Python Code Documentation

Function

`EAS_VAR`

Description

Performs graph selection for a vector autoregression model of order 1. Graph selection is taken to mean selecting the active/inactive components of the transition matrix. See the reference for further details.

Usage

`EAS_VAR`($\mathcal{Y}$, $\mathcal{X}$, steps, burnin, p_o=None, N=None, weights=None)

Arguments

$\mathcal{Y}$:	A $p \times n$ NumPy array of time-series values (increasing in time from left to right over the columns).
$\mathcal{X}$:	A $p \times n$ NumPy array of lagged time-series values (increasing in time from left to right over the columns).
steps:	The number of MCMC steps.
burnin:	The number of initial *steps* to discard.
p_o:	Upper bound on the number of predictors in the true model. Default is $\min\{p^2, n\}$.
N:	The number of importance samples used to estimate $\mathrm{E}(h(\beta_M))$ within the pseudo-marginal MCMC. Default is 200.
weights:	The weights used for proposing which components to make active or inactive as the MCMC samples index sets $G \subset \{1, \ldots, p^2\}$. Default is to use squared coefficient estimates from elastic net, via the `ElasticNetCV` function from the 'sklearn.linear_model' Python module, added by one tenth of the minimum squared coefficient estimate from elastic net if the elastic net estimates a nonempty model, else the weights are uniform.

Values

chain:	A $(steps - burnin) \times p^2$ NumPy array containing the MCMC sample path (or trace) over index sets G, after *burnin* number of steps.

postSample: A NumPy array containing the indices (in each row) for every G visited in the MCMC sample path.

postProbs: A list containing the relative frequencies for which each of the index sets G in *postSample* was visited in the MCMC sample path.

AcceptRatio: The number of MCMC steps in which a proposed index set G was accepted.

d: The d parameter value used in the h-function (see reference).

References

J P Williams, Y Xie, and J Hannig (2019+). The EAS approach for graphical selection consistency in vector autoregression models. *Submitted*.