Gaussian Data Example

Eric Dunipace

5/6/2019

Setup

Predictors

We can measure variable importance in a variety of settings. Our first example is in a setting with Normally distributed data. We generate $n = 1024$ predictor variables, X, from a Multivariate Normal,

$$X_i \sim N(0, \Sigma),$$

with $\Sigma_{i,i} = 1$ for all i from 1 to $p = 30$. The first 20 dimensions of X are correlated, and dimensions 21 to 30 are correlated with correlations 0.5. Thus, Σ is block diagonal.

Parameters

We generate parameters as follows

$$\beta_{0:5} \sim \text{Unif}(1, 2),$$

$$\beta_{6:10} \sim \text{Unif}(-2, -1),$$

$$\beta_{11:20} \sim \text{Unif}(0, 0.5),$$

and

$$\sigma^2 = 1.$$

Outcome

We sample the outcome as

$$Y_i \sim N(\beta_0 + X_i^\top \beta_{1:20}, \sigma^2)$$

```r
#### Load packages ####
require(SparsePosterior)
require(ggplot2)
require(CoarsePosteriorSummary)
require(rstan)
require(ggsci)
require(doParallel)
require(ggridges)
require(data.table)

rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores()-1)

group.names <- c("Selection","Loc./Scale", "Projection")

#### generate data ####
set.seed(9507123)
n <- 1024
```

```
p <- 31
nsamp <- 1000
nlambda <- 100
lambda.min.ratio <- 1e-10
gamma <- 1
pseudo.observations <- 0

target <- get_normal_linear_model()
param <- target$rparam()
p_star <- length(param$theta)

target$X$corr <- 0.5
X <- target$X$rX(n, target$X$corr, p)
Y <- target$rdata(n,X[,1:p_star], param$theta, param$sigma2)

x <- target$X$rX(1, target$X$corr, p)

true_mu_full <- X[,1:p_star, drop=FALSE] %*% c(param$theta)
true_mu <- x[,1:p_star,drop=FALSE] %*% c(param$theta)
```

Posterior Estimation

We put conjugate priors on our parameters,

$$\beta_{0:p} \sim N(0,1)$$

and

$$\sigma^2 \sim \text{Inv-Gamma}(10,10),$$

which facillitates quick estimation via known posterior distributions.

```
hyperparameters <- list(mu = NULL, sigma = NULL,
                         alpha = NULL, beta = NULL,
                         Lambda = NULL)
  hyperparameters$mu <- rep(0,p)
  hyperparameters$sigma <- diag(1,p,p)
  hyperparameters$alpha <- 10
  hyperparameters$beta <- 10
  hyperparameters$Lambda <- solve(hyperparameters$sigma)

# posterior <- target$rpost(nsamp, X, Y, method = "stan",
#                            stan_dir ="../../Stan/normal_horseshoe_noQR.stan",
#                            m0 = 20, scale_intercept = 2.5, chains = 4) # this can take a bit

posterior <- target$rpost(nsamp, X, Y, hyperparameters = hyperparameters, method="conjugate")


cond_mu <- x %*% posterior$theta
cond_mu_full <- X %*% posterior$theta

penalty.factor <- 1/sqrt(rowSums(posterior$theta^2))
```

We then utilize our 1) selection variable, 2) location/scale, and 3) projection methods to find coarse posteriors close to our full one. Note: this code may take a bit to run.

```r
adapt_file <- "gaussian_adapt.RData"
if(!file.exists(adapt_file)){
    theta_selection <- W2L1(X=X, Y = cond_mu_full,
                             theta=posterior$theta, penalty="selection.lasso",
                             nlambda = nlambda, lambda.min.ratio = lambda.min.ratio,
                             infimum.maxit=1e4, maxit = 1e3, gamma = gamma,
                             pseudo_observations = 0, display.progress = TRUE,
                      penalty.factor = penalty.factor, method="selection.variable")

  #these iterations may not be enough
  theta_adaptive <-W2L1(X=X, Y=cond_mu_full,
                             theta=posterior$theta, penalty="mcp",
                             nlambda = nlambda, lambda.min.ratio = lambda.min.ratio,
                             infimum.maxit=1e2, maxit = 5e2, gamma = gamma,
                             pseudo_observations = pseudo.observations, display.progress = TRUE,
                      penalty.factor = penalty.factor,
                       method="location.scale")
  theta_proj <- W2L1(X=X, Y=cond_mu_full,
                             theta=posterior$theta, penalty="mcp",
                             nlambda = nlambda, lambda.min.ratio = lambda.min.ratio,
                             infimum.maxit=1, maxit = 1e3, gamma = gamma,
                             pseudo_observations = pseudo.observations, display.progress = TRUE,
                      penalty.factor = penalty.factor, method="projection")
  save(theta_proj, theta_adaptive, theta_selection, file=adapt_file)
} else {
  load(adapt_file)
}


theta <- list(selection = NULL, adaptive = NULL, projection = NULL)

theta$selection <- extractTheta(theta_selection, posterior$theta)
theta$adaptive <- extractTheta(theta_adaptive, posterior$theta)
theta$projection <-  extractTheta(theta_proj, posterior$theta)
```

Then we measure our distance from the posterior predictive mean and posterior distribution.

```r
models <- list(Selection = theta_selection,
               "Loc./Scale" = theta_adaptive,
               Projection = theta_proj)
gauss_plot_file <- "w2plot.rds"
if ( !file.exists(gauss_plot_file) ){ #only run if file doesn't exist
  w2plot <- plot.compare(models, cond_mu_full, X, posterior$theta,
                        "w2", c("posterior","mean"), parallel = TRUE) # this can take a LONG time
  saveRDS( w2plot, gauss_plot_file )
} else {
  w2plot <- readRDS( gauss_plot_file )
}
```

We also measure the mean-squared error from the true mean, since it is known in this case.

```r
#these should run faster
mseplot_mean <- plot.compare(models, true_mu_full, X, posterior$theta,
                             "mse", c("mean"), parallel = TRUE)
mseplot_posterior <- plot.compare(models, c(param$theta, rep(0,p-p_star)), X, posterior$theta,
```

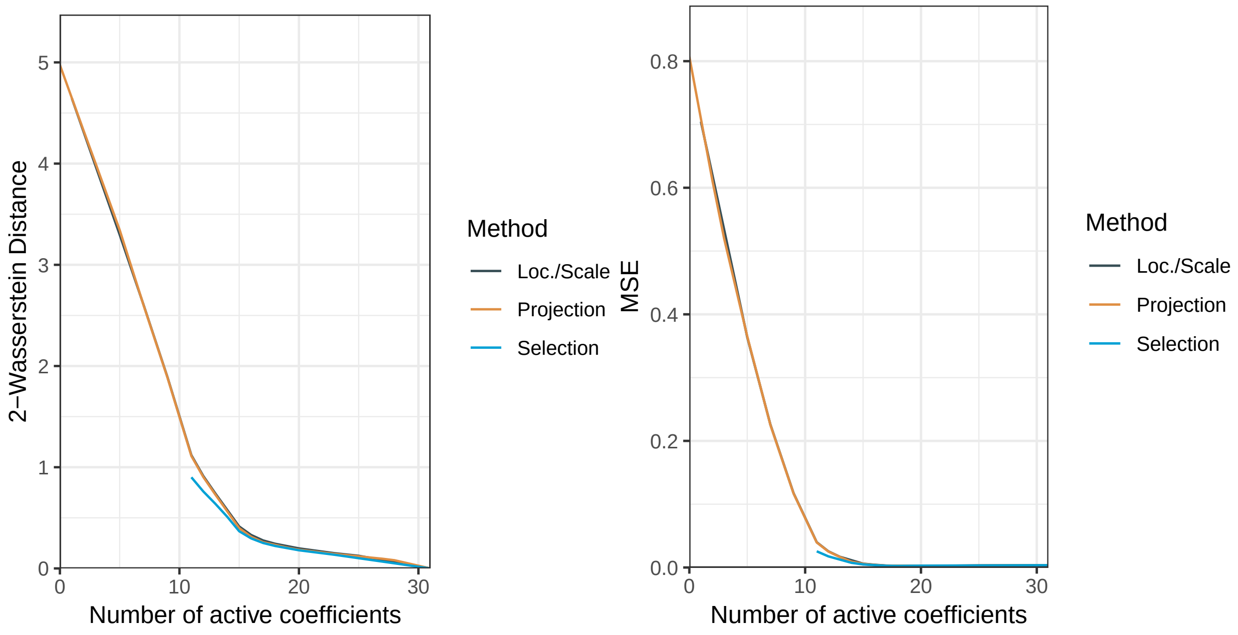


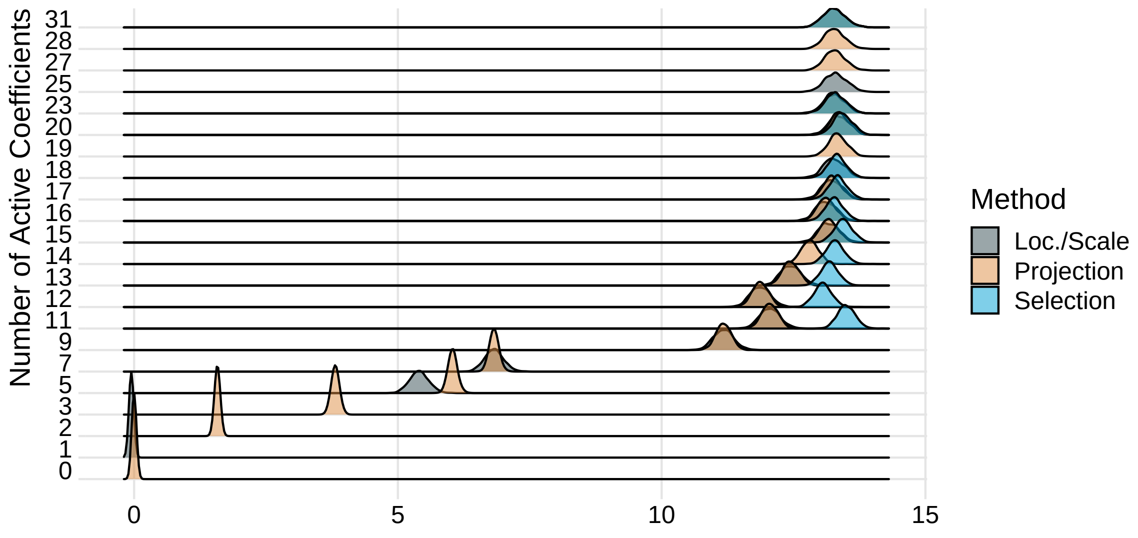
Figure 1: 2-Wasserstein distance between the full posterior for the regression coefficients and our coarsened version.

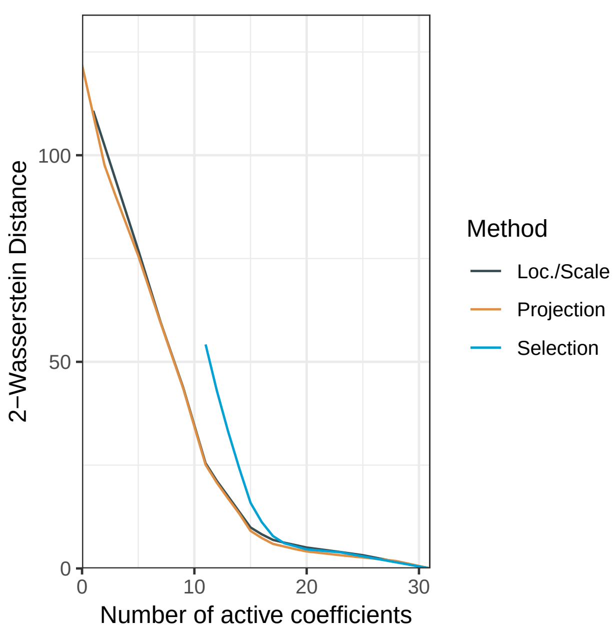
```
"mse", c("posterior"), parallel = TRUE)
```

Using the Wasserstein distance, we can see how the coarse versions move closer to both the full posterior. The coarse posteriors also converge in MSE to the true parameters

And we can look at the differences the posterior predictive mean and the full posterior predictive mean, as well as the MSE between the coarse posterior and true mean.

Finally, we can see as more covariates are active, the coarse distribution gets close to the truth. Importantly, it looks like the selection method does better here for this individual $i = 512$. However, overall, the projection and location/scale methods do better, which we can see in the above plots.





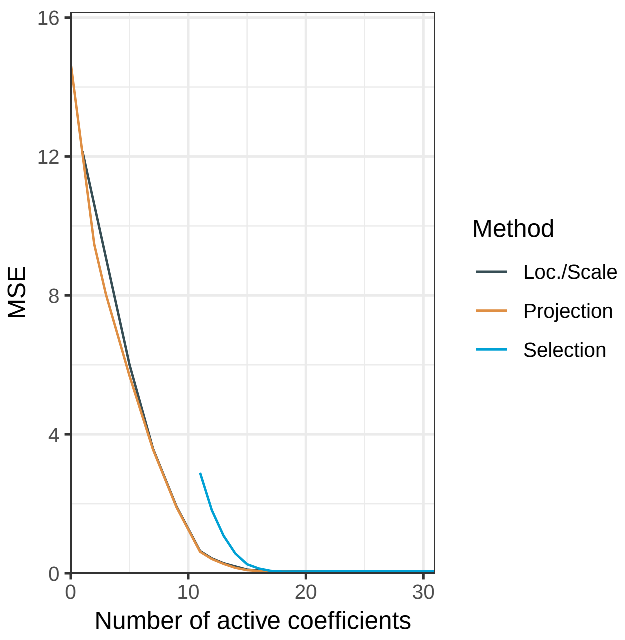


Figure 2: 2-Wasserstein distance between coarse and full posterior predictive means and MSE between posterior predictive means and the true means (calculated from the true parameters)