Latent Variable Multivariate Mixed-type Response Regression

Karl Oskar Ekvall

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 <- 100
type <- rep(1, 2) # 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] 0.6828132
```

```
# 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,
                           pgd = TRUE) # Fast and accurate for both pgd = T / F
```

```
## Change in parameters:  918.4455
## Change in parameters:  1.381737e-07
## Change in parameters:  3.424085e-12
## Change in parameters:  4.829968e-16
```

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

```
##
##  5.551115e-16 -1.110223e-15
```

```
fit$Sigma - Sigma_MLE
```

```
##               [,1]          [,2]
## [1,]  8.256303e-08 -6.656699e-09
## [2,] -6.656699e-09 -5.752276e-08
```

```
# 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),
                           pgd = FALSE,
                           Beta = Beta_MLE,
                           Sigma = Sigma_MLE,
                           psi = psi_true)
```

```
## Change in parameters:  1.207492e-16
```

```
fit$iter
```

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

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

```
##
## 1.110223e-16 0.000000e+00
```

```
fit$Sigma - Sigma_MLE
```

```
##               [,1]          [,2]
## [1,]  0.000000e+00 -1.110223e-16
## [2,] -1.110223e-16  0.000000e+00
```

```
# 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_rcpp(Y_T = t(Y), X_T = t(X), beta = fit$Beta,
                                  Sigma = fit$Sigma, W_T = t(fit$W), psi = psi_true,
                                  D1_T = t(D1), D2_T = t(D2))
```

```
## [1] -313.1999
```

```
lvmmrPQL:::working_ll_rcpp(Y_T = t(Y), X_T = t(X), beta = Beta_MLE,
                                  Sigma = Sigma_MLE, W_T = t(fit$W), psi = psi_true,
                                  D1_T = t(D1), D2_T = t(D2))
```

```
## [1] -313.1999
```

```
# 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] -313.1999
```

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

```
## [1] -313.1999
```

Example with mixed-type responses

```
set.seed(4)
n <- 1000
type <- c(1, 1, 2, 2, 3, 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
psi_true <- rep(1, r)
psi_true[type == "2"] <- 1 # Bernoulli does not suppose psi

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



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

```
## Change in parameters:  4.377447
## Change in parameters:  3.771977
## Change in parameters:  0.1269841
## Change in parameters:  0.06062333
## Change in parameters:  0.006096987
## Change in parameters:  0.001164503
## Change in parameters:  0.000301941
## Change in parameters:  1.260033e-05
## Change in parameters:  8.549044e-06
```

```
# Use starting values
fit_pgd <- lvmmrPQL::lvmmr_PQL(Y = Y, X = X, type = type, M = M,
                           relative = FALSE,
                           quiet = c(F, T, T, T),
                           maxit = c(50, 100, 500, 100),
                           tol = c(1e-5, 1e-7, 1e-10, 1e-8),
                           psi = psi_true,
                           pgd = TRUE,
                           Beta = fit_trust$Beta,
                           Sigma = fit_trust$Sigma,
                           W = fit_trust$W)
```

```
## Change in parameters:  0.04735362
## Change in parameters:  0.001653916
## Change in parameters:  0.0005094483
## Change in parameters:  2.213558e-05
## Change in parameters:  6.681294e-05
## Change in parameters:  5.035126e-06
```

```
# 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).
RMSE <- rbind(sqrt(colMeans((Y_new - lvmmrPQL::predict_lvmmr(X = X_new,
                                          Beta = fit_trust$Beta,
                                          sigma = sqrt(diag(fit_trust$Sigma)),
                                          type = type,
                                          num_nodes = 15))^2)),
      sqrt(colMeans((Y_new - lvmmrPQL::predict_lvmmr(X = X_new,
                                          Beta = fit_pgd$Beta,
                                          sigma = sqrt(diag(fit_pgd$Sigma)),
                                          type = type,
                                          num_nodes = 15))^2)),
sqrt(colMeans((Y_new - pred_GLM)^2)))
rownames(RMSE) <- c("Trust region", "PGD", "GLM")
RMSE
```

```
##                  [,1]     [,2]      [,3]      [,4]     [,5]     [,6]
## Trust region 1.413051 1.410204 0.4906523 0.4870166 4.574856 4.930073
## PGD          1.413054 1.410197 0.4906320 0.4870176 4.573034 4.928758
## GLM          1.425876 1.447995 0.4920662 0.4942357 4.832948 5.061954
```