Chapter 1

Examples

In the following section the Bayesian LMG implementation is presented on two examples. The first examples simulates data, the second examples uses real data.

Lets assume a simple model:

$$
\begin{align}
Y_i &\sim \mathcal{N}(\beta_0 + x_1\beta_1 + x_2\beta_2 + x_3\beta_3 + x_4\beta_4, \sigma^2), \tag{1.1}\\
\beta_1 &= 0.5, \beta_2 = 1, \beta_3 = 2, \beta_4 = 0, \sigma^2 = 1 \tag{1.2}\\
\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4 &\sim \mathcal{N}(0, 1) \tag{1.3}
\end{align}
$$

The values of the four predictors are sampled from a standard normal distribution. These values are then multiplied by the regression coefficients to obatin the dependent variable. A standard normal distributed error is added. Fifty observations were sampled.

The following Code was used to simulate the data :

```r
x1 <- rnorm(50, 0, 1); x2 <- rnorm(50, 0, 1)
x3 <- rnorm(50, 0, 1); x4 <- rnorm(50, 0, 1)
b1 <- 0.5; b2 <- 1; b3 <- 2; b4 <- 0

y <- b1*x1 + x2*b2 + b3*x3 + b4*x4 + rnorm(50, 0, 1)

df <- data.frame(y = y, x1 = x1, x2 = x2, x3 = x3, x4 = x4)
```

The model is fitted using the rstanarm package with the default priors for the regression and σ^2 parameter. For computational reasons a small burning periode of 1000 and a sample size of 1000 were chosen. For each posterior sample of the parameters the R^2 value is calculated. The R^2 of the submodels is then calculated by the conditional variance formula for each posterior sample.

```r
post2 <- stan_glm(y ~ 1 + x1 + x2 + x3 + x4,
                  data = df,
                  chains = 1, cores = 1)
```

```
##
## SAMPLING FOR MODEL 'continuous' NOW (CHAIN 1).
##
## Gradient evaluation took 0.000132 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 1.32 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.077623 seconds (Warm-up)
##                0.090766 seconds (Sampling)
##                0.168389 seconds (Total)
```

```
#posterior sample
post.sample <- as.matrix(post2)

#example of the first 10 posterior samples
post.sample[1:10,]
```

```
##           parameters
## iterations (Intercept)        x1        x2       x3        x4     sigma
##       [1,]  0.38240146 0.6761560 1.0736295 1.802415 0.2851963 0.9836808
##       [2,]  0.22356234 0.6970482 1.2601508 1.831950 0.4274150 1.1778742
##       [3,]  0.22926236 0.6704861 1.2175890 1.896573 0.3152204 1.2015425
##       [4,]  0.14122344 0.6621194 1.2281334 1.897343 0.2915838 1.2050201
##       [5,]  0.06846917 0.5128563 1.0836482 2.108757 0.2369996 0.7239385
##       [6,]  0.18249110 0.8330410 1.1844066 2.102656 0.4920880 0.9960685
##       [7,]  0.16108763 0.6867353 0.9972163 2.242331 0.6404463 1.3039173
##       [8,]  0.12771342 0.7132535 1.2105716 1.831717 0.2440454 0.8898359
##       [9,]  0.14336106 0.4831619 1.0644213 2.349252 0.3949097 0.9955502
##      [10,]  0.23785815 0.4704278 0.9126057 2.312152 0.4582012 1.0337756
```

```
#no need for the intercept, last parameter is sigma
post.sample <- post.sample[,-1]


#data frame with all submodels
df.rtwos <-rtwos(df[,2:5], post.sample)

df.rtwos[,1:5]

##                       X1          X2           X3           X4           X5
## none     0.000000e+00 0.000000000 0.000000e+00 0.000000e+00 0.0000000000
## x1       1.114441e-01 0.103104497 9.633566e-02 9.519329e-02 0.0620979108
## x2       3.712667e-01 0.387828823 3.766445e-01 3.811665e-01 0.3624957992
## x3       5.330658e-01 0.483609918 5.029314e-01 5.019089e-01 0.6697525164
## x4       2.345555e-05 0.001466615 3.347443e-05 6.779977e-06 0.0000177982
## x1 x2    4.141661e-01 0.424350727 4.098158e-01 4.133605e-01 0.3773894629
## x1 x3    6.797805e-01 0.619011096 6.311953e-01 6.288172e-01 0.7622104301
## x1 x4    1.151793e-01 0.111734242 9.971400e-02 9.762428e-02 0.0642354195
## x2 x3    7.623605e-01 0.732892589 7.404851e-01 7.432618e-01 0.8761592564
## x2 x4    3.892926e-01 0.417176576 3.951834e-01 3.976429e-01 0.3799425057
## x3 x4    5.344987e-01 0.483615269 5.042047e-01 5.038490e-01 0.6716570726
## x1 x2 x3 8.401879e-01 0.799755303 8.036483e-01 8.049840e-01 0.9176347851
## x1 x2 x4 4.403613e-01 0.463220254 4.355967e-01 4.365735e-01 0.3995229195
## x1 x3 x4 6.803972e-01 0.622417938 6.317208e-01 6.290170e-01 0.7622456935
## x2 x3 x4 7.673754e-01 0.745426349 7.461097e-01 7.477922e-01 0.8796711328
## all      8.510266e-01 0.820242179 8.146876e-01 8.143904e-01 0.9246266942
```

After the R^2 for each posterior sample and their corresponding submodels is calculated, the package hier.part is used to calculate the LMG value for each posterior sample.

```
# prepare data frame for LMG values

LMG.Vals<-matrix(0, 4, dim(df.rtwos)[2])

for(i in 1:dim(df.rtwos)[2]){

  gofn<-df.rtwos[,i]

  obj.Gelman<-partition(gofn, pcan = 4, var.names = names(df[,2:5]))

  LMG.Vals[,i]=obj.Gelman$IJ[,1]
}
```

```
# posterior LMG distribution of each variable
quantile(LMG.Vals[1,], c(0.025, 0.5, 0.975))

##       2.5%        50%      97.5%
## 0.03166254 0.07027119 0.12049975

quantile(LMG.Vals[2,], c(0.025, 0.5, 0.975))

##      2.5%       50%     97.5%
## 0.1770068 0.2589506 0.3423585

quantile(LMG.Vals[3,], c(0.025, 0.5, 0.975))

##      2.5%       50%     97.5%
## 0.4240630 0.5305730 0.6207969

quantile(LMG.Vals[4,], c(0.025, 0.5, 0.975))

##        2.5%         50%       97.5%
## 0.003023520 0.009984325 0.036070289

# Comparison to relaimpo package

fit <- lm(y~., data=df)

######## compare to relimp package

run<-boot.relimp(fit, fixed=TRUE)

booteval.relimp(run, bty = "perc", level = 0.95,
                sort = FALSE, norank = FALSE, nodiff = FALSE,
                typesel = c("lmg", "pmvd", "last", "first", "betasq", "pratt", "genizi", "c

## Response variable: y
## Total response variance: 7.600138
## Analysis based on 50 observations
##
## 4 Regressors:
## x1 x2 x3 x4
## Proportion of variance explained by model: 88.64%
## Metrics are not normalized (rela=FALSE).
##
## Relative importance metrics:
##
##           lmg
## x1 0.07118799
## x2 0.26455572
## x3 0.54037183
## x4 0.01026747
##
## Average coefficients for different model sizes:
##
##            1X       2Xs       3Xs       4Xs
## x1 0.69225408 0.6474797 0.6281180 0.6276475
## x2 1.57752966 1.4613948 1.3141362 1.1452477
## x3 2.30010895 2.2349317 2.1614571 2.0837687
## x4 0.09441481 0.2236843 0.3115496 0.3575779
##
##
##  Confidence interval information ( 1000 bootstrap replicates, bty= perc ):
## Relative Contributions with confidence intervals:
##
##                        Lower  Upper
##        percentage 0.95 0.95   0.95
## x1.lmg 0.0712     __C_ 0.0318 0.1242
## x2.lmg 0.2646     _B__ 0.1908 0.3526
## x3.lmg 0.5404     A___ 0.4547 0.6314
## x4.lmg 0.0103     ___D 0.0031 0.0358
##
## Letters indicate the ranks covered by bootstrap CIs.
## (Rank bootstrap confidence intervals always obtained by percentile method)
## CAUTION: Bootstrap confidence intervals can be somewhat liberal.
## NOTE: X-matrix has been considered as fixed for bootstrapping.
##
##
##  Differences between Relative Contributions:
##
##                           Lower   Upper
##           difference 0.95 0.95    0.95
## x1-x2.lmg -0.1934     *   -0.2927 -0.0972
## x1-x3.lmg -0.4692     *   -0.5785 -0.3532
## x1-x4.lmg  0.0609     *    0.0123  0.1152
## x2-x3.lmg -0.2758     *   -0.4292 -0.1193
## x2-x4.lmg  0.2543     *    0.1721  0.3389
## x3-x4.lmg  0.5301     *    0.4377  0.6207
##
## * indicates that CI for difference does not include 0.
## CAUTION: Bootstrap confidence intervals can be somewhat liberal.
## NOTE: X-matrix has been considered as fixed for bootstrapping.
```

The first example data are taken from the book Bayesian Regression Modeling with INLA. The data were about air pollution in 41 cities in the United States originally published in Everitt (2006). The data consits of the SO2 level as the dependent variable and six explanatory variables Two of the explanatory variables are are related to human ecology (pop, manuf) and four others are related to climate (negtemp, wind, precip, days).