<table>
<thead>
<tr>
<th>Statistica Sinica Preprint No: SS-2014-0151R1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title</strong></td>
</tr>
<tr>
<td><strong>Manuscript ID</strong></td>
</tr>
<tr>
<td><strong>URL</strong></td>
</tr>
<tr>
<td><strong>DOI</strong></td>
</tr>
<tr>
<td><strong>Complete List of Authors</strong></td>
</tr>
<tr>
<td><strong>Corresponding Author</strong></td>
</tr>
<tr>
<td><strong>E-mail</strong></td>
</tr>
</tbody>
</table>
Kernel additive sliced inverse regression *

Heng Lian and Qin Wang
University of New South Wales and Virginia Commonwealth University

Abstract

In recent years, nonlinear sufficient dimension reduction (SDR) methods have gained increasing popularity. However, while semiparametric models in regression have fascinated researchers for several decades with a large amount of literature, parsimonious structured nonlinear SDR has attracted little attention so far. In this paper, extending kernel sliced inverse regression, we study additive models in the context of SDR and demonstrate its potential usefulness due to its flexibility and parsimony. Theoretically we clarify the improved convergence rate using additive structure is due to faster rate of decay of the kernel’s eigenvalues. Additive structure also opens the possibility of nonparametric variable selection. This sparsification of the kernel, however, does not introduce additional tuning parameters, in contrast with sparse regression. Simulated and real data sets are presented to illustrate the benefits and limitations of the approach.

Keywords: Kernel method; Nonlinear dimension reduction; Sliced inverse regression; Variable selection.

*Heng Lian is Senior Lecturer, School of Mathematics and Statistics, University of New South Wales, Sydney Australia 2052 (Email: heng.lian@unsw.edu.au). Qin Wang is Assistant Professor, Department of Statistical Sciences & Operations Research, Virginia Commonwealth University, USA.
1 Introduction

In the classical theory of linear sufficient dimension reduction, with a $p$-dimensional predictor $X$ and a univariate response $Y$ as in the regression setting, we say the subspace spanned by a $p \times d$ matrix $B$ with $d \leq p$ is a sufficient dimension reduction space if $Y \perp X|B^TX$, where $\perp$ denotes independence. That is, $B^TX$ summarizes the information in the predictors relevant to predicting $Y$. Under mild assumptions, the intersection of all SDR spaces is itself an SDR space and termed the central subspace (Cook, 1994, 1996, 1998; Yin et al., 2008).

Under some mild assumptions including the linear design condition, sliced inverse regression (SIR) extracts directions in the central subspace by the eigenvectors of the matrix

$$Cov(X)^{-1}Cov(E[X|Y]),$$

(1)

which can be easily estimated by slicing the range of $Y$ given a sample and hence, the name of SIR (Li, 1991). Wu (2008) and Yeh et al. (2009) extended standard SIR to nonlinear dimension reduction via the kernel method. The kernel method is a popular trick in machine learning that maps the predictors into a typically infinite-dimensional space and performs linear operation in this new feature space, which corresponds to nonlinear operation when mapped back to the original space. The kernel trick can actually be formulated via the theory of reproducing kernel Hilbert spaces (RKHS) which is more familiar in the statistical literature. Fukumizu et al. (2004), Fukumizu et al. (2009) and Fukumizu and Leng (2014) proposed the use of cross-covariance operator on RKHS to characterize the conditional independence $Y \perp X|B^TX$ so as to achieve linear dimension reduction.

As for all other fully nonparametric approaches, the kernel method also suffers from low convergence rate when $p$ is sufficiently large. In regression, semiparametric models such as those with additive structures have a long history in statistics and have been thoroughly investigated (Stone, 1985; Liang et al., 2008; Xue, 2009; Wang et al., 2011; Ma, 2012).
However, such efforts are not present in nonlinear SDR. Commonly used product/tensor kernels in multi-dimensional setting try to incorporate all-way interactions which is infeasible and uninteresting, and using such kernels is sub-optimal when additive structures are present.

In Section 2, we show that additive structures in kernel SIR can be easily realized by using the additive kernel instead of the product/tensor kernel. However, establishing theoretical advantages for doing so is nontrivial. In this paper, we establish the convergence rate for kernel SIR (KSIR) and clarify that the faster convergence rate in kernel additive SIR (KASIR) is related to the faster rate of decay in the eigenvalues of the additive kernel operator. In the special case that the RKHS is the periodic Sobolev space and that the true dimension reduction directions have an additive structure, the convergence rate is $n^{-2m/(2m+1)}$ where $m$ is the smoothness parameter of the Sobolev space, which is the well-known optimal rate in regression (for fixed dimension asymptotics). Note that unlike nonparametric regression, kernel SIR requires no optimization procedure. In Section 3, we propose a method to sparsify the additive kernel which corresponds to sparse additive models in regression for nonparametric variable selection. However, the optimization problem in sparse kernel additive SIR (SKASIR) turns out to be much harder and thus we only investigate the case $p$ is relatively small (up to about 20) compared to sample size. Another notable practical difference from sparse additive regression is that no extra tuning parameter will be introduced in SKASIR. Section 4 contains our simulation studies as well as an application to a real data set to demonstrate the performance of KASIR and SKASIR, in comparison with KSIR and standard linear SIR. We conclude the paper with a discussion on limitations and future plans. The proof of the main result is contained in the supplementary material.

2 Kernel additive SIR using additive kernel

We first review some theory of RKHS and kernel sliced inverse regression proposed in Wu (2008); Yeh et al. (2009); Wu et al. (2013). Let $L^2(P_X)$ be the space of square integrable
functions with probability measure $P_X$ on $R^p$. Given a kernel $K(.,.)$ (a positive definite function on $R^p \times R^p$) with the spectral decomposition

$$K(x, y) = \sum_{j=1}^{\infty} \lambda_j \phi_j(x)\phi_j(y),$$

(2)

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ are the eigenvalues and $\phi_j$ are the eigenfunctions orthonormal in $L^2(P_X)$, the induced RKHS is

$$\mathcal{H}_K = \{f : f(x) = \sum_j a_j \phi_j(x), \text{ with } \sum_j a_j^2/\lambda_j < \infty\}.$$ 

The inner product on $\mathcal{H}_K$ is defined to be $\langle f, g \rangle_\mathcal{H} = \sum_j a_j b_j/\lambda_j$ if $f = \sum_j a_j \phi_j$ and $g = \sum_j b_j \phi_j$. It can be shown that $\langle f, K(., x) \rangle_\mathcal{H} = f(x)$ and thus $K$ is called the reproducing kernel. An alternative and popular view in the machine learning literature for kernel is to construct the feature map $\Phi : x \to \Phi(x) = (\sqrt{\lambda_1} \phi_1(x), \sqrt{\lambda_2} \phi_2(x), \ldots) \in l_2$, where $l_2$ is the space of square-summable sequences, and thus based on (2) $K(x, y)$ is just the inner product $\langle \Phi(x), \Phi(y) \rangle_2$ in $l_2$. Using this feature map, performing standard linear SIR in the feature space $l_2$ corresponds to nonlinear SIR in the original space $R^p$. More specifically, following the SIR procedure (1), we can extract directions $\beta \in l_2$ from the eigenvalue problem

$$\text{Cov}(E[\Phi(X)|Y])\beta = \mu \text{Cov}(\Phi(X))\beta,$$

where $\text{Cov}(\Phi(X)) = E[(\Phi(X) - E[\Phi(X)]) \otimes (\Phi(X) - E[\Phi(X)])]$, for example. In the following, we note that mathematically, since by the reproducing property $\langle K(., x), K(., y) \rangle_\mathcal{H} = K(x, y)$, we can just take the feature map to be $x \to K(., x) \in \mathcal{H}_K$, and the eigenvalue problem becomes

$$\Gamma f = \mu \Sigma f,$$

(3)
where $\Gamma = \text{Cov}(E[K(.|X)|Y])$ and $\Sigma = \text{Cov}(K(.|X))$. For simplicity of notation we also assume without loss of generality that $E_X K(.|X) = 0$, where the subscript in the expectation indicates the variable over which the expectation is taken, for clarity.

Given an i.i.d. sample $(X_i, Y_i), i = 1, \ldots, n$, the relevant covariance operators can be easily estimated by slicing moment estimators as in linear SIR. $\Sigma = \text{Cov}(K(.|X)) = E_X[K(.|X) \otimes K(.|X)]$ can be estimated by $\Sigma_n = n^{-1} \sum_i (K(.|X_i) - \overline{K(.)}) \otimes (K(.|X_i) - \overline{K(.)})$ where $\overline{K(.)} = n^{-1} \sum_i K(.|X_i)$. Under $F$ the assumption $E_X K(X, X) < \infty$, $\Sigma$ has a spectral decomposition, say given by

$$
\Sigma = \sum_j \lambda_j \psi_j \otimes \psi_j,
$$

with $\lambda_1 \geq \lambda_2 \geq \cdots$ and $\langle \psi_j, \psi_k \rangle_H = \delta_{jk}$. It is important to note that with the assumption $E_X K(.|X) = 0$, by direct calculation we can see $\Sigma f = E_X K(.|X) f(X)$ and thus the eigenvalues of $\Sigma$ and $K$ (as in (2)) are the same and that is the reason we denote the eigenvalues by $\lambda_j$ as in (2). However, the eigenvectors here are chosen to be orthonormal in $\mathcal{H}$ instead of in $L^2(P_X)$. Even without the assumption $E_X K(.|X) = 0$, it can be shown that eigenvalues of $\Sigma$ and $K$ decay at the same rate which does not affect our arguments, for example in Proposition 1 below, which only depend on the rate of decay of eigenvalues.

To obtain the slicing estimator of $\Gamma = \text{Cov}(E[K(.|X)|Y])$, the range of $Y$ is divided into $H$ slices and we can estimate $\Gamma$ by

$$
\Gamma_n = \sum_{h=1}^H \hat{p}_h (\overline{K_h(.|X)} - \overline{K(.)}) \otimes (\overline{K_h(.|X)} - \overline{K(.)}),
$$

where $\overline{K_h(.|X)}$ is the average of $K(.|X_i)$ concomitant to the $Y_i$ in the $h$-th slice, and $\hat{p}_n = n_h/n$ where $n_h$ is the number of observations in the $h$-th slice. Finally, to stabilize the eigenvalue problem, a scalar multiple of the identity operator $I$ is added to $\Sigma_n$ resulting in

$$
\Gamma_n \hat{f} = \mu(\Sigma_n + sI) \hat{f}.
$$

(4)
To find the eigenfunction above, the representer theorem previously shown for kernel SIR allows us to write \( \hat{f} = \sum_{i=1}^{n} c_i (K(., X_i) - \overline{K}(., \overline{X})) \) and plugging this expression into the above displayed equation, the eigenvalue problem can be written in terms of \( c = (c_1, \ldots, c_n) \) as

\[
KJKc = \mu (K + sI)c,
\]

which is used for computation, where \( K \) is the centered \( n \times n \) kernel matrix, \( J \) is the \( n \times n \) matrix with \( J_{ij} = 1/n_h \) if \( Y_i \) and \( Y_j \) are in the \( h \)-th slice and zero otherwise.

By the representer theorem, it is easy to see immediately that if we use kernel \( K \) that has an additive form, that is \( K(x, y) = K_1(x_1, y_1) + \cdots + K_p(x_p, y_p) \) for \( x = (x_1, \ldots, x_p) \), \( y = (y_1, \ldots, y_p) \) and \( p \) kernels \( K_1, \ldots, K_p \), then \( f \) (as well as all functions in \( H_K \)) also has this additive form. Theoretically, we need to assume the true \( f \) which are defined as the eigenfunction in (3) is in the RKHS generated by \( K(x, y) = K_1(x_1, y_1) + \cdots + K_p(x_p, y_p) \). This is equivalent to saying that one can write \( f(x) = f_1(x_1) + \cdots + f_p(x_p) \) and \( f_j \) is in the RKHS generated by \( K_j \).

Although it is trivial to incorporate additive structure into KSIR as explained above, it is nevertheless difficult to see how this additively structured kernel induces faster convergence rate. Wu et al. (2013) has shown the consistency of KSIR but a meaningful convergence rate remains elusive. The following main result in this paper clarifies the role of the kernel’s eigenvalues \( \lambda_1, \lambda_2, \ldots \), in determining the convergence rate. We first present the theorem for general kernels and then discuss its implication for the faster convergence rate for additive kernels. The following assumptions are used.

(A) \( \exists C > 0 \) such that \( K(x, x) < C \) for all \( x \) in the range of the predictor.

(B) \( \lambda_j \approx j^{-d} \) for some \( d > 1 \).

(C) The operator \( \Sigma^{-1} \Gamma \) has an eigenfunction \( f \) associated with its largest eigenvalue \( \mu \).

We also assume the multiplicity of the eigenspace associated with \( \mu \) is one so the
eigenfunction is identified up to sign. Similarly, let \( \hat{f} \) be the eigenfunction of \((\Sigma_n + sI)^{-1}\Gamma_n \) associated with its largest eigenvalue.

(D) The response is discrete and can take only values in \( \{y_1, \ldots, y_H\} \).

(E) Suppose the SDR space is generated by \( \{h_1, \ldots, h_r\} \subset \mathcal{H}_K \). We assume the linear design condition that \( E(g(x)|h_1(x), \ldots, h_r(x)) \) is linear in \( h_1(x), \ldots, h_r(x) \) for \( g \in \mathcal{H}_K \).

The assumption of uniform boundedness of \( K \) in (A) is required in our proof to show \( K(.,x) \otimes K(.,x) \) is a bounded operator for all \( x \). When we consider the range of \( x \) in a compact set as when we use the Sobolev kernel defined on \([0,1]\) later, assumption (A) is a very mild regularity assumption of the kernel (for example it is satisfied as long as the kernel is continuous). It is also satisfied by the Gaussian kernel. It also implies that \( \sum_j \lambda_j < \infty \). Assumption (B) imposes that the eigenvalues of the kernel decay with a polynomial rate. Assumption (B) was used in Blanchard et al. (2008); Caponnetto and De Vito (2007) to establish oracle inequalities for support vector machines classification and regression respectively. That eigenvalues play a critical role in convergence rate is expected in regression since the Rademacher complexity of the RKHS can be exactly characterized by these eigenvalues (Bartlett and Mendelson, 2003; Koltchinskii and Yuan, 2010; Raskutti et al., 2012). It is known that the polynomial decay assumption holds in some special cases as in section 1.1 of Koltchinskii and Yuan (2010). If this assumption does not hold, we can still possibly derive some rate in terms of the specific values of \( \lambda_1, \lambda_2, \ldots \). However, the expression would be messy and it would be hard to see the effect of eigenvalues on the convergence rate. With polynomial decay assumption \( \lambda_j \sim j^{-d} \), it can be seen clearly by our theoretical results that larger \( d \) leads to faster convergence rates. Assumption (C) simply re-states the estimator and the population counterpart for clearness. It can be shown that under our assumptions, the eigenspace of \((\Sigma_n + sI)^{-1}\Gamma_n \) associated with its largest eigenvalue also has multiplicity one with probability approaching one. For simplicity, we only consider the first dimension reduction direction which is the eigenfunction associated with the largest
eigenvalue. Rates for subsequent directions can be shown with some additional arguments. Assumption (D) is typically assumed in the SIR literature, for example in Cook and Ni (2005) to simplify analysis, which directly applies to classification problems and is also reasonable for regression since the slicing estimator will in effect quantize the responses. Assumption (E) is used in Wu (2008); Yeh et al. (2009); Wu et al. (2013). (E) implies that
\[ E[K(.,X)|Y = y_h] \in \text{span}\{\Sigma h_1, \ldots, \Sigma h_r\} \]
and in particular that \( \Sigma^{-1} \Gamma \) is a bounded operator which will be used in the proof.

**Theorem 1** Under assumptions (A)-(E), and let \( s = c_n n^{-d/(d+1)} \to 0 \) with \( c_n \to \infty \), we have

\[
\min_{c \in \{-1,1\}} E_{X^*}[(cf(X^*) - f(X^*))^2] = O_p(c_n n^{-d/(d+1)}),
\]
where \( X^* \) is an independent copy of \( X \).

We note that the constant \( c \) is necessary since the eigenfunction can only be identified up to sign change. This rate is optimal in regression up to an extra arbitrarily slowly diverging sequence \( c_n \). This might be due to our method of the proof but it is not clear to us how to get rid of this term.

From the above result, we can obtain improved convergence rate if the kernel’s eigenvalue has a fast decay to zero. For kernel methods, a commonly used kernel is the Gaussian kernel \( K(x,y) = \exp\{-a\|x - y\|^2\} \) or the more flexible form \( K(x,y) = \exp\{-\sum_{j=1}^{p} a_j (x_j - y_j)^2\} \).

More generally, kernels on a multi-dimensional domain can be constructed by a product/tensor operation from one-dimensional kernels with \( K(x,y) = \prod_{j=1}^{p} K_j(x_j, y_j) \). Another common example from the smoothing splines is the Sobolov space of order \( m \) with kernel given by \( K(s,t) = \sum_{\nu=1}^{m}(s-1)^{\nu-1}(t-1)^{\nu-1}/((\nu-1)!)^2 + \int_0^1 (s-u)^{m-1}(t-u)^{m-1}/((m-1)!^2)du, \) and the multivariate version is constructed by taking product of one-dimensional kernels. For \( m \)-th order Sobolev space of periodic functions, it is known that the eigenvalues decay at the rate \( j^{-2m} \).
Let $\mathcal{H}_j$ be the RKHS induced by one-dimensional kernel $K_j$. Suppose $\phi_{j1}, \phi_{j2}, \ldots$, are the eigenfunctions of $K_j$ with eigenvalue $\lambda_{jk} \asymp k^{-d}$. Let $K^{(p)}(x, y) = \prod_j K_j(x_j, y_j)$ and $K^{(s)}(x, y) = \sum_j K_j(x_j, y_j)$. The following simple proposition shows that the eigenvalues of $K^{(s)}$ decay at the same rate as each $K_j$ when the coordinates are independent.

**Proposition 1** Under the setup as described above, and assuming that the $p$ predictors are independent, the eigenvalues of $K^{(s)}(x, y)$ are of order $j^{-d}$.

**Proof.** The RKHS associated with $K^{(s)}$ is the space of functions of the form $\sum_{j=1}^p f_j(x_j)$ (Aronszajn, 1950) with $f_j \in \mathcal{H}_j$. Using $E_X f_j(X) = 0$, it is easy to see that $K f = (\sum_j K_j)(\sum_j f_j) = \sum_j K_j f_j$. Thus if $f = \sum_j f_j$ satisfies the eigenvalue equation $K f = \lambda f$, we have $K_j f_j = \lambda f_j$, which implies that for each $j$, either $f_j$ is an eigenvector of $K_j$ with (common) eigenvalue $\lambda$, or $f_j = 0$. This in turn means that the set of eigenvalues of $K$ is a subset of $\{\lambda_{jk}\}$ and the multiplicity of each eigenspace is at most $p$. Thus the decay rate of the eigenvalues of $K$ is the same as that of its additive component. \(\square\)

In a $p$-dimensional space, when the function $K(., y)$ is in the Hölder class of smoothness $2m - p$ for all $y$, eigenvalues of a positive definite kernel is upper bounded by $j^{-2m/p}$ under mild assumptions and a kernel can be constructed that achieves this rate of decay (Kühn, 1987). An illustration of this result is the $m$-th order Sobolev space of period functions on $[0, 1]$ whose kernel has smoothness $2m - 1$ with its eigenvalues known to exactly decay as $j^{-2m}$. In general, although it is unclear whether the rate of decay $j^{-2m/p}$ is achieved by a specific kernel, it is generally believed that $j^{-2m/p}$ is the typical rate, which leads to convergence rate of KSIR of $n^{-2m/(2m+p)}$, the same as the minimax rate for nonparametric regression in a $p$-dimensional space.

In particular, it seems hard to infer the eigenvalues of the product kernel $K^{(p)}$ based on the eigenvalues of $K_j$, although it is natural to conjecture that the eigenvalues of $K^{(p)}$ decay at a slower rate than those of $K_j$. To illustrate this numerically, we generate 100 sets of predictors with sample size $n = 300$ and $p = 5$ independently and uniformly distributed
Figure 1: Logarithms of eigenvalues for the product (black and solid curve) and additive kernel (red and dashed curve). Left: $p = 5$; Right: $p = 10$.

on $[0,1]^p$. We then construct the $n \times n$ kernel matrices of both the product type and the additive type using the kernel for the 2nd order Sobolev space. The averaged (over 100 data sets) logarithm of the largest 200 eigenvalues for the two kernel matrices are shown in the left panel of Figure 1. To facilitate comparison the eigenvalues are scaled such that the largest eigenvalue is always one. It is seen that the eigenvalues of the additive kernel (red and dashed curve) decay faster than the product kernel. The right panel of the same figure shows the results with $p = 10$. With larger $p$, the gap between the two curves visually becomes larger. Based on our theoretical result, the faster decay of the eigenvalues of the additive kernel will lead to a faster convergence rate, if the additive assumption is valid.

3 Sparse kernel additive SIR

The additive kernel used in the previous section is $K(x, y) = K_1(x_1, y_1) + \cdots + K_p(x_p, y_p)$ which is equally weighted. To take into account the different importance of different predictors, we can add a nonnegative weight to each component such that $K(x, y) = d_1 K_1(x_1, y_1) + \cdots + d_p K_p(x_p, y_p)$, where a larger weight $d_j$ roughly implies the more important role of the $j$-th predictor. In the extreme case, $d_j = 0$ will remove the $j$-th predictor from the model.
and thus perform variable selection in this nonparametric setting. We will use a data-driven procedure to determine the weights along with the sufficient dimension reduction space.

The variable selection problem for KASIR is fundamentally different from various sparse sliced inverse regression methods proposed previously for linear SDR (Li and Nachtsheim, 2006; Li, 2007; Bondell and Li, 2009; Chen et al., 2010). In linear SDR, sparse method naturally imposes sparsity on the eigenvector of an appropriately defined eigenvalue problem. In KASIR, although the SDR space is also estimated from the eigenvalue problem (5), sparsity of the eigenvector $c$ in (5) does not perform variable selection at all.

Rather, the form of the additive kernel with weights $d_1 K_1(x_1, y_1) + \cdots + d_p K_p(x_p, y_p)$ is similar to that used in multiple kernel learning, with the slight difference being that usually multiple kernel learning focuses on obtaining a good kernel for prediction rather than variable selection and thus the kernels are not necessarily defined on different predictors. Our additive kernel with weights is also related to COSSO (Lin and Zhang, 2006; Zhang, 2006; Storlie et al., 2011) which is an approach for component selection in additive splines or more generally splines analysis of variance. Although many kernel algorithms have been demonstrated to be able to incorporate kernel learning, for KASIR this poses much difficulty in computation.

It is well-known that the quotient trace problem

$$\max_{C \in \mathbb{R}^{p \times r}, \{d_j\}} \text{tr}(C^T K (K + sI) C^{-1} C^T K J C),$$

is solved by the eigenvalue problem $K J C_i = \mu_i (K + sI) C_i$ associated with the largest $r$ eigenvalues, where $C_i, i = 1, \ldots, r$ are the columns of $C$. Thus when weights are incorporated
in the kernel, we can solve the following

\[
\max_{C \in \mathbb{R}^{p \times r}, \{d_j\}} \text{tr}((C^T K(K + sI)C)^{-1}C^T K J K C) \\
\text{s.t. } \sum_j d_j = \tau, d_j \geq 0,
\]

where the kernel matrix \( K = \sum_j d_j K_j \) implicitly depends on \( d_j \). Unlike lasso problem where \( \tau \) is treated as a tuning parameter (Tibshirani, 1996), here we can set the bound to be just 1. Intuitively, when all \( d_j \) are multiplied by a positive constant, the RKHS generated does not change. Mathematically, it is straightforward to see that the constrained maximization problem with constraint \( \sum_j d_j = \tau \) and smoothing parameter \( s \) is the same as the problem with constraint \( \sum_j d_j = 1 \) and smoothing parameter \( s/\tau \), in the sense that the maximizer \( C \) and the maximum value are the same. Since we will choose the smoothing parameter \( s \) in the data-driven way, there is no loss of generality in setting \( \tau = 1 \). We also note that although it is probably more flexible to use two tuning parameters for controlling smoothness and variable selection separately, using one tuning parameter for both is also not rare in additive regression, as in COSSO (Lin and Zhang, 2006; Storlie et al., 2011).

Given \( \{d_j\} \), \( C \) can be obtained from the eigenvalue problem. However, given \( C \), the optimization problem is neither convex nor concave and finding \( \{d_j\} \) is hard. We use a general nonlinear solver for optimization over \( \{d_j\} \) after \( C \) is profiled out. More specifically, for given \( \{d_j\} \), we can solve the eigenvalue problem to get \( C \), written as \( C(\{d_j\}) \) to emphasize the dependence on \( \{d_j\} \). We then use a general nonlinear solver for the following problem

\[
\max_{d_{j,j=1...p}} \text{tr}((C(\{d_j\})^T K(K + sI)C(\{d_j\}))^{-1}C(\{d_j\})^T K J K C(\{d_j\})) \\
\text{s.t. } \sum_j d_j = 1, d_j \geq 0.
\]

Another slightly different implementation strategy is to alternatingly solve \( C \) for given \( \{d_j\} \)
and solve \( \{d_j\} \) for given \( C \) (again using a general nonlinear solver). However, this alternating algorithm has trouble in achieving convergence in our experience.

4 Numerical examples

4.1 Simulations

Three simulation examples are used to compare five methods including KSIR, KASIR, SKASIR and standard linear SIR and SSIR, a sparse version of linear SIR. The data are generated from the following three models.

1. \( Y_i = \frac{20 \sin(X_{i1}X_{i2})}{1 + \exp\{-3X_{i3}\}} + \epsilon_i, i = 1, \ldots, n \)

2. \( Y_i = 10(\sin(3X_{i1}) + X_{i2})\log(|\sin(3X_{i1}) + X_{i2}|) + \epsilon_i, i = 1, \ldots, n \)

3. \( Y_i = (1.5X_{i1} + 2X_{i2} - X_{i3}) \exp\{1.5X_{i1} + 2X_{i2} - X_{i3}\} + \epsilon_i, i = 1, \ldots, n \)

where \( \epsilon_i \sim \mathcal{N}(0, 1) \). The predictors are generated from a multivariate Gaussian distribution with mean zero and covariance \( \text{Cov}(X_{ij1}, X_{ij2}) = 0.2^{|j_1-j_2|} \) and then apply the standard normal cdf to transform them to the interval \([0, 1]\). The first example is a general nonlinear model. The second example has an additive structure while being nonlinear and KASIR is expected to perform well. The third example is linear.

We set the sample size to be \( n = 50 \) or 100, with dimension \( p = 10 \). We use the kernel for the 2nd order Sobolev space. KSIR uses the product of these one-dimensional kernels and KASIR/SKASIR uses the sum of these one-dimensional kernels. Nonlinear optimization in SKASIR is implemented using the nloptr package in R. Although linear SIR and SSIR could be implemented in the more traditional way and in particular there are several existing methods for sparse SIR, we actually treat linear SIR/SSIR as a special case of KSIR using the linear kernel \( K(s, t) = 1 + st \). This is mainly due to convenience for our implementation since we need to implement KSIR anyway. The smoothing provided by KSIR could also be
advantageous even in linear SIR. For the smoothing parameter $s$, 15 equally spaced value in $[-7, 5]$ are used on the logarithmic scale. Finally, the number of slices is set to be 10 in all numerical examples and generally the results are not sensitive to any reasonable choice of the number of slices.

To quantify the performance of different methods, we generated independent test data of size $n$ and computed the (absolute value of ) Spearman correlation between the estimated index and the response on the test data. This is possible since in all simulated examples we only have one index. The whole procedure is repeated 100 times in each scenario. The results are reported in Figures 2-4. In these figures, the first row shows the average Spearman correlations on test data for the five methods (KSIR, KASIR, SKASIR, SIR, SSIR, in color black, red, green, blue, magenta, respectively), with dotted curves showing the 0.1 and 0.9 quantiles over the 100 repetitions. The x-axis of this plot is the logarithm of the smoothing parameter $\log(s)$. The second and third row shows the values of $d_j$ for SKASIR and SSIR, respectively, using the smoothing parameter that achieves the largest correlation value.

For Example 1, the model does not have additive structure, thus it is somewhat surprising to see that when $n = 50$ both SSIR and SKASIR outperform KSIR, with SIR and KASIR (without variable selection) performing similarly to KSIR in terms of Spearman correlation. This actually suggests that the nonlinearity in Example 1 is not sufficiently strong and with small sample size, more parsimonious models can still be very competitive even though the model assumption is wrong. When $n = 100$, all methods become similar. We further performed simulations for this example with $n = 50$ and $p = 5$ with results shown in Figure 5. With a smaller dimension, we expect KSIR suffers less from curse of dimensionality and indeed the results demonstrate that with $p = 5$ KSIR has the best performance. Returning to Figure 2, both SKASIR and SSIR put large weights on the first two predictors, followed by the third predictor. This demonstrated actually that the contribution of the third predictor to the response is relatively small, as expected from the form of the regression function in Example 1, since the exponential function does not vary a lot on $[0, 1]$. Although the weights
$d_j$ for $j > 3$ can be shrunk to exactly zero in some cases, the kernel produced is often not sparse enough especially when $n = 100$. On the other hand, even when the weights are not exactly zero, the weights for irrelevant predictors are generally much smaller and thus can still provide information on variable importance.

Example 2 represents an ideal scenario for KASIR/SKASIR, which outperformed other methods. Sparse methods can also improve on its non-sparse counterpart, for both additive and linear SIR. Additive models correctly identify first two predictors as important, while linear methods only identify the second predictor. This is a natural behavior since in Example 2, the first predictor has a strong nonlinear effect in the index (and designed such that it has no obviously increasing or decreasing trend for $X_1 \in [0, 1]$) and thus it is hard for the linear method to pick out this predictor. Furthermore, SKASIR can select a much sparser model while in SSIR none of the weights are sufficiently close to zero. This can be explained by that since linear model is not sufficiently flexible, all predictors strive to compensate for this by playing some roles in prediction.

Example 3 has a linear index and thus linear standard SIR performs well and sparsity can also improve the performance to a small extent. However, we note that KASIR/SKASIR performs almost the same as linear methods. In particular, the curve for SKASIR follows closely that for SSIR and the curve for KASIR follows closely that for SIR. KSIR is the worst performer in this example. Both SKASIR and SSIR separate the first three predictors as important, although weights for SKASIR are somewhat less sparse.

For Example 2, we also carried out simulations with $n = 100$ and $p = 20$ with results shown in Figure 6. The results are qualitatively very similar to the case $p = 10$. The additive model KASIR/SKASIR has the best performance and the important predictors are correctly identified.
Figure 2: For Example 1, the Spearman correlation on test data for the five methods (KSIR, KASIR, SKASIR, SIR, SSIR, in color black, red, green, blue, magenta, respectively) is shown in the first row. The second (third) row shows the values of $d_j$ for SKASIR (SSIR).
Figure 3: For Example 2, the Spearman correlation on test data for the five methods (KSIR, KASIR, SKASIR, SIR, SSIR, in color black, red, green, blue, magenta, respectively) is shown in the first row. The second (third) row shows the values of $d_j$ for SKASIR (SSIR).
Figure 4: For Example 3, the Spearman correlation on test data for the five methods (KSIR, KASIR, SKASIR, SIR, SSIR, in color black, red, green, blue, magenta, respectively) is shown in the first row. The second (third) row shows the values of $d_j$ for SKASIR (SSIR).
Figure 5: For Example 1 with $n = 50$ and $p = 5$, the Spearman correlation on test data for the five methods (KSIR, KASIR, SKASIR, SIR, SSIR, in color black, red, green, blue, magenta, respectively) is shown in the first row. The left (right) panel on the second row shows the values of $d_j$ for SKASIR (SSIR).
Figure 6: For Example 2 with $n = 100$ and $p = 20$, the Spearman correlation on test data for the five methods (KSIR, KASIR, SKASIR, SIR, SSIR, in color black, red, green, blue, magenta, respectively) is shown in the first row. The left (right) panel on the second row shows the values of $d_j$ for SKASIR (SSIR).
4.2 NMMAPS data analysis

We use the NMMAPS (National Morbidity Mortality Air Pollution Study) database which contains daily mortality, weather and pollution data for 1987-2000. Here we only consider data for the year 1997. We will explore the relationship between daily mean ozone level and some predictors. The selected 7 explanatory variables include mean temperature, relative humidity, mean $CO_2$ level, mean $PM_{10}$ level, mean $SO_2$ level, daily humidity range, and daily temperature range. After excluding one day with missing observations, we have a sample size of $n = 364$. Scatterplot of the daily mean ozone level against the mean temperature in Figure 7 clearly shows some nonlinearity in the data, although this observation by itself does not mean nonlinear dimension reduction is more appropriate than linear dimension reduction.

For real data, it is harder to assess the performance of different methods. We randomly partition the whole data set into a training part and a testing part of equal sizes. We perform dimension reduction using the five methods on the training data, with a sequence of smoothing parameters as used in simulations, and consider the number of indices $r$ (projection directions) from 1 to 4. Gaussian process regression is then used to learn a nonparametric function that maps the values of the index to the response. We use the R package tpg for Gaussian process regression with the default parameters choices. We also
Table 1: Prediction MSE for the five SDR methods for the NMMAPS data.

<table>
<thead>
<tr>
<th>Method</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSIR</td>
<td>0.591</td>
</tr>
<tr>
<td>KASIR</td>
<td>0.592</td>
</tr>
<tr>
<td>SKASIR</td>
<td>0.595</td>
</tr>
<tr>
<td>SIR</td>
<td>0.608</td>
</tr>
<tr>
<td>SSIR</td>
<td>0.611</td>
</tr>
</tbody>
</table>

used the R package rpart for regression tree to fit nonparametric regression function with default parameters choices. However, the predictions with trees are generally worse than Gaussian process regression and thus we choose Gaussian process regression even though it is much slower. Using the estimated index and regression function, prediction mean squared errors (PMSE) on the test data is reported on Table 1 based on 100 random partitions of the data. The errors reported are based on the pair of \((r, s)\) values that produce the smallest error for each method. It is seen that KSIR and KASIR perform similarly, and better than linear methods. Kernel weights do not help in prediction in this data set although it may help in interpreting the importance of different predictors. The average kernel weights for SKASIR are \((0.375, 0.111, 0.171, 0.038, 0.051, 0.132, 0.122)\) showing that the most important predictor appears to be the mean temperature. We show the estimated component functions from the first dimension reduction direction for one of the 100 runs in Figure 8.

5 Conclusion and discussion

In this paper we considered kernel additive sliced inverse regression and its sparse version that can perform variable selection. The advantages of the additive structure come from the fast eigenvalue decay rate for the additive kernel compared to general kernels in multi-dimensional case. We showed via numerical studies that KASIR is flexible, parsimonious and reliable and SKASIR can further identify important predictors, a goal that fully nonparametric KSIR method cannot achieve. Although we do not consider a partially linear structure, this is straightforward by just using a linear kernel in the additive combination of kernels whenever the predictor is in the linear part. In particular, in this way we can deal with both continuous
Figure 8: Estimated component functions from the first dimension reduction direction for one of the 100 runs.
and discrete predictors simultaneously.

Due to the necessity of using general-purpose nonlinear optimization software, the computation of SKASIR may be too slow and unstable for high dimensional problems, and in particular we cannot carry out simulations with \( p \) much larger than 20 in R due to computational speed constraints. New formulations or new algorithms need to be proposed for it to work in problems with higher dimensions. In our formulation, given \( C \), the optimization problem for \( d_j \) can actually be posed as quadratically constrained quadratic programming (QCQP), which is well-known to be NP-hard in general. Although previously some QCQP problems can be solved by semidefinite programming after relaxation, the nonconvexity of our formulation prevents any easy route for doing so and we are not successful so far in this direction after some try. Asymptotic property of the sparse estimator is not established here due to the technical challenges and is worth further investigation.

In the simulations, it is seen that the estimated weights are typically not sparse enough and although some weights are small, they are not always shrunk sufficiently to be treated as zero. One can replace the constraint \( \sum_j d_j = 1 \) by \( \sum_j d_j^q = 1 \) with \( 0 < q < 1 \) which can potentially lead to a sparser solution, as have been demonstrated in regression (Huang et al., 2008).

In this paper, we only used one-dimensional Sobolev kernel as the building block of product and additive kernels. This is partly due to that kernel choice is a challenging topic in itself and it is hard to tune the kernel for different data sets. For kernel methods, Gaussian kernel is popularly used and gives satisfactory performance in various problems. However, we do not use Gaussian kernel for the following reason. In practice, the choice of the bandwidth parameter in the Gaussian kernel is critical for its performance and it is not clear how to choose these parameters in an efficient way. Even though some bandwidth selection methods could be used, based on heuristics or not, this extra complication would disturb the comparison between different methods. A good bandwidth choice method for the product kernel may not be good for the additive kernel, and vice versa, which makes
comparison between different methods complicated and dirty. Thus we use the Sobolev kernel for which there is no hyperparameters to choose. Careful treatment for kernels with hyperparameters needs further detailed investigation. Another related problem is automatic kernel choice. For example, in simulation example 1 we observe that if nonlinearity is not strong enough, a linear kernel may be prefered. Selection/combination of different kernels is an interesting direction for future research.

We note that a general and elegant theory of nonlinear SDR is reported in Lee et al. (2013). By using the definition of SDR $\sigma$-field to replace the concept of SDR space, linear design condition is not necessary for nonlinear SDR and they proposed generalized SIR. As shown in Lee et al. (2013), even without the linear design condition, KSIR can still be used to estimate the SDR $\sigma$-field. However, it seems challenging to establish asymptotic theory with this formulation.

The problem of determining the number of indices in the SDR space is important but challenging, which we did not try to address here. In the real data, the smallest PMSE is usually obtained when $r = 2$ suggesting that $r = 2$ may be appropriate. However, this method of choosing of $r$ is certainly problematic unless the goal is mean prediction. Finally, it is worthwhile to investigate the extension of other SDR method, such as SAVE (Cook and Weisberg, 1991; Cook, 2000; Zhu and Zhu, 2007; Zhu et al., 2010; Dong and Li, 2010), using product or additive kernels.

**Supplementary Materials**

The supplementary material online for this paper contains the technical proof of Theorem 1.
Acknowledgements

The authors sincerely thank Professor Hsin-Cheng Huang, an AE and two reviewers for their insightful comments that has led to significant improvements of the manuscript.

References


Supplementary Material for “Kernel additive sliced inverse regression”

Heng Lian and Qin Wang
University of New South Wales and Virginia Commonwealth University

Proof of the main theorem.

In the proofs, $C$ denotes a generic positive constant. We first note that $E_X [(\hat{f}(X^*) - f(X^*))^2] = \|\Sigma^{1/2}(\hat{f} - f)\|_H^2$. From (4), $\Sigma^{1/2}\hat{f}$ satisfies the eigenvalue equation

$$\Sigma^{1/2}(\Sigma_n + sI)^{-1}\Gamma_n\Sigma^{-1/2}(\Sigma^{1/2}\hat{f}) = \lambda\Sigma^{1/2}\hat{f}.$$ 

Similarly, $\Sigma^{1/2}f$ satisfies the eigenvalue equation

$$\Sigma^{1/2}\Sigma^{-1}\Gamma\Sigma^{-1/2}(\Sigma^{1/2}f) = \lambda\Sigma^{1/2}f.$$ 

Using the perturbation theory for operators, for example as in Chapter 4 of Kato (1995), we only need to show that $\|\Sigma^{1/2}(\Sigma_n + sI)^{-1}\Gamma_n\Sigma^{-1/2} - \Sigma^{1/2}\Sigma^{-1/2}\Gamma\Sigma^{-1/2}\|_2 = O_p(c_n n^{-d/(d+1)})$, where $\|A\|$ denotes the operator norm for an operator $A$ defined on $\mathcal{H}_K$. 

1
We write

\[
\|\Sigma^{1/2} (\Sigma_n + sI)^{-1} \Gamma_n \Sigma^{-1/2} - \Sigma^{1/2} \Sigma^{-1} \Sigma^{1/2} \| \\
= \|\Sigma^{1/2} ((\Sigma_n + sI)^{-1} \Gamma_n - (\Sigma + sI)^{-1} \Gamma + (\Sigma + sI)^{-1} \Gamma - \Sigma^{-1} \Gamma) \Sigma^{-1/2} \| \\
= \|\Sigma^{1/2} ((\Sigma + sI)^{-1} (\Gamma_n - \Gamma) + (\Sigma + sI)^{-1} (\Sigma - \Sigma_n)(\Sigma_n + sI)^{-1} \Gamma_n \\
+ (\Sigma + sI)^{-1} (\Sigma - \Sigma_n) \Gamma) \Sigma^{-1/2} \|
\]

\[
\leq \|\Sigma^{1/2} (\Sigma + sI)^{-1} (\Gamma_n - \Gamma) \Sigma^{-1/2} \| + \|\Sigma^{1/2} (\Sigma + sI)^{-1} (\Sigma - \Sigma_n)(\Sigma_n + sI)^{-1} \Gamma_n \Sigma^{-1/2} \|
\]

\[
+ \|\Sigma^{1/2} ((\Sigma + sI)^{-1} \Gamma - \Sigma^{-1} \Gamma) \Sigma^{-1/2} \|
\]

\[
=: (I) + (II) + (III).
\] (6)

To simplify the proofs and notations a little bit, we assume \( K(., X) = 0 \) in the following, since using \( \|K(., X)\|_H = O_p(n^{-1/2}) \), such terms does not lead to extra difficulties in the proof. By the same reason, we also replace \( \hat{p}_h \) by \( p_h = P(Y = y_h) \) in the following expression of \( \Gamma_n \). Obviously we can rewrite \( \Gamma_n \) and \( \Gamma \) as

\[
\Gamma_n = \sum_{h=1}^H \frac{1}{p_h} \left( \frac{1}{n} \sum_{i=1}^n K(., X_i)I\{Y_i = y_h\} \right) \otimes \left( \frac{1}{n} \sum_{i=1}^n K(., X_i)I\{Y_i = y_h\} \right),
\]

\[
\Gamma = \sum_{h=1}^H \frac{1}{p_h} E[K(., X)I\{Y = y_h\}] \otimes E[K(., X)I\{Y = y_h\}].
\] (7)

The term (III) is easy to deal with as follows. Using

\[
\|\Sigma^{1/2} (\Sigma + sI)^{-1} - \Sigma^{-1}(E[K(., X)I\{Y = y_h\}] \otimes E[K(., X)I\{Y = y_h\}]) \Sigma^{-1/2} \|^2 \\
= \|s \Sigma^{1/2} (\Sigma + sI)^{-1} (\Sigma^{-1} E[K(., X)I\{Y = y_h\}] \otimes (\Sigma^{-1} E[K(., X)I\{Y = y_h\}]) \Sigma^{-1/2} \|^2 \\
= O(\|s \Sigma^{1/2} (\Sigma + sI)^{-1} \|^2) \\
= O(s),
\]
we have \((III)^2 = O(s)\).

For the term (II), writing \(\Sigma(x) = K(., x) \otimes K(., x)\) and using that

\[
E\|\Sigma^{1/2}(\Sigma + sI)^{-1}\Sigma(x)\|^2_{HS} \\
= E\text{tr}(\Sigma(x)^2(\Sigma + sI)^{-1}\Sigma(\Sigma + sI)^{-1}) \\
\leq CE\text{tr}(\Sigma(x)(\Sigma + sI)^{-1}\Sigma(\Sigma + sI)^{-1}) \\
= C\text{tr}(\Sigma + sI)^{-1}\Sigma(\Sigma + sI)^{-1}) \\
= C\sum_j \frac{\lambda_j^2}{(\lambda_j + s)^2},
\]

where \(\|\cdot\|_{HS}\) is the Hilbert-Schmidt norm. In the inequality above we used that \(\|\Sigma(x)f\|_\mathcal{H} = \|K(., x)f(x)\|_\mathcal{H} = \sqrt{t_2(x)K(x, x)} \leq C\|f\|_\infty \leq C\|f\|_\mathcal{H}\) and thus \(\|\Sigma(x)\| \leq C\), and the inequality \(\text{tr}(AB) \leq \|A\|\text{tr}(B)\).

Thus using the Markov inequality, we have

\[
(II)^2 \\
\leq \|\Sigma^{1/2}(\Sigma + sI)^{-1}(\Sigma - \Sigma_n)\|^2 \cdot \|(\Sigma_n + sI)^{-1}\Gamma_n\Sigma_n^{-1/2}\|^2 \\
= O_p\left(\sum_j \frac{\lambda_j^2}{n(\lambda_j + s)^2}\right) \|(\Sigma_n + sI)^{-1}\Gamma_n\Sigma_n^{-1/2}\|^2.
\]

We will argue later that actually \(\|(\Sigma_n + sI)^{-1}\Gamma_n\Sigma_n^{-1/2}\|^2 = O_p(1)\).

The term (I) is more complicated. Let \(\Gamma_{nh}\) and \(\Gamma_h\) be the terms on the right hand side of the sums in (7) such that \(\Gamma_n = \sum_h p_h^{-1}\Gamma_{nh}, \Gamma = \sum_h p_h^{-1}\Gamma_h\). To bridge \(\Gamma_{nh}\) and \(\Gamma_h\), we further define

\[
\Gamma'_{nh} = \left(\frac{1}{n} \sum_{i=1}^n K(., X_i)I\{Y_i = y_h\}\right) \otimes E[K(., X)I\{Y = y_h\}].
\]
Since

\[ E[\Sigma^{1/2}(\Sigma + sI)^{-1}(K(., X)I\{Y = y_h\})] - E[\Sigma^{1/2}(\Sigma + sI)^{-1}(K(., X)I\{Y = y_h\})] \]

\[ \otimes (\Sigma^{-1/2}E[K(., X)I\{Y = y_h\})]\] 

\[ \leq CE \| \Sigma^{1/2}(\Sigma + sI)^{-1}(K(., X)I\{Y = y_h\}) - E[\Sigma^{1/2}(\Sigma + sI)^{-1}(K(., X)I\{Y = y_h\})]\|_H^2 \]

\[ \leq CE \| \Sigma^{1/2}(\Sigma + sI)^{-1}K(., X)I\{Y = y_h\}\|_H^2 \]

\[ = CE [I\{Y = y_j\}\langle \Sigma(\Sigma + sI)^{-2}K(., X), K(., X)\rangle_H]\]

\[ = CE [I\{Y = y_j\}tr(\Sigma(\Sigma + sI)^{-2}\Sigma(X))] \]

\[ = CE [tr(\Sigma(\Sigma + sI)^{-2}\Sigma(X))E[I\{Y = y_j\}|X]] \]

\[ \leq Ctr(\Sigma^2(\Sigma + sI)^{-2}) \]

\[ = C \sum_{j} \frac{\lambda_j^2}{(\lambda_j + s)^2} \]

by Markov inequality,

\[ \| \Sigma^{1/2}(\Sigma + sI)^{-1}(\Gamma_n - \Gamma_n')\Sigma^{-1/2} \|_2^2 = O_p \left( \sum_{j} \frac{\lambda_j^2}{n(\lambda_j + s)^2} \right) \].

Similarly one can show

\[ \| \Sigma^{1/2}(\Sigma + sI)^{-1}(\Gamma_n - \Gamma_n')\Sigma^{-1/2} \|_2^2 = O_p \left( \sum_{j} \frac{\lambda_j^2}{n(\lambda_j + s)^2} \right) \].

These imply that

\[ (II)^2 = O_p \left( \sum_{j} \frac{\lambda_j^2}{n(\lambda_j + s)^2} \right) \].

Once we have shown \( \| (\Sigma_n + sI)^{-1}\Gamma_n\Sigma_n^{-1/2} \| = O_p(1) \), the bounds given above will combine to obtain that (6) is bounded by \( O_P \left( s + \sum_j \frac{\lambda_j^2}{n(\lambda_j + s)^2} \right) \) and direct calculations by plugging in \( \lambda_j \asymp j^{-d} \) and \( s = c_n n^{-d/(d+1)} \) shows that this is \( O_P(c_n n^{-d/(d+1)}) \).
What is left is to show \( \| (\Sigma_n + sI)^{-1}\Gamma_n\Sigma^{-1/2}\| = O_p(1) \), which is equivalent to showing
\[
\| (\Sigma_n + sI)^{-1}\Gamma_n\Sigma^{-1/2} - \Sigma^{-1}\Gamma\Sigma^{-1/2}\| = O_p(1).
\]
Note this equation is actually similar to (6). Following similar lines that are used to upper bound the terms (I)-(III) in (6), we will get
\[
\| (\Sigma_n + sI)^{-1}\Gamma_n\Sigma^{-1/2} - \Sigma^{-1}\Gamma\Sigma^{-1/2}\|^2 = O_p\left( \sum_j \frac{\lambda_j}{n(\lambda_j + s)^2} \right)\| (\Sigma_n + sI)^{-1}\Gamma_n\Sigma^{-1/2}\|^2 + O_p(1 + \sum_j \frac{\lambda_j^{2}}{n(\lambda_j + s)^2}).
\]
When \( s = c_n n^{-d/(d+1)} \) with \( c_n \to \infty \), by direct calculations we have \( \sum_j \frac{\lambda_j}{n(\lambda_j + s)^2} = o(1) \) and thus the above displayed equation implies \( \| (\Sigma_n + sI)^{-1}\Gamma_n\Sigma^{-1/2}\| = O_p(1) \). □