Max–Planck–Institut für biologische Kybernetik
Max Planck Institute for Biological Cybernetics

Technical Report No. TR-148

Implicit Wiener Series

Part II: Regularised Estimation

P. V. Gehler[1] and M. O. Franz[1]

August 2006

[1] Department for Empirical Inference, email: pgehler;mof@tuebingen.mpg.de

Implicit Wiener Series

Part II: Regularised Estimation

P. V. Gehler and M. O. Franz

Abstract. Classical Volterra and Wiener theory of nonlinear systems does not address the problem of noisy measurements in system identification. This issue is treated in the present part of the report. We first show how to incorporate the implicit estimation technique for Volterra and Wiener series described in Part I into the framework of *regularised estimation* without giving up the orthogonality properties of the Wiener operators. We then proceed to a more general treatment of polynomial estimators (Volterra and Wiener models are two special cases) in the context of Gaussian processes. The implicit estimation technique from Part I can be interpreted as Gaussian process regression using a polynomial covariance function. Polynomial covariance functions, however, have some unfavorable properties which make them inferior to other, more localised covariance functions in terms of generalisation error. We propose to remedy this problem by approximating a covariance function with more favorable properties at a finite set of input points. Our experiments show that this additional degree of freedom can lead to improved performance in polynomial regression.

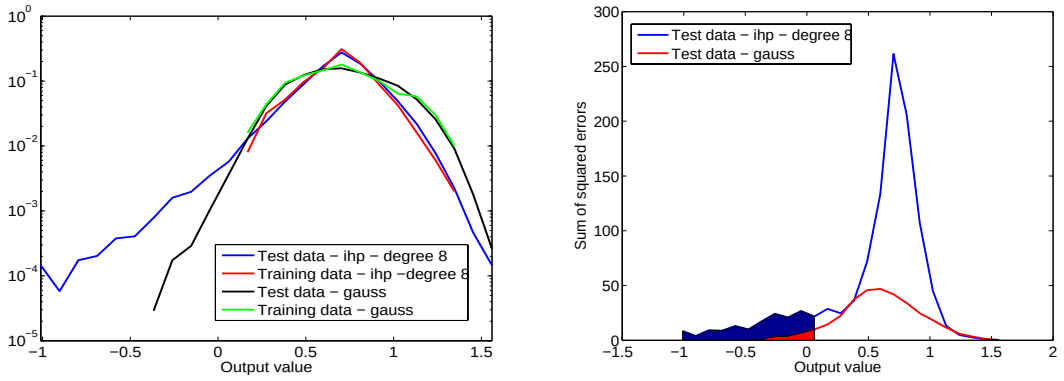
---

1 Introduction

In its classical formulation, the estimation of the Wiener series assumes noise-free measurements of the system outputs during system identification. This assumption was also adopted in the kernel-based implicit estimation procedure described in Part I of this report. For real, noise-contaminated data, the estimated Wiener series will model both signal and noise of the training data which results in reduced prediction performance on independent test sets. But even in the ideal case of noise-free signals, Volterra and Wiener models typically show a reduced generalisation performance as compared to other types of nonlinear models. Roughly speaking, this is due to the *divergence property* of polynomial models: for test inputs outside the range of the training data, polynomial models tend to assign output values well beyond the range of the training outputs. This results in a high sensitivity against outliers in the test data which in turn leads to a reduced prediction performance. An illustrative example of the divergence property is shown in Fig. 1. Here, we trained two models on 350 examples from the KIN40K dataset (Schwaighofer & Tresp, 2003), using either the inhomogeneous polynomial kernel $(1 + \mathbf{x}_1^\top \mathbf{x}_2)^p$ or the Gaussian kernel $\exp(-\|\mathbf{x_1} - \mathbf{x_2}\|^2)$. Fig. 1$a$. shows the histograms of the model outputs on the training set and independent test data. As expected, we find that the polynomial model produces output values on the test set that lie significantly beyond the range of the training outputs. Although these values constitute only a small fraction of the entire range, their contribution to the overall prediction error is disproportionately large (shaded regions in Fig. 1b.). Indeed, we find that the prediction performance of the polynomial model is considerable worse than that of the Gaussian kernel (cf. Sect. 4).

In the second part of this report, we address the generalisation problems of Volterra and Wiener system models using two closely related frameworks: regularisation and Gaussian processes. Regularisation is the standard approach in machine learning to address the generalisation problem. The basic idea is to restrict the possible solutions in a suitable manner that reflects the prior knowledge of the experimenter about the different characteristics of the true signal and the corrupting noise. For instance, noise often affects more the high-frequency components of a signal than its low-frequency components. This knowledge can be incorporated into an estimation problem by restricting the possible solutions to smooth functions. The learning procedure is thus prevented from modelling noise-corrupted fine details of the system output which typically results in an improved generalisation performance. In a similar train of thought, we can try to restrict our solutions not only to be smooth, but also to remain non-divergent in the input domains that are relevant to the problem at hand.

One approach to regularisation is to augment the mean squared error (MSE) objective function we used in Part I for estimating Volterra and Wiener models ($\mathbf{x}_i \in \mathbb{R}^m, i = 1 \dots N$ are the training inputs, y_i the corresponding



a. *b.*

Figure 1: Divergence property of polynomial models: *a.* logarithmic plot of the normalised output histograms of an eighth-order inhomogeneous polynomial (ihp) and a Gaussian model trained on 350 examples from the KIN40K dataset (39650 test examples, cf. Sect. 4); *b.* Sum of squared error per histogram bin for the polynomial and the Gaussian kernel. The shaded regions denote output values outside the range of the training data.

scalar training outputs, and $f(\mathbf{x}_i)$ the model outputs)

$$c((\mathbf{x}_1, y_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_N, y_N, f(\mathbf{x}_N))) = \frac{1}{N} \sum_{j=1}^{N} (f(\mathbf{x}_j) - y_j)^2 \tag{1}$$

with a functional $\Omega(f)$ that penalizes unwanted properties of the solutions such as overly fine detail or divergent behaviour. Choosing

$$c((\mathbf{x}_1, y_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_N, y_N, f(\mathbf{x}_N))) = \frac{1}{N} \sum_{j=1}^{N} (f(\mathbf{x}_j) - y_j)^2 + \Omega(\|f\|_{\mathbb{F}}). \tag{2}$$

(Ω is a nondecreasing function on $\mathbb{R}_+$ and $\|\cdot\|_{\mathbb{F}}$ is the norm in the function space used for the regression), and remembering from Part I that the polynomials constitute a reproducing kernel Hilbert space (RKHS), we are allowed the apply again the representer theorem which means that also the regularised solution can be expressed as a linear combination of polynomial kernel functions evaluated at the training points:

$$f(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}, \mathbf{x}_i), \quad \alpha_i \in \mathbb{R}. \tag{3}$$

$\Omega(f)$ is often given as a quadratic function of the weights α_i of the kernel expansion and can thus be written as

$$\Omega_R = \lambda\, \alpha^\top R \alpha, \quad \lambda > 0 \tag{4}$$

with a positive semidefinite matrix R. λ controls the tradeoff between the fidelity to the data and the penalty term. R is chosen to enforce the desired characteristics of the solutions. For instance, when choosing $R = I_N$ as in ridge regression, the RKHS norm of the solutions remain small which leads to smoother, less noise-sensitive solutions. Alternatively, a suitable choice of a rank-deficient R can selectively penalize noise-contaminated subspaces of the signal (Nowak, 1998).

When the penalising functional Ω_R is used in addition to the MSE on the training set, we obtain the implicit Volterra and Wiener series expansions $\sum_{n=0}^{p} H_n(\mathbf{x})$ and $\sum_{n=0}^{p} G_n(\mathbf{x})$

$$\sum_{n=0}^{p} G_n(\mathbf{x}) = \sum_{n=0}^{p} H_n(\mathbf{x}) = \mathbf{y}^\top (K_p + \lambda R)^{-1}\, \mathbf{k}(\mathbf{x}) \tag{5}$$

instead of

$$\sum_{n=0}^{p} G_n(\mathbf{x}) = \sum_{n=0}^{p} H_n(\mathbf{x}) = \mathbf{y}^\top K_p^{-1}\, \mathbf{k}(\mathbf{x}) \tag{6}$$

as before. Here, K_p denotes the Gram matrix computed with one of the polynomial kernels[1] $\sum_{n=0}^{p} a_n^2 (\mathbf{x}_1^\top \mathbf{x}_2)^n$, $(1 + \mathbf{x}_1^\top \mathbf{x}_2)^p$ or $\sum_{n=0}^{p} (\mathbf{x}_1^\top \mathbf{x}_2)^n$, $\mathbf{y} = (y_1, y_2, \ldots, y_N)^\top$ and $\mathbf{k}(\mathbf{x})$ is the coefficient vector

[1] In Part I, we derived the implicit representation of the implicit Volterra series only for the last kernel. Analogous derivations can be found for the other polynomial kernels (cf. Franz & Schölkopf, 2006)

$(k(\mathbf{x}, \mathbf{x}_1), k(\mathbf{x}, \mathbf{x}_2), \ldots, k(\mathbf{x}, \mathbf{x}_N))^\top \in \mathbb{R}^N$. As we said above, we obtain a less noise-sensitive solution when we choose $R = I_N$ which was our first motivation for using the regularisation framework. This choice, however, does not address our second problem of preventing the obtained Volterra and Wiener models from diverging in test input domains that are far from the training inputs. In the following sections, we show how to attain both objectives simultaneously using *Gaussian processes* (O'Hagan, 1978, overview in Rasmussen and Williams, 2006).

Before doing so, we have to clarify a further issue: if one is interested in single Wiener operators, the regularised estimation has a decisive disadvantage, namely that the Wiener operators computed according to (see Part I)

$$\begin{aligned} G_n(\mathbf{x}) &= \sum\nolimits_{i=0}^{n} G_i(\mathbf{x}) - \sum\nolimits_{i=0}^{n-1} G_i(\mathbf{x}) \\ &= \mathbf{y}^\top \left[K_n^{-1}\, \mathbf{k}^{(n)}(\mathbf{x}) - K_{n-1}^{-1}\, \mathbf{k}^{(n-1)}(\mathbf{x}) \right]. \end{aligned} \tag{7}$$

are no more orthogonal with respect to the input. However, orthogonality can be still enforced by considering the (smoothed) output of the regularised Wiener system on the training set

$$\tilde{y}^\top = \mathbf{y}^\top (K_p + \lambda R)^{-1}\, K \tag{8}$$

as a modified, "noise-corrected" training set. The regularised Wiener operators are then given as

$$G_n(\mathbf{x}) = \tilde{y}^\top \left[K_n^{-1}\, \mathbf{k}^{(n)}(\mathbf{x}) - K_{n-1}^{-1}\, \mathbf{k}^{(n-1)}(\mathbf{x}) \right] = \mathbf{y}^\top (K_p + \lambda R)^{-1}\, K \left[K_n^{-1}\, \mathbf{k}^{(n)}(\mathbf{x}) - K_{n-1}^{-1}\, \mathbf{k}^{(n-1)}(\mathbf{x}) \right] \tag{9}$$

constituting an orthogonal decomposition of the regularised solution over the training set.

The remainder of this text is organised as follows: we start in Sect. 2 by introducing the probabilistic framework of Gaussian processes and clarifying its relation to the regularisation approach described above. We will see that in Gaussian process point of view, the matrix R determines a prior distribution over the function space $\mathbb{F}$. In Sect. 3, we show how to choose this prior such that the solutions have (at least approximately) the properties of any other Gaussian process which can be prescribed by the experimenter. In particular, the desired Gaussian process can be chosen such that the solutions are smooth and non-divergent in at least some region of the feature space. In Sect. 4, we show in a number of experiments that this particular type of regularisation is indeed capable of remedying the above-mentioned estimation problems of the Volterra and Wiener approach to nonlinear system identification. As usual, we conclude with a discussion in Sect. 5.

2 Gaussian process Regression

In the standard regression setting, our task is to infer a functional relationship $f(\mathbf{x})$ from a set of observations $\{(\mathbf{x}_i, y_i) \,|\, i = 1, \ldots, N\}$ which can be used to predict the output $f(\mathbf{x}_*)$ on a test input $\mathbf{x}_*$. Formally, a Gaussian process is defined as a collection of random variables, any finite subset of which has a joint Gaussian distribution. In our context, the random variables are the possible outcomes $f(\mathbf{x})$ of the regression problem, with a continuous index $\mathbf{x}$. A Gaussian process is completely specified by its mean function $m(\mathbf{x})$[2] and covariance function $k(\mathbf{x}, \mathbf{x}')$

$$m(\mathbf{x}) = E[f(\mathbf{x})] \tag{10}$$

$$k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]. \tag{11}$$

A sample drawn from the Gaussian process for all possible values of $\mathbf{x}$ defines a function $f(\mathbf{x})$ over $\mathbf{x}$. Thus, the specification of a covariance function implies a distribution over functions. In Gaussian process regression, this distribution is taken as a prior over the possible solutions. Since this prior can be chosen such as to limit the possible outcomes of the regression procedure, an appropriate choice of the covariance is equivalent to regularisation. The predictive distribution for the test output is then found by applying the standard rules of probability theory to obtain the posterior probability distribution of $f(\mathbf{x}_*)$ at the test input $\mathbf{x}_*$, given the training set. From the posterior distribution, we can derive point estimates for the prediction by taking the mode or the mean of the posterior which both coincide in Gaussian processes.

Assuming that the observations are corrupted by additive i. i. d. Gaussian noise ϵ with zero mean and variance σ_n^2 such that $y = f(\mathbf{x}) + \epsilon$, the covariance of the noisy observations becomes

$$E[y_i y_j] = k(\mathbf{x}_i, \mathbf{x}_j) + \sigma_n^2 \delta_{ij}, \tag{12}$$

[2] Without loss of generality, $m(\mathbf{x})$ is usually assumed to be zero for notational convenience.

where δ_{ij} is the Kronecker delta. Using the Gram matrix $(K)_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, we can write the covariance matrix of $\mathbf{y}$ as

$$E[\mathbf{y}\mathbf{y}^\top] = K + \sigma_n^2 I. \tag{13}$$

From the definition of Gaussian processes, we know that any finite subset of observations will have a Gaussian distribution. In particular, the joint distribution of the N observations and the function value at the test location $\mathbf{x}_*$ can be described by a $(N+1) \times (N+1)$ covariance matrix and mean $\mathbf{0}$. We write the joint covariance in a partitioned form such that the joint distribution becomes

$$\begin{pmatrix} \mathbf{y} \\ f(\mathbf{x}_*) \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} K + \sigma_n^2 I & \mathbf{k}_* \\ \mathbf{k}_* & k_{**} \end{pmatrix}\right) \tag{14}$$

with the partitions being $E[\mathbf{y}\mathbf{y}^\top]$, the $N \times 1$ cross-covariance vector $(\mathbf{k}_*)_i = k(\mathbf{x}_i, \mathbf{x}_*)$, and the scalar prior covariance of the test output $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$. The predictive distribution of $f(\mathbf{x}_*)$ is obtained by conditioning the joint distribution of Eq. (14) on the observations. The result is again a Gaussian distribution with mean $\bar{f}_*$ and covariance $\sigma^2_{\bar{f}_*}$ (see, e.g., Rasmussen & Williams, 2006)

$$\bar{f}_* = \mathbf{k}_*^\top (K + \sigma_n^2 I)^{-1} \mathbf{y} \tag{15}$$

$$\sigma^2_{\bar{f}_*} = k_{**} - \mathbf{k}_*^\top (K + \sigma_n^2 I)^{-1} \mathbf{k}_*. \tag{16}$$

Note that – as before in the case of kernel regression – the predictive mean is a linear combination of the N kernel functions in $\mathbf{k}_*$ with weights $\alpha = (K + \sigma_n^2 I)^{-1}\mathbf{y}$. These weights are exactly the same as those obtained for the regularised Volterra series of Eq. (5) if we take one of the polynomial kernels as covariance function and the regulariser $R = I_N$. In other words, Gaussian process regression can be used as an alternative regression technique for estimating implicit Volterra and Wiener series. The interpretation of the polynomial kernel as a covariance function leads us also to an alternative explanation of the unfavorable generalisation properties of polynomial regression: a polynomial covariance function implies a high covariance for distant inputs. In most real-world problems we have the reverse situation, i.e. nearby inputs typically result in similar outputs. Therefore, in the Gaussian process view, a polynomial covariance implies a prior over the function space which favors functions not suited for most real-world problems.

The choice of the covariance function specifies both the space of functions that can be generated by the Gaussian process, and a probability measure on that space. It also determines the function basis in which the regression solutions are expressed: as before in the case of kernel regression, all solutions are linear combinations of the covariance function $k(\cdot, \mathbf{x}_i)$ evaluated for the training inputs. Consequently, the choice of the basis and the probability space are *tightly coupled* in the standard Gaussian process formulation. In our case, however, we would like to have the freedom of choosing the covariance function independently of the function basis, i.e., we would like to express our solution in a basis of polynomial kernel functions, but with a different prior covariance function $k_{\mathcal{GP}}(\cdot, \mathbf{x}_i)$ for the entire Gaussian process such that the regression solutions have the desired properties of smoothness and non-divergence.

A handle on how to decouple the basis and the covariance can be found by looking at an alternative derivation of Gaussian process regression, the so-called *weight-space view* on Gaussian processes [3]. Here, one assumes that $f(\mathbf{x})$ can be represented as weighted sum $f(\mathbf{x}) = \phi(\mathbf{x})^\top \mathbf{w}$ of m basis functions $\phi(\mathbf{x}) = (\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \ldots, \varphi_m(\mathbf{x}))^\top$, where the weights $\mathbf{w}$ from $\mathbb{R}^m$ are distributed according to $\mathcal{N}(0, \Sigma_w)$. This fits nicely into our framework: we always have a finite basis in which we express our solutions consisting either of monomials in explicit Wiener series or polynomial kernels in implicit Wiener series. The basis together with Gaussian prior on the weights again defines a distribution over functions with covariance function

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \Sigma_w \phi(\mathbf{x}_j) \tag{17}$$

and mean function

$$m(\mathbf{x}) = \phi(\mathbf{x})^\top E[\mathbf{w}] = 0. \tag{18}$$

When we substitute this covariance function into (15) and (16), we obtain

$$\bar{f}_* = \phi(\mathbf{x}_*)^\top \Sigma_w \Phi^\top (\Phi \Sigma_w \Phi^\top + \sigma_n^2 I)^{-1} \mathbf{y} \tag{19}$$

$$\sigma^2_{\bar{f}_*} = \phi(\mathbf{x}_*)^\top \Sigma_w \phi(\mathbf{x}_*) - \phi(\mathbf{x}_*)^\top \Sigma_w \Phi^\top (\Phi \Sigma_w \Phi^\top + \sigma_n^2 I)^{-1} \Phi \Sigma_w \phi(\mathbf{x}_*) \tag{20}$$

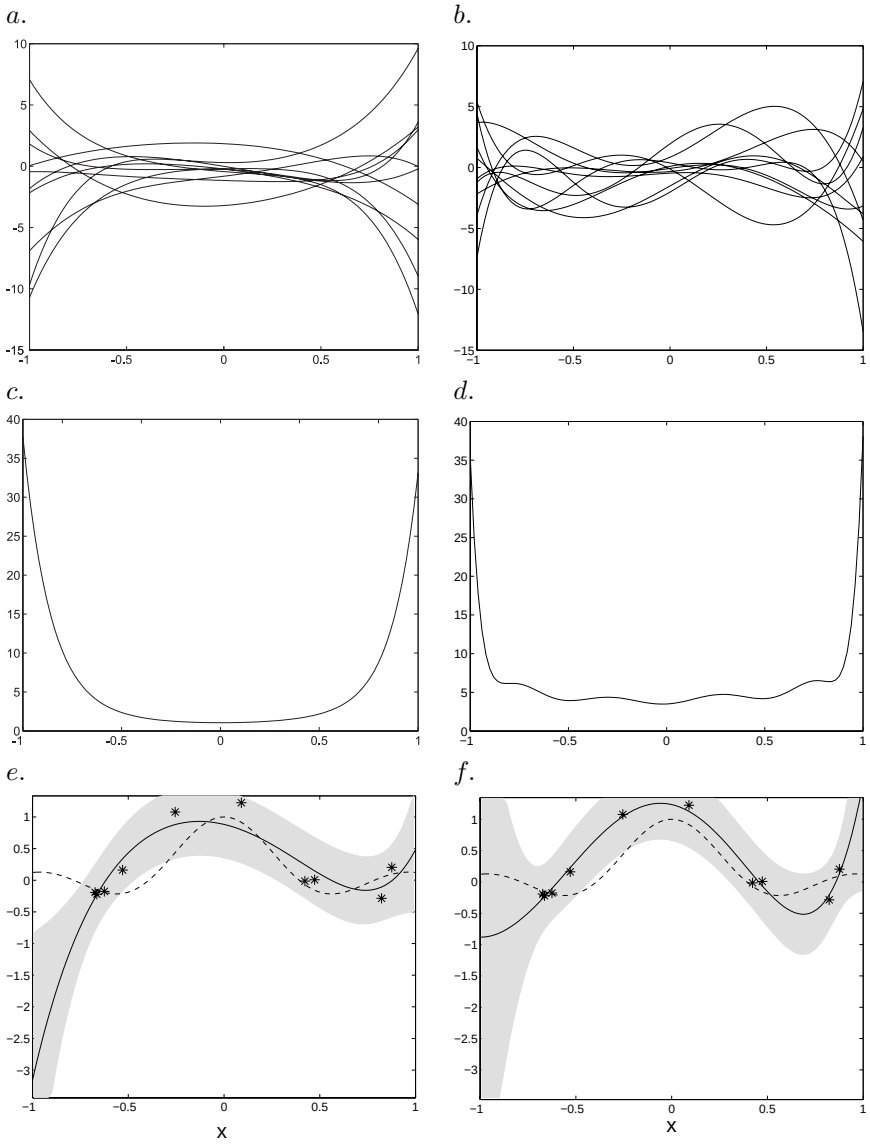


Figure 2: Toy experiment using the first 6 canonical polynomial as basis: a. 10 examples drawn from an isotropic prior in weight space and b. generated by a non-isotropic prior; c and d. Mean squared value in the interval $[-1, 1]$ of 1000 examples drawn from both distributions; e and f. Regression on 10 training samples (stars) for both Gaussian processes. The dashed line denotes the true function, the solid line the prediction from regression. The shaded areas show the 95%-confidence intervals.

where $\Phi = (\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \ldots, \phi(\mathbf{x}_n))$.

In the weight-space view, the implications of choosing a polynomial covariance can be seen immediately by sampling from the Gaussian process prior. Consider a simple toy example where the basis function set consists of the first six canonical polynomials $\phi_1(x) = 1, \phi_2(x) = x, \phi_3(x) = x^2, \ldots, \phi_6(x) = x^5$. Let us assume that the weights are distributed according to an isotropic Gaussian, i.e., $\mathbf{w} \sim \mathcal{N}(0, \sigma^2 I_6)$. In our first experiment, we draw samples from this distribution (Fig. 2a) and compute the mean square of $f(\mathbf{x})$ at all $x \in [-1, 1]$ for 1000 functions generated by the dictionary (Fig. 2b). It is immediately evident that our prior narrowly constrains the possible solutions around the origin while admitting a broad variety near the interval boundaries. If we do regression with this dictionary, the solutions tend to have a similar behaviour as long as they are not enough constrained by the data points (see the diverging solution at the left interval boundary in Fig. 2c. The function to be regressed is of

[3]The above derivation represents the so-called *function-space view*.

the form $t_i = \sin(ax_i)/(ax_i) + n_i$ with an additive Gaussian noise signal $n_i \sim \mathcal{N}(0, \sigma_\nu^2)$). This can lead to bad predictions in sparsely populated areas.

If we choose a non-isotropic prior on the weights instead of an isotropic one (in our example we force the overall covariance to be flat), we observe a different behaviour although we did not change the polynomial basis: the functions sampled from the prior show a richer structure (Fig. $2d$) with a relatively flat mean square value over the interval $[-1, 1]$ (Fig. $2e$). As a consequence, the predicted mean of the Gaussian process (solid line) usually shows a smaller tendency to diverge in the sparsely populated regions near the interval boundaries (see the left interval boundary region in Fig. $2f$). For both priors, the predicted variances (the shaded areas depict the 2σ-interval) become large in unpopulated regions. In the case of the isotropic prior, however, the true function values are still outside the predicted 2σ-interval.

3 Decoupling the covariance and basis

The two alternative views of Gaussian processes give us a handle on how to decouple the covariance and basis: we have to construct a suitable basis $\phi(\mathbf{x})$ from the kernel functions $k(\cdot, \mathbf{x}_i)$ along with a weight covariance Σ_w such that their covariance function $\phi(\mathbf{x}_i)^\top \Sigma_w \phi(\mathbf{x}_j)$ assumes the desired form $k_{\mathcal{GP}}(\mathbf{x}_i, \mathbf{x}_j)$. Of course, we cannot hope to approximate $k_{\mathcal{GP}}(\mathbf{x}_i, \mathbf{x}_j)$ at all possible input pairs, but a closer look at Eqns. (15) - (20) reveals that the covariance function is only evaluated at either training or test inputs. Consequently, we have to approximate our desired covariance only at a finite set of input points.

An obvious choice for $\phi(\mathbf{x})$ would be the polynomial kernel functions themselves (the *empirical kernel map*, see Schölkopf & Smola, 2002), but here we consider only the *Kernel PCA Map* (Schölkopf & Smola, 2002)

$$\phi(x) = K^{-\frac{1}{2}}(k(\mathbf{x}, \mathbf{x}_1), k(\mathbf{x}, \mathbf{x}_2), \ldots, k(\mathbf{x}, \mathbf{x}_n))^\top \tag{21}$$

which usually leads to a better conditioned regression problems in other contexts. Having specified our basis, we have to find a suitable Σ_w to approximate $k_{\mathcal{GP}}(\mathbf{x}_i, \mathbf{x}_j)$ on a finite set $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ of input points. Formally, we have a set of p^2 linear equations

$$k_{\mathcal{GP}}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \Sigma_w \phi(\mathbf{x}_j) \quad \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{S} \tag{22}$$

which, in general, cannot be solved exactly. An approximate solution is given by

$$\Sigma_w = ([\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_p)]^\top)^+ K_{\mathcal{GP}}(\mathcal{S})[\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_p)]^+, \tag{23}$$

where $K_{\mathcal{GP}}(\mathcal{S}) = (k_{\mathcal{GP}}(\mathbf{x}_i, \mathbf{x}_j))_{ij}$ $\forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{S}$ and A^+ denotes the Moore-Penrose pseudoinverse of A. The approximation accuracy clearly depends on the choice of the set $\mathcal{S}$ which describes the input region in which one wants to mimic the covariance function $C_{\mathcal{GP}}$. As already mentioned we are most often only interested in prediction and thus it is at hand to use the training and, if available, the test inputs in the calculation of Σ_w. Note that no output values enter Equation (23), so we can use any possible input set from the region of interest and are not restricted to the training or test set. It is conceivable to sample the set $\mathcal{S}$ artifically according to some arbritary distribution.

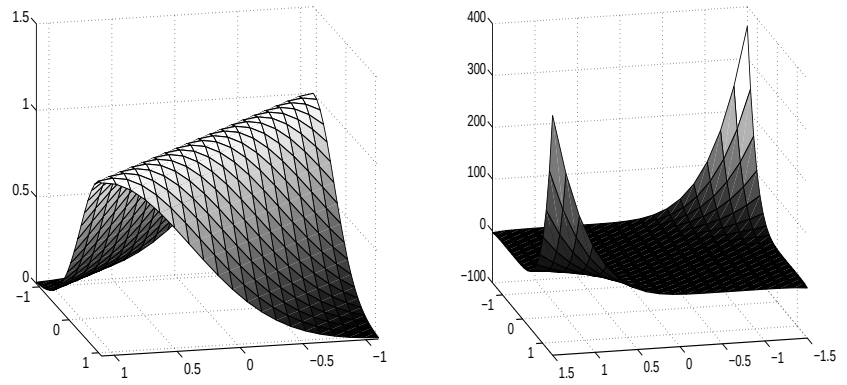
For our choice of the basis functions, Eq. (23) becomes

$$\Sigma_w = ((K^{-\frac{1}{2}} K(\mathcal{S}))^\top)^+ K_{\mathcal{GP}}(\mathcal{S})(K^{-\frac{1}{2}} K(\mathcal{S}))^+ \tag{24}$$

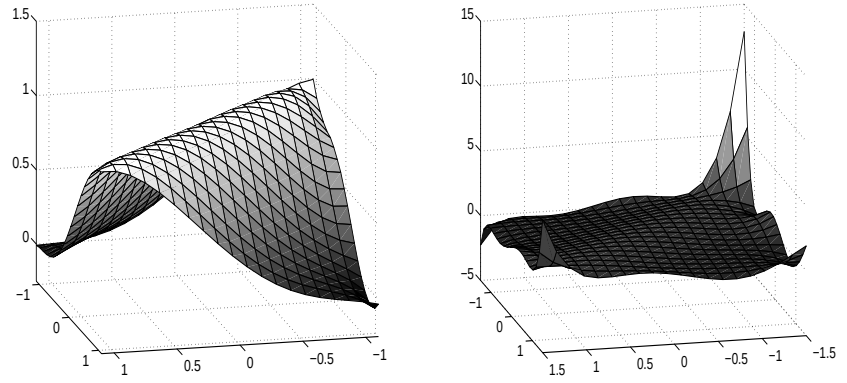
where we wrote the cross-covariance between the training points and the points in $\mathcal{S}$ as $(K(S))_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ $\forall \mathbf{x}_j \in \mathcal{S}, 1 \leq j \leq n$. The equations above hold for a general $\mathcal{S}$ and simplify to

$$\Sigma_w = K^{-\frac{1}{2}} K_{\mathcal{GP}} K^{-\frac{1}{2}} \tag{25}$$

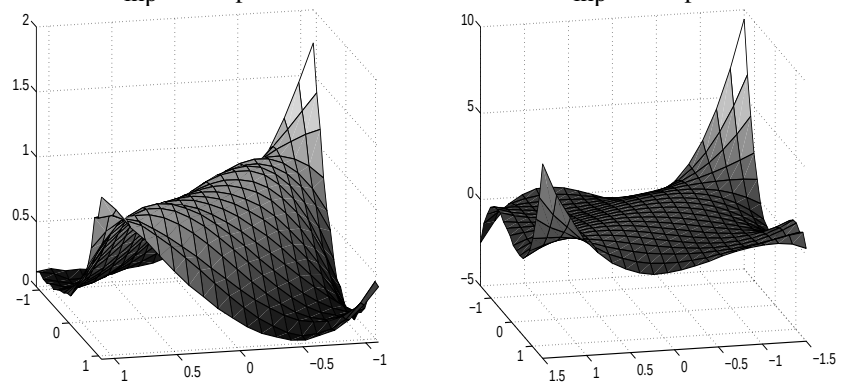
when only the training inputs are used. Note that for general sets $\mathcal{S}$ and kernels, positive definiteness of the resulting covariance matrix cannot be guaranteed. Although $K_{\mathcal{GP}}(\mathcal{S})$ is positive definite, $(K^{-\frac{1}{2}} K(\mathcal{S}))^+$ can be rank-deficient such that the overall covariance becomes positive semidefinite. In practice, however, this problem turned out to be less serious since we always approximate a valid covariance function. If the approximation is good enough this automatically leads to positive definite covariance matrices. Moreover, the size of the approximation set $\mathcal{S}$ typically exceeds that of the training set. In this case, $(K^{-\frac{1}{2}} K(\mathcal{S}))^+$ becomes only rank-deficient when the basis functions are linearly dependent.



(a) k_{gauss} (b) $k_{\text{ihp}}, p = 5$



(c) $k = \Phi_{\text{ihp}}^\top \Sigma_w \Phi_{\text{ihp}}$ degree $p = 5$ (d) $k = \Phi_{\text{ihp}}^\top \Sigma_w \Phi_{\text{ihp}}$ degree $p = 5$



(e) $k = \Phi_{\text{ihp}}^\top \Sigma_w \Phi_{\text{ihp}}$ degree $p = 3$ (f) $k = \Phi_{\text{ihp}}^\top \Sigma_w \Phi_{\text{ihp}}$ degree $p = 3$

Figure 3: Decoupling of basis and covariance in a one-dimensional toy example. (a) Gaussian covariance, (b) inhomogeneous polynomial kernel with degree 5, (c)+(d) inhomogeneous kernel with degree $p = 5$ adapted to the Gaussian covariance function on $[-1, 1]$. (e)+(f) same as (c)+(d), but for degree $p = 3$. Note that (c) and (d) ((e) and (f) resp.) show the same covariance functions, plotted on different input ranges.

In Figure 3 a one-dimensional toy example is plotted which illustrates the decoupling technique. An example of a polynomial covariance function (inhomogeneous polynomial kernel $k_{\text{ihp}}(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\top \mathbf{x}')^p$ with degree $p = 5$) is shown in Fig. 3 (b). The Gaussian covariance function $k_{\text{gauss}}(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2)$ is chosen as our target covariance $k_{\mathcal{GP}}(\mathbf{x}, \mathbf{x}')$ (Fig. 3(a)). The training inputs consisted of 10 points drawn uniformly from $[-1, 1]$. As our basis, we chose the Kernel PCA map (Eq. 21) for the inhomogeneous polynomial kernel and computed Σ_w according to Eq.(23) using an artificially created set $\mathcal{S}$ of 20 equidistant points in $[-1, 1]$. The resulting covariances for the degree $p = 5$ are shown in Figures 3(c)+(d), and for $p = 3$ in (e)+(f). Note that in both cases the left plot shows the range $[-1.1, 1.1]^2$ in which the approximation was computed, whereas the same function is shown in the right plot on a larger input region, namely $[-1.5, 1.5]^2$.

One can clearly see that the target covariance is well approximated in the domain of the data points. However, we can deform the covariance structure only locally leading to considerable deviations from the target outside $[-1, 1]^2$. The polynomial kernel with $p = 5$ results in a better approximation on $\mathcal{S}$ in this example, both visibly and numerically, but shows a stronger deviation outside of $\mathcal{S}$. However, it is not always the case that a higher degree leads to a better approximation. In the next section we show examples of the opposite behaviour.

4 Experiments

The decoupling approach was evaluated on three regression datasets: Boston Housing (Harrison & Rubinfeld, 1978), KIN40K (Schwaighofer & Tresp, 2003), and Stereopsis (Sinz, Quiñonero-Candela, Bakır, Rasmussen, & Franz, 2004). The boston dataset consists of 506 points in 13 dimensions. The results are averaged over 10 splits of the data with 10% used for testing and the rest for training. We created 5 splits of the 8-dimensional KIN40K dataset using 500 training and 39500 test examples. The Stereopsis dataset is the smallest among the three with 4 dimensions, 200 training and 792 testing examples. For this dataset, the task is to predict three different function values (3 dimensions) for each of which a single Gaussian process was trained and the given results are averaged over these three functions.

Approximation accuracy of the desired Gram matrix In our first experiment, approximation accuracy was measured using the 2-norm of the difference of the target Gram matrix and the obtained Gram matrix. In Figure 4 this quantity is shown as a function of the degree of the inhomogeneous polynomial kernel, along with the resulting difference in regression performance. Fig. 4(a) shows the results on one split of the Boston housing dataset, Fig. 4(b) the average over all three Gaussian Processes for the Stereopsis dataset. In both cases there is an optimal degree for both the approximation accuracy and the regression performance. Since polynomials with higher degree usually lead to ill-conditioned Gram matrices we included the condition numbers (ratio of the largest to the smallest singular value) of the covariance of the weights and the Gram matrix in the plot. One can see that the condition number of Σ_w and the approximation accuracy show similar behavior. In particular, the best approximation coincides with the best conditioned matrix Σ_w. Hence, we used the approximation accuracy to select the polynomial degree during model selection in the following experiment.

Regression performance In a baseline experiment, we used standard Gaussian process regression with the Gaussian kernel $k_{\text{gauss}}(\mathbf{x}, \mathbf{x}')$ and the inhomogeneous polynomial kernel $k_{\text{ihp}}(\mathbf{x}, \mathbf{x}')$. Model selection was done by maximizing the marginal log-likelihood (see, e.g., Rasmussen & Williams, 2006). The results of the baseline experiment are shown in the 3rd and 4th column of Table 1. The Gaussian covariance leads to a better regression performance on all three datasets although on the Stereopsis dataset the difference was very small.

To test our approach, we chose the Gaussian covariance as our target covariance as $k_{\mathcal{GP}}(\mathbf{x}, \mathbf{x}')$, and the Kernel PCA map computed for k_{ihp} as our basis. Calculation of Σ_w was done either on the training inputs only, or on both training and test inputs. Due to the required expensive matrix inverse we did not use all but only a subset of 2000 test points from the KIN40K dataset in the calculation of Σ_w. The results are summarized in the 5th and 6th column of Table 1. The model parameters σ and σ_n were again selected by maximizing the standard marginal log-likelihood criterion. The choice of the polynomial degree turned out to be sometimes problematic since the marginal log-likelihood as a function of polynomial degree tended to be very flat. We therefore used the above mentioned approximation accuracy of the covariance matrix as model selection criterion for the polynomial degree.

When only the training inputs were used for approximating the desired covariance, improvements over standard regression were only small, or - in the case of KIN40K - performance severely degraded because the training inputs were not sampled densely enough. Since the training error was the same as in the original Gaussian covariance,

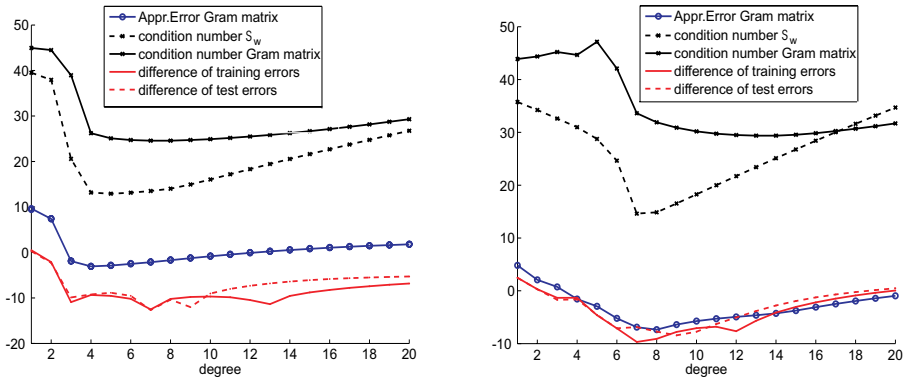


Figure 4: Condition number of Gram matrix and approximation accuracy as a function of the kernel degree. Experiments were done with the Kernel PCA map of the inhomogeneous polynomial kernel and are shown here on a log scale. Figure (a) shows the results on one split of the Boston dataset, Figure (b) the averages over all three Gaussian processes on the stereo dataset.

this is a clear indicator of overfitting to the training data. However, the results consistently improved on all datasets when the approximation was computed on both training and test inputs, so this should be the method of choice when the performance of polynomial regression is to be improved. The fact that the performance for the Boston Housing and Stereopsis datasets is slightly better than that of the Gaussian kernel is due to the approximation error of the target covariance. In principle, we should get the same results, but in these cases the approximated covariance yields *by accident* better results than the original one.

5 Discussion

In the second part of this report, we set out to offer some solutions to the severe generalisation problems of classical Wiener and Volterra models. As we have shown, both smoothness and non-divergence of the found solutions can be enforced by a special type of regularisation in the framework of Gaussian processes. We achieved this by approximating the covariance function of another Gaussian process with the desired properties (in our examples, this was the Gaussian covariance function) on a finite set of points. Our results show that the generalisation performance of the regularised Wiener models comes very close to that of the approximated Gaussian process. As a consequence, Wiener and Volterra models can be pushed to have a comparable performance to current state-of-the-art regression methods. Taken together, the methods presented in both parts of this report allow for extending the methodology of Wiener and Volterra analysis to novel application fields, largely alleviating the previous restrictions on input dimensionality, output noise and generalisation performance.

The idea of separately controlling the regularisation properties of the basis and the regulariser Σ_w has already been suggested by Smola and Schölkopf (1998) in the context of kernel regression. An application of this principle is the whitened regression approach of Franz, Kwon, Rasmussen, and Schölkopf (2004) in which Σ_w was chosen to flatten the covariance function over a given input range. Whitened regression was applied to the same scenario as considered here, i.e., improving polynomial regression for Wiener and Volterra analysis. Flattening alone, however,

Dataset		k_{gauss}	k_{ihp}	$\mathcal{S}_{\text{train}}$	$\mathcal{S}_{\text{train}} \cup \mathcal{S}_{\text{test}}$
Boston housing	train	3.58	5.11	3.58	3.59
	test	8.36	9.79	9.53	8.3
KIN40K	train	0.59	9.84	0.59	2.57
	test	10.41	21.07	114	13.76
Stereo	train	1.80	2.55	1.80	1.80
	test	3.26	3.38	3.30	3.25

Table 1: Averaged mean squared error on training and test set (10 folds for Boston, 5 for KIN40K, 3 for Stereo). For the inhomogeneous polynomial kernel with approximated Gaussian covariance, Σ_w is computed either on the training inputs only ($\mathcal{S}_{\text{train}}$), or on both training and test inputs ($\mathcal{S}_{\text{train}} \cup \mathcal{S}_{\text{test}}$).

turned out to be not sufficient to significantly improve polynomial regression performance. Our results suggest that approximating a localized covariance function instead leads to significant improvements in polynomial regression.

Walder, Schölkopf, and Chapelle (2006) proposed a method which computes the regulariser norm within the span of the chosen function basis. However, this method is not directly applicable to the case of polynomial basis functions as the involved integral is divergent. Still one might apply their method to this case by restricting it to a smaller domain e.g. by using an indicator function in the integral which would be similar to our approach of adapting the covariance on a finite domain only.

Finally, it should be pointed out that the presented method is not limited to approximating a Gaussian process with an infinite-dimensional function dictionary by a finite one (as is the case in our example of approximating a Gaussian covariance with polynomials). The presented derivation does not rely on this property. As a consequence, the method can be applied to arbitrary combinations of covariances and bases.

Acknowledgments

We are grateful for the discussions with Olivier Chapelle, Bernhard Schölkopf, and Carl Rasmussen who have helped us to develop and clarify many of the ideas presented here.

References

Franz, M. O., Kwon, Y., Rasmussen, C. E., & Schölkopf, B. (2004). Semi-supervised kernel regression using whitened function classes. In C. E. Rasmussen, H. H. Bülthoff, M. A. Giese, & B. Schölkopf (Eds.), *Pattern Recognition, Proc. 26th DAGM Symposium*, Vol. 3175 of *LNCS*, pp. 18 – 26 Berlin. Springer.

Franz, M. O., & Schölkopf, B. (2006). A unifying view of Wiener and Volterra theory and polynomial kernel regression. *Neural Computation*, **18**, 3097 – 3118.

Harrison, D., & Rubinfeld, D. (1978). Hedonic prices and the demand for clean air. *J. Environ. Economics & Management*, **5**, 81 – 102. Data available from http://lib.stat.cmu.edu/datasets/boston.

Nowak, R. (1998). Penalized least squares estimation of Volterra filters and higher order statistics. *IEEE Trans. Signal Proc.*, **46**(2), 419 – 428.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Cambridge, MA: MIT Press.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Schwaighofer, A., & Tresp, V. (2003). Transductive and inductive methods for approximate Gaussian process regression. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in Neural Information Processing Systems*, Vol. 15, pp. 953 – 960 Cambridge, MA. MIT Press.

Sinz, F., Quiñonero-Candela, J., Bakır, G. H., Rasmussen, C. E., & Franz, M. O. (2004). Learning Depth From Stereo. In C. E. Rasmussen, H. H. Bülthoff, M. A. Giese, & B. Schölkopf (Eds.), *Pattern Recognition, Proc. 26th DAGM Symposium*, Vol. 3175 of *LNCS*, pp. 245 – 252 Berlin. Springer.

Smola, A. J., & Schölkopf, B. (1998). On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, **22**, 211 –231.

Walder, C., Schölkopf, B., & Chapelle, O. (2006). Implicit surface modelling with a globally regularised basis of compact support. In *Computer Graphics Forum (Proc. EUROGRAPHICS)*, Vol. 25, pp. 635 – 644.