0.1 normal.net: Network Normal Regression for Continuous Proximity Matrix Dependent Variables

The Network Normal regression model is a close variant of the more standard least squares regression model (see `netlm`). Both models specify a continuous proximity matrix (a.k.a. sociomatricies, adjacency matrices, or matrix representations of directed graphs) dependent variable as a linear function of a set of explanatory variables. The network Normal model reports maximum likelihood (rather than least squares) estimates. The two models differ only in their estimate for the stochastic parameter σ.

Syntax

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

Additional Inputs

In addition to the standard inputs, `zelig()` takes the following additional options for network normal regression:

- `LF`: specifies the link function to be used for the network normal regression. Default is `LF="identity"`, but `LF` can also be set to `"log"` or `"inverse"` by the user.

Examples

1. Basic Example

   Load the sample data (see `?friendship` for details on the structure of the network dataframe):

   ```
   > data(friendship)
   ```

   Estimate model:

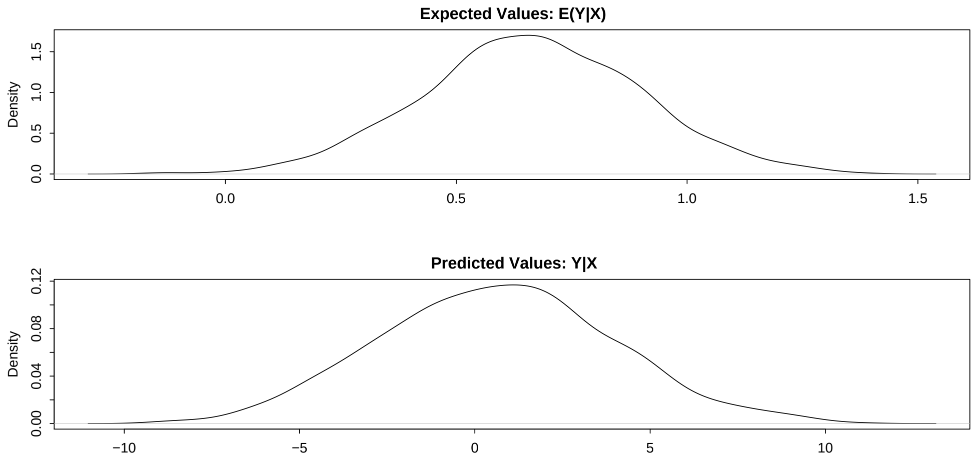
   ```
   > z.out <- zelig(perpower ~ friends + advice + prestige, model = "normal.net",
   +     data = friendship)
   > summary(z.out)
   ```

   Setting values for the explanatory variables to their default values:

   ```
   > x.out <- setx(z.out)
   ```

   Simulate fitted values.

```
> s.out <- sim(z.out, x = x.out)
> summary(s.out)
> plot(s.out)
```



Model

The normal.net model performs a Normal regression of the proximity matrix $\mathbf{Y}$, a $m \times m$ matrix representing network ties, on a set of proximity matrices $\mathbf{X}$. This network regression model is directly analogous to standard Normal regression element-wise on the appropriately vectorized matrices. Proximity matrices are vectorized by creating Y, a $m^2 \times 1$ vector to represent the proximity matrix. The vectorization which produces the Y vector from the $\mathbf{Y}$ matrix is performed 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 $\mathbf{Y}_{2,1}$ element is the second element of Y and so on. Once the input matrices are vectorized, standard Normal regression is performed.

Let Y_i be the continuous dependent variable, produced by vectorizing a continuous proximity matrix, for observation i.

- The *stochastic component* is described by a univariate normal model with a vector of means μ_i and scalar variance σ^2:

$$Y_i \sim \text{Normal}(\mu_i, \sigma^2).$$

- The *systematic component* is given by:

$$\mu_i = x_i \beta.$$

where x_i is the vector of k explanatory variables and β is the vector of coefficients.

Quantities of Interest

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

- The expected value (`qi$ev`) for the `normal.net` model is the mean of simulations from the stochastic component,
$$E(Y) = \mu_i = x_i\beta,$$
given a draw of β from its posterior.

- The predicted value (`qi$pr`) is a draw from the distribution defined by the set of parameters (μ_i, σ^2).

- The first difference (`qi$fd`) for the network Normal model is defined as
$$FD = \Pr(Y|x_1) - \Pr(Y|x)$$

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 = "normal.net", 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 systemic component λ.
  - `residuals`: the working residuals in the final iteration of the IWLS fit.
  - `linear.predictors`: fitted values. For the normal model, these are identical to fitted values.
  - `aic`: Akaike's Information Criterion (minus twice the maximized log-likelihood plus twice the number of coefficients).
  - `bic`: the Bayesian Information Criterion (minus twice the maximized log-likelihood plus the number of coefficients times log n).
  - `df.residual`: the residual degrees of freedom.
  - `df.null`: the residual degrees of freedom for the null model.
  - `zelig.data`: the input data frame if `save.data = TRUE`

- From `summary(z.out)`(as well as from `zelig()`), you may extract:
  - `mod.coefficients`: the parameter estimates with their associated standard errors, p-values, and t statistics.

  - `cov.scaled`: a $k \times k$ matrix of scaled covariances.
  - `cov.unscaled`: a $k \times k$ matrix of unscaled covariances.

- From the `sim()` output stored in `s.out`, you may extract:
  - `qi$ev`: the simulated expected probabilities for the specified values of `x`.
  - `qi$pr`: the simulated predicted values drawn from the distribution defined by (μ_i, σ^2).
  - `qi$fd`: the simulated first differences in the expected probabilities simulated from `x` and `x1`.

How to Cite

To cite the *normal.net* Zelig model:

> Skyler J. Cranmer. 2007. "normal.net: Network Normal Regression for Continuous Proximity Matrix Dependent Variables," in Kosuke Imai, Gary King, and Olivia Lau, "Zelig: Everyone's Statistical Software," `http://gking.harvard.edu/zelig`.

To cite Zelig as a whole, please reference these two sources:

> Kosuke Imai, Gary King, and Olivia Lau. 2007. "Zelig: Everyone's Statistical Software," `http://GKing.harvard.edu/zelig`.

> Kosuke Imai, Gary King, and Olivia Lau. 2007. "Toward A Common Framework for Statistical Analysis and Development," `http://gking.harvard.edu/files/abs/z-abs.shtml`.

See also

The network normal regression is part of the `netglm` package by Skyler J. Cranmer and is built using some of the functionality of the `sna` package by Carter T. Butts (Butts and Carley 2001).In addition, advanced users may wish to refer to `help(normal.net)`. Sample data are fictional.

Bibliography

Butts, C. and Carley, K. (2001), "Multivariate Methods for Interstructural Analysis," Tech. rep., CASOS working paper, Carnegie Mellon University.