GAUSSIAN PROCESSES FOR REGRESSION IN CHANNEL EQUALIZATION

Sebastián Caro[a†], *Fernando Pérez-Cruz*[b*] *and Juan José Murillo-Fuentes*[a†],

[a] Depto. Teoría de la Señal y Comunicaciones, Univ. de Sevilla.
Paseo de los Descubrimientos s/n, 41092 Sevilla, Spain.
{murillo,scaro}@us.es
[b] Depto. Teoría de la Señal y Comunicaciones, Univ. Carlos III de Madrid.
Avda de la Universidad 30. 28911. Leganés, Spain.
fernando@tsc.uc3m.es

ABSTRACT

In linear channels with additive white noise, linear symbol decision equalizers under perform. In this paper we present Gaussian Processes (GPs) for regression as new nonlinear equalizer for digital communications. GPs can be cast as nonlinear MMSE, a commom criterion in digital communications. Unlike other similar nonlinear kernel based methods such as the kernel adaline (KA) or SVMs, the solution of GPs is analytical and the hyperparameters can be easily learnt by maximum likelihood. Hence, by using GPs we avoid cross-validation or noise estimation, and improve convergence speeds. We present some experimental results regarding some linear and nonlinear channels to show that GPs clearly outperform linear and nonlinear state-of-the-art solutions.

1. INTRODUCTION

Channel equalization is a major issue in digital communications. Inter-symbol interference (ISI), which accounts for the linear distortion, is caused by the spreading of the symbol energy. Besides, nonlinear distortion is a consequence of the amplification and frequency conversion in transmitters and receivers. The objective of equalization is to minimize the effects of those distortions in order to reconstruct the transmitted sequence. In wireless communications, where bandwidth is a scarce resource and we send a training sequence every burst, short training sequences are a prerequisite.

Traditionally, equalization of linear channels has been considered equivalent to inverse filtering, where a linear transversal equalizer (LTE) is used to invert the channel response and its parameters are usually adjusted using minimum mean square error (MMSE) criteria. The optimal solution based on maximum likelihood has a complexity growing exponentially with the dimension. Alternatively, neural nets (NNs) can be used to solve this problem at a lower computational cost. Several NNs schemes have been used to solve this problem with varying degrees of success, such as the multi-layered perceptron (MLP) [1], radial basis functions (RBFs) [2], recurrent RBFs [3], self-organizing feature maps [4], wavelet neural networks [5], Kernel Adaline (KA) [6] and support vector machines (SVMs) [7]. Such structures usually outperform the LTE, especially when non-minimum phase channels are encountered. And they can also compensate for nonlinearities. The major drawback of such schemes is the need of long training sequences in order to work properly.

In this paper, we present a nonlinear estimation technique known as Gaussian Processes (GPs) for regression [8] as a novel channel equalizer. This approach has been already successfully applied to the multiuser detection problem in CDMA systems [9, 10]. GPs provide analytical answers to the estimation problem, avoiding solving an optimization problem. 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. These properties results, as illustrated in the experiments, in a remarkable reduction in the number of needed training samples. Even when we increase the length of the equalizer.

2. GAUSSIAN PROCESSES FOR REGRESSION

Gaussian Processes (GPs) for regression [11] 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-equalizer 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

* Supported through the Spanish MCYT Agency (TIC2003-02602).
† Supported through the Spanish MCYT Agency (TIC2003-03781).

[1] In [8], Williams introduces both views in a tutorial survey and shows their equivalency

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), $\mu_\mathbf{w} = \text{argmax}\{\log p(\mathbf{y}|\mathbf{X},\mathbf{w}) + \log p(\mathbf{w})\}$.

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

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

$$\sigma_{y^*}^2 = \boldsymbol{\phi}^\top(\mathbf{x}^*)\Sigma_\mathbf{w}\boldsymbol{\phi}(\mathbf{x}^*) = k(\mathbf{x}^*,\mathbf{x}^*) + \mathbf{k}^\top\mathbf{C}^{-1}\mathbf{k} \quad (4)$$

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 [8]. 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 $k(\mathbf{x}_i,\mathbf{x}_j)$. The matrix $\mathbf{C}$ represents 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 [12] 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 nonlinear 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:

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

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 [8].

The likelihood function of the hyperparamters is defined as:

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

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 channel equalization, since we face hard nonlinear problems at limited computational resources and short training sequences. By exploiting the GPs framework, as stated in this paper, we avoid them.

3. CHANNEL EQUALIZATION

A simple (and often used, e.g., [13]) model of a linear channel is given in Figure 1, where the transmitted digital signal $b(t)$ is a BPSK modulated signal, i.e., an independent and

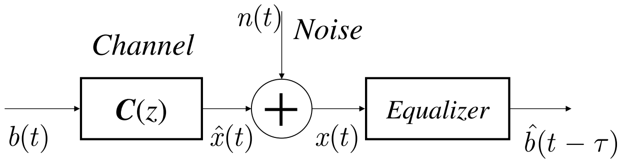

Figure 1: Simple discrete-time transmission channel model.

equiprobable sequence of symbols with values $\{-1, +1\}$. The linear time-invariant impulse response for the channel is given by

$$C(z) = \sum_{i=0}^{n_c-1} c_i z^{-i} \tag{7}$$

where n_c denotes the channel length and $n(t)$ is additive white Gaussian noise (AWGN). The task of the equalizer is to invert the response of the channel minimizing the noise and the ISI at the output. Then a detector provides an estimate of the delayed transmitted symbol denoted $\hat{b}(t-\tau)$ where τ is the equalizer delay. Since the equalizer block uses m received symbols to estimate $b(t-\tau)$, we form the equalizer input vector as

$$\mathbf{x}(t) = [x(t), x(t-1), \ldots, x(t-m+1)]^\top \tag{8}$$

where m is the equalizer length and we wish to compute $\hat{b}(t-\tau)$ given $\mathbf{x}(t)$.

The linear channel above can be modified by introducing a nonlinear function to model the receiver distortion. The model yields

$$\frac{\hat{X}(z)}{B(z)} = C(z) = \sum_{i=0}^{n_c-1} c_i z^{-i}$$
$$x(t) = f(\hat{x}(t)) + n(t) \tag{9}$$

where $f(\hat{x}(t))$ is a nonlinear function modelling the receiver. The distortion introduced in the transmitter could be modelled in the same way.

4. GAUSSIAN PROCESSES FOR CHANNEL EQUALIZATION

Since GPs predictions are given by (3), the GP-equalizer for a dispersive channel BPSK communications system decides which was the transmitted bit at time $(t-\tau)$ according to:

$$\hat{b}(t-\tau) = \mathrm{sign}(\mu_{y^*}) = \mathrm{sign}(\boldsymbol{\phi}^\top(\mathbf{x}^*)\mu_{\mathbf{w}}) = \mathrm{sign}(\mathbf{k}^\top \mathbf{C}^{-1}\mathbf{y}) \tag{10}$$

where $\mathbf{x}^*$ is the input vector to the equalizer. A known training sequence has to be transmitted to compute $\mu_{\mathbf{w}}$ prior to decide on the remaining bits. In that training data set, $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$, the input $\mathbf{x}_i$ is the received symbol vector (8)-(9) and the corresponding output y_i is $b(t-\tau)$. Regarding the covariance function, that in (5) is a good kernel for solving the GP-equalizer because it contains a linear and a nonlinear part. The optimal decision surface for equalization is nonlinear, and its deviation from the linear solution depends on the channel and the input dimension (length) of the equalizer. In some scenarios, e.g. for large dimensions, a linear detector is closer to the optimal decision surface, 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 symbol vectors form a constellation of 2^{n_c-1+m} clouds of points with Gaussian spread around its centres.

GPs predictions are given by (3), which is similar to the structure of a linear equalizer in that it computes the inner product between the received symbols and a pre-specified vector. Furthermore, when $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$, $\mu_{\mathbf{w}}$ yields

$$\mu_{\mathbf{w}} = \underset{\mathbf{w}}{\mathrm{argmin}}\{-\log p(\mathbf{b}|\mathbf{X},\mathbf{w}) - \log p(\mathbf{w})\} =$$
$$\underset{\mathbf{w}}{\mathrm{argmin}} \left\{ \frac{1}{2\sigma_\nu^2}\sum_{i=1}^{n}(b_i - \mathbf{w}^T\mathbf{x}_i)^2 + \frac{||\mathbf{w}||^2}{2\sigma_{\mathbf{w}}^2} \right\} \tag{11}$$

The only difference with the MMSE criterion is due to the second term in (11), 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-equalizer.

5. EXPERIMENTAL RESULTS

In this section we include some experimental results of channel equalization for a linear and a nonlinear channel. In both cases we first focus on the performance for different numbers of training samples at a fixed equalizer length and then for different equalizer lengths at a fixed training samples number. In all cases we depict the bit error rate (BER) along the normalized signal to noise ratio (SNR) for the linear MMSE, GP and SVM. The SVM-equalizer has been trained using a Gaussian kernel with its width equal to the noise standard deviation, as reported in [14], and its C parameter was chosen to minimise the BER. Eventually, the optimum Bayesian equalizer will be also included as reference.

As linear channel we used

$$C(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2} \tag{12}$$

in [6]. In Figure 2 we depict the BER for the linear MMSE ($\triangledown$), SVM ($+$) and GPs ($\diamond$) equalizers trained with $n = 50$, 100 and 400 samples. The length and delay of the equalizer was set to 4 and 1, respectively. We include the averaged results for 100 independent experiments and 10^5 test samples in each run. GP-equalizer provides remarkably the best results, close to optimal performance for $n = 400$. Besides, it can be seen that the MMSE-equalizer BER is always above 10^{-2}, even for high SNRs. In Figure 3 we repeated the same experiment for different equalizer lengths $m = 4$ ($\tau = 1$), 6 ($\tau = 2$) and 8 ($\tau = 3$), and 50 training samples. We observe that SVM and MMSE performance quite deteriorates as the equalizer length increases, while GPs performance improves towards the optimum. Besides, it is also interesting to point out that for $m = 6$ and $m = 8$ the GP-equalizer provides a good solution for just 50 training samples.

In Figure 4 we include the results of the experiment in Figure 2 considering a nonlinear channel as in (9) with $\frac{\hat{X}(z)}{B(z)}$

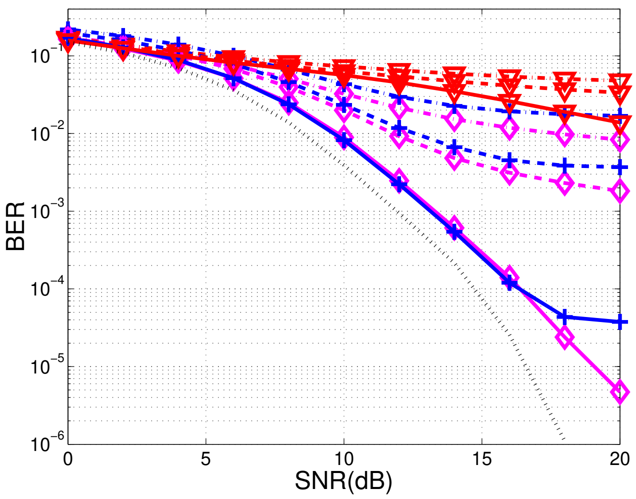

Figure 2: BER for a linear channel with optimal Bayesian (dotted), linear MMSE (▽), SVM (+) and GP's (⋄) equalization with $m = 4$ and $\tau = 1$, trained with 50 (dash-dotted), 100 (dashed) and 400 (solid) training samples.

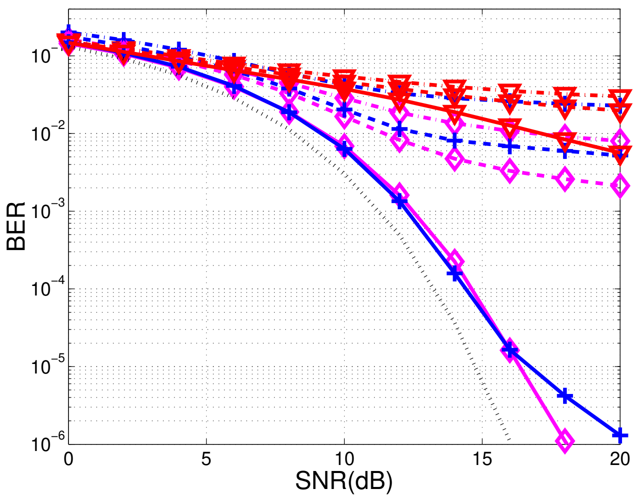

Figure 4: BER for a nonlinear channel with optimal Bayesian (dotted), linear MMSE (▽), SVM (+) and GP's (⋄) equalization with $m = 4$ and $\tau = 1$, trained with 50 (dash-dotted), 100 (dashed) and 400 (solid) training samples.

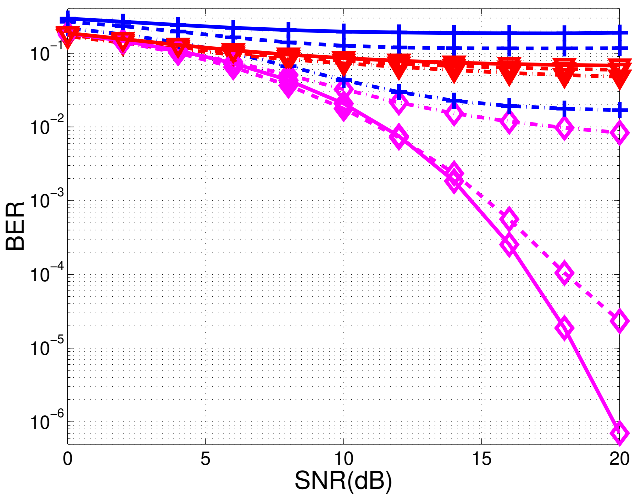

Figure 3: BER for a linear channel with linear MMSE (▽), SVM (+) and GP's (⋄) equalization with $n = 50$ training samples and length $m = 4$ (dash-dotted), $m = 6$ (dashed) and $m = 8$ (solid).

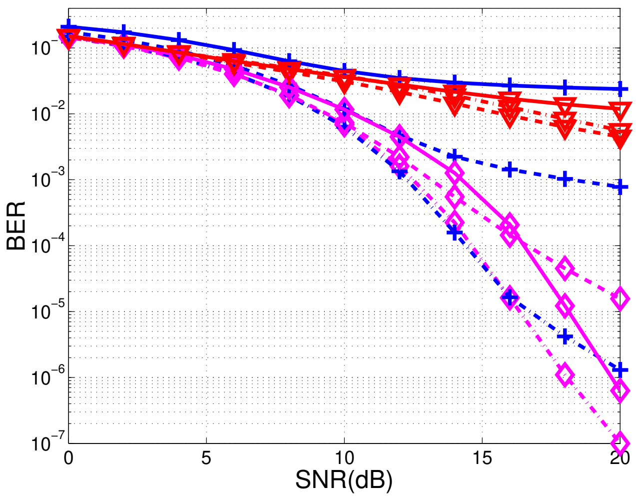

Figure 5: BER for a nonlinear channel with linear MMSE (▽), SVM (+) and GP's (⋄) equalization with $n = 400$ training samples and length $m = 4$ (dash-dotted), $m = 6$ (dashed) and $m = 8$ (solid).

as the linear channel above and nonlinear function

$$f(\hat{x}(t)) = \hat{x}(t) + 0.2\hat{x}^2(t) - 0.1\hat{x}^3(t) \tag{13}$$

Similar remarkable conclusions to those for Figure 2 and 3 can be drawn. The GPs based equalizer clearly outperforms other approaches, even if the channel is nonlinear, closing to the optimal Bayesian solution. Figure 5 depicts the results of the same scenario as in Figure 3 but for this nonlinear channel and $n = 400$ training samples. It can be observed that increasing the length of the equalizer yields an improvement of the BER, when the other methods fail to detect the transmitted symbols. Notice that, although the performance for $m = 8$ first slightly deteriorates, it will later tend to 0 for high SNR improving the curves for $m = 4$ and $m = 6$.

6. CONCLUSIONS

In this paper, we have presented a novel channel equalizer based on Gaussian Processes for regression. 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. Regarding the covariance matrix, in GPs it can be learned by using Maximum Likelihood, adjusting 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 in solving the channel equalization problem for digital communication systems. GPs can easily include a linear kernel to improve convergence and we do not need a pre-estimation of the noise. We have tested the GP-equalizer in a realistic scenario, including linear and nonlinear channels. In these scenarios we have shown that the GP-equalizer is able to provide accurate solutions with very short training sequences, a critical issue in equalization. Besides, in the experiments included, GPs exhibited a better performance than the linear MMSE and SVMs, in which the hyperparameters were specified beforehand. Furthermore, the GP-equalizer takes advantage of a larger input dimension to improve separability, while these approaches clearly fail.

REFERENCES

[1] G. J. Gibson, S. Siu, and C. F. N. Cowan, "The application of nonlinear structures to the reconstruction of binary signals," *IEEE Transaction on Signal Processing*, vol. 39, pp. 1877–1884, 1991.

[2] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Reconstruction of binary signals using an adaptive radial-basis-function equalizer," *Signal Processing*, vol. 22, pp. 77–83, 1991.

[3] J. Cid-Sueiro, A. Artés-Rodríguez, and A. R. Figueiras-Vidal, "Recurrent radial basis function networks for optimal symbol-by-symbol equalization," *Signal Processing*, vol. 40, pp. 53–63, 1994.

[4] T. Kohonen, K. Raivio, O. Simula, O. Venta, and J. Henriksson, "Combining linear equalization and self-organizing adaptation in dynamic discrete-signal detection," in *Proc. International Joint Conf. on Neural Networks*, San Diego, CA, June 1990, vol. 1, pp. 223–228.

[5] P. R. Chang and B. C. Wang, "Adaptive decision feedback equalization for digital satellites channels using multilayer neural networks," *IEEE Journal on Selected areas of communication*, vol. 13, no. 2, pp. 316–324, Feb. 1995.

[6] B. Mitchinson and Harrison R. F, "Digital communications channel equalization using the kernel adaline," *IEEE Trans. Communications*, vol. 50, no. 4, pp. 571–576, 2002.

[7] 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.

[8] Christopher K.I. Williams, "Prediction with gaussian processes: From linear regression to linear prediction and beyond," Tech. Rep., NCRG, Aston Univ., 1997, In M.I. Jordan ed., Learning and Inference in Graphical Models. Kluwer Academic Press. 1998.

[9] S. Caro J. Murillo-Fuentes and F. Perez-Cruz, "Gaussian processes for multiuser detection in CDMA receivers," in *In Advances in Neural Information Processing Systems, NIPS*, Vancouver, CA, Dec. 2005.

[10] F. Perez-Cruz and J. Murillo-Fuentes, "Gaussian processes for digital communications," in *In Proc. of the ICASSP*, Tolousse, Fr, May 2006, p. Accepted.

[11] Carl E. Rasmussen and Christopher K.I. Williams, *Gaussian Processes for Machine Learning*, Adaptive Computation and Machine Learning. The MIT Press, Cambridge, Massachusetts, first edition, 2006.

[12] 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.

[13] B. Mulgrew S. Chen and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–578, 1993.

[14] 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.