Latent Variable Multivariate Mixed-type Response Regression

Karl Oskar Ekvall

9/21/2020

Installation

The package can be installed from GitHub, using devtools.

```
# Currently private repository
# devtools::install_github("koekvall/lvmmrPQL")
```

Notation

The matrix of responses, Y, has n rows and r columns. The matrix of predictors, X, has nr rows and p columns; the first r rows of X are the design matrix for the r responses in the first row of Y, the next r rows of X are the design matrix for the second row of Y, and so on. Thus, \texttt{matrix(X %*% Beta, nrow = n, ncol = r, byrow = TRUE)} gives an $n \times r$ matrix whose ith row is the mean of the ith latent vector.

Example with normal responses

```
set.seed(4)
n <- 500
type <- rep(1, 5) # Only normal responses
r <- length(type)

# Each observation has its own intercept
X <- Matrix::kronecker(rep(1, n), diag(r))
Beta_true <- (1:r) / r

# Variance parameters, psi treated as known
Sigma_true <- 0.5^abs(outer(1:r, 1:r, FUN = "-"))
psi_true <- rep(0.5, r)

Y <- lvmmrPQL::generate_lvmmr(X = X, Beta = Beta_true, R = chol(Sigma_true),
                    type = type, psi = psi_true)
# No restrictions with normal responses
M <- matrix(NA, r, r)

# Compute MLEs
fit_MLE <- lm(Y ~ 1)
Beta_MLE <- c(coef(fit_MLE))
Sigma_MLE <- crossprod(residuals(fit_MLE)) / n - diag(psi_true)

# Does MLE exist? That is, is maximizer PD?
min(eigen(Sigma_MLE)$values)
```

```
## [1] 0.249554
```

```
# Skip W update; obj. fun, does not depend on W with mult. norm. resp.
# MLE of Beta does not depend on Sigma, so expect correct MLE for Beta
# regardless of whether algorithm finds MLE of Sigma.
fit <- lvmmrPQL::lvmmr_PQL(Y = Y, X = X, type = type, M = M,
                           relative = T,
                           quiet = c(F, T, T, T),
                           maxit = c(100, 100, 500, 0),
                           tol = c(1e-12, 1e-8, 1e-12, 1e-8),
                           psi = psi_true)
```

```
## Change in parameters:  1153.165
## Change in parameters:  5.128422e-08
## Change in parameters:  4.94989e-08
## Change in parameters:  9.554761e-09
## Change in parameters:  4.744042e-08
## Change in parameters:  2.081875e-14
```

```
# Difference to MLEs
fit$Beta - Beta_MLE
```

```
##
##  1.110223e-16  2.942091e-15 -4.218847e-15  1.554312e-15 -1.110223e-14
```

```
fit$Sigma - Sigma_MLE
```

```
##               [,1]          [,2]          [,3]          [,4]          [,5]
## [1,] -1.220655e-06  1.433181e-06  1.792936e-06 -5.529616e-08 -9.912511e-07
## [2,]  1.433181e-06  7.382859e-08  5.867502e-07  1.154297e-06 -3.624168e-07
## [3,]  1.792936e-06  5.867502e-07 -3.937836e-07  2.548942e-07  1.845730e-06
## [4,] -5.529616e-08  1.154297e-06  2.548942e-07  2.304977e-07  1.848131e-06
## [5,] -9.912511e-07 -3.624168e-07  1.845730e-06  1.848131e-06 -9.591571e-07
```

```
# With MLE as starting value
fit <- lvmmrPQL::lvmmr_PQL(Y = Y, X = X, type = type, M = M,
                           relative = T,
                           quiet = c(F, T, T, T),
                           maxit = c(100, 100, 500, 0),
                           tol = c(1e-12, 1e-8, 1e-12, 1e-8),
                           Beta = Beta_MLE,
                           Sigma = Sigma_MLE,
                           psi = psi_true)
```

```
## Change in parameters:  2.304933e-14
```

```
fit$iter
```

```
## [1] 1
```

```
# Difference to MLEs
fit$Beta - Beta_MLE
```

```
##
## -2.164935e-15 -1.004752e-14 -1.632028e-14 -2.409184e-14 -2.020606e-14
```

```
fit$Sigma - Sigma_MLE
```

```
##               [,1]          [,2]          [,3]          [,4]          [,5]
## [1,]  1.554312e-15  1.110223e-16 -1.665335e-15  2.220446e-16 -1.387779e-16
## [2,]  1.110223e-16 -4.551914e-15  4.218847e-15 -3.275158e-15  1.110223e-16
## [3,] -1.665335e-15  4.218847e-15 -9.547918e-15  6.328271e-15 -1.942890e-15
## [4,]  1.665335e-16 -3.219647e-15  6.328271e-15 -9.769963e-15  1.221245e-15
## [5,] -2.220446e-16  1.110223e-16 -2.053913e-15  1.221245e-15 -2.220446e-15
```

```r
# See that objective is correct
D1 <- t(lvmmrPQL:::get_cumulant_diffs(t(fit$W), type, 1))
D2 <- t(lvmmrPQL:::get_cumulant_diffs(t(fit$W), type, 2))

lvmmrPQL:::working_ll(Y = Y, X = X, Beta = fit$Beta, Sigma = fit$Sigma,
                      W = fit$W, psi = psi_true, D1 = D1, D2 = D2)
```

```
## [1] -3921.977
```

```r
lvmmrPQL:::working_ll(Y = Y, X = X, Beta = Beta_MLE, Sigma = Sigma_MLE,
                      W = fit$W, psi = psi_true, D1 = D1, D2 = D2)
```

```
## [1] -3921.977
```

```r
# Double check w. multivariate normal likelihood
Xb <- matrix(X %*% fit$Beta, nrow = n, ncol = r, byrow = T)
sum(mvtnorm::dmvnorm(x = Y - Xb, sigma = fit$Sigma + diag(psi_true), log = TRUE))
```

```
## [1] -3921.977
```

```r
sum(mvtnorm::dmvnorm(x = Y - predict(fit_MLE), sigma = Sigma_MLE + diag(psi_true), log = TRUE))
```

```
## [1] -3921.977
```

Example with mixed-type responses

```r
set.seed(4)
n <- 100
type <- c(1, 2, 3)
r <- length(type)

# Each observation has an intercept and one uniform predictor (SUR)
X <- as.matrix(Matrix::KhatriRao(matrix(runif(n * r, -1, 1), n, r),
                                 diag(1, r)))
X <- cbind(Matrix::kronecker(rep(1, n), diag(r)), X)

Beta_true <- c(1:(2 * r)) / (2 * r)

# Variance parameters, psi treated as known
Sigma_true <- matrix(0.9, r, r)
diag(Sigma_true) <- 1
Sigma_true <- 2 * Sigma_true
psi_true <- rep(1, r) # psi \neq 1 currently not supported Ber and Poi
psi_true[type == "1"] <- 1e-4

Y <- lvmmrPQL::generate_lvmmr(X = X, Beta = Beta_true, R = chol(Sigma_true),
                    type = type, psi = psi_true)
# No restrictions with normal and Poisson responses
M <- matrix(NA, r, r)
```

```r
diag(M)[type == 2] <- 1

fit <- lvmmrPQL::lvmmr_PQL(Y = Y, X = X, type = type, M = M,
                           relative = FALSE,
                           quiet = c(F, T, T, T),
                           maxit = c(50, 100, 500, 1000),
                           tol = c(1e-5, 1e-7, 1e-10, 1e-8),
                           psi = psi_true,
                           same = FALSE
                           )
```

```
## Change in parameters:  3.182361
## Change in parameters:  2.324259
## Change in parameters:  0.23449
## Change in parameters:  0.08154595
## Change in parameters:  0.05195326
## Change in parameters:  0.03076906
## Change in parameters:  0.01847363
## Change in parameters:  0.01085218
## Change in parameters:  0.006364319
## Change in parameters:  0.003711196
## Change in parameters:  0.00214906
## Change in parameters:  0.001236587
## Change in parameters:  0.0007094902
## Change in parameters:  0.0004060537
## Change in parameters:  0.0002319761
## Change in parameters:  0.0001322541
## Change in parameters:  7.532444e-05
## Change in parameters:  4.281531e-05
## Change in parameters:  2.434123e-05
## Change in parameters:  1.37993e-05
## Change in parameters:  7.845996e-06
```

```r
# Predict
n_pred <- 1e4

X_new <- as.matrix(Matrix::KhatriRao(matrix(runif(n_pred * r, -1, 1), n_pred, r),
                                     diag(1, r)))
X_new <- cbind(Matrix::kronecker(rep(1, n_pred), diag(r)), X_new)

Y_new <- lvmmrPQL::generate_lvmmr(X = X_new, Beta = Beta_true, R = chol(Sigma_true),
                    type = type, psi = psi_true)
Beta_GLM <- matrix(0, 2, r)
for(jj in 1:r){
   fam <- c("gaussian", "binomial", "poisson")[type[jj]]
   Beta_GLM[, jj] = coef(glm(Y[, jj] ~ 0 + X[seq(jj, nrow(X),  by = r),
                                               c(jj, r + jj)], family = fam))
}
Beta_GLM <- c(Beta_GLM)

Xb_GLM <- matrix(X_new %*% Beta_GLM, nrow = n_pred, ncol = r, byrow = T)
pred_GLM <- t(lvmmrPQL:::get_cumulant_diffs(t(Xb_GLM), type, 1))
```

```
# We win (sometimes and often small).
sqrt(colMeans((Y_new - lvmmrPQL::predict_lvmmr(X = X_new,
                                          Beta = fit$Beta,
                                          sigma = sqrt(diag(fit$Sigma)),
                                          type = type,
                                          num_nodes = 15))^2))
```

```
## [1]  1.4240197  0.4958211 17.9245047
```

```
sqrt(colMeans((Y_new - pred_GLM)^2))
```

```
## [1]  1.4275389  0.4965258 18.5522371
```