Semi-Supervised Kernel Regression Using Whitened Function Classes

Matthias O. Franz, Younghee Kwon, Carl Edward Rasmussen, and Bernhard Schölkopf

Max-Planck-Institut für biologische Kybernetik
Spemannstr. 38, 72076 Tübingen
{mof;kwon;carl;bs}@tuebingen.mpg.de,
http://www.kyb.tuebingen.mpg.de/

Abstract. The use of non-orthonormal basis functions in ridge regression leads to an often undesired non-isotropic prior in function space. In this study, we investigate an alternative regularization technique that results in an implicit whitening of the basis functions by penalizing directions in function space with a large prior variance. The regularization term is computed from unlabelled input data that characterizes the input distribution. Tests on two datasets using polynomial basis functions showed an improved average performance compared to standard ridge regression.

1 Introduction

Consider the following situation: We are given a set of N input values $\mathbf{x}_i \in \mathbb{R}^m$ and the corresponding N measurements of the scalar output values t_i. Our task is to model the output by linear combinations from a dictionary of fixed functions φ_i of the input $\mathbf{x}$, i.e.,

$$\hat{y}_i = \sum_{j=1}^{M} \gamma_j \varphi_j(\mathbf{x}_i), \quad \text{or more conveniently,} \quad \hat{y}_i = \gamma^\top \phi(\mathbf{x}_i), \tag{1}$$

using $\phi(\mathbf{x}_i) = (\varphi_1(\mathbf{x}_i), \varphi_2(\mathbf{x}_i), \dots)^\top$. The number of functions M in the dictionary can be possibly infinite as for instance in a Fourier or wavelet expansion. Often, the functions contained in the dictionary are neither normalized nor orthogonal with respect to the input. This situation is common in kernel ridge regression with polynomial kernels. Unfortunately, the use of a non-orthonormal dictionary in conjunction with the ridge regularizer $\|\gamma\|^2$ often leads to an undesired behaviour of the regression solutions since the constraints imposed by this choice rarely happen to reflect the - usually unknown - prior probabilities of the regression problem at hand. This can result in a reduced generalization performance of the solutions found.

In this study, we propose an approach that can alleviate this problem either when unlabelled input data is available, or when reasonable assumptions can be

made about the input distribution. From this information, we compute a regularized solution of the regression problem that leads to an *implicit whitening* of the function dictionary. Using examples from polynomial regression, we investigate whether whitened regression results in an improved generalisation performance.

2 Non-orthonormal functions and priors in function space

The use of a non-orthonormal function dictionary in ridge regression leads to a non-isotropic prior in function space. This can be seen in a simple toy example where the function to be regressed is of the form $t_i = \sin(ax_i)/(ax_i) + n_i$ with the input x_i uniformly distributed in $[-1, 1]$ and an additive Gaussian noise signal $n_i \approx N(0, \sigma_\nu^2)$. Our function dictionary consists of the first six canonical polynomials $\phi_1(x) = 1, \phi_2(x) = x, \phi_3(x) = x^2, \ldots, \phi_6(x) = x^5$ which are neither orthogonal nor normalized with respect to the uniform input. The effects on the type of functions that can be generated by this choice of dictionary can be seen in a simple experiment: we assume that the weights in Eq. 1 are distributed according to an isotropic Gaussian, i.e., $\gamma \approx N(0, \sigma^2 I_6)$ such that no function in the dictionary receives a higher a priori weight. Together with Eq. 1, these assumptions define a prior distribution over the functions $\hat{y}(x)$ that can be generated by our dictionary. In our first experiment, we draw samples from this distribution (Fig. 1*a*) and compute the mean square of $\hat{y}(x)$ at all $x \in [-1, 1]$ for 1000 functions generated by the dictionary (Fig. 1*b*). It is immediately evident that, given a uniform input, our prior narrowly constrains the possible solutions around the origin while admitting a broad variety near the interval boundaries. If we do ridge regression with this dictionary (here we used a Gaussian Process regression scheme, for details see [5]), 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. 1*c*). This can lead to bad predictions in sparsely populated areas.

If we choose a dictionary of orthonormal polynomials instead (in our example these are the first six Legendre polynomials), we observe a different behaviour: the functions sampled from the prior show a richer structure (Fig. 1*d*) with a relatively flat mean square value over the interval $[-1, 1]$ (Fig. 1*e*). As a consequence, the ridge regression solution usually does not diverge in sparsely populated regions near the interval boundaries (Fig. 1*f*).

The reason for this behaviour can be seen if one thinks of the functions as points in a function space. The dictionary defines a basis in a subspace such that all possible solutions of the form Eq. 1 are linear combinations of these basis vectors. Assuming an isotropic distribution of the weights, a non-orthogonal basis results in a non-isotropic distribution of points in function space. As a result, any new function to be expressed (or regressed) in this basis will have a larger probability if its projection onto the basis happens to be along a larger principal component, i.e., we have a *non-isotropic prior* in function space. Conversely, an orthonormal basis in conjunction with an isotropic weight distribution results in an isotropic prior in function space such that no specific function is preferred over

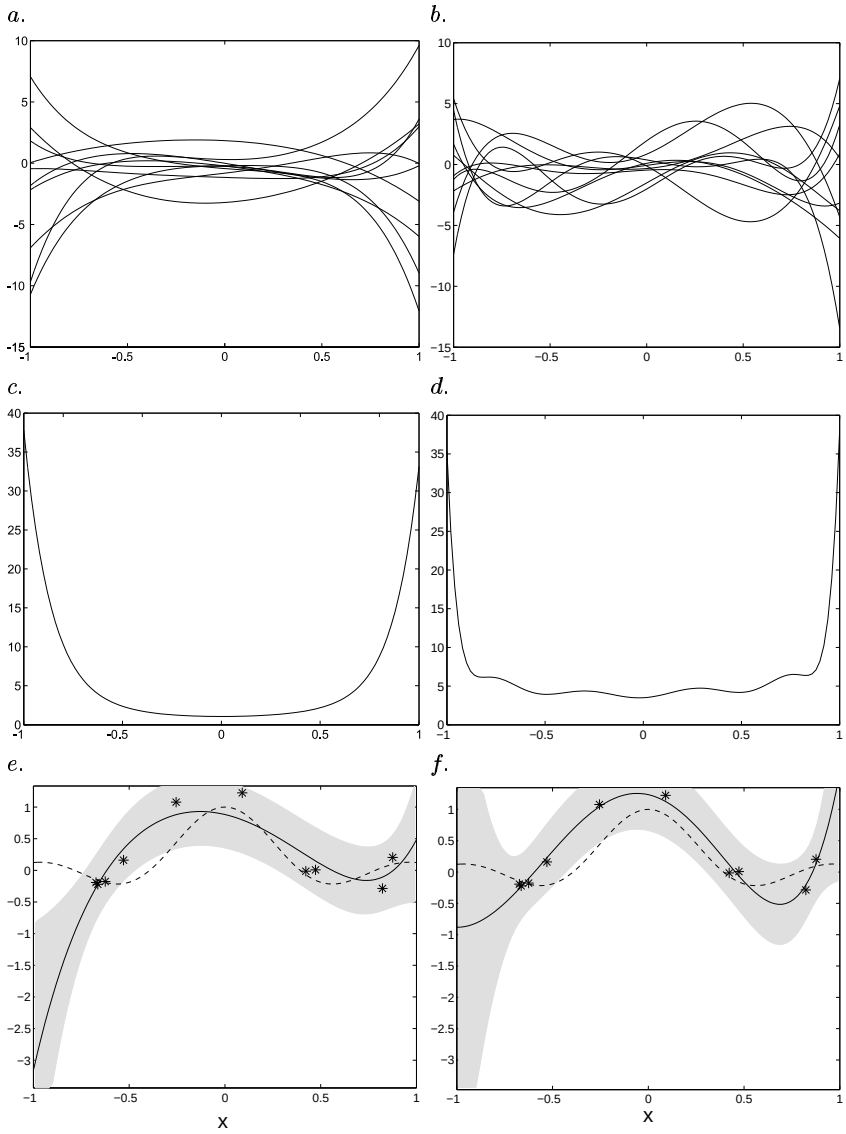

Fig. 1. Toy experiment using polynomial bases: *a.* 10 examples drawn from the prior in function space generated by the first 6 canonical polynomials and *b.* generated by the first 6 Legendre polynomials; *c.* Mean squared value in the interval $[-1, 1]$ of 1000 random linear combinations of the first 6 canonical polynomials and *d.*, of the first 6 Legendre polynomials; *e.* Regression on 10 training samples (stars) using the canonical polynomial basis and *f.*, the Legendre basis. The dashed line denotes the true function, the solid line the prediction from regression. The shaded areas show the 95%-confidence intervals.

another. This situation may often be preferable if nothing is known in advance about the function to be regressed.

3 Whitened regression

The standard solution to regression is to find the weight vector γ in Eq. 1 that minimizes the sum of the squared errors. If we put all $\phi(\mathbf{x}_i)$ into an $N \times M$ design matrix $\mathit{\Phi}$ with $\mathit{\Phi} = (\phi(\mathbf{x}_1)^\top, \phi(\mathbf{x}_2)^\top, \ldots \phi(\mathbf{x}_N)^\top)^\top$, the model (1) can be written as $\hat{\mathbf{y}} = \mathit{\Phi}\gamma$ such that the regression problem can be formulated as

$$\underset{\gamma}{\operatorname{argmin}} (\mathbf{t} - \mathit{\Phi}\gamma)^2. \tag{2}$$

The problem with this approach is that if the noises n_i are large, then forcing $\hat{\mathbf{y}}$ to fit as closely as possible to the data results in an estimate that models the noise as well as the function to be regressed. A standard approach to remedy such problems is known as the method of regularization in which the square error criterion is augmented with a penalizing functional

$$(\mathbf{t} - \mathit{\Phi}\gamma)^2 + \lambda J(\gamma), \quad \lambda > 0. \tag{3}$$

The penalizing functional J is chosen to reflect prior information that may be available regarding γ, λ controls the tradeoff between fidelity to the data and the penalty $J(\gamma)$. In many applications, the penalizing functional can be expressed as a quadratic form

$$J(\gamma) = \gamma^\top \Sigma_\gamma^{-1} \gamma \tag{4}$$

with a positive definite matrix Σ_γ^{-1}. The solution of the regression problem can be found analytically by setting the derivative of expression (3) with respect to γ to zero and solving for γ:

$$\gamma_{opt} = (\lambda \Sigma_\gamma^{-1} + \mathit{\Phi}^\top \mathit{\Phi})^{-1} \mathit{\Phi}^\top \mathbf{t}. \tag{5}$$

Based on γ_{opt}, we can predict the output for the new input $\mathbf{x}_*$ using

$$\hat{y}_* = \gamma_{opt}^\top \phi(\mathbf{x}_*) = \mathbf{t}^\top \mathit{\Phi} (\lambda \Sigma_\gamma^{-1} + \mathit{\Phi}^\top \mathit{\Phi})^{-1} \phi(\mathbf{x}_*) \tag{6}$$

Note that the solution depends only on products between basis functions evaluated at the training and test points. For certain function classes, these can be efficiently computed using *kernels* (see next section). In ridge regression, an isotropic penalty term on γ corresponding to $\Sigma_\gamma = \sigma^2 I_M$ is chosen. This can lead to a non-isotropic prior in function space as we have seen in the last section for non-orthonormal function dictionaries.

What happens if we transform our basis such that it becomes orthonormal? The proper transformation can be found if we know the covariance matrix C_ϕ of our basis with respect to the distribution of $\mathbf{x}$

$$C_\phi = E_\mathbf{x}[\phi(\mathbf{x})\phi(\mathbf{x})^\top] \tag{7}$$

where $E_\mathbf{x}$ denotes the expectation with respect to $\mathbf{x}$. The *whitening transform* is defined as

$$D = D^\top = C_\phi^{-\frac{1}{2}}. \tag{8}$$

The transformed basis $\tilde{\phi} = D\phi$ has an isotropic covariance as desired:

$$C_{\tilde{\phi}} = E_{\mathbf{x}}[\tilde{\phi}(\mathbf{x})\tilde{\phi}(\mathbf{x})^\top] = E_{\mathbf{x}}[D\phi(\mathbf{x})\phi(\mathbf{x})^\top D^\top] = DE_{\mathbf{x}}[\phi(\mathbf{x})\phi(\mathbf{x})^\top]D^\top = I_M. \quad (9)$$

Often, however, the matrix C_ϕ will not have full rank such that a true whitening transform cannot be found. In these cases, we propose to use a transform of the form

$$D = (C_\phi + I_M)^{-\frac{1}{2}}. \quad (10)$$

This choice prevents the amplification of possibly noise-contaminated eigenvectors of C_ϕ with small eigenvalues (since the minimal eigenvalue of $(C_\phi + I_M)$ is 1) while still leading to a whitening effect for eigenvectors with large enough eigenvalues.

When we substitute the transformed basis $\tilde{\phi} = D\phi$ into Eq. (5) using $\Sigma_\gamma = I_M$, we obtain

$$\gamma_{opt} = D^{-1}(\lambda(D^{-1})^2 + \boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^\top\mathbf{t}. \quad (11)$$

The prediction equation (6) is accordingly

$$\hat{y} = \mathbf{t}^\top\boldsymbol{\Phi}(\lambda(D^{-1})^2 + \boldsymbol{\Phi}^\top\boldsymbol{\Phi})^{-1}\phi(\mathbf{x}_*) \quad (12)$$

It follows that doing standard ridge regression with a withened, orthonormal basis is equivalent to doing regression in the original, non-orthonormal basis using the regularizer $J(\gamma) = \gamma^\top D^{-2}\gamma$. This allows us to use an implicitely whitened basis without the need to change the basis functions themselves. This is particularly useful when we do not have the freedom to choose our basis as, for instance, in kernel-based methods where the basis functions are determined by the kernel (see next section).

The proposed approach, however, suffers from a certain drawback because we need to know C_ϕ. In certain cases, the input distribution is known or can be approximated by reasonable assumptions such that C_ϕ can be computed beforehand. In other cases, *unlabelled data* is available which can be used to estimate C_ϕ. The training data itself, however, cannot be used to estimate C_ϕ since the estimate is proportional to $\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$. When substituted into Eq. (12) this amounts to no regularization at all. As a consequence, for the proposed approach to work it is absolutely necessary to obtain C_ϕ from data independent of the training data.

4 Whitened Kernel Regression

When the number of basis functions is large, a direct solution to the regression problem as described in the previous section becomes infeasible. Fortunately, there is a work-around to this problem for many important function classes: We noted in the previous section that the regression solution depends only on products between basis functions evaluated at the training and test points. For certain function dictionaries, the product between the functions evaluated at two input values $\mathbf{x}_1$ and $\mathbf{x}_2$ can be expressed as

$$\phi(\mathbf{x}_1)^\top\phi(\mathbf{x}_2) = k(\mathbf{x}_1, \mathbf{x}_2). \quad (13)$$

The function $k(\mathbf{x}_1, \mathbf{x}_2)$ on $\mathbb{R}^m \times \mathbb{R}^m$ is a *positive definite kernel* (for a definition see [3]). As a consequence, the evaluation of a possibly infinite number of terms in $\phi(\mathbf{x}_1)^\top \phi(\mathbf{x}_2)$ reduces to the computation of the kernel k directly on the input. Equation (13) is only valid for *positive definite* kernels, i.e., functions k with the property that the *Gram matrix* $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is positive definite for all choices of the $\mathbf{x}_1, \ldots, \mathbf{x}_N$. It can be shown that a number of kernels satisfies this condition including polynomial and Gaussian kernels [3].

A kernelized version of whitened regression is obtained by considering the set of n basis functions which is formed by the *Kernel PCA Map* [3]

$$\phi(\mathbf{x}) = K^{-\frac{1}{2}}(k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \ldots k(\mathbf{x}_N, \mathbf{x}))^\top. \tag{14}$$

The subspace spanned by the $\phi(\mathbf{x}_i)$ has the structure of a *reproducing kernel Hilbert space (RKHS)*. By carrying out linear methods in a RKHS, one can obtain elegant solutions for various nonlinear estimation problems [3], examples being Support Vector Machines. When we substitute this basis in Eq. (5), we obtain

$$\gamma_{opt} = (\lambda \Sigma_\gamma^{-1} + K)^{-1} K^{\frac{1}{2}} \mathbf{t} \tag{15}$$

using the fact that $\boldsymbol{\Phi} = K^{-\frac{1}{2}} K = K^{\frac{1}{2}} = \boldsymbol{\Phi}^\top$. By setting $\mathbf{k}(\mathbf{x}) = (k(\mathbf{x}_1, \mathbf{x}), k(\mathbf{x}_2, \mathbf{x}), \ldots k(\mathbf{x}_n, \mathbf{x}))^\top$, the prediction (6) becomes

$$\hat{y}_* = \mathbf{t}^\top (\lambda K^{\frac{1}{2}} \Sigma_\gamma^{-1} K^{-\frac{1}{2}} + K)^{-1} \mathbf{k}(\mathbf{x}_*). \tag{16}$$

It can be easily shown that this solution is exactly equivalent to Eq. 6 if Eq. 13 holds. When choosing $\Sigma_\gamma = I_N$, one obtaines the solution of standard kernel ridge regression [1]. Application of the whitening prior leads to

$$\hat{y}_* = \mathbf{t}^\top (\lambda R + K)^{-1} \mathbf{k}(\mathbf{x}_*) \tag{17}$$

Here, $C_\phi = K^{-\frac{1}{2}} C_\mathbf{k} K^{-\frac{1}{2}}$ and $C_\mathbf{k} = E_\mathbf{x}[\mathbf{k}(\mathbf{x})\mathbf{k}(\mathbf{x})^\top]$. This results in $R = K^{-\frac{1}{2}} C_\mathbf{k}$ or $R = K^{-\frac{1}{2}} C_\mathbf{k} + I_N$, depending on the choice of D.

5 Experiments

We compare whitened regression to ridge regression [1] using the kernelized form of Eq. 17 with $R = K^{-\frac{1}{2}} C_\mathbf{k} + I_N$ and Eq. 16 with $\Sigma_\gamma = I_N$, respectively. We consider three types of polynomial kernels that differ in the weights assigned to the different polynomial orders: the *summed polynomial kernel*

$$k_{sp}(\mathbf{x}_1, \mathbf{x}_2) = \sum\nolimits_{i=0}^{d} (\mathbf{x}_1^\top \mathbf{x}_2)^i; \tag{18}$$

the *adaptive polynomial kernel*

$$k_{ap}(\mathbf{x}_1, \mathbf{x}_2) = \sum\nolimits_{i=0}^{d} a_i (\mathbf{x}_1^\top \mathbf{x}_2)^i; \tag{19}$$

Table 1. Average squared error for whitened and ridge regression. Significant p-values < 0.1 are marked by a star.

Kernel	SUMMED POLYNOMIAL	ADAPTIVE POLYNOMIAL	INHOMOGENEOUS POLYNOMIAL
Sinc dataset			
Ridge regression	1.126	1.578	0.863
Whitened regression	0.886	0.592	0.787
p-value (t-test)	0.471	0.202	0.064*
Boston house-price			
Ridge Regression	18.99	16.37	18.74
Whitened Regression	12.83	15.78	13.08
p-value (t-test)	0.022*	0.817	0.053*

where the weights a_i are hyperparameters adapted during the learning process, and the *inhomogeneous polynomial kernel*

$$k_{ihp}(\mathbf{x}_1, \mathbf{x}_2) = (1 + \mathbf{x}_1^\top \mathbf{x}_2)^d = \sum\nolimits_{i=0}^{d} \binom{d-i}{i} (\mathbf{x}_1^\top \mathbf{x}_2)^i. \tag{20}$$

In both experiments, we used a 10 fold cross-validation setup with disjoint test sets. For each of the 10 partitions and the different kernels, we computed the squared error loss. In addition to the average squared loss, we tested the significance of the performance difference between whitened and standard regression using a t-test on the squared loss values.

1. Sinc dataset. Our first experiment is the $\sin(ax)/(ax)$ toy example ($a = 8$, noise variance $\sigma_\nu^2 = 0.05$) of Sec. 2 with disjoint training sets of 10 examples and disjoint test sets of 80 examples. We estimated the covariance $C_\mathbf{k}$ for Eq. 17 from 4000 additional unlabelled cases. The hyperparameters λ, and a_i were estimated by conjugate gradient descent on the analytically computed leave-one-out error [4], the best degree d was also chosen according to the smallest leave-one-out error for all orders up to 10.

2. Boston Housing. For testing whitened regression on real data, we took disjoint test sets of 50/51 examples and training sets of 455/456 examples from the Boston house-price dataset [2]. Note that due to dependencies in the training sets, independence assumptions needed for the t-test could be compromised. Since the Boston house-price dataset does not provide additional unlabelled data, we had to generate 2000 artificial unlabelled datapoints for each of the 10 trials based on the assumption that the input is uniformly distributed between the minima and maxima of the respective training set. The artificial datapoints were used to estimate $C_\mathbf{k}$. Instead of the leave-one-out error, we used conjugate gradient descent on a Bayesian criterion for selecting the hyperparameters, usually referred to as negative log evidence [5]. The maximal degree d tested was 5.

The results in Table 1 show that whitened regression performes on the average better than standard ridge regression. However the improvement appears to

be relatively small in many cases such that we get a significant result with $p < 0.1$ only for the inhomogeneous polynomial kernel on both datasets and for the summed polynomial kernel on the Boston house-price set. The weaker significance of the results on the Sinc dataset can be attributed to the very small number of training samples which leads to a large variance in the results.

The assumption of a uniformly distributed input in the Boston housing data seems to be useful as it leads to a general improvement of the results. The significantly better performance for the summed and the inhomogeneous polynomial kernel is mainly caused by the fact that often the standard ridge regression found only the linear solution with a typical squared error around 25, whereas whitened regression always extracted additional structure from the data with squared errors between 10 and 16.

6 Conclusion

Using a non-orthonormal set of basis function for regression can result in an often unwanted prior on the solutions such that an orthonormal or whitened basis is preferable for this task. We have shown that doing standard regression in a whitened basis is equivalent to using a special whitening regularizer for the non-orthonormal function set that can be estimated from unlabelled data.

Our results indicate that whitened regression using polynomial bases leads only to small improvements in most cases. In some cases, however, the improvement is significant, particularly in cases where the standard polynomial regression could not find a non-trivial solution. As a consequence, whitened regression is always an option to try when unlabelled data is available, or when reasonable assumptions can be made about the input distribution.

Acknowledgements. C.E.R. was supported by the German Research Council (DFG) through grant RA 1030/1.

1. N. Cristianini and J. Shawe-Taylor. *Support vector machines.* Cambridge University Press, Cambridge, 2000.
2. D. Harrison and D. Rubinfeld. Hedonic prices and the demand for clean air. *J. Environ. Economics & Management*, 5:81 – 102, 1978. Data available from http://lib.stat.cmu.edu/datasets/boston.
3. B. Schölkopf and A. J. Smola. *Learning with kernels.* MIT Press, Cambridge, MA, 2002.
4. V. Vapnik. *Estimation of dependences based on empirical data.* Springer, New York, 1982.
5. C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Processing Systems*, volume 8, pages 598 – 604, Cambridge, MA, 1996. MIT Press.