AME

Week 8: Maximum Likelihood Estimation

Sophie Bindslev, November 2022

UNIVERSITY OF COPENHAGEN

Today's Plan

- M-estimators
- Maximum Likelihood Estimators
- Variance Estimators
- Numerical Optimisers
- Your time to shine!

M-Estimators

- Last time we briefly considered that M-estimators may be viewed as solutions to optimisation problems
- The population problem:

$$\boldsymbol{\theta}_0 = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{argmin} E(q(\mathbf{w}, \boldsymbol{\theta})) \tag{1}$$

  where $q(\mathbf{w}, \boldsymbol{\theta})$ is a criterion function which depends on observables $\mathbf{w} = (\mathbf{x}, \mathbf{y})$ and parameters, $\boldsymbol{\theta}$
- The Sample analogue:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{argmin} \frac{1}{N} \sum_{i=1}^{N} q_i(\mathbf{w}, \boldsymbol{\theta}) \tag{2}$$

- Today we will look at maximum likelihood estimators, a sub-category of M-estimators

Maximum Likelihood Estimators I

- Maximum Likelihood estimates parameters of an assumed probability distribution given some observed data. Intuitively, we pick the parameters under which the data observed is most likely to have been generated by our assumed data generating process
- Say we believe our cross sectional data has IID normally distributed error terms, $u_i = y_i - \mathbf{x}_i\boldsymbol{\beta} \sim N(0, \sigma^2)$. Then, the distribution of y_i conditional on $\mathbf{x}_i$ is normal with variance σ^2
- The likelihood (probability density) of observing y_i conditional on $\mathbf{x}_i$ can then be written as

$$f(y_i \mid \mathbf{x}_i, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - \mathbf{x}_i\boldsymbol{\beta})^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{u_i^2}{2\sigma^2}} \qquad (3)$$

with parameters $\boldsymbol{\theta} = \boldsymbol{\beta}, \sigma$

Maximum Likelihood Estimators II

- MLE parameter estimates $\hat{\boldsymbol{\theta}}$ maximise the log-likelihood that the observed $\mathbf{y}$ were generated by the observed $\mathbf{x}$:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{argmin} - \frac{1}{N}\sum_{i=1}^{N} \ell_i(\mathbf{w}, \boldsymbol{\theta}) \tag{4}$$

- In the linear, normal case considered today the log-likelihood contribution of each observation pair $\mathbf{w}_i = (y_i, \mathbf{x}_i)$ is

$$\ell_i(\boldsymbol{\theta}) = -\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\sigma^2) - \frac{(y_i - \mathbf{x}_i\boldsymbol{\beta})^2}{2\sigma^2} \tag{5}$$

- We maximise the log-likelihood rather than the likelihood function to aid numerical precision (avoid computing the exponential, sums rather than products and, mainly, due to the risk of underflow)

Variance Estimation, IME

- Assuming $\mathbf{\Theta}$ is compact, $q(\mathbf{w}, .)$ is continuous and $\boldsymbol{\theta}_0$ solves the population problem (+ techn. assumptions) for consistency *and that* $\boldsymbol{\theta}_0$ is in the interior of $\mathbf{\Theta}$ and $q(\mathbf{w}, .)$ is twice continuously differentiable in $\boldsymbol{\theta}$, we have that our M-estimators are normally distributed with:

$$\sqrt{N}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0) \sim N(0, \mathbf{A}_0^{-1}\mathbf{B}_0\mathbf{A}_0^{-1}) \tag{6}$$

- To conduct inference we need an estimator of the "asymptotic"/approximate variance of our parameter estimates $\hat{\boldsymbol{\theta}}$
- Why *approximate* variance?
- In Maximum Likelihood the *"Information Matrix Equality"* (IME) implies that

$$\mathbf{B}_0 = \mathbf{A}_0 \tag{7}$$

Variance Estimators

- Under the IME we have a number of candidate variance estimators

  1) Hessian:

$$\widehat{Avar}(\hat{\boldsymbol{\theta}}) = \frac{1}{N}\hat{\mathbf{A}}^{-1} = \frac{1}{N}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{H}_i(\hat{\boldsymbol{\theta}})\right)^{-1} \tag{8}$$

  2) Outer product of the scores:

$$\widehat{Avar}(\hat{\boldsymbol{\theta}}) = \frac{1}{N}\hat{\mathbf{B}}^{-1} = \frac{1}{N}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{s}_i(\hat{\boldsymbol{\theta}})\mathbf{s}_i(\hat{\boldsymbol{\theta}})'\right)^{-1} \tag{9}$$

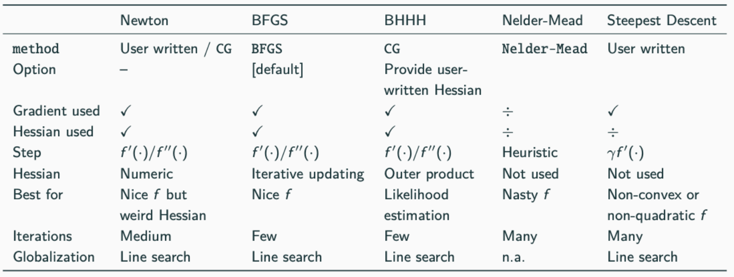
  3) Sandwich :

$$\widehat{Avar}(\hat{\boldsymbol{\theta}}) = \frac{1}{N}\hat{\mathbf{A}}^{-1}\hat{\mathbf{B}}\hat{\mathbf{A}}^{-1} \tag{10}$$

”Robust” Variance Estimators: A few notes of caution

- The sandwich variance estimator is sometimes referred to as more ”robust” since it is valid even if the model is misspecified
- However! If you believe your model is misspecified, then your parameter estimates are likely inconsistent
- Heteroskedastic errors, if unaccounted for, can be a consistency problem in MLE
- This is a cost of MLE: assuming we have the correct functional form of the entire conditional distribution of $y_i \mid \mathbf{x}_i$ gives us far richer results (not just the conditional mean, higher moments too) but also requires more assumptions to hold for consistency
- If you believe you know the form of heteroskedasticity you can (and should) account for it when deriving your likelihood function

Numerical Optimisers (again)



	Newton	BFGS	BHHH	Nelder-Mead	Steepest Descent
`method`	User written / CG	`BFGS`	CG	`Nelder-Mead`	User written
Option	–	[default]	Provide user-written Hessian		
Gradient used	✓	✓	✓	÷	✓
Hessian used	✓	✓	✓	÷	÷
Step	$f'(\cdot)/f''(\cdot)$	$f'(\cdot)/f''(\cdot)$	$f'(\cdot)/f''(\cdot)$	Heuristic	$\gamma f'(\cdot)$
Hessian	Numeric	Iterative updating	Outer product	Not used	Not used
Best for	Nice f but weird Hessian	Nice f	Likelihood estimation	Nasty f	Non-convex or non-quadratic f
Iterations	Medium	Few	Few	Many	Many
Globalization	Line search	Line search	Line search	n.a.	Line search

- You can implement different optimisers using `scipy.optimize.minimize` by specifying e.g. $\texttt{method} =' \texttt{BFGS}'$ or $\texttt{method} =' \texttt{Nelder} - \texttt{Mead}'$

Your time to shine!

- Fill in `estimation_ante.py` `LinearModel_ante.py` and solve the problem set
- Tip #1: Take a look at the documentation for `scipy.optimize.minimize` if you haven't already :)
- Tip #2: You can give your optimiser additional arguments using $**$`kwargs` in the `estimation_ante.py` file
- Tip #3: The minimiser returns an inverse Hessian which you can access by using `result.hess_inv`. **NB**: In our case the objective function is the mean of the negative log-likelihood contributions $-\frac{1}{N}\sum_{i=1}^{N}\ell_i$. Therefore, this inverse Hessian will already be the mean inverse Hessian i.e. $\hat{H}^{-1} = \frac{1}{N}\sum_{i=1}^{N}\hat{H}_i$ so there's no need to divide by N again when computing $\hat{A}$. There's no need to add a minus either (since using negative log-likelihood already)