0.1 netls: Network Least Squares Regression for Continuous Proximity Matrix Dependent Variables

Use network least squares regression analysis to estimate the best linear predictor when the dependent variable is a continuously-valued proximity matrix (a.k.a. sociomatrices, adjacency matrices, or matrix representations of directed graphs).

Syntax

```
> z.out <- zelig(y ~  x1 + x2, model = "netls", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

Examples

1. Basic Example with First Differences

   Load sample data and format it for social networkx analysis:

   ```
   > data(sna.ex)
   ```

   Estimate model:

   ```
   > z.out <- zelig(Var1 ~ Var2 + Var3 + Var4, model = "netls", data = sna.ex)
   ```

   Summarize regression results:

   ```
   > summary(z.out)
   ```

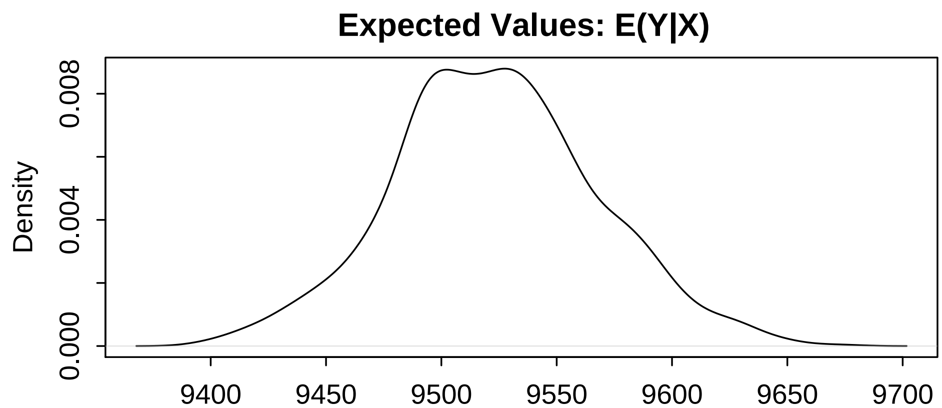
   Set explanatory variables to their default (mean/mode) values, with high (80th percentile) and low (20th percentile) for the second explanatory variable (Var3).

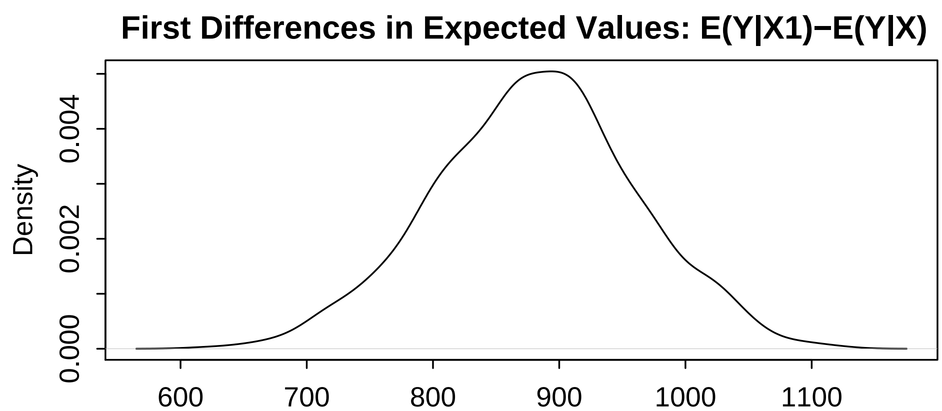
   ```
   > x.high <- setx(z.out, Var3 = quantile(sna.ex$Var3, 0.8))
   > x.low <- setx(z.out, Var3 = quantile(sna.ex$Var3, 0.2))
   ```

   Generate first differences for the effect of high versus low values of Var3 on the outcome variable.

   ```
   > try(s.out <- sim(z.out, x = x.high, x1 = x.low))
   > try(summary(s.out))

   > plot(s.out)
   ```





Model

The `netls` model performs a least squares regression of the sociomatrix $\mathbf{Y}$, a $m \times m$ matrix representing network ties, on a set of sociomatrices $\mathbf{X}$. This network regression model is a directly analogue to standard least squares regression element-wise on the appropriately vectorized matrices. Sociomatrices are vectorized by creating Y, an $m^2 \times 1$ vector to represent the sociomatrix. The vectorization which produces the Y vector from the $\mathbf{Y}$ matrix is preformed by simple row-concatenation of $\mathbf{Y}$. For example if $\mathbf{Y}$ is a 15×15 matrix, the $\mathbf{Y}_{1,1}$ element is the first element of Y, and the $\mathbf{Y}_{21}$ element is the second element of Y and so on. Once the input matrices are vectorized, standard least squares regression is performed. As such:

- The *stochastic component* is described by a density with mean μ_i and the common variance σ^2
$$Y_i \sim f(y_i|\mu_i, \sigma^2).$$

- The *systematic component* models the conditional mean as

$$\mu_i = x_i \beta$$

where x_i is the vector of covariates, and β is the vector of coefficients.

The least squares estimator is the best linear predictor of a dependent variable given x_i, and minimizes the sum of squared errors $\sum_{i=1}^{n}(Y_i - x_i\beta)^2$.

Quantities of Interest

The quantities of interest for the network least squares regression are the same as those for the standard least squares regression.

- The expected value (`qi$ev`) is the mean of simulations from the stochastic component,

$$E(Y) = x_i \beta,$$

given a draw of β from its sampling distribution.

- The first difference (`qi$fd`) is:

$$FD = E(Y|x_1) - E(Y|x)$$

Output Values

The output of each Zelig command contains useful information which you may view. For example, you run `z.out <- zelig(y   x, model="netls", data)`, then you may examine the available information in `z.out` by using `names(z.out)`, see the coefficients by using `z.out$coefficients`, and a default summary of information through `summary(z.out)`. Other elements available through the `$` operator are listed below.

- From the `zelig()` output stored in `z.out`, you may extract:
  - `coefficients`: parameter estimates for the explanatory variables.
  - `fitted.values`: the vector of fitted values for the explanatory variables.
  - `residuals`: the working residuals in the final iteration of the IWLS fit.
  - `df.residual`: the residual degrees of freedom.
  - `zelig.data`: the input data frame if `save.data = TRUE`
- From `summary(z.out)`, you may extract:
  - `mod.coefficients`: the parameter estimates with their associated standard errors, p-values, and t statistics.

$$\hat{\beta} = \left(\sum_{i=1}^{n} x_i' x_i \right)^{-1} \sum x_i y_i$$

- `sigma`: the square root of the estimate variance of the random error ε:

$$\hat{\sigma} = \frac{\sum (Y_i - x_i \hat{\beta})^2}{n - k}$$

- `r.squared`: the fraction of the variance explained by the model.

$$R^2 = 1 - \frac{\sum (Y_i - x_i \hat{\beta})^2}{\sum (y_i - \bar{y})^2}$$

- `adj.r.squared`: the above R^2 statistic, penalizing for an increased number of explanatory variables.
- `cov.unscaled`: a $k \times k$ matrix of unscaled covariances.

- From the `sim()` output stored in `s.out`, you may extract:
  - `qi$ev`: the simulated expected values for the specified values of `x`.
  - `qi$fd`: the simulated first differences (or differences in expected values) for the specified values of `x` and `x1`.

Contributors

The network least squares regression is part of the sna package by Carter T. Butts. Please cite the model as

.

In addition, advanced users may wish to refer to `help(netlm)`. Sample data are fictional. Skyler J. Cranmer added Zelig functionality.