Lecture Notes 8:
Empirical Applications

Ivan Rudik

ECON 509: Computational Methods

March 9, 2016

In addition to computing solutions to analytical models, we can use numerical methods to estimate models with real world data. The most common approach is Maximum Likelihood Estimation (MLE). There are also estimation procedures, like Nested Fixed Point (NXFP) for estimating dynamic structural models (Rust, 1987). We will go over some basic approaches to estimating models with numerical methods

1 Maximum Likelihood

When entering the empirical world, there is a set of data generating processes (DGPs) that may have generated the data we are using. Suppose that our data consist of realizations of some random variable $Z \sim f_Z(Z;\theta^*)$ where θ^* is a vector of true parameters governing the DGP. The likelihood function is simply $\mathcal{L}(\theta) = f_Z(Z;\theta)$: the pdf evaluated at some vector θ. If there are multiple random variables, the likelihood is based on the joint pdf, $\mathcal{L}(\theta) = f_{Z_1,Z_2,...Z_n}(Z_1, Z_2, ..., Z_n; \theta)$. If these random variables, $Z_1, Z_2, ..., Z_n$ are IID, then the likelihood is simply the product of the marginal pdfs, $\mathcal{L}(\theta) = \prod_{i=1}^n f_{Z_i}(Z_i;\theta)$. For tractability, we prefer to work with the log likelihood, $\log\mathcal{L}(\theta) = \sum_{i=1}^n \log f_{Z_i}(Z_i;\theta)$.

The maximum likelihood estimator of θ^* is the value of θ that maximizes the likelihood or log likelihood function. In effect, it finds the model, within the class of models f_Z that best matches the observed data (selecting parameters that maximize the likelihood that the data were generated by this class of models).

We can think about a simple exam, suppose X is distributed normally with some unknown mean and unit variance. The pdf of X is,

$$f_X(x;\mu) = \frac{1}{\sqrt{2\pi}}\, exp\left(-\frac{1}{2}\,(x-\mu)^2\right).$$

The likelihood is therefore,

$$\mathcal{L}(\mu) = \frac{1}{\sqrt{2\pi}}\, exp\left(-\frac{1}{2}\,(X-\mu)^2\right),$$

and the log likelihood is,

$$log\mathcal{L}(\mu) = -\frac{1}{2}log(2\pi) - \frac{1}{2}(X-\mu)^2.$$

What value of μ, our unknown parameter, maximizes our log likelihood? Simply $\hat{\mu}_{ML} = X$. If we have n random variables that are IID $N(\mu, 1)$ then our log likelihood is,

$$\mathcal{L}(\mu) = -n\frac{1}{2}log(2\pi) - \sum_{i=1}^{n}\frac{1}{2}(X_i-\mu)^2.$$

The argmax of the log likelihood is just, $\hat{\mu}_{ML} = \bar{X}$, the sample average of the $X_i's$.

Now we know how to maximize a likelihood, but why is this a logical strategy for estimating parameters of a model? Because we can appeal to large sample approximations. Suppose we have some random variable Y which is the ratio of the density function at some arbitrary θ to the density function at θ^*, the true theta, both evaluated at some X,

$$Y = f_X(X;\theta)/f_X(X;\theta^*)$$

. Suppose $g(\cdot)$ is the negative logarithm, $g(a) = -log(a), g'(a) = -1/a, g''(a) = 1/a^2 > 0$. $g(a)$ is clearly convex so Jensen's inequality states that,

$$E[g(Y)] \geq g(E[Y]).$$

This implies that,

$$E\left[-log\left(\frac{f_X(X;\theta)}{f_X(X;\theta^*)}\right)\right] \geq -log\left(E\left[\frac{f_X(X;\theta)}{f_X(X;\theta^*)}\right]\right).$$

What is the expectation over? The distribution of X: $f_X(x;\theta^*)$. This implies that the expectation on the right hand side is,

$$E\left[\frac{f_X(X;\theta)}{f_X(X;\theta^*)}\right] = \int \frac{f_X(X;\theta)}{f_X(X;\theta^*)} f_X(X;\theta^*)dx = \int f_X(X;\theta)dx = 1.$$

So for any value of θ, after taking a log we have that,

$$E\left[-log\left(\frac{f_X(X;\theta)}{f_X(X;\theta^*)}\right)\right] \geq 0$$

. This implies that,

$$E\left[logf_X(X;\theta^*)\right] \geq E\left[logf_X(X;\theta)\right], \ \forall\theta.$$

This tells us that the expected value of our log likelihood is maximized at the true value of θ, so we have a shot that our actual log likelihood is maximized near θ^*, so that $\hat{\theta}_{ML}$ is a good estimate of θ^*. We can illustrate this by considering an example where we have n Bernoulli trials with some unknown probability p^*. The joint density of these trials is,

$$f_{X_1,...,X_n} = (x_1,...,x_n,;p) = p^{\sum x_i}(1-p)^{N-\sum x_i}.$$

The log likelihood of this joint distribution is,

$$\mathcal{L}(p) = \sum X_i log(p) + (n - \sum X_i) log(1-p).$$

The MLE of this is, $\hat{p}_{ML} = \bar{X}$, the sample average. Now let us go to the expected log likelihood,

$$E\left[\mathcal{L}(p)\right] = E\left[\sum X_i log(p) + (n - \sum X_i) log(1-p)\right].$$

This reduces to,

$$E\left[\mathcal{L}(p)\right] = n \cdot p^* \cdot log(p) + n \cdot (1-p^*) \cdot log(1-p).$$

This function is maximized at $\hat{p} = p^*$, the true value. Therefore if we take the expected log likelihood (just multiply by $1/n$), we will have a sample average. Sample averages are subject to the law of large numbers so we should expect that the sample average log likelihood is close to the true expected log likelihood.

1.1 MLE in practice: Simulating data generating processes

How do we employ MLE in practice? We can learn how by simulating a data generating process (DGP) and then estimating its parameters.[1] Let us begin with a simple linear model,

$$Y = X\beta + \epsilon,$$

[1] This is a nice way to error check code for complex estimation problems. If you know how the data were generated, but your estimator doesn't reflect the parameters of the DGP, then your code must have an error or you got very unlucky with random draws.

or in other notation,

$$Y \sim N(X\beta, \sigma^2),$$

where Y is Nx1, X is NxK and β is Kx1, and $\epsilon \sim N(0, \sigma^2)$. We can simulate independent variables using any distribution we want, but let us begin with the multivariate normal,

```
N = 100;
X = mvnrnd([0 0 0], eye(3), N);
X = [ones(N, 1) X];
```

This simply draws N samples from our multivariate normal distribution of 3 mean zero, unit variance, independent normal distributions. We also want a constant term, so we must concatenate that in manually. Given these $X's$, we need a vector of true parameters and draws of random shocks to generate our dependent variable, Y,

```
true_betas = [0.1, 0.5, -0.3, 0.]';
epsilons = normrnd(0,1,[100,1]);
Y = X*true_betas + epsilons;
```

You can test that this worked by checking the mean of Y. It should be close to 0.1 (intercept term) since our X's were distributed normal with mean zero.

Now we wish to estimate the parameters of our model, the βs. How do we do this? We know that $Y \sim N(X\beta, \sigma^2)$, but we can rearrange this expression so that,

$$Y - X\beta \sim N(0, \sigma^2),$$

and we have a distribution that does not depend on X. Given the independence across observations, we have that the likelihood is,

$$\mathcal{L}(\beta, \sigma^2) = \prod_{i=1}^{N} \phi(Y_i - X_i\beta, \sigma^2 | \beta, \sigma^2),$$

where ϕ denotes the pdf of the normal distribution. Let $\rho = [\beta, \sigma^2]$. We will be estimating some $\hat{\rho}$ which is the argmax of the likelihood. We now need to estimate these parameters. To do so, we define a negative loglikelihood function to maximize,

```
function nloglikelihood = loglike(Y,X,rho)
    beta = rho(1:4);
    sigma2 = exp(rho(5));
    residual = Y - X*beta;
    nloglikelihood = normlike([0, sigma2], residual);
end
```

This function takes data Y and X, and an input of parameters rho, and returns the negative log likelihood. Notice that we take the exponential of the parameter that governs the residual variance. This is to ensure that guesses are strictly positive, we will have to undo this transformation later. Now that we have this function, we can send it to one of our maximization routines to search over parameters that minimize the negative log likelihood conditional on the data.

```
initial_rhos = [1 1 1 1 1]';
log_handle = @(rho)loglike(Y,X,rho);
mle = fminunc(log_handle,initial_rhos);
mle(5) = exp(mle(5));
```

This will yield an estimate of the true parameters. As we increase our sample size N, our estimates should get closer and closer to the true values via the law of large numbers argument. We have estimated our parameters, but as economists we also care about how precise our estimates are: we need to get standard errors. We get standard errors for MLE by bootstrapping. First we must construct a bootstrap function,

```
function samples = bootstrap_mle(N,X,Y,initial_rhos)
    options = optimset('display','off');
    sample_index = datasample(1:N,N);
    X_boot = X(sample_index,:);
    Y_boot = Y(sample_index,:);
    samples = fminunc(@(rho)loglike(Y_boot,X_boot,rho),initial_rhos,options);
    samples(5) = exp(samples(5));
end
```

This function takes in two sets of data, samples the sets of data with replacement, and then uses the re-sampled data to re-maximize the log likelihood. It then returns the optimal parameter values, $\hat{\rho}^b$ for samples $b = 1, ..., B$.

```
num_samples = 1000;
samples = zeros(num_samples,5);
for b = 1:num_samples
    samples(b,:) = bootstrap_mle(N,X,Y,initial_rhos);
end
```

```
% Calculate standard errors
bootstrapSE = std(samples,1);
```

With our bootstrap function, we then loop over how many times we want to bootstrap and get many different parameter estimates using different draws from our sample. Our bootstrapped standard errors are just the standard deviations of the parameter estimates. Typically we don't just stop at standard errors, we often want to test the null hypothesis that our parameters are zero. If we can safely assume the MLE is normally distributed, like it is here, then we can reject the hypothesis that the parameter is zero if it is at least 1.96 standard errors away from zero.

We can also do this without assuming a distribution by constructing non-parametric p-values.

References

Rust, John (1987) "Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher," *Econometrica*, Vol. 55, No. 5, pp. 999–1033.