

Lecture Note 9: Method of Simulated Scores, Gibbs Sampling, Bayes

We will continue working with the multinomial probit model; however the ideas here will apply to many similar models with latent variables.

Recall that the log likelihood in the MNP model was:

$$L(\theta) = \sum_{i=1}^n \sum_{c=1}^C d_{ic} \log \Pr(d_{ic} = 1 | X_i, \theta).$$

If we approximate $\Pr(d_{ic} = 1 | X_i, \theta)$ by the simple frequency estimator $\tilde{P}_S(d_{ic} = 1 | X_i, \theta)$, and maximize the approximate likelihood, this tends to work poorly unless S is extremely large. One problem that will arise is that we will occasionally get $\tilde{P}_S(d_{ic} = 1 | X_i, \theta) = 0$ (or close to 0).

McFadden suggested simulating $\Pr(d_{ic} = 1 | X_i, \theta)$, but using a method of moments formulation based on the viewing $\Pr(d_{ic} = 1 | X_i, \theta)$ as the conditional mean of d_{ic} . Then, simulation error adds noise to the moment equation but does not invalidate it even for finite S . In practice, there will be some loss of efficiency (relative to the infeasible ML estimator) unless one chooses the instruments carefully and uses a large number of simulations S .

Hajivassiliou and McFadden (1998) suggest an alternative approach based on simulating the score functions, and point out that a technique called Gibbs sampling can be used here.

Recall in the MNP model we set up last time,

$$X_i = \begin{pmatrix} x'_{i1} \\ \vdots \\ x'_{iC} \end{pmatrix}.$$

$$u_i = \begin{pmatrix} u_{i1} \\ \vdots \\ u_{iC} \end{pmatrix} | X_i, \theta \sim N(X_i \beta(\theta), X_i \Omega(\theta) X_i').$$

Simplify the notation by writing

$$\Omega_i(\theta) = X_i \Omega(\theta) X_i'.$$

Vector of choice indicators:

$$d_i = \begin{pmatrix} d_{i1} \\ \vdots \\ d_{iC} \end{pmatrix}.$$

Let τ be the function taking u_i into d_i :

$$d_i = \tau(u_i).$$

Let $T(\cdot)$ be the (set-valued) inverse:

$$T(d_i) = \{u_i : d_i = \tau(u_i)\}.$$

Then we can write the likelihood of the i th observation as:

$$\begin{aligned} l(\theta, d_i, X_i) &= \int_{T(d_i)} dN(u_i | X_i \beta(\theta), \Omega_i(\theta)) \\ &= \int_{T(d_i)} (2\pi)^{-C/2} |\Omega_i|^{-1/2} \exp\left(-\frac{1}{2}(u_i - X_i \beta)' \Omega_i^{-1} (u_i - X_i \beta)\right) du_i \end{aligned}$$

(Keep in mind that β and Ω_i are functions of θ .)

ML estimator:

$$\hat{\theta}_{ML} = \arg \max_{\theta} \frac{1}{n} \sum_{i=1}^n \log l(\theta, d_i, X_i).$$

Score equation:

$$\begin{aligned} 0 &= \frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial \theta} \log l(\hat{\theta}_{ML}, d_i, X_i) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{\frac{\partial}{\partial \theta} l(\hat{\theta}_{ML}, d_i, X_i)}{l(\hat{\theta}_{ML}, d_i, X_i)} \\ &= \frac{1}{n} \sum_{i=1}^n s_i(\hat{\theta}_{ML}, d_i, X_i). \end{aligned}$$

The idea is to replace s_i by a simulated version. For the MNP model (and many similar models), the score function turns out to have a particularly convenient form.

To derive a more explicit expression for

$$s_i(\theta, d_i, X_i) = \frac{\frac{\partial}{\partial \theta} l(\theta, d_i, X_i)}{l(\theta, d_i, X_i)},$$

we need to calculate $\frac{\partial}{\partial \theta} l(\theta, d_i, X_i)$. Notice that θ enters the likelihood through β and Ω_i . So first calculate

$$\frac{\partial}{\partial \beta} l(\theta, d_i, X_i)$$

$$\begin{aligned}
&= \frac{\partial}{\partial \beta} \int_{T(d_i)} (2\pi)^{-C/2} |\Omega_i|^{-1/2} \exp\left(-\frac{1}{2}(u_i - X_i\beta)' \Omega_i^{-1} (u_i - X_i\beta)\right) du_i \\
&= \int_{T(d_i)} (2\pi)^{-C/2} |\Omega_i|^{-1/2} \frac{\partial}{\partial \beta} \exp\left(-\frac{1}{2}(u_i - X_i\beta)' \Omega_i^{-1} (u_i - X_i\beta)\right) du_i \\
&= \int_{T(d_i)} (2\pi)^{-C/2} |\Omega_i|^{-1/2} \exp\left(-\frac{1}{2}(u_i - X_i\beta)' \Omega_i^{-1} (u_i - X_i\beta)\right) \frac{\partial}{\partial \beta} \left(-\frac{1}{2}(u_i - X_i\beta)' \Omega_i^{-1} (u_i - X_i\beta)\right) du_i
\end{aligned}$$

Using the fact that $\frac{d}{dx} x'Ax = x'(A + A')$, this equals

$$\begin{aligned}
&\int_{T(d_i)} (2\pi)^{-C/2} |\Omega_i|^{-1/2} \exp\left(-\frac{1}{2}(u_i - X_i\beta)' \Omega_i^{-1} (u_i - X_i\beta)\right) (u_i - X_i\beta)' X_i' \Omega_i^{-1} du_i \\
&= l(\theta, d_i, X_i) X_i' \Omega_i^{-1} \frac{\int_{T(d_i)} (u_i - X_i\beta) dN(u_i | X_i\beta, \Omega_i)}{\int_{T(d_i)} dN(u_i | X_i\beta, \Omega_i)} \\
&= l(\theta, d_i, X_i) X_i' \Omega_i^{-1} E[(u_i - X_i\beta) | u_i \in T(d_i), X_i, \theta]
\end{aligned}$$

To take the derivative with respect to the variance, let Γ_i satisfy $\Gamma_i \Gamma_i' = \Omega_i$. Then similar calculations give

$$\frac{\partial}{\partial \Gamma_i} l(\theta, d_i, X_i) = -l(\theta, d_i, X_i) \Omega_i^{-1} [I_C - E[(u_i - X_i\beta)(u_i - X_i\beta)' | u_i \in T(d_i), X_i, \theta] \Omega_i^{-1}] \Gamma_i$$

So the derivative with respect to θ has the form

$$\frac{\partial}{\partial \theta} l(\theta, d_i, X_i) = l(\theta, d_i, X_i) E[h(u_i - X_i\beta) | u_i \in T(d_i), X_i, \theta],$$

where $h(v)$ is a vector of partial derivatives of β and Γ_i with respect to θ multiplied by

$$\begin{bmatrix} X_i' \Omega_i^{-1} v \\ -\Omega_i^{-1} [I_C - vv' \Omega_i^{-1}] \Gamma_i \end{bmatrix}.$$

Thus the score function is

$$\begin{aligned}
s_i(\theta, d_i, X_i) &= \frac{\frac{\partial}{\partial \theta} l(\theta, d_i, X_i)}{l(\theta, d_i, X_i)} \\
&= E[h(u_i - X_i\beta) | u_i \in T(d_i), X_i, \theta]
\end{aligned}$$

To simulate the score, we need to be able to generate random draws from the distribution

$$u_i | u_i \in T(d_i), X_i, \theta.$$

This is not trivial due to the constraint $u_i \in T(d_i)$.

Hajivassiliou and McFadden suggest some different simulators for $u_i | u_i \in T(d_i), X_i, \theta$. We will focus on one simulator, called the Gibbs sampler, because it seems to work well in practice, and ends up

being very useful for Bayesian simulators to be studied next.

Gibbs Sampling

Suppose we want to simulate a random vector $Y = (Y_1, Y_2, \dots, Y_J)$ distributed according to a joint distribution F , but it is difficult to draw directly from the joint distribution. If we can draw from various conditional distributions of F , then we can construct a Markov chain that eventually converges to the joint distribution.

Simulation Algorithm:

- First, initialize $y^0 = (y_1^0, \dots, y_J^0)$ at some arbitrary values.
- Draw y_1^1 from the conditional distribution $F(Y_1|Y_2 = y_2^0, Y_3 = y_3^0, \dots, Y_J = y_J^0)$.
- Draw y_2^1 from the conditional distribution $F(Y_2|Y_1 = y_1^1, Y_3 = y_3^0, \dots, Y_J = y_J^0)$.
- Draw y_3^1 from the conditional distribution $F(Y_3|Y_1 = y_1^1, Y_2 = y_2^1, Y_4 = y_4^0, \dots, Y_J = y_J^0)$.
- ...
- Draw y_J^1 from the conditional distribution $F(Y_J|Y_1 = y_1^1, Y_2 = y_2^1, \dots, Y_{J-1} = y_{J-1}^1)$
- Draw y_1^2 from the conditional distribution $F(Y_1|Y_2 = y_2^1, Y_3 = y_3^1, \dots, Y_J = y_J^1)$
- ...

So, for $g = 1, \dots, G$,

- Y_1^g is drawn from $F(Y_1|Y_2 = y_2^{g-1}, Y_3 = y_3^{g-1}, \dots, Y_J = y_J^{g-1})$
- Y_2^g is drawn from $F(Y_2|Y_1 = y_1^g, Y_3 = y_3^{g-1}, \dots, Y_J = y_J^{g-1})$
- and so on.

Then $Y^g = (Y_1^g, \dots, Y_J^g)$ is a Markov chain: its distribution in round g depends on its value in round $g - 1$. Note that this Markov chain may have a continuous state space.

It can be shown that the invariant distribution of this Markov chain is the desired joint distribution F . That is, if Y^g were drawn from F , then Y^{g+1} would also have distribution F .

Under relatively weak conditions, the Markov chain converges to its invariant distribution. That is, for any starting value Y^0 , the distribution of Y^G converges to F .

The idea is to start the chain from some arbitrary value, iterate G times, and take Y^G as a draw from F .

Example: MNP latent utilities with $C = 3$

We want to simulate $u_i = (u_{i1}, u_{i2}, u_{i3})'$ from its distribution conditional on d_i .

Suppose $d_i = (1, 0, 0)'$, so that the first option was chosen by individual i . So conditioning on $u_i \in T(d_i)$ means conditioning on $u_{i1} \geq u_{i2}$ and $u_{i1} \geq u_{i3}$.

- Initialize $u_{i1}^0 = u_{i2}^0 = u_{i3}^0 = 0$.
- Draw $u_{i1}^1 | u_{i2}^0 = 0, u_{i3}^0 = 0$. This distribution is a normal distribution truncated so that $u_{i1} \geq 0$.
- Draw $u_{i2}^1 | u_{i1}^1, u_{i3}^0 = 0$. This distribution is normal, truncated so that $u_{i2}^1 \leq u_{i1}^1$.
- Draw $u_{i3}^1 | u_{i1}^1, u_{i2}^1$. This distribution is normal, truncated so that $u_{i3}^1 \leq u_{i1}^1$.
- \dots .

Iterating this G rounds, we get $u_i^G = (u_{i1}^G, u_{i2}^G, u_{i3}^G)$, which we treat as a random draw from $u_i | u_i \in T(d_i), X_i, \theta$.

In principle, G should go to infinity for the simulation to really generate a random draw from the desired distribution. Hajivassiliou and McFadden show that the convergence of the Gibbs sampling is “fast”, so that G only needs to grow at the rate $\log n$ for the bias in the score simulation to disappear.

In practice, the convergence is usually quite fast, so that this procedure works well even for fairly small G .

Note: usually we want to generate more than 1 simulation. So we would repeat the Gibbs sampling algorithm S times to get S simulations for $u_i | u_i \in T(d_i), X_i, \theta$. This is used to approximate

$$E[h(u_i - X_i\beta) | u_i \in T(d_i), X_i, \theta].$$

The MSS works for finite S , and is fully efficient (asymptotically equivalent to ML) if we let $S \rightarrow \infty$ at any rate.

Bayesian Estimation and Simulation

Recall the general Bayesian setup.

We have a parametric model with data

$$Z \sim P_\theta, \quad \theta \in \Theta.$$

Here you should think of Z as all the data observations, collected into a vector.

Let $p(z|\theta)$ denote the joint density of z given θ .

We select a prior distribution, a probability distribution on the parameter space. Write the prior density as $p(\theta)$. (We are abusing notation by using p for various densities, but it will be clear which one we mean by the arguments.)

This defines a joint density for (z, θ) :

$$p(z, \theta) = p(\theta)p(z|\theta).$$

We want to calculate the posterior

$$p(\theta|z) = \frac{p(z, \theta)}{p(z)} = \frac{p(\theta)p(z|\theta)}{\int p(\theta)p(z|\theta)d\theta}.$$

Note that the denominator $p(z) = \int p(\theta)p(z|\theta)d\theta$ does not depend on θ . It is a constant factor that normalizes the numerator $p(\theta)p(z|\theta)$ into a proper probability density. To emphasize this, we sometimes write

$$p(\theta|z) \propto p(\theta)p(z|\theta),$$

where \propto means “proportional to.”

In some cases, we can just calculate the numerator $p(\theta)p(z|\theta)$, and inspect the form of the unnormalized posterior density. If this is equal to some density (for θ) up to a constant, then we do not have to explicitly calculate the integral in the denominator.

This requires that the prior and the likelihood somehow “match” so that the unnormalized posterior has a nice form.

Example: Normal Model

Suppose that we observe random variables Y_i , $i = 1, \dots, n$, and we assume that

$$Y_i|\mu \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, 1), \quad i = 1, \dots, n \tag{1}$$

$$\begin{aligned} p(y_1, \dots, y_n|\mu) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y_i - \mu)^2\right) \\ &= \frac{1}{(2\pi)^{n/2}} \exp\left[-\frac{1}{2} \sum_{i=1}^n (y_i - \mu)^2\right]. \end{aligned}$$

We will suppose that μ is not known but is assumed to have a distribution with density $p(\mu)$. For concreteness, suppose $p(\mu)$ is a normal density with mean a and variance v . It will simplify some of the formulas if we work instead with the inverse of the variance $\tau := 1/v$. This is called the

precision. Then

$$p(\mu) = \frac{\sqrt{\tau}}{\sqrt{2\pi}} \exp\left(-\frac{\tau}{2}(\mu - a)^2\right). \quad (2)$$

The posterior is:

$$p(\mu|y_1, \dots, y_n) = \frac{p(\mu, y_1, \dots, y_n)}{p(y_1, \dots, y_n)} = \frac{p(\mu)p(y_1, \dots, y_n|\mu)}{p(y_1, \dots, y_n)}. \quad (3)$$

Recall that the denominator can be obtained by integrating the numerator with respect to μ :

$$p(y_1, \dots, y_n) = \int p(\mu, y_1, \dots, y_n) d\mu = \int p(\mu) f(y_1, \dots, y_n|\mu) d\mu$$

Intuitively, *the denominator is whatever constant c that makes*

$$p(\mu|y_1, \dots, y_n) = \frac{1}{c} \cdot p(\mu) f(y_1, \dots, y_n|\mu)$$

a proper density function, in the sense of integrating to 1. So we write

$$p(\mu|y_1, \dots, y_n) \propto p(\mu) f(y_1, \dots, y_n|\mu)$$

With some algebra, this can be shown to be proportional to:

$$\exp\left(-\frac{1}{2}(\tau + n)(\mu - \mu^*)^2\right), \quad (4)$$

where

$$\mu^* := \frac{\tau}{\tau + n} a + \frac{n}{\tau + n} \left(\frac{1}{n} \sum_{i=1}^n y_i \right)$$

The expression in (4) is proportional to the normal density for μ with mean μ^* and precision $(\tau + n)$.

We conclude that:

$$\mu|Y_1 = y_1, \dots, Y_n = y_n \quad \sim \quad \mathcal{N}(\mu^*, (\tau + n)^{-1}).$$

Notice that μ^* , the conditional mean of μ , is a weighted average of a and the sample mean $\sum_{i=1}^n y_i$, with the weight depending on τ and n .

Suppose we want to provide a *point estimate* d of μ , that minimizes the quadratic loss $(\mu - d)^2$. Since μ is regarded as having a distribution rather than being fixed, we could choose d to minimize

$$E((\mu - d)^2|y_1, \dots, y_n),$$

where the expectation is with respect to the conditional distribution of μ . Notice that as $\tau \rightarrow 0$ we get the “usual” estimator \bar{y} .