SNP-wise Regression Summary

Stuart Brabbs

Exact Function

We begin by defining an exact function to calculate the posterior inclusion probabilities (PIPs) of each SNP by cycling through all possible models and using the posterior model probabilities (PMPs) of each model:

```
exact = function(X,y,sigmaa) {
  snps <- dim(X)[2]
  nummodels <- 2^snps
  pips <- rep(0,snps)
  genweights <- snpwise_weights(X,y,sigmaa)

  snpnames <- c()
  for (i in c(1:snps)){
    newvar <- paste0("x", i)
    snpnames <- c(snpnames, newvar)
  }

  for (i in c(2:nummodels)){
    ourrow <- genweights[i,]
    ourmodel <- unlist(strsplit(as.character(ourrow$models), ","))
    for (j in c(1:snps)){
      if (snpnames[j] %in% ourmodel){
        pips[j] <- pips[j] + ourrow$pmps
      }
    }
  }

  toreturn <- data.frame(snpnames, pips)
  names(toreturn)[1] <- "snp"
  names(toreturn)[2] <- "pip"
  return(toreturn)
}
```

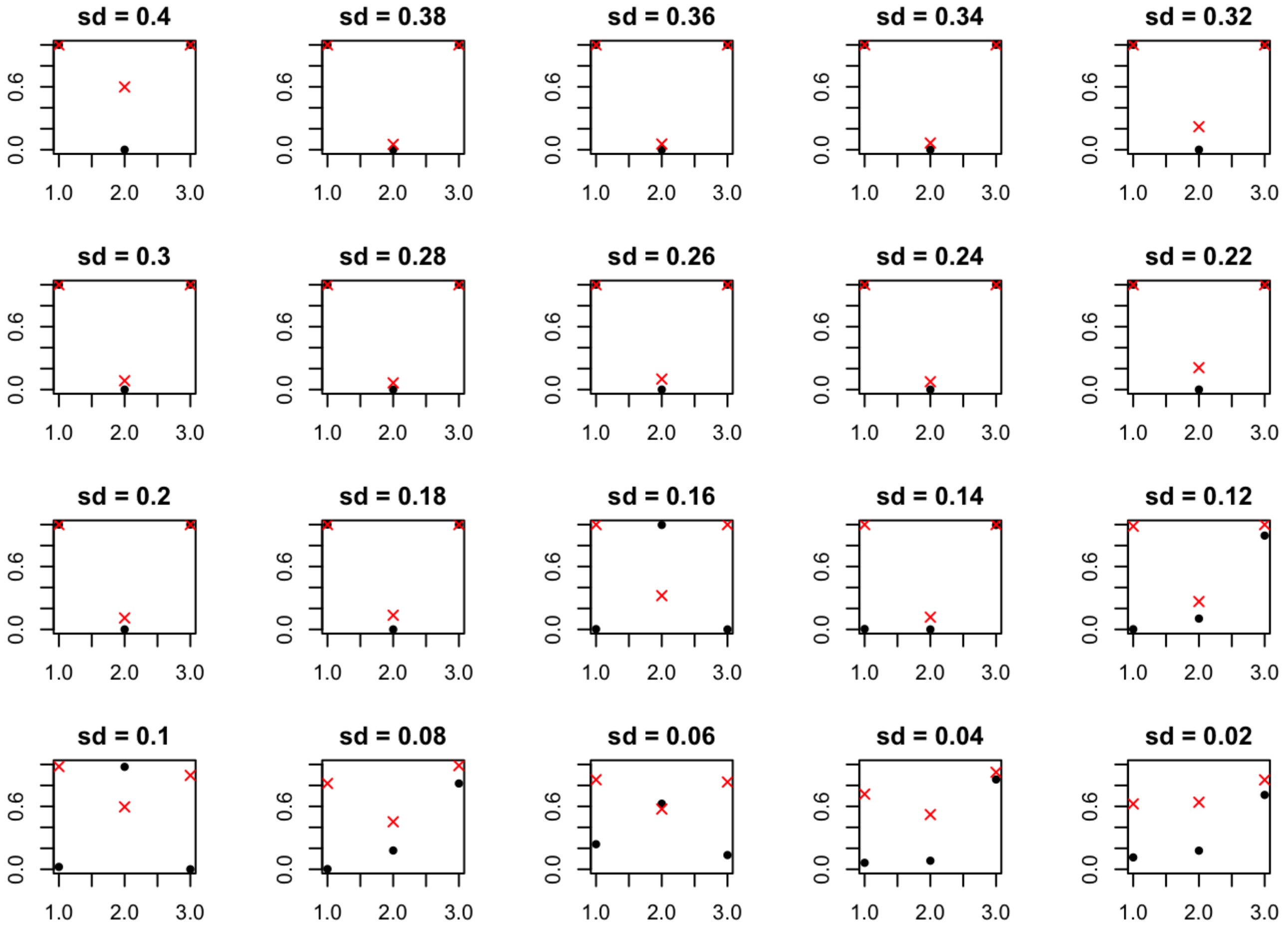
Three Body Problem

We now define the "three-body problem." The basic model of this experiment is a situation where some variable Y depends on variables X1 and X3, which are each highly correlated with a third variable X2, but only correlated with each other as an artifact of their correlation with X2. The true model is thus \(Y = \beta_1X_1 + \beta_3X_3\). However, competing models involving X2 are \(Y = \beta_2X_2\), \(Y = \beta_1X_1 + \beta_2X_2\), and \(Y = \beta_2X_2 + \beta_3X_3\). To simulate this data, we assume generate X1 and X3 separately and then generate X2 from the model \(X_2 = X_1 + X_3 + \epsilon\), \(\epsilon \sim N(0,1)\). We generate an example with n = 100 and 20 different correlations, and run the exact function on it:

SuSiE

We now run the same data with SuSiE, and compare the resulting PIPs with those of the exact function. SuSiE PIPs are black dots, and exact function PIPs are red crosses:

```
set.seed(100)
sds <- sort(seq(0.02, 0.4, by=0.02), decreasing = TRUE)
par(mfrow = c(4,5))
for (i in c(1:20)){
  x2 <- rnorm(1000)
  x1 <- x2 + rnorm(1000, sd = sds[i])
  x3 <- x2 + rnorm(1000, sd = sds[i])
  X <- cbind(x1, x2, x3)
  y <- x1 + x3 + rnorm(1000)

  exactreg <- exact(X,y,0.5)
  susiereg <- susie(X,y,L=2)

  title <- paste0("sd = ", sds[i])
  par(mar = c(3,2,2,3))
  plot(susiereg$pip, xlab = "Predictor", ylab = "PIP", main = title, pch = 20, ylim = c(
0,1))
  par(new=TRUE)
  plot(exactreg$pip, ylim = c(0,1), axes = FALSE, pch = 4, col = 2)
}
```

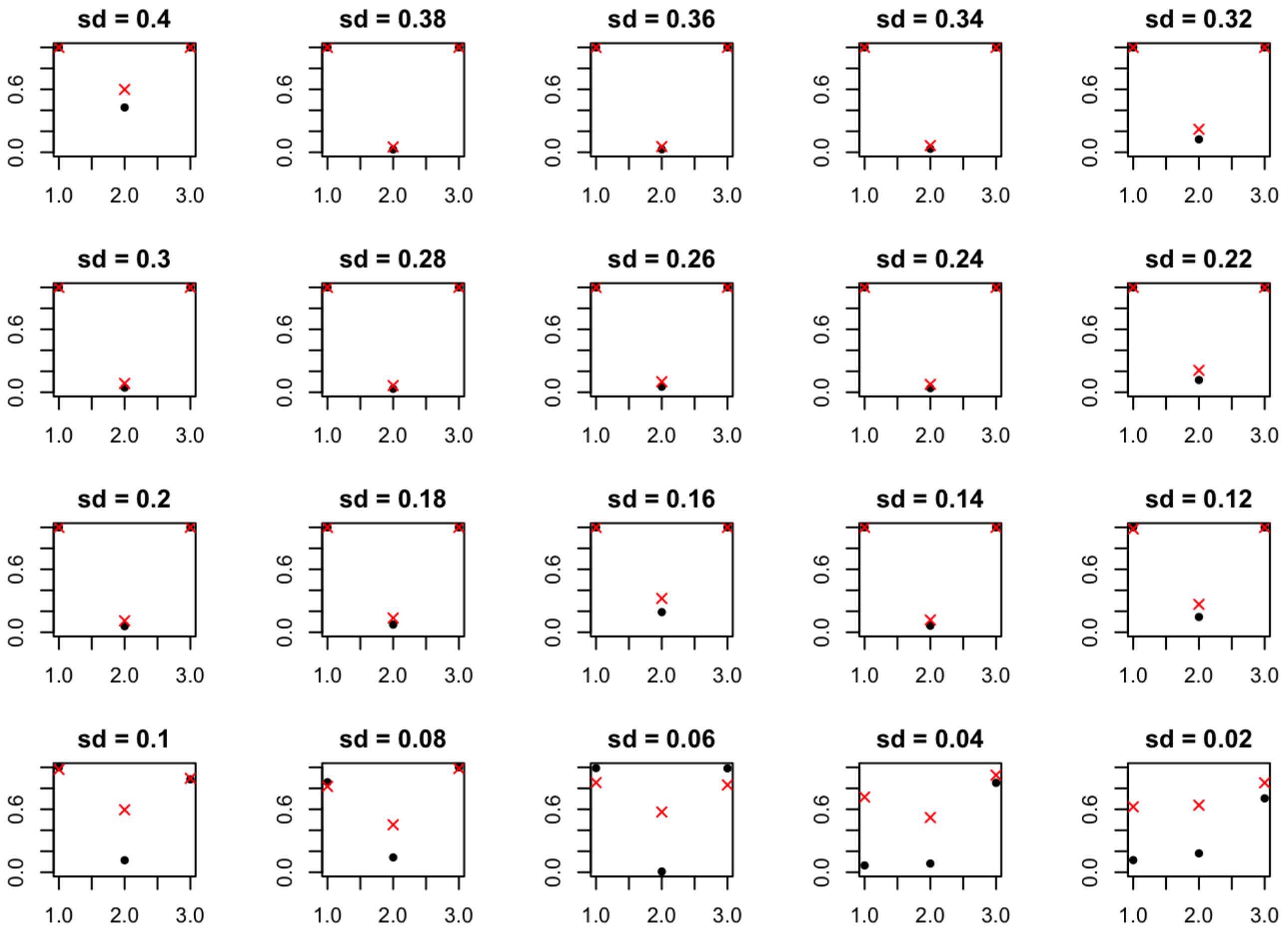


SNP-wise Regression

We now define SNP-wise regression, where we loop through each SNP, forcing it to be included in the model, do stepwise regression with it included, and add the corresponding PMP to those variables selected. We then run it on the same data from above and compare with the exact model, again with SNP-wise PIPs in black and exact PIPs as red crosses:

```
snpwisereg = function(X,y,sigmaa, priorpi){
  normalize <- 0
  #if no prior input, set to 1/number of parameters
  if (missing(priorpi)){
    p <- par
  }
  else {
    p <- 1/priorpi
  }

  df <- data.frame(X,y)
  #get number of variables
  snps <- dim(X)[2]
  #vector of variable names
  snpnames <- c()
  #vector of variable PIPs
  pips <- rep(0,snps)

  for (i in c(1:snps)){
    colnames(df)[i] <- paste0("x",i)
    snpnames <- c(snpnames, paste0("x",i))
  }
  genweights <- snpwise_weights(X,y,sigmaa,priorpi)

  #loop through each variable and stepwise regress while forcing to
  #include the variable
  for (i in c(1:snps)){
    reg <- stepwise(df, y = colnames(df)[snps+1], include = colnames(df)[i], selection =
"forward")
    selected <- reg$variate
    #filter out occasional duplicates in selected variables
    selected <- unique(selected)
    if (selected[1]=="intercept"){
      selected <- selected[-1]
    }
    incl <- ""
    numincl <- 0

    #make list of variables selected
    for (j in c(1:snps)) {
      if (paste0("x",j) %in% selected) {
        numincl <- numincl + 1
        if (incl == "") {
          incl <- paste0("x",j)
        }
        else {
          toincl <- paste0("x",j)
          incl <- paste(incl, toincl, sep=",")
        }
      }
    }

    #filter data to only include selected variables
    ourrow <- filter(genweights, models == incl)
    #update normalizing constant
    normalize <- normalize + ourrow$pmps
    #add model PMP to included variables' PIPs
    for (k in c(1:snps)){
      if (colnames(df)[k] %in% selected){
        pips[k] <- pips[k] + ourrow$pmps}
    }
  }

  pips <- pips/normalize

  #return snp names and corresponding PIPs
  toreturn <- data.frame(snpnames, pips)
  names(toreturn)[1] <- "snp"
  names(toreturn)[2] <- "pip"
  return(toreturn)
}
```

```
set.seed(100)
sds <- sort(seq(0.02, 0.4, by=0.02), decreasing = TRUE)
par(mfrow = c(4,5))
for (i in c(1:20)){
  x2 <- rnorm(1000)
  x1 <- x2 + rnorm(1000, sd = sds[i])
  x3 <- x2 + rnorm(1000, sd = sds[i])
  X <- cbind(x1, x2, x3)
  y <- x1 + x3 + rnorm(1000)

  exactreg <- exact(X,y,0.5)
  snpreg <- snpwisereg(X,y,0.5)

  title <- paste0("sd = ", sds[i])
  par(mar = c(3,2,2,3))
  plot(snpreg$pip, xlab = "Predictor", ylab = "PIP", main = title, pch = 20, ylim = c(0,
1))
  par(new=TRUE)
  plot(exactreg$pip, ylim = c(0,1), axes = FALSE, pch = 4, col = 2)
}
```



We can see that, compared to SuSiE, for the three-body problem SNP-wise regression holds up similarly to the exact function for significantly greater correlations.