Erasmus University Rotterdam

Thesis MSc Econometrics

Adaptive Estimation in Linear Regression Using Repeated Kernel Error Density Estimation

Author:
Hugo Reichardt

Supervisor:
Dr. Mikhail Zhelonkin

July 25, 2017

Abstract: In this thesis, I propose the Repeated Kernel Density-based Regression Estimator (RKDRE) for the linear regression model. The intuition is that the unknown error distribution can be approximated by using kernel density estimation on the residuals of an initial estimator. This density can then be maximized to obtain a new parameter estimate. The process of estimating the parameter and obtaining a density is repeated until convergence. RKDRE can be regarded as the multi-step version of KDRE as proposed by Yao and Zhao (2013). For computational convenience, I develop a constrained EM algorithm to perform the maximization. I show under relatively weak conditions that both KDRE and RKDRE converge almost surely to the true parameter. Also, I prove that using the conditions under which KDRE is adaptive (i.e., asymptotically normal and efficient), RKDRE is adaptive too. Even though the asymptotic properties of the estimators are the same, I show in a numerical study that RKDRE generally attains higher mean-square-error-efficiency. The overall performance of RKDRE is also arguably higher than any of the wide range of other adaptive estimators considered in the numerical study. The practical relevance of RKDRE is illustrated with an application to experimental research done in (Andrabi et al., 2017).

Key words: adaptive estimation; kernel density estimation; EM algorithm; semiparametric maximum likelihood

Contents

1 Introduction

Consider the general linear regression model

$$y_i = \boldsymbol{x_i'}\boldsymbol{\beta}_0 + \varepsilon_i, \tag{1}$$

where $\boldsymbol{x_i}$ are known $p \times 1$-vectors and $\boldsymbol{\beta}_0 \in \boldsymbol{\mathcal{B}} \subseteq \mathbb{R}^p$ is an unknown parameter vector including an intercept. The error terms ε_i are i.i.d. realizations of the unknown error density f. The error terms are independent of $\boldsymbol{x_i}$. Model (1) is semiparametric with $\boldsymbol{\beta}$ and f its parametric and non-parametric part, respectively. It is well known that, under the true density, maximum likelihood estimation (MLE) has several desirable properties such as consistency, asymptotic normality, and asymptotic efficiency. However, the specific form of f is unknown in almost all practical instances. In parametric regression, f is assumed to be the probability density function of a certain known distribution. The normal distribution is a convenient choice as it can be shown that in that case the MLE of $\boldsymbol{\beta}_0$ simplifies to the ordinary least squares estimate $\hat{\boldsymbol{\beta}}_{OLS}$. Naturally, an invalid assumption on the density function comes at a cost; in case of misspecification of the distribution, the MLE is in general neither consistent nor asymptotically efficient (Pawitan, 2001, p. 372). To overcome this problem, I suggest an adaptive estimator. That is, an estimator that has the same asymptotic distribution as the MLE in case f were known. The intuition is as follows; if we obtain an initial estimate that is roughly correct, the empirical distribution of the residuals corresponding to that estimate approximates the true density of the error terms. Then, we can estimate the distribution of the residuals by a kernel density estimator and perform standard maximum likelihood on the estimated kernel density. Then, we obtain residuals corresponding to a new estimate of $\boldsymbol{\beta}_0$ and, subsequently, perform maximum likelihood on the estimated density of those residuals. This process is repeated until the estimates converge.

To formally define the algorithm, I first define $f_n(x)$ as the Rosenblatt-Parzen kernel density estimator of the error terms (Rosenblatt et al., 1956; Parzen, 1962). That is,

$$f_n(x) = \frac{1}{nh_n} \sum_{i=1}^{n} K\left(\frac{x - \varepsilon_i}{h_n}\right), \tag{2}$$

where h_n is called the bandwidth. Similarly, $\hat{f}_n(x)$ is

$$\hat{f}_n(x) = \frac{1}{nh_n} \sum_{i=1}^{n} K\left(\frac{x - \hat{\varepsilon}_i}{h_n}\right), \tag{3}$$

where $e_i = y_i - \boldsymbol{x_i'}\hat{\boldsymbol{\beta}}$, $i = 1, 2, .., n$ are the residuals in model (1) corresponding to a certain estimate of $\boldsymbol{\beta}_0$.

Then, the Repeated Kernel Density-Based Regression Estimator (RKDRE) is computed as follows:

1. Initialize with $\hat{\boldsymbol{\beta}}^{(0)} = \hat{\boldsymbol{\beta}}_{OLS}$. Obtain the corresponding residuals $e_{i,OLS} = y_i - \boldsymbol{x}_{\boldsymbol{i}}'\hat{\boldsymbol{\beta}}_{OLS}$, $i = 1, 2, .., n$.

2. Estimate $f(x)$ by $\hat{f}_n^{(m)}(x)$, the kernel density estimator based on the residuals corresponding to $\hat{\boldsymbol{\beta}}^{(m-1)}$.

3. Compute $\hat{\boldsymbol{\beta}}^{(m)} = \arg\sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} \hat{Q}^{(m)} = \sum_{i=1}^{n} \ln \hat{f}_n^{(m)}(y_i - \boldsymbol{x}_{\boldsymbol{i}}'\boldsymbol{\beta})$ subject to $c(\boldsymbol{\beta}^{(m)}) = 0$ where $c(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\left[y_i - \boldsymbol{x}_{\boldsymbol{i}}'\boldsymbol{\beta}\right]$.

4. Repeat steps 2. and 3. until convergence.

To the best of my knowledge, none of the studies on semiparametric adaptive estimators have examined the performance of a repetitive procedure as described above. Most of the previously proposed estimators also use an initial estimate of $\boldsymbol{\beta}_0$, but then take only one final step to obtain the adaptive estimate. Two-step methods are usually enough to obtain desirable theoretical properties such as consistency and asymptotic efficiency. However, Mammen et al. (1996) show that the empirical distribution of the residuals depends strongly on the initial estimator. In particular, if one uses a maximum likelihood estimator based on an incorrectly specified error distribution G, the empirical distribution of the residuals is shifted toward G. Hence, by using $\hat{\boldsymbol{\beta}}_{OLS}$ as a convenient first-step estimator, the two-step method will be drawn towards the normal distribution (since OLS is equivalent to MLE under the normal distribution). This observation forms the intuition that, in finite samples, multi-step procedures may perform better than two-step procedures.

Only quite recently, Yao and Zhao (2013) researched the performance of the two-step version of the algorithm and introduced the term Kernel Density-Based Regression Estimator (KDRE) (although they do not impose the constraint that the residuals sum to zero in step 3). As far as I know, Yao and Zhao (2013) have been the first (and, as of yet, only) to discuss the two-step maximum likelihood procedure. In an extensive numerical study, I show that the performance (in the mean square error sense) of RKDRE is substantially better than that of KDRE. In fact, I find that RKDRE shows the best overall performance out of a wide range of investigated semiparametric adaptive estimators. Under some distributions, such as the variance-contaminated normal distribution and the log-normal distribution, the mean square error of RKDRE is up to two times lower than that of the second best estimator. Under the other investigated distributions, it is either the most efficient or close to the most efficient estimator. This corroborates the intuition that a two-step method usually not suffices to fully exploit the information contained in the data.

Even regardless of the strong performance of RKDRE, I claim that the numerical study is in itself a contribution to the literature. Most of the methods investigated have all been tested by means of simulation in earlier studies (while some more extensively than others). However, almost all of these simulations show the performance of the semiparametric estimator vis-à-vis a parametric estimator. For instance, Yao and Zhao (2013) only compare KDRE with OLS. I argue that it is more interesting to compare the performance of a semiparametric estimator against another as it

is a well-established fact that OLS is not efficient under the conditions that would lead the econometrician to adopt a semiparametric approach. Furthermore, the numerical study includes an examination of the proper choice of bandwidth for kernel estimators such as (R)KDRE; with that examination, I aim to fill another gap that has been left by the literature on adaptive estimators.

I build upon Yao and Zhao (2013) on a more technical note as well. First of all, I establish strong (almost sure) convergence to $\boldsymbol{\beta}_0$ under relatively weak conditions, both for the two-step (KDRE) and the proposed multi-step estimator (RKDRE). Yao and Zhao (2013) only prove asymptotic normality and efficiency for which they need stronger conditions. Another issue that is not touched upon by Yao and Zhao (2013) is the question of how to obtain standard errors. I suggest two different methods and prove consistency for both. In a subsequent numerical study, I compare the root mean squared standard error (as estimated by the two methods) with the actual root mean square error. I find that the methods estimate the standard error of the slope coefficient reasonably well. However, the standard errors of intercept coefficients are usually underestimated, such that, if these are of interest, bootstrapping is the preferred method.

Finally, I apply the adaptive estimators to the experimental research done in (Andrabi et al., 2017). The experiment explores the effect of provision of information (in the form of so-called report cards) to schools and households on test scores, school fees, and enrollment rates in 112 villages in Pakistan. In this paper, OLS is applied. However, I show that there is evidence to believe that adaptive estimators may be more efficient. I find that two out of the three main treatment effects found and described in the paper (i.e. the effect of the report cards on fees and on test scores) are not significantly different from zero when adaptive estimation is applied. Most interestingly, I find that the RKDRE estimate of the effect of the report cards on school fees is more than forty times smaller than the OLS estimate. None of the other adaptive estimators adjust the OLS estimate so dramatically. On top of that, I show that the prediction performance of RKDRE is superior to the prediction performance of the other estimators considered.

In the following section, I describe and explain adaptive estimators that are previously proposed in the literature. Then, in Section 3, I show the asymptotic properties of the RKDRE algorithm. Section 4 describes how RKDRE is made computationally feasible using an EM algorithm that performs the constrained maximization in step 3. of the algorithm. Section 5 shows a numerical study of the efficiency of RKDRE and the adaptive estimators discussed in Section 2. It also includes an analysis of the proper choice of bandwidth for the estimators that use kernel density estimation and a comparison of computation time of the different methods. In Section 6 and Section 7, I discuss the estimation of standard errors of the RKDRE algorithm and apply the adaptive estimators to the research in (Andrabi et al., 2017), respectively. I conclude in Section 8.

2 Literature on adaptive estimation

It is well known that the asymptotic variance of the MLE under the true distribution attains the Cramér-Rao lower bound (the lowest possible variance of an unbiased estimator): the inverse of the Fisher's information matrix $I_{\beta\beta}$. This means that MLE is asymptotically efficient. To formalize this concept, I use the following notation:

$$L_n(y_i|\boldsymbol{\beta}) = \ln f(y_i|\boldsymbol{\beta}) = \ln f(y_i - \boldsymbol{x_i'}\boldsymbol{\beta}) \tag{4}$$

$$d_\beta(\boldsymbol{\beta}) = \frac{\partial L_n}{\partial \boldsymbol{\beta}}(\boldsymbol{\beta}) = -f'(y_i|\boldsymbol{\beta}) f^{-1}(y_i|\boldsymbol{\beta}) \boldsymbol{x_i} \tag{5}$$

$$d_{\beta\beta}(\boldsymbol{\beta}) = \frac{\partial^2 L_n}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta'}}(\boldsymbol{\beta}) = \frac{f(y_i|\boldsymbol{\beta}) f''(y_i|\boldsymbol{\beta}) - f'^2(y_i|\boldsymbol{\beta})}{f^2(y_i|\boldsymbol{\beta})} \boldsymbol{x_i} \boldsymbol{x_i'}, \tag{6}$$

where $f'(u) = \partial f(u)/\partial u$, $f''(u) = \partial f'(u)/\partial u$, and $f'^2(u) = f'(u)f'(u)$. Then the information matrix can be defined as

$$\begin{aligned} I_{\beta\beta} &= E[d_\beta(\boldsymbol{\beta}_0) d_\beta(\boldsymbol{\beta}_0)'] \\ &= -E[d_{\beta\beta}(\boldsymbol{\beta}_0)], \end{aligned} \tag{7}$$

where the second equality holds under mild regularity conditions.[1] It is well-established that under Conditions (i)-(iv) of Theorem 3.5 below, and Conditions (i)-(v) of Theorem 3.3 in (Newey and McFadden, 1994, p. 2146) (under which also the conditions of the information matrix equality are satisfied (Newey and McFadden, 1994, p. 2146)), we obtain

$$\sqrt{n}\left(\hat{\boldsymbol{\beta}}_{ML} - \boldsymbol{\beta}_0\right) \xrightarrow{d} \mathcal{N}\left(0, I_{\beta\beta}^{-1}\right). \tag{8}$$

In the context of linear regression, an estimator is *adaptive* if it attains the same asymptotic distribution even if f is unknown (hence if it asymptotically normal and asymptotically efficient). It can be shown that not all parameters are adaptively estimable. Necessary conditions are derived by Begun et al. (1983); in the linear model in (1), the necessary condition for the slope estimates in $\boldsymbol{\beta}_0$ to be adaptively estimable is satisfied if $\boldsymbol{\beta}_0$ contains an intercept. If f is symmetric around zero, the necessary condition is also satisfied for the intercept (Pagan and Ullah, 1999, p.220). In this section, I describe several semiparametric estimators with adaptive properties. The overview presented is a comprehensive, but not exhaustive overview of the semiparametric methods that are proposed in the context of adaptive estimation. However, as Pagan and Ullah (1999, p .226) note, many other algorithms can be regarded as a special form of one of the methods below.

[1] Specifically, twice differentiability of $\ln f(y_i|\boldsymbol{\beta})$ and the assumption that interchanging the order of integration and differentiation is allowed, i.e.

$$\int \frac{\partial}{\partial \boldsymbol{\beta}} f(y|\boldsymbol{\beta}) dy = \frac{\partial}{\partial \boldsymbol{\beta}} \int f(y|\boldsymbol{\beta}) dy = 0.$$

2.1 SBS

Stone (1975); Bickel (1982); Schick (1993) (henceforth SBS) employed a two-step procedure. Denote $\tilde{\boldsymbol{\beta}}$ a certain $\sqrt{n}$-consistent estimator, i.e. $\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}_0 = O_p(n^{-\frac{1}{2}})$.[2] Then, the infeasible two-step estimator

$$\hat{\boldsymbol{\beta}} = \tilde{\boldsymbol{\beta}} + \frac{1}{n} I_{\beta\beta}^{-1}(\tilde{\boldsymbol{\beta}}) d_{\beta}(\tilde{\boldsymbol{\beta}})$$

can be shown to be asymptotically as efficient as the MLE. Note that the infeasibility of this estimator follows from the fact that f is unknown and hence $I_{\beta\beta}$ and d_{β} are unknown. The rather intuitive approach of SBS is to replace $d_{\beta}(\tilde{\boldsymbol{\beta}})$ by $\hat{d}_{\beta}(\tilde{\boldsymbol{\beta}}) = \sum_{i=1}^{n} \hat{f}_n'(y_i|\tilde{\boldsymbol{\beta}}) \hat{f}_n^{-1}(y_i|\tilde{\boldsymbol{\beta}}) \boldsymbol{x_i}$ where $\hat{f}_n(x)$ is defined as in (3) and $\hat{f}_n'(x)$ is its derivative with respect to x. Similarly, $I_{\beta\beta}^{-1}(\tilde{\boldsymbol{\beta}})$ is replaced with $n^2 \left[\sum \boldsymbol{x_i x_i'} \sum \left(\hat{f}_n'(y_i|\tilde{\boldsymbol{\beta}}) \hat{f}_n^{-1}(y_i|\tilde{\boldsymbol{\beta}}) \right)^2 \right]^{-1}$ (Pagan and Ullah, 1999, p .227). These estimates are based on the residuals from an initial estimator of $\boldsymbol{\beta}$ which is usually chosen to be the OLS estimator. The conditions under which this two-step approach can be shown to be asymptotically efficient have been researched extensively (Bickel, 1982; Manski, 1984; Andrews, 1994). Most importantly, the kernel estimator of the score function $f'(y_i|\boldsymbol{\beta}) f^{-1}(y_i|\boldsymbol{\beta})$ must be (i) i.i.d., and (ii) independent of $\boldsymbol{x_i}$. These conditions are restrictive and not easy to verify in practice (Yuan and De Gooijer, 2007, p. 845; Pagan and Ullah, 1999, p. 228). Bickel (1982) solved the i.i.d. problem by splitting the sample in two; one sub-sample to estimate the score and another to solve for $\boldsymbol{\beta}$. However, perhaps not surprisingly, Manski (1984) finds by means of simulation that the estimator works much better when the sample is not split (that is, if the estimated score and $\tilde{\boldsymbol{\beta}}$ are both computed using the entire sample). If (i) and (ii) are satisfied, a sufficient condition for adaptiveness is that (iii) $E\left[\left(f'(y_i|\boldsymbol{\beta}) f^{-1}(y_i|\boldsymbol{\beta}) - \hat{f}_n'(y_i|\tilde{\boldsymbol{\beta}}) \hat{f}_n^{-1}(y_i|\tilde{\boldsymbol{\beta}}) \right)^2 \right] \to 0$.

Since $\hat{f}_n(x)$ is present in the denominator of $\hat{d}_{\beta}$, unstable estimates may follow for near-zero values of $\hat{f}_n(x)$. Hence, Bickel (1982) suggest to trim the estimator of the kernel score as follows

$$\frac{\hat{f}_n'(y_i|\tilde{\boldsymbol{\beta}})}{\hat{f}_n(y_i|\tilde{\boldsymbol{\beta}})} = \begin{cases} \frac{\hat{f}_n'(y_i|\tilde{\boldsymbol{\beta}})}{\hat{f}_n(y_i|\tilde{\boldsymbol{\beta}})}, & \text{if } \left| y_i - \boldsymbol{x_i'}\boldsymbol{\beta} \right| \leq t_1,\ \hat{f}_n(y_i|\tilde{\boldsymbol{\beta}}) > t_2, \text{ and } \frac{\hat{f}_n'(y_i|\tilde{\boldsymbol{\beta}})}{\hat{f}_n(y_i|\tilde{\boldsymbol{\beta}})} < t_3 \\ 0, & \text{otherwise.} \end{cases}$$

This trimming mechanism ensures that near-zero values do not have unreasonably large influence on the estimate. Also, Bickel (1982, p. 665) shows that if $t_1 \to \infty$, $t_2 \to 0$, $t_3 \to \infty$, $h_n \to 0$, $\frac{t_1}{nh^3}$, and $h_n t_1 \to \infty$ as $n \to \infty$ then (iii) is satisfied. Hence, adaptiveness is established under the proper trimming parameters and condition (i) and (ii). Naturally, the growth rates of the trimming parameters are of little use to the practitioner and as such the choice for the trimming parameter is a practical disadvantage. Hsieh and Manski (1987) reduce the problem to selecting a

[2] If a sequence $X_n = O_p(a_n)$, this means that $\frac{X_n}{a_n}$ is *bounded in probability*. Formally, $\frac{X_n}{a_n}$ is bounded in probability if for every $\varepsilon > 0$, there exist a finite M and finite $N \in \mathbb{N}$ such that for all $n > N$

$$\Pr\left(\left| \frac{X_n}{a_n} \right| > M \right) < \varepsilon.$$

one-dimensional parameter t by suggesting the following relation between the trimming parameters:

$$t_1 = t, \quad t_2 = \exp\left(-\frac{t^2}{2}\right), \quad t_3 = t.$$

Hsieh and Manski (1987) vary t between 3, 4, and 8. For a sample size of 50, they find that $t = 8$ works best in almost all considered case.

2.2 LGMM(S)

Newey (1988) describes a two-step method that avoids kernel estimation. His approach is based on moment conditions that can be derived from certain assumptions on the error distribution. Two situations are analyzed. First, the case where the error terms are i.i.d. and independent of $\boldsymbol{x_i}$. This model implies the moment condition that any function of the errors are uncorrelated with any function of the regressors.[3] Second, the case where the distribution of ε_i is symmetric around zero conditional on the regressor $\boldsymbol{x_i}$. This second model allows for conditional heteroskedasticity, i.e. the variance of ε_i is allowed to depend on $\boldsymbol{x_i}$. The assumption that the errors are symmetrically distributed around zero yields the moment conditions that any odd function of the errors are uncorrelated with any function of the regressors. Hence, in both situations we can exploit moment restrictions to construct what Newey (1988) calls the Linearized General Method of Moments (LGMM) estimator. For later reference, I refer to the LGMM estimator based on the moment conditions following from the errors being i.i.d. and independent of $\boldsymbol{x_i}$ as **LGMM**, and to the LGMM estimator based on the moment conditions following from symmetry as **LGMMS**. For LGMM, natural moment conditions arise from the fact that $E\left[\boldsymbol{x_i}\left(\varepsilon_i^j - E\left[\varepsilon_i^j\right]\right)\right] = 0$ for $j = 1, 2, .., J$. However, Newey (1988) finds that these high-order 'raw' moments, $m_j(\varepsilon_i) = \varepsilon_i^j$, are sensitive to a fat-tailed error distribution. Estimates that are more robust against fat tails are obtained by using the 'transformed' powers, i.e.

$$m_j(\varepsilon_i) = \left(\frac{\varepsilon_i}{1 + |\varepsilon_i|}\right)^j,$$

or the 'weighted' powers, i.e.

$$m_j(\varepsilon_i) = \exp\left(-\frac{\varepsilon_i}{2}\right)\varepsilon_i^j.$$

Similarly, for LGMMS, we may use that $E\left[\boldsymbol{x_i}\varepsilon_i^{2j-1}\right] = 0$ for $j = 1, 2, .., J$.[4] As for LGMM, performance may be improved if we use the odd powers of the 'transformed' method instead. Note that for technical reasons the 'weighted' powers can not be used for LGMMS (Newey, 1988, p.315). In general, both for LGMM and LGMMS, we use the moment conditions that $E\left[\boldsymbol{x_i}\left(m_j(\varepsilon_i) - \mu_j\right)\right] = 0$ for $j = 1, 2, .., J$ where $u_j = E\left[m_j(\varepsilon_i)\right]$. To define the LGMM(S) estimator, I introduce the

[3] This result follows from the fact that functions of independent random variables are also independent.

[4] We do not subtract the mean here since for LGMMS the error terms are assumed to be symmetric around zero and for such random variables all odd-order moments are equal to zero (if they exist).

following notation:

$$\boldsymbol{\zeta}_i' = [m_1(\varepsilon_i) - \mu_1, m_2(\varepsilon_i) - \mu_2, .., m_J(\varepsilon_i) - \mu_J] \tag{9}$$

$$\boldsymbol{w'} = E[m_{1\varepsilon}(\varepsilon_i), m_{2\varepsilon}(\varepsilon_i), .., m_{J\varepsilon}(\varepsilon)] \tag{10}$$

$$\boldsymbol{V}_{\zeta\zeta} = \mathrm{Cov}(\boldsymbol{\zeta}_i) \tag{11}$$

where $m_{j\varepsilon}(\varepsilon) = \frac{\partial m_j(\varepsilon)}{\partial \varepsilon}$. Let $\hat{\varepsilon}_i$ denote the residuals of the initial estimate $\hat{\boldsymbol{\beta}}$, then the quantities in (9), (10), and (11) can be estimated by

$$\hat{\boldsymbol{\zeta}}_i' = [m_1(\hat{\varepsilon}_i) - \hat{\mu}_1, m_2(\hat{\varepsilon}_i) - \hat{\mu}_2, .., m_J(\hat{\varepsilon}) - \hat{\mu}_J] \tag{12}$$

$$\hat{\boldsymbol{w}}' = \left[\frac{1}{n}\sum m_{1\varepsilon}(\hat{\varepsilon}_i), \frac{1}{n}\sum m_{2\varepsilon}(\hat{\varepsilon}_i), .., \frac{1}{n}\sum m_{J\varepsilon}(\hat{\varepsilon}_i)\right] \tag{13}$$

$$\hat{\boldsymbol{V}}_{\zeta\zeta} = \frac{1}{n}\sum \hat{\boldsymbol{\zeta}}_i \hat{\boldsymbol{\zeta}}_i', \tag{14}$$

respectively, where $\hat{\mu}_j = \frac{1}{n}\sum m_j(\hat{\varepsilon}_i)$. Then, the LGMM(S) estimator is constructed as

$$\hat{\boldsymbol{\beta}}_{LGMM(S)} = \hat{\boldsymbol{\beta}} + \left[(\hat{\boldsymbol{w}}' \otimes \boldsymbol{X}'\boldsymbol{X})\left(\hat{\boldsymbol{V}}_{\zeta\zeta}^{-1} \otimes [\boldsymbol{X}'\boldsymbol{X}]^{-1}\right)(\hat{\boldsymbol{w}} \otimes \boldsymbol{X}'\boldsymbol{X})\right]^{-1} \times (\hat{\boldsymbol{w}}' \otimes \boldsymbol{X}'\boldsymbol{X})\left(\hat{\boldsymbol{V}}_{\zeta\zeta}^{-1} \otimes [\boldsymbol{X}'\boldsymbol{X}]^{-1}\right)(\boldsymbol{I}_J \otimes \boldsymbol{X}')\,\mathrm{vec}\left(\hat{\boldsymbol{\zeta}}\right) \tag{15}$$

where $\hat{\boldsymbol{\zeta}}$ is the $n \times J$ matrix $\left[\hat{\boldsymbol{\zeta}}_1', .., \hat{\boldsymbol{\zeta}}_n'\right]'$. Under certain assumptions, Newey (1988) proves asymptotic normality of both the LGMM and LGMMS estimator. In particular, it should hold that $J \to \infty$ and $\frac{J \ln J}{\ln n} \to 0$ as $n \to \infty$. Only for LGMMS, asymptotic efficiency is obtained (but not for LGMM). By means of simulation, Newey (1988) finds for LGMM that $J = 3$ performs best for sample sizes between $n = 50$ and $n = 200$. However, the mean square error efficiency of the estimator as a function of J flattens out as the sample size increases. Also, he finds that the 'transformed' method is in general preferred over the 'weighted' method. No numerical results for the LGMMS estimator are shown by (Newey, 1988).

2.3 KDRE

More recently, Yao and Zhao (2013) proposed the kernel density-based linear regression estimator (KDRE). The estimator is the unconstrained two-step version of the proposed RKDRE algorithm. That is, it follows from unconstrained maximization of the kernel likelihood function that is estimated on the basis of the residuals corresponding to an initial estimate. Under some conditions on the error terms, the regressors, the kernel, the bandwidth, and the initial estimator (i.e. Conditions (i)-(viii) of Theorem 3.6) Yao and Zhao (2013) prove that the KDRE algorithm is adaptive;[5] that is, asymptotically normal and efficient. For technical reasons, these properties are proven for a trimmed version. The untrimmed maximizer of the kernel likelihood is the solution to:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\hat{f}_n'(y_i|\boldsymbol{\beta})}{\hat{f}_n(y_i|\boldsymbol{\beta})}\boldsymbol{x_i'} = 0.$$

The trimmed version is then defined as the solution to

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\hat{f}_n'(y_i|\boldsymbol{\beta})}{\hat{f}_n(y_i|\boldsymbol{\beta})}\boldsymbol{x_i'}G_b\left(\hat{f}_n\left(y_i|\boldsymbol{\beta}\right)\right) = 0.$$

Here,

$$G_b(x) = \begin{cases} 0, & \text{if } x < b, \\ \int_b^x g_b(z)dz & \text{if } b \leq x \leq 2b, \\ 1, & \text{if } x > 2b, \end{cases} \tag{16}$$

where g_b is a four times continuously differentiable function with support on $[b, 2b]$ and $b \to 0$ if $n \to \infty$. This trimming function is introduced by Linton and Xiao (2007, p. 378) and they suggest the use of the beta function. For the purpose of KDRE, the trimming parameter is only used to simplify the proof, and is not used in the implementation. Yao and Zhao (2013) note that in practice the difference between the actual and the trimmed version is minimal. To simplify computation, Yao and Zhao (2013) develop an EM algorithm that is further discussed in Section 4.

2.4 YDG

Yuan and De Gooijer (2007) proposed another estimator based on estimating the error density by means of a kernel. The maximization criterion (applied to linear regression) is

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}}{\arg\max}\sum_{i=1}^{n}\ln\frac{1}{(n-1)h_n}\sum_{j\neq i}^{n}K\left(\frac{\left(y_i - \boldsymbol{x_i'}\boldsymbol{\beta}\right) - \left(y_j - \boldsymbol{x_j'}\boldsymbol{\beta}\right)}{h_n}\right). \tag{17}$$

Note that the kernel is based on $(n-1)$ observations. This is to avoid that the log-likelihood would contain artificial values of $K(0)$ for all $i = j$. Note that this method is a one-step approach and as such does not require an initial estimate. A disadvantage is the cancellation of the intercept coefficient in the difference $\boldsymbol{x_j'}\boldsymbol{\beta} - \boldsymbol{x_i'}\boldsymbol{\beta}$ in (17). To solve this, Yuan and De Gooijer (2007) proposed

[5] Conditions (i)-(viii) of Theorem 3.6 correspond to C1-C5 in (Yao and Zhao, 2013, p. 4506).

estimating

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}}{\arg\max} \sum_{i=1}^{n} \ln \frac{1}{(n-1)h_n} \sum_{j\neq i}^{n} K\left(\frac{r\left(y_i - \boldsymbol{x_i'}\boldsymbol{\beta}\right) - r\left(y_j - \boldsymbol{x_j'}\boldsymbol{\beta}\right)}{h_n}\right). \tag{18}$$

Yuan and De Gooijer (2007) suggest to use $r(z) = 10 \times \frac{\exp z}{1+\exp z}$. However, as Yao and Zhao (2013) note, this comes with an efficiency loss; $r(z) = z$ as in (17) is efficient in the sense that even though the intercept is cancelled out, the slope coefficients are adaptively estimated. Hence, Yao and Zhao (2013) suggest to compute

$$\hat{\boldsymbol{\beta}}^{*}_{YDG} = \underset{\boldsymbol{\beta^*}\in\boldsymbol{\mathcal{B}^*}}{\arg\max} \sum_{i=1}^{n} \ln \frac{1}{(n-1)h_n} \sum_{j\neq i}^{n} K\left(\frac{\left(y_i - \boldsymbol{x_i^{*\prime}}\boldsymbol{\beta}^*\right) - \left(y_j - \boldsymbol{x_j^{*\prime}}\boldsymbol{\beta}^*\right)}{h_n}\right), \tag{19}$$

where $\boldsymbol{x_i} = \left[1, \boldsymbol{x_i^*}\right]'$, and $\hat{\boldsymbol{\beta}}_{YDG} = \left[\hat{\alpha}_{YDG}, \hat{\boldsymbol{\beta}}^{*}_{YDG}\right]'$ and $\hat{\alpha}$ is the estimate of the intercept coefficient α. Subsequently, we set $\hat{\alpha}_{YDG} = \frac{1}{n}\sum\left[y_i - \boldsymbol{x_i^{*\prime}}\hat{\boldsymbol{\beta}}^{*}_{YDG}\right]$. The intercept estimate is not in general asymptotically efficient (Yao and Zhao, 2013, p.4503). However, despite this theoretical disadvantage, Yao and Zhao (2013) show in a numerical study that this implementation of YDG in general outperforms KDRE. Henceforth, if I refer to YDG, I refer to the estimate in (19).

3 Asymptotic properties

In this section, I derive the asymptotic properties of RKDRE. Section 3.1 contains the lemmas necessary for the proofs. In Section 3.2, I prove that the estimate of RKDRE converges almost surely to $\boldsymbol{\beta}_0$. The result also holds for KDRE (but was not proven by Yao and Zhao (2013)). Section 3.3 contains the proof that RKDRE belongs to the class of adaptive estimators, i.e. that it possesses the desired properties of asymptotic normality and efficiency.

3.1 Lemmas

Lemma 3.1. *If model* (1) *holds,* $\varepsilon_1, \varepsilon_2, .., \varepsilon_n$ *are i.i.d. with unknown density* $f(x)$ *where* f *is a uniformly continuous function that satisfies*

(i) $\int x f(x)dx = 0$

(ii) $0 < \int x^2 f(x)dx < \infty$,

$\{\boldsymbol{x_i}\}_{i=1}^{\infty}$ *satisfy,*

(iii) $\exists\ 0 < M < \infty$ *such that* $\left\|\boldsymbol{x_i}\right\| < M\ \forall\ i = 1, 2, .., n$

(iv) $\lim_{n\to\infty} \boldsymbol{S_n} = \boldsymbol{Q}$ *where* $\boldsymbol{S_n} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x_i x_i'}$,

and the following assumptions on the kernel function $K(x)$ *hold:*

(v) $K(x)$ *is uniformly bounded and* $\exists\ 0 < \rho < \infty$ *such that* $K(x) = 0\ \forall\ x : \|x\| \geq \rho$

(vi) $K(x)$ *is Riemann integrable on* $[-\rho, \rho]$

(vii) when $n \to \infty$, $0 < h_n \to 0$ *and* $\frac{\sqrt{n}h_n}{\ln n} \to \infty$,

then

$$\sup_{x\in\mathbb{R}} \left\| \hat{f}_{n,OLS}(x) - f(x) \right\| \overset{a.s.}{\to} 0 \tag{20}$$

where $\hat{f}_{n,OLS}(x)$ *is the kernel density estimator based on the OLS residuals* $e_{i,OLS} = y_i - \boldsymbol{x_i'}\hat{\boldsymbol{\beta}}_{OLS}$, $i = 1, 2, .., n$.

Proof: see Theorem 5 in (Zhang, 1990). □

Lemma 3.2. *Under the assumptions of Lemma 3.1, if any estimator* $\boldsymbol{\beta}^*$ *satisfies*

(i) $\Pr\left(\lim_{n\to\infty} \max|\boldsymbol{\beta}^* - \boldsymbol{\beta}_0| \leq \max\left|\boldsymbol{S_n^{-1}}\right| \left\|\ln \max\left|\boldsymbol{S_n^{-1}}\right|\right\|\right) = 1$ *where* $\max|A| = \max_{i,j}|a_{ij}|$ *where* a_{ij} *are the elements of a matrix* A

then

$$\sup_{x\in\mathbb{R}} \left\| \hat{f}_n^*(x) - f(x) \right\| \overset{a.s.}{\to} 0 \tag{21}$$

where $\hat{f}_n^*(x)$ *denotes the kernel density estimator based on the residuals* $e_i = y_i - \boldsymbol{x_i'}\boldsymbol{\beta}^*$, $i = 1, 2, .., n$

Proof: This follows immediately from Theorem 5 and Lemma 4 in (Zhang, 1990) in conjunction with Theorem 4 and (29) in (Chai et al., 1991). □

Lemma 3.3. *If there is a function $Q_0(\boldsymbol{\beta})$ such that*

(i) $Q_0(\boldsymbol{\beta})$ *is uniquely maximized at* $\boldsymbol{\beta}_0$

(ii) $\boldsymbol{\mathcal{B}}$ *is compact*

(iii) $Q_0(\boldsymbol{\beta})$ *is continuous*

(iv) $\sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} \left\| \hat{Q}_n(\boldsymbol{\beta}) - Q_0(\boldsymbol{\beta}) \right\| \xrightarrow{a.s.} 0$,

then

$$\hat{\boldsymbol{\beta}} \xrightarrow{a.s.} \boldsymbol{\beta}_0 \tag{22}$$

where $\hat{\boldsymbol{\beta}}$ *maximizes objective function* $\hat{Q}_n(\boldsymbol{\beta})$ *subject to* $\boldsymbol{\beta} \in \boldsymbol{\mathcal{B}}$. *The weak convergence result, i.e.* $\hat{\boldsymbol{\beta}} \xrightarrow{p} \boldsymbol{\beta}_0$ *can be obtained by replacing Condition (iv) by* $\sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} \left\| \hat{Q}_n(\boldsymbol{\beta}) - Q_0(\boldsymbol{\beta}) \right\| \xrightarrow{p} 0$.

Proof: see Theorem 2.1 in (Newey and McFadden, 1994). □

Lemma 3.4. *If* $f_n : \mathcal{B} \to \mathbb{R}$ *is a continuous function,* $\mathcal{B}$ *is compact, and* $f_n \xrightarrow{a.s.} f$, *then*

$$\lim_{n\to\infty} \int_{\mathcal{B}} f_n du = \int_{\mathcal{B}} f du. \tag{23}$$

Proof: since $\mathcal{B}$ is compact and f_n is continuous, the image $f_n(\mathcal{B})$ is a compact subset of $\mathbb{R}$ and hence, closed and bounded. Then, the result follows from the bounded convergence theorem (Wade, 1974). □

3.2 Almost sure convergence

Theorem 3.5. *Under the assumptions of Lemma 3.1 and*

(i) if $\hat{\boldsymbol{\beta}} \neq \boldsymbol{\beta}_0$, *then* $f(y_i|\hat{\boldsymbol{\beta}}) \neq f(y_i|\boldsymbol{\beta}_0)$

(ii) $\boldsymbol{\beta} \in \boldsymbol{\mathcal{B}}$ *where* $\boldsymbol{\mathcal{B}} \subseteq \mathbb{R}^p$ *is compact*

(iii) $\ln f(y_i|\boldsymbol{\beta})$ *are continuous at each* $\boldsymbol{\beta} \in \boldsymbol{\mathcal{B}}$ *with probability 1*

(iv) $E[\sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} |\ln f(y_i|\boldsymbol{\beta})|] < \infty$

then

$$\hat{\boldsymbol{\beta}}^{(m)} \xrightarrow{a.s.} \boldsymbol{\beta}_0 \tag{24}$$

where $\hat{\boldsymbol{\beta}}^{(m)}$ *maximizes the objective function* $\sum_{i=1}^{n} \ln \hat{f}_n^{(m)}(y_i|\boldsymbol{\beta})$ *subject to* $\boldsymbol{\beta} \in \boldsymbol{\mathcal{B}}$

Proof: as in Theorem 2.5 in (Newey and McFadden, 1994, p. 2131), I proceed by verifying the conditions in Lemma 3.3. Note that Conditions (i), (ii), and (iii) of Lemma 3.3 are conditions on the density and the parameter space of $\boldsymbol{\mathcal{B}}$; in Theorem 2.5, Newey and McFadden (1994) verify that these hold under the usual regularity conditions of MLE (i.e. Conditions (i)-(iv) of this theorem). Condition (iv) of Lemma 3.3 implies that we have to prove that

$$\sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} \left\| \frac{1}{n} \sum_{i=1}^{n} \ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) - E[\ln f(y_i|\boldsymbol{\beta})] \right\| \xrightarrow{a.s.} 0. \tag{25}$$

To that end, first note that since $\hat{f}_n^{(1)} = \hat{f}_{n,OLS}$, we have by Lemma 3.1

$$\sup_{x\in\mathbb{R}} \left\| \hat{f}_n^{(1)}(x) - f(x) \right\| \overset{a.s.}{\to} 0. \tag{26}$$

Now note that

$$\sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} \left\| \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) - f(y_i|\boldsymbol{\beta}) \right\| \leq \sup_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\| \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) - f(y_i|\boldsymbol{\beta}) \right\| \leq \sup_{x\in\mathbb{R}} \left\| \hat{f}_n^{(1)}(x) - f(x) \right\| \tag{27}$$

implying that

$$\sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} \left\| \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) - f(y_i|\boldsymbol{\beta}) \right\| \overset{a.s.}{\to} 0. \tag{28}$$

Condition (iii) implies that $\inf_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} f(y_i|\boldsymbol{\beta}) > 0$. Thus, $\exists\ \varepsilon > 0$ such that $\inf_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} f(y_i|\boldsymbol{\beta}) > \varepsilon$. Also, by (28) for any $\varepsilon > 0$,

$$\Pr\left(\lim_{n\to\infty} \sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} \left\| \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) - f(y_i|\boldsymbol{\beta}) \right\| < \varepsilon \right) = 1 \implies \Pr\left(\lim_{n\to\infty} \inf_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) > 0 \right) = 1. \tag{29}$$

This, together with Condition (ii), ensures that for n large enough both $\ln f(y_i|\boldsymbol{\beta})$ and $\ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta})$ are uniformly continuous with probability 1 such that by the uniform continuous mapping theorem (Kasy, 2015, p. 9)

$$\sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} \left\| \ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) - \ln f(y_i|\boldsymbol{\beta}) \right\| \overset{a.s.}{\to} 0. \tag{30}$$

Note that by Condition (ii) and (iii) $\ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta})$ is bounded and we may invoke the uniform law of large numbers such that,

$$\sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} \left\| \frac{1}{n}\sum_{i=1}^{n} \ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) - E[\ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta})] \right\| \overset{a.s.}{\to} 0. \tag{31}$$

Also, by Lemma 3.4

$$\lim_{n\to\infty} E[\ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta})] = E[\ln f(y_i|\boldsymbol{\beta})] \tag{32}$$

Now define the following variables

$$A \triangleq \left\| \frac{1}{n}\sum_{i=1}^{n} \ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) - E[\ln f(y_i|\boldsymbol{\beta})] \right\| \tag{33}$$

$$A_1 \triangleq \left\| \frac{1}{n}\sum_{i=1}^{n} \ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) - E[\ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta})] \right\| \tag{34}$$

$$A_2 \triangleq \left\| E[\ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta})] - E[\ln f(y_i|\boldsymbol{\beta})] \right\|. \tag{35}$$

Then, by the triangle inequality ($\|u+v\| \leq \|u\| + \|v\|$), we have $A \leq A_1 + A_2$, and by (31) and (32), $\sup_{\beta\in\mathcal{B}} A_1 \overset{a.s.}{\to} 0$ and $\lim_{n\to\infty} \sup_{\beta\in\mathcal{B}} A_2 = 0$. Condition (iv) of Lemma 3.3 follows:

$$\sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} \left\| \frac{1}{n}\sum_{i=1}^{n} \ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) - E[\ln f(y_i|\boldsymbol{\beta})] \right\| \overset{a.s.}{\to} 0. \tag{36}$$

Thus, by Lemma 3.3,

$$\hat{\boldsymbol{\beta}}^{(1)} \xrightarrow{a.s.} \boldsymbol{\beta}_0. \tag{37}$$

For the sake of completeness, I prove also that the constraint $c(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\left[y_i - \boldsymbol{x_i'}\boldsymbol{\beta}\right] = 0$ does not affect this result. Let $\mathcal{C} \subseteq \boldsymbol{\mathcal{B}}$ be the subset for which $c(\boldsymbol{\beta}) = 0$. That is, $\boldsymbol{\mathcal{C}} = \{\boldsymbol{\beta} \in \boldsymbol{\mathcal{B}} : c(\boldsymbol{\beta}) = 0\}$. First note that the set $\boldsymbol{\mathcal{C}}$ is the level set of the continuous function $c(\boldsymbol{\beta})$ such that $\boldsymbol{\mathcal{C}}$ is closed. Also, $\boldsymbol{\mathcal{C}}$ is bounded since $\boldsymbol{\mathcal{C}} \subseteq \boldsymbol{\mathcal{B}}$ and $\boldsymbol{\mathcal{B}}$ is bounded. Hence, $\boldsymbol{\mathcal{C}}$ is compact such that $\hat{\boldsymbol{\beta}}^{(1)} = \arg\sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{C}}}\sum_{i=1}^{n}\ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta})$. Denote $\tilde{\boldsymbol{\beta}} = \arg\sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}}\sum_{i=1}^{n}\ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta})$ as the global maximizer of the objective function over $\boldsymbol{\mathcal{B}}$. Newey and McFadden (1994, p. 2122) show that for (37) to hold, it suffices to prove that

$$\frac{1}{n}\sum_{i=1}^{n}\ln \hat{f}_n^{(1)}(y_i|\hat{\boldsymbol{\beta}}^{(1)}) \xrightarrow{a.s.} \frac{1}{n}\sum_{i=1}^{n}\ln \hat{f}_n^{(1)}(y_i|\tilde{\boldsymbol{\beta}}) \tag{38}$$

holds. For that purpose, define:

$$B \triangleq \left\| \sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{C}}} \frac{1}{n}\sum_{i=1}^{n}\ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) - \sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} \frac{1}{n}\sum_{i=1}^{n}\ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) \right\| \tag{39}$$

$$B_1 \triangleq \left\| \sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{C}}} \frac{1}{n}\sum_{i=1}^{n}\ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) - \sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{C}}} E[\ln f(y_i|\boldsymbol{\beta})] \right\| \tag{40}$$

$$B_2 \triangleq \left\| \sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{C}}} E[\ln f(y_i|\boldsymbol{\beta})] - \sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} E[\ln f(y_i|\boldsymbol{\beta})] \right\| \tag{41}$$

$$B_3 \triangleq \left\| \sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} E[\ln f(y_i|\boldsymbol{\beta})] - \sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} \frac{1}{n}\sum_{i=1}^{n}\ln \hat{f}_n^{(1)}(y_i|\boldsymbol{\beta}) \right\|. \tag{42}$$

Again by the triangle equality, $B \leq B_1 + B_2 + B_3$. From (36), it is easy to show that $B_1 \xrightarrow{a.s} 0$ and $B_3 \xrightarrow{a.s} 0$. To show that $B_2 \xrightarrow{a.s} 0$, first observe that by Conditions (i) and (ii) of Lemma 3.1 and the strong law of large numbers,

$$\Pr\left(\lim_{n\to\infty}\left[\frac{1}{n}\sum_{i=1}^{n}\left[y_i - \boldsymbol{x_i'}\boldsymbol{\beta}_0\right]\right] = 0\right) = 1. \tag{43}$$

This implies

$$\Pr\left(\lim_{n\to\infty}\boldsymbol{\beta}_0 \in \boldsymbol{\mathcal{C}}\right) = 1 \implies \Pr\left(\lim_{n\to\infty}\left[\arg\sup_{\boldsymbol{\beta}\in\boldsymbol{\mathcal{B}}} E[\ln f(y_i|\boldsymbol{\beta})]\right] \in \boldsymbol{\mathcal{C}}\right) = 1 \tag{44}$$

$$\implies \Pr\left(\lim_{n\to\infty} B_2 = 0\right) = 1 \tag{45}$$

and the last implies by definition of almost sure convergence that $B_2 \xrightarrow{a.s} 0$. Hence, $B \xrightarrow{a.s} 0$ and the constraint does not affect the result.

Lastly, to show that the algorithm asymptotically converges to $\boldsymbol{\beta}_0$, remark that (37) implies by Lemma 3.2 that $\sup_{x\in\mathbb{R}}\left\|\hat{f}_n^{(2)}(x) - f(x)\right\| \xrightarrow{a.s.} 0$ where $\hat{f}_n^{(2)}(x)$ is the kernel density estimator based on the residuals corresponding to $\hat{\boldsymbol{\beta}}^{(1)}$. Thus, by identical reasoning, we obtain $\hat{\boldsymbol{\beta}}^{(m)} \xrightarrow{a.s.} \boldsymbol{\beta}_0$ for $m = 1, 2, .., M$. $\square$

Remark: Conditions (i)-(iv) are the regularity conditions that are necessary for the convergence of MLE under the true density. Thus, the only additional conditions imposed are those in Lemma 3.1 of which Condition (i) of zero mean goes without loss of generality in the context of linear regression as we can always adjust the intercept parameter in β if the center of f is not zero. Condition (ii) may be restrictive in some cases as it rules out, for instance, the $t(v)$-distribution with $1 < v \leq 2$. However, in the numerical study in Section 5.3, we observe that RKDRE performs well for $t(2)$. In fact, its performance is best of all considered estimators under that distribution. Hence, the practical use of RKDRE does not seem to be restricted to distributions with finite variance. Conditions (iii) and (iv) are easy to verify in practice, and Conditions (v)-(vii) are technical requirements on the kernel and bandwidth. Note that (v) is not satisfied by the Gaussian kernel since that kernel does not have bounded support. In practice, however, the Gaussian kernel entails a significant computational advantage (see Section 4). As Silverman (1986, p. 43) notes, the kernels vary little in performance and it is legitimate (and even desirable) to base the choice of kernel on other considerations such as the computational effort involved.

3.3 Asymptotic normality and efficiency

Theorem 3.6. *If model* (1) *holds,* $\varepsilon_1, \varepsilon_2, .., \varepsilon_n$ *are i.i.d. with unknown density* $f(x)$ *where* f *is a continuous function symmetric around zero with bounded continuous derivatives that satisfies*

(i) $\int x f(x) dx = 0$

(ii) $0 < \int \left|x^3\right| f(x) dx < \infty$,

(iii) $E\left[\left(\frac{\partial \ln f(x)}{\partial x}\right)^2 + \frac{\partial^2 \ln f(x)}{\partial x^2} + \frac{\partial^3 \ln f(x)}{\partial x^3}\right] < \infty$

$\{\boldsymbol{x_i}\}_{i=1}^{\infty}$ *satisfy,*

(iv) $\exists\ 0 < M < \infty$ *such that* $\left\|\boldsymbol{x_i}\right\| < M\ \forall\ i = 1, 2, .., n$

(v) $\lim_{n\to\infty} \boldsymbol{S_n} = \boldsymbol{Q}$ *where* $\boldsymbol{S_n} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x_i x_i'}$,

$K(x)$ *is a symmetric and four times continuously differentiable function such that*

(vi) $\exists\ 0 < \rho < \infty$ *such that* $K(x) = 0\ \forall\ x : \|x\| \geq \rho$

holds, and

(vi) *when* $n \to \infty$, $nh_n^4 \to \infty$ *and* $nh_n^8 \to 0$,

(vii) $\hat{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}_0 = O_p(n^{-\frac{1}{2}})$,

then $\hat{\boldsymbol{\beta}}^{(m)}$ *for* $m = 1, 2.., M$ *is asymptotically normal and efficient. That is,*

$$\sqrt{n}(\hat{\boldsymbol{\beta}}^{(m)} - \boldsymbol{\beta}_0) \xrightarrow{d} \mathcal{N}\left(0, I_{\beta\beta}^{-1}\right). \tag{46}$$

Proof: To show that $\hat{\boldsymbol{\beta}}^{(1)}$ is asymptotically normal and efficient, see the proof of Theorem 2.1 in (Yao and Zhao, 2013). Then, the normality and efficiency of the following iterations follow trivially. The only condition on the initial estimator $\hat{\boldsymbol{\beta}}^{(0)}$ is that $\hat{\boldsymbol{\beta}}^{(0)} - \boldsymbol{\beta}_0 = O_p(n^{-\frac{1}{2}})$. For $\hat{\boldsymbol{\beta}}^{(1)}$ this follows from the proof in (Yao and Zhao, 2013). Hence, all following estimates also satisfy (46). □

Remark: As Yao and Zhao (2013, p. 4506) note, under Condition (i), (ii) and (iv) of Theorem 3.6, the condition on the initial estimate, i.e. Condition (viii) of Theorem 3.6, is satisfied for the OLS estimator. In the proof of Theorem 3.6, I have made the simplification to disregard the effect of the constraint that the residuals sum to zero. One might expect that this constraint yields a theoretical disadvantage in the sense that the Cramér-Rao lower bound is not achieved. A derivation of the effect of the constraint on normality and efficiency is not attempted here, and is as such left as a topic of further study.

4 EM algorithm

4.1 EM algorithm (Yao and Zhao, 2013)

I use an adapted version of the expectation-maximization (EM) algorithm suggested by Yao and Zhao (2013) to render the proposed algorithm computationally feasible. For the purpose of this section, let $\hat{\boldsymbol{\beta}}_{(k)}^{(m)}$ be the k-th iteration of the EM algorithm for the m-th step of the RKDRE algorithm. Also, for brevity, $K_h(x) = \frac{1}{h_n} K\left(\frac{x}{h_n}\right)$. Then, Yao and Zhao (2013) update the parameter as follows:

E-step:

$$p_{ij,(k+1)}^{(m)} = \frac{K_h\left(y_i - \boldsymbol{x_i'}\boldsymbol{\beta}_{(k)}^{(m)} - e_j^{(m-1)}\right)}{\sum_{l \neq i} K_h\left(y_i - \boldsymbol{x_i'}\boldsymbol{\beta}_{(k)}^{(m)} - e_l^{(m-1)}\right)}, \quad j \neq i \tag{47}$$

M-step:

$$\hat{\boldsymbol{\beta}}_{(k+1)}^{(m)} = \arg\max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \sum_{j \neq i} \left[p_{ij,(k+1)}^{(m)} \ln K_h\left(y_i - \boldsymbol{x_i'}\boldsymbol{\beta} - e_j^{(m-1)}\right)\right] \tag{48}$$

which has an analytical solution in case the Gaussian kernel, i.e. $K(x) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}x^2\right\}$, is used. Yao and Zhao (2013) show that the steps constitute an EM algorithm for the optimization of the log-likelihood over a leave-one-out kernel density estimate defined as

$$\tilde{f}_n(y_i|\boldsymbol{\beta}) = \frac{1}{(n-1)h_n} \sum_{j \neq i}^{n} K\left(\frac{y_i - \boldsymbol{x_i'}\boldsymbol{\beta} - e_j^{(m-1)}}{h_n}\right). \tag{49}$$

That is, $\tilde{Q}(\hat{\boldsymbol{\beta}}_{(k+1)}^{(m)}) \geq \tilde{Q}(\hat{\boldsymbol{\beta}}_{(k)}^{(m)})$ where

$$\tilde{Q}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \ln \frac{1}{(n-1)h_n} \sum_{j \neq i}^{n} K\left(\frac{y_i - \boldsymbol{x_i'}\boldsymbol{\beta} - e_j^{(m-1)}}{h_n}\right). \tag{50}$$

The EM algorithm entails a significant computational advantage over optimization of 50 by means of non-linear optimization routines.

4.2 Constrained EM algorithm

In contrast to the approach of Yao and Zhao (2013) as described above, I use a full-kernel method (where the kernel density is based on all n observations). The intuition behind that choice is as follows; consider the case where a certain residual $e_j^{(m-1)}$ is extremely large. In the full-kernel method, $p_{ij,(k+1)}^{(m)}$ would in that case be close to zero for all $i \neq j$ and close to one for $i = j$. This implies that the effect of the extreme residual is limited to the observation for which the following iteration of $\boldsymbol{\beta}$ is likely to lead to a residual that is similar in magnitude. Hence, the effect of the large residual on the maximization in (48) is small. In the leave-one-out method, the effect of the residual may be considerably larger as $p_{ij,(k+1)}^{(m)}$ is likely to have a substantial positive value for several observations. Then, the extreme residual would have a much larger influence on the estimation. This intuition is corroborated by numerical results. I find that in practice, the

performance of the leave-one-out method is inferior to the full-kernel method. In fact, the results of the numerical study shown in (Yao and Zhao, 2013) are also obtained by the full-kernel method (even though it is reported that the leave-one-out kernel is used).[6]

Another difference to the algorithm above is the imposed constraint that the residuals sum to zero, i.e. $c(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \boldsymbol{x_i'}\boldsymbol{\beta}) = 0$. This constraint is critical for the convergence of the algorithm as the intercept may 'wander off' if it is not imposed. Thus, I maximize

$$\hat{Q}(\boldsymbol{\beta}) = \sum_{i=1}^{n} \ln \frac{1}{nh_n} \sum_{j=1}^{n} K\left(\frac{y_i - \boldsymbol{x_i'}\boldsymbol{\beta} - e_j^{(m-1)}}{h_n}\right) \quad \text{s.t.} \quad \sum_{i=1}^{n}\left[y_i - \boldsymbol{x_i'}\boldsymbol{\beta}\right] = 0. \tag{51}$$

When K is the Gaussian kernel, this can be maximized by the following EM algorithm:

E-step:

$$p_{ij,(k+1)}^{(m)} = \frac{\exp\left\{-\frac{1}{2h_n^2}\left(y_i - \boldsymbol{x_i'}\boldsymbol{\beta}_{(k)}^{(m)} - e_j^{(m-1)}\right)^2\right\}}{\sum_{l=1}^{n}\exp\left\{-\frac{1}{2h_n^2}\left(y_i - \boldsymbol{x_i'}\boldsymbol{\beta}_{(k)}^{(m)} - e_l^{(m-1)}\right)^2\right\}} \tag{52}$$

M-step:

$$\begin{aligned}\hat{\boldsymbol{\beta}}_{(k+1)}^{(m)} = \hat{\boldsymbol{\beta}}_{OLS} &- \left[\sum_{i=1}^{n}\boldsymbol{x_i}\boldsymbol{x_i'}\right]^{-1}\sum_{i=1}^{n}\boldsymbol{x_i}\sum_{j=1}^{n}p_{ij,(k+1)}^{(m)}e_j^{(m-1)} \\ &-\frac{1}{n}\left[\sum_{i=1}^{n}\boldsymbol{x_i}\boldsymbol{x_i'}\right]^{-1}\sum_{i=1}^{n}\boldsymbol{x_i}\sum_{i=1}^{n}\sum_{j=1}^{n}p_{ij,(k+1)}^{(m)}e_j^{(m-1)}\end{aligned} \tag{53}$$

Theorem 4.1. *The objective function* (51) *decreases after each iteration of* (52) *and* (53) *until a fixed point is reached.*

Proof: under the Gaussian kernel, the constraint that $c(\boldsymbol{\beta}) = 0$, and a full-kernel method, the M-step in (48) becomes

$$\hat{\boldsymbol{\beta}}_{(k+1)}^{(m)} = \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n}\sum_{j=1}^{n}\left[p_{ij,(k+1)}^{(m)}\left(y_i - \boldsymbol{x_i'}\boldsymbol{\beta} - e_j^{(m-1)}\right)^2\right] \quad \text{s.t.} \quad \sum_{i=1}^{n}\left[y_i - \boldsymbol{x_i'}\boldsymbol{\beta}\right] = 0. \tag{54}$$

This can be solved by Lagrangian optimization. Define the Lagrangian $\mathcal{L}$ as

$$\mathcal{L}(\boldsymbol{\beta}, \lambda) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left[p_{ij,(k+1)}^{(m)}\left(y_i - \boldsymbol{x_i'}\boldsymbol{\beta} - e_j^{(m-1)}\right)^2\right] - \lambda\sum_{i=1}^{n}\left[y_i - \boldsymbol{x_i'}\boldsymbol{\beta}\right] \tag{55}$$

with first-order conditions

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\beta}} = -2\sum_{i=1}^{n}\sum_{j=1}^{n}\left[p_{ij,(k+1)}^{(m)}\boldsymbol{x_i}\left(y_i - \boldsymbol{x_i'}\boldsymbol{\beta} - e_j^{(m-1)}\right)\right] - \lambda\sum_{i=1}^{n}\boldsymbol{x_i} = 0 \tag{56}$$

$$\frac{\partial\mathcal{L}}{\partial\lambda} = \sum_{i=1}^{n}\left[y_i - \boldsymbol{x_i'}\boldsymbol{\beta}\right] = 0. \tag{57}$$

[6] I thank Weixin Yao of the University of California, Riverside for sharing his MATLAB code.

Since the first element of each explanatory variable, $x_{i1} = 1$, the first element of the first-order condition in (56) implies

$$\begin{aligned}
\lambda =& -\frac{2}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[p_{ij,(k+1)}^{(m)}\left(y_i - \boldsymbol{x_i'}\boldsymbol{\beta} - e_j^{(m-1)}\right)\right] \\
=& -\frac{2}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}\left[p_{ij,(k+1)}^{(m)}\left(y_i - \boldsymbol{x_i'}\boldsymbol{\beta}\right)\right] - \frac{2}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}p_{ij,(k+1)}^{(m)}e_j^{(m-1)} \\
=& -\frac{2}{n}\sum_{i=1}^{n}\left[y_i - \boldsymbol{x_i'}\boldsymbol{\beta}\right]\sum_{j=1}^{n}p_{ij,(k+1)}^{(m)} - \frac{2}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}p_{ij,(k+1)}^{(m)}e_j^{(m-1)} \\
=& -\frac{2}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}p_{ij,(k+1)}^{(m)}e_j^{(m-1)}
\end{aligned} \tag{58}$$

where the last equality follows from (57). Then, by plugging λ in (56), rearranging terms and using that $\sum_{j=1}^{n} p_{ij,(k+1)}^{(m)} = 1$, we obtain

$$\begin{aligned}
\hat{\boldsymbol{\beta}}_{(k+1)}^{(m)} =& \left[\sum_{i=1}^{n}\boldsymbol{x_i}\boldsymbol{x_i'}\right]^{-1}\sum_{i=1}^{n}\boldsymbol{x_i'}y_i - \left[\sum_{i=1}^{n}\boldsymbol{x_i}\boldsymbol{x_i'}\right]^{-1}\sum_{i=1}^{n}\boldsymbol{x_i}\sum_{j=1}^{n}p_{ij,(k+1)}^{(m)}e_j^{(m-1)} \\
&- \frac{1}{n}\left[\sum_{i=1}^{n}\boldsymbol{x_i}\boldsymbol{x_i'}\right]^{-1}\sum_{i=1}^{n}\boldsymbol{x_i}\sum_{i=1}^{n}\sum_{j=1}^{n}p_{ij,(k+1)}^{(m)}e_j^{(m-1)}.
\end{aligned} \tag{59}$$

Recognize that the first term is equal to $\hat{\boldsymbol{\beta}}_{OLS}$. Then, the fact that (52) and (53) are the E- and M-step, respectively, of an EM algorithm for (51) follows trivially from the proof of Theorem 2.2 in (Yao and Zhao, 2013, p.4511). □

The EM algorithm is considered converged in case $\max\left|\hat{\boldsymbol{\beta}}_{(k)}^{(m)} - \hat{\boldsymbol{\beta}}_{(k+1)}^{(m)}\right|$ is smaller than a threshold value where $\max|\boldsymbol{A}|$ denotes the largest (absolute) element in $\boldsymbol{A}$. In the m-th repetition of the RKDRE algorithm, the EM algorithm is initialized by the estimate of the $(m-1)$-th repetition. That is, $\hat{\boldsymbol{\beta}}_{(0)}^{(m)} = \hat{\boldsymbol{\beta}}^{(m-1)}$.

5 Numerical study

To assess small sample performance of the RKDRE estimator (and other semiparametric adaptive estimators), I perform several simulation studies. For the purpose of this section, the data generating process is

$$y_i = \boldsymbol{x_i'}\boldsymbol{\beta} + \varepsilon_i, \tag{60}$$

where $\boldsymbol{\beta}$ is a $p \times 1$ vector containing an intercept and the parameters corresponding to $p-1$ explanatory variables. In the comparative study in Section 5.3, I investigate performance for $p = 2$, $p = 5$, and $p = 10$. For $p = 10$, $\boldsymbol{\beta} = [1, -1, 2, -0.5, 3, 1, -1, 2, -0.5, 3]'$. For $p = 2$ and $p = 5$, $\boldsymbol{\beta}$ consists of the first two and five coefficients, respectively. The sample size n is varied over 50, 100, 500, and 1000. All simulation results are based on 500 replications. The explanatory variables in $\boldsymbol{x_i}$ are independent realizations of a standard normal distribution. For the error distributions, I draw from Hsieh and Manski (1987) and Yao and Zhao (2013). The distributions considered are

(A) standard normal distribution

(B) variance-contaminated normal distribution; $0.9\mathcal{N}(0, \frac{1}{9}) + 0.1\mathcal{N}(0, 9)$

(C) t-distribution with two degrees of freedom

(D) bimodal (symmetric) mixture of normal distributions; $0.5\mathcal{N}(-3, 1) + 0.5\mathcal{N}(3, 1)$

(E) uniform distribution on $\left[-\frac{1}{2}\sqrt{12}; \frac{1}{2}\sqrt{12}\right]$

(F) Gamma(2,2)[7]

(G) skewed mixture of normal distributions; $0.3\mathcal{N}(-1.4, 1) + 0.7\mathcal{N}(0.6, 0.16)$

(H) log-normal distribution; $\exp(Z)$ where $Z \sim \mathcal{N}(0, 1)$.

The distributions are centered and scaled to have mean zero and variance one (where necessary and possible). The $t(2)$-distribution is left unscaled as its variance is infinite. In Section 5.1 and Section 5.2, I discuss the choices for various parameters used in the simulation study. The reader that is less interested in such a discussion, may skip these subsections and proceed directly to the results of the numerical study as described in Section 5.3. Lastly, in Section 5.4, I compare the computation time of the different methods.

5.1 Implementation of existing semi-parametric estimators

All estimators discussed in Section 2 are implemented. Here, I discuss technical details on the parameters of these estimators that are necessary for replicability. I use OLS wherever an initial estimate is required. Also, I use the Gaussian kernel for all implemented kernel methods. The choice of bandwidth is discussed in Section 5.2.

For the SBS estimator, the trimming parameter t (as defined in Section 2) has to be specified. As Hsieh and Manski (1987), I find that $t = 8$ performs well for all investigated cases. Hsieh and

[7] Here, I use the shape-scale notation, i.e. if $X \sim \text{Gamma}(k, \theta)$, $E[X] = k\theta$ and $\text{Var}(X) = k\theta^2$.

Manski (1987) find for $n = 50$ and $p = 2$ that performance (in terms of mean square error) tends to increase when t is increased from 3 to 8 and tends to remain stable when t is increased from 8 to 16 (or so). When t is increased further, very rare outlying values are admitted and performance begins to deteriorate. Hsieh and Manski (1987) show that this pattern is relatively robust with respect to the error distribution. By experimentation not reported here, I find that this same pattern is observable for larger sample sizes and more explanatory variables as well such that I use $t = 8$ throughout this section.

Newey (1988) finds for the LGMM estimator that $J = 3$ works best when $n = 50$ and $p = 2$. I find that this holds for larger sample sizes and more explanatory variables as well. For certain distributions, the performance is relatively stable over J, whereas for some distributions (such as the bimodal mixture) a deviation from $J = 3$ leads to a sharp deterioration of performance. When n rises, the performance of J between 4 and 7 increases relatively to the performance of $J = 2$ and $J = 3$. However, for all cases included in the study, $J = 3$ performs either best or close to best such that I use $J = 3$ for the purpose of the simulation. Also, experimentation corroborated the finding of Newey (1988) that the LGMM estimator works better if it is based on the 'transformed' moments than if the 'raw' and/or 'weighted' moments are used. Hence, the results for the transformed method are shown here. Lastly, I observed that the above conclusions hold similarly for LGMMS such that I also use $J = 3$ transformed moments for the implementation of LGMMS.

Lastly, the implementation of YDG can be done by an EM algorithm and a non-linear optimization method. In Appendix A.1, I prove that the following EM algorithm increases the maximization criterion of YDG in (19) after each iteration.[8]

E-step:

$$p_{ij,(k+1)} = \frac{\exp\left\{-\frac{1}{2h_n^2}\left(y_i - y_j - \left(\boldsymbol{x}_{\boldsymbol{i}}^* - \boldsymbol{x}_{\boldsymbol{j}}^*\right)'\boldsymbol{\beta}^*\right)^2\right\}}{\sum_{j\neq i}^n \exp\left\{-\frac{1}{2h_n^2}\left(y_i - y_j - \left(\boldsymbol{x}_{\boldsymbol{i}}^* - \boldsymbol{x}_{\boldsymbol{j}}^*\right)'\boldsymbol{\beta}^*\right)^2\right\}}, \quad j \neq i$$

M-step:

$$\hat{\boldsymbol{\beta}}_{YDG,(k+1)}^* = \left[\sum_{i=1}^n \sum_{j\neq i}^n p_{ij,(k+1)} \left(\boldsymbol{x}_{\boldsymbol{i}}^* - \boldsymbol{x}_{\boldsymbol{j}}^*\right)\left(\boldsymbol{x}_{\boldsymbol{i}}^* - \boldsymbol{x}_{\boldsymbol{j}}^*\right)'\right]^{-1} \sum_{i=1}^n \sum_{j\neq i}^n \left(\boldsymbol{x}_{\boldsymbol{i}}^* - \boldsymbol{x}_{\boldsymbol{j}}^*\right)(y_i - y_j)$$

A disadvantage of this EM algorithm compared to the EM algorithms of (R)KDRE as formed by (52) and (53) is that it can not be simplified further, and hence computation is much slower. In fact, only for $p = 2$, the EM algorithm for YDG is marginally faster than using a non-linear optimization method (as in that case $\left(\boldsymbol{x}_{\boldsymbol{i}}^* - \boldsymbol{x}_{\boldsymbol{j}}^*\right)$ is a scalar). In all other instances, using a quasi-Newton optimization optimization method is faster. In the numerical study in Section 5.2 and 5.3, I use the quasi-Newton optimization method (BFGS) implemented in the R function `optim()`.

[8] Yao and Zhao (2013) note that an EM algorithm can be similarly used for the implementation of YDG, but this procedure is not made explicit.

The estimators are implemented by the R functions rkdre(), kdre(), rkdre(), ydg(), sbs(), and lgmm() provided in Appendix B.1-B.5. Note that lgmm() implements both LGMM and LGMMS depending on whether symmetric = FALSE or symmetric = TRUE, respectively. Also, even though I only show results for the 'transformed' LGMM(S), the code allows for the implementation of the 'raw' and 'weighted' moments as well. Lastly, ydg() allows both for non-linear optimization and optimization using the EM algorithm.

5.2 Choice of bandwidth

The problem of choosing the bandwidth parameter h is of crucial importance in density estimation (Silverman, 1986, p. 43; Pagan and Ullah, 1999, p. 49). It determines the smoothness of the estimated density. That is, if the bandwidth is low, the estimated density strongly 'follows' the data. This may come at the cost of consisting spurious patterns that do not reflect the true distribution but are merely a result of randomness. However, if the bandwidth is too high, the density will be 'oversmoothed' in the sense that it obscures the underlying structure. As Pagan and Ullah (1999) show, the oversmoothed (undersmoothed) estimate has smaller (larger) variance but larger (smaller) bias with respect to the true distribution.

One possible choice is to specify h subjectively, finding by eye the balance between under- and oversmoothing. This method is often used in data exploration (Silverman, 1986, p. 43). Surely, with regard to the proposed semiparametric estimator, practitioners would be better served if this subjective choice could be avoided by means of a rule-of-thumb. Silverman (1986) considers three different possibilities:

$$h_n = 1.06\sigma n^{-\frac{1}{5}} \tag{h.1}$$

$$h_n = 0.79Rn^{-\frac{1}{5}} \tag{h.2}$$

$$h_n = 0.90An^{-\frac{1}{5}}, \tag{h.3}$$

where σ is the standard deviation, R is the inter-quartile range, and $A = \min\left(\sigma, \frac{R}{1.34}\right)$. In practice, these values can be estimated by their sample equivalents. The fact that $h_n \propto n^{-\frac{1}{5}}$ has a technical reason; it can be shown that this choice of h_n minimizes the (approximation of) the Mean Integrated Square Error (MISE) defined as

$$\text{MISE} = \int \left[\left(\text{Bias } \hat{f}_n\right)^2 + \text{Var}\left(\hat{f}_n\right)\right] dx.$$

To render this intuitively plausible, Pagan and Ullah (1999, p. 25) show that $\left(\text{Bias}\hat{f}_n\right)^2 = O\left(h^4\right)$ and $\text{Var}\left(\hat{f}_n\right) = O\left(nh\right)^{-1}$ where O is the usual Big-O-notation. Hence, if these two terms are to be of the same order of magnitude, it must hold that $h_n \propto n^{-\frac{1}{5}}$. With respect to the proposed RKDRE algorithm, an additional advantage of specifying the bandwidth according to $h_n \propto n^{-\frac{1}{5}}$ is that the requirements on the bandwidth for almost sure convergence (i.e. $\frac{\sqrt{n}h_n}{\ln n} \to \infty$) and

Table 1: Mean (and standard deviation) of rule-of-thumb bandwidth under different distributions

	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)
$h_n = 1.06\hat{\sigma}n^{-\frac{1}{5}}$	0.420	0.405	1.116	0.419	0.420	0.417	0.454	0.401
	(0.028)	(0.098)	(0.787)	(0.014)	(0.019)	(0.048)	(0.041)	(0.129)
$h_n = 0.79\hat{R}n^{-\frac{1}{5}}$	0.415	0.160	0.531	0.586	0.530	0.380	0.359	0.210
	(0.049)	(0.021)	(0.088)	(0.028)	(0.054)	(0.053)	(0.100)	(0.037)
$h_n = 0.90\hat{A}n^{-\frac{1}{5}}$	0.341	0.136	0.452	0.356	0.357	0.318	0.300	0.178
	(0.031)	(0.018)	(0.075)	(0.012)	(0.016)	(0.040)	(0.075)	(0.031)

Standard deviations in parentheses. Results are based on 500 replications. Column headers (A)-(H) refer to the distributions described in this section. Bandwidth is estimated on OLS residuals of the samples considered in Table 2.

asymptotic normality and efficiency (i.e. $nh_n^4 \to \infty$ and $nh_n^8 \to 0$) are satisfied.[9]

If we assume that f is the density of a normal distribution and the Gaussian kernel is used, it can be shown that the rule-of-thumb in $(h.1)$ minimizes the approximated MISE (AMISE). Naturally, had we known that the errors are normally distributed, density estimation would not have been necessary. Therefore, different rule-of-thumbs are proposed, especially to increase performance in the presence of skewness, fat tails, and bi- or multi-modal distributions. Silverman (1986) finds that $(h.2)$ increases performance with respect to $(h.1)$ in case of skewness and heavy tails, but makes matters worse for bi-modal distributions. In that sense, he finds that $(h.3)$ is the 'best of both worlds', performing well for a wide range of range densities. Table 1 shows the mean (and standard deviation) of the bandwidth obtained by the three different rules-of-thumb on the OLS residuals for distribution (A)-(H). In general, we see that $(h.2)$ and $(h.3)$ are smaller than $(h.1)$. As Silverman (1986), we find that $(h.2)$ fails for the bi-modal distribution (E), since it leads to more smoothing than $(h.1)$ whereas in general less smoothing is required to properly capture the bimodality of the distribution.

Despite, the crucial importance of the bandwidth parameter, Yao and Zhao (2013) and Yuan and De Gooijer (2007) do not investigate the effect of different bandwidth parameters on the performance of YDG and KDRE, respectively. Also, even though Hsieh and Manski (1987) show simulation results for different constant values of h, to the best of my knowledge, no study has explored the performance of the SBS estimator under bandwidths based on rules-of-thumb. Hence, Table 2 shows the root mean square error (RMSE) of the estimators under different bandwidth parameters not only for RKDRE, but also for KDRE, YDG and SBS. The analysis is done for $n = 100$ and $p = 2$. This choice is based on computational convenience. However, when testing for several scenarios with larger sample size and/or a larger number of explanatory variables, I found that conclusions do not depend on a specific choice for n and/or p. The 'optimal' bandwidth found in this section is used in the analysis in Section 5.3.

[9] See Condition (vii) of Lemma 3.1 and C4 in (Yao and Zhao, 2013, p. 4506)

Table 2: Root mean square error of kernel-based estimators for different bandwidth values

	Intercept				Slope			
	RKDRE	KDRE	YDG	SBS	RKDRE	KDRE	YDG	SBS
(A) Standard normal distribution								
$h = 0.1$	0.1015	0.1014	0.1067	0.1021	0.1270	0.1023	0.3505	0.1134
$h = 0.2$	0.1023	0.1014	0.1015	0.1026	0.1368	0.1061	0.2036	0.1230
$h = 0.3$	0.1021	0.1027	0.1015	0.1049	0.1209	0.1072	0.1383	0.1180
$h = 0.4$	0.1019	0.1039	0.1015	0.1066	0.1109	0.1057	0.1174	0.1118
$h = 0.5$	0.1017	0.1048	0.1016	0.1084	0.1061	0.1041	0.1084	0.1078
$h = 0.6$	0.1017	0.1056	0.1016	0.1107	0.1037	0.1029	0.1045	0.1056
$h = 0.7$	0.1016	0.1063	0.1016	0.1136	0.1025	0.1022	0.1027	0.1042
$h = 0.8$	0.1016	0.1068	0.1016	0.1167	0.1019	0.1017	0.1020	0.1034
$h = 0.9$	0.1015	0.1070	0.1016	0.1201	0.1015	0.1015	0.1016	0.1028
$h_n = 1.06\hat{\sigma}n^{-\frac{1}{5}}$	0.1018	0.1039	0.1016	0.1066	0.1093	0.1054	0.1136	0.1106
$h_n = 0.79\hat{R}n^{-\frac{1}{5}}$	0.1018	0.1038	0.1017	0.1067	0.1100	0.1056	0.1160	0.1112
$h_n = 0.9\hat{A}n^{-\frac{1}{5}}$	0.1020	0.1031	0.1017	0.1054	0.1154	0.1068	0.1267	0.1153
(B) Variance-contaminated normal distribution								
$h = 0.1$	0.1011	0.1029	0.1657*	0.1028	0.0462	0.0773	1.4372*	0.0512
$h = 0.2$	0.1012	0.1034	0.1358*	0.1031	0.0410	0.0500	1.0093*	0.0491
$h = 0.3$	0.1012	0.1038	0.1140	0.1060	0.0404	0.0436	0.4915	0.0687
$h = 0.4$	0.1013	0.1046	0.1029	0.1109	0.0409	0.0429	0.2027	0.0990
$h = 0.5$	0.1013	0.1058	0.1014	0.1188	0.0421	0.0437	0.0618	0.1356
$h = 0.6$	0.1013	0.1075	0.1013	0.1303	0.0439	0.0452	0.0562	0.1741
$h = 0.7$	0.1013	0.1095	0.1013	0.1457	0.0462	0.0474	0.0518	0.2110
$h = 0.8$	0.1013	0.1117	0.1013	0.1643	0.0487	0.0497	0.0509	0.2426
$h = 0.9$	0.1013	0.1141	0.1013	0.1848	0.0513	0.0522	0.0504	0.2627
$h_n = 1.06\hat{\sigma}n^{-\frac{1}{5}}$	0.1013	0.1047	0.1016	0.1123	0.0413	0.0431	0.0856	0.1293
$h_n = 0.79\hat{R}n^{-\frac{1}{5}}$	0.1011	0.1033	0.1667*	0.1023	0.0418	0.0539	1.2722*	0.0465
$h_n = 0.9\hat{A}n^{-\frac{1}{5}}$	0.1011	0.1032	0.1581*	0.1025	0.0433	0.0594	1.3013*	0.0459
(C) t-distribution with two degrees of freedom								
$h = 0.1$	0.3367	0.3374	0.2720*	0.3377	0.2890	0.3105	1.9816*	0.3059
$h = 0.2$	0.3359	0.3375	0.3394*	0.3364	0.2255	0.2996	2.6408*	0.2701
$h = 0.3$	0.3320	0.3406	0.3483*	0.3364	0.1716	0.2764	2.8149*	0.2275
$h = 0.4$	0.3313	0.3420	0.4024*	0.3375	0.1543	0.2446	3.3200*	0.1936
$h = 0.5$	0.3312	0.3463	0.4203*	0.3384	0.1466	0.2127	3.3655*	0.1751
$h = 0.6$	0.3311	0.3440	0.3846*	0.3394	0.1428	0.1766	3.1670*	0.1659
$h = 0.7$	0.3312	0.3445	0.3824*	0.3406	0.1408	0.1669	3.0522*	0.1618
$h = 0.8$	0.3313	0.3449	0.3504*	0.3422	0.1403	0.1605	2.7421*	0.1620
$h = 0.9$	0.3314	0.3453	0.3650*	0.3444	0.1407	0.1566	2.6561*	0.1662
$h_n = 1.06\hat{\sigma}n^{-\frac{1}{5}}$	0.3330	0.3402	0.3713	0.4004	0.1478	0.1533	1.6033	1.3975
$h_n = 0.79\hat{R}n^{-\frac{1}{5}}$	0.3314	0.3436	0.4141*	0.3384	0.1451	0.1725	3.4324*	0.1538
$h_n = 0.9\hat{A}n^{-\frac{1}{5}}$	0.3314	0.3443	0.3895*	0.3377	0.1503	0.1852	3.2577*	0.1586
(D) Bi-modal mixture of normal distributions								
$h = 0.1$	0.1067	0.1071	0.1070	0.1071	0.0438	0.0621	0.0532	0.0518
$h = 0.2$	0.1068	0.1063	0.1069	0.1066	0.0361	0.0375	0.0358	0.0525
$h = 0.3$	0.1069	0.1061	0.1069	0.1065	0.0345	0.0349	0.0344	0.0899
$h = 0.4$	0.1069	0.1053	0.1069	0.1050	0.0341	0.0347	0.0343	0.1498
$h = 0.5$	0.1069	0.1032	0.1069	0.0983	0.0344	0.0356	0.0344	0.2246
$h = 0.6$	0.1069	0.0992	0.1069	0.0827	0.0359	0.0384	0.0355	0.2850
$h = 0.7$	0.1068	0.0931	0.1068	0.0670	0.0404	0.0445	0.0393	0.2790
$h = 0.8$	0.1068	0.0859	0.1068	0.0684	0.0498	0.0543	0.0486	0.1993
$h = 0.9$	0.1069	0.0796	0.1068	0.0714	0.0623	0.0656	0.0626	0.1095
$h_n = 1.06\hat{\sigma}n^{-\frac{1}{5}}$	0.1069	0.1051	0.1069	0.1046	0.0341	0.0347	0.0577	0.1617
$h_n = 0.79\hat{R}n^{-\frac{1}{5}}$	0.1069	0.1002	0.1069	0.0864	0.0353	0.0373	0.0590	0.2804
$h_n = 0.9\hat{A}n^{-\frac{1}{5}}$	0.1069	0.1058	0.1069	0.1061	0.0342	0.0346	0.0577	0.1194

Continued on next page

Table 2 – continued from previous page

	Intercept				Slope			
	RKDRE	KDRE	YDG	SBS	RKDRE	KDRE	YDG	SBS
(E) Uniform distribution								
$h = 0.1$	0.0955	0.0949	0.0955	0.0927	0.0831	0.0942	0.0689	0.0921
$h = 0.2$	0.0956	0.0908	0.0956	0.0858	0.0677	0.0771	0.0596	0.0841
$h = 0.3$	0.0957	0.0861	0.0957	0.0752	0.0657	0.0727	0.0633	0.0883
$h = 0.4$	0.0958	0.0817	0.0958	0.0650	0.0704	0.0748	0.0688	0.0888
$h = 0.5$	0.0958	0.0780	0.0959	0.0574	0.0755	0.0786	0.0744	0.0869
$h = 0.6$	0.0959	0.0755	0.0959	0.0528	0.0806	0.0827	0.0801	0.0841
$h = 0.7$	0.0960	0.0739	0.0960	0.0514	0.0853	0.0868	0.0857	0.0817
$h = 0.8$	0.0960	0.0733	0.0960	0.0524	0.0896	0.0906	0.0912	0.0814
$h = 0.9$	0.0960	0.0733	0.0961	0.0542	0.0932	0.0938	0.0962	0.0835
$h_n = 1.06\hat{\sigma}n^{-\frac{1}{5}}$	0.0958	0.0808	0.0958	0.0630	0.0712	0.0754	0.0697	0.0887
$h_n = 0.79\hat{R}n^{-\frac{1}{5}}$	0.0958	0.0773	0.0959	0.0561	0.0763	0.0793	0.0751	0.0835
$h_n = 0.9\hat{A}n^{-\frac{1}{5}}$	0.0958	0.0834	0.0958	0.0689	0.0683	0.0735	0.0662	0.0897
(F) Gamma(2,2)								
$h = 0.1$	0.1072	0.1071	0.1162*	0.1062	0.0881	0.0970	0.6053*	0.0930
$h = 0.2$	0.1069	0.1079	0.1086	0.1122	0.0714	0.0804	0.2804	0.0790
$h = 0.3$	0.1068	0.1151	0.1074	0.1377	0.0680	0.0738	0.1420	0.0764
$h = 0.4$	0.1069	0.1257	0.1067	0.1715	0.0695	0.0731	0.0824	0.0771
$h = 0.5$	0.1070	0.1366	0.1068	0.2062	0.0720	0.0747	0.0771	0.0788
$h = 0.6$	0.1071	0.1463	0.1069	0.2389	0.0749	0.0771	0.0765	0.0807
$h = 0.7$	0.1071	0.1541	0.1070	0.2685	0.0779	0.0797	0.0777	0.0825
$h = 0.8$	0.1071	0.1600	0.1071	0.2950	0.0808	0.0822	0.0794	0.0841
$h = 0.9$	0.1072	0.1642	0.1071	0.3189	0.0834	0.0845	0.0813	0.0854
$h_n = 1.06\hat{\sigma}n^{-\frac{1}{5}}$	0.1069	0.1331	0.1068	0.1890	0.0704	0.0735	0.0800	0.0786
$h_n = 0.79\hat{R}n^{-\frac{1}{5}}$	0.1069	0.1294	0.1068	0.1779	0.0690	0.0734	0.0864	0.0775
$h_n = 0.9\hat{A}n^{-\frac{1}{5}}$	0.1069	0.1224	0.1072	0.1555	0.0680	0.0736	0.1231	0.0773
(G) Skewed mixture of normal distributions								
$h = 0.1$	0.1156	0.1161	0.1185	0.1176	0.0693	0.0940	0.2290	0.0789
$h = 0.2$	0.1156	0.1199	0.1160	0.1219	0.0541	0.0640	0.0764	0.0582
$h = 0.3$	0.1156	0.1260	0.1157	0.1362	0.0506	0.0557	0.0538	0.0666
$h = 0.4$	0.1156	0.1351	0.1157	0.1655	0.0506	0.0550	0.0503	0.0868
$h = 0.5$	0.1157	0.1469	0.1157	0.2141	0.0524	0.0570	0.0507	0.1089
$h = 0.6$	0.1157	0.1611	0.1157	0.2805	0.0554	0.0604	0.0524	0.1249
$h = 0.7$	0.1158	0.1766	0.1158	0.3561	0.0595	0.0648	0.0551	0.1303
$h = 0.8$	0.1159	0.1921	0.1158	0.4298	0.0645	0.0698	0.0588	0.1257
$h = 0.9$	0.1159	0.2063	0.1158	0.4925	0.0701	0.0749	0.0634	0.1144
$h_n = 1.06\hat{\sigma}n^{-\frac{1}{5}}$	0.1157	0.1453	0.1157	0.2012	0.0511	0.0557	0.0502	0.1010
$h_n = 0.79\hat{R}n^{-\frac{1}{5}}$	0.1157	0.1428	0.1157	0.1895	0.0499	0.0554	0.0518	0.0842
$h_n = 0.9\hat{A}n^{-\frac{1}{5}}$	0.1156	0.1337	0.1156	0.1567	0.0500	0.0564	0.0547	0.0717
(H) Log-normal distribution								
$h = 0.1$	0.1049	0.1064	0.1256*	0.1106	0.0259	0.0624	0.8594*	0.0508
$h = 0.2$	0.1049	0.1120	0.1488*	0.1260	0.0279	0.0387	1.0176*	0.0660
$h = 0.3$	0.1050	0.1176	0.1541*	0.1650	0.0321	0.0371	1.0343*	0.0923
$h = 0.4$	0.1051	0.1238	0.1432*	0.2056	0.0366	0.0399	0.7955*	0.1129
$h = 0.5$	0.1051	0.1296	0.1372*	0.2448	0.0410	0.0435	0.8963*	0.1309
$h = 0.6$	0.1052	0.1347	0.1425	0.2824	0.0450	0.0472	0.3867	0.1475
$h = 0.7$	0.1052	0.1390	0.1381	0.3178	0.0487	0.0506	0.3093	0.1634
$h = 0.8$	0.1052	0.1426	0.1349	0.3513	0.0520	0.0537	0.2910	0.1793
$h = 0.9$	0.1052	0.1454	0.1057	0.3834	0.0551	0.0566	0.0579	0.1956
$h_n = 1.06\hat{\sigma}n^{-\frac{1}{5}}$	0.1051	0.1308	0.1057	0.2294	0.0377	0.0404	0.0807	0.1412
$h_n = 0.79\hat{R}n^{-\frac{1}{5}}$	0.1050	0.1163	0.1377*	0.1430	0.0283	0.0374	0.9028*	0.0721
$h_n = 0.9\hat{A}n^{-\frac{1}{5}}$	0.1049	0.1145	0.1452*	0.1306	0.0271	0.0394	0.9646*	0.0624

The sample size $n = 100$ and the number of variables (including intercept) $p = 2$. Results are based on 500 replications. For scenarios marked with *, the YDG estimator failed for some replications. In that case, the results are averages over the replications where it did not fail.

For RKDRE, we observe that the accuracy of the intercept estimate is very much robust against different values of the bandwidth. This is likely to be due to the fact that RKDRE uses the constraint that the residuals sum to zero (and the intercept mostly accounts for the satisfaction of this constraint). Hence, the bandwidth choice should be based on the effect on the accuracy of the slope parameter. For that matter, we see that for the standard normal distribution (A), the RMSE is a roughly monotonically decreasing function of h. To see why this makes sense, note that the Gaussian kernel is used such that the larger h, the closer the density represents a normal density. In Table 3 in Section 5.3, one may verify that the RMSE of OLS under this scenario is roughly 0.100. Hence, we see that if h increases the RMSE of the slope approaches the RMSE of OLS which is the MLE under (A), and thus, asymptotically efficient. In general, Table 2 shows that the rule-of-thumb based on the normal distribution as in (h.1) performs reasonable for RKDRE. However, for the skewed distribution, i.e. (F), (G), and (H), the robust rule-of-thumb as in (h.3) works best. Therefore, with regards to the RKDRE, I use (h.1) and (h.3) for symmetric and skewed distributions, respectively. I argue that such a distinction is reasonable since skewness can be detected relatively easily in practice. I also experimented with updating the bandwidth parameter of RKDRE after each step of the algorithm, but I found that this does not entail a meaningful change in performance. As an advantage of RKDRE over the other kernel estimators, we find that its performance is generally least sensitive to the value of the bandwidth.

As for RKDRE, KDRE performs best under the normal rule-of-thumb (h.1) for symmetric distributions. However, for asymmetric distributions we do not see an improvement (on the slope estimation) if we use (h.2) or (h.3) instead such that I use (h.1) throughout all distributions.

Table 2 shows that (h.1) works best for the YDG estimator, too. In particular, we find that the estimator may fail to obtain an estimate for lower values of the bandwidth if the distribution has heavy tails, i.e. for (B), (C), (F), and (H). To see why, consider the maximization criterion in (19) and let $\hat{\boldsymbol{\beta}}^{*(1)}$ be a starting value for the optimization routine. In case observation i is an outlier, $\left(y_i - \boldsymbol{x}_{\boldsymbol{i}}^{*\prime}\hat{\boldsymbol{\beta}}^{*(1)}\right) - \left(y_j - \boldsymbol{x}_{\boldsymbol{j}}^{*\prime}\boldsymbol{\beta}^{*(1)}\right)$ will be large (in absolute terms) for all $i \neq j$. This term is then divided by h_n, leading to a stronger inflation the closer h_n is to zero. Recall that I use the Gaussian kernel $K(x) \propto \exp\left\{-\frac{1}{2}x^2\right\}$, which for $|x| \gtrapprox 40$ is smaller than the 'smallest non-zero normalized floating-point number' in R. If that is the case for all $i \neq j$, the density estimated at i is equal to zero. Hence, taking the logarithm leads to a negative infinite value of the first function evaluation, causing the optimization routine to break down. Errors occurred in all scenarios marked with * in Table 2. For those scenarios, the percentage of replications that failed varied between 0.2% (one replication) and 57.2% (under (C)). In case of errors, the results shown are based on replications where the estimator did not fail. Not surprisingly, we see that the YDG estimator, even if it does not fail, performs in general badly in the presence of outliers. This fact is likely to explain why Yuan and De Gooijer (2007, p .856) shows performance of the estimator under a normal mixture distribution that is truncated on both tails.

Lastly, the SBS estimator generally shows the smallest RMSE for (h.3). In fact, the difference in

performance between ($h.2$) and the other two rules-of-thumb can be quite large, e.g. for (C), (D), (G), and (H). In general, we see that SBS is relatively sensitive to an improper value of the bandwidth. For instance, using ($h.2$) instead of ($h.3$) on the bi-modal distribution more than doubles the RMSE for SBS. For the other three estimators, we also see that ($h.2$) performs worst (on the slope) under (D), but the deterioration is not nearly as large.

Naturally, there are many more ways in which one can estimate a bandwidth value h from the data than the three rules-of-thumb considered. However, I also considered more complicated and data-based methods such as the method of Sheather and Jones (1991) (as implemented in R by bw.SJ()) and unbiased and biased cross-validation (as implemented by bw.ucv() and bw.bcv(), respectively), and found no substantial improvement over the simple rules-of-thumb considered here. Thus, considering the increased computation time that these methods entail, I have left them out of the analysis done here. As the values of RMSE for $h = 0.1, 0.2, .., 0.9$ serve to show, the fast and simple rule-of-thumb methods are generally already close (enough) to the 'optimal' bandwidth.

5.3 Comparative study

Table 3 and Table A1 (Appendix A.2) show for all investigated scenarios the RMSE of the slope and intercept coefficients, respectively. For $p = 5$ and $p = 10$, there are multiple slope coefficients; in that case, the reported RMSE on the slope is defined as:

$$\text{RMSE}\left(\hat{\boldsymbol{\beta}}\right) = \sqrt{\frac{1}{p-1}\sum_{j=2}^{p}\text{MSE}\left(\hat{\boldsymbol{\beta}}_j\right)}.$$

Several conclusion can be drawn from Table 3. First and foremost, I argue that RKDRE shows the best overall performance of all estimators considered. It improves upon KDRE for almost all investigated scenarios except the normal distribution. The performance for the variance-contaminated distribution (B) and the log-normal distribution (H) is especially remarkable. Under (H), the RMSE of the second most efficient estimators (KDRE and LGMM) is approximately 40% larger even for $n = 1000$, which means that the RKDRE is almost twice as efficient in the mean square error sense. Under (B) and (C), the RKDRE is also most efficient, but here the efficiency is gained mostly in the smaller samples ($n = 50$ and $n = 100$). Note especially the superior performance (R)KDRE in small samples of $t(2)$-distribution. By experimentation not reported here, I found that the comparative advantage of (R)KDRE over the other estimators is even higher when the degrees of freedom approach one. Furthermore, RKDRE performs best or close to best for distributions (D)-(G) as well. This is not to say of any other considered estimator. YDG performs well for (D), (E), and (G), but fails quite dramatically for distributions with fat tails such as (B), (C), and (H). Efficiency of the SBS estimator is in general low with respect to alternatives, but performance is especially weak under (C) and (D). LGMM is a reasonable estimator overall, but we see that efficiency is lost under distributions (E) and (F). This efficiency loss persists even for $n = 1000$. Also, it performs weaker than RKDRE under (B), (D), and (H) and its RMSE is up to two times larger than that of (R)KDRE in small samples of (C). Lastly, the LGMMS is by construction inef-

ficient when the error distribution is skewed (F)-(H). More surprisingly, however, its slope estimate is usually also no improvement over LGMM under symmetric distributions. Table A2 (Appendix A.2) shows the bias of the slope coefficients; we see that in general all estimators are virtually unbiased for $n \geq 100$. Only for the rather extreme $t(2)$-distribution, some bias still exists for $n = 500$.

Table A1 (Appendix A.2) shows that LGMMS does improve upon LGMM (and, in fact, upon all other estimators) in terms of the intercept coefficient. The RMSE of the intercept is up to three times as low as for the other estimators. Usually, the slope parameters of a regression model are of most interest to the researcher. However, if the intercept is of special interest the LGMMS estimator might be preferable. As may be expected, the efficiency of the LGMMS on the intercept vanishes when the distribution is not symmetric. The intercept estimators of the RKDRE, LGMM, YDG, and OLS show almost identical efficiency. Under the uniform distribution (E), KDRE and SBS are slightly more efficient on the intercept, but these two methods are, in turn, less efficient under the skewed distributions (F), (G), and (H). From Table A3 (Appendix A.2), we learn that the intercept bias of the different estimators is usually of similar magnitude in the symmetric cases. Under the asymmetric distributions, the bias of the intercept is much larger for KDRE, SBS and LGMMS than for the other estimators.

In conclusion, I find that the RKDRE estimator performs best, all things considered. It shows superior performance for almost all scenarios (other than the normal distribution). Only for a few scenarios, such as for $n = 50$ and $p = 5$ and/or $p = 10$ under (E) and (F), it shows some efficiency loss with respect to other estimators. However, note that for larger n, RKDRE is actually most efficient under (E) and (F). Also, the slight loss of efficiency of RKDRE under the normal distribution disappears when n increases.

Table 3: Comparison of root mean square error of the slope coefficient

p	2				5				10			
n	50	100	500	1000	50	100	500	1000	50	100	500	1000
(A) Normal distribution												
RKDRE	0.156	0.106	0.045	0.032	0.172	0.111	0.046	0.033	0.194	0.119	0.047	0.033
KDRE	0.149	0.103	0.044	0.033	0.158	0.106	0.046	0.033	0.165	0.109	0.046	0.032
YDG	0.162	0.113	0.047	0.032	0.178	0.115	0.049	0.034	0.198	0.125	0.049	0.034
SBS	0.166	0.111	0.046	0.033	0.178	0.114	0.047	0.033	0.181	0.117	0.047	0.033
LGMM	0.153	0.103	0.044	0.031	0.161	0.108	0.046	0.032	0.168	0.109	0.046	0.032
LGMMS	0.157	0.106	0.044	0.032	0.163	0.108	0.047	0.032	0.172	0.111	0.045	0.032
OLS	0.145	0.100	0.043	0.031	0.153	0.102	0.045	0.032	0.161	0.105	0.045	0.032
(B) Variance-contaminated normal distribution												
RKDRE	0.057	0.041	0.017	0.012	0.061	0.041	0.017	0.012	0.069	0.043	0.017	0.012
KDRE	0.063	0.044	0.017	0.012	0.073	0.045	0.018	0.012	0.098	0.051	0.018	0.012
YDG	0.213	0.099	0.019	0.014	0.185	0.095	0.021	0.014	0.180	0.089	0.021	0.014
SBS	0.067	0.044	0.018	0.013	0.074	0.046	0.018	0.013	0.095	0.052	0.019	0.013
LGMM	0.058	0.041	0.017	0.013	0.070	0.043	0.018	0.012	0.093	0.049	0.018	0.012
LGMMS	0.063	0.042	0.017	0.013	0.074	0.044	0.018	0.012	0.096	0.050	0.018	0.012
OLS	0.141	0.104	0.044	0.033	0.147	0.104	0.046	0.033	0.162	0.109	0.046	0.032

Continued on next page

Table 3 – continued from previous page

p	2				5				10			
n	50	100	500	1000	50	100	500	1000	50	100	500	1000
(C) t-distribution with two degrees of freedom												
RKDRE	0.217	0.142	0.064	0.041	0.234	0.154	0.064	0.043	0.257	0.161	0.064	0.044
KDRE	0.225	0.148	0.065	0.042	0.258	0.163	0.066	0.043	0.320	0.186	0.066	0.045
YDG	1.779	1.331	0.149	0.087	1.544	2.665	0.301	0.096	1.098	2.022	0.336	0.099
SBS	0.244	0.151	0.064	0.042	0.703	0.181	0.066	0.045	1.108	1.278	0.070	0.100
LGMM	0.277	0.152	0.060	0.040	0.399	0.172	0.063	0.041	0.413	0.383	0.069	0.044
LGMMS	0.325	0.159	0.060	0.040	0.425	0.176	0.065	0.041	0.440	0.368	0.084	0.045
OLS	0.507	0.330	0.164	0.107	0.515	0.350	0.159	0.114	0.516	0.512	0.180	0.117
(D) Bi-modal mixture of normal distributions												
RKDRE	0.052	0.033	0.015	0.010	0.053	0.034	0.014	0.010	0.076	0.035	0.015	0.010
KDRE	0.061	0.034	0.015	0.010	0.070	0.037	0.014	0.010	0.108	0.042	0.015	0.010
YDG	0.053	0.034	0.015	0.010	0.052	0.035	0.014	0.010	0.066	0.035	0.015	0.010
SBS	0.187	0.111	0.031	0.019	0.146	0.096	0.031	0.018	0.133	0.079	0.029	0.018
LGMM	0.063	0.037	0.015	0.010	0.075	0.039	0.015	0.011	0.111	0.048	0.015	0.011
LGMMS	0.070	0.037	0.015	0.010	0.082	0.041	0.015	0.011	0.119	0.050	0.015	0.011
OLS	0.155	0.096	0.044	0.031	0.151	0.103	0.044	0.032	0.161	0.104	0.045	0.032
(E) Uniform distribution												
RKDRE	0.116	0.069	0.025	0.015	0.132	0.076	0.025	0.016	0.170	0.084	0.025	0.016
KDRE	0.122	0.073	0.025	0.016	0.134	0.081	0.026	0.017	0.154	0.088	0.027	0.017
SBS	0.141	0.084	0.041	0.029	0.148	0.086	0.036	0.027	0.167	0.093	0.033	0.024
YDG	0.109	0.068	0.025	0.016	0.127	0.074	0.025	0.016	0.161	0.083	0.026	0.016
LGMM	0.125	0.082	0.035	0.025	0.134	0.085	0.034	0.025	0.152	0.089	0.035	0.025
LGMMS	0.113	0.065	0.027	0.018	0.127	0.075	0.026	0.018	0.151	0.083	0.028	0.019
OLS	0.150	0.102	0.045	0.032	0.151	0.104	0.045	0.033	0.162	0.105	0.045	0.032
(F) Gamma(2,2)												
RKDRE	0.104	0.068	0.026	0.017	0.124	0.072	0.027	0.018	0.151	0.083	0.027	0.017
KDRE	0.108	0.071	0.029	0.019	0.120	0.076	0.029	0.019	0.139	0.082	0.029	0.019
YDG	0.122	0.078	0.030	0.020	0.136	0.084	0.030	0.020	0.160	0.090	0.031	0.020
SBS	0.125	0.079	0.031	0.020	0.132	0.081	0.030	0.020	0.149	0.086	0.030	0.020
LGMM	0.106	0.071	0.031	0.021	0.118	0.076	0.031	0.021	0.137	0.081	0.031	0.021
LGMMS	0.144	0.094	0.042	0.028	0.150	0.099	0.043	0.029	0.163	0.101	0.042	0.029
OLS	0.142	0.098	0.046	0.030	0.149	0.103	0.047	0.032	0.161	0.104	0.045	0.032
(G) Skewed mixture of normal distributions												
RKDRE	0.079	0.052	0.022	0.015	0.087	0.054	0.022	0.015	0.115	0.058	0.022	0.015
KDRE	0.085	0.056	0.022	0.015	0.099	0.059	0.022	0.015	0.122	0.064	0.023	0.015
YDG	0.079	0.053	0.022	0.015	0.087	0.053	0.022	0.015	0.109	0.056	0.022	0.015
SBS	0.108	0.069	0.026	0.015	0.114	0.068	0.025	0.017	0.132	0.071	0.025	0.016
LGMM	0.077	0.052	0.021	0.015	0.120	0.057	0.022	0.015	0.125	0.063	0.023	0.015
LGMMS	0.125	0.088	0.035	0.025	0.147	0.094	0.036	0.026	0.178	0.100	0.039	0.026
OLS	0.142	0.104	0.050	0.034	0.149	0.102	0.049	0.036	0.174	0.105	0.050	0.035
(H) Log-normal distribution												
RKDRE	0.047	0.026	0.011	0.007	0.052	0.030	0.010	0.007	0.069	0.035	0.011	0.007
KDRE	0.066	0.037	0.015	0.010	0.074	0.043	0.015	0.010	0.095	0.052	0.016	0.010
YDG	0.108	0.172	0.022	0.013	0.163	0.144	0.023	0.013	0.168	0.130	0.023	0.013
SBS	0.098	0.060	0.029	0.017	0.085	0.052	0.022	0.015	0.097	0.053	0.019	0.014
LGMM	0.054	0.032	0.014	0.010	0.066	0.039	0.014	0.010	0.090	0.048	0.015	0.010
LGMMS	0.082	0.049	0.023	0.017	0.093	0.059	0.024	0.017	0.109	0.066	0.025	0.017
OLS	0.134	0.112	0.046	0.031	0.145	0.100	0.046	0.032	0.156	0.105	0.046	0.032

Results are based on 500 replications.

5.4 Computation time

Figure 1 shows average computation time (in seconds) over 50 simulations for the different estimators under varying values for n and p. I have left LGMMS out as its computational procedure (and hence computation time) is almost identical to LGMM. The left figure shows the computation time when $p = 2$ and n is increased. For the right figure $n = 500$, and the number of variables p is varied as specified. The computation time for YDG is obtained using the EM algorithm if $p = 2$ and using the optimization routine optim() if $p > 2$ (since this minimizes computation time). First, we see that for one explanatory variable, computation time of RKDRE, KDRE and YDG are in the same order of magnitude. For $n = 5000$, YDG is approximately two times slower than RKDRE and KDRE. Furthermore, we observe that the computation times of RKDRE and KDRE converge if n increases. To explain why, denote an iteration of the EM algorithm in RKDRE for a certain estimated density an 'EM-iteration' (i.e. an iteration in the maximization defined in Step 3 in Section 1), and an iteration from one estimated density to the next estimated density a β-iteration (i.e. an iteration of Step 2 in Section 1). Then, if n increases, the number of β-iterations generally decreases. For $n = 5000$, the RKDRE algorithm is usually converged after one β-iteration. Also, for later β-iterations, the number of EM-iterations are generally lower. This further decreases the difference in computation time between RKDRE and KDRE. For $n = 5000$, the RKDRE algorithm is still feasible (as it costs less than two minutes). However, considering that Table 3 shows that the difference in performance of the algorithms is small for large n, LGMM, being by far the fastest algorithm, might be preferred for large datasets (e.g. $n \gtrapprox 5000$).

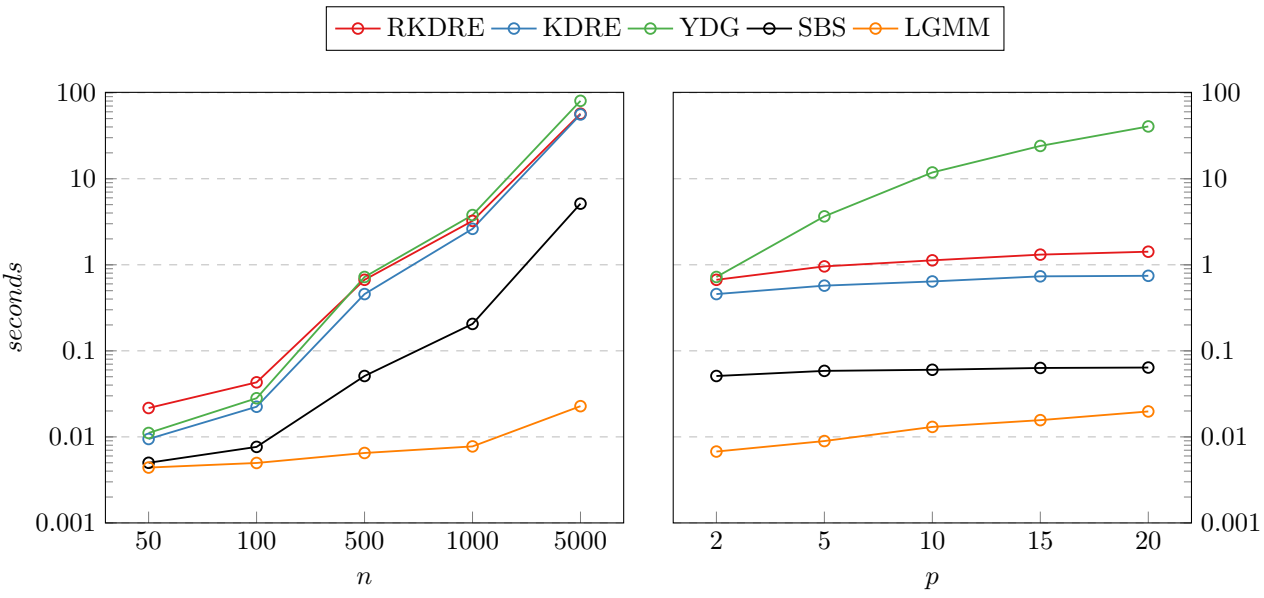


Figure 1: Average computation time in seconds for different values of n and p

In the right panel of Figure 1, we clearly observe that the computation time of YDG increases severely for an increasing number of explanatory variables, where the other estimators are less sensitive. Clearly, the non-linear optimization routine used in YDG becomes slower, the higher the dimensionality.

6 Standard errors

In this section, I investigate the estimation of standard errors of the RKDRE algorithm. In Theorem 3.6, we found the asymptotic distribution of the RKDRE algorithm. This suggests to estimate the standard errors accordingly. For that purpose, consider Theorem 6.1 and 6.2 below.

Theorem 6.1. *Under the assumptions of Theorem 3.6, the standard error of* $\hat{\boldsymbol{\beta}}^{(m)}$ *can be consistently estimated by*

$$\left[\sum_{i=1}^{n}\left(\hat{\Psi}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right)\right)^2 G_b\left(\hat{f}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right)\right)\boldsymbol{x_i}\boldsymbol{x_i'}\right]^{-1} \xrightarrow{p} \frac{1}{n}I_{\beta\beta}^{-1}, \tag{61}$$

where

$$\hat{\Psi}_n^{(m)}\left(y_i|\boldsymbol{\beta}\right) = \frac{\hat{f}_n^{\prime(m)}\left(y_i|\boldsymbol{\beta}\right)}{\hat{f}_n^{(m)}\left(y_i|\boldsymbol{\beta}\right)}, \tag{62}$$

and where G_b *is defined as in* (16).

Proof: By (46), $\hat{\boldsymbol{\beta}}^{(m)} \xrightarrow{d} \mathcal{N}\left(\boldsymbol{\beta}_0, \frac{1}{n}I_{\beta\beta}^{-1}\right)$. Yao and Zhao (2013, p. 4508) show in the proof of Theorem 3.6 that

$$C \triangleq -\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\Psi}_n^{(1)}\left(y_i|\boldsymbol{\beta}_0\right)\right)^2 G_b\left(\hat{f}_n^{(1)}\left(y_i|\boldsymbol{\beta}_0\right)\right)\boldsymbol{x_i}\boldsymbol{x_i'} \xrightarrow{p} -I_{\beta\beta}. \tag{63}$$

Since matrix inversion is a continuous transformation, by the continuous mapping theorem, this implies that $-C^{-1} \xrightarrow{p} I_{\beta\beta}^{-1}$. By definition of consistency of the initial estimator (C3) in (Yao and Zhao, 2013), $\hat{\boldsymbol{\beta}}^{(0)} \xrightarrow{p} \boldsymbol{\beta}_0$. Thus, by the fact that the kernel function K is four times continuously differentiable and continuity of $G_b(x)$, we may again invoke the continuous mapping theorem to conclude that

$$\left[\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\Psi}_n^{(1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(0)}\right)\right)^2 G_b\left(\hat{f}_n^{(1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(0)}\right)\right)\boldsymbol{x_i}\boldsymbol{x_i'}\right]^{-1} \xrightarrow{p} -C^{-1}. \tag{64}$$

To come to the final result, I first use that convergence in probability to a sequence converging in distribution implies convergence to the same distribution. That is, if $Y_n \xrightarrow{p} X_n$ and $X_n \xrightarrow{d} X$, then $Y_n \xrightarrow{d} X$. Since convergence in probability implies convergence in distribution, we have that $-C^{-1} \xrightarrow{d} I_{\beta\beta}^{-1}$. Thus, by (64), we obtain

$$\left[\frac{1}{n}\sum_{i=1}^{n}\left(\hat{\Psi}_n^{(1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(0)}\right)\right)^2 G_b\left(\hat{f}_n^{(1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(0)}\right)\right)\boldsymbol{x_i}\boldsymbol{x_i'}\right]^{-1} \xrightarrow{d} I_{\beta\beta}^{-1}. \tag{65}$$

Hence,

$$\left[\sum_{i=1}^{n}\left(\hat{\Psi}_n^{(1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(0)}\right)\right)^2 G_b\left(\hat{f}_n^{(1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(0)}\right)\right)\boldsymbol{x_i}\boldsymbol{x_i'}\right]^{-1} \xrightarrow{d} \frac{1}{n}I_{\beta\beta}^{-1}. \tag{66}$$

Lastly, note that $\frac{1}{n}I_{\beta\beta}^{-1}$ is a constant (as it is an expectation), and that convergence in distribution to a constant implies convergence in probability to that constant. This proves (61) for the first iteration. The result holds similarly for further iterations by the same argument as in Theorem 3.6. □

Theorem 6.2. *Under the assumptions of Theorem 3.6, the standard error of* $\hat{\boldsymbol{\beta}}^{(m)}$ *can also be consistently estimated by*

$$\begin{aligned}&\left[\sum_{i=1}^{n}\left(\hat{\Psi}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right)\right)^2 G_b\left(\hat{f}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right)\right)\boldsymbol{x_i}\boldsymbol{x_i'} - \right.\\ &\left.\sum_{i=1}^{n}\hat{\Phi}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right) G_b\left(\hat{f}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right)\right)\boldsymbol{x_i}\boldsymbol{x_i'}\right]^{-1} \xrightarrow{p} \frac{1}{n}I_{\beta\beta}^{-1}\end{aligned} \tag{67}$$

where

$$\hat{\Phi}_n^{(m)}\left(y_i|\boldsymbol{\beta}\right) = \frac{\hat{f}_n''^{(m)}\left(y_i|\boldsymbol{\beta}\right)}{\hat{f}_n^{(m)}\left(y_i|\boldsymbol{\beta}\right)}. \tag{68}$$

Proof: By the result in Theorem 6.1, we know that

$$\sum_{i=1}^{n}\left(\hat{\Psi}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right)\right)^2 G_b\left(\hat{f}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right)\right)\boldsymbol{x_i}\boldsymbol{x_i'} \xrightarrow{p} nI_{\beta\beta}. \tag{69}$$

Also, (Yao and Zhao, 2013, p. 4509) show that

$$\sum_{i=1}^{n}\hat{\Phi}_n^{(1)}\left(y_i|\boldsymbol{\beta}_0\right) G_b\left(\hat{f}_n^{(1)}\left(y_i|\boldsymbol{\beta}_0\right)\right)\boldsymbol{x_i}\boldsymbol{x_i'} \xrightarrow{p} 0. \tag{70}$$

By the same reasoning as in Theorem 6.1, we obtain that

$$\sum_{i=1}^{n}\hat{\Phi}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right) G_b\left(\hat{f}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right)\right)\boldsymbol{x_i}\boldsymbol{x_i'} \xrightarrow{p} 0. \tag{71}$$

Combining (69) and (71) leads to

$$\begin{aligned}&\sum_{i=1}^{n}\left(\hat{\Psi}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right)\right)^2 G_b\left(\hat{f}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right)\right)\boldsymbol{x_i}\boldsymbol{x_i'} - \\ &\sum_{i=1}^{n}\hat{\Phi}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right) G_b\left(\hat{f}_n^{(m+1)}\left(y_i|\hat{\boldsymbol{\beta}}^{(m)}\right)\right)\boldsymbol{x_i}\boldsymbol{x_i'} \xrightarrow{p} nI_{\beta\beta}.\end{aligned} \tag{72}$$

The result in (67) follows from the continuous mapping theorem. □

Essentially, what Theorem 6.2 shows is that the term that is added with respect to Theorem 6.1 is $o_p(1)$ and that this addition does not affect the consistency proven in Theorem 6.1.[10] To see why adding this term may be sensible, recall the information matrix $I_{\beta\beta}$ as in (7) that we aim to estimate:

$$\begin{aligned}I_{\beta\beta} &= -E\left[d_{\beta\beta}\left(\boldsymbol{\beta}_0\right)\right] = E\left[\frac{f'^2(y_i|\boldsymbol{\beta}) - f(y_i|\boldsymbol{\beta}_0)f''(y_i|\boldsymbol{\beta}_0)}{f^2(y_i|\boldsymbol{\beta}_0)}\boldsymbol{x_i}\boldsymbol{x_i'}\right]\\ &= E\left[\left(\Psi\left(y_i|\boldsymbol{\beta}_0\right)\right)^2\boldsymbol{x_i}\boldsymbol{x_i'}\right] - E\left[\Phi\left(y_i|\boldsymbol{\beta}_0\right)\boldsymbol{x_i}\boldsymbol{x_i'}\right].\end{aligned}$$

Even though the (approximation of) the second term is shown to converge to zero, in finite samples it may differ from zero. Hence, I separately investigate the accuracy of the estimated standard errors both when this term is excluded (Theorem 6.1) and when it is included (6.2).

[10] By definition, a random variable $X_n = o_p(1)$ if $X_n \xrightarrow{p} 0$.

Table 4: Ratio of root mean square standard error and actual root mean square error

	Intercept				Slope			
n	50	100	500	1000	50	100	500	1000
(A) Normal distribution								
SE1	1.250	1.187	1.065	1.085	1.171	1.171	1.082	1.014
SE2	0.911	0.935	0.942	0.990	0.811	0.894	0.952	0.922
Bootstrap	0.994	1.096	-	-	1.062	1.096	-	-
(B) Variance-contaminated normal distribution								
SE1	0.486	0.416	0.369	0.374	1.273	1.157	0.974	1.002
SE2	0.320	0.289	0.279	0.290	0.821	0.791	0.738	0.778
Bootstrap	1.018	0.969	-	-	1.133	1.041	-	-
(C) t-distribution with two degrees of freedom								
SE1	0.502	0.534	0.301	0.356	1.032	1.002	0.900	0.909
SE2	0.340	0.384	0.228	0.272	0.680	0.714	0.685	0.694
Bootstrap	0.969	1.102	-	-	1.049	1.062	-	-
(D) Bi-modal mixture of normal distributions								
SE1	0.524	0.436	0.381	0.354	1.598	1.344	1.188	1.141
SE2	0.361	0.332	0.330	0.318	1.069	1.004	1.027	1.020
Bootstrap	1.011	0.965	-	-	1.453	0.990	-	-
(E) Uniform distribution								
SE1	1.239	1.203	0.957	0.847	1.733	1.918	1.913	1.664
SE2	0.760	0.743	0.602	0.535	1.003	1.134	1.193	1.050
Bootstrap	0.983	0.996	-	-	1.264	1.202	-	-
(F) Gamma(2,2)								
SE1	1.005	0.935	0.827	0.725	1.387	1.424	1.341	1.304
SE2	0.640	0.632	0.608	0.547	0.835	0.931	0.981	0.975
Bootstrap	0.994	0.999	-	-	1.257	1.176	-	-
(G) Skewed mixture of normal distributions								
SE1	0.575	0.488	0.332	0.288	1.226	1.131	1.064	1.016
SE2	0.384	0.355	0.277	0.249	0.784	0.810	0.883	0.875
Bootstrap	0.963	0.902	-	-	1.292	1.095	-	-
(H) Log-normal distribution								
SE1	0.437	0.372	0.284	0.270	1.510	1.436	1.300	1.242
SE2	0.246	0.219	0.182	0.178	0.818	0.832	0.825	0.812
Bootstrap	0.997	0.988	-	-	1.376	1.205	-	-

The number of replications is 500. $p = 2$, and $\boldsymbol{\beta} = [1, -1]'$. The bandwidth suggested by Sheather and Jones (1991) as implemented in R by bw.SJ() is used. The number of bootstraps per replication is 100.

In Table 4, I show the ratio of the estimated root mean square standard error and the actual root mean square error.[11] SE1 and SE2 denote the standard errors computed based on Theorem 6.1 and 6.2, respectively. Even though the trimming parameter was necessary for technical reasons, in practice it entails no improvement and, hence, the reported standard errors are of the untrimmed version. Furthermore, I find that the accuracy of the standard errors is relatively sensitive to the bandwidth parameter. By experimentation, I find that the data-based bandwidth suggested by Sheather and Jones (1991) works best such that these are used in Table 4.[12] This is somewhat surprising as this bandwidth did not substantially improve upon the result when it was used in the estimation of $\boldsymbol{\beta}$. An explanation might lie in the fact that for the estimation of the standard error the kernel density (derivative) directly influences the result; in the estimation of $\boldsymbol{\beta}$ the kernel

[11] This procedure is adopted from Newey (1988).

[12] A technical explanation of the method of Sheather and Jones (1991) goes beyond the scope of this research, but the essence of the method is the inclusion of a non-stochastic term that reduces the bias without increasing variance.

density is used only to be maximized with respect to the parameter vector.

Indeed, we see that SE1 and SE2 can differ substantially. Under distributions that differ strongly from normal (e.g., (D), (E), (F), and (H)), SE2 performs in general better than SE1 with respect to the slope coefficient. The standard errors of the intercept term are underestimated for most considered distributions. For the skewed distributions, this does not come as a surprise; recall (from Section 2) that the intercept term is not in general adaptively estimable if f is not symmetric, and hence does not attain the Cramér-Rao lower bound. Under the symmetric distributions, based on theoretical results, one expects the RKDRE (and other estimators) to attain the CRLB for the intercept too. However, in finite samples, we see that this holds only (approximately) for LGMMS. In fact, using Table A1, we find for the scenarios for which the standard error of the intercept is underestimated, this underestimation is almost exactly accounted for by the difference in efficiency between LGMMS and the other estimators. Hence, the underestimation of the standard error is not so much a result of inaccurate estimation of the information matrix, but simply a result of the fact that RKDRE (and all other estimators except LGMMS) do not seem to attain the CRLB for the intercept, at least not in finite samples. Therefore, if the standard error of the intercept is of interest, it is recommended to obtain the standard error by other methods. In that respect, Table 4 shows that bootstrapping performs reasonable. Note that for computational reasons, the number of bootstrap replicates is limited to 100 and the bootstrap method is evaluated only for $n = 50$ and $n = 100$.

7 Application

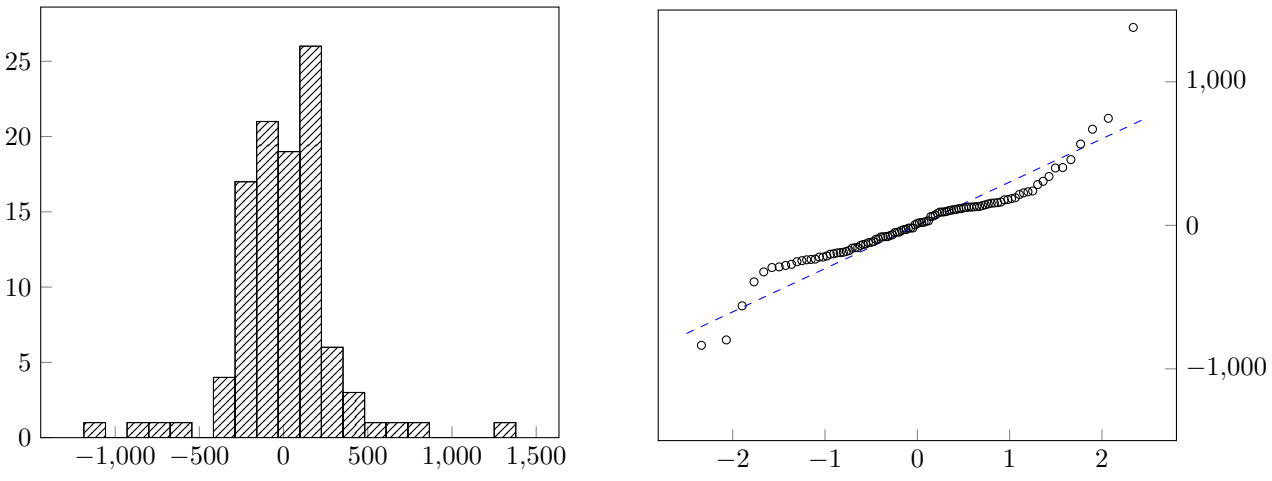
In this section, I apply RKDRE to the research described in (Andrabi et al., 2017), recently published in the American Economic Review. This research involves an experiment on the impact of providing information in the form of report cards on educational outcomes such as test scores, prices, and the enrollment rate. 56 out of $n = 112$ analyzed Pakistani villages were randomly assigned treatment. The report cards, given to both households and schools in treatment villages, included information on the performance of the child, the average score of different schools in the village, and the average village score in mathematics, English, and Urdu.[13] The main findings are: (1) private school fees decreased by 17 percent, (2) test scores increased by 0.11 standard deviations, and (3) primary enrollment increased with 4.5 percent. The models of interest are specified as followed:

$$F_{m2} = \alpha_{1d} + \beta_1 \cdot RC_m + \gamma_1 \cdot F_{m1} + \delta_1 \cdot X_{m1} + \varepsilon_m, \qquad \text{(Model 1)}$$

$$T_{m2} = \alpha_{2d} + \beta_2 \cdot RC_m + \gamma_2 \cdot T_{m1} + \delta_2 \cdot X_{m1} + \varepsilon_m, \qquad \text{(Model 2)}$$

$$E_{m2} = \alpha_{3d} + \beta_3 \cdot RC_m + \gamma_3 \cdot E_{m1} + \delta_3 \cdot X_{m1} + \varepsilon_m, \qquad \text{(Model 3)}$$

where F_{m2}, T_{m2}, and E_{m2} are average fees, test scores, and enrollment rate in the post-intervention year of village m, respectively. F_{m1}, T_{m1}, and E_{m1} denote the baseline measurement of the same variables. α_{id} are district fixed effects for model i; RC_m is the treatment dummy assignment to village m, which makes β_i the variable of interest, an estimate of the impact of the report card assignment. X_{m1} is a vector of village-level controls measured at baseline. All models in the paper are estimated using OLS. As Andrabi et al. (2017) note, under random assigment of treatment, the OLS coefficient is an unbiased estimate of the treatment effect. However, we have seen in Section 5.3, that adaptive methods may be more efficient if the error terms are not normally distributed. Figure 2 shows residual diagnostics of Model 1 after applying OLS; indeed, we see that the sample has fatter tails than we would expect based on normality.



(a) Histogram (b) Normal quantile plot

Figure 2: Residual diagnostics for Model 1

[13] In Appendix A.3, an example of an anonymized report can be found.

Furthermore, recall that Mammen et al. (1996) showed that the residuals of OLS (as the maximum likelihood estimator of the normal distribution) will usually be drawn towards the normal distribution, such that the residuals may even show closer resemblance to the normal distribution than the actual error distribution. For Model 2 and Model 3, the same diagnostics of the OLS residuals are shown in Figure A3 and A4, respectively. For Model 2, we find evidence of slight skewness, and the residuals of Model 3 seems to have fat tails too. However, the deviance from normality seems to be most pronounced for Model 1.

Table 5: Estimated effect of report cards for respective estimators

	OLS	RKDRE	KDRE	YDG	SBS	LGMM	LGMMS
$\hat{\beta}_1$	-187.0	-4.418	-85.55	-154.4	-47.74	-172.3	-182.9
	(65.91)	(38.76)	(63.42)	(59.90)	(73.34)	(64.54)	(67.65)
$\hat{\beta}_2$	0.114	0.084	0.092	0.094	0.088	0.088	0.099
	(0.046)	(0.050)	(0.048)	(0.060)	(0.054)	(0.045)	(0.050)
$\hat{\beta}_3$	0.032	0.028	0.030	0.018	0.030	0.029	0.027
	(0.014)	(0.014)	(0.012)	(0.019)	(0.013)	(0.012)	(0.012)

OLS coefficients are as shown in Table 3 (1) Panel C, Table 3 (4) Panel C, and Table 4 (1) Panel C in (Andrabi et al., 2017). Standard errors (in parentheses) for OLS are as reported in the paper; standard errors of other methods are obtained by bootstrap using 500 replications.

Table 5 shows the treatment effect in the three models as estimated by the investigated estimators. The OLS coefficients are as reported in (Andrabi et al., 2017). Clearly, we see that the adaptive estimators pull the estimated treatment effect towards zero for all models. The results for Model 1 are especially striking. The RKDRE estimate is more than forty times lower than OLS. Also, for Model 1, the estimates of the adaptive methods differ substantially. In that respect, it is interesting too investigate the prediction performance of the respective methods as reported in Table 6. We see that RKDRE shows the lowest median absolute prediction error (MAPE) for Model 1. Furthermore, observe that prediction performance is generally better for the estimators with a lower estimate of β_1, such as SBS and KDRE. Hence, we observe quite strong evidence to believe that the effect of the report cards on school fees, if it exists at all, is much lower than reported.

Table 6: Median absolute prediction error relative to OLS

	RKDRE	KDRE	YDG	SBS	LGMM	LGMMS
Model 1	0.720	0.858	0.906	0.769	0.992	1.001
Model 2	1.002	1.016	0.998	1.005	0.997	1.020
Model 3	0.972	0.975	1.029	0.963	0.965	0.971

The values are the ratio of the MAPE of the respective estimator and the MAPE of OLS. Hence, the lower the value, the better the prediction performance relative to OLS. The training set is a random sample of the data of size $\lceil 0.8n \rceil$. The number of replications is 500.

For Model 2 and 3, there is less difference between the estimates of the adaptive methods. Also, the estimate is adjusted less strongly with respect to OLS. Table 7 shows 95% confidence intervals for β_i as estimated by RKDRE using different standard errors; SE1 and SE2 are as defined in Section 6. For the bootstrap method, two confidence intervals are shown. The first is based on

(approximate) normality; that is, the confidence interval is the estimated coefficient $\hat{\beta}_i \pm 1.96 \times \text{SE}$ where SE is the standard deviation of the bootstrap estimates. The percentile method is obtained by taking the 0.025^{th} and 0.975^{th} quantile of the bootstrapped estimates as lower and upper bound, respectively.

Table 7: 95% confidence intervals of the effect of report cards

			Bootstrap	
	SE1	SE2	Normal	Percentile
Model 1	[−120.2, 111.3]	[76.87, 68.03]	[−80.38, 71.55]	[−88.11, 59.25]
Model 2	[−0.017, 0.184]	[0.004, 0.164]	[−0.015, 0.183]	[−0.011, 0.187]
Model 3	[−0.003, 0.060]	[0.006, 0.050]	[0.001, 0.055]	[0.002, 0.055]

For SE1, SE2, and the normal bootstrap, confidence intervals are computed as $\hat{\beta}_i \pm 1.96 \times \text{SE}$. For the percentile bootstrap confidence interval, the lower and upper bound are equal to the 0.025^{th} and 0.975^{th} quantile of the bootstrapped estimates of β, respectively. I use 500 bootstrap replicates.

The estimated treatment effect for Model 1 is clearly not significantly different from zero. For Model 2 and Model 3, the significance depends on the standard error of choice. From Table 4, we know that, for $n \approx 100$, SE2 usually slightly underestimates the standard error, whereas SE1 slightly overestimates the uncertainty, in particular for distributions with fat-tails such as (B), (C), and (H). The bootstrap standard errors are generally in between SE1 and SE2, such that these may be considered most reliable here.

The analysis in this section demonstrates the practical relevance of adaptive estimation in general and that of the proposed RKDRE algorithm in particular. None of the other adaptive estimators adjusted the treatment effect on school fees as far towards zero as RKDRE, while prediction performance provides evidence for the belief that RKDRE is in fact the preferred method for this model. Using RKDRE, we find that out of the three main findings described in (Andrabi et al., 2017), one can be considered not significant on any reasonable significance level. Also, Table 5 shows that the adaptive estimators find that the effect of report cards on test scores, which is the second out of the three result, is not significant on the 5% significance level. Lastly, the effect on the enrollment rate seems, even though marginally significant for RKDRE, also questionable.

8 Discussion

The proposed RKDRE algorithm is shown to have the adaptive property. That is, it is asymptotically normal and its asymptotic variance is equal to the Cramér-Rao lower bound. Also, I establish almost sure convergence. More importantly from a practical point of view, I find that the performance of RKDRE in simulation studies is second to none of the other considered adaptive estimators. For several distributions, it is up to twice as efficient in the mean square error sense than the second best estimator. Furthermore, for any other distribution, it is either the most efficient or very close to the most efficient estimator. All of the other estimators show a loss of efficiency for certain specific distributions. Furthermore, the use of RKDRE is made convenient by an EM algorithm that allows for fast computation and the availability of consistent standard errors.

Several avenues of further research may still be considered in the context of linear regression. First of all, other estimators can be adjusted to multi-step procedures as well. The intuition behind the importance of multiple steps is arguably less appealing for LGMM(S) and SBS than it is for (R)KDRE, as the maximum likelihood estimator of an estimated density that is likely to be adjusted towards the true error density in the process of repeated kernel density estimation. Regardless, one may also expect improvement of performance when these other methods use multiple steps instead of only two. Also, for the purpose of this research, the initial estimate was set equal to the OLS estimate. This choice was primarily based on computational convenience. Perhaps, efficiency may be further enhanced by a more prudent choice of the initial estimator. At last, the choice of bandwidth might be a source of further efficiency gain for the kernel estimators (i.e. (R)KDRE, SBS, and YDG). I have considered a vast range of bandwidth selection methods and found that the performance of the more complicated (and computationally intensive) methods is usually not an advantage over simple and fast rules-of-thumb. However, the bandwidth selection methods are constructed such as to minimize a certain loss function with respect to the true density; they are not tailored for the use in adaptive regression estimators. It is not unthinkable that, in this context, more appropriate bandwidth rules can be developed.

Lastly, the concept of RKDRE can be straightforwardly extended to nonlinear regression. It is expected that, as long as the nonlinear regression function is of known form, the asymptotic properties of adaptiveness continue to hold. However, the EM algorithm (at least in its present form) can only be applied to linear regression. This means that research in the direction of non-linear regression will be faced with computational issues, too.

Despite the fact that I focus on the advantages of RKDRE over existing adaptive estimators, I would like to conclude with the remark that in general all such estimators are considerably more efficient than OLS under distributions other than the normal distribution. By applying these methods to the research in (Andrabi et al., 2017), I show that their practical implications can be large. Therefore, it is the more remarkable that adaptive estimation is seldom applied in practice. Indeed, one may verify that articles on adaptive estimation, e.g. (Bickel, 1982) and (Newey, 1988),

are almost exclusively cited in research of methodological nature. I suppose that the difficulty of implementation may impede the use of adaptive methods in applied research. In that respect, I find it odd that (as far as I am aware) no software package exists that supports standard methods such as described in these papers. Also, as of yet, no practitioner's guide to adaptive methods is available. With such tools, the current existing gap between theory and practice may be bridged.

References

Andrabi, T., Das, J., and Khwaja, A. I. (2017). Report cards: The impact of providing school and child test scores on educational markets. *American Economic Review*, 107(6):1535–63.

Andrews, D. W. (1994). Asymptotics for semiparametric econometric models via stochastic equicontinuity. *Econometrica: Journal of the Econometric Society*, pages 43–72.

Begun, J. M., Hall, W., Huang, W.-M., and Wellner, J. A. (1983). Information and asymptotic efficiency in parametric-nonparametric models. *The Annals of Statistics*, pages 432–452.

Bickel, P. J. (1982). On adaptive estimation. *The Annals of Statistics*, pages 647–671.

Chai, G., Li, Z., and Tian, H. (1991). Consistent nonparametric estimation of error distributions in linear model. *Acta Mathematicae Applicatae Sinica (English Series)*, 7(3):245–256.

Hsieh, D. A. and Manski, C. F. (1987). Monte carlo evidence on adaptive maximum likelihood estimation of a regression. *The Annals of Statistics*, pages 541–551.

Kasy, M. (2015). Uniformity and the delta method. *Unpublished manuscript*.

Linton, O. and Xiao, Z. (2007). A nonparametric regression estimator that adapts to error distribution of unknown form. *Econometric Theory*, 23(3):371–413.

Mammen, E. et al. (1996). Empirical process of residuals for high-dimensional linear models. *The annals of statistics*, 24(1):307–335.

Manski, C. F. (1984). Adaptive estimation of non–linear regression models. *Econometric reviews*, 3(2):145–194.

Newey, W. K. (1988). Adaptive estimation of regression models via moment restrictions. *Journal of Econometrics*, 38(3):301–339.

Newey, W. K. and McFadden, D. (1994). Large sample estimation and hypothesis testing. *Handbook of econometrics*, 4:2111–2245.

Pagan, A. and Ullah, A. (1999). *Nonparametric econometrics*. Cambridge university press.

Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076.

Pawitan, Y. (2001). *In all likelihood: statistical modelling and inference using likelihood*. Oxford University Press.

Rosenblatt, M. et al. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837.

Schick, A. (1993). On efficient estimation in regression models. *The Annals of Statistics*, pages 1486–1521.

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 683–690.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*, volume 26. CRC press.

Stone, C. J. (1975). Adaptive maximum likelihood estimators of a location parameter. *The Annals of Statistics*, pages 267–284.

Wade, W. (1974). The bounded convergence theorem. *The American Mathematical Monthly*, 81(4):387–389.

Yao, W. and Zhao, Z. (2013). Kernel density-based linear regression estimate. *Communications in Statistics-Theory and Methods*, 42(24):4499–4512.

Yuan, A. and De Gooijer, J. G. (2007). Semiparametric regression with kernel error model. *Scandinavian Journal of Statistics*, 34(4):841–869.

Zhang, W. Y. (1990). On the congruent kernel estimate of error distributions in linear model (in chinese). *Journal of Sichuan University (Natural Science Edition)*, 27:132–144.

A Appendix

A.1 EM algorithm for YDG

Recall that YDG is the solution to the following maximization criterion:

$$\hat{\boldsymbol{\beta}}^*_{YDG} = \underset{\boldsymbol{\beta}^* \in \boldsymbol{\mathcal{B}}^*}{\arg\max} \sum_{i=1}^{n} \frac{1}{nh_n} \sum_{j \neq i}^{n} K\left(\frac{y_i - y_j - \left(\boldsymbol{x}^*_{\boldsymbol{i}} - \boldsymbol{x}^*_{\boldsymbol{j}}\right)' \boldsymbol{\beta}^*}{h_n} \right) \tag{73}$$

where $\boldsymbol{x_i} = \left[1, \boldsymbol{x}^*_{\boldsymbol{i}}\right]'$, and $\hat{\boldsymbol{\beta}}_{YDG} = \left[\hat{\alpha}_{YDG}, \hat{\boldsymbol{\beta}}^*_{YDG}\right]'$ and $\hat{\alpha}$ is the estimate of the intercept coefficient α. When K is the Gaussian kernel, this can be maximized by the following EM algorithm:

E-step:

$$p_{ij,(k+1)} = \frac{\exp\left\{ -\frac{1}{2h_n^2} \left(y_i - y_j - \left(\boldsymbol{x}^*_{\boldsymbol{i}} - \boldsymbol{x}^*_{\boldsymbol{j}}\right)' \boldsymbol{\beta}^* \right)^2 \right\}}{\sum_{j \neq i}^{n} \exp\left\{ -\frac{1}{2h_n^2} \left(y_i - y_j - \left(\boldsymbol{x}^*_{\boldsymbol{i}} - \boldsymbol{x}^*_{\boldsymbol{j}}\right)' \boldsymbol{\beta}^* \right)^2 \right\}}, \quad j \neq i \tag{74}$$

M-step:

$$\hat{\boldsymbol{\beta}}^*_{YDG,(k+1)} = \left[\sum_{i=1}^{n} \sum_{j \neq i}^{n} p_{ij,(k+1)} \left(\boldsymbol{x}^*_{\boldsymbol{i}} - \boldsymbol{x}^*_{\boldsymbol{j}}\right) \left(\boldsymbol{x}^*_{\boldsymbol{i}} - \boldsymbol{x}^*_{\boldsymbol{j}}\right)' \right]^{-1} \sum_{i=1}^{n} \sum_{j \neq i}^{n} \left(\boldsymbol{x}^*_{\boldsymbol{i}} - \boldsymbol{x}^*_{\boldsymbol{j}}\right) (y_i - y_j) \tag{75}$$

Theorem A.1. *The objective function* (73) *decreases after each iteration of* (74) *and* (75) *until a fixed point is reached.*

Proof: when we adjust (48) to the criterion in , under the Gaussian kernel, the M-step becomes

$$\hat{\boldsymbol{\beta}}_{YDG,(k+1)} = \underset{\boldsymbol{\beta}}{\arg\min} \sum_{i=1}^{n} \sum_{j \neq 1}^{n} p_{ij,(k+1)} \left(y_i - y_j - \left(\boldsymbol{x}^*_{\boldsymbol{i}} - \boldsymbol{x}^*_{\boldsymbol{j}}\right)' \boldsymbol{\beta}^* \right)^2$$

The first order condition yields

$$\sum_{i=1}^{n} \sum_{j \neq 1}^{n} p_{ij,(k+1)} \left(\boldsymbol{x}^*_{\boldsymbol{i}} - \boldsymbol{x}^*_{\boldsymbol{j}}\right) \left(\boldsymbol{x}^*_{\boldsymbol{i}} - \boldsymbol{x}^*_{\boldsymbol{j}}\right)' \hat{\boldsymbol{\beta}}^*_{YDG,(k+1)} = \sum_{i=1}^{n} \sum_{j \neq 1}^{n} p_{ij,(k+1)} \left(\boldsymbol{x}^*_{\boldsymbol{i}} - \boldsymbol{x}^*_{\boldsymbol{j}}\right)' (y_i - y_j),$$

and thus

$$\hat{\boldsymbol{\beta}}^*_{YDG,(k+1)} = \left[\sum_{i=1}^{n} \sum_{j \neq 1}^{n} p_{ij,(k+1)} \left(\boldsymbol{x}^*_{\boldsymbol{i}} - \boldsymbol{x}^*_{\boldsymbol{j}}\right) \left(\boldsymbol{x}^*_{\boldsymbol{i}} - \boldsymbol{x}^*_{\boldsymbol{j}}\right)' \right]^{-1} \sum_{i=1}^{n} \sum_{j \neq 1}^{n} p_{ij,(k+1)} \left(\boldsymbol{x}^*_{\boldsymbol{i}} - \boldsymbol{x}^*_{\boldsymbol{j}}\right)' (y_i - y_j)$$

Then, the fact that (74) and (75) are the E- and M-step, respectively, of an EM algorithm for (73) follows trivially from the proof of Theorem 2.2 in (Yao and Zhao, 2013, p.4511). □

A.2 Additional tables

Table A1: Comparison of root mean square error of the intercept

p	2				5				10			
n	50	100	500	1000	50	100	500	1000	50	100	500	1000
(A) Normal distribution												
RKDRE	0.145	0.105	0.043	0.032	0.149	0.107	0.045	0.031	0.155	0.103	0.043	0.033
KDRE	0.148	0.107	0.043	0.032	0.151	0.108	0.045	0.031	0.154	0.104	0.044	0.033
YDG	0.146	0.105	0.043	0.034	0.151	0.107	0.045	0.031	0.161	0.105	0.043	0.033
SBS	0.151	0.108	0.044	0.033	0.155	0.109	0.045	0.031	0.158	0.106	0.044	0.033
LGMM	0.146	0.105	0.043	0.031	0.148	0.107	0.045	0.031	0.153	0.103	0.043	0.033
LGMMS	0.156	0.110	0.043	0.032	0.160	0.113	0.045	0.031	0.162	0.110	0.044	0.033
OLS	0.146	0.105	0.043	0.031	0.148	0.106	0.045	0.031	0.152	0.102	0.043	0.033
(B) Variance-contaminated normal distribution												
RKDRE	0.142	0.101	0.044	0.032	0.137	0.101	0.044	0.031	0.139	0.099	0.044	0.031
KDRE	0.144	0.103	0.045	0.033	0.145	0.105	0.045	0.031	0.154	0.106	0.045	0.031
YDG	0.145	0.102	0.044	0.032	0.146	0.102	0.044	0.031	0.156	0.103	0.044	0.031
SBS	0.143	0.102	0.044	0.033	0.140	0.103	0.044	0.031	0.144	0.102	0.044	0.031
LGMM	0.142	0.101	0.044	0.032	0.137	0.102	0.044	0.031	0.141	0.100	0.044	0.031
LGMMS	0.059	0.042	0.017	0.012	0.066	0.042	0.018	0.012	0.083	0.049	0.018	0.013
OLS	0.144	0.102	0.044	0.032	0.139	0.102	0.044	0.031	0.151	0.104	0.046	0.033
(C) t-distribution with two degrees of freedom												
RKDRE	0.492	0.333	0.163	0.116	0.520	0.378	0.164	0.116	0.486	0.482	0.240	0.239
KDRE	0.508	0.343	0.168	0.118	0.545	0.392	0.171	0.120	0.520	0.501	0.243	0.241
YDG	0.513	0.340	0.122	0.094	0.526	0.577	0.138	0.085	0.576	0.487	0.127	0.090
SBS	0.765	0.338	0.164	0.116	1.772	0.482	0.164	0.116	1.382	3.317	0.568	2.025
LGMM	0.488	0.332	0.163	0.116	0.505	0.379	0.164	0.116	0.496	0.432	0.239	0.239
LGMMS	0.486	0.332	0.163	0.116	0.509	0.382	0.163	0.116	0.496	0.449	0.239	0.238
OLS	0.488	0.334	0.163	0.116	0.515	0.385	0.164	0.115	0.516	0.447	0.238	0.229
(D) Bi-modal mixture of normal distributions												
RKDRE	0.145	0.099	0.043	0.032	0.140	0.100	0.044	0.031	0.142	0.105	0.044	0.031
KDRE	0.142	0.099	0.043	0.032	0.138	0.100	0.044	0.032	0.154	0.114	0.046	0.031
YDG	0.144	0.099	0.043	0.032	0.140	0.100	0.044	0.031	0.155	0.131	0.045	0.031
SBS	0.143	0.101	0.044	0.032	0.139	0.102	0.044	0.032	0.148	0.109	0.045	0.031
LGMM	0.145	0.099	0.043	0.032	0.140	0.100	0.044	0.031	0.143	0.105	0.044	0.031
LGMMS	0.057	0.035	0.014	0.010	0.069	0.038	0.015	0.010	0.131	0.079	0.033	0.010
OLS	0.147	0.098	0.043	0.032	0.145	0.102	0.044	0.032	0.144	0.109	0.045	0.031
(E) Uniform distribution												
RKDRE	0.141	0.101	0.043	0.034	0.145	0.101	0.045	0.033	0.156	0.104	0.043	0.032
KDRE	0.120	0.085	0.036	0.029	0.131	0.087	0.038	0.029	0.146	0.093	0.037	0.028
YDG	0.141	0.101	0.043	0.034	0.145	0.101	0.045	0.033	0.157	0.104	0.043	0.032
SBS	0.104	0.072	0.032	0.026	0.131	0.083	0.034	0.027	0.149	0.095	0.035	0.026
LGMM	0.141	0.102	0.043	0.034	0.146	0.101	0.045	0.033	0.153	0.104	0.043	0.032
LGMMS	0.100	0.062	0.024	0.017	0.112	0.073	0.025	0.018	0.146	0.081	0.027	0.018
OLS	0.142	0.102	0.043	0.034	0.147	0.102	0.045	0.033	0.155	0.105	0.044	0.032

Continued on next page

Table A1 – continued from previous page

p	2				5				10			
n	50	100	500	1000	50	100	500	1000	50	100	500	1000
(F) Gamma(2,2)												
RKDRE	0.138	0.104	0.041	0.032	0.138	0.102	0.044	0.032	0.155	0.104	0.046	0.033
KDRE	0.153	0.126	0.076	0.065	0.154	0.126	0.080	0.066	0.165	0.127	0.080	0.065
YDG	0.138	0.104	0.041	0.032	0.138	0.102	0.045	0.032	0.156	0.106	0.046	0.033
SBS	0.173	0.150	0.094	0.080	0.159	0.135	0.092	0.077	0.168	0.122	0.083	0.070
LGMM	0.139	0.104	0.041	0.032	0.138	0.101	0.045	0.032	0.152	0.105	0.046	0.033
LGMMS	0.220	0.180	0.125	0.117	0.198	0.172	0.120	0.116	0.181	0.156	0.117	0.116
OLS	0.139	0.104	0.041	0.032	0.141	0.103	0.044	0.032	0.155	0.108	0.046	0.033
(G) Skewed mixture of normal distributions												
RKDRE	0.172	0.114	0.067	0.055	0.161	0.113	0.069	0.054	0.166	0.116	0.066	0.056
KDRE	0.202	0.145	0.091	0.075	0.187	0.140	0.093	0.075	0.197	0.145	0.089	0.077
YDG	0.171	0.114	0.067	0.055	0.161	0.113	0.068	0.054	0.167	0.116	0.066	0.056
SBS	0.236	0.157	0.086	0.068	0.204	0.148	0.089	0.068	0.200	0.150	0.085	0.070
LGMM	0.171	0.114	0.067	0.055	0.160	0.113	0.069	0.054	0.168	0.116	0.066	0.056
LGMMS	0.380	0.311	0.245	0.229	0.331	0.292	0.242	0.216	0.258	0.263	0.208	0.208
OLS	0.171	0.115	0.067	0.055	0.166	0.115	0.069	0.054	0.179	0.122	0.067	0.056
(H) Log-normal distribution												
RKDRE	0.147	0.110	0.044	0.032	0.146	0.098	0.046	0.029	0.144	0.101	0.045	0.032
KDRE	0.176	0.136	0.079	0.068	0.173	0.121	0.081	0.067	0.169	0.130	0.081	0.070
YDG	0.148	0.111	0.043	0.031	0.146	0.101	0.045	0.029	0.151	0.108	0.044	0.031
SBS	0.178	0.138	0.082	0.078	0.160	0.106	0.074	0.077	0.154	0.108	0.063	0.072
LGMM	0.148	0.110	0.044	0.032	0.145	0.098	0.046	0.029	0.146	0.102	0.045	0.032
LGMMS	0.189	0.137	0.072	0.058	0.191	0.149	0.082	0.061	0.175	0.154	0.098	0.069
OLS	0.148	0.108	0.044	0.032	0.149	0.100	0.046	0.030	0.152	0.107	0.046	0.032

Results are based on 500 replications.

Table A2: Comparison of bias of the slope coefficients

p	2				5				10			
n	50	100	500	1000	50	100	500	1000	50	100	500	1000
(A) Normal distribution												
RKDRE	0.003	0.000	0.000	0.000	0.005	0.003	0.001	0.001	0.006	0.003	0.002	0.002
KDRE	0.003	0.001	0.000	0.000	0.003	0.002	0.001	0.001	0.004	0.003	0.002	0.002
YDG	0.000	0.002	0.000	0.000	0.005	0.002	0.001	0.001	0.006	0.004	0.002	0.002
SBS	0.001	0.001	0.000	0.000	0.004	0.003	0.001	0.001	0.006	0.004	0.002	0.002
LGMM	0.004	0.000	0.000	0.000	0.004	0.002	0.001	0.001	0.003	0.004	0.002	0.002
LGMMS	0.006	0.000	0.000	0.000	0.004	0.002	0.001	0.001	0.004	0.004	0.002	0.002
OLS	0.004	0.002	0.001	0.000	0.004	0.003	0.001	0.001	0.003	0.004	0.002	0.002
(B) Variance-contaminated normal distribution												
RKDRE	0.002	0.002	0.000	0.000	0.002	0.000	0.001	0.000	0.003	0.002	0.001	0.000
KDRE	0.002	0.001	0.000	0.000	0.003	0.001	0.001	0.000	0.004	0.002	0.001	0.000
YDG	0.008	0.000	0.000	0.000	0.007	0.001	0.001	0.000	0.005	0.003	0.001	0.000
SBS	0.001	0.002	0.000	0.000	0.002	0.001	0.000	0.000	0.002	0.002	0.001	0.000
LGMM	0.000	0.001	0.000	0.000	0.002	0.001	0.001	0.000	0.003	0.002	0.001	0.000
LGMMS	0.000	0.002	0.000	0.000	0.002	0.001	0.000	0.000	0.002	0.002	0.001	0.000
OLS	0.001	0.001	0.000	0.002	0.005	0.004	0.002	0.001	0.006	0.003	0.002	0.001
(C) t-distribution with two degrees of freedom												
RKDRE	0.003	0.001	0.008	0.001	0.001	0.006	0.002	0.001	0.006	0.003	0.003	0.001
KDRE	0.002	0.003	0.009	0.001	0.001	0.007	0.002	0.001	0.009	0.004	0.003	0.001
YDG	0.008	0.039	0.010	0.003	0.042	0.090	0.012	0.005	0.014	0.036	0.010	0.002
SBS	0.001	0.002	0.011	0.001	0.007	0.006	0.001	0.001	0.029	0.021	0.002	0.003
LGMM	0.004	0.004	0.009	0.002	0.007	0.004	0.001	0.001	0.011	0.009	0.003	0.002
LGMMS	0.006	0.004	0.008	0.002	0.006	0.005	0.002	0.001	0.010	0.008	0.003	0.001
OLS	0.001	0.011	0.009	0.005	0.010	0.007	0.008	0.003	0.013	0.012	0.009	0.007
(D) Bi-modal mixture of normal distributions												
RKDRE	0.001	0.000	0.001	0.001	0.002	0.001	0.000	0.000	0.003	0.001	0.001	0.000
KDRE	0.002	0.000	0.001	0.001	0.002	0.001	0.000	0.000	0.003	0.001	0.001	0.000
YDG	0.000	0.000	0.001	0.001	0.003	0.001	0.000	0.000	0.003	0.001	0.001	0.000
SBS	0.003	0.003	0.001	0.002	0.007	0.003	0.001	0.000	0.002	0.001	0.001	0.001
LGMM	0.000	0.000	0.001	0.001	0.004	0.001	0.000	0.000	0.002	0.001	0.001	0.000
LGMMS	0.000	0.000	0.001	0.001	0.003	0.001	0.000	0.000	0.003	0.001	0.001	0.000
OLS	0.005	0.003	0.002	0.001	0.005	0.003	0.001	0.001	0.005	0.003	0.002	0.002
(E) Uniform distribution												
RKDRE	0.007	0.001	0.000	0.000	0.008	0.002	0.001	0.001	0.003	0.002	0.001	0.001
KDRE	0.009	0.002	0.000	0.000	0.008	0.002	0.001	0.001	0.004	0.003	0.001	0.001
YDG	0.007	0.001	0.000	0.000	0.008	0.002	0.002	0.001	0.004	0.002	0.001	0.001
SBS	0.005	0.003	0.000	0.000	0.005	0.003	0.001	0.001	0.003	0.003	0.002	0.001
LGMM	0.010	0.003	0.001	0.001	0.006	0.003	0.002	0.001	0.005	0.003	0.002	0.002
LGMMS	0.005	0.001	0.001	0.000	0.007	0.002	0.002	0.001	0.005	0.003	0.001	0.001
OLS	0.011	0.005	0.001	0.001	0.010	0.004	0.002	0.001	0.006	0.004	0.002	0.002

Continued on next page

Table A2 – continued from previous page

p	2				5				10			
n	50	100	500	1000	50	100	500	1000	50	100	500	1000
(F) Gamma(2,2)												
RKDRE	0.005	0.002	0.000	0.001	0.008	0.002	0.001	0.000	0.004	0.004	0.001	0.001
KDRE	0.006	0.003	0.000	0.001	0.008	0.003	0.001	0.000	0.005	0.005	0.001	0.001
YDG	0.001	0.002	0.002	0.001	0.007	0.003	0.001	0.001	0.006	0.005	0.001	0.001
SBS	0.003	0.001	0.000	0.002	0.006	0.003	0.001	0.001	0.006	0.005	0.001	0.000
LGMM	0.004	0.000	0.000	0.002	0.007	0.003	0.001	0.000	0.004	0.004	0.001	0.001
LGMMS	0.006	0.005	0.000	0.001	0.009	0.006	0.001	0.001	0.005	0.005	0.001	0.002
OLS	0.010	0.005	0.000	0.001	0.009	0.005	0.001	0.002	0.005	0.004	0.001	0.002
(G) Skewed mixture of normal distributions												
RKDRE	0.005	0.001	0.001	0.000	0.003	0.002	0.001	0.000	0.004	0.003	0.001	0.001
KDRE	0.005	0.001	0.000	0.000	0.004	0.002	0.000	0.000	0.005	0.002	0.001	0.000
YDG	0.004	0.002	0.001	0.000	0.004	0.003	0.000	0.000	0.004	0.003	0.001	0.000
SBS	0.007	0.003	0.001	0.001	0.005	0.002	0.000	0.000	0.005	0.003	0.001	0.001
LGMM	0.005	0.000	0.001	0.000	0.002	0.002	0.000	0.000	0.005	0.003	0.001	0.000
LGMMS	0.006	0.001	0.003	0.000	0.001	0.003	0.001	0.000	0.005	0.004	0.001	0.001
OLS	0.003	0.000	0.001	0.001	0.005	0.005	0.001	0.001	0.005	0.004	0.002	0.001
(H) Log-normal distribution												
RKDRE	0.003	0.002	0.000	0.000	0.002	0.001	0.001	0.000	0.004	0.001	0.000	0.000
KDRE	0.002	0.002	0.000	0.001	0.005	0.002	0.001	0.000	0.005	0.003	0.001	0.000
YDG	0.003	0.007	0.001	0.000	0.009	0.007	0.001	0.001	0.004	0.005	0.001	0.001
SBS	0.001	0.001	0.003	0.001	0.005	0.002	0.001	0.000	0.006	0.002	0.001	0.001
LGMM	0.002	0.003	0.000	0.000	0.005	0.002	0.001	0.000	0.005	0.002	0.001	0.001
LGMMS	0.004	0.001	0.003	0.001	0.006	0.002	0.001	0.001	0.006	0.004	0.001	0.001
OLS	0.004	0.006	0.003	0.000	0.007	0.003	0.002	0.001	0.007	0.005	0.002	0.001

Results are based on 500 replications. For $p = 5$ and $p = 10$, the bias of the slope coefficients is defined as the mean of the absolute bias of the $p - 1$ slope coefficients.

Table A3: Comparison of bias of the intercept coefficient

p	2				5				10			
n	50	100	500	1000	50	100	500	1000	50	100	500	1000
(A) Normal distribution												
RKDRE	0.008	0.001	0.002	0.001	0.003	0.000	0.002	0.003	0.008	0.002	0.001	0.002
KDRE	0.007	0.000	0.001	0.001	0.003	0.001	0.002	0.002	0.009	0.002	0.001	0.002
YDG	0.008	0.001	0.002	0.001	0.002	0.000	0.002	0.003	0.009	0.002	0.000	0.002
SBS	0.005	0.000	0.001	0.001	0.003	0.001	0.002	0.002	0.007	0.002	0.001	0.003
LGMM	0.008	0.001	0.002	0.001	0.003	0.001	0.003	0.003	0.009	0.003	0.000	0.002
LGMMS	0.008	0.002	0.001	0.001	0.002	0.000	0.002	0.002	0.009	0.001	0.000	0.002
OLS	0.008	0.001	0.002	0.001	0.002	0.001	0.003	0.003	0.008	0.002	0.000	0.002
(B) Variance-contaminated normal distribution												
RKDRE	0.003	0.001	0.000	0.001	0.013	0.001	0.001	0.002	0.003	0.007	0.000	0.002
KDRE	0.003	0.000	0.000	0.001	0.013	0.001	0.001	0.003	0.004	0.007	0.001	0.002
YDG	0.002	0.002	0.000	0.001	0.012	0.001	0.001	0.002	0.001	0.008	0.000	0.002
SBS	0.003	0.001	0.001	0.001	0.014	0.001	0.001	0.003	0.005	0.005	0.000	0.002
LGMM	0.003	0.001	0.000	0.001	0.013	0.001	0.001	0.002	0.004	0.007	0.000	0.002
LGMMS	0.001	0.000	0.000	0.000	0.000	0.003	0.001	0.000	0.002	0.003	0.001	0.001
OLS	0.003	0.001	0.000	0.001	0.014	0.001	0.001	0.002	0.004	0.007	0.000	0.002
(C) t-distribution with two degrees of freedom												
RKDRE	0.015	0.002	0.001	0.002	0.005	0.004	0.000	0.006	0.019	0.011	0.008	0.010
KDRE	0.015	0.002	0.002	0.002	0.007	0.004	0.000	0.005	0.022	0.013	0.008	0.011
YDG	0.017	0.001	0.006	0.002	0.003	0.008	0.004	0.006	0.009	0.002	0.004	0.003
SBS	0.021	0.003	0.002	0.002	0.038	0.006	0.000	0.006	0.050	0.075	0.025	0.091
LGMM	0.015	0.002	0.001	0.002	0.004	0.005	0.000	0.006	0.019	0.010	0.008	0.010
LGMMS	0.015	0.002	0.001	0.002	0.004	0.005	0.000	0.006	0.020	0.011	0.008	0.010
OLS	0.015	0.001	0.001	0.002	0.006	0.004	0.001	0.006	0.025	0.012	0.009	0.010
(D) Bi-modal mixture of normal distributions												
RKDRE	0.000	0.005	0.002	0.000	0.008	0.000	0.001	0.000	0.004	0.006	0.000	0.000
KDRE	0.001	0.004	0.001	0.000	0.010	0.000	0.001	0.000	0.005	0.007	0.000	0.000
YDG	0.000	0.005	0.002	0.000	0.008	0.000	0.001	0.000	0.004	0.006	0.000	0.000
SBS	0.001	0.004	0.001	0.000	0.008	0.000	0.000	0.000	0.008	0.007	0.000	0.001
LGMM	0.000	0.005	0.002	0.000	0.007	0.000	0.001	0.000	0.006	0.006	0.000	0.000
LGMMS	0.002	0.001	0.001	0.000	0.004	0.001	0.000	0.000	0.006	0.003	0.002	0.000
OLS	0.000	0.004	0.002	0.000	0.008	0.000	0.001	0.000	0.007	0.006	0.001	0.002
(E) Uniform distribution												
RKDRE	0.014	0.004	0.003	0.002	0.013	0.003	0.003	0.002	0.007	0.001	0.000	0.001
KDRE	0.010	0.004	0.002	0.002	0.012	0.003	0.003	0.001	0.005	0.001	0.000	0.001
YDG	0.014	0.004	0.003	0.002	0.013	0.003	0.003	0.002	0.007	0.002	0.000	0.001
SBS	0.008	0.004	0.001	0.002	0.014	0.002	0.002	0.001	0.004	0.002	0.000	0.000
LGMM	0.014	0.004	0.003	0.002	0.013	0.004	0.003	0.002	0.005	0.001	0.000	0.001
LGMMS	0.008	0.001	0.002	0.001	0.011	0.003	0.003	0.000	0.003	0.003	0.002	0.000
OLS	0.013	0.004	0.003	0.002	0.012	0.004	0.003	0.002	0.006	0.001	0.000	0.001

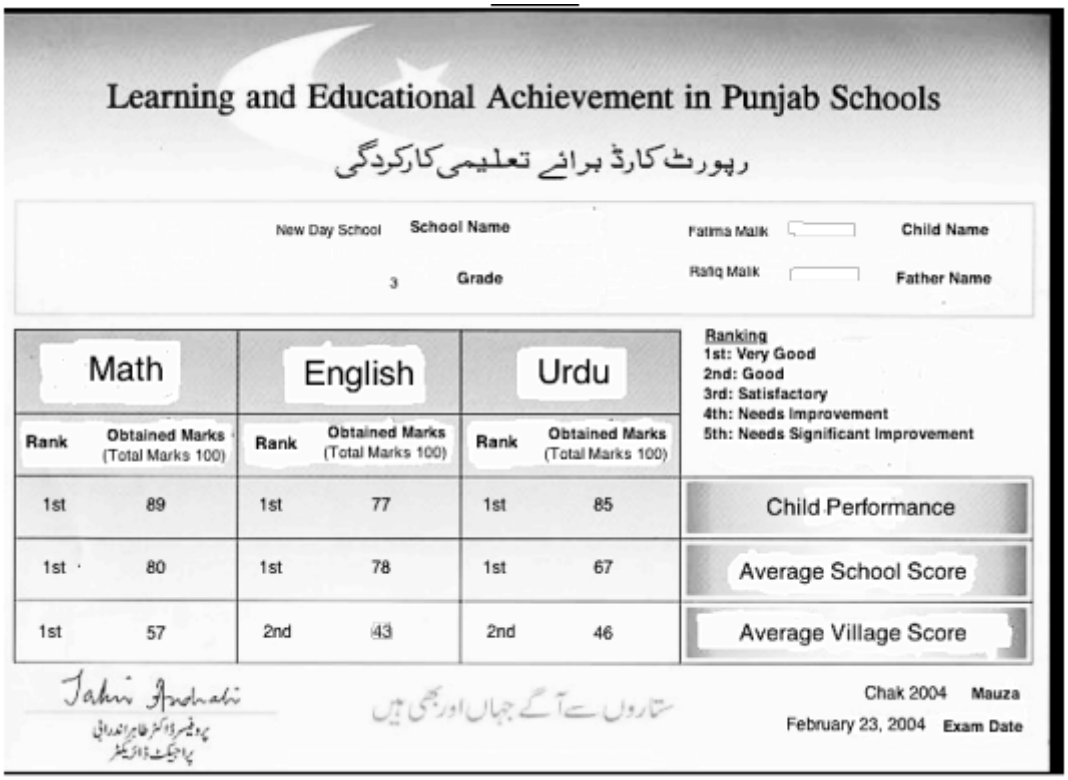
Continued on next page

Table A3 – continued from previous page

p	2				5				10			
n	50	100	500	1000	50	100	500	1000	50	100	500	1000
(F) Gamma(2,2)												
RKDRE	0.011	0.007	0.000	0.001	0.009	0.000	0.003	0.000	0.008	0.000	0.003	0.002
KDRE	0.048	0.060	0.062	0.055	0.047	0.063	0.064	0.057	0.050	0.058	0.064	0.055
YDG	0.011	0.007	0.002	0.001	0.009	0.001	0.003	0.000	0.003	0.001	0.003	0.002
SBS	0.073	0.090	0.082	0.073	0.051	0.072	0.077	0.069	0.042	0.050	0.067	0.060
LGMM	0.011	0.008	0.000	0.001	0.009	0.001	0.003	0.000	0.007	0.000	0.003	0.002
LGMMS	0.107	0.123	0.115	0.112	0.103	0.113	0.109	0.110	0.057	0.094	0.107	0.110
OLS	0.011	0.008	0.000	0.001	0.008	0.000	0.003	0.000	0.004	0.000	0.003	0.002
(G) Skewed mixture of normal distributions												
RKDRE	0.046	0.047	0.046	0.043	0.033	0.042	0.048	0.041	0.054	0.040	0.043	0.045
KDRE	0.097	0.091	0.075	0.066	0.076	0.084	0.077	0.065	0.096	0.081	0.071	0.068
YDG	0.047	0.047	0.046	0.043	0.033	0.042	0.048	0.041	0.056	0.040	0.043	0.045
SBS	0.107	0.089	0.066	0.057	0.079	0.082	0.068	0.056	0.095	0.078	0.063	0.060
LGMM	0.046	0.047	0.046	0.043	0.031	0.043	0.048	0.041	0.053	0.040	0.043	0.045
LGMMS	0.079	0.127	0.187	0.201	0.092	0.066	0.187	0.180	0.020	0.010	0.133	0.177
OLS	0.045	0.047	0.046	0.043	0.030	0.044	0.048	0.041	0.055	0.042	0.043	0.045
(H) Log-normal distribution												
RKDRE	0.003	0.002	0.000	0.001	0.000	0.009	0.001	0.001	0.003	0.001	0.002	0.002
KDRE	0.063	0.062	0.061	0.058	0.056	0.052	0.061	0.058	0.047	0.061	0.063	0.060
YDG	0.004	0.002	0.000	0.001	0.001	0.009	0.001	0.000	0.001	0.003	0.000	0.002
SBS	0.073	0.068	0.068	0.072	0.036	0.033	0.059	0.071	0.020	0.028	0.047	0.066
LGMM	0.003	0.002	0.001	0.001	0.000	0.009	0.001	0.001	0.001	0.002	0.002	0.024
LGMMS	0.040	0.038	0.032	0.032	0.108	0.009	0.050	0.040	0.110	0.113	0.075	0.050
OLS	0.003	0.003	0.000	0.001	0.010	0.008	0.001	0.001	0.001	0.001	0.002	0.003

Results are based on 500 replications.

A.3 Example of report card



Learning and Educational Achievement in Punjab Schools

رپورٹ کارڈ برائے تعلیمی کارکردگی

New Day School — School Name — Fatima Malik — Child Name

3 — Grade — Rafiq Malik — Father Name

Math		English		Urdu	
Rank	Obtained Marks (Total Marks 100)	Rank	Obtained Marks (Total Marks 100)	Rank	Obtained Marks (Total Marks 100)
1st	89	1st	77	1st	85
1st	80	1st	78	1st	67
1st	57	2nd	43	2nd	46

Ranking
1st: Very Good
2nd: Good
3rd: Satisfactory
4th: Needs Improvement
5th: Needs Significant Improvement

Child Performance

Average School Score

Average Village Score

Tahir Andrabi
پروفیسر ڈاکٹر طاہر اندرابی
پراجیکٹ ڈائریکٹر

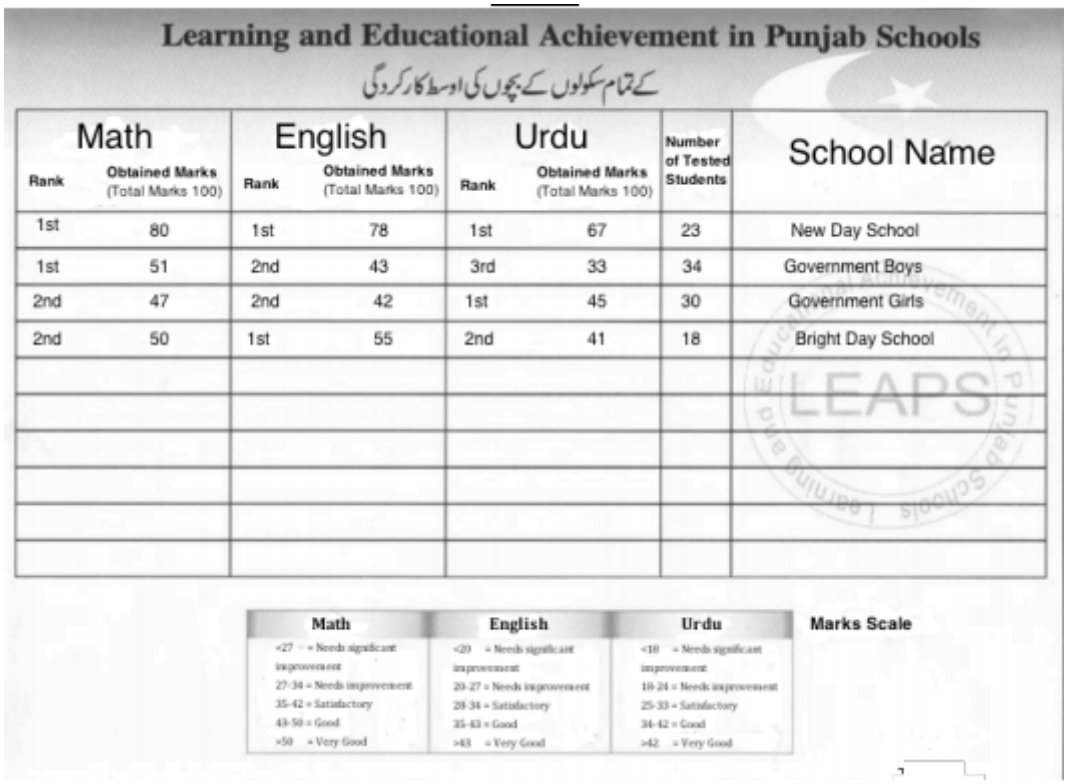
ستاروں سے آگے جہاں اور بھی ہیں

Chak 2004 — Mauza

February 23, 2004 — Exam Date

Figure A1: Card 1 of the report card

Source: Andrabi et al. (2017, Online appendix)



Learning and Educational Achievement in Punjab Schools

کے تمام سکولوں کے بچوں کی اوسط کارکردگی

Math		English		Urdu		Number of Tested Students	School Name
Rank	Obtained Marks (Total Marks 100)	Rank	Obtained Marks (Total Marks 100)	Rank	Obtained Marks (Total Marks 100)		
1st	80	1st	78	1st	67	23	New Day School
1st	51	2nd	43	3rd	33	34	Government Boys
2nd	47	2nd	42	1st	45	30	Government Girls
2nd	50	1st	55	2nd	41	18	Bright Day School

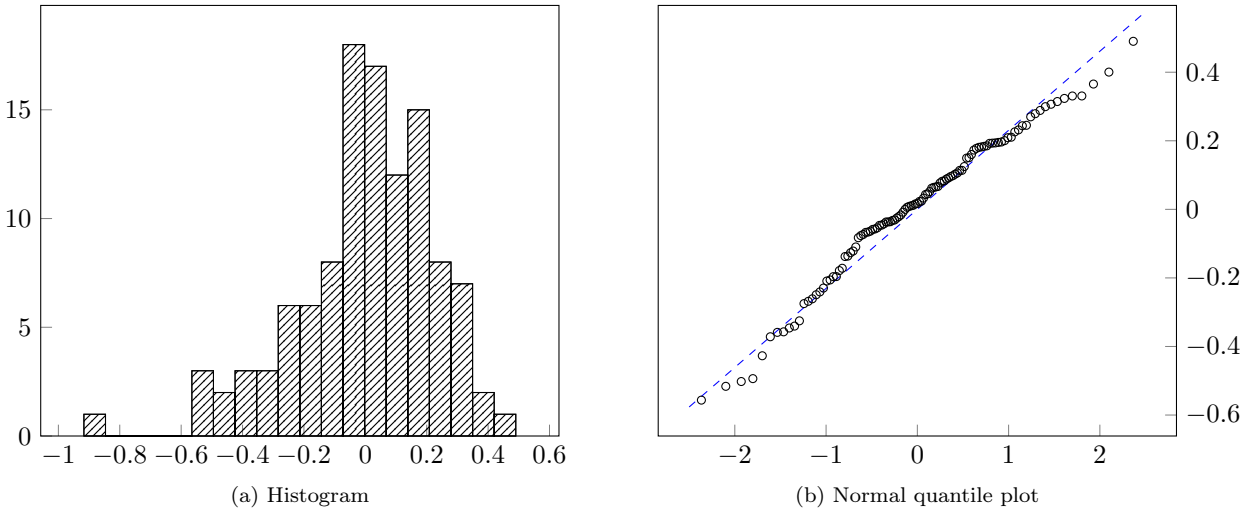
Math	English	Urdu
<27 = Needs significant improvement	<20 = Needs significant improvement	<18 = Needs significant improvement
27-34 = Needs improvement	20-27 = Needs improvement	18-24 = Needs improvement
35-42 = Satisfactory	28-34 = Satisfactory	25-33 = Satisfactory
43-50 = Good	35-43 = Good	34-42 = Good
>50 = Very Good	>43 = Very Good	>42 = Very Good

Marks Scale

Figure A2: Card 2 of the report card

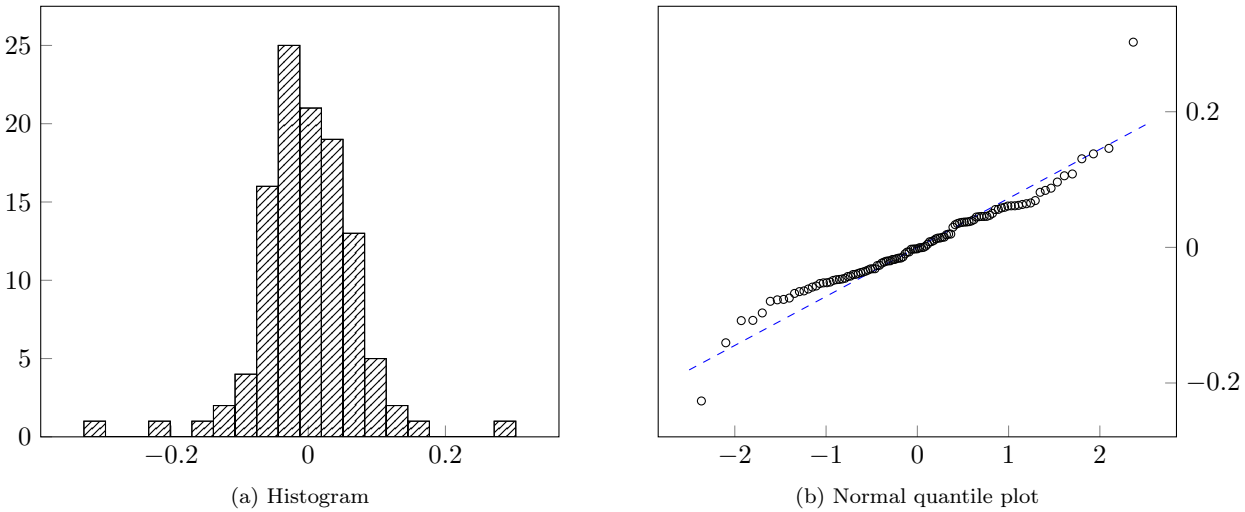
Source: Andrabi et al. (2017, Online appendix)

A.4 Residual diagnostics for Model 2 and Model 3



(a) Histogram (b) Normal quantile plot

Figure A3: Residual diagnostics for Model 2



(a) Histogram (b) Normal quantile plot

Figure A4: Residual diagnostics for Model 3

B R code

B.1 rkdre()

```
rkdre <- function(data, formula, tol_beta = 1e-12, tol_EM = 1e-12, max.iter = 100,
      bandwidth = c("normal","robust1","robust2","SJ","bcv","ucv"),
      update_bw = FALSE){

  n <- nrow(data)
  resid <- NULL
  h <- NULL
  std_error_trimmed <- NULL
  std_error_untrimmed <- NULL

  #compute OLS estimate
  OLS_model <- lm(data = data, formula = formula)
  beta_OLS <- OLS_model$coefficients

  #set initial estimate
  beta <- beta_OLS

  #set initial bandwidth
  if (bandwidth == "normal"){
    h <- 1.06 * sd(OLS_model$residuals) * n^(-0.2)
  } else if (bandwidth == "robust1"){
    h <- 0.79 * IQR(OLS_model$residuals) * n^(-0.2)
  } else if (bandwidth == "robust2"){
    h <- 0.9 * min(sd(OLS_model$residuals),IQR(OLS_model$residuals)/1.34) * n^(-0.2)
  } else if (bandwidth == "SJ"){
    h <- bw.SJ(OLS_model$residuals)
  } else if (bandwidth == "bcv"){
    h <- bw.bcv(OLS_model$residuals)
  } else if (bandwidth == "ucv"){
    h <- bw.ucv(OLS_model$residuals)
  }

  #create matrix of explanatory variables with intercept
  X <- model.matrix(formula, as.data.frame(data))

  #column of dependent variable
  y <- data[,which(colnames(data) == all.vars(formula)[1])]

  #create A for in EM loop
  A <- solve( t(X) %*% X )

```

```
converged_beta <- 0
counter <- 0

while (converged_beta == 0 && counter < max.iter){ #start beta loop

  counter <- counter + 1
  beta_prev <- beta
  beta_EM <- beta_prev
  converged_EM <- 0

  #create residuals
  resid <- y - X %*% beta_prev

  if (update_bw == TRUE){
    #update bandwidth
    if (bandwidth == "normal"){
      h <- 1.06 * sd(resid) * n^(-0.2)
    } else if (bandwidth == "robust1"){
      h <- 0.79 * IQR(resid) * n^(-0.2)
    } else if (bandwidth == "robust2"){
      h <- 0.9 * min(sd(resid),IQR(resid)/1.34) * n^(-0.2)
    } else if (bandwidth == "SJ"){
      h <- bw.SJ(resid)
    } else if (bandwidth == "bcv"){
      h <- bw.bcv(resid)
    } else if (bandwidth == "ucv"){
      h <- bw.ucv(resid)
    }
  }

  resid_matrix <- do.call(rbind, replicate(n, t(resid), simplify=FALSE))

  while (converged_EM == 0){ #start EM loop

    beta_prev_EM <- beta_EM

    resid_EM <- y - X %*% beta_prev_EM

    p_matrix <- matrix(rep(resid_EM,n), nrow = n, ncol = n) - resid_matrix

    p_matrix <- exp( -0.5 * (p_matrix / h )^2 )

    p_matrix <- p_matrix / ( rowSums(p_matrix) )

```

```
86        #split beta_update in two parts (beta_OLS and A %*% (B-C))
87        #A is solve( t(X) %*% X) is created outside the loops
88
89        PE_sum <- p_matrix %*% resid
90
91        B <- colSums(X * as.vector(PE_sum))
92
93        C <- mean(PE_sum) * colSums(X)
94
95        #update beta
96        beta_EM <- beta_OLS - A %*% (B - C)
97
98
99        if(max((beta_EM - beta_prev_EM)^2) < tol_EM){
100         converged_EM <- 1
101         beta <- beta_EM
102         print(beta)
103       }
104
105     } #end of EM loop
106
107     if (max((beta - beta_prev)^2) < tol_beta){
108       converged_beta <- 1
109       resid <- y - X %*% beta
110     }
111   } #end of beta loop
112
113   if (counter >= max.iter){
114     warning('RKDRE did not converge in specified maximum number of iterations')
115   }
116
117   return(list(coefficients = beta, iterations = counter, residuals = resid))
118 }
```

B.2 kdre()

```
kdre <- function(data, formula, tol_EM = 1e-12,
                 bandwidth = c("normal","robust1","robust2","SJ","bcv","ucv")){

  beta <- NULL
  h <- NULL
  resid <- NULL
  n <- nrow(data)

  OLS_model <- lm(data = data, formula = formula)
  beta_OLS <- OLS_model$coefficients

  #set initial bandwidth
  if (bandwidth == "normal"){
    h <- 1.06 * sd(OLS_model$residuals) * n^(-0.2)
  } else if (bandwidth == "robust1"){
    h <- 0.79 * IQR(OLS_model$residuals) * n^(-0.2)
  } else if (bandwidth == "robust2"){
    h <- 0.9 * min(sd(OLS_model$residuals),IQR(OLS_model$residuals)/1.34) * n^(-0.2)
  } else if (bandwidth == "SJ"){
    h <- bw.SJ(OLS_model$residuals)
  } else if (bandwidth == "bcv"){
    h <- bw.bcv(OLS_model$residuals)
  } else if (bandwidth == "ucv"){
    h <- bw.ucv(OLS_model$residuals)
  }

  #create matrix of explanatory variables with intercept
  X <- model.matrix(formula, as.data.frame(data))

  #column of dependent variable
  y <- data[,which(colnames(data) == all.vars(formula)[1])]

  #create A for in EM loop
  A <- solve( t(X) %*% X )

  #initialize EM algorithm
  beta <- beta_OLS
  converged_EM <- 0

  #create residuals
  resid_OLS <- y - X %*% beta

  resid_matrix_OLS <- do.call(rbind, replicate(n, t(resid_OLS), simplify=FALSE))
```

```
44
45  while (converged_EM == 0){ #start EM loop
46
47    beta_prev <- beta
48
49    resid <- y - X %*% beta_prev
50
51    #create matrix of classification probabilities
52    p_matrix <- matrix(rep(resid,n), nrow = n, ncol = n) - resid_matrix_OLS
53    p_matrix <- exp( -0.5 * (p_matrix / h )^2  )
54    p_matrix <- p_matrix / ( rowSums(p_matrix)  )
55
56    PE_sum <- p_matrix %*% resid_OLS
57
58    #update beta
59    beta <- beta_OLS - A %*% colSums(X * as.vector(PE_sum))
60
61    if(max((beta - beta_prev)^2) < tol_EM){
62      converged_EM <- 1
63      resid <- y - X %*% beta
64    }
65
66  } #end of EM loop
67
68  return(list(coefficients = beta, residuals = resid))
69 }
```

B.3 ydg()

```
ydg <- function(data, formula, tol_EM = 1e-7, optim = c("EM", "ML"),
                bandwidth = c("normal","robust1","robust2","SJ","bcv","ucv")){

  beta <- NULL
  h <- NULL
  n <- nrow(data)

  OLS_model <- lm(data = data, formula = formula)
  beta_OLS <- OLS_model$coefficients
  p <- length(beta_OLS)

  #set initial bandwidth
  if (bandwidth == "normal"){
    h <- 1.06 * sd(OLS_model$residuals) * n^(-0.2)
  } else if (bandwidth == "robust1"){
    h <- 0.79 * IQR(OLS_model$residuals) * n^(-0.2)
  } else if (bandwidth == "robust2"){
    h <- 0.9 * min(sd(OLS_model$residuals),IQR(OLS_model$residuals)/1.34) * n^(-0.2)
  } else if (bandwidth == "SJ"){
    h <- bw.SJ(OLS_model$residuals)
  } else if (bandwidth == "bcv"){
    h <- bw.bcv(OLS_model$residuals)
  } else if (bandwidth == "ucv"){
    h <- bw.ucv(OLS_model$residuals)
  }

  #create matrix of explanatory variables with intercept
  X <- model.matrix(formula, as.data.frame(data))

  #column of dependent variable
  y <- data[,which(colnames(data) == all.vars(formula)[1])]

  if (optim == "EM"){

    if (p == 2){

      #create matrix of differences in explanatory variables (xi- xj)
      X_diff <- rep(X[,2],n) - do.call(rbind, replicate(n, t(X[,2]), simplify=FALSE))

      #create matrix of differences in dependent variable (yi- yj)
      y_diff <- rep(y,n) - do.call(rbind, replicate(n, t(y), simplify=FALSE))

      beta_EM <- beta_OLS[-1]
```

```
    converged_EM <- 0

    while (converged_EM == 0){

      beta_prev_EM <- beta_EM

      #create residual matrix
      resid_matrix <- y_diff - X_diff * beta_prev_EM

      #create matrix of classification probabilities
      p_matrix <- exp(-0.5 * ( resid_matrix / h )^2 )
      #use leave one out
      diag(p_matrix) <- 0
      #divide by sum of the rows such that probabilities sum to 1
      p_matrix <- p_matrix / rowSums(p_matrix)

      #update beta
      beta_EM <- 1/( sum(p_matrix * X_diff^2 )) * sum(p_matrix * X_diff * y_diff)

      if(max((beta_EM - beta_prev_EM)^2) < tol_EM){
        converged_EM <- 1
        beta <- beta_EM
      }
    } # end of EM loop
  }

  if (p > 2){
    #create three-dimensional array of differences in indep. variables (xi- xj)
    X_diff <- array(0,dim = c(n,n,(p-1)))
    for (i in 1:(p-1)){
      X_diff[,,i] <- rep(X[,(i+1)],n) -
        do.call(rbind, replicate(n, t(X[,(i+1)]), simplify=FALSE))
    }

    #create matrix of differences in dependent variable (yi- yj)
    y_diff <- rep(y,n) - do.call(rbind, replicate(n, t(y), simplify=FALSE))

    beta_EM <- beta_OLS[-1]
    converged_EM <- 0

    while (converged_EM == 0){ #start EM loop

      beta_prev_EM <- beta_EM
      resid <- y - as.matrix(X[,-1]) %*% beta_prev_EM
```

```
88
89        #create residual matrix
90        resid_matrix <- matrix(rep(resid,n), nrow = n, ncol = n)
91        resid_matrix <- resid_matrix -t(resid_matrix)
92
93        #create matrix of classification probabilities
94        p_matrix <- exp(-0.5 * ( resid_matrix / h )^2 )
95        #use leave one out
96        diag(p_matrix) <- 0
97        #divide by sum of the rows such that probabilities sum to 1
98        p_matrix <- p_matrix / rowSums(p_matrix)
99
100       #the inner apply computes the outer product of X_diff[i,j,]
101       #the outer products are structured with array
102       #sweep multiplies each outer product element with p[i,j]
103       #the outer apply sums over the matrices of outer products
104       A <- apply(sweep(array(apply(X_diff, c(1,2), function(x){x %*% t(x)}),
105                 dim = c(p-1,p-1,n,n)), c(3,4), FUN = "*",p_matrix), c(1,2), sum)
106       B <- apply(sweep(X_diff, c(1,2), FUN = "*", (y_diff * p_matrix)),3,sum)
107
108       #update beta
109       beta_EM <- solve(A) %*% B
110
111       if(max((beta_EM - beta_prev_EM)^2) < tol_EM){
112         converged_EM <- 1
113         beta <- beta_EM
114       }
115     } #end of EM loop
116   }
117 }
118
119 if (optim == "ML"){
120
121   fn <- function(beta, y, X, n, h){
122
123     #create residuals
124     resid <- y - as.matrix(X[,-1]) %*% beta
125
126     #create matrix of differences
127     resid_matrix <- matrix(rep(resid,n), nrow = n, ncol = n)
128     resid_matrix <- ( resid_matrix - t(resid_matrix) ) / h
129
130     #compute Gaussian kernel scores
131     K_matrix <- exp(-0.5 * resid_matrix^2)
```

```
132       #use leave one out density
133       diag(K_matrix) <- 0
134
135       #compute leave one out density estimates
136       f_vector <- 1/( (n-1) * h ) * rowSums(K_matrix)
137
138       #compute (negative) likelihood
139       fn <- -1*sum(log(f_vector))
140       fn
141     }
142
143     #if initial value is infinite, return NULL
144     if ( is.infinite(fn(beta = beta_OLS[-1],y = y, X = X, n = n, h = h)) == TRUE ){
145       return(list(coefficients = rep(NaN,(nvar+1))))
146       warning('initial estimated log-likelihood in YDG is (minus) infinite')
147     }
148
149     beta <- optim(par = beta_OLS[-1], fn = fn, method = "BFGS",
150                   y = y, X = X, n = n, h = h)$par
151   }
152
153   #compute intercept term
154   intercept <- mean(y - as.matrix(X[,-1]) %*% beta)
155   beta <- c(intercept,beta)
156
157   return(list(coefficients = beta))
158 }
```

B.4 sbs()

```
sbs <- function(data, formula, t = 8,
                bandwidth = c("normal","robust1","robust2","SJ","bcv","ucv")){

  n <- nrow(data)
  h <- NULL

  #compute trimming parameters
  a <- t
  b <- exp(-0.5*t^2)
  c <- t

  OLS_model <- lm(data = data, formula = formula)
  beta <- OLS_model$coefficients

  #set initial bandwidth
  if (bandwidth == "normal"){
    h <- 1.06 * sd(OLS_model$residuals) * n^(-0.2)
  } else if (bandwidth == "robust1"){
    h <- 0.79 * IQR(OLS_model$residuals) * n^(-0.2)
  } else if (bandwidth == "robust2"){
    h <- 0.9 * min(sd(OLS_model$residuals),IQR(OLS_model$residuals)/1.34) * n^(-0.2)
  } else if (bandwidth == "SJ"){
    h <- bw.SJ(OLS_model$residuals)
  } else if (bandwidth == "bcv"){
    h <- bw.bcv(OLS_model$residuals)
  } else if (bandwidth == "ucv"){
    h <- bw.ucv(OLS_model$residuals)
  }

  #create matrix of explanatory variables with intercept
  X <- model.matrix(formula, as.data.frame(data))

  #column of dependent variable
  y <- data[,which(colnames(data) == all.vars(formula)[1])]

  #compute OLS residuals
  resid <- y - X %*% beta

  #create matrix for score computation
  resid_matrix <- matrix(rep(resid,n), nrow = n, ncol = n)

  resid_matrix <- ( resid_matrix - t(resid_matrix) ) / h

```

```
  exp_resid_matrix <- exp( -0.5 * (resid_matrix^2) )

  #compute kernel density (up to scale factor)
  f_u <- rowSums( exp_resid_matrix )

  #compute derivative of kernel density (up to scale factor)
  df_u <- rowSums( -1 * resid_matrix * exp_resid_matrix )

  #compute score function
  score <- 1/h * df_u / f_u

  #trim score
  trim_matrix <- cbind(resid,f_u,score)

  trimmed_score <- apply(trim_matrix, 1, function(x){
    if (abs(x[1]) < a && x[2] > b && abs(x[3]) < c ){
      x <- x[3]
    }
    else{
      x <- 0
    }
    return(x)
  })

  score_X <- X * trimmed_score

  #compute the SBS estimator
  beta <- beta - n / sum(trimmed_score^2) * solve(t(X) %*% X) %*% colSums(score_X)

  resid <- y - X %*% beta

  return(list(coefficients = beta, residuals = resid, h = h))
}
```

B.5 lgmm()

```
lgmm <- function(data, formula, J = 3, symmetric = TRUE,
                 transformed = TRUE, weighted = FALSE){

  n <- nrow(data)
  resid_transformed <- NULL
  resid_weighted <- NULL
  zeta_matrix <- NULL
  w_matrix <- NULL

  #create matrix of explanatory variables with intercept
  X <- model.matrix(formula, as.data.frame(data))

  #column of dependent variable
  y <- data[,which(colnames(data) == all.vars(formula)[1])]

  #starting values for optimization routine
  OLS_model <- lm(data = data, formula = formula)
  beta <- OLS_model$coefficients
  sigma <- sd(OLS_model$residuals)

  #create residuals
  resid <- y - X %*% beta

  if (symmetric == TRUE){
    if (transformed == FALSE && weighted == FALSE){

      #create zeta matrix
      zeta_matrix <- resid
      for (j in 2:J){
        zeta_matrix <- cbind(zeta_matrix,resid^( 2*j -1 ))
      }

      #create w_matrix
      w_matrix <- rep(1,n)
      for (j in 2:J){
        w_matrix <- cbind(w_matrix,( 2*j-1 )*resid^( 2*j - 2 ))
      }
    } else if (transformed == TRUE){

      resid_transformed <- resid / (1 + abs(resid))

      #create zeta matrix
      zeta_matrix <- resid_transformed
```

```
    for (j in 2:J){
      zeta_matrix <- cbind(zeta_matrix,resid_transformed^( 2*j -1 ))
    }

    zeta_matrix <- scale(zeta_matrix, center = TRUE, scale = FALSE)

    #create vector that is 1 (-1) for positive (negative) residuals
    ind <- rep(-1,n)
    ind[which(resid >= 0)] <- 1

    #compute 'inner' derivative of w
    w_deriv <- ( rep(1,n) + abs(resid) - resid * ind ) / ( 1 + abs(resid) )^2
    w_matrix <- w_deriv
    for (j in 2:J){
      w_matrix <- cbind(w_matrix,( 2*j-1 ) * resid_transformed^( 2*j - 2 )
                        * w_deriv)
    }

  }
} else if (symmetric == FALSE){
  if (transformed == FALSE && weighted == FALSE){

    #create zeta matrix
    zeta_matrix <- resid
    for (j in 2:J){
      zeta_matrix <- cbind( zeta_matrix,resid^j )
    }
    zeta_matrix <- scale(zeta_matrix, center = TRUE, scale = FALSE)
    #create w_matrix
    w_matrix <- rep(1,n)
    for (j in 2:J){
      w_matrix <- cbind(w_matrix, j * resid^( j - 1 ))
    }

  } else if (transformed == TRUE){

    resid_transformed <- resid / (1 + abs(resid))

    #create zeta matrix
    zeta_matrix <- resid_transformed
    for (j in 2:J){
      zeta_matrix <- cbind(zeta_matrix,resid_transformed^j)
    }

```

```
88      zeta_matrix <- scale(zeta_matrix, center = TRUE, scale = FALSE)
89
90      #create vector that is 1 (-1) for positive (negative) residuals
91      ind <- rep(-1,n)
92      ind[which(resid >= 0)] <- 1
93
94      #compute 'inner' derivative of w
95      w_deriv <- ( rep(1,n) + abs(resid) - resid * ind ) / ( 1 + abs(resid) )^2
96      w_matrix <- w_deriv
97      for (j in 2:J){
98        w_matrix <- cbind(w_matrix, j * resid_transformed^( j-1 ) * w_deriv)
99      }
100   } else if (weighted == TRUE){
101
102     #create weights of residuals
103     resid_weights <- as.vector(exp(-0.5 * resid^2))
104
105     zeta_matrix <- resid
106     for (j in 2:J){
107       zeta_matrix <- cbind(zeta_matrix,resid^( 2*j -1 ))
108     }
109     zeta_matrix <- zeta_matrix * resid_weights
110
111     zeta_matrix <- scale(zeta_matrix, center = TRUE, scale = FALSE)
112
113     w_matrix <- ( c(1,n) - resid^2 ) * resid_weights
114     for (j in 2:J){
115       w_matrix <- cbind(w_matrix, ( j * resid^(j-1) - resid^(j+1) )
116                         * resid_weights )
117     }
118   }
119 }
120
121 #create covariance matrix of moments
122 V <- ( n-1 ) / n * cov(zeta_matrix)
123
124 #compute sample moments
125 w <- colMeans(w_matrix)
126
127 #precompute some matrices
128 A <- t(X) %*% X
129 wX <- kronecker(t(w),A)
130 VinvAinv <- kronecker(solve(V),solve(A))
131
```

```
132   #update beta
133   beta <- beta + solve( wX %*% VinvAinv %*% t(wX) ) %*% wX %*% VinvAinv %*%
134     kronecker(diag(J),t(X)) %*% as.vector(zeta_matrix)
135
136   return(list(coefficients = beta))
137 }
```