rSGDLM
An R Package for Simultaneous Graphical DLMs

Lutz F. Gruber

1 Loading the Package

The package is loaded into R by executing the command

```
library(rSGDLM)
```

This will automatically also load the *Rcpp* package, which is required to operate *rSGDLM*.

Once the package has loaded successfully, generate a module by

```
M = Module("gpuSGDLM", "rSGDLM")
```

This module provides access to the underlying object oriented structure of the wrapped C++/cuda-implemented software.

2 Initializing a Simultaneous Graphical DLM

A Simultaneous Graphical DLM object can be created from the module `M` by

```
sgdlm1 = new(M$SGDLM, no_gpus)
```

where the variable `no_gpus` specifies the number of GPU devices utilized by the package. If no value is provided, the software defaults to using one GPU.

3 Specifying the Simultaneous Parental Sets

The state vectors $\boldsymbol{\theta}_{jt}$ of all series $j = 1:m$ are combined in a joint state matrix

$$\begin{aligned}
\boldsymbol{\Theta}_t &= \begin{pmatrix} \vdots & \vdots & & \vdots \\ \boldsymbol{\theta}_{1t} & \boldsymbol{\theta}_{2t} & \cdots & \boldsymbol{\theta}_{mt} \\ \vdots & \vdots & & \vdots \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{\phi}_{1t} \in \mathbb{R}^{p_{1\phi}} & \boldsymbol{\phi}_{2t} \in \mathbb{R}^{p_{2\phi}} & \cdots & \boldsymbol{\phi}_{mt} \in \mathbb{R}^{p_{m\phi}} \\ \boldsymbol{\gamma}_{1t} \in \mathbb{R}^{p_{1\gamma}} & \boldsymbol{\gamma}_{2t} \in \mathbb{R}^{p_{2\gamma}} & \cdots & \boldsymbol{\gamma}_{mt} \in \mathbb{R}^{p_{m\gamma}} \\ \mathbf{0} \in \mathbb{R}^{\max_j(p_{j\phi}+p_{j\gamma})-p_{1\phi}-p_{1\gamma}} & \mathbf{0} \in \mathbb{R}^{\max_j(p_{j\phi}+p_{j\gamma})-p_{2\phi}-p_{2\gamma}} & \cdots & \mathbf{0} \in \mathbb{R}^{\max_j(p_{j\phi}+p_{j\gamma})-p_{1\phi}-p_{m\gamma}} \end{pmatrix} \\
&\in \mathbb{R}^{\max_j(p_{j\phi}+p_{j\gamma})\times m}
\end{aligned}$$

where each series state vector $\boldsymbol{\theta}_{jt}$, $j = 1:m$, concatenates that series' dynamic regression coefficients $\boldsymbol{\phi}_{jt}$, simultaneous parental coefficients $\boldsymbol{\gamma}_{jt}$, and filling zeros if $p_{j\phi} + p_{j\gamma} < \max_j(p_{j\phi} + p_{j\gamma})$. The dimensions $p_j := p_{j\phi} + p_{j\gamma}$ are stored in the vector `p`.

The simultaneous parental sets $sp(j)$, $j = 1 : m$, are encoded by setting all entries $\boldsymbol{\phi}_{jt}$ as well as all filling zeros in $\boldsymbol{\Theta}_t$ to `NA` and replacing all entries $\boldsymbol{\gamma}_{jt}$ by the position number of their parental series in $\boldsymbol{\Gamma}_t$,

$$\mathrm{pos}(\boldsymbol{\Gamma}_t) = \begin{pmatrix} 0 & m & 2m & \cdots & (m-1)m \\ 1 & m+1 & 2m+1 & \cdots & (m-1)m+1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m-1 & 2m-1 & 3m-1 & \cdots & m^2-1 \end{pmatrix}.$$

With simultaneous parental sets encoded in the variable `sp` as above, parental sets are then created via

```
sgdlm1$setSimultaneousParents(p, sp)
```

Example: Consider the model

$$\begin{aligned} y_{1t} &= x_{1t,1}\phi_{1t,1} + x_{1t,2}\phi_{1t,2} + y_{3t}\gamma_{1t,1} + \nu_{1t} \\ y_{2t} &= x_{2t,1}\phi_{2t,1} + y_{1t}\gamma_{2t,1} + y_{3t}\gamma_{2t,2} + y_{4t}\gamma_{2t,3} + \nu_{2t} \\ y_{3t} &= x_{3t,1}\phi_{3t,1} + x_{3t,2}\phi_{3t,2} + x_{3t,3}\phi_{3t,3} + \nu_{3t} \\ y_{4t} &= y_{1t}\gamma_{4t,1} + y_{2t}\gamma_{4t,2} + \nu_{4t} \end{aligned}$$

so that

$$\begin{aligned} \boldsymbol{\phi}_{1t} &= (\phi_{1t,1}, \phi_{1t,2}) \in \mathbb{R}^2 & \qquad \boldsymbol{\gamma}_{1t} &= (\gamma_{1t,1}) \in \mathbb{R}^1 \\ \boldsymbol{\phi}_{2t} &= (\phi_{2t,1}) \in \mathbb{R}^1 & \boldsymbol{\gamma}_{2t} &= (\gamma_{2t,1}, \gamma_{2t,2}, \gamma_{2t,3}) \in \mathbb{R}^3 \\ \boldsymbol{\phi}_{3t} &= (\phi_{3t,1}, \phi_{3t,2}, \phi_{3t,3}) \in \mathbb{R}^3 & \boldsymbol{\gamma}_{3t} &= () \in \mathbb{R}^0 \\ \boldsymbol{\phi}_{4t} &= () \in \mathbb{R}^0 & \boldsymbol{\gamma}_{4t} &= (\gamma_{4t,1}, \gamma_{4t,2}) \in \mathbb{R}^2 \end{aligned}$$

and $p = (2+1, 1+3, 3+0, 0+2) = (3, 4, 3, 2)$. The simultaneous coefficients matrix $\boldsymbol{\Gamma}_t$ is

$$\boldsymbol{\Gamma}_t = \begin{pmatrix} 0 & 0 & \gamma_{1t,1} & 0 \\ \gamma_{2t,1} & 0 & \gamma_{2t,2} & \gamma_{2t,3} \\ 0 & 0 & 0 & 0 \\ \gamma_{4t,1} & \gamma_{4t,2} & 0 & 0 \end{pmatrix} \text{ with } \mathrm{pos}(\boldsymbol{\Gamma}_t) = \begin{pmatrix} 0 & 4 & \mathbf{8} & 12 \\ \mathbf{1} & 5 & \mathbf{9} & \mathbf{13} \\ 2 & 6 & 10 & 14 \\ \mathbf{3} & \mathbf{7} & 11 & 15 \end{pmatrix}.$$

The corresponding position entry of $\gamma_{1t,1}$ is 8, $\gamma_{2t,1}$ is 1, $\gamma_{2t,2}$ is 9, $\gamma_{2t,3}$ is 13, $\gamma_{4t,1}$ is 3 and $\gamma_{4t,2}$ is 7. Here

$$\boldsymbol{\Theta}_t = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\theta}_{1t} & \boldsymbol{\theta}_{2t} & \boldsymbol{\theta}_{3t} & \boldsymbol{\theta}_{4t} \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} = \begin{pmatrix} \phi_{1t,1} & \phi_{2t,1} & \phi_{3t,1} & \boldsymbol{\gamma}_{4t,1} \\ \phi_{1t,2} & \boldsymbol{\gamma}_{2t,1} & \phi_{3t,2} & \boldsymbol{\gamma}_{4t,2} \\ \boldsymbol{\gamma}_{1t,1} & \boldsymbol{\gamma}_{2t,2} & \phi_{3t,3} & 0 \\ 0 & \boldsymbol{\gamma}_{2t,3} & 0 & 0 \end{pmatrix}$$

so the parental matrix is

$$sp = \begin{pmatrix} \mathtt{NA} & \mathtt{NA} & \mathtt{NA} & 3 \\ \mathtt{NA} & 1 & \mathtt{NA} & 7 \\ 8 & 9 & \mathtt{NA} & \mathtt{NA} \\ \mathtt{NA} & 13 & \mathtt{NA} & \mathtt{NA} \end{pmatrix}.$$

Create p and sp via

```
p = c(3,4,3,2)
sp = matrix(c(NA, NA, 8, NA, NA, 1, 9, 13, NA, NA, NA, NA, 3, 7, NA, NA), 4)
```

and then set

```
sgdlm1$setSimultaneousParents(p, sp)
```

4 Setting Discount Factors

Discount factors β_j and Δ_j are specified via

```
sgdlm1$setDiscountFactors(beta, Delta)
```

where `beta` is an m-vector of discount factors for residual volatilities and `Delta` is a $\max(p) \times \max(p) \times m$ matrix that specifies element-wise the discount factors for states. The β and δ notation is the same as in the references.

5 Setting Initial Parameters

Initial prior parameters $\mathbf{a}_{jt_0}$, $\mathbf{R}_{jt_0}$, r_{jt_0} and c_{jt_0} or posterior parameters $\mathbf{m}_{jt_0}$, $\mathbf{C}_{jt_0}$, n_{jt_0} and s_{jt_0} are set via

```
sgdlm1$setPriorParameters(a, R, r, c)
```

or

```
sgdlm1$setPosteriorParameters(m, C, n, s)
```

Here $\mathbf{a}_{jt_0}$ or $\mathbf{m}_{jt_0}$ is a $\max(p) \times m$ matrix of which each column specifies the mode of $\boldsymbol{\theta}_{jt_0}$, and is filled up with zeros if $p_j < \max(p)$. $\mathbf{R}_{jt_0}$ or $\mathbf{C}_{jt_0}$ is a $\max(p)\times\max(p)\times m$ array where each $\max(p)\times\max(p)$ sub-matrix specifies the covariance factor of $\boldsymbol{\theta}_{jt_0}$, and entries outside the left-upper $p_j \times p_j$ block are ignored. r_{jt_0} or n_{jt_0} and c_{jt_0} or s_{jt_0} are m-vectors. Again, notation is as in the references.

Example: In the running example, set prior parameters as

```
a = array(0, c(max(p), 4))
R = array(0.0001*diag(max(p)), c(max(p), max(p), 4))
r = rep(1, 4)
c = rep(.01, 4)
sgdlm1$setPriorParameters(a, R, r, c)
```

6 Forward Filtering

Naive posteriors: On observing $\mathbf{y}_t$, the within-series posteriors are computed by applying

```
sgdlm1$computePosterior(y_t, F_t)
```

where `y_t` is a m-vector of observations and `F_t` is a $\max(p) \times m$ array of which each column j contains the external predictors $\mathbf{x}_{jt}$ and simultaneous values $y_{sp(j),t}$, possibly extended by zeros. The resulting naive posterior parameters $\widetilde{\mathbf{m}}_{jt}$, $\widetilde{\mathbf{C}}_{jt}$, $\widetilde{n}_{jt}$ and $\widetilde{s}_{jt}$ are stored in `sgdlm1` and can be obtained by

```
sgdlm1$getParameters()
```

VB posterior: The variational Bayes-decoupled posterior parameters $\mathbf{m}_{jt}$, $\mathbf{C}_{jt}$, n_{jt} and s_{jt} are obtained by

```
parVB = sgdlm1$computeVBPosterior(no_simulations, batch_size)
```

where the calculations are based on `no_simulations` importance samples, and `batch_size` importance samples are evaluated by one GPU at once. The latter parameter may have to be set substantially smaller than `no_simulations`, depending on available GPU device memory.

This function returns the VB parameters $\mathbf{m}_{jt}$, $\mathbf{C}_{jt}$, n_{jt} and s_{jt}, the importance weights $(\alpha_i)_{i=1:no\_simulations}$, and the total sum of the determinant weights $\sum_i |\mathbf{I}-\boldsymbol{\Gamma}_t^i|$. Upon verification of the results, the VB-decoupled parameters can be accepted via

```
sgdlm1$setPosteriorParameters(parVB$m, parVB$C, parVB$n, parVB$s)
```

The VB-decoupled parameters are not automatically stored in `sgdlm1`.

Step-ahead prior: The model implemented in the references has random walk evolutions for states. For such models, the step-ahead priors with the implied $\mathbf{G}_{jt} = \mathrm{diag}(1 \cdots 1)$ are computed by

```
sgdlm1$computePrior()
```

and the prior parameters $\mathbf{a}_{j,t+1}$, $\mathbf{R}_{j,t+1}$, $r_{j,t+1}$ and $c_{j,t+1}$ are automatically stored in `sgdlm1`. They can be retrieved using

```
sgdlm1$getParameters()
```

For general state evolution matrices $\mathbf{G}_{j,t+1}$, the step-ahead priors are computed by

```
sgdlm1$computeEvoPrior(G_tp1)
```

where `G_tp1` is a $\max(p) \times \max(p) \times m$ array and the j-th $\max(p) \times \max(p)$ sub-matrix specifies series j's state evolution matrix $\mathbf{G}_{j,t+1}$. If $p_j < \max(p)$, the off-diagonal entries of $\mathbf{G}_{j,t+1}$ outside of the left-upper $p_j \times p_j$ sub-matrix must be set to zero, and the diagonal entries must be set to one.

One-step Forecasts: Given external predictors $\mathbf{x}_{j,t+1}$, one-step ahead forecasts are computed by

```
y.forecasts = sgdlm1$computeForecast(no_simulations, batch_size, F_tp1)
```

where `F_tp1` is a $\max(p) \times m$ matrix of which the j-th column contains the external predictors $\mathbf{x}_{j,t+1}$ and padding zeros.