GAUSSIAN PROCESSES FOR REGRESSION IN CHANNEL EQUALIZATION

Sebastián Caro\textsuperscript{a}\textsuperscript{†}, Fernando Pérez-Cruz\textsuperscript{b}* and Juan José Murillo-Fuentes\textsuperscript{a}\textsuperscript{†},

\textsuperscript{a} Depto. Teoría de la Señal y Comunicaciones, Univ. de Sevilla.
Paseo de los Descubrimientos s/n, 41092 Sevilla, Spain.
\{murillo,scaro\}@us.es

\textsuperscript{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 common 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. Unlike other nonlinear kernel based methods, such as kernels, 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.

\footnote{Supported through the Spanish MCYT Agency (TIC2003-02602).} \footnote{Supported through the Spanish MCYT Agency (TIC2003-03781).}
the weight vector \( \mathbf{w} \) using Bayes theorem:

\[
\begin{align*}
    p(\mathbf{w}|D) &= \frac{p(\mathbf{y}|\mathbf{X}, \mathbf{w})p(\mathbf{w})}{p(\mathbf{y}|\mathbf{X})} \\
    &= \frac{1}{p(\mathbf{y}|\mathbf{X})} \prod_{i=1}^{n} \exp \left( -\frac{(y_i - \mathbf{w}^T \phi(x_i))^2}{2\sigma_w^2} \right) \exp \left( -\frac{||\mathbf{w}||^2}{2\sigma_w^2} \right) \\
    &= \mathcal{N}(\mathbf{w}; \mu_w, \Sigma_w)
\end{align*}
\]

where \( \mu_w = \Sigma_w \Phi / \sigma_w^2, \Sigma_w^{-1} = \Phi^T \Phi / \sigma_w^2 + 1/\sigma_w^2 \), \( y = [y_1, \ldots, y_n]^T \), \( \Phi = [\phi(x_1), \ldots, \phi(x_n)]^T \) and \( \mathbf{X} = [x_1, \ldots, x_n]^T \). Actually, the mean of the posterior can be computed as the maximum a posteriori (MAP) of (1), \( \mu_w = \arg\max_{\mathbf{w}} \{ \log p(\mathbf{y}|\mathbf{X}, \mathbf{w}) + \log p(\mathbf{w}) \} \). The prediction for \( y^* \) is obtained integrating out the posterior over \( \mathbf{w} \) times its likelihood:

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

where

\[
    \begin{align*}
    &\mu_{y^*} = \phi^T(\mathbf{x}^*) \mu_w = k^T \mathbf{C}^{-1} y \\
    &\sigma_{y^*}^2 = \phi^T(\mathbf{x}^*) \Sigma_w \phi(\mathbf{x}^*) = k^T \mathbf{C}^{-1} k
\end{align*}
\]

being \( k(x_i, x_j) = \phi^T(x_i) \phi(x_j) \), \( \mathbf{C}_{ij} = k(x_i, x_j) + \sigma^2 \delta_{ij} \) and \( k = [k(x_1, x_1), \ldots, k(x_n, x_n)] \). 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 \( \mathbf{w}, \mu_w \), and the input vector, \( \phi(\mathbf{x}^*) \).

### 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 \( \phi(x_i) \) 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 \( k(x_i, x_j) = x_i^T x_j \) (notice that \( \phi(x) = x \)). We then need to specify the value of \( \sigma_w^2/\sigma_w^2 \) 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 \( \{x_i\}_{i=1}^n \), as it represents the covariance matrix of a multidimensional Gaussian distribution. A versatile covariance function, typically used in the literature, is described as follows:

\[
    k(x_i, x_j) = \alpha_1 \exp \left( -\sum_{\ell=1}^{d} \gamma_\ell (x_{i\ell} - x_{j\ell})^2 \right) + \alpha_2 x_i^T x_j + \alpha_3 \delta_{ij}
\]

The hyperparameters \( \alpha_1 \) and \( \gamma_\ell \) have to be non-negative to ensure that \( k(x_i, x_j) \) is a kernel function, i.e., every matrix \( \mathbf{C} \) formed by any set of input vectors must be positive semi-definite. 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 \( \alpha_2 \gamma_\ell \) 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 \( \gamma_\ell \) 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 hyperparameters is defined as:

\[
    p(\mathbf{y}|\mathbf{x}, \theta) = \frac{1}{\sqrt{(2\pi \mathbf{C}_\theta)^{d}}} \exp \left( -\frac{1}{2} \mathbf{y}^T \mathbf{C}_\theta^{-1} \mathbf{y} \right)
\]

where \( \theta = [\log \alpha_1, \log \gamma_2, \log \alpha_3, \log \gamma_1, \ldots, \log \gamma_d] \) represents the hyperparameters of the covariance function and we have added the subscript \( \theta \) to \( \mathbf{C} \) to explicitly indicate that the covariance matrix depends on the hyperparameters. We have used the logarithm of the hyperparameters to deal with an unconstrained optimization problem over \( \theta \). 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 hand-based model to describe a dispersive communication channel, yet typically used [14].
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:

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

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 typically computes 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 $x(t)$ prior to computing the weight vector to obtain solutions that achieve minimum BER.

Figure 1: Simple discrete-time transmission channel model.

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

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

(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:

$$\hat{b}(t-\tau) = \text{sign}(\mu_{\nu_s}) = \text{sign}(\phi^\top(x^*)\mu_{\nu_w}) = \text{sign}(k^\top C^{-1} y)$$

(10)

where $x^*$ 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 $\phi(x) = x$, $\mu_{\nu_w}$ is computed as:

$$\mu_{\nu_w} = \arg \min_w \left\{ -\log p(b|X, w) - \log p(w) \right\}$$

$$= \arg \min_w \left\{ \frac{1}{2\sigma_w^2} \sum_{i=1}^{n} (b_i - w^\top x_i)^2 + \frac{||w||^2}{2\sigma_w^2} \right\}$$

(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 $C^{-1} y$. For each new input sample $x^*$, we need to compute the kernel vector $k = [k(x^*, x_1), \ldots, k(x^*, x_n)]$ and predict the incoming symbol, $\text{sign}(k^\top C^{-1} y)$. Hence, $C^{-1} y$ 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 compared 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 para-
metric 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 chan-
nel 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 dif-
ferent equalizer lengths at a fixed-number of training sam-
pies. 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
normalized signal to noise ratio (SNR) for the linear MMSE,
ples. In all cases we depict the bit error rate (BER) along the
different equalizer lengths at a fixed-number of training sam-
les. In each problem at hand.

As linear channel we used

\[ C(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2} \]  

(12)
as proposed in [7] to model radiocommunication channels. In
Figure 2 we depict the BER for the linear MMSE (∇), SVM (+) and GPs (⋄) 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 re-
results for 100 independent experiments and 10^3 test samples
in each run. GP-equalizer provides remarkably the best re-
sults, 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. Be-
sides, 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 equali-
izers. 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 non-
linearity 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:

\[ f(\hat{x}(t)) = \hat{x}(t) + 0.2\hat{x}^2(t) - 0.1\hat{x}^3(t) \]  

(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 chan-
nel 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 re-
results than the other equalizers.

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 likeli-
hood. 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 commu-
nication systems. GPs covariance matrix can include a lin-
ear term to improve convergence and we do not need a pre-
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