

## **Implementation**

### *Fitting $g$ across size classes*

#### Inputs

- CP trial data, observation columns, predictor columns
- SE trial data, observation columns, predictor columns
- Search schedules
- Size class factor
- Number of iterations

Run CPmodsacrosssizes

Pass output to graphing and table representation functions

- Allow user to select from the model fits for each size class
- Store the choice of model for each size class

Run ThetaCPcreateacrosssizes

Run SEmodsacrosssizes

Pass output to graphing and table representation functions

- Allow user to select from the model fits for each size class
- Store the choice of model for each size class

Run ThetaSEcreateacrosssizes

Run gcreateacrosssizes

Produces an array of  $g$  values with dimensionality (Niter, 1, Nss, Ncellcombos, Nsizeclasses)

### *Estimating $M$ across splits*

#### Inputs

- Array from gcreateacrosssizes
- Proportion weighted area searched (PWAS) for each size class x turbine x search schedule combination
- Carcass observations, split column

Run Mhatgenerator

Produces an array of  $\hat{M}$  values with dimensionality (Niter, Nss, Nturbines, Nsplitcategories, Nsizeclasses)

Pass output to graphing and table representation functions

- Allow user to condense data across grouping of interest

- Allow user to evaluate posterior distributions with regards to alpha values

## **Functions**

CPmodsacrosssizes: Wrapper function for fitting the CP models for all size classes

Inputs: CP trial data, CP observation columns, CP predictor columns, size class column

- Determine the number of size classes
- Create an empty list, length = number of size classes
- For each size class
  - Restrict the data to the size class
  - Run all 20 possible models
  - Store the model list within the element of the main list corresponding to the size class
- Return list of lists of model fits

SEmodscrosssizes: Wrapper function for fitting the SE models for all size classes

Inputs: SE trial data, SE observation columns, SE predictor columns, size class column

- Determine the number of size classes
- Create an empty list, length = number of size classes
- For each size class
  - Restrict the data to the size class
  - Run SEmodsetfit (producing length-25 list)
  - Store the length-25 list within the element of the main list corresponding to the size class
- Return list of lists of model fits

gcreateacrosssizes: Wrapper function for fitting the g distributions for all size classes

Inputs: array of CP theta tables, array of SE theta tables, search schedule,  
 list from CPmodsetfit, element # or name of selected model  
 list from SEmodsetfit, element # or name of selected model

- Run gcellsetup to create the cell setup matrix
- Determine the number of cell combos (Ncellcombos) from the cell setup matrix
- Create an empty array (Niter, 1, Nss, Ncellcombos, Nclasses)
- For each size class
  - For each cell
    - For each search schedule
      - Run gvec using the CP and SE tables according to the cell within the size class
      - Store column vector in the Niter x 1 table central to the multidimensional array
- Return multidimensional array of g values

ThetaCPcreateacrosssizes

input: list from CPmodscrosssizes, vector of element #s selected models, Niterations  
 action: draw Niter samples from the selected model for each cell for each size class  
 output: multidimensional array (Niter, 2, Ncellcp, Nsizeclasses)

### ThetaSEcreateacrosssizes

input: list from SEmodsaacrosssizes, vector of element #s selected models, Niterations  
action: draw Niter samples from the selected model for each cell for each size class  
output: multidimensional array (Niter, 2, Ncellse, Nsizeclasses)

### gvec

input: CP data table w/Niter rows and 2 cols, SE data table w/Niter rows and 2 cols, search schedule  
action: calculate g for a given value of each CP and SE parameter and search schedule  
output: g data table w/Niter rows and 1 col

### Mhatgenerator

Inputs: g array, PWAS for each size class x turbine x search schedule combo, carcass observations, carcass split column

-Determine the number of split categories, size classes, turbines, and cell combinations

-Create output array (Niter, Nss, Nturbines, Nsplitcategories, Nsizeclasses)

-Note that there is no dimension for Ncells because we sum across within the function before returning

-Also other arrays are created for utility and could be exported if desired

-For each size class q

-For each split category l

-For each turbine k

-For each Search Schedule j

-For each cell i

-subset the inputs to  $X_{ijklq}$  (length 1),  $a_{jkq}$  (length 1),  $g_{ijq}$  (length Niter)

-Draw  $\tilde{X}_{ijklq} \sim \text{Bin}\left(\frac{X_{ijklq}}{g_{ijq}}, g_{ijq}\right)$

-Calculate  $\tilde{M}_{ijklq} = \frac{\tilde{X}_{ijklq}}{g_{ijq}}$

-Sum  $\tilde{M}$  across all cells within the search schedule

-Calculate  $\hat{M}_{jkq} = \frac{\sum_{l=1}^{N_{cells}} \tilde{M}_{ijklq}}{a_{jkq}}$

-Return multidimensional array of  $\hat{M}$

## Function coding checklist

Coded:

SEmodscrosssizes

ThetaSEcreateacrosssizes

CPmodscrosssizes

ThetaCPcreateacrosssizes

gcreateacrosssizes

Mhatgenerator

logit

alogit

factorcombinations

pkfunction

gvec

ppersist

graphing functions

SE analysis

CP analysis

g

Mhat

Condense Mhat estimation

Evaluate Mhat given alpha