FUNCTIONAL ESTIMATOR SELECTION TECHNIQUES FOR PRODUCTION FUNCTIONS AND FRONTIERS

A Dissertation

by

JOSE LUIS PRECIADO ARREOLA

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Andrew Luke Johnson
Committee Members,	Yu Ding
	Natarajan Gautam
	Anirban Bhattacharya
Head of Department,	César Octavio Malavé

May 2016

Major Subject: Industrial Engineering

ABSTRACT

This dissertation provides frameworks to select production function estimators in both the state-contingent and the general monotonic and concave cases. It first presents a Birth-Death Markov Chain Monte Carlo (BDMCMC) Bayesian algorithm to endogenously estimate the number of previously unobserved states of nature for a state-contingent frontier. Secondly, it contains a Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm to determine a parsimonious piecewise linear description of a multiplicative monotonic and concave production frontier. The RJMCMC based algorithm is the first computationally efficient one-stage estimator of production frontiers with potentially heteroscedastic inefficiency distribution and environmental variables. Thirdly, it provides general framework, based on machine learning concepts, repeated learning-testing and parametric bootstrapping techniques, to select the best monotonic and concave functional estimator for a production function from a pool of functional estimators. This framework is the first to test potentially nonlinear production function estimators on actual datasets, rather than extrapolation of Monte Carlo simulation results.

DEDICATION

A mis papás, por su inmenso amor, esfuerzos e incondicional apoyo a su hijo. A mi hermano, por ser mi mejor amigo y quien mejor entiende mis curas y tristezas. A mi hermana, por ser una persona inteligente y en constante crecimiento.

English Translation:

To my parents, for their immense love, effort and unconditional support to their son. To my brother, for being my best friend and whom best understands my jokes and my grief. To my sister, for being a smart person who is in constant growth.

ACKNOWLEDGMENTS

I would first like to thank my advisor, Dr. Andy Johnson, for taking me as his student almost four years ago. His academic guidance, along with his great qualities both as a person, such as his patience and care for others, and as a researcher, such as his curiosity and openness to ideas, have resulted in many learnings. I will not deny this was a really harsh road to transit, but it would have been plainly impossible to do it without a mentor like you. Thanks also to my committee members, Dr. Ding, Dr. Gautam and Dr. Bhattacharya for taking the time to go over my work.

Next, I would like to thank my friends from the lab, current and past, who have contributed to make a great work and friendship ecosystem across these years. Jayanth, thanks for your deeply rooted human values, your sincerity, sense of responsibility and willingness to learn. Seokjun, it was my pleasure being around such an empathetic, loyal and intelligent person. Daisuke, for inspiring us with your outstanding analytical capabilities and for your humility despite them. Marcella, for all your positive energy, the true and deep way in which you care about people, your great ability to listen and for not being afraid of a challenging thesis. Kevin, for reminding me how hard things can be at the beginning and how much better they can go after. Hankyul, for being a great example of how to keep one's cool under pressure and showing what great patience in spite of a long project looks like. Anna Paula, for helping me index this thesis in such a selfless way. Anurag, for improving my management skills and being a good buddy at designing programming solutions. Mukilan and Rajesh, for the fun times and hard work at the beginnings of the NOV project.

Thanks to the good friends I made outside of the lab over these last three years. Axel, for being the best 15-day roommate in the history of the world, for motivating many discussions and laughs and for the very many shared stories. Selina, thanks for getting my humor on all of its levels, for being such an inclusive person and for all the bachata we made. Kristyna, for all the long and fun talks, for making my 30th birthday plain unforgettable and for being as hilarious and good-hearted of a person as they come. Daphne, for being a real-stuff escape for this liberal mind in this town over these last semesters. Thanks to Nick, because only through his help I could shed light on issues I thought were previously unsolvable and became a happier person. I look forward to tackle them for good. Thanks to Monica for her support, caring personality, and for being my long-distance partner in crime for our Ph.D adventures. To end this section, thanks to the relationships that regardless of their duration left a positive and joyful mark in these years.

A mis papás, quienes han sido una fuente de infinito apoyo, amor y consuelo desde que tengo memoria. Gracias por todas las tribulaciones y problemas que tuvieron que superar para que yo tuviera las oportunidades que tuve. Estoy seguro de que no hubiera llegado muy lejos sin ustedes y me siento increíblemente afortunado por todas las rarísimas circunstancias que tuvieron que darse para que estuvieran conmigo sanos y salvos en esta fase de la vida y espero que aún mejores tiempos estén por venir. Me siento extremadamente afortunado de haber tenido papás como ustedes y los amo. A mi hermano, por ser mi mejor amigo, por la habilidad que en conjunto tenemos para ir y venir a nuestra niñez, adolescencia y adultez en cualquier conversación. Por escucharme y entender qué tan difícil es el camino de mejorar, ya que tú mismo lo vives a diario. No te puedo explicar qué tan más feliz es mi existencia

porque estás tú en ella. A mi hermana, por crecer tanto y por ser inteligente y aterrizada en tu forma de vivir la vida. Tu crecimiento inspira mi crecimiento (Namasté).

A mis amadísimos abuelos, a mi tía Tencha y a mi nina Ana Laura quienes ya no están con nosotros, pero cuyo amor y esfuerzos para criarme bien fueron indispensables para convertirme en quién soy. De igual manera gracias a mis muy numerosos tíos, tías y primos, en especial a la Chomis, el Checho, la None, el Mashombre, Ana Maria y Maria Teresa.

English Translation of previous two paragraphs:

To my parents, who have been an infinite source of support, love and comfort. Thanks for all the struggle and the problems you had to overcome for me to have the opportunities I had. I am sure that I would have not gotten very far without you and I feel incredibly fortunate for all the exceptional circumstances that had to take place for you to be safe and sound with me at this stage of life. I also expect that even better times are to come. I feel extremely lucky of having had parents like you and I love you. To my brother, for being my best friend, for the joint ability we have to go back and forth to our childhood, teenage and adulthood in any given conversation. For listening to me and understanding how difficult the road to improvement is, as you transit it on a daily basis. I cannot explain you how happier my existence is because of your presence in it. To my sister, for growing so much, for being intelligent and down to earth in your way of living life. Your growth inspires my growth, Namaste.

To my immensely beloved grandparents, my great aunt Tencha and my godmother Ana Laura, who are not with us anymore, but whose love and effort to raise me well were

indispensable to become who I am. Likewise, thanks to my very numerous uncles, aunts and cousins, especially Chomis, Checho, None, Mashombre, Ana Maria and Maria Teresa.

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iii

ACKNOWLEDGMENTS iv

TABLE OF CONTENTS viii

LIST OF FIGURES xi

LIST OF TABLES xiii

CHAPTER I INTRODUCTION AND LITERATURE REVIEW 1

I.1 Literature Specific to Chapter II: Bayesian Model Selection and Model Estimation in a State-Contingent Production Framework 4
I.2 Literature Specific to Chapter III: Multivariate Bayesian Constrained Regression with Inefficiency and Flexible Variances 7
I.3 Literature Specific to Chapter IV: Evaluating Monotonic and Concave Production Function Estimators for Use on Manufacturing Survey Data 12

CHAPTER II ENDOGENOUS ESTIMATION OF THE NUMBER OF STATES ON A STATE-CONTINGENT PRODUCTION FRONTIER 18

II.1 Model 19
II.1.1 Production Frontier Regression Model 20
II.1.2 Bayesian Hierarchical Model 21
II.1.3 Prior Probability Distributions 23
II.1.4 Likelihood Function 24
II.1.5 Conditional Posterior Distributions 25
II.1.6 Bayesian Inference via BDMCMC 26
II.2 Empirical Application 28
II.2.1 Prior Parameter Values 30
II.2.2 Posterior Parameter Estimates and Interpretation 32
II.2.3 Incorporating Monotonicity and Convexity Constraints into the Frontier Model 41
II.2.4 Goodness of Fit 41
II.2.5 Sensitivity Analysis 44
II.2.6 Label-Switching 48

CHAPTER III A ONE-STAGE MULTIVARIATE SEMI-NONPARAMETRIC BAYESIAN CONCAVE REGRESSION METHOD TO FIT STOCHASTIC PRODUCTION FRONTIERS 50

III.1 Methodology 52
III.1.1 Production Frontier Model 52
III.1.2 Multivariate Nonparametric Bayesian Concave Regression 52
III.1.3 A Multiplicative Production Frontier 55
III.1.4 The Proposed MBCR-I Algorithm 58
III.1.5 Additional Computational Considerations due to Inefficiency Modeling 60
III.1.6 MBCR-I as a One-Stage Estimator for Stochastic Frontiers 61
III.2. Monte Carlo Simulations 62
III.2.1 Evaluation Based on Four Data Generation Processes 64
III.2.2 Discussion of Simulation Results and Recommendations to Use MBCR-I 75
III.3. Extensions 77
III.3.1 Flexible Time Trend 77
III.3.2 Contextual Variables 78
III.4 Empirical Application 79
III.4.1 Posterior Estimated Frontier and Interpretation 81

CHAPTER IV EVALUATING MONOTONIC AND CONCAVE PRODUCTION FUNCTION ESTIMATORS FOR USE ON MANUFACTURING SURVEY DATA 87

IV.1 Convex Adaptively Partitioned Nonparametric Least Squares 88
IV.1.1 Production Function Model 88
IV.1.2 Convex Adaptively Partitioned Nonparametric Least Squares 89
IV.1.3 CAP-NLS as a Series of Quadratic Programs 93
IV.2 Experiments on Simulated Data 94
IV.2.1 Bivariate Input Cobb-Douglas DGP 97
IV.2.2 Trivariate Input Cobb-Douglas DGP 101
IV.2.3 Four-variate Input Cobb-Douglas DGP 105
IV.2.4 Insights and Implications from Examples IV.2.1 - IV.2.3 108
IV.2.5 Scalability of CAP-NLS to Larger Datasets 109
IV.3 Chilean Annual National Industrial Survey 111
IV.3.1 Dataset and Considerations 111
IV.3.2 Methodology to Compare Functional Estimator Performance on Real Data 112
IV.3.3 Functional Estimator Comparison Results 115
IV.3.4 Replicate-Specific Insights for Predictive Error 120
IV.3.5. Estimator Performance Measures as Function of Subsample Size and Surveying Implications 120

CHAPTER V CONCLUSIONS 125

V.1 Conclusions Specific to Chapter II 125
V.2 Conclusions Specific to Chapter III 127
V.3 Conclusions Specific to Chapter IV 129

REFERENCES 132

APPENDIX A CONDITIONAL POSTERIOR HYPERPARAMETERS FOR CHAPTER II 143

APPENDIX B TRACE PLOTS OF ESTIMATED PARAMETERS FOR CHAPTER II 144

APPENDIX C FULL SENSITIVITY ANALYSIS AND GOODNESS OF FIT RESULTS FOR CHAPTER II 150

APPENDIX D FULLY BAYESIAN MCBR-I 155

APPENDIX E FRONTIER CHARACTERISTICS FOR 2008-2009 CROSS SECTIONAL DATASETS 157

APPENDIX F THE FLEXIBILITY TRADEOFF BETWEEN PRODUCTION FUNCTION AND INEFFICIENCY DISTRIBUTION ASSUMPTIIONS 159

APPENDIX G A COARSE GRID SEARCH ALGORITHM TO COMPUTE MPSS ON A MULTIVARIATE SETTING 162

APPENDIX H RESULTS FOR HOMOSCEDASTIC ADDITIVE ERROR SIMULATED DATASETS 163

APPENDIX I PARAMETRIC BOOTSTRAP ALGORITHM TO CALCULATE EXPECTED OPTIMISM 181

APPENDIX J COBB-DOUGLAS RESULTS WITH MULTIPLICATIVE RESIDUAL ASSUMPTION FOR CHILEAN MANUFACTURING DATA 183

LIST OF FIGURES

Page

Figure 1 Directed acyclic graph for the hierarchical model used for parameter estimation. All parameters to be inferred are in circles. 23

Figure 2 Posterior distribution for the number of components in the mixture for dummy time trend. 33

Figure 3 Dummy trend compared against the state-contingent linear time trends. 40

Figure 4 Goodness of fit results for dummy model. Results for full posterior scenario (top left panel), 4% outliers removed for full posterior scenario (top right panel), results for mode scenario(bottom left panel), 4% outliers removed for mode scenario (bottom right panel). 46

Figure 5 Rankings of the three alternative prior TE lower bound assumptions against the base value of 0.7 for the dummy trend model. 47

Figure 6 Technical Efficiency distribution for low (top panel), base (middle panel) and high (bottom panel) values of τ^* for the dummy trend model. 47

Figure 7 Distribution of intercept (expected output at time 0) for three states, the mode of J. Fitted curves for dummy time trend model (left panel), linear time trend model results (right panel). Solid line denotes distribution for state 1 intercept, dashed line denotes distribution for state 2 intercept and dotted line denotes distribution for state 3 intercept. 49

Figure 8 Three basis regions defined by the current number of hyperplanes $K^{(t)}=3$ (top left); proposed basis regions for removal of the second hyperplane (top right); proposal basis regions for hyperplane addition and split of third basis region (bottom left); and, proposal basis regions for hyperplane addition and split of third basis region using a different splitting knot and the same splitting direction (bottom right). 54

Figure 9 Bivariate input Cobb-Douglas DGP results for small noise settings 99

Figure 10 Bivariate input Cobb-Douglas DGP results for large noise settings 101

Figure 11 Trivariate input Cobb-Douglas DGP results for small noise settings 103

Figure 12 Trivariate input Cobb-Douglas DGP results for large noise settings 104

Figure 13 Four-variate input Cobb-Douglas DGP results for small noise settings 107

Figure 14 Four-variate input Cobb-Douglas DGP results for large noise settings 108

Figure 15 Replicate-specific results on predictive component of R^2_{FS} for varying degrees of CAP-NLS-related estimation improvement..... 122

Figure 16 Best method's R^2_{FS} as function of relative subset size for selected industries. CAP-NLS was chosen as best method for industry codes 2899, 2010, 2811 and 2520, while CDA was chosen for industry code 1541. 124

LIST OF TABLES

Page

Table 1 Summary Statistics 29

Table 2 Summary of Prior Values for Model Hyperparameters 31

Table 3a Estimated Production Frontier Coefficient Means for BDMCMC Linear, Dummy Time Trend and O'Donnell and Griffiths (2006) Models at the Posterior Mode J=3 36

Table 3b Estimated Production Frontier Coefficient Standard Deviations for BDMCMC Linear, Dummy Time Trend and O'Donnell and Griffiths (2006) Models at the Posterior Mode J=3 37

Table 4 MSE, RMSE, %RMSE Comparison Between Different Models 43

Table 5 Sensitivity Analysis on Hyperparameter λ, Prior Mean on the Number of States for the Dummy Time Trend Model 44

Table 6 Results for Example 1: Univariate Cobb-Douglas Frontier with Homoscedastic Inefficiency Terms 65

Table 7 Estimator Robustness Analysis for Example 1: Univariate Cobb-Douglas Frontier with Homoscedastic Inefficiency Terms 66

Table 8 Results for Example 2: Bivariate Cobb-Douglas Frontier with Homoscedastic Inefficiency Terms 68

Table 9 Estimator Robustness Analysis for Example 2: Bivariate Cobb-Douglas Frontier with Homoscedastic Inefficiency Terms 69

Table 10 Results for Example 3: Trivariate Cobb-Douglas Frontier with Homoscedastic Inefficiency Terms 70

Table 11 Estimator Robustness Analysis for Example 3: Trivariate Cobb-Douglas Frontier with Homoscedastic Inefficiency Terms 72

Table 12 Results for Example 4: Trivariate Cobb-Douglas Frontier with Heteroscedastic Inefficiency Terms 74

Table 13 Estimator Robustness Analysis for Example 3: Trivariate Cobb-Douglas Frontier with Heteroscedastic Inefficiency Terms 74

Table 14 Descriptive Statistics for Concrete Products Dataset 80

Table 15 MBCR-I S Cross Sectional and Panel Production Frontier Fitting Statistics... 82

Table 16 MBCR-I S Cross Sectional Production Frontier Characterization for 2007 83

Table 17 MBCR-I S Cross Sectional Production Frontier Characterization for 2010 84

Table 18 MBCR-I Panel Production Frontier Characterization 84

Table 19 Time Dummy Variable Coefficients..... 85

Table 20 Most Productive Scale Size for Selected Capital/Labor Ratios for Cross Sectional and Panel Production Frontiers 86

Table 21 Number of Hyperplanes and Runtime for Bivariate Input Cobb-Douglas DGP 99

Table 22 Number of Hyperplanes and Runtime for Trivariate Input Cobb-Douglas DGP 104

Table 23 Number of Hyperplanes and Runtime for Four-variate Input Cobb-Douglas DGP..... 106

Table 24 Number of Hyperplanes and Runtime for Trivariate Input Cobb-Douglas DGP on Larger Datasets..... 111

Table 25 Method Comparison Across the 5 Largest Sampled Industries from the Chilean Annual National Industrial Survey 118

Table 26 Ratio of CDA to Best Model Performance..... 119

Table 27 Most Frequently Selected Best Method for Different Sample Size Ranges... 119

CHAPTER I*

INTRODUCTION AND LITERATURE REVIEW

Efficiency Analysis is a framework under which we measure the ability a firm has to produce a particular output bundle given a set of inputs and environmental conditions. More specifically, we are interested in benchmarking the firm's performance in the production of these outputs (Coelli *et al.* 2005), as compared to the optimally attainable output levels, costs or profits. The relative performance of a firm, depending on the comparison criteria, is called either *Technical Efficiency* in the case of output levels or *Allocative Efficiency* in terms of picking the best mix of outputs and inputs. The concepts of Technical Efficiency and price/cost related efficiencies relative to an optimal estimated standard were first explored by Debreu (1951) and Farrell (1957). If we focus on comparing output levels, we can describe the feasible input-output combinations in terms of a set, called the *production possibility set* (Koopmans 1951). Moreover, in the case where we consider outputs to be aggregable into a single output, its maximum level for all given input levels can be summarized in the form of a function. This function is known as a *production function* or the *production frontier*. Thus, as this function is not directly observed, we need to estimate it with a dataset of firms that are producing the same output and using the same input bundle.

---

* Section I.1 reprinted with permission from "A birth-death Markov Chain Monte Carlo Method to Estimate the Number of States in a State-Contingent Production Frontier Model" by Preciado Arreola, J.L. and A. L. Johnson, 2015. *American Journal of Agricultural Economics*, 1267-1285, Copyright by American Agricultural Economics Association.

Historically, there have been two frameworks to estimate production frontiers. On one side, Data Envelopment Analysis (DEA) (Charnes, Cooper and Rhodes, 1978) is a nonparametric technique that fits a convex hull to the observed input-output data. DEA does not assume the functional form of the production frontier *a priori* and it results in a function described by a set of enveloping hyperplanes. Classic Stochastic Frontier Analysis (SFA) (Aigner, Lovell and Schmidt, 1977; Meeusen and van den Broeck, 1977), on the other hand, estimates the parameters of a functional form chosen by the researcher. More recently, novel estimation procedures have been developed to overcome the inherent shortcomings of both DEA and the classic parametric SFA. Specifically, DEA has been shown to be a special case of Stochastic Nonparametric Envelopment of Data (StoNED) method, Kuosmanen and Johnson (2010) and Kuosmanen and Kortelainen (2012). Moreover, the StoNED method specifies using a shape constrained nonparametric estimation method in the first stage, and thus generalizing classic parametric SFA.

However, in spite of relaxing assumptions such as the deterministic nature of DEA and the parametric forms of classical SFA, these and other recent production function and frontier estimation methods have generally not addressed basic model selection issues. In other words, either parametric functional estimators are without theoretical justification or nonparametric methods have been proposed without concern for their parsimony, thus affecting their predictive ability, or even their in-sample fit[1]. Thus, the objective of this research is to expand the literature on model selection strategies for production

[1] See Chapter 7 of Hastie, Tibshirani and Friedman (2009) for a discussion of overfitting and its consequences on in-sample and predictive error of a model.

function/frontier functional estimators. We note that except for the specific state-contingent production function/frontier application in Chapter 2, the remaining functional estimators and framework represent not only improvements, but the first explicit efforts to incorporate model selection strategies to production functions and frontiers.

This dissertation is organized as follows: in the remaining sections of this chapter, we present the literature review specific to chapters II, III, IV and V. In Chapter II, we propose a one-stage method to estimate state-contingent production frontiers which simultaneously estimates the state-contingent frontier descriptions and the number of states of nature, thus endogenously controlling the complexity of the functional estimator, under the assumption of a Translog parametric production function. The method is introduced for the purposes of analyzing rice-farm performance. In Chapter III, we present a one-stage shape-constrained semiparametric stochastic frontier method, which results in more parsimonious and economically sound frontiers than state-of-the art estimators. The method in Chapter III augments both the Bayesian SFA and the general SFA literature by being the first shape-constrained model to include highly flexible descriptions of both the production function and the inefficiency distributions while maintaining scalability in the number of observations used for estimation. In Chapter IV, we first present an adaptively-partitioned functional estimator that improves the in-sample and predictive performance of existing methods. Then, we proceed to introduce a general framework to evaluate production function estimators on real manufacturing survey data. In Chapter V we conclude and summarize the lines of research that are now open.

I.1 Literature Specific to Chapter II: Bayesian Model Selection and Model Estimation in a State-Contingent Production Framework

State-contingent production provides a flexible framework for estimating production functions (Quiggin and Chambers 2006), particularly in industries with subsets of firms that operate in environments lacking data at the producer level. Using rice farming as an example, data on each farm's rainfall, temperature, pests, diseases, and other environmental factors are potentially observable, but rarely collected at the farm level. It is possible, however, to employ state-contingent production frontiers to separate the impacts of inefficiency and state-dependent production conditions by allowing different parameters for each state of nature. The simultaneous estimation of production frontiers, using Stochastic Frontier Analysis (SFA) for example, and creation of unobserved variable clusters, needed to define the states of nature, can be addressed by using Latent Class Stochastic Frontier Models (LCSFM), either from a sampling framework, an example of which appears in Orea and Kumbhakar (2004), or from a Bayesian framework via Markov Chain Monte Carlo (MCMC) methods. We select the Bayesian framework, because it allows us to obtain the complete posterior distribution of the number of states and to calculate weights for our output predictions. Moreover, MCMC methods make it relatively easy to add common constraints, such as monotonicity in inputs, to the production frontier estimation algorithm, and to model the observed outputs as the outcome of a finite mixture of Gaussian distributions.

Several uses of the state-contingent approach have been described in the production economics literature. Chavas (2008) develops an input cost-minimizing methodology under a state-contingent technology. Nauges, O'Donnell and Quiggin (2010) use the state-contingent approach and MCMC methods to model production with Constant Elasticity of Substitution

(CES) flexible production functions. Other studies include O'Donnell and Shankar (2010) and Serra and Lansink (2010). The work most closely related to ours is O'Donnell and Griffiths (2006), who estimate a state-contingent frontier model using MCMC methods, specifically a Gibbs sampler, for the same data set we analyze. Further, O'Donnell and Griffiths determine the number of states of nature by minimizing the Bayesian Information criterion (BIC) for a model in which only the intercepts of the production function and the precision parameters (the inverses of the variances of the noise distribution) are state-contingent. Then they impose the BIC-minimizing number of states to a model for which the input slopes are also state-contingent and monotonicity of the production function is imposed. We caution, however, that the BIC-minimizing number of states can differ across models, since the number of parameters increases for models in which the state-contingent input slopes are estimated. In other words, the number of states imposed may be inappropriate even though imposing the number of states calculated for a simpler model is computationally easier.

Further, in the frontier-estimation models discussed above, the number of states is either fixed or obtained by using BIC as model selection criterion. Biernacki, Celeux and Govaert (1998) observe that the assumptions needed in order to use BIC as a model selection criterion, do not hold for mixture models. For example, the needed assumption of all model parameters being well inside of the parameter space is violated when evaluating models for which the number of states is larger than the true one, as state probabilities for some states will approach zero which is a boundary value for a probability parameter, making BIC inappropriate. Biernacki, Celeux and Govaert (2000) and Biernacki and Govaert (1997) show that other criteria, such as the Integrated Classification Likelihood (ICL) and the

Classification Likelihood Criterion (CLC), are often better suited for this type of application. Banfield and Raftery (1993) show that Approximate Weight of Evidence (AWE), derived as a Bayes Factor approximation, can be used to compute the posterior probabilities for each number of states given the data. It is also possible to use these posterior probabilities to weight the chosen model, instead of simply choosing one with the best score. Regardless of the model selection criterion chosen, a two-stage procedure is required, i.e. the first stage estimates all plausible models, and the second stage scores and obtains the posterior probabilities to weight the models.

In contrast, endogenous estimation of the number of states involves solving the model selection and estimation problems simultaneously. Specifically, selection is inferring the number of states in the model and estimation is calculating the coefficients for the regression models given a certain number of states. Both endogenous estimation and the model selection criteria mentioned above allow identification of the best model; however endogenous estimation also calculates weighted output across all plausible models in a single run of the estimation algorithm. Moreover, although in both endogenous estimation and the two-stage procedure described above the range of possible values for the number of states has to be specified a priori, in the endogenous estimation algorithm presented in Chapter II, only the plausible values of this range are visited by the estimation algorithm significantly reducing computational complexity.

Our survey of the literature indicates that efforts to endogenously estimate the number of components in a Gaussian mixture in an econometric context have been limited. For example, in a macroeconomic setting Chopin and Pelgrin (2004) develop a general hidden Markov chain and estimate all transition probabilities where the number of transition

probabilities grows quadratically in the number of states. Other methods can be used to estimate a mixture model where the number of components is unknown, such as Reversible Jump Markov Chain Monte Carlo (RJMCMC) introduced by Green (1995) and first applied to mixture modeling by Richardson and Green (1997). Nevertheless, Richardson and Greene's RJMCMC algorithm is not directly applicable in a regression context, because the equations for the parameter means become datapoint specific and cause estimation difficulties (Kottas and Gelfland 2001). Some efforts to tailor RJMCMC algorithms within a regression context have been successful, such as Denison, Mallick and Smith (1998) and Biller (2000).

However, we select the birth-death Markov Chain Monte Carlo (BDMCMC) method (Stephens 2000), which was first used in a regression context by Hurn, Justel and Robert (2003). BDMCMC computes the rates of a simpler hidden Markov Chain, which they characterize as a birth and death process. By reducing the number of parameter estimates, the use of BDMCMC is better suited for applications with limited data. We note also that the computational process becomes simpler compared to RJMCMC, because the use of BDMCMC reduces the types of jumps across states, while still achieving consistent estimation results.

I.2 Literature Specific to Chapter III: Multivariate Bayesian Constrained Regression with Inefficiency and Flexible Variances

In Chapter III, we move away from the very specific state-contingent context to a general monotonic and concave production function. Further, we strive to achieve greater flexibility than allowed by the Translog function in our production frontier estimation. The

estimation of flexible production, cost, and utility functions that globally satisfy certain second order shape restrictions consistent with economic theory, such as monotonicity and convexity/concavity, remains challenging (Diewert and Wales, 1987). Isotonic Regression, Constraint Weighted Bootstrapping, and Data Sharpening have been devised to impose such restrictions (Henderson and Parmeter, 2009), but no solution methods so far are able to both impose concavity and handle very large datasets. Motivated by the challenge, we are also interested in models that allow for firms to make errors in optimization and thus model firm inefficiency. Because the inefficiency is not directly measurable, we use a Stochastic Frontier Analysis (SFA) framework (Aigner et al., 1977). Recently, several non-parametric estimators that include inefficiency, such as Kumbhakar et al. (2007)'s estimator, Stochastic Nonparametric Envelopment of Data (Kuosmanen and Kortelainen, 2012), and Constraint Weighted Bootstrapping (Du, Parmeter and Racine, 2013) have been devised, but even these methods are limited by computational issues or homoscedasticity assumptions on the inefficiency term or both. To the best of our knowledge, Kumbhakar et al. (2007) is the only frontier production function estimation method that allows the inefficiency term to be heteroscedastic without additional observable variables that predict inefficiency; however, this method does not impose shape constraints.

A survey of the relevant literature indicates that Banker and Maindiratta (1992) are the first to propose an estimator with second order shape restrictions and a composed error term; however, their maximum likelihood methods are only tractable for small instances and no applications or Monte Carlo studies of their method have been reported. Allon et al. (2005) use entropy methods, but little is known about their estimator's computational performance for datasets of more than a few hundred observations. Least Squares

approaches, such as Convex Nonparametric Least Squares (CNLS) (Kuosmanen, 2008) or the Penalized Least Squares splines approach by Wu and Sickles (2013), which allow estimation of production frontiers with minimal assumptions about the residual term require a separable and homoscedastic distribution for the inefficiency term.[2] Constraint Weighted Bootstrapping (CWB) (Du, Parmeter and Racine (2013)) can impose a vast array of derivative-based constraints, including both monotonicity and concavity. The concavity constraints in CNLS and CWB require satisfying $O(n^2)$ constraints simultaneously which is computationally difficult (Lee et al. 2013). Moreover, their use in a stochastic frontier setting requires a two-stage procedure, such as Kuosmanen and Kortelainen (2012), which does not allow a feedback structure between the frontier estimation procedure and the inefficiency estimation procedure. Thus, prior distributional assumptions about the inefficiency distribution, namely the family of distributions and homoscedasticity, are imposed throughout the frontier estimation procedure.

An alternative two-stage procedure by Simar and Zelenyuk (2011) adapts the Local Maximum Likelihood method developed by Kumbhakar et al. (2007) to shape constrained estimation by using Data Envelopment Analysis on the fitted values obtained from the Local Maximum Likelihood method. The approach is constrained by the Local Maximum Likelihood method, and scalability appears to be limited to a few hundred observations (Kumbhakar et al., 2007).

[2] While the homoscedastic assumption can be relaxed by using a specific parametric form on heteroscedasticity, see Kuosmanen and Kortelainen (2012), general models of heteroscedasticity are not available.

The first Bayesian SFA semiparametric estimators in the general multivariate setting proposed by Koop and Poirier (2004) and Griffin and Steel (2004) do not address the imposition of a concavity constraint on the estimations. O'Donnell and Coelli (2005) impose homogeneity, monotonicity, and convexity in inputs on a parametric multi-output, multi-input production frontier for a Panel dataset by means of a Metropolis-Hastings (M-H) random walk algorithm and restricting the Hessian matrix. While the approach is feasible in the parametric setting, a nonparametric equivalent requires numerical estimates of the production function derivatives, similar to Du et al. (2013), and significant computational effort.

When shape restrictions between the dependent variable and each regressor are imposed separately, Shively et al. (2011) estimate Bayesian shape-constrained nonparametric regressions using fixed and free-knot smoothing splines. The knots are endogenously inferred using a Reversible Jump Markov Chain Monte Carlo (RJMCMC) (Green 1995) algorithm. Unfortunately, the method can result in complex and numerous conditions, and consequently low acceptance rates within the parameter-sampling rejection algorithm. Furthermore, the RJMCMC algorithm cannot be directly extended to a general multivariate setting. Meyer, Hackstadt and Hoeting (2011), who sample from a set of basis functions for which the imposition of concavity constraints only relies on the non-negativity of their coefficients, thus reducing complexity, avoid rejection sampling or M-H methods. Nevertheless, the monotonicity and concavity constraints are still imposed only separately between each regressor and the dependent variable and a partial linear model form is assumed for the multivariate case.

The only Bayesian semi-nonparametric constrained method for a general multivariate context is the Neural Cost Function (Michaelides et al. 2015). Even though it can impose shape restrictions a priori to estimate a cost function, the method relies on an exogenous model selection criterion to select the number of intermediate variables, focuses on an average cost function rather than a frontier, and potentially has an overfitting problem due to near perfect correlation between predicted and actual costs.

Hannah and Dunson (2013) and Hannah and Dunson (2011) propose two adaptive regression-based multivariate nonparametric convex regression methods for estimating conditional means: Least-Squares based Convex Adaptive Partitioning (CAP), and a Bayesian method, Multivariate Bayesian Convex Regression (MBCR), both of which scale well in large data. Unlike CAP, the Markov Chain Monte Carlo nature of MBCR (Hannah and Dunson, 2011, henceforth H-D) allows to create extensions of the method in a modular manner and without risking its convergence guarantees. MBCR approximates a general convex multivariate regression function with the maximum value of a random collection of hyperplanes. Additions, removals, and changes of proposed hyperplanes are done through a RJMCMC algorithm. MBCR's attractive features include the block nature of its parameter updating, which causes parameter estimate autocorrelation to drop to zero in tens of iterations in most cases, the ability to span all convex multivariate functions without need for any acceptance-rejection samplers, scalability to a few thousand observations, and relaxation of the homoscedastic noise assumption.

I.3 Literature Specific to Chapter IV: Evaluating Monotonic and Concave Production Function Estimators for Use on Manufacturing Survey Data

Despite proposing more parsimonious alternatives than existing methods for frontier estimation in Chapter III, an explicit general framework for model selection in production frontiers does not exist. To accomplish this, we shift to the two-stage production frontier estimation paradigm and focus on model selection on the estimation of production functions. Moreover, due to computational challenges of our semiparametric Bayesian approach in small sample sizes, we demonstrate our framework on frequentist estimators of both parametric and nonparametric nature. Our model selection framework is based on basic machine learning concepts and algorithms.

Key concepts from the machine learning literature such as sample-specificity and prediction error over-optimism have been critically overlooked in the assessment of production functions. The importance of these issues becomes immediately evident when estimating a production function using non-exhaustive survey data.[3] These machine learning concepts allow us to investigate the extent to which an estimated production function characterizes both the set of surveyed establishments and the set of un-surveyed establishments for a particular industry. Classical frontier estimators such as Stochastic Frontier Analysis (SFA, Aigner et al., 1977), and Data Envelopment Analysis (DEA, Banker et al., 1984), as well as more recent developments such as Stochastic DEA (Simar and Zelenyuk, 2008), or Convex Nonparametric Least Squares (CNLS, Kuosmanen, 2008),

[3] The U.S. annual survey of manufacturers is conducted annually, except for years ending in 2 and 7 in which a Census is performed, Foster et al. (2008). This survey includes approximately 50,000 establishments selected from the census universe of 346,000 or approximately a 15% sampling, Fort et al. (2014).

Constraint Weighted Bootstrapping (CWB, Du et al., 2013) and Shape-Constrained Kernel Weighted Least Squares (SCKLS, Yagi et al., 2016) have all justified their use based on Monte Carlo simulation results for production functions estimated from random samples drawn from a known data generation process (DGP) and evaluated estimator performance on the same sample with which they were estimated.[4] The literature provides independent comparisons between some of these methods considering a more ample set of scenarios in a Monte Carlo simulation framework, but still using the same dataset to fit the production function/frontier and test its goodness-of-fit, see Andor and Hesse (2012) as an example. Furthermore, numerous applied studies to fit production functions and frontiers with real data have been conducted using the aforementioned methods without assessing estimator sample-specificity or even comparing the performance of multiple estimators on the actual application datasets.[5] Finally, the Monte Carlo simulations used to illustrate the performance of existing estimators do not explore the accuracy with which the estimator predicts the noise level of the DGP, which further obscures insights about the performance of a functional estimator.

[4] A body of economic literature of a less computational and more aggregate nature is the growth accounting literature, Solow (1957) and Barro and Sala-i-Martin (2004). These methods rely on price information, the cost minimization assumption and parametric forms to deterministically compute the coefficients of a first order approximation of a general production function using observed input cost shares (see for example Syverson, 2011). This literature's main model adequacy check is to compute the R-squared value on the full dataset of interest, again forgoing insights about the adequacy of the estimator in survey to full sample data or in less aggregate datasets with noisier input cost share measurements.

[5] For DEA-based studies, see the survey paper by Emrouznejad, Parker and G. Tavares (2008). For relevant SFA applications see Section 2.10 on Greene (1993). Practical studies involving more recent methods include Mekaroonreung and Johnson (2012), Eskelinen and Kuosmanen (2013), among others. This is also common in the productivity literature, see for example Olley and Pakes (1997), Levinsohn and Petrin (2003) and De Loecker et al. (2015).

Thus, we consider a model selection strategy that improves on the current paradigm for selecting a production function model among a pool of models generated by different functional estimators both on simulated and the actual application datasets. The common framework we will follow in both simulated and real data encompasses three elements: estimation of the optimism-corrected in-sample error (defined in Section 3) for the observed establishment set, use of a learning set-testing set context to estimate the predictive error on the unobserved establishment set (Hastie, Tibshirani and Friedman, 2009 pp. 222) and a finite-sample weighting, which acknowledges the existence of a finite set of establishments and thus weights the in-sample and predictive errors proportionally to the survey size. For simulated datasets, we take advantage of the practically infinite data-generating capability the Monte Carlo context. We estimate the expected optimism-corrected in-sample error by fitting a learning set and then evaluating the fitted function error on data drawn from the same DGP and restricting input vectors to the set observed in the learning set, but with potentially different output vectors due to random noise. To estimate the predictive error, we compute the mean-squared error for production functions fitted from random learning datasets on large, previously unobserved, testing sets. This is an important measure, providing estimates of the functional estimator's expected predictive error for an arbitrary unobserved establishment. Additionally, we also compute the performance against the known true DGP in both the in-sample and learning-to-testing set contexts.

In case of the manufacturing survey data, for which we do not know the data generating process of the production function a priori, we have to follow a different approach to estimate the expected in-sample and predictive errors. For the in-sample error, given the different natures of the considered estimators, we cannot simply compute Mallow's - C_p

(1973), Akaike Information Criterion (AIC, 1974) or similar optimism penalization, which is specific to linear models. Thus, we employ the parametric bootstrap approach by Efron (2004) to estimate in-sample optimism for potentially non-linear functional estimators. For our predictive error estimation, we estimate production functions using random subsamples of the survey data and assess the predictive error of the fit, thus following a cross-validation strategy (see Stone, 1974; Allen, 1974 and Geisser, 1975 for seminal work on cross-validation). The universal nature of cross-validation is key in comparing production function estimators (Arlot and Celisse, 2010), as they usually arise from different regression paradigms and have different sets of assumptions. More specifically, Hastie, Tibshirani and Friedman (2009, pp. 230) mention that when evaluating nonlinear, adaptive regression techniques[6] it is usually difficult to estimate the effective number of parameters, and cross-validation is one of the few available model selection strategies available. There exist several cross-validation methods, among which two frequently used non-exhaustive methods, k-fold cross-validation and repeated learning-testing (RLT) (Breiman et al, 1984) have significant computational advantages over methods that exhaustively explore all the possible combinations of learning and validation sets, such as leave-one out cross-validation (Stone, 1974, Geisser, 1975) or leave-p-out cross-validation (Shao, 1993). Unlike k-fold cross-validation, the variance of RLT can be controlled by increasing the number of replicates given any learning set size (Burman, 1989). A key aspect of our functional estimator selection strategy is that we do not use RLT to estimate the predictive error of a learning set

[6] All the nonparametric estimators we will evaluate on this chapter fall into that category. More generally, Burman (1989) shows that k-fold cross-validation has $O(n^{-2})$ bias against the true predictive error on a general model selection problem after applying a simple correction described in the application section.

of the size of our census set, as is normally the objective on cross-validation, but rather to obtain the expected predictive error for a dataset of the size of the learning set itself, this variance-controlling feature is the main feature of interest. Thus, we proceed to evaluate the fits of our different production function estimators on a RLT cross-validation scheme.

Furthermore, we use our model selection strategy to assess the required survey sizes to obtain reliable production function estimates for each of the studied industries. This type of insight is valuable to the Census Bureaus that need to determine the size of the surveys in the years in which Censuses are not performed. Development of an optimism-corrected model selection method which allows the evaluation of estimator performance in both simulated datasets and the actual manufacturing survey data is the main contribution of Chapter IV. Additionally, evaluation of existing estimators on simulated data resulted in valuable general insights about the different estimators' performance and led to the development of a novel nonparametric shape-constrained production function estimator. We describe this estimator which performs robustly on a learning set – testing set environment. Further, we argue that relative performance of an estimator on the real application dataset should be the main criterion to follow when choosing a production frontier estimation for that dataset, as the application may have characteristics that favor the use of that particular estimator.

The functional estimator we propose as an additional contribution to our model selection framework, Convex Adaptively Partitioned Nonparametric Least Squares (CAP-NLS), integrates the idea of adaptive partitioning from CAP (Hannah and Dunson, 2013) with the global optimization strategy of the CNLS estimator. The CNLS estimator is an example of a sieve estimator which is extremely flexible and is optimized to fit the observed

data set, White and Wooldridge (1991) and Chen (2007). Alternatively, the adaptive least squares-based CAP developed in the machine-learning literature has demonstrated good predictive performance by integrating model estimation and selection strategies, (and thus resulting in parsimonious functional estimates) as opposed to only optimizing fit on the observed dataset. Specifically, Hannah and Dunson (2013) recognize that the CNLS estimator overfits the observed dataset at the boundaries of the data, thus affecting the quality of prediction for the true underlying function. Other researchers, such as Huang and Szepesvári (2014) and Balázs, György and Szepesvári (2015) build examples in which CNLS estimation results in infinite Mean Squared Error due to overfitting of the sample.

CHAPTER II*

ENDOGENOUS ESTIMATION OF THE NUMBER OF STATES ON A STATE-CONTINGENT PRODUCTION FRONTIER

This chapter is the first work to endogenously estimate the number of states in a state-contingent stochastic production frontier, as well as the first to perform a goodness of fit analysis on a state-contingent production frontier model. We apply the proposed BDMCMC method to a case study of 44 rice farms in the Philippines operating between 1990 and 1997, which were investigated in Villano, O'Donnell and Battese (2004) and later in O'Donnell and Griffiths (2006). In addition, we relax the linear time trend assumption of the models in O'Donnell and Griffiths (2006) for changes in production over time, to consider a model with a state-independent dummy time trend, which more accurately separates the frontier shifts over time from state-contingent yield variability, inefficiency, and noise. We discuss the implications of the use of both the linear time trend and dummy time trend models on the estimated posterior parameters, the relevant economic interpretations, and the existence of state-specific production. We describe the use of Mean Squared Error (MSE) and visual tools, such as Quantile-Quantile (QQ) plots, to compare the performance of a model using the full distribution of the number of states versus a model using only the mode of this distribution.

---

* Reprinted with permission from "A birth-death Markov Chain Monte Carlo Method to Estimate the Number of States in a State-Contingent Production Frontier Model" by Preciado Arreola, J.L. and A. L. Johnson, 2015. *American Journal of Agricultural Economics*, 1267-1285, Copyright by American Agricultural Economics Association.

The chapter is organized as follows: The Model section, Section II.1, presents the functional form and our assumptions about the production frontier, the dependence and distributional assumptions about the frontier parameters to be estimated, and the steps followed to conduct inference. The Empirical Application section, Section II.2, describes the rice farm case study including the application-specific prior parameter values, the estimation results, our economic interpretations, and our incorporation of some common economic constraints. We test the fit and robustness of our results and discuss our sensitivity analyses. We also assess the degree of label-switching present in our results.

II.1 Model

We estimate a production frontier and the number of states using a BDMCMC algorithm with a "nested" Gibbs sampler. This Gibbs sampler draws from the conditional posterior distributions of the parameters to be estimated and infers the frontier coefficients, precisions, state probabilities, state allocations and inefficiency terms of the observations in the dataset for a *fixed* number of states. In order to obtain expressions that are proportional to these conditional posterior distributions of each parameter, we must multiply the joint *prior* distribution against the likelihood function - the joint density of all output levels in our dataset conditioned on all other parameters - to obtain an unnormalized joint posterior distribution. The joint prior distribution is defined by the assumed prior distributions on all parameters and by the Bayesian Hierarchical model that specifies the dependence structure between them.

The BDMCMC algorithm selects the number of states by computing the birth and death rates for states through likelihood comparisons (see step 1 of the algorithm in the

Bayesian Inference via BDMCMC subsection below) between the current model and models where an individual state is removed. Once these rates are obtained at each pass of the BDMCMC algorithm, a birth or death happens within a Markov Chain describing the number of states. Stephens (2000) proves that the stationary distribution of this Markov Chain, obtained after a large enough number of births and deaths, converges to the correct posterior distribution for the number of states. Moreover, Stephens (2000) proves that using the Gibbs sampler to obtain draws from all frontier parameters each time a state is born or dies results in a valid joint distribution of all of the parameters estimated within the BDMCMC algorithm, which simultaneously solves both the estimation and model selection problems.

II.1.1 Production Frontier Regression Model

We first define a state-contingent production frontier regression model (henceforth, the production frontier) that will be estimated by our BDMCMC algorithm

$$\ln Y = f_j(X_1, \ldots, X_k)+\varepsilon, \tag{II.1}$$

where f_j is the production function associated with state j, and $\varepsilon=v-u$ is a composite error term obtained by subtracting a non-negative variable u representing the technical inefficiency of a farm relative to the efficient frontier from a random effect v, which follows a finite normal scale mixture distribution centered at 0. We assume all f_j's have the same functional form; however, the production frontier coefficients are state-dependent following O'Donnell and Griffiths (2006)'s most general random effects model. We scale all input variables to their mean values and adapt the model for a panel context resulting in

$$\ln Y_{it}=\mathbf{d}_{it}\boldsymbol{\beta}_{\mathbf{0}}+(\mathbf{x}_{it}\otimes\mathbf{d}_{it})'\boldsymbol{\beta}_{\mathbf{-0}}+ v_{it}-u_i. \tag{II.2}$$

Note that we assume a fully separable intercept term $\boldsymbol{\beta}_{\mathbf{0}}$ from the chosen functional

form. To have state-varying coefficients, we define J to be the number of states and $\boldsymbol{d}_{it}$ as a vector containing J state-specific binary dummy variables. Moreover, $\boldsymbol{\beta}_0=[\beta_{01},\ldots,\beta_{0J}]'$ is a vector of state-varying intercepts, $\otimes$ is the Kronecker product, $\boldsymbol{\beta}_{-0}=[\boldsymbol{\beta}_{-01},\ldots,\boldsymbol{\beta}_{-0J}]'$ is a vector of state-varying slope coefficients and $\boldsymbol{\beta}=[\boldsymbol{\beta}_0,\ \boldsymbol{\beta}_{-0}]'$ is a vector containing the coefficients of all J frontiers. The $\boldsymbol{d}_{it}$ vector is used to specify the distributions of the random effects $v_{it}\sim N(0,[\mathbf{d}_{it}\mathbf{h}]^{-1})$, where $\boldsymbol{h}$ is the vector of state-specific precisions. Finally, the output vector Y_{it} is the rice yield of farm i at time t.

Next, we consider two distinct time trend assumptions and fit separate models for each: in the first model, the linear time trend model, β_{TRj} (included in $\boldsymbol{\beta}_{-0j}$), is a scalar and in the second model, the dummy time trend model, it is a vector of state-independent dummy coefficients for each year, i.e. $\boldsymbol{\beta}_{TRj}=[\beta_{TR2j},\ldots,\beta_{TRTj}]$ with $\beta_{TRtj}=\beta_{TRtk}\ \forall\ j,k\in\{1,\ldots,J\}$, for each year $t\in\{2,\ldots,T\}$ in the timespan. The dummy time trend model is important, because the results produced by models with less flexible time trends, i.e. the linear trend model and the models estimated by O'Donnell and Griffiths (2006), could lead us to attribute output variations over time to the state-specific effects on mean output, to noise, and/or to inefficiency.

II.1.2 Bayesian Hierarchical Model

To estimate the parameters of the production frontier with a Gibbs sampler, as done in our application of BDMCMC, we define a dependence structure using a Bayesian Hierarchical model. Figure 1 illustrates this hierarchical structure; the circles indicate parameters to be inferred during the BDMCMC algorithm, the squares indicate known or

assumed quantities, and the arrows indicate that the value of the parameter located at the end of an arrow depends on the value of the parameter at its beginning (see the Prior and Conditional Posterior Distributions subsections for details of the explicit mathematical prior and posterior dependence among the parameters).

Similar to Richardson and Green (1997), our hierarchical model extends the models in O'Donnell and Griffiths (2006) by including *g* and *m* as hyperparameters of Θ (the precision of the random effect associated with each state), which allows for smooth precision estimates among the states. Similar to O'Donnell and Griffiths (2006) and Hurn, Justel and Robert (2003), the output is dependent on the input matrix $\boldsymbol{X}$ leading to a standard regression structure. For clarity, we include *J*, the number of states in the model, and the hyperparameter for its prior mean, λ. Further, we define $\boldsymbol{\pi}$, a vector of state probabilities (or the probability that any given observation is assigned to a particular component of the mixture) and ρ to be a hyperparameter of u related to median efficiency. Finally, note that the dependence structure detailed in Figure 1 can also be described as the joint prior distribution, given by

$$p(\boldsymbol{\beta},\mathbf{h},\mathbf{d},u,\boldsymbol{\pi},\rho,\Theta,J) = p(\boldsymbol{\beta}|J,\xi,\kappa)\, p(\mathbf{h}|J,\nu,\Theta)\, p(\boldsymbol{\pi}|J,\delta)\, p(\mathbf{d}|\pi)\, p(u|\rho,\zeta)\, p(\Theta|g,m) \qquad \text{(II.3)}$$

$$p(\rho|\tau^{*})\, p(J|\lambda)$$

and the likelihood function $p(y|\mathbf{X},\boldsymbol{\beta},\mathbf{h},\mathbf{d},u,\boldsymbol{\pi},\rho,\Theta,J)$. Note the output, *y*, appears in the likelihood function which is further discussed in the *likelihood function* section below, but not in the joint prior distribution.

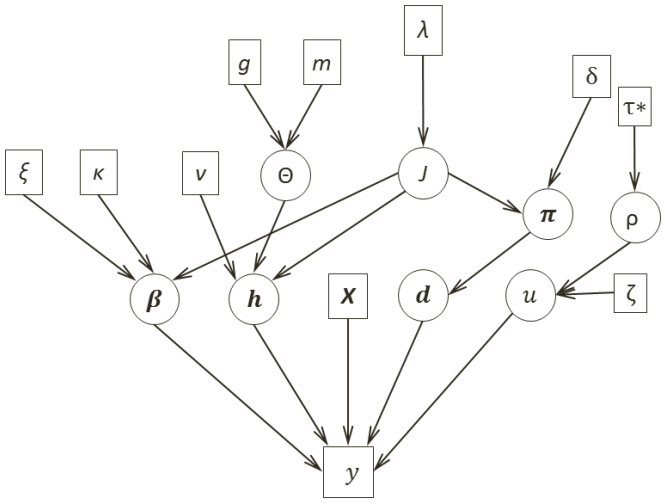

Figure 1 Directed acyclic graph for the hierarchical model used for parameter estimation. All parameters to be inferred are in circles.

II.1.3 Prior Probability Distributions

To obtain conditional posterior distributions from which the Gibbs sampler will draw, we define the *prior* probability distributions for the parameters to be estimated. The prior distributions are dependent on the known prior hyperparameter values. Our production frontier includes a large number of parameters to be estimated and thus requires us to define proper prior distributions on all parameters. A commonly used prior distribution for the regression coefficients and precisions of a production frontier in a Bayesian framework is the Normal – Inverse Gamma (NIG). We also use the prior distributions to impose structure to the states being estimated. Although a loss function has been used to address the state identifiability problem (Hurn, Justel and Robert 2003), we use a labeling restriction that ranks the states of nature from "least favorable" to "most favorable" in order to gain knowledge about the relative desirability of each state. We define our labeling restriction to constrain the expected log-output from a lower-indexed state to be on average less than that

of a higher-indexed state with the NIG assumption to obtain

$\boldsymbol{\beta} \sim N(\boldsymbol{\xi},\boldsymbol{\kappa})\cdot I(E(\ln Y_{it}|x_{it}=0,j=1) \leq \ldots \leq E(\ln Y_{it}|x_{it}=0,j=J))$, where $\boldsymbol{\xi}$, $\boldsymbol{\kappa}$ are fixed and the indicator function describes the labeling restriction.

Consistent with the NIG prior, we assign a gamma prior for each state-specific precision, h_j, such that $h_j \sim \Gamma(\nu,\Theta)$ for all $j \in \{1,\ldots,J\}$, where its hyperparameter Θ has a prior $\Gamma(g,m)$ distribution and ν is fixed. We choose a uniform prior for $\boldsymbol{\pi}$ over *J* states, a Multinomial($\boldsymbol{\pi}$) distribution for $\boldsymbol{d}$ and following Stephens (2000), we choose the number of states to follow a prior Poisson distribution, Po(λ). Also, we assume a (properly normalized) truncated exponential prior distribution for the inefficiency term of the *i*th farm, u_i. Thus, $u_i \sim c\cdot expo(\rho)\cdot I(u_i \leq -\log(\zeta))$, where *c* is a normalization constant and $-\log(\zeta)$ is an upper bound on the inefficiency term. Note that this upper bound translates into a lower bound ζ on the technical efficiency of each farm (TE_i) due to $TE_i = \exp(-u_i)$ [7]. Finally, we assume that ρ, the prior hyperparameter of the u_i terms, follows an expo $(-1/\ln(\tau^*))$ distribution, where we assume that τ^* is a prior estimate of the median technical efficiency.

II.1.4 Likelihood Function

Having specified the prior distributions and dependence structure for our parameters, we now need to specify a likelihood function to multiply with them in order to obtain expressions that are proportional to the conditional posterior parameter distributions that our Gibbs sampler will draw from. We construct the standard likelihood for a mixture of

[7] While there are justifications for this in some industries, in this chapter, this assumption is needed for estimation proposes.

Gaussians with unobserved state allocations by weighting the Gaussian likelihoods with different coefficients and precisions, each of which corresponds to a particular component in the mixture, times the component's relative frequency within the mixture. In equation (II.3) below, the total number of farms is denoted by *N* and the index of the last time period denoted by *T*. Below y denotes the vector of all observations $\ln(Y_{it})$. Note that we use "…" to denote conditioning an all parameter models except for the one of interest.

$$y|\ldots\sim(2\pi)^{-NT/2}\prod_{i=1}^{N}\prod_{t=1}^{T}\left\{\sum_{j=1}^{J}\pi_j\sqrt{h_j}\exp[-0.5h_j(\ln(Y_{it})-\beta_{0j}-\mathbf{x_{it}'\beta_{-0j}}+u_i)^2]\right\} \qquad \text{(II.4)}$$

II.1.5 Conditional Posterior Distributions

We multiply the likelihood against the joint prior distribution assumed for our parameters to obtain an expression that is proportional to their joint posterior distribution, which we then condition. To obtain valid parameter estimates, given a fixed *J*, for the production frontier coefficients $\boldsymbol{\beta}$, precisions $\mathbf{h}$, state probabilities $\boldsymbol{\pi}$, the dummy variables to assign observations to states $\mathbf{d}$, the inefficiency terms u_i and hyperparameters Θ and ρ, we need to draw from their conditional posterior distributions using the Gibbs sampler until the parameter estimates reach stationarity. For our production frontier model, and given our choices of dependence structure, prior distributions and likelihood, the conditional posteriors for $\boldsymbol{\beta}$, $\mathbf{h}$, $\boldsymbol{\pi}$ and $\mathbf{d}$ correspond to the results of a NIG finite Gaussian mixture model for fixed *J*, while the ones corresponding to u_i, Θ and ρ are straightforward to derive. Note that the conditional posterior distribution for u_i is not conjugate, it is a 2-sided truncated normal, but has a truncated exponential prior distribution. For simplicity, we define $\mathbf{z_{it}}=(\mathbf{d_{it}'},(\mathbf{x_{it}}\otimes\mathbf{d_{it}})')'$.

Noting that parameters n_j refer to the number of observations allocated to state *j*, the conditional posterior distributions for our parameters are given by the following[8]:

$$\boldsymbol{\pi}|\ldots\sim D(\delta+n_1,\ldots,\delta+n_k) \tag{II.5}$$

$$\boldsymbol{\beta}|\ldots\sim N(\bar{\boldsymbol{\xi}},\bar{\boldsymbol{\kappa}})\cdot I(E(\ln Y_{iy}|x_{it}=0,j=1)\leq\ldots\leq E(\ln Y_{iy}|x_{it}=0,j=J)) \tag{II.6}$$

$$h_j|\ldots\sim\Gamma\{v+\tfrac{1}{2}n_j,\ \Theta+\tfrac{1}{2}\textstyle\sum_{it:z_{it}=j}[y_i+u_i-(\beta_{0j}+\mathbf{x_{it}}\boldsymbol{\beta}_{-0j})]^2\} \tag{II.7}$$

$$\mathbf{d}|\ldots=\textstyle\prod_{i=1}^{N}\prod_{t=1}^{T}f_M(\mathbf{d_{it}}|1,\bar{\mathbf{d}}_{\mathbf{it}}) \tag{II.8}$$

$$\Theta|\ldots\sim\Gamma(g+Jv,\ m+\textstyle\sum_j h_j) \tag{II.9}$$

$$u_i|\ldots\sim N(\mu_{ui},\sigma_{ui}^2)\cdot I(0\leq u_i\leq -\log(TE_l)) \tag{II.10}$$

$$\rho^{-1}|\ldots\sim\Gamma\left(\frac{N+1}{\mathbf{u}'\mathbf{j}_N-\ln(\tau^*)},2(N+1)\right). \tag{II.11}$$

II.1.6 Bayesian Inference via BDMCMC

Having specified the conditional posterior distributions from which we will sample, the last step is to simultaneously estimate the parameters of our state-contingent production frontier model and its number of states. Prior to explaining this, we will discuss some technicalities of the random variable simulation process conducted in our Gibbs sampler, as random variable generators do not suffice for simulation of our model's parameters. Efficient sampling from the restricted normal distribution of $\boldsymbol{\beta}$ becomes critically important as the dimensionality of the model increases, for larger values of *J*, therefore we incorporate an efficient sampling method (Geweke 1991) when the standard accept-reject criterion fails to

[8] See Appendix A for the posterior hyperparameter expressions.

produce a satisfactory draw from $\boldsymbol{\beta}$'s posterior distribution after 30 trials. We select this number to ensure a reasonable running time, while frequently sampling from the simple accept-reject sampler, and because Geweke's sampler produces draws whose variance depends on the number of iterations of its internal algorithm, which we want to avoid where possible. Unlike O'Donnell and Griffiths (2006), we use the 2-sided truncated normal sampler (Robert 1995) to simulate from the posterior distribution. Finally, we nest our Gibbs sampler within the BDMCMC algorithm and solve the model estimation and model selection problems simultaneously by adapting the algorithm outlined in Stephens (2000). Letting $\boldsymbol{\eta}^{(s)}=(\boldsymbol{\beta}^{(s)},\mathbf{h}^{(s)})'$, the steps in our BDMCMC algorithm are:

1. Run the Birth and Death process, starting at time s for a fixed time s_0, fixing $\mathbf{d}$, ρ^{-1},$\mathbf{u}$,Θ,$\boldsymbol{\pi}$ and $\boldsymbol{\eta}$. Keep $J^{(s+s_0)}$ as a draw from the conditional posterior of *J*.

    a) Fix a birth rate $\lambda b = \lambda$.

    b) Compute death rate for each component $\delta_j(\boldsymbol{\pi},\boldsymbol{\eta})=\frac{L((\boldsymbol{\pi},\boldsymbol{\eta})\backslash(\boldsymbol{\pi}_j,\boldsymbol{\eta}_j))}{L(\boldsymbol{\pi},\boldsymbol{\eta})}$ $\forall j$.

    c) Compute total death rate $\delta(\boldsymbol{\pi},\boldsymbol{\eta})=\sum_j \delta_j(\boldsymbol{\pi},\boldsymbol{\eta})$.

    d) Compute next time s_{new} until a birth or death occurs from an expo$(\lambda_b+\delta(\boldsymbol{\pi},\boldsymbol{\eta}))$ distribution and let $s^*=s+ s_{new}$.

    e) Decide type of jump: Birth with probability $\frac{\lambda_b}{\lambda_b+\delta(\boldsymbol{\pi},\boldsymbol{\eta})}$ or death with probability $\frac{\delta(\boldsymbol{\pi},\boldsymbol{\eta})}{\lambda_b+\delta(\boldsymbol{\pi},\boldsymbol{\eta})}$.

    f) Adjust sizes of $\mathbf{d}$, $\boldsymbol{\pi}$ and $\boldsymbol{\eta}$ according to the type of jump. For birth jumps, generate $\boldsymbol{\pi}$ according to a Beta$(1,k)$ distribution and generate a new component for $\boldsymbol{\eta}$ from its prior. Assign the new component an index j^* such that it does not violate

the labeling restriction. For death jumps, select a component to die with probability $\delta_j(\boldsymbol{\pi},\boldsymbol{\eta})/\delta(\boldsymbol{\pi},\boldsymbol{\eta})$.

g) Repeat until $s^* \geq s + s_0$.

2. Draw $\mathbf{d}^{(s+1)}$ from $p(\mathbf{d}| J^{(s+1)}, \boldsymbol{\pi}^{(s+s_0)}, \boldsymbol{\eta}^{(s+s_0)}, \Theta^{(s)}, \rho^{-1(s)}, \mathbf{u}^{(s)})$.

3. Draw $\Theta^{(s+1)}$ from $p(\Theta| J^{(s+1)}, \boldsymbol{\pi}^{(\mathbf{s+s_0})}, \boldsymbol{\eta}^{(s+s_0)}, \mathbf{d}^{(s+1)}, \rho^{-1(s)}, \mathbf{u}^{(s)})$.

4. Draw $\boldsymbol{\pi}^{(s+1)}$ from $p(\boldsymbol{\pi}| J^{(s+1)}, \boldsymbol{\eta}^{(s+s_0)}, \mathbf{d}^{(\mathbf{s+1})}, \Theta^{(s+1)}, \rho^{-1(s)}, \mathbf{u}^{(s)})$.

5. Draw $\boldsymbol{\eta}^{(s+1)}$ from $p(\boldsymbol{\eta}| J^{(s+1)}, \boldsymbol{\pi}^{(s+1)}, \mathbf{d}^{(s+1)}, \Theta^{(s+1)}, \rho^{-1(s)}, \mathbf{u}^{(s)})$.

6. Draw $\rho^{(s+1)}$from $p(\rho| J^{(s+1)}, \boldsymbol{\pi}^{(s+1)}, \boldsymbol{\eta}^{(s+1)}, \mathbf{d}^{(s+1)}, \Theta^{(s+1)}, \mathbf{u}^{(\mathbf{s})})$.

7. Draw $\mathbf{u}^{(s+1)}$from $p(\mathbf{u}| J^{(s+1)}, \boldsymbol{\pi}^{(s+1)}, \boldsymbol{\eta}^{(s+1)}, \mathbf{d}^{(\mathbf{s+1})}, \Theta^{(s+1)}, \rho^{-1(s+1)})$.

Note that the choice of λ_b is inconsequential for the algorithm to converge, because λ_b can take any finite positive value (Stephens 2000). Since the birth or death jumps of the algorithm must consider the labeling restriction, we add another instruction to Step 1, (f), which assigns the newly generated index component j^* such that its intercept is larger than that of the (j^*-1)th component and smaller than that of the (j^*+1)th component. Finally, note that drawing from our Gibbs sampler takes place on steps 2-5.

II.2 Empirical Application

As mentioned previously, our dataset contains observations for 44 rice farms operating in the Tarlac region of the Philippines between 1990 and 1997. Each observation, which refers to a farm *i* at year *t*, comprises the value for the yield (in tons), the output variable *Y*, and the values for Area planted (hectares planted), Labor used (person-days), and Fertilizer (kilograms of nitrogen, phosphorus and potassium or NPK fertilizer) used, the input

variables, X_1, X_2 and X_3, respectively. Summary statistics for our database, identical to the ones shown in O'Donnell and Griffiths (2006), are shown in Table 1. We use a translog production function, for which the input vector is

$$\mathbf{x_{it}}=[TR_{it}, \ln(X_{1it}), \ln(X_{2it}), \ln(X_{3it}), 0.5\ln(X_{1it})^2, \ln(X_{1it})\ln(X_{2it}),$$
$$\ln(X_{1it})\ln(X_{3it}), 0.5\ln(X_{1it})^2, \ln(X_{2it})\ln(X_{3it}), 0.5\ln(X_{3it})^2]', \quad \text{(II.12)}$$

where TR is either a scalar or a vector of the binary dummy variables, depending on which model we consider. We scale all input variables at their means and define all of the dummy variables of the dummy time trend model as the difference in the yearly shift of the production year against the base year, 1990, following Baltagi and Griffin (1988); note that this base year has no associated time dummy variable. Since the variables are scaled at their input means, ordering states by their intercept values ensures that the mean log-output of a higher-indexed state will be greater than that of a lower-indexed state, i.e. $I(\beta_{01}\leq\beta_{02}\leq\ldots\leq\beta_{0J})$ is equivalent to $I(E(\ln Y_{iy}|\mathbf{x_{it}}=\mathbf{0},j=1)\leq\ldots\leq E(\ln Y_{iy}|\mathbf{x_{it}}=\mathbf{0},j=J))$.

Table 1 Summary Statistics

Variables	Mean	SD	Minimum	Maximum
Y = Rice output (tons)	6.47	5.08	0.09	31.10
X_1 = Area	2.12	1.45	0.20	7.00
X_2 = Labor	107.20	76.65	8.00	436.00
X_3 = Fertilizer	187.05	168.59	3.40	1030.90

The current section is organized as follows: the application-specific prior parameter values are detailed on the Prior Parameter Values subsection, the estimation results and economic interpretations are discussed on the Posterior Parameter Estimates and

Interpretation subsection; also, some common economic constraints incorporated to our models are discussed in the Incorporating Monotonicity and Convexity constraints to the Frontier Model subsection. We then proceed to test the fit and robustness of our results, in the Goodness of Fit subsection and Sensitivity Analysis and a Label-switching subsection respectively.

II.2.1 Prior Parameter Values

We begin by selecting a value of $\mathbf{1}_J$ for hyperparameter δ, which is a J-dimensional vector of ones, to reflect the least possible prior knowledge about the probability occurrence of each state of nature. We select prior values for the slopes and intercepts, ξ and covariance matrix $\boldsymbol{\kappa}$, so that the intercept terms of ξ comply with our labeling restriction and the slopes of the main variables have a relatively small value with a large variability and thus are not significant *a priori*. We assign no prior significance and large variability to the coefficients corresponding to the interaction terms. We select precision-related hyperparameters, ν, g and m, based on the range of the errors resulting from a frequentist regression using a translog functional form and considering a linear time trend. We use 3, O'Donnell and Griffiths (2006)'s result for the number of states, as the prior value for λ, which is the mean of the distribution describing the number of states. Finally, we consider a prior median Technical Efficiency (TE) $\tau^*$of 0.875 with a prior lower bound of 0.7 (see corresponding subsections for the sensitivity analysis performed on λ, τ^* and the prior lower bound of TE). Table 2 summarizes the prior values for the hyperparameters.

Table 2 Summary of Prior Values for Model Hyperparameters

Hyperparameter	Prior Value	Justification
Intercepts on ξ,$\boldsymbol{\kappa}$	$(2j\text{-}1)/2J^{th}$ percentile of output vector, where j is the state number ,2.25	Comply with labeling restriction. Assign large prior variance.
Time trend on ξ,$\boldsymbol{\kappa}$ (for linear time trend model)	0.02,0.15	Annual output growth rate from database. Assign large prior variance.
Time dummies on ξ,$\boldsymbol{\kappa}$ (for dummy trend model)	0.02*(year-1990), 0.15	Annual output growth rate from database. Assign large prior variance.
Hyperparameter	Prior Value	Justification
Slopes on ξ,$\boldsymbol{\kappa}$	0.5,6.5	Small value, not significant due to its large variance.
Interaction terms on ξ,$\boldsymbol{\kappa}$	0,26	Consider no impact of input interactions a priori. Also, assign large prior variance.
δ	$\mathbf{1}_J$	Consider all states to be equiprobable a priori; assign smallest possible weight to prior distribution.
ν	2	Same as Richardson and Green (1997) to ensure data-dependent vague precision prior; $\{e_{it}\}$ are the errors of a least squares regression with the translog function.
g	0.2	
m	$100*g/(\nu*(\max_{i,t} e_{it} - \min_{i,t} e_{it})^2)$	
λ	3	Best estimation obtained by O'Donnell and Griffiths (2006)
Max. of number of allowed states	100	Needed for BDMCMC; choose very large value so as not to affect estimation
τ^*	0.875	Assumed prior median technical efficiency
ζ	0.7	Assumed prior lower bound for technical efficiency

II.2.2 Posterior Parameter Estimates and Interpretation

We run both the linear and dummy time trend models as well as their monotonicity-constrained versions. Here, we discuss the results of the monotonicity-constrained linear and dummy time trend models and compare them against those of the monotonicity-constrained fully state-contingent model in O'Donnell and Griffiths (2006). We run 5500 iterations for all our models, discarding the first 500 as burn-in. From the observed mixing on the trace plots for the number of states, precisions, state probabilities and frontier coefficients (available in Appendix B for both the monotonicity-constrained dummy time trend model and the monotonicity-constrained linear time trend model), this period appears to be sufficient. After estimating the frontier, we find that the posterior distribution of the number of states of nature, *J*, is unimodal with a mode 3 for both monotonicity-constrained models. Moreover, the highest probability region of the posterior distribution of *J* allows us to create a 90% credible set including the values 2, 3 or 4 (Figure 2). This distribution, which provides evidence of at least two major groups of farms based on differences in environmental conditions, supports our state-contingent hypothesis.

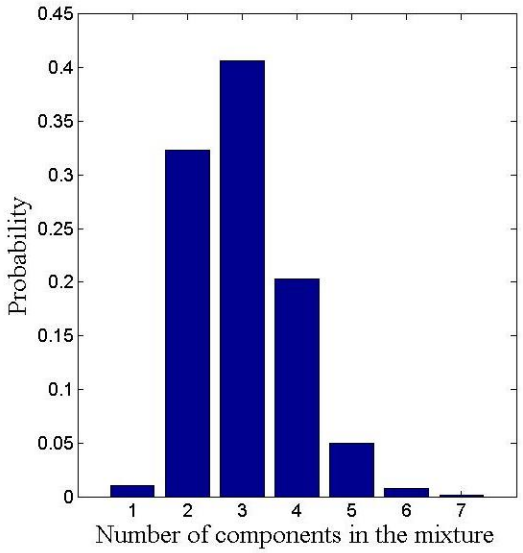

Figure 2 Posterior distribution for the number of components in the mixture for dummy time trend.

Moreover, if hyperparameter Θ is held fixed, our linear time trend model is equivalent to the fully state-contingent model in O'Donnell and Griffiths (2006), but with Θ held fixed, our estimation procedure results in a posterior distribution for the number of states with a mode of 2 versus the 3 states estimated in O'Donnell and Griffiths (2006). This latter finding suggests that O'Donnell and Griffiths (2006) overestimate the number of states by relying on BIC as the model-selection criterion. Tables 3a and 3b shows the results for our models for $J=3$ and for the fully state-contingent models in O'Donnell and Griffiths (2006).

The expected efficient log-output for the first state (the intercept term) in our two models, is not as low as the estimates in O'Donnell and Griffiths (2006). Hence, we infer that factors other than land area, labor and fertilizer have fewer detrimental effects relative to the results of O'Donnell and Griffiths (2006). For instance, if we exponentiate the intercepts to obtain expected efficient yields for the first year of the timespan, 1990, we find that for state

2 of our dummy time trend model this figure is only 6% higher than that of the most unfavorable state (state 1) whereas this same comparison results in a 200% difference for O'Donnell and Griffiths (2006).

We define four confidence levels (99% - highly; 95% - moderately; 90% - mildly; and 85% - barely significant)[9], to compare our two models with O'Donnell and Griffiths (2006). For our two models and O'Donnell and Griffiths (2006), the Area, Labor, and the state-contingent intercepts are significant at approximately the same levels for all states. The primary difference is in the estimated effect of fertilizer, which is highly significant for all states in our dummy model, moderately significant for two states in our linear model, and highly significant only for state 1 in O'Donnell and Griffiths (2006); recall that state 1 has unfavorable (low expected yield) weather conditions. Unlike O'Donnell and Griffiths (2006), however, our two models do not estimate negative fertilizer elasticities for state 1. O'Donnell and Griffiths (2006) posit excessive fertilizer application as detrimental to rice yield in bad weather conditions. We suggest that the disagreement in our parameter estimates and O'Donnell and Griffiths (2006)'s are likely driven by the differences between our worst state's expected yield results. O'Donnell and Griffiths (2006)'s worst state is associated with a significantly lower yield compared to ours (see Tables 3a and 3b).

O'Donnell and Griffiths (2006)'s significantly lower estimates of the yield in state 1 are likely to have effects on the parameter estimates of the production function associated with that state. Our positive elasticity results for fertilizer are supported by SriRamaratnam et

[9] For the sake of computational simplicity, we use classical hypothesis testing, because we only want to evaluate the intercepts, time trends, and elasticities, which have a posterior normal distribution. Thus, the normality of classical hypothesis testing holds.

al. (1987). They show that for yield per unit area levels similar to the ones obtained in our state 1, fertilizer, ranging from very low to very high concentrations per unit of area, has a positive effect on field yield. Thompson et al. (1985), show similar results in the context of ratoon crops. Thus, we conclude that for all states estimated in our application, fertilizer has a non-decreasing relationship to yield. Furthermore, this result does not contradict the notion of fertilizer being a risky input in a profit-maximization context, as the increase in yield may not out weight its acquisition and application costs. Moreover, O'Donnell and Griffiths (2006) shows moderately or barely significant elasticities for the squared fertilizer term for all states. Combining these with their fertilizer elasticities, O'Donnell and Griffiths (2006)'s results and ours are roughly consistent for states 2 and 3, suggesting a moderately to barely significant positive elasticity of fertilizer on rice yields.

The planted area elasticities of our two models seem to be slightly higher for states 1 and 3 and significantly higher for state 2, suggesting a higher per hectare yield relative to O'Donnell and Griffiths (2006)'s results. For labor, our estimated elasticities and O'Donnell and Griffiths (2006) are similar. Finally, and unlike O'Donnell and Griffiths (2006), our two models show no significant second-order or interaction terms and first order input coefficients have a sum close to unity. From these results, we conclude that approximately constant returns-to-scale characterize the Tarlac region's rice production.

Table 3a Estimated Production Frontier Coefficient Means for BDMCMC Linear, Dummy Time Trend and O'Donnell and Griffiths (2006) Models at the Posterior Mode J=3

		Means					
		Linear Trend		Dummy Trend		O'Donnell and Griffiths (2006) SC-all	
Coefficient	State	Free	Mono	Free	Mono	Free	Mono
Intercept	1	1.886****	1.886****	1.917****	1.918****	1.118****	1.112****
	2	2.002****	2.002****	1.983****	1.976****	1.814****	1.803****
	3	2.098****	2.101****	2.061****	2.049****	2.082****	2.079****
Time Trend	1	0.024*	0.023*	Multiple	Multiple	0.028**	0.029**
	2	0.011	0.012	Multiple	Multiple	-0.014	-0.013
	3	0.013	0.014	Multiple	Multiple	0.009	0.010
ln(Area)	1	0.708****	0.696****	0.735****	0.623****	0.615***	0.434***
	2	0.645****	0.637****	0.66****	0.584****	0.133	0.143**
	3	0.556****	0.546****	0.586****	0.577****	0.561*	0.369***
ln(Labor)	1	0.016	0.036	-0.033	0.127	-0.333	0.107
	2	0.071	0.079	0.044	0.138	0.024	0.119**
	3	0.231**	0.241**	0.201	0.204**	-0.106	0.184**
ln(Fertilizer)	1	0.197	0.193	0.212	0.187***	-0.199	-0.368***
	2	0.151**	0.151**	0.183	0.18***	0.112	0.049
	3	0.15**	0.152**	0.159	0.169***	0.313**	0.231
ln(Area)^2/2	1	-0.259	-0.297	-0.256***	-0.297	0.011	0.006
	2	-0.558	-0.545	-0.40****	-0.398	-0.408	-0.395
	3	-0.637	-0.619	-0.59****	-0.567	-0.690	-0.158
ln(Area)*	1	0.420	0.481	0.413	0.454	0.176	0.213
ln(Labor)	2	0.611	0.592	0.494	0.504	0.336	0.391
	3	0.740	0.731	0.732	0.700	0.700	0.379
ln(Area)*	1	0.157	0.141	0.165	0.093	0.171	0.048
ln(Fert.)	2	0.159	0.169	0.156	0.106	-0.005	-0.048
	3	0.039	0.033	0.047	0.047	-0.529	-0.492
ln(Labor)^2/	1	-0.441	-0.537	-0.442**	-0.399	-0.580	-0.366
2	2	-0.493	-0.481	-0.496**	-0.501	-0.145	-0.066
	3	-0.772	-0.746	-0.827***	-0.779	-0.221	-0.408
ln(Labor)*	1	-0.307	-0.278	-0.287	-0.262	-0.288**	-0.223
ln(Fert.)	2	-0.349	-0.346	-0.269	-0.228	-0.181	-0.204
	3	-0.145	-0.153	-0.106	-0.111	0.555	0.637**
ln(Fert.)^2/2	1	0.151	0.133	0.119	0.136	-0.172	-0.228*
	2	0.129	0.117	0.085	0.091	0.152**	0.145**
	3	0.035	0.046	0.005	0.016	-0.395*	-0.385*
Precision	1	11.540	11.433	14.616	14.342	5.81	5.648
	2	12.809	12.985	14.752	14.701	8.513	8.528
	3	15.138	15.248	16.807	16.582	8.303	8.346
State Prob.	1	0.296	0.293	0.347	0.330	0.312	0.306
	2	0.345	0.344	0.325	0.335	0.363	0.364
	3	0.358	0.361	0.327	0.333	0.325	0.330

Note: *, **, ***, and **** denote significance at 15%, 10%, 5% and 1%, respectively

Table 3b Estimated Production Frontier Coefficient Standard Deviations for BDMCMC Linear, Dummy Time Trend and O'Donnell and Griffiths (2006) Models at the Posterior Mode *J=3*

		Standard Deviations					
		Linear Trend		Dummy Trend		O'Donnell and Griffiths (2006) SC-all	
Coefficient	State	Free	Mono	Free	Mono	Free	Mono
Intercept	1	0.102	0.098	0.075	0.071	0.200	0.196
	2	0.079	0.077	0.070	0.066	0.087	0.087
	3	0.085	0.085	0.080	0.074	0.108	0.103
Time Trend	1	0.020	0.020	Multiple	Multiple	0.018	0.018
	2	0.015	0.015	Multiple	Multiple	0.016	0.016
	3	0.014	0.014	Multiple	Multiple	0.014	0.014
ln(Area)	1	0.270	0.268	0.060	0.158	0.296	0.239
	2	0.209	0.206	0.204	0.157	0.195	0.103
	3	0.192	0.192	0.230	0.161	0.471	0.241
ln(Labor)	1	0.268	0.265	0.215	0.109	0.242	0.094
	2	0.206	0.203	0.199	0.115	0.184	0.090
	3	0.173	0.171	0.188	0.131	0.347	0.144
ln(Fertilizer)	1	0.174	0.177	0.218	0.105	0.239	0.227
	2	0.122	0.119	0.197	0.096	0.124	0.110
	3	0.102	0.108	0.170	0.090	0.201	0.190
ln(Area)^2/2	1	0.742	0.744	0.128	0.641	0.339	0.344
	2	0.738	0.688	0.113	0.674	0.933	0.927
	3	0.921	0.867	0.100	0.837	1.795	1.416
ln(Area)*	1	0.659	0.647	0.602	0.567	0.341	0.334
ln(Labor)	2	0.656	0.636	0.709	0.607	0.647	0.641
	3	0.780	0.764	0.905	0.770	1.237	1.154
ln(Area)*	1	0.414	0.407	0.517	0.330	0.241	0.229
ln(Fert)	2	0.380	0.360	0.633	0.341	0.456	0.415
	3	0.430	0.407	0.815	0.401	0.581	0.578
ln(Labor)^2/2	1	0.882	0.896	0.327	0.786	0.510	0.489
	2	0.861	0.864	0.361	0.811	0.733	0.729
	3	0.993	0.981	0.417	0.975	1.332	1.296
ln(Labor)*	1	0.374	0.399	0.735	0.301	0.223	0.218
ln(Fertilizer)	2	0.365	0.354	0.837	0.327	0.459	0.437
	3	0.404	0.391	0.993	0.372	0.495	0.484
ln(Fertilizer)^2/2	1	0.281	0.288	0.297	0.205	0.189	0.186
	2	0.241	0.232	0.351	0.235	0.109	0.106
	3	0.297	0.280	0.390	0.267	0.358	0.357
Precision	1	3.075	3.014	3.555	3.870	1.082	1.084
	2	3.221	3.254	3.866	4.415	1.298	1.286
	3	3.639	3.706	4.494	0.094	1.466	1.513
State Prob.	1	0.091	0.091	0.087	0.097	0.060	0.063
	2	0.107	0.109	0.092	0.087	0.050	0.051
	3	0.093	0.091	0.081	0.060	0.058	0.062

The time trend effects are only significant for state 1 in O'Donnell and Griffiths (2006)'s results, indicating either technological progress regarding the methods to handle bad weather conditions, or increasingly more benign "bad" weather conditions throughout the timespan. Our state-contingent linear trend model, which shows that the time trend coefficients are barely significant for state 1, partially supports this insight. Our time trend coefficient for state 3, which is the largest in magnitude, aligns with O'Donnell and Griffiths (2006). Furthermore, our dummy time trend model gives an additional insight into changes in the production environment of the Tarlac region over time. Thus far, the only indicator of the overall change in production conditions over time is O'Donnell and Griffiths (2006)'s model with J=1 (equivalent to the RE model in the O'Donnell and Griffiths (2006) paper), which indicates a significant linear trend coefficient of 0.014. Figure 3, shows a more complicated pattern of output fluctuation over time, as the dummy time trend model's coefficient values relative to the base year are at least barely significant for every year except for 1991 and 1994, and they show that both "good" (1997) and "bad" (1996) years occur.

The state-contingent intercepts for the dummy time trend model are slightly more similar to one another than for the linear time trend model, while the estimated precisions are larger for the dummy trend model. Both of these results indicate that the dummy trends capture a larger percentage of output variability. This finding translates into a smaller percentage of the expected output being assigned to the unobserved variables modeling the different states (for details, see the Label-switching subsection). In general, the precisions of the noise terms are larger in our two models than in O'Donnell and Griffiths (2006), which is consistent with the smaller MSE obtained using our models (for details, see the Goodness of Fit subsection). The technical efficiency estimates for our two models have a Spearman

correlation coefficient of roughly 80% if compared to O'Donnell and Griffiths (2006), meaning that the efficiency rankings do not change radically (see top figure on page 47). For the 44 rice farms as a group, the mean technical efficiency estimated using either the dummy or the linear time trend models is lower than O'Donnell and Griffiths (2006). However, if we set a lower bound on technical efficiency to 0.8, our result is less than 1% different from that of O'Donnell and Griffiths (2006). In any case, O'Donnell and Griffiths (2006)'s TE *distribution* differs from the results of our two models, based on a series of Kolmogorov-Smirnov tests with p-values lower than 0.1% for the null hypothesis of distributional equality. One exception occurs when we compare O'Donnell and Griffiths (2006)'s TE distribution with our linear model with a lower bound on technical efficiency set to 0.8, where the p-value for the null hypothesis is equal to 0.105. Hence, the hypothesis is not rejected at the usual significance levels, indicating statistically equivalent distributions.

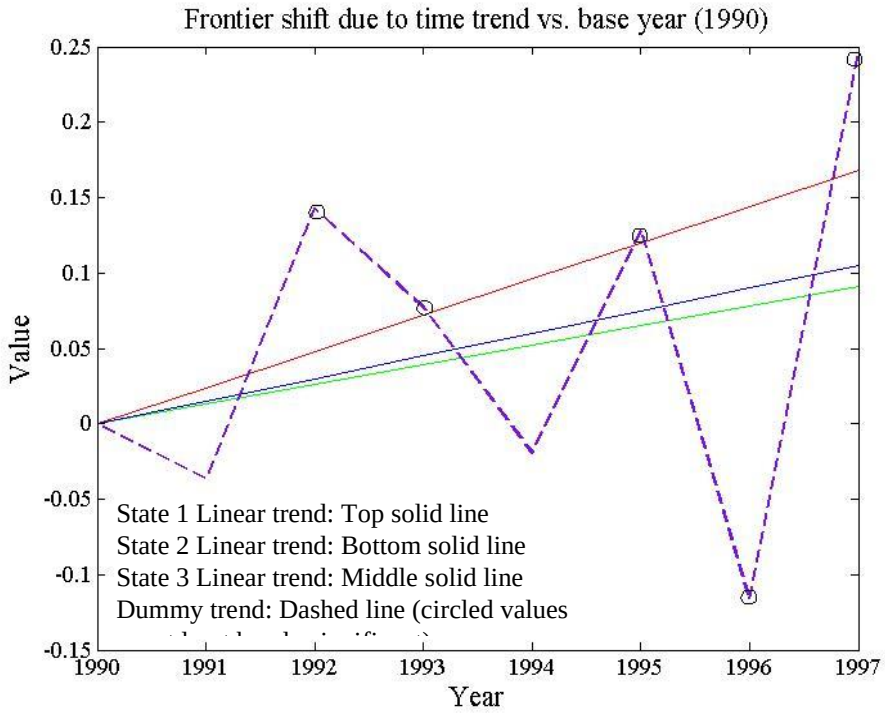

Figure 3 Dummy trend compared against the state-contingent linear time trends.

When comparing the random effects model against a fully state-contingent model, O'Donnell and Griffiths (2006) find a 7% higher mean TE when using the state-contingent model. While we can neither validate nor refute O'Donnell and Griffiths (2006)'s result, since our mean TE depends on the prior lower bound, we can assess the degree to which mean TE increases when using a state-contingent frontier, by estimating a random effects model with the same value for the lower bound on TE. For the dummy time trend model, a less than 1% increase in the TE is observed when considering a state-contingent frontier instead of a random effects frontier suggesting that state-contingency does not account for as much variability in output as O'Donnell and Griffiths (2006)'s results indicate.

II.2.3 Incorporating Monotonicity and Convexity Constraints into the Frontier Model

To impose monotonicity of inputs, we use an accept-reject method to sample from the truncation region for the multivariate normal draws of **β** (Area, Labor, Fertilizer). We define the truncated region by the monotonicity and labeling restrictions. Our BDMCMC algorithm produces satisfactory monotonicity-constrained draws for both the linear time trend and the dummy time trend models. Tables 3a and 3b shows that there is not a large difference between the restricted and unrestricted models, implying that the unrestricted models nearly - satisfy monotonicity. Regarding the imposition of a convexity constraint, we form the Hessian matrix to test quasi-convexity in the inputs used as suggested by O'Donnell and Coelli (2005). Given the selected dataset, production frontier model, and the distributional assumptions on the parameters, some observations do not comply with a convexity assumption. Hence, we conclude that by using a simple accept-reject method, it is not possible to estimate a convexity-constrained version of our two models using our BDMCMC algorithm for this data set.

II.2.4 Goodness of Fit

To assess the goodness of fit (GOF) of both our linear time trend and dummy time trend models, we compare the MSEs and the quantiles of the posterior distribution of the observed residuals with those of a Gaussian mixture distribution. First, we generate predicted output values from our state-contingent models by simulating from the posterior distribution of the number of states and then simulating from the posterior distribution of π, given the simulated value of J. Drawing the values for these parameters tells us both the state to which an observation is assigned and which production function will predict the output level.

Second, we compare these predicted output levels to the observed output level to obtain a prediction error and to calculate MSE. Third, we compare the prediction error quantiles against those of a Gaussian mixture error with mean zero and the corresponding state-specific precisions in order to validate the distribution of the observed prediction errors. Fourth, since both the observed and theoretical standard errors depend on the generated values of the uniform random numbers we use to draw from the aforementioned posterior distributions, we run the error-generating algorithm until both the observed and theoretical standard error vectors reach stationarity.

We consider two simulation scenarios: the full posterior scenario, which weights predictions from all values of J using the full distribution for the number of states described previously, and the posterior mode scenario, which gives full weighting to the mode of the posterior number of states. Table 4, listing the MSE values obtained for the monotonicity constrained linear and dummy time trend models, shows that the %RMSE is practically identical for all of our models and scenarios and is below 5%, indicating a good in sample performance of our estimated state-contingent production frontier model. Figure 4 and Appendix C show the QQ plots for the stationary vectors for the dummy time trend model and the linear time trend model, respectively. Note that the full posterior scenario exhibits a slightly better fit, and is consistent with our conclusions using MSE. Moreover, the two models predict different sets of outliers, each containing 15 observations, or roughly 4% of the total sample (the right column of figure 4 shows the outlier-free fit results). Based on the minimal difference in MSE obtained when using the mode for J instead of its full distribution, we conclude: 1) for our application, using the mode of the number of states to predict efficiency provides a reasonable fit; and 2) re-running the model assuming the mode

of the posterior distribution of J is the correct number of states gives a parsimonious and nearly as well-fitting model (bottom panels of figure 4). Our conclusions suggest that in this particular application using the mode of the posterior distribution for the number of states is a reasonable choice due to the small MSE difference versus using the full posterior distribution although this may not be the case in a general setting. Comparing both scenarios' GOF shows that using a model based on a point estimate of the number of states, such as using the mode of the posterior distribution or the BIC-minimizing number of states, is in fact a special case of the full posterior model we develop. Table 4 also compares the MSE figures from our models and O'Donnell and Griffiths (2006). The MSE obtained from our models is roughly ¼ the magnitude of the MSE obtained by using O'Donnell and Griffiths (2006)'s estimated frontier and could be a result of the additional smoothing provided in the estimation of the precision parameters, meaning that our models provide a better explanation of the variability in output.

Table 4 MSE, RMSE, %RMSE Comparison Between Different Models

MSE	Linear	Dummy	ODG
$J = 3$	0.075	0.074	0.290
Full	0.075	0.072	-
RMSE	Linear	Dummy	ODG
$J = 3$	0.274	0.272	0.538
Full	0.273	0.269	-
%RMSE	Linear	Dummy	ODG
$J = 3$	4.7%	4.7%	9.2%
Full	4.7%	4.6%	-

II.2.5 Sensitivity Analysis

Implementing the BDMCMC algorithm allows us to analyze the estimated posterior probability distribution for the number of states of nature. To determine the model's robustness, we investigate whether our results depend on the mean value of the prior distribution on the number of states of nature, λ. We assign integer values ranging from 1 to 5 for lambda. In all instances, the posterior shows evidence of at least 2 states.

For both the linear and dummy trend models, the posterior mean and mode for *J* indicate that *J*=3 in most cases. The 90% Highest Posterior Density (HPD) set for the value of *J* seems to be invariant over the distinct choices for λ. Finally, the maximum number of possible states *J*=100 is far from being reached during the estimation process, since 8 is the highest number of components at any point of time for all choices of λ. Therefore, we conclude that this parameter is immaterial to the estimation of the model. Table 5 summarizes the findings for the dummy time trend model.

Table 5 Sensitivity Analysis on Hyperparameter λ, Prior Mean on the Number of States for the Dummy Time Trend Model

Prior lambda	Posterior mode for *J*	Posterior mean for *J*	P(*J*=2|*X*)	P(*J*=3|*X*)	90% HPD[a]
1	3	3.02	0.3162	0.4076	{2,3,4}
2	3	3.07	0.2938	0.413	{2,3,4}
3	3	3.07	0.306	0.4018	{2,3,4}
4	3	3.1	0.285	0.395	{2,3,4}
5	3	3.1	0.3056	0.386	{2,3,4}

[a] Highest Posterior Density set

To conduct sensitivity analysis on our assumptions about the inefficiency levels, we vary the prior lower bound on TE from its base value of 0.7 on the range [0.5, 0.8] in 0.1 increments. that the mean of the posterior distribution for TE shifts upward relative to the prior lower bound, and that the distribution changes to accommodate the smaller range of TE. Figure 5, the scatterplot of the rankings considering all sensitivity scenarios against our base assumption, shows that there are no significant changes in the inefficiency rankings of the firms. Computing Spearman's correlation coefficients for the base scenario rankings against those of the three alternative lower bound values shows that the lowest coefficient is 96.4% for the dummy time trend model, which still indicates a strong relationship between the rankings. Thus, we conclude that while the relative efficiency rankings do not change depending on the value of the lower bound on TE, our inefficiency distribution is sensitive to this assumption.

Finally, we also perform sensitivity analysis on the prior median level of TE. We consider three different values for τ^*: a low value of 0.825, O'Donnell and Griffiths (2006) base value of 0.875, and a high value of 0.925. Figure 6 shows that the inefficiency distribution is approximately independent on the choice of τ^* for our dummy time trend model. The efficiency rankings are approximately maintained for the three scenarios, with the lowest Spearman's correlation coefficient being greater than 99% overall compared to the base scenario. The results for our linear time trend model for all the sensitivity analyses performed are similar and their details can be found on Appendix C.

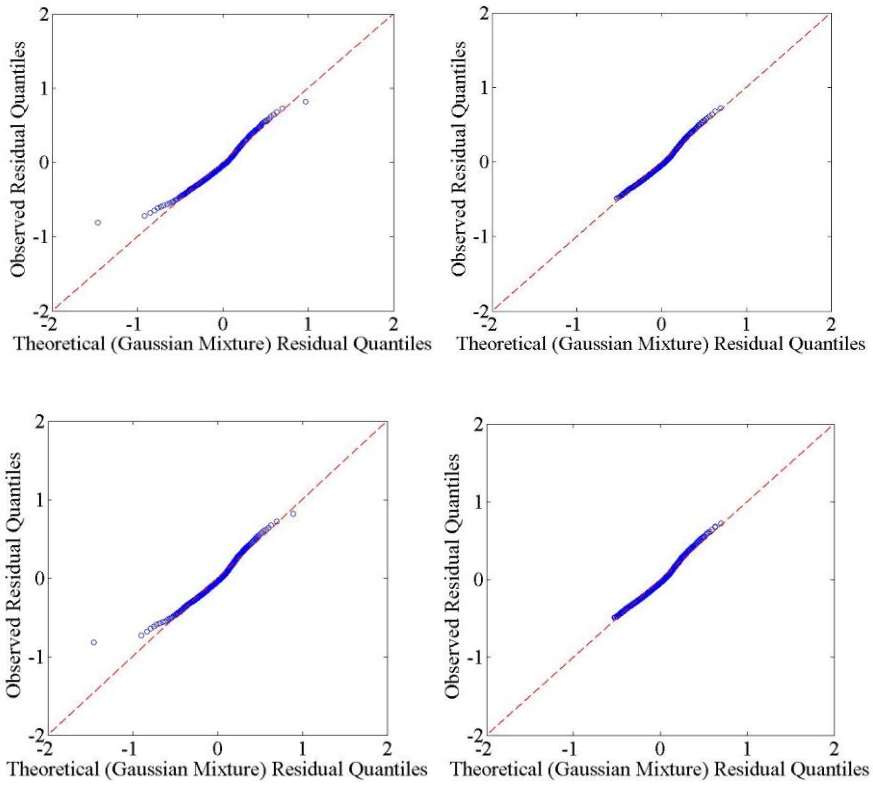

Figure 4 Goodness of fit results for dummy model. Results for full posterior scenario (top left panel), 4% outliers removed for full posterior scenario (top right panel), results for mode scenario(bottom left panel), 4% outliers removed for mode scenario (bottom right panel).

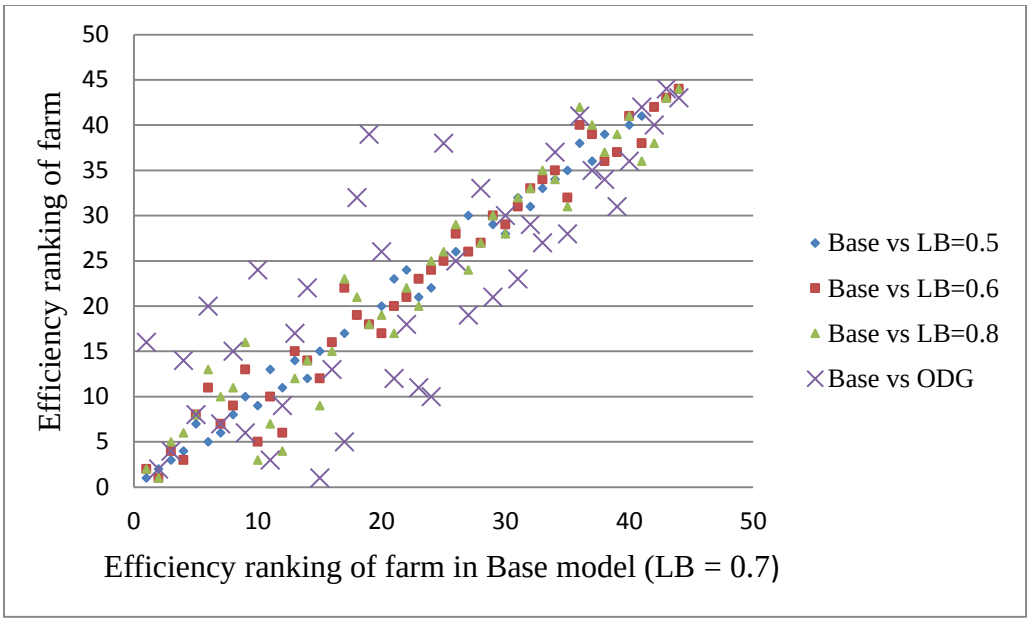

Figure 5 Rankings of the three alternative prior TE lower bound assumptions against the base value of 0.7 for the dummy trend model.

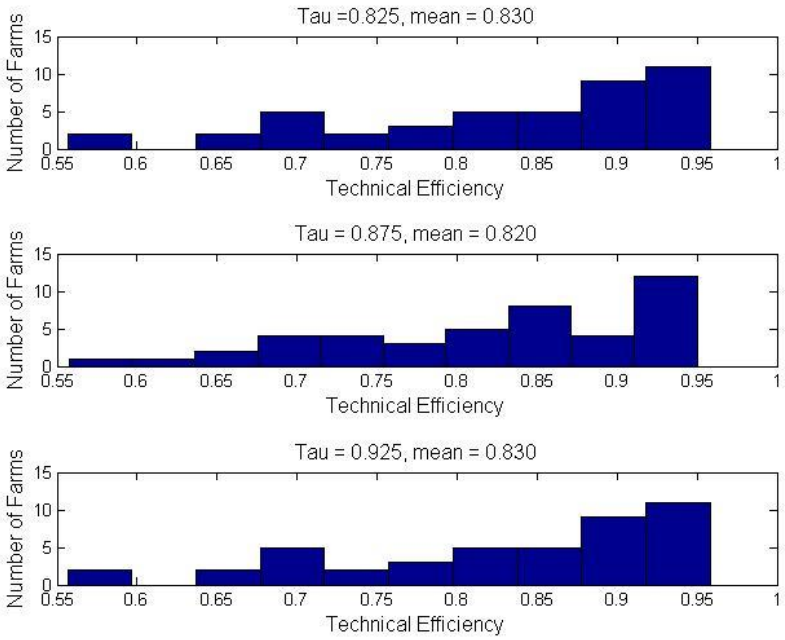

Figure 6 Technical Efficiency distribution for low (top panel), base (middle panel) and high (bottom panel) values of τ^* for the dummy trend model.

II.2.6 Label-Switching

Our labeling restriction alternative addresses the mixture component label-switching problem by setting a labeling restriction to ensure that the drawn intercept value from a lower-indexed state has a lower value than that of a higher-indexed state on each iteration of the BDMCMC algorithm. Since the drawn values differ for each iteration, varying degrees of overlap between the intercept distributions can exist. Figure 7 (left panel) shows that the state-contingent linear time trend model gives a large degree of overlap between these distributions, i.e. the distribution of the first state's intercept has approximately a 50% overlap with the second state's intercept and the second state's intercept distribution has almost a 55% overlap with the third state's intercept distribution. Figure 7 (right panel) shows that the overlap is slightly more significant for the dummy trend model, i.e. if we plot all of the intercept distributions together, the compound histogram is slightly closer to unimodality, making the state-contingency of the intercepts less clear. Restated, the level of state-contingency of mean log-output diminishes only slightly when using a more flexible model to capture the time shifting effect, and continues to support the state-contingent hypothesis about the mean yield per state. Moreover, some coefficients, such as the labor-related components, are noticeably state-contingent when comparing states 1 and 3 (see the dummy time trend model results in table 3a).

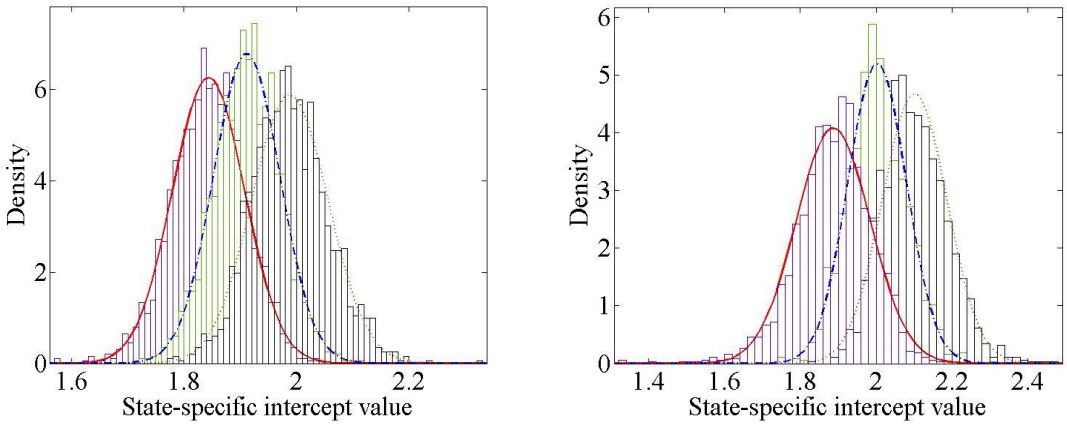

Figure 7 Distribution of intercept (expected output at time 0) for three states, the mode of J. Fitted curves for dummy time trend model (left panel), linear time trend model results (right panel). Solid line denotes distribution for state 1 intercept, dashed line denotes distribution for state 2 intercept and dotted line denotes distribution for state 3 intercept.

CHAPTER III

A ONE-STAGE MULTIVARIATE SEMI-NONPARAMETRIC BAYESIAN CONCAVE REGRESSION METHOD TO FIT STHOCASTIC PRODUCTION FRONTIERS

This chapter describes a method to estimate a production frontier that satisfies the axioms of monotonicity and concavity in a semi-nonparametric Bayesian setting. An inefficiency term that allows for significant departure from the homoscedastic prior distributional assumption is jointly estimated in a single stage. Our method is the first to estimate a nonparametric shape constrained frontier jointly with a flexible non-homoskedastic inefficiency term allowing investigation regarding the tradeoff between the flexibility of a nonparametric production function and flexibility of the inefficiency term in cross-sectional data. Our Monte Carlo simulation experiments demonstrate that the frontier and efficiency estimations are competitive, economically sound, and allow for the analysis of larger datasets than existing nonparametric methods. We use the proposed method to investigate the Japan's concrete industry for the period of 2007-2010. The results show that concrete industry efficiency, a critical component of construction industry efficiency, is relatively high providing support for the continued use of funding construction projects as a method of economic stimulus.

To model inefficiency, this chapter extends MBCR to an MBCR-based semi-nonparametric SFA method. Developing our estimator in the Bayesian context allows us to learn about the inefficiency distribution beyond prior assumptions and to obtain heteroscedastic firm-specific inefficiency estimates that are shrunk both to local variance parameters and to a common population value. The shape-constrained frontier and the

components of the error term are jointly estimated in a single stage. The proposed method, MBCR with Inefficiency (henceforth MBCR-I) is computationally efficient and provides straightforward inference, returning the posterior distributions of the estimated parameters. These characteristics are unique among the estimators available in the literature. Specifically, MBCR-I is unique among SFA estimators in the literature by combining a one-stage framework, shape constraints on the frontier, heteroscedastic posterior distributions of inefficiency that can depart from the homoscedastic prior, a heteroscedastic error term, and computational feasibility for large datasets. Further, due to its one-stage framework, MBCR-I allows the analyst to explore how the inclusion of an inefficiency term affects the detail with which a highly flexible frontier specification is estimated.

The remainder of this chapter is organized as follows. Section III.1 describes the SFA model that we use to fit the observed data, the H-D's MBCR regression method, and our proposed method, MBCR-I) and its characteristics. Section III.2 presents our Monte Carlo simulations for comparing the performance of MBCR-I against Stochastic Nonparametric Envelopment of Data (StoNED), a nonparametric method to fit production frontiers, on the basis of three criteria: functional estimation accuracy, mean inefficiency estimation accuracy and estimator variability across replicates from the same DGP for several generated datasets. Further, we discuss MBCR-I's capability to produce full-dimensional hyperplanes (Olesen and Petersen, 2003) and the dataset characteristics for which MBCR-I use is recommended. Section III.3 discusses several extensions allowing MBCR-I to model time trends of the frontier shift for Panel data and datasets with contextual variables. Section III.4 applies MBCR-I to estimate a production frontier for the Japanese concrete industry and analyzes the substitution rates, most productive scale size, and inefficiency estimates.

III.1 Methodology

III.1.1 Production Frontier Model

We define the regression model for our semi-nonparametric estimation procedure as

$$Y=f(\mathbf{X})e^{\varepsilon}, \tag{III.1}$$

where Y represents observed output, $f(\mathbf{X})$ denotes the best attainable output level, given a certain input mix $\mathbf{X}=(X_1, \ldots, X_k)'$, and $\varepsilon=v-u$ is a composite error term obtained by subtracting a non-negative, skewed random variable u representing a firm's technical inefficiency from a symmetric random effect v, which we term *noise*, assuming a mean 0. For our estimation purpose, we use the firm-specific equation in (III.2) to derive our likelihood function for v_i.

$$\ln(Y_i) = \ln\big(f(X_{1i}, \ldots, X_{ki})\big) + v_i - u_i, \quad i=1,\ldots,n. \tag{III.2}$$

For notational simplicity, we let $f_i=f(X_{1i}, \ldots, X_{ki})$ and $\mathbf{X_i}=X_{1i}, \ldots, X_{ki}$. This allows us to describe the decreasing marginal productivity (concavity) property in terms of $\nabla f(\mathbf{X})$, i.e., the gradient of f with respect to **X,**

$$f(\mathbf{X}_i) \leq + f\big(\mathbf{X}_j\big) + \nabla f(\mathbf{X}_j)^T\big(\mathbf{X}_i - \mathbf{X}_j\big) \ \forall i,j. \tag{III.3}$$

Given that the constraints in (III.3) hold, the additional constraint $\nabla f(\mathbf{X}_i) > \mathbf{0}$ $\forall i$ imposes monotonicity.

III.1.2 Multivariate Nonparametric Bayesian Concave Regression

Even though equation (III.3) leads to a series of pairwise constraints that can be difficult to impose for even moderate datasets, H-D note that the global concavity constraint

is met automatically for the class of functions encompassing the minimums of K hyperplanes. Moreover, H-D prove that the piecewise planar functions estimated by the MBCR algorithm are able to consistently estimate any continuous and concave function. Thus, following H-D, we estimate the function

$$\hat{f}(\mathbf{X})=\min_{k\in\{1,\ldots,K\}} \alpha_k+\beta_k^T\mathbf{X} \tag{III.4}$$

to approximate the concave function $f(\mathbf{X})$.

The estimation procedure in H-D is based on proposing additions, relocations, and removals of hyperplanes, the coefficients of which are determined by fitting Bayesian linear regressions. The MBCR algorithm fits $K^{(t)}$ approximating hyperplanes at a given iteration *t*. Given the current $K^{(t)}$ hyperplanes, H-D partition the set of all observations into subsets C_k, $k\in\{1,\ldots, K\}$, term each subset a *basis region,* and define each one by $C_k=\left\{i:k=\arg\min_{k\in\{1,\ldots, K\}} \alpha_k+\beta_k^T\mathbf{X}_i\right\}$. The MBCR algorithm decides the type of move based on the current status of the Markov Chain. After choosing the type of move, the procedures (see Figure 8) are:

Hyperplane addition: A basis region is split, which creates two proposed basis regions. To divide each region, we consider *L* different proposal splitting knots and M different proposal search directions along each knot,[10] i.e, each region now has *LM* splitting proposals.

[10] The number of hyperplanes estimated is sensitive to the knot and direction selection criteria when adding a hyperplane. Hannah and Dunson (2011) create the knot and direction proposals randomly and we also implement this knot and direction proposal generation scheme.

Relocation: The current basis regions are kept except for minor changes due to refitting.

Removal: Basis regions are proposed for the $K^{(t)}$-1 remaining hyperplanes.

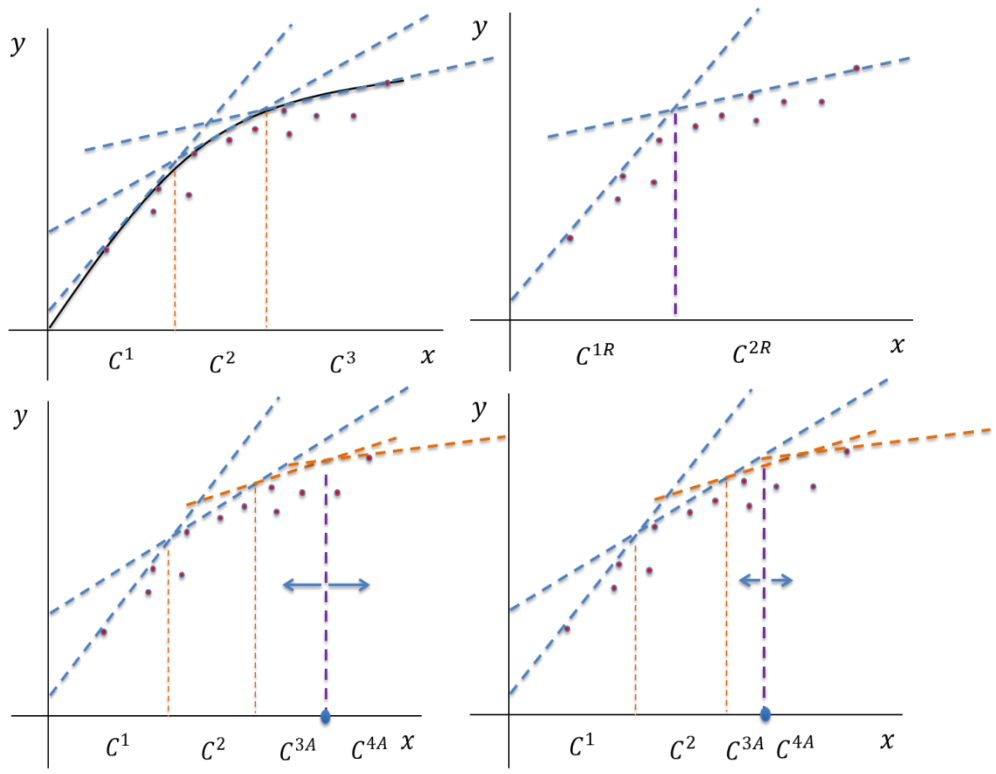

Figure 8 Three basis regions defined by the current number of hyperplanes $K^{(t)}$=3 (top left); proposed basis regions for removal of the second hyperplane (top right); proposal basis regions for hyperplane addition and split of third basis region (bottom left)); and, proposal basis regions for hyperplane addition and split of third basis region using a different splitting knot and the same splitting direction (bottom right).

Regardless of the move type chosen at iteration *t*, the hyperplane coefficients for each new basis region proposal are obtained after refitting. As is common in Bayesian analysis, prior distributional assumptions are placed on the parameters to be estimated by Bayesian

linear regression for each basis region. Due to the difficulties resulting from the homoscedasticity assumption, H-D specify hyperplane-specific Gaussian noise distributions with variances $(\sigma_k^2)_{k=1}^K$.

III.1.3 A Multiplicative Production Frontier

To fit the model described in (III.1) using (III.2), we assume v_i follows the Gaussian mixture distribution $v_i \sim N(\ln(Y_i) - \ln(\hat{f}_i) + u_i, \sigma_{[i]}^2)$, where $\sigma_{[i]}^2$ is the noise variance of the basis region that includes the ith observation. Unlike H-D, which consider a conjugate Multivariate Normal-Inverse Gamma (NIG) prior for estimating the proposal distributions of the hyperplane coefficients $(\alpha_k,\beta_k)_{k=1}^K$ and the hyperplane-specific noise variances $(\sigma_k^2)_{k=1}^K$, we cannot rely on such conjugate proposal distributions for $(\alpha_k,\beta_k)_{k=1}^K$.

Specifically, the logarithm operator applied to $\widehat{f_i}$ prevents the Multivariate Normal distribution on $(\alpha_k,\beta_k)_{k=1}^K$ from being conjugate, given the Gaussian mixture likelihood function on v_i, shown in (III.5). We prioritize computational performance and forgo the ability to draw from the full posterior distributions of $(\alpha_k,\beta_k)_{k=1}^K$, instead estimating the hyperplane parameters by nonlinear least squares with a lower bound of 0 for all β_k's to impose monotonicity.[11] Like Denison, Mallick and Smith (1998), who compute regression coefficients by least squares, but conduct the remaining analysis on a Bayesian framework, we conduct a Bayesian analysis of the remaining parameters to preserve the key MBCR

[11] Given a vague prior on each (α_k, β_k), this is equivalent to the Maximum a Posteriori (MAP) estimate obtained from a Bayesian estimation; see Appendix D for a fully Bayesian version of the algorithm.

property of the endogenous estimation of K. Recalling equation (III.4), we estimate $f_i=f(\mathbf{X}_i;\theta)$ by

$$Y_i= \hat{f}_i e^{v_i} e^{-u_i}, \qquad v_i \sim N\left(\ln(Y_i)- \ln\left(\hat{f}_i\right)+u_i, \sigma^2_{[i]}\right), \qquad u_i \sim H \tag{III.5}$$

$$\beta_k > \mathbf{0}\ , \quad \sigma^2_k \sim IG(a,b), \quad k=1,\dots, K$$

$$K-1 \sim \text{Poisson}(\lambda)$$

As mentioned, $\beta_k > \mathbf{0}$, $k=1,\dots, K$ are necessary to impose monotonicity, whereas the concavity constraints are automatically satisfied, given the construction of the function set from which we choose $\hat{f}$. Initially, we consider different options for *H*, the distribution of the prior inefficiency terms u_i, the most general of which correspond to the Gamma distribution with two unknown continuous parameters, $\Gamma(P,\theta)$ as in Tsionas (2000). While Tsionas (2000) is able to estimate the shape parameter *P*, which Ritter and Simar (1997) show to be difficult in a frequentist setting unless several thousand observations are available, Tsionas obtains parameter estimates close to their true valuesonly when at least 1000 observations are available. Moreover, since Tsionas (2000) estimates *P* and θ in a parametric regression setting, we expect that in a nonparametric setting this moderate-sample bias will be larger if *P* is at all identifiable. In fact, our experiments with generated datasets indicate that *P* is not identifiable in our nonparametric setting, even if a few thousand observations are available and a single input is considered. Therefore, we evaluate scenarios considering the Exponential and Half-Normal prior inefficiency distributions first presented in van den Broeck et al. (1994). Moreover, we place a Poisson prior on the number of hyperplanes, *K*. While this prior is not to be multiplied against the likelihood function in order to obtain a posterior distribution, we still need it to determine the addition, relocation, and removal

probabilities at each iteration of MBCR as described by equations (III.6), where $c \in (0,0.5]$ is a tunable parameter.

$$b_{K^{(t)}}=c \min\left\{1, \frac{p(K^{(t)}+1)}{p(K^{(t)})}\right\}, \quad d_{K^{(t)}}=c \min\left\{1, \frac{p(K^{(t)}-1)}{p(K^{(t)})}\right\}, \quad r_{K^{(t)}}=1-b_{K^{(t)}}- d_{K^{(t)}} \tag{III.6}$$

Equation (III.7) describes the mathematical program used to fit the hyperplanes and obtain $(\alpha_k,\beta_k)_{k=1}^{K}$, where $Y_{i[k]}$ and $\mathbf{X}_{i[k]}$ refer to the *i*th observation in basis region C_k and n_k refers to the number of observations in the basis region. Due to the conjugacy of the IG prior with each σ_k^2, we can easily sample these posterior variances for each of the proposed basis regions by using (III.8). For all cases, we assume that θ, the scale parameter of our prior inefficiency distribution, has a prior $\Gamma(w_0)$ distribution, as shown in (III.10). We also assume $w_0=-1/\ln(\tau^*)$ and that τ^* is a prior estimate of the median Technical Efficiency. The posterior distributions for either the Exponential or the Half-Normal prior assumptions on u_i are the Truncated Normals shown in (III.9a) and (III.9b), respectively, and $\varepsilon_i=\ln(Y_i)-\ln(x_i'\beta)$ denotes the residuals.

$$\min_{\alpha_k,\beta_k} \sum_{i=1}^{n_k} ((\ln(Y_i)+u_i-\ln(\alpha_k+\beta_k^T\mathbf{X}_i))^2 \text{ subject to } \beta_k>0, \quad k=1,\dots,K \tag{III.7}$$

$$\sigma_k^2 \sim IG(a_k^*,b_k^*), \quad k=1,\dots,K, \text{ where} \tag{III.8}$$

$$a_k^*=\tilde{a}+\frac{n_k}{2} \quad , \quad b_k^*=\tilde{b}+\frac{1}{2}\left(\sum_{i=1}^{n_k} (\ln(Y_{i[k]})+u_i-\ln(\alpha_k+\beta_k^T\mathbf{X}_{i[k]}))^2\right)$$

$$u_i|\dots \propto \exp(-1/2\sigma_{u_i}^2 (\mu_{u_i}-u_i)), \quad u_i\geq 0, \quad i=1,\dots,n \text{ , where } \mu_{u_i}=-(\varepsilon_i+\theta\sigma_{[i]}^2), \quad \sigma_{u_i}^2=\sigma_{[i]}^2 \tag{III.9a}$$

$$u_i|\dots \propto \exp(-1/2\sigma_{u_i}^2 (\mu_{u_i}-u_i)), \quad u_i\geq 0, \quad i=1,\dots,n \text{ where } \mu_{u_i}=\frac{-\sigma_{0u}^2\varepsilon_i}{\sigma_{0u}^2+\sigma_{[i]}^2}, \quad \sigma_{u_i}^2=\frac{\sigma_{0u}^2\sigma_{[i]}^2}{\sigma_{0u}^2+\sigma_{[i]}^2} \tag{III.9b}$$

$$\theta|\dots \sim \Gamma(n+1, w_0+\sum_{i=1}^{n} u_i) \tag{III.10}$$

III.1.4 The Proposed MBCR-I Algorithm

We propose an algorithm, MBCR-I, with a smoothed and a non-smooth variant. Our Metropolis-Hastings algorithm first calculates the block $(\alpha_k,\beta_k,\sigma_k^2)_{k=1}^{K}$ from our multiplicative error version of MBCR, then draws the block $(u_i)_{i=1}^{n}$ on a Gibbs step, and ends by drawing θ on another Gibbs step. After verifying that MBCR's fast convergence is around 100 iterations for the examples presented in H-D,[12] we consider a burn-in period for the Metropolis-Hastings sampling algorithm of 150 iterations which is safely beyond the needed convergence period. Then, we monitor mean squared error at iteration *t*, MSE*y* (*t*) = $\frac{1}{n}\sum_{i=1}^{n}(\widehat{Y}_i-Y_i)^2$, where $\widehat{Y}_i=\hat{f}_i e^{-\hat{u}_i}$. We declare that the MBCR-I algorithm has reached stationarity when the running median does not change significantly and the variability across iterations is constant for at least 200 iterations.[13] Finally, as we usually obtain a few hundred draws from the sampling algorithm, we average the functional estimates across iterations to obtain a smoothed estimator, or we select a single iteration for a non-smooth estimator, resulting in two versions of the MBCR-I algorithm, henceforth MBCR-I S and MBCR-I NS. We find that the non-smooth estimator performs better in small datasets for which the inefficiency model is mis-specified, although it relies on a heuristic criterion to select the best iteration and inference is not possible, whereas the smoothed estimator performs well in all other settings, without relying on heuristics and inference is available directly from MBCR-I's output.

[12] The numerical results are available from the author upon request.
[13] Under this stopping criterion, MBCR-I rarely needs more than 1000 iterations to reach stationarity.

Our iteration selection criterion for the non-smooth estimator is motivated by observing that for small sample sizes (n<200), MBCR-I can overfit the data with highly flexible inefficiency terms. To prevent overfitting, we choose a stationary iteration with relatively conservative MSE*y*. [14] Finally, we note that as *n* increases, the iteration selection criterion becomes irrelevant at n≥300 as the variability decreases across the iterations. This decrease in variability is the reason why the smoothed and non-smooth estimators are increasingly similar in n. We summarize the MBCR-I algorithm as follows:

0. Let t = 1, *K* = 1 and set $t_{Burn\text{-}in}$, draw $(u_i)_{i=1}^n$ and θ from their priors.

1. Use MBCR to get $((\alpha_k,\beta_k,\sigma_k^2)_{k=1}^K,K)^{(t)}$.

2. Draw $(u_i^{(t)})_{i=1}^n$ from (III.9a) or (III.9b), depending on the prior assumption.

3. Draw $\theta^{(t)}$ from (III.10).

4. If $t>t_{Burn\text{-}in}$, save $((\alpha_k,\beta_k,\sigma_k^2)_{k=1}^K,(u_i)_{i=1}^n,\theta,K)^{(t)}$ draw and compute

$MSE_y^{(t)}=\frac{1}{n}\sum_{i=1}^n(\widehat{Y}_i-Y_i)^2$. Otherwise, go back to 1.

5. Stop when the cumulative median of MSE_y meets the stationarity criterion. Otherwise, go back to 1.

6a. To obtain a smoothed estimator: Average $(\hat{f}_i)_{i=1}^n$ across the stationary iterations for the f estimator and average mean inefficiency across the stationary iterations to obtain $\widehat{E(\bar{u})}$.

[14] Our simulations consider the iteration with maximum MSE_y within the described subset of iterations.

6b. To obtain a non-smooth estimator: Choose an iteration according to the iteration selection criterion and return the parameters $((\alpha_k,\beta_k)_{k=1}^{K},(u_i)_{i=1}^{n},\theta,K)$ associated with that iteration.

III.1.5 Additional Computational Considerations due to Inefficiency Modeling

Incorporating inefficiency into the MBCR algorithm requires the following adaptations and augmentations. First, we need an efficient and robust Truncated Normal sampler for the inefficiency terms to quickly sample from extreme tails and avoid stalling. We use a MATLAB implementation of Chopin's (2011) fast truncated normal sampling algorithm by Mazet (2014), because other samplers do not achieve the degree of accuracy or posterior coverage needed to make MBCR-I computationally feasible. Second, as described in the MBCR-I algorithm, at any given iteration *t*, we sample the inefficiency draws $(u_i)_{i=1}^{n}$ as a block after computing the hyperplane coefficients and simulating the associated variances as a different parameter block. Here, the only step of the algorithm in which the model size is allowed to change is when we draw the block $((\alpha_k,\beta_k,\sigma_k^2)_{k=1}^{K},K)$.

Nevertheless, we observe that due to the differences in the $(u_i)_{i=1}^{n}$ values from iteration *t* to iteration *t+1*, the number of hyperplanes supporting a positive number of observations can change even if the move at iteration *t+1* is only a relocation (or conversely, the number of hyperplanes supporting a positive number of observations remains the same even if the move is an addition or removal). We automatically reject such proposal distributions and we draw different $(u_i)_{i=1}^{n}$ values, given the $(\alpha_k,\beta_k,\sigma_k^2)_{k=1}^{K}$ of iteration *t*. If this rejection policy results in stalling, measured as the time taken to generate the (*t*+1)*th* draw

compared to the average time to generate a draw, we restart the Markov Chain $((\alpha_k,\beta_k)_{k=1}^{K},(u_i)_{i=1}^{n},\theta,K)$ at its value on a randomly chosen previous iteration. The restarting policy is analogous to H-D's chain restarting policy when the number of tries to produce the $(t+1)$*th* draw of the chain goes above a preset threshold. Third, we run a small number of warm-up iterations, i.e., 20, in our simulation scenarios. In these iterations, which are a subset of the burn-in iterations, we draw $(u_i)_{i=1}^{n}$ from their prior, and we run MBCR to get a good initial guess of *K*, as opposed to the $K = 1$ starting value chosen by H-D. Otherwise, the $(u_i)_{i=1}^{n}$ draws from the initial iterations will be heavily overestimated and complicate, or even prevent, the MBCR-I algorithm from running fluently. An alternative to the warm-up iterations is to use the multiplicative-error MBCR estimates of $((\alpha_k,\beta_k)_{K}^{k=1},K)$ for the same dataset. Section III.2.6 below and Appendix F illustrate our use of the latter strategy when analyzing the Japanese concrete industry and show that allowing flexible inefficiency terms decreases the number of hyperplanes with which the frontier is fit.

III.1.6 MBCR-I as a One-Stage Estimator for Stochastic Frontiers

Unlike StoNED (Kuosmanen and Kortelainen, 2012) and Constraint Weighted Bootstrapping (CWB) (Du et al. 2013), the ability of MBCR-I to significantly depart from prior distributional assumptions on u_i makes the method more robust against model mis-specifications for the inefficiency term. Our posteriors show that besides globally shrinking the inefficiency terms using either θ or σ_{0u}^2, we can *locally* shrink them with the $\sigma_{[i]}^2$

parameters.[15] Moreover, within the bounds established by the shrinking parameters, doing so allows each u_i to have a potentially *different* posterior distribution. In Section 3, we explain how the posterior specification in (III.9a) allows a significantly better prediction of mean inefficiency and the production frontier when the inefficiency distribution is mis-specified. Our posterior specifications are not a source of additional inaccuracy when the prior distributions are in fact correct. Finally, the use of a one-stage framework imposes a correctly-skewed distribution of ε_i at each iteration of the MBCR-I algorithm and avoids the wrong skewness issues of two-stage methods (see, for example, Almanidis and Sickles, 2012).

III.2. Monte Carlo Simulations

This section describes Monte Carlo simulations and their results comparing the performance of MBCR-I versus StoNED on the Data Generation Processes (DGPs) used in Kuosmanen and Kortelainen (2012), some of which are presented in Simar and Zelenyuk (2011).[16] These DGPs, henceforth Example 1 through 4, are based on Cobb-Douglas production functions and they explore the performance of both estimators as dimensionality, noise-to-signal ratio, and sample size vary. Example 2 is added to the DGPs in Kuosmanen and Kortelainen (2012) for completeness, as they do not include a bivariate input example. Example 4 assesses the robustness of each method against mis-specification on the prior

[15] Shrinking refers to the common parameter shrinkage concept in Hierarchical Regression Models (HRM), where parameters are constrained by a common distribution. For further discussion see Gelman and Hill (2006).
[16] CWB is a state-of-the-art two-stage method that can be used to estimate production frontiers. Nevertheless, its application is not straightforward for the proposed DGPs, because the CWB formulation in Du, Parmeter and Racine (2013) considers an additive error structure.

inefficiency distribution. For all four examples we consider three noise-to-signal scenarios, ρ_{nts}=1,2,3, and vary the number of observations, n=100, 200, 300, and 500. StoNED is the only shape constrained frontier estimation method that can handle more than a few hundred observations under a multiplicative error assumption, and therefore is the most natural benchmark for comparison.

We compare the estimators based on criteria including quality and degree of variability in frontier estimates and quality of the inefficiency estimate. We measure the frontier estimation performance as MSE $f=\frac{1}{n}\sum_{i=1}^{n}(\hat{f}_i-f_i)^2$. As measures of degree of variability in our frontier estimates, we also report the number of replications needed to obtain stable estimates in terms of near-constant mean and standard deviation of MSE f. Instead of using MSE $u=\frac{1}{n}\sum_{i=1}^{n}(\hat{u}_i-u_i)^2$ to measure the accuracy of our inefficiency estimation as in Kuosmanen and Kortelainen (2012), we use $\widehat{E(\bar{u})}-E(\bar{u})=\frac{1}{n}\sum_{i=1}^{n}\hat{u}_i-\frac{1}{n}\sum_{i=1}^{n}u_i$, the mean inefficiency prediction deviation.[17] MBCR-I is advantageous because its estimates typically are more consistent with production theory. Specifically, we report the number of observations supported by fully dimensional hyperplanes and the percentage of replicates for which StoNED has negatively skewed residuals.

For MBCR-I, we conduct a MATLAB implementation, considering an Exponential prior with parameter θ. We randomly draw the prior values used for the θ parameter from ranges described in each of the examples. In Tables 6-13 we show results for both MBCR-I S

[17] The metric $MSE\ u$ focuses on firm specific efficiency estimates from the Jondrow et al. (1982) estimator that have been shown to be inconsistent, Greene (2008). Instead, we measure the quality of the inefficiency estimate based on the population parameter $E(\bar{u})$.

and MBCR-I NS. In the case of StoNED, since multiplicative CNLS is a mathematical program with a generally nonlinear objective function its solutions are sensitive to the choice of starting point and solver. Thus, we conduct several implementations of CNLS and select the one with the lowest MSE f for each example. Our MATLAB CNLS implementations use the built-in fmincon solver and the KNITRO solver. Our GAMS implementations uses the MINOS 5.5 solver. For every implementation, we consider different starting points, such as the (global) optimal solution of additive CNLS, single hyperplane solutions, and full vectors of zeros.

III.2.1 Evaluation Based on Four Data Generation Processes

Example 1: Univariate Cobb-Douglas frontier with homoscedastic inefficiency terms

We fit the univariate Cobb-Douglas frontier $Y_i=x_i^{0.5}e^{-u_i}e^{v_i}$, considering a homoscedastic DGP with u_i~Expo($\mu_u=\sigma_u=1/6$) and v_i~$N(0,\sigma_v^2)$, where $\sigma_v=\rho_{nts}\sigma_u$. We randomize the nearly uniformative prior on θ across replicates by choosing $v_0=1$ and drawing w_0 uniformly on the (0.1, 0.2) range. Table 6 shows that in terms of MSE f, StoNED only outperforms MBCR-I S for a noise-to-signal ratio of 1 and less than 300 observations. MBCR-I S and StoNED perform similarly for a noise-to-signal ratio of 1 and $n \geq 300$. For $\rho_{nts}=2,3$, MBCR-I S outperforms StoNED for all n.[18] In terms of the quality of efficiency estimates, all of the estimators perform similarly for a noise-to-signal ratio of 1, but the

[18] We display results for our MATLAB fmincon implementation, which outperforms Kuosmanen and Kortelainen's (2012) results for a low noise-to-signal ratio, $\rho_{nts} = 1$, and has similar performance in Kuosmanen and Kortelainen's high noise-to-signal ratio, $\rho_{nts} = 2$. We did not compare performance when $\rho_{nts} = 3$, because Kuosmanen and Kortelainen do not estimate this scenario. See Kuosmanen and Kortelainen (2012) for a demonstration of StoNED's superior performance relative to standard implementations of SFA and DEA for all scenarios that included $\rho_{nts} > 0$.

MBCR-I estimators are superior when there is a larger noise-to-signal ratio. Finally, the results for StoNED show that non-full dimensional hyperplanes support 1%-8% of the observations.

Table 6 Results for Example 1: Univariate Cobb-Douglas Frontier with Homoscedastic Inefficiency Terms

		MSE *f*			$\widehat{E(\bar{u})}$ -E($\bar{u}$)			% Non-Full Dimensional	
ρ_{nts}	n	Sto NED	MBCR-I NS	MBCR-I S	Sto NED	MBCR-I NS	MBCR-I S	Sto NED	MBCR-I NS
1	100	**0.003**	0.0046	0.0036	**0.01**	**0.01**	**0.01**	4%	**0%**
	200	**0.001**	0.0020	0.0024	**0.01**	**0.01**	**0.01**	2%	**0%**
	300	0.000	0.0014	**0.0007**	**0.01**	**0.01**	**0.01**	3%	**0%**
	500	**0.000**	0.0008	0.0006	**0.01**	**0.01**	**0.01**	1%	**0%**
2	100	0.005	0.0056	**0.0035**	**0.02**	**-0.02**	**-0.02**	6%	**0%**
	200	0.006	0.0052	**0.0035**	0.05	**-0.03**	**-0.03**	2%	**0%**
	300	0.006	0.0055	**0.0028**	0.05	**0.01**	-0.05	3%	**0%**
	500	0.006	0.0049	**0.0023**	0.07	0.05	**-0.02**	3%	**0%**
3	100	0.033	**0.0097**	0.0153	0.09	**0.00**	-0.01	8%	**0%**
	200	0.046	**0.0054**	0.0061	0.18	**0.02**	**0.02**	7%	**0%**
	300	0.042	0.0059	**0.0046**	0.18	**-0.03**	-0.04	8%	**0%**
	500	0.041	0.0050	**0.0028**	0.16	**-0.06**	-0.07	2%	**0%**

Table 7 shows that the percentage of replicates with a negatively skewed ε_i distribution for StoNED is non-decreasing for the noise-to-signal ratio, ρ_{nts}, and non-increasing in the sample size, n, as expected. As explained in the previous section, this problem does not affect MBCR-I, because it is a one-stage method and automatically imposes correct skewness on the distribution of ε_i. For MSE f, both methods need a small

number of replicates to reach relatively constant[19] values; StoNED needs only 10 replicates for all scenarios, whereas MBCR-I needs 20 replicates for 2 of the 12 considered scenarios We attribute the variability of MBCR-I's prediction error, which is smaller both in absolute and relative terms as quantified by the standard deviation and the coefficient of variation of MSE f across all replicates, as the result of non-smooth MBCR-I fitting the production frontier with a smaller number of hyperplanes than StoNED.

Table 7 Estimator Robustness Analysis for Example 1: Univariate Cobb-Douglas Frontier with Homoscedastic Inefficiency Terms

		Replicates for MSE f convergence		Standard Deviation of MSE f		MSE f coefficient of variation		% Negative Skew
ρ_{nts}	n	Sto NED	MBCR-I	Sto NED	MBCR-I	Sto NED	MBCR-I	Sto NED
1	100	10	10	0.0046	**0.0019**	132%	**42%**	10%
	200	10	10	**0.0011**	0.0016	108%	**64%**	0%
	300	10	20	**0.0009**	0.0018	**101**	128%	0%
	500	10	10	**0.0005**	**0.0005**	85%	**63%**	0%
2	100	10	10	0.0056	**0.0036**	104%	**64%**	10%
	200	10	10	0.0058	**0.0018**	85%	**34%**	0%
	300	10	10	**0.0022**	0.0038	**35%**	69%	0%
	500	10	10	0.0048	**0.0047**	**71%**	95%	0%
3	100	10	20	0.0246	**0.0077**	**74%**	75%	20%
	200	10	10	0.0278	**0.0024**	60%	**45%**	0%
	300	10	10	0.0167	**0.0019**	40%	**37%**	0%
	500	10	10	0.0142	**0.0033**	**34%**	57%	0%

[19] We define relatively constant as within a 5% difference across replicates for both the running mean and running standard deviation of $MSE\ f$.

Example 2: Bivariate Cobb-Douglas frontier with homoscedastic inefficiency terms

We fit the bivariate Cobb-Douglas frontier $Y_i=x_{1i}^{0.4}x_{2i}^{0.5}e^{-u_i}e^{v_i}$, where we consider a homoscedastic distribution for both noise and inefficiency, u_i~expo($\mu_u=\sigma_u=1/6$) and v_i~$N(0,\sigma_v^2)$, where $\sigma_v=\rho_{nts}\sigma_u$[20], the same inefficiency assumptions as in Example 1. We also consider the same prior assumptions for θ as in Example 1. MBCR-I's estimators performance in terms of functional fit, MSE *f*, is better in all scenarios with a noise-to-signal ratio greater than 1. Despite StoNED's lower MSE *f* in the $\rho_{nts}=1$ scenarios, MBCR-I's estimates give a more economically sound description of the frontier, because more of its hyperplanes are full-dimensional. In Example 2, non-full dimensional hyperplanes support between 14% and 23% of the observations for StoNED, whereas it is always less than 6% for MBCR-I. Finally, MBCR-I NS performs well when the number of observations is low and MBCR-I S well estimates inefficiency consistently.

Table 8 shows that predictions of the mean inefficiency are competitive for both StoNED and the MBCR-I estimators across the different scenarios varying the noise-to-signal ratio and number of observations, with the exception of the $\rho_{nts}=3$ scenarios, where only MBCR-I S performs well. In Example 1 and the less noisy scenarios of Example 2, StoNED's functional estimates improve, as measured by MSE f, as the number of observations increases typical of any consistent estimator. However, in the high dimensionality and high noise scenarios for Example 2, StoNED's ability to fit the function,

[20] To make the CNLS problem feasible to solve, we reduced our optimality tolerance from 10^{-10} to 10^{-4} for the n =500 scenarios on our GAMS with MINOS 5.5 implementation. Kuosmanen and Kortelainen (2012) did not perform n =500 simulation scenarios in any of their multivariate examples.

measured by MSE f, decreases as the number of observations increases. StoNED's erratic performance relates to the increase in local optima for the optimization problem associated with the first step of StoNED, CNLS, and occurs across all of the MATLAB and GAMS implementations. Even though we implement versions of CNLS which use global solvers,[21] the solvers cannot find solutions for data instances with more than 100 observations. MBCR-I, which uses an adaptive partitioning strategy rather than a full dataset optimization strategy does not suffer from these solution algorithm complexity issues.

Table 8 Results for Example 2: Bivariate Cobb-Douglas Frontier with Homoscedastic Inefficiency Terms

		MSE *f*			$\widehat{E(\bar{u})}$ -E($\bar{u}$)			% Non-Full Dimensional	
ρ_{nts}	n	Sto NED	MBCR-I NS	MBCR-I S	Sto NED	MBCR-I NS	MBCR-I S	Sto NED	MBCR-I NS
1	100	**0.00**	0.0032	0.0038	**-0.02**	-0.07	0.03	16%	**2%**
	200	**0.00**	**0.0015**	0.0018	-0.02	**-0.01**	0.04	15%	**0%**
	300	**0.00**	0.0020	0.0016	-0.02	**-0.01**	0.02	17%	**0%**
	500	**0.00**	0.0010	0.0012	**-0.01**	**0.01**	0.02	10%	**6%**
2	100	0.00	0.0043	**0.0040**	**0.02**	-0.09	**0.01**	18%	**1%**
	200	0.00	**0.0039**	0.0050	**0.01**	-0.04	0.06	19%	**0%**
	300	0.00	0.0040	**0.0036**	**0.01**	-0.08	0.03	20%	**0%**
	500	0.00	0.0029	**0.0014**	0.07	-0.08	**0.00**	16%	**0%**
3	100	0.02	**0.0085**	0.0175	0.10	**-0.06**	0.12	23%	**0%**
	200	0.02	**0.0045**	0.0057	0.14	-0.11	**0.00**	23%	**5%**
	300	0.03	0.0050	**0.0046**	0.17	-0.11	**-0.02**	22%	**2%**
	500	0.02	0.0047	**0.0032**	0.13	-0.12	**-0.06**	14%	**3%**

[21] We attempted to use global nonlinear optimization algorithms such as MSNLP, BARON and ANTIGONE.

Table 9 shows that the percentage of negatively skewed replicates also exhibits roughly consistent behavior throughout the different noise-to-signal ratio and the number of observations. However, for this more computationally challenging example, the percentage of negatively skewed replicates for StoNED is in general higher than in Example 1, and thus predicts negligible inefficiency levels more frequently. Further, both StoNED and MBCR-I need more replicates for their estimates to stabilize if compared with the simpler Example 1. However, we note that MBCR-I functional estimate converges in significantly less replicates than StoNED.

Table 9 Estimator Robustness Analysis for Example 2: Bivariate Cobb-Douglas Frontier with Homoscedastic Inefficiency Terms

		Replicates for MSE *f* convergence		Standard Deviation of MSE *f*		MSE *f* coefficient of variation		% Negative Skew
ρ_{nts}	n	Sto NED	MBCR-I	Sto NED	MBCR-I	Sto NED	MBCR-I	Sto NED
1	100	50	10	0.0024	**0.0013**	113%	**40%**	10%
	200	50	10	0.0017	**0.0007**	120%	**48%**	10%
	300	50	40	0.0020	**0.0013**	145%	**65%**	10%
	500	10	20	**0.0002**	0.0007	**56%**	69%	0%
2	100	20	10	0.0032	**0.0017**	56%	**40%**	15%
	200	20	10	0.0038	**0.0017**	75%	**43%**	20%
	300	20	20	0.0024	**0.0015**	54%	**39%**	20%
	500	10	20	0.0023	**0.0017**	**49%**	58%	0%
3	100	20	10	0.0164	**0.0045**	69%	**53%**	15%
	200	20	10	0.0130	**0.0012**	50%	**27%**	5%
	300	20	20	0.0120	**0.0024**	**39%**	48%	0%
	500	10	10	0.0141	**0.0013**	52%	**34%**	10%

Example 3: Trivariate Cobb-Douglas frontier with homoscedastic inefficiency terms

We consider the Trivariate Cobb-Douglas frontier, $y_i = x_{1,i}^{0.4} x_{2,i}^{0.3} x_{3,i}^{0.2} e^{-u_i} e^{v_i}$. The distributional assumptions for the noise and inefficiency terms are the same as those of Examples 1 and 2, $u_i \sim \text{expo}(\mu_u = 1/6)$ and $v_i \sim N(0, \sigma_v^2)$. Prior assumptions on θ are the same as in previous examples. Table 10 shows that in this higher-dimensional setting, CNLS has a poor functional fit, MSE *f*, for scenarios with a large number of observations, n = 500, and for scenarios with a high noise-to-signal ratio, $\rho_{nts} = 3$.[22]

Table 10 Results for Example 3: Trivariate Cobb-Douglas Frontier with Homoscedastic Inefficiency Terms

		MSE *f*			$\widehat{E(\bar{u})} - E(\bar{u})$			% Non-Full Dimensional	
ρ_{nts}	n	Sto NED	MBCR-INS	MBCR-IS	Sto NED	MBCR-INS	MBCR-IS	Sto NED	MBCR-INS
1	100	**0.0015**	0.0055	0.0056	**-0.01**	**-0.01**	0.07	33%	**5%**
	200	**0.0014**	0.0030	0.0033	-0.02	**0.01**	0.06	25%	**12%**
	300	**0.0020**	0.0026	0.0028	**0.02**	**0.02**	0.05	23%	**1%**
	500	0.0070	0.0019	**0.0013**	-0.08	**0.01**	0.03	6%	**0%**
2	100	**0.0053**	0.0078	0.0072	**0.00**	-0.07	0.04	38%	**8%**
	200	**0.0054**	0.0060	0.0070	**0.03**	-0.07	0.09	38%	**10%**
	300	0.0075	0.0047	**0.0045**	0.07	**0.04**	**0.04**	34%	**4%**
	500	0.0147	0.0039	**0.0031**	0.14	-0.07	**0.01**	**6%**	**6%**
3	100	0.0286	**0.0076**	0.0088	0.12	-0.10	**0.01**	42%	**2%**
	200	0.0243	0.0077	**0.0063**	0.12	-0.10	**0.01**	38%	**1%**
	300	0.0218	0.0058	**0.0042**	0.13	-0.10	**-0.02**	32%	**3%**
	500	0.0257	0.0066	**0.0034**	0.11	-0.12	**-0.04**	14%	**0%**

[22] Again, the StoNED results from our GAMS with MINOS 5.5 implementation were similar or better than Kuosmanen and Kortelainen (2012). Kuosmanen and Kortelainen (2012) did not consider n =500 scenarios. To make the CNLS problem feasible to solve, we reduced our optimality tolerance from 10^{-10} to 10^{-2} for the n =500 scenarios.

While the results for the noise-to-signal ratio equal to 1 scenarios are similar to Examples 1 and 2, with the MBCR-I estimators only being competitive in some of the scenarios in Example 3, the proportion of observations supported by non fully-dimensional hyperplanes fit by StoNED first-stage, CNLS, increases and ranges between 6% and 42%. For larger noise-to-signal ratios, ρ_{nts}=2,3, the performance comparison is similar to Examples 1 and 2, with MBCR-I S performing the best in most of the scenarios. The variability in the functional fit, Standard Deviation of MSE f, and negative skewness results in Table 11 show behavior similar to Examples 1 and 2, with MBCR-I showing a lower inter-replicate variability in functional fit, MSE f.

Example 4: Trivariate Cobb-Douglas frontier with heteroscedastic inefficiency terms

We consider a heteroscedastic inefficiency DGP, where $u_i|\mathbf{x}_i \sim N^+(0,\sigma_{0u}(x_{1,i}+x_{2,i}))$, where σ_u=0.3. The noise distribution is a homoscedastic Normal $v_i \sim N(0,\sigma_v^2)$, where $\sigma_v = \rho_{nts}\cdot\sigma_u\cdot\sqrt{(\pi-2)/\pi}$. The production frontier is $y_i = x_{1,i}^{0.4}x_{2,i}^{0.3}x_{3,i}^{0.2}e^{-u_i}e^{v_i}$, as in Example 3. The hyperparameter for our Exponential prior on inefficiency, θ, is lower than in Examples 1, 2, and 3 due to the scale of the data, although still nearly uninformative with v_0=1 and w_0 drawn uniformly from the range (0, 0.1). Unlike Tables 6, 8, and 10, we include an additional set of results for StoNED from a different implementation.[23] We note that even if this

[23] The GAMS with MINOS 5.5 implementation is our main implementation for this example. The additional results are from our MATLAB fmincon implementation. Our main implementation was similar to the MSE *f* results in Kuosmanen and Kortelainen (2012) for all $\rho_{nts} = 1$ scenarios. None of our implementations achieved Kuosmanen and Kortelainen (2012)'s $\rho_{nts} = 2, n = 300$ MSE *f* results.

alternative implementation has better functional fit, MSE *f* has inconsistent behavior with regard to the percentage or negatively-skewed replicates.

Table 11 Estimator Robustness Analysis for Example 3: Trivariate Cobb-Douglas Frontier with Homoscedastic Inefficiency Terms

		Replicates for MSE *f* convergence		Standard Deviation of MSE *f*		MSE *f* coefficient of variation		% Negative Skew
ρ_{nts}	n	Sto NED	MBCR-I	Sto NED	MBCR-I	Sto NED	MBCR-I	Sto NED
1	100	20	10	**0.0006**	0.0020	39%	**37%**	0%
	200	50	10	**0.0010**	0.0012	71%	**41%**	4%
	300	50	20	0.0033	**0.0011**	163%	**43%**	2%
	500	10	10	0.0054	**0.0008**	76%	**40%**	0%
2	100	20	10	**0.0025**	0.0027	48%	**34%**	15%
	200	20	10	0.0025	**0.0017**	46%	**29%**	10%
	300	20	20	0.0065	**0.0021**	87%	**47%**	0%
	500	10	20	0.0057	**0.0015**	**38%**	**38%**	0%
3	100	50	20	0.0204	**0.0037**	71%	**48%**	10%
	200	20	10	0.0129	**0.0031**	53%	**41%**	10%
	300	20	10	0.0130	**0.0022**	60%	**39%**	10%
	500	10	10	0.0197	**0.0026**	77%	**40%**	10%

Table 12 shows that the functional fit, MSE *f*, for the MBCR-I estimators is lower for all scenarios, with MBCR-I NS having significantly better performance when the number of observations is small and both MBCR-I estimators perform similarly for the larger n scenarios. Reasons for MBCR-I's good functional fit are the updating of prior assumptions about the inefficiency term and incorporating hyperplane-specific noise variances into the posterior distribution of the observation's inefficiency term, u_i. Moreover, MBCR-I's mean inefficiency predictions are more accurate in 10 of the 12 scenarios. Conversely, StoNED's

predictions for the mean inefficiency level are highly biased (even for ρ_{nts}=1); however, this bias becomes smaller for our alternative implementation used in scenarios with 500 observations.

Comparing MBCR-I's results in Table 12 with MBCR-I's results for the correctly specified DGP in Table 10 shows that the heteroscedastic inefficiency specification only has a significantly detrimental impact on MSE f for 4 of the 12 scenarios. Besides these, the MSE f results for the heteroscedastic inefficiency example are at most 20% larger, and in fact smaller in the majority of scenarios. For both StoNED and MBCR-I, the percentage of observations supported by non fully-dimensional hyperplanes is similar to Example 3. We conclude that even for moderate sample sizes and large noise-to-signal ratios, MBCR-I is relatively robust to mis-specification of the inefficiency term. Finally, the DGP mis-specification impacts the variability of MBCR-I's functional fit across replicates less than StoNED's as measured by the Standard Deviation of MSE f column in Table 13.

Table 12 Results for Example 4: Trivariate Cobb-Douglas Frontier with Heteroscedastic Inefficiency Terms

		MSE *f*			$E(\widehat{\bar{u}})$ -$E(\bar{u})$			% Non-Full Dimensional	
ρ_{nts}	n	Sto NED	MBCR-I NS	MBCR-I S	Sto NED	MBCR-I NS	MBCR-I S	Sto NED	MBCR-I NS
1	100	0.0058	**0.0046**	0.0071	0.10	**0.02**	0.11	25%	**6%**
	200	0.0047	**0.0025**	0.0033	0.10	**0.02**	0.07	24%	**4%**
	300	0.0047	**0.0033**	0.0045	0.10	**0.06**	0.10	23%	**1%**
	500	0.0499, 0.0028	**0.0015**	0.0019	0.33, **0.02**	0.04	0.04	27%	**1%**
2	100	0.0507	**0.0127**	0.0194	0.28	**0.12**	0.22	45%	**0%**
	200	0.0407, 0.0158	**0.0064**	0.0126	0.27, 0.08	**0.07**	0.16	35%	**5%**
	300	0.0590, 0.0191	**0.0042**	0.0052	0.33, 0.12	**0.00**	0.06	31%	**3%**
	500	0.1345, 0.0084	**0.0020**	0.0023	0.47, 0.05	**0.03**	**0.03**	12%	**0%**
3	100	0.1492	**0.0107**	0.0205	0.38,	**0.04**	0.18	48%	**0%**
	200	0.1955, 0.1286	**0.0061**	0.0068	0.50, 0.33	**0.03**	0.06	47%	**4%**
	300	0.2446, 0.1046	**0.0050**	**0.0050**	0.58, 0.29	**0.00**	0.04	30% 5%	**5%**
	500	0.2610, 0.0061	**0.0034**	0.0038	0.63**, -0.02**	0.04	0.04	20%	**5%**

Table 13 Estimator Robustness Analysis for Example 3: Trivariate Cobb-Douglas Frontier with Heteroscedastic Inefficiency Terms

		Replicates for MSE *f* convergence		Standard Deviation of MSE *f*		MSE *f* coefficient of variation		% Negative Skew
ρ_{nts}	n	Sto NED	MBCR-I	Sto NED	MBCR-I	StoNED	MBCR-I	StoNED
1	100	100	20	10	0.0041	**0.0027**	71%	**55%**
	200	200	20	10	0.0023	**0.0011**	48%	**44%**
	300	300	20	10	0.0028	**0.0017**	59%	**51%**
	500	500	10	10	0.0314, 0.0019	**0.0008**	63%, 66%	**58%**

Table 13 (continued)

		Replicates for MSE *f* convergence		Standard Deviation of MSE *f*		MSE *f* coefficient of variation		% Negative Skew
ρ_{nts}	n	Sto NED	MBCR-I	Sto NED	MBCR-I	StoNED	MBCR-I	StoNED
2	100	100	20	10	0.0405	**0.0079**	80%	**62%**
	200	200	20	10	0.0233, 0.0216	**0.0051**	**57%**, 136%	81%
	300	300	30	10	0.0399, 0.0173	**0.0032**	**68%**, 131%	75%
	500	500	10	20	0.0826, 0.0122	**0.0005**	61%, 146%	**26%**
3	100	100	50	20	0.1277, 0.1428	**0.0067**	86%, 108%	**63%**
	200	200	50	10	0.0954, 0.1088	**0.0018**	49%, 85%	**30%**
	300	300	40	10	0.1298, 0.1010	**0.0026**	53%, 142%	**51%**
	500	500	10	10	0.0641, 0.0028	**0.0011**	**25%**, 46%	33%

III.2.2 Discussion of Simulation Results and Recommendations to Use MBCR-I

As Tables 6–13 show, MBCR-I is best for scenarios with relatively noisy data and/or when the inefficiency distribution is unknown.[24] Relative to the benchmark method StoNED, MBCR-I is also competitive for lower noise-to-signal ratios in datasets where n≥300. Due to MBCR-I's one-stage nature, the residuals from this estimator are correctly skewed even for

[24] An example of possible mis-specification of the inefficiency term appears in public sector applications where firms do not compete and efficient behavior does not result. Therefore, the distribution of inefficiency is unlikely to have a mode of zero and thus both an exponential or half-normal assumption regarding the inefficiency distributions is mis-specified.

smaller datasets. Since the iteration-selection criterion is of no clear benefit for the sample sizes where MBCR-I is recommended, we choose the MBCR-I S estimator, because it eliminates the heuristic component of our algorithm. We also note that MBCR-I requires significantly higher computational times than CNLS for n<500, partly because it gives a full posterior distribution of results rather than a point estimate. Nevertheless, due to its adaptive regression nature, MBCR-I only fits regression parameters for subsets of the dataset and so computational time increases slowly in n. [25] At n=500, the computational time for both MBCR-I and Multiplicative-error StoNED is about 45 minutes.[26] MBCR-I is the only feasible existing axiomatic concavity constrained frontier estimation method to fit datasets with 1000≤n≤6000 observations. Thus, we recommend MBCR-I S for most datasets with 300≤n≤6000, or for smaller datasets when imposing the proper skewness on the residual distributions is needed. Oh et al. (2015) and Crispim Sarmento et al. (2015) analyze datasets where Multiplicative CNLS is not applicable due to dataset size, but both papers report successfully applying variants of MBCR-I.[27]

We also recommend MBCR-I when inference on the frontier is needed. As a Bayesian method, MBCR-I produces credible intervals for MBCR-I S, our smoothed estimator, unlike CAP and StoNED, where it is computationally burdensome, even for moderate datasets, to obtain inference results by running the method repeatedly followed by bagging, smearing, or random partitioning (Hannah and Dunson, 2012). Finally, we

---

[25] MBCR-I's computational time increased from ~10 min for $n = 100$ to ~45 min for $n = 500$, whereas it increased from ~0.03min for $n = 100$ to ~45 min for $n = 500$ for Multiplicative CNLS.
[26] The CNLS speed-up algorithm proposed by Lee et al. (2013) on an additive error setting did not show time savings in our multiplicative setting.
[27] While MBCR-I integrates inefficiency into MBCR, it also has computational differences, as described in Section 2.5. Some of these are exploited to allow the inclusions of z-variables as discussed in Section 4.

recommend MBCR-I when obtaining a higher rate of fully dimensional hyperplanes is an important property of the frontier to be estimated.

III.3. Extensions

We consider several extensions to MBCR-I to make the method useful for a larger set of applications. We note that more general models for each extension are possible and are avenues for future research.

III.3.1 Flexible Time Trend

MBCR-I is attractive for use with Panel data because it can fit shape constrained production frontiers for moderate datasets. To model technical progress over time, we consider a vector of dummy time effects to act as frontier-shifting factors (Baltagi and Griffin, 1988). We estimate the following model,

$$Y_{it}=f(\mathbf{X}_{it})e^{v_{it}}e^{-u_i}e^{\boldsymbol{\gamma}\mathbf{d}_{it}}, \qquad i = 1,..,n;\ t = 1,\dots T. \tag{III.11}$$

We let $\boldsymbol{\gamma}=(\gamma_2,\dots,\gamma_T)$ and $\mathbf{d}_{it}$ is a row vector of dummy variables which has a 1 on the (t-1)*th* entry (and is a zero vector for observations on the first time period) and zeros on all other entries. Recalling MBCR-I's Gaussian mixture likelihood function, we know the hyperplane-specific noise variances $(\sigma_k^2)_{k=1}^K$ from the MBCR step of our algorithm (step 1). We let $\mathbf{D}=(\mathbf{d}_{11},\dots,\mathbf{d}_{nT})'$, collect the $(\sigma_{[i]}^2)_{i=1}^n$ terms on diagonal matrix Σ_v and consider the Multivariate Normal prior $\boldsymbol{\gamma}\sim MVN(\mu_{\gamma 0},\Sigma_{\gamma 0})$ to obtain the conjugate posterior shown in (12). Then, we add a step between steps 3 and 4 to draw $\boldsymbol{\gamma}^{(t)}$. Specifically,

$$\boldsymbol{\gamma}|\dots \sim MVN\left(\mu_{\gamma 1},\Sigma_{\gamma 1}\right), \quad k=1,\dots, K, \text{ where} \tag{III.12}$$

$$\mu_{\gamma 1}= (\Sigma_{\gamma 0}^{-1}+\mathbf{D}'\Sigma_v^{-1}\mathbf{D})^{-1}(\Sigma_{\gamma 0}^{-1}\mu_{\gamma 0}+\mathbf{D}'\Sigma_v^{-1}\mathbf{D}\hat{\boldsymbol{\gamma}})$$

$$\Sigma_{\gamma 1}= (\Sigma_{\gamma 0}^{-1}+\mathbf{D}'\Sigma_v^{-1}\mathbf{D})^{-1},$$

where $\hat{\boldsymbol{\gamma}}$ is the OLS estimator of $r_i=\ln(Y_i)-\ln(\hat{f}_i)+u_i$ using $\mathbf{D}$ as a predictor. Since drawing from (III.12) is not computationally demanding, this estimator gives roughly the same computational performance as MBCR-I for nT observations, thus opening the possibility to fit nonparametric multiplicative production frontiers to Panel datasets up to a few thousand observations. To adapt the MBCR-I algorithm to Panel data, we modify equations (III.9a) or (III.9b) to draw $(u_i)_{i=1}^n$ using information from the whole Panel rather than one observation. In the exponential inefficiency prior case, we consider the posterior hyperparameters analogous to O'Donnell and Griffiths (2006), who also consider a Gaussian mixture likelihood.

$$\sigma_{ui}^2=\left[\sum_{t=1}^T \sigma_{[it]}^{-2}\right]^{-1}, \qquad \mu_{ui}=\left\{\sum_{t=1}^T (\sigma_{[it]}^{-2})\left[\ln(\hat{f}_i)+\boldsymbol{\gamma}\mathbf{d}_{it}-\ln(Y_i)\right]-\theta\right\}\sum_{t=1}^T \sigma_{[it]}^{-2} \qquad \text{(III.13)}$$

III.3.2 Contextual Variables

Incorporating contextual variables allows MBCR-I to estimate production functions that are multiplicatively affected by factors beyond a firm's control, or factors that are not inputs, but that the firm can control (see Johnson and Kuosmanen, 2011). We consider a parametric specification for the effect of contextual variables and fit the model

$$Y_i=f(\mathbf{X}_i)e^{v_i}e^{-u_i}e^{\boldsymbol{\delta}\mathbf{z}_i}, \qquad i=1,..,n \qquad \text{(III.14)}$$

where $\mathbf{z}_{it}=(z_{1it},\ldots,z_{Rit})'$ is the R-dimensional vector of z-variables for firm *i* at time *t* and $\boldsymbol{\delta}=(\delta_1,\ldots,\delta_R)$ are the coefficients for each contextual variable. We note that (III.14) is the same as (III.11), with $\boldsymbol{\delta}$ and $\mathbf{z}_i$ playing the role of $\boldsymbol{\gamma}$ and d_{it}, respectively. Thus, the posterior

simulation approach is the same as (III.12) with matrix $\mathbf{Z}=(\mathbf{z}'_{11},\ldots,\mathbf{z}'_{nT})'$ in place of $\mathbf{D}$ and the prior parameters $(\mu_{\delta 0},\Sigma_{\delta 0})$ in place of $(\mu_{\gamma 0},\Sigma_{\gamma 0})$. We can also include both contextual variables and time effects with a common covariance matrix between their regression coefficients.

III.4 Empirical Application

Japan's recent plan for economic reforms, popularly termed "Abenomics", is lead by Prime Minister Shinzo Abe. The Prime Minister has recommended increased activity in the construction sector as an important economic driver, yet many politicians and pundits argue that Japan's construction industry is inefficient ("Japan and Abenomics", 2013). Our empirical application investigates the efficiency of operations in the Japanese concrete industry, a critical component of the country's construction industry.

We construct a dataset using the Census of Manufacturers collected by Japan's Ministry of Economy, Trade and Industry (METI) for concrete products. The data include all establishments in Japan with at least four workers.[28] We define Capital and Labor as the input variables and Value Added as the output variable. Capital and Value Added are measured in tens of thousands of Yen and Labor is measured in number of Employees. We report results for Cross Sectional datasets for 2007 and 2010 and a balanced Panel dataset for 2007-2010. See Appendix E for the Cross Sectional results for 2008 and 2009. We note even our smallest Cross Sectional dataset of 1,652 observations exceeds the capacity of existing

[28] The equation used to calculate value added is different for firms with more than 30 employees than for firms with less than 30 employees. Excluding the firms with more than 30 employees did not result in significant changes to our frontier estimates, thus we present results for the full dataset.

methods to fit a semi-nonparametric shape constrained production function. Table 14 presents summary statistics.

Table 14 Descriptive Statistics for Concrete Products Dataset

Year		2007	2008	2009	2010
Number of observations – Cross Sectional		1,929	1,715	1,714	1,652
Capital	Mean	16,419	15,809	15,662	14,215
	Median	2,000	2,000	2,000	2,000
	Standard Deviation	76,401	74,300	69,332	58,486
Labor	Mean	16.68	17.51	16.60	16.32
	Median	12	13	12	12
	Standard Deviation	15.80	15.83	15.39	15.29
Year		2007	2008	2009	2010
Value Added	Mean	19,004	18,481	18,393	16,520
	Median	11,515	11,391	11,092	10,587
	Standard Deviation	26,221	21,619	20,842	17,407

Year		2007	2008	2009	2010
Number of observations – Panel		1,382	1,382	1,382	1,382
Capital	Mean	16,313	16,475	15,901	15,536
Year		2007	2008	2009	2010
Capital	Median	2,000	2,000	2,000	2,000
	Standard Deviation	77,269	77,426	70,606	63,081
Labor	Mean	18.44	18.25	17.72	17.09
	Median	14	14	14	13
	Standard Deviation	16.38	16.20	15.91	15.75
Year		2007	2008	2009	2010
Value Added	Mean	20,084	19,434	19,377	17,648
	Median	13,502	12,385	12,548	11,795
	Standard Deviation	22,475	20,264	20,665	17,839

We specify prior values for the parameters K, $(\sigma_k^2)_{k=1}^K$, v_0, w_0, θ, $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$. To determine the prior for K, we run multiplicative-error MBCR (without modeling inefficiency), and hypothesize that this prior value of K is likely to be larger than the number of hyperplanes estimated by MBCR-I, because MBCR will capture more of the output variability with functional complexity, whereas MBCR-I can also use inefficiency. Unlike the K=1 assumption, our MBCR-based prior on K implies that we have the prior belief that the frontier has curvature and is more complex than a linear function. See Appendix F for a detailed discussion.[29] Additionally, we consider priors v_0=1, w_0=0.1 and θ=$v_0 w_0$ for the inefficiency-related parameters. Finally, we consider a wide-support, nearly uninformative prior IG(1,0.01) for each σ_k^2. In the Panel model, we consider $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ to have near-vague MVN(0,M) prior distributions, where M=$\text{diag}_T(2000)$.

Due to our multiplicative error structure and the use of logarithms, we eliminate firms with negative Value Added (~1% of the initial observations). After an initial fit, 3% of the observations significantly deviate from our prediction. These observations correspond to the largest observations in terms of Capital, Value Added, or both, and thus we exclude these observations as outliers. Firm exclusion is consistent across all datasets.

III.4.1 Posterior Estimated Frontier and Interpretation

Table 15 shows the fitting statistics for the two datasets. The number of hyperplanes needed by our Gaussian Mixture model is relatively small, which implies that firms operate

[29] This assumption did not have a significant impact on our parameter estimates compared to the $K = 1$ assumption, but helped MBCR-I to be more computationally efficient and resulted in the insights summarized in *Appendix F*.

in a few clusters with locally constant marginal productivity. We estimate a median inefficiency of 28-29% for the Cross Sectional and 45% for the Panel datasets, respectively. The percentage of output variability explained by the joint production function and inefficiency model is between 66-76% for the Cross Sectional and above 80% for the Panel datasets, respectively. The number of hyperplanes fitted for the Panel dataset is larger, due to both MBCR-I's ability to produce finer estimates of the production function when more data is available, and to the lower noise level of the firms operating throughout the study period. We report the percentage of output variation explained by our model and the part that remains unexplained we report as noise in the columns labeled % Model and % Noise, respectively.

Table 15 MBCR-I S Cross Sectional and Panel Production Frontier Fitting Statistics

	# Hyperplanes fitted	Median Inefficiency	% Model	% Noise
2007	2.11	29%	66%	34%
2008	2.03	28%	75%	25%
2009	2.60	29%	70%	30%
2010	2.36	28%	76%	24%
Panel	3.63	45%	81%	19%

Tables 16, 17 and 18 show the economic quantities of interest for MBCR-I S, i.e., the marginal productivities of Capital and Labor, Capital to Labor Elasticity of Substitution and Technical Efficiency of the fitted production frontier models according to their minimum, maximum, and quartile-specific values across observations. Table 18 shows Technical Efficiencies on a firm-specific basis, rather than an observation-specific basis. While we

observe non-constant elasticities of substitution across the samples, Tables 16 and 17 show that a single component of their respective Gaussian mixtures support at least half of the observations, indicating that a majority group faces the same marginal productivity and Elasticity of Substitution. Regarding Technical Efficiency, in 2010 413 out of 1,652 establishments (25%) operate below 53% efficiency. This level of inefficiency is consistent across all years and the Panel dataset analysis, indicating a large potential for improvement for this subset of establishments. The Cross Sectional results for 2007 and 2010 (and the 2008 and 2009 results in Appendix E), are similar in most of the estimated frontier and efficiency characteristics, indicating firm's consistency performance over time. Table 18 shows that the Panel data model gives a smoother, more detailed estimate of the frontier. Lastly, firm-specific technical efficiencies are lower at all quantiles for Cross Sectional versus the Panel data results.

Table 16 MBCR-I S Cross Sectional Production Frontier Characterization for 2007

	Marginal Product		Elasticity of Substitution ($x10^{-4}$)	Technical Efficiency
	Capital	Labor	Capital/Labor	Firm-specific %
Min	0.0446	1,013	0.4278	0.060%
25th percentile	1.0534	1,222	8.6449	50.89%
Median	1.0534	1,222	8.6449	70.94%
75th percentile	1.0534	1,222	8.6449	86.61%
Max	3.7182	1,218	43.189	99.97%

Table 17 MBCR-I S Cross Sectional Production Frontier Characterization for 2010

	Marginal Product		Elasticity of Substitution ($x10^{-4}$)	Technical Efficiency
	Capital	Labor	Capital/Labor	Firm-specific %
Min	0.0289	803.1	0.3342	3.46%
25th percentile	0.9880	1074	9.2108	52.98%
Median	0.9880	1074	9.2108	72.80%
75th percentile	0.9880	1074	9.2108	87.66%
Max	6.7196	1074	98.7883	97.97%

Table 18 MBCR-I Panel Production Frontier Characterization

	Marginal Product		Elasticity of Substitution ($x10^{-4}$)	Technical Efficiency
	Capital	Labor	Capital/Labor	Firm-specific %
Min	0.0299	988	0.2850	2.78%
25th percentile	0.8758	1173	7.2168	36.53%
Median	1.8886	1318	14.0175	55.40%
75th percentile	3.0354	1417	21.4405	77.80%
Max	3.7157	1971	26.3485	99.95%

The lower Technical Efficiency levels for the Panel model are conservative estimates, because as Table 19 shows, the frontier contracts in each time period after 2007 are likely due to the global financial crisis. Given that 2007 is the base year for the Panel analysis and that our assumption about uniform yearly frontier shifts across firms is an approximation, we suggest that firm-specific inefficiency terms absorb some of the frontier shrinkage effect. Finally, we note that the frontier multiplier for year *t* is given by e^{γ_t}, as defined in Section

III.4.1. Table 19 shows Maximum a Posteriori (MAP)[30] and 90% credible interval values for the year-specific frontier shifts for the Panel model. Further, Table 19 also confirms that the global financial crisis of 2007 has a significant effect on the future output of establishments.

Table 19 Time Dummy Variable Coefficients

	2008	2009	2010
MAP	-0.0595	-0.0628	-0.1063
Frontier Multiplier	0.9422	0.9391	0.8992
90% Credible Interval	(0.9186, 0.9665)	(0.9154, 0.9649)	(0.8762, 0.9248)

Table 20 shows the Most Productive Scale size results, conditional on Capital/Labor ratio distributions at the 10, 25, 50, 75, and 90 percentiles (see Appendix G for a full discussion). The Capital/Labor ratio distribution is similar for the Cross Sectional and Panel datasets. The MPSS results for the cross sectional datasets are similar at all percentiles of the Capital/Labor ratio distribution. Finally, the larger MPSS values for the Panel frontier reflect a higher probability of well-established firms with superior technologies to appear in all years of the census. These firms have a large weight in determining the frontier in the Balanced Panel and contribute to lower Technical Efficiency levels.

[30] The MAP value is the highest density point of the simulated posterior distribution for the parameter of interest.

Table 20 Most Productive Scale Size for Selected Capital/Labor Ratios for Cross Sectional and Panel Production Frontiers

Model	Capital/Labor ratio percentile	10%	25%	50%	75%	90%
Cross Sectional 2007	Capital/Labor	37	75	166.7	400	1,500
	MPSS Capital	1,243	1,281	1,368	1,594	2,638
	MPSS Labor	32	17	8	4	2
Cross Sectional 2010	Capital/Labor	37.5	80.0	166.7	416.7	1451.2
	MPSS Capital	1,102	1,140	1,224	1,446	2,477
	MPSS Labor	28	14	8	4	2
Panel 2007–2010	Capital/Labor	38.9	76.9	162.2	389.9	1456.0
	MPSS Capital	1,490	1,596	1,827	2,438	5,278
	MPSS Labor	38	21	11	6	4

CHAPTER IV

EVALUATING MONOTONIC AND CONCAVE PRODUCTION FUNCTION ESTIMATORS FOR USE ON MANUFACTURING SURVEY DATA

In this chapter, we propose selecting an estimator based on a weighting of its in-sample and predictive performance on actual application datasets rather than on learning Mean Squared Error performance in Monte Carlo simulations. The method provides insights to Census Bureaus regarding the survey sizes needed to reliably estimate industry population-level production functions. We demonstrate this selection method for both parametric and shape-constrained nonparametric production function estimators. We compare Cobb Douglas functional assumptions, CNLS, CAP, and CAP-NLS. We find CAP-NLS has lower Mean Squared Error across learning set sizes compared to alternative nonparametric approaches when used to estimate a population production function for simulated data. For survey data, specifically the 2010 Chilean Annual National Industrial Survey and four regressors, we find that the best method depends on the specific dataset, but with CAP-NLS and additive-residual Cobb-Douglas specification performing similarly for at all sample sizes, and CAP matching their performance at $n>50$. Further, we find that the additive-residual Cobb-Douglas specification describes at least 90% as much variance as the best estimator in practically all considered cases. We construct simulation-based confidence intervals on the goodness-of-fit of a survey-fitted production function depending on its sample size, thus creating a survey size recommendation criterion for census bureaus.

The chapter is organized as follows: In Section IV.1, we present Convex Adaptively Partitioned Nonparametric Least Squares (CAP-NLS), a method that integrates CAP and

CNLS using an adaptive partitioning strategy, but at the same time uses the Afriat (1967; 1972) inequalities and global optimization. In Section IV.2, we perform Monte Carlo simulation analysis to demonstrate CAP-NLS's superior performance at estimating production functions relative to CAP and CNLS for both in-sample and learning set-to-testing set scenarios. In Section IV.3, we fit production data for the five industries with the largest sample sizes in the 2010 Chilean Annual National Industrial Survey and evaluate the performance of CAP-NLS relative to CAP, CNLS and the Cobb-Douglas parametric models in terms of its ability to produce reliable industry-wide production function estimates from non-exhaustive survey datasets.

IV.1 Convex Adaptively Partitioned Nonparametric Least Squares

IV.1.1 Production Function Model

We define the regression model for our nonparametric estimation procedure as

$$Y=f(\mathbf{X})+\varepsilon \tag{IV.1}$$

where Y represents observed output, $f(\mathbf{X})$ denotes the attainable output level given a certain input mix $\mathbf{X}=(X_1, \ldots, X_d)'$, d is the dimensionality of the input vector and ε is a symmetric random term, which we call *noise*, assuming a mean 0. For our estimator, we use the establishment-specific equation (IV.2) to derive our objective function.

$$Y_i=f(X_{1i}, \ldots, X_{di})+\varepsilon_i, \quad i=1,\ldots,n \tag{IV.2}$$

For notational simplicity, we let $f_i=f(X_{1i}, \ldots, X_{di})$ and $\mathbf{X_i}=X_{1i}, \ldots, X_{di}$. We describe the decreasing marginal productivity (concavity) property in terms of $\nabla f(\mathbf{X})$, i.e., the gradient of f with respect to X,

$$f(\mathbf{X}_i) \leq + f(\mathbf{X}_j) + \nabla f(\mathbf{X}_j)^T (\mathbf{X}_i - \mathbf{X}_j) \ \forall i,j. \quad \text{(IV.3)}$$

Given that the constraints in (IV.3) hold, the additional constraint $\nabla f(\mathbf{X}_i) > \mathbf{0}$ $\forall i$ imposes monotonicity.

IV.1.2 Convex Adaptively Partitioned Nonparametric Least Squares

Throughout this chapter, we consider nonparametric approximation of $f(\mathbf{X})$ with piecewise linear estimators. These estimators can consistently describe a general concave function allowing the concavity constraints in (IV.3) to be written as a system of linear inequalities. The CNLS estimator is a sieve estimator and the most flexible piecewise linear estimator consistent with the functional description in (IV.1)-(IV.3), Kuosmanen (2008). Nevertheless, this estimator imposes condition (IV.3) by a set of numerous pairwise constraints, which requires significant computational enhancements to be applied on moderate datasets (see Lee et al., 2013 and Mazumder et al., 2015). More importantly, CNLS results in a parameter-intensive representation of $f(\mathbf{X})$, as it allows for potentially *N* distinct hyperplanes. This highly detailed sample-specific fit limits the estimator's ability to predict the performance unobserved establishments from the same industry. Moreover, even if a full census is fitted with CNLS, researchers have ignored that the CNLS fit is specific to the particular vector of (unobserved) noise terms and thus descriptive ability on the observed dataset does not fully represent descriptive ability on the unobserved set of establishments. The extent to which CNLS over fits the data and the observed noise vector will be explored and illustrated in Sections IV.2 and IV.3. Furthermore, allowing for such a large number of distinct hyperplanes is an issue for production function estimation also from a purely economic point of view, because individual establishment observations can specify their own

vector of marginal products meaning they can place zero weight on some set of inputs excluding those inputs from that establishment's production function. This implies that even if the establishment uses those inputs intensively, the establishment can ignore those inputs recorded in the data when evaluating the establishment's performance.

To address these computational and representational issues, we propose to partition the dataset into input-space defined subsets, which we will refer to as *Basis Regions* and estimate one hyperplane per basis region. We use the notation [i] to denote the index of the basis region to which observation i is assigned for a given input set partitioning proposal and let K be the number of basis regions. Then, we can approximate concave function $f(\mathbf{X})$ at input vector $\mathbf{X}_i$ with estimator

$$\hat{f}_K(\mathbf{X}_i) = \beta^*_{0[i]} + \boldsymbol{\beta}^{*T}_{-0[i]}\mathbf{X}_i \tag{IV.4}$$

$$\left(\beta^*_{0k}, \boldsymbol{\beta}^*_{-0k}\right)_{k=1}^{K} = \underset{(\beta_{0k},\ \boldsymbol{\beta}_{-0k})_{k=1}^{K}}{\operatorname{argmin}} \sum_{i=1}^{n} \left(\beta_{0[i]} + \boldsymbol{\beta}^{T}_{-0[i]}\mathbf{X}_i - Y_i\right)^2$$

$$\text{s.t. } \beta_{0[i]} + \boldsymbol{\beta}^{T}_{-0[i]}\mathbf{X}_i \leq \beta_{0k} + \boldsymbol{\beta}^{T}_{-0k}\mathbf{X}_i \quad \forall\, i=1,\ldots N, k=1,\ldots,K$$

$$\boldsymbol{\beta}_{-0k} \geq \mathbf{0} \ \forall k=1,\ldots,K,$$

where the *kth* basis region is fitted by a hyperplane with parameters $\boldsymbol{\beta}_k = (\beta^*_{0k}, \boldsymbol{\beta}^*_{-0k})$. Note that the total number of Afriat inequality constraints is NK, as opposed to the N(N-1) constraints implied by (IV.3). Further, note that (IV.4) estimates $f(\mathbf{X})$ *conditionally* on an input-space partition. Thus, to obtain an unconditional estimator of $f(\mathbf{X})$, we need to explore different input-space partitions. Consequently, we need to nest the solution to problem (IV.4) into an algorithm that proposes partitions resulting in a more parsimonious estimator of $f(\mathbf{X})$ than the CNLS solution.

Hannah and Dunson (2013) propose a partitioning strategy, which both explores the input space adaptively and easily allows nesting. Firstly, their strategy greedily selects models with incrementally better fits as the number of hyperplanes grows. Secondly, they transition from simpler (initially linear) to more detailed models of the concave function and select a model that results in the best tradeoff between model fit and the number of parameters used. We estimate the function $\hat{f}(\mathbf{X})$ by iteratively solving (IV.4) inside of the partitioning proposal strategy. At each iteration, the strategy evaluates KML partition-splitting proposals, where M, a tunable parameter, is the number of random input-space location proposals for a new knot at each iteration, L=d is the number of randomly proposed directions, given the current basis regions and a proposed new knot location, that will define the new dataset partition, and K is the current number of partitions at the current iteration. Our full estimation algorithm, which nests (IV.4) in the adaptive partitioning strategy, can be succinctly described by the following algorithm where tunable parameter n_0 is bounded below by 2(d+1):

Algorithm 1. CAP-NLS Estimator

1. Start with K=1 and fit (IV.4).
2. Consider splitting each of the current K hyperplanes at M random knot locations in L random directions.
3. Fit (IV.4) for each for each of the (at most) KML partition proposals with at least $n_0/2$ observations on each basis region.
4. Select the proposal that minimizes MSE and let K=K+1.

5. Check if each of the current partitions has at least n_0 observations. If yes, stop; if not go to Step 2

To ensure model parsimony, we select the smallest model (in terms of K) for which MSE is within a prespecified tolerance of the MSE of the largest Kconsidered by Algorithm 1[31]. Even though CAP and CAP-NLS use the same partitioning strategy, there are three main differences between CAP-NLS and CAP. Firstly, CAP-NLS imposes concavity via the Afriat Inequalities, rather than a minimum-of-hyperplanes construction. Secondly, CAP-NLS requires solving a global optimization problem rather than multiple localized optimization problems. As we will observe in our experiments on Sections IV.2 and IV.3, this additional structure results in increased rates of convergence and improved robustness against local monotonicity violations common in manufacturing survey data. Thirdly, CAP-NLS does not require a refitting step, as the observation-to-basis region correspondence is kept before and after fitting problem (IV.4). Furthermore, CAP-NLS is differentiated from CNLS not only in its one-to-many observation to hyperplane mapping and iterative nature, but also because CAP-NLS requires at least 2(d+1) observations per partition to fit each hyperplane. This property is the key for CAP-NLS's superior performance on the learning-to-full sampling settings.

[31] The tolerance is set to 1% in all of our examples. Initially we do not use the Generalized Cross Validation (GCV) score approximation used by Hannah and Dunson (2013) because they state GCV's predictive results are only comparable with full cross validation strategies for problems with $n \geq 5000$, which are larger than the datasets we consider.

IV.1.3 CAP-NLS as a Series of Quadratic Programs

Taking advantage of the linearly-constrained quadratic programming structure of CAP-NLS is essential to make it computationally feasible. We first need to write Problem (IV.4) in the standard form

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\boldsymbol{\beta}^T H\boldsymbol{\beta}+\boldsymbol{\beta}^T g \qquad \text{(IV.5)}$$

$$\text{s.t. } A\boldsymbol{\beta}\leq 0, \quad \boldsymbol{\beta}\geq \mathbf{l}.$$

Thus, starting with the objective function from (IV.4), we let $\widetilde{\mathbf{X}}=(\mathbf{1}, \mathbf{X})$ and write

$$\min_{(\beta_{0k},\, \boldsymbol{\beta}_{-0k})_{k=1}^K} \sum_{i=1}^n \left(\beta_{0[i]}+\boldsymbol{\beta}_{-0[i]}^T\mathbf{X}_i-Y_i\right)^2 = \min_{(\boldsymbol{\beta}_k)_{k=1}^K} \sum_{i=1}^n \left(\boldsymbol{\beta}_{[i]}^T\widetilde{\mathbf{X}}_i-Y_i\right)^2 =\ldots \qquad \text{(IV.6)}$$

$$= \min_{(\boldsymbol{\beta}_k)_{k=1}^K} \frac{1}{2}\sum_{i=1}^n \left(\boldsymbol{\beta}_{[i]}^T\widetilde{\mathbf{X}}_i\right)^2 - \sum_{i=1}^n \left(\boldsymbol{\beta}_{[i]}^T\widetilde{\mathbf{X}}_i Y_i\right)$$

where we have dropped constant $\sum_{i=1}^n Y_i^2$ and multiplied times one half. To write the last expression in (IV.6) in standard form, we first write $\sum_{i=1}^n \left(\boldsymbol{\beta}_{[i]}^T\widetilde{\mathbf{X}}_i\right)^2$ using matrix operations. We define observation-to-hyperplane $n(d+1)\times K(d+1)$-dimensional mapping matrix P with elements $P\big((i-1)*(d+1)+j,([i]-1)*(d+1)+i\big)=\widetilde{X}_{ij}$, $i=1,\ldots,n$, $j=1,\ldots d+1$ and all other elements equal to zero. Similarly, we define $n\times n(d+1)$-dimensional observation-specific vector product matrix S, with elements $S(i,(i-1)*(d+1)+l)=1$ for $l=1,..,3$, $i=1,\ldots,n$, $j=1,\ldots d+1$. We then concatenate vectors $(\boldsymbol{\beta}_k)_{k=1}^K$ in $K(d+1)\times 1$-dimensional vector $\boldsymbol{\beta}$. It follows that

$$\sum_{i=1}^n \left(\boldsymbol{\beta}_{[i]}^T\widetilde{\mathbf{X}}_i\right)^2 =\boldsymbol{\beta}^T P^T\left(S^T S\right)P\boldsymbol{\beta} \quad \text{and} \quad \sum_{i=1}^n \left(\boldsymbol{\beta}_{[i]}^T\widetilde{\mathbf{X}}_i Y_i\right) =\boldsymbol{\beta}^T P^T S^T\mathbf{Y}, \qquad \text{(IV.7)}$$

from which we can easily see that $H=P^T\left(S^T S\right)P$ and $g=-P^T S^T\mathbf{Y}$. To write in the Afriat Inequality constraints as $nK\times K(d+1)$ - dimensional matrix A, we let elements $A\big(K(i-1)+k,j+(d+1)([i]-1)\big)= \widetilde{X}_{ij}$, $i=1,\ldots,n$, $j=1,\ldots d+1$, $k=1,\ldots,K$ and let all other

elements equal zero. Finally, we define $K(d+1)$ – dimensional vector $\mathbf{l}$ to have elements $\mathbf{l}\left((k-1)(d+1)+1\right)=0$, $k=1,\ldots,K$ and all other elements be equal to negative infinity.

IV.2 Experiments on Simulated Data

We compare CAP-NLS to a correctly specified parametric estimator, a monotonically-constrained version of CAP, and CNLS via Monte Carlo-simulations. We consider Data Generation Processes (DGP) based on Cobb-Douglas functions and observe our estimates for the expected *in-sample* error of the production function estimators against the true DGP, $E(Err_{ISf})$, where the expectation is taken against all possible learning sets. Contrastingly, we also estimate the following expected quantities: *learning-to-testing set* or *predictive* error against the true DGP, (Err_f) , *in-sample error against observed output*, $E(Err_{ISy})$, and *predictive error against observed output* $E(Err_y)$ (Hastie, Tibshirani and Friedman, 2009 pp. 228-229). While performance against the true DGP is important to assess the functional estimator's finite-sample bias, the functional estimator's estimates compared to the observed output are the only performance metrics available to the researcher when considering a real dataset. On a real dataset, the estimate of in-sample error against observed output would be the most reliable fitting diagnostic when working with a *census* or *full set* of establishments. Conversely, the estimator's estimate compared to an additional sample drawn from the same DGP, which defines the expected *predictive error*, is the primary diagnostic when assessing the fit of a functional estimator obtained from estimation on a learning set relative to a much larger population. Consequently, assessing the fit of a functional estimator

on a finite census from a non-exhaustive sample, would have to weight these two errors by the relative sizes of the sets of observed and unobserved establishments.

Thus, we estimate the expected in-sample error against the true DGP for a learning set of size nLearn , $E(Err_{ISf}^{nL})$, by $E\widehat{(Err_{ISf}^{nL})}= \overline{MSE_{ISf}^{nL}}= \sum_{v=1}^{V} \sum_{i=1}^{n_L} \left(\hat{f}_{vLi}^{nL}-f_{vLi}\right)^2 /nV$ [32] for each functional estimator**,** where $\hat{f}_{vL}^{nL}$ is the production function estimate obtained with the vth learning set and learning set of size nL, f_{vLi} is the ith observation of the vth learning set, n_L is the size of the learning set and V is the number of different considered learning sets. Analogously, we estimate the expected predictive error against the true DGP for a learning set of size nLearn, $E(Err_{f}^{nL})$, by computing the averaged Mean-Squared Error across the V learning-testing set combinations of the same DGP, $E\widehat{(Err_{f}^{nL})}= \overline{MSE_{f}^{nL}}= \sum_{v=1}^{V} \sum_{i=1}^{n_T} \left(\hat{f}_{vLi}^{nL}-f_{vTi}\right)^2 /nV$, where we choose the size of the testing set, $n_T = 1000$, and f_{vTi} is the ith output observation of the vth testing set. When estimating $E(Err_{ISy}^{nL})$, unlike when estimating $E(Err_{ISf}^{nL})$ we need to vary the random component of each observation of each learning set in order to avoid over-optimism (Hastie, Tibshirani and Friedman, 2009 pp. 228). Thus, we generate W different sets of noise terms[33] for each learning set and estimate $E\widehat{(Err_{ISy}^{nL})}= \overline{MSE_{yIS}^{nL}}= \sum_{v=1}^{V} \sum_{w=1}^{W} \sum_{i=1}^{n_L} \left(\hat{f}_{vLi}- f(\mathbf{x}_{vLi})+\varepsilon_{wTi}\right)^2 /nVW$, where $\mathbf{x}_{vLi}$ is the ith input vector of the

[32] Note that the estimator "hat" character is over $E\left(Err_{ISf}^{nL}\right)$ rather than Err_{ISf}^{nL}, the in-sample error for the particular learning set with which the production function was fitted.

[33] Computing the in-sample error provides a more realistic estimate of the quality of the production function on a full set than the *learning error* $\overline{MSE_{Ly}^{nL}} = \sum_{v=1}^{V} \sum_{i=1}^{n_L} \left(\hat{f}_{vi} - Y_{vi}\right)^2 /nv$, as it averages performance across many possible ε_i residual values for the learning set input vector.

vth learning set and ε_{wTi} is the ith residual of the wth testing set. Finally, we compute $E\widehat{(Err_{y}^{nL})}= \overline{MSE_{y}^{nL}}= \sum_{v=1}^{V} \sum_{i=1}^{n_T} (\hat{f}_{vLi}-Y_{vTi})^2/nV$ to estimate the predictive error against observed outputs, where Y_{vTi} is the ith output observation of the vth testing set.

We consider results for two types of scenarios, the *full-sample* or *census* scenarios, and *learning set-to-full set* with finite full set scenarios. For the *full-sample* scenarios, we report $E\widehat{(Err_{lSf}^{n})}$ and $E\widehat{(Err_{lSy}^{n})}$. For the *learning set-to-full set* scenarios, we compute an estimator for the *Full Set* error

$$E\widehat{(Err_{FS\cdot}^{nL})}=\overline{MSE_{FS}^{nL}}=(n_L/n)\, E\widehat{(Err_{lS\cdot}^{nL})}+((n_F-n_L)/n)\, E\widehat{(Err_{\cdot}^{nL})} \qquad \text{(IV.8)}$$

,where FS stands for Full Set and this can be calculated for either f or y allowing f or y take the place of the dot operator. Note that $E\widehat{(Err_{lSy}^{nL})}$, $E\widehat{(Err_{y}^{nL})}$ and $E\widehat{(Err_{FSy}^{nL})}$ are also estimators for the noise level σ^2 of the DGP and thus we can use σ^2 as a benchmark for their estimations. Further, without our corrections for over-optimism computing an estimator $\hat{\sigma}^2$ would be complicated by the nonparametric nature of the regression methods used to fit the production functions.[34] In our learning-to-testing scenarios, we compare the three methods' performance on 100 learning-testing set pairs, V=100, using learning datasets of size n_L= 30, 50, 80, 100, 150, 200, 240 and 300. For our *full-sample* scenarios, we consider n_{Learn}= 100, 200, 300. For all scenarios, we consider 30 randomly drawn sets of noise testing vectors, W= 30 to compute the in-sample portion of (IV.8). Additionally, we estimate the correctly-specified parametric estimator for our DGP. The true parametric form is never known in an

[34] Specifically, if we intend to use the learning set's residual sum of squares, calculation of an estimator $\hat{\sigma}^2$ would require knowledge of the functional estimator's effective number of parameters. However, effective parameters can be difficult to calculate for nonparametric and particularly sieve estimators.

application and thus the correctly-specified parametric estimator could not be selected as an estimator for a practitioner; however, these estimation results are best-case benchmarks.

In what follows we present or our estimates of expected *Full Set* errors measured against the true DGP, $\overline{\text{MSE}}^{\text{nL}}_{\text{FSf}}$, and expected fraction of unexplained variance on the *Full Set*, $\overline{\text{MSE}}^{\text{nL}}_{\text{FSy}}/\text{var}(Y_{FS})$, respectively. Also, note that the expected Full Set error is equal to expected In-Sample error for the full-sample scenarios. Due to the extensive nature of our results, we present them in graphical form. Tables for all experiments can be found in Appendix H. We record and report other relevant performance indicators, such as the number of hyperplanes fitted and the estimation time.

IV.2.1 Bivariate Input Cobb-Douglas DGP

We consider the DGP $Y_i=X_{i1}^{0.4}X_{i2}^{0.5}+\varepsilon_i$, where $\varepsilon_i \sim N(0,\sigma^2)$ and σ= 0.01, 0.05, 0.1, 0.2, 0.3, 0.4 for our six noise settings, which we split into low and high-noise settings. Furthermore, we assume $X_{ij}\sim\text{Unif}(0.1,1)$ for j=1,2 and i=1,…,n_L. Our first observation from Figures 9 and 10 is that our estimated expected full set error results, on all learning-to-testing set scenarios for CNLS exceed the scale of the y-axes (due to very high predictive error values); thus we only present the full set error values for CNLS. The top set of graphs in Figure 9 demonstrates that CAP-NLS has similar to slightly better expected Full Set error values performance than both CAP and CNLS on full set scenarios, while CAP-NLS clearly outperforms both methods on learning-to-testing set scenarios. However, we gain further insight by looking at the bottom panels of Figure 9, we see that for low-noise scenarios, σ =0.01, 0.05, 0.1, where the percentage of output variance due to noise is 0.3%,

6.5% and 22% respectively. We observe that the improvement of CAP-NLS against CAP rarely exceeds 2% of the variance of the full dataset. In other words, an r-squared measurement would differ by less than two percent. We observe that the difference between the correctly-specified parametric estimator and CAP-NLS is also within 2% on nearly all cases.

Surprisingly, Figure 10 shows that for our large noise settings, σ =0.2, 0.3, 0.4, CAP-NLS and the other nonparametric estimators are competitive with the correctly-specified parametric estimator for large sample size or large noise scenarios. This performance gap reduction against the correctly specified parametric estimator is at least in part due to the generally nonlinear objective function of the Least-Squares estimator of Cobb-Douglas with an additive error term (the correctly specified error structure for our DGP), which can potentially lead to multiple local optimal solutions. We note that if we had instead considered a multiplicative error structure on the DGP, for which neither of the nonparametric estimators has a convex programming formulation, the Cobb-Douglas function would have been easy to estimate. Further, only CAP-NLS and the parametric estimator perform consistently regardless of the full set size on these higher noise settings and the performance gap between CAP-NLS and CAP increases to more significant levels, as the latter becomes unstable. From the lower panels of Figures 9 and 10, we see that all the estimators approach the true noise-to-total variance level (labeled *True Variance Level* on the graphs for notational ease) as the learning set size increases, regardless of the noise setting. Finally and as expected, we observe very high correlation between expected Full Set error measured against the true DGP and true noise-to-total variance level.

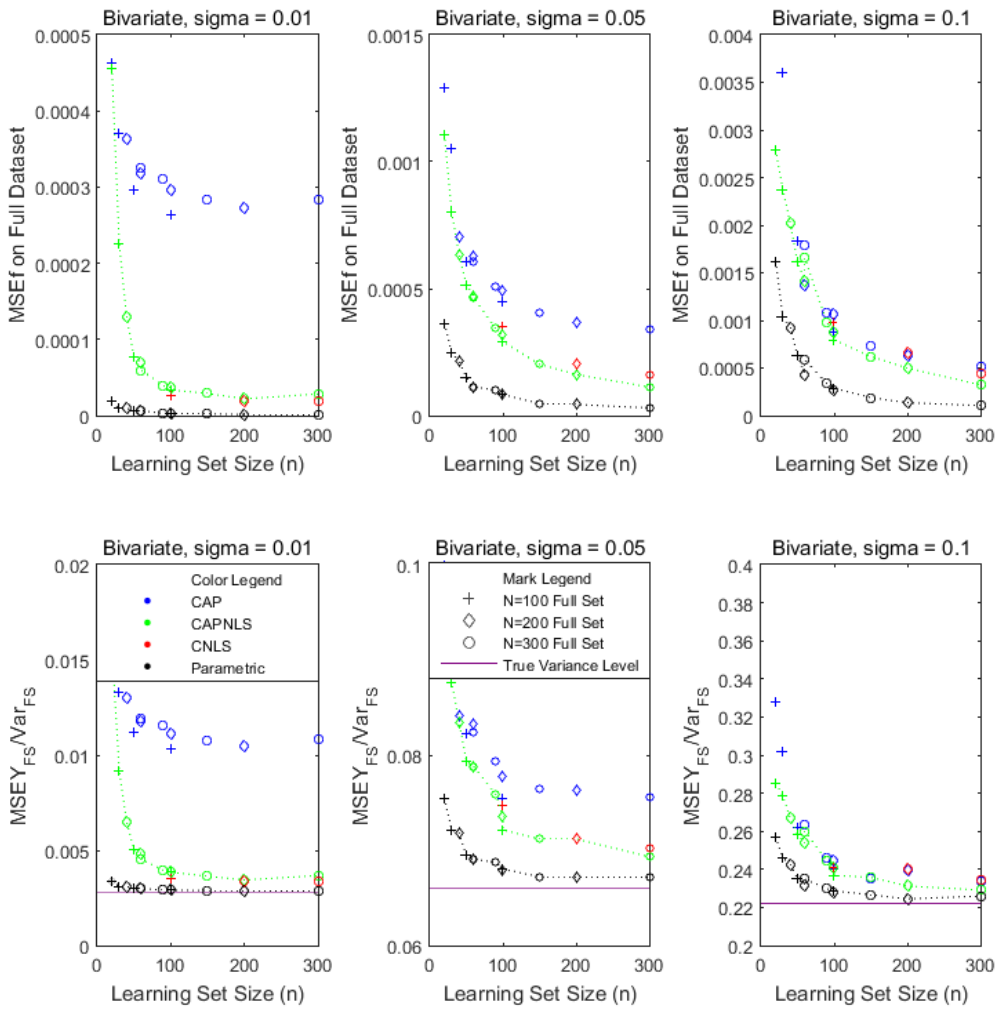

Figure 9 Bivariate input Cobb-Douglas DGP results for small noise settings

Table 21 Number of Hyperplanes and Runtime for Bivariate Input Cobb-Douglas DGP

		CAP-NLS			CAP			CNLS		
σ	*n*	100	200	300	100	200	300	100	200	300
0.01	*K*	9	12	12	2	2	2	93	164	242
	Time (s)	4	15	27	1	0.56	0.78	1	8	22

Table 21 (continued)

		CAP-NLS			CAP			CNLS		
0.05	*K*	8	10	12	2	2	2	80	135	198
	Time (s)	5	12	30	0.42	0.60	0.77	1	6	23
0.1	*K*	9	10	11	2	2	2	60	148	172
	Time (s)	4	17	40	0.42	0.54	0.75	1	8	26
0.2	*K*	9	10	11	2	2	2	54	101	157
	Time (s)	5	16	32	0.47	0.66	0.79	1	8	23
0.3	*K*	8	10	11	2	2	2	50	98	147
	Time (s)	5	15	28	0.41	0.58	0.71	1	8	22
0.4	*K*	8	10	11	2	2	2	47	90	135
	Time (s)	4	15	29	0.46	0.62	0.74	1	7	22

The number of hyperplanes fitted for the Full Sample scenarios are reported in Table 21 for each of the three nonparametric estimators. Larger values indicate a more complex production function using more hyperplanes to characterize the curvature. While CNLS fits a much larger number of hyperplanes relative to CAP-NLS, CAP fits functions that are only slightly more complex than linear, by employing 2 hyperplanes in all estimates. Finally, although CAP-NLS's runtimes are the highest, they are still small in absolute terms.

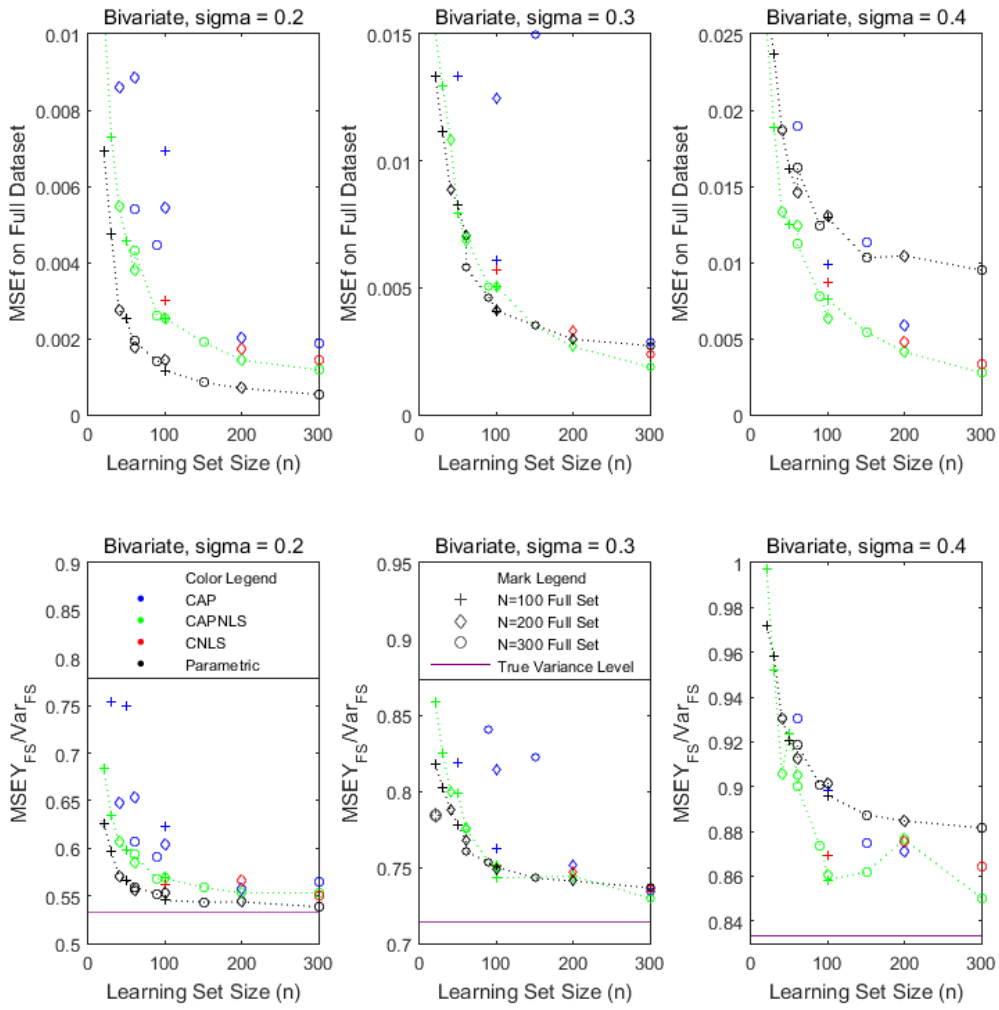

Figure 10 Bivariate input Cobb-Douglas DGP results for large noise settings

IV.2.2 Trivariate Input Cobb-Douglas DGP

We consider the DGP $Y_i=X_{i1}^{0.4}X_{i2}^{0.3}X_{i3}^{0.2}+\varepsilon_i$, where $\varepsilon_i \sim N(0,\sigma^2)$ $\sigma = 0.01, 0.05, 0.1, 0.2, 0.3, 0.4$ for our six noise settings considering the same small and large noise split as the previous example and $X_{ij} \sim Unif(0.1,1)$ for $j=1,2,3$, $i=1,\ldots,n_L$. We first note that CNLS's expected Full Set errors again exceed the displayed range for the learning-to-full set scenarios for this trivariate input example regardless of the noise level. Again this

is a consequence of poor predictive error values, which is in part linked to the higher proportion of non-fully dimensional hyperplanes[35] CNLS fits. The second key observation is that the higher dimensional form of the parametric estimator, as compared to Example IV.2.1, results in increased estimation complexity. This results in higher errors for the parametric estimator, which exceed the very small scale of most panels in Figure 11.

However, as the longer error scales of Figure 12 show, the errors given by the parametric estimator are not as high as those for CNLS in learning-to-full settings. Furthermore, we observe that CAP's expected performance deteriorates relative to Example IV.2.1 and the performance gain obtained by employing CAP-NLS is relevant in an increased number of settings. Like in Example IV.2.1, as the learning set grows, the expected Full Set errors gap between CAP-NLS and the correctly specified parametric estimator either favors CAP-NLS at every learning set size or becomes more favorable for CAP-NLS as the learning set size increases. Finally, we note from Figure 12 that CAP-NLS can accurately recover a production function even when noise composes nearly 85% of the variance, as in the σ=0.4 results.

[35] These are hyperplanes which have zero coefficients on some input dimensions, implying it is possible to obtain output without using the zero-coefficient inputs. Olesen and Petersen (1996) first explore the existence of these type of hyperplanes on a DEA context.

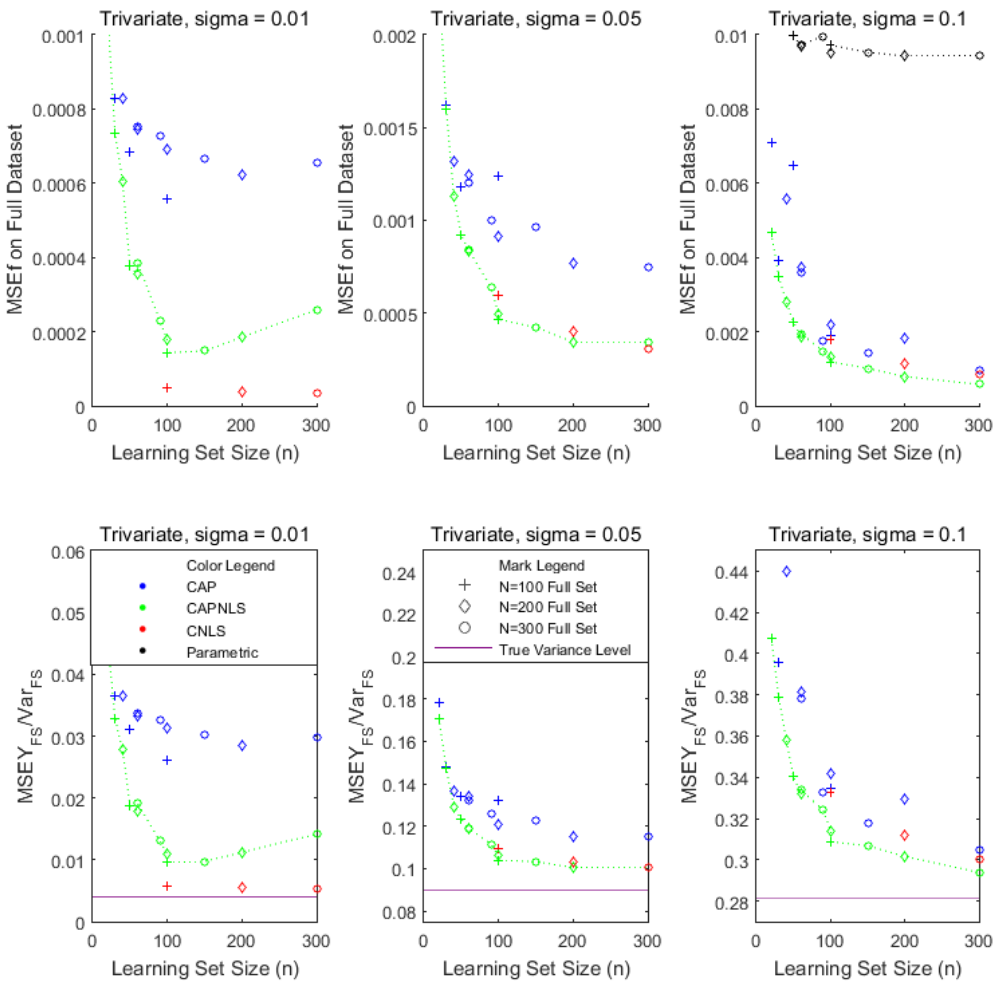

Figure 11 Trivariate input Cobb-Douglas DGP results for small noise settings

In Table 22, we observe that all methods fit slightly more hyperplanes than for the Bivariate-input example. Increase in the number of hyperplanes with increased dimensionality is moderate for both CAP-NLS and CAP at all settings. For CNLS, while the number of hyperplanes does not significantly increase for n=100, it significantly increases for the two larger datasets. Furthermore, the runtimes for all methods are higher than in the previous example, with CAP-NLS's times nearly doubling; however, CAP-NLS's run times are below one minute for all scenarios.

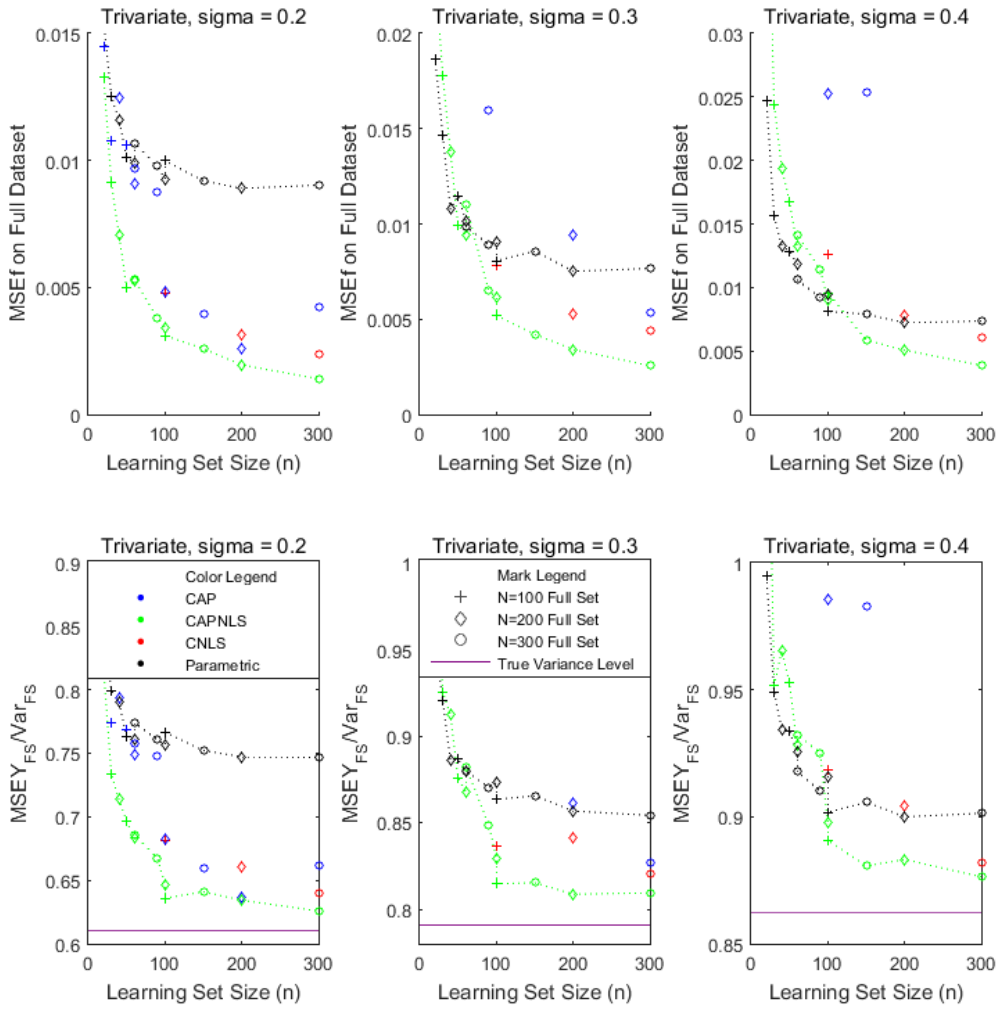

Figure 12 Trivariate input Cobb-Douglas DGP results for large noise settings

Table 22 Number of Hyperplanes and Runtime for Trivariate Input Cobb-Douglas DGP

		CAP-NLS			CAP			CNLS		
σ	*n*	100	200	300	100	200	300	100	200	300
0.01	*K*	9	12	13	2	2	2	96	193	294
	Time (s)	5	28	51	0.59	0.78	1	1	9	27

Table 22 (continued)

		CAP-NLS			CAP			CNLS		
σ	*n*	100	200	300	100	200	300	100	200	300
0.05	*K*	10	11	12	2	2	3	80	169	235
	Time (s)	5	28	45	0.53	0.83	1	1	9	28
0.1	*K*	8	12	14	2	2	3	75	136	199
	Time (s)	6	24	54	0.54	1	1	1	10	24
0.2	*K*	8	11	12	2	3	2	61	126	193
	Time (s)	8	23	50	0.52	0.93	1	1	9	28
0.3	*K*	8	11	12	2	3	3	57	123	184
	Time (s)	5	23	49	0.53	0.92	1	1	9	28
0.4	*K*	8	11	12	2	3	3	54	115	179
	Time (s)	5	25	49	0.52	0.92	1	1	9	29

IV.2.3 Four-variate Input Cobb-Douglas DGP

We consider the DGP $Y_i=X_{i1}^{0.3}X_{i2}^{0.25}X_{i3}^{0.25}X_{i4}^{0.1}+\varepsilon_i$, where $\varepsilon_i\sim N(0,\sigma^2)$ and $\sigma=0.01, 0.05, 0.1, 0.2, 0.3, 0.4$ for our six noise settings and $X_{ij}\sim Unif(0.1,1)$ for $j=1,2,3,4$; $i=1,\ldots,n_L$. We first observe that Table 23 shows that the number of hyperplanes needed to fit this four-variate input production function does not significantly increase from the trivariate-input case of Example IV.2.2 for any of the methods. Contrastingly, CAP-NLS has 40-60% longer runtimes as compared to the trivariate-input case. However, this runtime increase with dimensionality is not a severe concern as input information to fit a production function (or

output in case of a cost function) rarely exceeds four variables. Furthermore, the maximum recorded runtime for CAP-NLS was still short and thus is not large in absolute terms.

Figures 13 and 14 show that for this higher dimensional example, the parametric estimator can only predict the true function up to a certain accuracy, namely $\overline{\text{MSE}}^{\text{nL}}_{\text{FSf}}$=0.015, and tends to plateau at this error level even as learning set size increases. Moreover, the benefits of CAP-NLS over the other nonparametric methods are similar the results in Example IV.2.2 for the small noise settings, and significantly larger for the large noise settings. Finally, the gap between CAP-NLS and all the other functional estimators, parametric or nonparametric, increases indicating CAP-NLS's superior performance for all noise settings and learning set sizes in this higher-dimensional example.

Table 23 Number of Hyperplanes and Runtime for Four-variate Input Cobb-Douglas DGP

		CAP-NLS			CAP			CNLS		
σ	*n*	100	200	300	100	200	300	100	200	300
0.01	*K*	7	11	12	2	2	2	98	194	234
	Time (s)	5	29	70	0.42	1	2	1	8	24
0.05	*K*	7	12	13	2	2	2	87	170	215
	Time (s)	4	33	65	0.39	1	1	1	9	27
0.1	*K*	7	12	12	2	2	2	55	166	207
	Time (s)	4	30	62	0.60	1	2	1	10	32
0.2	*K*	7	12	13	2	2	2	63	132	192
	Time (s)	4	36	79	0.45	1	2	1	9	31
0.3	*K*	7	12	12	2	2	2	59	122	192
	Time (s)	4	36	74	0.46	1	2	1	10	32
0.4	*K*	7	12	12	2	2	2	57	122	186
	Time (s)	4	36	75	0.48	1	2	1	10	33

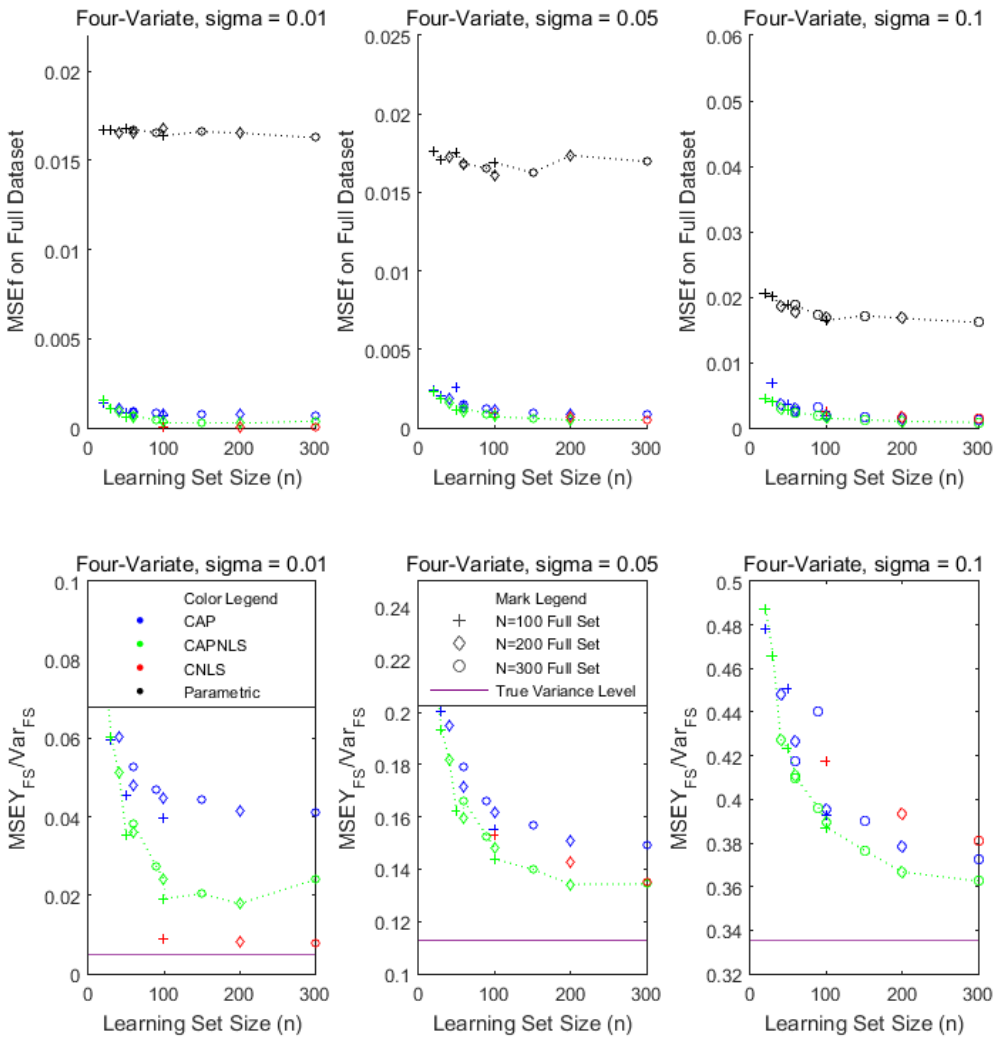

Figure 13 Four-variate input Cobb-Douglas DGP results for small noise settings

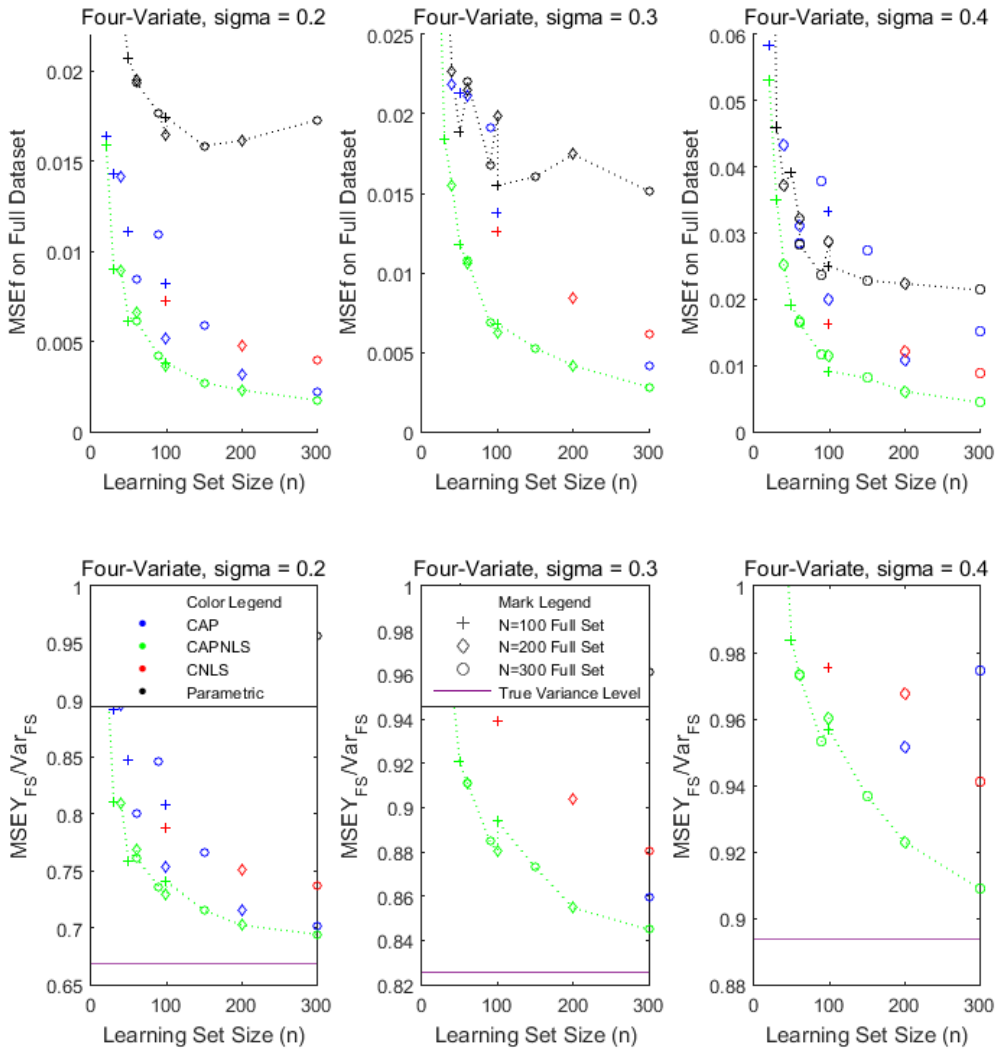

Figure 14 Four-variate input Cobb-Douglas DGP results for large noise settings

IV.2.4 Insights and Implications from Examples IV.2.1 - IV.2.3

Throughout our examples, we observed that CAP-NLS was the only functional estimator which performed robustly on a learning-to-full set basis across all dimensionalities and noise levels, while also being the nonparametric estimator with lowest in-sample error on nearly all the full set scenarios. CNLS's overfitting of the learning set (as observed by the large number of hyperplanes it fitted in all examples) has a severe detrimental effect on the

expected Full Set error due to low predictive power (i.e., high expected predictive error). However, this overfitting issue did not heavily affect its in-sample performance, as observed through CNLS's expected Full Set error in the full set scenarios and thus could still be robust and a candidate estimator for comparison on census datasets. CAP showed good performance on both full set and learning-to-full set scenarios for small noise settings at all dimensionalities, however, as the noise level increased, CAP's learning-to-full set performance deteriorates.

Expected Full Set error is similar for all nonparametric methods, except for CAP in the high noise settings with 3 or 4 inputs, on which its performance deteriorates. CAP-NLS and CNLS perform similarly on the full set scenarios in all cases. Runtimes for CAP-NLS are the only ones to deteriorate significantly with dimensionality and are the largest of the three nonparametric methods in all cases. However, these times are still small in relative terms, not being larger than 2 minutes for any fitted dataset. Finally, while dimensionality of production functions is typically low and therefore CAP-NLS's scalability in dimensionality would not be a concern, scalability in *n* could be an issue to fit large production datasets.[36] The following subsection proposes a modification to CAP-NLS to address this potential issue.

IV.2.5 Scalability of CAP-NLS to Larger Datasets

To demonstrate the performance of CAP-NLS in large data sets, we revisit the DGP used Example IV.2.2, specifically, $Y_i=X_{i1}^{0.4}X_{i2}^{0.3}X_{i3}^{0.2}+\varepsilon_i$, where $\varepsilon_i \sim N(0,\sigma^2)$, $\sigma=0.1$ and

[36] Oh et al. (2015) or Crispim Sarmento et al. (2015) discuss computational challenges for fitting existing nonparametric piecewise linear estimators in large application datasets.

X_{ij}~Unif(0.1,1) for j=1,2,3, i=1,…,n and n=500, 1000, 2000, 3000, and 5000. The estimator's performance is reported in Table 24.[37] We first conduct standard CAP-NLS analysis and report learning errors, number of fitted hyperplanes and runtime results. Runtimes for datasets up to 2,000 observations are well below the one hour threshold; however, we see significant scalability challenges for datasets larger than 2000 observations. Thus, we apply the Fast CAP stopping criterion in Hannah and Dunson (2013), which measures the GCV score improvement by the addition of one more hyperplane and stops the algorithm if no improvement has been achieved in two consecutive additions, although unlike Fast CAP, we apply it directly to the learning error against observations. We denote the results for those runs with the CAP-NLSF superscript and observe that differences are minimal if compared to following our standard partitioning strategy. However, this alternative stopping rule results in a highly scalable algorithm which is able to fit datasets up to 5,000 observations in around 40 minutes.

[37] Due to the increased computational burden of using larger datasets, we present results for a single replicate of the DGP for each sample size and only include learning set results. For this section we choose to report $RMSE$ results rather than MSE ones, as the latter were very small and differences are indistinguishable across settings.

Table 24 Number of Hyperplanes and Runtime for Trivariate Input Cobb-Douglas DGP on Larger Datasets

n	500	1000	2000	3000	5000
$RMSEf_{Learn}^{CAP\text{-}NLS}$	0.025	0.022	0.025	0.025	0.024
$RMSEf_{Learn}^{CAP\text{-}NLSF}$	0.023	0.025	0.026	0.027	0.028
$MSEy_{Learn}^{CAP\text{-}NLS}$	0.009	0.010	0.011	0.011	0.011
$MSEy_{Learn}^{CAP\text{-}NLSF}$	0.010	0.011	0.010	0.011	0.011
$K^{CAP\text{-}NLS}$	5	5	7	5	5
$K^{CAP\text{-}NLSF}$	4	4	5	4	4
$Time(min)^{CAP\text{-}NLS}$	3	8	43	114	367
$Time(min)^{CAP\text{-}NLSF}$	3	5	10	11	41

IV.3 Chilean Annual National Industrial Survey

IV.3.1 Dataset and Considerations

The Chilean Annual Industrial Survey (ENIA, by its initials in Spanish) is a census of all industrial establishments with 10 or more employees which are located inside the Chilean territory. The survey's main goal is to characterize the manufacturing activity of the country in terms of input usage, manufactured products and means of production utilized in the diverse transformation processes. We focus on the 5 largest 4-digit industries in terms of sample size and removing only a very limited set of establishments. Specifically we only remove observations with non-positive value added or input values for any of the used input variables. Further, in this section we will refer to the learning sets as the survey subsamples and to the full sets as the survey full sample.

We use this dataset to illustrate three points which have been largely overlooked in the production function estimation literature and that are key in working with national survey data on manufacturing. First, as real production data is highly clustered around a particular

scale size and input ratios, the data lacks the more complex curvature of data simulated from monotonic and concave DGPs. In view of this difference, the performance of estimators and their resulting rankings could vary significantly between Monte Carlo simulation experiments and the estimators' performance on survey data. We investigate whether this is the case, by assessing the ability of the estimators used in Section IV.2 to fit industry-specific data from the Chilean manufacturing dataset on a subsample to full sample setting. Second, we illustrate the replicate-specific performance of the selected functional estimators. Third and lastly, we graphically explore the increase in explanatory capability of our fitted production functions as a function of the relative size of survey subsample to survey full sample and discuss practical survey sample size implications.

IV.3.2 Methodology to Compare Functional Estimator Performance on Real Data

We compare additive error formulations of CAP-NLS, CAP and CNLS. Further, we also consider the additive-error Cobb-Douglas formulation used in Example IV.2.3, $Y = X_1^{\alpha_1} X_2^{\alpha_2} X_3^{\alpha_3} X_4^4 + \varepsilon$, which we label CDA. As customary, we restrict all input powers to be nonnegative for the Cobb-Douglas functional estimator. In total, we compare 4 different functional estimators. We center our comparison on the estimated expected error on the full survey set of establishments given a survey subset size $E(\widehat{Err}_{FSy}^{nL})$, but report the scale-invariant quantity $R_{FS}^2 = \max(1 - E(\widehat{Err}_{FSy}) / Var(y_{FS}), 0)$, where $Var(y_{FS})$ is the sample variance of the output on the full industry dataset. Note that the definition of R_{FS}^2 implies that if the evaluated estimator fails to explain more variability than the mean value added on the survey full sample, we will use the mean value added as our estimator instead.

To compute $\mathrm{E}\left(\widehat{\mathrm{Err}_{\mathrm{FSy}}^{\mathrm{nL}}}\right)$ we rely, as in our simulated data section, in separate estimations of the expected predictive $\mathrm{E}\left(\mathrm{Err}_{\mathrm{y}}^{\mathrm{nL}}\right)$ and in-sample $\mathrm{E}\left(\mathrm{Err}_{\mathrm{ISy}}^{\mathrm{nL}}\right)$ errors, which we later weight by the relative size of the observed and unobserved establishment sets. However, unlike in Section IV.2, we cannot generate more data from the same DGP as that of the observed dataset, let alone vectors of residuals with the same level of noise as the DGP and thus cannot compute error estimators $\overline{\mathrm{MSE}_{\mathrm{y}}^{\mathrm{nL}}}$ and $\overline{\mathrm{MSE}_{\mathrm{ISy}}^{\mathrm{nL}}}$. To circumvent these issues, we estimate $\mathrm{E}\left(\mathrm{Err}_{\mathrm{y}}^{\mathrm{nL}}\right)$ via a RLT procedure and we estimate $\mathrm{E}\left(\mathrm{Err}_{\mathrm{ISy}}^{\mathrm{nL}}\right)$ by summing the *learning error* $\mathrm{MSE}_{\mathrm{yL}}^{\mathrm{nL}}$ for a $\mathrm{n_L}$-sized learning set and a parametric bootstrap covariance penalty estimator $\widehat{\mathrm{E}(\omega^{\mathrm{nL}})}$ for expected *in-sample optimism* $\mathrm{E}(\omega^{\mathrm{nL}})$ (Efron, 2004). For the RLT procedure, we consider 20%, 30%, 40% and 50% learning subsets and V=100 replicates to understand the predictive power of subsample-fitted functional estimators when inferring the industry-level production function as the subsample size increases.[38] For the bootstrap procedure we consider B=500 parametric bootstrap replicates.

We compute our expected predictive error estimate given by RLT, $\mathrm{MSE}_{\mathrm{RLT}}^{\mathrm{nL}}=\sum_{\alpha=1}^{\mathrm{V}}\frac{\mathrm{n_L^{\alpha}}}{\mathrm{n}}\sum_{\mathrm{i}\notin\{\alpha\}}\left(\hat{\mathrm{f}}_{\mathrm{i}}^{\alpha}-\mathrm{Y_i}\right)^2/\mathrm{n_T^{\alpha}}$, where $\{\alpha\}$ is the index set of the α^{th} learning set, $\hat{\mathrm{f}}_{\mathrm{i}}^{\alpha}$ are the estimated functional values obtained from the α^{th} learning set, and $\mathrm{n_T^{\alpha}}=\mathrm{n}-\mathrm{n_L^{\alpha}}$, where $\mathrm{n_L^{\alpha}}$ is the size of the α^{th}learning set. As we only aim to estimate expected predictive error for a set of the size of our *learning set*, our RLT estimator does not have the bias described by Burman (1989) when estimating the usual cross-validation objective, which is the expected

[38] We reemphasize that unlike the usual goal of cross-validation procedures, of which the goal is to estimate $E\left(Err_y^n\right)$, our goal is to estimate $E\left(Err_y^{nL}\right)$

predictive error for a set of the size of our *full set*. Furthermore, Burman (1989) shows that the variance of RLT can be partially controlled with the number of replicates V. Finally, we acknowledge that independently of V, the variance of our RLT expected predictive error estimate could increase with the learning set size, as the testing set size decreases, given our finite full survey. However, we do not observe an increase in variance in our estimates, as will be shown on Section IV.3.5.

As mentioned before, to compute the estimator for expected[39] in-sample error $E\left(\widehat{Err}_{ISy}^{nL}\right)$, we add the learning error MSE_{yL}^{nL} and a covariance penalty term $E(\widehat{\omega}^{nL})$ to account for expected optimism $E(\omega^{nL})$. If a uniformly weighted squared loss function (i.e., $g\left(\widehat{Y_i},Y_i\right)=\widehat{Y_i}-Y_i$ in our notation) and an arbitrary estimator $\widehat{Y_i}$ are considered, Efron (2004) shows that

$$E(\widehat{\omega}^{nL})=\frac{2}{n_L}\sum_{i=1}^{n_L} cov(\widehat{Y_i},Y_i). \tag{IV.9}$$

We note that if the functional estimators being considered are all of the linear smoother form $\widehat{\mathbf{Y}}=S\mathbf{Y}$, this penalty term can be written in terms of the trace of the linear smoothing matrix S. However, not all the functional estimators we evaluate are of this form, as the Cobb-Douglas functional estimator is clearly not linear in parameters. Thus, and to accommodate a much larger class of functional estimators in our model evaluation framework, we use the parametric bootstrap algorithm by Efron (2004), which directly estimates $cov(\widehat{Y_i},Y_i)$. We leave the details about this algorithm for Appendix I. Thus, our full expression for $E\left(\widehat{Err}_{FSy}^{nL}\right)$ for learning set sizes of size nL is

[39] Again, expectations and averages over the error and optimism metrics discussed are done over all possible learning sets of a given size.

$$E\left(\widehat{Err}_{FSy}^{nL}\right)= (n_T/n)_{MSE_{RLT}^{nL}}+ (n_L/n)(MSE_{yL}^{nL}+ E(\hat{\omega}^{nL})) \quad \text{(IV.10)}$$

We clarify that by using a uniformly-weighted error measure, such as our $E(\widehat{Err}_{FSy}^{nL})$ estimator to evaluate the different functional estimators, we assume the data is truly homoscedastic[40]. Thus, if we intended to use multiplicative or other residual assumptions, our error estimators would need to reflect a similar residual-weighting scheme.

To define the inputs and output for our production function, we follow the KLEMS framework by fitting the Value-Added production function

$$VA=Y-M=f(KLES) \quad \text{(IV.11)}$$

where *VA* is value added, *Y* is output, *M* is intermediate goods, *K* is capital stock, *L* are labor man-hours, *E* is energy and *S* is service expenditures, respectively. All variables except for *L* are measured in thousands of Chilean pesos. These variables are readily found in the Chilean manufacturing dataset, except for Energy, for which we also add the fuel expenditures costs. From the high explanatory value of our models for all industries, we argue that use of K, L, E and S is sufficient to describe variability in VA.

IV.3.3 Functional Estimator Comparison Results

The *Best Method* field of Table 25 names the functional estimator with highest R_{FS}^2 for each subset size. Ties are defined as R_{FS}^2 values within 2% of the best one in which case multiple functional estimators listed. Further, the table includes a field for K_{nL}^{CAPNLS}, the average number of CAP-NLS hyperplanes fitted to either the learning sets in case of 20, 30,

[40] Clearly a different assumption could be made within this same framework by introducing a weighting function on the individual residual terms, however we focus on the uniformly weighted error.

40 and 50 percent subset sizes, or the bootstrapped sets used to compute $E(\widehat{\omega}^{nL})$ in the case of the full set. The average number of CAP-NLS hyperplanes fitted allows us to compare the complexity of the estimated production functions relative to those considered in Section IV.2. As expected due to the simpler curvature and more concentrated nature of real manufacturing survey data relative to Monte Carlo simulated data, the number of CAP-NLS hyperplanes fitted for data sets with 100 or 200 observations is generally smaller than those fitted to similar sample sizes in Example IV.2.3, in which the production function also has a four-dimensional input space.

Further exploring our results for Chilean manufacturing data, we observe both similarities and discrepancies regarding the insights obtained testing estimators via Monte Carlo simulations. The clearest similarity to all our low noise settings[41], is that we observe multiple ties across functional estimators in terms of R^2_{FS}, which mean several estimators are able to describe the production function with the same accuracy. Thus, the model selection results are consistent with the small noise setting results for all of our small noise simulated data examples. Discrepancies include better CDA performance for larger datasets, regardless of the residual noise level, which are closer to the insights obtained with lower-dimensional example IV.2.1. Surprisingly, CDA's performance is remarkably well especially if we consider that now the true DGP is unknown.[42] Thus, in Table 26 we explore the capabilities

[41] The maximum attainable (i.e., using the full set as the learning set) noise-to-total variance levels of our real datasets are very similar to those of our low noise settings. To see this, compare 1-$\overline{MSE^{nL}_{FSy}}/var(Y_{FS})$ on our low noise settings against the R^2_{FS}results the 100% survey real datasets.

[42] Recall we use the Cobb-Douglas function with an additive error term. We stress that the necessity for a relationship between the Cobb-Douglas production function and the use of a multiplicative residual is primarily a historical carry-over. The Cobb-Douglas functional form was introduced in a time where by taking logs and estimating a linear in parameters estimator was the only computationally feasible way to estimate such a function.

of the CDA parametric estimator against the best estimate achieved for each subset size. We observe that in general the CDA estimator describes nearly as much variance as the best estimator. Further, we include equivalent results to those of Table 26, but rather considering the classical multiplicative error assumption for Cobb-Douglas (labeled CDM) in Appendix J. The results in Appendix J show that a multiplicative error assumption when fitting the Cobb-Douglas model is a significantly better model for industry codes 2899 and 2010 (even if we compute its in-sample and predictive errors with the additive error assumption we have followed in our exposition) and a significantly worse model for industry code 1541.

Table 27 indicates the best estimator in the Chilean manufacturing dataset is perhaps more closely related to the learning set size, regardless of the residual noise level. CAP-NLS is dominant for very small learning set sizes (less than 50 observations). Contrastingly, CAP-NLS, CAP and CDA perform similarly for larger datasets. Thus, we observe that the additional structure of CAP-NLS relative to CAP seems to lose its benefits as the learning set size increases for our application datasets, which is a much more direct statement than we could make from extrapolating results across our 3 small noise settings of Example IV.2.3. Some insights obtained evaluating estimators on the actual application dataset were not observed on simulated data. For instance, our simulated data examples showed potential problems when fitting the CDA model at high dimensionalities or high noise settings. However, we see that for the application datasets considered, CDA is a reliable production function estimator at learning sets of all considered sizes.

Table 25 Method Comparison Across the 5 Largest Sampled Industries from the Chilean Annual National Industrial Survey

Industry Name and Code	n	Survey Size	R^2_{FS}	K_{nL}^{CAPNLS}	Best Method
Other Metal Products (2899)	144	20%	50%	1	CAP-NLS, CDA
		30%	60%	2	CAP-NLS, CDA
		40%	64%	2	CAP-NLS, CDA
		50%	72%	3	CAP-NLS
		100%	88%	7	CAP-NLS
Wood (2010)	150	20%	35%	1	CDA
		30%	40%	1	CAP-NLS, CDA
		40%	47%	2	CAP-NLS, CDA
		50%	52%	3	CAP-NLS, CDA
		100%	66%	6	CAP-NLS
Structural Use Metal (2811)	161	20%	77%	1	CAP-NLS, CAP
		30%	82%	2	CAP-NLS
		40%	87%	3	CAP-NLS, CAP
		50%	90%	4	CAP-NLS
		100%	95%	9	CAP-NLS, CAP
Plastics (2520)	249	20%	54%	2	CAP-NLS, CAP, CDA
		30%	57%	3	CDA
		40%	57%	5	CAP-NLS, CAP, CDA
		50%	60%	7	CAP-NLS, CAP, CDA
		100%	64%	11	CAP-NLS, CAP, CDA
Bakeries (1541)	250	20%	72%	3	CAP
		30%	77%	3	CAP
		40%	78%	4	CAP, CDA
		50%	85%	4	CAP
		100%	99%	5	CAP-NLS, CAP, CDA

Table 26 Ratio of CDA to Best Model Performance

Industry Name and Code	n	Survey Size	R^2_{FS}	R^2_{CDA}	Ratio vs. Best Method
Other Metal Products (2899)	144	20%	50%	49%	CDA ties for Best Method
		30%	60%	59%	CDA ties for Best Method
		40%	64%	64%	CDA ties for Best Method
		50%	72%	60%	0.83 vs. CAP-NLS
		100%	88%	79%	0.90 vs. CAP-NLS
Wood (2010)	150	20%	35%	35%	CDA ties for Best Method
		30%	40%	40%	CDA ties for Best Method
		40%	47%	47%	CDA ties for Best Method
		50%	52%	51%	CDA ties for Best Method
		100%	66%	62%	0.94 vs. CAP-NLS
Structural Use Metal (2811)	161	20%	77%	69%	0.90 vs. CAP-NLS
		30%	82%	76%	0.93 vs. CAP-NLS
		40%	87%	81%	0.93 vs. CAP-NLS
		50%	90%	87%	0.97 vs. CAP-NLS
		100%	95%	91%	0.96 vs. CAP-NLS
Plastics (2520)	249	20%	54%	53%	CDA ties for Best Method
		30%	57%	57%	CDA ties for Best Method
		40%	57%	57%	CDA ties for Best Method
		50%	60%	60%	CDA ties for Best Method
		100%	64%	64%	CDA ties for Best Method
Bakeries (1541)	250	20%	72%	61%	0.85 vs. CAP
		30%	77%	71%	0.92 vs. CAP
		40%	78%	78%	CDA ties for Best Method
		50%	85%	82%	0.96 vs. CAP
		100%	99%	99%	CDA ties for Best Method

Table 27 Most Frequently Selected Best Method for Different Sample Size Ranges

	Learning Set Size			
Times selected as "Best Method"	29 - 50	51 - 80	81 - 149	150+
CAP-NLS	7	5	3	4
CAP	3	3	4	3
CDA	5	2	2	2

IV.3.4 Replicate-Specific Insights for Predictive Error

In this subsection, we investigate in more detail the robustness of the estimators we evaluated for the Chilean manufacturing dataset. In order to do this, we display a learning subsample-by-learning subsample results of the *predictive component* of R^2_{FS}, namely the component associated with $(n_T/n)_{MSE^{nL}_{RLT}}$. For clarity of exposition and conciseness, we focus on the results for our three nonparametric estimators, CAP-NLS, CAP and CNLS at all non-exhaustive subsample sizes[43]. Figure 15 gives a subset-by-subset analysis of R^2_{FS} for all folds considered in our analysis and across all the studied industries for 5 randomly selected replicates of the total 100 RLT replicates. We observe that while CAP-NLS and CAP's results are similar identical for all industries. Contrastingly, while CNLS performs similarly for many of the depicted subsamples, it exhibits higher variance. Moreover, this variance is large enough that a $R^2_{FS}=0$ result is not rate, which indicates lack of predictive power. Finally, although such scenarios are also observed for CAP and CAP-NLS, they are so with a rarer frequency.

IV.3.5. Estimator Performance Measures as Function of Subsample Size and Surveying Implications

We apply the results from our framework to make recommendations about the minimal size that a randomly-sampled production survey needs to have in order to represent a census. We compute simulation-based confidence intervals on R^2_{FS} across the replicates of

[43] Analyses including parametric estimators and the full sample yield similar insights and thus are of limited added value.

our RLT results. As mentioned before, increased testing set variance as the learning set size increases did not seem to be large enough so as to affect the variance of our estimates across. We concentrate on describing this procedure for the best method for every industry (we consider CAP-NLS for all industries, except Bakeries, for which we consider the CAP results) at all the considered non-full survey learning set sizes. However, we note that this procedure to analyze subset R^2_{FS} performance for any other estimator or combination of estimators would be the same. Figure 16 shows the learning subset-specific results for the Best Method in terms of goodness-of-fit, R^2_{FS}, for different industries. If we compare to the insights on Figure 15, we can observe that the variance of R^2_{FS} and overall predictive power is significantly enhanced by inclusion of the in-sample component of the expected Full Set error.

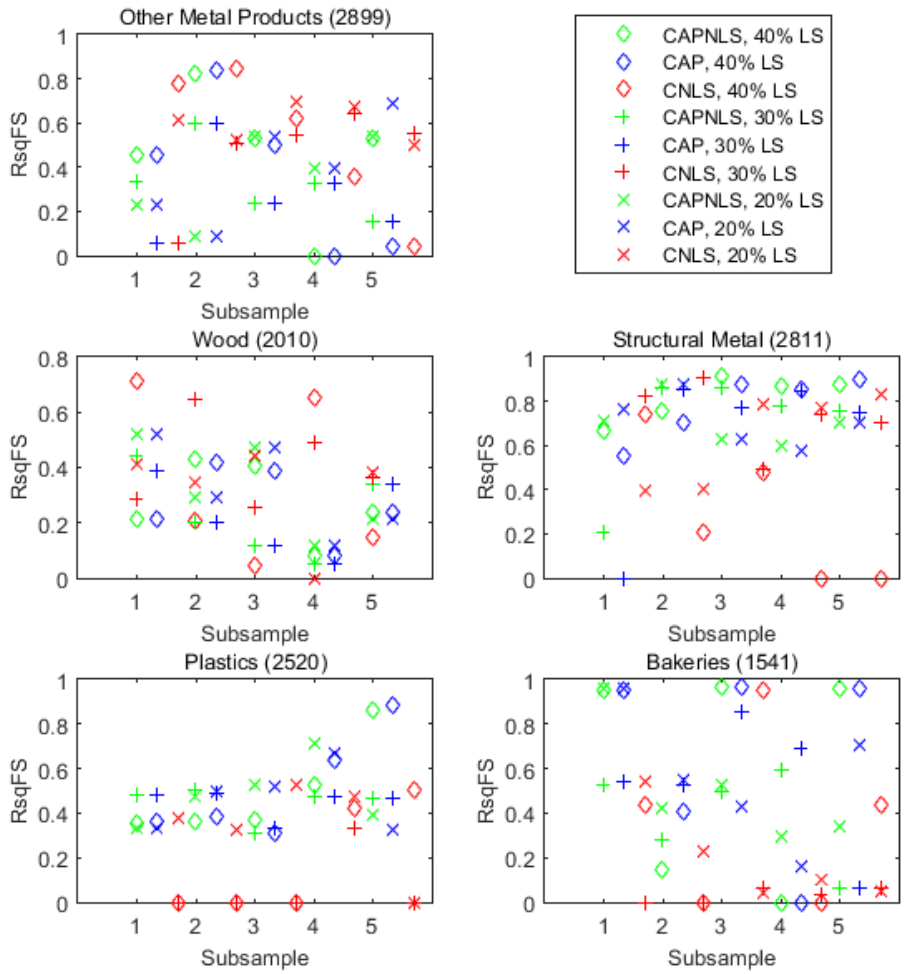

Figure 15 Replicate-specific results on predictive component of R^2_{FS} for varying degrees of CAP-NLS-related estimation improvement.

We note that the mean goodness-of-fit is increasing for all industries with different degrees of diminishing returns to larger survey subsamples. Industry-dependent degrees of variance reduction are also observed. These results are of significant practical importance, as countries such as the United States and Mexico only conduct full economic censuses on a quinquennial basis, while taking surveys on the years in between. Although we acknowledge that the R^2_{FS} results we obtain are specific to the particular observed survey full set, we argue

they can still be of at least qualitative value for survey design on the years following the census. Specifically, to use the data from the Census year to inform the sample size needed in following years, the researcher would need to assume that both the set of establishments within an industry and the complexity of the production function have not changed significantly over this time period. Under these conditions, this analysis could help guide the subset-size calculation strategy for each industry. For example, if production functions with 75% of the predictive power of a census-fitted production function are desired, the relative survey sample size would need to be approximately 40%, 45%, <20%, <20% and 25% for industry codes 2899, 2010, 2811, 2520 and 1541, respectively.

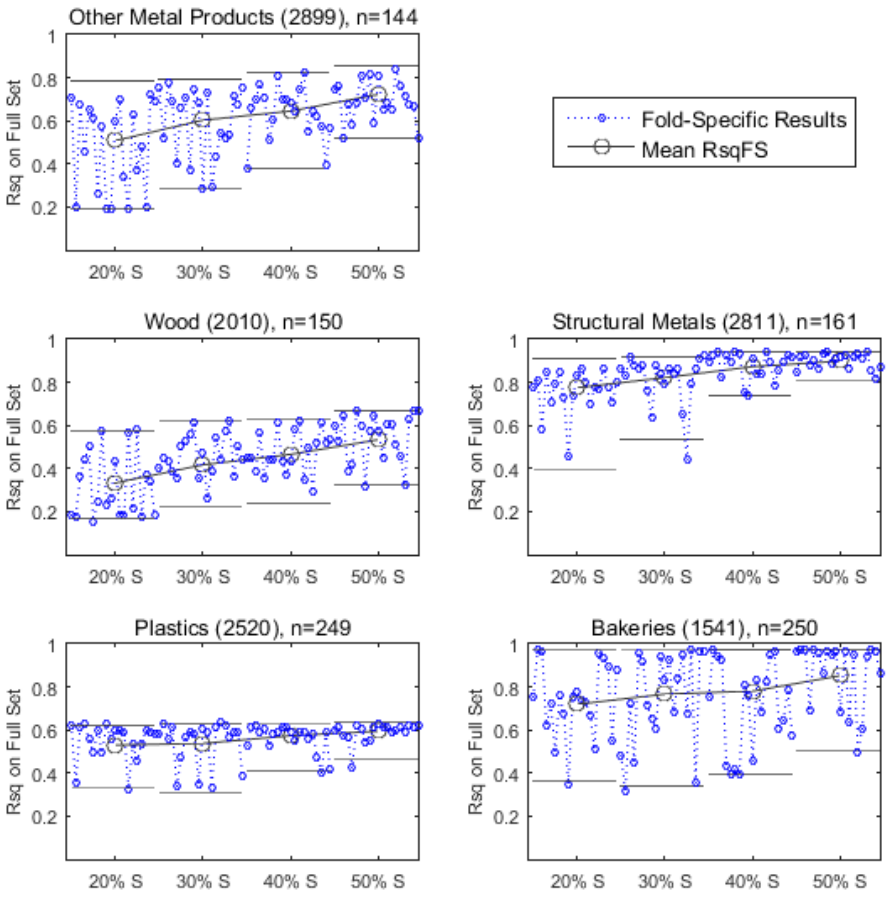

Figure 16 Best Method's R^2_{FS} as function of relative subset size for selected industries. CAP-NLS was chosen as Best Method for industry codes 2899, 2010, 2811 and 2520, while CDA was chosen for industry code 1541.

CHAPTER V*

CONCLUSIONS

We have constructed specific models, as well as a general framework that address model parsimony in the estimation of production functions and frontiers. In the process, we have shifted the state of the art in both one-stage and two-stage production frontier estimators. Concerning one-stage estimators, the specific Bayesian models built for state-contingent and concave-constrained frontiers are an improvement in either flexibility, generality, scalability, or more than one of these attributes as compared to existing methods. Moreover, they both rely on different RJMCMC algorithms which control model complexity. Additionally, both models allow for flexible inefficiency distributions that significantly depart from prior assumptions due to their mixture modeling context. In the realm of two-stage frontier estimation, we have laid out a general framework to evaluate production function estimators based on their weighted in-sample and predictive performance. Moreover, we have proposed an adaptively partitioned functional estimator which performs robustly under this evaluation metric.

V.1 Conclusions Specific to Chapter II

Chapter II describes a new BDMCMC algorithm that efficiently estimates the number of states in a state-contingent model. This approach improves upon the Bayesian estimation of state-contingent models, because the posterior distribution of this parameter can be

---

* Section V.1 reprinted with permission from "A birth-death Markov Chain Monte Carlo Method to Estimate the Number of States in a State-Contingent Production Frontier Model" by Preciado Arreola, J.L. and A. L. Johnson, 2015. *American Journal of Agricultural Economics*, 1267-1285, Copyright by American Agricultural Economics Association.

visualized, thus providing more insights into the nature of the unobserved variables generating the states. Notably, after only one run of the model, there is enough information to weight the outputs obtained for models with different numbers of states in a straightforward manner. Computing a goodness of fit analysis for a state-contingent model of production allows determining whether the full posterior distribution of the number of states enhances the model significantly versus using only its mode, by assessing the difference in MSE of both scenarios. The experimental results derived from a case study of 44 rice farms in the Tarlac region of the Philippines shows an insignificant difference between using the mode of the posterior distribution versus using the complete distribution. Utilizing a state-independent dummy time trend, we estimate the differences in mean output levels across states to be smaller than O'Donnell and Griffiths (2006)'s estimates.

Our finding that a state-contingent linear time trend could not explain the complex time-shifting effect of the frontier suggests that changing weather patterns from year to year have a non-linear effect on output. The similarity in the mean output levels of states in our dummy time trend model suggests that bad/good years affect the rice-producing region more uniformly than indicated by O'Donnell and Griffiths (2006). Nevertheless, evidence of state-contingency can still be argued by the differences in labor elasticities we find for states 1 and 3 as well as the slight differences in the state-contingent mean output. The unimodal posterior distribution on the number of states indicates that the interactions between unobserved variables are complex and probably interdependent. We suggest that the inability to impose a convexity constraint could be due to the limited flexibility of the parametric framework, distributional assumptions, or the presence of outliers in the dataset. A complementary or alternative explanation for the inability to impose convexity also could relate to the significant yearly shifts in the observed log-output, for example, in 1996; there is a distinct possibility that the group of farms as a whole was on a non-convex portion of a

classical s-shaped yield curve, since the mean log-output in 1996 was well below the average of the full timespan. Comparing the results of our two models shows that our efficiency rankings are roughly consistent with O'Donnell and Griffiths (2006), and that the main benefit of the O'Donnell and Griffiths (2006) model, the ability to avoid misinterpretation of state-contingency as inefficiency, is maintained. In terms of input elasticities, our Area and Labor elasticities are similar to O'Donnell and Griffiths (2006), although we estimate a positive fertilizer elasticity for our lowest output state whereas O'Donnell and Griffiths (2006) finds a negative effect of fertilizer. The difference is driven by the higher average yield estimates in state 1 we obtain as compared to O'Donnell and Griffiths (2006). Our positive fertilizer elasticities, given our estimated yield per unit area in that state, are consistent with the previous rice crop literature. Finally, our models give slightly larger yield estimates and exhibit a significantly lower MSE.

Further work remains to be done regarding the estimation of state-contingent frontiers. A key limitation is the challenges regarding imposing a convexity constraint. This is likely an issue related to this data set. Also currently our model needs a prespecified lower bound on technical efficiency to be estimated. While we have performed sensitivity analysis on the related parameter, the approach in Mahendran et al. (2012) could be used to select this parameter optimally in terms of a specific criterion, such as the MSE.

V.2 Conclusions Specific to Chapter III

Chapter III described MBCR-I, a one-stage Bayesian semi non-parametric method to fit concave and monotonic stochastic frontiers using Reversible Jump Markov Chain Monte Carlo techniques. We believe that this chapter discusses the first single-stage method that allows a shaped constrained production frontier to be estimated nonparametrically, relaxes the homoscedastic assumption on the inefficiency term and estimates the impact of

environmental variables for the analysis of cross sectional data. Further, the one-stage nature of MBCR-I revealed the tradeoffs between modeling the frontier and the inefficiency term, i.e., more flexible models of the inefficiency term lead to coarser estimates of the production frontier and vice versa.

Computational modifications relative to MBCR allowed for both the presence of inefficiency and the multiplicative residuals structure of the standard SFA model. MBCR-I's performance in the Monte Carlo simulation study showed increased accuracy in large noise-to-signal ratio scenarios versus StoNED and for large sample sizes at any noise-to-signal ratio. MBCR-I handled datasets of a few thousand observations efficiently, which suggested its use for long panels of moderate samples and very large cross sections. MBCR-I was increasingly robust to mis-specification of the inefficiency model due to its ability to learn from data and to consider locally shrunk individual inefficiency posteriors.

MBCR-I was empirically tested by using data on Japan's concrete firms operating between 2007 and 2010. Computation of input productivities, elasticities of substitution, inefficiency distributions, frontier-shifting effects and most productive scale sizes were demonstrated. MBCR-I's Bayesian framework made straightforward inference possible for the frontier-shifting effects.

There was limited evidence for the criticism that important parts of the construction industry offered significant room for efficiency improvements. Between 2007–2010, efficiency levels were stable and relatively high. Japan was significantly affected by the global financial crisis (Fukao and Yuan, 2009) as shown by the value-added output steadily declined over the 2007-2010 time period.

Future research should consider a more extensive exploration of the tradeoffs that estimating both a shape-constrained nonparametric production frontier and a flexible inefficiency term imply. A second research path should consider developing contextual variable models that do not rely on parametric assumptions. While we obtained credible intervals for all estimated parameters, given our non-linear least squares step to draw the hyperplane-specific regression coefficients, our credible intervals on both the production frontier and contextual variables were conditional on the point estimates obtained in that step. Although this assumption is not unreasonable, because it considers the best hyperplane fits conditional on all other parameters, future research, should also extend MBCR-I to draw the regression coefficients with a more efficient Bayesian algorithm in order to obtain full posterior distributions (and credible intervals) for the estimated parameters. Compared with StoNED, MBCR-I fitted a very small number of hyperplanes on highly clustered datasets, especially for low-dimensional input vectors. Thus, future research on alternative strategies for proposing knots and directions when adding hyperplanes should lead to more detailed frontier estimations.

V.3 Conclusions Specific to Chapter IV

This chapter describes two main contributions to the production function estimation literature. Firstly, we introduced CAP-NLS, a nonparametric estimator, which imposes global optimization and does not require refitting in contrast to CAP, as well as additional smoothing relative to CNLS. We formulated a homoscedastic version of CAP-NLS as a series of quadratic programs, which allowed for its computational feasibility. We demonstrated that CAP-NLS's additional structure relative to CAP and parsimonious

structure relative to CNLS translates into superior performance, smaller sensitivity against noise and input vector dimensionality, increased robustness on learning-to-full dataset settings and a much faster empirical rate of convergence on simulated data when the noise level is high relative to the full variance of the output and has similar performance to CAP when the noise level is relatively low to the full variance of the output.

Secondly and more importantly, we constructed a framework to test the adequateness of a production function estimator on real data. Specifically, we established a procedure based on repeated learning-testing and parametric bootstrapping that is able to assess the quality of subsample-fitted production functions to fit full survey (census) samples. Further, this procedure allows the relative quality of a subsample-fitted production function compared to full sample-fitted production function to be quantified. We demonstrated that unlike for simulated data, CAP-NLS, CAP and a Cobb-Douglas specification performed similarly for our application datasets. Our functional estimator selection procedure is not limited to CAP-NLS, the other tested methods or even nonparametric functional estimators, and thus should be routinely used for model selection of econometrically-estimated production functions. Finally, we discovered that the commonly-used Cobb Douglas production function can lead to very competitive approximations on the Chilean manufacturing dataset at all learning set sizes if an additive residual assumption is considered.

We locate the work done in this chapter within the production function estimation literature with the following assertion: Production function estimation had a "first generation" of estimators, which consisted of DEA and SFA. Later, "second generation" methods which overcame the clear shortcomings of DEA (its deterministic nature) and SFA (its parametric assumptions) arose in the production function literature. These methods, such

as CNLS and CWB, represented a first attempt to develop a nonparametric shape-constrained estimator with an explicit noise component in the model. However, this chapter demonstrates that such methods, which are based on optimization of specific observed datasets have important challenges when survey data is used to provide insights to an industry population. Thus, we conclude that a new generation of estimators which is able to overcome these challenges is needed. CAP and CAP-NLS, along other smoothed versions of Least Squares-based estimators, such as the estimators in Yagi et al. (2015) and Mazumder et al. (2015), are members of this new generation.

We have presented CAP-NLS emphasizing its practical benefits. However, open avenues of research such as proving consistency and setting bounds on CAP-NLS' fast rate of convergence remain open. Incorporation of smoothing strategies, such as the one presented in Mazumder et al. (2015), also are outstanding future lines of work. Regarding our procedures to assess the descriptive power of non-parametric methods in real data, further work can be done in replicating the procedure across a broader array of datasets, as we have restricted this exposition to the largest industries in the Chilean manufacturing dataset.

REFERENCES

Afriat, S. N. (1967). "The construction of utility functions from expenditure data." *International Economic Review* 8: 67-77.

Afriat, S. N. (1972). "Efficiency estimation of production functions." *International Economic Review* 13: 568-598.

Aigner, D., Lovell, C.A.K., and P. Schmidt (1977). "Formulation and estimation of stochastic frontier production function models." *Journal of Econometrics* 6: 21–37.

Allen, D. M. (1974). "The relationship between variable selection and data augmentation and a method for prediction." *Technometrics*, 16:125–127.

Allon, G., Beenstock, M., Hackman, S., Passy, U. and A. Shapiro (2007). "Nonparametric estimation of concave production technologies by entropic methods." *Journal of Applied Econometrics* 22: 795–816.

Almanidis, P. and R.C. Sickles (2011). The skewness issue in stochastic frontier models: Fact or fiction? in: I. van Keilegom and P. W. Wilson (Eds.), *Exploring Research Frontiers in Contemporary Statistics and Econometrics*. Springer Verlag, Berlin Heidelberg, pp. 201-228.

Alvarez, R. and R. A. Lopez (2008). "Is Exporting a Source of Productivity Spillovers?" *Review of World Economics* 144: 723–49.

Andor M. and F. Hesse (2014). "The StoNED age: the departure into a new era of efficiency analysis? A Monte Carlo comparison of StoNED and the "oldies" (SFA and DEA)." *Journal of Productivity Analysis* 41: 85-109.

Arlot, S. and A. Celisse (2010). "A survey of cross-validation procedures for model selection." *Statistics Surveys* 4: 40—79.

Balázs, G., András G., and C. Szepesvári (2015). "Near-optimal max-affine estimators for convex regression." *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics* 56-64.

Baltagi, B.H. and J.M Griffin (1988). "A General Index of Technical Change." *Journal of Political Economy* 96: 20-41.

Banfield, J.D. and A.E. Raftery (1993). "Model-Based Gaussian and Non-Gaussian Clustering." *Biometrics* 49: 803-821.

Banker R.D, Charnes A. and W.W. Cooper (1984). "Some models for estimating technical and scale inefficiencies in data envelopment analysis." *Management Science* 30: 1078–1092.

Barro, R.J. and X. Sala-I-Martin (2004). *Economic Growth*, MIT Press: Boston MA.

Biernacki C., G. Celeux and G. Govaert (1998). "Assessing a Mixture Model for Clustering with the Integrated Completed Likelihood." Technical Report No. 3521, INRIA, Rhône-Alpes.

———. 2000. "Assessing a Mixture Model for Clustering with the Integrated Completed Likelihood." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22: 719-725.

Biernacki, C. and G. Govaert (1997). "Using the Classification Likelihood to Choose the Number of Clusters." *Computing Science and Statistics* 29: 451-457

Biller, C. (2000). "Adaptive Bayesian Regression Splines in Semiparametric Generalized Linear Models." *Journal of Computational and Graphical Statistics* 9: 122-140.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and regression trees*. Wadsworth Statistics/Probability Series. Wadsworth Advanced Books and Software, Belmont, CA.

Breiman, L. (1992). “The little bootstrap and other methods for dimensionality selection in regression: X-Fixed predictor error.” *Journal of the American Statistical Association* 87: 738-754.

Burman, P. (1989). “A comparative study of ordinary cross-validation, *v*-fold cross-validation and the repeated learning-testing methods.” *Biometrika*, 76:503–514.

Charnes, A., Cooper, W.W. and E. Rhodes (1978). “Measuring the efficiency of decision making units”. *European Journal of Operational Research* 2:429-444.

Chavas, J.P. (2008). “A Cost Approach to Economic Analysis Under State-Contingent Production Uncertainty.” *American Journal of Agricultural Economics* 90:435–446.

Chopin, N. (2011). “Fast simulation of truncated Gaussian distributions.” *Statistics and Computing* 21: 275–288.

Chopin, N. and F. Pelgrin (2004). “Bayesian Inference and State Number Determination for Hidden Markov Models: An Application to the Information Content of the Yield Curve About Inflation.” *Journal of Econometrics* 123: 327–344.

Coelli, T.J., Rao, D.S.P., O'Donnell, C.J., Battese, G.E. (2005). *An Introduction to Efficiency and Productivity Analysis*. Springer.

Debreu, G. (1951). “The coefficient of resource utilization.” *Econometrica* 19: 273-292.

Denison, D., Mallick, B. and A. Smith (1998). “Automatic Bayesian curve fitting.” *Journal of the Royal Statistical Society: Series B* 60: 333–350.

Diewert, W. E and T.J. Wales (1987). “Flexible Functional Forms and Global Curvature Conditions.” *Econometrica* 55: 43–68.

Du P., Parmeter C.F. and J.S. Racine (2013). “Nonparametric kernel regression with multiple predictors and multiple shape constraints.” *Statistica Sinica* 23: 1347–1371.

Efron, B. (2004). "The estimation of prediction error: covariance penalties and cross-validation." *Journal of the American Statistical Association,* 99: 619–642.

Emrouznejad A., Parker B.R., G. Tavares (2008). "Evaluation of research in efficiency and productivity: a survey and analysis of the first 30 years of scholarly literature in DEA." *Socio-Economic Planning Sciences* 42: 151–157.

Farrell, M.J., 1957."The Measurement of Productive Efficiency". *Journal of the Royal Statistical Society: Series A* 120: 253-290.

Fukao, K. and T. Yuan. "Why is Japan so heavily affected by the global economic crisis? An analysis based on the Asian international input-output tables." *VOX CEPR's Policy Portal: Research-based policy analysis and commentary from leading economists,* 8 June 2009. Retrieved 27 September 2015.

Geisser, S. (1975). "The predictive sample reuse method with applications." *Journal of the American Statistical Association* 70:320–328.

Gelfand, A.E. and A. Kottas (2001). "Bayesian Semiparametric Median Regression Modeling." *Journal American Statistical Association* 96: 1458 – 1468.

Gelman, A. and J. Hill (2006). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, Cambridge, Massachusetts.

Geweke, J. (1991). Efficient Simulation from the Multivariate Normal and Student t Distributions Subject to Linear Constraints. In E. M. Keramidas and S. M. Kaufman, eds. *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*. Seattle: Interface Foundation of America, pp. 571–78.

Griffin, J. E. and M.F.J. Steel (2004). "Semiparametric Bayesian inference for stochastic frontier models." *Journal of Econometrics* 123: 121–152.

Green, P. J. (1995). "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination." *Biometrika* 82: 711-732.

Greene, W (1993). The econometric approach to efficiency analysis in H.O. Fried, C.A.K. Lovell, S.S. Schmidt (Eds.), *The Measurement of Productive Efficiency: Techniques and Applications*, Oxford University Press, Oxford pp. 68–11

Griffin, R. C., M.E. Rister, J.M. Montgomery, and F.T. Turner (1985). "Scheduling Inputs With Production Functions: Optimal Nitrogen Programs for Rice." *Southern Journal of Agricultural Economics* 17: 159-168.

Hannah, L.A. and D. Dunson (2011). "Bayesian Nonparametric Multivariate Convex Regression." arXiv preprint :1109.0322v1,

Hannah, L. A. and D. Dunson (201). "Ensemble Methods for Convex Regression with Applications to Geometric Programming Based Circuit Design." *Proceedings of the 29th International Conference on Machine Learning (ICML).*

Hannah, L. A. and D. Dunson (2013). "Multivariate Convex Regression with Adaptive Partitioning." *Journal of Machine Learning Research* 14: 3207–3240.

Hastie T, R. Tibshirani, and J.H. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer: New York, NY.

Hildreth, C. (1954) Point estimates of ordinates of concave functions. *Journal of the American Statistical Association* 49: 598–619.

Huang, R. and C. Szepesvári (2014). "A finite-sample generalization bound for semiparametric regression: Partially linear models." *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*.

Henderson D.J. and C.F. Parmeter (2009). Imposing Economic Constraints on Nonparametric Regression: Survey, Implementation and Extensions, in Q. Li and J. S. Racine, (Eds.), *Advances in Econometrics: Nonparametric Methods.* Elsevier Science, Volume 25, 433–469.

Huang, R. and C. Szepesvári (2014). "A finite-sample generalization bound for semiparametric regression: Partially linear models." *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*.

Hurn, M., A. Justel and C.P. Robert (2003). "Estimating Mixtures of Regressions." *Journal of Computational and Graphical Statistics* 12: 1-25.

"Japan and Abenomics: Taxing times." *The Economist*. 5 October 2013. Retrieved 28 January 2014.

Johnson, A.L. and T. Kuosmanen (2011). "One-stage Estimation of the Effects of Operational Condiitions and Practices on Productive Performance: Asymptotically Normal and Efficient, Root-N Consistent StoNEZD Method." *Journal of Productivity Analysis* 36: 219-230.

Jondrow, J., Lovell, C. K., Materov, I. S., and P.Schmidt (1982). "On the estimation of technical inefficiency in the stochastic frontier production function model." *Journal of Econometrics* 19: 233-238.

Koop, G. and D.J. Poirier (2004). "Bayesian variants of some classical semiparametric regression techniques." *Journal of Econometrics* 123: 259-282.

Koopmans T. (1951). "Efficient allocation of resources". *Econometrica* 19: 455-465.

Kumbhakar, S. C., and C.A.K. Lovell (2003). *Stochastic frontier analysis*. Cambridge University Press.

Kumbhakar S.C., Park, B.U., Simar, L. and E.G. Tsionas (2007). “Nonparametric stochastic frontiers: a local maximum likelihood approach.” *Journal of Econometrics* 137: 1–27.

Kuosmanen T. (2008). “Representation theorem for convex nonparametric least squares.” *Econometrics Journal* 11: 308–325.

Kuosmanen T. and A.L. Johnson (2010). “Data Envelopment Analysis as Nonparametric Least-Squares Regression.” *Operations Research* 58: 149-160.

Kuosmanen T. and M. Kortelainen (2012). “Stochastic non-smooth envelopment of data: semi-parametric frontier estimation subject to shape constraints.” *Journal of Productivity Analysis* 38:11-28.

Lee, C.Y., Johnson, A.L., Moreno-Centeno, E. and T. Kuosmanen (2013). “A more efficient algorithm for convex nonparametric least squares.” *European Journal of Operational Research* 227: 391-400.

Mahendran, N., Wang, Z., Hamze, F., and de Freitas, N (2012). “Adaptive MCMC with Bayesian Optimization.” *Journal of Machine Learning Research - Proceedings Track* 22: 751-760.

Mazet, V., Pseudo random Numbers from a Truncated Gaussian Distribution. Computer software. Université De Strasbourg, 09 Sept. 2014. Web. 27 June 2015.

Mazumder, R., et al. (2015). "A Computational Framework for Multivariate Convex Regression and its Variants." arXiv preprint: 1509.08165.

Meeusen, W. and van den Broeck, J. (1977). “Efficiency Estimation from Cobb-Douglas Production Functions with Composed Error”. *International Economic Review* 18: 435-444.

Mekaroonreung, M. and A.L. Johnson (2012). “Estimating the Shadow Prices of SO2 and NOx for U.S. Coal Power Plants: A Convex Nonparametric Least Squares Approach.” *Energy Economics* 34: 723-732.

Meyer, M.C., Hackstadt, A., and J.A. Hoeting (2011). “Bayesian Estimation and Inference for Generalized Partial Linear Models using Shape-Restricted Splines.” *Journal of Nonparametric Statistics* 23: 867-884.

Michaelides, P. G., Tsionas, E. G., Vouldis, A. T. and K.N. Konstantakis (2015). “Global approximation to arbitrary cost functions: A Bayesian approach with application to US banking.” *European Journal of Operational Research* 241: 148–160.

Nauges, C., C.J. O'Donnell and J. Quiggin (2011). "Uncertainty and Technical Efficiency in Finnish Agriculture: a State-Contingent Approach." *European Review of Agricultural Economics* 38: 449-467.

O'Donnell, C.J. and T.J. Coelli (2005). "A Bayesian Approach to Imposing Curvature on Distance Functions." *Journal of Econometrics* 126: 493-523.

O'Donnell, C.J. and W. E. Griffiths (2006). "Estimating State-Contingent Production Frontiers." *American Journal of Agricultural Economics* 88: 249-266.

O'Donnell, C.J. and S.Shankar (2010). "Regulatory Reform and Economic Performance in US Electricity Generation Estimating State-allocable Production Technologies When there are Two States of Nature and State Allocations of Inputs are Unobserved." CEPA Working Papers Series WP042010, School of Economics, University of Queensland, Australia.

Oh, H., A.L. Johnson, L. Lucianetti, and S. Youn (2015). “The Effect of Performance Measurement Systems on Productive Performance: An Empirical Study of Italian Manufacturing Firms.” Available at SSRN: http://ssrn.com/abstract=2677354.

Olesen O.B. and N.C. Petersen (1996). “Indicators of Ill-Conditioned Data Sets and Model Misspecification in Data Envelopment Analysis: An Extended Facet Approach.” *Management Science* 42: 205-219.

Orea, L. and S.C. Kumbhakar (2004). “Efficiency Measurement Using a Latent Class Stochastic Frontier Model.” *Empirical Economics* 29:169-183.

Quiggin, J. C. and R.G. Chambers (2006). "The State-contingent Approach to Production Under Uncertainty." *Australian Journal of Agricultural and Resource Economics* 50: 153-169.

Richardson, S. & P.J. Green (1997). "On Bayesian Analysis of Mixtures with an Unknown Number of Components (with discussion)." *Journal of the Royal Statistical Society: Series B* 59: 731–792.

Robert, C.P. (1995). "Simulation of Truncated Normal Variables." *Statistics and computing* 5: 121-125.

Sarmento, M., Johnson, A.L., Preciado Arreola, J.L., and Ferrier, G.D (2016). “Cost Efficiency of U.S. Hospitals: A Semi-parametric Bayesian Analysis.” Working paper, Department of Industrial and Systems Engineering, Texas A&M University, College Station, Texas.

Serra, T., S.Stefanou and A.O.Lansink (2010). "Dynamic State Contingent Production and Capital Accumulation in U.S. Agriculture." *European Review of Agricultural Economics* 37: 293–312.

Shao, J. (1993). “Linear model selection by cross-validation.” *Journal of the American Statistical Association* 88:486–494.

Shively T., Walker S. and P. Damien (2011). “Nonparametric function estimation subject to monotonicity, convexity and other shape constraints.” *Journal of Econometrics* 161: 166–181.

Simar L. and V. Zelenyuk (2011). “Stochastic FDH/DEA estimators for frontier analysis.” *Journal of Productivity Analysis* 36: 1–20.

Solow, R. (1957). "Technical change and the aggregate production function." *Review of Economics and Statistics* 39: 312–320.

Stone, M. (1974). “Cross-validatory choice and assessment of statistical predictions (with discussion and a reply by the authors).” *Journal of the Royal Statistical Society Series B* 36:111–147.

Syverson, C (2004). “Market Structure and Productivity: A concrete example.” *Journal of Political Economy* 112: 1181–1222.

Syverson, C. (2011). "What Determines Productivity?" *Journal of Economic Literature* 49: 326-365.

Tsionas E.G. (2000). “Full likelihood inference in normal-gamma stochastic frontier models.” *Journal of Productivity Analysis* 13: 179–201.

Ramaratnam, S., D.A. Bessler, M.E. Rister, J. Novak, and J.E. Matocha. (1987). "Fertilization Under Uncertainty: An Analysis Based on Subjective Producer Yield Expectations." *American Journal of Agricultural Economics* 69: 349-357.

Stephens, M. (2000). "Bayesian Analysis of Mixtures with an Unknown Number of Components - an Alternative to Reversible Jump Methods." *Annals of Statistics* 28: 40-74.

van den Broeck, J., G. Koop, Osiewalski J., and M. Steel (1994). “Stochastic Frontier Models: a Bayesian Perspective.” *Journal of Econometrics* 61: 273–303.

Villano, R.A., C.J. O'Donnell, and G.E. Battese (2004). "An Investigation of Production Risk, Risk Preferences and Technical Efficiency: Evidence from Rainfed Lowland Rice Farms in the Philippines." Paper presented at Asia-Pacific Productivity Conference, Brisbane, 14–16 July.

Wu, Ximing and R.Sickles (2013). "Semiparametric Estimation under Shape Constraints." Working paper, Department of Agricultural Economics Texas A&M University, College Station, Texas.

Yagi D., Johnson, A.L. and T. Kuosmanen (2016). "Shape constrained kernel weighted least squares for the estimation of production functions." Working Paper. Department of Industrial and Systems Engineering, Texas A&M University.

APPENDIX A

CONDITIONAL POSTERIOR HYPERPARAMETERS FOR CHAPTER II

The following are the mathematical expressions for the posterior hyperparameters $\bar{\xi}$, $\bar{\kappa}$, which partially define the restricted multivariate normal conditional posterior distribution of β. The posterior multinomial parameter $\bar{d}_{it}$, which describes a probability vector for states 1 to *J* from which a multinomial draw d_{it} is to be simulated. Finally, parameters μ_{ui} and σ^2_{ui} partially describe the truncated normal conditional posterior distribution of u_i, the inefficiency term associated to farm *i*.

$$\bar{\kappa}= \left((\underline{\kappa} \otimes I_J)^{-1} + \sum_{i=1}^{N} \sum_{t=1}^{T} (d_{it}' h) z_{it} z_{it}' \right)^{-1} ; \tag{A.1}$$

$$\bar{\xi}= \bar{\kappa} \left((\underline{\kappa} \otimes I_J)^{-1} \underline{\xi} + \sum_{i=1}^{N} \sum_{t=1}^{T} (d_{it}' h) z_{it} y_{it} \right)^{-1} ; \tag{A.2}$$

$$\bar{d}_{it}=\left[\frac{\pi_1 f_N(y_{it}|\beta_{01}+(x_{it}\otimes i_1)'\beta_{-0}, h_1^{-1})}{\sum_{j=1}^{J} \pi_j f_N(y_{it}|\beta_{0j}+(x_{it}\otimes i_j)'\beta_{-0}, h_j^{-1})}, \ldots, \frac{\pi_J f_N(y_{it}|\beta_{0J}+(x_{it}\otimes i_J)'\beta_{-0}, h_J^{-1})}{\sum_{j=1}^{J} \pi_j f_N(y_{it}|\beta_{0j}+(x_{it}\otimes i_j)'\beta_{-0}, h_j^{-1})} \right]' ; \tag{A.3}$$

$$\mu_{ui}= \frac{\sum_{t=1}^{T} (\mathbf{d}_{it}'\mathbf{h})(\mathbf{z}_{it}'\boldsymbol{\beta}-y_{it})-\lambda^{-1}}{\sum_{t=1}^{T} \mathbf{d}_{it}'\mathbf{h}} \text{ and } \sigma^2_{ui}= \left[\sum_{t=1}^{T} \mathbf{d}_{it}'\mathbf{h} \right]^{-1} ; \tag{A.4}$$

APPENDIX B

TRACE PLOTS OF ESTIMATED PARAMETERS FOR CHAPTER II

Trace plots for the main estimated parameters for both the linear and dummy time trend models are included in this appendix. All of them exhibit stationary behavior, indicating proper mixing.

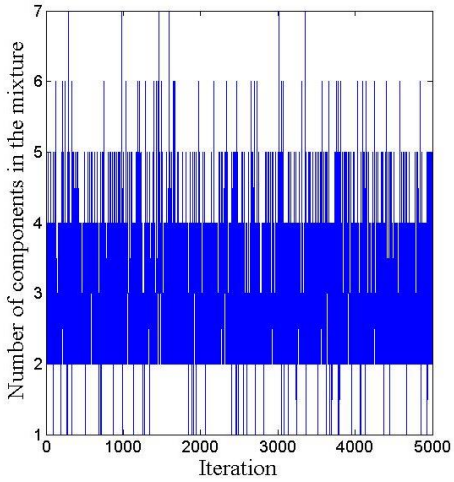

Figure B1 Trace plots for number of components in the mixture after burn-in for dummy time trend model.

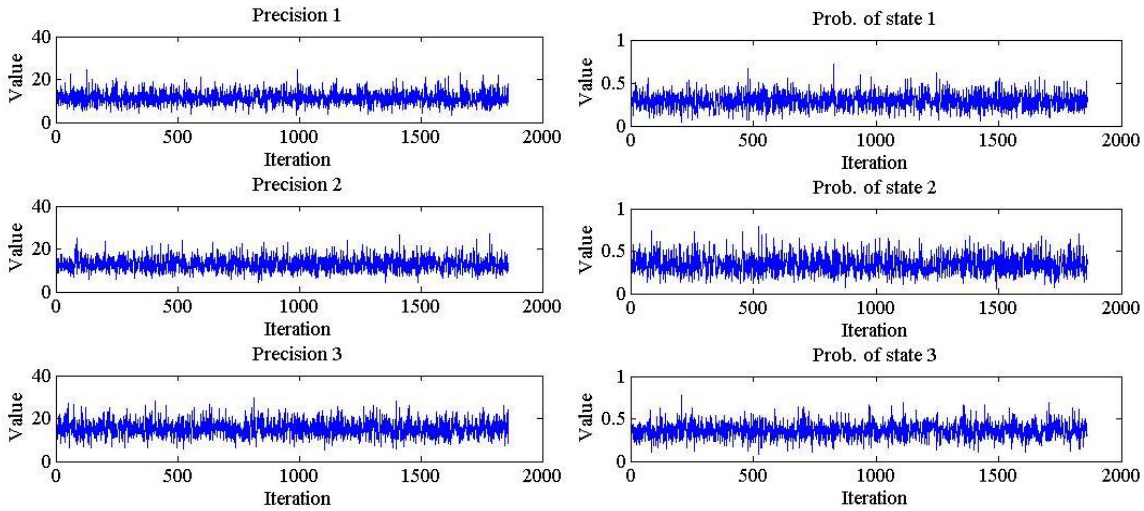

Figure B2 Trace plots for precisions and state probabilites for dummy time trend model.

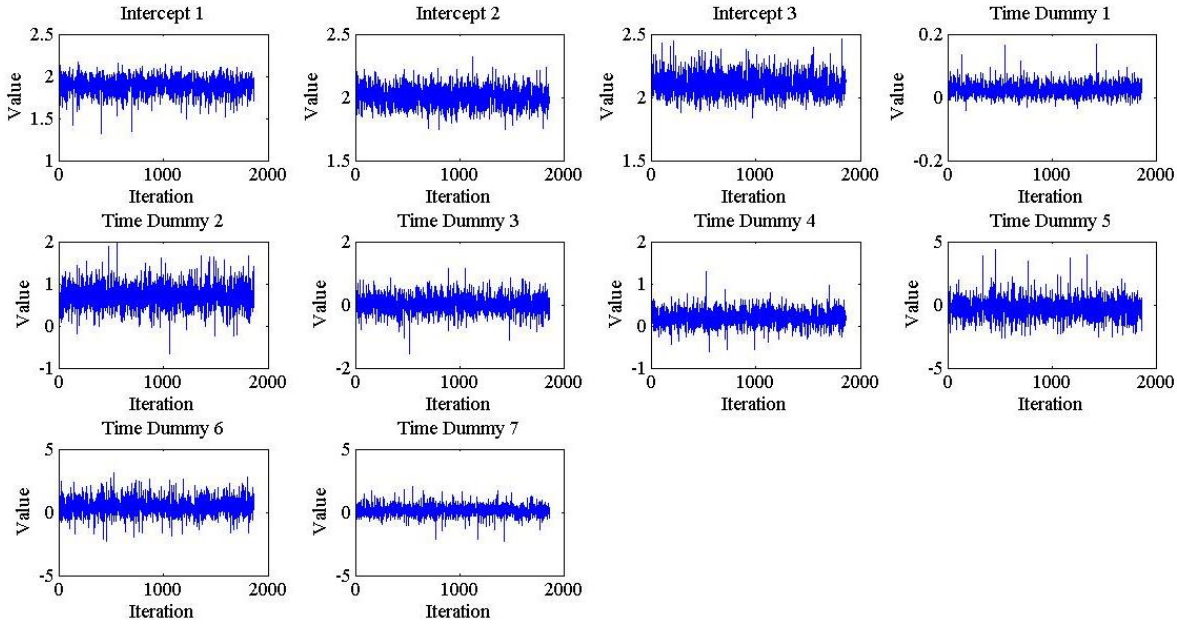

Figure B3 Trace plots for intercepts and dummy variables for dummy time trend model

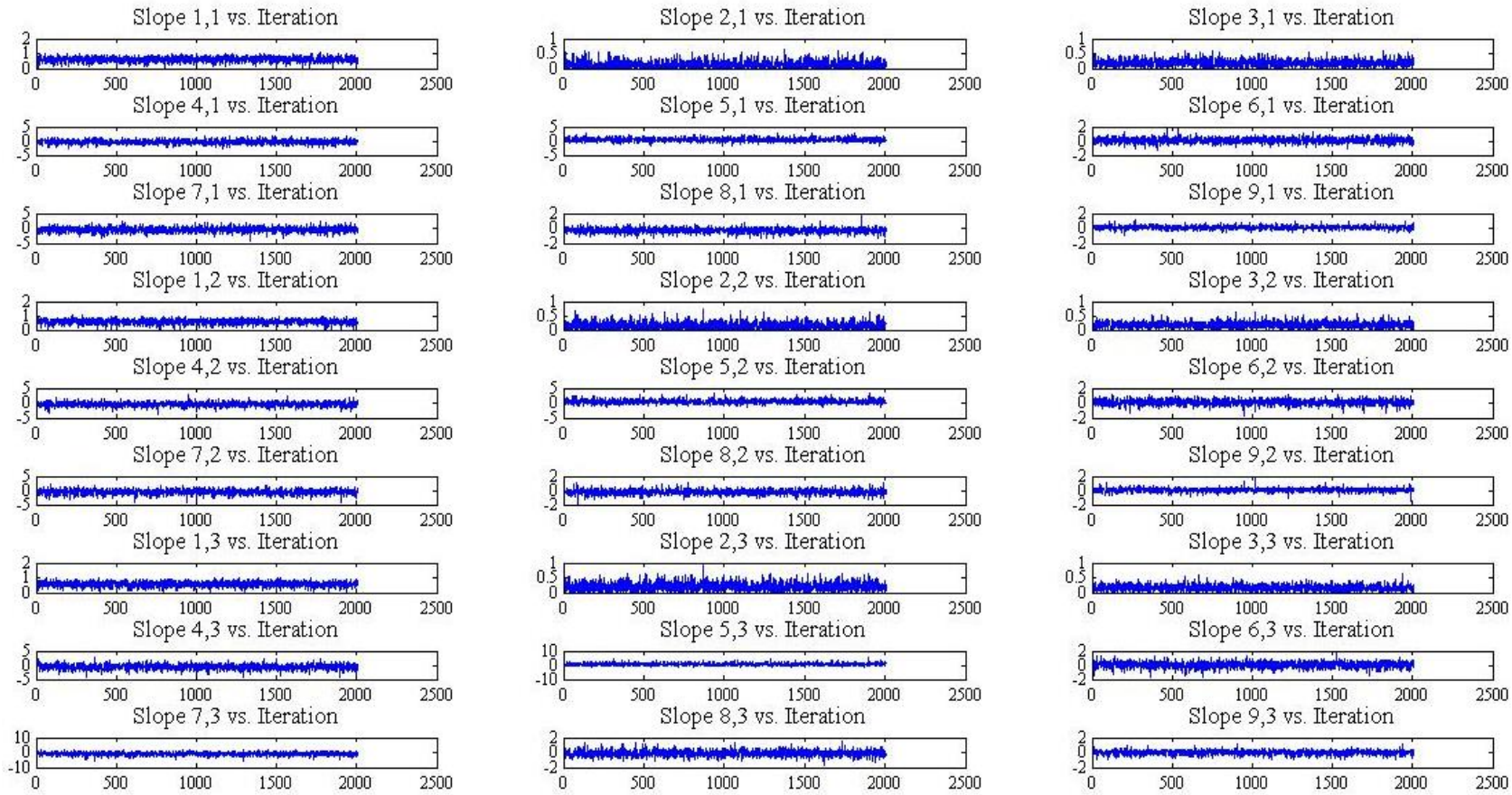

Figure B4 Trace plots for slopes for dummy trend model

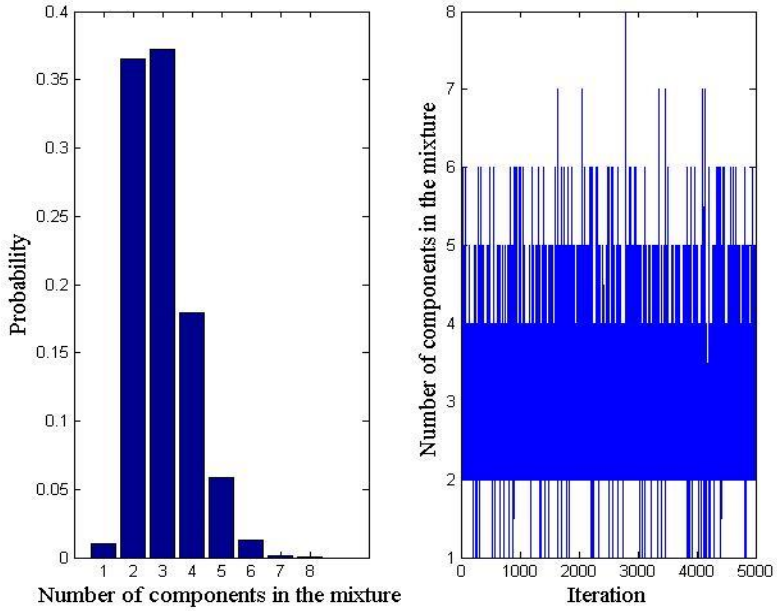

Figure B5 Number of components for linear trend model and trace plot comparing their draws against the iteration number.

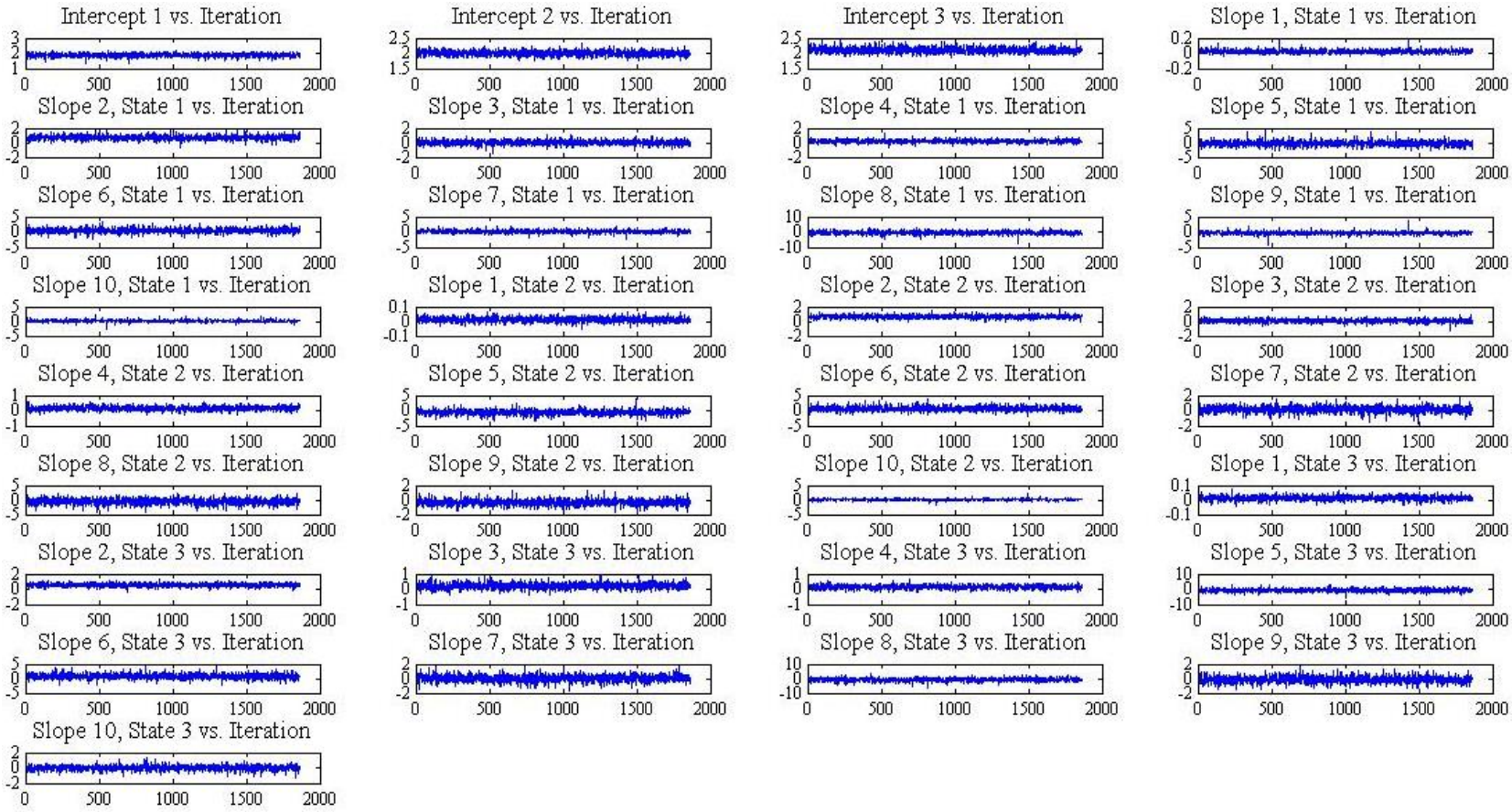

Figure B6 Trace plots of intercepts and slopes for linear time trend model.

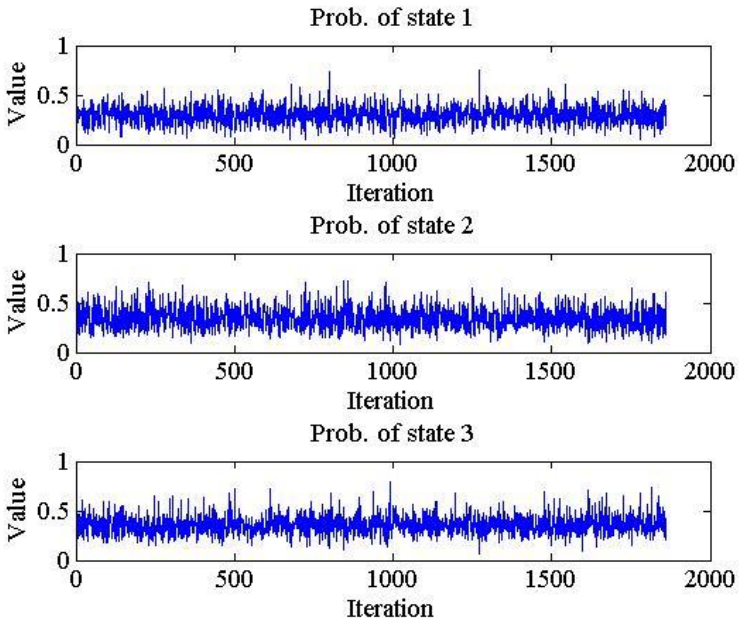

Figure B7 Trace plots of state probabilities for linear trend model

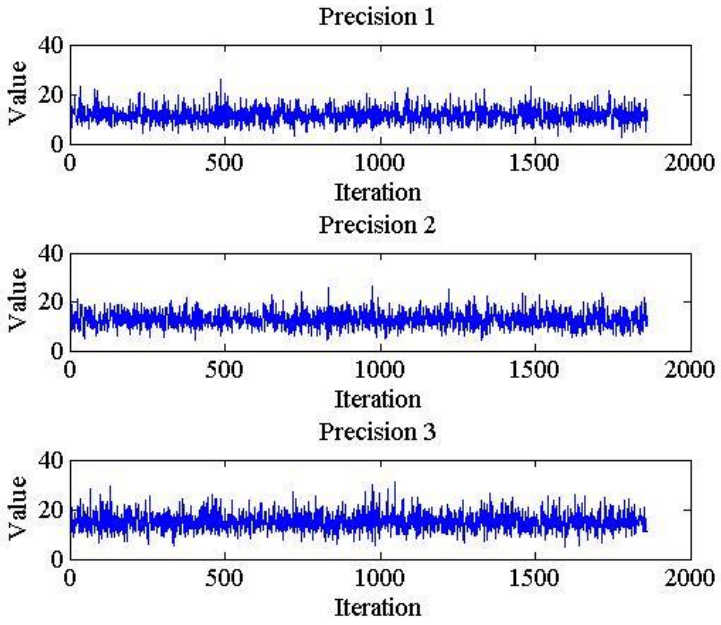

Figure B8 Trace plots of precisions for linear trend model

APPENDIX C

FULL SENSITIVITY ANALYSIS AND GOODNESS OF FIT RESULTS FOR CHAPTER II

This section contains sensitivity analysis and goodness of fit results for both the linear time trend and dummy time trend models. Our first results, shown in table C1, show the posterior technical efficiency distribution for the dummy time trend model with different assumptions for the prior lower bound on technical inefficiency.

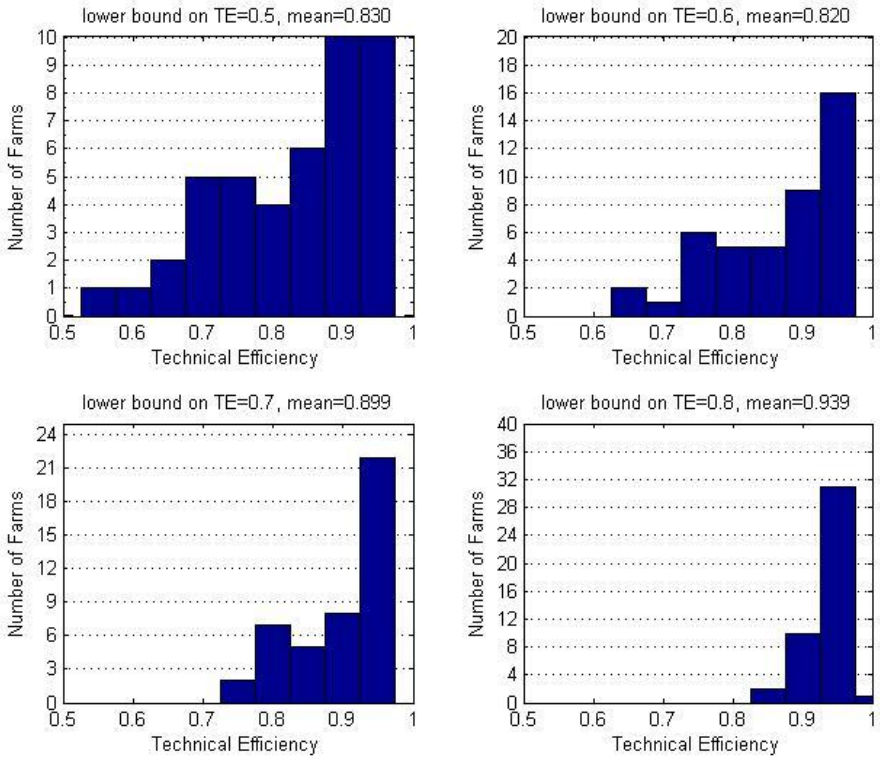

Figure C1 Technical efficiency distribution under different assumptions for lower bound of TE for the dummy time trend model

The reminder of this appendix focuses on results for the linear time trend model. As mentioned on this chapter, the efficiency rankings do not significantly change across the

considered range of prior values for ζ and τ^*. Goodness of fit results exhibit similar behavior to those of the dummy trend model presented here.

Table C1. Sensitivity Analysis on Hyperparameter λ, Prior Mean on the Number of States for the Linear Trend Model

Prior lambda	Posterior mode for *J*	Posterior mean for *J*	P(*J*=2|*X*)	P(*J*=3|*X*)	90% HPD[a]
1	2	2.83	0.398	0.386	{2,3,4}
2	3	2.87	0.381	0.384	{2,3,4}
3	3	2.93	0.370	0.372	{2,3,4}
4	3	2.95	0.357	0.375	{2,3,4}
5	3	2.87	0.381	0.384	{2,3,4}

[a] Highest Posterior Density set

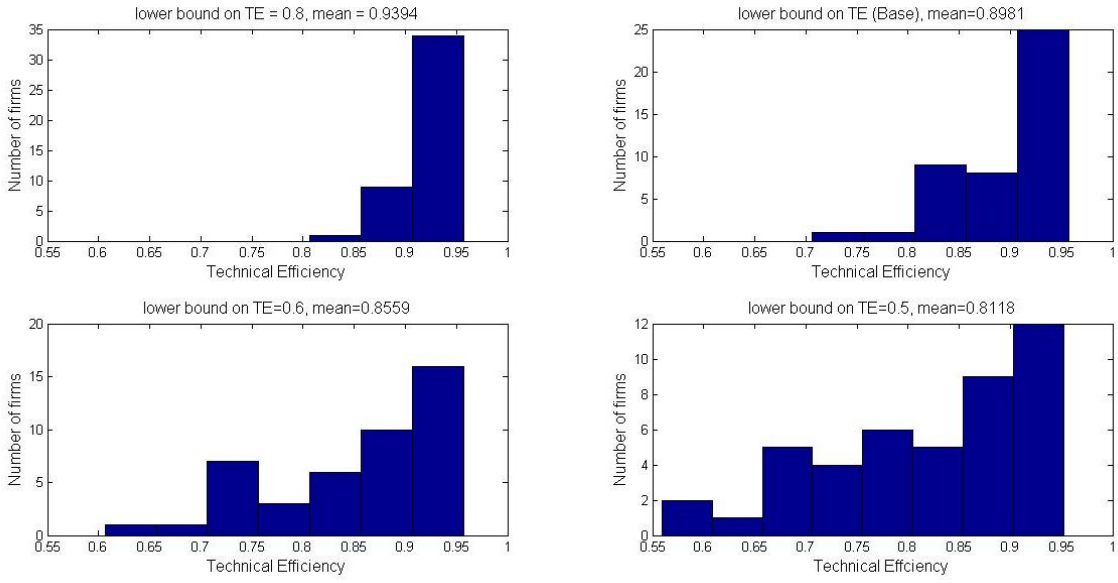

Figure C2 Technical efficiency distribution under different assumptions for lower bound of TE for the linear time trend model.

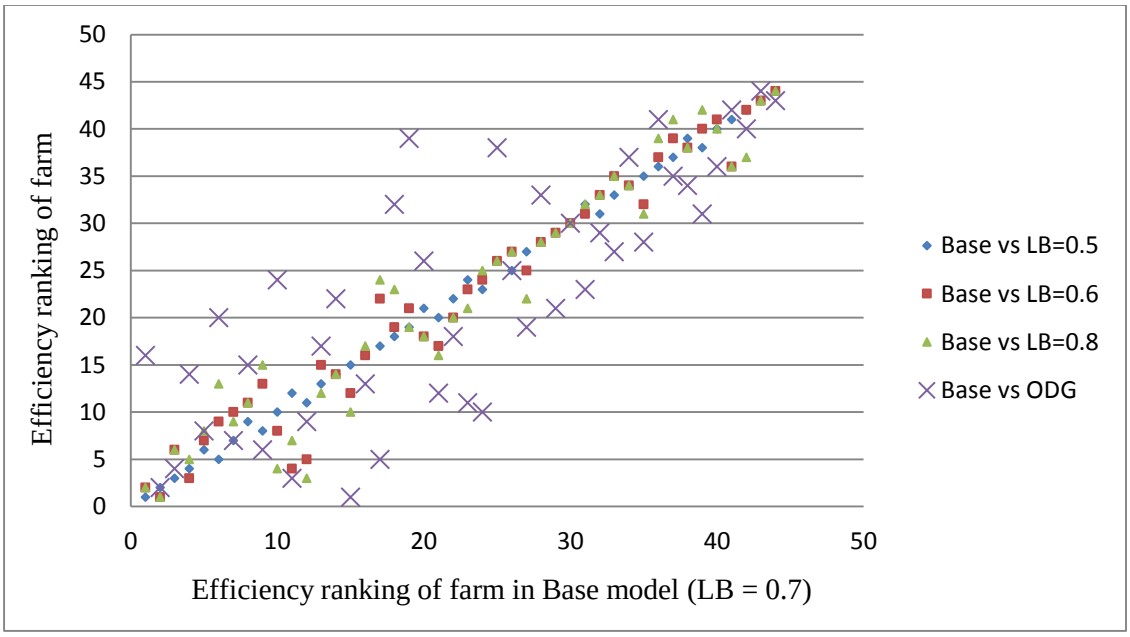

Figure C3 Rankings of the three alternative prior TE lower bound assumptions and O'Donnell and Griffiths (2006)'s monotonicity-constrained SC model against the base value of 0.7 for the linear model.

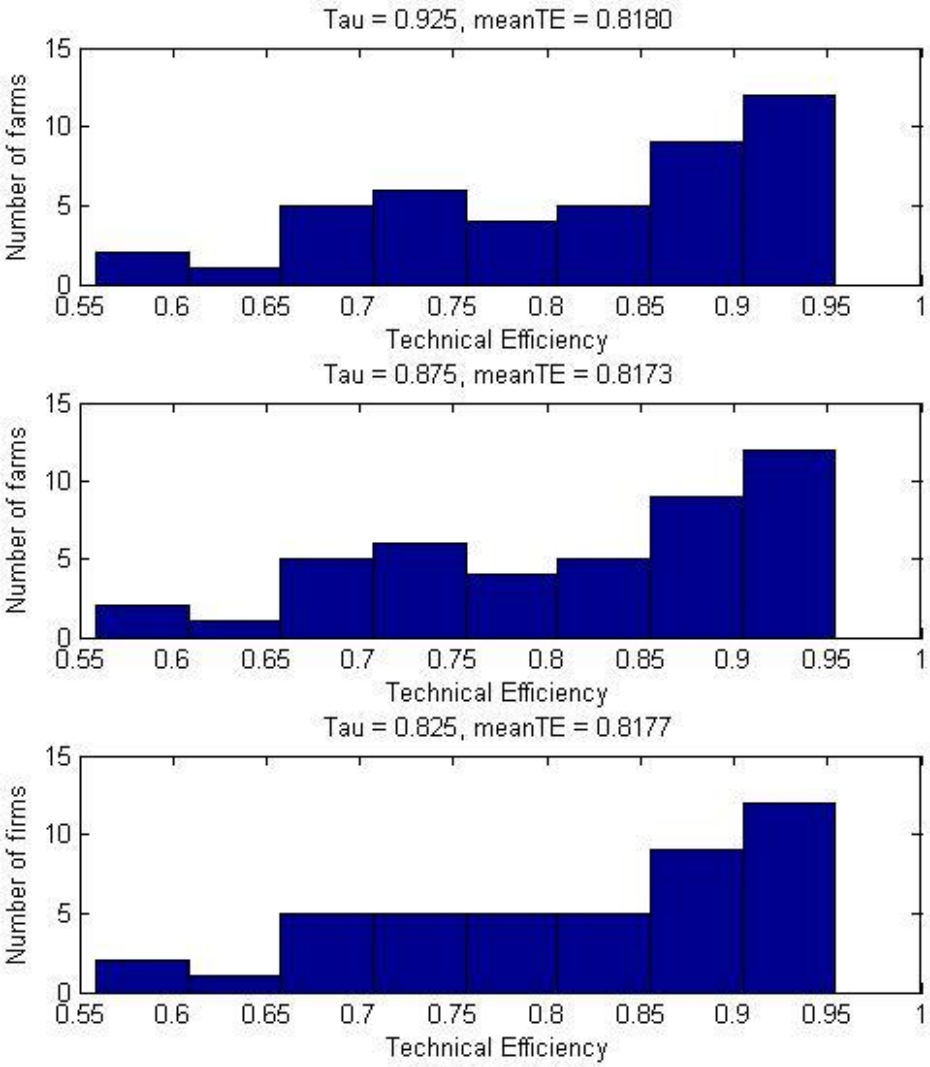

Figure C4 Technical Efficiency distribution for low (top panel), base (middle panel) and high (bottom panel) values of τ^* for the linear time trend model.

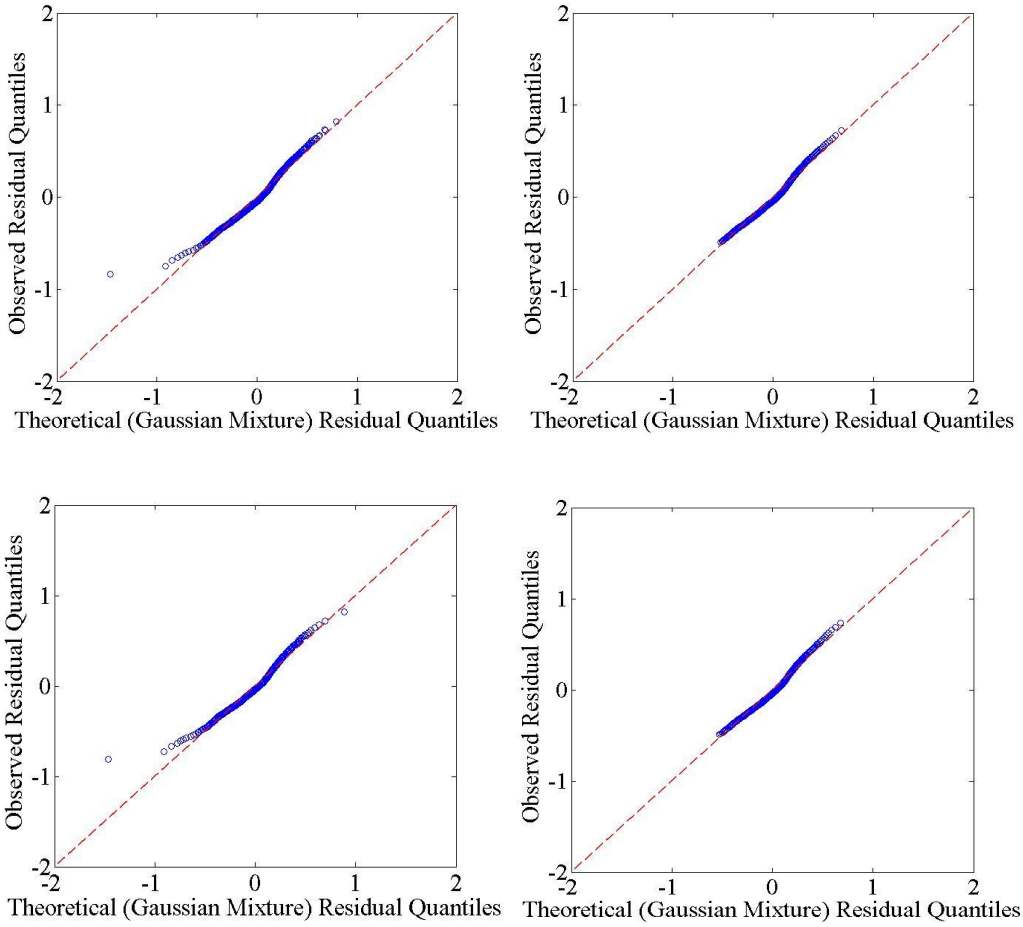

Figure C5 Goodness of fit results for linear model. Full results for full posterior scenario(top left panel), 4% outliers removed for full posterior scenario (top right panel), full results for mode scenario(bottom left panel), 4% outliers removed for mode scenario (bottom right panel).

APPENDIX D

FULLY BAYESIAN MBCR-I

As mentioned in Section III.1.3, we solve mathematical program (III.7) instead of simulating $(\alpha_k,\beta_k)_{k=1}^K$ from its posterior distribution due to the difficulty of obtaining good proposal distributions. Computational feasibility of the original, additive error structure, MCBR algorithm depends on the availability of good proposal distributions for $(\alpha_k,\beta_k)_{k=1}^K$ (Hannah and Dunson, 2011). While H-D take advantage of conjugacy to compute such proposal distributions, we are not able to do so due to the logarithm operator present in our likelihood function, $\prod_{i=1}^n N\left(\ln(Y_i)- \ln\left(\alpha_{[i]}+\beta_{[i]}^T\mathbf{X}_i\right)+u_i,\sigma_{[i]}^2\right)$. Here, we propose a naïve way to obtain full conditional posterior distributions for $(\alpha_k,\beta_k)_{k=1}^K$. The algorithm to draw (α_k,β_k) for each basis region is as follows:

1. Solve mathematical program (III.7) to obtain candidate distribution mean estimates (α_{k1},β_{k1}) for the given basis region.

2. Draw the candidate values $(\alpha_{kCand},\beta_{kCand})$ from a (d+1)-dimensional Multivariate Gaussian distribution $N_{(d+1)}(\left(\alpha_{k1},\beta_{k1}\right),\Sigma_{\alpha,\beta})$, where $\Sigma_{\alpha,\beta}=\eta\begin{bmatrix}\sigma_{\alpha_k}^2 & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \sigma_{\beta_{dk}}^2\end{bmatrix}$, where $\eta>1$ is a tunable parameter, and $\sigma_{\alpha_k},\ldots,\sigma_{\beta_{dk}}^2$ are the standard errors of α_{k1} and each component of β_{k1} obtained after solving (7).

3. Draw σ_{kCand}^2 from (III.8), assuming $(\alpha_{kCand},\beta_{kCand})$.

4. Compute the Metropolis-Hastings acceptance probability a_{Prob} for $(\alpha_{kCand},\beta_{kCand},\sigma_{kCand}^2)$ using (D1)-(D4):

D1. $a_{Prob}=\frac{p(\alpha_{kCand},\beta_{kCand},\sigma^2_{kCand})L_k(\alpha_{kCand},\beta_{kCand},\sigma^2_{kCand})}{q(\alpha_{kCand},\beta_{kCand},\sigma^2_{kCand}|\alpha_{kCurr},\beta_{kCurr},\sigma^2_{kCurr})}\Big/\frac{p(\alpha_{kCurr},\beta_{kCurr},\sigma^2_{kCurr})L_k(\alpha_{kCurr},\beta_{kCurr},\sigma^2_{kCurr})}{q(\alpha_{kCurr},\beta_{kCurr},\sigma^2_{kCurr}|\alpha_{kCand},\beta_{kCand},\sigma^2_{kCand})}$

D2. $p(\alpha,\beta,\sigma^2)\sim N_{(d+1)}IG(\mathbf{0},MI,\tilde{a},\tilde{b})$, where *M* is a large number and *I* is the identity matrix

D3. $L_k(\alpha,\beta,\sigma^2)=\prod_{i=1}^{n_k} N\left(\ln(Y_i)-\ln\left(\alpha+\beta^T\mathbf{X}_i\right)+u_i,\sigma^2\right)$

D4. $q\left(\alpha_2,\beta_2,\sigma^2{}_2\middle|\alpha_1,\beta_1,\sigma^2{}_1\right)\sim N_{(d+1)}IG(\mu_1,\Sigma_1,a_1,b_1)$, where (μ_1,Σ_1,a_1,b_1) are the hyperparameters associated with $(\alpha_1,\beta_1,\sigma^2{}_1)$.

5. If $a_{Prob}>Unif(0,1)$, accept draw $\left(\alpha_{kCand},\beta_{kCand},\sigma^2_{kCand}\right)$; otherwise, go back to step 2.

6. After accepting a predefined number n_{bi} of burn-in draws, accept the $(n_{bi}+1)$th accepted draw as a valid draw from the posterior distribution of $(\alpha_k,\beta_k,\sigma^2_k)$.

APPENDIX E

FRONTIER CHARACTERISTICS FOR 2008-2009 CROSS SECTIONAL DATASETS

Tables E1 and E2 show that the Marginal Product and Technical Efficiency quantiles are similar to those of 2010 for almost all cases, which is also true for the Elasticity of Substitution between Capital and Labor, except for its maximum values. In the case of MPSS shown in Table E3, there are no significant departures from the values for 2007 and 2010.

Table E1 MBCR-I S Cross Sectional Production Frontier Characterization for 2008

	Marginal Product		Elasticity of Substitution ($x10^{-4}$)	Technical Efficiency
	Capital	Labor	Capital/Labor	Firm-specific %
Min	0.0093	1,119	0.0767	1.26%
25th percentile	1.1209	1,202	9.3052	51.90%
Median	1.1209	1,206	9.3052	71.77%
75th percentile	1.1209	1,206	9.3052	87.07%
Max	1.4449	1,262	13.142	99.97%

Table E2 MBCR-I S Cross Sectional Production Frontier Characterization for 2009

	Marginal Product		Elasticity of Substitution ($x10^{-4}$)	Technical Efficiency
	Capital	Labor	Capital/Labor	Firm-specific %
Min	0.0662	910	0.5608	0.97%
25th percentile	1.0010	1,191	8.4305	50.82%
Median	1.0010	1,191	8.4305	71.04%
75th percentile	1.0012	1,191	8.4328	86.72%
Max	10.0217	1,220	134.53	99.97%

Table E3 Most Productive Scale Size for Selected Capital/Labor Ratios for Cross Sectional Frontiers 2008-2009

Model	Capital/Labor ratio percentile	10%	25%	50%	75%	90%
Cross Sectional 2008	Capital/Labor	38	75	166.7	400	1,434
	MPSS Capital	1,226	1,266	1,356	1,573	2,445
	MPSS Labor	32	16	8	4	2
Cross Sectional 2009	Capital/Labor	37.5	76.9	166.7	391.3	1,470
	MPSS Capital	1,217	1,254	1,334	1,546	2,510
	MPSS Labor	31	16	8	4	2

APPENDIX F

THE FLEXIBILITY TRADEOFF BETWEEN PRODUCTION FUNCTION AND INEFFICIENCY DISTRIBUTION ASSUMPTIIONS

The potential exists for a flexibility tradeoff between modeling the production function nonparametrically and modeling the unobserved inefficiency without distribution assumptions. Since our simulated datasets consist of well-distributed data along the input-output space, we do not need an MBCR prior to obtain computationally efficient runs of MBCR-I. Nevertheless, the more clustered and uneven distribution of observations in our application datasets benefit computationally from a more informative prior on the curvature of the function given by MBCR. When we run MBCR on both datasets, the number of hyperplanes needed to describe the production frontier decreases when inefficiency is introduced. The results for MBCR-I in Table F1 and Table 15 are the same.

Table F1 Fitting Statistics Comparison for MBCR and MBCR-I for the Japanese Concrete Industry Dataset.

MBCR	# Hyperplanes fitted	Median Inefficiency	% SS Model	% SS Noise
Cross Sectional 2007	2.22	0%	55%	45%

Table F1 (continued) Fitting Statistics Comparison for MBCR and MBCR-I for the Japanese Concrete Industry Dataset.

Cross Sectional 2008	2.18	0%	55%	45%
Cross Sectional 2009	3	0%	53%	47%
Cross Sectional 2010	4	0%	55%	45%
Panel	4	0%	56%	54%

MBCR-I	# Hyperplanes fitted	Median Inefficiency	% SS Model	% SS Noise
Cross Sectional 2007	2.11	29%	66%	34%
Cross Sectional 2008	2.03	28%	75%	25%
Cross Sectional 2009	2.60	29%	70%	30%
Cross Sectional 2010	2.36	27%	76%	24%
Panel	3.63	43%	81%	19%

Note that an omitted inefficiency model results in a more complex nonparametric shape-constrained frontier estimation, whereas the use of a moderately flexible inefficiency model results in frontier estimations with fewer hyperplanes. We believe very general inefficiency distribution specifications, especially non-parametric specifications considering non-monotonic or heteroscedastic behaviors, will result in coarser production frontier estimates. That is, if both the frontier specification and the estimated values are very flexible, we may encounter an identification problem. We emphasize that this insight is only possible due to the one-stage nature of MBCR-I.

APPENDIX G

A COARSE GRID SEARCH ALGORITHM TO COMPUTE MPSS ON A MULTIVARIATE SETTING

Calculate MPSS at a given Capital/Labor ratio as follows:

1. Let $\boldsymbol{\omega}=(\omega_1,\ldots,\omega_L)$ be a uniform grid of L points across the Labor axis.
2. Set R=Capital/Labor to a value of interest; for example, a percentile of the empirical Capital/Labor distribution.
3. Evaluate the predicted frontier output value $\hat{f}_l(R\cdot\omega_l,\omega_l)$ for each $l\in\{1,\ldots L\}$.
4. Compute distance D_l from $(R\cdot\omega_l,\omega_l)$ to the origin for each $l\in\{1,\ldots L\}$.
5. Calculate $MPSS^{*}=\max_{l\in\{1,\ldots,L\}}\hat{f}(R\cdot\omega_l,\omega_l)/D_l$.

APPENDIX H

RESULTS FOR HOMOSCEDASTIC ADDITIVE ERROR SIMULATED DATASETS

d = 2, sigma = 0.2		nFull = 100			nFull = 200			nFull = 300		
nL/nF	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0025	0.0069	0.0030	0.0014	0.0020	0.0017	0.0012	0.0019	0.0015
100%	MSEFTesting	0.0028	0.0073	292.9882	0.0014	0.0020	93.3874	0.0012	0.0019	168.9078
100%	MSEFCensus	0.0025	0.0069	0.0030	0.0014	0.0020	0.0017	0.0012	0.0019	0.0015
80%	MSEFInSample	0.0045	0.0160	0.0054	0.0024	0.0053	0.0029	0.0019	0.0273	0.0023
80%	MSEFTesting	0.0047	0.0173	56.2393	0.0026	0.0055	414.9028	0.0019	0.0249	148.8055
80%	MSEFCensus	0.0046	0.0166	28.1224	0.0025	0.0054	207.4528	0.0019	0.0261	74.4039
50%	MSEFInSample	0.0066	0.0145	0.0081	0.0035	0.0085	0.0041	0.0026	0.0045	0.0031
50%	MSEFTesting	0.0076	0.0169	3.6138	0.0039	0.0090	76.2141	0.0026	0.0044	819.0860
50%	MSEFCensus	0.0073	0.0162	2.5321	0.0038	0.0089	53.3511	0.0026	0.0045	573.3612
30%	MSEFInSample	0.0093	0.0153	0.0117	0.0051	0.0081	0.0063	0.0041	0.0048	0.0047
30%	MSEFTesting	0.0112	0.0188	3.7562	0.0056	0.0087	50.9327	0.0044	0.0055	85.4171
30%	MSEFCensus	0.0108	0.0181	3.0073	0.0055	0.0086	40.7474	0.0043	0.0054	68.3346
100%	MSEYInSample	0.0426	0.0467	0.0422	0.0415	0.0418	0.0425	0.0415	0.0424	0.0412
100%	MSEYTesting	0.0428	0.0472	293.0517	0.0415	0.0420	93.4378	0.0415	0.0421	168.9763
100%	MSEYCensus	0.0426	0.0467	0.0422	0.0415	0.0418	0.0425	0.0415	0.0424	0.0412
80%	MSEYInSample	0.0449	0.0551	0.0447	0.0428	0.0451	0.0421	0.0418	0.0660	0.0421
80%	MSEYTesting	0.0448	0.0573	56.2969	0.0426	0.0455	414.9679	0.0420	0.0649	148.8733
80%	MSEYCensus	0.0449	0.0562	28.1708	0.0427	0.0453	207.5050	0.0419	0.0654	74.4577
50%	MSEYInSample	0.0470	0.0552	0.0494	0.0430	0.0484	0.0440	0.0418	0.0437	0.0440
50%	MSEYTesting	0.0478	0.0571	3.6587	0.0442	0.0492	76.2684	0.0429	0.0446	819.1599
50%	MSEYCensus	0.0476	0.0565	2.5759	0.0439	0.0490	53.4011	0.0426	0.0444	573.4251
30%	MSEYInSample	0.0509	0.0581	0.0519	0.0437	0.0467	0.0480	0.0436	0.0444	0.0446
30%	MSEYTesting	0.0514	0.0589	3.7986	0.0459	0.0490	50.9833	0.0447	0.0457	85.4786
30%	MSEYCensus	0.0513	0.0588	3.0493	0.0455	0.0486	40.7962	0.0445	0.0455	68.3918
100%	K (Full Census)	8.6	2.22	54.14	10.46	2.12	101.2	10.82	2.06	157.12
100%	Time (Full Census)	4.5916	0.4667	1.1398	15.9820	0.6556	7.6636	32.2248	0.7935	22.9329

Table H1. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=2, sigma = 0.2

d = 2, sigma = 0.3		nFull = 100			nFull = 200			nFull = 300		
nLearn/nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0051	0.0061	0.0057	0.0027	0.0030	0.0033	0.0019	0.0029	0.0024
100%	MSEFTesting	0.0056	0.0067	308.9863	0.0028	0.0031	102.7481	0.0019	0.0029	198.8724
100%	MSEFCensus	0.0051	0.0061	0.0057	0.0027	0.0030	0.0033	0.0019	0.0029	0.0024
80%	MSEFInSample	0.0075	0.0122	0.0084	0.0048	0.0111	0.0057	0.0035	0.0150	0.0042
80%	MSEFTesting	0.0084	0.0144	10.9325	0.0052	0.0139	217.2453	0.0035	0.0150	116.0226
80%	MSEFCensus	0.0079	0.0133	5.4704	0.0050	0.0125	108.6255	0.0035	0.0150	58.0134
50%	MSEFInSample	0.0118	0.0264	0.0141	0.0067	0.0349	0.0077	0.0048	0.0153	0.0055
50%	MSEFTesting	0.0134	0.0305	10.6080	0.0069	0.0353	32.6064	0.0052	0.0158	186.1256
50%	MSEFCensus	0.0129	0.0293	7.4298	0.0069	0.0352	22.8268	0.0051	0.0156	130.2895
30%	MSEFInSample	0.0134	0.0215	0.0176	0.0100	0.1021	0.0116	0.0067	0.0410	0.0079
30%	MSEFTesting	0.0155	0.0273	3.4131	0.0110	0.1009	36.0041	0.0071	0.0442	51.8847
30%	MSEFCensus	0.0151	0.0261	2.7340	0.0108	0.1012	28.8056	0.0070	0.0436	41.5094
100%	MSEYInSample	0.0936	0.0960	0.0946	0.0938	0.0947	0.0941	0.0919	0.0925	0.0927
100%	MSEYTesting	0.0957	0.0967	309.1160	0.0930	0.0932	102.8571	0.0924	0.0934	199.0073
100%	MSEYCensus	0.0936	0.0960	0.0946	0.0938	0.0947	0.0941	0.0919	0.0925	0.0927
80%	MSEYInSample	0.1025	0.1018	0.0998	0.0937	0.1014	0.0955	0.0937	0.1023	0.0956
80%	MSEYTesting	0.0987	0.1045	11.0346	0.0953	0.1038	217.3697	0.0937	0.1050	116.1571
80%	MSEYCensus	0.1006	0.1032	5.5672	0.0945	0.1026	108.7326	0.0937	0.1036	58.1263
50%	MSEYInSample	0.1043	0.1174	0.1068	0.0982	0.1232	0.0943	0.0931	0.1049	0.0981
50%	MSEYTesting	0.1039	0.1209	10.7094	0.0976	0.1259	32.7146	0.0957	0.1063	186.2580
50%	MSEYCensus	0.1040	0.1199	7.5287	0.0978	0.1251	22.9285	0.0949	0.1058	130.4100
30%	MSEYInSample	0.1170	0.1185	0.1057	0.0966	0.1957	0.1048	0.0972	0.1292	0.0958
30%	MSEYTesting	0.1060	0.1177	3.5072	0.1018	0.1915	36.1073	0.0978	0.1348	51.9937
30%	MSEYCensus	0.1082	0.1179	2.8269	0.1008	0.1924	28.9068	0.0977	0.1337	41.6141
100%	K (Full Census)	8.84	2.16	49.78	9.84	2.12	94.96	10.7	2.26	143.94
100%	Time (Full Census)	4.6754	0.4517	1.1370	15.0989	0.6220	7.6569	30.8449	0.7676	22.5022

Table H2. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=2, sigma = 0.3

d = 2, sigma = 0.4		nFull = 100			nFull = 200			nFull = 300		
nLearn/nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0076	0.0099	0.0087	0.0041	0.0059	0.0048	0.0027	0.1022	0.0033
100%	MSEFTesting	0.0081	0.0105	281.3251	0.0043	0.0064	198.6378	0.0027	0.1093	89.5596
100%	MSEFCensus	0.0076	0.0099	0.0087	0.0041	0.0059	0.0048	0.0027	0.1022	0.0033
80%	MSEFInSample	0.0124	0.0408	0.0136	0.0062	0.1578	0.0073	0.0054	0.0112	0.0063
80%	MSEFTesting	0.0125	0.0398	10.2707	0.0064	0.1723	118.2487	0.0055	0.0115	198.9407
80%	MSEFCensus	0.0125	0.0403	5.1422	0.0063	0.1650	59.1280	0.0054	0.0113	99.4735
50%	MSEFInSample	0.0177	0.1723	0.0200	0.0117	0.0647	0.0129	0.0074	0.0720	0.0085
50%	MSEFTesting	0.0193	0.2279	3.2368	0.0127	0.0584	147.9070	0.0080	0.0729	720.6396
50%	MSEFCensus	0.0188	0.2112	2.2717	0.0124	0.0603	103.5388	0.0078	0.0726	504.4503
30%	MSEFInSample	0.0232	0.0693	0.0265	0.0129	0.0381	0.0146	0.0105	0.0176	0.0118
30%	MSEFTesting	0.0262	0.0731	6.4528	0.0134	0.0435	15.3265	0.0114	0.0193	50.9989
30%	MSEFCensus	0.0256	0.0723	5.1675	0.0133	0.0424	12.2641	0.0112	0.0190	40.8015
100%	MSEYInSample	0.1648	0.1725	0.1669	0.1683	0.1672	0.1681	0.1633	0.2634	0.1660
100%	MSEYTesting	0.1681	0.1705	281.5458	0.1646	0.1666	198.8305	0.1636	0.2701	89.7579
100%	MSEYCensus	0.1648	0.1725	0.1669	0.1683	0.1672	0.1681	0.1633	0.2634	0.1660
80%	MSEYInSample	0.1816	0.2004	0.1754	0.1640	0.3209	0.1647	0.1654	0.1644	0.1697
80%	MSEYTesting	0.1731	0.2001	10.4477	0.1664	0.3323	118.4455	0.1656	0.1715	199.1661
80%	MSEYCensus	0.1774	0.2003	5.3116	0.1652	0.3266	59.3051	0.1655	0.1679	99.6679
50%	MSEYInSample	0.1890	0.3318	0.1760	0.1736	0.2192	0.1708	0.1646	0.2319	0.1741
50%	MSEYTesting	0.1801	0.3886	3.4051	0.1739	0.2193	148.1010	0.1690	0.2339	720.8490
50%	MSEYCensus	0.1828	0.3716	2.4364	0.1738	0.2193	103.7219	0.1677	0.2333	504.6465
30%	MSEYInSample	0.2091	0.2434	0.1826	0.1702	0.1975	0.1816	0.1739	0.1718	0.1687
30%	MSEYTesting	0.1870	0.2339	6.6162	0.1748	0.2048	15.4998	0.1725	0.1804	51.1890
30%	MSEYCensus	0.1914	0.2358	5.3295	0.1739	0.2034	12.4362	0.1728	0.1786	40.9849
100%	K (Full Census)	8.34	2.18	46.24	10.02	2.32	93.9	10.66	2.12	137.24
100%	Time (Full Census)	4.5859	0.4570	1.1464	15.6152	0.6628	7.4554	30.7520	0.7735	23.0779

Table H3. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=2, sigma = 0.4

d = 3, sigma = 0.2		nFull = 100			nFull = 200			nFull = 300		
nLearn/nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0031	0.0173	0.0048	0.0019	0.0026	0.0031	0.0014	0.0042	0.0024
100%	MSEFTesting	0.0034	0.0205	591.2154	0.0020	0.0028	1240.05	0.0014	0.0045	1174.25
100%	MSEFCensus	0.0031	0.0173	0.0048	0.0019	0.0026	0.0031	0.0014	0.0042	0.0024
80%	MSEFInSample	0.0047	0.0099	0.0074	0.0033	0.0046	0.0050	0.0025	0.0038	0.0040
80%	MSEFTesting	0.0053	0.0114	53.6331	0.0035	0.0051	357.01	0.0027	0.0041	907.15
80%	MSEFCensus	0.0050	0.0106	26.8203	0.0034	0.0048	178.51	0.0026	0.0039	453.57
50%	MSEFInSample	0.0075	0.0088	0.0121	0.0048	0.0083	0.0070	0.0036	0.0083	0.0055
50%	MSEFTesting	0.0098	0.0116	4.6067	0.0055	0.0094	142.51	0.0039	0.0090	251.86
50%	MSEFCensus	0.0091	0.0107	3.2283	0.0053	0.0091	99.76	0.0038	0.0087	176.30
30%	MSEFInSample	0.0097	0.0106	0.0136	0.0060	0.0115	0.0089	0.0049	0.0090	0.0072
30%	MSEFTesting	0.0142	0.0154	21.3351	0.0073	0.0127	11.92	0.0054	0.0099	59.93
30%	MSEFCensus	0.0133	0.0145	17.0708	0.0070	0.0125	9.5384	0.0053	0.0097	47.9491
100%	MSEYInSample	0.0416	0.0574	0.0447	0.0415	0.0417	0.0433	0.0410	0.0433	0.0419
100%	MSEYTesting	0.0429	0.0599	591.2827	0.0416	0.0423	1240.13	0.0413	0.0444	1174.34
100%	MSEYCensus	0.0416	0.0574	0.0447	0.0415	0.0417	0.0433	0.0410	0.0433	0.0419
80%	MSEYInSample	0.0463	0.0496	0.0472	0.0416	0.0448	0.0446	0.0418	0.0428	0.0431
80%	MSEYTesting	0.0449	0.0510	53.6810	0.0431	0.0446	357.08	0.0422	0.0436	907.21
80%	MSEYCensus	0.0456	0.0503	26.8641	0.0423	0.0447	178.56	0.0420	0.0432	453.62
50%	MSEYInSample	0.0448	0.0496	0.0538	0.0437	0.0488	0.0477	0.0437	0.0495	0.0467
50%	MSEYTesting	0.0494	0.0512	4.6504	0.0452	0.0491	142.58	0.0438	0.0488	251.93
50%	MSEYCensus	0.0480	0.0507	3.2714	0.0448	0.0490	99.82	0.0437	0.0490	176.36
30%	MSEYInSample	0.0512	0.0491	0.0528	0.0461	0.0507	0.0505	0.0441	0.0499	0.0478
30%	MSEYTesting	0.0539	0.0551	21.3806	0.0469	0.0523	11.9685	0.0451	0.0496	59.9883
30%	MSEYCensus	0.0534	0.0539	17.1150	0.0467	0.0520	9.5849	0.0449	0.0496	48.0002
100%	K (Full Census)	8.46	2.42	60.88	11.46	2.64	125.88	12.42	2.46	193.14
100%	Time (Full Census)	5.0010	0.5231	1.2491	23.2401	0.9280	8.6596	49.3375	1.1707	28.3374

Table H4. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=3, sigma = 0.2

d = 3, sigma = 0.3		nFull = 100			nFull = 200			nFull = 300		
nLearn/nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0052	0.0699	0.0078	0.0034	0.0094	0.0053	0.0026	0.0053	0.0044
100%	MSEFTesting	0.0057	0.0753	782.22	0.0035	0.0095	717.85	0.0026	0.0055	1235.09
100%	MSEFCensus	0.0052	0.0699	0.0078	0.0034	0.0094	0.0053	0.0026	0.0053	0.0044
80%	MSEFInSample	0.0095	0.0429	0.0144	0.0058	0.0303	0.0088	0.0041	0.1537	0.0064
80%	MSEFTesting	0.0104	0.0491	119.14	0.0064	0.0312	768.10	0.0043	0.1663	926.53
80%	MSEFCensus	0.0099	0.0460	59.5793	0.0061	0.0307	384.05	0.0042	0.1600	463.26
50%	MSEFInSample	0.0145	0.0218	0.0209	0.0090	0.0835	0.0129	0.0063	0.0146	0.0091
50%	MSEFTesting	0.0191	0.0323	56.0073	0.0096	0.0964	70.1889	0.0066	0.0165	271.27
50%	MSEFCensus	0.0178	0.0291	39.2114	0.0094	0.0925	49.1361	0.0065	0.0160	189.89
30%	MSEFInSample	0.0192	0.0240	0.0244	0.0124	0.0195	0.0176	0.0092	0.0619	0.0132
30%	MSEFTesting	0.0246	0.0365	13.3936	0.0142	0.0253	12.1662	0.0114	0.0651	163.46
30%	MSEFCensus	0.0235	0.0340	10.7197	0.0138	0.0242	9.7365	0.0110	0.0645	130.77
100%	MSEYInSample	0.0927	0.1595	0.0952	0.0920	0.0980	0.0957	0.0921	0.0941	0.0933
100%	MSEYTesting	0.0947	0.1642	782.36	0.0926	0.0985	717.97	0.0923	0.0952	1235.25
100%	MSEYCensus	0.0927	0.1595	0.0952	0.0920	0.0980	0.0957	0.0921	0.0941	0.0933
80%	MSEYInSample	0.0999	0.1336	0.1004	0.0934	0.1195	0.0964	0.0923	0.2408	0.0939
80%	MSEYTesting	0.0994	0.1381	119.24	0.0954	0.1200	768.22	0.0933	0.2553	926.66
80%	MSEYCensus	0.0996	0.1359	59.6733	0.0944	0.1198	384.15	0.0928	0.2480	463.38
50%	MSEYInSample	0.0985	0.1061	0.1168	0.0980	0.1770	0.1024	0.0974	0.1085	0.0994
50%	MSEYTesting	0.1082	0.1211	56.1061	0.0990	0.1857	70.3067	0.0962	0.1062	271.40
50%	MSEYCensus	0.1053	0.1166	39.3093	0.0987	0.1831	49.2454	0.0966	0.1069	190.01
30%	MSEYInSample	0.1180	0.1179	0.1075	0.1065	0.1062	0.1086	0.0983	0.1542	0.1028
30%	MSEYTesting	0.1137	0.1255	13.4911	0.1031	0.1144	12.2663	0.1009	0.1546	163.58
30%	MSEYCensus	0.1146	0.1240	10.8144	0.1038	0.1128	9.8348	0.1004	0.1545	130.89
100%	K (Full Census)	8.32	2.22	56.74	11.12	2.54	122.6	12.16	2.54	184.08
100%	Time (Full Census)	5.0325	0.5227	1.2846	23.2854	0.9178	8.6964	49.1229	1.1300	27.6230

Table H5. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=3, sigma = 0.3

d = 3, sigma = 0.4		nFull = 100			nFull = 200			nFull = 300		
nLearn /nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0096	0.3470	0.0126	0.0051	0.043	0.0078	0.0039	0.0836	0.0061
100%	MSEFTesting	0.0104	0.3180	940.716	0.0052	0.042	1025.041	0.0040	0.0848	1404.569
100%	MSEFCensus	0.0096	0.3470	0.0126	0.0051	0.043	0.0078	0.0039	0.0836	0.0061
80%	MSEFInSample	0.0154	0.0340	0.0195	0.0089	0.024	0.0138	0.0056	0.0240	0.0086
80%	MSEFTesting	0.0181	0.0397	75.3542	0.0092	0.025	669.2816	0.0061	0.0267	1050.675
80%	MSEFCensus	0.0168	0.0369	37.6869	0.0090	0.025	334.6477	0.0058	0.0253	525.3422
50%	MSEFInSample	0.0204	0.0294	0.0280	0.0125	0.032	0.0171	0.0105	0.0478	0.0142
50%	MSEFTesting	0.0261	0.0372	12.2046	0.0136	0.033	77.9379	0.0118	0.0544	287.5137
50%	MSEFCensus	0.0243	0.0349	8.5516	0.0132	0.032	54.5617	0.0114	0.0524	201.2638
30%	MSEFInSample	0.0389	0.0442	0.0405	0.0183	0.035	0.0226	0.0131	0.0948	0.0168
30%	MSEFTesting	0.0483	0.0601	0.6002	0.0197	0.039	30.7514	0.0144	0.0928	100.0441
30%	MSEFCensus	0.0464	0.0570	0.4882	0.0194	0.038	24.6056	0.0142	0.0932	80.0387
100%	MSEYInSample	0.1653	0.5069	0.1705	0.1639	0.202	0.1678	0.1626	0.2403	0.1637
100%	MSEYTesting	0.1687	0.4762	940.937	0.1635	0.201	1025.241	0.1634	0.2442	1404.848
100%	MSEYCensus	0.1653	0.5069	0.1705	0.1639	0.202	0.1678	0.1626	0.2403	0.1637
80%	MSEYInSample	0.1772	0.1952	0.1713	0.1656	0.181	0.1710	0.1626	0.1801	0.1636
80%	MSEYTesting	0.1765	0.1979	75.5363	0.1675	0.183	669.5001	0.1643	0.1847	1050.904
80%	MSEYCensus	0.1768	0.1965	37.8538	0.1666	0.182	334.8355	0.1634	0.1824	525.5342
50%	MSEYInSample	0.1583	0.1813	0.1859	0.1719	0.195	0.1742	0.1727	0.2134	0.1768
50%	MSEYTesting	0.1845	0.1956	12.3735	0.1724	0.191	78.1271	0.1713	0.2140	287.7428
50%	MSEYCensus	0.1766	0.1913	8.7172	0.1723	0.192	54.7413	0.1717	0.2138	201.4730
30%	MSEYInSample	0.2056	0.2143	0.1961	0.1845	0.194	0.1830	0.1726	0.2593	0.1743
30%	MSEYTesting	0.2070	0.2189	0.7630	0.1779	0.198	30.9247	0.1732	0.2513	100.2345
30%	MSEYCensus	0.2067	0.2180	0.6496	0.1792	0.197	24.7763	0.1730	0.2529	80.2225
100%	K (Full Census)	7.84	2.38	54.3	11.3	2.56	115.12	12.38	2.82	178.82
100%	Time (Full Census)	4.9215	0.5182	1.2876	24.9839	0.916	8.7400	49.3384	1.1136	28.9099

Table H6. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=3, sigma = 0.4

d = 4, sigma = 0.2		nFull = 100			nFull = 200			nFull = 300		
nLearn/ nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0038	0.0082	0.0072	0.0023	0.0031	0.0048	0.0017	0.0022	0.0040
100%	MSEFTesting	0.0041	0.0084	366.5847	0.0023	0.0034	1201.9533	0.0018	0.0023	2096.69
100%	MSEFCensus	0.0038	0.0082	0.0072	0.0023	0.0031	0.0048	0.0017	0.0022	0.0040
80%	MSEFInSample	0.0056	0.0101	0.0105	0.0036	0.0049	0.0071	0.0026	0.0055	0.0054
80%	MSEFTesting	0.0067	0.0120	13.2270	0.0038	0.0054	569.4433	0.0028	0.0063	1140.86
80%	MSEFCensus	0.0061	0.0110	6.6187	0.0037	0.0052	284.7252	0.0027	0.0059	570.4375
50%	MSEFInSample	0.0082	0.0109	0.0154	0.0058	0.0209	0.0105	0.0039	0.0113	0.0083
50%	MSEFTesting	0.0094	0.0157	10.0245	0.0069	0.0237	70.0020	0.0043	0.0107	268.51
50%	MSEFCensus	0.0090	0.0143	7.0218	0.0066	0.0229	49.0046	0.0042	0.0109	187.96
30%	MSEFInSample	0.0115	0.0117	0.0180	0.0073	0.0107	0.0130	0.0057	0.0073	0.0109
30%	MSEFTesting	0.0170	0.0175	0.6706	0.0094	0.0150	3.4263	0.0062	0.0088	43.73
30%	MSEFCensus	0.0159	0.0164	0.5401	0.0089	0.0141	2.7436	0.0061	0.0085	34.9876
100%	MSEYInSample	0.0443	0.0483	0.0471	0.0420	0.0428	0.0449	0.0415	0.0420	0.0441
100%	MSEYTesting	0.0438	0.0481	366.6145	0.0421	0.0432	1202.0056	0.0414	0.0419	2096.74
100%	MSEYCensus	0.0443	0.0483	0.0471	0.0420	0.0428	0.0449	0.0415	0.0420	0.0441
80%	MSEYInSample	0.0444	0.0496	0.0500	0.0436	0.0449	0.0467	0.0429	0.0454	0.0452
80%	MSEYTesting	0.0463	0.0517	13.2687	0.0435	0.0452	569.4794	0.0427	0.0462	1140.94
80%	MSEYCensus	0.0453	0.0506	6.6593	0.0436	0.0450	284.7631	0.0428	0.0458	570.49
50%	MSEYInSample	0.0472	0.0483	0.0510	0.0447	0.0602	0.0504	0.0441	0.0509	0.0477
50%	MSEYTesting	0.0490	0.0554	10.0640	0.0464	0.0633	70.0421	0.0440	0.0505	268.54
50%	MSEYCensus	0.0485	0.0533	7.0601	0.0459	0.0624	49.0446	0.0440	0.0506	187.99
30%	MSEYInSample	0.0544	0.0485	0.0557	0.0456	0.0489	0.0549	0.0446	0.0459	0.0510
30%	MSEYTesting	0.0567	0.0572	0.7107	0.0491	0.0547	3.4666	0.0457	0.0483	43.77
30%	MSEYCensus	0.0562	0.0555	0.5797	0.0484	0.0536	2.7843	0.0455	0.0478	35.02
100%	K (Full Census)	7.04	2.02	63.02	11.7	2.16	132.46	12.92	2.08	192.36
100%	Time (Full Census)	4.0210	0.4488	1.3800	35.6645	1.2499	9.3455	78.7762	1.6994	31.3752

Table H7. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=4, sigma = 0.2

d = 4, sigma = 0.3		nFull = 100			nFull = 200			nFull = 300		
nLearn/ nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0068	0.0138	0.0126	0.0042	0.0267	0.0084	0.0028	0.0042	0.0061
100%	MSEFTesting	0.0074	0.0157	330.2075	0.0043	0.0294	956.4055	0.0028	0.0043	2061.1434
100%	MSEFCensus	0.0068	0.0138	0.0126	0.0042	0.0267	0.0084	0.0028	0.0042	0.0061
80%	MSEFInSample	0.0113	0.0198	0.0199	0.0063	0.1558	0.0127	0.0051	0.0505	0.0098
80%	MSEFTesting	0.0123	0.0227	6.9686	0.0061	0.1795	231.3875	0.0054	0.0542	846.5812
80%	MSEFCensus	0.0118	0.0213	3.4943	0.0062	0.1677	115.7001	0.0053	0.0523	423.2955
50%	MSEFInSample	0.0156	0.0245	0.0255	0.0096	0.0188	0.0177	0.0066	0.0188	0.0122
50%	MSEFTesting	0.0196	0.0330	1.7114	0.0110	0.0222	126.0220	0.0070	0.0193	256.3481
50%	MSEFCensus	0.0184	0.0305	1.2056	0.0106	0.0212	88.2207	0.0069	0.0192	179.4474
30%	MSEFInSample	0.0245	0.0259	0.0350	0.0139	0.0176	0.0216	0.0100	0.0517	0.0178
30%	MSEFTesting	0.0364	0.0400	5.9935	0.0159	0.0229	1.7440	0.0110	0.0516	54.2180
30%	MSEFCensus	0.0340	0.0372	4.8018	0.0155	0.0218	1.3996	0.0108	0.0517	43.3780
100%	MSEYInSample	0.0974	0.1035	0.1023	0.0932	0.1164	0.0985	0.0921	0.0937	0.0959
100%	MSEYTesting	0.0968	0.1052	330.2890	0.0940	0.1190	956.5211	0.0920	0.0936	2061.2556
100%	MSEYCensus	0.0974	0.1035	0.1023	0.0932	0.1164	0.0985	0.0921	0.0937	0.0959
80%	MSEYInSample	0.0992	0.1081	0.1101	0.0964	0.2460	0.1022	0.0952	0.1406	0.0997
80%	MSEYTesting	0.1015	0.1120	7.0599	0.0955	0.2691	231.4696	0.0951	0.1440	846.7029
80%	MSEYCensus	0.1003	0.1101	3.5850	0.0959	0.2575	115.7859	0.0951	0.1423	423.4013
50%	MSEYInSample	0.1033	0.1076	0.1059	0.0977	0.1058	0.1086	0.0966	0.1075	0.1017
50%	MSEYTesting	0.1088	0.1223	1.8006	0.0999	0.1112	126.1123	0.0963	0.1088	256.4295
50%	MSEYCensus	0.1071	0.1178	1.2922	0.0993	0.1095	88.3111	0.0964	0.1084	179.5311
30%	MSEYInSample	0.1213	0.1096	0.1187	0.0994	0.1023	0.1145	0.0971	0.1408	0.1064
30%	MSEYTesting	0.1257	0.1293	6.0830	0.1053	0.1124	1.8337	0.0999	0.1407	54.3085
30%	MSEYCensus	0.1248	0.1254	4.8902	0.1041	0.1104	1.4899	0.0993	0.1407	43.4681
100%	K (Full Census)	6.94	2.26	59.42	11.94	2.22	122.16	12.4	2.18	192.14
100%	Time (Full Census)	3.8932	0.4635	1.3946	36.0235	1.2504	9.5493	73.5000	1.6338	31.5927

Table H8. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=4, sigma = 0.3.

d = 4, sigma = 0.4		nFull = 100			nFull = 200			nFull = 300		
nLearn/ nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0091	0.0333	0.0162	0.0061	0.0109	0.0120	0.0044	0.0151	0.0088
100%	MSEFTesting	0.0104	0.0336	269.5879	0.0063	0.0123	1809.4380	0.0044	0.0149	2638.1731
100%	MSEFCensus	0.0091	0.0333	0.0162	0.0061	0.0109	0.0120	0.0044	0.0151	0.0088
80%	MSEFInSample	0.0183	0.0637	0.0296	0.0110	0.0194	0.0177	0.0078	0.0270	0.0141
80%	MSEFTesting	0.0200	0.0755	14.6548	0.0120	0.0206	378.6360	0.0084	0.0279	671.0586
80%	MSEFCensus	0.0191	0.0696	7.3422	0.0115	0.0200	189.3269	0.0081	0.0274	335.5364
50%	MSEFInSample	0.0300	0.0387	0.0424	0.0160	0.0289	0.0272	0.0109	0.0369	0.0189
50%	MSEFTesting	0.0371	0.0489	3.0819	0.0170	0.0320	26.7000	0.0119	0.0383	403.4055
50%	MSEFCensus	0.0349	0.0459	2.1701	0.0167	0.0311	18.6981	0.0116	0.0379	282.3895
30%	MSEFInSample	0.0411	0.0441	0.0526	0.0225	0.0361	0.0367	0.0159	0.0257	0.0274
30%	MSEFTesting	0.0562	0.0620	1.6564	0.0258	0.0450	102.2316	0.0166	0.0291	32.9122
30%	MSEFCensus	0.0532	0.0585	1.3356	0.0251	0.0432	81.7926	0.0164	0.0284	26.3352
100%	MSEYInSample	0.1713	0.1918	0.1746	0.1652	0.1703	0.1732	0.1627	0.1745	0.1685
100%	MSEYTesting	0.1694	0.1928	269.7363	0.1657	0.1716	1809.6336	0.1630	0.1736	2638.4058
100%	MSEYCensus	0.1713	0.1918	0.1746	0.1652	0.1703	0.1732	0.1627	0.1745	0.1685
80%	MSEYInSample	0.1736	0.2194	0.1877	0.1728	0.1786	0.1773	0.1675	0.1868	0.1738
80%	MSEYTesting	0.1786	0.2344	14.8167	0.1710	0.1797	378.7881	0.1679	0.1874	671.2451
80%	MSEYCensus	0.1761	0.2269	7.5022	0.1719	0.1792	189.4827	0.1677	0.1871	335.7094
50%	MSEYInSample	0.1864	0.1869	0.1833	0.1722	0.1825	0.1876	0.1708	0.1946	0.1775
50%	MSEYTesting	0.1958	0.2078	3.2389	0.1751	0.1903	26.8610	0.1705	0.1973	403.5389
50%	MSEYCensus	0.1930	0.2015	2.3222	0.1742	0.1880	18.8590	0.1706	0.1965	282.5305
30%	MSEYInSample	0.2115	0.1926	0.2005	0.1761	0.1884	0.2024	0.1723	0.1813	0.1872
30%	MSEYTesting	0.2149	0.2207	1.8168	0.1847	0.2040	102.3966	0.1747	0.1874	33.0729
30%	MSEYCensus	0.2142	0.2151	1.4936	0.1830	0.2009	81.9577	0.1742	0.1861	26.4957
100%	K (Full Census)	6.94	2.02	56.82	11.8	2.38	122.26	12.22	2.1	185.84
100%	Time (Full Census)	3.8504	0.4765	1.4262	35.9447	1.2765	9.8326	75.0789	1.6454	32.8098

Table H9. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=4, sigma = 0.4

d = 2, sigma = 0.3		nFull = 100			nFull = 200			nFull = 300		
nLearn/ nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0000	0.0003	0.0000	0.0000	0.0003	0.0000	0.0000	0.0003	0.0000
100%	MSEFTesting	0.0000	0.0003	87.91	0.0000	0.0003	76.24	0.0000	0.0003	55.36
100%	MSEFCensus	0.0000	0.0003	0.0000	0.0000	0.0003	0.0000	0.0000	0.0003	0.0000
80%	MSEFInSample	0.0001	0.0002	0.0000	0.0000	0.0003	0.0000	0.0000	0.0003	0.0000
80%	MSEFTesting	0.0001	0.0004	69.49	0.0000	0.0003	138.34	0.0000	0.0003	252.95
80%	MSEFCensus	0.0001	0.0003	34.74	0.0000	0.0003	69.1709	0.0000	0.0003	126.4761
50%	MSEFInSample	0.0001	0.0002	0.0000	0.0000	0.0002	0.0000	0.0000	0.0003	0.0000
50%	MSEFTesting	0.0003	0.0004	34.57	0.0001	0.0004	251.85	0.0000	0.0003	74.69
50%	MSEFCensus	0.0002	0.0004	24.19	0.0001	0.0003	176.29	0.0000	0.0003	52.28
30%	MSEFInSample	0.0002	0.0002	0.0001	0.0001	0.0002	0.0000	0.0000	0.0002	0.0000
30%	MSEFTesting	0.0005	0.0005	1.54	0.0001	0.0004	38.20	0.0001	0.0003	132.17
30%	MSEFCensus	0.0005	0.0005	1.23	0.0001	0.0004	30.56	0.0001	0.0003	105.73
100%	MSEYInSample	0.0001	0.0004	0.0001	0.0001	0.0004	0.0001	0.0001	0.0004	0.0001
100%	MSEYTesting	0.0001	0.0004	87.91	0.0001	0.0004	76.24	0.0001	0.0004	55.36
100%	MSEYCensus	0.0001	0.0004	0.0001	0.0001	0.0004	0.0001	0.0001	0.0004	0.0001
80%	MSEYInSample	0.0002	0.0003	0.0001	0.0001	0.0004	0.0001	0.0001	0.0004	0.0001
80%	MSEYTesting	0.0002	0.0005	69.49	0.0001	0.0004	138.34	0.0001	0.0004	252.95
80%	MSEYCensus	0.0002	0.0004	34.74	0.0001	0.0004	69.17	0.0001	0.0004	126.47
50%	MSEYInSample	0.0002	0.0003	0.0001	0.0001	0.0003	0.0001	0.0001	0.0003	0.0001
50%	MSEYTesting	0.0004	0.0005	34.57	0.0002	0.0005	251.85	0.0001	0.0004	74.69
50%	MSEYCensus	0.0003	0.0005	24.19	0.0002	0.0004	176.30	0.0001	0.0004	52.28
30%	MSEYInSample	0.0003	0.0003	0.0002	0.0002	0.0003	0.0001	0.0001	0.0003	0.0001
30%	MSEYTesting	0.0006	0.0006	1.5435	0.0002	0.0005	38.20	0.0002	0.0004	132.17
30%	MSEYCensus	0.0006	0.0006	1.2348	0.0002	0.0005	30.56	0.0002	0.0004	105.73
100%	K (Full Census)	9.1200	2.0000	89.66	11.1000	2.0000	159.20	12.0000	2.0600	221.32
100%	Time (Full Census)	5.0034	0.4731	1.0831	16.5378	0.6387	7.0558	32.8468	0.7533	21.8691

Table H10. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=2, sigma = 0.01

d = 2, sigma = 0.4		nFull = 100			nFull = 200			nFull = 300		
nLearn/ nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0003	0.0004	0.0004	0.0002	0.0004	0.0002	0.0001	0.0003	0.0002
100%	MSEFTesting	0.0003	0.0005	344.2265	0.0002	0.0004	104.3058	0.0001	0.0004	208.4270
100%	MSEFCensus	0.0003	0.0004	0.0004	0.0002	0.0004	0.0002	0.0001	0.0003	0.0002
80%	MSEFInSample	0.0004	0.0005	0.0005	0.0003	0.0005	0.0004	0.0002	0.0004	0.0002
80%	MSEFTesting	0.0006	0.0007	56.7960	0.0003	0.0005	163.3753	0.0002	0.0004	276.1076
80%	MSEFCensus	0.0005	0.0006	28.3982	0.0003	0.0005	81.6878	0.0002	0.0004	138.0539
50%	MSEFInSample	0.0006	0.0008	0.0008	0.0004	0.0005	0.0005	0.0003	0.0004	0.0003
50%	MSEFTesting	0.0009	0.0012	22.0658	0.0005	0.0007	237.7376	0.0004	0.0005	212.6012
50%	MSEFCensus	0.0008	0.0011	15.4463	0.0005	0.0006	166.4165	0.0003	0.0005	148.8209
30%	MSEFInSample	0.0008	0.0009	0.0010	0.0005	0.0006	0.0006	0.0004	0.0005	0.0005
30%	MSEFTesting	0.0012	0.0014	1.5623	0.0007	0.0007	34.0284	0.0005	0.0006	92.0718
30%	MSEFCensus	0.0011	0.0013	1.2501	0.0006	0.0007	27.2228	0.0005	0.0006	73.6575
100%	MSEYInSample	0.0027	0.0029	0.0028	0.0027	0.0029	0.0027	0.0026	0.0029	0.0027
100%	MSEYTesting	0.0028	0.0030	344.2366	0.0027	0.0029	104.3114	0.0026	0.0029	208.4361
100%	MSEYCensus	0.0027	0.0029	0.0028	0.0027	0.0029	0.0027	0.0026	0.0029	0.0027
80%	MSEYInSample	0.0029	0.0031	0.0030	0.0027	0.0029	0.0028	0.0027	0.0029	0.0027
80%	MSEYTesting	0.0031	0.0032	56.8033	0.0028	0.0030	163.3829	0.0027	0.0029	276.1184
80%	MSEYCensus	0.0030	0.0031	28.4031	0.0028	0.0029	81.6928	0.0027	0.0029	138.0605
50%	MSEYInSample	0.0032	0.0032	0.0033	0.0029	0.0031	0.0030	0.0029	0.0029	0.0028
50%	MSEYTesting	0.0034	0.0037	22.0713	0.0030	0.0032	237.7481	0.0029	0.0030	212.6125
50%	MSEYCensus	0.0033	0.0035	15.4509	0.0030	0.0032	166.4246	0.0029	0.0030	148.8296
30%	MSEYInSample	0.0034	0.0033	0.0036	0.0031	0.0030	0.0031	0.0029	0.0031	0.0029
30%	MSEYTesting	0.0037	0.0039	1.5654	0.0032	0.0032	34.0332	0.0030	0.0031	92.0791
30%	MSEYCensus	0.0036	0.0038	1.2530	0.0032	0.0032	27.2272	0.0030	0.0031	73.6638
100%	K (Full Census)	9.1600	2.0000	75.3400	10.7400	2.0600	133.4800	11.4400	2.0400	192.1800
100%	Time (Full Census)	4.9175	0.4587	1.1222	16.5608	0.6480	7.6042	32.8749	0.7590	23.0120

Table H11. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=2, sigma = 0.05

d = 2, sigma = 0.3		nFull = 100			nFull = 200			nFull = 300		
nLearn/ nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0008	0.0009	0.0010	0.0005	0.0006	0.0007	0.0003	0.0005	0.0004
100%	MSEFTesting	0.0009	0.0010	370.9240	0.0005	0.0007	218.0245	0.0003	0.0005	141.7350
100%	MSEFCensus	0.0008	0.0009	0.0010	0.0005	0.0006	0.0007	0.0003	0.0005	0.0004
80%	MSEFInSample	0.0015	0.0017	0.0018	0.0008	0.0010	0.0011	0.0006	0.0007	0.0007
80%	MSEFTesting	0.0017	0.0020	130.4580	0.0009	0.0011	219.5168	0.0007	0.0008	329.2754
80%	MSEFCensus	0.0016	0.0018	65.2299	0.0009	0.0011	109.7589	0.0006	0.0007	164.6381
50%	MSEFInSample	0.0020	0.0029	0.0027	0.0013	0.0013	0.0015	0.0009	0.0010	0.0011
50%	MSEFTesting	0.0025	0.0039	17.0756	0.0015	0.0014	170.5202	0.0010	0.0011	133.5198
50%	MSEFCensus	0.0024	0.0036	11.9537	0.0014	0.0014	119.3646	0.0010	0.0011	93.4642
30%	MSEFInSample	0.0023	0.0036	0.0032	0.0018	0.0485	0.0022	0.0013	0.0016	0.0016
30%	MSEFTesting	0.0029	0.0049	1266.1997	0.0021	0.0473	74.8704	0.0017	0.0018	195.1175
30%	MSEFCensus	0.0028	0.0047	1012.9604	0.0020	0.0475	59.8968	0.0017	0.0018	156.0943
100%	MSEYInSample	0.0107	0.0108	0.0108	0.0104	0.0108	0.0108	0.0103	0.0105	0.0105
100%	MSEYTesting	0.0109	0.0110	370.9495	0.0105	0.0107	218.0414	0.0104	0.0106	141.7571
100%	MSEYCensus	0.0107	0.0108	0.0108	0.0104	0.0108	0.0108	0.0103	0.0105	0.0105
80%	MSEYInSample	0.0115	0.0116	0.0119	0.0108	0.0109	0.0109	0.0105	0.0104	0.0107
80%	MSEYTesting	0.0117	0.0120	130.4817	0.0109	0.0111	219.5377	0.0107	0.0108	329.3072
80%	MSEYCensus	0.0116	0.0118	65.2468	0.0108	0.0110	109.7743	0.0106	0.0106	164.6590
50%	MSEYInSample	0.0124	0.0128	0.0127	0.0111	0.0114	0.0118	0.0108	0.0109	0.0114
50%	MSEYTesting	0.0126	0.0139	17.0899	0.0115	0.0115	170.5413	0.0111	0.0112	133.5433
50%	MSEYCensus	0.0125	0.0136	11.9667	0.0114	0.0114	119.3825	0.0110	0.0111	93.4837
30%	MSEYInSample	0.0124	0.0139	0.0130	0.0115	0.0589	0.0124	0.0112	0.0117	0.0119
30%	MSEYTesting	0.0130	0.0150	1266.2176	0.0122	0.0574	74.8874	0.0118	0.0119	195.1395
30%	MSEYCensus	0.0128	0.0148	1012.9767	0.0120	0.0577	59.9124	0.0117	0.0118	156.1140
100%	K (Full Census)	8.9200	2.1000	61.9200	10.6600	2.1200	120.6400	11.1800	2.1000	169.9800
100%	Time (Full Census)	4.8844	0.4632	1.1780	16.5917	0.6587	7.6625	33.8377	0.7826	23.3258

Table H12. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=2, sigma = 0.1

d = 3, sigma = 0.2		nFull = 100			nFull = 200			nFull = 300		
nLearn/nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0001	0.0006	0.0001	0.0002	0.0006	0.0000	0.0003	0.0007	0.0000
100%	MSEFTesting	0.0002	0.0008	413.5722	0.0002	0.0007	423.1706	0.0003	0.0007	714.7379
100%	MSEFCensus	0.0001	0.0006	0.0001	0.0002	0.0006	0.0000	0.0003	0.0007	0.0000
80%	MSEFInSample	0.0003	0.0005	0.0001	0.0001	0.0006	0.0001	0.0001	0.0006	0.0000
80%	MSEFTesting	0.0005	0.0009	32.9737	0.0002	0.0008	187.1054	0.0002	0.0007	429.8912
80%	MSEFCensus	0.0004	0.0007	16.4869	0.0002	0.0007	93.5527	0.0001	0.0007	214.9456
50%	MSEFInSample	0.0004	0.0004	0.0001	0.0002	0.0005	0.0001	0.0002	0.0006	0.0001
50%	MSEFTesting	0.0009	0.0010	4.2185	0.0004	0.0008	79.8074	0.0003	0.0008	325.6861
50%	MSEFCensus	0.0007	0.0008	2.9529	0.0004	0.0007	55.8652	0.0002	0.0007	227.9803
30%	MSEFInSample	0.0004	0.0004	0.0001	0.0003	0.0005	0.0001	0.0002	0.0005	0.0001
30%	MSEFTesting	0.0012	0.0012	0.3865	0.0007	0.0009	20.5994	0.0004	0.0008	80.4559
30%	MSEFCensus	0.0011	0.0011	0.3092	0.0006	0.0008	16.4795	0.0004	0.0008	64.3647
100%	MSEYInSample	0.0002	0.0007	0.0001	0.0003	0.0007	0.0001	0.0004	0.0008	0.0001
100%	MSEYTesting	0.0003	0.0009	413.5735	0.0003	0.0008	423.1714	0.0004	0.0008	714.7400
100%	MSEYCensus	0.0002	0.0007	0.0001	0.0003	0.0007	0.0001	0.0004	0.0008	0.0001
80%	MSEYInSample	0.0004	0.0006	0.0002	0.0002	0.0007	0.0001	0.0002	0.0007	0.0001
80%	MSEYTesting	0.0006	0.0010	32.9743	0.0003	0.0009	187.1063	0.0003	0.0008	429.8925
80%	MSEYCensus	0.0005	0.0008	16.4872	0.0003	0.0008	93.5532	0.0002	0.0008	214.9463
50%	MSEYInSample	0.0005	0.0005	0.0002	0.0003	0.0006	0.0002	0.0003	0.0007	0.0002
50%	MSEYTesting	0.0010	0.0011	4.2188	0.0005	0.0009	79.8086	0.0004	0.0009	325.6881
50%	MSEYCensus	0.0008	0.0009	2.9532	0.0005	0.0008	55.8661	0.0003	0.0008	227.9817
30%	MSEYInSample	0.0005	0.0005	0.0002	0.0004	0.0006	0.0002	0.0003	0.0006	0.0002
30%	MSEYTesting	0.0013	0.0013	0.3867	0.0008	0.0010	20.6000	0.0005	0.0009	80.4571
30%	MSEYCensus	0.0012	0.0012	0.3094	0.0007	0.0009	16.4800	0.0005	0.0009	64.3657
100%	K (Full Census)	8.5200	2.0000	94.7200	11.7800	2.1600	185.3600	12.5600	2.1600	272.9800
100%	Time (Full Census)	5.4161	0.5377	1.1190	25.9682	0.8869	7.8553	53.5792	1.1314	26.0566

Table H13. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=3, sigma = 0.01

d = 3, sigma = 0.3		nFull = 100			nFull = 200			nFull = 300		
nLearn/ nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0005	0.0012	0.0006	0.0003	0.0008	0.0004	0.0003	0.0007	0.0003
100%	MSEFTesting	0.0006	0.0014	762.8861	0.0004	0.0008	975.1576	0.0004	0.0008	823.8718
100%	MSEFCensus	0.0005	0.0012	0.0006	0.0003	0.0008	0.0004	0.0003	0.0007	0.0003
80%	MSEFInSample	0.0008	0.0010	0.0009	0.0004	0.0008	0.0006	0.0004	0.0009	0.0005
80%	MSEFTesting	0.0011	0.0013	144.0639	0.0005	0.0010	601.6849	0.0004	0.0010	608.9285
80%	MSEFCensus	0.0009	0.0012	72.0324	0.0005	0.0009	300.8427	0.0004	0.0010	304.4645
50%	MSEFInSample	0.0011	0.0011	0.0013	0.0007	0.0010	0.0009	0.0005	0.0009	0.0007
50%	MSEFTesting	0.0018	0.0018	12.5961	0.0009	0.0013	122.5333	0.0007	0.0011	627.3650
50%	MSEFCensus	0.0016	0.0016	8.8177	0.0008	0.0012	85.7736	0.0006	0.0010	439.1557
30%	MSEFInSample	0.0013	0.0014	0.0015	0.0008	0.0009	0.0010	0.0006	0.0009	0.0008
30%	MSEFTesting	0.0025	0.0027	4.1813	0.0012	0.0014	68.7782	0.0009	0.0013	236.6423
30%	MSEFCensus	0.0023	0.0024	3.3453	0.0011	0.0013	55.0228	0.0008	0.0012	189.3140
100%	MSEYInSample	0.0029	0.0037	0.0031	0.0028	0.0032	0.0029	0.0028	0.0032	0.0028
100%	MSEYTesting	0.0030	0.0038	762.8987	0.0029	0.0033	975.1682	0.0029	0.0033	823.8849
100%	MSEYCensus	0.0029	0.0037	0.0031	0.0028	0.0032	0.0029	0.0028	0.0032	0.0028
80%	MSEYInSample	0.0034	0.0037	0.0034	0.0029	0.0033	0.0030	0.0028	0.0034	0.0030
80%	MSEYTesting	0.0035	0.0038	144.0707	0.0030	0.0035	601.6945	0.0029	0.0035	608.9365
80%	MSEYCensus	0.0034	0.0037	72.0371	0.0030	0.0034	300.8487	0.0029	0.0034	304.4697
50%	MSEYInSample	0.0037	0.0038	0.0038	0.0032	0.0035	0.0034	0.0030	0.0034	0.0032
50%	MSEYTesting	0.0043	0.0043	12.5997	0.0034	0.0038	122.5413	0.0032	0.0035	627.3812
50%	MSEYCensus	0.0041	0.0041	8.8209	0.0033	0.0037	85.7799	0.0031	0.0035	439.1678
30%	MSEYInSample	0.0038	0.0042	0.0039	0.0033	0.0035	0.0034	0.0031	0.0034	0.0034
30%	MSEYTesting	0.0050	0.0052	4.1844	0.0037	0.0039	68.7844	0.0034	0.0038	236.6521
30%	MSEYCensus	0.0048	0.0050	3.3483	0.0036	0.0038	55.0282	0.0033	0.0037	189.3224
100%	K (Full Census)	8.7400	2.4600	80.1600	11.8400	2.3400	154.8200	12.6800	2.2200	229.7600
100%	Time (Full Census)	5.5555	0.5153	1.2451	25.9777	0.9054	8.4133	55.2034	1.1440	27.9051

Table H14. MSE_f, MSE_{lSf}, MSE_{FSf}, MSE_y, MSE_{lSy}, MSE_{FSy}, time and *K* results for d=3, sigma = 0.05

d = 3, sigma = 0.4		nFull = 100			nFull = 200			nFull = 300		
nLearn/nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0012	0.0019	0.0018	0.0008	0.0018	0.0011	0.0006	0.0010	0.0008
100%	MSEFTesting	0.0014	0.0022	978.6187	0.0009	0.0019	1562.1317	0.0006	0.0010	1432.7275
100%	MSEFCensus	0.0012	0.0019	0.0018	0.0008	0.0018	0.0011	0.0006	0.0010	0.0008
80%	MSEFInSample	0.0019	0.0062	0.0027	0.0012	0.0021	0.0018	0.0010	0.0013	0.0014
80%	MSEFTesting	0.0026	0.0068	83.7592	0.0014	0.0023	545.8418	0.0011	0.0015	711.7752
80%	MSEFCensus	0.0023	0.0065	41.8809	0.0013	0.0022	272.9218	0.0010	0.0014	355.8883
50%	MSEFInSample	0.0028	0.0031	0.0040	0.0017	0.0035	0.0025	0.0013	0.0016	0.0019
50%	MSEFTesting	0.0038	0.0043	2.3712	0.0019	0.0038	128.7269	0.0015	0.0019	272.2648
50%	MSEFCensus	0.0035	0.0039	1.6611	0.0019	0.0037	90.1096	0.0015	0.0018	190.5859
30%	MSEFInSample	0.0031	0.0049	0.0047	0.0022	0.0049	0.0034	0.0017	0.0032	0.0025
30%	MSEFTesting	0.0050	0.0076	0.3994	0.0029	0.0058	32.1949	0.0020	0.0037	216.3175
30%	MSEFCensus	0.0047	0.0071	0.3205	0.0028	0.0056	25.7566	0.0019	0.0036	173.0545
100%	MSEYInSample	0.0110	0.0119	0.0118	0.0107	0.0117	0.0111	0.0104	0.0108	0.0107
100%	MSEYTesting	0.0113	0.0121	978.6446	0.0108	0.0118	1562.1564	0.0106	0.0110	1432.7618
100%	MSEYCensus	0.0110	0.0119	0.0118	0.0107	0.0117	0.0111	0.0104	0.0108	0.0107
80%	MSEYInSample	0.0116	0.0160	0.0126	0.0111	0.0120	0.0118	0.0109	0.0112	0.0112
80%	MSEYTesting	0.0125	0.0167	83.7772	0.0113	0.0122	545.8659	0.0109	0.0114	711.7959
80%	MSEYCensus	0.0121	0.0163	41.8949	0.0112	0.0121	272.9388	0.0109	0.0113	355.9035
50%	MSEYInSample	0.0130	0.0138	0.0133	0.0116	0.0130	0.0123	0.0115	0.0119	0.0119
50%	MSEYTesting	0.0137	0.0141	2.3829	0.0119	0.0138	128.7483	0.0115	0.0118	272.2914
50%	MSEYCensus	0.0135	0.0140	1.6720	0.0118	0.0135	90.1275	0.0115	0.0118	190.6075
30%	MSEYInSample	0.0125	0.0161	0.0137	0.0123	0.0155	0.0135	0.0116	0.0128	0.0125
30%	MSEYTesting	0.0149	0.0175	0.4102	0.0128	0.0156	32.2087	0.0119	0.0136	216.3393
30%	MSEYCensus	0.0145	0.0173	0.3309	0.0127	0.0156	25.7697	0.0119	0.0134	173.0739
100%	K (Full Census)	8.4200	2.5800	69.2400	11.6200	2.4400	140.6800	12.7000	2.3800	211.5800
100%	Time (Full Census)	5.3340	0.5241	1.2568	24.1476	0.9520	8.8607	53.7186	1.1511	29.1077

Table H15. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=3, sigma = 0.1

d = 4, sigma = 0.2		nFull = 100			nFull = 200			nFull = 300		
nLearn/nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0003	0.0007	0.0001	0.0003	0.0007	0.0001	0.0004	0.0007	0.0001
100%	MSEFTesting	0.0005	0.0010	161.6089	0.0003	0.0009	365.6785	0.0005	0.0008	584.3493
100%	MSEFCensus	0.0003	0.0007	0.0001	0.0003	0.0007	0.0001	0.0004	0.0007	0.0001
80%	MSEFInSample	0.0004	0.0005	0.0001	0.0003	0.0007	0.0001	0.0003	0.0007	0.0001
80%	MSEFTesting	0.0008	0.0011	8.6689	0.0005	0.0009	204.9950	0.0004	0.0009	263.8065
80%	MSEFCensus	0.0006	0.0008	4.3345	0.0004	0.0008	102.4976	0.0003	0.0008	131.9033
50%	MSEFInSample	0.0005	0.0005	0.0001	0.0004	0.0006	0.0001	0.0003	0.0006	0.0001
50%	MSEFTesting	0.0013	0.0014	1.2656	0.0007	0.0010	25.1762	0.0005	0.0009	91.1381
50%	MSEFCensus	0.0011	0.0011	0.8860	0.0006	0.0009	17.6233	0.0004	0.0008	63.7967
30%	MSEFInSample	0.0008	0.0006	0.0001	0.0005	0.0005	0.0001	0.0004	0.0006	0.0001
30%	MSEFTesting	0.0017	0.0016	0.1625	0.0010	0.0013	1.8949	0.0007	0.0010	22.4333
30%	MSEFCensus	0.0015	0.0014	0.1301	0.0009	0.0011	1.5159	0.0007	0.0009	17.9467
100%	MSEYInSample	0.0004	0.0008	0.0002	0.0004	0.0008	0.0002	0.0005	0.0008	0.0002
100%	MSEYTesting	0.0006	0.0011	161.6089	0.0004	0.0010	365.6792	0.0005	0.0009	584.3495
100%	MSEYCensus	0.0004	0.0008	0.0002	0.0004	0.0008	0.0002	0.0005	0.0008	0.0002
80%	MSEYInSample	0.0005	0.0006	0.0002	0.0004	0.0008	0.0002	0.0004	0.0008	0.0002
80%	MSEYTesting	0.0009	0.0012	8.6690	0.0006	0.0010	204.9950	0.0005	0.0010	263.8074
80%	MSEYCensus	0.0007	0.0009	4.3346	0.0005	0.0009	102.4976	0.0004	0.0009	131.9038
50%	MSEYInSample	0.0006	0.0006	0.0002	0.0005	0.0007	0.0002	0.0004	0.0007	0.0002
50%	MSEYTesting	0.0014	0.0014	1.2657	0.0008	0.0011	25.1763	0.0006	0.0010	91.1379
50%	MSEYCensus	0.0012	0.0012	0.8860	0.0007	0.0010	17.6235	0.0005	0.0009	63.7966
30%	MSEYInSample	0.0009	0.0007	0.0002	0.0006	0.0006	0.0002	0.0005	0.0007	0.0002
30%	MSEYTesting	0.0018	0.0017	0.1627	0.0011	0.0014	1.8950	0.0008	0.0011	22.4334
30%	MSEYCensus	0.0016	0.0015	0.1302	0.0010	0.0012	1.5160	0.0008	0.0010	17.9468
100%	K (Full Census)	7.2600	2.0000	96.1600	11.8800	2.0000	190.2800	12.5000	2.0600	276.8200
100%	Time (Full Census)	4.3435	0.4711	1.1840	37.5370	1.2684	8.2187	75.3199	1.6647	26.7982

Table H16. MSE_f, MSE_{lSf}, MSE_{FSf}, MSE_y, MSE_{lSy}, MSE_{FSy}, time and *K* results for d=4, sigma = 0.01

d = 4, sigma = 0.3		nFull = 100			nFull = 200			nFull = 300		
nLearn/nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0007	0.0010	0.0009	0.0005	0.0009	0.0007	0.0005	0.0008	0.0005
100%	MSEFTesting	0.0009	0.0012	440.3271	0.0006	0.0010	1105.7870	0.0006	0.0009	630.5477
100%	MSEFCensus	0.0007	0.0010	0.0009	0.0005	0.0009	0.0007	0.0005	0.0008	0.0005
80%	MSEFInSample	0.0009	0.0024	0.0013	0.0007	0.0010	0.0009	0.0005	0.0009	0.0007
80%	MSEFTesting	0.0014	0.0029	20.5826	0.0009	0.0012	701.4585	0.0007	0.0011	1027.9976
80%	MSEFCensus	0.0011	0.0026	10.2919	0.0008	0.0011	350.7297	0.0006	0.0010	513.9992
50%	MSEFInSample	0.0012	0.0014	0.0016	0.0009	0.0011	0.0012	0.0007	0.0010	0.0009
50%	MSEFTesting	0.0021	0.0023	2.8493	0.0012	0.0015	43.0414	0.0010	0.0013	281.3078
50%	MSEFCensus	0.0018	0.0020	1.9949	0.0011	0.0013	30.1293	0.0009	0.0012	196.9158
30%	MSEFInSample	0.0015	0.0014	0.0020	0.0011	0.0014	0.0015	0.0008	0.0011	0.0012
30%	MSEFTesting	0.0025	0.0026	0.2964	0.0017	0.0020	3.9116	0.0013	0.0016	45.9066
30%	MSEFCensus	0.0023	0.0024	0.2375	0.0016	0.0019	3.1296	0.0012	0.0015	36.7255
100%	MSEYInSample	0.0032	0.0034	0.0034	0.0030	0.0034	0.0032	0.0030	0.0033	0.0030
100%	MSEYTesting	0.0034	0.0037	440.3277	0.0031	0.0035	1105.7959	0.0030	0.0034	630.5537
100%	MSEYCensus	0.0032	0.0034	0.0034	0.0030	0.0034	0.0032	0.0030	0.0033	0.0030
80%	MSEYInSample	0.0034	0.0048	0.0038	0.0032	0.0034	0.0034	0.0030	0.0034	0.0032
80%	MSEYTesting	0.0038	0.0053	20.5851	0.0034	0.0037	701.4593	0.0032	0.0036	1028.0035
80%	MSEYCensus	0.0036	0.0051	10.2944	0.0033	0.0036	350.7313	0.0031	0.0035	514.0033
50%	MSEYInSample	0.0037	0.0038	0.0038	0.0033	0.0035	0.0037	0.0032	0.0034	0.0033
50%	MSEYTesting	0.0046	0.0047	2.8517	0.0036	0.0039	43.0442	0.0035	0.0038	281.3081
50%	MSEYCensus	0.0043	0.0045	1.9974	0.0035	0.0038	30.1320	0.0034	0.0037	196.9167
30%	MSEYInSample	0.0042	0.0038	0.0043	0.0035	0.0038	0.0041	0.0033	0.0035	0.0036
30%	MSEYTesting	0.0050	0.0051	0.2989	0.0042	0.0045	3.9139	0.0038	0.0041	45.9092
30%	MSEYCensus	0.0049	0.0048	0.2400	0.0040	0.0043	3.1319	0.0037	0.0040	36.7281
100%	K (Full Census)	7.0800	2.0000	76.1800	11.8400	2.0600	154.8800	12.7800	2.0600	229.4800
100%	Time (Full Census)	4.1409	0.4665	1.3064	35.8179	1.2787	9.3666	77.7415	1.6637	29.6764

Table H17. MSE_f, MSE_{ISf}, MSE_{FSf}, MSE_y, MSE_{ISy}, MSE_{FSy}, time and *K* results for d=4, sigma = 0.05

d = 4, sigma = 0.4		nFull = 100			nFull = 200			nFull = 300		
nLearn/nFull	Metric	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS	CAPNLS	CAP	CNLS
100%	MSEFInSample	0.0015	0.0018	0.0025	0.0010	0.0013	0.0017	0.0008	0.0012	0.0014
100%	MSEFTesting	0.0017	0.0022	322.4462	0.0010	0.0015	1624.4689	0.0009	0.0013	2548.5803
100%	MSEFCensus	0.0015	0.0018	0.0025	0.0010	0.0013	0.0017	0.0008	0.0012	0.0014
80%	MSEFInSample	0.0025	0.0030	0.0043	0.0015	0.0017	0.0025	0.0012	0.0015	0.0019
80%	MSEFTesting	0.0031	0.0043	20.1828	0.0017	0.0020	398.8514	0.0013	0.0017	1202.0194
80%	MSEFCensus	0.0028	0.0036	10.0935	0.0016	0.0019	199.4270	0.0012	0.0016	601.0107
50%	MSEFInSample	0.0031	0.0060	0.0055	0.0021	0.0025	0.0035	0.0017	0.0030	0.0029
50%	MSEFTesting	0.0044	0.0073	0.4835	0.0026	0.0030	49.4704	0.0019	0.0033	372.3199
50%	MSEFCensus	0.0040	0.0069	0.3401	0.0024	0.0028	34.6304	0.0018	0.0032	260.6248
30%	MSEFInSample	0.0033	0.0034	0.0060	0.0023	0.0028	0.0041	0.0021	0.0022	0.0035
30%	MSEFTesting	0.0048	0.0047	0.5589	0.0030	0.0037	11.4710	0.0024	0.0027	96.0842
30%	MSEFCensus	0.0045	0.0045	0.4483	0.0029	0.0035	9.1776	0.0024	0.0026	76.8680
100%	MSEYInSample	0.0115	0.0117	0.0124	0.0109	0.0113	0.0117	0.0108	0.0111	0.0114
100%	MSEYTesting	0.0116	0.0121	322.4528	0.0110	0.0115	1624.4880	0.0108	0.0112	2548.6096
100%	MSEYCensus	0.0115	0.0117	0.0124	0.0109	0.0113	0.0117	0.0108	0.0111	0.0114
80%	MSEYInSample	0.0122	0.0126	0.0141	0.0115	0.0116	0.0125	0.0112	0.0115	0.0119
80%	MSEYTesting	0.0130	0.0142	20.1930	0.0117	0.0120	398.8582	0.0113	0.0117	1202.0398
80%	MSEYCensus	0.0126	0.0134	10.1035	0.0116	0.0118	199.4354	0.0112	0.0116	601.0258
50%	MSEYInSample	0.0129	0.0154	0.0145	0.0118	0.0124	0.0135	0.0117	0.0128	0.0129
50%	MSEYTesting	0.0143	0.0172	0.4933	0.0124	0.0129	49.4806	0.0118	0.0133	372.3238
50%	MSEYCensus	0.0139	0.0167	0.3497	0.0122	0.0127	34.6405	0.0118	0.0131	260.6305
30%	MSEYInSample	0.0139	0.0127	0.0152	0.0119	0.0123	0.0145	0.0118	0.0120	0.0135
30%	MSEYTesting	0.0147	0.0146	0.5691	0.0129	0.0136	11.4810	0.0123	0.0126	96.0948
30%	MSEYCensus	0.0145	0.0143	0.4583	0.0127	0.0133	9.1877	0.0122	0.0124	76.8785
100%	K (Full Census)	6.9800	2.0000	67.6400	11.9000	2.0400	142.6800	12.5400	2.0000	217.2600
100%	Time (Full Census)	4.0251	0.4753	1.3533	36.7225	1.2767	9.4859	75.9045	1.6847	30.9289

Table H18. MSE_f, MSE_{lSf}, MSE_{FSf}, MSE_y, MSE_{lSy}, MSE_{FSy}, time and *K* results for d=4, sigma = 0.1

APPENDIX I

PARAMETRIC BOOTSTRAP ALGORITHM TO CALCULATE EXPECTED OPTIMISM

We apply the following algorithm from Efron (2004) to compute in-sample optimism. First, we assume a Gaussian density $p(\mathbf{Y})=N(\widehat{\mathbf{Y}}, \hat{\sigma}^2\mathbf{I})$, where $\widehat{\mathbf{Y}}$ is the vector of estimated output values of the estimator for which we are assessing the in-sample optimism. $\hat{\sigma}^2$ is obtained from the residuals of a "big" model presumed to have negligible bias. Given CNLS's unbiasedness, high flexibility and use of a very complex description of the production function, we choose it as our "big" model. While obtaining an unbiased estimate for σ^2 from CNLS's residuals is complicated as there are no formal results regarding the effective number of parameters CNLS uses, using MSE^{CNLS}_{yLearn} as $\hat{\sigma}^2$ results in a downward biased estimator of σ^2. This downward bias in fact results in improved efficiency for the parametric bootstrap algorithm and is an example of a "little" bootstrap (Breiman, 1992). Thus, we let $\hat{\sigma} = MSE^{CNLS}_{yLearn}$. Efron (2004) then suggests to run a large number *B* of simulated observations $\mathbf{Y}^*$ from $p(\mathbf{Y})$, fit them to obtain estimates $\widehat{\mathbf{Y}^*}$ and estimate $cov_i=cov(\widehat{Y_i},Y_i)$ computing

$$\widehat{cov}_i= \sum_{b=1}^{B} \widehat{Y_i^{*b}}(Y_i^{*b}-Y_i^{*\cdot})/(B-1) \; ; \; Y_i^{*\cdot}= \sum_{b=1}^{B} Y_i^{*b}/B \qquad (I.1)$$

Finally, we select $B = 500$ for all our experiments based on observed convergence of the $\sum_{i=1}^{n} \widehat{cov}_i$ quantity.

Further, we note that if the researcher is not comfortable with the assumption made about the size of MSE^{CNLS}_{yLearn} relative to σ^2, sensitivity analysis (by adding a multiplier $c>1$, such that $p(\mathbf{Y})=N(\widehat{\mathbf{Y}}, c\hat{\sigma}^2\mathbf{I})$) can be performed to test against it. Finally, we note that other

distributions different than Gaussian can be considered to draw the bootstrapped $\mathbf{Y}^*$ vectors. This is especially useful when considering inefficiency, as skewed distributions can be also included.

APPENDIX J

COBB-DOUGLAS RESULTS WITH MULTIPLICATIVE RESIDUAL ASSUMPTION

FOR CHILEAN MANUFACTURING DATA

Industry Name and Code	n	Survey Size	R^2_{FS}	R^2_{CDM}	Ratio vs. Best Method
Other Metal Products (2899)	144	20%	50%	82%	CDM is Best Method
		30%	60%	85%	CDM is Best Method
		40%	64%	86%	CDM is Best Method
		50%	72%	86%	CDM is Best Method
		100%	88%	87%	CDM ties for Best Method
Wood (2010)	150	20%	35%	45%	CDM is Best Method
		30%	40%	50%	CDM is Best Method
		40%	47%	51%	CDM is Best Method
		50%	52%	53%	CDA ties for Best Method
		100%	66%	62%	0.94 vs. CAP-NLS
Structural Use Metal (2811)	161	20%	77%	79%	CDM ties for Best Method
		30%	82%	81%	CDM ties for Best Method
		40%	87%	84%	0.97 vs. CAP-NLS
		50%	90%	85%	0.94 vs. CAP-NLS
		100%	95%	92%	0.97 vs. CAP-NLS
Plastics (2520)	249	20%	54%	56%	CDM ties for Best Method
		30%	57%	56%	CDM ties for Best Method
		40%	57%	57%	CDM ties for Best Method
		50%	60%	57%	CDM ties for Best Method
		100%	64%	60%	0.94 vs. CAP-NLS
Bakeries (1541)	250	20%	72%	46%	0.64 vs. CAP
		30%	77%	50%	0.65 vs. CAP
		40%	78%	50%	0.64 vs. CAP
		50%	85%	51%	0.60 vs. CAP
		100%	99%	58%	0.59 vs. CAP

Table J1. Ratio of CDM to Best Model Performance