Nonparametric Bayes Subject to Overidentified Moment Conditions *

A. Ronald Gallant
Penn State University

First draft: January 1, 2020
This draft: July 23, 2021

Forthcoming, *Journal of Econometrics*

Paper:www.aronaldg.org/papers/npb.pdf

Code:http:www.aronaldg.org/webfiles/npb

Slides:www.aronaldg.org/papers/npbclr.pdf

*Address correspondence to A. Ronald Gallant, P.O. Box 659, Chapel Hill NC 27514, USA, phone 919-428-1130; email aronldg@gmail.com.
I am indebted to Shengbo Zhu for calling my attention to Zappa, Holmes-Cerfon, and Goodman (2018).
© 2020 A. Ronald Gallant

Abstract

Nonparametric Bayesian estimation subject to overidentified moment equations is a challenge because the support of the posterior is a manifold of lower dimension than the number of model parameters. The manifold therefore has Lebesgue measure zero thus inhibiting the use of the most commonly used Bayesian estimation method: MCMC (Markov Chain Monte Carlo). This study proposes an effective MCMC algorithm and algorithms for estimating scale and the normalizing constant. The algorithms are illustrated with two illustrative applications.

Keywords and Phrases: Method of moments, Bayesian inference

JEL Classification: C11, C14, C15, C32, C36, C58

1 Introduction

An appealing approach to statistical analysis is to set forth a sieve likelihood

$$f(y \,|\, x, \rho) = \prod_{t=1}^{n} f(y_t \,|\, x_{t-1}, \rho), \tag{1}$$

where y_t is a column vector and x_{t-1} is a matrix of exogenous and predetermined variables with a fixed number of rows and a number of columns that are either fixed, such as in a VAR or a cross-sectional application, or that are increasing with t, such as in a VAR-GARCH model. The y and x represent objects that contain the observed y_t and x_{t-1}. The parameters of the sieve are elements of the vector ρ whose dimension is determined by the order of the sieve. An example is the SNP-ARCH sieve derived from the Hermite polynomials (Gallant and Tauchen, 2017), which is the one used here in our illustrative applications.

The parameters in (1) are to be estimated by Bayesian methods subject to moment conditions[1]

$$0 = q(\rho, \theta) = \frac{1}{n} \sum_{t=1}^{n} \int m(y, x_{t-1}, \rho, \theta) f(y \,|\, x_{t-1}, \rho)\, dy, \quad q \in \mathbb{R}^m \tag{2}$$

support conditions

$$h(\rho, \theta) > 0, \quad h \in \mathbb{R}^l \tag{3}$$

and a prior

$$\pi(\rho, \theta). \tag{4}$$

Letting[2] $\mathsf{x} = (\rho, \theta)$, the support of the posterior is the manifold

$$M = \left\{ \mathsf{x} \in \mathbb{R}^{d_a} : \; q_i(\mathsf{x}) = 0, \; i = 1, \ldots, m, \; h_j(\mathsf{x}) > 0, \; j = 1, \ldots, l \right\} \tag{5}$$

The sieve parameters ρ are induced in $q(\rho, \theta)$ by the integration but they may also appear explicitly in $m(y_t, x_{t-1}, \rho, \theta)$. The parameters θ are those that appear only in the moment functions $m(y_t, x_{t-1}, \rho, \theta)$. We assume overidentification, i.e., that the dimension m of $q(\rho, \theta)$

[1] One can integrate with respect to the distribution of x_{t-1} rather than the empirical distribution of x_{t-1} as in (2) if it is available.

[2] In this paper, sans serif x and y are distinguished from italic x and y; the former referring to parameters and the later to data. This is to maintain compatibility with both econometric conventions and numerical analysis conventions.

is larger than the dimension of θ. The just identified case, where the dimensions are equal, is not an interesting mathematical challenge.[3] The overidentified case estimated by Bayesian methods is interesting because the support of the posterior density is singular with respect to Lebesgue measure (Born, Shephard, and Solgi, 2018). A solution to the computational problem requires notions from geometric measure theory.

The economic relevance of what is proposed here is primarily methodological. GMM (Generalized Method of Moments) is the leading frequentist approach to inference for partially specified models. For those instances where the use of prior information is desired, it is of value to have a viable Bayesian counterpart. The obvious approach is to emulate GMM directly and estimate θ and those elements of ρ upon which $m(y_t, x_{t-1}, \rho, \theta)$ depends using the MCMC-GMM method proposed by Chernozhukov and Hong (2003). MCMC-GMM can be formally justified in the Bayesian context (Gallant, 2016). But it has some unappealing features: A Jacobian term is missing if one follows the Chernozhukov and Hong (2003) recipe; its derivation takes application specific, human, analytical effort to obtain (Gallant, 2020). A continuously updated weighting matrix is required, which use detracts from the numerical stability of the Chernozhukov and Hong (2003) MCMC algorithm. A requisite distributional assumption, while implied by asymptotics under weak regularity conditions (Gallant, 2020), nonetheless takes effort to verify in finite samples and does fail dramatically in some examples (Gallant, 2020). Of the compromises entailed by use of MCMC-GMM as a Bayesian estimator, the most serious is the degradation caused by the continuously updated weighting matrix of which the use thereof is a logical necessity in the Bayesian context. In addition, one is needlessly disregarding the information in the data regarding $f(y \mid x, \rho)$.

There is a literature that addresses estimation of (1) subject to (2) that is summarized in Born, Shephard, and Solgi (2018) and in Schennach (2005). Of this literature, there are four papers that are directly relevant here: Born, Shephard, and Solgi (2018), Shin (2015), Schennach (2005), and Gallant, Hong, Leung, and Li (2019).

The first two papers, Born, Shephard, and Solgi (2018) and Shin (2015) solve the computational problem exactly using MCMC methods, but under restrictions on the likelihood

[3]The algorithm of Subsection 2.1 works for just identified models but is inefficient for that purpose due to unnecessary linear algebra and nonlinear equation solving; see Subsection 2.5.

(1). Born, Shephard, and Solgi require that (1) has discrete support, which makes (2) a sum involving probability weights and their corresponding support. Their innovation is to derive a Jacobian term from geometric measure theoretic considerations that enables an MCMC sampler. Their paper contains numerous examples. Shin presumes that (1) is a mixture of specific parametric distributions with random weights drawn from a discrete distribution. The constraint (2) becomes a constraint on the discrete distribution of the random weights. His examples are from macro economics.

The second two papers, Schennach (2005) and Gallant, Hong, Leung, and Li (2019) propose approximate MCMC methods based on asymptotics that cannot impose (2) exactly in finite samples and therefore cannot exactly restrict posterior draws to the manifold M. The method of Gallant, Hong, Leung, and Li (2019) is simple and easily implemented although the mathematics to verify the asymptotics are involved. It is based on a prior $\pi_\lambda(\mathsf{x})$ that increasingly penalizes deviations of x from M as λ increases. One uses MCMC and increases λ until the MCMC chain just fails to mix, as must happen for large enough λ according to the results of Born, Shephard, and Solgi (2018). To have a name, we shall call it the λ-prior method. It is of value in the present context because the method proposed here requires a starting value of x for its MCMC chain whose distance from M is within a small tolerance. The λ-prior method provides this starting value.[4]

To compute estimates, we rely on an amazingly innovative MCMC algorithm due to Zappa, Holmes-Cerfon, and Goodman (2018). Their innovation is to use a nonlinear equation solver twice during the course of computations to impose a detailed balance condition on draws restricted to the manifold M.

To compute standard deviations we rely on a Fast Marching Algorithm (Memoli and Sapiro, 2001) when x has dimension of five or less and on Dijkstra's Algorithm (Dijkstra, 1959) for higher dimensions. For the normalizing constant we rely on an algorithm due to Zappa, Holmes-Cerfon, and Goodman (2018).

Code, including a User's Guide, is at `http://www.aronaldg.org/webfiles/npb`. We proceed to a description of the proposed methodology.

[4]Suggestions for choosing λ to use the λ-prior method as an estimator are in Subsection 3.2.

2 The Algorithms

2.1 The MCMC Surface Sampling Algorithm

Traditional econometric notation regards the italic letters y_t and x_{t-1} as observed data representing dependent variables and explanatory variables, respectively. Similarly for y and x that here represent containers of such data. The Surface Sampling Algorithm uses these letters to represent MCMC draws and proposals that, from our perspective, are draws and proposals of (ρ, θ). We resolve this clash of notation by using sans serif letters in describing the Surface Sampling Algorithm with these correspondences: x and y represent values for (ρ, θ). X_k and Y_k represent either (ρ, θ) regarded as random variables or as their *ex post* values as elements of an MCMC chain, as determined by context.

The first issue that needs to be addressed is the computation of the integral

$$\int m(y, x_{t-1}, \rho, \theta) f(y \,|\, x_{t-1}, \rho)\, dy \tag{6}$$

that appears in (2). We use Gaussian quadrature (Golub and Welsch, 1969) to integrate (6). In particular, we use Gauss-Hermite quadrature because that rule is best suited to the SNP-ARCH sieve that we use in the illustrative examples of Section 3. A sieve with different tail behavior would entail a different quadrature rule as discussed in Subsection 2.5.

For a univariate function $g(u)$ our quadrature rule of order I has the form

$$\int g(u) e^{-\frac{1}{2}u^2} du \doteq \sum_{i=1}^{I} \tilde{w}_i \, g(\tilde{u}_i). \tag{7}$$

The standard Gauss-Hermite formula is $\int g(u) e^{-u^2}\, du \doteq \sum_{i=1}^{I} \hat{w}_i \, g(\hat{u}_i)$. The transformation to get from the standard rule (Golub and Welsch, 1969) to ours is $\tilde{w}_i = \sqrt{2}\hat{w}_i$ and $\tilde{u}_i = \sqrt{2}\hat{u}_i$.

To get a multivariate rule for $z = (z_1, \ldots, z_J)$ of dimension J, let λ_k denote a vector of dimension J whose elements $\lambda_{j,k}$ are a permutation of the numbers $\{1, 2, \ldots, I\}$ and let $\{\lambda_k\}_{k=1}^{K}$ denote the set of all distinct such λ_k. The multivariate abcissae and weights are $z_k = (\tilde{u}_{\lambda_{1,k}}, \ldots, \tilde{u}_{\lambda_{J,k}})$ and $w_k = \prod_{j=1}^{J} \tilde{w}_{\lambda_{j,k}}$. We shall also need $e_k = \prod_{j=1}^{J} \exp\left(-\frac{1}{2}(\tilde{u}_{\lambda_{j,k}})^2\right)$.

The SNP code normalizes its data prior to use and presumably the moment conditions are coded to expect data in its natural units. With this convention assumed, let $\bar{y}$ and

$S = RR^\top$ be the sample mean and variance of the data. The integral is computed as

$$\int m(y, x_{t-1}, \rho, \theta) f(y \,|\, x_{t-1}, \rho)\, dy \doteq \sum_{k=1}^{K} \frac{w_k}{e_k} m(\bar{y} + Rz_k,\, x_{t-1}, \rho, \theta) f(z_k \,|\, x_{t-1}, \rho) \tag{8}$$

This integration strategy avoids dependence of the abcissae and weights on ρ, which would make derivation of analytical derivatives of the right hand side of (8) a nightmare and would increase the cost of coding and evaluating numerical derivatives.[5] It is predicated on an a assumption that the range of the abcissae $\{z_k\}_{k=1}^K$ of rule (8) approximate the range of the observed data after the observed data have been rescaled to have sample mean zero and sample variance the identity. We check this requirement by making sure that the quantiles from 0.05 to 0.95 of each element of the rescaled data are within the range of the abcissae $\{\tilde{u}_i\}_{i=1}^I$ of rule (7). We increase the order I of rule (7) to expand the range of the abcissae if necessary.

Our exposition of the Surface Sampling Algorithm will be more detailed than Zappa, Holmes-Cerfon, and Goodman's exposition and will be in terms of our specific choice of software components.

Define

$$Q_{\mathsf{x}} = \left[\frac{\partial}{\partial \mathsf{x}} q_1(\mathsf{x}),\, \ldots,\, \frac{\partial}{\partial \mathsf{x}} q_m(\mathsf{x}) \right], \tag{9}$$

which is the transpose of the Jacobian of $q(\mathsf{x})$ and has dimension d_a by m. Put $A = [Q_{\mathsf{x}} \,|\, 0\,]$, which is a square matrix of dimension d_a by d_a whose last $d = d_a - m$ columns are filled with zeros. Apply the singular value decomposition algorithm (Businger and Golub, 1969) to obtain $A = USV^\top$; U will be orthogonal and S diagonal with the first m diagonal entries positive and the remainder zero. If S is not such, Q_{x} does not have full rank, which violates a regularity condition of the Surface Sampling Algorithm. Partition U as $\left[T_{\mathsf{x}}^\perp \,|\, T_{\mathsf{x}}\right]$, where $T_{\mathsf{x}}^\perp$ has m columns and T_{x} has d columns.

A step in the chain requires solving

$$q(\mathsf{x} + v + Q_{\mathsf{x}} a) = 0 \tag{10}$$

for a. We use Newton's method (Suli and Mayers, 2003) with Fletcher line search (Fletcher, 1987) from a start of $a = 0$. The solution tolerance `tol` and iteration limit `nmax` are tuning

[5] A call to the generic SNP code returns $\frac{\partial}{\partial \rho} f(y \,|\, x_{t-1}, \rho)$ as well as $f(y \,|\, x_{t-1}, \rho)$ which facilitates the computation of the derivatives of (8).

parameters. The equation solver is used twice. Its performance *per se* is not critical; what is critical is that it be the identical solver in both instances.

In (10), v is a proposal drawn as follows: Draw $z_1, \ldots, z_d$ independently from the normal $n(z \,|\, 0, s^2)$ density, where the standard deviation s is a tuning parameter. Put $z = (z_1, \ldots, z_d)^\top$. Then $v = T_{\mathsf{x}} z$ is the draw from the proposal density $p(v)$. In the Metropolis-Hastings step below, evaluate $p(v)$ by $n_d(z \,|\, 0, s^2 I) = n_d(T_{\mathsf{x}}^\top v \,|\, 0, s^2 I)$ and $p(v')$ by $n_d(T_{\mathsf{y}}^\top v' \,|\, 0, s^2 I)$, where $n_d(\cdot \,|\, \mu, \Sigma)$ is the d-dimensional, multivariate normal density.

A difficulty with Zappa, Holmes-Cerfon, and Goodman's $p(v)$ is that there is no control over the relative scaling of the elements of v. This can be partially remedied with an alternative $p(v)$ as follows: Let S be a diagonal matrix of positive tuning parameters $s_1, s_2, \ldots, s_{d_a}$ along the diagonal. Let $R_s = T_{\mathsf{x}}^\top S$. Draw $u_1, \ldots, u_{d_a}$ independently from the standard normal $n(u \,|\, 0, 1)$ and put $z = R_s u$. Then $v = T_{\mathsf{x}} z$ is the draw from the alternative proposal density $p(v)$. Let $\Sigma_s = R_s R_s^\top$. In the Metropolis-Hastings step below, evaluate the alternative $p(v)$ by $n_d(z \,|\, 0, \Sigma_s) = n_d(T_{\mathsf{x}}^\top v \,|\, 0, \Sigma_s)$ and $p(v')$ by $n_d(T_{\mathsf{y}}^\top v' \,|\, 0, \Sigma_s)$.

An iteration of the Surface Sampling Algorithm has the property that if X_k is a draw from the posterior $p(\mathsf{x} \,|\, y) \propto f(y \,|\, \mathsf{x}) \pi(\mathsf{x})$ subject to (2) and (3), then so is X_{k+1}. An iteration of the Surface Sampling Algorithm proceeds as follows.

1. **Begin:** $\mathsf{x} = \mathsf{X}_k$ (X_k must be in M).

2. **Proposal:**

   (a) Calculate Q_{x} according to (9).

   (b) Compute $T_{\mathsf{x}}^\perp$ and T_{x} using the SVD as described above.

   (c) Draw $v \sim p(v)$ as described above.

3. **Projection to M:**

   (a) Solve $q(\mathsf{x} + v + Q_{\mathsf{x}} a) = 0$ for a using Newton's method.

   (b) If Newton's method fails, put $\mathsf{X}_{k+1} = \mathsf{x}$. **Done**.

   (c) Else $\mathsf{y} = \mathsf{x} + v + Q_{\mathsf{x}} a$. **Continue.**

4. **Inequality check:**

(a) If $h_i(\mathsf{y}) < 0$ for some i, put $\mathsf{X}_{k+1} = \mathsf{x}$. **Done**.

(b) Else y satisfies (3). **Continue.**

5. **Metropolis-Hastings acceptance/rejection step:**

   (a) Calculate Q_{y} according to (9).

   (b) Compute $T_{\mathsf{y}}^{\perp}$ and T_{y} using the SVD as described above.

   (c) Find $v' \in T_{\mathsf{y}}$ and $w' \in T_{\mathsf{y}}^{\perp}$ so that $\mathsf{x} = \mathsf{y} + v' + w'$.[6]

   (d) $P_a = \min\left(1, \frac{f(y\,|\,\mathsf{y})\pi(\mathsf{y})p(v')}{f(y\,|\,\mathsf{x})\pi(\mathsf{x})p(v)}\right)$

   (e) Generate $U \sim$ Uniform(0,1).

   (f) If $U > P_a$, put $\mathsf{X}_{k+1} = \mathsf{x}$. **Done**.

   (g) Else **Continue.**

6. **Reverse Projection:**

   (a) Solve $q(\mathsf{y} + v' + Q_{\mathsf{y}}a) = 0$ for a using Newton's method.

   (b) If Newton's method fails, put $\mathsf{X}_{k+1} = \mathsf{x}$. **Done**.

   (c) Else accept move, $\mathsf{X}_{k+1} = \mathsf{y}$. **Done**.

2.2 The λ-prior Method

The λ-prior method is simple: One merely draws from the posterior

$$p(\rho, \theta\,|\,y, x) \propto f(y\,|\,x, \rho)\pi(\rho, \theta)\pi_\lambda(\rho, \theta) \tag{11}$$

by MCMC (Gamerman and Lopes, 2006) subject to the support conditions (3), where

$$\pi_\lambda(\rho, \theta) = \exp\left[-\lambda\frac{n}{2}\sum_{i=1}^{m} q_i^2(\rho, \theta)\right]. \tag{12}$$

Above, $q(\rho, \theta)$ is given by (2) and $\pi(\rho, \theta)$ by (4). One can use other distances in the exponent such as $\sum_{i=1}^{m} |q_i(\rho, \theta)|$ but experience suggests that (12) works best in the applications encountered to date for the purpose of finding a draw (ρ, θ) close to the manifold M.

[6] I.e., put $z = [T_{\mathsf{y}}^{\perp}\,|\,T_{\mathsf{y}}]^{\top}(\mathsf{x} - \mathsf{y})$, then $w' = T_{\mathsf{y}}^{\perp}z$ and $v' = T_{\mathsf{y}}z$.

It is obvious at sight that the (ρ, θ) draws in the MCMC chain will be forced closer to the manifold M as λ increases. Less obvious is that there is a limit to how large λ can be before the MCMC chain fails to mix. But this is indeed the case as discussed in theory by Born, Shephard, and Solgi (2018) and as borne out in practice. In our experience many draws of (ρ, θ) within the chain get close enough to the boundary of M to provide a start X_k for the first step of the Surface Sampling Algorithm whether the chain mixes or not.

2.3 Estimating Scale

On a submanifold $M \subset \mathbb{R}^{d_a}$ of dimension $d < d_a$, distance is computed along geodesics. One computes distance $\delta_M(s, p)$ by traversing a geodesic from a starting point s to an end point p and accumulating (infinitesimal increments of) a weight function defined on M.[7] Average squared distance is computed by integrating $[\delta_M(s, p)]^2$ as a function of the end point p with respect to the probability distribution over the manifold. The mean $\bar{\mathsf{x}}$ is defined as that starting point that minimizes average squared distance. Variance is computed similarly by accumulating distance elementwise over a geodesic to obtain a vector $D_M(\bar{\mathsf{x}}, p)$ and then integrating $D_M(\bar{\mathsf{x}}, p) D_M^\top(\bar{\mathsf{x}}, p)$ as a function of p with respect to the probability distribution. If one has a sample from the distribution, e.g., MCMC draws, one averages distances over the sample to estimate the mean and variance instead of integrating with respect to a distribution on the manifold.

The question then becomes how to compute a geodesic on a manifold when one only has a point cloud.

Distance along a geodesic on a d-dimensional submanifold of $\mathbb{R}^{d_a}$ satisfies the intrinsic Eikonal distance equation

$$\begin{aligned} &\|\nabla_M \delta_M(s, p)\| = 1 \quad p \in M \\ &\delta_M(s, s) = 0 \end{aligned} \tag{13}$$

where $\nabla_M \delta_M(s, p)$ denotes intrinsic differentiation, $\delta_M(s, p)$ denotes intrinsic distance as described above, s is the starting point, and p is the end point.

[7] See Memoli and Sapiro (2001, Subsection 1.1) for details; non-Euclidean distance is obtained by making the right hand side of (13) and (14) a weight function other than $g \equiv 1$.

If one puts an ϵ-offset on the submanifold M to obtain a d_a-dimensional subset M_ϵ of $\mathbb{R}^{d_a}$, then one can solve, instead, the extrinsic Eikonal distance equation

$$\|\nabla\delta(s,p)\| = 1 \quad p \in M_\epsilon \tag{14}$$
$$\delta(s,s) = 0$$

where δ is Euclidean distance and differentiation is the usual one. One can construct such an M_ϵ as the union of ϵ-balls centered at the draws of an MCMC chain on the manifold M provided ϵ is large enough that M_ϵ is a connected set. In view of the fact that the contours of the posterior density determined by the Surface Sampling Algorithm are not spheres, our ϵ-balls for determining $\mathcal{G}_\epsilon$ are actually rectangles with sides k equal to $\Delta \max\{|\mathsf{x}_{k,i} - \mathsf{x}_{k,i-1}| : \mathsf{x}_i \in \mathcal{D}\}$ where $\mathcal{D} = \{\mathsf{x}_i\}_{i=1}^N$ denotes the ordered Surface Sampling MCMC chain and $\mathsf{x}_{k,i}$ denotes the kth element of x_i. The approximation of the solution of (13) by the solution of (14) improves as ϵ decreases. To decrease ϵ while retaining connected M_ϵ usually requires an increase in the length N of the MCMC chain $\mathcal{D}$. See Memoli and Sapiro (2005).

Standard algorithms for the solution of (14) produce as a by-product the geodesic that connects the starting point s to the end point p.

The Fast Marching Algorithm (Sethian, 1996) is frequently used to solve (14); see Sethian (2010) for an excellent introductory exposition. For this algorithm, Memoli and Sapiro (2001, Section 3) provide the upwind equation and the neighbor checking modification needed to adapt the Fast Marching Algorithm to M_ϵ constructed from a point cloud as above. Unfortunately, the Fast Marching Algorithm requires that M_ϵ be placed on a Euclidean grid by interpolation. The demands of a grid on computer memory limit the applicability of the implementations of the Fast Marching Algorithm known to us to problems where d_a is less than about five.

For dimensions higher than five the best available method appears to be Dijkstra's algorithm (Dijkstra, 1959). If M_ϵ is a connected set, then the MCMC draws may be viewed as nodes p_j of a graph $\mathcal{G}_\epsilon$ connected by edges $e_{j,j'}$ that have length $\delta(p_j, p_{j'})$ and that stay within M_ϵ. From a start s, Dijkstra's algorithm finds the shortest path that traverses edges to every node p_j. These distances will be larger than those of the Fast Marching Algorithm, were it applicable, because the Fast Marching Algorithm is not constrained to follow edges.

Computations are as follows.

The Surface Sampling MCMC chain $\mathcal{D} = \{\mathsf{x}_i\}_{i=1}^N$ will contain duplicates due to rejections. They are easily detected because they must occur in succession with probability one. Nodes are the distinct points $\{p_j\}_{j=1}^{N^*}$ and $j(i)$ is the mapping from draw index i to node index j. Dijkstra's algorithm gives the distance $\delta(s, p_j)$ along edges from s to every node p_j and the path $(j_1^p, j_2^p, ..., j_k^p)$ that connects them, where j_1^p refers to starting node s and j_k^p to node p_j

First one chooses a Δ and constructs the graph $\mathcal{G}_\epsilon$. If $\mathcal{G}_\epsilon$ is not connected, Dijkstra's algorithm will return ∞ for the distance from s to an isolated node. One can find the smallest admissible Δ by increasing Δ until Dijkstra's algorithm does not find isolated nodes.

The estimated posterior mean $\bar{\mathsf{x}}$ is the start s that minimizes $\frac{1}{N}\sum_{i=1}^N \delta^2(s, p_{j(i)})$.

To compute scale, for the path $(j_1^p, j_2^p, ..., j_k^p)$ that connects s to p_j put[8]

$$D_j = \mathrm{diag}[\mathrm{sgn}(p_j - \bar{\mathsf{x}})] \sum_{\ell=2}^{k} |p_{j_\ell^p} - p_{j_{\ell-1}^p}| \quad D_j \in \mathbb{R}^{d_a} \tag{15}$$

where the signum and absolute value functions are applied elementwise to $p_j - \bar{\mathsf{x}}$ and $p_{j_\ell^p} - p_{j_{\ell-1}^p}$, respectively. The estimated variance matrix is

$$V = \frac{1}{N}\sum_{i=1}^{N} D_{j(i)} D_{j(i)}^\top. \tag{16}$$

The Δ that determines the graph $\mathcal{G}_\epsilon$ is a tuning parameter. As yet we do not know the optimal choice. Too small and one is essentially forcing Dijkstra's algorithm to traverse the entire Surface Sampling MCMC chain to find a path. Too large and nodes that should not be connected by edges are. Our thinking at present is to increase Δ until standard deviations are larger than but reasonable relative to those returned by the λ-prior method with λ chosen to get a close match to GMM estimates of the location and scale for θ.

A lower bound on variance is obtained by making Δ large enough that every draw is connected to all of the other draws. The sample variance computed directly from the MCMC draws $\mathcal{D} = \{\mathsf{x}_i\}_{i=1}^N$ is a slightly smaller[9] lower bound but much cheaper to compute.

Unfortunately, computing the posterior mean is an order N^2 computation and is quite time consuming. Efforts to improve it by eliminating seemingly irrelevant comparisons have

[8] This expression is wrong in versions of the MS prior to June 11, 2021; that includes the published version.

[9] Because the sample mean is not restricted to be one of the draws.

so far caused erroneous computation of the posterior mean. The posterior mean does not change much with changes in Δ so that the search for the best Δ can avoid computation of the mean except at intervals along the Δ search path.

2.4 Estimating the Normalization Constant

Consider the normalization constant, aka marginal likelihood or marginal data density,

$$Z = \int_M f(y \,|\, \mathsf{x})\, \pi(\mathsf{x})\, d\sigma(\mathsf{x}), \tag{17}$$

where $\sigma(\mathsf{x})$ is d-dimensional Hausdorff measure on $\mathbb{R}^{d_a}$. If a mapping from $\mathbb{R}^d$ to M can be found, then computing (17) can be accomplished by Riemann integration after multiplication by a Jacobian term (Morgan, 2016, p. 29). The strategy proposed by Zappa, Holmes-Cerfon, and Goodman (2018) consists of successively reducing the domain of integration until such a mapping can be found. The remaining part of the integral can be computed from Surface Sampling draws. As above, their algorithm is presented here in terms of our specific choices of software components and options.

Let x_0 be that draw with the highest value of $f(y \,|\, \mathsf{x})\pi(\mathsf{x})$ in the set of Surface Sampler MCMC draws that are to be used for estimation and inference; i.e., an estimate of the posterior mode. Let $\mathcal{D}_0 = \{\mathsf{x}_i\}_{i=1}^{n_0}$ be n_0 subsequent Surface Sampler draws. Duplicates in $\mathcal{D}_0$ due to rejections are easily eliminated because they must occur in succession. Let $\mathcal{D}_0^e$ denote the subset of $\mathcal{D}_0$ remaining after elimination. Compute the Euclidean norms $\mathcal{N}_0 = \{\|\mathsf{x} - \mathsf{x}_0\| \,:\, \mathsf{x} \in \mathcal{D}_0^e\}$. Let $r_0 = \max \mathcal{N}_0$, r_1 the 90th percentile, r_2 the 80th, and so on until r_9 the 10th. Let B_i be a closed ball in $\mathbb{R}^{d_a}$ with center x_0 and radius r_i. Note that $B_0 \supset B_1 \supset \ldots \supset B_9$. Let

$$Z_i = \int_{M \cap B_i} f(y \,|\, \mathsf{x})\, \pi(\mathsf{x})\, d\sigma(\mathsf{x}).$$

For k yet to be determined, note that

$$Z = Z_k \prod_{i=0}^{k-1} \frac{Z_i}{Z_{i+1}} = Z_k \prod_{i=0}^{k-1} R_i.$$

Now $\frac{Z_{i+1}}{Z_i} = \frac{1}{Z_i} \int_{M \cap B_i} I_{B_{i+1}}(\mathsf{x}) f(y \,|\, \mathsf{x})\, \pi(\mathsf{x})\, d\sigma(\mathsf{x})$. Therefore, if we add the constraint $\|\mathsf{x} - \mathsf{x}_0\| <= r_i$ to the support conditions (3), generate n_i draws from the Surface Sampling Algorithm and count the number $N_{i,i+1}$ of those draws that are in B_{i+1}, then an estimate

of $\frac{Z_{i+1}}{Z_i}$ is $\frac{N_{i,i+1}}{n_i}$. Thus, an estimate of R_i is $\hat{R}_i = \frac{n_i}{N_{i,i+1}}$. The number of draws n_i and the starting values of the Surface Sampler to compute the R_i are tuning parameters. We set the starts to x_0 and the n_i to a single value on the order of 10,000.

Now consider determination of k. Start with a first guess at k, which is a tuning parameter that we set to 5, and a tuning parameter n_k that we set to N. Compute Q_{x_0} and T_{x_0}, i.e., compute Q_{x} and T_{x} as described in Subsection 2.1 with x set to x_0. For $i = 1, \ldots, n_k$, draw u_i from the uniform distribution on a ball of dimension d and radius r_k and put $v_i = T_{\mathsf{x}_0} u_i$. Note that $\mathsf{x}_i = x_0 + v_i \in B_k$ because T_{x_0} has orthonormal columns. Project onto M by solving $q(\mathsf{x}_0 + v_i + Q_{\mathsf{x}} a) = 0$ given by (10) for a, then setting $\mathsf{y}_i = \mathsf{x}_0 + v_i + Q_{x_0} a$. If the projection fails for some i, abort, increase the guessed value for k by one, and repeat. If $k = 9$ fails, one can set $r_{10} = r_9/2$, and try again. Repeat thus with successive divisions by 2. For us so far, failure to accept $k = 5$ has never happened.

The Jacobian term that corresponds to x_i and y_i is $J_i = \det(T_{x_0}^\top T_{y_i})$.

We can now compute Z_k by Monte Carlo integration as follows. Compute

$$\hat{I} = \frac{1}{n_k} \sum_{i=1}^{n_k} I_{B_k}(\mathsf{y}_i) \, (J_i)^{-1} \exp\left[\log f(y \,|\, \mathsf{y}_i) + \log \pi(\mathsf{y}_i) - \log f(y \,|\, \mathsf{x}_0) - \log \pi(\mathsf{x}_0)\right].$$

Then

$$\log Z_k = (d/2) \log \pi - \log \Gamma(d/2 + 1) + d \log(r_k) + \log(\hat{I}) + \log f(y \,|\, \mathsf{x}_0) + \log \pi(\mathsf{x}_0)$$

and

$$\log Z = \log Z_k + \sum_{i=0}^{k-1} \log \hat{R}_i.$$

2.5 Regularity Conditions

The class of models for which the SNP sieve is dense is specified in Gallant and Nychka (1987). The most stringent of the requirements there are that densities in this class must possess a moment generating function and must be dominated by a Sobelev norm. Usually domination has the effect of requiring the parameters of models in the class to be in a compact set. The algorithm for computing the normalizing constant described in Subsection 2.4 also requires the parameters $\mathsf{x} = (\rho, \theta)$ to be in a compact set. In most applications the compactness requirement is honored in the breech: One can set bounds on the parameters so large that

they might as well be absent so there is no point to bother with imposing them in the first place. Relevant is that the algorithm in Subsection 2.4 does not make use of bounds, were they present, to determine the radius of B_0.

The choice of quadrature rule in Subsection 2.1 is dictated by the tail behavior of the density $f(y \mid x_{t-1}, \rho)$; see Golub and Welsch (1969). For instance, in the example of Subsection 3.2, were we to work with gross returns instead of log gross returns, the appropriate expansion for $f(y \mid x_{t-1}, \rho)$ would be the Laguerre polynomials instead of the Hermite polynomials of the SNP density and the appropriate quadrature rule would be Gauss-Laguerre. Similarly, if $f(y \mid x_{t-1}, \rho)$ had bounded support, Gauss-Legendre would be the appropriate rule.

The manifold M given by (5) must be connected. The Jacobian transpose Q_x given by (9) must exist and be full rank on M. Because of our use of quadrature, for integration, it is the right hand side of (8) that must be differentiable. The SNP density is differentiable in ρ and usually $m(y, x_{t-1}, \rho, \theta)$ does not depend on ρ so that it is differentiablity of $m(y, x_{t-1}, \rho, \theta)$ with respect to θ that is usually required.

Technically, the starting value of the Surface Sampler must be in M; practically, very close to M. Violation of this requirement causes the Surface Sampling Chain to stick at the starting value.

Zappa, Holmes-Cerfon, and Goodman (2018) provide detailed proofs of why Step 3, Projection, and Step 6, Reverse Projection, of the Surface Sampling Algorithm, imply that detailed balance holds and why no Jacobian terms are present in P_a of Step 5d.

The implementation described here is intended for overidentified moment conditions. If, instead, the moment conditions are just identified, i.e., the dimension m of $q(\rho, \theta)$ is the same as the dimension of θ, then ρ is unrestricted, can be sampled directly by MCMC, and can provide corresponding θ draws by solving $q(\rho_i, \theta) = 0$ for θ at each draw ρ_i. Examination of Step 3, Projection, of the Surface Sampling Algorithm reveals that this is what the Surface Sampling Algorithm will do but accompanied by linear algebra that is wasteful for that purpose. Step 6, Reverse Projection, is completely wasteful for that purpose. We confirmed that the Surface Sampling Algorithm does behave as just stated by trying it on a trivial, just identified model with a weak prior.

If $q(\rho, \theta) = 0$ does not have a solution, then the Surface Sampling Algorithm will not work at all because one cannot even provide a starting value. If one persist with a start that, perforce, does not solve $q(\rho, \theta) = 0$, then the Surface Sampling Algorithm cannot get past the Projection step and will stick.

3 Examples

3.1 A Simple Instrumental Variables Example

Consider a simulation of the demand and supply system

$$x_t = (\sigma_x + \rho_x x_{t-1}) z_{1,t} \tag{18}$$

$$\log q_{d,t} = a_1 + a_2 \log p_t + \sigma_d z_{2,t} \tag{19}$$

$$\log q_{s,t} = b_1 + b_2 \log p_t + x_t + \sigma_s z_{3,t} \tag{20}$$

with solution $(\log p_t, \log q_t)$ under $q_t = q_{d,t} = q_{s,t}$, where $\sigma_x = 3$, $\rho_x = 0.2$, $a_1 = 12$, $a_2 = -2$, $b_1 = 3$, $b_2 = 4$, $\sigma_d = \sigma_s = 0.1$, $z_{i,t}$ standard normal, and sample size $n = 500$. Note that the supply shifter x_t is heteroscedastic with variance dependent on x_{t-1} whence the same for price p_t and quantity q_t. The data are $y_t = (\log p_t, \log q_t, x_t)$ for $t = 1, 2, \ldots, n$.

The SNP likelihood used for estimation is normal with heteroscedastic errors that depend on past values of y_t:

$$y_t \sim n_3(y_t \,|\, y_{t-1}, \mu, \Sigma) \tag{21}$$

$$\Sigma = RR' + P(y_{t-1} - \mu)(y_{t-1} - \mu)' P', \tag{22}$$

where R is upper triangular, and P is diagonal. Thus,

$$\rho = (\mu_1, \mu_2, \mu_3, R_{1,1}, R_{1,2}, R_{2,2}, R_{1,3}, R_{2,3}, R_{3,3}, P_{1,1}, P_{2,2}, P_{3,3}) \in \mathbb{R}^{12}.$$

A set of moment conditions for estimation of the demand equation (19) are

$$m_{d,1}(y_t, y_{t-1}, \rho, \theta) = \log q_t - a_1 - a_2 \log p_t \tag{23}$$

$$m_{d,2}(y_t, y_{t-1}, \rho, \theta) = x_t m_{d,1}(y_t, y_{t-1}, \rho, \theta) \tag{24}$$

$$m_{d,3}(y_t, y_{t-1}, \rho, \theta) = x_{t-1} m_{d,1}(y_t, y_{t-1}, \rho, \theta) \tag{25}$$

$$\theta = (a_1, a_2)$$

$$\rho \quad \text{not used}$$

Table 1 about here.

The prior for ρ is independent normal with location the unconstrained maximum likelihood estimates of (21) and scale twice the maximum likelihood standard errors. The prior for $\theta = (a_1, a_2)$ is independent normal with means $(12, -2)$ and standard deviations $(2, 2)$.

The support conditions on R and P of (22) are that diagonals of R_0 must be positive, the first diagonal element P must be positive, and the eigenvalues of the companion matrix of Σ must be less than one in absolute value. In addition, a_1 must be positive and a_2 negative.

We ran the Surface Sampling Algorithm, with moment conditions, prior, and support conditions as immediately above and retained 50,000 draws after transients had died out. The tuning parameters were `tol` $= 0.001$, `nmax` $= 20$. The scaling matrix S of the proposal density $p(v)$ had diagonal elements 0.006 corresponding to μ_i and $P_{i,i}$ and 0.005 corresponding to $R_{i,j}$ and a_i. We used a five point rule for (7) whence (8) is a 125 point rule.

Because the data are simulated, we were able to start the Surface Sampling Algorithm from the maximum likelihood estimates of ρ and the known values of θ. Use of the λ-prior method to get starts was not necessary. The discussion of λ-prior and Surface Sampling tuning in the next subsection is more relevant to applications because that example is for real data with answers not known in advance.

Parameter estimates are shown in Table 1. The posterior mode is used as the estimate of location for reasons discussed in the next subsection. As discussed in Subsection 2.1, SNP uses normalized data so estimated μ and RR' should be near zero and the identity, respectively; the moment equations use raw data so estimated a_i are in natural units. Reduced form and nonparametric Bayes estimates of ρ differ little because this is simulated data and (ρ, θ) is on the manifold (5). The effect of imposing the moment conditions (23), (24), and (25) to reduce standard deviations computed from MCMC draws, as discussed next.

Scale marked Hi Std.Dev. in Table 1 is computed as described in Subsection 2.3 with tuning parameter $\Delta = 2$. These are sometimes called "Intrinsic Standard Deviations" because the distance between points on the singular manifold (5) is computed by following the geodesic that connects them. Scale marked Lo Std.Dev. in Table 1 is the sample variance computed directly from the MCMC draws. These are sometimes called "Extrinsic Standard Deviations". The intrinsic distance between any two points on the manifold is greater than

or equal to extrinsic distance whence intrinsic standard deviations are greater than or equal to extrinsic standard deviations. See Subsection 2.3 for details.

3.2 Extraction of the Stochastic Discount Factor

As a substantive example, we consider the specification and extraction of the *ex post* stochastic discount factor.

Let $R_t = \frac{P_t + D_t}{P_{t-1}}$ denote the gross return to a security whose price is P_t at time t and that pays a dividend D_t at time t. Let $r_t = \log(P_t + D_t) - \log(P_{t-1})$ denote its geometric return. For any security, the stochastic discount factor satisfies

$$1 = \int \mathrm{SDF}_t(y)\, R_t(y)\, f(y \,|\, x_{t-1}, \rho)\, dy \tag{26}$$

provided that, indeed, the SDF_t and R_t are functions of y. The density $f(y_t \,|\, x_{t-1}, \rho)$ is that given by (1) and x_{t-1} is the time $t-1$ information set of the conditional expectation (26).

Table 2 about here.

As data we take $y_{1,t}$ to be daily, inflation adjusted, geometric returns on the S&P500 stock index (including distributions) and $y_{1,t}$ the same for the NASDAQ stock index for January 1, 2010, to December 31, 2018, which are $n = 2264$ bi-variate observations. Sources are the Center for Research in Security Prices at the Wharton Research Data Services web site (http://wrds.wharton.upenn.edu) for the S&P500, the St. Louis Federal Reserve bank (https://fred.stlouisfed.org) for the NASDAQ, and the Bureau of Economic Analysis (https://www.bea.gov) for the GDP deflator used to adjust returns for inflation. The returns are expressed as a percentage, i.e., $y_t = 100 r_t$. simple statistics are shown as Table 2 together with simple statistics on the extracted SDF. These data are available at `www.aronaldg.org/webfiles/data` as files `stocks_s.dat` for data and `stocks_s.doc` for documentation.

We consider a log linear and log quadratic specification of $sdf = \log(\mathrm{SDF})$; viz.,

$$sdf_l(y_t) = a_0 + a_1 f_t \tag{27}$$

$$sdf_q(y_t) = a_0 + a_1 f_t + a_2 f_t^2, \tag{28}$$

where $f_t = \frac{1}{2}(y_{1,t}/100 + y_{2,t}/100)$.

For the likelihood (1) we use a bivariate SNP-ARCH model. This model has an SNP innovation density with VAR location and diagonal ARCH scale. It is parameterized as follows:

$$y \sim \frac{[\mathcal{P}(z)]^2 n_2(y \,|\, \mu, \Sigma)}{\int [\mathcal{P}(s)]^2 n_2(s \,|\, 0, I)\, ds} \tag{29}$$

where $\mathcal{P}(z)$ is evaluated at $z = \Sigma^{-1/2}(y - \mu)$, $n_2(y|\, \mu, \Sigma)$ is the bivariate normal density, and

$$\mathcal{P}(z) = a_{01}z_2 + a_{02}z_2^2 + a_{03}z_2^3 + a_{04}z_2^4 + a_{05}z_1 + a_{06}z_1^2 + a_{07}z_1^3 + a_{08}z_1^4.$$

The location and scale are

$$\mu = b_0 + By_{t-1}$$

$$\Sigma = R_0R_0' + P(y_{t-1} - b_0 - By_{t-2})(y_{t-1} - b_0 - By_{t-2})'P'$$

where R_0 is upper triangular and P is diagonal. The parameters of $\mathcal{P}(z)$, the elements of b_0 and B, and the non-zero elements of R_0 and P are the elements of ρ; see Table 3 for the ordering of parameters within ρ. The information set is $x_{t-1} = (y_{t-1}, y_{t-2})$.

The moment conditions defining (2) that we use in estimation are:

$$m_1(y_t, x_{t-1}, \rho, \theta) = 1.0 - \exp[sdf_q(y_t) + r_{1,t}] \tag{30}$$

$$m_2(y_t, x_{t-1}, \rho, \theta) = 1.0 - \exp[sdf_q(y_t) + r_{2,t}] \tag{31}$$

$$m_3(y_t, x_{t-1}, \rho, \theta) = y_{1,t-1}m_1(y_t, x_{t-1}, \rho, \theta) \tag{32}$$

$$m_4(y_t, x_{t-1}, \rho, \theta) = y_{2,t-1}m_2(y_t, x_{t-1}, \rho, \theta) \tag{33}$$

$$r_t = y_t/100$$

$$\theta = (a_0, a_1, a_2)$$

$$\rho \quad \text{not used}$$

For the linear SDF given by (27), the moment are the same but with sdf_l replacing sdf_q in (30) and (31) and with $\theta = (a_0, a_1)$.

The prior for ρ is independent normal with location and scale the SNP-ARCH unconstrained maximum likelihood estimated parameters and standard errors. Admittedly this is a data dependent, independence prior, but it is so loose that we think this consideration can

be dismissed. The prior for $\theta = (a_0, a_1, a_2)$ is independent normal with means $(0, -1, 0)$ and standard deviations $(1, 1, 1)$. This prior loosely implies a variant of CAPM (capital asset pricing model).

The support conditions apply to R_0 and P of the scale function and B of the location function. They are that the diagonals of R_0 be positive and that the first diagonal element of the diagonal matrix P be positive. In addition, the eigenvalues of the companion matrices for the location function and the scale function are required to be less than one in absolute value. Their values for unconstrained estimation of the SNP-ARCH model by maximum likelihood are 0.0784306 and 0.242951, respectively, so that it is unlikely that these last two support conditions ever bind.[10]

We ran the λ-prior chain with the moment conditions, prior, and support conditions described immediately above to get a start using $\lambda = 10^{-12}$; the resulting estimates and standard deviations are shown as the last two columns of Table 3. To comment briefly, when $\lambda = 10^{-14}$ estimates are about the same except that estimated $a_3 = -0.03185$, and standard deviations are one order of magnitude smaller than those shown in the last column of Table 3. For $\lambda = 10^{-16}$, estimates are again about the same but $a_3 = 0.011962$ and standard deviations are two orders of magnitude smaller than shown in the last column of Table 3. This behavior is the main problem with using the λ-prior method for estimation over a singular manifold such as (5): There is no known objective way to choose λ.

If one does wish to use the λ-prior method for estimation rather than the Surface Sampling approach, our suggestion is to estimate the location and scale of the parameters of the moment equations by GMM and choose λ to get the closest match possible to the GMM estimates. The benefit over GMM being that one thus acquires an estimate of a likelihood approximately subject to those moment conditions. See, e.g., the CRRA (constant relative risk aversion) asset pricing example in the slides at `www.aronaldg.org/papers/npbclr.pdf`, which compares estimates for various GMM estimation methods, the Surface Sampling method, and a λ-prior fit that matches to GMM estimates.

We then ran the Surface Sampling Algorithm, using Zappa, Holmes-Serfon, and Good-

[10]The volatility eigenvalues for an SNP-GARCH or VAR-GARCH model are larger than one in absolute value. All the multivariate GARCH models we tried in our specification search had explosive volatility estimates.

man's $p(v)$ with the same moment conditions, prior, and support conditions as immediately above and with tuning parameters $s_i = 0.001815$, for all diagonal elements of S, `tol` $= 0.001$, and `nmax` $= 20$.[11] We used a five point rule for (7) whence (8) is a 25 point rule. There was little sensitivity to `tol`; projection onto M was more accurate than `tol`.[12] There was some sensitivity to S. We collected 100,000 draws well after transients had died out. The Surface Sampling Algorithm was remarkably effective in keeping all MCMC draws of (ρ, θ) on the manifold M. The largest value of $q(\rho, \theta)$ in the 200,000 MCMC draws over both models was less than 5.0e-10.

Obviously examining the posterior by computing the posterior mean by averaging over the MCMC draws of $\mathtt{x} = (\rho, \theta)$ makes no sense because averages of draws have to be computed with respect to a notion of distance confined to the manifold M. Unfortunately, the latter computation depends on a tuning parameter Δ; see Subsection 2.3. Therefore, to interpret results as regard location, we pick the draw that has the highest posterior mode. Parameter estimates are shown in Table 3. The normalizing constants are computed with $n_i = 10000$ as -6154.5364 and -6153.0577 for the linear and quadratic models, respectively, using the algorithm described in Subsection 2.4, giving a posterior probability of 0.81 in favor of the quadratic SDF.

Table 3 about here.

An advantage of a nonparametric Bayesian strategy subject to moment conditions as opposed to Bayesian method of moments (Gallant, 2016, 2020) is that it permits a more exhaustive analysis of the data. Available for various hypotheticals are plots of the conditional density, estimates of functionals of the density, the posterior moments of such functionals, impulse response functions, persistence assessment via fiber bundles, etc.; see, e.g., Gallant, Rossi, and Tauchen (1992, 1993) for examples. We provide an illustration here: Figure 1, overplots a stationary density whose first four moments reasonably match those shown in Table 2 and the conditional density $f(y \,|\, x_{t-1}, \rho)$ given by (29) estimated subject to moment conditions (30) through (33) evaluated at ρ for the quadratic in Table 3. The conditioning

[11] Trace plots indicated that unequal s_i were not required. The odd value of the s_i is due to moving the s_i up and down by multiplicative factors such as 0.9, 1.1, etc. It does not indicate requisite precision for s_i.

[12] This is apparently a consequence of using Fletcher line search.

events x_{t-1} are the days of the largest crashes and rallies on the S&P 500 and NASDAQ, the values of which are shown in the table legend. Volatility after a crash or rally is roughly 2.5 times larger than stationary volatility. The next day move of the mean slightly offsets the crash or rally, but in view of the volatility, the best that can be said is that anything can happen the day after a crash or rally. And that the move is likely to be large.

Figure 1 about here.

4 Conclusion

This paper has addressed computational problems that arise in nonparametric Bayesian estimation subject to overidentified moment conditions. The difficulty is that the support of the posterior is a manifold of lower dimension than the number of model parameters which inhibits the use of MCMC methods. This paper proposes an effective MCMC algorithm to sample the posterior, methods for estimating scale from a point cloud on the manifold, and a method to compute the normalizing constant of the posterior density. Effectiveness is illustrated by an a simple instrumental variables illustration and by extraction of the SDF (stochastic discount factor) from daily S&P500 and NASDAQ returns.

5 References

Bornn, Luke, Neil Shephard, and Reza Solgi (2018), "Moment Conditions and Bayesian Nonparametrics," *Journal of the Royal Statistical Society, Series B* 81(1), 5–43.

Businger, Peter A., and Gene H. Golub (1969), "Singular Value Decomposition of a Complex Matrix," *Communications of the ACM* 12, 564–565.

Chernozhukov, Victor, and Han Hong (2003), "An MCMC Approach to Classical Estimation," *Journal of Econometrics* 115(2), 293–346.

Dijkstra, E. W. (1959), "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik* 1 269–271

Fletcher, R. (1987), *Practical Methods of Optimization, 2nd Edition*, Wiley, New York, 26–40.

Gallant, A. Ronald (2016), "Reflections on the Probability Space Induced by Moment Conditions with Implications for Bayesian Inference," *Journal of Financial Econometrics* 14, 229–247.

Gallant, A. Ronald (2020), "Complementary Bayesian Method of Moments Strategies," *Journal of Applied Econometrics* 35, 422–439.

Gallant, A. Ronald, Han Hong, Michael P. Leung, and Jessie Li (2019), "Constrained and Penalized Estimation using MCMC," Manuscript, http://www.aronaldg.org/papers/ce.pdf.

Gallant, A. Ronald, and Douglas W. Nychka (1987), "Semi-Nonparametric Maximum Likelihood Estimation," *Econometrica* 55, 363–390.

Gallant, A. Ronald, Peter E. Rossi, and George E. Tauchen (1992), "Stock Prices and Volume," *The Review of Financial Studies* 5, 199–242.

Gallant, A. Ronald, Peter E. Rossi, and George E. Tauchen (1993), "Nonlinear Dynamic Structures," *Econometrica* 61, 871–907.

Gallant, A. Ronald, and George Tauchen (2017), "SNP: A Program for Nonparametric Time Series Analysis, Version 9.1 User's Guide," Manuscript, www.aronaldg.org/webfiles/snp

Gamerman, Dani and Hedibert F. Lopes (2006), *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference (2nd ed.),* Boca Raton, FL: Chapman and Hall.

Golub, Gene H., and John H. Welsch (1969), "Calculation of Gaussian Quadrature Rules," *Mathematics of Computation* 23, 221–230.

Memoli, Facundo, and Guillermo Sapiro (2001), "Fast Computation of Weighted Distance Functions and Geodesics on Implicit Hyper-surfaces," *Journal of Computational Physics* 173, 730–764.

Memoli, Facundo, and Guillermo Sapiro (2005), "Distance Functions and Geodesics on Sub-manifolds of $\mathcal{R}^d$ and Point Clouds," *SIAM Journal of Applied Mathematics* 65, 1227–1260.

Morgan, Frank (2016), *Geometric Measure Theory, A Beginner's Guide, 5th Edition,* San Diego CA, Academic Press.

Sethian, J. A. (1996), "A Fast Marching Level Set Method for Monotonically Advancing Fronts," *Proceedings of the National Academy of Sciences* 93, 1591–1595. https://doi.org/10.1073/pnas.93.4.1591

Sethian, J. A. (2010), "Fast Marching Methods: A Boundary Value Formulation," Manuscript: https://math.berkeley.edu/∼sethian/2006/Explanations/fast_marching_explain.html

Schennach, Susanne M. (2005), "Bayesian Exponentially Tilted Empirical Likelihood," *Biometrika* 92, 31–46.

Shin, Minchuyl (2015), "Bayesian GMM," Working paper, Department of Economics, University of Illinois. http://www.econ.uiuc.edu/ mincshin/BGMM_ver05

Suli, Endre, and David Mayers (2003), *An Introduction to Numerical Analysis*, Cambridge University Press.

Zappa, Emilio, Miranda Holmes-Cerfon,and Jonathan Goodman (2018), "Monte Carlo on Manifolds: Sampling Densities and Integrating Functions," *Communications on Pure and Applied Mathematics* 71, 2609–2647.

Table 1. Demand and Supply Example Parameter Estimates

	Reduced Form		Nonparametric Bayes		
Parameter	Estimate	Std.Dev.	Estimate	Lo Std.Dev.	Hi Std.Dev.
μ_1	0.00417	0.04558	0.00010	0.03231	0.13842
μ_2	-0.00081	0.04540	0.02049	0.03490	0.16887
μ_3	-0.00147	0.04548	-0.00886	0.03532	0.16060
$R_{1,1}$	0.98833	0.03900	0.98584	0.03057	0.13707
$R_{1,2}$	-0.00174	0.02361	0.00485	0.01940	0.08361
$R_{2,2}$	0.99996	0.03781	1.01112	0.03184	0.12225
$R_{1,3}$	-0.00400	0.02157	-0.00470	0.01900	0.07417
$R_{2,3}$	-0.00064	0.02448	0.00894	0.01837	0.07259
$R_{3,3}$	0.99425	0.03531	0.98554	0.03009	0.13362
$P_{1,1}$	0.14013	0.16614	0.15897	0.08124	0.28866
$P_{2,2}$	0.07580	0.24505	0.05796	0.10914	0.40459
$P_{3,3}$	-0.10131	0.19240	-0.13080	0.12833	0.42660
a_1			11.99235	0.01066	0.04075
a_2			-1.99928	0.00678	0.02540

The data are a simulation of the demand and supply system (18) through (20). The reduced form estimates are maximum likelihood estimates for the heteroskedastic normal likelihood (21). The nonparametric Bayes estimates are computed using the Surface Sampling Algorithm for the likelihood (21) subject to moment conditions (2) as determined by (23) through (25). Average autocorrelations for MCMC chains at lag 50 were 0.60. The estimates shown are the those with the highest posterior mode in 50,000 Surface Sampling draws collected after transients dissipated. Scale marked Hi Std.Dev. is computed as described in Subsection 2.3 with tuning parameter $\Delta = 2$. Scale marked Lo Std.Dev. is the sample variance computed directly from the MCMC draws. Recall the coding conventions of Subsection 2.1: SNP uses normalized data so estimated μ and RR' should be near zero and the identity, respectively; the moment equations use raw data so estimated a_i are in natural units.

Table 2. Statistics for Data and SDF

	S&P500	NASDAQ	sdf_l	sdf_q
mean	0.04181	0.04076	-0.04142	-0.04143
std dev	0.94408	1.08461	0.99988	0.99982
skewness	-0.40315	-0.45746	0.43621	0.39168
kurtosis	4.43556	3.48237	4.03297	3.97506
	Correlations			
	S&P500	NASDAQ	sdf_l	sdf_q
S&P500	1.00000	0.95356	-0.98664	-0.98664
NASDAQ	0.95356	1.00000	-0.98989	-0.98985
sdf_l	-0.98664	-0.98989	1.00000	0.99998
sdf_q	-0.98664	-0.98985	0.99998	1.00000

Statistics computed from the data, which are daily, inflation adjusted, geometric returns including dividends, expressed as a percent, on the S&P500 and NASDAQ stock indexes, and from the extracted log SDFs, sdf_l, sdf_q, expressed as a percent, from January 1, 2010, to December 31, 2018, which are $n = 2264$ observations.

Table 3. Linear and Quadratic SDF Parameter Estimates

	Linear SDF			Quadratic SDF				
	Surface Sampling			Surface Sampling			λ-prior	
		Low	High		Low	High		
Parm	Est	Sdev	Sdev	Est	Sdev	Sdev	Est	Sdev
a_{01}	0.11862	0.03268	0.16628	0.14440	0.03642	0.27323	0.15385	0.00562
a_{02}	0.01399	0.02812	0.18932	0.01115	0.02840	0.20699	-0.03775	0.01708
a_{03}	0.02636	0.01529	0.10085	0.02136	0.01432	0.12896	0.00749	0.01230
a_{04}	0.08140	0.01068	0.09451	0.08448	0.01052	0.12633	0.07901	0.01236
a_{05}	-0.04835	0.02502	0.11964	-0.05492	0.02075	0.14644	-0.04508	0.00500
a_{06}	-0.02255	0.02092	0.15480	-0.03920	0.01923	0.17775	-0.02550	0.01672
a_{07}	-0.03390	0.01371	0.10433	-0.03312	0.01353	0.13481	-0.02797	0.00875
a_{08}	0.15204	0.01152	0.08969	0.15183	0.01112	0.15327	0.15182	0.00947
$b_{0,1}$	0.12552	0.04330	0.18922	0.14067	0.03439	0.19499	0.12560	0.00364
$b_{0,2}$	-0.23809	0.06133	0.27905	-0.29055	0.06939	0.46305	-0.28476	0.00685
$B_{1,1}$	-0.04869	0.01531	0.08063	-0.03979	0.01379	0.11784	-0.05413	0.00425
$B_{2,1}$	-0.06418	0.01957	0.11300	-0.04801	0.01837	0.16854	-0.06388	0.00740
$B_{1,2}$	-0.00307	0.01955	0.10959	-0.00388	0.01850	0.17987	0.00157	0.01408
$B_{2,2}$	-0.02274	0.01960	0.10424	-0.02041	0.02147	0.18976	-0.02974	0.02036
$R_{0,1,1}$	0.83003	0.03037	0.16300	0.83914	0.02620	0.23278	0.82794	0.02032
$R_{0,1,2}$	-0.03206	0.01125	0.08293	-0.03956	0.01081	0.14755	-0.03682	0.01012
$R_{0,2,2}$	0.95712	0.04378	0.27808	0.98015	0.04376	0.26173	1.02870	0.02727
$P_{1,1}$	0.45712	0.05754	0.26017	0.51896	0.05586	0.29325	0.49067	0.05880
$P_{2,2}$	0.09592	0.05605	0.24186	0.11608	0.05281	0.28683	0.18463	0.05458
a_1	-2.45e-6	5.30e-6	3.23e-5	2.33e-5	1.57e-5	0.00007	2.62e-5	6.46e-6
a_2	-0.99733	0.01139	0.05765	-0.99814	0.00933	0.07215	-0.99574	0.00497
a_3				-0.25289	0.14899	0.41864	0.23894	0.06461

Surface sampling parameter estimates are for the SNP-ARCH likelihood (29) estimated from daily, inflation adjusted returns on the S&P500 and NASDAQ indices (including distributions) from January 1, 2010, to December 31, 2018 under moment conditions (2) as determined by (30) through (33). The prior for ρ is independent normal with location and scale the SNP-ARCH unconstrained maximum likelihood estimated parameters and standard errors. The prior for $\theta = (a_0, a_1, a_2)$ is independent normal with means $(0, -1, 0)$ and standard deviations $(1, 1, 1)$. The support conditions are normalizing sign restrictions on variance parameters and that the eigenvalues of the companion matrices for location and scale are less than one in absolute value. Average autocorrelations for both the linear and quadratic MCMC chains at lag 50 were 0.87. The estimates shown are the those with the highest posterior mode in 100,000 draws collected after transients had dissipated. Estimates of scale are computed as described in Subsection 2.3 with tuning parameter $\Delta = 4$. The normalizing constants are computed with $n_i = 10,000$ as -6154.5364 and -6153.0577 for the linear and quadratic models, respectively, using the algorithm described in Subsection 2.4, giving a posterior probability of 0.81 in favor of the quadratic SDF. The λ-prior estimates are computed with $\lambda = 10^{12}$ from 50,000 MCMC draws for the same likelihood, moment conditions, prior, and support conditions.

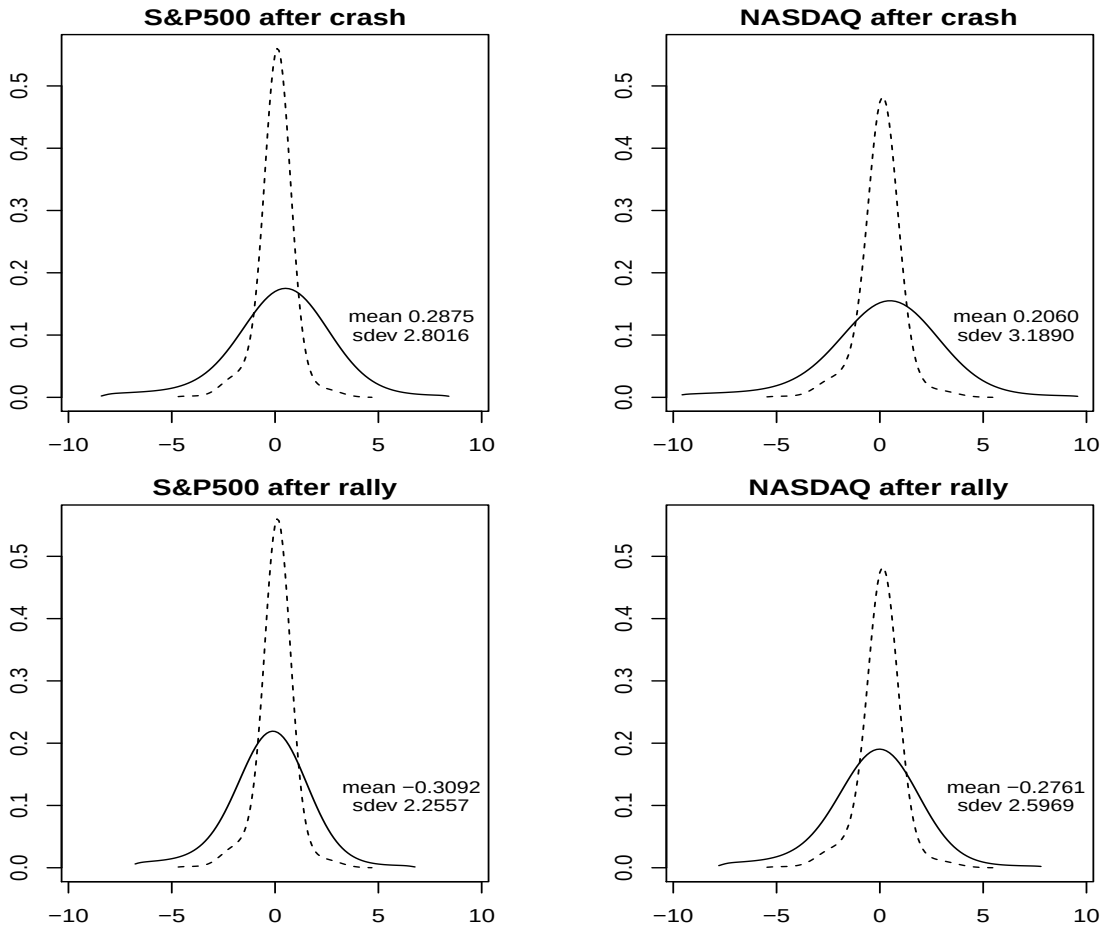

Figure 1. Crash and Rally The SNP-ARCH conditional density $f(y \mid x_{t-1}, \rho)$ given by (29) is plotted for the day after the conditioning event (solid line). The density is estimated subject to the moment conditions (2) determined by (30) through (33) and evaluated at the parameters shown for quadratic in Table 3. The conditioning events are the largest crash, $x_{t-1} = (-6.65, -7.15, -0.07, -0.94)$, which occurred on the NASDAQ on August 8, 2010, and the largest rally, $x_{t-1} = (4.98, 5.67, -2.69, -2.24)$, which occurred on the NASDAQ on December 26, 2018. Overplotted is a nonparametric estimate of the stationary density of the data whose first four moments reasonably match those shown in Table 2. Means and standard deviations in the figures refer to the solid line.