0.1 irt1d: One Dimensional Item Response Model

Given several observed dependent variables and an unobserved explanatory variable, item response theory estimates the latent variable (ideal points). The model is estimated using the Markov Chain Monte Carlo algorithm via a Gibbs sampler and data augmentation. Use this model if you believe that the ideal points lie in one dimension, and see the k-dimensional item response model (Section **??**) for k hypothesized latent variables.

Syntax

```
> z.out <- zelig(cbind(Y1, Y2, Y3) ~ NULL, model = "irt1d", data = mydata)
```

Inputs

irt1d accepts the following argument:

- **Y1, Y2**, and **Y3**: Y1 contains the items for subject "Y1", Y2 contains the items for subject "Y2", and so on.

Additional arguments

irt1d accepts the following additional arguments for model specification:

- **theta.constraints**: a list specifying possible equality or inequality constraints on the ability parameters θ. A typical entry takes one of the following forms:

  - **varname = list()**: by default, no constraints are imposed.
  - **varname = c**: constrains the ability parameter for the subject named **varname** to be equal to **c**.
  - **varname = "+"**: constrains the ability parameter for the subject named **varname** to be positive.
  - **varname = "-"**: constrains the ability parameter for the subject named **varname** to be negative.

  The model also accepts the following arguments to monitor the sampling scheme for 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 to 20,000).

- **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.

- **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.

- **theta.start**: starting values for the subject abilities (ideal points), either a scalar or a vector with length equal to the number of subjects. If a scalar, that value will be the starting value for all subjects. The default is `NA`, which sets the starting values based on an eigenvalue-eigenvector decomposition of the agreement score matrix formed from the model response matrix (`cbind(Y1, Y2, ...)`).

- **alpha.start**: starting values for the difficulty parameters α, either a scalar or a vector with length equal to the number of the items. If a scalar, the value will be the starting value for all α. The default is `NA`, which sets the starting values based on a series of probit regressions that condition on `theta.start`.

- **beta.start**: starting values for the β discrimination parameters, either a scalar or a vector with length equal to the number of the items. If a scalar, the value will be the starting value for all β. The default is `NA`, which sets the starting values based on a series of probit regressions conditioning on `theta.start`.

- **store.item**: defaults to `TRUE`, storing the posterior draws of the item parameters. (For a large number of draws or a large number observations, this may take a lot of memory.)

- **drop.constant.items**: defaults to `TRUE`, dropping items with no variation before fitting the model.

`irt1d` accepts the following additional arguments to specify prior parameters used in the model:

- **t0**: prior mean of the subject abilities (ideal points). The default is 0.

- **T0**: prior precision of the subject abilities (ideal points). The default is 0.

- **ab0**: prior mean of (α, β). It can be a scalar or a vector of length 2. If it takes a scalar value, then the prior means for both α and β will be set to that value. The default is 0.

- **AB0**: prior precision of (α, β). It can be a scalar or a 2×2 matrix. If it takes a scalar value, then the prior precision will be `diag(AB0,2)`. The prior precision is assumed to be same for all the items. The default is 0.25.

Zelig users may wish to refer to `help(MCMCirt1d)` 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(SupremeCourt)
   > names(SupremeCourt) <- c("Rehnquist", "Stevens", "OConnor", "Scalia",
   +     "Kennedy", "Souter", "Thomas", "Ginsburg", "Breyer")
   ```

   Fitting a one-dimensional item response theory model using `irt1d`:

   ```
   > z.out <- zelig(cbind(Rehnquist, Stevens, OConnor, Scalia, Kennedy,
   +     Souter, Thomas, Ginsburg, Breyer) ~ NULL, data = SupremeCourt,
   +     model = "irt1d", B0.alpha = 0.2, B0.beta = 0.2, burnin = 500,
   +     mcmc = 10000, thin = 20, verbose = TRUE)
   ```

   Checking for convergence before summarizing the estimates:

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

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

> summary(z.out)
```

Model

Let Y_i be a vector of choices on J items made by subject i for $i = 1, \ldots, n$. The choice Y_{ij} is assumed to be determined by an unobserved utility Z_{ij}, which is a function of the subject i's abilities (ideal points) θ_i and item parameters α_j and β_j as follows:

$$Z_{ij} \;=\; -\alpha_j + \beta_j'\theta_i + \epsilon_{ij}.$$

- The *stochastic component* is given by

$$\begin{aligned} Y_{ij} &\sim& \text{Bernoulli}(\pi_{ij}) \\ &=& \pi_{ij}^{Y_{ij}}(1 - \pi_{ij})^{1-Y_{ij}}, \end{aligned}$$

where $\pi_{ij} = \Pr(Y_{ij} = 1) = E(Z_{ij})$.

The error term in the unobserved utility equation is independently and identically distributed with

$$\epsilon_{ij} \sim \text{Normal}(0, 1).$$

- The *systematic component* is given by

$$\pi_{ij} = \Phi(-\alpha_j + \beta_j'\theta_i),$$

where $\Phi(\cdot)$ is the cumulative density function of the standard normal distribution with mean 0 and variance 1, θ_i is the subject ability (ideal point) parameter, and α_j and β_j are the item parameters. Both subject abilities and item parameters are estimated from the model, such that the model is identified by placing constraints on the subject ability parameters.

- The *prior* for θ_i is given by

$$\theta_i \;\sim\; \text{Normal}\left(t_0, T_0^{-1}\right)$$

- The joint *prior* for α_j and β_j is given by

$$(\alpha_j, \beta_j)' \;\sim\; \text{Normal}\left(ab_0, AB_0^{-1}\right)$$

where ab_0 is a 2-vector of prior means and AB_0 is a 2×2 prior precision matrix.

Output Values

The output of each Zelig command contains useful information which you may view. For example, if you run:

```
z.out <- zelig(cbind(Y1, Y2, Y3) ~ NULL, model = "irt1d", 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 subject abilities(ideal points). If `store.item = TRUE`, the estimated item parameters α and β are also contained in `coefficients`.
  - `data`: the name of the input data frame.
  - `seed`: the random seed used in the model.

- Since there are no explanatory variables, the `sim()` procedure is not applicable for item response models.

How to Cite

To cite the *irt1d* Zelig model use:

> Ben Goodrich and Ying Lu. 2007. "irt1d: One Dimensional Item Response Mode," 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. 2008. "Toward A Common Framework for Statistical Analysis and Development," *Journal of Computational and Graphical Statistics*, forthcoming, `http://gking.harvard.edu/files/abs/z-abs.shtml`.

See also

The unidimensional item-response function 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). Sample data are adapted from Martin and Quinn (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.*