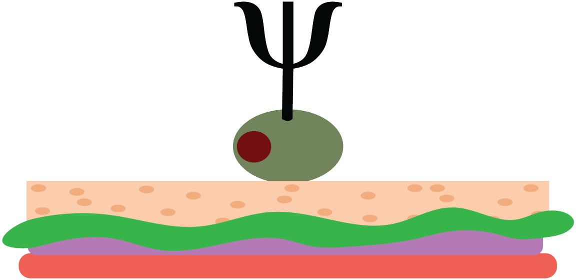
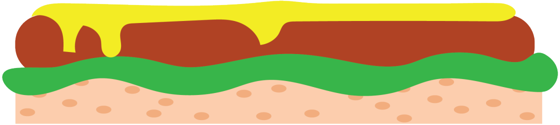


ABC's of M-estimation



Paul Zivich, Rachael Ross, Stephen Cole

University of North Carolina at Chapel Hill

Acknowledgements

Supported by NIH T32-AI007001 (PNZ), R01-DA056407 (RKR), R01-AI157758 (SRC), P30-AI50410 (SRC).

pzivich@unc.edu pzivich

Run Code

github.com/pzivich/ABCs_of_M-estimation

- Open your preferred statistical software
- Open corresponding `mean.*` script
- Run the full script
- Should see the following output

```
Closed-form:  8.0
Root-finder:  8.0
95% CI: [ 0.8, 15.2]
```

Overview

Practical Applications of M-estimation

Three use-cases

- Marginal structural model with inverse probability weights
- Bridged treatment comparisons
- Higher-order evidence

While other estimators can be used, M-estimators have advantages

Marginal Structural Models

Interested in the marginal structural model (MSM):[1]

$$E[Y^a] = \alpha_0 + \alpha_1 a$$

Estimate with $E[Y] = \hat{\alpha}_0 + \hat{\alpha}_1 A_i$ with weights $\frac{1}{\widehat{\Pr}(A=a|W)}$

Challenge: variance of $\hat{\alpha}$ depends on variance of $\widehat{\Pr}(A=a|W)$

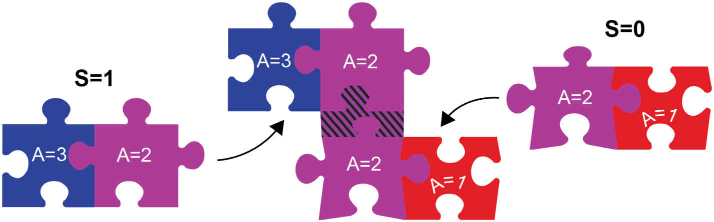
- Commonly use the 'GEE' trick
- Bootstrap is computationally intensive

M-estimation

- Not conservative & computationally simpler than bootstrap
- Application[2]

[1] Robins et al. (2000) *Epidemiology*
[2] Reifeis & Hudgens (2022) *American Journal of Epidemiology*

Bridged Treatment Comparisons



Challenge: multiple sets of weights

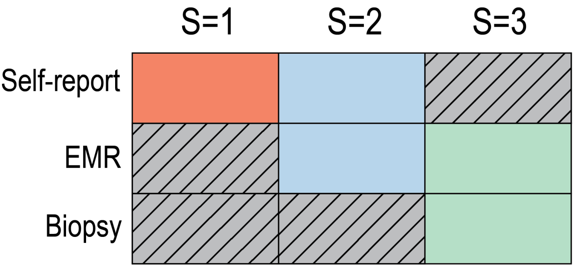
- Weights for treatment, missing outcomes, transportability

M-estimation

- Computationally efficient variance estimator
- Application[3]

[3]Shook-Sa et al. *arXiv:2305.00845*

Higher-Order Evidence



Challenge: multiple sensitivity (Se) & specificity (Sp)

- Variance depends on variance of Se & Sp

M-estimation

- Solve for all parameters simultaneously
- Application[4]

[4] Cole et al. (2023) *Under Review*

Overview

Section 1: introduction

Break (15min)

Section 2: applied examples

Break (15min)

Section 3: extensions, cautions, conclusions

Overview

Section 1: introduction

Break (15min)

Section 2: applied examples

Break (15min)

Section 3: extensions, cautions, conclusions

Overview: Section 1

Review notation / definitions

M-estimator by-hand

M-estimator with computer

Some statistical properties

Notation

Review notation and mathematical operations used

- If unfamiliar with something, don't worry!
- Operations will be contextualized in following sections
- Operations will also be done by the computer
- Definitions can be returned to later

Notation – Basics

O_i: observed data for unit i

- $O_i = (X_i, Y_i)$
- $O_i = (W_i, A_i, Y_i)$

$\sum_{i=1}^{n} i \ = \ 1 + 2 + ... + n$: cumulative sum

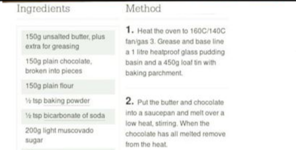
$\prod_{i=1}^{n} i \ = \ 1 \times 2 \times ... \times n$: cumulative product

expit$(a) = 1/(1 + \exp(-a))$

Notation – Basics

estimand θ
(parameter of interest)

estimator $\hat{\theta}$


Ingredients

150g unsalted butter, plus extra for greasing
150g plain chocolate, broken into pieces
150g plain flour
½ tsp baking powder
½ tsp bicarbonate of soda
200g light muscovado sugar

Method

1. Heat the oven to 160C/140C fan/gas 3. Grease and base line a 1 litre heatproof glass pudding basin and a 450g loaf tin with baking parchment.

2. Put the butter and chocolate into a saucepan and melt over a low heat, stirring. When the chocolate has all melted remove from the heat.

estimate 0.5


[5]

[5]Estimand also denoted by θ_0 or θ^*

Notation – Matrix Algebra

Transpose

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \\ e & f \end{bmatrix} \quad \mathbf{A}^T = \begin{bmatrix} a & c & e \\ b & d & f \end{bmatrix}$$

Notation – Matrix Algebra

Dot product (multiplication)

$$\mathbf{B}\ \mathbf{C} = \begin{bmatrix} a & b \end{bmatrix} \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} a\ w + b\ y & a\ x + b\ z \end{bmatrix}$$

- Elements in row of 1st must match elements in column of 2nd
  - $\mathbf{C}\ \mathbf{B}$ would not be defined
- Output has rows of 1st and columns of 2nd
- Symmetric matrices have same shape

Notation – Matrix Algebra

Inverse of matrix

$$\mathbf{D} = \begin{bmatrix} w & x \\ y & z \end{bmatrix} \quad \mathbf{D}^{-1} = \frac{1}{w \ z - x \ y} \begin{bmatrix} z & -y \\ -x & w \end{bmatrix}$$

- Only applies to matrices with same number of rows and columns

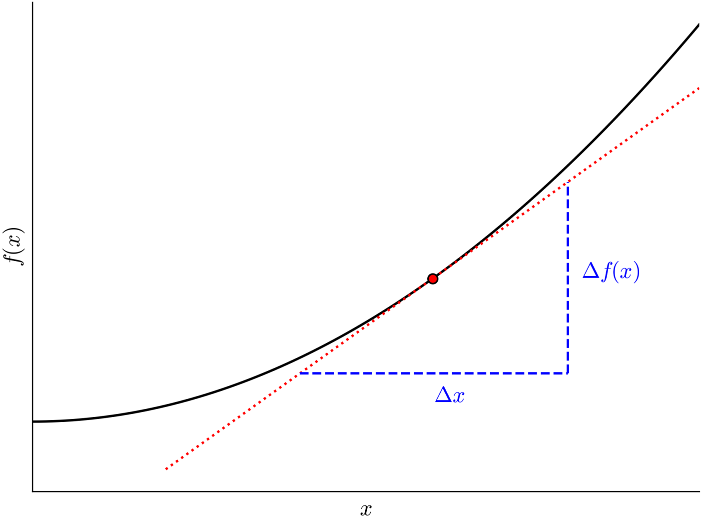
Derivatives – Basics

$$f'(x) = \frac{d}{dx} f(x)$$

Helpful to think of derivative as slope of tangent line at a point

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Derivatives – Basics



Derivatives – Generalizations

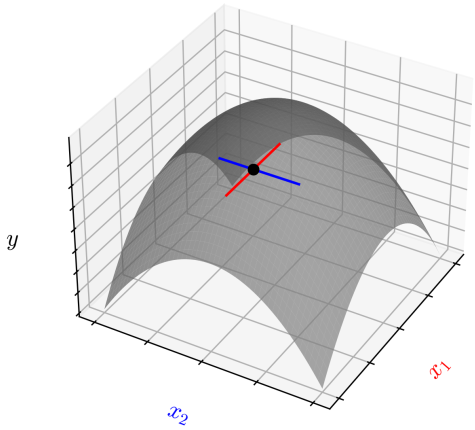
If $\boldsymbol{x} = (x_1, x_2, ..., x_m)$ and $f(\boldsymbol{x}) = y$, then the partial derivative is

$$\frac{\partial}{\partial x_1} f(\boldsymbol{x})$$

The gradient is

$$\nabla f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(\boldsymbol{x}) \\ \frac{\partial}{\partial x_2} f(\boldsymbol{x}) \\ \vdots \\ \frac{\partial}{\partial x_m} f(\boldsymbol{x}) \end{bmatrix}$$

Derivatives – Generalizations



Derivatives – Generalizations

The Hessian is

$$\Delta H_f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2}{\partial x_1 \partial x_1} f(\boldsymbol{x}) & \dots & \frac{\partial^2}{\partial x_1 \partial x_m} f(\boldsymbol{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_m \partial x_1} f(\boldsymbol{x}) & \dots & \frac{\partial^2}{\partial x_m \partial x_m} f(\boldsymbol{x}) \end{bmatrix}$$

- Jacobian (transpose gradient, ∇^T) of the gradient

Derivatives – Generalization

Function

$$f(x_1, x_2) = y$$

Gradient

$$\nabla f(x_1, x_2) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x_1) \\ \frac{\partial}{\partial x_2} f(x_2) \end{bmatrix}$$

Hessian

$$\Delta H_f(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial^2}{\partial x_1 \partial x_1} f(x_1, x_2) & \frac{\partial^2}{\partial x_1 \partial x_2} f(x_1, x_2) \\ \frac{\partial^2}{\partial x_2 \partial x_1} f(x_1, x_2) & \frac{\partial^2}{\partial x_2 \partial x_2} f(x_1, x_2) \end{bmatrix}$$

Notation – M-estimation

Estimating *function*

$$\psi(O_i; \theta)$$

Estimating *equation*

$$\sum_{i=1}^{n} \psi(O_i; \theta)$$

Definition: M-estimator

An M-estimator, $\hat{\theta}$, is the solution to

k-dimensional estimating function

k-dimensional parameter

$$\sum_{i=1}^{n} \boxed{\psi\left(O_i , \hat{\theta} \right)} = 0$$

Observation i

root: where $f(x) = 0$

- Don't worry if any of the above isn't clear yet

M-estimator for the mean

Problem: Learn the Mean

Want to learn the population mean

- Estimand: $\mu = E[Y]$

Suppose we have the following observations to estimate μ

$$7, 1, 5, 3, 24$$

Usual method

$$\underbrace{\hat{\mu}}_{\text{Estimator for mean}} = \frac{1}{\underbrace{n}_{\text{Total number of units}}} \sum_{i=1}^{n} \underbrace{Y_i}_{\text{Observed value for unit } i}$$

Applying to data in example (estimate)

$$\frac{7+1+5+3+24}{5} = \frac{40}{5} = 8$$

but let's use M-estimation instead

M-estimator steps

1. Determine estimating function
2. Find the roots of the estimating equations
3. Estimate variance via the sandwich

1. Determine Estimating Function

Goal: rewrite mean as a function that is equal to zero

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} Y_i \qquad \text{definition}$$

$$\hat{\mu}\; n = \sum_{i=1}^{n} Y_i \qquad \text{multiply by } n$$

$$0 = \sum_{i=1}^{n} (Y_i) - \hat{\mu}\; n \qquad \text{subtract } \hat{\mu} n$$

$$0 = \sum_{i=1}^{n} (Y_i) - \sum_{i=1}^{n} (\hat{\mu})$$

$$0 = \sum_{i=1}^{n} (Y_i - \hat{\mu})$$

1. Determine Estimating Function

This formula is our M-estimator for the mean

$$\sum_{i=1}^{n} \overbrace{\psi}^{\text{Estimating function}} (\underbrace{O_i}_{\text{Observation } i} , \overbrace{\hat{\theta}}^{\text{Parameter}}) = \sum_{i=1}^{n} (\underbrace{Y_i}_{\text{Observation } i} - \overbrace{\hat{\mu}}^{\text{Parameter}}) = 0$$

2. Root-finding

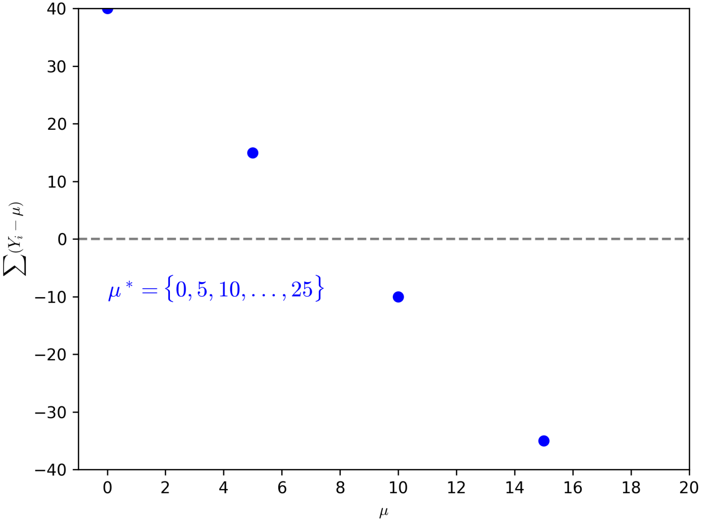
How can we find $\hat{\mu}$?

- Ignore the closed-form solution for the time

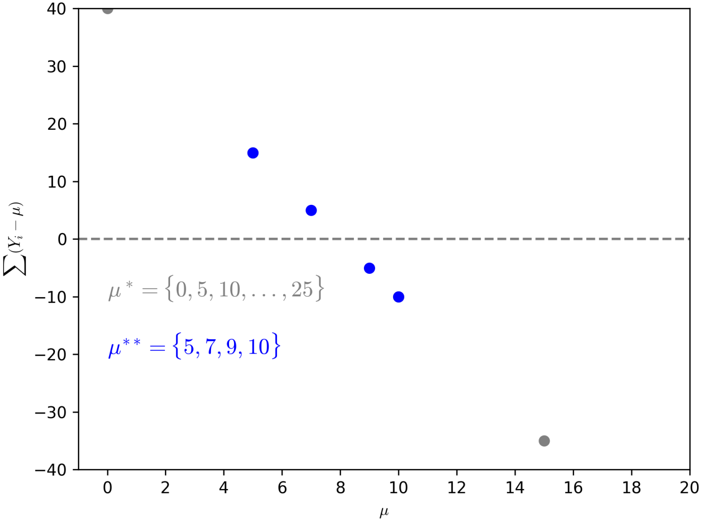
Broadly

- Take some guesses at $\hat{\mu}$, denoted as $\hat{\mu}^*$
- Compute $\sum_{i=1}^{n} \psi(O_i; \hat{\mu}^*)$
- Find the guesses that are close to zero
- Generate some new guesses, $\hat{\mu}^{**}$
- Repeat process until we find $\hat{\mu}$

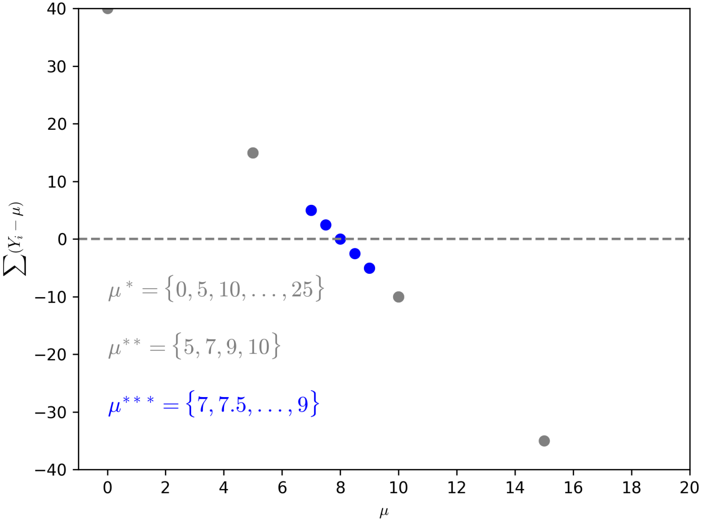
2. Root-finding



2. Root-finding



2. Root-finding



3. Variance

Closed-form estimator[6]

$$\widehat{Var}(\hat{\mu}) = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{\mu})^2$$

but let's use M-estimation instead

[6]Note: n is often replaced by $n-1$ in practice, which can lead to differences for small sample sizes

3. Sandwich Variance Estimator

Sandwich variance

Filling (meat) matrix

$$\underbrace{V(\hat{\theta})} = \underbrace{B(\hat{\theta})^{-1}} \; \underbrace{F(\hat{\theta})} \left(\underbrace{B(\hat{\theta})^{-1}} \right)^T$$

(inverse of) Bread matrix

3. Sandwich Variance Estimator

Bread matrix

$$B(\hat{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \left[-\psi'(O_i, \hat{\theta}) \right]$$

Partial derivatives (Jacobian)

Filling matrix

$$F(\hat{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \left[\psi(O_i, \hat{\theta}) \; \psi(O_i, \hat{\theta})^T \right]$$

Dot product of estimating functions

Baking the Bread: By-Hand

Need the derivative of $\psi(O_i;\mu)$

$$\begin{aligned}\psi'(O_i;\hat{\mu}) =& \frac{d}{d\hat{\mu}}\psi(O_i;\hat{\mu}) && \text{definition of derivative}\\ =& \frac{d}{d\hat{\mu}}(Y_i - \hat{\mu}) && \text{definition of estimating function}\\ =& -1\end{aligned}$$

Therefore

$$\boxed{\frac{1}{n}\sum_{i=1}^{n}\left[-\psi'(O_i,\hat{\theta})\right]} = \frac{1}{n}\sum_{i=1}^{n}\left[-\boxed{-1}\right] = 1$$

Definition of Bread

From derivative above

Cooking the Filling: By-Hand

Definition of Filling

$$\boxed{\frac{1}{n}\sum_{i=1}^{n}\left[\psi(O_i,\hat{\theta})\psi(O_i,\hat{\theta})^T\right]} = \frac{1}{n}\sum_{i=1}^{n}\left[\boxed{(Y_i-\hat{\mu})(Y_i-\hat{\mu})}\right]$$

Plugging in estimating function

Therefore

$$\frac{1}{5}\sum_{i=1}^{5}\left[(Y_i-8)^2\right] = 68$$

Assembling the Sandwich: By-Hand

$$\underbrace{\overbrace{V(\hat{\mu})}^{\text{Sandwich variance}}}_{} = \underbrace{1^{-1} \overbrace{68}^{\text{Filling}} \left(1^{-1}\right)^{T}}_{\text{Bread}} = 68$$

Confidence intervals

$$\hat{\mu} \pm z_{\alpha} \sqrt{\frac{V(\hat{\mu})}{n}} = 8 \pm 1.96 \sqrt{\frac{68}{5}} = (0.8, 15.2)$$

Computation for M-estimators

Computation for M-estimators

Solved for M-estimator of mean by-hand

- By-hand is not needed

Instead, consider how M-estimators can be implemented

- Root-finding
- Approximation of derivatives
- Matrix algebra

Follow along in `mean.R`, `mean.sas`, or `mean.py`

- Start of code inputs data and sets up estimating equations

Root-Finding – Algorithms

Performed a by-hand search for $\hat{\mu}$

- Similar to the *bisection method*

Variety of multidimensional root-finding algorithms exist

- Secant method (quasi-Newton)
- Levenberg-Marquardt
- Powell hybrid method

Root-Finding – Code

Under `Root-finding` see implementation

- SAS – `nlplm`
- R – `rootSolve::multiroot`
- Python – `scipy.optimize.root`

Derivatives – Back to the Definition

Derivative of function

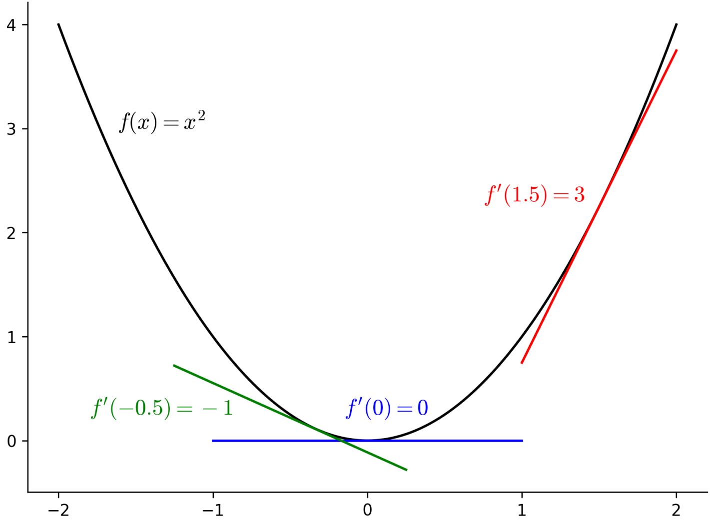
Change in output (rise)

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Behavior as h becomes small

Divided change in input (run)

Derivatives – Intuition



Derivatives – Numerical Approximation

Central Difference Method[7]

$$\underbrace{\tilde{f}'(x)}_{\text{Approximation}} = \frac{f(\underbrace{x+a}_{\text{Slightly above } x}) - f(\underbrace{x-a}_{\text{Slightly below } x})}{2\,a}$$

Here a is a small value (e.g., 1×10^{-9})

[7] Automatic differentiation, which computes the derivatives exactly via the chain rule, could be used instead

Baking the Bread – Code

Under `Baking the bread` see implementation

- SAS – `nlpfdd`
- R – `numDeriv::jacobian`
- Python – `scipy.optimize.approx_fprime`

Cooking the Filling – Code

Under `Cooking the filling` see implementation

- Transpose
  - SAS – `'`
  - R – `base::t`
  - Python – `numpy.transpose`
- Dot product
  - SAS – `*`
  - R – `%*%`
  - Python – `numpy.dot`

Assembling the Sandwich – Code

Under `Assembling the sandwich` see implementation

- Inverse
  - SAS – `inv`
  - R – `base::solve`
  - Python – `numpy.linalg.inv`

Implications

To implement an M-estimator, we only need to provide

- Valid estimating functions
- Data

Everything else can be done by the computer

- Potential to simplify complex analyses
- Open-source libraries
  - R: `geex`[8]
  - Python: `delicatessen`[9]

[8] Saul & Hudgens (2020) *Journal of Statistical Software*
[9] Zivich et al. (2022) *arXiv:2203.11300*

Extensions

But Why M-estimation?

So far, all we've done is calculate the mean in a complicated way

So why bother with M-estimation?

- Flexibility of the framework
    - Extensions of these basics
    - Simplified proofs for properties of estimators

How M-estimators are extended

As will be seen in applied examples

1. Stacking estimating functions
2. Automation of delta method

Stacking estimating functions

Often want to estimate more than 1 parameter

- Regression models
- Effect measure modification
- Inverse probability weighting requires estimating propensity scores

Stacking Estimating Functions

M-estimators extend by stacking estimating functions

$$\sum_{i=1}^{n} \begin{bmatrix} \psi_{\theta_1}(O_i; \hat{\theta}) \\ \psi_{\theta_2}(O_i; \hat{\theta}) \\ \psi_{\theta_3}(O_i; \hat{\theta}) \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} = \mathbf{0}$$

- Easy to stack estimating functions together
- Unlike maximizing a likelihood
  - Likelihood has a single value for individual contribution
  - Need each parameter to contribute correctly
  - More difficult to combine likelihood functions

Stacking Estimating Functions

Example

$$\sum_{i=1}^{n} \begin{bmatrix} \psi_{\theta_1}(O_i;\theta) \\ \psi_{\theta_2}(O_i;\theta) \end{bmatrix} = \sum_{i=1}^{n} \begin{bmatrix} Y_i - \theta_1 \\ (Y_i - \theta_1)^2 - \theta_2 \end{bmatrix} = \mathbf{0}$$

- Stacking important when parameter depends on other parameters
- Concept explored further in applications

Delta Method

Theorem: smooth function of AN estimator is also AN[10]

Application:

$$\underbrace{Var\left\{ \boxed{g(\alpha)} \right\}}_{} \approx g'(\alpha)\ \Sigma_\alpha\ g'(\alpha)$$

Transformation of α

Covariance of α

Derivative of transformation

[10] AN: asymptotically normal

Delta Method

Many variance formulas you know are Delta method results

- $Var(RD)$, $Var(\log(RR))$, $Var(\log(OR))$
- Formulas follow from Delta method argument
- Don't need to manually solve due to known formulas
  - Not always the case

Delta Method with M-estimation

The estimating function for the transformed parameter, θ_t is

$$\psi_{g(\theta)}(O_i; \theta, \theta_t) = g(\theta) - \theta_t$$

- Estimating function does not depend on data

Therefore, the stacked estimating equations are

$$\sum_{i=1}^{n} \begin{bmatrix} \psi^*(O_i; \theta) \\ \psi_{g(\theta)}(O_i; \theta, \theta_t) \end{bmatrix} = 0$$

Delta Method with M-estimation

Following some derivatives and matrix algebra

$$V(\theta, \theta_t) = \begin{bmatrix} V^*(\theta) & g'(\theta)V^*(\theta) \\ V^*(\theta)g'(\theta)^T & g'(\theta)V^*(\theta)g'(\theta) \end{bmatrix}$$

where

Sandwich covariance for θ

$$V(\theta_t) = \underbrace{g'(\theta)}\ \overbrace{V^*(\theta)}\ \underbrace{g'(\theta)}$$

Derivative of transformation

- which is the same result from the delta method!

M-estimators automate the Delta method

Robust Variance

To close this section, let's discuss the robust variance

- The sandwich variance is also known as the 'robust' variance
- 'Robust' designates that the variance estimator is robust to certain assumptions[11]
    - Variance estimator is consistent when parametric model is wrong
    - However this has some difficulties
- Relates back to Maximum Likelihood Estimation
    - The variance can be estimated two ways

[11] See Mansournia et al. (2021) *International Journal of Epidemiology* for further details

Robust Variance

Variance estimators

1. Inverse Hessian of the log-likelihood
   - Equivalent to $B(\theta)^{-1}$

2. Residuals of the score function
   - Equivalent to $F(\theta)^{-1}$

- When the model is correctly specified
  - These variance estimators asymptotically equivalent
  - $B(\theta) = F(\theta)$

Robust Variance

When the model is not correctly specified

- $B(\theta) \neq F(\theta)$
- By combining, sandwich is robust to assumptions
  - Variance estimator is consistent even if model is wrong

- Example: log-Poisson model to estimate the risk ratio
  - Here, estimated variance is too large

Warning[12]

- Does not correct for bias in parameter estimates

[12] See Freedman DA *Am Stat* 2006 for details

Overview

Section 1: introduction

Break (15min)

Section 2: applied examples

Break (15min)

Section 3: extensions, cautions, conclusions