernel in Support Vector Machines (SVMs) or any other kernel method, see [9] for further details.

If we design the regressor to be linear, we set $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$ (notice that $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$). We then need to specify the value of $\sigma_\nu^2/\sigma_{\mathbf{w}}^2$ to reach the desired solution. If this value is set to a small constant, which ensures that the matrix $\mathbf{C}$ is non-singular, The GP provides the same solution as the linear MMSE regressor.

We can also specify other covariance functions that yields non-linear regression estimates. The definition of the covariance function must capture any available information about the problem at hand. Typically a parametric form is proposed and its hyperparameters adjust for each particular instantiation of the regression problem. The chosen covariance function must construct positive definite matrices, for any set of input vectors $\{\mathbf{x}_i\}_{i=1}^n$, as it represents the covariance matrix of a multidimensional Gaussian distribution. A versatile covariance function, typically used in the literature, is described as follows:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \alpha_1 \exp\left(-\sum_{\ell=1}^{d} \gamma_\ell (x_{i\ell} - x_{j\ell})^2\right) + \alpha_2 \mathbf{x}_i^\top \mathbf{x}_j + \alpha_3 \delta_{ij} \tag{5}$$

Each term is weighted by an hyperparameter α_i, that need to be positive to construct positive definite matrices. Hence, we define $\boldsymbol{\theta} = [\log\alpha_1, \log\alpha_2, \log\alpha_3, \log\gamma_1, \ldots, \log\gamma_d]$ for the covariance function in (5), where we have used the logarithm of the hyperparamters to deal with an unconstrained optimization problem over $\boldsymbol{\theta}$. This covariance function contains 3 terms. The second term is the linear covariance function. Therefore, the GP model contains as a particular case the linear regressor ($\alpha_1 = 0$). The third term correspond to $\frac{\sigma_\nu^2}{\sigma_{\mathbf{w}}^2}\delta_{ij}$ in the definition of $\mathbf{C}$, which is considered as an extra hyperparameter of the covariance function. The first term is a radial basis kernel with a different length-scale for each input dimension. This term allows to construct generic nonlinear regression functions and eliminate those components that do not affect the solution, by setting its γ_ℓ to zero.

To set the hyperparameters of the covariance function for each specific problem, we define the likelihood function given the training set and compute its maximum. The maximum likelihood hyperparameters are used in (3) and (4) to predict the outputs to new input vectors. We can also define a prior over these hyperparameters, compute its posterior, and integrate them out to obtain predictions (similarly as we did for the weight vector in (2)). But, the posterior is non-analytical and the integration has to be done using sampling. Although this second approach is more principled, it is computational intensive and it will not be feasible for communications systems. For the interested readers, further details can be found in [7].

The likelihood function of the hyperparamters is defined as:

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{\sqrt{|2\pi\mathbf{C}_{\boldsymbol{\theta}}|}} \exp\left(-\frac{1}{2}\mathbf{y}^\top \mathbf{C}_{\boldsymbol{\theta}}^{-1}\mathbf{y}\right) \tag{6}$$

where $\boldsymbol{\theta}$ represents the hyperparameters of the covariance function and we have added the subscript $\boldsymbol{\theta}$ to $\mathbf{C}_{\boldsymbol{\theta}}$ to explicitly indicate that the covariance matrix depends on the hyperparamters. The negative log-likelihood of (6) can be minimised with any off-the-shelf optimiser.

Gaussian Processes for regression is a general nonlinear regression tool that, given the covariance function, provides an analytical solution to any regression estimation problem. It does not only provide point estimates, but it also gives confidence intervals for them. In GPs for regression, we perform the optimization step to set the hyperparameters of the covariance function by maximum likelihood. These hyperparameters have to be pre-specified for other nonlinear estimation tools as SVMs, or estimated by means of cross-validation. However, cross-validation need long training sequences, limiting the number of hyperparameters. Besides, this means solving multiple optimization problems first to obtain the best hyperparameters, increasing the computational burden. These are remarkable drawbacks in digital communications, since we face hard non-linear problems at limited computational resources and short training sequences. By exploiting the GPs framework, as stated in this paper, we avoid them.

3. MMSE MULTI-USER DETECTOR

The discrete baseband synchronous CDMA model in [8] transmits K bits (one per user) per unit of time, $\mathbf{b} = [b_1, b_2, ..., b_K]^\top$. Each user's bit is multiplied by its spreading code $\mathbf{h}_j$ with L chips and suffers an attenuation given by a_j. The L-chip signal at the receiver end is given by:

$$\mathbf{x} = \mathbf{G}\begin{bmatrix} \mathbf{HA} & 0 & \ldots & 0 \\ 0 & \mathbf{HA} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \mathbf{HA} \end{bmatrix}\begin{bmatrix} \mathbf{b}^t \\ \mathbf{b}^{t-1} \\ \vdots \\ \mathbf{b}^{t-M+1} \end{bmatrix} + \mathbf{n} = \mathbf{PB} + \mathbf{n} \tag{7}$$

where $\mathbf{H}$ is an $L \times K$ matrix whose columns contains the spreading codes, $\mathbf{A}$ is a $K \times K$ diagonal matrix with each user's amplitude as entries, $\mathbf{n}$ is an L-dimensional vector with additive white Gaussian noise (AWGN) and the K-dimensional $\mathbf{b}^t$ vector represents the transmitted bits at time t. We have pre-multiplied the received chips by $\mathbf{G}$ to include the effect of a dispersive channel in the CDMA system. The matrix $\mathbf{P}$ summarises the effect of the channel, the spreading codes and attenuations, and $\mathbf{B}$ is a KM-dimensional vector representing the transmitted bits.

Throughout the paper, we consider a channel with inter-symbolic interference characterised by its discrete channel impulse response:

$$\mathbf{G}(z) = \sum_{i=0}^{n_c-1} g_i z^{-i} \tag{8}$$

For this channel model the $L \times LM$ matrix $\mathbf{G}$ is described by:

$$\mathbf{G} = \begin{bmatrix} g_0 & g_1 & \ldots & g_{n_c-1} & & & \\ & g_0 & g_1 & \ldots & g_{n_c-1} & & \\ & & \ddots & \ddots & \ldots & \ddots & \\ & & & g_0 & g_1 & \ldots & g_{n_c-1} \end{bmatrix} \tag{9}$$

The matrix is completed with zeros to ensure it contains LM columns.

The objective for the MUD receiver is to recover the transmitted bit for each user. The standard linear MMSE-MUD receiver [8] is given by:

$$\hat{b}_j = \text{sign}(\mathbf{x}^\top \mathbf{v}_{\text{nc}-\text{mmse}}) = \text{sign}(\mathbf{x}^\top \mathbf{R}_{\mathbf{xx}}^{-1}\mathbf{p}_j) \tag{10}$$

which is known as the non-centralised receiver as it works directly over the received chips. The vector $\mathbf{p}_j$ is the j^{th} column of $\mathbf{P}$.

The centralised version projects the received signal to a lower dimensional space by multiplying the received bits by the matrix containing the chips, i.e. $\mathbf{r} = \mathbf{H}^\top \mathbf{x}$. In AWGN memoryless channels, there is no loss of information after this projection [8]. The linear centralized MMSE-MUD is described by:

$$\hat{b}_j = \text{sign}(\mathbf{r}^\top \mathbf{v}_{\text{c}-\text{mmse}}) = \text{sign}(\mathbf{r}^\top \mathbf{R}_{\mathbf{rr}}^{-1}\mathbf{H}^\top \mathbf{p}_j) \tag{11}$$

4. GP AS A NON-LINEAR MMSE

GPs predictions are given by (3), which is similar to (10), as it computes the inner product between the received chips and a pre-specified vector. Therefore, the GP-MUD for a CDMA system decides which was the transmitted bit for the j^{th} user according to:

$$\widehat{b}_j = \text{sign}(\mu_{y^*}) = \text{sign}(\boldsymbol{\phi}^\top(\mathbf{x})\mu_{w_j}) \tag{12}$$

This solution is identical to (10) when $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$. In this case μ_{w_j} yields

$$\mu_{w_j} = \underset{\mathbf{w}_j}{\text{argmin}}\{-\log p(\mathbf{b}|\mathbf{X}, \mathbf{w}_j) - \log p(\mathbf{w}_j)\} = \underset{\mathbf{w}_j}{\text{argmin}} \left\{ \frac{1}{2\sigma_\nu^2} \sum_{i=1}^{n} (b_i - \mathbf{w}_j^T \mathbf{x}_i)^2 + \frac{||\mathbf{w}_j||^2}{2\sigma_{\mathbf{w}_j}^2} \right\} \tag{13}$$

The only difference with the MMSE criterion is due to the second term in (13), i.e. the log of the prior. But its effects in the solution will fade away as we increase the number of examples and the sum in the first term will converge to its expectation. As in general, GP for regression will not use a covariance function that yields linear regressors its decisions can be interpreted as a nonlinear MMSE-MUD. The GP-MUD cannot be computed blindly, as the linear MMSE is, because we have a nonlinear transformation of the received symbols, prior to the detection process. A known training sequence has to be transmitted to compute μ_{w_j} prior to decide on the remaining bits. GPs can similarly be compared to the centralised MMSE-MUD, if we used $\mathbf{r} = \mathbf{H}^\top \mathbf{x}$ instead of $\mathbf{x}$ in (12).

The covariance function in (5) is a good kernel for solving the GP-MUD, because it contains a linear and a nonlinear part. The optimal decision surface for MUD is nonlinear, unless the spreading codes are orthogonal to each other, and its deviation from the linear solution depends on how strong the correlations between codes are. In most cases, a linear detector is very close to the optimal decision surface, as spreading codes are almost orthogonal, and only a minor correction is needed to achieve the optimal decision boundary. In this sense the proposed GP covariance function is ideal for the problem. The linear part can mimic the best linear decision boundary and the nonlinear part modifies it, where the linear explanation is not optimal. Also using a radial basis kernel for the nonlinear part is a good choice to achieve nonlinear decisions. Because, the received chips form a constellation of 2^K clouds of points with Gaussian spread around its centres.

5. EXPERIMENTAL RESULTS

In this section we include some simulation results for the synchronous CDMA system in (7). For comparison purposes, we include the BER for the GP-MUD, SVM-MUD in [6], and MMSE-MUD in [8]. We also depict the BER for the ideal case, memoryless channel without users interference, and the BER for the optimum centralized detector, as described in [6]. The SVM-MUD has been trained using a Gaussian kernel with its width equal to the noise standard deviation, as reported in [6], and its C parameter was chosen to minimise the BER on the test set.

We aim to illustrate the fast convergence of the GP in comparison to the SVM-MUD in [6] and the suboptimal decisions provided by centralised MUDs, when the channel is not memoryless. First, we reproduce the Example 2 in [6], where $K = 3$, $L = 8$ and $a_i = 1$, and we report the BER for Users 2 and 3. The spreading codes are described in [6] and the Channel response is given by:

$$c(z) = 0.4 + 0.9z^{-1} + 0.4z^{-2} \tag{14}$$

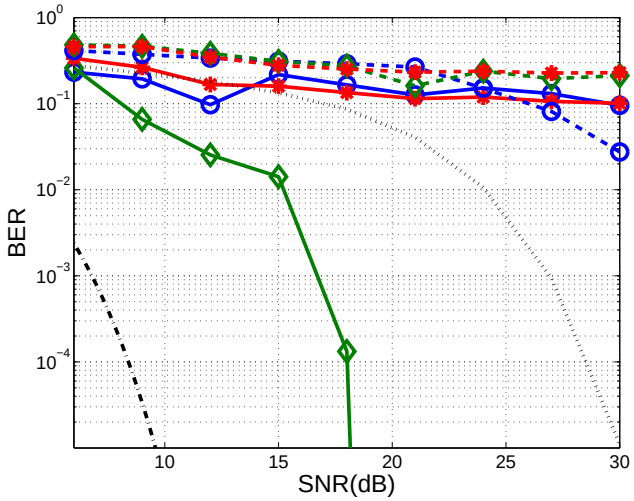

Fig. 1. BER for User 2 in a CDMA scenario with 3 users, $L = 8$ and 20 training samples. MMSE ($\circ$), SVM ($*$) and GP ($\diamond$) with non-centralized (solid) and centralized (dashed) MUD.

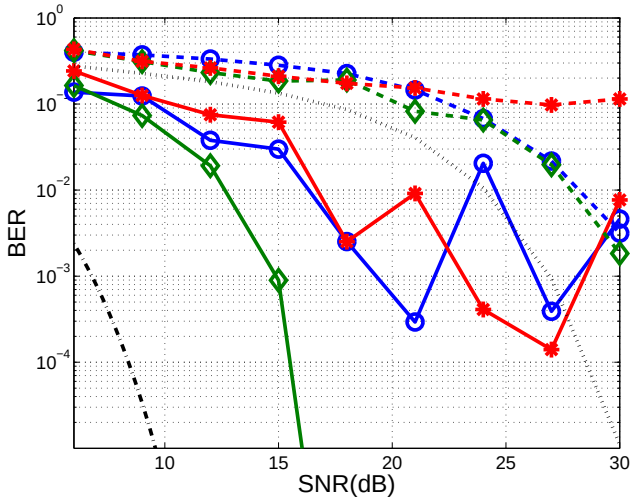

Fig. 2. BER for User 2 in a CDMA scenario with 3 users, $L = 8$ and 50 training samples. MMSE ($\circ$), SVM ($*$) and GP ($\diamond$) with non-centralized (solid) and centralized (dashed) MUD.

In Figure 1 and 2 we depict the BER for User 2 with 20 and 50 training samples respectively. We include the averaged results for 50 independent experiments and 10^5 test samples in each run. For 20 training examples, Figure 1, only the non-centralised GP-MUD detector provides meaningful results, while the others just report chance level performance. For 50 training examples in Figure 1 the non-centralised GP-MUD still provides remarkably the best results. The non-centralised SVM and MMSE-MUD are comparable to the GP-MUD for low SNR, but they fail to reduce the BER as the SNR increases. In the centralised case, GP and MMSE-MUD provide similar results closing to optimal performance, but the SVM-MUD is not able to reduce its error below 0.1. For longer training sets, more than 100 samples, the SVM-MUD exhibits the same performance as the GP and MMSE-MUD and they all tend to the optimum.

In Figure 3, we show the BER for User 3 with 100 training examples. It can be observed that the SVM and GP-MUD achieve the same BER, while the linear detectors under perform. In this case, the

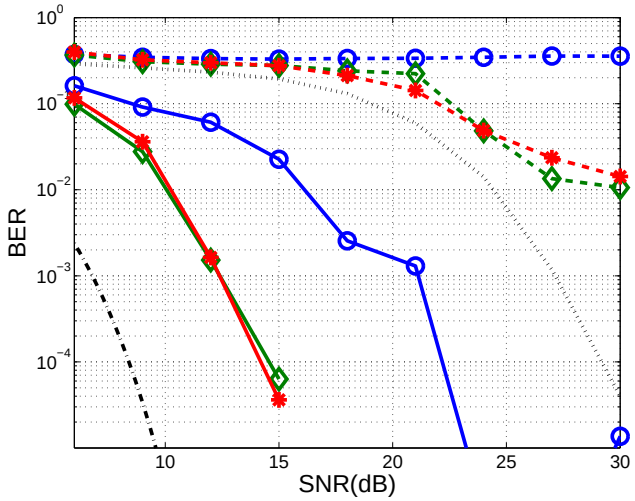

Fig. 3. BER for User 3 in a CDMA scenario with 3 users, $L = 8$ and 100 training samples. MMSE ($\circ$), SVM ($*$) and GP ($\diamond$) with non-centralized (solid) and centralized (dashed) MUD.

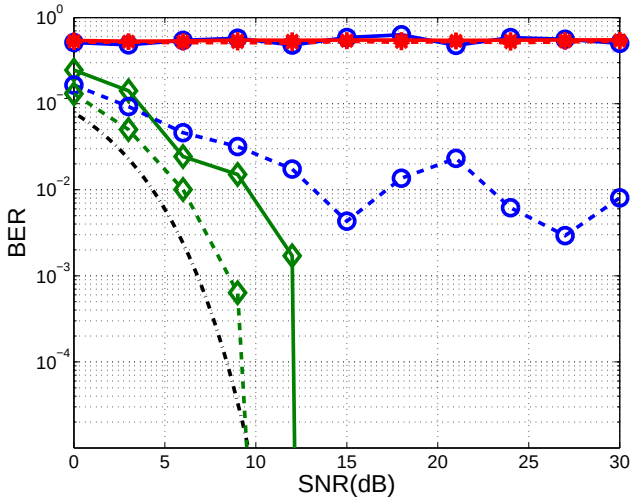

Fig. 4. BER for a CDMA scenario with $K = 8$ users and Gold sequences with $L = 31$. Non-centralized MMSE ($\circ$), SVM ($*$) and GP ($\diamond$) MUD for 30 (solid) and 200 (dashed) training samples. The UoI's power is 30dB bellow the interferer users. Also, the BER with memoryless channel and no interference (dash-dotted).

optimal solution is heavily nonlinear and the linear detectors cannot provide accurate answers. The SVM and GP will achieve the optimal BER performance in the centralised case for training sequences with more than 400 bits, as shown in [6]. In this example, it is clearly seen that centralised detectors are suboptimal, as relevant information is lost when reducing the problem dimensionality from L to K.

Finally, we include a more realistic scenario with $K = 8$ users and Gold spreading sequences with $L = 31$. The amplitudes of the interferer users are between 0 and 30 dB above the user of interest, so we can also check the performance of the GP-MUD detector for the near-far problem [8]. We also used the channel described in (14). In Figure 4 we plot the averaged results for 50 independent experiments with 30 (solid) and 200 (dashed) training samples. The BER was computed using 10^5 bits for each experiment. We only report the non-centralised detectors, as we have shown in the previous experiments that the centralised detectors are suboptimal for dispersive channels. The GP-MUD detector provides meaningful solutions for training sequences shorter that the spreading code, while the MMSE and the SVM-MUD provides chance level detection. For 200 training samples the GP detector is close to optimal performance, while the SVM is still providing 50% error rate. At this point, the MMSE detector is starting to reduce the BER, but BER is always above 10^{-3}, even for high SNRs.

6. CONCLUSIONS

In this paper, we have presented the Gaussian Processes for regression framework for digital communications. GPs are used to construct nonlinear regressors, according to the Minimum Mean Square Error criterion. The main advantages of the GPs are twofold. First, the solution given by the GPs is analytical, given its covariance matrix. Hence, it allows learning the covariance matrix by Maximum Likelihood, being able to adjust the hyperparameters in a single optimization step. Second, GPs provide confidence intervals, which can be used to assess the quality of the estimates. These characteristics differentiate them with respect to other nonlinear tools as SVMs or Neural Nets, in which an optimization step is needed to obtain the point estimates and no confidence intervals are provided.

We have shown that this framework is very useful for solving the MUD problem in CDMA systems compared to the linear MMSE criterion and SVMs, in which the hyperparameters are specified beforehand. We have tested the GP and SVM-MUD in a realistic scenario, in which the codes for the other users are unknown; the users arrive with different signal power; and we have dispersive channels. For this scenario we have shown that the GP-MUD is able to converge to the optimal solution providing accurate answers with very short training sequences, shorter than the chip length.

7. REFERENCES

[1] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Adaptative equalization of finite non-linear channels using multilayer perceptrons," *Signal Processing*, vol. 10, pp. 107–119, 1990.

[2] G. C. Orsak B. Aazhang, B. P. Paris, "Neural networks for multiuser detection in code-division multiple-access communications," *IEEE Transactions on Communications*, vol. 40, pp. 1212–1222, 1992.

[3] U. Mitra and H. V. Poor, "Neural network techniques for adaptive multiuser demodulation," *IEEE Journal Selected Areas on Communications*, vol. 12, pp. 1460–1470, 1994.

[4] V. N. Vapnik, *Statistical Learning Theory*, Wiley, 1998.

[5] F. Pérez-Cruz, A. Navia-Vázquez, P. L. Alarcón-Diana, and A. Artés-Rodríguez, "SVC-based equalizer for burst TDMA transmissions," *Signal Processing*, vol. 81, no. 8, pp. 1681–1693, Aug. 2001.

[6] L. Hanzo S. Chen, A. K. Samingan, "Support vector machine multiuser receiver for DS-CDMA signals in multipath channels," *IEEE Transactions on Neural Network*, vol. 12, no. 3, pp. 604–611, December 2001.

[7] C. Williams, "Prediction with gaussian processes: From linear regression to linear prediction and beyond," .

[8] S. Verdú, *Multiuser Detection*, Cambridge University Press, 1998.

[9] F. Pérez-Cruz and O. Bousquet, "Kernel methods and their potential use in signal processing," *Signal Processing Magazine*, vol. 21, no. 3, pp. 57–65, 2004.