NIH Public Access
Author Manuscript
Comput Stat Data Anal. Author manuscript; available in PMC 2014 December 01.

Published in final edited form as:
Comput Stat Data Anal. 2013 December ; 68: 190–201. doi:10.1016/j.csda.2013.06.016.

Kernel Continuum Regression☆

Myung Hee Lee* and
Department of Statistics, Colorado State University, Fort Collins, CO 80525, U.S.A.

Yufeng Liu
Department of Statistics and Operations Research, Carolina Center for Genome Sciences, Department of Biostatistics, University of North Carolina, Chapel Hill, NC 27599

Abstract

The continuum regression technique provides an appealing regression framework connecting ordinary least squares, partial least squares and principal component regression in one family. It offers some insight on the underlying regression model for a given application. Moreover, it helps to provide deep understanding of various regression techniques. Despite the useful framework, however, the current development on continuum regression is only for linear regression. In many applications, nonlinear regression is necessary. The extension of continuum regression from linear models to nonlinear models using kernel learning is considered. The proposed kernel continuum regression technique is quite general and can handle very flexible regression model estimation. An efficient algorithm is developed for fast implementation. Numerical examples have demonstrated the usefulness of the proposed technique.

Keywords

Continuum Regression; Kernel regression; Ordinary Least Squares; Principal Component Regression; Partial Least Squares

1. Introduction

Regression is one of the most fundamental and useful statistical techniques. It helps to relate explanatory variables with a response variable and build predictive models. Ordinary Least Squares (OLS) regression estimates the conditional mean of the response variable given covariates and is commonly used in practice. Despite its simple implementation and good interpretability, OLS may face numerical unstability when there exists multi-collinearity among covariates or when the dimension of covariates is relatively high. In that case, Ridge Regression (RR), which can be viewed as a penalized approach, may serve as an alternative.

Another popular group of regression techniques is to perform regression analysis based on a small number of linear transformations of the explanatory variables. For example, Principal Component Regression (PCR) first summarizes multiple explanatory variables, which can

☆This article has supplementary material online.

© 2013 Elsevier B.V. All rights reserved.

*To whom correspondence should be addressed. Tel: 1-970-491-6682, Fax: 1-970-491-7895, mhlee@stat.colostate.edu (Myung Hee Lee), yfliu@email.unc.edu (Yufeng Liu).

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

NIH-PA Author Manuscript
NIH-PA Author Manuscript
NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

be high dimensional, into a few principal component directions and then performs regression on those principal component directions. These principal component directions are orthogonal to each other, yet contain most of the variations in the explanatory variables. Thus, PCR can circumvent the potential numerical difficulty of OLS. Partial Least Squares (PLS) is a related regression technique and it has been widely used in the field of chemometrics. Similar to PCR, PLS also uses a small number of linear transformations of the covariates for regression. The main difference of PLS from PCR is that PCR finds those transformations without the use of the response variable while PLS makes use of both covariates and the response variable to seek for suitable transformations.

With various regression techniques available, it would be desirable to study the differences and connections among these methods. Stone and Brooks (1990) pointed out that the seemingly different regression procedures such as OLS, PLS, and PCR differ only in one aspect: the target quantity maximized at the first step when finding linear transformations of the explanatory variables. Based on this analogy, they formulated a richer family of regression methods, Continuum Regression (CR), by introducing a continuum parameter which connects these three methods.

Similar to other methods, CR also aims to find directional vectors to transform the explanatory variables into new latent predictors which are orthogonal to each other and are constructed as linear combinations of the original predictors. There are two aspects of the latent predictors: one is the variation of the original predictors explained by each latent predictor and the other is the correlation between each latent predictor with the response. The quantity for the CR to maximize involves both variance and correlation of the latent variable with a parameter controlling the relative proportion of two terms. With its flexible construction, the CR is quite general and it contains OLS and PCR as the two extremes and PLS in the middle. In particular, the OLS ignores the variance of the latent variable and maximizes the correlation between the observed response vector and the predicted response vector. In contrast, PCR finds the regression directional vector so that the variance of the latent predictor is maximized. Interestingly, PLS essentially maximizes the covariance between the observed and the predicted response vectors. Besides these three special cases, CR also covers many other methods in the whole spectrum. Frank and Friedman (1993) provides a nice overview of CR and other related regression techniques. Sundberg (1993) and Björkström and Sundberg (1999) reveal some close connection between the RR and the set of CR. Chen and Cook (2010) investigated some asymptotic properties of CR.

The CR approach is potentially useful when the relationship between the response and the explanatory variable is linear. However, when the true relationship is nonlinear, the predictive performance of the CR family can be improved if the model is built as a nonlinear function of the explanatory variables.

In the literature, there has been some work in this direction which generalizes some special cases of CR via kernel learning. The nonlinear generalization of the building blocks for PCR, i.e., nonlinear Principal Component Analysis (PCA) has been studied in the field of pattern recognition, where lower dimensional feature extraction of high dimensional data becomes an important task. See Schölkopf et al. (1998),Mika et al. (1999), and Shawe-Taylor and Cristianini (2004) for more details.Rosipal et al. (2000b) andRosipal et al. (2000a) deal with nonlinear generalization of PCR, which uses nonlinear PCA as the latent variables in the regression analysis. As in the linear case, the feature extraction based on PCA is done not specifically to the regression problem at hand, and consequently the predictive performance of PCR is usually not as good as PLS. Walczak and Massart (1996) and Rosipal and Trejo (2001) generalize the PLS to incorporate nonlinear cases.

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

In this present paper, we extend the linear CR model to nonlinear CR model using the powerful kernel trick concept in machine learning. The proposed kernel CR (kCR) incorporates the special cases such as kernel OLS, kernel PLS and kernel PCR in one unified framework. If a linear kernel map is chosen, the kCR is the same as the ordinary CR. Section 2 provides mathematical formulation of kCR. The first part is devoted to present systematic ways to construct latent variables from an optimization point of view, and the second part is to run a regression analysis with the selected latent variables. Section 3 gives the details of the algorithm for solving the optimization problem in the first step. Numerical performance of the proposed method is investigated in Section 4 through simulation examples and real data analysis. We conclude the paper with some brief discussion in Section 5.

2. Continuum Regression and Its Kernel Extension

In this section, we first briefly review the linear CR in Section 2.1 and then introduce its kernel extension in Section 2.2.

2.1. Review of Linear Continuum Regression

Suppose that we have n pairs of data points for regression, $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ are the explanatory variables and $y_i \in \mathbb{R}$ is the response variable. Define the $n \times d$ input data matrix as $\mathbf{X} = [\mathbf{x}_1,\ldots, \mathbf{x}_n]^T$, where each row vector $\mathbf{x}_i$ represents a d-dimensional input vector for $i = 1, \ldots, n$. The output data vector is denoted by $\mathbf{y} = (y_1, \ldots, y_n)^T$. We assume that the data are mean-centered so that each column sum of the matrix $\mathbf{X}$ is $\mathbf{0}$. Denote X and Y as the random predictor vector and response variable respectively. Furthermore, define the scatter matrix of the data $\mathbf{X}$ as $\mathbf{S}_{d\times d} = \mathbf{X}^T\mathbf{X}$ and the cross covariance matrix between $\mathbf{X}$ and $\mathbf{y}$ as $\mathbf{s} = \mathbf{X}^T\mathbf{y}$.

We now describe the linear CR technique in terms of its optimization criterion. CR is essentially a two-step regression procedure where in the first step, one finds a set of direction vectors in the input variable space, $\mathbb{R}^d$, and makes projections of data onto the subspace generated by these vectors. In the second step, we use these extracted features as regressors to build a regression model to predict the value of the response variable Y. In particular, for a given parameter $\alpha \in [0, 1]$, suppose that the first k direction vectors $\mathbf{c}_1, \ldots, \mathbf{c}_k$ have been constructed and we want to find $\mathbf{c}_{k+1}$ by maximizing

$$T(\mathbf{c}) = \mathrm{Cov}(\mathbf{c}^T X, Y)^2 \mathrm{Var}(\mathbf{c}^T X)^{\alpha/(1-\alpha)-1} = (\mathbf{c}^T\mathbf{X}^T\mathbf{y})^2 (\mathbf{c}^T\mathbf{X}^T\mathbf{X}\mathbf{c})^{\alpha/(1-\alpha)-1} = (\mathbf{c}^T\mathbf{s})^2 (\mathbf{c}^T\mathbf{S}\mathbf{c})^{\alpha/(1-\alpha)-1} \quad (1)$$

subject to the constraints $|\mathbf{c}|^2 = 1$ and $\mathrm{Corr}(\mathbf{c}^T X, \mathbf{c}_j^T X) = \mathbf{c}^T\mathbf{S}\mathbf{c}_j = 0$, $j = 1, \ldots, k$.

Stone and Brooks (1990) showed that CR includes OLS and PCR at the two extremes, $\alpha = 0$ and $\alpha \to 1$, respectively. Specifically, OLS can be viewed as maximizing correlation. In particular, the multiple correlation coefficient is maximized over all direction vectors $\mathbf{c}$ of the correlation between $\mathbf{y}$ and $\mathbf{X}\mathbf{c}$. When $\alpha = 0$, the optimization criterion in (1) reduces to the multiple correlation coefficient for OLS. For the other extreme of this family with $\alpha \to 1$, the variance term in (1) dominates the optimization criterion and the role of $\mathbf{y}$ will be ignored. In that case, CR finds PCR directions which give linear combinations of d covariates with maximum variations. In this family, PLS can be viewed as a compromise of the two extremes at $\alpha = 0.5$. Once these direction vectors are obtained, one can construct the corresponding latent predictors and build regression models using these new predictors. In this case, the regression models are linear as the latent predictors are linear combinations of the original predictors.

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

PLS has been extensively used in the field of chemometrics since it was first introduced by Wold (1975). It was presented in an iterative algorithmic form as an alternative to OLS in the presence of high multicollinearity among input variables. Since its introduction, there have been various algorithms proposed for PLS (Helland, 1990; Naes and Martens, 1985), but we will stick to the version given as an optimization solution in Stone and Brooks (1990). PLS shares similarity with PCR in the sense that it extracts potential regressors by creating a set of orthogonal transformation of input variables. It is different since it directly makes use of the output variable when creating a set of latent variables - the selected latent variable maximizes the covariance between the output variable among all linear combinations of input variables. A least squares regression technique is applied once the latent variables are constructed.

As discussed earlier, the optimization problem (1) covers many regression techniques, including OLS, PLS, and PCR as special cases. In the next section, we discuss the extension of CR to the more general kernel framework.

2.2. Kernel Continuum Regression

Linear CR introduced by Stone and Brooks (1990) provides a nice regression framework which includes several well known regression methods as well as many other new techniques in this family. It helps us to better understand regression tools. Despite its usefulness, the original proposal was restricted to the linear setting. In this section, we explain how to extend the linear CR to a general kernel version.

Consider a mapping φ from $X \in R^d$ to a feature space, $\mathcal{F}$, and we propose to build a linear regression model using a set of latent variables in the feature space. Let $\langle \cdot, \cdot \rangle$ be the inner product defined in $\mathcal{F}$. For presentation of the algorithm, we use some matrix notations for the feature mapped data. Denote the feature mapped data matrix as $\varphi(\mathbf{X}) = [\varphi(\mathbf{x}_1), \ldots, \varphi(\mathbf{x}_n)]^{\mathrm{T}}$ and the $n \times n$ kernel matrix as $\mathbf{K}_{(i,j)} = \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle$. The $n \times 1$ vector storing the projection of each data vector onto the direction vector $\mathbf{c} \in \mathcal{F}$ as entries will be denoted by $\langle \mathbf{c}, \varphi(\mathbf{X}) \rangle$. Note that our use of the kernel function relates to the kernel trick concept used in machine learning. For example, the well known Support Vector Machine (Vapnik, 1998; Cristianini and Shawe-Taylor, 2000) utilizes the kernel trick to achieve nonlinear learning.

Let us define the transformed continuum parameter as $\gamma = \alpha/(1-\alpha) > 0$. Then the problem of finding direction vectors (1) in the feature space can be modified to the following:

$$\max T_\phi(\mathbf{c}) = \mathrm{Cov}(\langle \mathbf{c}, \phi(X) \rangle, Y)^2 \mathrm{Var}(\langle \mathbf{c}, \phi(X) \rangle)^{\gamma-1} = \{\langle \mathbf{c}, \phi(\mathbf{X}) \rangle^{\mathrm{T}} \mathbf{y}\}^2 \{\langle \mathbf{c}, \phi(\mathbf{X}) \rangle^{\mathrm{T}} \langle \mathbf{c}, \phi(\mathbf{X}) \rangle\}^{\gamma-1} \quad (2)$$

subject to the two constraints: $\langle \mathbf{c}, \mathbf{c} \rangle = 1$ and $\langle \mathbf{c}, \varphi(\mathbf{X}) \rangle^{\mathrm{T}} \langle \mathbf{c}_j, \varphi(\mathbf{X}) \rangle = 0$, $j = 1, \ldots, k$.

Next we show that the maximum value can be always achieved in the subspace generated by the mapped data vectors, $\mathcal{R}_{\varphi(\mathbf{X})} = \mathcal{C}\{\varphi(\mathbf{x}_1), \ldots, \varphi(\mathbf{x}_n)\} \subset \mathbb{R}^\infty$. Let $\mathbf{c}$ be any unit vector in the feature space and P and Q be the projection and the orthogonal projection maps onto $\mathcal{R}_{\varphi(\mathbf{X})}$, respectively. Consider the projection of $\mathbf{c}$ onto $\mathcal{R}_{\varphi(\mathbf{X})}$ and let us compare the objective function T_φ evaluated at $\mathbf{c}$ and $\tilde{\mathbf{c}} = P\mathbf{c}/|P\mathbf{c}|$, then we have

$$T_\phi(\mathbf{c}) = \{\langle P\mathbf{c}+Q\mathbf{c}, \phi(\mathbf{X}) \rangle^{\mathrm{T}} \mathbf{y}\}^2 \{\langle P\mathbf{c}+Q\mathbf{c}, \phi(\mathbf{X}) \rangle^{\mathrm{T}} \langle P\mathbf{c}+Q\mathbf{c}, \phi(\mathbf{X}) \rangle\}^{\gamma-1} = \{\langle P\mathbf{c}, \phi(\mathbf{X}) \rangle^{\mathrm{T}} \mathbf{y}\}^2 \{\langle P\mathbf{c}, \phi(\mathbf{X}) \rangle^{\mathrm{T}} \langle P\mathbf{c}, \phi(\mathbf{X}) \rangle\}^{\gamma-1} \leq \{\langle \tilde{\mathbf{c}}, \phi(\mathbf{X}) \rangle^{\mathrm{T}} \mathbf{y}\}^2 \{\langle \tilde{\mathbf{c}}, \phi(\mathbf{X}) \rangle^{\mathrm{T}} \langle$$

Projection of any direction vector onto the subspace, $\mathcal{R}_{\varphi(\mathbf{X})}$, will result in an improvement to achieve the maximization of T_φ. Consequently, we can restrict the solution $\mathbf{c}$ to be in $\mathcal{R}_{\varphi(\mathbf{X})}$, i.e., we search for the maximizer of the form $\mathbf{c} = \sum_{i=1}^n a_i \phi(\mathbf{x}_i) = \phi(\mathbf{X})^{\mathrm{T}} \mathbf{a}$. Using this

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

representation, a simple algebraic calculation shows that the i-th data vector projected on the j-th direction vector $\mathbf{c}_j$ can be expressed as a matrix multiplication with $\langle \mathbf{c}_j, \varphi(\mathbf{x}_i)\rangle$ = i-th row of $\mathbf{K} \cdot \mathbf{a}_j$. This means that if we use these transformed data in the analysis, a direct mapping $\varphi(\mathbf{x}_i)$ is not necessary to acquire as long as the kernel matrix $\mathbf{K}$ and the weight vector $\mathbf{a}_j$'s are available.

Based on the above considerations, the objective function (2) can be written as the following:

$$T_\phi(\mathbf{c}) = \{\langle \mathbf{a}^{\mathrm{T}}\phi(\mathbf{X}), \phi(\mathbf{X})\rangle^{\mathrm{T}}\mathbf{y})^2\}^2 \{\langle \mathbf{a}^{\mathrm{T}}\phi(\mathbf{X}), \phi(\mathbf{X})\rangle^{\mathrm{T}} \langle \mathbf{a}^{\mathrm{T}}\phi(\mathbf{X}), \phi(\mathbf{X})\rangle\}^{\gamma-1} = \{\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{y}\}^2 \{\mathbf{a}^{\mathrm{T}}\mathbf{K}^2\mathbf{a}\}^{\gamma-1}.$$

Furthermore, the orthonormality constraints can be rewritten as

$$\langle \mathbf{c}, \mathbf{c}\rangle = \mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a} = 1$$

and

$$\langle \mathbf{c}, \phi(\mathbf{X})\rangle^{\mathrm{T}} \langle \mathbf{c}_j, \phi(\mathbf{X})\rangle = \mathbf{a}^{\mathrm{T}}\mathbf{K}^2\mathbf{a}_j = 0,$$

where $\mathbf{c}_j = \varphi(\mathbf{X})^{\mathrm{T}}\mathbf{a}_j$ for some $\mathbf{a}_j \in \mathbb{R}^n$, $j = 1,\ldots, k$, are the k previously constructed continuum direction vectors. Consequently, for a given $\gamma > 0$, the optimization problem (2) in the feature space $\mathscr{F}$, can be formulated as an n-dimensional optimization problem:

$$\max_{\mathbf{a}\in\mathbb{R}^n} (\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{y})^2 (\mathbf{a}^{\mathrm{T}}\mathbf{K}^2\mathbf{a})^{\gamma-1} \quad (3)$$

subject to $\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a} = 1$ and $\mathbf{a}^{\mathrm{T}}\mathbf{K}^2\mathbf{a}_j = 0$ for $j = 1,\ldots, k$. Note that the problem depends on the data vectors only through the kernel matrix $\mathbf{K}$. The detailed algorithm on how to solve this dual problem (3) is presented in Section 3.

Once $\mathbf{a}_1,\ldots, \mathbf{a}_k$ are solved, we can get the first k kernel continuum directional vectors $\mathbf{c}_1,\ldots, \mathbf{c}_k \in \mathscr{F}$ via $\mathbf{c}_j = \varphi(\mathbf{X})^{\mathrm{T}}\mathbf{a}_j$. Next, we extract information by projecting the mapped data onto these direction vectors and use them as regressors for the regression fit procedure in the second step. Specifically, we use the continuum latent variables, $\langle (X), \mathbf{c}_1\rangle,\ldots, \langle \varphi(X), \mathbf{c}_k\rangle$, as regressor variables in a linear regression model as $Y = \beta_1 \langle (X), \mathbf{c}_1\rangle + \cdots + \beta_k \langle \varphi(X), \mathbf{c}_k\rangle + \varepsilon$.

Utilizing transformed data, a simple expression for the OLS estimate of the coefficient vector $\beta = (\beta_1,\ldots, \beta_k)$ is obtained as $\hat{\beta} = (\tilde{\mathbf{X}}^{\mathrm{T}}\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^{\mathrm{T}}\mathbf{y}$, where (i, j)-th entry of $\tilde{\mathbf{X}}_{n\times k}$ is the projection of the i-th data vector onto the j-th continuum direction vector, $\langle \mathbf{c}_j, \varphi(\mathbf{x}_i)\rangle$, i.e., $\tilde{\mathbf{X}}_{n\times k} = \mathbf{K}[\mathbf{a}_1,\ldots, \mathbf{a}_k]$.

Prediction of the level of the response variable can be made on any input data point $x \in \mathbb{R}^d$, $\hat{Y} = \hat{\beta}_1\langle \varphi(x), \mathbf{c}_1\rangle + \cdots + \hat{\beta}_k\langle \varphi(x), \mathbf{c}_k\rangle$. Suppose that we have a set of testing points $\{\mathbf{X}_i^*\}_{i=1,\cdots,n_t}$ to make predictions on y values, then the vector representation of the set of predicted values is given as follows:

$$\hat{\mathbf{y}} = \phi(\mathbf{X}^*)[\mathbf{c}_1, \cdots, \mathbf{c}_k]\hat{\beta} = \mathbf{K}_t[\mathbf{a}_1, \cdots, \mathbf{a}_k]\hat{\beta},$$

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

where $\mathbf{K}_t$ is an $n_t \times n$ matrix whose (i, j)-th element is $\langle \phi(\mathbf{x}_i^*), \phi(\mathbf{x}_j) \rangle$ for $i = 1, \ldots, n_t$ and $j = 1, \ldots, n$. Therefore, we can achieve nonlinear regression modeling building and prediction using the kernel function, without the necessity of having explicit $\varphi(\cdot)$. The kernel representation of the function can also be shown using the famous Representer Theorem by Kimeldorf and Wahba (1971) when a reproducing kernel is applied. In that case, the corresponding functional space is a reproducing kernel Hilbert space.

3. Construction Algebra for Kernel CR

Fast and simple implementation is essential for most statistical techniques. In this section, we discuss how to implement kCR. Note that our proposed kCR includes the original linear CR as a special case when the linear kernel is applied.

If the kernel matrix $\mathbf{K}$ is not a full rank matrix, the representation of the solution of a in (3) is not uniquely defined. In order to avoid ambiguity in the representation, we write the solution as the linear combination of the eigenvectors of the kernel matrix corresponding to nonzero eigenvalues. Let $\mathbf{K}_{(n\times n)}=\mathbf{U}_{(n\times m)}\mathbf{E}_{(m\times m)}\mathbf{U}^{\mathrm{T}}_{(m\times n)}$ be the eigen-decomposition of the kernel matrix $\mathbf{K}$ where m is the rank of $\mathbf{K}$ and $\mathbf{E}$ is the diagonal matrix of positive eigenvalues of $\mathbf{K}$. Then, the solution of the optimization problem can be expressed as $\mathbf{a}_{k+1} = \mathbf{U}\mathbf{E}^{-1/2}\mathbf{z}$, for some $\mathbf{z} \in \mathbb{R}^m$. It can be directly checked that the maximization problem (3) can be formulated as the following:

$$\max_{\mathbf{z}} (\mathbf{z}\mathbf{d})^2 (\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{z})^{\gamma-1}, \quad (4)$$

subject to $\mathbf{z}^{\mathrm{T}}\mathbf{z} = 1$ and $\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{Z}_k = 0$, where $\mathbf{d} = \mathbf{E}^{1/2}\mathbf{U}^{\mathrm{T}}\mathbf{y}$ and $\mathbf{Z}_k = [\mathbf{z}_1, \ldots, \mathbf{z}_k]$.

The problem above is equivalent to the maximization of its Lagrangian form:

$$T^*(\mathbf{z}) := (\mathbf{z}^{\mathrm{T}}\mathbf{d})^2 (\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{z})^{\gamma-1} - \lambda(\mathbf{z}^{\mathrm{T}}\mathbf{z} - 1) - 2\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{Z}_k\Lambda_k,$$

subject to $\mathbf{z}^{\mathrm{T}}\mathbf{z} = 1$ and $\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{Z}_k = 0$, where $\Lambda_k = [\lambda_1, \ldots, \lambda_k]$ and $\lambda, \lambda_1, \ldots, \lambda_k$ are the Lagrange multipliers. The maximizer should be the solution to the following equation:

$$\frac{\partial T^*}{\partial \mathbf{z}} = 2(\mathbf{z}^{\mathrm{T}}\mathbf{d})(\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{z})^{\gamma-1}\mathbf{d} + 2(\gamma-1)(\mathbf{z}^{\mathrm{T}}\mathbf{d})^2(\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{z})^{\gamma-2}\mathbf{E}\mathbf{z} - 2\lambda\mathbf{z} - 2\mathbf{E}\mathbf{Z}_k\Lambda_k = 0. \quad (5)$$

Now pre multiply $\mathbf{z}^{\mathrm{T}}$ to (5), then we have

$$(\mathbf{z}^{\mathrm{T}}\mathbf{d})^2(\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{z})^{\gamma-1} + (\gamma-1)(\mathbf{z}^{\mathrm{T}}\mathbf{d})^2(\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{z})^{\gamma-1} - \lambda\mathbf{z}^{\mathrm{T}}\mathbf{z} - \mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{Z}_k\Lambda_k = 0.$$

Since $\mathbf{z}^{\mathrm{T}}\mathbf{z} = 1$ and $\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{Z}_k = 0$, we can express λ in terms of the data as $\lambda = \gamma(\mathbf{z}^{\mathrm{T}}\mathbf{d})^2(\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{z})^{\gamma-1}$. Plugging this back into the equation (5), we obtain

$$(\mathbf{z}^{\mathrm{T}}\mathbf{d})(\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{z})^{\gamma-1}\mathbf{d} + (\gamma-1)(\mathbf{z}^{\mathrm{T}}\mathbf{d})^2(\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{z})^{\gamma-2}\mathbf{E}\mathbf{z} - \gamma(\mathbf{z}^{\mathrm{T}}\mathbf{d})^2(\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{z})^{\gamma-1}\mathbf{z} - \mathbf{E}\mathbf{Z}_k\Lambda_k = 0. \quad (6)$$

For the sake of simplicity, let us write the scalars as

$$\tau_{\mathbf{z}} = \mathbf{z}^{\mathrm{T}}\mathbf{d} \quad \rho_{\mathbf{z}} = \mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{z}. \quad (7)$$

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

Writing the equation (6) along with the linear constraint, $\mathbf{z}^{\mathrm{T}}\mathbf{E}\mathbf{Z}_k = 0$, in a matrix form, we obtain the following:

$$\begin{pmatrix} \gamma\tau_{\mathbf{z}}^2\rho_{\mathbf{z}}^{\gamma-1}\mathbf{I}+(1-\gamma)\tau_{\mathbf{z}}^2\rho_{\mathbf{z}}^{\gamma-2}\mathbf{E} & \mathbf{E}\mathbf{Z}_k \\ \mathbf{Z}_k^{\mathrm{T}}\mathbf{E} & 0 \end{pmatrix}\begin{pmatrix} \mathbf{z} \\ \Lambda_k \end{pmatrix}=\begin{pmatrix} \tau_{\mathbf{z}}\rho_{\mathbf{z}}^{\gamma-1}\mathbf{d} \\ \mathbf{0} \end{pmatrix}. \quad (8)$$

To simplify the expression, we introduce partitioned matrix notations, and rewrite the equation above as

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^{\mathrm{T}} & \mathbf{0} \end{pmatrix}\begin{pmatrix} \mathbf{z} \\ \Lambda_k \end{pmatrix}=\begin{pmatrix} \mathbf{q} \\ \mathbf{0} \end{pmatrix},$$

where $\mathbf{A}=\gamma\tau_{\mathbf{z}}^2\rho_{\mathbf{z}}^{\gamma-1}\mathbf{I}+(1-\gamma)\tau_{\mathbf{z}}^2\rho_{\mathbf{z}}^{\gamma-2}\mathbf{E}$, $\mathbf{B}=\mathbf{E}\mathbf{Z}_k$, and $\mathbf{q}=\tau_{\mathbf{z}}\rho_{\mathbf{z}}^{\gamma-1}\mathbf{d}$. Using the following fact on the inverse of the partitioned matrix,

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^{\mathrm{T}} & \mathbf{0} \end{pmatrix}^{-1}=\begin{pmatrix} \mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}(\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1} & \mathbf{A}^{-1}\mathbf{B}(\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ (\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1} & (\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1} \end{pmatrix}$$

and imposing the unit vector condition, the unit vector solution $\mathbf{z}$ to (8) becomes

$$\mathbf{z}=\frac{\mathbf{M}\mathbf{q}}{\|\mathbf{M}\mathbf{q}\|}, \quad (9)$$

where $\mathbf{M} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1}$. Note that $\mathbf{z}$ does not depend on $\tau_{\mathbf{z}}$ since this quantity is canceled out when normalized. It is important to point out that everything in (9) is known except for the scalar $\tau_{\mathbf{z}}$, which is determined by $\mathbf{z}$.

Combining the iterative updating scheme between $\mathbf{z}$ and ρ using the relationship (7) and (9), we are given a fixed point problem $\rho = g(\rho)$, where $g(\rho) = \mathbf{z}(\rho)^{\mathrm{T}}\mathbf{E}\mathbf{z}(\rho)$ and $\mathbf{z}(\rho)$ is the vector evaluated at ρ as in the function on the right hand side of (9). From the equation (7), it is clear that the solution ρ lies between $[e_m, e_1]$, where e_1 (e_m) is the largest (smallest) eigenvalue of $\mathbf{E}$. This observation essentially brings down our solution search into a 1-dimensional closed space.

A nutshell to summarize the kCR algorithm to find the maxima $\mathbf{z}$ for the problem (4) is shown below.

Step 1. Fix τ as an arbitrary value, say $\tau = 1$.

Step 2. Fix the $\rho \in [e_m, e_1]$.

Step 3. Compute $\mathbf{z}(\rho)=\frac{\mathbf{M}\mathbf{q}}{\|\mathbf{M}\mathbf{q}\|}$, where

$$\mathbf{M}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}(\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{A}^{-1}, \ \mathbf{A}=\gamma\tau^2\rho^{\gamma-1}\mathbf{I}+(1-\gamma)\tau^2\rho^{\gamma-2}\mathbf{E}, \ \mathbf{B}=\mathbf{E}\mathbf{Z}_k, \ \mathbf{q}=\tau\rho^{\gamma-1}\mathbf{d}.$$

Step 4. Evaluate $g(\rho) = \mathbf{z}(\rho)^{\mathrm{T}}\mathbf{E}\mathbf{z}(\rho)$.

Step 5. Based on Steps 1-4, solve the nonlinear equation $\rho = \mathbf{z}(\rho)^{\mathrm{T}}\mathbf{E}\mathbf{z}(\rho)$.

Let the solution be ρ^*.

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

Step 6. If there exist multiple roots for the nonlinear equation in Step 5, take the value that actually gives rise to $\mathbf{z}(\rho^*)$ which achieves the largest objective function value (4) among all the roots ρ^*'s. Let this optimal value be ρ^*.

Step 7. Complete search for the k-th continuum weight vector $\mathbf{a}_k$ by computing $\mathbf{z}(\rho^*)$ as in Step 3 and let $\mathbf{a}_k = \mathbf{U}\mathbf{E}^{-\frac{1}{2}}\mathbf{z}(\rho^*)$.

Note that the main challenging step is Step 5 of the algorithm where a numerical equation needs to be solved. However, since the equation is only a one-dimensional problem, typically it can be computed efficiently.

4. Numerical Study

In this section, we examine the performance of the proposed kCR using both simulated and real data examples. Our goal is to show the flexibility of kernel learning to achieve nonlinear regression function estimation as well as to examine the effect of the kCR parameter α.

4.1. Simulation

We generate an $n \times d$ dimensional $\mathbf{X}$ data matrix from a t-dimensional factor model,

$$\mathbf{X} = \mathrm{T}\mathbf{P}' + \mathbf{E},$$

where $\mathbf{T}$ is an $n \times t$ random matrix and $\mathbf{P}$ is a fixed $d \times t$ orthonormal matrix and $\mathbf{E}$ is a noise matrix. We set $k = 3$, $d = 7$, $n = 100$, $\beta_1 = \ldots = \beta_d = 1$. Each row of $\mathbf{T}$ is generated from the t dimensional Gaussian distribution $N_t(0, \sum)$, where $\sum = diag(3, 3, 3, 1, \ldots, 1)$, and each row of $\mathbf{E}$ is generated from $N_d(0, .5^2 I)$.

For the relationship between X and Y, we consider four different nonlinear regression problems and investigate the finite sample performance of kCR as described below. Let

$$f_1(x) = \sin(x);\ f_2(x) = \sin|x|/|x|;\ f_3(x) = 1/2\sqrt{x+10} - 1;\ f_4(x) = 5 + x - 5x^2 + .2x^3;\ f_5(x, y) = 5 + xy^2.$$

With these f functions, we consider four different settings as follows.

Setting 1: $Y = X_1\beta_1 + \cdots + X_d\beta_d + \varepsilon$;

Setting 2: $Y = 5f_2(X_1) + 5f_1(X_2 - 5) + f_4(X_3) + X_4\beta_4 + \cdots + X_d\beta_d + \varepsilon$;

Setting 3: $Y = 2f_2(X_1) + f_1(X_2/3) + f_2(X_2) + X_3\beta_3 + \cdots + X_d\beta_d + \varepsilon$;

Setting 4: $Y = 5f_2(X_1) + 10f_3(X_2) + f_5(X_2, X_3) + X_4\beta_4 + \cdots + X_d\beta_d + \varepsilon$.

Clearly, Setting 1 contains a linear regression problem. Settings 2-4 contain nonlinear functions. To illustrate this, Figure 1 displays scatter plots between (X_i, Y) for $i = 1, 2, 3$ for Setting 2. The true nonlinear relationship between X_i's and Y are also shown as dotted red curves. Clearly, the relationship between X_i and Y is nonlinear.

When fitting the kCR, we consider three different choices of kernels: i) the linear kernel $K(x, y) = x^{\mathrm{T}}y$, ii) the polynomial, $K(x, y) = (x^{\mathrm{T}}y + c)^m$, and iii) the Gaussian kernel $K(x, y) = e^{-|x-y|^2/h}$. For the polynomial kernel, we set $m = 2$, choose $c = -\min_{i,j}(x_i^{\mathrm{T}}x_j)$, and scale the kernel matrix so that the entries of K do not exceed 1. This step was needed to make the polynomial kernel stable. For the Gaussian kernel, the parameter h is chosen as the first quartile of all pairwise squared distances between X data vectors. There are two tuning parameters in kCR, $\alpha \in [0, 1]$ and the number of components to be used in the regression,

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

denoted by k. The parameter k can be as large as the rank of the kernel matrix **K** in theory, but in practice we want to keep k relatively small to avoid overfitting. We ran kCR with α ranging from 0 to 1 with an increment of 0.1 and k from 1 to 5. The tuning parameters are selected to minimize the validation error on an independent tuning set of size $n = 100$. Once the tuning parameters are selected, the test error is validated on an independent dataset of size $m = 1000$.

In order to see the effect of tuning parameter α, we present fitted values using the Gaussian kCR with three different values of α = 0, 0.5, and 1 in Figure 2. Scatter plot of Y against X_1 for Setting 2 is shown in Figure 2. The fitted Y values using the Gaussian kernel are overlaid as magenta squares. We set the number of latent variables as $k = 1$. The top (middle and bottom) panel shows kCR with α = 0 (α = .5 and α = 1). The predicted value on the top panel is very wiggly since it incorporates the sampling artifact greatly whereas the α = 1 case does not pick the relationship with Y well since the latent variables are extracted to explain most variability of X data.

Figures 3 and 4 show the boxplots of the test errors for kCR over 50 replications. For Setting 1 (top panel in Figure 3), the linear kernel gives better performance. This is expected since the underlying relationship is linear. In Setting 2, all three kernels show similar performance. In both cases considered, the family of kCR does not vastly improve the performance from the individual regression fit (either α = 0, 0.5 or 1), possibly because the nonlinearity is relatively mild. In Settings 3 and 4, the Gaussian kernel is the best, followed by the polynomial kernel, and the linear kernel shows the worst performance. As the true relationship is highly nonlinear, using a model that accounts for nonlinearity greatly improves the performance. Results from the two nonlinear kernels, Polynomial and Gaussian, are comparable.

Table 1 reports the average selected tuning parameters for kCR over 50 replications. The standard errors are reported in parentheses. The linear kernel tends to find a smaller number of latent variables than the Gaussian kernel does.

For the remaining section, in an effort to evaluate the computational cost of the algorithm developed in Section 3, we consider different combinations of (n, d) and compare the computation time. We use Setting 2 to create the data set. In this set up, the first 4 factors relate to Y in nonlinear fashions whereas the other factors have a linear relationship with Y. Dimensions of X are set as $d = 7, 14, 28, 56, 112, 224, 448$ and three different sample sizes are considered with $n = 50, 100$ and 200. We used the tic/toc function in MATLAB to measure the computation time in seconds and the average time over 5 repeats are reported in Figure 5. In the algorithm, one needs to perform eigen-decomposition of the kernel matrix, **K**. For this reason, the computational time increases as n increases, however, the computation cost does not increase rapidly with dimensions.

4.2. Real Data Analysis

In this section, we examine the performance of the proposed kCR using two publicly available data sets; diabetes (available at http://www.stanford.edu/~Hastie/Papers/LARS/) and Boston housing (available at http://archive.ics.uci.edu/ml/datasets/Housing). In the diabetes data set, 10 clinical variables (age, BMI, serum measurements, etc.) as well as a quantitative measurements for the progress of the disease (the response variable) were collected from $n = 442$ diabetes patients. A detailed explanation can be found inEfron et al. (2004). The Boston housing data set contains $n = 506$ census tracts of Boston from the 1970 census, which includes 12 continuous and 1 binary covariates (per capita crime rate by town, average number of rooms per dwelling, full-value property-tax rate, etc). The quantitative response variable is the median house value in USD.

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

We randomly split each data set into a training set and a test set, each of which approximately contained 2/3 and 1/3 of the whole data, respectively. Then, all methods were tuned by 10-fold CV within the training set. Once the tuning parameters get selected, the tuned methods are fit and tested on the test set. Shown in Figures 6 and 7 are the box plots of test error rates over 50 random splits. From the results, we can see that for the Diabetes data, all methods perform similarly while linear methods give the best overall performance. However, kCR gives comparable performance, despite the use of a much larger functional space through kernel learning. For the Boston housing data example, kCR methods yield much better performance than linear methods. This indicates that the underlying relationship is likely to be nonlinear. Our finding matches the previous analysis of this dataset in the literature, see for exampleZhang et al. (2011).

5. Discussion

Regression analysis is one of the most fundamental statistical tools. CR offers an attractive framework connecting many linear regression techniques in a spectrum. In this paper, we extend linear CR into a much more flexible kernel setting. Our proposed kCR covers the linear CR as a special case. Furthermore, the proposed kCR includes the kernel PLS and kernel PCR as members as well. An efficient algorithm is developed for fast computation. Our numerical examples demonstrate the usefulness of our proposed methodology.

Our current kCR does not perform variable selection. In many applications, variable selection can be useful to obtain accurate and interpretable models. To achieve that goal, one may consider more flexible forms of kernel functions to allow the method to automatically remove variables. Existing literature in nonlinear variable selection, for example, the COSSO (Lin and Zhang, 2007) and KNIFE (Allen, 2013), can be useful here. In that case, the corresponding computational algorithm can be much more challenging. Further investigation will be pursued.

Another line of extension is to rubustify kCR. The sample covariance and the sample variance used to extract features in the first construction step may not be robust in the presence of outliers, which may affect estimation of the regression coefficients at the second step. For the linear continuum regression case, a robust version was introduced by Serneels et al. (2005). A robust counterpart for kCR will be useful for practical problems.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

The authors would like to thank the editor, the associate editor, and three anonymous reviewers for their constructive comments and suggestions. Yufeng Liu’s research was partially supported by the NIH grant NIH/NCI R01 CA-149569.

References

Allen GI. Automatic feature selection via weighted kernels and regularization. Journal of Computational and Graphical Statistics. 2013 To Appear.

Björkström A, Sundberg R. A generalized view on continuum regression. Scandinavian Journal of Statistics. 1999; 26(1):17–30.

Chen X, Cook D. Some insights into continuum regression and its asymptotic properties. Biometrika. 2010; 97(4):985–989.

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

Cristianini, N.; Shawe-Taylor, J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press; 2000.

Efron B, Hastie T, Johnstone I, Tibshirani R. Least angle regression. The Annals of Statistics. 2004; 32(2):407–499.

Frank I, Friedman J. A statistical view of some chemometrics regression tools. Technometrics. 1993; 35(2):109–135.

Helland IS. Partial least squares regression and statistical models. Scandinavian Journal of Statistics. 1990; 17:97–114.

Kimeldorf G, Wahba G. Some results on tchebycheffian spline functions. Journal of Mathematical Analysis and Applications. 1971; 33(1):82–95.

Lin Y, Zhang HH. Component selection and smoothing in multi-variate nonparametric regression. Annals of Statistics. 2007; 34(5):2272–2297.

Mika S, Schölkopf B, Smola AJ, Müller K-R, Scholz M, Rätsch G. Kernel pca and de-noising in feature spaces. Advances in Neural Information Processing Systems. 1999; 11:536–542.

Naes T, Martens H. Comparison of prediction methods for multi-collinear data. Communications in Statistics - Simulation and Computation. 1985; 14(3):545–576.

Rosipal R, Girolami M, Trejo LJ. Kernel PCA for feature extraction and de-noising in non-linear regression. Neural Computing and Applications. 2000a; 10:231–243.

Rosipal R, Trejo LJ. Kernel partial least squares regression in reproducing kernel hilbert space. Journal of Machine Learning Research. 2001; 2(2):97–123.

Rosipal, R.; Trejo, LJ.; Cichocki, A. Kernel principal component regression with EM approach to nonlinear principal components extraction. Tech. rep. 2000b.

Schölkopf B, Smola A, Müller K-R. Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation. 1998; 10:1299–1319.

Serneels S, Filzmoser P, Croux C, Van Espen PJ. Robust continuum regression. Chemometrics and Intelligent Laboratory Systems. 2005; 76:197–204.

Shawe-Taylor, J.; Cristianini, N. Kernel Methods for Pattern Analysis. Cambridge University Press; 2004.

Stone M, Brooks RJ. Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression. Journal of the Royal Statistical Society Series B. Methodological. 1990; 52(2):237–269. with discussion and a reply by the authors.

Sundberg R. Continuum regression and ridge regression. Journal of the Royal Statistical Society. Series B (Methodological). 1993; 55(3):653–659.

Vapnik, VN. Statistical Learning Theory. Wiley-Interscience; 1998.

Walczak B, Massart D. The radial basis functions - partial least squares approach as a flexible non-linear regression technique. Analytica Chimica Acta. 1996; 331:177–185.

Wold S. Soft modeling by latent variables; the nonlinear iterative partial least squares approach. Perspectives in Probability and Statistics. Papers in Honour of M. S. Bartlett. 1975:520–540.

Zhang HH, Cheng G, Liu Y. Linear or nonlinear? automatic structure discovery for partially linear models. Journal of the American Statistical Association. 2011; 106:1099–1112. [PubMed: 22121305]

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

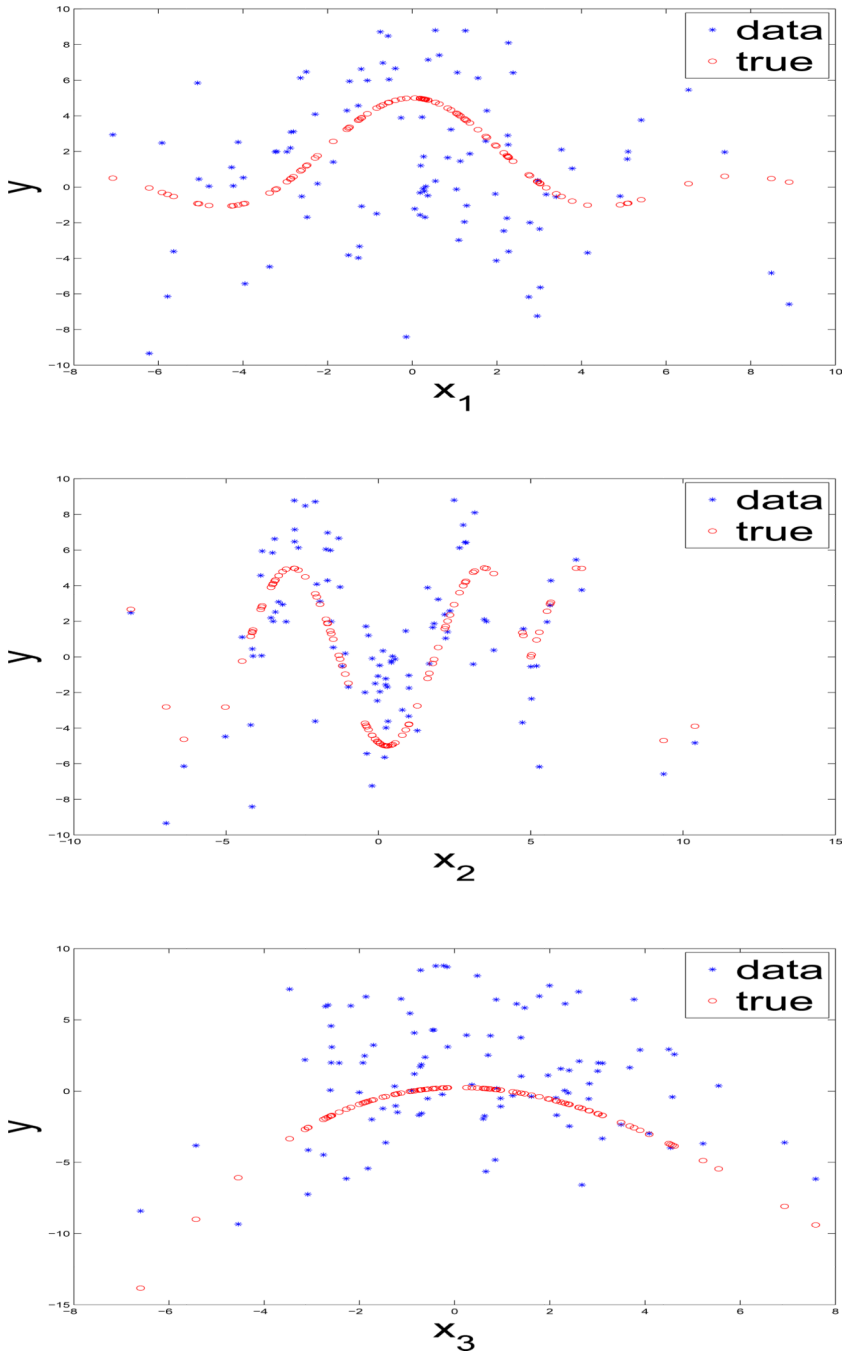

Figure 1.
Scatter plots of (X_i, Y) for Setting 2. Top: Y against X_1, Middle: Y against X_2, Bottom: Y against X_3.

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

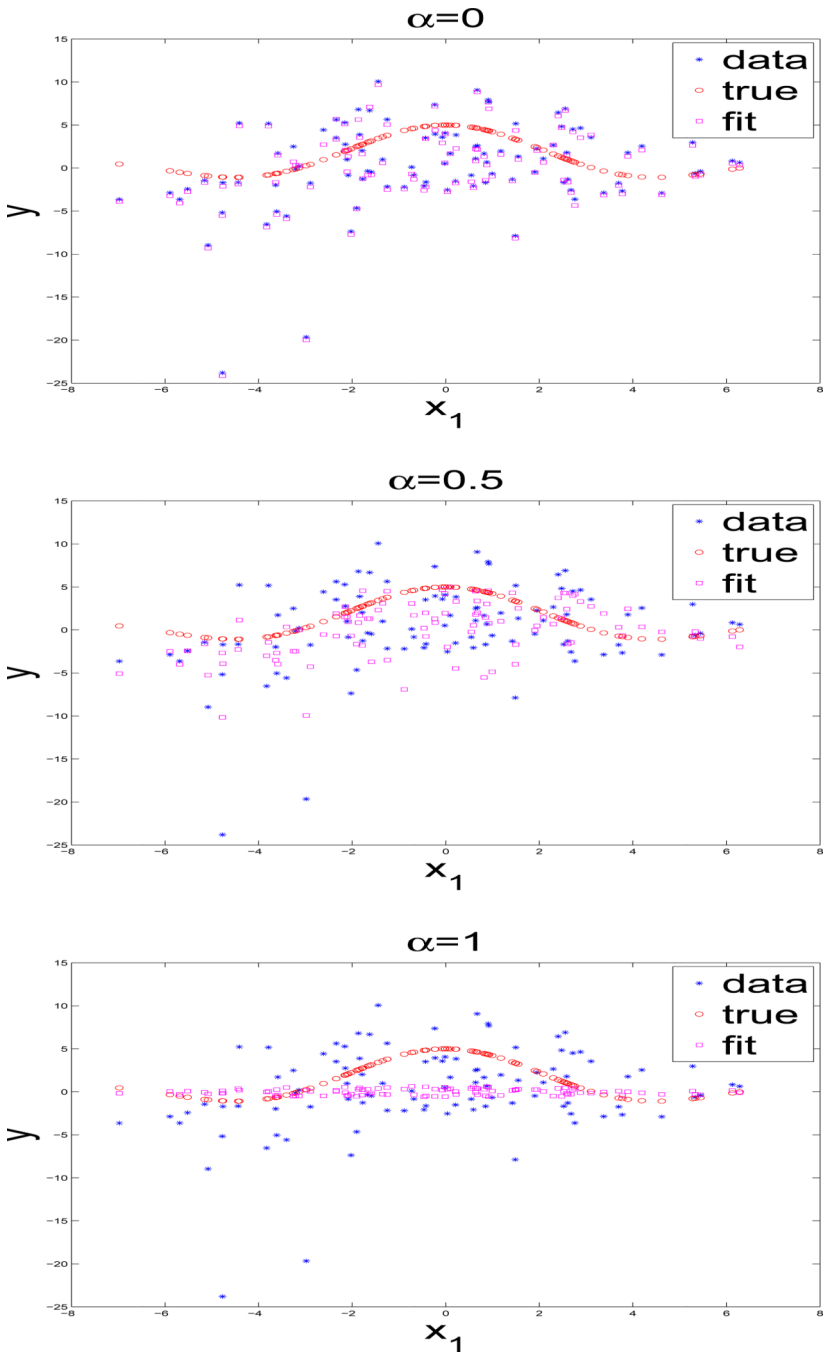

Figure 2.
Fitting results for Setting 2: fitted Y against X_1 with α = 0 (top), .5 (middle), and 1 (bottom) Gaussian kCR. The kCR with small α captures the relationship with Y.

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

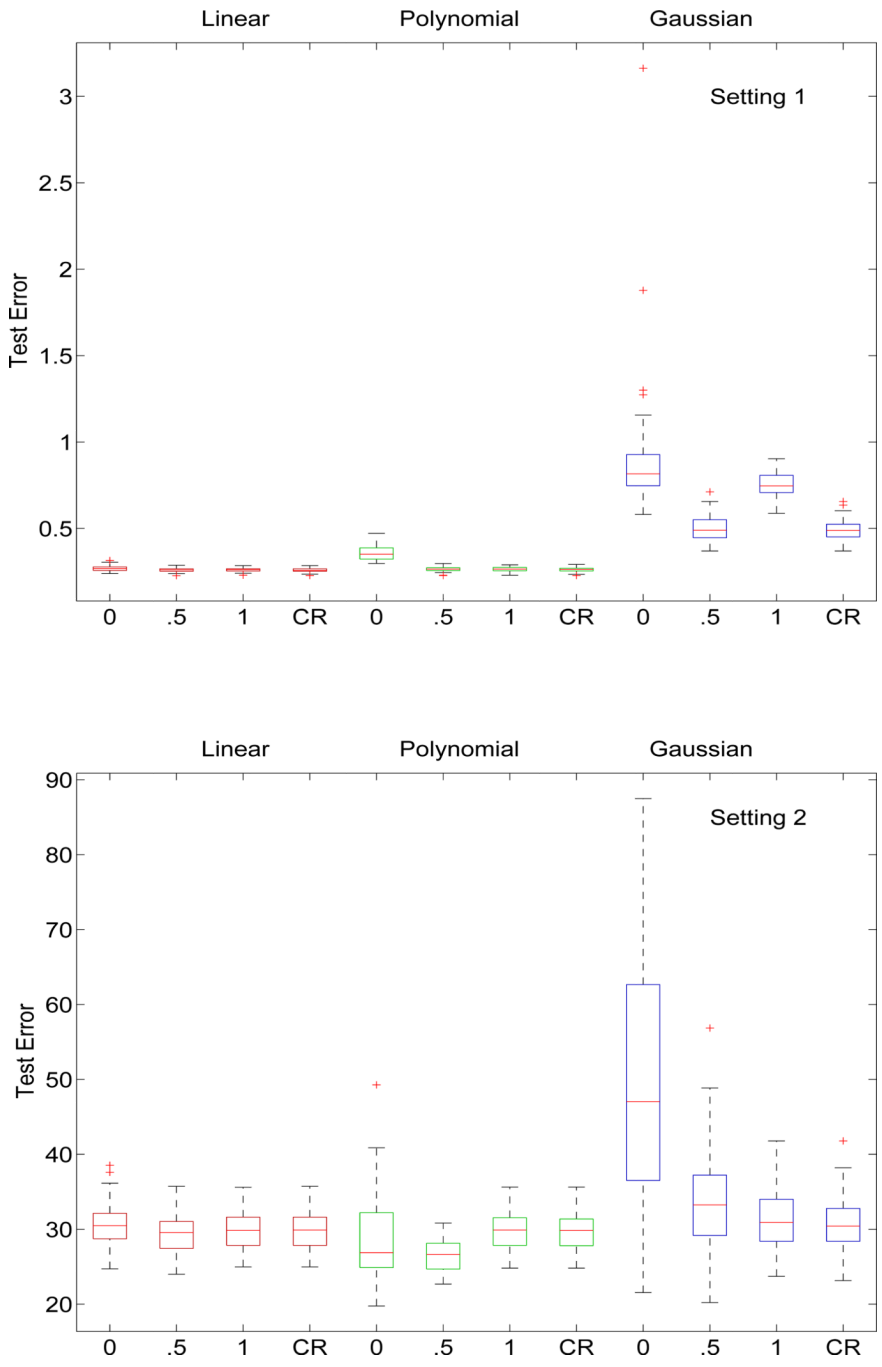

Figure 3.
Box plots of test errors for the tuned kCRs over 50 replications. Top: Setting 1. Bottom: Setting 2

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

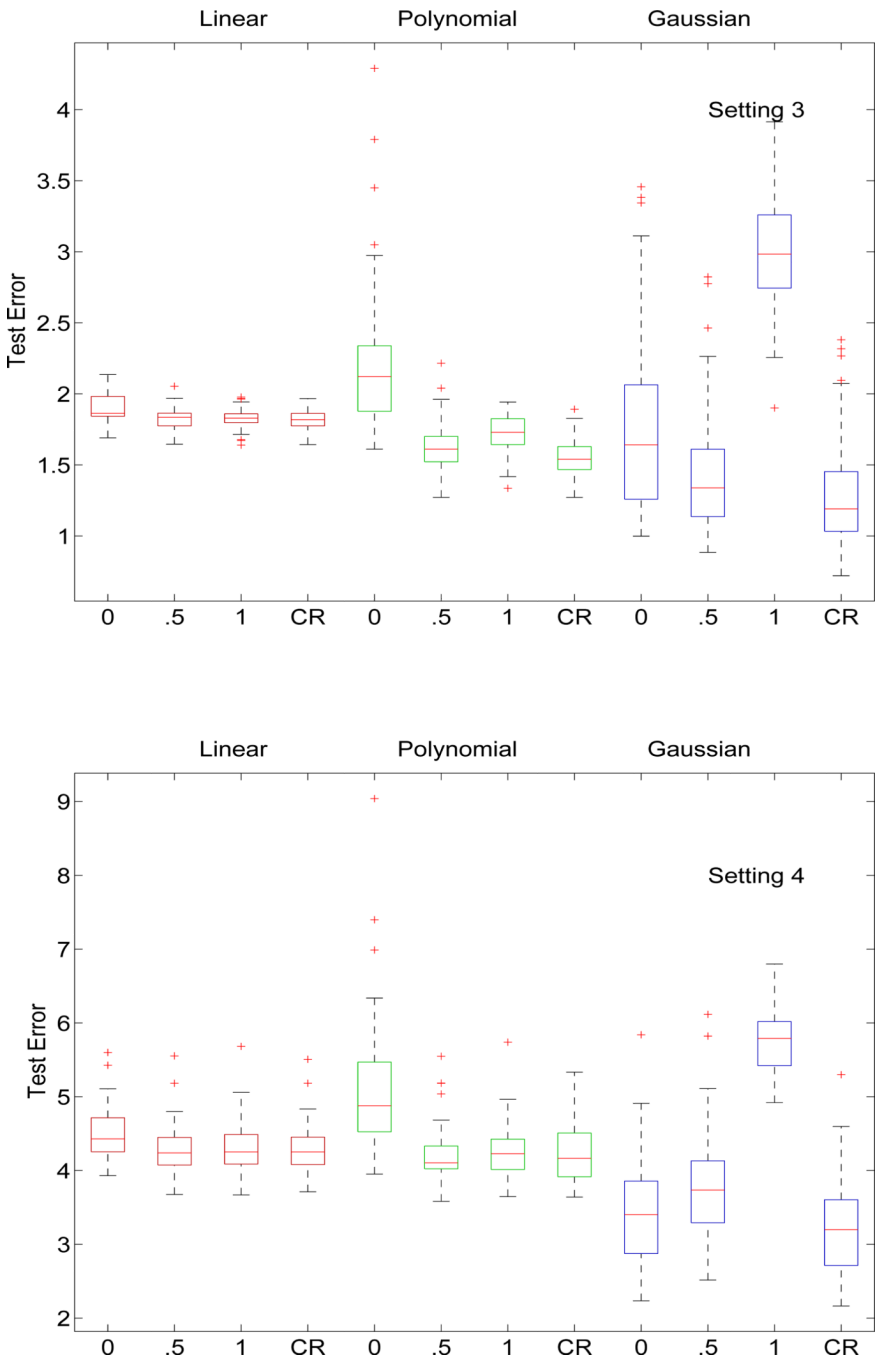

Figure 4.
Box plots of test errors for the tuned kCRs over 50 replications. Top: Setting 3. Bottom: Setting 4

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

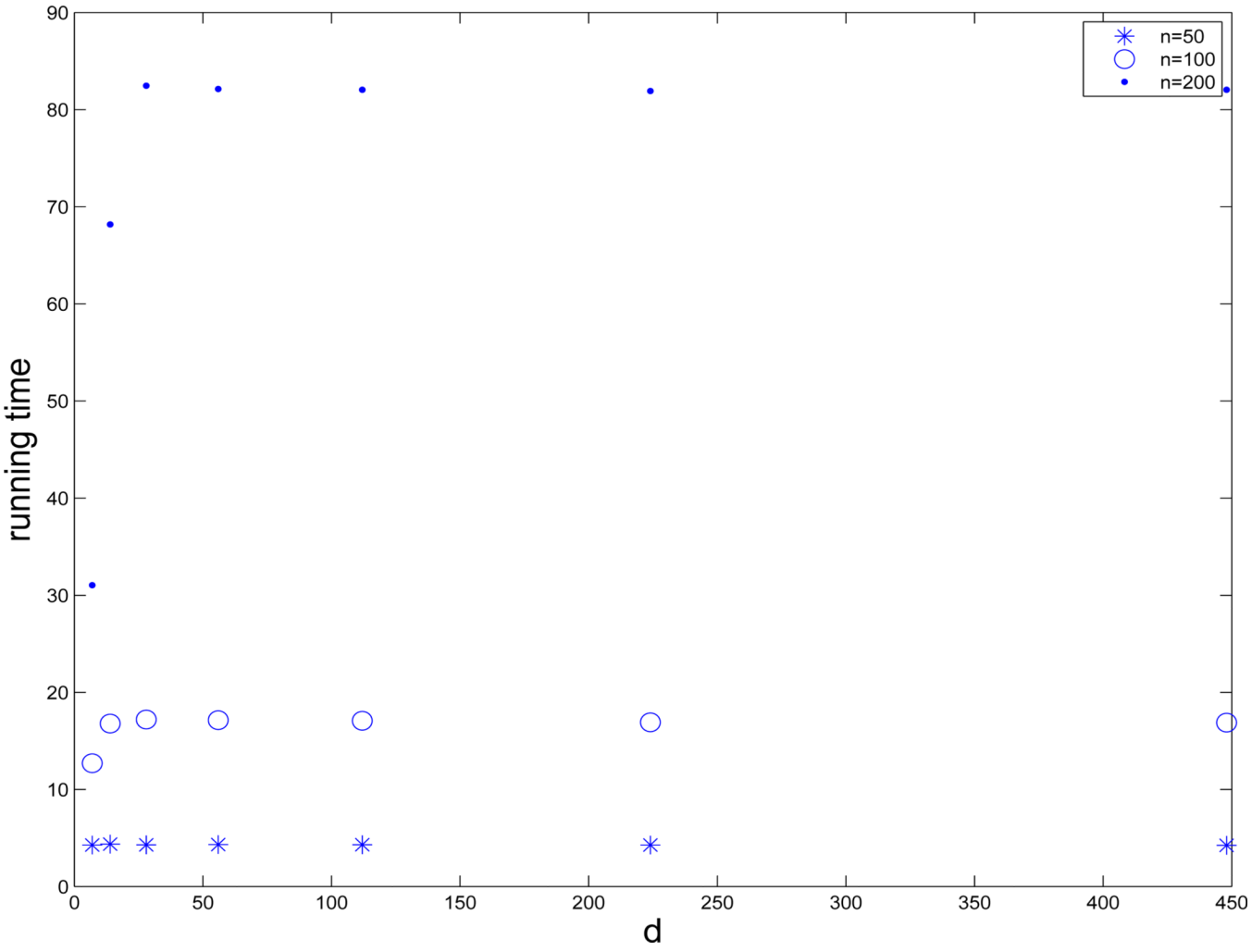

Figure 5.
Plot of running time (in seconds on *y*-axis) versus dimensionality (*d* on *x*-axis) and the sample size *n*. The symbols *, ○ and · are for $n = 50$, 100 and 200 respectively.

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

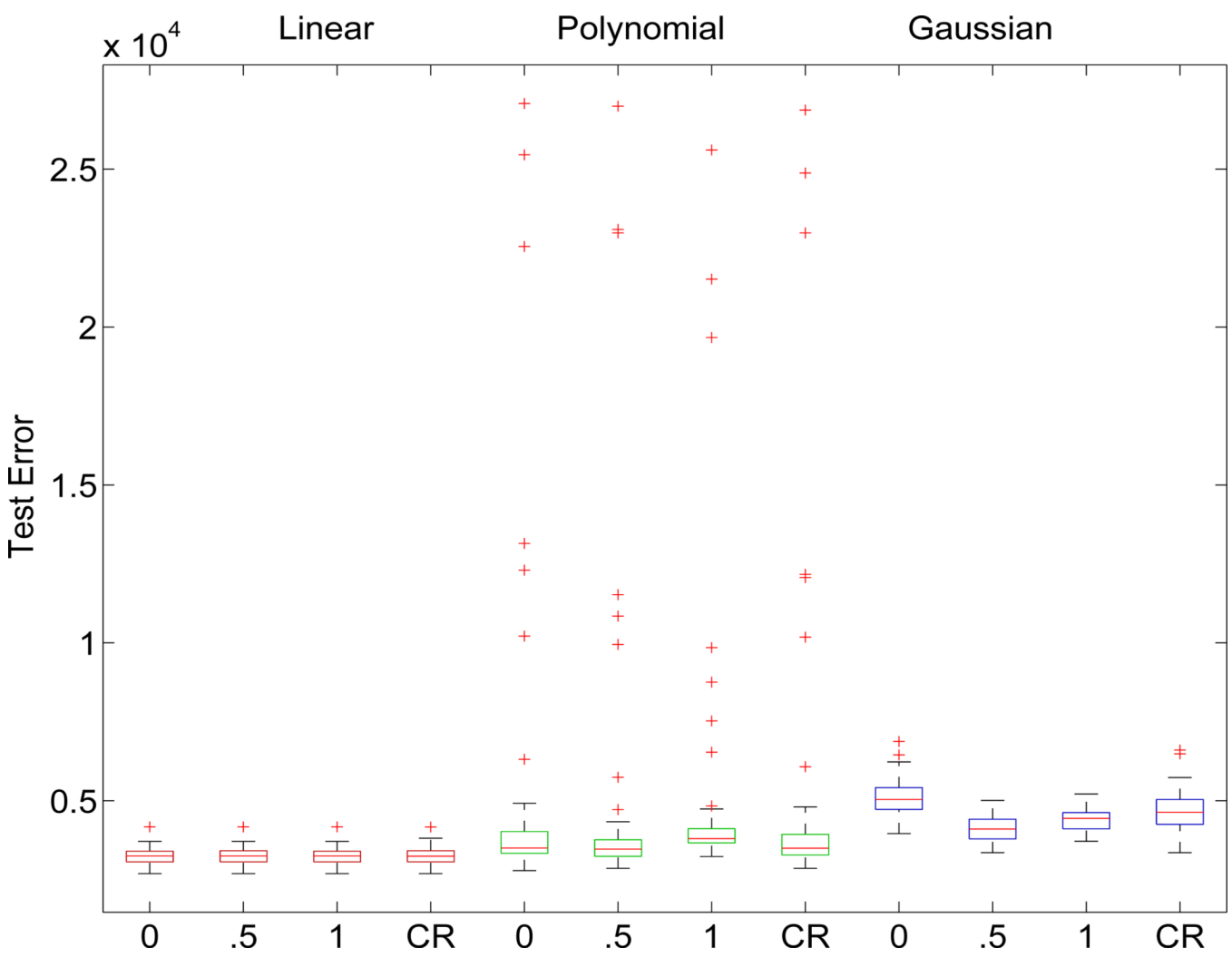

Figure 6.
Diabetes data set: Box plots of test errors for the tuned kCRs over 50 replications.

NIH-PA Author Manuscript

NIH-PA Author Manuscript

NIH-PA Author Manuscript

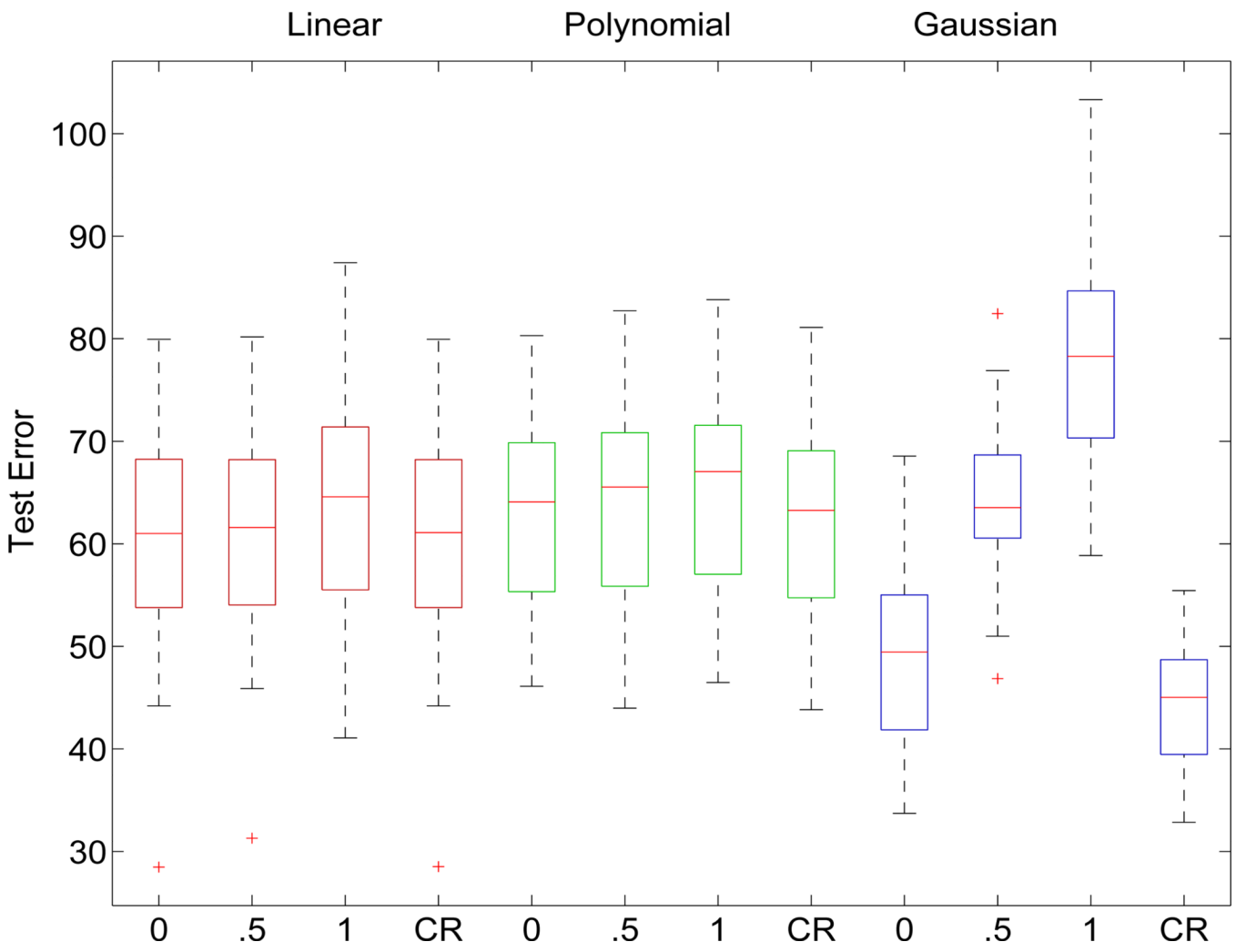

Figure 7.
Boston Housing data set: Box plots of test errors for the tuned kCRs over 50 replications.

NIH-PA Author Manuscript
NIH-PA Author Manuscript
NIH-PA Author Manuscript

Table 1

Average selected tuning parameters (α, *k*) over 50 replications. Standard deviations are reported in parentheses.

	Linear		Polynomial		Gaussian	
	α	*k*	α	*k*	α	*k*
Setting 1	0.53 (0.24)	2.38 (1.05)	0.70 (0.24)	2.98 (1.29)	0.38 (0.15)	3.18 (1.42)
Setting 2	0.90 (0.18)	2.04 (1.32)	0.98 (0.04)	1.94 (1.00)	0.88 (0.17)	2.80 (1.31)
Setting 3	0.64 (0.27)	2.38 (1.09)	0.44 (0.30)	2.84 (1.74)	0.35 (0.13)	3.54 (1.01)
Setting 4	0.63 (0.26)	2.34 (1.17)	0.54 (0.26)	3.28 (1.51)	0.27 (0.15)	3.32 (1.41)

Comput Stat Data Anal. Author manuscript; available in PMC 2014 December 01.