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

Linear equalizers underperform in dispersive channels with additive white noise, because optimal decision functions are nonlinear. In this paper we present Gaussian Processes (GPs) for regression as new nonlinear equalizer for digital communication systems. GPs can be cast as nonlinear MMSE, a commom criterion in digital communications. Unlike other nonlinear kernel based methods, such as kernel adaline or support vector machines, the solutions produced by GPs are analytical, and the hyperparameters can be readily learnt by maximum likelihood. Hence, we avoid cross-validation or noise estimation, and improve convergence speed. We present experimental results, over linear and nonlinear channel models, to show that GP-equalizers outperform linear and nonlinear state-of-the-art solutions.

1. INTRODUCTION

Channel equalization is a major issue in digital communications, because the channel affects the transmitted sequence with both linear and nonlinear distortions. Inter-symbol interference, which accounts for the linear distortion, occurs as a consequence of the limited bandwidth of the channel and consists of spreading the received symbol energy through several time intervals. The channel cannot be considered linear due to the presence of nonlinear devices such as amplifiers and converters [1]. Channel equalization minimizes those distortions to recover the transmitted sequence. In wireless communications, in which bandwidth is a scarce resource and we need to send a training sequence in 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) criterion. The optimal solution based on maximum likelihood sequence estimation has a complexity that grows exponentially with the dimension of the channel impulsive response (Viterbi Algorithm), and an unknown delay. Alternatively, neural networks (NNs) can be used to solve this problem at a lower computational cost. Several NNs schemes have been proposed to address this problem with varying degrees of success, such as the multi-layered perceptron (MLP) [2], radial basis functions (RBFs) [3], recurrent RBFs [4], self-organizing feature maps [5], wavelet neural networks [6], Kernel Adaline (KA) [7] and support vector machines (SVMs) [8]. Such structures usually outperform the LTE, especially when non-minimum phase channels are encountered. They can also compensate for nonlinearities in the channel. The major drawback of such schemes is the need for long training sequences to achieve optimal equalization (Bayes error).

In this paper, we present a nonlinear estimation technique known as Gaussian Processes (GPs) for regression [9] as a novel channel equalization tool. This approach has been already successfully applied to the multiuser detection problem in CDMA systems [10, 11]. GPs provide analytical answers to the estimation 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 result, as illustrated in the experiments, in a remarkable reduction in the number of needed training samples, even when increased the order of the equalizer.

2. GAUSSIAN PROCESSES FOR REGRESSION

Gaussian Processes (GPs) for regression [12] 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 [9], 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), $\mu_\mathbf{w} = \mathrm{argmax}_\mathbf{w}\{\log p(\mathbf{y}|\mathbf{X},\mathbf{w}) + \log p(\mathbf{w})\}$. 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 [9]. 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, $\boldsymbol{\phi}(\mathbf{x}^*)$.

2.1 Covariance Matrix

To get the estimation given by a GP model for regression, we only need to specify its covariance function $k(\cdot,\cdot)$. This 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 SVMs, or any other kernel method, see [13] 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 adjusted 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:

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

The hyperparameters α_i and γ_ℓ have to be non-negative to ensure that $k(\mathbf{x}_i,\mathbf{x}_j)$ is a kernel function, i.e., every matrix $\mathbf{C}$ formed by any set of input vectors must be positive semidefinite. 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 (i.e. $\alpha_1 = 0$). 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 [9].

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 $\boldsymbol{\theta} = [\log\alpha_1,\log\alpha_2,\log\alpha_3,\log\gamma_1,\ldots,\log\gamma_d]$ 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. We have used the logarithm of the hyperparamters to deal with an unconstrained optimization problem over $\boldsymbol{\theta}$. 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 train 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 needs long training sequences, limiting the number of hyperparameters that can be adjusted. Besides, it 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

In Figure 1 we depict a simple band-based model to describe a dispersive communication channel, yet typically used [14].

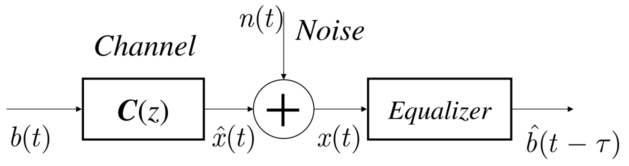

Figure 1: Simple discrete-time transmission channel model.

The transmitted signal $b(t)$ is a BPSK modulated signal, i.e., an independent and equiprobable sequence of symbols with values $\{-1,+1\}$, and $n(t)$ represents additive white Gaussian noise (AWGN). 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. A linear transversal equalizer (LTE) is typically used to invert the channel response for recovering the transmitted signals [15]. The LTE linearly combines the last m samples from the channel to estimate the transmitted bit, i.e.:

$$\hat{b}(t-\tau) = \operatorname{sign}\left(\mathbf{w}^\top \mathbf{x}(t)\right) \tag{8}$$

where $\mathbf{x}(t) = [x(t), x(t-1), \ldots, x(t-m+1)]$ and τ is a delay to ensure that enough information is available in the LTE to estimate the transmitted bit. The weight vector $\mathbf{w}$ is typically computed applying the MMSE criterion [15], as it offers a trade-off between minimising the effects of the dispersive nature of the channel and its noise. Nonlinear channel equalizers transform $\mathbf{x}(t)$ prior to computing the weight vector to obtain solutions that achieve minimum BER.

The linear channel above can be modified by introducing a nonlinear function to model the receiver distortion, i.e.:

$$x(t) = f(\hat{x}(t)) + n(t) \tag{9}$$

where $f(\cdot)$ is the nonlinear function modelling the nonlinearities in the receiver.

4. GAUSSIAN PROCESSES FOR CHANNEL EQUALIZATION

GPs predictions are analytical and can be computed using (3). The GP-equalizer for a dispersive channel decides which was the transmitted bit according to:

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

where $\mathbf{x}^*$ is the input vector to the equalizer. The GP-equalizer is similar to the LTE, as it computes the inner product between the received symbols and a pre-specified vector. Furthermore, when $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$, $\mu_{\mathbf{w}}$ is computed as:

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

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 on 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 GPs for regression will not use a covariance function that yields linear regressors, its decisions can be interpreted as a nonlinear MMSE-equalizer.

The transmitter will need to send a known sequence to the receiver to train the hyperparameters of the covariance function, which are obtained as explained in Section 2.1, and construct the weight vector $\mathbf{C}^{-1}\mathbf{y}$. For each new input sample $\mathbf{x}^*$, we need to compute the kernel vector $\mathbf{k} = [k(\mathbf{x}^*, \mathbf{x}_1), \ldots, k(\mathbf{x}^*, \mathbf{x}_n)]$ and predict the incoming symbol, $\operatorname{sign}(\mathbf{k}^\top \mathbf{C}^{-1}\mathbf{y})$. Hence, $\mathbf{C}^{-1}\mathbf{y}$ is a prespecified vector, given by the training set, and predictions are obtained by its inner product with the kernel vector of the new input with the training set. In Section 2 we mention that GPs provide error bars for our estimates (4). These error bars are obtained assuming we are solving a regression problem with Gaussian noise. But as we are actually solving a detection problem, therefore they are meaningless for our application. Still in other regression problems they provide accurate description of the standard deviation of our predictions. The mean prediction still provides a good estimate of the transmitted bits as the square-loss can be used as a proxy to solve classification problems.

The covariance function in (5) is a good kernel for solving the GP-equalizer, because it contains a linear and a nonlinear part. The optimal decision surface is nonlinear. In minimum-phase channels a linear solution can provide an approximation to the optimal solution, but still it will be suboptimal and a nonlinear part is needed to improve the results obtained by the linear equalizer. 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 symbols form a constellation of clouds of points with Gaussian spread around its centres.

GPs provide a great advantage compare to other nonlinear tools for solving the channel equalization problem. SVMs, KA or NNs use the training sequence to adjust the weights in a predefined nonlinear structure. GP uses the training sequence to search for the best nonlinear structure for each instantiation of the problem, because the set of weights can be computed analytically. SVMs or KA need to fix the width of the kernel a priori, as the only way to search for its optimum is by cross-validation means, while GPs find this optimal width parameter. MLP or RBFN have to specify their structure a priori (number of layers, neurons per layer, etc.), while GPs find their structure by maximum likelihood. This advantages will allow GP-equalizer to outperform these nonlinear tools to solve the channel equalization problem.

KA uses the same criterion to train the equalizer as GP does (least squares). But while KA trains the equalizer with a fix kernel using early stopping [7], GPs solve the equalizer analytically and trains for the best kernel parameters. Hence using the same criterion is able to train the equalizer much more accurately. Finally, we believe GP is the natural way to extend the MMSE criterion for nonlinear estimations. If the nonlinear structure is known its solution is straight forward, as in the linear case. Furthermore, its structure is learnt from the data, so optimal results are expected if a versatile parametric form for the covariance matrix can be described for each problem at hand.

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 fixed-length equalizer and then for different equalizer lengths at a fixed-number of training samples. 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 [16], and its C parameter was chosen to minimise the BER. The optimum Bayesian equalizer will be also included as reference.

As linear channel we used

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

as proposed in [7] to model radiocommunication channels. 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, respectively, set to 4 and 1. 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 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. Notice that, being a linear channel, a higher equalizer length helps designing good linear equalizers. Thanks to the linear part of the GP kernel, we reach these solutions with just a few training samples.

We have repeated the previous experiments adding a nonlinearity to the receiver to show that GP-equalizers also deal with nonlinearities in the channel model. We have used the nonlinear model in [7] that models typical nonlinearities in digital communication receivers:

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

Similar remarkable conclusions to those for Figure 2 and 3 can be drawn. The GP-equalizer clearly outperforms the 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. Since in this experiment the channel is nonlinear, GPs do not clearly improve with the length of the equalizer. Besides, we need a large enough number of training samples to tune the non-linear part of the GP kernel. Anyway, the GPs again present quite better results than the other equalizers.

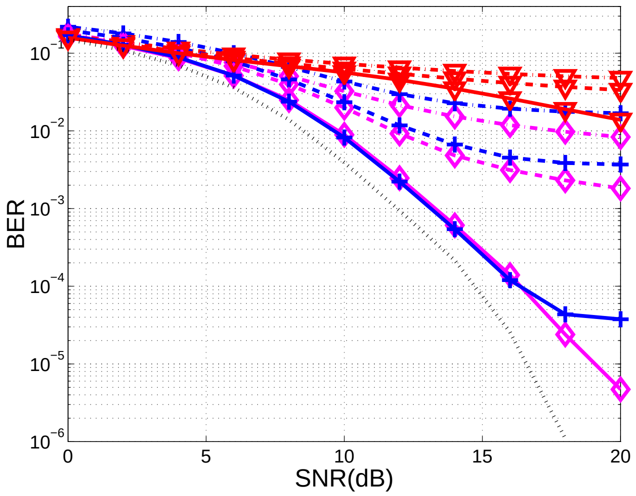

Figure 2: BER for a linear channel with optimal Bayesian (dotted), linear MMSE ($\triangledown$), SVM ($+$) and GP's ($\diamond$) equalization with $m = 4$ and $\tau = 1$, trained with 50 (dash-dotted), 100 (dashed) and 400 (solid) training samples.

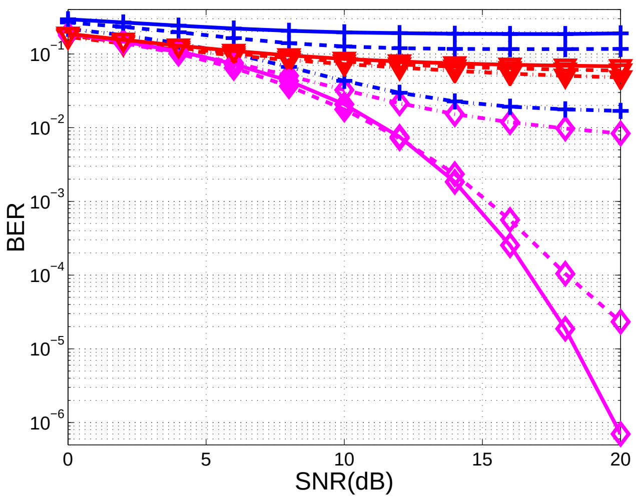

Figure 3: BER for a linear channel with linear MMSE ($\triangledown$), SVM ($+$) and GP's ($\diamond$) equalization with $n = 50$ training samples and length $m = 4$ (dash-dotted), $m = 6$ (dashed) and $m = 8$ (solid).

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 solution given by the GPs is analytical, given its covariance matrix. Furthermore the covariance matrix in GPs can be learnt by maximum likelihood. These characteristics differentiate them with respect to other nonlinear tools as SVMs or Neural Nets, in which an optimization step is needed to obtain weight vector, and its hyperparameters/structure will have to be prespecified.

We have shown that this framework is very useful for solving the channel equalization problem in digital communication systems. GPs covariance matrix can include a linear term to improve convergence and we do not need a pre-

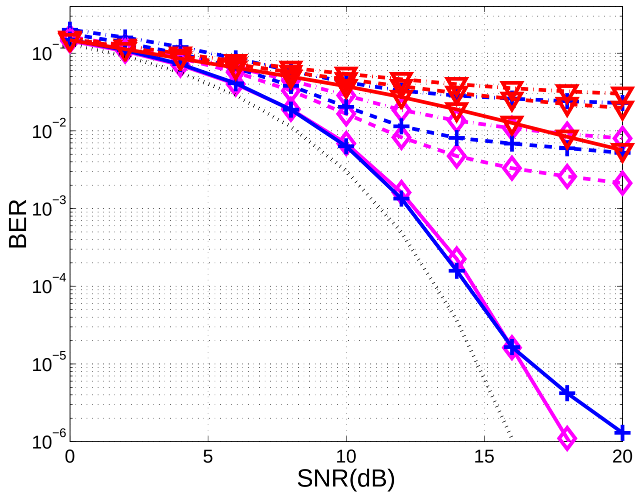

Figure 4: BER for a nonlinear channel with optimal Bayesian (dotted), linear MMSE ($\triangledown$), SVM ($+$) and GP's ($\diamond$) equalization with $m = 4$ and $\tau = 1$, trained with 50 (dash-dotted), 100 (dashed) and 400 (solid) training samples.

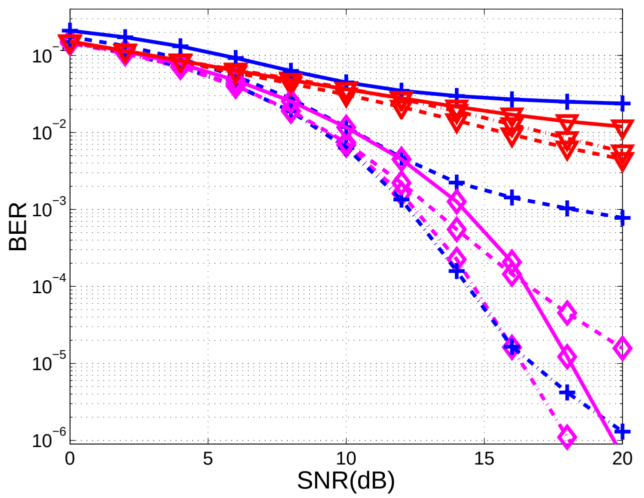

Figure 5: BER for a nonlinear channel with linear MMSE ($\triangledown$), SVM ($+$) and GP's ($\diamond$) equalization with $n = 400$ training samples and length $m = 4$ (dash-dotted), $m = 6$ (dashed) and $m = 8$ (solid).

estimation of the noise in the channel. 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] R. Parisi, E. D. Di Caudio, G. Orlandi, and B. D. Rao, "Fast adaptive digital equalization by recurent neural networks," *IEEE Trans on Signal Processing*, vol. 45, no. 11, pp. 2731–2739, Nov. 1997.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] C.K.I. Williams, "Prediction with gaussian processes: From linear regression to linear prediction and beyond," In M.I. Jordan ed., Learning and Inference in Graphical Models. Kluwer Academic Press. 1998.

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

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

[12] C.E. Rasmussen and C.K.I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[13] 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.

[14] 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.

[15] J.G. Proakis, *Digital Communications*, McGraw-Hill, third edition, 1995.

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