Surrogate Outcome Regression Analysis

Zachary R. McCaw

2022-08-05

Contents

Setting

For each of n independent subjects, suppose two continuous outcomes are potentially observed. Let T_i denote the *target* outcome, and let S_i denote the *surrogate* outcome. Group the target and surrogate outcomes into a bivariate outcome vector $Y_i = (T_i, S_i)'$. For each subject, either the target or the surrogate is potentially missing. Suppose the target mean depends on a vector of covariates x_i, and the surrogate mean depends on a vector of covariates z_i:

$$\mu_{T,i} = \mathbb{E}(T_i|x_i) = x_i'\beta$$
$$\mu_{S,i} = \mathbb{E}(S_i|z_i) = z_i'\alpha$$

Let $\mu_i = (\mu_{T,i}, \mu_{S,i})'$ denote the mean vector. Consider the bivariate normal regression model:

$$\begin{pmatrix} T_i \\ S_i \end{pmatrix} \Big| (x_i, z_i) \sim N\left\{ \begin{pmatrix} x_i'\beta \\ z_i'\alpha \end{pmatrix}, \begin{pmatrix} \Sigma_{TT} & \Sigma_{TS} \\ \Sigma_{ST} & \Sigma_{SS} \end{pmatrix} \right\}$$

This package provides methods for estimation of the model parameters (β, α, Σ), and for inference on components of the target regression parameters β. In the case of bilateral (target, surrogate) missingness, estimation is performed via an expectation maximization (EM) procedure. In the case of unilateral target missingness, estimation is performed via an accelerated, generalized least squares (GLS) procedure.

Example Data

Below, data are simulated for $n = 10^3$ subjects. The target `X` and surrogate `Z` design matrices each contain an intercept and three standard normal covariates. The regression coefficient for the target outcome is $\beta = (-1, 0.1, -0.1, 0)$. The regression coefficient for the surrogate outcome is $\alpha = (1, -0.1, 0.1, 0)$. The target and surrogate outcome each have unit variance $\Sigma_{TT} = \Sigma_{SS} = 1$. The target-surrogate covariance, equivalently the correlation, is $\Sigma_{TS} = \Sigma_{ST} = 0.5$. An outcome matrix for which 10% of the target outcomes and 20% of the surrogate outcomes are missing completely at random is simulated using `rBNR`.

```r
library(SurrogateRegression)
set.seed(100)

# Observations.
n <- 1e3

# Target design.
X <- cbind(1, matrix(rnorm(3 * n), nrow = n))

# Surrogate design.
Z <- cbind(1, matrix(rnorm(3 * n), nrow = n))

# Target parameter.
b <- c(-1, 0.1, -0.1, 0)

# Surrogate parameter.
a <- c(1, -0.1, 0.1, 0)

# Covariance matrix.
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)

# Generate data.
Y <- rBNR(X, Z, b, a, t_miss = 0.1, s_miss = 0.2, sigma = sigma);
t <- Y[, 1]
s <- Y[, 2]
```

Formatting Assumptions

The target and surrogate outcome vectors (`t`, `s`) both have length n. The unobserved values of the target or surrogate outcome are set to `NA`. The target `X` and surrogate `Z` model matrices are numeric, with all factors and interactions expanded. The model matrices contain no missing values.

Estimation

Estimation of the bivariate normal regression model is performed using `Fit.BNR`. If the surrogate outcome vector `s` contains missing values, or if the surrogate design matrix `Z` differs from the target design matrix `X`, then the EM algorithm is applied. Otherwise, estimation is performed via GLS, which is significantly faster.

```r
# Fit bivariate normal regression model.
fit <- FitBNR(
  t = t,
  s = s,
  X = X,
  Z = Z
)
show(fit)
```

```
## Objective increment:  0.761
## Objective increment:  0.00146
## Objective increment:  8.23e-05
## Objective increment:  5.28e-06
## Objective increment:  3.81e-07
```

```
## 4 update(s) performed before tolerance limit.
##
##     Outcome Coefficient    Point     SE        L         U         p
## 1    Target          x1 -1.01000 0.0327 -1.08000 -0.951000 4.49e-212
## 2    Target          x2  0.08040 0.0286  0.02430  0.136000  4.96e-03
## 3    Target          x3 -0.12700 0.0305 -0.18700 -0.067300  3.15e-05
## 4    Target          x4 -0.05620 0.0283 -0.11200 -0.000717  4.71e-02
## 5 Surrogate          z1  0.95800 0.0345  0.89100  1.030000 4.38e-170
## 6 Surrogate          z2 -0.12000 0.0332 -0.18500 -0.055400  2.84e-04
## 7 Surrogate          z3  0.06120 0.0323 -0.00215  0.125000  5.83e-02
## 8 Surrogate          z4  0.00226 0.0323 -0.06100  0.065500  9.44e-01
##
##         Covariance Point     SE     L     U
## 1           Target 0.981 0.0461 0.909 1.060
## 2 Target-Surrogate 0.475 0.0387 0.436 0.514
## 3        Surrogate 0.995 0.0495 0.920 1.080
```

The output is an object of class `bnr` with these slots:

- `@Covariance` containing the target-surrogate covariance matrix.

```
round(fit@Covariance, digits = 3)
```

```
##           Target Surrogate
## Target     0.981     0.475
## Surrogate  0.475     0.995
```

- `@Covariance.info` containing the information matrix for $(\Sigma_{TT}, \Sigma_{TS}, \Sigma_{SS})$.

```
round(fit@Covariance.info, digits = 3)
```

```
##                     Target-Target Target-Surrogate Surrogate-Surrogate
## Target-Target             719.391         -587.791             140.337
## Target-Surrogate         -587.791         1494.113            -579.423
## Surrogate-Surrogate       140.337         -579.423             648.569
```

- `@Covariance.tab` containing the estimated covariance parameters in tabular format.

```
fit@Covariance.tab
```

```
##         Covariance     Point         SE         L         U
## 1           Target 0.9810061 0.04612446 0.9091096 1.0585884
## 2 Target-Surrogate 0.4752001 0.03874548 0.4364546 0.5139456
## 3        Surrogate 0.9951732 0.04949877 0.9195614 1.0770021
```

- `@Regression.info` containing the information matrix for (β, α).

```
round(fit@Regression.info, digits = 3)
```

```
##          x1       x2       x3       x4       z1      z2      z3      z4
## x1 1132.136   33.834   -1.348   -5.086 -443.251  -0.294  -5.118 -23.263
## x2   33.834 1227.144   28.815  -64.616  -10.920  11.653  11.865 -18.835
## x3   -1.348   28.815 1083.294 -104.610   -3.128  37.159   4.925  23.347
## x4   -5.086  -64.616 -104.610 1264.986  -10.382   1.933 -47.296 -15.857
## z1 -443.251  -10.920   -3.128  -10.382 1015.535 -15.388  -0.737  22.757
## z2   -0.294   11.653   37.159    1.933  -15.388 912.145 -14.971  37.458
## z3   -5.118   11.865    4.925  -47.296   -0.737 -14.971 961.074  49.477
## z4  -23.263  -18.835   23.347  -15.857   22.757  37.458  49.477 967.085
```

- `@Regression.tab` containing the estimated regression parameters in tabular format.

```
fit@Regression.tab
```

```
##     Outcome Coefficient       Point         SE            L            U
## 1    Target          x1 -1.01497945 0.03265703 -1.078986058 -0.950972845
## 2    Target          x2  0.08038492 0.02861041  0.024309552  0.136460295
## 3    Target          x3 -0.12712578 0.03053943 -0.186981954 -0.067269600
## 4    Target          x4 -0.05617237 0.02829387 -0.111627342 -0.000717395
## 5 Surrogate          z1  0.95840033 0.03447531  0.890829960  1.025970698
## 6 Surrogate          z2 -0.12041438 0.03317267 -0.185431617 -0.055397150
## 7 Surrogate          z3  0.06122701 0.03233566 -0.002149727  0.124603746
## 8 Surrogate          z4  0.00226177 0.03225221 -0.060951395  0.065474934
##               p
## 1 4.492902e-212
## 2  4.959709e-03
## 3  3.145379e-05
## 4  4.710898e-02
## 5 4.384665e-170
## 6  2.835007e-04
## 7  5.829375e-02
## 8  9.440921e-01
```

- `@Residuals` containing the target and surrogate residuals.

```
round(head(fit@Residuals), digits = 3)
```

```
##   Target Surrogate
## 1 -0.363        NA
## 2  0.063     0.622
## 3 -0.537    -0.816
## 4 -1.117    -1.129
## 5  0.503    -0.042
## 6     NA     0.764
```

Inference

Wald and Score tests on β are specified using a logical vector `is_zero`, with length equal to the number of columns in the target model matrix `X`, and indicating which regression coefficients are zero under the *null hypothesis*. At least one element of `is_zero` must be `TRUE` (i.e. a test must be specified) and at least one element of `is_zero` must be `FALSE` (i.e. a null model must be estimable).

Below, various hypotheses are tested on the example data. The first is an overall test of $H_0 : \beta_1 = \beta_2 = \beta_3 = 0$, which is false. The second assesses $H_0 : \beta_1 = \beta_2 = 0$, which is again false, leaving β_3 unconstrained. The final considers $H_0 : \beta_3 = 0$, which is true, leaving β_1 and β_2 unconstrained. All models include an intercept β_0 under the null.

```
cat("Joint score test of b1 = b2 = b3 = 0","\n")
test_spec <- c(FALSE, TRUE, TRUE, TRUE)
signif(TestBNR(t, s, X, Z, is_zero = test_spec, report = FALSE, test = "Wald"), digits = 2)
signif(TestBNR(t, s, X, Z, is_zero = test_spec, report = FALSE, test = "Score"), digits = 2)

cat("\n","Joint score test of b1 = b2 = 0, treating b3 as a nuisance","\n")
test_spec <- c(FALSE, TRUE, TRUE, FALSE)
signif(TestBNR(t, s, X, Z, is_zero = test_spec, report = FALSE, test = "Wald"), digits = 2)
signif(TestBNR(t, s, X, Z, is_zero = test_spec, report = FALSE, test = "Score"), digits = 2)
```

```r
cat("\n","Individual score test of b3 = 0, treating b2 and b3 as nuisances","\n")
test_spec <- c(FALSE, FALSE, FALSE, TRUE)
signif(TestBNR(t, s, X, Z, is_zero = test_spec, report = FALSE, test = "Wald"), digits = 2)
signif(TestBNR(t, s, X, Z, is_zero = test_spec, report = FALSE, test = "Score"), digits = 2)
```

```
## Joint score test of b1 = b2 = b3 = 0
##    Wald      df       p
## 2.8e+01 3.0e+00 3.8e-06
##   Score      df       p
## 2.7e+01 3.0e+00 6.2e-06
##
##  Joint score test of b1 = b2 = 0, treating b3 as a nuisance
##    Wald      df       p
## 2.5e+01 2.0e+00 4.2e-06
##   Score      df       p
## 2.4e+01 2.0e+00 6.2e-06
##
##  Individual score test of b3 = 0, treating b2 and b3 as nuisances
##  Wald    df     p
## 3.900 1.000 0.047
## Score    df     p
## 3.900 1.000 0.048
```