# Causal inference using the g-formula in Stan

Leah Comment

Department of Biostatistics
Harvard T.H. Chan School of Public Health

January 12, 2018

# Crash course on causal inference

- Goal: learn about causal mechanisms using observational data
- Why?
  - Useful for identifying targets for policy intervention
  - Can create projections for what *would* occur after some policy change
  - Need to make decisions even when conclusive data are not available
- Caveats:
  - Correlation still $\neq$ causation; more about formalizing *what would be necessary* for that to hold
  - Not going to be very rigorous today

# The potential outcomes framework

$$
\begin{tikzcd}
\mathbf{Z} \arrow[dr] \arrow[drr, bend left] & & \\
& A \arrow[r] & Y
\end{tikzcd}
$$

- Some treatment or exposure $A$
- Outcome of interest is $Y$
- Under some assumptions, the **potential outcome** $Y_a$ is the value $Y$ would take on if $A$ were set to $a$
- For binary $A$:
  - Average treatment effect: $\mathbb{E}\left[Y_1 - Y_0\right]$
  - Average treatment effect on treated: $\mathbb{E}\left[Y_1 - Y_0 | A = 1\right]$
- Often need to adjust for a set of baseline confounders $\mathbf{Z}$

# The g-formula for standardization

**g-formula:** $\mathbb{E}[Y_a] = \sum_z \mathbb{E}[Y|A=a, Z=z]\, P(Z=z)$

- This requires no unmeasured confounding given $Z$: $Y_a \perp\!\!\!\perp A|Z$
- Average treatment effect of changing $A$ from $a$ to $a^*$ for whole population: $\mathbb{E}[Y_{a^*}] - \mathbb{E}[Y_a]$
- Common (frequentist) approach is to adopt parameteric models for $Y|A, Z$ and use empirical distribution of $Z$ for $P(Z=z)$
- Frequentist bootstrap used for inference

# A Bayesian version of the g-formula

Adopting parametric models indexed by $\boldsymbol{\theta}$, the Bayesian g-formula is:

$$p(y_a|o) = \int\int p(\tilde{y}|a, \tilde{\mathbf{z}}, \theta)p(\tilde{\mathbf{z}}|\boldsymbol{\theta})p(\boldsymbol{\theta}|o)d\boldsymbol{\theta}d\tilde{\mathbf{z}}$$

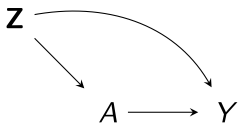
- $p(\tilde{y}_a|o)$
  - Distribution of $Y$ we would expect to see if $A$ were set to $a$ in some population with same:
    - Underlying confounder distribution (comparability)
    - Data-generating parameters (causal transportability)
- This integrates over uncertainty in $\boldsymbol{\theta}$
- Causal estimands usually compare means of $p(\tilde{y}_1|o)$ and $p(\tilde{y}_0|o)$
- See paper by Keil et al for more details (Keil et al. 2015)

# Causal inference with Stan

Two components to Bayesian causal inference with the g-formula:

- Get posterior samples of parameters $\boldsymbol{\theta}$
  - Learn from data in `data` block
  - Fit parametric models in `model` block
- Do causal inference using posterior predictive draws of potential outcomes
  - Use confounder distribution from `data` block (may or may not be same data)
  - Sample potential outcomes in the `generated quantities` block

# A simple example




- Nothing in particular assumed about distribution of $\mathbf{Z}$
- Binary $A$
- Binary $Y$

# Simple example: model

Assume $Y$ is generated according to logistic model:

$$\operatorname{logit}\left(P(Y_i = 1|A_i, Z_i)\right) = \alpha_0 + \alpha_A A_i + \boldsymbol{\alpha}_Z' \mathbf{Z}_i$$

# Simple example: code

https://github.com/lcomm/stancon2018/simple_mc.stan

```
data {
  // number of observations
  int<lower=0> N;
  // number of columns in design matrix excluding A
  int<lower=0> P;
  // design matrix, excluding treatment A
  matrix[N, P] X;
  // observed treatment
  vector[N] A;
  // outcome
  int<lower=0,upper=1> Y[N];
}
```

# Simple example: code

https://github.com/lcomm/stancon2018/simple_mc.stan

```
transformed data {
  // make vector of 1/N for (classical) bootstrapping
  real one = 1;
  vector[N] boot_probs = rep_vector(one/N, N);
}
```

# Simple example: code

https://github.com/lcomm/stancon2018/simple_mc.stan

```
parameters {
  // regression coefficients
  vector[P + 1] alpha;
}

transformed parameters {
  vector[P] alphaZ = head(alpha, P);
  real alphaA = alpha[P + 1];
}
```

# Simple example: code

https://github.com/lcomm/stancon2018/simple_mc.stan

```
model {
  // priors for regression coefficients
  for (p in 1:(P + 1)) {
    alpha[p] ~ normal(0, 2.5);
  }

  // likelihood
  Y ~ bernoulli_logit(X * alphaZ + A * alphaA);
}
```

# Simple example: code

```
generated quantities {
  // weights for the bootstrap
  int<lower=0> counts[N] = multinomial_rng(boot_probs, N);

  // calculate ATE in the bootstrapped sample
  real ATE = 0;
  vector[N] Y_a1;
  vector[N] Y_a0;
  for (n in 1:N) {
    // sample Ya where a = 1 and a = 0
    Y_a1[n] = bernoulli_logit_rng(X[n] * alphaZ + alphaA);
    Y_a0[n] = bernoulli_logit_rng(X[n] * alphaZ);

    // add contribution of this observation
    ATE = ATE + (counts[n] * (Y_a1[n] - Y_a0[n]))/N;
  }
}
```

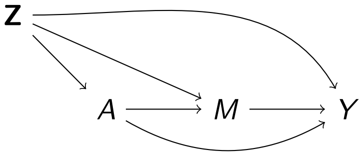
# Simple example: more on the ATE calculation

- Remember: we want $\mathbb{E}[Y_1] - \mathbb{E}[Y_1]$, which marginalizes over $\mathbf{Z}$
- Weighted average of causal effects for different $\mathbf{Z}$ values (like $P(Z = z)$ in the frequentist g-formula)
- Weights can be 0, $1/N$, $2/N$, ..., 1
- Not quite a bootstrap sample, but conceptually similar

# Mediation analysis

- Mediation analysis seeks to understand more about causal mechanisms of actions
- For every causal intermediate ("mediator") $M$, we can decompose the total effect of an exposure into two parts:
  - Part mediated by $M$ (natural indirect effect; NIE)
  - Part enacted through other pathways (natural direct; NDE)
- Policymakers want to target the causal paths with biggest impact
- Variations of mediation can be useful for policy where "exposure" is non-manipulable

# A mediation example




- Nothing in particular assumed about distribution of $Z$
- Binary treatment $A$
- Binary mediator $M$
- Binary outcome $Y$

# Mediation: models

Assume $M$ and $Y$ are generated according to logistic models:

$$\operatorname{logit}\left(P(M_i = 1|A_i, Z_i)\right) = \alpha_0 + \alpha_A A_i + \boldsymbol{\alpha}_Z' \mathbf{Z}_i$$

$$\operatorname{logit}\left(P(Y_i = 1|A_i, M_i, Z_i)\right) = \alpha_0 + \alpha_A A_i + \boldsymbol{\alpha}_Z' \mathbf{Z}_i$$

# Mediation: code

https://github.com/lcomm/stancon2018/mediation_mc.stan

Changes to data and model blocks are the addition of a model for *M*

```
data {
  ...
  vector[P + 1] beta_m;
  cov_matrix[P + 1] beta_vcv;
  ...
}
...
model {
  ...
  M ~ bernoulli_logit(X * betaZ + A * betaA);
  Y ~ bernoulli_logit(X * alphaZ + A * alphaA + Mv * alphaM);
  ...
}
```

# Mediation: code

https://github.com/lcomm/stancon2018/simple_mc.stan

Calculation of NDE is done in `generated quantities` block:

```
// calculate NDE in the bootstrapped sample
real NDE = 0;
vector[N] M_a0;
vector[N] Y_a1Ma0;
vector[N] Y_a0Ma0;
for (n in 1:N) {
  // sample Ma where a = 0
  M_a0[n] = bernoulli_logit_rng(X[n] * betaZ);

  // sample Y_(a=1, M=M_0) and Y_(a=0, M=M_0)
  Y_a1Ma0[n] = bernoulli_logit_rng(X[n] * alphaZ + M_a0[n] * alphaM +
                                    alphaA);
  Y_a0Ma0[n] = bernoulli_logit_rng(X[n] * alphaZ + M_a0[n] * alphaM);

  // add contribution of this observation to the bootstrapped NDE
  NDE = NDE + (counts[n] * (Y_a1Ma0[n] - Y_a0Ma0[n]))/N;
}
```

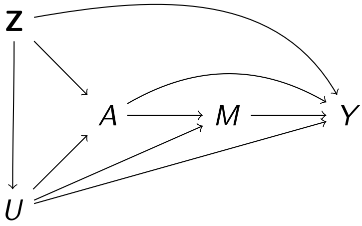
# Data integration for unmeasured confounding

- Policymakers usually have to make decisions based on available data
- We rarely have the ideal data set $\rightarrow$ often lack important confounders
- This is problematic for causal inference
- Analysts may struggle to communicate the additional uncertainty to the decision maker

# Prior information to the rescue

- Thankfully, all is not lost!
- We often have *some* information about the unmeasured confounder in another data source
- We can derive informative priors from the external data source

# Revisiting mediation example: new structure




- Now we have an unmeasured binary baseline confounder $U$

# Revisiting mediation example: new models

Assume the following generative models:

$$\operatorname{logit}\left(P(U_i = 1|\mathbf{Z}_i, A_i)\right) = \gamma_0 + \boldsymbol{\gamma}'_Z \mathbf{Z}_i$$

$$\operatorname{logit}\left(P(M_i = 1|\mathbf{Z}_i, A_i, U_i)\right) = \beta_0 + \boldsymbol{\beta}'_Z \mathbf{Z}_i + \beta_U U_i + \beta_A A_i$$

$$\operatorname{logit}\left(P(Y_i = 1|A_i, Z_i, U_i, M_i)\right) = \alpha_0 + \boldsymbol{\alpha}'_Z \mathbf{Z}_i + \alpha_U U_i + \alpha_A A_i + \alpha_M M_i$$

# Marginalization over unmeasured confounder

- Full data likelihood (i.e., if $U$ were measured)

$$\prod_{i=1}^{n} f(y_i|\boldsymbol{\alpha}, \mathbf{z}_i, a_i, m_i, u_i) f(m_i|\boldsymbol{\beta}, \mathbf{z}_i, a_i, u_i) f(u_i|\boldsymbol{\gamma}, \mathbf{z}_i)$$

- Marginalizing likelihood over binary $U$

$$\prod_{i=1}^{n} \left[ \sum_{u=0}^{1} f(y_i|\boldsymbol{\alpha}, \mathbf{z}_i, a_i, m_i, u_i = u) f(m_i|\boldsymbol{\beta}, \mathbf{z}_i, a_i, u_i = u) P(U_i = u|\boldsymbol{\gamma}, \mathbf{z}_i) \right]$$

# Incorporation of prior information

Obviously, parameters involving $U$ are unidentifiable in the original data set

- Fit maximum likelihood models in supplemental
- Use MLE from external data as priors in main analysis
  - Point estimates as prior means
  - Variance-covariance matrices as prior variances on parameter vectors
- Other data integration possibilities exist, but this one:
  - Sidesteps data privacy concerns that hinder data sharing
  - Keeps interpretability of confounder distribution

# Unmeasured confounding in mediation: code

https://github.com/lcomm/stancon2018/mediation_unmeasured_mc.stan

Likelihood in model block becomes a mixture:

```
// likelihood
for (n in 1:N) {
  // contribution if U = 0
  ll_0 = ...;

  // contribution if U = 1
  ll_1 = ...;

  // contribution is summation over U possibilities
  target += log_sum_exp(ll_0, ll_1);
}
```

# Unmeasured confounding in mediation: code

Informative priors (based on `R` model fits) are passed in as data

```
model {
  ...
  // informative priors
  alpha ~ multi_normal(alpha_m, alpha_vcv);
  beta  ~ multi_normal(beta_m, beta_vcv);
  gamma ~ multi_normal(gamma_m, gamma_vcv);
  ...
}
```

# Unmeasured confounding in mediation: code

Recreating the data-generating sequence $\mathbf{Z} \to U \to A \to M \to Y$

```
for (n in 1:N) {
  // sample U
  U[n] = bernoulli_logit_rng(pU1[n]);

  // sample M_a where a = 0
  M_a0[n] = bernoulli_logit_rng(X[n] * betaZ + U[n] * betaU);

  // sample Y_(a=0, M=M_0) and Y_(a=1, M=M_0)
  Y_a0Ma0[n] = bernoulli_logit_rng(X[n] * alphaZ + M_a0[n] * alphaM +
                                   U[n] * alphaU);
  Y_a1Ma0[n] = bernoulli_logit_rng(X[n] * alphaZ + M_a0[n] * alphaM +
                                   alphaA + U[n] * alphaU);
  ...
}
```

# Summary

- Bayesian causal inference with the parametric g-formula is a powerful tool
- The `generated quantities` block allows us to sample potential outcomes for new observations based on model for data
- Prior information is a nice way to integrate data sources and perform informed sensitivity analyses

Acknowledgments

- Collaborators Brent Coull and Linda Valeri
- NIH grants T32ES007142 and T32CA009337
- StanCon reviewers for helpful comments

# References


Imbens, G.W., and D.B. Rubin. 2015. *Causal Inference in Statistics, Social, and Biomedical Sciences*. Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction. Cambridge University Press.

Keil, Alexander P, Eric J Daza, Stephanie M Engel, Jessie P Buckley, and Jessie K Edwards. 2015. "A Bayesian Approach to the G-Formula." *Statistical Methods in Medical Research*. SAGE Publications Sage UK: London, England.

VanderWeele, T. 2015. *Explanation in Causal Inference: Methods for Mediation and Interaction*. Oxford University Press. https://books.google.com/books?id=K6cgBgAAQBAJ.