FUNDAÇÃO GETULIO VARGAS
ESCOLA DE ECONOMIA DE SÃO PAULO

VICTOR MARTINS ORESTES

A FRAMEWORK FOR SOLVING NON-LINEAR DSGE MODELS

São Paulo

2019

VICTOR MARTINS ORESTES

A FRAMEWORK FOR SOLVING NON-LINEAR DSGE MODELS

Dissertação apresentada à Escola de Economia de São Paulo como pré-requisito à obtenção de título de mestre em Economia de Empresas.

Orientador: Ricardo Pereira Masini.

Coorientador: Bernardo Guimarães.

São Paulo

2019

Orestes, Victor Martins.
A framework for solving non-linear DSGE models / Victor Martins Orestes. - 2019.
48 f.

Orientador: Ricardo Pereira Masini.
Co-orientador: Bernardo de Vasconcellos Guimarães.
Dissertação (mestrado CMEE) – Fundação Getulio Vargas, Escola de Economia de São Paulo.

1. Macroeconomia - Modelos matemáticos. 2. Equilíbrio econômico - Modelos matemáticos. 3. Modelos não-lineares (Estatística). 4. Estatística matemática - Processamento de dados. 5. Análise de regressão. I. Masini, Ricardo Pereira. II. Guimarães, Bernardo de Vasconcellos. III. Dissertação (mestrado CMEE) – Escola de Economia de São Paulo. IV. Fundação Getulio Vargas. V. Título.

CDU 330.101.541

Ficha Catalográfica elaborada por: Isabele Oliveira dos Santos Garcia CRB SP-010191/O
Biblioteca Karl A. Boedecker da Fundação Getulio Vargas - SP

VICTOR MARTINS ORESTES

A FRAMEWORK FOR SOLVING NON-LINEAR DSGE MODELS

Dissertação apresentada à Escola de Economia de São Paulo como pré-requisito à obtenção de título de mestre em Economia de Empresas.

Data de aprovação: __/__/____

Banca examinadora:

Prof. Dr. Ricardo Pereira Masini
FGV-EESP (Orientador)

Prof. Dr. Bernardo De Vasconcellos Guimarães
FGV-EESP (Coorientador)

Prof. Dr. Marcelo Cunha Medeiros
PUC - Rio

Agradecimentos

Ao meu orientador, professor Ricardo Masini, por todo o empenho, por ter acreditado neste trabalho desde o início e por sempre estar disposto a me ajudar com os problemas. Ao, também orientador, professor Bernardo Guimarães, por ter me acompanhado desde o começo do curso, sempre me guiando na direção correta. Agradeço aos dois por todo o apoio, amizade e confiança depositada em mim. Seus ensinamentos e conselhos foram inestimáveis para minha formação, indo muito além desta dissertação.

Ao professor Marcelo Medeiros, membro da minha banca, pelas importantes sugestões e críticas.

Ao corpo docente da EESP-FGV, por todas as aulas. Em especial, agradeço aos professores Daniel Monte e Victor Filipe Martins da Rocha.

Aos meus colegas de turma, pela amizade a auxílio. Em especial, agradeço ao Ariel, Bruna, Gabriel, João, Luis, Leonardo, Nicolas, Renan, Tulio e aos membros da Salinha.

Aos meus pais, Léa e Orestes, por todo amor e apoio incondicional. Nada disso teria sido possível sem a sua dedicação .

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Resumo

Propomos um arcabouço para resolver modelos DSGE não lineares. Para tanto, sorteamos uma amostra do espaço de estado, que é usada para estimar uma aproximação para as funções valor ou política de interesse. Utilizando técnicas de estatística de alta dimensão podemos atenuar o problema da dimensionalidade, ao mesmo tempo que mantemos a flexibilidade, garantias teóricas de convergência e limite superior para os erros. Em particular, propomos dois métodos diferentes: uma projeção regularizada e um algoritmo baseado em *Support Vector Machines* (SVM). Para ilustrar estes métodos de solução, aplicamos o primeiro algoritmo para resolver um modelo de crescimento básico, que tem uma solução linear conhecida, e mostramos que ele tem boa precisão e seleciona corretamente os coeficientes de uma base polinomial. Além disso, aplicamos o algoritmo de SVM para resolver um modelo Novo Keynesiano com um *Zero Lower Bound* (ZLB) e comparamos nossos resultados com os do método Smolyak, que é amplamente utilizado na literatura. Mostramos que este último superestima o impacto do ZLB na economia, alcançando uma precisão menor do que a da nossa solução.

Palavras-chave: modelos DSGE; Métodos de solução não lineares; Alta Dimensão; LASSO; Support Vector Machines; Zero lower bound

Abstract

We propose a framework to solve non-linear DSGE models combining approximation and estimation techniques. Instead of relying on a fixed grid, we use Monte Carlo methods to draw samples from the state space, which are used to estimate an approximation for the value or policy functions of interest. By using estimators from high-dimensional statistics we can attenuate the curse of dimensionality while maintaining flexibility, theoretical guarantees for convergence and upper bound for the errors. In particular, we propose two different methods: a regularized projection and a support vector machine algorithm. To illustrate these solution procedures, we apply the first algorithm to solve a standard growth model, which has a known linear solution, and show that it achieves a good accuracy, correctly shrinking the coefficients of a polynomial basis. Moreover, we use the support vector machine algorithm to solve a New Keynesian model with a Zero Lower Bound (ZLB) and compare our results with the ones from the Smolyak Method, which is widely used in the literature. We show that the latter overestimate the impact of the ZLB in the economy, achieving a lower accuracy than the one from our solution.

Keywords: DSGE models; Non-linear Solution methods; High-Dimensional; LASSO; Support Vector Machines; Zero lower bound

List of Figures

List of Tables

Contents

1 Introduction

Most of the objects of interest in dynamic stochastic general equilibrium (DSGE) modeling (value functions or policy functions) can be represented as a fixed point of some operator Γ acting on a functional space $\mathcal{F}$. Elements of $\mathcal{F}$ are functions defined over a state space $\mathcal{X}$, $f : \mathcal{X} \to \mathbb{R}^m$. The objective is to find $f^* \in \mathcal{F}$, such that:

$$\Gamma(f^*) = f^*$$

For a large class of problems, it is unfeasible to achieve a closed form solution for f^*. As a consequence, one must resort to numerical methods for an approximate solution, which usually consists of iterative procedures. However, these methods suffer from a phenomenon known as “curse of dimensionality". In short, computational costs tend to increase exponentially with the dimension of the state space.

In this paper, we propose a solution framework that combines approximation and estimation techniques. First, we approximate an element of $\mathcal{F}$ into a convenient space. Because we do not have a closed form for the approximation, we treat this as an estimation problem, relying on Monte Carlo methods to draw an i.i.d sample from the state space and using it to estimate the approximation. Finally, at each iteration step, we replace the original operator Γ by this estimation, then reducing the number of necessary evaluations at each step. Moreover, we do not use a fixed grid like most of the literature on numerical solutions for DSGE models.

However, the reduction in the number of evaluations has an accuracy cost. Specifically, our framework has two additional sources of imprecision: approximation and estimation error. Furthermore, for a fixed number of draws, there is an inherent tradeoff when attempting to attenuate these two errors. We propose using techniques from high-dimensional statistics to deal with this tradeoff. In particular, we propose two alternative solution algorithms: A high-dimensional projection and a support vector machine (SVM) algorithm. Moreover, we show that they share a very similar theoretical framework, provide sufficient conditions for convergence and derive error bounds.

The regularized projection algorithm allows us to approximate the function in a high-dimensional (but finite) dimensional space. The regularization techniques allows to get a good estimator for this projection even when the dimensional of the basis function is greater than the number of draws. This approach is especially useful when the solution is sparse (in terms of the chosen basis).

On the other hand, the SVM algorithm allows for approximations in a particular infinite dimensional space, a Reproducing Kernel Hilbert Space (RKHS). This is possible

because point evaluation in a RKHS can be represented as an inner product, hence evaluating an infinite dimensional basis is not necessary. The idea of applying this kind of approach to value function iteration was first proposed (from a Bayesian perspective) by Scheidegger and Bilionis (2019). We choose to present it in the framework of the RKHS for two reasons. First, it is easier to generalize to policy function iteration problems, where its important to use take into account the correlation between the functions within the estimation procedure. Second, it allows to encompass the algorithm in a more general theoretical framework.

Both procedures allow us to use convenient basis function or kernels to attenuate the curse of dimensionality, while maintaining the flexibility to account for specifies of a given model. To exemplify this point, we apply our solution procedure to solve the New Keynesian Model with a zero lower bound (ZLB) constraint presented by Fernández-Villaverde et al. (2015). In such model it is reasonable to expect a sharp change in the policy functions around the region where the ZLB binds. Consequently, we use the SVM algorithm with a non-stationary kernel, which is a class of kernels designed specifically to accommodate this variable smoothness in the function of interest (Paciorek and Schervish (2004)).

Our results differ from those obtained by Fernández-Villaverde et al. (2015), which employ the Smolyak method[1] to solve the model. Particularly, our solution indicates that the ZLB binds less often and has lesser impact on the variables[2]. Moreover, our solution achieves smaller Euler Equation Errors, indicating that it has more precision than the Smolyak's solution, which may suffer from a poor performance of Polynomial based approximations to capture the sharp changes around the ZLB. Moreover, our results highlights the importance of choosing an appropriate approximation procedure, taking into account the nature of the problem[3].

This paper is directly related to the literature of numerical methods for finding non-linear, global solutions of DSGE models. In particular, we are inserted on the branch that iterate on function approximations, instead of constructing a fine grid and defining a interpolation rule between the grid's points as in Richter et al. (2014). For excellent surveys on the literature of solution methods, see Maliar and Maliar (2014) and Fernández-

[1] The Smolyak method method combines projection (using a Chebyshev Polynomial basis) with a grid shrinking tool to avoid the exponential increase in the number of dimensions(see Judd et al. (2014) for a full description of and some variants).

[2] This result has been noted before when comparing polynomial and linear interpolation in a full grid-based solution. Richter et al. (2014) use a grid with around 70000 points to solve a ZLB model with 3 state variables. They compare the results using a linear interpolation and a interpolation with Chebyshev polynomials. The authors find that the ZLB binds more often when using the polynomial interpolation.

[3] Several other papers have also employed (a close variation) of the Smolyak Method with a Chebyshev basis for solving ZLB models. Take for instance Albertini and Poirier (2015), BoraUgan Aruoba et al. (2017) and Debortoli et al. (2019)

Villaverde et al. (2016).

There is a wide literature that combines projection techniques with tools to shrink the grid (which remain fixed), see Arcidiacono et al. (2013), Judd et al. (2014), Brumm and Scheidegger (2017) and Maliar and Maliar (2015) for notable examples. Our projection approach mainly differs from this literature by using Monte Carlo methods to draw a sample from the state space and performing a high-dimensional estimation in order to attenuate the approximation error.

The use of Monte Carlo methods in the iterative procedure is based and closed related to the work of Scheidegger and Bilionis (2019). They present an special case of our SVM algorithm, from a Bayesian perspective, to solve Value Function iteration problems. We extend the algorithm to accommodate correlation between the policy functions in the estimations. Moreover, by presenting on the RKHS perspective we were able to include this algorithm in a more broad theoretical framework. The authors presents a dimension reduction technique, known as active subspaces. It is possible to generalize this technique to multi output functions(Constantine et al. (2018)), however, since for most policy function iteration problems we do not have information about the gradient of the functions, we do not include this as an alternative.

In a broader sense, our work is related to the literature that uses randomization to break the curse of dimensionality. Usually, this is used in the context of integration, see for Novak and Wozniakowski (2010) for theoretical aspects and see Rust (1997) and Pál and Stachurski (2013) for applications to computing integrals in value function iteration algorithms. Here, we focus on using randomization for computing function approximation (see Kunsch (2017) and Novak and Wozniakowski (2012) for a general mathematical treatment).

Regarding estimation, we use techniques from high-dimensional statistics. Especially, we use the LASSO and the Support Vector Machine regression, see , respectively, Bühlmann and Van De Geer (2011) and Steinwart and Christmann (2008) for further discussion.

Finally, our work is also related to the literature of solution methods for New Keynesian Models with the Zero Lower Bound, see Hirose and Sunakawa (2019) for a survey of this literature.

In Section 2 we present the solution algorithms. In section 3 we focus on the theoretical aspects of the solution procedures. In Section 4 we apply the algorithms to solve two models: a standard growth model and a New Keynesian Model with a Zero Lower bound.

2 The Solution Framework

2.1 Overview

Solving a DSGE model usually involve finding a fixed point of an operator Γ. Our focus will be on the two most common methods: value function iteration and time iteration[1].

2.1.1 Value function iteration procedure

The problem may admit a recursive formulation of the following form:

$$V^*(x) = \sup_{y \in A(x)} \left[u(x,y) + \beta \int_{\mathcal{X}} V^*(x') Q(dx'|x,y) \right]$$

Where $u(\cdot)$ can be, for instance, a utility function. Then V^* is a fixed point of the operator Γ, defined as:

$$\Gamma(f) = \sup_{y \in A(x)} \left[u(x,y) + \beta \int_{\mathcal{X}} f(x') Q(dx'|x,y) \right] \tag{2.1.1}$$

2.1.2 Time iteration procedure

Even when the model does not admit a recursive formulation, which is the case for most of the New Keynesian DSGE models, we can represent (under some regularity conditions) the solution path of a generic DSGE model as:

$$\int_{\mathcal{X}} G(x_t, x_{t+1}, y_t, y_{t+1}) Q(dx_{t+1}|x_t, y_t) = 0, \quad t \geq 0$$

where $G : \mathcal{X}^2 \times \mathcal{Y}^2 \to \mathbb{R}^\ell$ and $Q(\cdot|X_t, Y_t)$ is a probability measure on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ conditional on the σ-algebra generated by (X_t, Y_t). Under some technical assumptions, the equilibrium concept above define a time-invariant policy function $y_t = f^*(x_t)$. Define the time iteration operator as:

$$\Gamma(f) := \left\{ g \in \mathcal{F} \mid \int_{\mathcal{X}} G(x, x', g(x), f(x')) Q(dx'|x, f(x)) = 0 \right\} \tag{2.1.2}$$

Then f^* will be a fixed point of Γ.

Our solution framework consists of embedding an estimation procedure within the iterative algorithm. The main objective is to avoid the curse of dimensionality while maintaining flexibility to incorporate model specific characteristics.

[1] The solution of the model can be characterized by the fixed point, as in Fernández-Villaverde et al. (2015), or be a process within the solution algorithm, such as an inner-loop, as in Aiyagari (1994)

2.2 Approximating the operator fixed point

Let Γ be the operator acting on some functional space $\mathcal{H}$, with respective norm $\|\cdot\|_{\mathcal{H}}$, whose fixed point ($\Gamma(f^*) = f^*$) we want to approximate.

Given an initial guess, $f_0 \in \mathcal{H}$, the standard iterative procedure at step s, $\{\Gamma^{(s)}(f_0)\}_{s=0}^{\infty}$, is defined recursively as

$$\Gamma^{(s)}(f_0) := \Gamma(\Gamma^{(s-1)}(f_0)) \quad \text{and} \quad \Gamma^0(f_0) := f_0$$

When $\mathcal{H}$ is a Banach space and Γ is a contraction, the above sequence converges to the unique fixed point of Γ, f^*. Even though it is possible to immediately derive a numerical grid-based algorithm from this fact, evaluating the operator $\Gamma^{(s)}$ may be computationally very costly when each element of $\mathcal{H}$ has a high-dimensional domain.

For instance, assume that $\mathcal{H}$ is composed by functions from a compact set $\mathcal{X} \subseteq \mathbb{R}^d$ to $\mathbb{R}$, $f : \mathcal{X} \to \mathbb{R}$. Although $\Gamma(f)$ is not available analytically, $\forall x \in \mathcal{X}$ we can evaluate pointwise $\Gamma(f(x))$. The algorithm would then proceed by defining a grid over $\mathcal{X}$ and iterating over it[2] until a convergence criterion is met. However, the grid's cardinality grows exponentially with d. For illustration, a grid consisting of 50 points for each dimension would have a total number of 312.50 million points when $d = 5$. When pointwise evaluations of Gamma are costly, this approach is computationally unfeasible.

Intuitively, the previous solution method is trying to "approximate" the continuous by constructing a fine grid. This is necessary because finding a good approximation in an arbitrary infinite dimensional space is hard. As a consequence, our approach will rely on searching for an approximation in another, more convenient, space. By constructing a measure on the state space space (usually an uniform when the space is compact), we will be able to generate a sample and estimate the approximation using relatively less points than the standard grid iteration.

Algorithm 1 presents a general version of the proposed solution framework. Formally, be Π_J an operator that maps $\mathcal{H}$ into a space $\mathcal{G}_J$. The modified operator $\Omega_J := \Pi_J \circ \Gamma$ will be the base of our iterative procedure. At each iteration step, we draw an *i.i.d.* sample, $\{x_i\}_{i=1}^n$ from $\mathcal{X}$ and obtain an estimation of Ω_J, denoted by $\widehat{\Omega}_{n,J}$. The framework's flexibility comes from the possible choices for Π_J and $\widehat{\Omega}_J$. Still, they must follow some guidelines, for each $f \in \mathcal{H}$, we expect that:

(i) $\Omega_J(f)$ is a good approximation of $\Gamma(f)$.

(ii) $\widehat{\Omega}_{n,J}(f)$ is a good estimator for $\Omega_J(f)$.

(iii) $\widehat{\Omega}_{n,J}(f)$ is computationally feasible to obtain.

[2] When necessary, this procedure may also include a interpolation method.

The guidelines highlight two sources of error in the algorithm: approximation and estimation error. On one hand, a very simple approximation, such as a linear one, can be accurately estimated with few draws, but at the cost of having a potentially higher approximation error. On the other hand, by increasing the approximation complexity, with the same number of draws, we may have a higher estimation error.

To deal with these tradeoffs, we suggest using techniques from high-dimensional statistics. Particularly, we propose 2 possible algorithms. The first is a regularized projection algorithm, which allows to search in a high-dimensional space. The second is a support vector machine algorithm. This algorithm allow us to search into another infinite dimensional space, a Reproducing Kernel Hilbert Space (RKHS), whose structure permits expressing point evaluations as inner products, it can be seen as a generalization of the algorithm proposed (from a Bayesian's perspective) by Scheidegger and Bilionis (2019).

Algorithm 1 General Algorithm

Define an initial guess, $f^{(0)} \in \mathcal{H}$
Define the operator Π_J
Define a sampler from $\mathcal{X}$
Define an estimator $\widehat{\Omega}_{n,J}$ for Ω_J
Define a positive integer n and a real $\bar{\epsilon} > 0$.
while $\epsilon > \bar{\epsilon}$ **do**
  Generate n independent draws from the sampler to obtain $x_1, \ldots, x_N$
  Using the draws $\{x_i\}_{i=1}^n$ and $\{\Gamma(f^s(x_i))\}_{i=1}^n$, estimate $\widehat{\Omega}_{n,J}(f^s)$
  Set $f^{next}(x) = \widehat{\Omega}_{n,J}(f^s)(x)$
  Calculate (an approximation for) the error $\epsilon = \|f^{next} - f^s\|_{\mathcal{H}}$
  Set $f^s = f^{next}$
end while

2.3 High-Dimensional Projection Algorithm

In this section we present the regularized projection algorithm. Be $\mathcal{X} \subseteq \mathbb{R}^d$ a compact subspace and μ a probability measure on $\mathcal{X}$. We assume[3] that $\mathcal{H}$ is a space containing functions from $\mathcal{X}$ to $\mathbb{R}$. Be $\Psi := (\psi_1, \ldots, \psi_J)$ a basis for a J-dimensional space $\mathcal{G}_J \subseteq \mathcal{H}$. Define Π_J such that, for any $f \in \mathcal{H}$,

$$\Omega_J(f) := \Pi_J(\Gamma(f)) = \Psi\beta_J(f) \quad \text{where} \quad \beta_J(f) := \mathbb{E}[\Psi'\Psi]^{-1}\mathbb{E}[\Psi'\Gamma(f)]$$

Where the expectation is taken with respect to μ. The $(J \times m)$ matrix $\beta_J(f)$ is unfeasible to obtain in practice, mainly due to the second term integral, but we can estimate it using n independent draws from $\mathcal{X}$ (with respect to μ). Whereas a simple OLS could be used if

[3] An extension of the algorithm to time iteration with multiple policy functions is straightforward. It will be necessary to perform the estimation step independently for each function and adjust the convergence criterion to be the norm of all individual errors.

$J << n$, by employing a regularization technique it is feasible to have $J >> n$, which is an alternative to potentially reduce the approximation error. Particularly, we use the least absolute shrinkage and selection operator(LASSO):

$$\widehat{\beta}_J(f) = \underset{\beta}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (\Gamma_Z(f(x_i)) - \Psi_Z(x_i)\beta)^2 + \lambda_n \|\beta\|_{\ell_1}$$

Where λ_n is the regularization parameter. $\Psi_Z(x_i) = (\psi_1(x_i)_Z, \ldots, \psi_J(x_i)_Z)$ is the standardized (each entry with sample mean 0 and variance 1) vector of basis function evaluated at the sample and $\Gamma_Z(f(x_n))$ is standardized to have sample mean 0.

For instance, in the value function iteration we would have:

$$\Gamma(f(x_n)) = \sup_{y \in A(x_n)} \left[u(x_n, y) + \beta \int_{\mathcal{X}} f(x') Q(dx'|x_n, y) \right]$$

Or for the time iteration procedure:

$$\int_{\mathcal{X}} G(x_n, x', \Gamma(f(x_n)), f(x')) Q(dx'|x_n, f(x)) = 0$$

Algorithm 2 High-Dimensional Projection Algorithm

Define an initial guess, for $f^{(0)} : \mathcal{X} \to \mathbb{R}$
Define a set of basis functions $\{\psi_j : \mathcal{X} \to \mathbb{R}, 1 \leq j \leq J\}$
Define a sampler from $\mathcal{X}$
Define a positive integer n and a real $\bar{\epsilon} > 0$.
while $\epsilon > \bar{\epsilon}$ **do**
  Generate n independent draws from the sampler to obtain $x_1, \ldots, x_n$
  Standardize each entry $\psi_j(x_i)_Z = \frac{\psi_j(x_i) - \bar{\psi_j}}{\widehat{\sigma_j}}$
  Denote $\Psi_i = (\psi_1(x_i)_Z, ..., \psi_J(x_i)_Z)$
  Denote $\Psi_Z = (\Psi_1, ..., \Psi_n)'$ a $(n \times J)$ matrix.
  Find $y_i = \Gamma(f^s(x)) - \bar{\Gamma}$.
  Denote $\mathbf{Y} := (y_1, \ldots, y_n)'$;
  Compute λ_n
  Perform the LASSO to obtain

$$\hat{\beta} := \underset{\beta}{\operatorname{argmin}} \frac{1}{n} \|\mathbf{Y} - \Psi_Z \beta\|_{\ell_2}^2 + \lambda_n \|\beta\|_{\ell_1}$$

  Define the next step function as

$$f^{next}(x) := \widehat{\beta}^0 + \sum_{j=1}^{J} \frac{\widehat{\beta}_j}{\widehat{\sigma_j}} \psi_j(x)$$

  where $\widehat{\beta}^0 = \bar{\Gamma} - \sum_{j=1}^{J} \frac{\widehat{\beta}_j}{\widehat{\sigma_j}} \bar{\psi_j}$
  Calculate (an approximation for) the error $\epsilon = \|f^s - f^{next}\|_{\mathcal{H}}$
  Set $f^s = f^{next}$
end while

The estimation will be:

$$\widehat{\Omega}_{n,J}(f) := \widehat{\beta}_0 + \sum_{j=1}^{J} \frac{\widehat{\beta}_J^j(f)}{\widehat{\sigma_j}} \psi_j$$

Where $\bar{\psi}_j$, $\widehat{\sigma_j}$ are the sample mean and standard deviation of $\{\psi_j(x_i)\}$. $\bar{\Gamma}$ is the sample mean of $\{\Gamma(f(x_i)\}$ and $\widehat{\beta}_0 = \bar{\Gamma} - \sum_{j=1}^{J} \frac{\widehat{\beta}_J^j(f)}{\widehat{\sigma_J}} \bar{\psi}_j$.

The convergence criterion is set regarding the $\mathcal{L}^2$ norm, which we approximate via Monte Carlo integration. The iterative procedure continues until achieving a convergence criterion $\bar{\epsilon}$.

2.3.1 Choice of the regularization parameter

The parameter λ_n can be chosen by k-fold cross validation. However, we will need to perform the cross validation at each iteration step, which tends to be time consuming. When this is a problem, it is possible to choose the regularization parameter using a Bayesian Information Criteria[4].

2.4 Support Vector Machine Algorithm

Now we set $\mathcal{G}$ to be an infinite dimensional space with a particular structure: it is a Reproducing Kernel Hilbert Space (RKHS). In this space, point evaluations can be represented as inner products, thus bypassing direct evaluation of an infinite dimensional basis. This method, known as "kernel trick", allows to approximate a function directly in $\mathcal{G}$.

We follow Alvarez et al. (2012) to define a kernel for vector valued functions. A kernel is a function $\mathbf{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{m \times m}$, such that $\forall x, x' \in \mathcal{X}$, $\mathbf{K}(x, x')$ is a positive semi-definite matrix. A Reproducing Kernel Hilbert Space is a Hilbert space $\mathcal{H}_{\mathbf{K}}$ of functions $f : \mathcal{X} \to \mathbb{R}^m$, such that, for every $\mathbf{c} \in \mathbb{R}^m$ and $x \in \mathcal{X}$, $\mathbf{K}(x, \cdot)\mathbf{c}$ belongs to $\mathcal{H}_{\mathbf{K}}$. Moreover K has the reproducing property:

$$\langle f, \mathbf{K}(\cdot, x)\mathbf{c} \rangle_{\mathcal{H}_{\mathbf{K}}} = f(x)'\mathbf{c}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}_{\mathbf{K}}}$ is the inner product in $\mathcal{H}_{\mathbf{K}}$. Consequently, the norm of $\mathcal{H}_{\mathbf{K}}$ is defined implicitly by this inner product. Additionally, if $\mathbf{K}$ is a kernel then there exists a unique (up to an isometry) RKHS of functions which admits $\mathbf{K}$ as the reproducing kernel (Micchelli and Pontil (2005)).[5]

[4] Operationally, one would construct a grid over possible values for λ_n and select the one that minimizes the Bayesian Information Criteria. This approach can be further refined in a high-dimensional setting, see Chen and Chen (2008)

[5] For a detailed explanation of the theory regarding RKHS, see Chapter 4 of Carmeli et al. (2010)

Example: The Gaussian Radial Basis Function kernel is given by:

$$K_\lambda(x, x') := \exp\left(-\frac{\|x - x'\|_2^2}{\gamma^2}\right)$$

$\forall \gamma > 0$, this kernel is dense in the space L^2 (see Theorem 4.63 from Steinwart and Christmann (2008)). Scheidegger and Bilionis (2019) use a close variant in their solution procedure. Moreover, this kernel can be extended to vector valued functions:

$$\mathbf{K}(x, x') = K_\lambda(x, x') \otimes \mathbf{B}$$

Where $\mathbf{B}$ is a $m \times m$ positive definite matrix that captures the correlations between the outputs. Note that, by the Cholesky Decomposition, $\mathbf{B} = LL'$, where L is a lower triangular matrix, so $\mathbf{B}$ has a total of $(m(m+1)/2)$ hyperparameters.

2.4.1 Approximating via a RKHS

Be $\mathbf{K}$ a bounded and measurable kernel[6] and $\mathcal{G}$ the RKHS that admits $\mathbf{K}$ as reproducing Kernel. Denote by $\|\cdot\|_\mathcal{G}$ the norm of $\mathcal{G}$. Consider a finite sample of n i.i.d draws from $\mathcal{X}$. To estimate the approximation, we perform the following Support vector machine regression:

$$\widehat{\Pi}_n(f) = \underset{g \in \mathcal{G}}{\operatorname{argmin}} \quad \frac{1}{n}\sum_{i=1}^{n} \|g(x_i) - f(x_i)\|_2^2 + \lambda_n \|g\|_\mathcal{G}^2$$

Where $\lambda_n > 0$ is a regularization parameter. By the Representer Theorem (see Micchelli and Pontil (2005) and Alvarez et al. (2012)) the unique solution is given by:

$$\widehat{\Pi}_n(f) = \mathbf{K}_x' \mathbf{c}$$

Where $\mathbf{c} = [\tilde{\mathbf{K}} + \lambda_n \mathbf{I}]^{-1}\mathbf{f}]$ and

$$\tilde{\mathbf{K}} = \begin{bmatrix} \mathbf{K}(\mathbf{x}_1, \mathbf{x}_1) & \dots & \mathbf{K}(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ \mathbf{K}(\mathbf{x}_n, \mathbf{x}_1) & \dots & \mathbf{K}(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}$$

$\mathbf{f} = (f(x_1), \dots f(x_n))'$. Finally $\mathbf{K}_x$ is a function of x, given by:

$$\mathbf{K}_x(x) = \begin{bmatrix} \mathbf{K}(x, \mathbf{x}_1) & \dots & \mathbf{K}(x, \mathbf{x}_n) \end{bmatrix}'$$

The representer theorem reduces the problem of searching in an infinite dimensional space to finding a finite number of coefficients. Algorithm 3 presents the iterative procedure, which uses the operator:

$$\widehat{\Omega}_n = \widehat{\Pi}_n \circ \Gamma$$

[6] This will be the case, for instance, if the kernel is continuous.

2.4.2 Hyperparameters Selection

Be $vec(\cdot)$ the vectorization operator. At each iteration step, denote by $\mathbf{Y} = (\Gamma(f^s(x_1)), \ldots, \Gamma(f^s(x_n)))'$. We choose all hyperparameters as to maximize:

$$(\lambda_n, L, \gamma) = \underset{\lambda_n, L, \gamma}{\operatorname{argmax}} -\frac{1}{2} vec(\mathbf{Y})' \left(\tilde{\mathbf{K}} \otimes (LL') + \lambda_n \mathbf{I}\right)^{-1} vec(\mathbf{Y}) - \ln |\tilde{\mathbf{K}} \otimes (LL^t) + \lambda_n \mathbf{I})|$$

This heuristic comes from the log-likelihood of a Matrix Normal Distribution. The maximization is implemented utilizing a close variation of the Matlab code provided by Chen et al. (2017).

Algorithm 3 Reproducing Kernel algorithm

Define an initial guess for the policy function $f^{(0)} : \mathcal{X} \to \mathbb{R}^m$
Define a kernel $\mathbf{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{m \times m}$ with a vector of hyperparameters (γ, λ_n)
Define a sampler from $\mathcal{X}$
Define a positive integer n and a real $\bar{\epsilon} > 0$.
while $\epsilon > \bar{\epsilon}$ **do**
  Generate n independent draws from the sampler to obtain $x_1, \ldots, x_n$
  Find $y_i = \Gamma(f^s(x_i))$.
  Denote $\mathbf{Y} := (y_1, \ldots, y_n)'$.
  Compute γ and λ_n
  Define the next step function as:

$$f(x)^{next} := \mathbf{K}_x(x)'[\tilde{\mathbf{K}} + \lambda_n \mathbf{I}]^{-1}\mathbf{Y}$$

  Calculate (an approximation for) the error $\epsilon = \|f^s - f^{next}\|_{\mathcal{L}^2}$
  Set $f^s = f^{next}$
end while

2.5 Choice of Algorithm

The choice of algorithm will depend on previous information about the model. If the model is generally well behaved, i.e, it does not contain kinks, convergence issues and other problems, the high-dimensional projection algorithm associated with a polynomial basis (when the dimension of the state space is low) or with a radial basis function (for high-dimensional state space) tends to work well, especially when employing cross validation to choose the regularization. On the other hand, if there is issues regarding convergence or kinks, the SVM algorithm can be adapted to deal with these issues. Moreover, when there is doubt if it's possible to use a good approximation in a finite dimensional space (theoretically, this is related to the norm of the projection coefficients, see section 3.4.1), it is better to use the SVM algorithm.

2.6 Computational Aspects: Evaluating Integrals

We will need to compute integrals over the state space $\mathcal{X}^e \subseteq \mathcal{X}$ of exogenous state variables, usually in the form of an expectation:

$$\mathbb{E}[f] := \int_{\mathcal{X}^e} f dQ$$

Where Q is a distribution on $\mathcal{X}^e$ (conditional on previous state realizations).

An analytic form for $E[f]$ is usually not available. Numerical evaluations are possible using Curbature methods, which use a set of n points from the state space and assigned weights to compute the integral:

$$\mathbb{E}[f] \approx \sum_{i=1}^{n} w_i f(\mathbf{x}_i) = \mathbf{w}'\mathbf{f}$$

Where $\mathbf{x}_i \in \mathcal{X}^e$, $w_i \in \mathbb{R}$ and $\mathbf{w} = (w_1, ..., w_n)'$.

2.6.1 Importance Sampling

Assume that we can draw i.i.d samples from a distribution P, such that, $P(x) > 0$ if $f(x)Q(x) \neq 0$. Then be $\{\mathbf{x}_i\}_{i=1}^n$ an i.i.d. sample from P, the approximation will be:

$$\mathbb{E}[f] \approx \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{x}_i) \frac{Q(x)}{P(x)}$$

Note that the standard Monte Carlo integration is recovered if $P = Q$, in which case the weights are constant, $w_i = \frac{1}{n}$.

2.6.2 Monomial Rule

If it is possible to to write:

$$\mathbb{E}[f] = \int_{\mathbb{R}^d} g(x) \exp\left(-\sum_{i=1}^{d} x_i^2\right) dx$$

We employ a Monomial rule proposed by Stroud(1971). This is the fastest method by approximating integrals and usually can be applied when the exogenous shocks are Gaussian.

2.6.3 Kernel Curbature

Be $\mathcal{G}$ a RKHS with a bounded and measurable kernel. Be $\{\mathbf{x}_i\}_{i=1}^n$ a set of n i.i.d draws from $\mathcal{X}^e$, consider the following approximation for f:

$$\widehat{f} = \text{argmin}_{g \in \mathcal{G}} \quad \frac{1}{n} \sum_{i=1}^{n} \|g(x_i) - f(x_i)\|_2^2 + \lambda_n \|g\|_{\mathcal{G}}^2$$

As stated in section 2.4:

$$\widehat{f} = K'_x[\tilde{\mathbf{K}} + \lambda_n \mathbf{I}]^{-1}\mathbf{f}$$

So it is possible to use the approximation:

$$\mathbb{E}[f] \approx \mathbb{E}[\widehat{f}] = \left(\int_{\mathcal{X}^e} \mathbf{K}_x(x)Q(dx)\right)' [\tilde{\mathbf{K}} + \lambda_n \mathbf{I}]^{-1}\mathbf{f}$$

In this case the weights are $\mathbf{w}'_K := (\int \mathbf{K}_x(x)Q(dx))' [\tilde{\mathbf{K}} + \lambda_n \mathbf{I}]^{-1}$. For this rule to be computationally feasible, we must have a closed form solution for the kernel mean[7].

[7] Closed form solutions can be found for several pairs of kernels and distributions, see Briol et al. (2019)

3 Theoretical Framework

3.1 Overview

The objective of this section is to present a theoretical framework for the previous algorithms. We are mainly interested in sufficient conditions to ensure some properties of the modified iterative procedure. Particularly if it preserves the contraction of Γ, and if we can find a bound for the errors. Both properties are presented in Poposition 2. Below we define the theoretical framework and elicit the assumptions that guarantees our results.

Be $\mathcal{X} \subseteq \mathbb{R}^d$, the Borel set of $\mathcal{X}$ is denoted by $\mathcal{B}(\mathcal{X})$. Be μ a probability measure on $\mathcal{X}$. Let Y be a Hilbert space endowed with inner product $\langle \cdot, \cdot \rangle_Y$. From now own we consider the (separable) Hilbert space:

$$\mathcal{H} = L^2(\mathcal{X}, \mathcal{B}(\mathcal{X}), \mu)$$

Which contains all Borel-measurable functions $f : \mathcal{X} \to Y$, such that $\|f\|_{\mathcal{H}} < \infty$, where $\| \cdot \|_{\mathcal{H}}$ is the norm associated with the inner product of $\mathcal{H}$, given by:

$$\langle f, g \rangle_{\mathcal{H}} = \int_{\mathcal{X}} \langle f(x), g(x) \rangle_Y \, \mu(dx)$$

$\Gamma : \mathcal{H} \to \mathcal{H}$ will be the operator whose fixed point f^* we want to approximate. Our approach relies on constructing an approximation operator indexed by $J \in \mathbb{N}$, $\Pi_J : \mathcal{H} \to \mathcal{G}_J \subseteq \mathcal{G}$ and $\mathcal{G} \subseteq \mathcal{H}$. Moreover, let $\{\mathcal{G}_j\}_{j=1}^{\infty}$ be a collection of nested subsets of $\mathcal{G}$, i.e., $\mathcal{G}_J \subseteq \mathcal{G}_{J+1}$. We will use the following modified operator as the basis of our iterative procedure:

$$\Omega_J := \Pi_J \circ \Gamma$$

For each J, consider a sequence of random operators $\{\widehat{\Omega}_{n,J}\}_{n=1}^{\infty}$ from $\mathcal{H}$ to $\mathcal{H}$, which represents the estimations. Given an initial guess $f^0 \in \mathcal{H}$ and $n \in \mathbb{N}$, the step s of the modified iterative procedure is defined recursively as

$$\widehat{\Omega}_{n,J}^{(s)}(f^0) = \widehat{\Omega}_{n,J}(\widehat{\Omega}_{n,J}^{(s-1)}(f^0))$$

Where $\widehat{\Omega}_{n,J}^{(0)}(f^0) = f^0$.

3.2 Assumptions

In this section we elicit our assumptions. The first one ensures that Γ is a contraction.

Assumption 1. (Contraction Mapping) $\Gamma : \mathcal{F} \to \mathcal{F}$ is a contraction mapping, i.e, $\forall f, g \in \mathcal{H}$, $\exists\ \kappa \in (0,1)$, such that t:

$$\|\Gamma(f) - \Gamma(g)\|_{\mathcal{H}} \leq \kappa \|f - g\|_{\mathcal{H}}$$

We denote by $\underline{\kappa}$ the contraction coefficient of Γ :

$$\underline{\kappa} := \inf \quad \{\kappa \in (0,1) : \forall f, g \in \mathcal{H} \quad \|\Gamma(f) - \Gamma(g)\|_{\mathcal{H}} \leq \kappa \|f - g\|_{\mathcal{H}}\}$$

Its important to discuss this assumption. Usually in value function iteration problems the contraction is in a Banach Space endowed with a sup norm ($\|\cdot\|_\infty$), mainly because of the Blackwell Sufficient Conditions. However, this space is not generally a Hilbert Space, take for instance L^∞. Moreover, being a contraction regarding the sup norm does not necessarily imply being a contraction on a Hilbert Space norm (nor the inverse). However, we lose some important properties regarding approximation methods when working on general Banach Spaces. For the projection, we cannot guarantee unicity and that it is a non-expansive mapping (which is useful for preserving the operator's contraction property). Moreover, for the SVM algorithm, the convergence is related to the loss function and we would not have a closed form solution for the estimation if we pick the maximum loss, nor the theoretical results for consistency.

Although is possible to employ non-expansive approximations [1] with respect to $\|\cdot\|_\infty$, this class is very restrict and does not include even methods widely employed in the literature, such as projections on polynomial basis. As a consequence, if we restrict our focus to non-expansionary maps in Banach spaces we lose the flexibility in choosing the approximation. However, in the first example of section 4 we show numeric convergence of the high-dimensional projection algorithm to the analytic solution on a basic growth model, where we only have theoretical results for the sup norm or the Thompson Metric (Montrucchio (1998)).

The next assumption ensures that $\mathcal{G}$ is a good approximation space.

Assumption 2. (Approximation Space) $\mathcal{G}$ is a dense (with respect to the norm of $\mathcal{H}$) subset of $\mathcal{F}$, i.e., $\text{cl}(\mathcal{G}) = \mathcal{F}$ and

$$\bigcup_{J \geq 1} \mathcal{G}_J = \mathcal{G}$$

Remark 1. This assumption also encompass the case where $\mathcal{G}_J = \mathcal{G}\ \forall\ J$.

By assumption 3, the approximation operator maintains the contraction property of Γ and converges pointwise as J increases.

[1] For the application on value function iteration problems see (Pál and Stachurski (2013)). However, when the approximation is composed of finite linear functionals, neither deterministic nor Monte Carlo methods can break the curse of dimensionality(see Novak and Woźniakowski (2009) and Kunsch (2016), respectively).

Assumption 3. (Approximation Operator) We have the following properties for Π_J, $\forall f, g \in \mathcal{H}$:

(i) Π_J is a non-expansive operator, i.e.,

$$\|\Pi_J(f) - \Pi_J(g)\|_{\mathcal{H}} \leq \kappa \|f - g\|_{\mathcal{H}}$$

(ii) The sequence $\{\Pi_J \circ \Gamma\}_{J=1}^{\infty}$ converges pointwise to Γ. Moreover, the convergence is uniform on any compact subset of $\mathcal{H}$.

For instance, if $\mathcal{G}_J$ is a closed and convex subset of $\mathcal{G}$ of dimension J and Π_J is the projection operator, then assumption 3 holds (see subsection 3.4.1). If, for all J, Π_J is the identity operator then the standard iterative procedure is recovered.

Finally, the next assumptions ensures that we can consistently estimate the approximation.

Assumption 4. (Consistency of the estimator) $\forall J \in \mathbb{N}, \forall f \in \mathcal{H}$,

$$\widehat{\Omega}_{n,J}(f) \xrightarrow{p} \Omega_J(f)$$

3.3 General Results

The first result, Proposition 1, bounds the distance between the fixed points of Ω_J and Γ. Moreover, we can see that the approximation error is intensified by the iterative procedure.

Proposition 1. *Under Assumptions 1-3, the following are properties of the iterative procedure* Ω_J .

(i) Ω_J *is a contraction mapping with contraction coefficient* $\kappa_J \leq \underline{\kappa}$*, thus admits a unique fixed point* f_J^**.*

(ii) Be f^* *the fixed point of* Γ*, then:*

$$\|f_J^* - f^*\| \leq \frac{\|f^* - \Pi_j(f^*)\|}{1 - \kappa_J}$$

Remark 2. Notice that f^* is *not* necessarily the fixed point of the operator Ω_J for any J, because $\Omega_J(f^*) = \Pi_J(\Gamma(f^*)) = \Pi_J(f^*)$.This will not be equal f^*, unless $f^* \in \mathcal{G}_J$ for some J.

Also $\Pi_J(f^*)$ is *not* necessarily the fixed point of Ω_J either because $\Omega_J(\Pi_J(f^*)) = \Pi_J(\Gamma(\Pi_J(f^*)))$. Once again unless $\Gamma(\Pi_J(f^*)) = f^*$, which would be the case if $f^* \in \mathcal{G}_J$ for some J.

The next proposition presents our main theoretical result.

Proposition 2. *Under assumptions 1-4, the following are properties of the iterative procedure* $\widehat{\Omega}^{(s)}_{n,J}(f^0)$. $\forall s \geq 1$, $\forall f, g \in \mathcal{H}$*:*

(i) (Consistency) $\widehat{\Omega}^{(s)}_{n,J}(f) \xrightarrow{p} \Omega^{(s)}_{J}(f)$.

(ii) (Uniform Convergence in probability) Be $\mathcal{F}$ *a compact subset of* $\mathcal{H}$, $\forall s \geq 1$ *we have* $\sup_{f \in \mathcal{F}} \|\widehat{\Omega}^{(s)}_{n,J}(f) - \Omega^{(s)}_{J}(f)\| \xrightarrow{p} 0$.

(iii) (Contraction)

$$\|\widehat{\Omega}^{(s)}_{n,J}(f) - \widehat{\Omega}^{(s)}_{n,J}(g)\|_{\mathcal{H}} \leq (o_p(1) + \underline{\kappa}^s)\|f - g\|_{\mathcal{H}}$$

(iv) (Upper Bound for error) Be f^* *the fixed point of* Γ*, then*

$$\|\widehat{\Omega}^{(s)}_{n,J}(f) - f^*\| \leq o_p(1) + \underline{\kappa}^s\|f - f^*\| + \frac{\|f^* - \Pi_j(f^*)\|}{1 - \underline{\kappa}}$$

(v) (Limits) $\widehat{\Omega}^{(s)}_{n,J}(f) \xrightarrow{p} f^*$ *as* $n \to \infty$ *followed by* $s \to \infty$ *and* $J \to \infty$.

Where all the convergence in probability happens when $n \to \infty$

By proposition 2, our iterative procedure inherits the consistency of the estimator, including uniform convergence on any compact subset of $\mathcal{H}$[2]. Moreover, there is an additive upper bound for the iterative procedure's error: the first term comes from the estimation error, the second term is the error relative to the initial guess[3], which goes to 0 with the number of iterative steps and the third term is the approximation error. The bound express the tradeoff between approximation accuracy and estimation accuracy discussed in section 2.

3.4 Applications

3.4.1 High-Dimensional Projection Algorithm

In order to apply the results from Proposition 2 to the high-dimensional projection algorithm we first need to write it down in terms of the theoretical framework.

To satisfy assumption 2, $\mathcal{G}$ has to be a dense subset of $\mathcal{H}$. Moreover, $\forall J \in \mathbb{N}$, $\mathcal{G}_J$ has to be a closed subset of $\mathcal{G}$ with dimension J, thus be $\Psi := (\psi_1, \ldots, \psi_J)$ a basis for $\mathcal{G}$.

[2] The uniform converge result is a direct consequence of pointwise convergence and stochastic equicontinuity (Newey (1991)). See the proof of lemma 6.

[3] It is not possible to get rid of this term, since it comes from the original operator Γ.

The operator Π_J will be the nearest-point projection:

$$\Pi_J(f) = \underset{g \in \mathcal{G}_J}{\operatorname{argmin}} \|f - g\|$$

Since $\mathcal{G}_J$ is closed (and consequently convex), we can apply the Hilbert Space Projection Theorem (Rudin (2006), pg. 80) to conclude that the projection is well-defined for every $f \in \mathcal{H}$ and J. The unique minimizer is given by the orthogonal projection of f onto $\mathcal{G}_J$. In particular, for a given basis, it can be expressed as:

$$\Psi' \beta_J(f) \quad \text{where} \quad \beta_J(f) := \mathbb{E}[\Psi'\Psi]^{-1}\mathbb{E}[\Psi'\Gamma(f)]$$

Again, the expectation is taken with respect to μ. Proposition 3 ensures that assumption 3 holds.

Proposition 3. *Be* $\Pi_J(f) = \operatorname{argmin}_{g \in \mathcal{G}_J} \|f - g\|$ *and define* $\Omega_J = \Pi_J \circ \Gamma$. *The sequence* $\{\Omega_J\}_{J=1}^{\infty}$ *converges pointwise to* Γ.

Next, we elicit the assumptions for the consistency of the LASSO. It is important to emphasize that we are interest only in prediction consistency, i.e., for $f \in \mathcal{H}$, we want :

$$\|\Psi \widehat{\beta}_J(f) - \Psi \beta_J(f)\| \xrightarrow{p} 0$$

We will need to bound higher order moments to be able to work in a high-dimensional setting.

Assumption 5. $\forall f \in \mathcal{H}$, $\mathbb{E}[|\Gamma(f)|^{2q}] < \infty$ and $\forall j \in \{1, \ldots, J\}$, $\mathbb{E}[|\psi_j|^{2q}] < \infty$ for some $q > 2$.

Assumption 5 will always be true, for instance, if $\mathcal{X}$ is compact, $\Gamma(f)$ is bounded in $\mathcal{X}$ when [4] f is bounded in $\mathcal{X}$ and we choose a bounded basis function. Since the draws from the state space are *i.i.d.*, this is all we need to bound the prediction error. The assumption that $q > 2$ is there to ensure that we can actually work in a high-dimensional setting, with $J >> n$. For consistency, we will need a last assumption about the norm of β_J does not grows too fast.

Assumption 6. $\|\beta_J\|_{\ell 1}$ is of smaller order than $\frac{\sqrt{N}}{J^{\frac{1}{q}}}$.

Be $\mathbf{x} = (x_1, \ldots, x_n)$ an i.i.d. sample from $\mathcal{X}$. Define by $\Psi(\mathbf{x})$ as the $N \times J$ matrix consisting of the basis functions evaluated at the sample, and by $\widehat{\beta}_J(f)$ the associated LASSO estimation, given a current guess f, the next proposition bounds the prediction error.

[4] It is enough to pick an initial guess that is also bounded in $\mathcal{X}$

Proposition 4. *(Consistency) For some $t > 0$, let the regularization parameter be:*

$$\lambda_n = t\frac{J^{\frac{1}{q}}}{\sqrt{N}}$$

Then, under assumption 5, with probability at least $1 - \alpha$, where, for some constant C_q,

$$\alpha = \frac{C_q}{t^q}$$

We have:

$$\frac{1}{n}\|\Psi(\boldsymbol{x})\widehat{\beta}_J(f) - \Psi(\boldsymbol{x})\beta_J(f)\|_2^2 \leq 2\lambda_n\|\beta_J\|_{\ell 1}$$

Moreover, given assumption 6, it shows that is sufficient to take λ_n of order[5] $\frac{J^{\frac{1}{q}}}{\sqrt{N}}$ to ensure consistency of the LASSO.

3.4.2 Reproducing kernel Algorithm

We just need to choose a RKHS $\mathcal{G}$ that is dense in $\mathcal{H}$ for assumption 2 to hold. Given that, assumption 3 will be true, since we have $\forall\ J$, $\mathcal{G}_J = \mathcal{G}$. To ensure consistency it is sufficient[6] that $\lambda_n \to 0$ as $n \to \infty$.

[5] Under stronger assumptions, such as a sub-Gaussian error,one could achieve an order of $\sqrt{\frac{\log J}{n}}$ (see chapter 6 of Bühlmann and Van De Geer (2011).

[6] The prove of consistency for the case where f is real valued can be found in chapter 5 of Steinwart and Christmann (2008), for more general cases, including vector valued functions, see theorem 4.4 and corollary 4.6 of Combettes et al. (2018)

4 Numerical Examples

4.1 A basic growth model

Consider the following discrete time stochastic growth model where, given a initial capital stock k_0, a planner choose a sequence $\{c_t, k_{t+1}\}_{t\geq 0}$ to maximize

$$\mathbb{E}_0 \sum_{t\geq 0} \beta^t u(c_t) \quad \text{subject to} \quad k_{t+1} + c_t = A_t h(k_t)$$

Where $\beta \in (0,1)$ is the discount factor, $A_t = \exp(a_t)$ and $a_t \sim N(0, \sigma_a^2)$ i.i.d. The optimality conditions imply:

$$v(c) = \beta \mathbb{E}[v(c')A'p(k')]$$
$$k' = A'h(k) - c$$

where $v := du/dc$ is the marginal utility and $p := dh/dk$ is the marginal productivity of capital. Define $y = Ah(k)$, by doing this we can reduce number of state variables to just 1. Combining the two optimality conditions with the last definition, the problem simplifies to:

$$v(c) = \beta \mathbb{E}[v(c')A'p(y-c)]$$

Be $\mathcal{X} \subseteq R_{++}$ the state space defined by $\mathcal{X} = [\underline{y}, \bar{y}]$. We can define the time iteration operator as:

$$\Gamma(f) := \{g \in C_b(\mathcal{X}) \mid v(g) - \beta p(y - f(y))\mathbb{E}[v(f(y'))A'] = 0\}$$

For the particular case when $u(c) = \ln(c)$, $h(k) = k^\alpha$and $\alpha \in (0,1)$, the time iteration operator Γ becomes:

$$\Gamma(f) := \left\{ g \in C_b(\mathcal{X}) \mid \frac{1}{g} - \beta\alpha(y - f(y))^{\alpha-1}\mathbb{E}_t\left[\frac{A'}{f(A'(y-f(y))^\alpha)}\right]\right\}$$

Where $C_b(\mathcal{X})$ is the space of continuous and bounded functions on $\mathcal{X}$. Note that this is a subset of the $L^2(\mathcal{X}, \mathcal{B}(\mathcal{X}), \mu)$, where μ is a uniform measure on $\mathcal{X}$. However, we do not have a proof that this operator is a contraction in the L^2 norm, only for the case where $C_b(\mathcal{X})$ is endowed with the sup norm (Coleman (1990)).

Furthermore, the problem has a closed form solution (see chapter 3 of Ljungqvist and Sargent (2004)) given by

$$f^*(y) = (1 - \alpha\beta)y$$

4.1.1 Solution Procedure

We set μ to be a uniform measure on $\mathcal{X}$ and use the High-Dimensional Projection algorithm. To test whether the algorithm correctly shrinks the coefficients we use a polynomial basis:

$$\psi_j(x) = x^{j-1}$$

At each iteration step, to compute $\Gamma(f^s(x))$ we need to compute the integral:

$$\mathbb{E}_t\left[\frac{A'}{f^s(A'(y - f^s(y))^\alpha)}\right]$$

Since the shocks are Gaussian, we compute the integrals using a degree 11 monomial rule from Stroud (1971). At each iteration we make $n = 100$ draws from $\mathcal{X}$ and use a 10-fold cross validation to select λ_n. We test with $J = 10, 100$ and 1000. The calibration of the model follows: $\alpha = 0.3$, $\beta = 0.98$ and $\sigma_a = 1$.

4.1.2 Results

We first set $\lambda_N = 0$ to test the OLS result(Using J=10). Not surprisingly it converges to the true policy function, since the solution is linear. The results also gets very close to the true solution when employing the lasso algorithm for the three considered dimensions, although with a small bias due to the regularization.

4.2 A New Keynesian Model with a Zero Lower Bound

4.2.1 Overview

In this example we solve the model presented by Fernández-Villaverde et al. (2015). It consists of a baseline New Keynesian model with a Zero Lower Bound on the nominal interest rate.

There is a representative household that makes decision about consumption, savings and supply labor (in a competitive market) for the intermediate goods producers. The intermediate goods producers operate in a monopolistic competition market structure. They are also subject to Calvo pricing, i.e, at each period they have a probability θ of not being able to change prices. They supply the intermediate goods for a final goods producers, who aggregate and sell the final consumption good to the households in a competitive market. The Central Bank fixes the one-period nominal interest rate according to a Taylor Rule and the Government consumes and finance its spending with lump-sum taxes. The economy is subject to four shocks: the discount factor, productivity, interest rate and government spending shock.

4.2.2 The Model

For simplicity we write down the equations that characterize the equilibrium conditions. We refer the reader to Fernández-Villaverde et al. (2015) for a complete derivation.

The equilibrium is given by a sequence of variables:

$$\{y_t, c_t, l_t, x_{1,t}, x_{2,t}, w_t, \Pi_t, \Pi_t^*, \Delta_t, R_t, Z_t, \beta_t, A_t, m_t, g_t, s_{g,t}\}_{t=0}^{\infty}$$

Where the variables denote, respectively, output, consumption, labor supply, two auxiliary variables regarding the firms pricing decision, real wage, gross inflation, the ration between target price level and actual price level, price dispersion, gross interest rate, gross "shadow" interest rate(ignoring the bound), discount factor, productivity, monetary shock, government spending and the fiscal shock. The equations that characterize the equilibrium are:

$$\frac{1}{c_t} = R_t E_t \left[\frac{\beta_{t+1}}{c_{t+1}\Pi_{t+1}} \right] \tag{4.2.1}$$

$$w_t = \chi l_t^{\varphi} c_t \tag{4.2.2}$$

$$x_{1,t} = \frac{\varepsilon - 1}{\varepsilon} x_{2,t} \tag{4.2.3}$$

$$x_{1,t} = \frac{y_t}{c_t}\frac{w_t}{A_t} + \theta E_t[\beta_{t+1}\Pi_{t+1}^{\varepsilon} x_{1,t+1}] \tag{4.2.4}$$

$$x_{2,t} = \Pi_t^* \left(\frac{y_t}{c_t} + \theta E_t \left[\beta_{t+1} \frac{\Pi_{t+1}^{\varepsilon-1}}{\Pi_{t+1}^* x_{2,t+1}} \right] \right) \tag{4.2.5}$$

$$1 = \theta \Pi_t^{\varepsilon-1} + (1-\theta)(\Pi_t^*)^{1-\varepsilon} \tag{4.2.6}$$

$$\Delta_t = \theta \Pi_t^{\varepsilon} \Delta_{t-1} + (1-\theta)(\Pi_t^*)^{-\varepsilon} \tag{4.2.7}$$

$$g_t = s_{g,t} y_t \tag{4.2.8}$$

$$Z_t = \frac{1}{\beta} \left(\frac{\Pi_t}{\Pi} \right)^{\phi_\pi} \left(\frac{y_t}{y} \right)^{\phi_y} m_t \tag{4.2.9}$$

$$R_t = max[Z_t, 1] \tag{4.2.10}$$

$$y_t = c_t + g_t \tag{4.2.11}$$

$$y_t = \frac{A_t}{\Delta_t} l_t \tag{4.2.12}$$

$$\beta_t = \beta^{1-\rho_b} \beta_{t-1}^{\rho_b} \exp(\sigma_b u_t^b) \tag{4.2.13}$$

$$A_t = A^{1-\rho_a} A_{t-1}^{\rho_a} \exp(\sigma_a u_t^a) \tag{4.2.14}$$

$$s_{g,t} = s_g^{1-\rho_g} s_{g,t-1}^{\rho_g} \exp(\sigma_g u_t^g) \tag{4.2.15}$$

$$m_t = \exp(\sigma_m u_t^m) \tag{4.2.16}$$

Where $u_t^i \sim N(0,1)$. Variables without time subscripts refer to the respective deterministic steady-state value, which we can compute analytically. In the model Equation (4.2.1) is the Euler Equation for consumption. Equation (4.2.2) is the household labor supply. Equations (4.2.3)-(4.2.5) comes from the firms' optimal pricing decision. Equations (4.2.6) and (4.2.7) are the laws of motion for inflation and price dispersion. Equation (4.2.8) is the fiscal rule. Equation (4.2.9) is the Taylor Rule and equation (4.2.10) is the zero lower bound constraint. Equations (4.2.11) and (4.2.12) are the market clearing conditions. Finally, equations (4.2.13) - (4.2.16) describe the shocks.

4.2.3 Calibration

We follow the specification which Fernández-Villaverde et al. (2015) can solve using the most precise Smolyak algorithm[1]. Table 1 presents the parameter calibration.

Table 1 – Parameter Calibration

	Parameter Calibration	**Value**
β	Steady State Discount rate	0.994
χ	Labor disutility	1
φ	Frisch elasticity	1
ε	Elasticity of Substitution	6
θ	Calvo Pricing Parameter	0.75
s_g	Steady State Government Share	0.2
ϕ_π	Taylor Rule: Response to inflation	2.5
ϕ_y	Taylor Rule: Response to output	0.25
Π	Steady State Inflation	1.005
ρ_b	discount rate persistence	0.78
ρ_a	productivity persistence	0.9
ρ_g	government spenditure persistence	0.8
σ_b	discount rate shock volatility	0.0025
σ_a	productivity shock volatility	0.0025
σ_g	fiscal shock volatility	0.0025
σ_m	monetary shock volatility	0.0025

[1] The authors cannot use a more precise version of the Smolyak algorithm, which uses a degree 8 Chebyshev polynomial approximation, to solve for the baseline calibration, due to convergence problems. Instead, in an alternative, more stable, calibration they don't experience this issue. We follow this alternative specification because we think it provides a better comparison for our solution algorithm, since we do not have to worry if the convergence to the solution is well defined

4.2.4 State Space Representation

There are five state variables[2] in the model. We work with them in logs. Thus, a state S_t will be defined by:

$$S_t = (\ln(\Delta_{t-1}), \ln(\beta_t), \ln(m_t), \ln(s_{g,t}), \ln(A_t))$$

For a given point draw of the state space, we need to solve the system (3)-(18), taking as given the current policies guess for each endogenous variables to compute the expectations. In fact, 3 time invariant policy functions ($c_t = c(S_t)$, $\Pi_t = \Pi(S_t)$) and $x_{1,t} = x_1(S_t)$) characterize all the system, i.e., it is possible to recover all the remaining endogenous variables. Again, to avoid having any negative value, we define and update the guesses in the log of the variables. Formally, at a iteration s the guess will be:

$$f^s(S_t) = \begin{bmatrix} \ln c^s(S_t) \\ \ln \Pi^s(S_t) \\ \ln x_1^s(S_t) \end{bmatrix}$$

To compute $\Gamma(f^s(S_t))$ we use the solver presented in the appendix of Fernández-Villaverde et al. (2015). To evaluate the solver we need to compute three integrals.

$$E_t\left[\frac{\beta_{t+1}}{c^s(S_{t+1})\Pi^s(S_{t+1})}\right], E_t[\beta_{t+1}\Pi^s(S_{t+1})^{\varepsilon}x_1^s(S_{t+1})], \theta E_t\left[\beta_{t+1}\frac{\Pi^s(S_{t+1})^{\varepsilon-1}}{\Pi^*(f^s(S_{t+1}))x_2(f^s(S_{t+1}))}\right]$$

Since there are 4 exogenous Gaussian shocks we can use a degree 11 monomial rule from Stroud (1971). The state space $\mathcal{X}$ is constructed to contain 99% of the exogenous shocks' realizations. For instance the bounds for the variable A_t would be $0 \pm 2.58\sigma_A$, where σ_A is the unconditional variance of $\ln(A_t)$ which depends on σ_a and ρ_a. The bound for the endogenous state variable $\ln(\Delta_{t-1})$ is $[\ln(1), \ln(1.005)]$.

4.2.5 Solution Algorithm

We consider a uniform measure on $\mathcal{X}$ and use the RKHS algorithm, with $n = 750$ draws from the state space at each iteration step. The kernel is on the form

$$\mathbf{K}(x, x') = K_\lambda(x, x') \otimes \mathbf{B}$$

[2] Even though the exogenous state variables are auto correlated random variables, it is possible to define the state space only containing exogenous i.i.d. shocks by including the lagged realization of these variables. Here, this is not necessary for the solution procedure and we save in the dimensionality by reducing the problem to 5 state variables.

And for $K_\lambda(x, x')$, we chose the neural network covariance function (see chapter 4 of Williams and Rasmussen (2006)).

$$K_\lambda(\mathbf{x}, \mathbf{y}) = \sigma_f^2 \sin^{-1}\left(\frac{\mathbf{x}'\mathbf{y}}{\sqrt{(\ell^2 + \mathbf{x}'\mathbf{x})(\ell^2 + \mathbf{y}'\mathbf{y})}}\right)$$

The above kernel is non stationary, meaning it does not depend only on the Euclidean distance between points. This characteristic may be important when dealing with the zero lower bound because we expect a sudden change in the behaviour of the policy functions when the constraint starts binding.

There are a total of 6 hyperparameters: two from K_γ, $\gamma = (\ell, \sigma_f^2)$, three from $\mathbf{B}$ and λ_n. The hyperparameters are chosen by the maximum likelihood procedure described in section 2.4.

Finally, we set the converge criterion, $\bar{\varepsilon} = 10^{-6}$ and, at each iteration step, compute the L_2 norm using Monte Carlo integration with $100,000$ points.

To solve the model we start with $\rho_b = 0$, when the ZLB does not bind, and gradually increase this parameter to the desired value using the last solution as an initial guess.

4.2.6 Results

We compare our results with the ones from the Fernández-Villaverde et al. (2015). The authours use the Smolyak Method, which consists in combining a projection into a Chebyshev polynomial basis[3] with grid selection method(see Judd et al. (2014) for details).

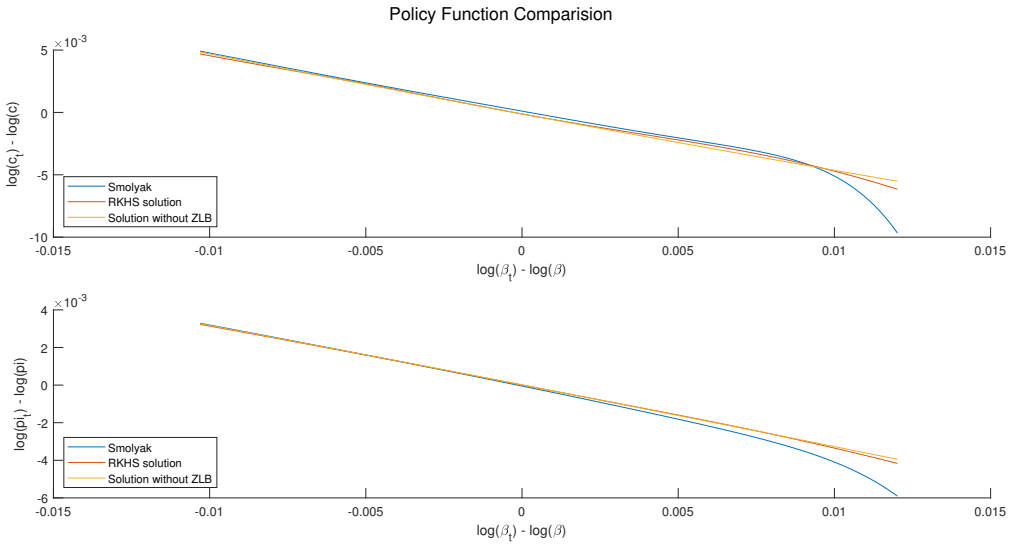
Our results indicates that the solution by Fernández-Villaverde et al. (2015) *over-estimates* the effects of the zero lower bound, both in the extensive and intensive margin. Moreover, we test solutions' accuracy by computing the consumption Euler Equation Errors and show that ours have lower average error.

Figure 1 illustrates the intensive margin. It consists of the plots for consumption and inflation policy functions, where we vary the discount rate shock and keep the other state variables constant on their (deterministic) steady-state values. Our solution presents only a slight detachment from the non-linear solution of the unconstrained model(without the ZLB), whereas the Smolyak solution greatly differs from it.

Figure 2 illustrates the extensive margin. We consider two possible scenarios: one that features combinations of the discount rate shock and fiscal shocks and another that consider combinations of the discount rate and productivity shocks. The other state variables are kept constant in their (deterministic) steady-state valued. We can see that

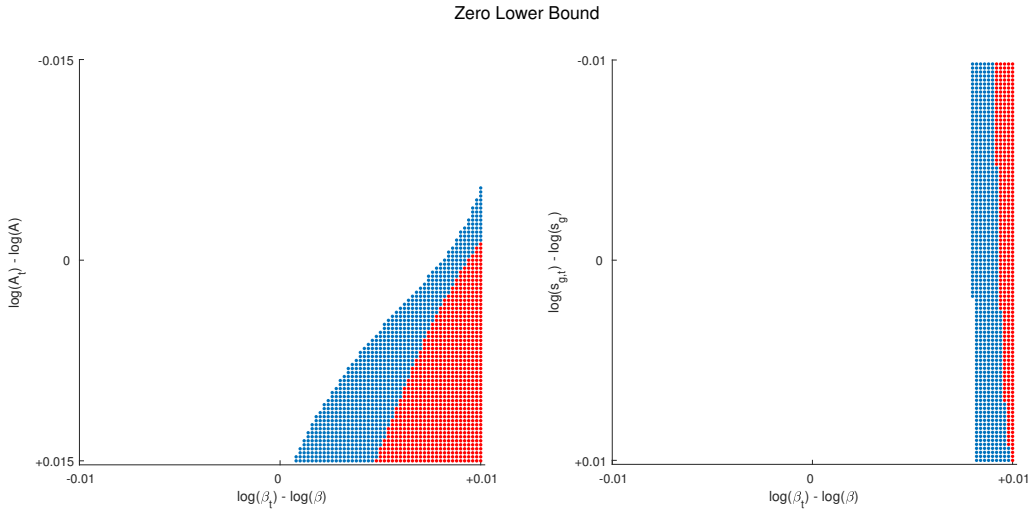
[3] In this particular case, Fernández-Villaverde et al. (2015) use a degree 8 approximation

Figure 1 – Comparison Between Solution Methods



whenever the ZLB binds in our solution, it binds in the the Smolyak's solution. Moreover, the ZLB binds far more often in the last solution.

Figure 2 – Regions where the ZLB is binding



In the red region the ZLB bounds in our solution and in the Smolyak solution. In the blue one it only binds in the Smolyak solution.

4.2.7 Accuracy

To address Accuracy of the model we compute the consumption Euler Equation Errors(EEr). Be $\widehat{f} = (\widehat{c}, \widehat{\Pi}, \widehat{x}_1)$ the solution obtained. Given a state $S_t \in \mathcal{X}$, the error is defined as:

$$\text{EEr} = \log_{10}\left(\frac{\left|\widehat{c}(S_t) - R(\widehat{f}(S_t))E_t\left[\frac{\beta_{t+1}}{\widehat{c}(S_{t+1})\widehat{\Pi}(S_{t+1})}\right]\right|}{\widehat{c}(S_t)}\right)$$

Table 2 presents the calculated EEr. Our solution achieves overall smaller EEr. Moreover, as show in figure 3, the gain in performance is achieved exactly when the ZLB binds on the Smolyak Solution. In this case, the Euler Equation Erros are computed by varying only the discount rate shock, exactly as in Figure 1.

Figure 3 – Consumption Euler Equation Errors

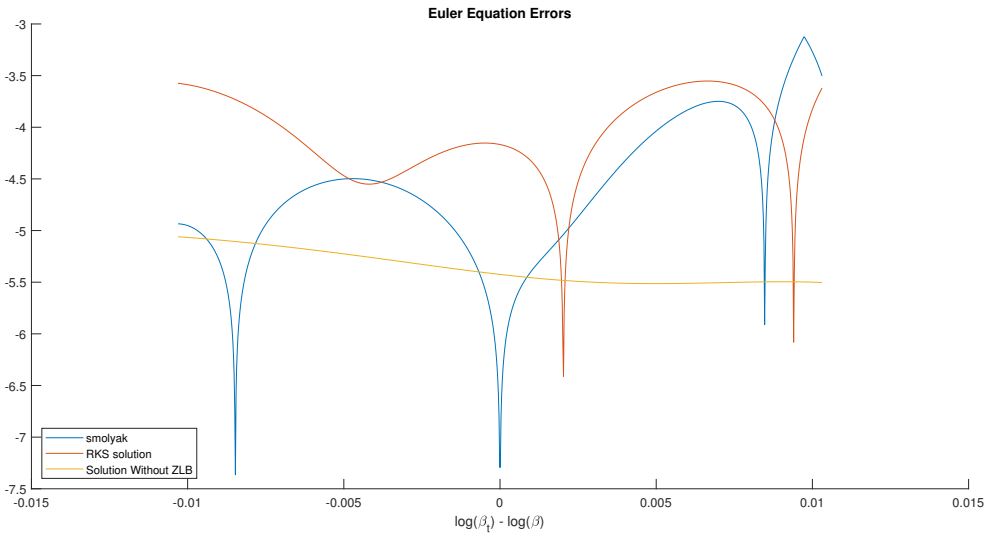


Table 2 – Euler Equation Error and ZLB frequency

Algorithm	%ZLB	EEr Mean	EEr Min	EEr Max
RKHS	7.4%	-3.9187	-9.2084	-2.0011
Smolyak	14.87%	-3.3007	-8.7431	-1.9089

We conjecture what is driving the contrast in the solutions. If the true solution has a sharp change in the derivatives of the policy functions around the binding region, a polynomial basis approximation will have a problem fitting such functions. As a consequence, if the basis overestimates the region in the space which the ZLB binds, it will also overestimate the impact of the ZLB. Specifically, when the ZLB binds, due to the persistence of the shock, it will take a while to move out of the region, a period when the interest rate will be held constant. Since the variations in the interest rate that stabilize the economy, the impact of the ZLB will be greater as the binding region is larger(up to a point where there is no equilibrium with price stability), i.e., the policy function will change more in relation to the ones from a unconstrained economy,

5 Conclusion

We propose a solution framework that combines approximation with estimation techniques from high-dimensional statistics to solve non-linear DSGE models. Our framework allows to deal with the curse of dimensonality, while still maintaining flexibility in terms of the choice of approximation. Moreover, we show that this algorithm has some theoretical guarantees regarding convergence and approximation quality.

To better illustrate these points, we apply our solution to a New Keynesian Model with a ZLB. We show that our solution is more accurate than the usual Smolyak method, which employ an approximation using a Chebyshev polynomial basis. In fact, our results also indicates that the Smolyak's solution overestimates the impact of the ZLB. We conjecture that these results are likely being driven by some sort of kinks in the policies function around the binding region, rendering good polynomial approximations harder to obtain.

One further step would involve constructing an easy to use toolbox, including different options for basis functions, kernels and general options within our framework. Moreover, our framework also benefits from the recent tools of parallelization (see Fernández-Villaverde and Valencia (2018)), which opens the possibility of optimizing the algorithms.

Another direction would involve extending our framework to models where the state space is infinite dimensional, such as heterogeneous agents model with whole distributions as a state variables. In this case the work would involve reducing the dimensionality of the state variables within the model, to be able to compute the value function or policy function iteration operator for a given state realization.

Bibliography

Aiyagari, S. R. (1994). Uninsured idiosyncratic risk and aggregate saving. *The Quarterly Journal of Economics 109*(3), 659–684.

Albertini, J. and A. Poirier (2015). Unemployment benefit extensions at the zero lower bound. *Review of Economic Dynamics 18*(4), 733–751.

Alvarez, M. A., L. Rosasco, and N. D. Lawrence (2012). Kernels for vector-valued functions: A review. *Foundations and Trends® in Machine Learning 4*(3), 195–266.

Arcidiacono, P., P. Bayer, F. A. Bugni, and J. James (2013). Approximating high-dimensional dynamic models: Sieve value function iteration. In *Structural Econometric Models*, pp. 45–95. Emerald Group Publishing Limited.

BoraUgan Aruoba, S., P. Cuba-Borda, and F. Schorfheide (2017). Macroeconomic dynamics near the zlb: A tale of two countries. *The Review of Economic Studies 85*(1), 87–118.

Briol, F.-X., C. J. Oates, M. Girolami, M. A. Osborne, D. Sejdinovic, et al. (2019). Probabilistic integration: A role in statistical computation? *Statistical Science 34*(1), 1–22.

Brumm, J. and S. Scheidegger (2017). Using adaptive sparse grids to solve high-dimensional dynamic models. *Econometrica 85*(5), 1575–1612.

Bühlmann, P. and S. Van De Geer (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

Carmeli, C., E. De Vito, A. Toigo, and V. Umanitá (2010). Vector valued reproducing kernel hilbert spaces and universality. *Analysis and Applications 8*(01), 19–61.

Chen, J. and Z. Chen (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika 95*(3), 759–771.

Chen, Z., B. Wang, and A. N. Gorban (2017). Multivariate gaussian and student $-t$ process regression for multi-output prediction. *arXiv preprint arXiv:1703.04455*.

Coleman, W. J. (1990). Solving the stochastic growth model by policy-function iteration. *Journal of Business & Economic Statistics 8*(1), 27–29.

Combettes, P. L., S. Salzo, and S. Villa (2018). Regularized learning schemes in feature banach spaces. *Analysis and Applications 16*(01), 1–54.

Constantine, P., Y. Marzouk, C. Prieur, and O. Zahm (2018). Gradient-based dimension reduction of multivariate vector-valued functions. *arXiv preprint arXiv:1801.07922*.

Debortoli, D., J. Galí, and L. Gambetti (2019). On the empirical (ir)relevance of the zero lower bound constraint. Working Paper 25820, National Bureau of Economic Research.

Fernández-Villaverde, J., G. Gordon, P. Guerrón-Quintana, and J. F. Rubio-Ramirez (2015). Nonlinear adventures at the zero lower bound. *Journal of Economic Dynamics and Control 57*, 182–204.

Fernández-Villaverde, J., J. F. Rubio-Ramírez, and F. Schorfheide (2016). Solution and estimation methods for dsge models. In *Handbook of Macroeconomics*, Volume 2, pp. 527–724. Elsevier.

Fernández-Villaverde, J. and D. Z. Valencia (2018). A practical guide to parallelization in economics. Working Paper 24561, National Bureau of Economic Research.

Hirose, Y. and T. Sunakawa (2019). Review of solution and estimation methods for nonlinear dynamic stochastic general equilibrium models with the zero lower bound. *The Japanese Economic Review 70*(1), 51–104.

Judd, K. L., L. Maliar, S. Maliar, and R. Valero (2014). Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain. *Journal of Economic Dynamics and Control 44*, 92–123.

Kunsch, R. J. (2016). Bernstein numbers and lower bounds for the monte carlo error. In *Monte Carlo and Quasi-Monte Carlo Methods*, pp. 471–488. Springer.

Kunsch, R. J. (2017). High-dimensional function approximation: Breaking the curse with monte carlo methods. *arXiv preprint arXiv:1704.08213*.

Ljungqvist, L. and T. J. Sargent (2004). *Recursive macroeconomic theory*. MIT press.

Maliar, L. and S. Maliar (2014). Numerical methods for large-scale dynamic economic models. In *Handbook of computational economics*, Volume 3, pp. 325–477. Elsevier.

Maliar, L. and S. Maliar (2015). Merging simulation and projection approaches to solve high-dimensional problems with an application to a new keynesian model. *Quantitative Economics 6*(1), 1–47.

Micchelli, C. A. and M. Pontil (2005). Kernels for multi–task learning. In *Advances in neural information processing systems*, pp. 921–928.

Montrucchio, L. (1998). Thompson metric, contraction property and differentiability of policy functions. *Journal of economic behavior & organization 33*(3-4), 449–466.

Newey, W. (1991). Uniform convergence in probability and stochastic equicontinuity. *Econometrica 59*(4), 1161–67.

Novak, E. and H. Woźniakowski (2009). Approximation of infinitely differentiable multivariate functions is intractable. *Journal of Complexity 25*(4), 398–404.

Novak, E. and H. Wozniakowski (2010). *Tractability of Multivariate Problems, Volume II, Standard Information for Functionals*. European Mathematical Society.

Novak, E. and H. Wozniakowski (2012). *Tractability of Multivariate Problems, Volume III, Standard Information for Operators*. European Mathematical Society.

Paciorek, C. J. and M. J. Schervish (2004). Nonstationary covariance functions for gaussian process regression. In *Advances in neural information processing systems*, pp. 273–280.

Pál, J. and J. Stachurski (2013). Fitted value function iteration with probability one contractions. *Journal of Economic Dynamics and Control 37*(1), 251–264.

Petrov, P. and V. Petrov (1995). *Limit Theorems of Probability Theory: Sequences of Independent Random Variables*. Oxford science publications. Clarendon Press.

Richter, A. W., N. A. Throckmorton, and T. B. Walker (2014). Accuracy, speed and robustness of policy function iteration. *Computational Economics 44*(4), 445–476.

Rudin, W. (2006). *Real and complex analysis*. Tata McGraw-Hill Education.

Rust, J. (1997). Using randomization to break the curse of dimensionality. *Econometrica 65*(3), 487–516.

Scheidegger, S. and I. Bilionis (2019). Machine learning for high-dimensional dynamic stochastic economies. *Journal of Computational Science*.

Steinwart, I. and A. Christmann (2008). *Support vector machines*. Springer Science & Business Media.

Stroud, A. (1971). *Approximate calculation of multiple integrals*. Prentice-Hall series in automatic computation. Prentice-Hall.

Williams, C. K. and C. E. Rasmussen (2006). *Gaussian processes for machine learning*. MIT press.

Appendix

APPENDIX A – Proofs

A.1 Proof of Proposition 1

We now proceed with the proof of proposition 1. We first start proving that the operator Ω_J is also a contraction.

Lemma 1. *For each $J \in \{1, 2, \dots\}$, Ω_J is a contraction mapping.*

Proof. Fix $J \in \{1, 2, \dots\}$. We have by assumption that Γ is a contraction and we also know that the nearest point projection onto any closed and convex subset of a Hilbert Space is nonexpansive, then the following inequality holds for some $\kappa \in (0, 1)$:

$$\|\Omega_J(f) - \Omega_J(g)\| = \|\Pi_J(\Gamma(f)) - \Pi_J(\Gamma(g))\| \leq \|\Gamma(f) - \Gamma(g))\| \leq \kappa\|f - g\|.$$

□

Define f_J^* as the unique fixed point of the operator Ω_J, i.e, $f_J^* = \Omega_J(f_J^*)$. We now prove that the sequence $\{f_J^*\}_{J=1}^{\infty}$ actually converges to f^*.

We have that $\{\Omega_J\}_{J=1}^{\infty}$ converges pointwise to Γ, moreover,$\forall J$ Ω_J has a contraction modulus $\kappa \in (0, 1)$, which is also a contraction modulus of Γ, with this we can get the following result[1]:

Lemma 2. *The sequence of fixed points $\{f_J^*\}_{J=1}^{\infty}$ converges to f^*, the fixed point of Γ. Moreover,*

$$\|f_J^* - f^*\| \leq \frac{\|f^* - \Pi_j(f^*)\|}{1 - \underline{\kappa}}$$

Proof. We begin with the following inequality:

$$\|f_J^* - f^*\| = \|\Omega_J(f_J^*) - \Gamma(f^*)\| \leq \|\Omega_J(f_J^*) - \Omega_J(f^*)\| + \|\Omega_J(f^*) - \Gamma(f^*)\|$$

For all J, Ω_J is a contraction mapping, then[2] :

$$\|\Omega_J(f_J^*) - \Omega_J(f^*)\| \leq \underline{\kappa}\|f_J^* - f^*\|$$

Combining these two inequalities we get:

$$\|f_J^* - f^*\| \leq \frac{\|\Omega_J(f^*) - \Gamma(f^*)\|}{1 - \underline{\kappa}} = \frac{\|f^* - \Pi_j(f^*)\|}{1 - \underline{\kappa}}$$

[1] Given the previous lemmas, Proposition 1 will be a particular case of Lemma 2.1 from Rust (1997) applied to the projection operator instead;

[2] This comes as a corollary from lemma 1

Since by assumption 3 $\{\Omega_J\}_{J=1}^{\infty}$ converges pointwise to Γ, so we can conclude:

$$\lim_{J\to\infty} \|f_J^* - f^*\| = 0$$

□

A.2 Proof of proposition 2

The first lemma will ensure that the estimator $\widehat{\Omega}_{n,J}$ will be the stochastic analogous of a contraction.

Lemma 3. $\forall f, g \in \mathcal{H}$, $\exists\, \kappa \in (0,1)$, *such that,*

$$\|\widehat{\Omega}_{n,J}(f) - \widehat{\Omega}_{n,J}(g)\|_{\mathcal{H}} \leq (o_p(1) + \kappa)\|f - g\|_{\mathcal{H}}$$

Proof. If $f = g$ almost everywhere, the result is trivial since $\|\widehat{\Omega}_{n,J}(f) - \widehat{\Omega}_{n,J}(g)\| = 0$. Now assume f is not everywhere equal g. By triangle inequality:

$$\|\widehat{\Omega}_{n,J}(f) - \widehat{\Omega}_{n,J}(g)\|_{\mathcal{H}} \leq \|\widehat{\Omega}_{n,J}(f) - \Omega_J(f)\|_{\mathcal{H}} + \|\widehat{\Omega}_{n,J}(g) - \Omega_J(g)\|_{\mathcal{H}} + \|\Omega_J(f) - \Omega_J(g)\|_{\mathcal{H}}$$

Dividing and multiplying the first two terms by $\|f - g\|_{\mathcal{H}}$ and using the contraction property of Ω_J, we get that $\exists \kappa \in (0,1)$, such that:

$$\|\widehat{\Omega}_{n,J}(f) - \widehat{\Omega}_{n,J}(g)\|_{\mathcal{H}} \leq \left[\frac{\|\widehat{\Omega}_{n,J}(f) - \Omega_J(f)\|_{\mathcal{H}}}{\|f - g\|_{\mathcal{H}}} + \frac{\|\widehat{\Omega}_{n,J}(g) - \Omega_J(g)\|_{\mathcal{H}}}{\|f - g\|_{\mathcal{H}}} + \kappa\right] \|f - g\|_{\mathcal{H}}$$

Define $B_N = \left[\frac{\|\widehat{\Omega}_{n,J}(f) - \Omega_J(f)\|_{\mathcal{H}}}{\|f - g\|_{\mathcal{H}}} + \frac{\|\widehat{\Omega}_{n,J}(g) - \Omega_J(g)\|_{\mathcal{H}}}{\|f - g\|_{\mathcal{H}}} + \kappa\right]$. Since both $\widehat{\Omega}_J(f) \xrightarrow{p} \Omega_J(f)$ and $\widehat{\Omega}_J(g) \xrightarrow{p} \Omega_J(g)$, we can apply the continuous mapping to get that:

$$B_n = [o_p(1) + o_p(1) + k] = [o_p(1) + k]$$

□

The following lemma establishes that each step of the iterative procedure is also consistent.

Lemma 4. $\forall s \geq 1$ $\widehat{\Omega}_{n,J}^{(s)}(f) \xrightarrow{p} \Omega_J^{(s)}(f)$.

Proof. The proof is by induction. Assume the property holds $s \geq 1$, we show that it holds for s+1. By the triangle inequality and the definition of $\widehat{\Omega}^{(s)}$:

$$\|\widehat{\Omega}_{n,J}^{(s+1)}(f) - \Omega_J^{(s+1)}(f)\| \leq \|\widehat{\Omega}_{n,J}(\widehat{\Omega}_{n,J}^{(s)}(f)) - \widehat{\Omega}_{n,J}(\Omega_J^{(s)}(f))\| + \|\widehat{\Omega}_{n,J}(\Omega_J^{(s)}(f)) - \Omega(\Omega_J^{(s)}(f))\|$$

Applying lemma 3, $\exists \kappa \in (0,1)$ and B_N, such that, B_N converges in probability to κ as $N \to \infty$ and:

$$\|\widehat{\Omega}_{n,J}^{(s+1)}(f) - \Omega_J^{(s+1)}(f)\|_{\mathcal{H}} \leq B_N \|\widehat{\Omega}_{n,J}^{(s)}(f) - \Omega_J^{(s)}(f)\|_{\mathcal{H}} + \|\widehat{\Omega}_{n,J}(\Omega_J^{(s)}(f)) - \Omega(\Omega_J^{(s)}(f))\|_{\mathcal{H}}$$

Since $B_n \xrightarrow{p} k$ and $\widehat{\Omega}_{n,J}^{(s)}(f) \xrightarrow{p} \Omega_J^{(s)}(f)$ by the induction hypothesis, then $B_N(\widehat{\Omega}_{n,J}^{(s)}(f) - \Omega_J^{(s)}(f)) \xrightarrow{p} 0$. Moreover,by assumption 4 $\forall f \in \mathcal{H}$, $\widehat{\Omega}_{n,J}(f) \xrightarrow{p} \Omega_J(f)$. With this we can conclude that:

$$\widehat{\Omega}_{n,J}^{(s+1)}(f) \xrightarrow{p} \Omega_J^{(s+1)}(f)$$

□

The next lemma extends the contraction result for all the iteration steps.

Lemma 5. $\forall f, g \in \mathcal{H}$, $\exists\, \kappa \in (0,1)$, *such that,*

$$\|\widehat{\Omega}_{n,J}^{(s)}(f) - \widehat{\Omega}_{n,J}^{(s)}(g)\|_{\mathcal{H}} \leq (o_p(1) + \kappa)\|f - g\|_{\mathcal{H}}$$

Proof. The proof is analogous to the case of lemma 3, since $\Omega_J^{(s)}$ is a contraction. □

Similarly as the deterministic case, we can provide a strong convergence on any compact subset of $\mathcal{H}$.

Lemma 6. *(Uniform convergence in probability) Be $\mathcal{F}$ a compact subset of $\mathcal{H}$, $\forall s \geq 1$ we have* $\sup_{f \in \mathcal{F}} \|\widehat{\Omega}_{n,J}^{(s)}(f) - \Omega_J^{(s)}(f)\| \xrightarrow{p} 0$

Proof. Combining the results from lemma 4 and 5 with the fact that $\Omega_J^{(s)}$ is equicontinuous (because it is a contraction) we are able to apply corollary 2.2 from Newey (1991) to get the desired result. □

We are now able to prove the main result of the upper bound for the error.

Lemma 7. *Be f^* the fixed point of Γ, then*

$$\|\widehat{\Omega}_{n,J}^{(s)}(f) - f^*\| \leq o_p(1) + \underline{\kappa}^s \|f - f^*\| + \frac{\|f^* - \Pi_j(f^*)\|}{1 - \underline{\kappa}}$$

Proof. By triangle inequality:

$$\|\widehat{\Omega}_{n,J}^{(s)}(f) - f_J^*\| \leq \|\widehat{\Omega}_{n,J}^{(s)}(f) - \Omega_J^{(s)}(f)\| + \|\Omega_J^{(s)}(f) - f_J^*\| + \|f_J^* - f^*\|$$

Applying lemma 6, we conclude that the first term on the right hand side is $o_p(1)$, $n \to \infty$, the second term comes from the fact that Ω_J is a contraction, and admits $\underline{k}$ as contraction modulus. The last term comes from Proposition 1. □

As n increases, the first term of the right hand side of the inequality from lemma 7 goes to 0. The second term goes to 0 as s increases,due to the properties of the contraction. The last terms goes to 0 as J by assumption 3. Thus we conclude this subsection with the last lemma.

Lemma 8. *Be $f \in \mathcal{H}$ and f^* the fixed point of Γ, then $\widehat{\Omega}^{(s)}_{n,J}(f) \xrightarrow{p} f^*$ as $n \to \infty$ followed by $s \to \infty$ and $J \to \infty$.*

A.3 Proof of proposition 3

Proof. We first prove pointwise convergence. $\mathcal{G}$ is a subset of $\mathcal{F}$, then it is separable since $\mathcal{H}$ is separable. This fact ensures that $\mathcal{G}$ admits an orthonormal basis (ONB). Moreover since this set is dense on $\mathcal{H}$, any ONB basis of $\mathcal{G}$ is also an ONB of $\mathcal{H}$. Be $\{e_J\}_{J=1}^{\infty}$ one of those ONB. Without loss of generality we have $\mathcal{G}_J = \text{span}(e_1, e_2, ..., e_J)$. We can apply the Hilbert Space Projection Theorem to conclude that:

$$\Omega_J(f) = \Pi_j(\Gamma(f)) = \sum_{i=1}^{J} \langle \Gamma(f), e_i \rangle e_i$$

$\{e_J\}_{J=1}^{\infty}$ is also a basis for $\mathcal{H}$, then:

$$\Gamma(f) = \sum_{i=1}^{\infty} \langle \Gamma(f), e_i \rangle e_i$$

Then we have:

$$\|\Omega_J(f) - \Gamma(f)\|^2 = \| \sum_{i=J+1}^{\infty} \langle \Gamma(f), e_i \rangle e_i \|^2 = \sum_{i=J+1}^{\infty} |\langle \Gamma(f), e_i \rangle|^2$$

Where the last inequality is due to the Parseval's identity(Rudin (2006) pg. 85). Moreover this identity also implies that $\|\Gamma(f)\|^2 = \sum_{i=1}^{\infty} |\langle \Gamma(f), e_i \rangle|^2$, so we know this series converge(since the norm is well defined) and as a corollary we have:

$$\lim_{J \to \infty} \sum_{i=J+1}^{\infty} |\langle \Gamma(f), e_i \rangle|^2 = \lim_{J \to \infty} \|\Omega_J(f) - \Gamma(f)\|^2 = 0$$

We conclude that $\{\Omega_J\}_{J=1}^{\infty}$ converges pointwise to Γ. Because as a corollary of lemma 1 $\{\Omega_J\}_{J=1}^{\infty}$ is equicontinuous, we get the desired result. □

A.4 Proof of Proposition 4

Be $\{x_i\}_{i=1}^{n}$ an i.i.d. sample from $\mathcal{X}$ and $\widehat{\beta}_J(f)$ obtained from the LASSO estimation, for simplicity we will just refer to it as $\widehat{\beta}_J$ and will not work with the standardized variables as in section 2. We define the sample prediction error by

$$\varepsilon = \Psi(\mathbf{x})\beta_J - \Psi(\mathbf{x})\widehat{\beta}_J$$

Where $\Psi(\mathbf{x})$ is the $n \times J$ matrix consisting of the basis function evaluated at the sample.The next lemma bounds the sample prediction error.

Lemma 9.

$$\frac{1}{n}\|\varepsilon\|_2^2 \leq \left(\max_{j\in\{1,\dots,J\}} \frac{2}{n}\left|\varepsilon'\psi_j(\boldsymbol{x})\right|\right)\|\widehat{\beta}_J - \beta_J\|_{\ell 1} + \lambda_n(\|\beta_J\|_{\ell 1} - \|\widehat{\beta}_J\|_{\ell 1})$$

Proof. Since $\widehat{\beta_J}$ is the maximizer in the LASSO optimization problem, then(see lemma 6.1 of Bühlmann and Van De Geer (2011)):

$$\frac{1}{n}\|\varepsilon\|_2^2 \leq \left|\frac{2}{n}\varepsilon'\Psi(\mathbf{x})(\widehat{\beta}_J - \beta_J)\right| + \lambda_n(\|\beta_J\|_{\ell 1} - \|\widehat{\beta}_J\|_{\ell 1})$$

Note that, by Holder's inequality,

$$\left|\frac{2}{n}\varepsilon'\Psi(\mathbf{x})(\widehat{\beta}_J - \beta_J)\right| \leq \left(\max_j \frac{2}{n}\left|\varepsilon'\psi_j(\mathbf{x})\right|\right)\|\beta_J - \widehat{\beta}_J\|_{\ell 1}$$

Combining these two statements we get the desired result. □

The next lemma will bound the probability of the first term in the inequality presented in lemma 9.

Lemma 10. *For some constant C_q:*

$$\mathbb{P}\left(\max_{j\in\{1,\dots,J\}} \frac{2}{n}\left|\varepsilon'\psi_j(\boldsymbol{x})\right| \geq \lambda\right) \leq \frac{JC_q}{n^{\frac{q}{2}}\lambda^q}$$

Where the probability is taken with respect to μ.

Proof. By assumption 5, $\mathbb{E}[|\varepsilon|^{2q}] < \infty$. Then, applying Holder's inequality we get that $\forall j \in \{1, \dots, J\}$,

$$\mathbb{E}[|\varepsilon\psi_J|^q] \leq \infty$$

Then by the union bound, followed by Markov's Inequality and Marcinkiewicz-Zygmund inequality (see Petrov and Petrov (1995)) we have:

$$\begin{aligned}\mathbb{P}\left(\frac{2}{n}\max_j\left|\varepsilon'\psi_j(\mathbf{x})\right| \geq \lambda\right) &\leq J\max_j \mathbb{P}\left(\frac{2}{n}\left|\varepsilon'\psi_j(\mathbf{x})\right| \geq \lambda\right) \\ &\leq \frac{2J}{\lambda^q n^q}\mathbb{E}\left[\left|\sum_{i=1}^n \varepsilon_i\psi_J(x_i)\right|^q\right] \leq C_q n^{\frac{q}{2}}\frac{J}{\lambda^q n^q} = \frac{JC_q}{\lambda^q n^{\frac{q}{2}}}\end{aligned}$$

□

Using Lemmas 9 and 10 we are able to prove prediction consistency for the LASSO. The next lemma is based on Corollary 6.1 of Bühlmann and Van De Geer (2011).

Lemma 11. *For some $t > 0$, let the regularization parameter be:*

$$\lambda_n = t\frac{J^{\frac{1}{q}}}{\sqrt{N}}$$

Then with probability at least $1 - \alpha$, where, for some constant C_q,

$$\alpha = \frac{C_q}{t^q}$$

We have:

$$\frac{1}{n}\|\varepsilon\|_2^2 \leq 2\lambda_n\|\beta_J\|_{\ell 1}$$

Proof. By lemma 10 we have that:

$$\mathbb{P}\left(\max_{j\in\{1,\dots,J\}} \frac{2}{n}\left|\varepsilon'\psi_j(\mathbf{x})\right| \geq \lambda\right) \leq \frac{JC_q}{n^{\frac{q}{2}}\lambda^q}$$

For some constant C_q. Pick $\lambda_n = t\frac{J^{\frac{1}{q}}}{\sqrt{N}}$ for some $t > 0$. Defined the probability α by:

$$\alpha := \mathbb{P}\left(\max_{j\in\{1,\dots,J\}} \frac{2}{n}\left|\varepsilon'\psi_j(\mathbf{x})\right| \geq \lambda_n\right) \leq \frac{JC_q}{n^{\frac{q}{2}}}\frac{n^{\frac{q}{2}}}{Jt^q} = \frac{C_q}{t^q}$$

Then, applying lemma 9, with probability $1 - \alpha$ we have:

$$\begin{aligned}
\frac{1}{n}\|\varepsilon\|_2^2 &\leq \left(\max_{j\in\{1,\dots,J\}} \frac{2}{n}\left|\varepsilon'\psi_j(\mathbf{x})\right|\right)\|\widehat{\beta}_J - \beta_J\|_{\ell 1} + \lambda_n(\|\beta_J\|_{\ell 1} - \|\widehat{\beta}_J\|_{\ell 1}) \\
&\leq (\lambda_n)\,\|\widehat{\beta}_J - \beta_J\|_{\ell 1} + \lambda_n(\|\beta_J\|_{\ell 1} - \|\widehat{\beta}_J\|_{\ell 1}) = \lambda_n\left(\|\widehat{\beta}_J - \beta_J\|_{\ell 1} + \|\beta_J\|_{\ell 1} - \|\widehat{\beta}_J\|_{\ell 1}\right) \\
&\leq 2\lambda_n\|\beta_J\|_{\ell 1}
\end{aligned}$$

Where the last inequality comes from the triangle inequality. □