2011

Optimal predictive kernel regression via feature space principle components

Ernest Fokoue

Optimal Predictive Kernel Regression via Feature Space Principal Components

Ernest Fokoué
Center for Quality and Applied Statistics, Rochester Institute of Technology
98 Lomb Memorial Drive, Rochester, NY 14623, USA
eMail: ernest.fokoue@rit.edu

Abstract

We propose a simple use of principal component analysis in feature space that allows the derivation of optimal predictive kernel regression. The proposed approach is shown to perform well on both artificial and real data. Despite its incredible simplicity, the proposed method is found to compete very well with sophisticated statistical approaches like the Relevance Vector Machine and the Support Vector Machine.

Keywords: *Kernel Regression, Sparsity, Principal Component Analysis.*

1 Introduction

We are given a training set $\mathcal{D} = \{(\mathbf{x}_i, \mathrm{y}_i), i = 1, \cdots, n : \ \mathbf{x}_i \in \mathcal{X} \subset I\!R^p, \mathrm{y}_i \in I\!R\}$ where y_i are realizations of $Y_i = f^*(\mathbf{x}_i) + \epsilon_i$, and ϵ_i is the noise term. For simplicity and without loss of generality, we shall assume throughout this paper that the data are standardized. We assume that the true function f^* can be approximated by

$$f_n(\mathbf{x}) = \sum_{j=1}^{n} \mathrm{w}_j \mathcal{K}(\mathbf{x}, \mathbf{x}_j), \tag{1}$$

where $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to I\!R$ is the Gaussian radial basis function kernel defined by

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\omega^2}\right), \tag{2}$$

for some bandwidth $\omega > 0$. Now, the function f_n as defined in Eq. (1) is called a radial basis function (RBF) with weights $\mathrm{w}_1, \mathrm{w}_2, \cdots, \mathrm{w}_n$ and centers $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$, assumed to be pairwise distinct. We shall consider $\mathcal{H}_\mathcal{K}$, the family of radial basis functions (RBF) generated by $\mathcal{K}$, i.e.

$$\mathcal{H}_\mathcal{K} = \left\{ h : I\!R^p \to I\!R : \ \forall \mathbf{x} \in I\!R^p, \ h(\mathbf{x}) = \sum_{i=1}^{k} \mathrm{w}_i \mathcal{K}\left(\mathbf{x}, \xi_i\right) : k \in \mathbb{N}, \xi_i \in I\!R^p, \mathrm{w}_i \in I\!R \right\}$$

The great appeal of $\mathcal{H}_{\mathcal{K}}$ resides in the fact that, for any given continuous function f, there exist (a) a kernel $\mathcal{K}(\cdot,\cdot)$, (b) an optimum number of basis functions k, (c) a set of weights $\{\mathrm{w}_i\}_{i=1}^k$ and (d) a set of centers $\{\xi_i\}_{i=1}^k$ such that the corresponding function $h_k \in \mathcal{H}_{\mathcal{K}}$ approximates f to any desired precision. This approach to regression really became popular after the invention and publication by [1] of the Relevance Vector Machine (RVM). [7] provides one of the earliest and most comprehensive coverage of kernel methods as used in machine learning. However, it is important to note that kernel methods - as they are called in the machine learning community - owe the vast popularity partly to a strong theoretical justification by Kimerdolf and Wahba (1971)'s representer theorem. The reader is referred to [11] for a more detailed account of the above fundamental result. Kernel methods have come under intense and careful scrutiny in recent years. Researchers - probably fueled by the guarantee provided by such important results as the representer theorem - continually dig deeper into the framework, exploring a variety of aspects of it. One of the most important aspects of kernel regression - besides the crucial issue of the choice of the kernel - is the search for a sparse representation. Indeed, sparsity was the professed motivation of [1], and later of [3], [2]. In fact, for most situations and indeed most kernels, the statistical estimation of the weights w_j's by traditional error minimization (least squares) or density maximization (MLE) methods turns out to be an ill-posed problem, for which there is no hope of a decent solution without some form of regularization or constraints to help stabilize the solution. Of course, regularization in and of itself does not necessarily yield a sparse solution. Indeed, the form of the regularizer and/or appropriate subsequent refinements performed on the regularized solution are the keys to obtaining the desired level of sparsity.

Definition 1 *Given a vector* $\mathbf{w}^\top = (\mathrm{w}_1, \cdots, \mathrm{w}_n) \in I\!R^n$*, we define* $\mathtt{supp}(\mathbf{w}) = \{j : \ \mathrm{w}_j \neq 0\}$*, and the zero-norm of* $\mathbf{w}$ *will simply be*

$$\|\mathbf{w}\|_0 = |\mathtt{supp}(\mathbf{w})| = \textit{number of nonzeroes entries in } \mathbf{w}$$

Definition 2 *Let* $s \in \mathbb{N}$ *be a natural number. We shall say that a solution* $\mathbf{w}^*$ *is s-sparse if* $\|\mathbf{w}^*\|_0 \leq s$*, i.e. if* $\mathbf{w}^*$ *has at most s nonzero entries.*

Throughout this paper, we make the usual assumption that the noise terms are independent zero-mean Gaussian random variables with the same variance σ^2, i.e. $\epsilon_i \overset{iid}{\sim} \boldsymbol{\mathcal{N}}(0, \sigma^2)$. As a result, we have the likelihood

$$p(\mathbf{y} \,|\, \mathbf{w}, \sigma^2) = \boldsymbol{\mathcal{N}}_n(\mathbf{y} \,|\, \mathbf{K}\mathbf{w}, \sigma^2 \mathbf{I}_n), \tag{3}$$

corresponding to the model of equation (1) conveniently rewritten in vector-matrix form as

$$\mathbf{y} = \mathbf{K}\mathbf{w} + \boldsymbol{\epsilon}, \tag{4}$$

where $\mathbf{y} = (\mathrm{y}_1, \mathrm{y}_2, \cdots, \mathrm{y}_n)^\mathsf{T}$, $\mathbf{w} = (\mathrm{w}_1, \mathrm{w}_2, \cdots, \mathrm{w}_n)^\mathsf{T}$, $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \cdots, \epsilon_n)^\mathsf{T}$ and $\mathbf{K} = (\mathbf{K}_{ij})$ where $\mathbf{K}_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|),\ i, j = 1, \cdots, n$. We ideally seeks to solve

$$\min_{\tilde{\mathbf{w}} \in I\!R^n} \|\tilde{\mathbf{w}}\|_0 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{K}\tilde{\mathbf{w}}\|^2 < \eta$$

where η might be related to the variance of the noise term ϵ.

1.1 Main result

We consider obtaining a least squares solution to the estimation of the weight vector $\mathbf{w}$. However, it turns out that

$$\hat{\mathbf{w}}_{\texttt{ols}} = (\mathbf{K}^\top \mathbf{K})^{-1} \mathbf{K}^\top \mathbf{y}$$

cannot be obtained in practice, because the matrix $(\mathbf{K}^\top \mathbf{K})$ is typically ill-conditioned. Various authors have restored to techniques of regularization such Ridge Regression [8] and LASSO [9] to isolate a unique solution. A typical ridge regression solution would be of the form

$$\hat{\mathbf{w}}_{\texttt{ridge}} = (\mathbf{K}^\top \mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{K}^\top \mathbf{y}$$

where the ridge constant (tuning parameter) can be estimated through cross validation or generalized cross validation. It turns out that the ridge regression approach in this context is fraught with various difficulties due to the nature of the matrix $\mathbf{K}$.

1.2 Our approach

In this paper, however, we consider a straightforward technique based on the spectral decomposition of $(\mathbf{K}^\top \mathbf{K})$, namely

$$\mathbf{K}^\top \mathbf{K} = \mathbf{P} \mathbf{\Lambda} \mathbf{P}^\top.$$

- We use $\mathbf{\Lambda}$ to determine q, the number of relevant rows to retain in $\mathbf{P}$ and then form the relevant projection matrix $\mathbf{Q} \in \mathbb{R}^{n \times q}$ made up of the first q columns of $\mathbf{P}$.
- We then form the new $n \times q$ relevant "design" matrix

$$\mathbf{Z} = \mathbf{K}\mathbf{Q}$$

- Then we get the q-dimensional estimate $\hat{\mathbf{w}}_{\texttt{pck}}$ of the relevant weights

$$\hat{\mathbf{w}}_{\texttt{pck}} = (\mathbf{Z}^\top \mathbf{Z})^{-1} \mathbf{Z}^\top \mathbf{y}$$

- For each new vector $\mathbf{x}_{\texttt{new}}$, we first compute

$$\mathbf{a}_{\texttt{new}} = (\mathcal{K}(\mathbf{x}_{\texttt{new}}, \mathbf{x}_j), \mathcal{K}(\mathbf{x}_{\texttt{new}}, \mathbf{x}_2), \cdots, \mathcal{K}(\mathbf{x}_{\texttt{new}}, \mathbf{x}_n))^\top,$$

  then we crucially form the q-dimensional relevant version of $\mathbf{a}_{\texttt{new}}$ as

$$\mathbf{z}_{\texttt{new}} = \mathbf{Q}^\top \mathbf{a}_{\texttt{new}}$$

- The point estimate of the response at $\mathbf{x}_{\texttt{new}}$ is then given by

$$\hat{\mathbf{y}}_{\texttt{new}} = \mathbf{z}_{\texttt{new}}^\top \hat{\mathbf{w}}_{\texttt{pck}} = \hat{f}_{\texttt{pck}}(\mathbf{x}_{\texttt{new}})$$

The immediate gain here lies in the fact that we use only standard spectral decomposition elements, and not complex computation is involved. We are therefore in the presence of a computationally efficient approach to prediction in kernel regression.

An even bigger benefit comes from the fact our initial computational results on both artificial and real life data show that our straightforward and intuitively simple technique works very well, and even outperforms sophisticated techniques like the relevance vector machine and the support vector machine.

$$R_m(f) = \frac{1}{m} \sum_{(\mathbf{x}_{\texttt{new}}, \mathrm{y}_{\texttt{new}}) \in \texttt{Test}} (\mathrm{y}_{\texttt{new}} - f(\mathbf{x}_{\texttt{new}})^2$$

which the empirical version of

$$R(f) = \mathbb{E}[(Y - f(X))^2] = \int_{(\mathbf{x}_{\texttt{new}}, \mathrm{y}_{\texttt{new}})} (\mathrm{y}_{\texttt{new}} - f(\mathbf{x}_{\texttt{new}})^2 dP(\mathbf{x}_{\texttt{new}}, \mathrm{y}_{\texttt{new}})$$

2 Numerical explorations

2.1 First artificial example

As our first illustrative example, we consider recovering the univariate sinc function from noisy observations. The choice of the sinc function is motivated by the fact that it has become the *de facto* benchmark example in nearly all papers on kernel regression since its use by Vladimir Vapnik in some of the earliest computational examples of statistical learning theory.

$$f(\mathrm{x}) = \frac{\sin \mathrm{x}}{\mathrm{x}} \quad \mathrm{x} \in [-10, +10]$$

We generate $n = 99$ points with noise variance $\sigma^2 = 0.3^2$.

	PCK	SVM	RVM
Training Error	0.0297	0.0300	0.0307
Test Error	0.2036	0.2145	0.2093

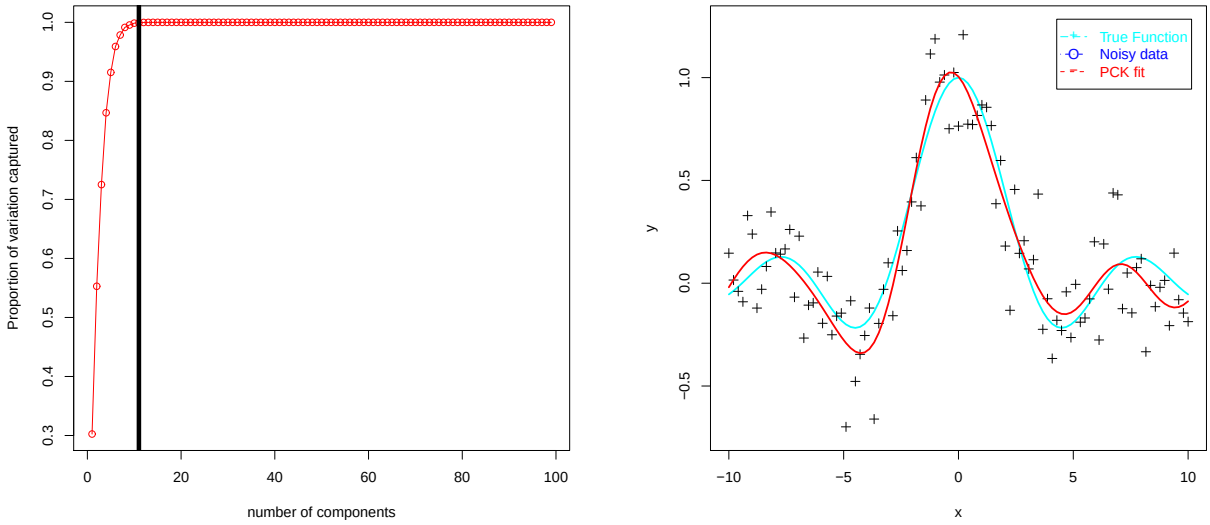


Figure 1: *Results of the PCK algorithm for the hill function. The panel on the left shows that our technique picks up around* 11 *relevant components. Note that this corresponding to essentially to the totality of the variation.*

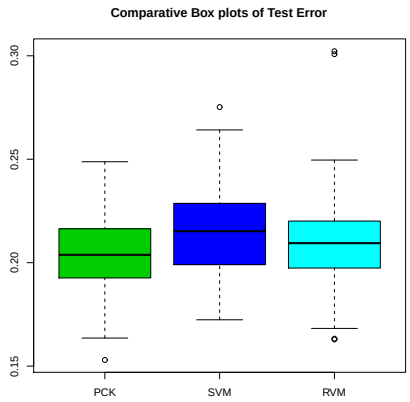


Figure 2: *Comparison of the predictive performances. PCK seems to have a minute edge, but clearly, the three methods are performing equally well on this data.*

2.2 Second Artificial Example

As our second artificial example, we consider exploring the hill function defined as

$$f(\mathrm{x}) = -\mathrm{x} + \sqrt{2}\sin(\pi^{3/2}\mathrm{x}^2) \quad \text{with} \quad \mathrm{x} \in [-1, +1]$$

This function is somewhat qualitatively different from the previous one. Like before, we generate $n = 99$ points, and use a noise variance of $\sigma^2 = 0.3^2$.

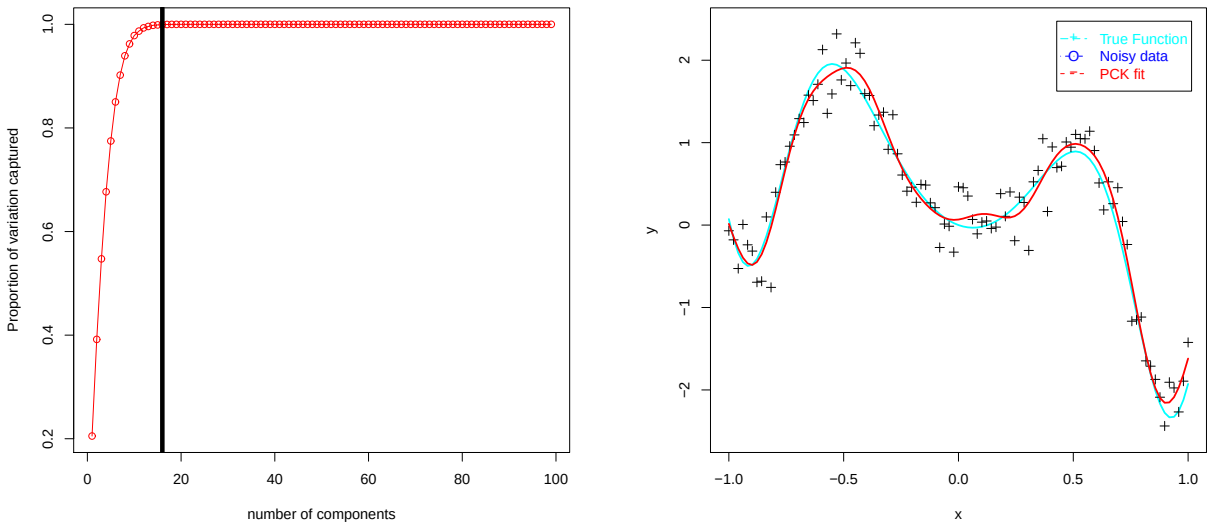


Figure 3: *Results of the PCK algorithm for the hill function. The panel on the left shows that our technique picks up around* 16 *relevant components. Note that this corresponding to essentially to the totality of the variation.*

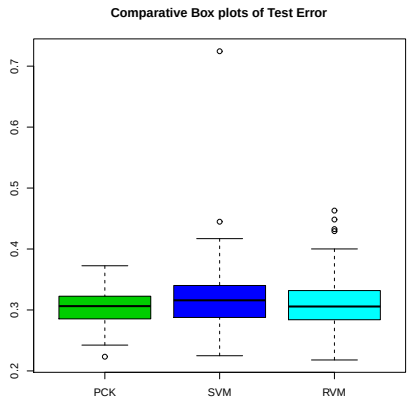


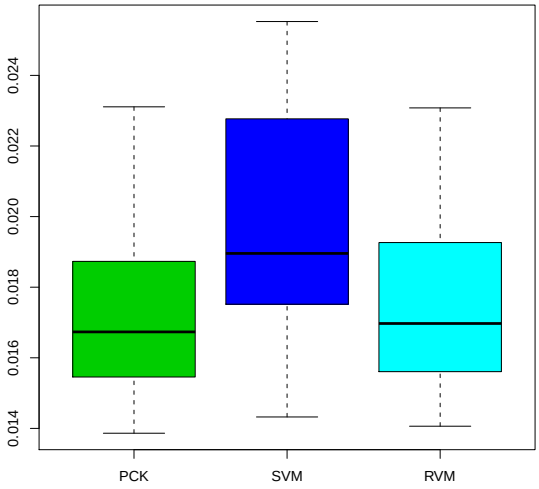
Figure 4: *Comparison of the predictive performances. PCK seems to have a minute edge, but clearly, the three methods are performing equally well on this data.*

2.3 Real life example

The Boston housing data set is a well known benchmark data set often used to test the performance of regression methods.

	PCK	SVM	RVM
Training Error	0.0002	0.0002	0.0001
Test Error	0.0174	0.0201	0.0181

For this data set, the Principal Component approach retains 49 relevant components.



2.4 Summary of performances

	PCK	SVM	RVM
Sinc	0.2036	0.2145	0.2093
Hill	0.0109	0.0210	0.0170
Boston	0.0171	0.0197	0.0181

Table 1: *Comparison of the predictive performance of three methods based on the root mean squared error on the test set. In this case,* $m = 50$ *replications is used each for each data set. The proposed method PCK is compared to both SVM and RVM*

3 Conclusion

We have proposed a simple use of principal component analysis in feature space that allows the derivation of optimal predictive kernel regression. The proposed approach is shown to perform well on both artificial and real data. Despite its incredible simplicity, the proposed method is found to compete very well with sophisticated statistical approaches like the Relevance Vector Machine and the Support Vector Machine. The proposed method merits to be considered seriously because of simplicity. An aspect worth investigating in our future work is the derivation of way to trace back the relevant vectors themselves rather than settle with the relevant components.

References

[1] Tipping, M. E. (2001). Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, **1**, 211-244.

[2] Fokoué, E. and Sun, D. and Goel, P. (2009). Fully Bayesian Analysis of the Relevance Vector Machine with Consistency Inducing Priors. *Statistical Methodology*. Invited Paper, to appear.

[3] Fokoué, E. (2008). Estimation of Atom Prevalence for Optimal Prediction. *Contemporary Mathematics*. Vol **443**, pp 103-129, The American Mathematical Society.

[4] Fokoué, E. (2008). Stabilization of Atom Selection for Optimal Prediction via Mixture Modelling of the Sample Path. *Technical Report*. Number EPF-08-9-1. Department of Mathematics, Kettering University, 1700 West Third Avenue, Flint, Michigan, USA. (2008) *Submitted to Computational Statistics and Data Analysis*

[5] Fokoué, E. and P. Goel (2007). An Optimal Experimental Design Perspective on the Relevance Vector Machine, *Technical Report*. Number EPF-07-12-1. Department of Mathematics, Kettering University, 1700 West Third Avenue, Flint, Michigan, USA.

[6] Zhang, Z., Jordan, M. I. and Yeung, D. (2008). Posterior Consistency of the Silverman g-Prior in Bayesian Model Choice. *Technical Report*, Number xx, Department of Electrical Engineering and Computer Science, University of California, Berkeley, California, USA.

[7] Scholkopf, B. and Smola, A. (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond. MIT Press, 2002.

[8] Grandvalet, Y. (1998). Least absolute shrinkage is equivalent to quadratic penalization. In L. Niklasson,M. Boden, and T. Ziemske, editors, ICANN98, volume 1 of Perspectives in Neural Computing, pages 201206. Springer, 1998.

[9] Tibshirani, R. J. (1996). Regression Shrinkage and Selection via the LASSO.. *J. Roy. Statist. Soc. Ser. B*, **58**, 267-288.

[10] F. Liang and R. Paulo and G. Molina and M. Clyde and J. O. Berger (2008). Mixtures of g-priors for Bayesian Variable Selection. *J. Amer. Statist. Assoc.*, **103**, 410-423.

[11] Wahba, G (2000). An Introduction to Model Building with Reproducing Kernel Hilbert Spaces, *Technical Report No. 1020*, Department of Statistics, University of Wisconsin 1210 West Dayton St. Madison, WI 53706, USA, April 18, 2000