University of Zurich

Master Thesis

GMM Estimation with Many Moment Conditions Using Regularized Jackknife GMM

Name: Luca Gaegauf

Student number: 10-750-453

Supervisor: Dr. Damian Kozbur
Department of Economics

November 29, 2017

Contents

List of Figures

List of Tables

Glossary

CUE	Continuous updating estimator
DGP	Data generating process
GIV	Generalized instrumental variable
GMM	Generalized method of moments
HK	Hansen and Kozbur (2014)
i.i.d.	Independent and identically distributed
IQR	Interquartile range
IV	Instrumental variable
JIVE	Jackknife IV estimator
JIVE1	Phillips and Hale's (1977) JIVE form
JIVE2	Angrist, Imbens, and Krueger's (1995) JIVE form
JGMM	Jackknife GMM
Med. Bias	Median bias
MSE	Mean squared error
NW	Newey and Windmeijer (2009)
RJIVE	Regularized JIVE
RJGMM	Regularized JGMM
RJGMM$\star$	Oracle version of RJGMM

Notation

Data parameters:

n	Sample size
p	Dimension of structural parameter
m	Dimension of moment conditions
m^*	Dimension of selected moment conditions
β	Structural parameter of interest
B	Compact parameter search space
μ_n^2	Concentration parameter
Π	First-stage structural parameter

Data

w	Data ($w = (y, X, Z)$ if data generating process is GIV)
U, ϵ	First and second-stage error terms
σ_ϵ^2, σ_U^2	Variance of error terms
$\sigma_{\epsilon U}$	Covariance of error terms

GMM specific notation:

$g(\cdot)$	GMM moment condition
$G(\cdot)$	Derivative of $g(\cdot)$
$\Omega(\cdot)$	Variance of $g(\cdot)$
W	GMM weighting matrix
Λ (λ)	Ridge penalty matrix (term)
$\hat{A}$ ($\hat{A}^\Lambda$)	(Regularized) GMM selection matrix

Estimators:

$\ddot{\beta}$	Two-step GMM estimator
$\breve{\beta}$	Jackknife GMM estimator
$\tilde{\beta}$	RJGMM estimator
$\dot{\beta}$ ($\dot{\beta}_i$)	preliminary consistent estimators (on the without-i^{th} sample)
$\hat{\beta}$	Generic GMM estimator

Indexes:

i	i^{th} observation
$-i$	All but the i^{th} observation

Misc. notation:

ι_q	$1 \times q$ vector of ones
0_q	$1 \times q$ vector of zeros
I_q	$q \times q$ identity matrix

Abstract

Using many moment conditions is a well-known remedy for generalized method of moments' (GMMs) imprecision. However, it comes at the cost of an increased bias and/or instability of the weighting matrix. We focus on scenarios where the dimension of available moment conditions may be larger than the sample size and where the signal contained in the moments may not be sparse. In such situations, consistent estimation of the standard GMM estimator is not possible without regularization. Model selection methods have demonstrated their ineptitude in these scenarios. We propose a jackknife GMM (JGMM) estimator with regularization at each jackknife iteration. We use Monte Carlo simulations to investigate the proposed ridge-regularized JGMM (RJGMM) estimator's finite-sample properties and provide evidence that it performs favorably relative to other estimation strategies that are valid in many-weak moment settings.

Acknowledgments

I am deeply indebted to my adviser Damian Kozbur for his guidance throughout this thesis. I am also grateful to Chris Leuenberger and Fred Gaegauf for the provision of additional computers to complete the simulations on time and to David Eichmann, Marco Kaiser, and Suzanne Ariza for their comments.

1 Introduction

Generalized method of moments (GMM, Hansen, 1982) has become a popular tool in econometrics research due to its ability to be applied to situations requiring few assumptions. However, this generality often comes at the cost of poor finite-sample properties and/or low precision. Many generalized instrumental variable (GIV) applications experience these difficulties. For examples, see Hansen and Singleton (1982) and Fuhrer et al. (1995) for Euler equations, Alonso-Borrego and Arellano (1999), Blundell and Bond (1998), and Holtz-Eakin et al. (1988) for panel data, and Angrist and Krueger (1991) for natural experiments. Making use of more moment conditions has been found to improve the estimator's accuracy. Hansen, Hausman, and Newey (2008) have found narrower confidence intervals when extending Angrist and Krueger's (1991) study by using 180 instruments rather than the three used in the original. Despite the lure of increased efficiency, GMM is known to have a bias that grows with the number of moment conditions (see Bekker, 1994; Donald et al., 2009; Newey and Smith, 2004). If the dimension of the moment conditions grows too fast relative to the sample size, the GMM estimator performs poorly in simulations and can become inconsistent.[1] It is likely that applied economics research will embrace high-dimensional data with the rise of data availability and quality, and signs of this adoption are already present (see Belloni et al., 2012, 2014, 2016; Gautier and Tsybakov, 2011). For this research to flourish, continued attention must be given to GMM's many-moment bias and inconsistency.

In this article, we propose an estimation and inference strategy that is applicable in scenarios with very many moment conditions. The method we consider employs a combination of jackknife and ridge-regularization. The resulting regularized jackknife GMM (RJGMM) remains valid when the number of moment conditions exceeds the number of observations. The use of jackknife corrects for the weak-moment bias and the use of regularization at each jackknife iteration helps to avoid overfitting in the presence of high-dimensional data by stabilizing the covariance matrix. Furthermore, our model does not require that the signal in the moment conditions be sparse. That is, we do not require that there exists a low-dimensional subset of the moment conditions that captures the signal. Since we do not assume sparsity, it allows us to consider scenarios where the moment conditions are individually weak and, hence, moment selection strategies are void. Coupling jackknife with regularization enables us to extract sufficient valid information from the data without overfitting our estimator on noise. We provide evidence based on Monte Carlo simulations that suggest that the RJGMM has favorable finite-sample properties relative to other GMM estimators that are robust to many weak moment conditions.

Our work contributes to the literature on estimation strategies that are robust to many weak moment

[1]GMM estimators that are valid in high-dimensional settings assume that the number of moment conditions grows at a fraction the sample size (*e.g.*, Bekker, 1994; Hansen et al., 1996; Newey and Windmeijer, 2009).

conditions. While the traditional GMM estimator is biased and inconsistent in this setting, alternative asymptotics that allow the number of moment conditions, m, to grow with, but not exceed, the sample size, n, (Han and Phillips, 2006; Stock and Yogo, 2004), perform well. For example, Han and Phillips (2006) and Newey and Windmeijer (2009, NW hereafter) find that the generalized empirical likelihood (Smith, 1997) and jackknife GMM (JGMM, NW) estimators have smaller biases and are consistent. Hansen, Heaton, and Yaron's (1996) continuous updating estimator (CUE) has been suggested as a solution to GMM's finite-sample bias and has been demonstrated by NW to have favorable asymptotic properties in many weak moment settings. NW show that these estimators perform well in simulations where the number of moment conditions is set to an appreciable fraction of the sample size. Similar findings exist for the linear case (*e.g.,* Chao et al., 2012). However, many-moments asymptotics is limited by the fact that the data must contain more observations than moments. This paper extends existing research by considering cases where the dimension of the moment conditions can be larger than the sample size and, therefore, where regularization is necessary.

RJGMM also contributes to the growing literature on GMM application on high-dimensional data through moment selection. These selection methods either focus on selecting moment conditions based only on their validity or based on both their validity and relevance. For example, Andrews (1999) proposes a selection method to identify the largest set of valid moments based on the J-statistic and the number of moment conditions, Hall and Peixe (2003) and Hall et al. (2007) use canonical correlations and long run canonical correlations, respectively, to eliminate invalid and redundant moments, and Inoue (2006) uses a bootstrap procedure. Caner (2009) and Belloni et al. (2012), among others, apply LASSO in high-dimensional settings. What these methods have in common is that they assume a sparse signal, in the sense that there is a relatively small subset of the available moment conditions that are able to approximate the signal contained in the data about the parameter of interest. This assumption excludes the possibility of having many weak moments that each contributes a small amount to the total signal. As we consider regularization through shrinkage rather than selection, our approach allows for application in settings where sparsity is not given.

Finally, this paper complements the literature on shrinkage-based regularization that addresses the issue of an ill-conditioned covariance matrix relevant to many estimators. Liao (2013) proposes a GMM shrinkage device for the selection of valid moment conditions, Fan and Liao (2011) investigate penalized GMM and penalized empirical likelihood estimation in ultra high-dimensional models where the number of parameters is allowed to grow faster than the sample size, and Cheng and Liao (2015) use penalized GMM to combine the selection of valid and relevant moments with efficient estimation. Shrinkage estimation for homoskedastic linear instrumental variable (IV) models is covered by Chamberlain and Imbens (2004), Okui (2011), and Carrasco and Tchuente (2016). The closest papers to our approach are Carrasco (2012), Carrasco and Tchuente (2016), and Hansen and Kozbur (2014, HK hereafter), whom all allow for $m > n$. Carrasco studies efficient estimation with many IVs by regularization but restricts the behavior of the covariance of the instruments. HK derived the limiting distribution of the regularized jackknife IV estimator (RJIVE) under alternative asymptotics for linear endogenous variable models with dense instruments. In essence, the RJGMM is a generalization of RJIVE where we additionally relax the assumption of homoskedasticity. We compare the performance of these two models in the simulation study and find slight discrepancies. Finally, many shrinkage-based estimation strategies require the modeler to rank or partition the moment conditions prior to estimation (*e.g.*, Kuersteiner, 2002; Okui, 2011). Our approach does not require any prior knowledge concerning the strength or validity of the moment conditions.

In Section 2 a brief literature review focused on the development of the model will be presented. Section 3 presents the estimator. Monte Carlo results are reported in Section 4. Section 5 concludes the report and offers possible avenues for future research. The Appendix holds supplementary simulation results and the code used to run the simulations.

2 A Model For Many Dense Moment Conditions

This section provides an intuitive discussion of the proposed model. The model under consideration is similar to conventional GMM models where the focus is on semi-parametric data with a finite-dimensional parameter space. It differs, however, in two respects. First, unlike conventional GMM research, we allow the number of moment conditions to be large relative to the sample size. Second, unlike other models that can be applied in settings where the set of moment conditions is large, we allow the moment conditions to be dense in the sense that all moments may be valid yet individually weak, rendering common selection-based regularization strategies moot. We, therefore, address the need for regularization with a shrinkage-based strategy.

2.1 The model

For this research, we consider a sequence of models with i.i.d. data for which the GMM moment restriction (equation 1) holds. As the proposed model is an extension of NW's JGMM, we borrow their notation, allowing a direct comparison of the models to be made. To describe the model, let $g(w; \beta)$ be an m-dimensional vector of functions of the data, w, and a p-dimensional parameter vector of interest, β, where

$m \geq p$. Furthermore, let $\{w_i\}_{i=1}^n$ be (i.i.d.) observations of the data vector w. The population moment restriction defines the true parameter vector, β_0,

$$E\left[g(w_i;\beta_0)\right] = 0. \tag{1}$$

The expectation $E\left[\cdot\right]$ is taken with respect to the distribution of w_i for sample size n. The dependence on n is suppressed for notational convenience.

For the ease of exposition, let the following definitions for the moment condition functions and its derivative and variance hold

$$\begin{array}{lllll}
g_i(\beta) = g(w_i;\beta), & g_i = g_i(\beta_0), & \bar{g}(\beta) = E[g_i(\beta)], & \bar{g} = \bar{g}(\beta_0), & \hat{g}(\beta) = \sum_{i=1}^n g_i(\beta)/n, \\
G_i(\beta) = \partial g_i(\beta)/\partial\beta, & G_i = G_i(\beta_0), & G(\beta) = E\left[G_i(\beta)\right], & G = G(\beta_0), & \hat{G}(\beta) = \partial\hat{g}(\beta)/\partial\beta, \\
\Omega_i(\beta) = g_i(\beta)g_i(\beta)', & \Omega_i = \Omega_i(\beta_0), & \Omega(\beta) = E[\Omega_i(\beta)], & \Omega = \Omega(\beta_0), & \hat{\Omega}(\beta) = \sum_{i=1}^n \Omega_i(\beta)/n.
\end{array}$$

Furthermore, let the $-i$ subscript denote the function variation of the without-i^{th} sample. That is, let

$$\hat{g}_{-i}(\beta) = \textstyle\sum_{j\in\{1,\ldots,n\}\setminus i} g_j(\beta)/(n-1), \quad \hat{G}_{-i}(\beta) = \partial\hat{g}_{-i}(\beta)/\partial\beta, \quad \hat{\Omega}_{-i}(\beta) = \sum_{j\in\{1,\ldots,n\}\setminus i} \Omega_j(\beta)/(n-1)$$

be the sample average, sample derivative, and sample variance of the moment conditions for the without-i^{th} sample, respectively.

Finally, we introduce the GIV structural form. This form is relevant for economics research as it covers the form of linear IV data, panel data, and Euler equation data such as Hansen and Singleton's (1982) consumption-based asset pricing model. Using the notation above, let $w_i = (y_i, x_i, Z_i)$ and let the model be described by

$$\begin{aligned} y_i &= f(x_i, \epsilon_i; \beta_0), \\ x_i &= h(Z_i, U_i; \Pi), \end{aligned} \tag{2}$$

where y_i is a scalar dependent variable, x_i is a p-dimensional treatment variable, Z_i is an m-dimensional vector of instruments, U_i and ϵ_i are the first and second-stage error terms, respectively, β_0 is the structural parameter of interest, and Π is an m-dimensional set of first-stage coefficients. $E[U_i\epsilon_i] \neq 0$ causes endogeneity of the treatment variables. Specific implementations of this form will be explored in the simulation study in Section 4. The key feature is that the GMM moment condition is defined by $g_i(\beta) = Z_i\epsilon_i(\beta)$ and the moment restriction takes the form $E[Z_i\epsilon_i(\beta_0)] = 0$.

2.2 GMM with a high-dimensional set of moments conditions

We focus on the case where the number of moment conditions in $g_i(\beta)$ is large. Such cases arise naturally in panel and time series settings where the number of lagged variables grows quickly in the number of time-periods. Contrarily, using more moments for estimation can be employed as a strategy, as suggested by Carrasco and Tchuente (2016), to "boost" the identifying signal for the parameter. Examples of this can be found in Hansen et al. (2008) and Dagenais and Dagenais (1997) who interact and take higher order powers of existing instruments. When the dimension of $g_i(\beta)$ is large, regularization is beneficial to avoid overfitting. In the following, we introduce the problems that are commonly related to high-dimensional settings.

The standard approach to solving GMM – Hansen's (1982) two-step GMM estimator – is consistent under standard asymptotics, however, is known to have poor precision in simulations. While adding more moment conditions increases the asymptotic efficiency, it also increases the small-sample bias. If the number of moment conditions grows fast enough, it can lead to inconsistency of the estimator. To illustrate these problems, consider the estimator that minimizes the two-step objective function

$$\ddot{\beta} = \arg\min_{\beta \in B} \hat{g}(\beta)' \ddot{W} \hat{g}(\beta)/2.$$

Under standard asymptotics, the weighting matrix $\ddot{W} = \hat{\Omega}(\dot{\beta})^{-1}$, for a consistent preliminary estimate, $\dot{\beta}$, minimizes the asymptotic variance of $\ddot{\beta}$.

The many-moment bias of the two-step estimator is explained by the fact that when m is large (and fixed), the objective function is not minimized at truth. NW illustrate this by deconstructing the expectation of the objective function into a signal term and a noise term (the first and second terms of the last equality of equation 3, respectively).

$$\begin{aligned} E\left[\hat{g}(\beta)' W \hat{g}(\beta)/2\right] &= E\left[\sum_{i \neq j} g_i(\beta)' W g_j(\beta) + \sum_{i=1}^{n} g_i(\beta)' W g_i(\beta)\right] /2n^2 \\ &= (1 - n^{-1}) \bar{g}(\beta)' W \bar{g}(\beta)/2 + E\left[g_i(\beta)' W g_i(\beta)\right]/2n \\ &= (1 - n^{-1}) \bar{g}(\beta)' W \bar{g}(\beta)/2 + tr(W\Omega(\beta))/2n \end{aligned} \tag{3}$$

where W is some non-random weighting matrix. Under the assumption of strong moment conditions, that is moment conditions with non-vanishing derivatives, $\bar{g}(\beta) = 0$ uniquely at β_0. Consequently, the signal term is minimized at truth. In general, the noise term does not have a zero derivative at β_0 when $G_i(\beta)$ and $g_i(\beta)$ are correlated. Furthermore, the noise term grows proportionally to m (Newey and

Smith, 2004). The correlation between the "own observation" is the source of GMM's many-moment bias. Under these conditions, the estimator obtains consistency as the signal dominates the noise when the sample size is increased.

Estimation with many weak moment conditions, that is moment conditions where their expected first derivatives vanish, suffers a similar problem.[2] However, asymptotically, consistency of the standard GMM estimator is violated. Unlike with strong moment conditions, when moment conditions are weak, $\bar{g}(\beta)$ is flat and zero in a neighborhood around β_0 and hence the signal does not dominate the noise as the sample size is increased (*c.f.*, Han and Phillips (2006) for a more detailed explanation of these processes).

Because of this risk of overfitting when working with high-dimensional moment conditions, it is helpful to employ regularization. GMM estimators that are valid under many moments use regularization. For example, jackknife-type GMM estimators (*e.g.*, NW's JGMM and Hansen et al.'s (1996) CUE[3]) regularize implicitly. In practice, these estimators rely on the assumption that the sample covariance matrix is of full rank and are therefore only well-defined when $m < n$.[4] Other estimators that make use of selection algorithms (*e.g.*, Cheng and Liao, 2015; Luo and Chernozhukov, 2014; Shi, 2016) allow for m to grow faster than the sample size, n, but require that the dimension of the selected moment conditions, m^*, be an appreciable fraction of n. It is clear that when $m, m^* > n$, consistent estimation of β_0 can only be obtained through the use of additional dimension reduction or regularization techniques.

2.3 Regularizing with dense moments conditions

A large majority of the research conducted on regularizing GMM focuses on selecting an optimal subset of the moment conditions. The goal of selection methods is to select an m^*-dimensional subset of the moment conditions, $g_S(w_i;\beta)$ where $g(w_i;\beta) = [g_S(w_i;\beta)', g_D(w_i;\beta)']'$ and $m^* < m$. These methods satisfy the corresponding moment restriction, $E[g_S(w_i;\beta_0)] = 0$, if $E[g(w_i;\beta_0)] = 0$. The most basic selection strategies are based on intuition or a simple formal mechanism. For GIV data generating processes (DGPs), this includes random selection of instruments or a factor decomposition of the instrument space. In panel and time series settings, where the number of lagged variables naturally grows quickly, authors have had success with factor decomposition. To highlight a few, Forni et al. (2000, 2004) and Stock and Watson (2002) recommend the use of principal component analysis in time series or panel data applications, and Kapetanios and Marcellino (2010) develop a similar strategy for many weak moment settings. However, inclusion or exclusion of the wrong moment conditions leads to inefficiency and, if

[2] Formally, moment conditions are said to be weak when their derivatives vanish at a rate of $O(n^{-1/2})$ (Stock and Wright, 2000).

[3] Donald and Newey (2000) demonstrate that under certain assumptions CUE can be interpreted as a jackknife estimator.

[4] NW assume that covariance matrix' eigenvalues are bounded from below.

too much relevant information contained in the moment conditions is discarded and insufficient signal is maintained, identification is in jeopardy (Dominguez and Lobato, 2004).

Data-driven regularization is a natural countermeasure against the risk of basic selection methods discarding relevant signal. In IV settings this means making use of the information contained in the first-stage dependent variable, x_i, as well as the instrument set, Z_i. However, Shi (2016) points out that we cannot blindly select IVs in the first-stage of non-linear problems as the gradient explicitly depends on β, and hence rudimentary selection procedures are ruled out. Research in data-driven selection algorithms falls into two categories. Following Andrews (1999), moment conditions are selected based on their validity, in the sense that $g_S(w_i;\beta) = \{g_j(w_i;\beta) \mid E[g_j(w_i;\beta)] = 0,\ j = 1, ..., m\}$ are selected and the rest, $g_D(w_i;\beta) = \{g_j(w_i;\beta) \mid E[g_j(w_i;\beta)] \neq 0,\ j = 1, ..., m\}$, are discarded (see, *e.g.*, Andrews and Lu, 2001; Liao, 2013). In contrast, in the vein of Breusch et al. (1999), the set of valid moment conditions can be augmented by those that increase the efficiency of the estimator – relevant moment conditions (see, *e.g.*, Cheng and Liao, 2015; Hall and Peixe, 2003; Kuersteiner, 2002). The formal validity of these methods rely on the existence of a small m^*-dimensional subset of $g(w_i;\beta)$ that can be estimated from the data, and can well-approximate the signal up to a small error.[5] Much more common is that while m is allowed to grow with n, the number of candidate valid moment conditions, m^*, is fixed, allowing these methods to benefit from somewhat standard asymptotics (as in Cheng and Liao, 2015; Hall and Peixe, 2003; Okui, 2011). Intuitively, sparsity demands that the majority of the signal is contained in a small set of the considered moment conditions.

In order to differentiate between sparse and dense signal cases, consider the following argument. Standard many moment asymptotics for GMM, such as those used by Bekker (1994), allow m to grow at a negligible fraction of n.[6] Sparsity generalizes this by allowing m to grow faster than n (*i.e.*, $m/n \to \infty$) but imposes that there exists a subset of the moment conditions, $g_S(w_i;\beta)$ with $\dim(g_S(w_i;\beta)) = m^*$ and $m^* = o(n)$, that can approximate the signal.[7] The intuition behind sparsity can made clearer through the use of the concentration parameter. In IV literature, it is common for many moment asymptotics to focus on the information contained in the instruments through the use of the concentration parameter, which is formally defined on a linear model with one endogenous variable as $\mu_n^2 = n\Pi_n' E[Z_i Z_i']\Pi_n/\sigma_u^2$.[8,9] The value of the concentration parameter measures the information contained in the instruments on the parameter of interest, β_0, where high values correspond to high information. While there is no direct GMM equivalent of the concentration parameter, its intuition applies and GMM researchers often draw

[5] Luo and Chernozhukov (2014) assume that the sparse set of the moment conditions estimates the true model up to an approximation error of $o(\sqrt{log(m \vee n)/n})$.

[6] Formally, $m/n \to \rho$, $\rho \in [0, 1)$ is assumed.

[7] Luo and Chernozhukov (2014) and Shi (2016) allow $m = o(exp(n^{1/3}))$ and $m^* = o((n/logm)^{1/5})$.

[8] The linear model is defined as in equation 2 where $f(x_i, \epsilon_i;\beta) = x_i\beta + \epsilon_i$, $h(Z_i, U_i;\Pi) = Z_i'\Pi + U_i$, and x_i is a scalar. Also, let $E[U_i^2|Z_i] = \sigma_u^2$.

[9] The reader is referred to HK for a more thorough explanation with linear data.

comparisons to benefit understanding. Sparsity allows for $m > n$ but imposes (1) that $\mu_n^2/n \geq C$ for some positive constant C and (2) that the concentration parameter corresponding to the m^*-dimensional subset, $\mu_{S,n}^2$, approximates the m-dimensional set's concentration parameter to a small discrepancy.[10] These two conditions state (1) that the information contained in the instruments does not converge to zero and (2) that the information lost by the discarded variables be low and asymptotically negligible.

This paper targets settings with different data signal requirements. Namely, we are interested in scenarios where any attempt at selecting an optimal subset of the moment conditions will result in a lack of identification due to a weak-moment problem. That is, because we assume that there is no subset of the moment conditions that contains enough information to identify β_0, every $m^* = o(n)$ dimensional subset of the moment conditions, $g_S(w_i;\beta)$, satisfies $\mu_{S,n}^2/n \to 0$. Instead, we build on the work of NW and Han and Phillips (2006), and only require that the signal contained in the complete set of moments is large.[11] By assuming a dense signal, we assume that a combination of the moments holds the full signal and that there is no subset of moments on which a majority of the signal is concentrated. Our approach can be used as an estimation and inference strategy in situations like these.

When the true set of moment conditions strongly identifies the structural parameters and is sparse, the standard asymptotics would apply if the identities of the valid moments were known. Under these circumstances, consistent model selection is possible. Liao (2013) demonstrates that his selection procedure satisfies the oracle property and that efficiency can be improved for strongly identified parameters by adding more valid moments.[12] This means that, given the correct application of a high-dimensional selection model, selection mistakes are unlikely. However, consistent moment selection is not feasible in a model with (very) many weak moments. Caner (2009) demonstrate that under Stock and Wright's (2000) weak moment assumption, LASSO-type GMM is inconsistent. The LASSO is unable to correctly identify the true set of moment conditions without tightening the assumption concerning the strength of the moment conditions. Cheng and Liao (2012) find that including relatively weak moments does not influence their model's ability to correctly identify valid moments, but does influence its ability to identify redundant ones. The source of this conflict stems from the fact that (1) selection causes the convergence of the concentration parameter and (2) that weak moments approximately satisfy the moment restriction in a neighborhood around β_0. Therefore, of the moments with a flat and zero $\bar{g}_j(\beta)$ ($j \in \{1,...,m\}$), those with the largest information content have a higher probability of being selected. In GIV settings, selection mistakes happen when the discarded moments are correlated with the first-stage error (U_i).

[10] Formally, $\mu_n^2 = \mu_{S,n}^2 + O(m^*)$.

[11] Han and Phillips (2006) and NW allow each concentration parameter μ_{jn}^2 ($j = 1,...,k$) to grow and converge with the sample size ($\mu_n = min_{1\leq j\leq p}\mu_{jn} \to \infty$, $\mu_{jn}/\sqrt{n} \to 0$) but require the signal to grow faster than the dimension of the moment conditions (m/μ_n^2 bounded).

[12] The oracle property states that the model performs as well as if the identities of the valid moments were known.

Rather than using selection-based devices, one could also employ shrinkage-based regularization schemes. Shrinkage methods address the issue of unstable covariance matrices that show up in the definitions of many estimators. Kuersteiner (2002, 2012) and Canay (2010) propose kernel weighted GMM estimators that focus on mean squared error (MSE) minimization. Okui (2011) introduces a shrinkage device that shrinks a subset of the weighting matrix. The most similar approach to ours is the work of Carrasco (Carrasco, 2012; Carrasco and Nayihouba, 2017; Carrasco and Noumon, 2011; Carrasco and Tchuente, 2015, 2016) where she (and coauthors) consider different regularization techniques, including ridge-regularization, that directly target the inverse of the covariance matrix. Although these papers are similar to ours in that they employ shrinkage rather than selection devices, they differ by the restrictions that are placed on the data.[13] As HK identify, we are still prone to the difficulties caused by many weak moments, discussed above. Like selection techniques, shrinkage techniques shrink those moments with high information content more than those with low. This can, given the correlation with the first-stage error, result in a similar violation of the moment restriction as above.

To handle the bias introduced by regularization, we follow the work of HK and make use of the jackknife. The JGMM estimator, which takes the generic form

$$\breve{\beta} = \arg\min_{\beta \in B} \sum_{i=1}^{n} \hat{g}_{-i}(\beta)' \breve{W} g_i(\beta)/2n.$$

where $\breve{W}$ is a jackknife GMM weighting matrix, has been shown to be able to deal with many weak moments (*e.g.*, NW or Kézdi et al. 2002). The first-order condition

$$\sum_{i=1}^{n} \hat{G}_{-i}(\breve{\beta})' \breve{W} g_i(\breve{\beta}) = 0$$

demonstrates that $\hat{G}_{-i}(\beta)$ is independent of $g_i(\beta)$ by construction, breaking the correlation responsible for the violation of the exclusion restriction.

3 Ridge-Regularized Jackknife GMM

In this section, we will cover the details of the regularized jackknife GMM. Following the work of HK, we opt for ridge-regularization as a shrinkage device. Since we are primarily interested in settings where signal sparsity is not guaranteed, it is prudent to regularize via shrinkage. Selection methods, including methods that combine selection and shrinkage, result in a violation of the moment restriction. Furthermore, we

[13]Carrasco's research restricts the structure of the covariance operator and Kuersteiner and Okui's work assumes different signal concentration to us.

suspect that the asymptotic theory of RJGMM with ridge-regularization will nicely augment the theory on JGMM, developed by NW, in future research.

3.1 The ridge-regularized jackknife GMM

To reiterate, the two-step GMM is biased when the number of moment conditions is large relative to the sample size. If m grows fast enough, it can cause instability in the covariance matrix, leading to inconsistency of the estimate, $\hat{\Omega}$. Therefore, the main challenge in high-dimensional research is to estimate the optimal weighting matrix, $\hat{\Omega}^{-1}$. Rather than minimizing an objective function, this paper approaches the problem by attempting to create a linear combination of the moment conditions to gain the most information about the parameter. Hansen (1982) shows that a GMM estimator can be obtained from the $p \times m$ selection matrix $A = G'\Omega^{-1}$ by estimating

$$G'\Omega^{-1}\bar{g}(\beta) = 0$$

and shows that this equation is asymptotically equivalent to the two-step's first-order condition. Yet, when m is large, the selection matrix approach offers an easier and more accurate estimation approach than simply estimating the optimal weighting matrix. A feasible version of the two-step GMM via the selection matrix approach is

$$\hat{G}(\dot{\beta})'\hat{\Omega}(\dot{\beta})^{-1}\hat{g}(\ddot{\beta}) = 0$$

for a consistent preliminary estimate, $\dot{\beta}$.

Difficulties arise when the number of moments exceeds the sample size. When $m > n$, the rank of $\hat{\Omega}(\beta)$ is maximally n and, therefore, it does not have a well-defined inverse. We address this by considering the Tikhonov (ridge) regularization as used by Carrasco (2012) and HK. For expository purposes, denote the feasible two-step regularized GMM selection matrix by $\hat{A}^{\Lambda} = \hat{G}(\dot{\beta})'(\hat{\Omega}(\dot{\beta}) + \Lambda'\Lambda)^{-1}$ where Λ is a positive definite ridge penalty matrix. The corresponding estimator solves

$$\hat{G}(\dot{\beta})'(\hat{\Omega}(\dot{\beta}) + \Lambda'\Lambda)^{-1}\hat{g}(\hat{\beta}) = 0.$$

Since $\hat{\Omega}(\beta)$ may be near-singular when the dimension of $g_i(\beta)$ is comparable to the sample size, it behaves poorly more generally than when $m > n$ and $\hat{\Omega}(\beta)$ is singular by design. By adding a positive definite matrix, $\Lambda'\Lambda$, to the positive semi-definite $\hat{\Omega}(\dot{\beta})$, we ensure that the inverse is always well-defined. It is clear that the role of penalty matrix is to stabilize the inverse of the sample covariance matrix. The ridge penalty enables the estimator to avoid overfitting during estimation on a high-dimensional set of moment conditions. Common implementations of ridge-regularization in literature set $\Lambda = \lambda^{1/2} I_m$ where λ is a scalar penalty parameter and I_m is an $m \times m$ identity matrix.[14]

Conducting the same exercise as in Section 2.2 quickly demonstrates that the many-moment bias is still an issue. Furthermore, the presence of many weak moments, as can be expected with a dense signal, causes asymptotic inconsistency. Carrasco (2012), and others who employ shrinkage techniques, restrict the behavior of the covariance operator. Although this assumption is not always restrictive in practice, it rules out some general cases that are favorable in other many weak moment applications, like NW's. Alternatively, we employ the use of jackknife to deal with the correlation between $G_i(\beta)$ and $g_i(\beta)$ present in dense signal settings. We define the selection matrix of the sample that uses all but the i^{th} observation as $\hat{A}^{\Lambda}_{-i} = \hat{G}_{-i}(\dot{\beta}_i)'(\hat{\Omega}_{-i}(\dot{\beta}_i) + \Lambda'\Lambda)^{-1}$. The preliminary estimate, $\dot{\beta}_i$, is generated on the same sample. Again, we are trying to find a linear combination of the without-i^{th} moment conditions that would be most informative about the signal contained in the i^{th} moment conditions about β_0. Note that the covariance matrix, $\hat{\Omega}_{-i}(\beta)$, is a generalization of Phillips and Hale's (1977) JIVE1 rather than the JIVE2 equivalent employed by NW. Repeating this for each observation i, the RJGMM estimator $\tilde{\beta}$ solves

$$\sum_{i=1}^{n} \hat{A}^{\Lambda}_{-i} g_i(\tilde{\beta}) = 0, \text{ where } \hat{A}^{\Lambda}_{-i} = \hat{G}_{-i}(\dot{\beta}_i)'(\hat{\Omega}_{-i}(\dot{\beta}_i) + \Lambda'\Lambda)^{-1},$$

and $\dot{\beta}_i$ is a preliminary consistent estimator for each $i = 1, ..., n$.

4 Simulation Study

We use Monte Carlo simulations to test the finite-sample performance of the RJGMM relative to other common GMM estimators and conduct two types of simulations. In Section 4.1 we investigate the performance of our estimator under various data structures and DGPs. In Section 4.2 we investigate the finite-sample asymptotic properties of our estimator. All simulations are variations of the GIV DGP as outlined in equation 2:

$$y_i = f(x_i, \epsilon_i; \beta_0),$$
$$x_i = h(Z_i, U_i; \Pi),$$

where β_0 is the parameter of interest and the moment restriction takes the form $E[Z_i \epsilon_i(\beta_0)] = 0$. In all of the simulations, we let the treatment variable, x_i, be scalar and the first-stage regression be linear,

[14] We do not investigate the influence of the selection of the penalty term. However, the simulation results suggest that further research is warranted.

$h(Z_i, U_i; \Pi) = Z_i'\Pi + U_i$. The instruments and error terms are normally distributed with mean zero

$$Z_i \sim N(0, \Omega), \quad (\epsilon_i, U_i) \sim N\left(0, \begin{pmatrix} \sigma_\epsilon^2 & \sigma_{\epsilon U} \\ \sigma_{\epsilon U} & \sigma_U^2 \end{pmatrix}\right).$$

The variance of the first-stage and the second-stage errors are set to $\sigma_U^2 = n\Pi'\Omega\Pi/\mu_n^2$ and $\sigma_{\epsilon_i}^2 = 2$, respectively, and the correlation between them is set to $\text{corr}(\epsilon_i, U_i) = 0.6$. The instruments are drawn with a variance $\sigma_{Z_{ij}}^2 = 0.3$ $(j = 1, ..., m)$ and a correlation $\text{corr}(Z_{ij}, Z_{ik}) = 0.5^{|j-k|}$ $(j \neq k, j, k = 1, ..., m)$. The rest of the parameters are set according to the simulation design. The simulations were conducted in Matlab R2017b and the code is provided in Appendix 6.2.

4.1 Fixed sample size simulations

In this section, we present the results for two simulations – a linear and non-linear second-stage relationship. We fix the sample size $n = 100$ and set the number of instruments to $m = 95$ and $m = 190$. Furthermore, as we are explicitly interested in the application of the estimator on dense first-stage relationships, we employ the following differentiation: for the dense setting, we set $\Pi = (-1)^j(\iota_{0.4m}, 0_{0.6m})'/m^{1/2}$ where ι_p is a $1 \times p$ vector of ones, 0_q is a $1 \times q$ vector of zeros, and j indexes the elements of Π $(j = 1, ..., m)$; in the sparse case, we set $\Pi = (-1)^j(\iota_5, 0_{m-5})'/m^{1/2}$, so that only the first five instruments are valid. These first-stage coefficients were chosen to "challenge" the one-step estimator and offer a fair comparison. We consider two variations of the concentration parameter: $\mu^2 = 30$ and $\mu^2 = 150$, where the former provides a weak signal from the set of instruments and the latter a strong signal.

The first design considers a standard linear IV model

$$\begin{aligned} y_i &= x_i\beta_0 + \epsilon_i \\ x_i &= Z_i'\Pi + U_i \end{aligned}$$

for which the corresponding moment restriction $E[Z_i(y_i - x_i\beta_0)] = 0$ holds. The parameter of interest, β_0, is set to one. This simulation design differs from NW in that (1) we explicitly differentiate between dense and sparse settings and (2) we allow the number of moment conditions, m, to be greater than the sample size, n.

In addition to RJGMM, we consider 5 alternative estimators: One-step GMM, two-step GMM, jackknife GMM, CUE, and HK's RJIVE.

$$\begin{aligned}
\hat{\beta}_{1step} &= \arg\min_{\beta\in B} \hat{g}(\beta)' I_m \hat{g}(\beta), \text{ where } I_m \text{ is the } m\times m \text{ identity matrix,}\\
\hat{\beta}_{2step} &= \arg\min_{\beta\in B} \hat{g}(\beta)'\hat{\Omega}(\hat{\beta}_{1step})^{-1}\hat{g}(\beta),\\
\hat{\beta}_{CUE} &= \arg\min_{\beta\in B} \hat{g}(\beta)'\hat{\Omega}(\beta)^{-1}\hat{g}(\beta),\\
\hat{\beta}_{RJIVE} &= \left(\sum_{i=1}^{n} \hat{\Pi}^{\Lambda}_{-i}{}' Z_i x_i'\right)^{-1} \sum_{i=1}^{n} \hat{\Pi}^{\Lambda}_{-i}{}' Z_i y_i, \text{ where } \hat{\Pi}^{\Lambda}_{-i} = (Z_{-i}'Z_{-i} + \Lambda'\Lambda)^{-1} Z_{-i}'X_{-i},^{15}\\
\hat{\beta}_{(R)JGMM} &: \sum_{i=1}^{n} \hat{G}_{-i}(\dot{\beta}_i)'(\hat{\Omega}_{-i}(\dot{\beta}_i) + \Lambda'\Lambda)^{-1} g_i(\hat{\beta}_{(R)JGMM}) = 0, \text{ where } \dot{\beta}_i = \arg\min_{\beta\in B} \hat{g}_{-i}(\beta)' I_m \hat{g}_{-i}(\beta).
\end{aligned}$$

The one-step and two-step GMM weighting matrices are I_m and $\hat{\Omega}(\hat{\beta}_{1step})^{-1}$, respectively. JGMM is computed as RJGMM with a penalty of zero ($\Lambda = 0_{m\times m}$). CUE optimizes the weighting matrix during estimation and is consequently less stable. We, therefore, initialize the solver using a grid of initial β values, $\{0.1, 0.3, 0.7, 0.9\}$, and record the estimator that minimizes the CUE objective function.[16] RJIVE is solved using a closed form solution. RJGMM and RJIVE's penalty matrices are set to $\Lambda = \lambda I_m$, where $\lambda = (var(x)m/n)^{1/2}$ and $var(x)$ is the in-sample variance of the treatment variable, which is calculated every iteration.[17] The preliminary $\dot{\beta}_i$, which is used to build JGMM and RJGMM's selection matrices, is estimated by running one-step GMM on the without-i^{th} sample.[18] Both CUE and JGMM are valid under many weak moment conditions ($m < n$) and, therefore, provide a good benchmark against which to compare RJGMM. Since the two-step GMM, CUE, and JGMM are ill-defined when $m > n$, we randomly select 95 moment conditions to train the estimators. In every iteration, the same subset is used to train all three models. Since the two-step and one-step estimators are the standard GMM models, they each provide a natural benchmark for both the $m < n$ and $m \geq n$ cases, respectively. Finally, we also estimate an oracle version of the RJGMM estimator (RJGMM$\star$), in which the preliminary estimate used to estimate the selection matrix, is set to β_0.

The results are based on 1'500 simulation iterations. We report the median bias (Med. Bias), interquartile range (IQR), and MSE. In each simulation, we center the data to ensure that each variable's in-sample mean is zero before running the various models. The results for the simulation using linear data with $m = 95$ are presented in Table 1. Panel A and B show the results for concentration parameters $\mu^2 = 30$ and $\mu^2 = 150$, respectively. The results for the one-step and two-step GMM are as expected. Both estimators are dominated by bias that increases with signal sparsity but have a low variance relative

[15] X_{-i} is an $(n-1)$-dimensional vector of stacked x_i values and Z_{-i} is an $(n-1)\times m$ dimensional matrix of stacked Z_i values with the i^{th} terms being omitted from each.

[16] We use the `fminunc(·)` solver in Matlab R2017b.

[17] The order of RJIVE's covariance matrix, $Z_{-i}'Z_{-i}$, requires $\lambda = (var(x)m)^{1/2}$.

[18] The simulation results suggest that further research into an estimation strategy that yields an optimal $\dot{\beta}_i$ could improve the small-sample performance of the RJGMM.

to the other estimators. CUE outperforms the two traditional GMM estimators in all settings relative to the median bias. JGMM fares worst, as it is unable to handle the high dimensionality of the moment conditions. In all of the settings, RJGMM and RJIVE offer bias improvements relative to the one-step and two-step estimators, however, this comes at the cost of higher variance. Their high MSE scores are due to few large outliers. As expected, the regularized estimators offer superior performance relative to the other estimators in the dense setting with $\mu^2 = 150$. Note that it is likely that the discrepancies between RJGMM and RJIVE in terms of RJIVE's superior IQR, comes from the difference in homoskedasticity assumptions. In most settings, the oracle version of the RJGMM has a slightly lower bias and variance.

The results for the simulation using linear data with $m = 190$ are presented in Table 2. Panel A and B show the results for concentration parameters $\mu^2 = 30$ and $\mu^2 = 150$, respectively. The rough trends align with the previous results. The standard GMM estimators suffer from higher biases than their $m < n$ counterparts. JGMM remains unstable and performs worst. CUE's performance is relatively stable across all settings. Again, the regularized estimators demonstrate their strength in the dense settings. The oracle version of the RJGMM has a lower bias than both the RJGMM and RJIVE in all settings, however, the RJIVE is superior regarding the IQR. Finally, although the performance of all of the estimators is dominated by bias in the sparse settings, the regularized estimators offer a clear bias reduction.

The second simulation design considers a non-linear second-stage relationship of the exponential form

$$y_i = exp(x_i\beta_0) + \epsilon_i$$
$$x_i = Z_i'\Pi + U_i$$

for which the corresponding moment restriction $E[Z_i(y_i - exp(x_i\beta_0))] = 0$ holds. The parameter of interest, β_0, is set to one. This structural form is similar to Stock and Wright's (2000) capital asset pricing model (Hausman et al., 2011). We use the same set of comparison models as in the previous simulation with the exception of the RJIVE. The results are based on 1'500 simulation iterations and we report the median bias, IQR, and MSE. In each simulation we center Z_i, x_i, ϵ_i, and U_i such that they have an in-sample mean of zero and, consequently, the dependent variable, y_i, has an in-sample mean of one.

The results for the simulation using $m = 95$ are presented in Table 3 and $m = 190$ are presented in Table 4. Panels A and B present the results for $\mu^2 = 30$ and $\mu^2 = 150$, respectively. The results of this simulation design reflect the findings of the linear case. In all of the setups, the one-step and two-step estimators are dominated by bias and benefit from a relatively low variance. CUE's performance is stable across the trials. JGMM's performance is also comparatively strong in the sparse weak signal cases with

$m = 95$. In the dense signal simulations, RJGMM outperforms CUE with a slightly higher variance. When increasing the number of moment conditions to $m = 190$, RJGMM has a lower bias and variance relative to its $m = 95$ counterparts. In the majority of the settings, except $m = 190$ with a sparse strong signal, the RJGMM outperforms the oracle version.

The results of the simulations suggest that the RJGMM estimator complements existing approaches. RJGMM has a lower median bias than the comparison estimators and has superior performance in the dense signal settings with $\mu^2 = 150$ (for both $m < n$ and $m > n$). Given the theoretical restrictions, RJGMM could also offer an effective estimation strategy in linear settings with heteroskedasticity relative to the RJIVE, however, additional research is necessary to validate this claim. The results that CUE outperforms JGMM support the conclusions drawn by NW. Our results also agree with Carrasco and Tchuente (2016), in that the ridge-regularization leads to favorable performance even when the signal is weak. Finally, the discrepancy between the RJGMM and the oracle version highlights the difficulty inherent with any two-step GMM method where a preliminary consistent estimate is needed for estimation.

4.2 Finite-sample asymptotic simulations

The final simulation considers a second non-linear design

$$
\begin{aligned}
y_i &= \left((x_i\beta_0)^2\right)^{exp\{\gamma_0\}/2} + \epsilon_i \\
x_i &= Z_i'\Pi + U_i
\end{aligned}
$$

for which the corresponding moment restriction $E[Z_i(y_i - ((x_i\beta_0)^2)^{exp\{\gamma_0\}/2})] = 0$ holds. The parameters of interest are set to $(\beta_0, \gamma_0) = (1, 0.5)$. The structure of the second-stage equation enforces that the support of the parameters is real ($supp(\{\beta, \gamma\}) \in \mathbb{R}^2$). We set the signal concentration to be dense in the sense that $\Pi = (-1)^j(\iota_{0.4m}, 0_{0.6m})'/m^{1/2}$. Finally, we let the sample size and concentration parameter grow: $n = \{100, 200, 300, 400, 500\}$ and $\mu_n^2 = 1.5n$, respectively; and investigate three scenarios of m: fixed ($m = 50$), growing but less than the sample size ($m = 0.95n$), and growing and larger than the sample size ($m = 1.9n$). In the third case, $m/2$ moment conditions are selected at random each iteration for the two-step estimator, CUE, and JGMM. The specifications of the other DGP parameters are adopted from the previous simulation's dense setting. To compare the performance we use the same estimators as the previous non-linear simulation (one-step GMM, two-step GMM, CUE, JGMM, and oracle RJGMM). Again, we center the data in each simulation to ensure that each variable's in-sample mean is zero.[19]

The results are based on 1'500 simulation iterations. The results for the $m = 50$, $m = 0.95n$, and

[19]We include results for more simulation where $\Pi = m^{-1/2}(\iota_{0.4m}, 0_{0.6m})'$ in Appendix 6.1.2. These results are included for the reader's interest and are only briefly referenced in the conclusion of this section.

$m = 1.9n$ cases are presented in Figures 1, 2, and 3, respectively. The corresponding tables are included in Appendix 6.1.1, in Tables A1, A2, and A3, for the reader's interest. The first, second, and third panels (rows) of the figures present the results for median bias, IQR, and MSE, respectively. The columns, from left to right, present the results for parameter β, parameter γ, and the mean, respectively. Note that the IQR for Figure 1 and MSE for all figures are presented in log scale.

We begin with Figure 1 ($m = 50$; Table A1). The standard one-step and two-step estimators have stable performance across all sample sizes. The bias for both parameters is positive and constant, and the variance is low. CUE, JGMM, and RJGMM seem to balance their biases across the parameters. They balance the estimation error by underestimating β ($\hat{\beta} \approx 0$) and overestimating γ. As n grows, JGMM and RJGMM are able to break free from this tendency and estimate β and γ closer to the truth. As n increases, RJGMM has the lowest absolute median bias for the β parameter. The oracle version of RJGMM has the lowest bias and is close to median unbiased on average. Regarding the variance of the estimators, they all suffer from large outliers that inflate their MSE values. However, apart from CUE, most simulation iterations yield estimates in a narrow interval. Both the JGMM and RJGMM's MSE values decrease with sample size. The results of JGMM and its regularized version are very similar, suggesting a possible misspecification of the penalty parameter.

Next, Figure 2 (Table A2) presents the results for the simulations where we allow the number of moment conditions to grow at $m = 0.95n$. We see a similar trend in the biases of the estimators as in Figure 1. One-step, two-step, JGMM, and RJGMM all have constant biased across n, where the one-step and two-steps biases are positive. The others are positive for β and negative for γ. Notice again, that the average of RJGMM's biases is close to zero and that the oracle version is mean median unbiased. The most interesting difference is CUE's improvement in bias, however, regarding the MSE, it has similar problems as in Figure 1. The oracle version of the RJGMM seems to have a bias that converges to zero. The IQR values of the estimators are all in a similar range. The IQR of RJGMM converges towards the oracle version's IQR. As expected, the RJGMM has a higher variance than the traditional GMM estimators and, with JGMM, has MSE values that decrease with sample size.

The results for the third case, $m = 1.9n$, are presented in Figure 3 (Table A3). One-step's and two-step's β biases are similar to the previous case and have slightly higher γ biases. CUE's bias is also slightly higher than the previous simulation since it is not estimated on the full signal. Again, CUE, JGMM, and RJGMM underestimate β and over estimate γ, yet they are all close to median unbiased on average. RJGMM$\star$ is close to median unbiased for both parameters. The IQR values are very similar to the IQR results from Figure 2. The variance of RJGMM and RJGMM$\star$ are stable over n and slightly higher than the one-step and two-step estimators. Again, the RJGMM and RJGMM$\star$'s IQR values converge. Finally, we see a quick convergence of RJGMM's MSE for the β parameter over n.

The simulation results confirm the findings from Section 4.1. Overall, we are able to see that the RJGMM estimator has advantages relative to the benchmark estimators. The findings demonstrate that the RJGMM estimator is on average median unbiased, however, difficulties can occur during estimation.[20] In high-dimensional settings the estimator has nice convergence properties of the MSE. The results highlight the need for an estimation strategy for the preliminary estimate in order to be able to benefit from the oracle's performance. Finally, the findings suggest that further investigation into the influence of the penalty term, λ, is warranted as the RJGMM results do not differ as substantially from the JGMM results, as they do in Section 4.1.

We conclude this section with a brief discussion of the supplementary simulation results included in Appendix 6.1.2. These results provide additional information about the properties of our estimator under simulation conditions more conventional to literature. Figures A1, A2, and A3, which present the results corresponding to Figures 1, 2, and 3 with $\Pi = m^{-1/2}(\iota_{0.4m}, 0_{0.6m})'$, show improved performance of the RJGMM regarding the bias. The RJGMM is median unbiased relative to the γ parameter and has a lower biased relative to β. The MSE converges quickly when m is allowed to grow with n. It is also noteworthy that under this design, the RJGMM and its oracle version display similar results. Figure A4 presents the results for linear data corresponding to the simulation in Section 4.1 with $\Pi = m^{-1/2}(\iota_{0.4m}, 0_{0.6m})'$. These results show that the RJGMM is median unbiased and has IQR values that are slightly larger than the two-step's values, but lower than CUE's and JGMM's. The MSE values have clear convergence rates with growing n. These simulation results suggest that further research into RJGMM is warranted.

5 Conclusion

In this paper, we provide an estimator for settings with high-dimensional moment conditions where the signal is dispersed across the full set of moments. Finite-sample simulation studies provide evidence that this estimator has favorable properties for estimation under these assumptions. In dense settings, the performance of the RJGMM is comparable to that of HK's RJIVE. Further research is necessary to determine the relative strengths and weaknesses of these two estimators. The simulations display that RJGMM occasionally has difficulty balancing the parameter, however, as sample size increases, this problem tends to dampen. It is clear that further research into the properties of RJGMM is necessary to determine its area and extent of applicability. Still, under conventional simulation designs used in

[20] Upon investigation of the results, it was noticed that the RJGMM "quasi-objective function" (*c.f.*, Section 6.2.2, Line 108) was not convex. A similar approach to CUEs', where different initialization values for the solver were attempted, might have yielded better results.

literature, RJGMM provides a simple and effective alternative for estimation and inference in high-dimensional settings.

We suggest three avenues of further research. The next logical step for future research is the development of RJGMM's asymptotic theory. The simulation results suggest that, under assumptions similar to NW's, RJGMM could be shown to be asymptotically efficient relative to CUE. A more extensive comparison of the estimator to HK's RJIVE would also be facilitated. Second, Carrasco and Nayihouba's (2017) findings suggest that (1) research into data-driven selection of the penalty term would extend our understanding of the estimator and (2) research into additional regularization techniques could offer novel estimation strategies relevant for high-dimensional settings. Finally, this paper can be extended by applying RJGMM on time series or panel data and could help address the poor finite-sample problems common to GMM in these scenarios (*e.g.,* Blundell and Bond, 1998).

Hereby I declare that the presented paper is all my own work and was produced without the use of other than the stated resources. All passages that are directly or indirectly taken from published or unpublished texts are marked as such. This paper has not been presented in the same or a similar form or in extracts in the context of another exam.

Place, date	Signature

Table 1: Linear data simulation results for $m = 95$ moment conditions

	Dense Signal			Sparse Signal		
	Med. Bias	IQR	MSE	Med. Bias	IQR	MSE
	A. Concentration parameter $\mu^2 = 30$					
One-step	1.614	0.601	2.788	3.786	1.466	15.500
Two-step	1.625	0.607	2.796	3.782	1.439	15.609
CUE	-0.280	1.143	23.951	-0.264	1.200	45.679
JGMM	1.983	5.823	20483.086	4.051	14.276	1004363.620
RJGMM	1.027	3.469	73152.290	2.076	8.389	81294.420
RJIVE	0.845	2.991	49750.353	1.708	5.992	10922.125
RJGMM$\star$	0.761	3.113	831.668	1.675	7.812	8092.460
	B. Concentration parameter $\mu^2 = 150$					
One-step	1.648	1.047	3.201	3.623	2.447	16.025
Two-step	1.674	1.030	3.238	3.676	2.381	16.348
CUE	-0.300	1.102	57.531	-0.283	1.334	1075.086
JGMM	2.570	10.243	35132.867	4.611	25.108	709367.048
RJGMM	0.097	3.024	2624.334	0.165	7.570	58132.222
RJIVE	0.105	1.962	19.148	0.390	4.436	21.215
RJGMM$\star$	0.139	2.631	1754.384	0.362	7.203	2645.827

Notes: $n = 100$; $\beta_0 = 1$; 1'500 replications; $\text{corr}(\epsilon_i, U_i) = 0.6$. I report the median bias (Med. Bias), interquartile range (IQR), and mean squared error (MSE). The oracle version of the RJGMM is denoted by RJGMM$\star$.

Table 2: Linear data simulation results for $m = 190$ moment conditions

	Dense Signal			Sparse Signal		
	Med. Bias	IQR	MSE	Med. Bias	IQR	MSE
	A. Concentration parameter $\mu^2 = 30$					
One-step	1.699	0.508	3.026	5.790	1.821	34.897
Two-step	1.701	0.595	3.067	5.749	2.025	35.652
CUE	-0.310	1.092	26.922	-0.307	1.342	164001.852
JGMM	1.631	5.985	5843.182	6.093	19.885	90798.658
RJGMM	1.473	3.902	905.112	4.616	13.677	25109.879
RJIVE	1.393	3.392	4865.027	4.238	7.155	2409.946
RJGMM$\star$	1.232	3.836	6170.370	2.870	15.788	22797.557
	B. Concentration parameter $\mu^2 = 150$					
One-step	1.861	0.886	3.830	5.958	3.129	40.961
Two-step	1.856	0.996	3.935	6.056	3.444	43.365
CUE	-0.310	0.952	225.883	-0.322	1.501	85.333
JGMM	2.006	9.931	112224.210	7.877	34.932	401862.375
RJGMM	0.857	4.567	16188.399	2.250	16.700	5901.006
RJIVE	0.627	3.284	7491.500	3.556	5.400	36.087
RJGMM$\star$	0.680	4.343	1641.062	1.230	15.089	285443.600

Notes: $n = 100$; $\beta_0 = 1$; 1'500 replications; $\text{corr}(\epsilon_i, U_i) = 0.6$. I report the median bias (Med. Bias), interquartile range (IQR), and mean squared error (MSE). The oracle version of the RJGMM is denoted by RJGMM$\star$. For two-step GMM, CUE, and JGMM, 95 moment conditions were selected randomly in each iteration. The same moment conditions were selected for each estimator each iteration. The preliminary estimate of β_0 for the two-step estimator was generated on the reduced set of moment conditions.

Table 3: Exponential data simulation results for $m = 95$ moment conditions

	Dense Signal			Sparse Signal		
	Med. Bias	IQR	MSE	Med. Bias	IQR	MSE
	A. Concentration parameter $\mu^2 = 30$					
One-step	0.665	0.273	0.501	2.329	0.840	5.772
Two-step	0.668	0.266	0.504	2.334	0.816	5.794
CUE	-0.296	1.107	384.401	-0.279	1.216	2101.349
JGMM	0.190	1.343	1.411	1.000	3.513	11.454
RJGMM	0.244	1.376	1.214	0.695	3.532	10.138
RJGMM$\star$	0.405	1.518	1.590	1.238	4.990	15.089
	B. Concentration parameter $\mu^2 = 150$					
One-step	1.021	0.583	1.170	2.813	1.755	9.157
Two-step	1.016	0.553	1.181	2.842	1.694	9.333
CUE	-0.300	1.169	768.241	-0.294	1.345	4148.592
JGMM	0.346	2.390	4.624	1.142	6.307	32.276
RJGMM	0.043	2.924	4.558	0.182	6.922	27.170
RJGMM$\star$	0.150	2.828	4.569	0.215	6.701	26.125

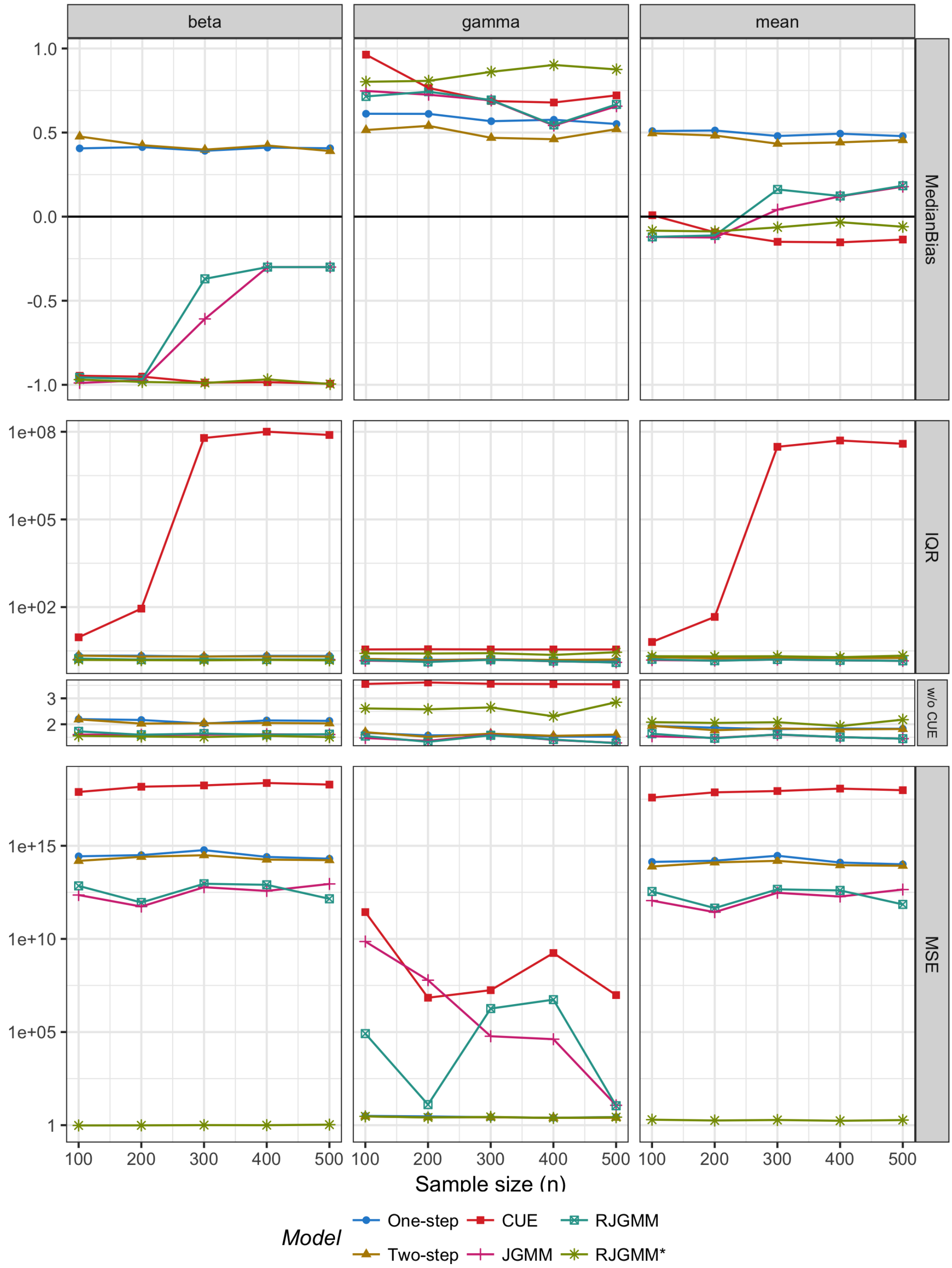
Notes: $n = 100$; $\beta_0 = 1$; 1'500 replications; $\text{corr}(\epsilon_i, U_i) = 0.6$. I report the median bias (Med. Bias), interquartile range (IQR), and mean squared error (MSE). The oracle version of the RJGMM is denoted by RJGMM$\star$.

Table 4: Exponential data simulation results for $m = 190$ moment conditions

	Dense Signal			Sparse Signal		
	Med. Bias	IQR	MSE	Med. Bias	IQR	MSE
	A. Concentration parameter $\mu^2 = 30$					
One-step	0.697	0.265	0.537	3.654	1.030	13.941
Two-step	0.698	0.285	0.534	3.654	1.132	13.945
CUE	-0.311	1.021	298.289	-0.297	1.417	2376.256
JGMM	0.200	1.394	1.435	1.441	4.992	22.051
RJGMM	0.038	1.096	0.657	0.478	2.758	6.819
RJGMM$\star$	0.145	1.279	0.937	0.947	5.851	22.289
	B. Concentration parameter $\mu^2 = 150$					
One-step	1.108	0.515	1.336	4.617	2.057	23.313
Two-step	1.095	0.557	1.345	4.719	2.189	23.921
CUE	-0.291	0.927	357.586	-0.314	1.517	1091.132
JGMM	0.337	2.422	4.955	1.985	9.140	69.157
RJGMM	0.069	1.742	2.227	0.558	5.297	26.052
RJGMM$\star$	0.253	2.371	3.936	0.442	8.761	46.943

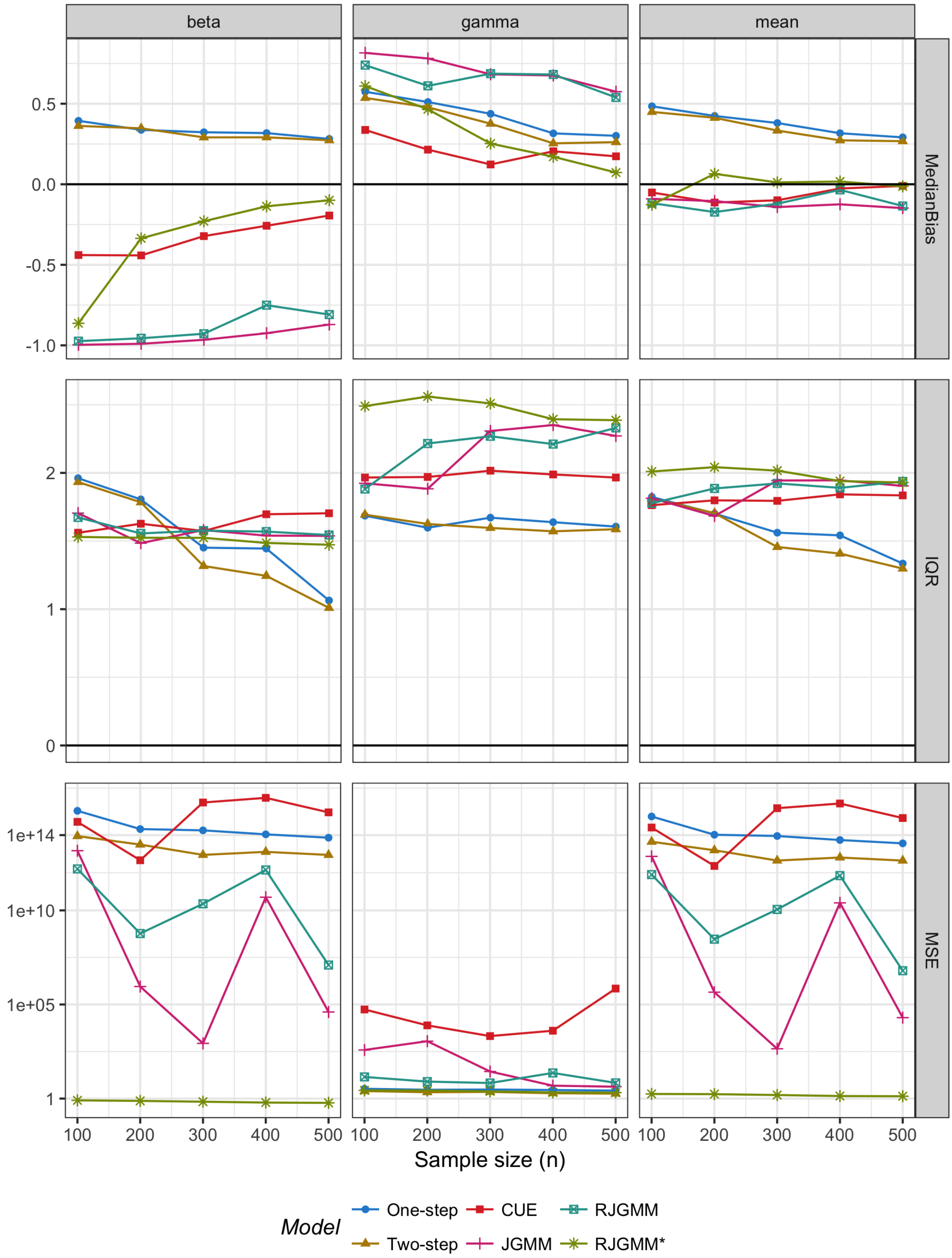
Notes: $n = 100$; $\beta_0 = 1$; 1'500 replications; $\text{corr}(\epsilon_i, U_i) = 0.6$. I report the median bias (Med. Bias), interquartile range (IQR), and mean squared error (MSE). The oracle version of the RJGMM is denoted by RJGMM$\star$. For two-step GMM, CUE, and JGMM, 95 moment conditions were selected randomly in each iteration. The same moment conditions were selected for each estimator each iteration. The preliminary estimate of β_0 for the two-step estimator was generated on the reduced set of moment conditions.

Figure 1: Non-linear data simulation results for $m = 50$ moment conditions



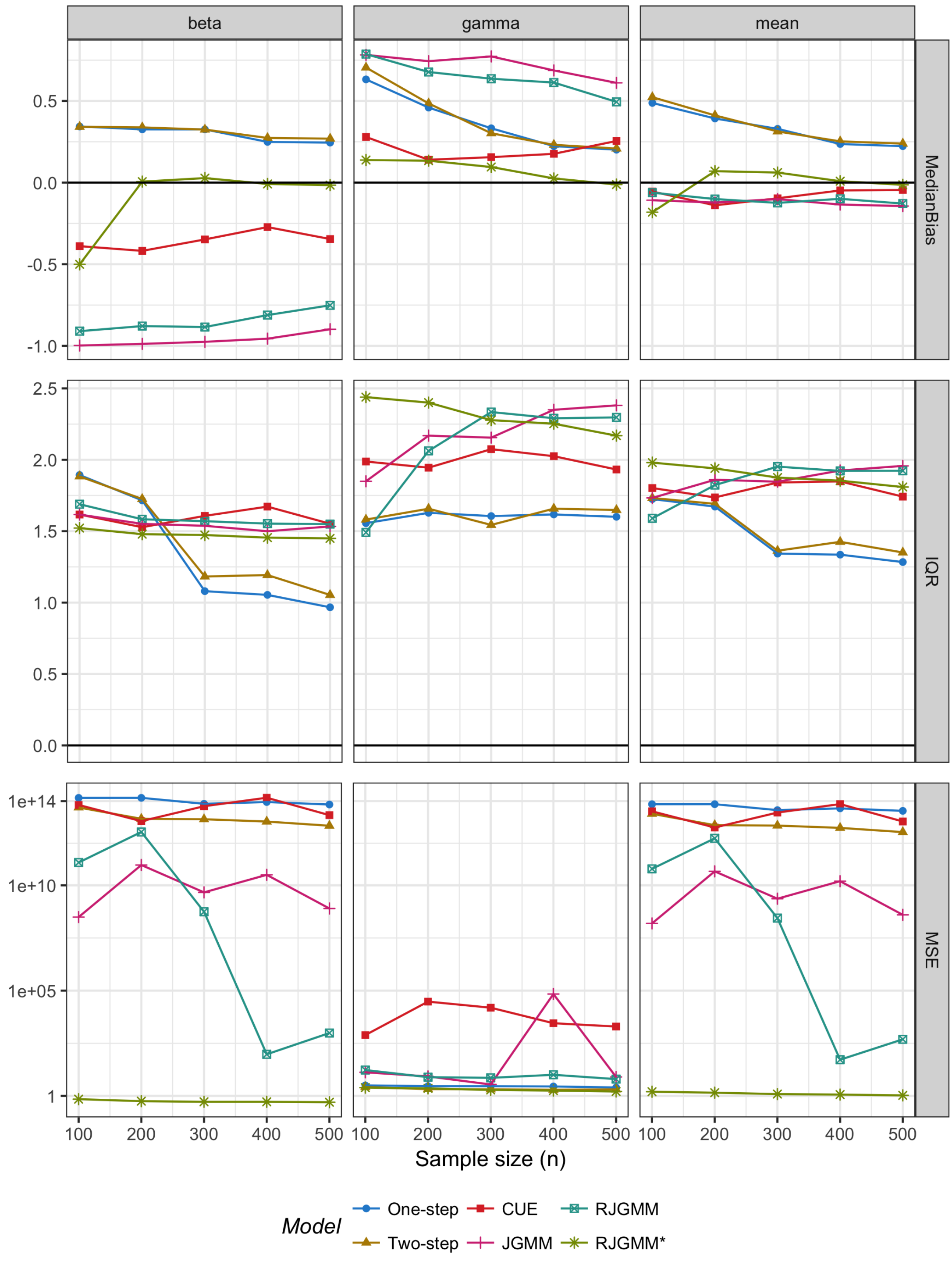
Notes: 1'500 replications; $(\beta_0, \gamma_0) = (1, 0.5)$; $\text{corr}(\epsilon_i, U_i) = 0.6$; Dense signal; $\mu_n^2 = 1.5n$. The three columns from right to left present the results for the β parameter, γ parameter, and the average results. The three rows from top to bottom present the median bias, IQR, and MSE. The oracle version of the RJGMM is denoted by RJGMM$\star$. The second level of the second panel (IQR) replots the IQR for the set of models excluding CUE in order to better distinguish the lines. The preliminary estimate of (β_0, γ_0) for the two-step estimator was generated on the reduced set of moment conditions.

Figure 2: Non-linear data simulation results for $m = 0.95n$ moment conditions



Notes: 1'500 replications; $(\beta_0, \gamma_0) = (1, 0.5)$; $\text{corr}(\epsilon_i, U_i) = 0.6$; Dense signal; $\mu_n^2 = 1.5n$. The three columns from right to left present the results for the β parameter, γ parameter, and the average results. The three rows from top to bottom present the median bias, IQR, and MSE. The oracle version of the RJGMM is denoted by RJGMM$\star$. The preliminary estimate of (β_0, γ_0) for the two-step estimator was generated on the reduced set of moment conditions.

Figure 3: Non-linear data simulation results for $m = 1.9n$ moment conditions



Notes: 1'500 replications; $(\beta_0, \gamma_0) = (1, 0.5)$; $\text{corr}(\epsilon_i, U_i) = 0.6$; Dense signal; $\mu_n^2 = 1.5n$. The three columns from right to left present the results for the β parameter, γ parameter, and the average results. The three rows from top to bottom present the median bias, IQR, and MSE. For the two-step GMM, CUE, and JGMM, $m/2$ moment conditions were selected randomly in each iteration. The same moment conditions were selected for each estimator, each iteration. The oracle version of the RJGMM is denoted by RJGMM$\star$. The preliminary estimate of (β_0, γ_0) for the two-step estimator was generated on the reduced set of moment conditions.

6 Appendix

6.1 Supplementary simulation results

6.1.1 Simulation results for Section 4.2

We provide the results used to create Figures 1, 2, and 3 in Tables A1, A2, and A3, respectively. Information concerning how the results were generated can be found in Section 4.2.

Table A1: Non-linear data simulation results for $m = 50$ moment conditions

		β					γ				
	n	100	200	300	400	500	100	200	300	400	500
One-step	MB	0.406	0.414	0.392	0.411	0.407	0.612	0.611	0.568	0.576	0.551
	IQR	2.200	2.166	2.029	2.149	2.132	1.670	1.571	1.592	1.524	1.524
	MSE	2.721e+14	3.159e+14	5.859e+14	2.530e+14	2.025e+14	3.155	2.967	2.665	2.492	2.7234
Two-step	MB	0.478	0.425	0.398	0.423	0.390	0.514	0.540	0.469	0.460	0.520
	IQR	2.189	2.027	2.039	2.050	2.036	1.694	1.516	1.639	1.555	1.601
	MSE	1.54e+14	2.55e+14	3.11e+14	1.81e+14	1.69e+14	2.959	2.638	2.665	2.464	2.494
CUE	MB	-0.946	-0.952	-0.987	-0.984	-0.994	0.963	0.765	0.688	0.679	0.721
	IQR	9.246	88.614	6.12e+07	1.01e+08	7.76e+07	3.555	3.611	3.561	3.550	3.541
	MSE	7.75e+17	1.48e+18	1.74e+18	2.35e+18	1.93e+18	2.75e+11	6.95e+06	1.77e+07	1.75e+09	9.64e+06
JGMM	MB	-0.989	-0.973	-0.608	-0.30	-0.30	0.748	0.725	0.691	0.542	0.657
	IQR	1.605	1.593	1.605	1.614	1.606	1.471	1.361	1.594	1.398	1.282
	MSE	2.27e+12	5.46e+11	5.97e+12	3.77e+12	8.96e+12	7.24e+09	6.08e+07	59467.487	41328.836	11.662
RJGMM	MB	-0.956	-0.966	-0.37	-0.30	-0.30	0.715	0.743	0.693	0.545	0.667
	IQR	1.728	1.597	1.644	1.601	1.619	1.545	1.323	1.576	1.412	1.271
	MSE	7.03e+12	8.99e+11	9.14e+12	8.04e+12	1.42e+12	82351.843	12.72	1.82e+06	5.49e+06	11.056
RJGMM⋆	MB	-0.969	-0.983	-0.989	-0.968	-0.995	0.802	0.807	0.861	0.902	0.875
	IQR	1.554	1.528	1.509	1.553	1.506	2.611	2.577	2.651	2.308	2.849
	MSE	0.967	0.978	1.010	10000	1.067	2.970	2.587	2.787	2.447	2.692

Notes: 1'500 replications; $(\beta_0, \gamma_0) = (1, 0.5)$; $\text{corr}(\epsilon_i, U_i) = 0.6$; Dense signal; $\mu_n^2 = 1.5n$. I report the median bias (MB), interquartile range (IQR), and mean squared error (MSE). The preliminary estimate of (β_0, γ_0) for the two-step estimator was generated on the reduced set of moment conditions.

Table A2: Non-linear data simulation results for $m = 0.95n$ moment conditions

		β					γ				
	n	100	200	300	400	500	100	200	300	400	500
One-step	MB	0.394	0.338	0.323	0.318	0.282	0.574	0.511	0.437	0.316	0.301
	IQR	1.961	1.806	1.451	1.445	1.064	1.686	1.598	1.672	1.638	1.606
	MSE	1.95e+15	2.10e+14	1.79e+14	1.10e+14	7.30e+13	3.322	2.940	3.029	2.849	2.659
Two-step	MB	0.362	0.347	0.291	0.292	0.273	0.536	0.478	0.376	0.254	0.262
	IQR	1.933	1.784	1.317	1.244	1.009	1.693	1.625	1.596	1.571	1.587
	MSE	8.86e+13	3.14e+13	8.89e+12	1.29e+13	8.86e+12	2.507	2.176	2.264	1.928	1.834
CUE	MB	-0.44	-0.442	-0.322	-0.258	-0.194	0.337	0.215	0.123	0.205	0.174
	IQR	1.561	1.627	1.573	1.696	1.704	1.966	1.970	2.016	1.988	1.966
	MSE	5.07e+14	4.61e+12	5.40e+15	9.57e+15	1.62e+15	54269.61	7768.615	2093.955	4033.895	701574.338
JGMM	MB	-0.997	-0.99	-0.966	-0.925	-0.871	0.816	0.781	0.683	0.676	0.575
	IQR	1.705	1.484	1.581	1.540	1.538	1.924	1.883	2.308	2.351	2.271
	MSE	1.48e+13	899098.575	859.914	5.02e+10	39908.634	382.855	1119.88	27.486	4.854	4.273
RJGMM	MB	-0.974	-0.956	-0.928	-0.751	-0.809	0.740	0.611	0.687	0.682	0.539
	IQR	1.673	1.556	1.577	1.569	1.544	1.882	2.216	2.269	2.211	2.330
	MSE	1.60e+12	5.90e+08	2.25e+10	1.41e+12	1.25e+07	14.149	7.997	6.720	23.058	6.827
RJGMM$\star$	MB	-0.863	-0.336	-0.23	-0.137	-0.099	0.610	0.465	0.253	0.171	0.073
	IQR	1.530	1.524	1.524	1.486	1.473	2.490	2.560	2.510	2.394	2.387
	MSE	0.805	0.753	0.681	0.615	0.590	2.717	2.685	2.410	2.116	2.072

Notes: 1'500 replications; $(\beta_0, \gamma_0) = (1, 0.5)$; $\text{corr}(\epsilon_i, U_i) = 0.6$; Dense signal; $\mu_n^2 = 1.5n$. I report the median bias (MB), interquartile range (IQR), and mean squared error (MSE). The preliminary estimate of (β_0, γ_0) for the two-step estimator was generated on the reduced set of moment conditions.

Table A3: Non-linear data simulation results for $m = 1.9n$ moment conditions

		β					γ				
	n	100	200	300	400	500	100	200	300	400	500
One-step	MB	0.344	0.326	0.326	0.249	0.245	0.632	0.460	0.333	0.223	0.202
	IQR	1.894	1.716	1.080	1.054	0.967	1.556	1.629	1.606	1.617	1.601
	MSE	1.43e+14	1.43e+14	7.48e+13	9.06e+13	6.91e+13	3.170	2.912	2.900	2.815	2.517
Two-step	MB	0.342	0.338	0.325	0.274	0.269	0.705	0.485	0.302	0.231	0.209
	IQR	1.884	1.726	1.183	1.193	1.054	1.581	1.658	1.544	1.658	1.649
	MSE	5.01e+13	1.45e+13	1.38e+13	1.07e+13	6.78e+12	2.638	2.101	2.051	1.966	2.035
CUE	MB	-0.389	-0.418	-0.348	-0.272	-0.346	0.280	0.140	0.156	0.176	0.255
	IQR	1.617	1.528	1.607	1.672	1.552	1.988	1.945	2.075	2.025	1.932
	MSE	6.58e+13	1.12e+13	5.72e+13	1.45e+14	2.16e+13	776.096	30148.999	15488.134	2837.704	1974.945
JGMM	MB	-0.998	-0.988	-0.975	-0.956	-0.898	0.782	0.744	0.773	0.687	0.611
	IQR	1.617	1.552	1.538	1.500	1.533	1.850	2.169	2.155	2.350	2.381
	MSE	3.12e+08	9.25e+10	4.64e+09	3.16e+10	7.96e+08	13.264	8.425	3.496	69060.123	7.936
RJGMM	MB	-0.91	-0.879	-0.885	-0.811	-0.752	0.787	0.677	0.636	0.613	0.494
	IQR	1.689	1.584	1.570	1.553	1.550	1.491	2.062	2.334	2.291	2.297
	MSE	1.21e+11	3.42e+12	5.58e+08	94.812	969.998	16.959	7.784	7.226	10.094	6.293
RJGMM$\star$	MB	-0.501	0.006	0.028	-0.008	-0.016	0.138	0.134	0.096	0.026	-0.012
	IQR	1.522	1.479	1.473	1.455	1.450	2.439	2.401	2.278	2.253	2.169
	MSE	0.702	0.564	0.526	0.524	0.503	2.464	2.285	1.921	1.790	1.624

Notes: 1'500 replications; $(\beta_0, \gamma_0) = (1, 0.5)$; $\text{corr}(\epsilon_i, U_i) = 0.6$; Dense signal; $\mu_n^2 = 1.5n$. I report the median bias (MB), interquartile range (IQR), and mean squared error (MSE). For the two-step GMM, CUE, and JGMM, $m/2$ moment conditions were selected randomly in each iteration. The same moment conditions were selected for each estimator each iteration. The preliminary estimate of (β_0, γ_0) for the two-step estimator was generated on the reduced set of moment conditions.

6.1.2 Supplementary simulations

Additional simulations were conducted and the results are presented here for the reader's interest. We wish to highlight that under more common simulation designs (*e.g.*, those in HK and NW), the results of the simulations are clearer than the ones presented in the main section. This demonstrates that more research is needed in order to understand the properties of the estimator in less conventional settings. We run two additional simulation designs. The first simulation uses a non-linear DGP from Section 4.2 and the second simulation uses the linear DGP from Section 4.1. The rest of the simulation design is adopted from Section 4.2 with the following exception: the first stage coefficient is set to $\Pi = m^{-1/2}(\iota_{0.4m}, 0_{0.6m})'$. We only run 700 simulation iterations due to time constraints. The results for the non-linear simulation are presented in Figures A1, A2, and A3 and the results from the linear simulation are present the results in Figure A4. Note that the results are not discussed with the same depth as the main results as we only intend to provide reader's additional information and possibly bestow more confidence in the RJGMM's performance.

Figures A1, A2, and A3 present the results that correspond to Figures 1, 2, and 3 with the updated first-stage coefficients. All three figures show lower median biases for all of the estimators relative to their Figure 1, 2, and 3 counterparts. The IQR values in the $m = 50$ case remain relatively constant. Only CUE has a large improvement in performance. Relative to the MSE, the estimators have a slight improvement in performance on the β parameter and a large improvement on the γ parameter. RJGMM's MSE for γ converges slowly as n grows in the $m = 50$ case. In Figure A2, RJGMM's bias is lower than in Figure 2. JGMM and CUE improve the least. Regarding the IQR, RJGMM's value is similar to the one-step's β value and slightly higher for γ. There is also a clear convergence of the MSE values of the RJGMM. The results in Figure A3 do not differ substantially from Figure A2.

The results presented in Figure A4 show a very clear picture. The RJGMM is median unbiased in every scenario, dominating the other models. The cost, is a slightly higher IQR relative to the two-step GMM estimator, however, the variance of the RJGMM is lower than that of CUE and JGMM. The results also support NW's findings that the JGMM performs worse then CUE in the cases where m is allowed to grow. The MSE paints a similar picture, in that the RJGMM estimator does not suffer from as many outliers as it did in the previous simulations. Interesting to note is also the one-step and two-step's convergence of bias in the $m = 50$ case. RJGMM and the oracle version seem to perform equally well in terms of bias. The IQR of the estimators converges nicely in the $m = 50$ case, as does the MSE in all cases. RJIVE has a lower IQR than RJGMM and the regularized estimators have the lowest MSE.

Figure A1: Supplementary non-linear simulation results for m = 50

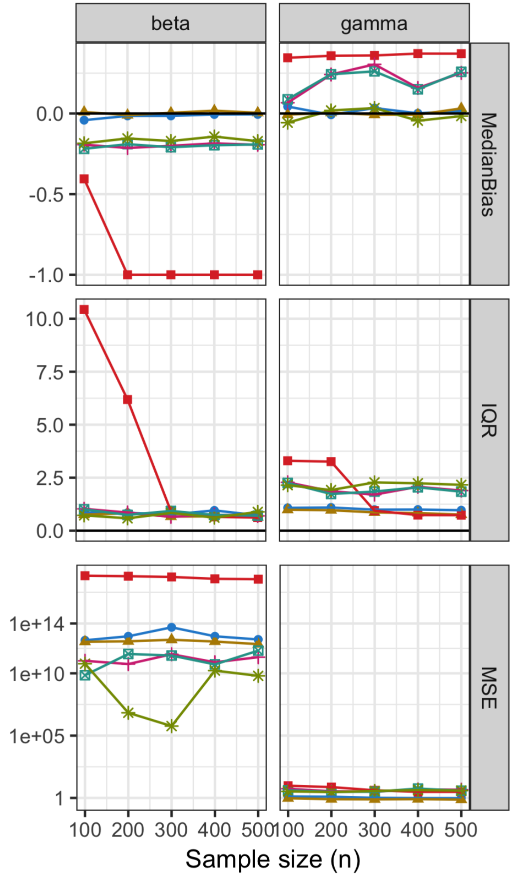


Figure A2: Supplementary non-linear simulation results for m = 0.95n

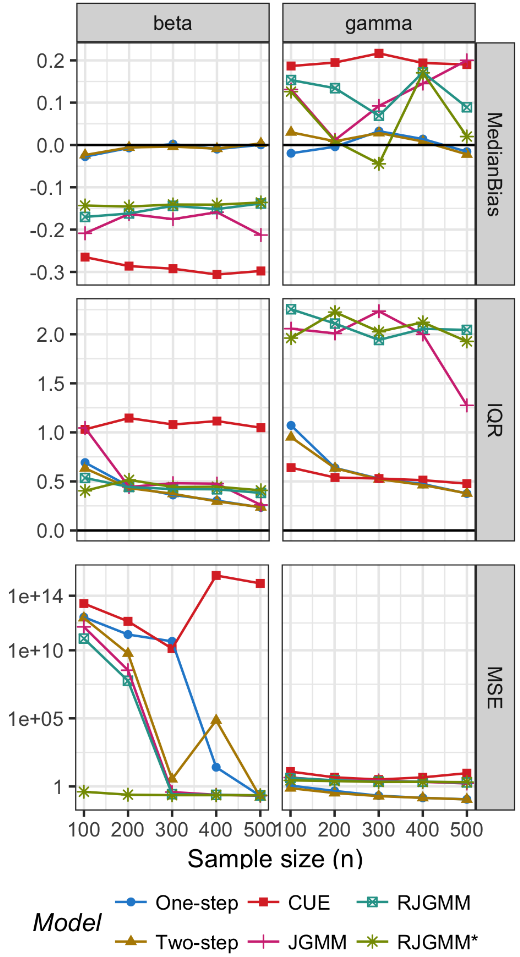
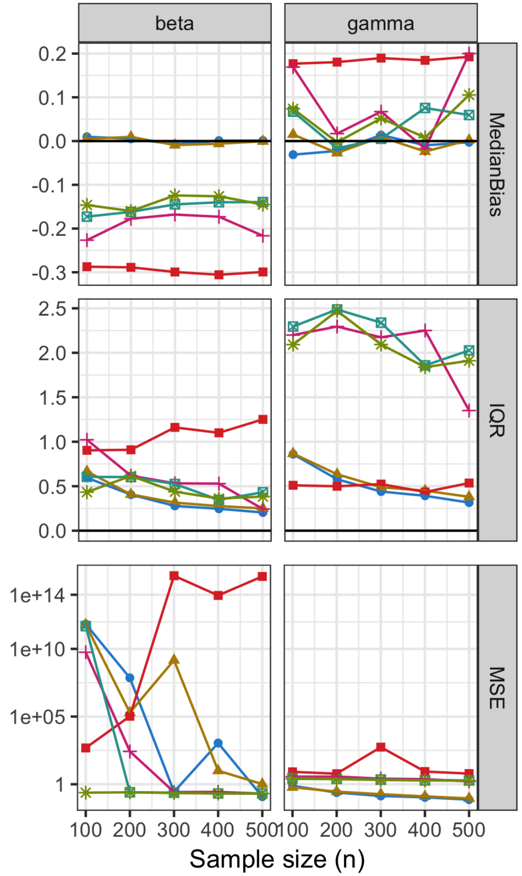
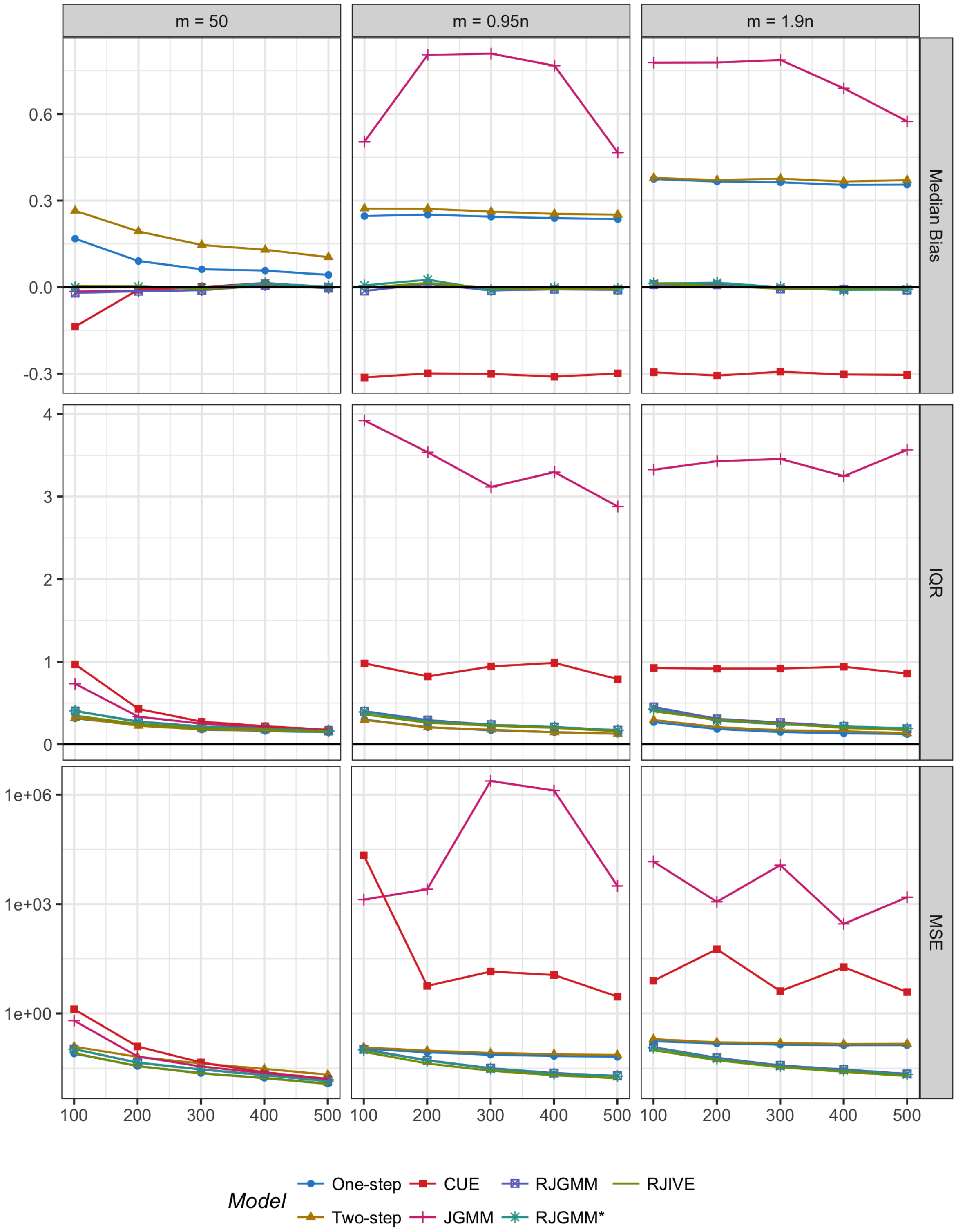


Figure A3: Supplementary non-linear simulation results for m = 1.9n



Notes: 700 replications; $(\beta_0, \gamma_0) = (1, 0.5)$; $\text{corr}(\epsilon_i, U_i) = 0.6$; Dense signal; $\mu_n^2 = 1.5n$. The 2 columns for each figure present the results for the β and γ parameter, respectively. The three rows from top to bottom present the median bias, IQR, and MSE. In the $m = 1.9n$ case, $m/2$ moment conditions were selected randomly in each iteration for the two-step GMM, CUE, and JGMM. The same moment conditions were selected for each estimator each iteration. The oracle version of the RJGMM is denoted by RJGMM$\star$. The preliminary estimate of (β_0, γ_0) for the two-step estimator was generated on the reduced set of moment conditions.

Figure A4: Supplementary linear simulation results with $m \in \{50, 0.95n, 1.9n\}$



Notes: 700 replications; $\beta_0 = 1$; $\text{corr}(\epsilon_i, U_i) = 0.6$; Dense signal; $\mu_n^2 = 1.5n$. The three columns from right to left present the results for the $m = 50$, $m = 0.95n$, and $m = 1.9n$. The three rows from top to bottom present the median bias, IQR, and MSE. In the $m = 1.9n$ case, $m/2$ moment conditions were selected randomly in each iteration for the two-step GMM, CUE, and JGMM. The same moment conditions were selected for each estimator each iteration. The oracle version of the RJGMM is denoted by RJGMM$\star$. The preliminary estimate of β_0 for the two-step estimator was generated on the reduced set of moment conditions.

6.2 Matlab code

The Matlab code used to run the simulations is presented below. Each script is presented in its own subsection. `RJGMM_NL_sim.m` and `RJGMM.m` are the main scripts. The former contains the simulation code and the latter contains the RJGMM estimation code. The rest of the files are dependencies. For more information, the reader is referred to the comments in each script.

6.2.1 Monte Carlo simulation code for Section 4.2 (`RJGMM_NL_sim.m`)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Non-linear RJGMM simulation script                   (RJGMM_NL_sim.m)
% Author: Luca Gaegauf
% Supervisor: Dr. Damian Kozbur
% Project: Master thesis (2017)
% Date: 15.11.2017
% Matlab: R2017b
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This script runs the non-linear simulations where the sample size,
% number of moment conditions, and concentration parameter is allowed
% to grow (see Section 4.2).
%
% Dependencies for this script include DGP.m, RJGMM.m, RJGMM_oracle.m,
% f_GMM.m, f_evalStats.m, and f_buildVariance.m. Note that
% RJGMM_oracle.m is not included in the appendix (RJGMM_oracle.m is
% RJGMM.m where theta_prelim is set to theta0).
%
% Simulation parameters:
%   n  = [100, 200, 300, 400, 500]
%   m  = [50, 0.95 * n, 1.9 * n]
%   CP = 1.5 * n
%   theta0 = [1, 0.5]
%
% We compare RJGMM to one-step GMM, two-step GMM, CUE, JGMM, and RJGMM
% oracle.
%
% For more information on the simulation design see Section 4.
%
% Plots were generated in R.
%

% Setup workspace ----------------------------------------------------
warning('off','optim:fminunc:SwitchingMethod')

% Data generating process --------------------------------------------
% Set data_form to 'NL2'
% data_form = 'NL2'
%     y = ((x * beta)^2)^(exp(gamma)/2) + e,
%     x = z * pi + u,
data_form = 'NL2';

% DGP.m script returns:
%   y_fn = @(x, e, theta) ...,
%   x_fn = @(z, u, pi) ...,
%   M_fn = @(x, y, z, theta) ...,
%   theta0, and
%   p (= dim(theta0)).
```

```
DGP

% Other DGP parameters: Correlation between the first-stage and
% second-stage errors Corr(eps, U)
rho = 0.6;

% Simulation Parameters -------------------------------------------------
% Number of simulation iterations
sim_max = 1500;

% Array of sample sizes
n_list = 100:100:500;

% Output table ----------------------------------------------------------
% Create an empty table to record and export the simulation results
variable_names = {'Model'};
for i = 1:p
    variable_names = [variable_names, {['medianBias_', num2str(i)], ...
        ['IQR_', num2str(i)], ...
        ['MSE_', num2str(i)]}]; %#op
end
variable_names = [variable_names, {'signal', 'n', 'm', 'm_used', 'CP', '
    lambda'}];
output_table = cell2table(cell(size(variable_names) - [1,0]), '
    VariableNames', variable_names);

% Begin Simulation ====================================================
% Run the simulation for n in n_list
for n = n_list

    % Three dimension of moment conditions scenarios
    for m_gr8r_n = 0:2
        % 1) the number of moment conditions grows with but is less than
        % the sample size (m = 0.95 * n)
        if m_gr8r_n == 0
            m_const = ceil(0.95 * n);

        % 2) the number of moment conditions grows with and is greater
        % than the sample size (m = 1.9 * n)
        elseif m_gr8r_n == 1
            m_const = ceil(1.9 * n);

        % 3) the number of moment conditions is fixed (m = 50)
        else
            m_const = 50;
        end

        % Instrumental variables covariance ---------------------------
        % The variance of the instruments is set to 0.3. The correlation
        % between the ith and jth instruments is set to 0.5^(|j-i|).
        % Therefore, their variance is set to 0.3 + 0.5^(|j-i|).

        % V_z is the covariance matrix. Set the diagonal to 0.3
        V_z = eye(m_const) * 0.3;

        % Create all combinations of vectors 1:m and 1:m.
        % i.e. vec_ix = [1,1; 2,1; 3,1;...; m,m-1; m,m]
        vec_ix = combvec(1:m_const, 1:m_const)';
```

```
        % Remove diagonal elements of vector of indexes. vec_ix will be
        % used to index positions of the covariance matrix V_z. Since we
        % already determined the variances, we remove all the indexes
        % pointing to diagonal elements.
        % i.e. vec_ix = [2,1; 3,1;...; m,m-1]
        vec_ix = vec_ix(vec_ix(:, 1) ~= vec_ix(:, 2), :);

        % Returns the linear index equivalent to the row and column
        % subscripts
        % i.e. idx = [2; 3; ...; (m*m)-1]
        idx = sub2ind(size(V_z), vec_ix(:, 1), vec_ix(:, 2));

        % Update the off-diagonal elements of the convariance matrix to
        % cov(Z_i, Z_j) = sqrt(Var(Z_i)) * sqrt(Var(Z_j)) *
        %                                               corr(Z_i, Z_j)
        % where corr(Z_i, Z_j) = 0.5^|i-j|, and
        % Var(Z_i) = Var(Z_i) = 0.3
        V_z(idx) = (sqrt(0.3) ^ 2) * (0.5 .^ (abs(vec_ix(:, 1) - vec_ix
            (:, 2))));

        % Error covariance ------------------------------------------------
        % Set iota_m such that the signal is dense.
        % i.e. iota_m = [1, ..., 1, 0, ..., 0] (40% of the elements are
        % ones, the remaining are zeros)
        iota_m = [ones(ceil(m_const * 0.4), 1); zeros(floor(m_const *
            0.6), 1)];

        % Set the concentration parameter
        CP = 1.5 * n;

        % First-stage coefficients
        pi = ((-1 * iota_m).^((1:m_const)')) / sqrt(m_const);

        % Variance of first-stage error
        % Var(U) = n * pi' * Var(Z) * pi / CP
        V_U = n * (pi' * V_z * pi) / CP;

        % Variance of second stage error (set to 2)
        V_eps = 2;

        % Covariance of first-stage and second-stage errors
        % Cov(eps, U) = corr(eps, U) * sqrt(Var(eps)) * sqrt(Var(U))
        V_eps_U = rho * sqrt(V_eps) * sqrt(V_U);

        % Error covariance matrix
        V_err = [V_eps, V_eps_U; V_eps_U, V_U];

        % Storing variables ---------------------------------------------
        theta_1S     = zeros(sim_max, p);
        theta_2S     = zeros(sim_max, p);
        theta_CUE    = zeros(sim_max, p);
        theta_J1GMM  = zeros(sim_max, p);
        theta_RJGMM  = zeros(sim_max, p);
        theta_RJGMMo = zeros(sim_max, p);

        penalty_RJGMM = zeros(sim_max, 1);

```

```
        % Loop through simulation iterations --------------------------
        for nsim = 1:sim_max
            rng(nsim); % set seed
            m = m_const; % reset m

            % Generate mean zero shocps with V_err covariance structure
            E = randn(n, 2) * chol(V_err); e = E(:, 1); u = E(:, 2);
            e = e - mean(e); u = u - mean(u);

            % Generate mean zero instruments with V_z covariance
            % structure
            z = randn(n, m_const) * chol(V_z);
            z = z - mean(z);

            % Generate endogenous treatment variable
            x = x_fn(z, u, pi);

            % Generate outcome variable
            y = y_fn(x, e, theta0);

            % Main model ==============================================

            % RJGMM with JIVE1 ----------------------------------------
            % Ridge regression penalty term
            RJGMM_penalty = var(x) * m / n; penalty_RJGMM(nsim) = RJGMM_penalty;
            % Estimate RJGMM
            theta_RJGMM(nsim, :) = RJGMM(M_fn, x, y, z, p, RJGMM_penalty);

            % RJGMM with JIVE1 (oracle) -------------------------------
            % Estimate RJGMM with preliminary theta value set to the
            % truth
            theta_RJGMMo(nsim, :) = RJGMM_oracle(M_fn, x, y, z, p, theta0, RJGMM_penalty);

            % Comparison models =======================================

            % One-step GMM (with full set of variables) ---------------
            % One-step GMM weighting matrix
            W_1S = @(theta) eye(m);
            % Estimate one-step GMM
            theta_1S(nsim, :) = f_GMM(M_fn, x, y, z, p, W_1S);

            % The rest of the models are not well-defined when the
            % number of moment conditions is greater than sample size.
            % Therefore, if m > n we select a random sample of the
            % moment conditions.
            if m > n
                % Select m/2 instruments (randomly)
                m2 = ceil(m / 2);
                red_samp = datasample(1:m, m2, 'Replace', false);

                % Subset instruments
                z2 = z(:, red_samp);
            else
                m2 = m; z2 = z;
            end
```

```

        % One-step GMM (with subset of variables) -----------------
        % This estimator is only used as the preliminary estimate of
        % the two-step GMM estimator.

        % One-step GMM weighting matrix
        W_1Ss = @(theta) eye(m2);
        % Estimate one-step GMM
        theta_1Ss = f_GMM(M_fn, x, y, z2, p, W_1Ss);

        % Two-step GMM --------------------------------------------
        % The preliminary estimator for the weighting matrix is
        % theta_1Ss.

        % Two-step GMM weighting matrix
        Vhat_2S = f_buildVariance(M_fn, x, y, z2, theta_1Ss);
        W_2S = @(theta) inv(Vhat_2S);
        % Estimate two-step GMM
        theta_2S(nsim, :) = f_GMM(M_fn, x, y, z2, p, W_2S);

        % CU GMM ---------------------------------------------------
        % CUE optimizes the weighting matrix simultaneously and does
        % not require a preliminary estimate for theta.

        % CUE weighting matrix
        Vhat_CUE = @(theta) f_buildVariance(M_fn, x, y, z2, theta);
        W_CUE = @(theta) inv(Vhat_CUE(theta));
        % Estimate CUE
        theta_CUE(nsim, :) = f_GMM(M_fn, x, y, z2, p, W_CUE);

        % JGMM with JIVE1 ------------------------------------------
        % The JIVE1 jackknife GMM solves RJGMM with a penalty of
        % zero

        % Estimate JGMM
        theta_J1GMM(nsim, :) = RJGMM(M_fn, x, y, z2, p);
    end

    % Output ======================================================
    % Calculate the median bias, IQR, and MSE of the estimators
    es_1S     = f_evalStats(theta_1S, theta0);
    es_2S     = f_evalStats(theta_2S, theta0);
    es_CUE    = f_evalStats(theta_CUE, theta0);
    es_J1GMM  = f_evalStats(theta_J1GMM, theta0);
    es_RJGMM  = f_evalStats(theta_RJGMM, theta0);
    es_RJGMMo = f_evalStats(theta_RJGMMo, theta0);

    % Round to 4 digits
    es_1S     = round(es_1S, 4);
    es_CUE    = round(es_CUE, 4);
    es_CUE    = round(es_CUE, 4);
    es_J1GMM  = round(es_J1GMM, 4);
    es_RJGMM  = round(es_RJGMM, 4);
    es_RJGMMo = round(es_RJGMMo, 4);

    % Build the output table
    constVars = {'dense', n, m_const};

```

```
        new_rows = [
            ['1 step', num2cell(es_1S(1:end)),      constVars, m_const,
                CP, NaN(1)];
            ['2 step', num2cell(es_2S(1:end)),      constVars, m2,
                CP, NaN(1)];
            ['CUE',    num2cell(es_CUE(1:end)),     constVars, m2,
                CP, NaN(1)];
            ['JGMM',   num2cell(es_J1GMM(1:end)),   constVars, m2,
                CP, NaN(1)];
            ['RJGMM',  num2cell(es_RJGMM(1:end)),   constVars, m_const,
                CP, mean(penalty_RJGMM)];
            ['RJGMMO', num2cell(es_RJGMMo(1:end)), constVars, m_const,
                CP, mean(penalty_RJGMM)]
            ];
        output_table = [output_table; new_rows]; %#ok

        % Display results
        disp('-----------------------------------------------------------
            ---------------------------------------')
        disp('Simulation')
        disp('-----------------------------------------------------------
            ---------------------------------------')
        disp('Parameters')
        disp(['#obs = ', num2str(n), '   #instr = ', num2str(m_const), '
            /', num2str(m2), '        corr(e,u) = ', num2str(rho)])
        disp(['CP = ', num2str(CP), '     #simulations = ', num2str(
            sim_max), '    design: ', signal])

        disp('------------------------------------------------------------
            -------------------------------------')
        disp('Model     Med. Bias (b)     IQR (b)             MSE (b)
                Med. Bias (c)     IQR (c)             MSE (c)')
        disp(['1 step    ', num2str(es_1S(1:end),     '%+1.4f \t')])
        disp(['2 step    ', num2str(es_2S(1:end),     '%+1.4f \t')])
        disp(['CUE       ', num2str(es_CUE(1:end),    '%+1.4f \t')])
        disp(['JGMM      ', num2str(es_J1GMM(1:end),  '%+1.4f \t')])
        disp(['RJGMM     ', num2str(es_RJGMM(1:end),  '%+1.4f \t')])
        disp(['RJGMMo    ', num2str(es_RJGMMo(1:end), '%+1.4f \t')])
        disp('------------------------------------------------------------
            -------------------------------------')
    end
end

% View results
%output_table

% Save results
writetable(output_table, [data_form '_n' num2str(n) '_reps' num2str(
    sim_max) '_sim.csv'])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                  END                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

6.2.2 RJGMM function (RJGMM.m)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RJGMM script                                                (RJGMM.m)
```

```
% Author: Luca Gaegauf
% Supervisor: Dr. Damian Kozbur
% Project: Master thesis (2017)
% Date: 15.11.2017
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This script estimates the ridge regularized jackknife GMM using the
% selection matrix approach.
%

function thetaRJGMM = RJGMM(M, x, y, z, p, lambda)
% RJGMM  estimate regularized jackknife GMM
%
% Input:
%   M (function handle): M = @(x, y, z, theta) ...
%       e.g. M = @(x, y, z, theta) z' * (y - x * theta);
%   x, y, z (double): data.
%       dim(x) = n x 1
%       dim(y) = n x 1
%       dim(z) = n x m
%   p (double): dim(lambda) = 1
%       Number of parameters to estimate.
%   lambda (double): dim(lambda) = 1
%       If lambda = 0 or empty, then JGMM will be estimated.
%
% Output:
%   - thetaRJGMM (double):
%       dim(thetaRJGMM) = 1 x p

% Error handling -----------------------------------------------------------
if nargin < 5 % if less than 5 arguments are provided
    error('Minimum required input: M, x, y, z, p. See help RJGMM for
        more information.');
elseif nargin == 5 % if not penalty term is provided estimate JGMM
    lambda = 0;
end

% RJGMM --------------------------------------------------------------------
[n, m] = size(z);

Ahati = cell(n, 1);
Mi_cell = cell(n, 1);

% Generate an array of ith moment condition function handles
for j = 1:n
    Mi_cell{j} = @(theta) M(x(j,:), y(j,:), z(j,:), theta);
end

% Jackknife iterations
for i = 1:n
    % Data (without-ith sample)
    zi = z((1:n) ~= i, :);
    xi = x((1:n) ~= i, :);
    yi = y((1:n) ~= i, :);

    % Moment condition variance matrix --------------------------------
    % Estimate a preliminary theta for the selection matrix using
    % 1-step GMM of without-ith sample.
```

```
    % Weighting matrix
    W_1S = @(theta) eye(m);
    % Minimize GMM objective function
    theta_prelim = f_GMM(M, xi, yi, zi, p, W_1S);
    % Vhati of without-ith sample variance of moment condition at
    % theta_prelim
    Vhat = f_buildVariance(M, xi, yi, zi, theta_prelim);

    % For RJGMM_oracle.m theta_prelim is set to theta0

    % Moment condition derivative -------------------------------------
    % Without-ith sample average of moment condition
    Mi = @(theta) M(xi, yi, zi, theta);

    % Matrix of perturbations
    epsilon = 1e-4;
    % diag(repmat(epsilon, 1, p))
    %  = [epsilon, 0, 0, ..., 0;
    %      0, epsilon, 0, ..., 0;
    %      ...
    %      0, 0, ..., 0, epsilon]
    d = diag(repmat(epsilon, 1, p));
    % Derivative of moment condition
    % E.g. dim(theta_prelim) = 2:
    % dMi =
    %  [Mi(theta_prelim + [0, epsilon]) -  Mi(theta_prelim)) / epsilon,
    %   Mi(theta_prelim + [epsilon, 0]) -  Mi(theta_prelim)) / epsilon]
    dMi = cellfun(@(esp) {(Mi(theta_prelim + esp) -  Mi(theta_prelim)) /
        epsilon}, num2cell(d, 2));

    % Selection matrix ------------------------------------------------
    % Calculate the selection matrix on the without-ith sample
    % Ahat_{-i} = D(theta_prelim)'(V(theta_prelim) + penalty)^{-1}
    Ahati{i} = cell2mat(dMi')' * inv(lambda * eye(m) + Vhat); %#ok
end

% Calculate the ith optimal moment condition using the selection
% matrix (Ahat_{-i} g_i(theta))
% AhatiMi is a cell array of function handles dependent on theta
AhatiMi = cellfun(@(Ai, M_i) {@(theta) Ai * M_i(theta)}, Ahati , Mi_cell
    ) ;

% Sum (over i) optimal moment conditions
% sum_{i=1}^n Ahat_{-i} g_i(theta)
Ahatsum = @(~) 0;
for jj = 1:n
    Ahatsum = @(theta) Ahatsum(theta) + AhatiMi{jj}(theta);
end

% Create RJGMM selection matrix quasi objective function
RJGMMObj = @(theta) Ahatsum(theta)' * Ahatsum(theta);

% Estimate RJGMM
fminunc_options = optimset('Display','off');
thetaRJGMM = fminunc(RJGMMObj, 0.7 * ones(1, p), fminunc_options);

end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                  END                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

6.2.3 GMM function (f_GMM.m)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GMM script                                                  (f_GMM.m)
% Author: Luca Gaegauf
% Supervisor: Dr. Damian Kozbur
% Project: Master thesis (2017)
% Date: 15.11.2017
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This script estimates GMM. It can estimate one-step, two-step, and
% continuous updating estimator.
%

function thetaGMM = f_GMM(M, x, y, z, p, W)
% F_GMM  estimate GMM (E(M(theta0)) = 0)
%                 theta_hat = argmin M(theta)' * W(theta) * M(theta)
%          with data of the following form:
%                 y = f(x, e; theta0),
%                 z = g(z, u; pi0),
%          where M(theta) = z'e(theta).
%
%   theta_1Step = F_GMM(M, @(theta) eye(m), theta_init) estimates one-
   step
%                  GMM.
%   theta_2Step = F_GMM(M, @(theta) inv(Vhat(theta_1Step)), theta_init)
%                  estimates two-step GMM.
%   theta_CUE   = F_GMM(M, @(theta) inv(Vhat(theta)), theta_init)
   estimates
%                  continuous updating GMM estimator
%
% Input args:
%   M (anonymous function): moment condition with q parameters
%       dim(M) = q x 1
%   x, y, z (double): data.
%       dim(x) = n x 1
%       dim(y) = n x 1
%       dim(z) = n x m
%   W (anonymous function): weighting matrix
%       dim(W) = m x m
%   p (double): dim(lambda) = 1
%       Number of parameters to estimate.
%
% Output:
%   - thetaRJGMM (double):
%       dim(thetaRJGMM) = 1 x p

% Error handling -------------------------------------------------------
if nargin < 6 % if less than 6 arguments are provided
    error('Required input: M, x, y, z, p, W. See help f_GMM for more
       information.');
end

% If M isn't a function handle
```

```
if ~isa(M, 'function_handle')
    error('M must be anonymous function.');
end

% If W isn't a function handle
if ~isa(W, 'function_handle')
    error('W must be anonymous function. Try: @(theta) W,');
end

% GMM ----------------------------------------------------------------
% Store the GMM objective function
GMMObj = @(theta) M(x, y, z, theta)' * W(theta) * M(x, y, z, theta);

% Estimate GMM
fminunc_options = optimset('Display','off');
thetaGMM = fminunc(GMMObj, 0.7 * ones(1, p), fminunc_options);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                END                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

6.2.4 Data generating process (DGP.m)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Data generating process script                               (DGP.m)
% Author: Luca Gaegauf
% Supervisor: Dr. Damian Kozbur
% Project: Master thesis (2017)
% Date: 15.11.2017
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This script initializes the data structure. There are 3 DGP types:
% linear (1 param), exponential (1 param), and non-linear (2 param).
%
% The script creates:
%   x_fn (function_handle): the functional form of the treatment
%                           variable
%   y_fn (function_handle): the functional form of the dependent
%                           variable
%   M_fn (function_handle): the functional form of the moment
%                           condition
%   p (double): dimension of the parameter of interest
%   theta0 (double): parameter of interest
%     Individal parameters of interest: alpha0, beta0, ...
%

if strcmp(data_form, 'linear')
    x_fn = @(z, u, pi) z * pi + u;
    y_fn = @(x, e, theta) x * theta + e;
    M_fn = @(x, y, z, theta) z' * (y - x * theta);
    p = 1;
    beta0 = 1; theta0 = beta0;
elseif strcmp(data_form, 'exp')
    x_fn = @(z, u, pi) z * pi + u;
    y_fn = @(x, e, theta) exp(x * theta) + e;
    M_fn = @(x, y, z, theta) z' * (y - exp(x * theta));
```

```
    p = 1;
    beta0 = 1; theta0 = beta0;
elseif strcmp(data_form, 'NL2')
    x_fn = @(z, u, pi) z * pi + u;
    y_fn = @(x, e, theta) (((x * theta(1)).^2).^(exp(theta(2))/2)) + e;
    M_fn = @(x, y, z, theta) z' * (y - (((x * theta(1)).^2).^(exp(theta(2))/2)));
    p = 2;
    beta0 = 1; gamma0 = 0.33;
    theta0 = [beta0, gamma0];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                END                                  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

6.2.5 Build variance function (f_buildVariance.m)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Generate moment condition variance matrix script  (f_buildVariance.m)
% Author: Luca Gaegauf
% Supervisor: Dr. Damian Kozbur
% Project: Master thesis (2017)
% Date: 15.11.2017
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This function generates the sample variance matrix of the moment
% conditions at theta_prelim.
%

function varMatrix = f_buildVariance(M, x, y, z, theta_prelim)
% buildVariance  build a variance matrix of moment condition M
%
% Input:
%   M (function handle): M = @(x, y, z, theta) ...
%       e.g. M = @(x, y, z, theta) z' * (y - x * theta);
%   x, y, z (double): data.
%       dim(x) = n x 1
%       dim(y) = n x 1
%       dim(z) = n x m
%   theta_prelim (double): dim(theta_prelim) = p
%       Preliminary theta at which to calculate variance
%
% Output:
%   varMatrix (double): variance matrix of moment condition M
%       dim(varMatrix) = m x m

[n, m] = size(z);

% n x m matrix of ith moment conditions (stacked)
Mi = zeros(n, m);
% Calculate the moment condition using the ith observation of the data
% at theta_prelim
for i = 1:n
    Mi(i,:) = M(x(i,:), y(i,:), z(i,:), theta_prelim)';
end

% Calculate the variance of the moment condition at theta_prelim
```

```
varMatrix = Mi' * Mi / n;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                END                                    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

6.2.6 Evaluation statistics function (f_evalstats.m)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Evaluation statistics script                          (f_evalStats.m)
% Author: Luca Gaegauf
% Supervisor: Dr. Damian Kozbur
% Project: Master thesis (2017)
% Date: 15.11.2017
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This script calculates the median bias, interquartile range, and
% mean squared error of a vector of estimated parameters.
%

function evalStats = f_evalStats(estimates, truth)
% f_EVALSTATS   calculate median bias, IQR, and MSE of estimated
%               parameters
%
%   evalStats(1,:) for median bias,
%   evalStats(2,:) for IQR,
%   evalStats(3,:) for MSE.
%   evalStats(:,i) for stats on ith parameter.
%
% Input args:
%   - estimates (double):
%       dim(estimates) = r x p
%   - truth (double):
%       dim(truth) = 1 x p
%
% Output:
%   - evalStats (double):
%       dim(evalStats) = 3 x p

% Error handling -------------------------------------------------------
if nargin < 2 % if less than 2 arguments are provided
    error('Minimum required input: estimates, truth. See help f_evalStats for more information.');
end

% Evaluation statistics ------------------------------------------------
[sim_max, p] = size(estimates);

evalStats = zeros(3, p);
% Iterate over the parameters
for nparam = 1:p
    % Calculate median bias
    evalStats(1, nparam) = median(estimates(:, nparam) - truth(nparam), 'omitnan');

    % Calculate IQR
```

```
    evalStats(2, nparam) = iqr(estimates(:, nparam));

    % Calculate MSE
    evalStats(3, nparam) = mean((estimates(:, nparam) - truth(nparam) .* ones(sim_max, 1)).^2, 'omitnan');
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                END                                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

6.2.7 Monte Carlo simulation code for Section 4.1 (RJGMM_sim.m)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RJGMM simulation script                                (RJGMM_sim.m)
% Author: Luca Gaegauf
% Supervisor: Dr. Damian Kozbur
% Project: Master thesis (2017)
% Date: 15.11.2017
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% This script runs the non-linear simulations where the sample size,
% number of moment conditions, and concentration parameter is allowed
% to grow.
%
% Dependencies for this script include DGP.m, RJGMM.m, RJGMM_oracle.m,
% f_GMM.m, rjive.m, f_evalStats.m, and f_buildVariance.m. Note that
% RJGMM_oracle.m and rjive.m is not included in the appendix. For
% RJGMM_oracle.m theta_prelim is set to theta0 (in RJGMM.m)
%
% Simulation parameters:
%   n  = 100
%   m  = [0.95 * n, 1.9 * n]
%   CP = 1.5 * n
%   signal sparsity = [dense, sparse]
%   signal strength = [strong, weak]
%   theta0          = [1] if linear or exponential
%                   = [1, 0.5] if non-linear
%
% We compare RJGMM to one-step GMM, two-step GMM, CUE, JGMM, and RJGMM
% oracle. If the data form is linear we also compare it to RJIVE.
%
% For more information on the simulation design see Section 4.
%

% Setup workspace -----------------------------------------------------
warning('off','optim:fminunc:SwitchingMethod')

% Data generating process ---------------------------------------------
% Set data_form to 'linear', 'exp', or 'NL2'
% data_form = 'linear'
%     y = x * beta + e,
%     x = z * pi + u,
% data_form = 'exp'
%     y = exp(x * beta) + e,
%     x = z * pi + u,
% data_form = 'NL2'
```

```
%      y = ((x * beta)^2)^(exp(gamma)/2) + e,
%      x = z * pi + u,
data_form = 'linear';

% DGP script returns:
%    y_fn = @(x, e, theta) ...,
%    x_fn = @(z, u, pi) ...,
%    M_fn = @(x, y, z, theta) ...,
%    theta0, and
%    p (= dim(theta0)).
DGP

% Other data parameters
% Correlation between the first-stage and second-stage errors
% Corr(eps, U)
rho = 0.6;

% Simulation Parameters ------------------------------------------------
% Number of simulation iterations
sim_max = 1500;

% Sample size
n = 100;

% Output table ---------------------------------------------------------
% Create an empty table to record and export the simulation results
variable_names = {'Model'};
for i = 1:p
    variable_names = [variable_names, {['medianBias_', num2str(i)], ...
        ['IQR_', num2str(i)], ...
        ['MSE_', num2str(i)]}]; %#ok
end
variable_names = [variable_names, {'signal', 'n', 'm', 'm_used', 'CP', 'lambda'}];
output_table = cell2table(cell(size(variable_names) - [1,0]), 'VariableNames', variable_names);

% Begin Simulation =====================================================
% Three dimension of moment conditions scenarios
for m_gr8r_n = 0:1
    % 1) the number of moment conditions grows with but is less than
    % the sample size (m = 0.95 * n)
    if m_gr8r_n == 0
        m_const = ceil(0.95 * n);

    % 2) the number of moment conditions grows with and is greater
    % than the sample size (m = 1.9 * n)
    elseif m_gr8r_n == 1
        m_const = ceil(1.9 * n);
    end

    % Instrumental variables covariance -------------------------------
    % The variance of the instruments is set to 0.3. The correlation
    % between the ith and jth instruments is set to 0.5^(|j-i|).
    % Therefore, their variance is set to 0.3^2 + 0.5^(|j-i|).

    % V_z is the covariance matrix. Set the diagonal to 0.3
    V_z = eye(m_const) * 0.3;
```

```
    % Create all combinations of vectors 1:m and 1:m.
    % i.e. vec_ix = [1,1; 2,1; 3,1;...; m,m-1; m,m]
    vec_ix = combvec(1:m_const, 1:m_const)';

    % Remove diagonal elements of vector of indexes. vec_ix will be
    % used to index positions of the covariance matrix V_z. Since we
    % already determined the variances, we remove all the indexes
    % pointing to diagonal elements.
    % i.e. vec_ix = [2,1; 3,1;...; m,m-1]
    vec_ix = vec_ix(vec_ix(:, 1) ~= vec_ix(:, 2), :);

    % Returns the linear index equivalent to the row and column
    % subscripts
    % i.e. idx = [2; 3; ...; (m*m)-1]
    idx = sub2ind(size(V_z), vec_ix(:, 1), vec_ix(:, 2));

    % Update the off-diagonal elements of the convariance matrix to
    % cov(Z_i, Z_j) = sqrt(Var(Z_i)) * sqrt(Var(Z_j)) *
    %                                                corr(Z_i, Z_j)
    % where corr(Z_i, Z_j) = 0.5^|i-j|, and
    % Var(Z_i) = Var(Z_i) = 0.3
    V_z(idx) = (sqrt(0.3) ^ 2) * (0.5 .^ (abs(vec_ix(:, 1) - vec_ix(:,
       2))));

    for signal_n = 1:2
        if signal_n == 1 % Dense signal
            signal = 'dense';
            % Set iota_m such that the signal is dense.
            % i.e. iota_m = [1, ..., 1, 0, ..., 0] (40% of the elements
            % are ones, the remaining are zeros)
            iota_m = [ones(ceil(m_const * 0.4), 1); zeros(floor(m_const
               * 0.6), 1)];
        elseif signal_n == 2 % Sparse signal
            % Set iota_m such that the signal is sparse.
            % i.e. iota_m = [1, 1, 1, 1, 1, 0, ..., 0] (5 elements are
            % ones, the remaining are zeros)
            signal = 'sparse';
            iota_m = [ones(5, 1); zeros(m_const - 5, 1)];
        end

        for signalStrength_n = 1:2
            if signalStrength_n == 1 % Strong signal
                CP = 150;
            elseif signalStrength_n == 2 % Weak signal
                CP = 30;
            end

            % Error covariance ----------------------------------------
            % First-stage coefficients
            pi = ((-1 * iota_m).^((1:m_const)')) / sqrt(m_const);

            % Variance of first-stage error
            % Var(U) = n * pi' * Var(Z) * pi / CP
            V_U = n * (pi' * V_z * pi) / CP;

            % Variance of second stage error (set to 2)
            V_eps = 2;
```

```
        % Covariance of first-stage and second-stage errors
        % Cov(eps, U) = corr(eps, U) * sqrt(Var(eps)) *
        %                                          sqrt(Var(U))
        V_eps_U = rho * sqrt(V_eps) * sqrt(V_U);

        % Error covariance matrix
        V_err = [V_eps, V_eps_U; V_eps_U, V_U];

        % Storing variables ---------------------------------------
        theta_1S      = zeros(sim_max, p);
        theta_2S      = zeros(sim_max, p);
        theta_CUE     = zeros(sim_max, p);
        theta_J1GMM   = zeros(sim_max, p);
        theta_RJGMM   = zeros(sim_max, p);
        theta_RJGMMo  = zeros(sim_max, p);
        % If data_from == 'linear' include RJIVE simulation
        theta_RJIVE   = zeros(sim_max, p);

        penalty_RJGMM = zeros(sim_max, 1);

        % Loop through simulation iterations ----------------------
        for nsim = 1:sim_max
            rng(nsim); % set seed
            m = m_const; % reset m

            % Generate mean zero shocks with V_err covariance
            % structure
            E = randn(n, 2) * chol(V_err); e = E(:, 1); u = E(:, 2);
            e = e - mean(e); u = u - mean(u);

            % Generate mean zero instruments with V_z covariance
            % structure
            z = randn(n, m_const) * chol(V_z);
            z = z - mean(z);

            % Generate endogenous treatment variable
            x = x_fn(z, u, pi);

            % Generate outcome variable
            y = y_fn(x, e, theta0);

            % Main model ==========================================

            % RJGMM with JIVE1 ------------------------------------
            % Ridge regression penalty term
            RJGMM_penalty = var(x) * m / n; penalty_RJGMM(nsim) =
                RJGMM_penalty;
            % Estimate RJGMM
            theta_RJGMM(nsim, :) = RJGMM(M_fn, x, y, z, p,
                RJGMM_penalty);

            % RJGMM with JIVE1 (oracle) ---------------------------
            % Estimate RJGMM with preliminary theta value set to
            % the truth
            theta_RJGMMo(nsim, :) = RJGMM_oracle(M_fn, x, y, z, p,
                theta0, RJGMM_penalty);

```

```
            % Comparison models ===================================
            % Modeling RJIVE with JIVE1 ---------------------------
            % Minimize obj function (with array of starting points)
            theta_RJIVE(nsim, :) = rjive(y, x, z, RJGMM_penalty * n)
                ;

            % One-step GMM (with full set of variables) -----------
            % One-step GMM weighting matrix
            W_1S = @(theta) eye(m);
            % Estimate one-step GMM
            theta_1S(nsim, :) = f_GMM(M_fn, x, y, z, p, W_1S);

            % The rest of the models are not well-defined when the
            % number of moment conditions is greater than sample
            % size. Therefore, if m > n we select a random sample
            % of the moment conditions.
            if m > n
                % Select m/2 instruments (randomly)
                m2 = ceil(m / 2);
                red_samp = datasample(1:m, m2, 'Replace', false);

                % Subset instruments
                z2 = z(:, red_samp);
            else
                m2 = m; z2 = z;
            end

            % One-step GMM (with subset of variables) -------------
            % This estimator is only used as the preliminary
            % estimate of the two-step GMM estimator.

            % One-step GMM weighting matrix
            W_1Ss = @(theta) eye(m2);
            % Estimate one-step GMM
            theta_1Ss = f_GMM(M_fn, x, y, z2, p, W_1Ss);

            % Two-step GMM ----------------------------------------
            % The preliminary estimator for the weighting matrix is
            % theta_1Ss.

            % Two-step GMM weighting matrix
            Vhat_2S = f_buildVariance(M_fn, x, y, z2, theta_1Ss);
            W_2S = @(theta) inv(Vhat_2S);
            % Estimate two-step GMM
            theta_2S(nsim, :) = f_GMM(M_fn, x, y, z2, p, W_2S);

            % CU GMM ----------------------------------------------
            % CUE optimizes the weighting matrix simultaneously and
            % does not require a preliminary estimate for theta.

            % CUE weighting matrix
            Vhat_CUE = @(theta) f_buildVariance(M_fn, x, y, z2,
                theta);
            W_CUE = @(theta) inv(Vhat_CUE(theta));
            % Estimate CUE
            theta_CUE(nsim, :) = f_GMM(M_fn, x, y, z2, p, W_CUE);

            % JGMM with JIVE1 -------------------------------------
```

```
        % The JIVE1 jackknife GMM solves RJGMM with a penalty
        % of zero

        % Estimate JGMM
        theta_J1GMM(nsim, :) = RJGMM(M_fn, x, y, z2, p);
    end

    % Output ==================================================
    % Calculate the median bias, IQR, and MSE of the estimators
    es_1S     = f_evalStats(theta_1S, theta0);
    es_2S     = f_evalStats(theta_2S, theta0);
    es_CUE    = f_evalStats(theta_CUE, theta0);
    es_J1GMM  = f_evalStats(theta_J1GMM, theta0);
    es_RJGMM  = f_evalStats(theta_RJGMM, theta0);
    es_RJGMMo = f_evalStats(theta_RJGMMo, theta0);
    es_RJIVE  = f_evalStats(theta_RJIVE, theta0);

    % Round to 4 digits
    es_1S     = round(es_1S, 4);
    es_CUE    = round(es_CUE, 4);
    es_CUE    = round(es_CUE, 4);
    es_J1GMM  = round(es_J1GMM, 4);
    es_RJGMM  = round(es_RJGMM, 4);
    es_RJGMMo = round(es_RJGMMo, 4);
    es_RJIVE  = round(es_RJIVE, 4);

    % Build the output table
    constVars = {signal, n, m_const};

    new_rows = [
        ['1 step', num2cell(es_1S(1:end)),     constVars,
           m_const, CP, NaN(1)];
        ['2 step', num2cell(es_2S(1:end)),     constVars, m2,
                CP, NaN(1)];
        ['CUE',    num2cell(es_CUE(1:end)),    constVars, m2,
                CP, NaN(1)];
        ['JGMM',   num2cell(es_J1GMM(1:end)),  constVars, m2,
                CP, NaN(1)];
        ['RJGMM',  num2cell(es_RJGMM(1:end)),  constVars,
           m_const, CP, mean(penalty_RJGMM)];
        ['RJGMMO', num2cell(es_RJGMMo(1:end)), constVars,
           m_const, CP, mean(penalty_RJGMM)];
        ['RJIVE',  num2cell(es_RJIVE(1:end)'), constVars,
           m_const, CP, mean(penalty_RJGMM)]
        ];
    output_table = [output_table; new_rows]; %#ok

    % Display results
    disp('--------------------------------------------------------
       ------------------------------------------')
    disp('Simulation')
    disp('--------------------------------------------------------
       ------------------------------------------')
    disp('Parameters')
    disp(['#obs = ', num2str(n), '   #instr = ', num2str(m_const
       ), '/', num2str(m2), '        corr(e,u) = ', num2str(rho)
       ])
    disp(['CP = ', num2str(CP), '     #simulations = ', num2str(
```

```
                sim_max), '     design: ', signal])

            disp('-----------------------------------------------------
                ------------------------------------------')
            disp('Model      Med. Bias (b)      IQR (b)           MSE (b)
                     Med. Bias (c)      IQR (c)           MSE (c)')
            disp(['1 step     ', num2str(es_1S(1:end),      '%+1.4f \t')])
            disp(['2 step     ', num2str(es_2S(1:end),      '%+1.4f \t')])
            disp(['CUE        ', num2str(es_CUE(1:end),     '%+1.4f \t')])
            disp(['JGMM       ', num2str(es_J1GMM(1:end),   '%+1.4f \t')])
            disp(['RJGMM      ', num2str(es_RJGMM(1:end),   '%+1.4f \t')])
            disp(['RJGMMo     ', num2str(es_RJGMMo(1:end), '%+1.4f \t')])
            disp(['RJIVE      ', num2str(es_RJIVE(1:end)', '%+1.4f \t')])
            disp('-----------------------------------------------------
                ------------------------------------------'
        end
    end
end


% View results
%output_table

% Save results
writetable(output_table, [data_form '_n'  num2str(n) '_reps' num2str(
    sim_max) '_sim.csv'])

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                                   END                                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

References

Alonso-Borrego, C. and Arellano, M. (1999). Symmetrically normalized instrumental-variable estimation using panel data. *Journal of Business & Economic Statistics*, 17(1):36–49.

Andrews, D. W. (1999). Consistent moment selection procedures for generalized method of moments estimation. *Econometrica*, 67(3):543–563.

Andrews, D. W. and Lu, B. (2001). Consistent model and moment selection procedures for GMM estimation with application to dynamic panel data models. *Journal of Econometrics*, 101(1):123–164.

Angrist, J. D. and Krueger, A. B. (1991). Does compulsory school attendance affect schooling and earnings? *The Quarterly Journal of Economics*, 106(4):979–1014.

Bekker, P. A. (1994). Alternative approximations to the distributions of instrumental variable estimators. *Econometrica: Journal of the Econometric Society*, pages 657–681.

Belloni, A., Chen, D., Chernozhukov, V., and Hansen, C. (2012). Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica*, 80(6):2369–2429.

Belloni, A., Chernozhukov, V., and Hansen, C. (2014). Inference on treatment effects after selection among high-dimensional controls. *The Review of Economic Studies*, 81(2):608–650.

Belloni, A., Chernozhukov, V., Hansen, C., and Kozbur, D. (2016). Inference in high-dimensional panel models with an application to gun control. *Journal of Business & Economic Statistics*, 34(4):590–605.

Blundell, R. and Bond, S. (1998). Initial conditions and moment restrictions in dynamic panel data models. *Journal of econometrics*, 87(1):115–143.

Breusch, T., Qian, H., Schmidt, P., and Wyhowski, D. (1999). Redundancy of moment conditions. *Journal of econometrics*, 91(1):89–111.

Canay, I. A. (2010). Simultaneous selection and weighting of moments in GMM using a trapezoidal kernel. *Journal of Econometrics*, 156(2):284–303.

Caner, M. (2009). LASSO-type GMM estimator. *Econometric Theory*, 25(1):270–290.

Carrasco, M. (2012). A regularization approach to the many instruments problem. *Journal of Econometrics*, 170(2):383–398.

Carrasco, M. and Nayihouba, A. (2017). A regularization approach to the dynamic panel data model estimation.

Carrasco, M. and Noumon, N. (2011). Optimal portfolio selection using regularization. Technical report, Discussion paper.

Carrasco, M. and Tchuente, G. (2015). Regularized liml for many instruments. *Journal of Econometrics*, 186(2):427–442.

Carrasco, M. and Tchuente, G. (2016). Efficient estimation with many weak instruments using regularization techniques. *Econometric Reviews*, 35(8-10):1609–1637.

Chamberlain, G. and Imbens, G. (2004). Random effects estimators with many instrumental variables. *Econometrica*, 72(1):295–306.

Chao, J. C., Swanson, N. R., Hausman, J. A., Newey, W. K., and Woutersen, T. (2012). Asymptotic distribution of jive in a heteroskedastic iv regression with many instruments. *Econometric Theory*, 28(1):42–86.

Cheng, X. and Liao, Z. (2012). Select the valid and relevant moments: A one-step procedure for GMM with many moments.

Cheng, X. and Liao, Z. (2015). Select the valid and relevant moments: An information-based LASSO for GMM with many moments. *Journal of Econometrics*, 186(2):443–464.

Dagenais, M. G. and Dagenais, D. L. (1997). Higher moment estimators for linear regression models with errors in the variables. *Journal of Econometrics*, 76(1):193–221.

Dominguez, M. A. and Lobato, I. N. (2004). Consistent estimation of models defined by conditional moment restrictions. *Econometrica*, 72(5):1601–1615.

Donald, S. G., Imbens, G. W., and Newey, W. K. (2009). Choosing instrumental variables in conditional moment restriction models. *Journal of Econometrics*, 152(1):28–36.

Donald, S. G. and Newey, W. K. (2000). A jackknife interpretation of the continuous updating estimator. *Economics Letters*, 67(3):239–243.

Fan, J. and Liao, Y. (2011). Ultra high dimensional variable selection with endogenous covariates. *Manuscript. Princeton University.*

Forni, M., Hallin, M., Lippi, M., and Reichlin, L. (2000). The generalized dynamic-factor model: Identification and estimation. *The review of Economics and Statistics*, 82(4):540–554.

Forni, M., Hallin, M., Lippi, M., and Reichlin, L. (2004). The generalized dynamic factor model consistency and rates. *Journal of Econometrics*, 119(2):231–255.

Fuhrer, J. C., Moore, G. R., and Schuh, S. D. (1995). Estimating the linear-quadratic inventory model maximum likelihood versus generalized method of moments. *Journal of Monetary Economics*, 35(1):115–157.

Gautier, E. and Tsybakov, A. (2011). High-dimensional instrumental variables regression and confidence sets. *arXiv preprint arXiv:1105.2454.*

Hall, A. R., Inoue, A., Jana, K., and Shin, C. (2007). Information in generalized method of moments estimation and entropy-based moment selection. *Journal of Econometrics*, 138(2):488–512.

Hall, A. R. and Peixe, F. P. (2003). A consistent method for the selection of relevant instruments. *Econometric Reviews*, 22(3):269–287.

Han, C. and Phillips, P. C. (2006). GMM with many moment conditions. *Econometrica*, 74(1):147–192.

Hansen, C., Hausman, J., and Newey, W. (2008). Estimation with many instrumental variables. *Journal of Business & Economic Statistics*, 26(4):398–422.

Hansen, C. and Kozbur, D. (2014). Instrumental variables estimation with many weak instruments using regularized jive. *Journal of Econometrics*, 182(2):290–308.

Hansen, L. P. (1982). Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the Econometric Society*, pages 1029–1054.

Hansen, L. P., Heaton, J., and Yaron, A. (1996). Finite-sample properties of some alternative GMM estimators. *Journal of Business & Economic Statistics*, 14(3):262–280.

Hansen, L. P. and Singleton, K. J. (1982). Generalized instrumental variables estimation of nonlinear rational expectations models. *Econometrica: Journal of the Econometric Society*, pages 1269–1286.

Hausman, J., Lewis, R., Menzel, K., and Newey, W. (2011). Properties of the cue estimator and a modification with moments. *Journal of econometrics*, 165(1):45–57.

Holtz-Eakin, D., Newey, W., and Rosen, H. S. (1988). Estimating vector autoregressions with panel data. *Econometrica: Journal of the Econometric Society*, pages 1371–1395.

Inoue, A. (2006). A bootstrap approach to moment selection. *The Econometrics Journal*, 9(1):48–75.

Kapetanios, G. and Marcellino, M. (2010). Factor-GMM estimation with large sets of possibly weak instruments. *Computational Statistics & Data Analysis*, 54(11):2655–2675.

Kézdi, G., Hahn, J., and Solon, G. (2002). Jackknife minimum distance estimation. *Economics Letters*, 76(1):35–45.

Kuersteiner, G. M. (2002). Rate-adapative GMM estimators for linear time series models.

Kuersteiner, G. M. (2012). Kernel-weighted GMM estimators for linear time series models. *Journal of Econometrics*, 170(2):399–421.

Liao, Z. (2013). Adaptive GMM shrinkage estimation with consistent moment selection. *Econometric Theory*, 29(5):857–904.

Luo, Y. and Chernozhukov, V. (2014). Selecting informative moments via LASSO.

Newey, W. K. and Smith, R. J. (2004). Higher order properties of GMM and generalized empirical likelihood estimators. *Econometrica*, 72(1):219–255.

Newey, W. K. and Windmeijer, F. (2009). Generalized method of moments with many weak moment conditions. *Econometrica*, 77(3):687–719.

Okui, R. (2011). Instrumental variable estimation in the presence of many moment conditions. *Journal of Econometrics*, 165(1):70–86.

Phillips, G. D. A. and Hale, C. (1977). The bias of instrumental variable estimators of simultaneous equation systems. *International Economic Review*, pages 219–228.

Shi, Z. (2016). Econometric estimation with high-dimensional moment equalities. *Journal of Econometrics*, 195(1):104–119.

Smith, R. J. (1997). Alternative semi-parametric likelihood approaches to generalised method of moments estimation. *The Economic Journal*, 107(441):503–519.

Stock, J. H. and Watson, M. W. (2002). Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics*, 20(2):147–162.

Stock, J. H. and Wright, J. H. (2000). GMM with weak identification. *Econometrica*, 68(5):1055–1096.

Stock, J. H. and Yogo, M. (2004). Asymptotic distributions of instrumental variables statistics with many instruments.