SNP-Wise Selection Experiment

Stuart Brabbs

6/30/2021

Our goal is to investigate a potential selection algorithm where one-by-one we force each variable (SNP) to be in the model, and then run stepwise forward (or backward) selection on the remaining variables conditional on the other variable already in the model. We repeat this for each variable. Uncertain at the moment whether we should place some sort of weight on each step based on the likelihood of the variable's inclusion or not.

The R package "StepReg" is particularly helpful in investigating this because in addition to specifying type of selection, it also allows one to force it to include a certain variable in the selection. This hints at a possible iterative process where in addition to the stepwise selection process, we have an outer loop where we iterate through each variable as the required selection.

We start with a simple example of 3 variables x1, x2, and x3, and outcome y. x2 is correlated with x1 and x3, but only x1 and x3 have a true causal effect on y. We run forward (and then backward) selection without forcing the inclusion of any variable, and then one-by-one force the inclusion of each one, and compare the results.

```
set.seed(100)
x1 <- rnorm(1000)
x3 <- rnorm(1000)
x2 <- x1 + x3 + rnorm(1000, sd = 0.5)
y <- x1 + x3 + rnorm(1000)

df1 <- data.frame(x1, x2, x3, y)

reg0 <- stepwise(df1, y = "y", selection = "forward")
reg1 <- stepwise(df1, y = "y", include = "x1", selection = "forward")
reg2 <- stepwise(df1, y = "y", include = "x2", selection = "forward")
reg3 <- stepwise(df1, y = "y", include = "x3", selection = "forward")

reg4 <- stepwise(df1, y = "y", selection = "backward")
reg5 <- stepwise(df1, y = "y", include = "x1", selection = "backward")
reg6 <- stepwise(df1, y = "y", include = "x2", selection = "backward")
reg7 <- stepwise(df1, y = "y", include = "x3", selection = "backward")
```

For the forward selection, the following variables are selected for each forced variable:

Forced Var.	Selected Vars. (In Order)
None	X2 X1 X3
X1	(X1) X3
X2	(X2) X1 X3
X3	(X3) X1

For the backward selection, the following variables are selected for each forced variable:

Forced Var.	Selected Vars.
None	X1 X3
X1	(X1) X3
X2	X1 (X2)
X3	X2 (X3)

We can see that there are significant changes in selection when we force the inclusion of a variable. In forward selection, the true model is selected when X1 or X3 is forced, while forcing X2 (or none) selects every variable. In backward selection, only forcing X1 (or none) led to the true model being selected.

Of course, we cannot simply average over these and be happy because that would assume each variable is equally likely to be selected. We may need some sort of weight to place on the model produced from forcing each variable. This may be less necessary, however, if we have many variables.

If we go simply by averaging over the three forced variables, for forward selection we see that X1 and X3 are selected 100% of the time, and X2 is selected just 1/3 of the time, and then only when we force it. For backward selection, X1, X2, and X3 are each selected 2/3 of the time.

One possibility for weights could simply be to weight by the SuSiE PIPs. For this example, the SuSiE PIPs (L=3) are:

```
X <- cbind(x1, x2, x3)
sus1 <- susie(X, y, L=3)
sus1$pip
```

```
## x1 x2 x3
##  1  0  1
```

This is, of course, a rather simple example, so the PIPs in this case assign 1 to X1 and X3 and 0 to X2. If we proceed with the above idea, however, we can assign the weights to produce new selection "probabilities" (for lack of a better word) for each variable, starting with the forward stepwise:

$$X1 : (PIP(X1) * 1 + PIP(X2) * 1 + PIP(X3) * 1)/(PIP(X1) + PIP(X2) + PIP(X3)) = (1 * 1 + 0 * 1 + 1 * 1)/(1 + 0 + 1) = 1$$

$$X2 : (1 * 0 + 0 * 1 + 0 * 1)/(1 + 0 + 1) = 0$$

$$X3 : (1 * 1 + 0 * 1 + 1 * 1)/(1 + 0 + 1) = 1$$

In the forward case little changes except that X2's selection chances go to 0. For backward stepwise, we have:

$$X1 : (1 * 1 + 0 * 1 + 1 * 0)/(1 + 0 + 1) = 1/2$$

$$X2 : (1 * 0 + 0 * 1 + 1 * 1)/(1 + 0 + 1) = 1/2$$

$$X3 : (1 * 1 + 0 * 0 + 1 * 1)/(1 + 0 + 1) = 1$$

We thus see that weighting like this changes the output from backward stepwise selection. Whereas before each variable was selected 2/3 of the time, here X3 is always selected, while X1 and X2 are slightly lowered to 1/2. Thus this system of weights may be somewhat beneficial, but gives slightly mixed results in this simple example. SuSiE PIP weights may work more effectively if we have many causal variables, however.

Another possibility is to use relative weights analysis (RWA). RWA computes a form of relative importance for predictors in modeling the outcome. For our example, RWA returns the following weights (which sum to 100):

```
rwa1 <- rwa(df1, "y", c("x1", "x2", "x3"))
rwa1$result[c(-2,-4)]
```

```
##   Variables Rescaled.RelWeight
## 1        x1           37.58940
## 2        x2           32.26463
## 3        x3           30.14597
```

If we then weight our results from the original selection by these weights (divided by 100), for forward stepwise we get:

$$X1 : (0.3759 * 1 + 0.3226 * 1 + 0.3015 * 1) = 1$$

$$X2 : (0.3759 * 0 + 0.3226 * 1 + 0.3015 * 0) = 0.3226$$

$$X3 : (0.3759 * 1 + 0.3226 * 1 + 0.3015 * 1) = 1$$

and for backward stepwise we get:

$$X1 : (0.3759 * 1 + 0.3226 * 1 + 0.3015 * 0) = 0.6985$$

$$X2 : (0.3759 * 0 + 0.3226 * 1 + 0.3015 * 1) = 0.6241$$

$$X3 : (0.3759 * 1 + 0.3226 * 0 + 0.3015 * 1) = 0.6774$$

Thus while the forward selection case remains about the same (though X2 is given a slightly higher value), with backward selection the values deviate from 2/3 for each, with X1 and X3 being higher and X2 being lower. RWA may be a good option for weighting in the SNP-wise algorithm if we can better define what the output number indicates.