0.1 oprobit.bayes: Bayesian Ordered Probit Regression

Use the ordinal probit regression model if your dependent variables are ordered and categorical. They may take either integer values or character strings. The model is estimated using a Gibbs sampler with data augmentation. For a maximum-likelihood implementation of this models, see Section **??**.

Syntax

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

Additional Inputs

`zelig()` accepts the following arguments to monitor the Markov chain:

- `burnin`: number of the initial MCMC iterations to be discarded (defaults to 1,000).
- `mcmc`: number of the MCMC iterations after burnin (defaults 10,000).
- `thin`: thinning interval for the Markov chain. Only every `thin`-th draw from the Markov chain is kept. The value of `mcmc` must be divisible by this value. The default value is 1.
- `tune`: tuning parameter for the Metropolis-Hasting step. The default value is `NA` which corresponds to 0.05 divided by the number of categories in the response variable.
- `verbose`: defaults to `FALSE` If `TRUE`, the progress of the sampler (every 10%) is printed to the screen.
- `seed`: seed for the random number generator. The default is `NA` which corresponds to a random seed 12345.
- `beta.start`: starting values for the Markov chain, either a scalar or vector with length equal to the number of estimated coefficients. The default is `NA`, which uses the maximum likelihood estimates as the starting values.

Use the following parameters to specify the model's priors:

- `b0`: prior mean for the coefficients, either a numeric vector or a scalar. If a scalar value, that value will be the prior mean for all the coefficients. The default is 0.

- `B0`: prior precision parameter for the coefficients, either a square matrix (with dimensions equal to the number of coefficients) or a scalar. If a scalar value, that value times an identity matrix will be the prior precision parameter. The default is 0 which leads to an improper prior.

Zelig users may wish to refer to `help(MCMCoprobit)` for more information.

Convergence

Users should verify that the Markov Chain converges to its stationary distribution. After running the `zelig()` function but before performing `setx()`, users may conduct the following convergence diagnostics tests:

- `geweke.diag(z.out$coefficients)`: The Geweke diagnostic tests the null hypothesis that the Markov chain is in the stationary distribution and produces z-statistics for each estimated parameter.
- `heidel.diag(z.out$coefficients)`: The Heidelberger-Welch diagnostic first tests the null hypothesis that the Markov Chain is in the stationary distribution and produces p-values for each estimated parameter. Calling `heidel.diag()` also produces output that indicates whether the mean of a marginal posterior distribution can be estimated with sufficient precision, assuming that the Markov Chain is in the stationary distribution.
- `raftery.diag(z.out$coefficients)`: The Raftery diagnostic indicates how long the Markov Chain should run before considering draws from the marginal posterior distributions sufficiently representative of the stationary distribution.

If there is evidence of non-convergence, adjust the values for `burnin` and `mcmc` and rerun `zelig()`.

Advanced users may wish to refer to `help(geweke.diag)`, `help(heidel.diag)`, and `help(raftery.diag)` for more information about these diagnostics.

Examples

1. Basic Example
   Attaching the sample dataset:

   ```
   > data(sanction)
   ```

   Estimating ordered probit regression using `oprobit.bayes`:

   ```
   > z.out <- zelig(ncost ~ mil + coop, model = "oprobit.bayes", data = sanction,
   +     verbose = TRUE)
   ```

   Creating an ordered dependent variable:

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

Checking for convergence before summarizing the estimates:

```
> heidel.diag(z.out$coefficients)

> raftery.diag(z.out$coefficients)

> summary(z.out)
```

Setting values for the explanatory variables to their sample averages:

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

Simulating quantities of interest from the posterior distribution given: `x.out`.

```
> s.out1 <- sim(z.out, x = x.out)
> summary(s.out1)
```

2. Simulating First Differences
   Estimating the first difference (and risk ratio) in the probabilities of incurring different level of cost when there is no military action versus military action while all the other variables held at their default values.

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

> s.out2 <- sim(z.out, x = x.high, x1 = x.low)
> summary(s.out2)
```

Model

Let Y_i be the ordered categorical dependent variable for observation i which takes an integer value $j = 1, \ldots, J$.

- The *stochastic component* is described by an unobserved continuous variable, Y_i^*,

$$Y_i^* \sim \text{Normal}(\mu_i, 1).$$

  Instead of Y_i^*, we observe categorical variable Y_i,

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

  where τ_j for $j = 0, \ldots, J$ are the threshold parameters with the following constraints, $\tau_l < \tau_m$ for $l < m$, and $\tau_0 = -\infty, \tau_J = \infty$.

The probability of observing Y_i equal to category j is,

$$\Pr(Y_i = j) \quad = \quad \Phi(\tau_j \mid \mu_i) - \Phi(\tau_{j-1} \mid \mu_i) \text{ for } j = 1, \ldots, J$$

where $\Phi(\cdot \mid \mu_i)$ is the cumulative distribution function of the Normal distribution with mean μ_i and variance 1.

- The *systematic component* is given by

$$\mu_i = x_i \beta,$$

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

- The *prior* for β is given by

$$\beta \sim \text{Normal}_k \left(b_0, B_0^{-1} \right)$$

where b_0 is the vector of means for the k explanatory variables and B_0 is the $k \times k$ precision matrix (the inverse of a variance-covariance matrix).

Quantities of Interest

- The expected values (`qi$ev`) for the ordered probit model are the predicted probability of belonging to each category:

$$\Pr(Y_i = j) = \Phi(\tau_j \mid x_i \beta) - \Phi(\tau_{j-1} \mid x_i \beta),$$

given the posterior draws of β and threshold parameters τ from the MCMC iterations.

- The predicted values (`qi$pr`) are the observed values of Y_i given the observation scheme and the posterior draws of β and cut points τ from the MCMC iterations.

- The first difference (`qi$fd`) in category j for the ordered probit model is defined as

$$\text{FD}_j = \Pr(Y_i = j \mid X_1) - \Pr(Y_i = j \mid X).$$

- The risk ratio (`qi$rr`) in category j is defined as

$$\text{RR}_j = \Pr(Y_i = j \mid X_1) \ / \ \Pr(Y_i = j \mid X).$$

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

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

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 observations in the treatment group that belong to category j.

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

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

  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 observations in the treatment group that belong to category j.

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 = "oprobit.bayes", data)
```

then you may examine the available information in `z.out` by using `names(z.out)`, see the draws from the posterior distribution of the `coefficients` by using `z.out$coefficients`, and view a default summary of information through `summary(z.out)`. Other elements available through the `$` operator are listed below.

- From the `zelig()` output object `z.out`, you may extract:
  - `coefficients`: draws from the posterior distributions of the estimated coefficients β and threshold parameters τ. Note, element τ_1 is normalized to 0 and is not returned in the `coefficients` object.
  - `zelig.data`: the input data frame if `save.data = TRUE`.
  - `seed`: the random seed used in the model.
- From the `sim()` output object `s.out`:
  - `qi$ev`: the simulated expected values (probabilities) of each of the J categories for the specified values of `x`.
  - `qi$pr`: the simulated predicted values (observed values) for the specified values of `x`.
  - `qi$fd`: the simulated first difference in the expected values of each of the J categories for the values specified in `x` and `x1`.
  - `qi$rr`: the simulated risk ratio for the expected values of each of the J categories simulated from `x` and `x1`.
  - `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

To cite the *oprobit.bayes* Zelig model use:

> Ben Goodrich and Ying Lu. 2007. "oprobit.bayes: Bayesian Ordered Probit Regression," 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

Bayesian ordinal probit regression is part of the MCMCpack library by Andrew D. Martin and Kevin M. Quinn (Martin and Quinn 2005). The convergence diagnostics are part of the CODA library by Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines (Plummer et al. 2005).

Bibliography

Martin, A. D. and Quinn, K. M. (2005), *MCMCpack: Markov chain Monte Carlo (MCMC) Package*.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2005), *coda: Output analysis and diagnostics for MCMC*.