0.1 `ei.dynamic`: Quinn's Dynamic Ecological Inference Model

Given contingency tables with observed marginals, ecological inference (EI) models estimate each internal cell value for each table. Quinn's dynamic EI model estimates a dynamic Bayesian model for 2×2 tables with temporal dependence across tables (units). The model is implemented using a Markov Chain Monte Carlo algorithm (via a combination of slice and Gibbs sampling). For a hierarchical Bayesian implementation of EI see Quinn's dynamic EI model (Section **??**). For contingency tables larger than 2 rows by 2 columns, see R×C EI (Section **??**).

Syntax

```
> z.out <- zelig(cbind(t0, t1) ~ x0 + x1, N = NULL,
                 model = "MCMCei.dynamic", data = mydata)
> x.out <- setx(z.out, fn = NULL, cond = TRUE)
> s.out <- sim(z.out, x = x.out)
```

Inputs

- `t0`, `t1`: numeric vectors (either counts or proportions) containing the column margins of the units to be analyzed.
- `x0`, `x1`: numeric vectors (either counts or proportions) containing the row margins of the units to be analyzed.
- `N`: total counts in each contingency table (unit). If `t0`,`t1`, `x0` and `x1` are proportions, you must specify `N`.

Additional Inputs

In addition, `zelig()` accepts the following additional inputs for `ei.dynamic` to monitor the convergence of the Markov chain:

- `burnin`: number of the initial MCMC iterations to be discarded (defaults to 5,000).
- `mcmc`: number of the MCMC iterations after burnin (defaults to 50,000).
- `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 of 12345.

The model also accepts the following additional arguments to specify priors and other parameters:

- `W`: a $p \times p$ numeric matrix describing the structure of the temporal dependence among elements of θ_0 and θ_1. The default value is 0, which constructs a weight matrix corresponding to random walk priors for θ_0 and θ_1 (assuming that the tables are equally spaced throughout time, and that the elements of `t0`, `t1`,`x0`,`x1` are temporally ordered).

- `a0`: $a_0/2$ is the shape parameter for the Inverse Gamma prior on σ_0^2. The default is 0.825.

- `b0`: $b_0/2$ is the scale parameter for the Inverse Gamma prior on σ_0^2. The default is 0.0105.

- `a1`: $a_1/2$ is the shape parameter for the Inverse Gamma prior on σ_1^2. The default is 0.825.

- `b1`: $b_1/2$ is the scale parameter for the Inverse Gamma prior on σ_1^2. The default is 0.0105.

Users may wish to refer to `help(MCMCdynamicEI)` for more options.

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 examples
   Attaching the example dataset:

   ```
   > data(eidat)
   ```

   Estimating the model using `ei.dynamic`:

   ```
   > z.out <- zelig(cbind(t0, t1) ~ x0 + x1, model = "ei.dynamic",
   +     data = eidat, mcmc = 40000, thin = 10, burnin = 10000, verbose = TRUE)
   > summary(z.out)
   ```

   Setting values for in-sample simulations given the marginal values of `t0`, `t1`, `x0`, and `x1`:

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

   In-sample simulations from the posterior distribution:

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

   Summarizing in-sample simulations at aggregate level weighted by the count in each unit:

   ```
   > summary(s.out)
   ```

   Summarizing in-sample simulations at unit level for the first 5 units:

   ```
   > summary(s.out, subset = 1:5)
   ```

Model

Consider the following 2×2 contingency table for the racial voting example. For each geographical unit $i = 1, \ldots, p$, the marginals t_i^0, t_i^1, x_i^0, and x_i^1 are known, and we would like to estimate n_i^{00}, n_i^{01}, n_i^{10}, and n_i^{11}.

	No Vote	Vote	
Black	n_i^{00}	n_i^{01}	x_i^0
White	n_i^{10}	n_i^{11}	x_i^1
	t_i^0	t_i^1	N_i

The marginal values x_i^0, x_i^1, t_i^0, t_i^1 are observed as either counts or fractions. If fractions, the counts can be obtained by multiplying by the total counts per table $N_i = n_i^{00}+n_i^{01}+n_i^{10}+n_i^{11}$, and rounding to the nearest integer. Although there are four internal cells, only two unknowns are modeled since $n_i^{01} = x_i^0 - n_i^{00}$ and $n_i^{11} = s_i^1 - n_i^{10}$.

The hierarchical Bayesian model for ecological inference in 2×2 is illustrated as following:

- The *stochastic component* of the model assumes that

$$\begin{aligned} n_i^{00} \mid x_i^0, \beta_i^b &\sim \text{Binomial}\left(x_i^0, \beta_i^b\right), \\ n_i^{10} \mid x_i^1, \beta_i^w &\sim \text{Binomial}\left(x_i^1, \beta_i^w\right) \end{aligned}$$

  where β_i^b is the fraction of the black voters who vote and β_i^w is the fraction of the white voters who vote. β_i^b and β_i^w as well as their aggregate summaries are the focus of inference.

- The *systematic component* of the model is

$$\begin{aligned} \beta_i^b &= \frac{\exp \theta_i^0}{1 - \exp \theta_i^0} \\ \beta_i^w &= \frac{\exp \theta_i^1}{1 - \exp \theta_i^1} \end{aligned}$$

  The logit transformations of β_i^b and β_i^w, θ_i^0, and θ_i^1 now take value on the real line. (Future versions may allow β_i^b and β_i^w to be functions of observed covariates.)

- The *priors* for θ_i^0 and θ_i^1 are given by

$$\begin{aligned} \theta_i^0 \mid \sigma_0^2 &\propto \frac{1}{\sigma_0^p} \exp\left(-\frac{1}{2\sigma_0^2}\theta_0' P \theta_0\right) \\ \theta_i^1 \mid \sigma_1^2 &\propto \frac{1}{\sigma_1^p} \exp\left(-\frac{1}{2\sigma_1^2}\theta_1' P \theta_1\right) \end{aligned}$$

where P is a $p \times p$ matrix whose off diagonal elements P_{ts} $(t \neq s)$ equal $-W_{ts}$ (the negative values of the corresponding elements of the weight matrix W), and diagonal elements $P_{tt} = \sum_{s \neq t} W_{ts}$. Scale parameters σ_0^2 and σ_1^2 have hyperprior distributions as given below.

- The *hyperpriors* for σ_0^2 and σ_1^2 are given by

$$\begin{aligned} \sigma_0^2 &\sim \text{Inverse Gamma}\left(\frac{a_0}{2}, \frac{b_0}{2}\right), \\ \sigma_1^2 &\sim \text{Inverse Gamma}\left(\frac{a_1}{2}, \frac{b_1}{2}\right), \end{aligned}$$

  where $a_0/2$ and $a_1/2$ are the shape parameters of the (independent) Gamma distributions while $b_0/2$ and $b_1/2$ are the scale parameters.

  The default hyperpriors for μ_0, μ_1, σ_0^2, and σ_1^2 are chosen such that the prior distributions for β^b and β^w are flat.

Output Values

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

```
> z.out <- (cbind(t0, t1) ~ x0 + x1, N = NULL,
          model = "ei.dynamic", data = mydata)
```

then you may examine the available information in `z.out` by using `names(z.out)`, see the draws from the posterior distribution of the quantities of interest 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 parameters.
  - `data`: the name of the input data frame.
  - `N`: the total counts when the inputs are fractions.
  - `seed`: the random seed used in the model.
- From `summary(z.out)`, you may extract:
  - `summary`: a matrix containing the summary information of the posterior estimation of β_i^b andβ_i^w for each unit and the parameters μ_0, μ_1, σ_1 and σ_2 based on the posterior distribution. The first p rows correspond to β_i^b, $i = 1, \ldots p$, the row

names are in the form of `p0table`i. The $(p+1)$-th to the $2p$-th rows correspond to β_i^w, $i = 1, \ldots, p$. The row names are in the form of `p1table`i. The last four rows contain information about μ_0, μ_1, σ_0^2 and σ_1^2, the prior means and variances of θ_0 and θ_1.

- From the `sim()` output object `s.out`, you may extract quantities of interest arranged as arrays indexed by simulation × column × row × observation, where column and row refer to the column dimension and the row dimension of the ecological table, respectively. In this model, only 2×2 contingency tables are analyzed, hence column= 2 and row= 2 in all cases. Available quantities are:
  - `qi$ev`: the simulated expected values of each internal cell given the observed marginals.
  - `qi$pr`: the simulated expected values of each internal cell given the observed marginals.

Contributors

The dynamic EI model was developed in

.

The function is part of the MCMCpack library by Andrew D. Martin and Kevin M. Quinn. If you use this model, please cite:

.

The convergence diagnostics are part of the CODA library by Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. These diagnostics should be cited as:

.

Sample data are adapted from

.

Ben Goodrich and Ying Lu enabled `ei.dynamic` to work with Zelig.