Package ‘SMFilter’

October 12, 2022

Title  Filtering Algorithms for the State Space Models on the Stiefel Manifold
Version  1.0.3
Description  Provides the filtering algorithms for the state space models on the Stiefel manifold as well as the corresponding sampling algorithms for uniform, vector Langevin-Bingham and matrix Langevin-Bingham distributions on the Stiefel manifold.
Depends  R (>= 3.0.0)
License  GPL-3
Encoding  UTF-8
LazyData  true
RoxygenNote  6.1.1
URL  https://github.com/yukai-yang/SMFilter
BugReports  https://github.com/yukai-yang/SMFilter/issues
Suggests  knitr, rmarkdown, ggplot2
VignetteBuilder  knitr
NeedsCompilation  no
Author  Yukai Yang [aut, cre]
Maintainer  Yukai Yang <yukai.yang@statistik.uu.se>
Repository  CRAN
Date/Publication  2018-12-12 22:20:03 UTC

R topics documented:

FDist2 ................................................................. 2
FilterModel1 .......................................................... 3
FilterModel2 .......................................................... 4
rmLB_sm .............................................................. 6
runif_sm .............................................................. 7
rvlb_sm .............................................................. 8
SimModel1 ............................................................ 9
FDist2

Compute the squared Frobenius distance between two matrices.

**Description**

This function Compute the squared Frobenius distance between two matrices.

**Usage**

```r
FDist2(mX, mY)
```

**Arguments**

- `mX`: a $p \times r$ matrix where $p \geq r$.
- `mY`: another $p \times r$ matrix where $p \geq r$.

**Details**

The Frobenius distance between two matrices is defined to be

$$d(X, Y) = \sqrt{\text{tr}(A'A)}$$

where $A = X - Y$.

The Frobenius distance is a possible measure of the distance between two points on the Stiefel manifold.

**Value**

the Frobenius distance.

**Author(s)**

Yukai Yang, <yukai.yang@statistik.uu.se>

**Examples**

```r
FDist2(runif_sm(1,4,2)[1,,], runif_sm(1,4,2)[1,,])
```
Description

This function implements the filtering algorithm for the type one model. See Details part below.

Usage

FilterModel1(mY, mX, mZ, beta, mB = NULL, Omega, vD, U0, method = "max_1")

Arguments

- mY: the matrix containing $Y_t$ with dimension $T \times p$.
- mX: the matrix containing $X_t$ with dimension $T \times q_1$.
- mZ: the matrix containing $Z_t$ with dimension $T \times q_2$.
- beta: the $\beta$ matrix.
- mB: the coefficient matrix $B$ before $mZ$ with dimension $p \times q_2$.
- Omega: covariance matrix of the errors.
- vD: vector of the diagonals of $D$.
- U0: initial value of the alpha sequence.
- method: a string representing the optimization method from c('max_1','max_2','max_3','min_1','min_2').

Details

The type one model on Stiefel manifold takes the form:

$$y_t = \alpha_t \beta' x_t + B z_t + \varepsilon_t$$

$$\alpha_{t+1} | \alpha_t \sim ML(p, r, \alpha_t D)$$

where $y_t$ is a $p$-vector of the dependent variable, $x_t$ and $z_t$ are explanatory variables with dimension $q_1$ and $q_2$, $x_t$ and $z_t$ have no overlap, matrix $B$ is the coefficients for $z_t$, $\varepsilon_t$ is the error vector.

The matrices $\alpha_t$ and $\beta$ have dimensions $p \times r$ and $q_1 \times r$, respectively. Note that $r$ is strictly smaller than both $p$ and $q_1$. $\alpha_t$ and $\beta$ are both non-singular matrices. $\alpha_t$ is time-varying while $\beta$ is time-invariant.

Furthermore, $\alpha_t$ fulfills the condition $\alpha_t' \alpha_t = I_r$, and therefore it evolves on the Stiefel manifold. $ML(p, r, \alpha_t D)$ denotes the Matrix Langevin distribution or matrix von Mises-Fisher distribution on the Stiefel manifold. Its density function takes the form

$$f(\alpha_{t+1}) = \frac{etr \{ D \alpha_t' \alpha_{t+1} \}}{\alpha F_1(\frac{p}{2}; \frac{1}{4} D^2)}$$

where $etr$ denotes $\exp(tr())$, and $F_1(\frac{p}{2}; \frac{1}{4} D^2)$ is the (0,1)-type hypergeometric function for matrix.
FilterModel2

Filtering algorithm for the type two model.

Value

an array aAlpha containing the modal orientations of alpha in the prediction step.

Author(s)

Yukai Yang, <yukai.yang@statistik.uu.se>

Examples

iT = 50
ip = 2
ir = 1
iqx = 4
iqz=0
ik = 0
Omega = diag(ip)*.1

if(iqx==0) mX=NULL else mX = matrix(rnorm(iT*iqx),iT, iqx)
if(iqz==0) mZ=NULL else mZ = matrix(rnorm(iT*iqz),iT, iqz)
if(ik==0) mY=NULL else mY = matrix(0, ik, ip)

alpha_0 = matrix(c(runif_sm(num=1,ip=ip,ir=ir)), ip, ir)
beta = matrix(c(runif_sm(num=1,ip=ip*ik+iqx,ir=ir)), ip*ik+iqx, ir)

mB=NULL
vD = 100

ret = SimModel1(iT=iT, mX=mX, mZ=mZ, mY=mY, alpha_0=alpha_0, beta=beta, mB=mB, vD=vD, Omega=Omega)
mYY=as.matrix(ret$dData[,1:ip])
fil = FilterModel1(mY=mYY, mX=mX, mZ=mZ, beta=beta, mB=mB, Omega=Omega, vD=vD, U0=alpha_0)

Description

This function implements the filtering algorithm for the type two model. See Details part below.

Usage

FilterModel2(mY, mX, mZ, alpha, mB = NULL, Omega, vD, U0, 
method = "max_1")
Arguments

- \texttt{mY} the matrix containing \( Y_t \) with dimension \( T \times p \).
- \texttt{mX} the matrix containing \( X_t \) with dimension \( T \times q_1 \).
- \texttt{mZ} the matrix containing \( Z_t \) with dimension \( T \times q_2 \).
- \texttt{alpha} the \( \alpha \) matrix.
- \texttt{mB} the coefficient matrix \( B \) before \( mZ \) with dimension \( p \times q_2 \).
- \texttt{Omega} covariance matrix of the errors.
- \texttt{vD} vector of the diagonals of \( D \).
- \texttt{U0} initial value of the alpha sequence.
- \texttt{method} a string representing the optimization method from c(’max_1’,’max_2’,’max_3’,’min_1’,’min_2’).

Details

The type two model on Stiefel manifold takes the form:

\[
y_t = \alpha \beta_t' x_t + B' z_t + \varepsilon_t
\]

\[
\beta_{t+1} | \beta_t \sim ML(q_1, r, \beta, D)
\]

where \( y_t \) is a \( p \)-vector of the dependent variable, \( x_t \) and \( z_t \) are explanatory variables with dimension \( q_1 \) and \( q_2 \), \( x_t \) and \( z_t \) have no overlap, matrix \( B \) is the coefficients for \( z_t \), \( \varepsilon_t \) is the error vector.

The matrices \( \alpha \) and \( \beta_t \) have dimensions \( p \times r \) and \( q_1 \times r \), respectively. Note that \( r \) is strictly smaller than both \( p \) and \( q_1 \). \( \alpha \) and \( \beta_t \) are both non-singular matrices. \( \beta_t \) is time-varying while \( \alpha \) is time-invariant.

Furthermore, \( \beta_t \) fulfills the condition \( \beta_t' \beta_t = I_r \), and therefore it evolves on the Stiefel manifold.

\( ML(p, r, \beta, D) \) denotes the Matrix Langevin distribution or matrix von Mises-Fisher distribution on the Stiefel manifold. Its density function takes the form

\[
f(\beta_{t+1}) = \frac{\exp \{ D \beta_{t+1} \}}{\,_{0}F_{1}(\frac{p+1}{2}; \frac{1}{4} D^2)}
\]

where \( \exp(\text{tr}()) \), and \( \,_{0}F_{1}(\frac{p+1}{2}; \frac{1}{4} D^2) \) is the \((0,1)\)-type hypergeometric function for matrix.

Value

an array \texttt{aAlpha} containing the modal orientations of alpha in the prediction step.

Author(s)

Yukai Yang, <yukai.yang@statistik.uu.se>
Examples

```r
iT = 50
ip = 2
ir = 1
iqx = 4
iqz = 0
ik = 0
Omega = diag(ip) * 0.1

if(iqx == 0) mX = NULL else mX = matrix(rnorm(iT * iqx), iT, iqx)
if(iqz == 0) mZ = NULL else mZ = matrix(rnorm(iT * iqz), iT, iqz)
if(ik == 0) mY = NULL else mY = matrix(0, ik, ip)

alpha = matrix(c(runif_sm(num=1, ip=ip, ir=ir)), ip, ir)
beta_0 = matrix(c(runif_sm(num=1, ip=ip*ik+iqx, ir=ir)), ip*ik+iqx, ir)
mB = NULL
vD = 100

ret = SimModel2(iT = iT, mX = mX, mZ = mZ, mY = mY, alpha = alpha, beta_0 = beta_0, mB = mB, vD = vD)
mYY = as.matrix(ret$dData[, 1:ip])
fit = FilterModel2(mY = mYY, mX = mX, mZ = mZ, alpha = alpha, mB = mB, Omega = Omega, vD = vD, U0 = beta_0)
```

---

**Description**

This function draws a sample from the matrix Langevin-Bingham on the Stiefel manifold.

**Usage**

```r
rmLB_sm(num, mJ, mH, mC, mX, ir)
```

**Arguments**

- `num` number of observations or sample size.
- `mJ` symmetric ip*ip matrix
- `mH` symmetric ir*ir matrix
- `mC` ip*ir matrix
- `mX` ip*ir matrix, the initial value
- `ir` ir
Details

The matrix Langevin-Bingham distribution on the Stiefel manifold has the density kernel:

\[ f(X) \propto \text{etr}\{HX'JX + C'X\} \]

where \( X \) satisfies \( X'X = I_r \), and \( H \) and \( J \) are symmetric matrices.

Value

an array containing a sample of draws from the matrix Langevin-Bingham on the Stiefel manifold.

Author(s)

Yukai Yang, <yukai.yang@statistik.uu.se>


---

runif_sm

Sample from the uniform distribution on the Stiefel manifold.

Description

This function draws a sample from the uniform distribution on the Stiefel manifold.

Usage

runif_sm(num, ip, ir)

Arguments

num number of observations or sample size.

ip the first dimension \( p \) of the matrix.

ir the second dimension \( r \) of the matrix.

Details

The Stiefel manifold with dimension \( p \) and \( r \) (\( p \geq r \)) is a space whose points are \( r \)-frames in \( R^p \). A set of \( r \) orthonormal vectors in \( R^p \) is called an \( r \)-frame in \( R^p \). The Stiefel manifold is a collection of \( p \times r \) full rank matrices \( X \) such that \( X'X = I_r \).

Value

an array with dimension num, ip and ir containing a sample of draws from the uniform distribution on the Stiefel manifold.
Author(s)
Yukai Yang, <yukai.yang@statistik.uu.se>

Examples

```r
runif_sm(10, 4, 2)
```

---

**rvlb_sm**

*Sample from the vector Langevin-Bingham on the Stiefel manifold.*

**Description**

This function draws a sample from the vector Langevin-Bingham on the Stiefel manifold.

**Usage**

```r
rvlb_sm(num, mA, vc, vx)
```

**Arguments**

- `num`: number of observations or sample size.
- `mA`: the matrix A which is symmetric ip*ip matrix.
- `vc`: the vector c with dimension ip.
- `vx`: the vector x, the initial value.

**Details**

The vector Langevin-Bingham distribution on the Stiefel manifold has the density kernel:

\[ f(X) \propto \text{etr} \{ x'Ax + c'x \} \]

where \( x \) satisfies \( x'x = 1 \), and \( A \) is a symmetric matrix.

**Value**

an array containing a sample of draws from the vector Langevin-Bingham on the Stiefel manifold.

**References**


**Author(s)**

Yukai Yang, <yukai.yang@statistik.uu.se>
Simulate from the type one state-space Model on Stiefel manifold.

**Description**

This function simulates from the type one model on Stiefel manifold. See Details part below.

**Usage**

```r
SimModel1(iT, mX = NULL, mZ = NULL, mY = NULL, alpha_0, beta,
           mB = NULL, Omega = NULL, vD, burnin = 100)
```

**Arguments**

- `iT` the sample size.
- `mX` the matrix containing $X_t$ with dimension $T \times q_1$.
- `mZ` the matrix containing $Z_t$ with dimension $T \times q_2$.
- `mY` initial values of the dependent variable for $k=1$ up to 0. If `mY = NULL`, then no lagged dependent variables in regressors.
- `alpha_0` the initial alpha, $p \times r$.
- `beta` the $\beta$ matrix, $i q x p r k y_{1,t-1} y_{1,t-2} ... y_{2,t-1} y_{2,t-2} ...$
- `mB` the coefficient matrix $B$ before $mZ$ with dimension $p \times q_2$.
- `Omega` covariance matrix of the errors.
- `vD` vector of the diagonals of $D$.
- `burnin` burn-in sample size (matrix Langevin).

**Details**

The type one model on Stiefel manifold takes the form:

\[
y_t = \alpha_t \beta' x_t + B z_t + \epsilon_t
\]

\[
\alpha_{t+1} | \alpha_t \sim ML(p, r, \alpha_t D)
\]

where $y_t$ is a $p$-vector of the dependent variable, $x_t$ and $z_t$ are explanatory variables with dimension $q_1$ and $q_2$, $x_t$ and $z_t$ have no overlap, matrix $B$ is the coefficients for $z_t$, $\epsilon_t$ is the error vector.

The matrices $\alpha_t$ and $\beta$ have dimensions $p \times r$ and $q_1 \times r$, respectively. Note that $r$ is strictly smaller than both $p$ and $q_1$. $\alpha_t$ and $\beta$ are both non-singular matrices. $\alpha_t$ is time-varying while $\beta$ is time-invariant.

Furthermore, $\alpha_t$ fulfills the condition $\alpha_t' \alpha_t = I_r$, and therefore it evolves on the Stiefel manifold. $ML(p, r, \alpha_t D)$ denotes the Matrix Langevin distribution or matrix von Mises-Fisher distribution on the Stiefel manifold. Its density function takes the form

\[
f(\alpha_{t+1}) = \exp(\text{tr}(\alpha_{t+1}' \alpha_t)) \cdot \frac{\text{etr}\{D \alpha_t' \alpha_{t+1}\}}{\Gamma\left(\frac{p}{2}\right) \Gamma\left(\frac{1}{4} D^2\right)}
\]

where $\text{etr}$ denotes $\exp(\text{tr}())$, and $\Gamma\left(\frac{p}{2}\right) \Gamma\left(\frac{1}{4} D^2\right)$ is the $(0,1)$-type hypergeometric function for matrix. Note that the function does not add intercept automatically.
Value

A list containing the sampled data and the dynamics of alpha.

The object is a list containing the following components:

- **dData**: a data.frame of the sampled data
- **aAlpha**: an array of the $\alpha_t$ with the dimension $T \times p \times r$

Author(s)

Yukai Yang, <yukai.yang@statistik.uu.se>

Examples

```r
iT = 50 # sample size
ip = 2 # dimension of the dependent variable
ir = 1 # rank number
iqx=2 # number of variables in X
iqz=2 # number of variables in Z
ik = 1 # lag length

if(iqx==0) mX=NULL else mX = matrix(rnorm(it*iqx),iT, iqx)
if(iqz==0) mZ=NULL else mZ = matrix(rnorm(iT*iqz),iT, iqz)
if(ik==0) mY=NULL else mY = matrix(0, ik, ip)

alpha_0 = matrix(c(runif_sm(num=1,ip=ip,ir=ir)), ip, ir)
beta = matrix(c(runif_sm(num=1,ip=ip*ik+iqx,ir=ir)), ip*ik+iqx, ir)
if(ip*ik+iqz==0) mB=NULL else mB = matrix(c(runif_sm(num=1,ip=(ip*ik+iqz)*ip,ir=1)), ip, ip*ik+iqz)
vd = 50

ret = SimModel1(iT=iT, mX=mX, mZ=mZ, mY=mY, alpha_0=alpha_0, beta=beta, mB=mB, vD=vd)
```

SimModel2

Simulate from the type two state-space Model on Stiefel manifold.

Description

This function simulates from the type two model on Stiefel manifold. See Details part below.

Usage

```r
SimModel2(iT, mX = NULL, mZ = NULL, mY = NULL, beta_0, alpha,
           mB = NULL, Omega = NULL, vD, burnin = 100)
```
Arguments

iT the sample size.
mX the matrix containing X_t with dimension $T \times q_1$.
mZ the matrix containing Z_t with dimension $T \times q_2$.
mY initial values of the dependent variable for $ik-1$ up to 0. If $mY = \text{NULL}$, then no lagged dependent variables in regressors.
beta_0 the initial beta, $iqx+ip*ik$, $y_{1,t-1},y_{1,t-2},...,y_{2,t-1},y_{2,t-2},...$
alpha the $\alpha$ matrix, $p \times r$.
mB the coefficient matrix $B$ before $mZ$ with dimension $p \times q_2$.
Omega covariance matrix of the errors.
vD vector of the diagonals of $D$.
burnin burn-in sample size (matrix Langevin).

Details

The type two model on Stiefel manifold takes the form:

$$y_t = \alpha \beta'_t x_t + B' z_t + \epsilon_t$$

$$\beta_{t+1}|\beta_t \sim ML(q_1, r, \beta_t D)$$

where $y_t$ is a $p$-vector of the dependent variable, $x_t$ and $z_t$ are explanatory variables with dimension $q_1$ and $q_2$, $x_t$ and $z_t$ have no overlap, matrix $B$ is the coefficients for $z_t$, $\epsilon_t$ is the error vector.

The matrices $\alpha$ and $\beta_t$ have dimensions $p \times r$ and $q_1 \times r$, respectively. Note that $r$ is strictly smaller than both $p$ and $q_1$. $\alpha$ and $\beta_t$ are both non-singular matrices. $\beta_t$ is time-varying while $\alpha$ is time-invariant.

Furthermore, $\beta_t$ fulfills the condition $\beta'_t \beta_t = I_r$, and therefore it evolves on the Stiefel manifold. $ML(p, r, \beta_t D)$ denotes the Matrix Langevin distribution or matrix von Mises-Fisher distribution on the Stiefel manifold. Its density function takes the form

$$f(\beta_{t+1}) = \frac{\exp \left\{ D \beta'_t \beta_{t+1} \right\}}{\text{etr}(D^2)}$$

where $\text{etr}$ denotes $\exp(\text{tr}())$, and $\text{etr}(D^2)$ is the (0,1)-type hypergeometric function for matrix.

Note that the function does not add intercept automatically.

Value

A list containing the sampled data and the dynamics of beta.

The object is a list containing the following components:

dData a data.frame of the sampled data
aBeta an array of the $\beta_t$ with the dimension $T \times q_1 \times r$

Author(s)

Yukai Yang. <yukai.yang@statistik.uu.se>
Examples

iT = 50
ip = 2
ir = 1
iqx = 3
iqz = 2
ik = 1

if(iqx==0) mX=NULL else mX = matrix(rnorm(iT*iqx),iT, iqx)
if(iqz==0) mZ=NULL else mZ = matrix(rnorm(iT*iqz),iT, iqz)
if(ik==0) mY=NULL else mY = matrix(0, ik, ip)

alpha = matrix(c(runif_sm(num=1,ip=ip,ir=ir)), ip, ir)
beta_0 = matrix(c(runif_sm(num=1,ip=ip*ik+iqx,ir=ir)), ip*ik+iqx, ir)
if(ip*ik+iqz==0) mB=NULL else mB = matrix(c(runif_sm(num=1,ip=(ip*ik+iqz)*ip,ir=1)), ip, ip*ik+iqz)
vD = 50

ret = SimModel2(iT=iT, mX=mX, mZ=mZ, mY=mY, alpha=alpha, beta_0=beta_0, mB=mB, vD=vD)

SMFilter

SMFilter: a package implementing the filtering algorithms for the state-space models on the Stiefel manifold.

Description

The package implements the filtering algorithms for the state-space models on the Stiefel manifold. It also implements sampling algorithms for uniform, vector Langevin-Bingham and matrix Langevin-Bingham distributions on the Stiefel manifold.

Details

Two types of the state-space models on the Stiefel manifold are considered.

The type one model on Stiefel manifold takes the form:

\[
y_t = \alpha_t' \beta x_t + B z_t + \varepsilon_t
\]

\[
\alpha_{t+1} | \alpha_t \sim ML(p, r, \alpha_t D)
\]

where \(y_t\) is a \(p\)-vector of the dependent variable, \(x_t\) and \(z_t\) are explanatory variables wit dimension \(q_1\) and \(q_2\), \(x_t\) and \(z_t\) have no overlap, matrix \(B\) is the coefficients for \(z_t\), \(\varepsilon_t\) is the error vector.

The matrices \(\alpha_t\) and \(\beta\) have dimensions \(p \times r\) and \(q_1 \times r\), respectively. Note that \(r\) is strictly smaller than both \(p\) and \(q_1\). \(\alpha_t\) and \(\beta\) are both non-singular matrices. \(\alpha_t\) is time-varying while \(\beta\) is time-invariant.

Furthermore, \(\alpha_t\) fulfills the condition \(\alpha_t' \alpha_t = I_r\), and therefore it evolves on the Stiefel manifold.
ML \( (p, r, \alpha, D) \) denotes the Matrix Langevin distribution or matrix von Mises-Fisher distribution on the Stiefel manifold. Its density function takes the form

\[
f(\alpha_{t+1}) = \frac{\exp\left\{D\alpha_{t}^\prime \alpha_{t+1}\right\}}{0F_1\left(\frac{p}{2}; \frac{1}{4}D^2\right)}
\]

where \( \exp(\text{tr}()) \) denotes the matrix exponential function. \( 0F_1 \) is the (0,1)-type hypergeometric function for matrix.

The type two model on Stiefel manifold takes the form:

\[
y_t = \alpha^\prime x_t + B^\prime z_t + \varepsilon_t
\]

\[
\beta_{t+1} | \beta_t \sim ML(q_1, r, \beta, D)
\]

where \( y_t \) is a \( p \)-vector of the dependent variable, \( x_t \) and \( z_t \) are explanatory variables with dimensions \( q_1 \) and \( q_2 \), respectively. \( x_t \) and \( z_t \) have no overlap, matrix \( B \) is the coefficients for \( z_t \), \( \varepsilon_t \) is the error vector. The matrices \( \alpha \) and \( \beta_t \) have dimensions \( p \times r \) and \( q_1 \times r \), respectively. Note that \( r \) is strictly smaller than both \( p \) and \( q_1 \), \( \alpha \) and \( \beta_t \) are both non-singular matrices. \( \beta_t \) is time-varying while \( \alpha \) is time-invariant.

Furthermore, \( \beta_t \) fulfills the condition \( \beta^\prime_t \beta_t = I_r \), and therefore it evolves on the Stiefel manifold.

ML \( (p, r, \beta, D) \) denotes the Matrix Langevin distribution or matrix von Mises-Fisher distribution on the Stiefel manifold. Its density function takes the form

\[
f(\beta_{t+1}) = \frac{\exp\left\{D\beta_{t}^\prime \beta_{t+1}\right\}}{0F_1\left(\frac{p}{2}; \frac{1}{4}D^2\right)}
\]

where \( \exp(\text{tr}()) \) denotes the matrix exponential function. \( 0F_1 \) is the (0,1)-type hypergeometric function for matrix.

Author and Maintainer

Yukai Yang
Department of Statistics, Uppsala University
<yukai.yang@statistik.uu.se>
Sampling

- `runif_sm` sample from the uniform distribution on the Stiefel manifold.
- `rvlb_sm` sample from the vector Langevin-Bingham distribution on the Stiefel manifold.
- `rmLB_sm` sample from the matrix Langevin-Bingham distribution on the Stiefel manifold.

Other Functions

- `version()` shows the version number and some information of the package.

Description

This function shows the version number and some information of the package.

Usage

```
version()
```

Author(s)

Yukai Yang. <yukai.yang@statistik.uu.se>
Index

* filtering
  FilterModel1, 3
  FilterModel2, 4
* simulation
  SimModel1, 9
  SimModel2, 10
* utils
  FDist2, 2
  rmLB_sm, 6
  runif_sm, 7
  rvlb_sm, 8
  version, 14

FDist2, 2
FilterModel1, 3, 13
FilterModel2, 4, 13

rmLB_sm, 6, 14
runif_sm, 7, 14
rvlb_sm, 8, 14

SimModel1, 9, 13
SimModel2, 10, 13
SMFilter, 12
SMFilter-package (SMFilter), 12

version, 14, 14