Online Appendix: mrobust Methods and Formulas

Appendix for the paper, "Model Uncertainty and Robustness: A Computational Framework for Multi-Model Analysis"

This appendix describes the technical details of the Stata Module "mrobust" body of commands.

Part 1: Combinations Algorithm

To generate the list of all possible *p* control terms, the mrobust algorithm counts in binary up to 2^p. The steps are as follows.

1. Create a vector *v* of zeros of length *p*, where *p* is the number of possible control variables. Each entry of *v* is a 0/1 dummy for inclusion of the corresponding variable in the model.

   ex: $p = 5$

Variable Index:	1	2	3	4	5
Initial v:	0	0	0	0	0

2. To generate the next combination, iterate through *v*, starting at index $j = 1$.

   If the j^{th} position of $v = 0$, then set that position to 1 and proceed to step 3.
   Otherwise the j^{th} position of $v = 1$, set that position to 0 and repeat step #2 starting at index $j + 1$.
3. Add this combination to the list and/or estimate the corresponding model. Repeat step #2 until reaching 2^n combinations.

4. Example:

Variable Index:		1	2	3	4	5
Combination	#1	1	0	0	0	0
	#2	0	1	0	0	0
	#3	1	1	0	0	0
	#31	1	1	1	1	1
	#32	0	0	0	0	0

Further Considerations for Either | Or Terms

For a control term x1 with no "either | or " option, there are 2 possible states for the term:
1) In
2) Out.
So for *p* rotated control terms, there are 2^p combinations.

Adding an either | or option (x1 | x1') gives us 2 options for the form of x1, increasing the number of possible states for the term to 3:

1) In, taking the form x1
2) In, taking the form x1'
3) Out.
Each additional "either | or" option for of x1 increases the number of possible states by one. So when a control term includes n "either | or" variable options, then the number of models is multiplied by n + 1.

Example:

x1 x2 (x3 | x3') (x4 | x4')

There are really 4 rotated terms.
Term #1 can be In or Out
Term #2 can be In or Out
Term #3 has 3 options(x3, x3', or Out)
Term #4 has 3 options (x4, x4', or Out)
Total # models = 2*2*3*3

In order to generate the combinations, we delineate control terms by the outermost parentheses. For each combination of *control terms*, for each term containing "either | or" options, there are more than one possible *variables* to include one at a time. To take these options into account, we use a recursive algorithm to generate the list of all possible sets of variables to fulfill each combination of control terms.

Example:

Given the combination of control terms

x2 (x3 | x3') (x4 | x4')

We generate the following sets of control variables and estimate a model for each.

x2 x3 x4
x2 x3' x4
x2 x3 x4'
x2 x3' x4'

Part 2: `mrobust` formulas

In all of the formulas, X is the variable of interest.

Basic Statistics

Mean b = simple average of the coefficient estimates _b[X] for all the models.

Sampling SE = sqrt(sum over all models(_se[X]^2))
**_se[X] is accessed from the saved results from each regression and is calculated according to the user's specification of vce (robust or cluster), or the default conventional calculation.

Modeling SE = sqrt(variance of _b[X] over all models)

Total SE = sqrt((Sampling SE)^2 + (Modeling SE)^2)

Robustness Ratio = (mean b) / (total SE)

Mean R^2 = simple average of the R^2 values for all the models.

Special Considerations for Odds Ratios (OR)

(Logistic, Poisson IRR, and NBreg IRR estimation commands. These commands report odds ratios. The saved coefficients _b[X] are the log-odds. We execute all the calculations directly on _b[X] and convert to odds ratio terms at the end for display.)

Mean(OR) = exp(mean(_b[X])
Sampling SE = Mean(OR)*sqrt(sum of _se[X]^2)
Modeling SE = Mean(OR)*sqrt(variance of _b[X])
Total SE = sqrt((sampling SE)^2 + (modeling SE)^2)
= Mean(OR)*Total SE of _b[X]

Robustness Ratio = mean(_b[X])/sqrt(variance of the total distribution of _b[X])
= robustness ratio for the corresponding logit command.

Pos = 1 if OR > 1; equivalently, _b[X] > 0
Sig = 1 based on p-value for _b[X]

Significance Calculations

For reg, areg, rreg, xtreg(fe), use the t-distribution with df = residual degrees of freedom of the model:
t = _b[intvar]/_se[intvar]
P-value = 2*ttail(e(df_r), abs(t))

For all other models, and all models when using the nonparametric bootstrap, use the standard normal distribution:
z = _b[intvar]/_se[intvar]
P-value = 2*(1-normal(abs(z)))

Default alpha = .05, option alpha(X) sets alpha to the value X passed in.
Coefficients with p-value < = alpha are counted as significant.

Influence Calculations

We run three OLS regressions on the model results to determine the marginal effect of variable inclusion and functional form on 1) value of the estimate, 2) probability of a statistically significant estimate, and 3) probability of a positive estimate.

For these regressions each model estimated becomes a row/observation. The respective y- variables are

1) _b[varint]
2) dummy for significant: 1 if P-value of _b[varint] <= alpha, 0 else
3) dummy for positive: 1 if _b[varint] > 0, 0 else

The x-variables for all of the regressions are dummies for inclusion of each of the rotated control terms, and dummies for the different functional forms, dependent variables, and/or independent variables, if more than one choice is specified for any of these categories.

Bootstrap Option

bs(bs_type) performs resampling of the data or estimate to generate a sampling distribution of size B = 50 for the parameter of interest for each of the J models. These B*J bootstrapped estimates are saved in the results file, composing the total sampling+modeling distribution, which is used to calculate the total SE and the robustness intervals. bs_type may be par (parametric) or nonpar (nonparametric).

Under the parametric bootstrap, the B estimates for each model are random samples from the normal distribution with mean _b[X] and SE _se[X].

Under the nonparametric bootstrap, Stata's bootstrap command is executed on each model, resampling the actual data points and re-estimating the model B times to generate B parameter estimates. Additional bootstrap options accommodate data-specific resampling requirements such as strata(*varlist*) or cluster(*varlist*)), and these options are passed in directly to Stata's bootstrap on each iteration.

Intervals

Modeling Distribution 95% interval = 2.5 and 97.5 percentiles of the estimates

Modeling Distribution Extreme Bounds = Min and Max of the estimates

Total Distribution Parametric = Mean(b) $\pm$ 2*(Total SE)
**Where the sampling part of the Total SE comes from the user-specified SEs [robust, clustered, etc])

Total Distribution (Bootstrap) = 2.5 and 97.5 percentiles of the total bootstrapped distribution