0.1 `factor.ord`: Ordinal Data Factor Analysis

Given some unobserved explanatory variables and observed ordinal dependent variables, this model estimates latent factors using a Gibbs sampler with data augmentation. For factor analysis for continuous data, see Section ??. For factor analysis for mixed data (including both continuous and ordinal variables), see Section ??.

Syntax

```
> z.out <- zelig(cbind(Y1 ,Y2, Y3) ~ NULL, factors = 1,
                 model = "factor.ord", data = mydata)
```

Inputs

`zelig()` accepts the following arguments for `factor.ord`: :

- `Y1, Y2`, and `Y3`: variables of interest in factor analysis (manifest variables), assumed to be ordinal variables. The number of manifest variables must be greater than the number of the factors.
- `factors`: number of the factors to be fitted (defaults to 1).

Additional Inputs

In addition, `zelig()` accepts the following arguments for model specification:

- `lambda.constraints`: list that contains the equality or inequality constraints on the factor loadings. A typical entry in the list has one of the following forms:
  - `varname = list()`: by default, no constraints are imposed.
  - `varname = list(d, c)`: constrains the dth loading for the variable named `varname` to be equal to `c`;
  - `varname = list(d, "+")`: constrains the dth loading for the variable named `varname` to be positive;
  - `varname = list(d, "-")`: constrains the dth loading for the variable named `varname` to be negative.

  The first column of Λ should not be constrained in general.
- `drop.constantvars`: defaults to `TRUE`, dropping the manifest variables that have no variation before fitting the model.

The model accepts the following arguments to monitor the convergence of 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.
- `tune`: tuning parameter for Metropolis-Hasting sampling, either a scalar or a vector of length K. The value of the tuning parameter must be positive. The default value is 1.2.
- `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.
- `Lambda.start`: starting values of the factor loading matrix Λ for the Markov chain, either a scalar (all unconstrained loadings are set to that value), or a matrix with compatible dimensions. The default is `NA`, such that the start values for the first column are set based on the observed pattern, while the remaining columns have start values set to 0 for unconstrained factor loadings, and -1 or 1 for constrained loadings (depending on the nature of the constraints).
- `store.lambda`: defaults to `TRUE`, which stores the posterior draws of the factor loadings.
- `store.scores`: defaults to `FALSE`. If `TRUE`, stores the posterior draws of the factor scores. (Storing factor scores may take large amount of memory for a a large number of draws or observations.)

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

- `l0`: mean of the Normal prior for the factor loadings, either a scalar or a matrix with the same dimensions as Λ. If a scalar value, that value will be the prior mean for all the factor loadings. Defaults to 0.
- `L0`: precision parameter of the Normal prior for the factor loadings, either a scalar or a matrix with the same dimensions as Λ. If `L0` takes a scalar value, then the precision matrix will be a diagonal matrix with the diagonal elements set to that value. The default value is 0, which leads to an improper prior.

Zelig users may wish to refer to `help(MCMCordfactanal)` 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(newpainters)
   ```

   Factor analysis for ordinal data using `factor.ord`:

   ```
   > z.out <- zelig(cbind(Composition, Drawing, Colour, Expression) ~
   +     NULL, data = newpainters, model = "factor.ord", factors = 1,
   +     L0 = 0.5, burin = 5000, mcmc = 30000, thin = 5, tune = 1.2,
   +     verbose = TRUE)
   ```

   Checking for convergence before summarizing the estimates:

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

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

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

   ```
   > summary(z.out)
   ```

Model

Let Y_i be a vector of K observed ordinal variables for observation i, each ordinal variable k for $k = 1, \ldots, K$ takes integer value $j = 1, \ldots, J_k$. The distribution of Y_i is assumed to be governed by another k-vector of unobserved continuous variable Y_i^*. There are d underlying factors.

- The *stochastic component* is described in terms of the latent variable Y_i^*:

$$Y_i^* \sim \text{Normal}_K(\mu_i, I_K),$$

  where $Y_i^* = (Y_{i1}^*, \ldots, Y_{iK}^*)$, and μ_i is the mean vector for Y_i^*, and $\mu_i = (\mu_{i1}, \ldots, \mu_{iK})$. Instead of Y_{ik}^*, we observe ordinal variable Y_{ik},

$$Y_{ik} = j \text{ if } \gamma_{(j-1),k} \leq Y_{ik}^* \leq \gamma_{jk} \text{ for } \quad j = 1, \ldots, J_k, k = 1, \ldots, K.$$

  where $\gamma_{jk}, j = 0, \ldots, J$ are the threshold parameters for the kth variable with the following constraints, $\gamma_{lk} < \gamma_{mk}$ for $l < m$, and $\gamma_{0k} = -\infty, \gamma_{J_k k} = \infty$ for any $k = 1, \ldots, K$. It follows that the probability of observing Y_{ik} belonging to category j is,

$$\Pr(Y_{ik} = j) = \Phi(\gamma_{jk} \mid \mu_{ik}) - \Phi(\gamma_{(j-1),k} \mid \mu_{ik}) \text{ for } j = 1, \ldots, J_k$$

  where $\Phi(\cdot \mid \mu_{ik})$ is the cumulative distribution function of the Normal distribution with mean μ_{ik} and variance 1.

- The *systematic component* is given by,

$$\mu_i = \Lambda \phi_i,$$

  where Λ is a $K \times d$ matrix of factor loadings for each variable, ϕ_i is a d-vector of factor scores for observation i. Note both Λ and ϕ need to be estimated.

- The independent conjugate *prior* for each element of Λ, Λ_{ij} is given by

$$\Lambda_{ij} \sim \text{Normal}(l_{0_{ij}}, L_{0_{ij}}^{-1}) \text{ for } i = 1, \ldots, k; \quad j = 1, \ldots, d.$$

- The *prior* for ϕ_i is,

$$\phi_{i(2:d)} \sim \text{Normal}(0, I_{d-1}), \text{ for } \quad i = 2, \ldots, n.$$

  where I_{d-1} is a $(d-1) \times (d-1)$ identity matrix. Note the first element of ϕ_i is 1.

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), model = "factor.ord", 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 factor loadings, the estimated cut points γ for each variable. Note the first element of γ is normalized to be 0. If `store.scores=TRUE`, the estimated factors scores are also contained in `coefficients`.
  - `zelig.data`: the input data frame if `save.data = TRUE`.
  - `seed`: the random seed used in the model.
- Since there are no explanatory variables, the `sim()` procedure is not applicable for factor analysis models.

How to Cite

To cite the *factor.ord* Zelig model:

> Ben Goodrich and Ying Lu. 2007. ”factor.ord: Ordinal Data Factor Analysis” 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`.
>
> 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.