Nonparametric Bayes Subject to Overidentified Moment Conditions

A. Ronald Gallant
Penn State University

Paper: http://www.aronaldg.org/papers/npb.pdf
Slides: http://www.aronaldg.org/papers/npbclr.pdf
Code: http://www.aronaldg.org/webfiles/npb

Outline

- Brief Introduction to Bayesian Estimation
  - Best reference: Lindley, Dennis (1985), *Making Decisions, Second Edition*, Wiley.

- MCMC
  - Best reference: Gamerman, D., and Lopes, H. F., (2006), *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference (2nd Edition)*, Chapman & Hall.

- Nonparametric Bayes Subject to Overidentified Moment Conditions

- Comparisons

Bayesian Inference

- Bayesian inference is based on the posterior, which is the likelihood times the prior divided by a normalization factor: $p(\theta \,|\, x) = \ell(\theta \,|\, x)\pi(\theta) / \int \ell(\theta \,|\, x)\pi(\theta)\, d\theta$
  - E.g., to get a confidence interval, integrate an indicator function with respect to the posterior. E.g., $P[\theta_i \in (a,b)] = \int I(a < \theta_i < b) p(\theta \,|\, x) d\theta$
- The normalization factor is hard to compute.
- MCMC allows one to sample the posterior without knowing the normalization factor.
  - E.g., to get a confidence interval, average an indicator function over the MCMC draws.
- A GMM criterion function times a Jacobian term can be used as a likelihood. $\ell(\theta \,|\, x) = J(x,\theta) \exp\left[\frac{\sqrt{n}}{2} \bar{m}'(x,\theta) W^{-1}(x,\theta) \bar{m}(x,\theta)\right]$
  - Gallant, A. Ronald (2020), "Complementary Bayesian Method of Moments Strategies," *Journal of Applied Econometrics* 35, 422–439.

MCMC

- Posterior: $p(\theta \,|\, x) = \frac{\ell(\theta \,|\, x)\pi(\theta)}{\int \ell(\theta \,|\, x)\pi(\theta)\, d\theta}$

- Proposal transition density: $T(\theta_{old}, \theta_{new})$

- Proposal: Draw θ_{prop} from $T(\theta_{old}, \theta)$

- Put θ_{new} to θ_{prop} with probability

$$\alpha = \text{min}\left[1, \frac{\pi(\theta_{prop})\ell(\theta_{prop})T(\theta_{prop}, \theta_{old})}{\pi(\theta_{old})\ell(\theta_{old})T(\theta_{old}, \theta_{prop})}\right]$$

- Put θ_{new} to θ_{old} with probability $1 - \alpha$.

- If θ_{old} is distributed as $p(\theta \,|\, x)$, then so is θ_{new}.

Why Does This Work?

Let x be the old and y the new and let $f(\cdot)$ be the product of the prior and the likelihood of the previous slide. The proposal density is $T(x,y)$ and the transition density determined by the chain is

$$A(x,y) = T(x,y)\,\mathrm{min}\left\{1, \frac{f(y)T(y,x)}{f(x)T(x,y)}\right\}$$

for $y \neq x$ and

$$A(x,x) = 1 - \int I(x,y)\,A(x,y)\,dy,$$

where

$$I(x,y) = \begin{cases} 1 & y \neq x \\ 0 & y = x \end{cases}$$

Detailed Balance

For $x \neq y$

$$f(x)A(x,y) = \min\{f(x)T(x,y), f(y)T(y,x)\}$$

which implies that $f(x)A(x,y)$ is symmetric, i.e. that

$$f(y)A(y,x) = f(x)A(x,y).$$

Symmetry holds trivially for $x = y$.

This symmetry condition is called the detailed balance condition and implies, among other things, that the chain defined by $A(x,y)$ is reversible.

Conditional Expectation

Let

$$I(x,y) = \begin{cases} 1 & y \neq x \\ 0 & y = x \end{cases}$$

Then

$$\mathcal{E}\Big[g(Y)|x\Big] = \int g(y) I(x,y) A(x,y)\, dy + g(x) A(x,x)$$

Unconditional Expectation

$$\int \mathcal{E}[g(Y)|x] f(x)\, dx$$

$$\begin{aligned}
&= \int\!\!\int g(y) I(x,y) A(x,y) f(x) dx dy + \int g(x) A(x,x) f(x) dx \\
&= \int\!\!\int g(y) I(x,y) A(y,x) f(y) dx dy + \int g(x) A(x,x) f(x) dx \\
&= \int g(y) f(y) \int I(x,y) A(y,x) dx dy + \int g(x) A(x,x) f(x) dx \\
&= \int g(y) f(y) [1 - A(y,y)]\, dy + \int g(x) A(x,x) f(x)\, dx \\
&= \int g(y) f(y)\, dy
\end{aligned}$$

Stationary Density of the Chain

The fact that the equation

$$\int \mathcal{E}[g(Y)|x]f(x)\,dx = \int g(y)f(y)\,dy$$

holds for all integrable $g(y)$ implies that $f(y)$ is the stationary density of the MCMC chain with transition density $A(x, y)$.

Bayes Subject to Moment Conditions

The parameters $(\rho, \theta) \in \mathbb{R}^{d_a}$ of the likelihood

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

are to be estimated subject to the moment conditions

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

the support conditions

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

and the prior

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

Nonparametric Bayes

- Bayesian estimation can be regarded as nonparametric when

$$f(y_t \mid x_{t-1}, \rho)$$

is a sieve.

- A sieve is a density with a variable number K of parameters

$$\rho = (\rho_1, \rho_2, \ldots, \rho_K)$$

that is dense for some norm, e.g. Sobolev norm, as $K \to \infty$.

  – We use the SNP time series sieve in the application (Gallant and Tauchen, 1989, ECTA).

  – Which paper considers the same problem as here from a frequentist perspective.

A Much Better Bayesian GMM

With respect to Bayesian GMM al. la. Chernozhukov and Hong (2003, JoE)

- Same asymptotic efficiency (were one a frequentist)

- No continuously updated weighting matrix

- No auxiliary distributional assumption.

- No missing Jacobian term

Overidentification

- The support of the posterior is the manifold

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

- The problem is interesting when θ is overidentified, i.e., when the dimension m of q is larger than the dimension of θ because then M is singular with respect to Lebesgue measure on R^{d_a}.
  - Whence standard MCMC (Markov Chain Monte Carlo) methods cannot be used to estimate (ρ, θ).
  - Otherwise the problem is boring.

Clash of Notation

To adhere to the notational conventions of both the econometric and numerical analysis literature:

- Italic represents data: x_t, y_t, x, y
  - x_t, y_t are what is observed at time t, have a fixed number of rows, but the columns of x_t, the information set, can increase with t if $f(y_t|x_t, \rho)$ is recursive.
  - x contains all the observed x_t and y the same for y_t
- Sans serif represents parameters: x, y, X_k, Y_k
  - x and y represent values of (ρ, θ)
  - $\mathsf{X_k}$, $\mathsf{Y_t}$ represent either (ρ, θ) considered as a random variable or their *ex post* values as draws in an MCMC chain.

Relevant Literature

- Born, Shephard, and Solgi (2018, JRSSb)

- Shin (2015, Working paper),

- Schennach (2005, Biometrika)

- Gallant, Hong, Leung, and Li (2019, Working paper)

- 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.

Computing the Integral – 1

- Start with a univariate Gauss-Hermite rule

$$\int g(u)e^{-\frac{1}{2}u^2}du \doteq \sum_{i=1}^{I} \tilde{w}_i \, g(\tilde{u}_i). \tag{6}$$

  – Critical: make sure that the 5% and 95% quantiles of the elements $y_{i,t}$ of the data are within the min and max of the of the u_i.

  – Either increase I or rescale the data if not.

Computing the Integral – 2

- Multivariate Gauss-Hermite rule

$$\int \cdots \int g(y_1, \ldots, y_J)\, dy_1, \ldots, dy_J \doteq \sum_{k=1}^{K} \frac{w_k}{e_k} g(y_k), \qquad (7)$$

- For λ_k a permutation of $\{1, 2, \ldots, I\}$
  - $y_k = (\tilde{u}_{\lambda_{1,k}}, \ldots, \tilde{u}_{\lambda_{J,k}})$
  - $w_k = \prod_{j=1}^{J} \tilde{w}_{\lambda_{j,k}}$
  - $e_k = \prod_{j=1}^{J} \exp\left(-\frac{1}{2}(\tilde{u}_{\lambda_{j,k}})^2\right)$

- This form because analytic derivatives of $m(y_k, x_{t-1}, \rho, \theta)$ are required and would be a nightmare to obtain if y_k and w_k depended on (ρ, θ), which would be the case for a standard rule.

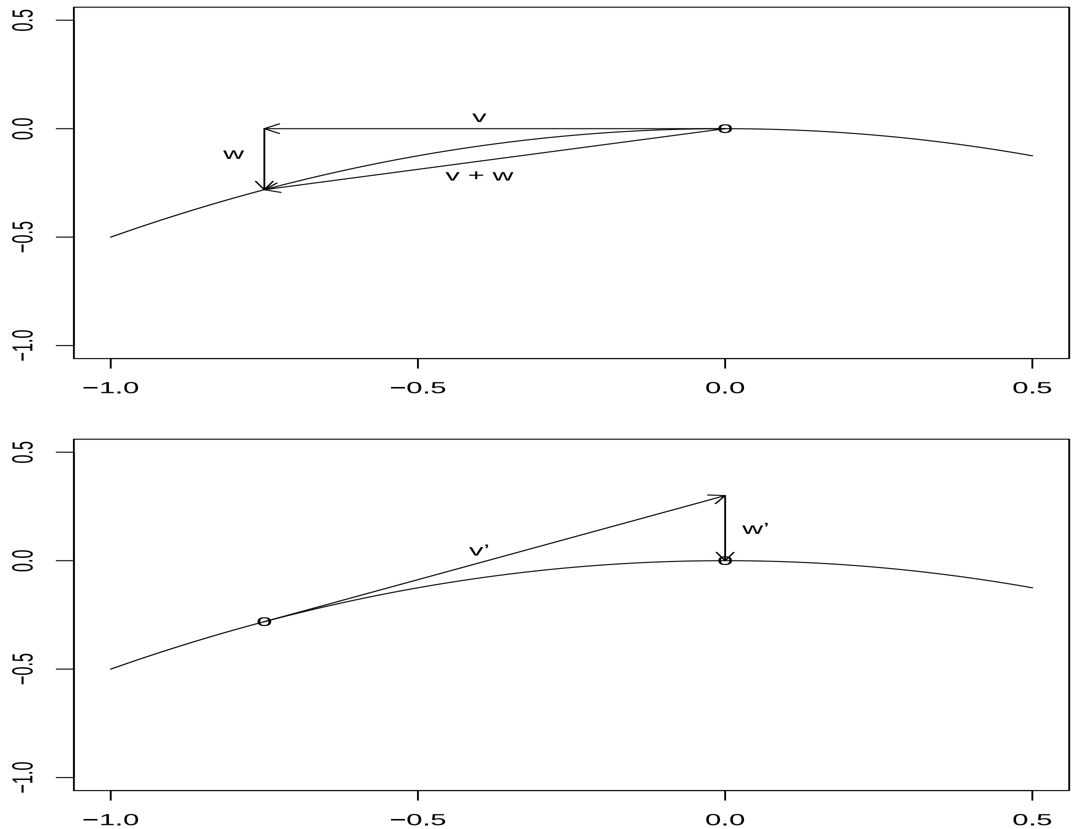

Figure 1. Zappa et al's Innovation The embedded manifold M given by (5) is illustrated by the curved line. The upper panel shows a move at the Projection Step of the Surface Sampling Algorithm. It consists of a move v tangent to M followed by a perpendicular move w onto M. The lower panel shows the Reverse Projection Step. If the proposed move is accepted and the reverse projection succeeds, then the draw satisfies the detailed balance condition on M. The nonlinear equation solver used to compute w and w' must be same in both instances.

Surface Sampling Algorithm: Begin

- **Begin:** $\mathsf{X}_k = \mathsf{x} = (\rho, \sigma)$
  - X_k must be in M
  - Use λ-prior method, described later, to find X_k
- Notation for subsequent steps
  - Q_{x} is the transpose of the Jacobian of $q(\mathsf{x})$
  - Apply SVD algorithm to $A = [Q_{\mathsf{x}}|0] \rightarrow [T_{\mathsf{x}}^{\perp}|T_{\mathsf{x}}]$
  - $p(v)$ the proposal density for v shown in Figure 1
  - $\mathsf{x} \in \mathbb{R}^{d_a}$, $q(\mathsf{x}) \in \mathbb{R}^m$, $d = d_a - m$, $T_{\mathsf{x}}^{\perp}$ is $d_a \times m$, T_{x} is $d_a \times d$.

Surface Sampling Algorithm: Proposal

- **Proposal:**

  1. Calculate Q_{x}, the transpose of the Jacobian of $q(\mathsf{x})$
  2. Compute $T_{\mathsf{x}}^{\perp}$ and T_{x} using the SVD as described above.
  3. Draw z from $N_d(0, s^2 I)$; $v = T_{\mathsf{x}} z$ is the draw from $p(v)$.

Surface Sampling Algorithm: Projection

- **Projection:**

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

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

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

Surface Sampling Algorithm: Inequality Check

- **Inequality check:**

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

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

Surface Sampling Algorithm: Metropolis Step

- **Metropolis-Hastings acceptance/rejection step:**
  1. Calculate Q_{y}
  2. Compute $T_{\mathsf{y}}^{\perp}$ and T_{y} using the SVD as described above.
  3. Find $v' \in T_{\mathsf{y}}$ and $w' \in T_{\mathsf{y}}^{\perp}$ so that $\mathsf{x} = \mathsf{y} + v' + w'$.*
  4. $P_a = \min\left(1, \frac{f(y \,|\, \mathsf{y})\pi(\mathsf{y})p(v')}{f(y \,|\, \mathsf{x})\pi(\mathsf{x})p(v)}\right)$
  5. Generate $U \sim$ Uniform(0,1).
  6. If $U > P_a$, put $\mathsf{X}_{k+1} = \mathsf{x}$. **Done**.
  7. Else **Continue.**

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

Surface Sampling Algorithm: Reverse Projection Step

- **Reverse Projection:**

  1. Solve $q(\mathsf{y} + v' + Q_{\mathsf{y}} a) = 0$ for a using Newton's method.
  2. If Newton's method fails, put $\mathsf{X}_{k+1} = \mathsf{x}$. **Done**.
  3. Else accept move, $\mathsf{X}_{k+1} = \mathsf{y}$. **Done**.

λ-prior Method

- Used to get starting values for the Surface Sampling Algorithm.

- The λ-prior method is simple: Draw from the posterior

$$p(\rho, \theta \,|\, y, x) \propto f(y \,|\, x, \rho)\pi(\rho, \theta)\pi_\lambda(\rho, \theta) \tag{8}$$

  by MCMC subject to the support conditions (3), where

$$\pi_\lambda(\rho, \theta) = \exp\left[-\lambda \frac{n}{2} \sum_{i=1}^{m} q_i^2(\rho, \theta)\right]. \tag{9}$$

  - Large λ forces the (ρ, θ) draws to be near M.

  - Will fail for λ too large because M is singular.

Standard Deviations

- 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 some norm defined on M.
  - 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 $D_M(\bar{\mathsf{x}},p)D^\top(\bar{\mathsf{x}},p)$ 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.

Geodesics from a Point Cloud

- All we have are the Surface Sampling MCMC draws.
  - Which lie on the d-dimensional submanifold $M \subset \mathbb{R}^{d_a}$,
  - $d < d_a$
- The question becomes how to compute a geodesic on a manifold when one only has a point cloud.

Geodesics – 1

- Distance along a geodesic satisfies the intrinsic Eikonal distance equation

$$\begin{aligned} \|\nabla_M \delta_M(s,p)\| &= 1 \quad p \in M \\ \delta_M(s,s) &= 0 \end{aligned} \tag{10}$$

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.

Geodesics – 2

- 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

$$\begin{aligned} &\|\nabla\delta(s,p)\| = 1 \quad p \in M_\epsilon \\ &\delta(s,s) = 0 \end{aligned} \tag{11}$$

  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.

Geodesics – 3

- Standard algorithms for the solution of (10) produce as a by-product the geodesic that connects the starting point s to the end point p.

Fast Marching Algorithm

- The Fast Marching Algorithm (Sethian, 1996, Proc.Natl.Acad.) is frequently used to solve (10)
  - Memoli and Sapiro (2001, Comp.Phsics.) provide the upwind equation and the neighbor checking modification to adapt the Fast Marching Algorithm to a point cloud.

- Unfortunately, the Fast Marching Algorithm requires that M_ϵ be placed on a Euclidean grid which limits the Fast Marching Algorithm to problems where $d_a < 5$

Dijkstra's Algorithm

- 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 $e_{j,j'}$ with length $\delta(p_j, p_{j'})$.

- From a start s, Dijkstra's algorithm finds the shortest path that traverses edges to every node p_j. (Dijkstra, 1959, Numerische Mathematik)

- Distances will be larger than those of the Fast Marching Algorithm because the Fast Marching Algorithm is not constrained to follow edges.

- Used by Google Maps.

Tuning Dijkstra's Algorithm – 1

- The ϵ that determines the graph $\mathcal{G}_\epsilon$ is a tuning parameter.

- 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.

- Way too large is the same as computing sample variance matrix directly from the MCMC draws.

Tuning Dijkstra's Algorithm – 2

- Upper bound: increase ϵ until standard errors are larger than returned by the λ-prior method but reasonable relative to the λ-prior method.

- Lower bound: Computing sample variance matrix directly from the MCMC draws $\mathcal{D} = \{\mathrm{x}_i\}_{i=1}^{N}$.

Normalization Constant

- The normalizing constant, aka marginal likelihood or marginal data density, is

$$Z = \int_M f(y \,|\, \mathrm{x}) \, \pi(\mathrm{x}) \, d\sigma(\mathrm{x}), \tag{12}$$

  where $\sigma(\mathrm{x})$ is d-dimensional Hausdorff measure on $\mathbb{R}^{d_a}$.

- If a mapping from $\mathbb{R}^d$ to M can be found, then computing (12) can be accomplished by Riemann integration after multiplication by a Jacobian term

- The strategy is to reduce the domain of integration until a mapping can be found.

- The remaining part of the integral can be computed from Surface Sampling draws.

Reduction via Concentric Balls

- x_0 the estimated posterior mode
- $\mathcal{D}^{\rceil}{}_0 = \{\mathsf{x}_i\}_{i=1}^{n_0}$ be n_0 draws with duplicates that occur in succession deleted
- Compute the Euclidean norms $\mathcal{N}_0 = \{\|\mathsf{x} - \mathsf{x}_0\| : \mathsf{x} \in \mathcal{D}_0^e\}$.
- $r_0 = \max \mathcal{N}_0$, r_1 the 90th percentile, r_2 the 80th, and so on until r_9 the 10th.
- B_i a closed ball in $\mathbb{R}^{d_a}$ with center x_0 and radius r_i.
- $B_0 \supset B_1 \supset \ldots \supset B_9$.

Domain Reduction

- 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 $\frac{Z_{i+1}}{Z_i} = \frac{1}{Z_i} \int_{M \cap B_i} I_{B_{i+1}}(\mathsf{x}) f(y \,|\, \mathsf{x})\, \pi(\mathsf{x})\, d\sigma(\mathsf{x})$.

- Append $\|\mathsf{x} - \mathsf{x}_0\| <= r_i$ to the support conditions (3), generate n_i draws, let $N_{i,i+1}$ be those draws that are in B_{i+1}.

- A estimate of $\frac{Z_{i+1}}{Z_i}$ is $\frac{N_{i,i+1}}{n_i}$, whence $\widehat{R}_i = \frac{n_i}{N_{i,i+1}}$.

Find k

- Start at $k = 5$
- Compute Q_{x} and T_{x} as described earlier
- For $i = 1, \ldots, n_k$, draw u_i from the uniform distribution on a ball of dimension d and radius r_k
- Put $v_i = T_{\mathsf{x}_0} u_i$ and project to $y_i \in M$ as described earlier
- If projection fails for some i, abort, increase the guessed value for k by one, and repeat.

Compute Z_k

- Jacobian is $J_i = \det(T_{x_0}^{\top} T_{y_i})$.

- Compute Z_k by Monte Carlo integration as follows:

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

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

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

Example: CRRA Moment Function

- Parameter: $\theta = (\beta, \gamma) =$ (discount factor, risk aversion)

- Data: $x_t = \begin{pmatrix} x_{1t} \\ x_{2t} \end{pmatrix} = \begin{pmatrix} \mathsf{lsr}_t \\ \mathsf{lcg}_t \end{pmatrix} = \begin{pmatrix} \text{log stock returns} \\ \text{log endowment growth} \end{pmatrix}$

- Moments: $m(x_t, \theta) = \begin{pmatrix} 1 \\ \mathsf{lsr}_{t-1} \\ \mathsf{lcg}_{t-1} \end{pmatrix} [1 - \exp(\log \beta - \gamma \mathsf{lcg}_t + \mathsf{lsr}_t)]$

- Adjustment: $\mathsf{adj}(x, \theta) = 4(1 - e)^2 \left| \frac{1 - \tanh\left(\frac{1}{4} z_1\right)}{1 - \left[\tanh\left(\frac{1}{4} z_1\right)\right]^2} \right|$

Moment Weighting

Setup: Classical GMM with selective weighting.
I.e., Diagonal matrix with ones and zeros along the diagonal

Moments	β	γ
e & e×lsr	0.9994	3.96
e & e×lcg	0.9811	0.44
e×lsr & e×lcg	0.9999	3.88
e & e×lsr & e×lcg	0.9993	3.94

Conclusion: One can produce any desired answer with GMM by choosing the weighting matrix appropriately.

Example: SNP Sieve

$f(x_t \,|\, x_{t-1}, x_{t-2}, x_{t-3}, \ldots, \rho)$ (recursive)

- $x_t = \begin{pmatrix} x_{1t} \\ x_{2t} \end{pmatrix} = \begin{pmatrix} \mathsf{lsr}_t \\ \mathsf{lcg}_t \end{pmatrix} = \begin{pmatrix} \text{log stock returns} \\ \text{log endowment growth} \end{pmatrix}$

- Garch(1,1)
  - Diagonal ARCH term
  - Scalar GARCH term

- Hermite error density
  - Polynomial of degree four in u times a normal density $n(u \,|\, 0, \Sigma)$

Example: Support and Prior

- Support: $0 < \beta < 0.99999 \quad 0 < \gamma < 100$

- Prior: $n(\beta \,|\, 0.9975, 0.001^2) \times n(\gamma \,|\, 4.00, 2.00^2)$
  0.9975 quarterly discount is 0.99 annual.

Table 1. Estimates

	W=I		2SL2		Cont. Up.		λ-Prior		NP Bayes		
parm	est	sdev	est	sdev	est	sdev	est	sdev	est	lo sdev	hi sdev
a_{01}							0.2149	0.0713	0.2254	0.0769	0.24839
a_{02}							0.0608	0.0597	0.0732	0.0578	0.19428
a_{03}							-0.0862	0.0294	-0.0774	0.0319	0.09312
a_{04}							0.0805	0.0274	0.0816	0.0295	0.10939
a_{05}							-0.0121	0.0501	-0.0539	0.0408	0.08623
a_{06}							-0.0742	0.0542	-0.0367	0.0393	0.10223
a_{07}							-0.0738	0.0317	-0.0521	0.0258	0.07523
a_{08}							0.0953	0.0340	0.0918	0.0351	0.09023
$b_{0,1}$							0.0584	0.0454	0.0946	0.0471	0.12701
$b_{0,2}$							-0.3341	0.1153	-0.3166	0.1269	0.40114
$B_{1,1}$							0.0572	0.0592	0.0153	0.0141	0.05128
$B_{2,1}$							0.2490	0.0558	0.2294	0.0478	0.16735
$B_{1,2}$							-0.0887	0.0443	0.0034	0.0117	0.02781
$B_{2,2}$							0.1690	0.0359	0.1369	0.0396	0.10595
$R_{0,1,1}$							0.3059	0.0338	0.2952	0.0359	0.08214
$R_{0,1,2}$							-0.0207	0.0156	-0.0293	0.0150	0.04815
$R_{0,2,2}$							0.4657	0.0400	0.4685	0.0373	0.09390
$P_{1,1}$							0.5492	0.0509	0.5500	0.0553	0.16231
$P_{2,2}$							-0.0551	0.0701	-0.0677	0.0682	0.20625
$Q_{1,1}$							0.8344	0.0290	0.8220	0.0320	0.09005
β	0.9975	0.0010	0.9974	0.0010	0.9974	0.0010	0.9980	0.0010	0.9980	0.0010	0.00323
γ	3.9844	1.8386	3.1116	0.7195	3.0416	0.7226	4.5299	1.2248	3.0500	0.7174	1.73501