1 `cloglog.net`: Network Complementary Log Log Regression for Dichotomous Proximity Matrix Dependent Variables

Use network complementary log log regression analysis for a dependent variable that is a binary valued proximity matrix (a.k.a. sociomatricies, adjacency matrices, or matrix representations of directed graphs).

1.1 Syntax

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

1.2 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(friends ~ advice + prestige + perpower, model = "cloglog.net", data =
   > summary(z.out)
   >
   ```

   Setting values for the explanatory variables to their default values:

   ```
   >   x.out <- setx(z.out)
   >
   ```
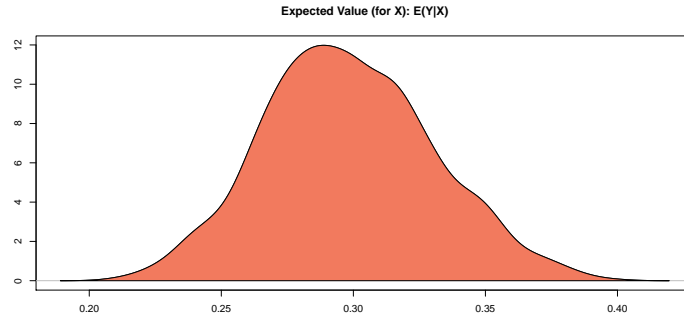
   Simulating quantities of interest from the posterior distribution.

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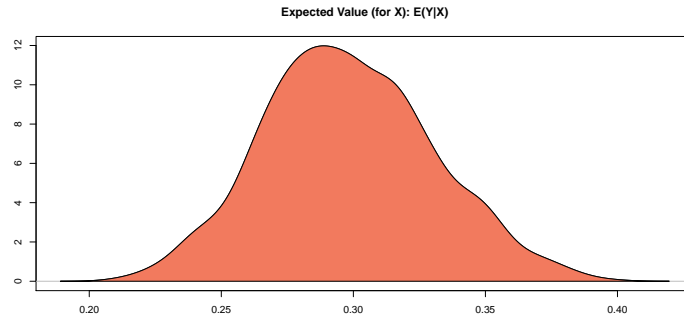
2. Simulating First Differences

   Estimating the risk difference (and risk ratio) between low personal power (25th percentile) and high personal power (75th percentile) while all the other variables are held at their default values.

Expected Value (for X): E(Y|X)

```
> x.high <- setx(z.out, perpower = quantile(friendship$perpower, prob = 0.75))
> x.low  <- setx(z.out, perpower = quantile(friendship$perpower, prob = 0.25))
> s.out2 <- sim(z.out, x = x.high, x1 = x.low)
> summary(s.out2)
> plot(s.out2)
>
```


Expected Value (for X): E(Y|X)

1.3   Model

The `cloglog.net` model performs a complementary log log 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 complementary log log 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 complementary log log regression is performed.

2

Let Y_i be the binary dependent variable, produced by vectorizing a binary proximity matrix, for observation i which takes the value of either 0 or 1.

- The *stochastic component* is given by

$$Y_i \sim \text{Bernoulli}(\pi_i)$$

  where $\pi_i = \Pr(Y_i = 1)$.

- The *systematic component* is given by:

$$\pi_i = 1 - \exp[\exp(-x_i\beta)]$$

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

1.4   Quantities of Interest

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

- The expected values (`qi$ev`) for the `cloglog.nett` model are simulations of the predicted probability of a success:

$$E(Y) = \pi_i = 1 - \exp[\exp(-x_i\beta)],$$

  given draws of β from its sampling distribution.

- The predicted values (`qi$pr`) are draws from the Binomial distribution with mean equal to the simulated expected value π_i.

- The first difference (`qi$fd`) for the network complementary log log model is defined as

$$FD = \Pr(Y = 1|x_1) - \Pr(Y = 1|x)$$

1.5   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 = "cloglog.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 explanatory variables.

- **residuals**: the working residuals in the final iteration of the IWLS fit.
- **linear.predictors**: the vector of $x_i\beta$.
- **aic**: Akaikeś 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$ev1**: the simulated expected probabilities for the specified values of `x`.
  - **qi$pr1**: the simulated predicted values for the specified values of `x`.
  - **qi$fd**: the simulated first differences in the expected probabilities simulated from `x` and `x1`.

How to Cite Network Log-Log Regeression

Olivia Lau Matt Owen, Kosuke Imai and Gary King. *cloglog.net: Network Complementary Log Log Regression for Dichotomous Proximity Matrix Dependent Variables*, 2011

How to Cite the Zelig Software Package

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`.

Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

See also

The network complementary log log 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 [1].In addition, advanced users may wish to refer to `help(netbinom)`. Sample data are fictional.

2 `gamma.net`: Network Gamma Regression for Continuous, Positive Proximity Matrix Dependent Variables

Use the network gamma regression model if you have a positive-valued dependent variable that is a binary valued proximity matrix (a.k.a. sociomatricies, adjacency matrices, or matrix representations of directed graphs). The gamma distribution assumes that all waiting times are complete by the end of the study (censoring is not allowed).

2.1 Syntax

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

2.2 Additional Inputs

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

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

2.3 Examples

1. Basic Example

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

   ```
   > data(friendship)
   >
   >
   >
   ```
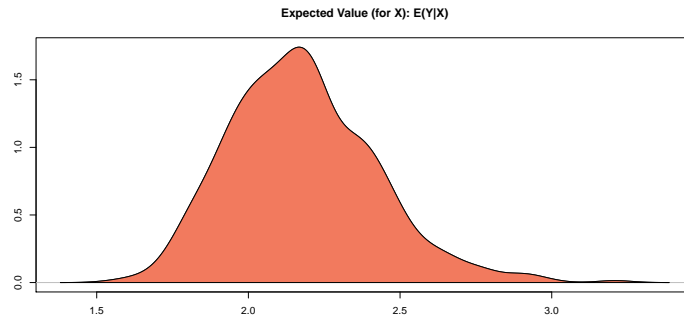
   Estimate model:

```
> z.out <- zelig(per ~ perpower, LF="inverse",  model="gamma.net", data=friendship)
> summary(z.out)
>
```

Setting values for the explanatory variables to their default values:

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

Simulating quantities of interest from the posterior distribution.

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


2. Simulating First Differences

```
> x.low <- setx(z.out, numst2 = 0)
> x.high <- setx(z.out, numst2 = 1)
> s.out2 <- sim(z.out, x = x.low, x1 = x.high)
> summary(s.out2)
> plot(s.out2)
>
```

2.4   Model

The `gamma.net` model performs a gamma 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 gamma 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 $\mathbf{Y}_{1,1}$

Expected Value (for X): E(Y|X)
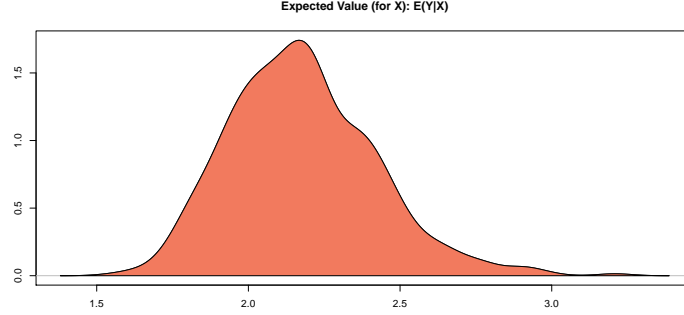
element is the first element of Y, and the $\mathbf{Y}_{2,1}$ element is the second element of Y and so on. Once the input matrices are vectorized, standard gamma regression is performed.

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

- The Gamma distribution with scale parameter α has a *stochastic component* given by

$$
\begin{aligned}
Y &\sim \text{Gamma}(y_i|\lambda_i, \alpha) \\
f(y) &= \frac{1}{\alpha^{\lambda_i}\Gamma\lambda_i} y_i^{\lambda_i - 1} \exp - \left[\frac{y_i}{\alpha}\right]
\end{aligned}
$$

for $\alpha, \lambda_i, y_i > 0$.

- The *systematic component* is given by:

$$
\lambda_i = \frac{1}{x_i\beta}.
$$

2.5 Quantities of Interest

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

- The expected values (`qi$ev`) are simulations of the mean of the stochastic component given draws of α and β from their posteriors:

$$
E(Y) = \alpha_i\lambda.
$$

- The predicted values (`qi$pr`) are draws from the gamma distribution for each set of parameters (α, λ_i).

- The first difference (`qi$fd`) for the network gamma model is defined as

$$
FD = \Pr(Y|x_1) - \Pr(Y|x)
$$

7

2.6 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 = "gamma.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 explanatory variables.
    - `residuals`: the working residuals in the final iteration of the IWLS fit.
    - `linear.predictors`: the vector of $x_i\beta$.
    - `aic`: Akaikeś 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 a distribution defined by (α_i, λ).
    - `qi$fd`: the simulated first differences in the expected probabilities simulated from `x` and `x1`.

How to Cite Network Gamma Regeression

Olivia Lau Matt Owen, Kosuke Imai and Gary King. *gamma.net: Network Gamma Regression for Continuous, Positive Proximity Matrix Dependent Variables*, 2011

How to Cite the Zelig Software Package

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`.

> Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

See also

The network gamma 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 [1].In addition, advanced users may wish to refer to `help(gamma.net)`. Sample data are fictional.

3 `logit.net`: Network Logistic Regression for Dichotomous Proximity Matrix Dependent Variables

Use network logistic regression analysis for a dependent variable that is a binary valued proximity matrix (a.k.a. sociomatricies, adjacency matrices, or matrix representations of directed graphs).

3.1 Syntax

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

3.2 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(friends ~ advice + prestige + perpower, model = "logit.net", data = fr
> summary(z.out)
>
```
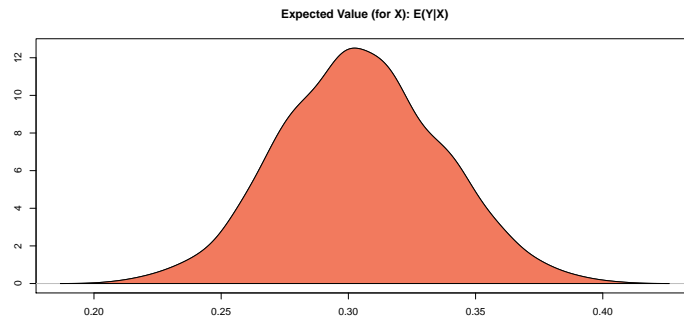
Setting values for the explanatory variables to their default values:

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

Simulating quantities of interest from the posterior distribution.

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

Expected Value (for X): E(Y|X)

2. Simulating First Differences

Estimating the risk difference (and risk ratio) between low personal power
(25th percentile) and high personal power (75th percentile) while all the
other variables are held at their default values.

```
> x.high <- setx(z.out, perpower = quantile(friendship$perpower, prob = 0.75))
> x.low  <- setx(z.out, perpower = quantile(friendship$perpower, prob = 0.25))
> s.out2 <- sim(z.out, x = x.high, x1 = x.low)
> summary(s.out2)
> plot(s.out2)
>
```

3.3   Model

The logit.net model performs a logistic 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 logistic regression
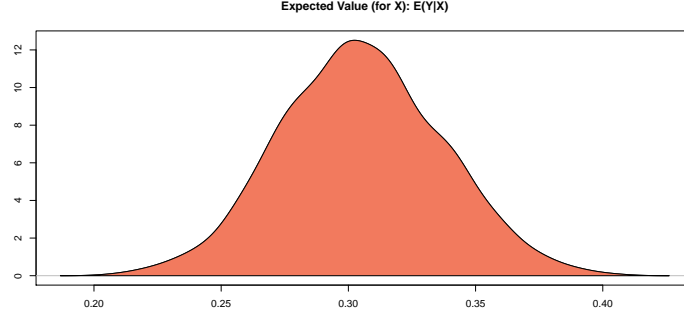
Expected Value (for X): E(Y|X)

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 logistic regression is performed.

Let Y_i be the binary dependent variable, produced by vectorizing a binary proximity matrix, for observation i which takes the value of either 0 or 1.

- The *stochastic component* is given by

$$
\begin{aligned}
Y_i & \sim \text{Bernoulli}(y_i | \pi_i) \\
& = \pi_i^{y_i} (1 - \pi_i)^{1 - y_i}
\end{aligned}
$$

where $\pi_i = \Pr(Y_i = 1)$.

- The *systematic component* is given by:

$$
\pi_i = \frac{1}{1 + \exp(-x_i \beta)}.
$$

where x_i is the vector of k covariates for observation i and β is the vector of coefficients.

3.4 Quantities of Interest

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

- The expected values (`qi$ev`) for the `logit.net` model are simulations of the predicted probability of a success:

$$
E(Y) = \pi_i = \frac{1}{1 + \exp(-x_i \beta)},
$$

given draws of β from its sampling distribution.

11

- The predicted values (`qi$pr`) are draws from the Binomial distribution with mean equal to the simulated expected value π_i.

- The first difference (`qi$fd`) for the network logit model is defined as

$$FD = \Pr(Y = 1|x_1) - \Pr(Y = 1|x)$$

3.5   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 = "logit.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 explanatory variables.
  - `residuals`: the working residuals in the final iteration of the IWLS fit.
  - `linear.predictors`: the vector of $x_i\beta$.
  - `aic`: Akaikeś 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 for the specified values of `x`.
  - `qi$fd`: the simulated first differences in the expected probabilities simulated from `x` and `x1`.

How to Cite Network Log-Log Regeression

Olivia Lau Matt Owen, Kosuke Imai and Gary King. *logit.net: Logistic Regression for Dichotomous Dependent Variables*, 2011

How to Cite the Zelig Software Package

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`.

> Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

See also

The network logistic 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 [1].In addition, advanced users may wish to refer to `help(netgamma)`. Sample data are fictional.

4 `ls.net`: 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).

4.1 Syntax

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

4.2 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 = "ls.net", 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)
```
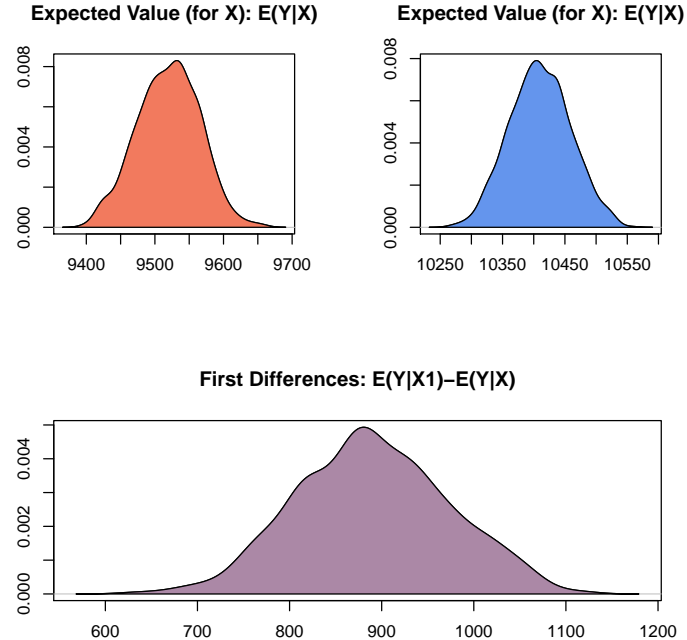
4.3 Model

The `ls.net` 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 $\sum_{i=1}^{n}(Y_i - x_i\beta)^2$.

4.4 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:
$$FD = E(Y|x_1) - E(Y|x)$$

4.5 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="ls.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 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`.

How to Cite

How to Cite Network Linear Model

Matt Owen, Kosuke Imai, Olivia Lau, and Gary King. *ls.net: Network Least Squares Regression for Continuous Proximity Matrix Dependent Variables*, 2011

How to Cite the Zelig Software Package

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`.

> Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

See also

The network least squares regression is part of the sna package by Carter T. Butts [1].In addition, advanced users may wish to refer to `help(netlm)`.

5   `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.

5.1   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)
```

5.2   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.

5.3   Examples

1. Basic Example

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

   ```
   > data(friendship)
   >
   >
   >
   ```
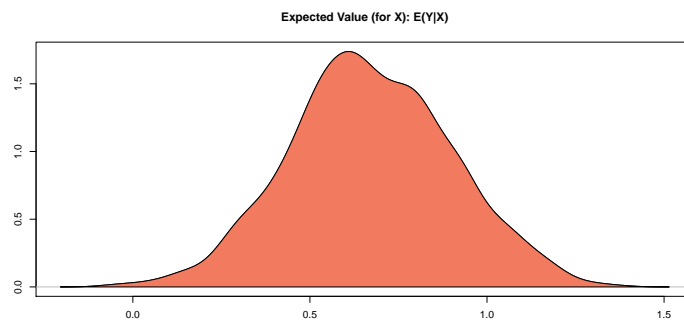
   Estimate model:

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


Expected Value (for X): E(Y|X)

5.4   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.

5.5 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

$$FD = \Pr(Y|x_1) - \Pr(Y|x)$$

5.6 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 Network Normal Regression

Matt Owen, Kosuke Imai, Olivia Lau, and Gary King. *normal.net: Network Normal Regression for Continuous Proximity Matrix Dependent Variables*, 2011

How to Cite the Zelig Software Package

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`.

Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

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 [1].In addition, advanced users may wish to refer to `help(normal.net)`. Sample data are fictional.

6 `poisson.net`: Network Poisson Regression for Count Proximity Matrix Dependent Variables

Use the ordinal logit regression model if your dependent variable is ordered and categorical, either in the form of integer values or character strings.

6.1 Syntax

```
> z.out <- zelig(as.factor(Y) ~ X1 + X2, model = "poisson.net", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

If `Y` takes discrete integer values, the `as.factor()` command will order automatically order the values. If `Y` takes on values composed of character strings, such as "strongly agree", "agree", and "disagree", `as.factor()` will order the values in the order in which they appear in `Y`. You will need to replace your dependent variable with a factored variable prior to estimating the model through `zelig()`. See Section **??** for more information on creating ordered factors and Example 1 below.

6.2 Example

1. Creating An Ordered Dependent Variable

   Load the sample data:

   ```
   > data(sanction)
   ```

   Create an ordered dependent variable:

   ```
   > sanction$ncost <- factor(sanction$ncost, ordered = TRUE,
   +                          levels = c("net gain", "little effect",
   +                          "modest loss", "major loss"))
   ```

   Estimate the model:

```
>    # z.out <- zelig(ncost ~ mil + coop, model = "poisson.net", data = sanction)
```

Set the explanatory variables to their observed values:

```
>    # x.out <- setx(z.out, fn = NULL)
```

Simulate fitted values given `x.out` and view the results:

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

>    # summary(s.out)
```

2. First Differences

Using the sample data `sanction`, estimate the empirical model and re-turning the coefficients:

```
>    # z.out <- zelig(as.factor(cost) ~ mil + coop, model = "ologit",
>    #                    data = sanction)

>    # summary(z.out)
```

Set the explanatory variables to their means, with `mil` set to 0 (no military action in addition to sanctions) in the baseline case and set to 1 (military action in addition to sanctions) in the alternative case:

```
>    # x.low <- setx(z.out, mil = 0)
>    # x.high <- setx(z.out, mil = 1)
```

Generate simulated fitted values and first differences, and view the results:

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

6.3   Model

Let Y_i be the ordered categorical dependent variable for observation i that takes one of the integer values from 1 to J where J is the total number of categories.

- The *stochastic component* begins with an unobserved continuous variable, Y_i^*, which follows the standard logistic distribution with a parameter μ_i,

$$Y_i^* \sim \text{Logit}(y_i^* \mid \mu_i),$$

to which we add an observation mechanism

$$Y_i = j \quad \text{if} \quad \tau_{j-1} \leq Y_i^* \leq \tau_j \quad \text{for} \quad j = 1, \ldots, J.$$

where τ_l (for $l = 0, \ldots, J$) are the threshold parameters with $\tau_l < \tau_m$ for all $l < m$ and $\tau_0 = -\infty$ and $\tau_J = \infty$.

- The *systematic component* has the following form, given the parameters τ_j and β, and the explanatory variables x_i:

$$\Pr(Y \leq j) \;=\; \Pr(Y^* \leq \tau_j) \;=\; \frac{\exp(\tau_j - x_i\beta)}{1 + \exp(\tau_j - x_i\beta)},$$

which implies:

$$\pi_j \;=\; \frac{\exp(\tau_j - x_i\beta)}{1 + \exp(\tau_j - x_i\beta)} - \frac{\exp(\tau_{j-1} - x_i\beta)}{1 + \exp(\tau_{j-1} - x_i\beta)}.$$

6.4 Quantities of Interest

- The expected values (`qi$ev`) for the ordinal logit model are simulations of the predicted probabilities for each category:

$$E(Y = j) \;=\; \pi_j \;=\; \frac{\exp(\tau_j - x_i\beta)}{1 + \exp(\tau_j - x_i\beta)} - \frac{\exp(\tau_{j-1} - x_i\beta)}{1 + \exp(\tau_{j-1} - x_i\beta)},$$

given a draw of β from its sampling distribution.

- The predicted value (`qi$pr`) is drawn from the logit distribution described by μ_i, and observed as one of J discrete outcomes.

- The difference in each of the predicted probabilities (`qi$fd`) is given by

$$\Pr(Y = j \mid x_1) \;-\; \Pr(Y = j \mid x) \quad \text{for} \quad j = 1, \ldots, J.$$

- In conditional prediction models, the average expected treatment effect (`att.ev`) for the treatment group is

$$\frac{1}{n_j} \sum_{i:t_i=1}^{n_j} \left\{ Y_i(t_i = 1) - E[Y_i(t_i = 0)] \right\},$$

where t_i is a binary explanatory variable defining the treatment ($t_i = 1$) and control ($t_i = 0$) groups, and n_j is the number of treated observations in category j.

- In conditional prediction models, the average predicted treatment effect (`att.pr`) for the treatment group is

$$\frac{1}{n_j} \sum_{i:t_i=1}^{n_j} \left\{ Y_i(t_i = 1) - Y_i(\widehat{t_i = 0}) \right\},$$

where t_i is a binary explanatory variable defining the treatment ($t_i = 1$) and control ($t_i = 0$) groups, and n_j is the number of treated observations in category j.

6.5 Output Values

The output of each Zelig command contains useful information which you may view. For example, if you run `z.out <- zelig(y ~ x, model = "ologit", 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 object `z.out`, you may extract:

  - `coefficients`: parameter estimates for the explanatory variables.
  - `zeta`: a vector containing the estimated class boundaries τ_j.
  - `deviance`: the residual deviance.
  - `fitted.values`: the $n \times J$ matrix of in-sample fitted values.
  - `df.residual`: the residual degrees of freedom.
  - `edf`: the effective degrees of freedom.
  - `Hessian`: the Hessian matrix.
  - `zelig.data`: the input data frame if `save.data = TRUE`.

- From `summary(z.out)`, you may extract:

  - `coefficients`: the parameter estimates with their associated standard errors, and t-statistics.

- From the `sim()` output object `s.out`, you may extract quantities of interest arranged as arrays. Available quantities are:

  - `qi$ev`: the simulated expected probabilities for the specified values of `x`, indexed by simulation $\times$ quantity $\times$ `x`-observation (for more than one `x`-observation).
  - `qi$pr`: the simulated predicted values drawn from the distribution defined by the expected probabilities, indexed by simulation $\times$ `x`-observation.
  - `qi$fd`: the simulated first difference in the predicted probabilities for the values specified in `x` and `x1`, indexed by simulation $\times$ quantity $\times$ `x`-observation (for more than one `x`-observation).
  - `qi$att.ev`: the simulated average expected treatment effect for the treated from conditional prediction models.
  - `qi$att.pr`: the simulated average predicted treatment effect for the treated from conditional prediction models.

How to Cite Network Poisson Regression

Matt Owen, Kosuke Imai, Olivia Lau, and Gary King. *poisson.net: Network Poisson Regression for Count Proximity Matrix Dependent Variables*, 2011

How to Cite the Zelig Software Package

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`.

> Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

See also

The network poisson 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 [1].In addition, advanced users may wish to refer to `help(poisson.net)`. Sample data are fictional.

7 `probit.net`: Network Probit Regression for Dichotomous Proximity Matrix Dependent Variables

Use network probit regression analysis for a dependent variable that is a binary valued proximity matrix (a.k.a. sociomatricies, adjacency matrices, or matrix representations of directed graphs).

7.1 Syntax

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

7.2 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(friends ~ advice + prestige + perpower, model = "probit.net", data = f
> summary(z.out)
>
```
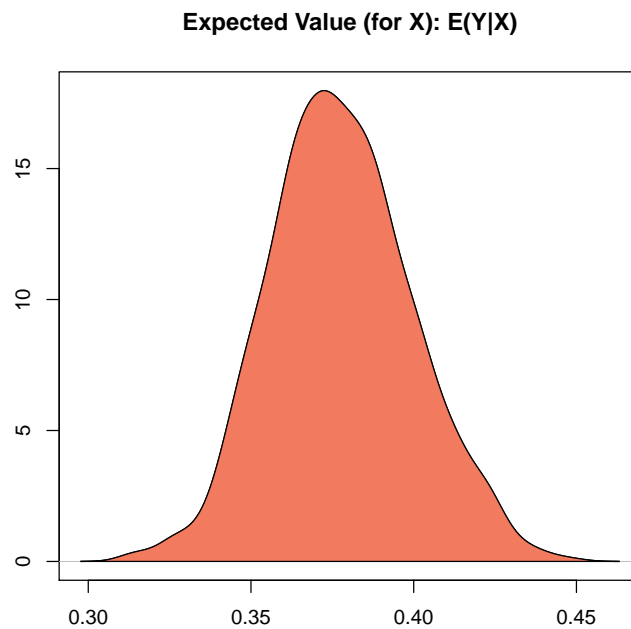
Setting values for the explanatory variables to their default values:

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

Simulating quantities of interest from the posterior distribution.

```
> s.out <- sim(z.out, x = x.out)
> summary(s.out)
> plot(s.out)
>
```
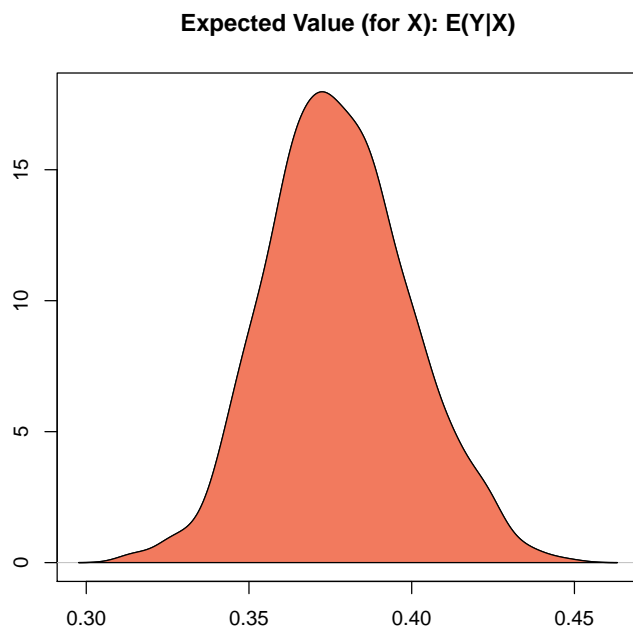
Expected Value (for X): E(Y|X)

2. Simulating First Differences

   Estimating the risk difference (and risk ratio) between low personal power
   (25th percentile) and high personal power (75th percentile) while all the
   other variables are held at their default values.

```
> x.high <- setx(z.out, perpower = quantile(friendship$perpower, prob = 0.75))
> x.low  <- setx(z.out, perpower = quantile(friendship$perpower, prob = 0.25))
> s.out2 <- sim(z.out, x = x.high, x1 = x.low)
> summary(s.out2)
> plot(s.out2)
>
```

Expected Value (for X): E(Y|X)

7.3   Model

The `probit.net` model performs a probit 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 probit 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 probit regression is performed.

Let Y_i be the binary dependent variable, produced by vectorizing a binary proximity matrix, for observation i which takes the value of either 0 or 1.

- The *stochastic component* is given by

$$Y_i \sim \text{Bernoulli}(\pi_i)$$

where $\pi_i = \Pr(Y_i = 1)$.

- The *systematic component* is given by:

$$\pi_i = \mathbf{\Phi}(x_i\beta).$$

where $\mathbf{\Phi}(\mu)$ is the cumulative distribution function of the Normal distribution with mean 0 and unit variance.

7.4 Quantities of Interest

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

- The expected values (`qi$ev`) for the `probit.net` model are simulations of the predicted probability of a success:

$$E(Y) = \pi_i = \mathbf{\Phi}(x_i\beta),$$

given draws of β from its sampling distribution.

- The predicted values (`qi$pr`) are draws from the Binomial distribution with mean equal to the simulated expected value π_i.

- The first difference (`qi$fd`) for the network probit model is defined as

$$FD = \Pr(Y = 1|x_1) - \Pr(Y = 1|x)$$

7.5 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 = "probit.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 explanatory variables.
    - `residuals`: the working residuals in the final iteration of the IWLS fit.
    - `linear.predictors`: the vector of $x_i\beta$.

- **aic**: Akaikeś 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 for the specified values of `x`.
  - **qi$fd**: the simulated first differences in the expected probabilities simulated from `x` and `x1`.

How to Cite Network Poisson Regression

Matt Owen, Kosuke Imai, Olivia Lau, and Gary King. *poisson.net: Network Poisson Regression for Count Proximity Matrix Dependent Variables*, 2011

How to Cite the Zelig Software Package

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`.

> Imai, Kosuke, Gary King, and Olivia Lau. (2008). "Toward A Common Framework for Statistical Analysis and Development." Journal of Computational and Graphical Statistics, Vol. 17, No. 4 (December), pp. 892-913.

See also

The network probit 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 [1].In addition, advanced users may wish to refer to `help(netpoisson)`. Sample data are fictional.

References

[1] C.T. Butts and K.M. Carley. Multivariate methods for interstructural analysis. Technical report, CASOS working paper, Carnegie Mellon University, 2001.

[2] Olivia Lau Matt Owen, Kosuke Imai and Gary King. *cloglog.net: Network Complementary Log Log Regression for Dichotomous Proximity Matrix Dependent Variables*, 2011.

[3] Olivia Lau Matt Owen, Kosuke Imai and Gary King. *gamma.net: Network Gamma Regression for Continuous, Positive Proximity Matrix Dependent Variables*, 2011.

[4] Olivia Lau Matt Owen, Kosuke Imai and Gary King. *logit.net: Logistic Regression for Dichotomous Dependent Variables*, 2011.

[5] Matt Owen, Kosuke Imai, Olivia Lau, and Gary King. *ls.net: Network Least Squares Regression for Continuous Proximity Matrix Dependent Variables*, 2011.

[6] Matt Owen, Kosuke Imai, Olivia Lau, and Gary King. *normal.net: Network Normal Regression for Continuous Proximity Matrix Dependent Variables*, 2011.

[7] Matt Owen, Kosuke Imai, Olivia Lau, and Gary King. *poisson.net: Network Poisson Regression for Count Proximity Matrix Dependent Variables*, 2011.