

ABC's of M-estimation



Paul Zivich, Rachael Ross, Stephen Cole

University of North Carolina at Chapel Hill

Thank you for attending

✉ pzivich@unc.edu

🔗 github.com/pzivich/ABCs_of_M-estimation

Feedback on workshop: forms.gle/fAK6nQFj8zRknFQj7

Overview

Section 1: introduction

Section 2: applied examples

Three use-cases of M-estimators

- Estimate the variance of a marginal structural model
- Bridged treatment comparisons
- Sensitivity analyses

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

Variance with IPW

IPW for estimating marginal structural model (MSM)¹

- Model propensity score
- Calculate weighted mean

Variance estimation is complicated

- Variance for MSM depends on variance of propensity scores
- Commonly use the 'GEE' trick
- Bootstrap is computationally intensive

M-estimation

- Not overly conservative
- Computationally simpler than bootstrap

¹Robins et al. (2000) *Epidemiology*

Bridged Treatment Comparisons

Suppose we want to learn the causal effect of **A** vs **C**²

- Only have data that compares **A** vs **B** and **B** vs **C**

Bridged treatment comparisons link across data source

- Compare **A** to **C** through **B**
- Analytically account for differences between source
- Further correct for confounding or missing outcomes

M-estimation

- Consistently estimate the variance with models
- Holds for not identically distributed data

²Zivich et al. *arXiv:2206.04445*

Sensitivity Analysis

Missing data is a common problem

- Descriptive, predictive, causal

Missingness may depend on the missing variable

- Whether Y is observed depends on Y
- Missing not at random
- Sensitivity analyses to assess the impact

M-estimation

- Avoid having to use the bootstrap
- Evaluate a large number of scenarios

Section 1: introduction

Section 2: applied examples

Overview: Section 1

Review notation / definitions

M-estimator by-hand

M-estimator computationally

Some useful properties of M-estimators

O_i : observed data for unit i

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

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

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

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

Notation – Basics

estimand
(parameter of interest)

 θ 

estimator

 $\hat{\theta}$

Ingredients

150g unsalted butter, plus extra for greasing
150g plain chocolate, broken into pieces
150g plain flour
1/2 tsp baking powder
1/2 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 

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

Inverse of matrix

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

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

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

Helpful to think of derivative as slope at a point

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

If $\mathbf{x} = (x_1, x_2, \dots, x_m)$, then the partial derivative is

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

The gradient is

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

The Hessian is

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

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

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

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

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

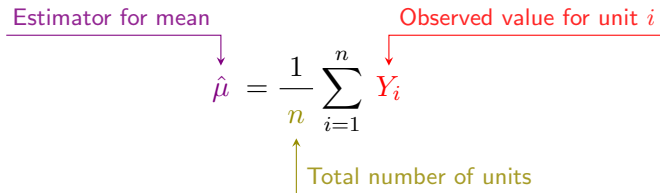
Want to learn the population mean

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

Suppose we have the following observations to estimate μ

7, 1, 5, 3, 24

Usual method



The diagram illustrates the formula for the sample mean estimator. A purple line labeled "Estimator for mean" points to the symbol $\hat{\mu}$. A red line labeled "Observed value for unit i " points to the variable Y_i in the summation. A yellow line labeled "Total number of units" points to the variable n in the denominator. The formula is
$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n Y_i$$

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

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

definition

$$\hat{\mu} n = \sum_{i=1}^n Y_i$$

multiply by n

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

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

The diagram shows the formula for an M-estimator for the mean, with annotations identifying its components. The formula is presented in two equivalent forms. The first form is $\sum_{i=1}^n \psi(O_i, \hat{\theta}) = 0$, where $\psi(O_i, \hat{\theta})$ is enclosed in a light blue box. A blue arrow labeled "Estimating function" points to this box. A red arrow labeled "Observation i " points to the O_i term. A purple arrow labeled "Parameter" points to the $\hat{\theta}$ term. The second form is $\sum_{i=1}^n (Y_i - \hat{\mu}) = 0$, where $(Y_i - \hat{\mu})$ is enclosed in a light blue box. A red arrow labeled "Observation i " points to the Y_i term. A purple arrow labeled "Parameter" points to the $\hat{\mu}$ term.

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

Alternative Derivation

Mean is value where

$$\min \sum_{i=1}^n \rho(O_i; \hat{\mu}) = \min \sum_{i=1}^n (Y_i - \hat{\mu})^2$$

At the minimum (or maximum), the slope is zero

$$\sum_{i=1}^n \rho'(O_i; \hat{\mu}) = 0$$

Therefore

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

2. Root-finding

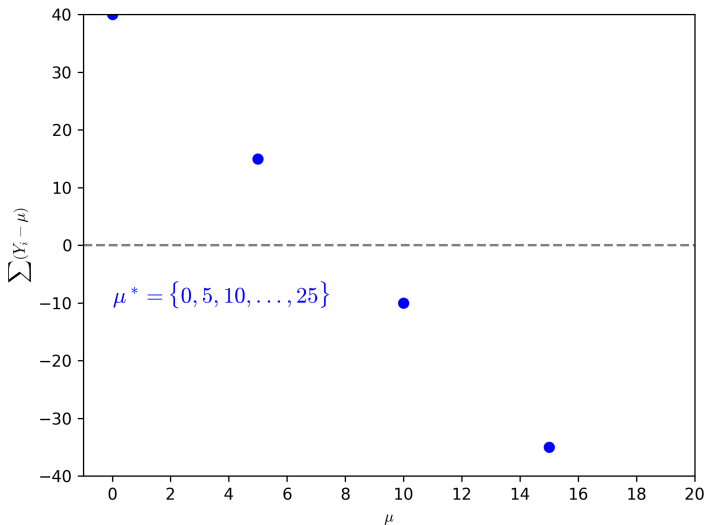
How can we find $\hat{\mu}$?

- Ignore the closed-form solution for the time

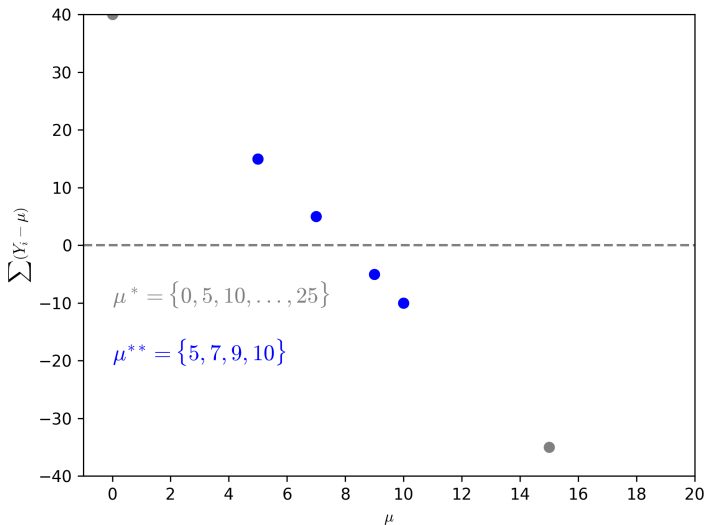
Broadly

- 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 $\hat{\mu}$

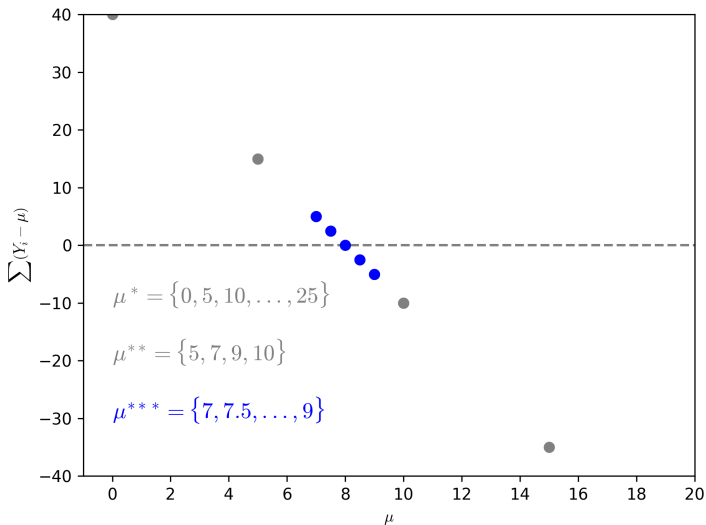
2. Root-finding



2. Root-finding



2. Root-finding



3. Variance

Closed-form estimator

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

but let's use M-estimation instead

3. Sandwich Variance Estimator

The diagram illustrates the Sandwich Variance Estimator formula: $V(\hat{\theta}) = B(\hat{\theta})^{-1} F(\hat{\theta}) (B(\hat{\theta})^{-1})^T$. The components are color-coded and labeled as follows:

- Sandwich variance:** A purple label with an arrow pointing to the $V(\hat{\theta})$ term, which is enclosed in a purple box.
- Filling (meat) matrix:** A red label with an arrow pointing to the $F(\hat{\theta})$ term, which is enclosed in a red box.
- (inverse of) Bread matrix:** A blue label with two arrows pointing to the $B(\hat{\theta})^{-1}$ terms, which are enclosed in blue boxes.

The formula is presented as: $V(\hat{\theta}) = B(\hat{\theta})^{-1} F(\hat{\theta}) (B(\hat{\theta})^{-1})^T$

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

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

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

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

Definition of Bread 

From derivative above 

Cooking the Filling: By-Hand

Definition of Filling

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

Plugging in estimating function

Therefore

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

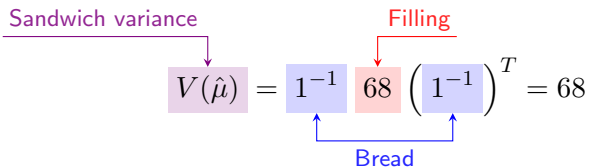
Assembling the Sandwich: By-Hand

Sandwich variance

Filling

$$V(\hat{\mu}) = 1^{-1} 68 \left(1^{-1}\right)^T = 68$$

Bread



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

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

Under **Root-finding** see implementation

- SAS – `nlp1m`
- R – `rootSolve::multiroot`
- Python – `scipy.optimize.root`

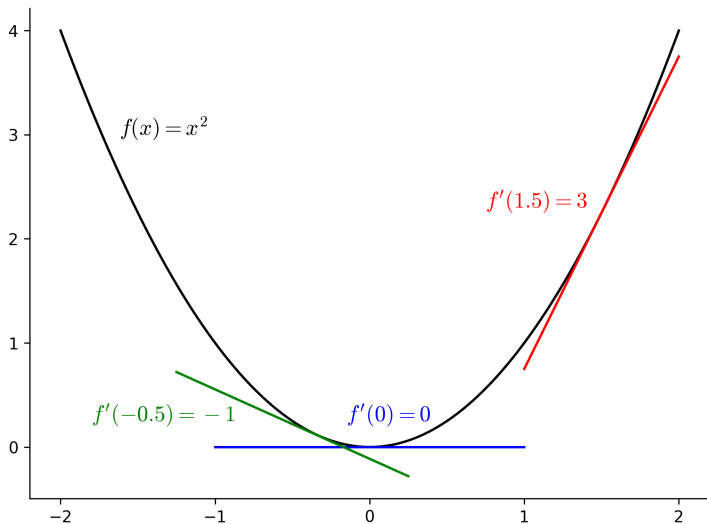
Derivatives – Back to the Definition

The diagram illustrates the definition of a derivative with the following components and annotations:

- Derivative of function**: A black line points from this text to the $f'(x)$ term in the equation.
- Behavior as h becomes small**: A blue line points from this text to the $\lim_{h \rightarrow 0}$ term.
- Change in output (rise)**: A red line points from this text to the numerator $f(x+h) - f(x)$.
- Divided change in input (run)**: A purple line points from this text to the denominator h .

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

Derivatives – Intuition



Derivatives – Numerical Approximation

Central Difference Method

Approximation

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

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

Under **Baking the bread** see implementation

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

Under **Cooking the filling** see implementation

- Transpose
 - SAS – ‘
 - R – `base::t`
 - Python – `numpy.transpose`
- Dot product
 - SAS – *
 - R – `%*%`
 - Python – `numpy.dot`

Under **Assembling the sandwich** see implementation

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

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

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

$$\sum_{i=1}^n \begin{bmatrix} \psi_1(O_i; \theta) \\ \psi_2(O_i; \theta) \\ \psi_3(O_i; \theta) \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix} = \mathbf{0}$$

- 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

Example

$$\sum_{i=1}^n \begin{bmatrix} \psi_1(O_i; \theta) \\ \psi_2(O_i; \theta) \end{bmatrix} = \begin{bmatrix} Y_i - \theta_1 \\ X_i - \theta_2 \end{bmatrix} = \mathbf{0}$$

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

Theorem: smooth function of AN estimator is also AN



Application:

$$\text{Transformation of } \alpha \rightarrow \text{Var} \left\{ g(\alpha) \right\} \approx \begin{matrix} \text{Covariance of } \alpha \\ \downarrow \\ g'(\alpha) \quad \Sigma_{\alpha} \quad g'(\alpha) \\ \uparrow \qquad \qquad \uparrow \\ \text{Derivative of transformation} \end{matrix}$$

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

The estimating function for the transformed parameter, θ_t is

$$\psi_t(O_i; \theta, \theta_t) = g(\theta) - \theta_t$$

- Estimating function does not depend on data

Therefore, the stacked estimating equations are

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

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

$$V(\theta_t) = \begin{array}{|c|c|c|} \hline g'(\theta) & V^*(\theta) & g'(\theta) \\ \hline \end{array}$$

Derivative of transformation

Sandwich covariance for θ

- which is the same result from the delta method!

M-estimators automate the Delta method

Statistical Properties of M-estimators

Key Statistical Properties

Consistency (C)

- Estimator converges to the estimand as $n \rightarrow \infty$ in probability
- As your data increases, the estimate goes to the true value

Asymptotic Normality (AN)

- As $n \rightarrow \infty$ the distribution of $\hat{\theta}$ converges to a normal distribution

Desirable properties for estimators

- C: inconsistent estimators may not give 'right' answer
 - Even with massive amounts of data
- AN: Wald-type confidence intervals are justified

CAN Proofs with M-estimation

If you can prove that the estimating equations are unbiased³

- Following some regularity conditions
- Sufficient to prove M-estimator is CAN



³Section 7.8 (pg 327) of Boos & Stefanski's *Essential Statistical Inference*

The sandwich variance is also known as the 'robust' variance

Maximum likelihood estimation

- 1 Inverse Hessian of the log-likelihood
 - Equivalent to the bread, $B(\theta)$
- 2 Residuals of the score function
 - Equivalent to the filling, $F(\theta)$
- These variance estimators asymptotically equivalent
 - When the model is correctly specified
 - $B(\theta) = F(\theta)$
 - Hessian is generally most efficient in finite samples

When the model is not correctly specified

- $B(\theta) \neq F(\theta)$
- Example: log-Poisson model to estimate the risk ratio
 - Here, estimated variance is too large
- Here, sandwich variance works
 - By combining, sandwich is robust to assumptions
 - Variance estimator is consistent even if model is wrong

Caution:⁴

- Does not correct for bias in parameter estimates
- Okay to use for log-Poisson because unbiased for RR
- Otherwise inference is no longer for original θ

⁴See Freedman DA *Am Stat* 2006 for details

Applied Examples