0.1 bprobit: Bivariate Logistic Regression for Two Dichotomous Dependent Variables

Use the bivariate probit regression model if you have two binaryrun dependent variables (Y_1, Y_2), and wish to model them jointly as a function of some explanatory variables. Each pair of dependent variables (Y_{i1}, Y_{i2}) has four potential outcomes, $(Y_{i1} = 1, Y_{i2} = 1)$, $(Y_{i1} = 1, Y_{i2} = 0)$, $(Y_{i1} = 0, Y_{i2} = 1)$, and $(Y_{i1} = 0, Y_{i2} = 0)$. The joint probability for each of these four outcomes is modeled with three systematic components: the marginal $\Pr(Y_{i1} = 1)$ and $\Pr(Y_{i2} = 1)$, and the correlation parameter ρ for the two marginal distributions. Each of these systematic components may be modeled as functions of (possibly different) sets of explanatory variables.

Syntax

```
> z.out <- zelig(list(mu1 = Y1 ~ X1 + X2,
                      mu2 = Y2 ~ X1 + X3,
                      rho = ~ 1),
                 model = "bprobit", data = mydata)
> x.out <- setx(z.out)
> s.out <- sim(z.out, x = x.out)
```

Input Values

In every bivariate probit specification, there are three equations which correspond to each dependent variable (Y_1, Y_2), and the correlation parameter ρ. Since the correlation parameter does not correspond to one of the dependent variables, the model estimates ρ as a constant by default. Hence, only two formulas (for μ_1 and μ_2) are required. If the explanatory variables for μ_1 and μ_2 are the same and effects are estimated separately for each parameter, you may use the following short hand:

```
> fml <- list(cbind(Y1, Y2) ~ X1 + X2)
```

which has the same meaning as:

```
> fml <- list(mu1 = Y1 ~ X1 + X2, mu2 = Y2 ~ X1 + X2, rho = ~1)
```

You may use the function `tag()` to constrain variables across equations. The `tag()` function takes a variable and a label for the effect parameter. Below, the constrained effect of `x3` in both equations is called the `age` parameter:

```
> fml <- list(mu1 = y1 ~ x1 + tag(x3, "age"), mu2 = y2 ~ x2 + tag(x3,
+     "age"))
```

You may also constrain different variables across different equations to have the same effect.

Examples

1. Basic Example

   Load the data and estimate the model:

```
> data(sanction)

> z.out1 <- zelig(cbind(import, export) ~ coop + cost + target,
+      model = "bprobit", data = sanction)
```

   By default, `zelig()` estimates two effect parameters for each explanatory variable in addition to the correlation coefficient; this formulation is parametrically independent (estimating unconstrained effects for each explanatory variable), but stochastically dependent because the models share a correlation parameter.

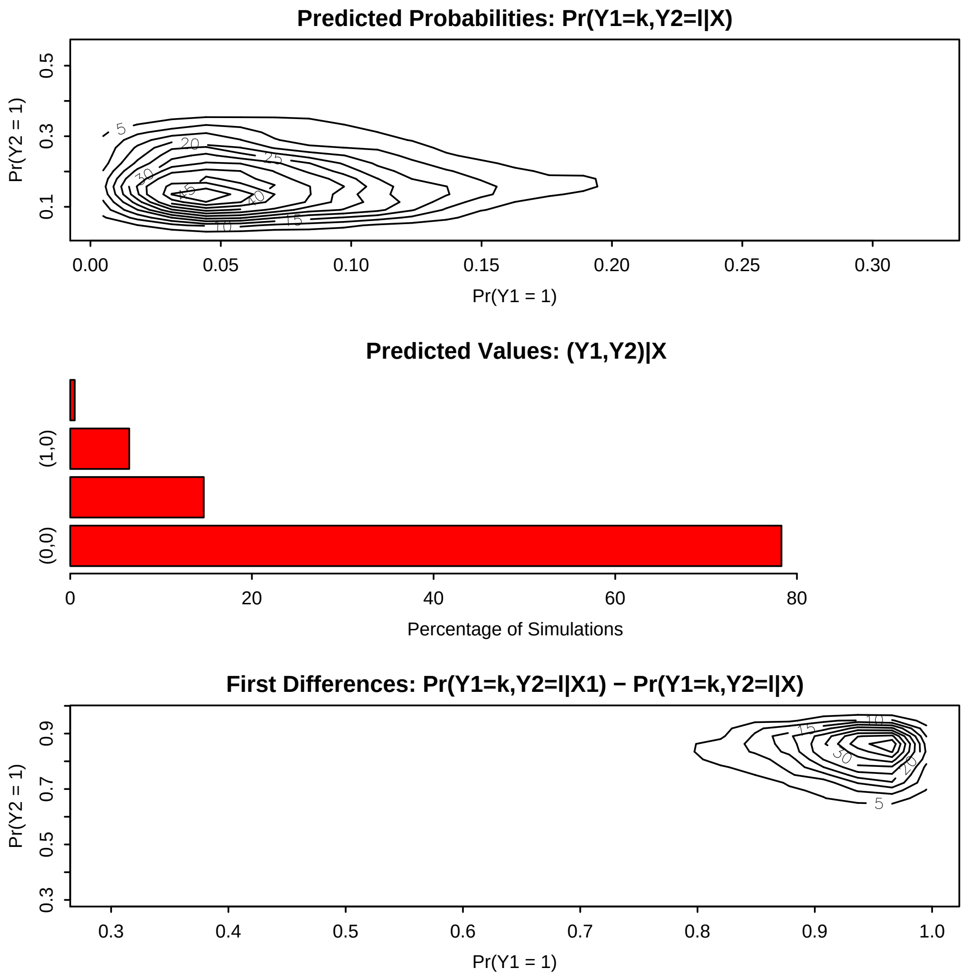
   Generate baseline values for the explanatory variables (with cost set to 1, net gain to sender) and alternative values (with cost set to 4, major loss to sender):

```
> x.low <- setx(z.out1, cost = 1)
> x.high <- setx(z.out1, cost = 4)
```

   Simulate fitted values and first differences:

```
> s.out1 <- sim(z.out1, x = x.low, x1 = x.high)
> summary(s.out1)

> plot(s.out1)
```



2. Joint Estimation of a Model with Different Sets of Explanatory Variables

   Using the sample data `sanction`, estimate the statistical model, with `import` a function of `coop` in the first equation and `export` a function of `cost` and `target` in the second equation:

```
> fml2 <- list(mu1 = import ~ coop, mu2 = export ~ cost + target)

> z.out2 <- zelig(fml2, model = "bprobit", data = sanction)
> summary(z.out2)
```

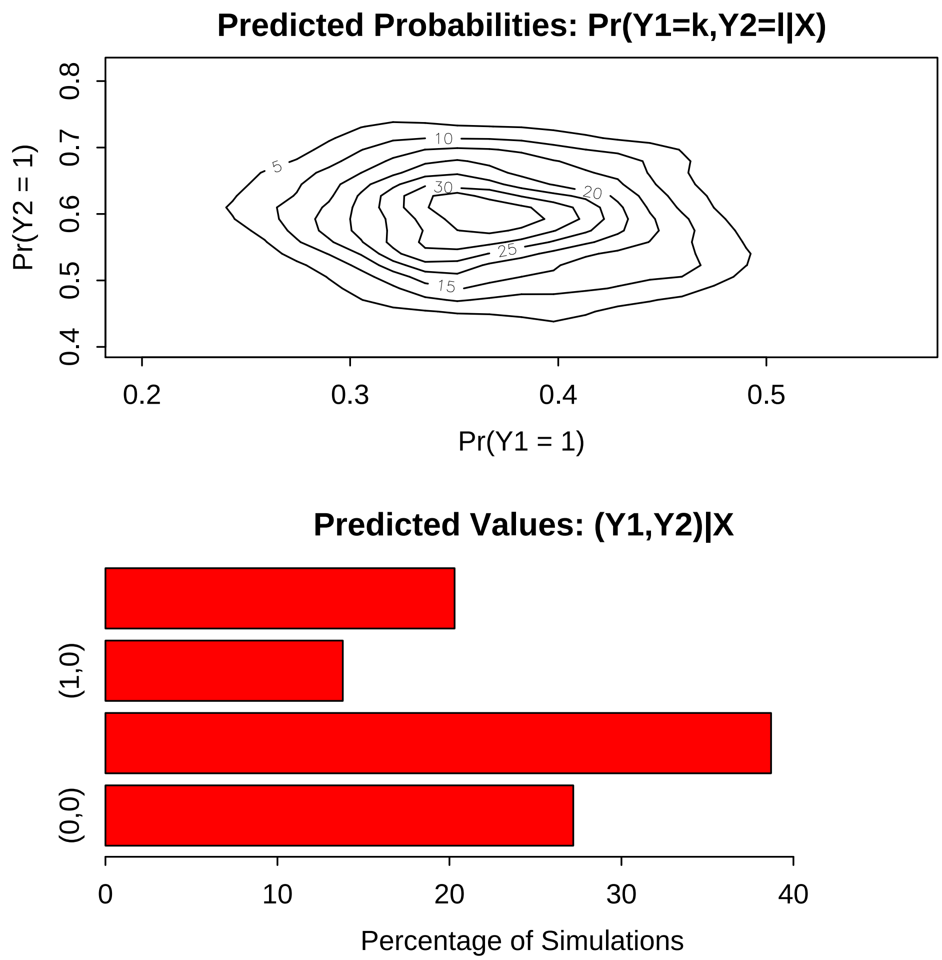
   Set the explanatory variables to their means:

```
> x.out2 <- setx(z.out2)
```

   Simulate draws from the posterior distribution:

```
> s.out2 <- sim(z.out2, x = x.out2)
> summary(s.out2)

> plot(s.out2)
```



3. Joint Estimation of a Parametrically and Stochastically Dependent Model

   Using the sample data `sanction`. The bivariate model is parametrically dependent if Y_1 and Y_2 share some or all explanatory variables, *and* the effects of the shared explanatory variables are jointly estimated. For example,

```
> fml3 <- list(mu1 = import ~ tag(coop, "coop") + tag(cost, "cost") +
+      tag(target, "target"), mu2 = export ~ tag(coop, "coop") +
+      tag(cost, "cost") + tag(target, "target"))

> z.out3 <- zelig(fml3, model = "bprobit", data = sanction)
> summary(z.out3)
```

Note that this model only returns one parameter estimate for each of `coop`, `cost`, and `target`. Contrast this to Example 1 which returns two parameter estimates for each of the explanatory variables.

Set values for the explanatory variables:

```
> x.out3 <- setx(z.out3, cost = 1:4)
```

Draw simulated expected values:

```
> s.out3 <- sim(z.out3, x = x.out3)
> summary(s.out3)
```

Model

For each observation, define two binary dependent variables, Y_1 and Y_2, each of which take the value of either 0 or 1 (in the following, we suppress the observation index i). We model the joint outcome (Y_1, Y_2) using two marginal probabilities for each dependent variable, and the correlation parameter, which describes how the two dependent variables are related.

- The *stochastic component* is described by two latent (unobserved) continuous variables which follow the bivariate Normal distribution:

$$\begin{pmatrix} Y_1^* \\ Y_2^* \end{pmatrix} \sim N_2 \left\{ \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right\},$$

  where μ_j is a mean for Y_j^* and ρ is a scalar correlation parameter. The following observation mechanism links the observed dependent variables, Y_j, with these latent variables

$$Y_j = \begin{cases} 1 & \text{if } Y_j^* \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

- The *systemic components* for each observation are

$$\begin{aligned} \mu_j &= x_j \beta_j \quad \text{for} \quad j = 1, 2, \\ \rho &= \frac{\exp(x_3 \beta_3) - 1}{\exp(x_3 \beta_3) + 1}. \end{aligned}$$

Quantities of Interest

For n simulations, expected values form an $n \times 4$ matrix.

- The expected values (`qi$ev`) for the binomial probit model are the predicted joint probabilities. Simulations of β_1, β_2, and β_3 (drawn form their sampling distributions) are substituted into the systematic components, to find simulations of the predicted joint probabilities $\pi_{rs} = \Pr(Y_1 = r, Y_2 = s)$:

$$\begin{aligned}
\pi_{11} &= \Pr(Y_1^* \geq 0, Y_2^* \geq 0) &= \int_0^\infty \int_0^\infty \phi_2(\mu_1, \mu_2, \rho)\, dY_2^*\, dY_1^* \\
\pi_{10} &= \Pr(Y_1^* \geq 0, Y_2^* < 0) &= \int_0^\infty \int_{-\infty}^0 \phi_2(\mu_1, \mu_2, \rho)\, dY_2^*\, dY_1^* \\
\pi_{01} &= \Pr(Y_1^* < 0, Y_2^* \geq 0) &= \int_{-\infty}^0 \int_0^\infty \phi_2(\mu_1, \mu_2, \rho)\, dY_2^*\, dY_1^* \\
\pi_{11} &= \Pr(Y_1^* < 0, Y_2^* < 0) &= \int_{-\infty}^0 \int_{-\infty}^0 \phi_2(\mu_1, \mu_2, \rho)\, dY_2^*\, dY_1^*
\end{aligned}$$

where r and s may take a value of either 0 or 1, ϕ_2 is the bivariate Normal density.

- The predicted values (`qi$pr`) are draws from the multinomial distribution given the expected joint probabilities.

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

$$\mathrm{FD}_{rs} = \Pr(Y_1 = r, Y_2 = s \mid x_1) - \Pr(Y_1 = r, Y_2 = s \mid x).$$

- The risk ratio (`qi$rr`) for each of the predicted joint probabilities are given by

$$\mathrm{RR}_{rs} = \frac{\Pr(Y_1 = r, Y_2 = s \mid x_1)}{\Pr(Y_1 = r, Y_2 = s \mid x)}.$$

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

$$\frac{1}{\sum_{i=1}^n t_i} \sum_{i:t_i=1}^n \{Y_{ij}(t_i = 1) - E[Y_{ij}(t_i = 0)]\} \text{ for } j = 1, 2,$$

where t_i is a binary explanatory variable defining the treatment ($t_i = 1$) and control ($t_i = 0$) groups. Variation in the simulations are due to uncertainty in simulating $E[Y_{ij}(t_i = 0)]$, the counterfactual expected value of Y_{ij} for observations in the treatment group, under the assumption that everything stays the same except that the treatment indicator is switched to $t_i = 0$.

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

$$\frac{1}{\sum_{i=1}^n t_i} \sum_{i:t_i=1}^n \left\{ Y_{ij}(t_i = 1) - \widehat{Y_{ij}(t_i = 0)} \right\} \text{ for } j = 1, 2,$$

where t_i is a binary explanatory variable defining the treatment ($t_i = 1$) and control ($t_i = 0$) groups. Variation in the simulations are due to uncertainty in simulating $\widehat{Y_{ij}(t_i = 0)}$, the counterfactual predicted value of Y_{ij} for observations in the treatment group, under the assumption that everything stays the same except that the treatment indicator is switched to $t_i = 0$.

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 = "bprobit", data)`, then you may examine the available information in `z.out` by using `names(z.out)`, see the `coefficients` by using `z.out$coefficients`, and obtain a default summary of information through `summary(z.out)`. Other elements available through the `$` operator are listed below.

- From the `zelig()` output object `z.out`, you may extract:
  - `coefficients`: the named vector of coefficients.
  - `fitted.values`: an $n \times 4$ matrix of the in-sample fitted values.
  - `predictors`: an $n \times 3$ matrix of the linear predictors $x_j \beta_j$.
  - `residuals`: an $n \times 3$ matrix of the residuals.
  - `df.residual`: the residual degrees of freedom.
  - `df.total`: the total degrees of freedom.
  - `rss`: the residual sum of squares.
  - `y`: an $n \times 2$ matrix of the dependent variables.
  - `zelig.data`: the input data frame if `save.data = TRUE`.
- From `summary(z.out)`, you may extract:
  - `coef3`: a table of the coefficients with their associated standard errors and t-statistics.
  - `cov.unscaled`: the variance-covariance matrix.
  - `pearson.resid`: an $n \times 3$ matrix of the Pearson residuals.
- From the `sim()` output object `s.out`, you may extract quantities of interest arranged as arrays indexed by simulation $\times$ quantity $\times$ `x`-observation (for more than one `x`-observation; otherwise the quantities are matrices). Available quantities are:
  - `qi$ev`: the simulated expected values (joint predicted probabilities) for the specified values of `x`.
  - `qi$pr`: the simulated predicted outcomes drawn from a distribution defined by the joint predicted probabilities.

– `qi$fd`: the simulated first difference in the predicted probabilities for the values specified in `x` and `x1`.
– `qi$rr`: the simulated risk ratio in the predicted probabilities for given `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 *bprobit* Zelig model use:

> Kosuke Imai,Gary King, and Olivia Lau. 2007. "bprobit: Bivariate Probit Regression for Two Dichotomous Dependent Variable," 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 bivariate probit function is part of the VGAM package by Thomas Yee (Yee and Hastie 2003). In addition, advanced users may wish to refer to `help(vglm)` in the VGAM library. Additional documentation is available at http://www.stat.auckland.ac.nz/˜ yee.Sample data are from Martin (1992)

Bibliography

Martin, L. (1992), *Coercive Cooperation: Explaining Multilateral Economic Sanctions*, Princeton University Press, please inquire with Lisa Martin before publishing results from these data, as this dataset includes errors that have since been corrected.

Yee, T. W. and Hastie, T. J. (2003), "Reduced-rank vector generalized linear models," *Statistical Modelling*, 3, 15–41.