Package ‘MsdeParEst’

August 10, 2017

Type Package

Title Parametric estimation in mixed-effects stochastic differential equations

Version 0.1

Date 2017-07-18

Author Maud Delattre, Charlotte Dion

Maintainer Maud Delattre <maud.delattre@agroparistech.fr>

Description Parametric estimation in stochastic differential equations with random effects in the drift, or in the diffusion or both. Approximate maximum likelihood methods are used.

License GPL-3

Imports MASS, sde, moments, mvtnorm, methods, graphics

RoxygenNote 6.0.1

R topics documented:

`MsdeParEst-package` *Parametric estimation in mixed-effects stochastic differential equations*

Description

Parametric estimation in mixed-effects stochastic differential equations

Details

This package is dedicated to parametric estimation in the following mixed-effects stochastic differential equations:

$$dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma_j \; a(X_j(t))dW_j(t),$$

$j = 1, \ldots, N$, where the $(W_j(t))$ are independant Wiener processes. The volatility function $a(x)$ is known and can be either $a(x) = 1$ (Ornstein-Uhlenbeck process) or $a(x) = \sqrt{x}$ (Cox-Ingersoll-Ross process).

Different estimation methods are implemented depending on whether there are random effects in the drift and/or in the diffusion coefficient:

1. The diffusion coefficient is fixed $\sigma_j \equiv \sigma$ and the parameters in the drift are Gaussian random variables:
   (a) either $\alpha_j \equiv \alpha$ and $\beta_j \sim N(\mu, \Omega), j = 1, \ldots, N$,
   (b) or $\beta_j \equiv \beta$ and $\alpha_j \sim N(\mu, \Omega), j = 1, \ldots, N$,
   (c) or $(\alpha_j, \beta_j) \sim N(\mu, \Omega), j = 1, \ldots, N$.

   μ, Ω and potentially the fixed effects σ, α, β are estimated as proposed in:

   - Maximum Likelihood Estimation for Stochastic Differential Equations with Random Effects, Delattre, M., Genon-Catalot, V. and Samson, A. *Scandinavian Journal of Statistics 40(2) 2012* 322-343.

- Parametric inference for discrete observations of diffusion processes with mixed effects, Delattre, M., Genon-Catalot, V. and Laredo, C. *Stochastic Processes and their Applications*. To appear. (Available pre-publication hal-0133263).

The extension to mixtures of Gaussian distributions is also implemented by following:

- Mixtures of stochastic differential equations with random effects: application to data clustering, Delattre, M., Genon-Catalot, V. and Samson, A. *Journal of Statistical Planning and Inference 173 2016* 109-124.

2. The coefficients in the drift are fixed $\alpha_j \equiv \alpha$ and $\beta_j \equiv \beta$ and the diffusion coefficient $1/\sigma_j^2$ follows a Gamma distribution $1/\sigma_j^2 \sim \Gamma(a, \lambda), j = 1, \ldots, N$. a, λ, and potentially the fixed effects α and β are estimated by the method published in:

   - Estimation of population parameters in stochastic differential equations with random effects in the diffusion coefficient, Delattre, M., Genon-Catalot, V. and Samson, A. *ESAIM:PS 19 2015* 671-688.

3. There are random effects in the drift and in the diffusion, such that $1/\sigma_j^2 \sim \Gamma(a, \lambda)$ and

   (a) either $\alpha_j \equiv \alpha$ and $\beta_j|\sigma_j \sim N(\mu, \sigma_j^2\Omega)$,

   (b) or $\beta_j \equiv \beta$ and $\alpha_j|\sigma_j \sim N(\mu, \sigma_j^2\Omega)$,

   (c) or $(\alpha_j, \beta_j)|\sigma_j \sim N(\mu, \sigma_j^2\Omega)$.

   a, λ, μ, Ω and potentially the fixed effects α and β are estimated by following:

   - Estimation of the joint distribution of random effects for a discretely observed diffusion with random effects, Delattre, M., Genon-Catalot, V. and Laredo, C. *2017* (Prepublication hal-01446063).

Some examples and more details on how to use the different functions of the package are available on the package's vignette by typing **vignette("MsdeParEst")** in the R console.

---

bx *Computation Of The Drift Coefficient*

Description

Computation of the drift coefficient

Usage

```
bx(x)
```

Arguments

x	vector of data

Value

b	The drift is $b(x, \phi) = \phi_1 b_1(x) + \phi_2 b_2(x)$, the output is the vector $(b_1, b_2)^t$

`class.mixture.pred-class`

S4 class for the parametric estimation results when the random effects in the drift follow mixture of normal distributions

Description

S4 class for the parametric estimation results when the random effects in the drift follow mixture of normal distributions

Slots

`estim` object of class Mixture.fit.class containing the results of the model estimation

`phipred` numeric 1, 2, or c(1,2)

`Xpred` matrix of predicted trajectories (dimensions)

`idexpred` matrix of values on which the estimation of the density of the random effects is done

`class.pred-class`

S4 class for the estimation results in the mixed SDE with random effects in the drift, in the diffusion or both

Description

S4 class for the estimation results in the mixed SDE with random effects in the drift, in the diffusion or both

Slots

`estim` object of class Fit.class containing the results of the model estimation

`phipred` matrix of simulated values for the random effects in the drift that are used for prediction (dimensions)

`Xpred` matrix of simulated trajectories used for prediction (dimensions)

`idexpred` vector of indexes of the true trajectories that are used for prediction `contrastGamma` *Contrast based on the Euler approximation of the likelihood for parameter estimation when there is one random effect in the diffusion coefficient and no random effect in the drift.*

Description

Computation of the contrast based on the Euler approximation of -2 log-likelihood for the estimation of the mixed SDE:

$dXj(t) = (\alpha - \beta X_j(t))dt + \sigma_j a(X_j(t))dW_j(t)$

with Gamma distribution for $1/\sigma_j^2$ and fixed parameters in the drift.

Usage

```
contrastGamma(a, lambda, U, V, S, K, drift.fixed)
```

Arguments

`a`	value of the shape of the Gamma distribution.
`lambda`	value of the scape of the Gamma distribution.
`U`	matrix of M sufficient statistics U (see `UVS`).
`V`	list of the M sufficient statistics matrix V (see `UVS`).
`S`	vector of the M sufficient statistics S (see `UVS`).
`K`	number of times of observations.
`drift.fixed`	values of the fixed effects in the drift.

Value

`L`	value of the contrast

References

Estimation of population parameters in stochastic differential equations with random effects in the diffusion coefficient, M. Delattre, V. Genon-Catalot and A. Samson, *ESAIM: Probability and Statistics 2015*, Vol 19, **671 – 688**

Parametric inference for discrete observations of diffusion processes with mixed effects, M. Delattre, V. Genon-Catalot and C. Laredo, *Preprint 2016, hal-01332630*

contrastNormal — *Computation of the contrast used for the estimation of the normal conditional distribution of the random effects in the drift in mixed SDE with random effects in the drift and in the diffusion coefficient*

Description

Computation of the contrast used for the estimation of the parameters of the Gaussian conditional distribution of the random effects in the drift α_j, β_j when the SDE includes random effects in the drift and in the diffusion coefficient:

$dXj(t) = (\alpha_j - \beta_j Xj(t))dt + \sigma_j a(Xj(t))dWj(t)$.

Usage

```
contrastNormal(mu, omega, U, V, S, K, estimphi, drift.random)
```

Arguments

mu	value of the mean of the Gaussian distribution.
omega	value of the standard deviation of the Gaussian distribution.
U	matrix of M sufficient statistics U (see UVS).
V	list of the M sufficient statistics matrix V (see UVS).
S	vector of the M sufficient statistics S (see UVS).
K	number of times of observations.
estimphi	matrix of the M x 2 estimated parameters (α_j, β_j).
drift.random	random effects in the drift: 1 if one additive random effect, 2 if one multiplicative random effect or c(1,2) if 2 random effects.

Value

L	value of the contrast

References

Estimaton of the joint distribution of random effects for a discretely observed diffusion with random effects, M. Delattre, V. Genon-Catalot and C. Laredo, *Preprint, hal-01446063*.

`discr` *Simulation Of Random Variables*

Description

Simulation of (discrete) random variables from a vector of probability (the nonparametrically estimated values of the density renormalised to sum at 1) and a vector of real values (the grid of estimation)

Usage

```
discr(x, p)
```

Arguments

`x`	n real numbers
`p`	vector of probability, length n

Value

y a simulated value from the discrete distribution

`EM` *EM algorithm for mixtures of stochastic differential equations with random effects in the drift and a fixed effect in the diffusion coefficient*

Description

EM algorithm for parameter estimation in the mixed SDE

$dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma a(X_j(t))dW_j(t)$

with random effects in the drift α_j, β_j following a mixture of Gaussian distributions.

Usage

```
EM(U, V, S, K, drift.random, start, Niter = 10, drift.fixed = NULL,
  sigma = NULL)
```

Arguments

U	matrix of M sufficient statistics U (see UVS).
V	list of the M sufficient statistics matrix V (see UVS).
S	vector of the M sufficient statistics S (see UVS).
K	number of times of observations.
drift.random	random effects in the drift: 1 if one additive random effect, 2 if one multiplicative random effect or c(1,2) if 2 random effects.
start	list of starting values: mu, omega, mixt.prop, respectively for the mean and the standard deviation of the Gaussian distributions and the mixing proportions. mixt.prop is a vector of length N, where N stands for the number of mixture components. mu is a N x 2 matrix, first (resp. second) column is the mean of α_j (resp. β_j) if α_j (resp. β_j) is random, the fixed effect value otherwise. omega is a N x 2 matrix, the components corresponding to a fixed effect should be set to 0.
Niter	number of iterations. Defaults to 10.
drift.fixed	NULL if the fixed effects in the drift are estimated, vector of the N values of the fixed effect otherwise. Default to NULL.
sigma	value for the diffusion parameter if known (not estimated), NULL otherwise. Defaults to NULL.

Value

mu	estimated value of the mean at each iteration of the algorithm. Niter x N x 2 array.
omega	estimated value of the standard deviation at each iteration of the algorithm. Niter x N x 2 array.
mixt.prop	estimated value of the mixture proportions at each iteration of the algorithm. Niter x N matrix.
sigma	value of the diffusion parameter.
probindi	posterior component probabilites. M x N matrix.
BIChere	BIC indicator
AIChere	AIC indicator

References

Mixtures of stochastic differential equations with random effects: application to data clustering, M. Delattre, V. Genon-Catalot and A. Samson, *Journal of Statistical Planning and Inference 2016*, Vol 173, **109–124**

`EstParamGamma` *Estimation In Mixed Stochastic Differential Equations with fixed effects in the drift and one random effect in the diffusion coefficient*

Description

Parameter estimation of the mixed SDE with Gamma distribution of the diffusion random effect and fixed effects in the drift:

$dXj(t) = (\alpha - \beta Xj(t))dt + \sigma_j a(Xj(t))dWj(t), 1/\sigma_j^2 \sim \Gamma(a, lambda)$,

done with `likelihoodGamma`.

Usage

```
EstParamGamma(U, V, S, SigDelta, K, drift.param = NULL)
```

Arguments

`U`	matrix of M sufficient statistics U (see `UVS`).
`V`	list of the M sufficient statistics matrix V (see `UVS`).
`S`	vector of the M sufficient statistics S (see `UVS`).
`SigDelta`	vector of the M constant terms of the individual likelihood (see `UVS`).
`K`	number of times of observations.
`drift.param`	values of the fixed effects in the drift if know, NULL if the fixed effects are estimated. Defaults to NULL.

Value

`mu`	values of the fixed effects in the drift.
`a`	estimated value of the shape of the Gamma distribution.
`lambda`	estimated value of the scale of the Gamma distribution.
`BIChere`	BIC indicator.
`AIChere`	AIC indicator.

References

Estimation of population parameters in stochastic differential equations with random effects in the diffusion coefficient, M. Delattre, V. Genon-Catalot and A. Samson, *ESAIM: Probability and Statistics 2015*, Vol 19, **671 – 688**

`EstParamNormal` *Estimation In Mixed Stochastic Differential Equations with random effects in the drift and fixed effect in the diffusion coefficient*

Description

Estimation of the parameters of the mixed SDE:

$dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma a(X_j(t))dW_j(t),$

with Normal distribution of the random effects in the drift α_j, β_j and fixed parameter σ in the diffusion. Done with `likelihoodNormal`.

Usage

```
EstParamNormal(U, V, S, SigDelta = 0, K, drift.fixed = NULL, sigma = NULL,
  drift.random, discrete = 1)
```

Arguments

`U`	matrix of M sufficient statistics U (see `UVS`).
`V`	list of the M sufficient statistics matrix V (see `UVS`).
`S`	vector of the M sufficient statistics S (see `UVS`).
`SigDelta`	vector of the M constant terms of the individual likelihood (see `UVS`). Required only if discrete = 1. Defaults to 0.
`K`	number of times of observations.
`drift.fixed`	NULL if thz fixed effect in the drift is estimated, value of the fixed effect otherwise. Default to NULL.
`sigma`	value of the fixed effect in the diffusion if known (not estimated), NULL otherwise. Defaults to NULL.
`drift.random`	random effects in the drift: 1 if one additive random effect, 2 if one multiplicative random effect or c(1,2) if 2 random effects.
`discrete`	1 for discrete observations, 0 otherwise. If discrete = 0, the exact likelihood associated with continuous observations is discretized. If discrete = 1, the likelihood of the Euler scheme of the mixed SDE is computed. Defaults to 1.

Value

`mu`	estimated value of the mean of the Normal distribution
`omega`	estimated value of the standard deviation of the Normal distribution
`sigma`	value of the diffusion coefficient
`BIChere`	BIC indicator
`AIChere`	AIC indicator

References

Maximum likelihood estimation for stochastic differential equations with random effects, M. Delattre, V. Genon-Catalot and A. Samson, *Scandinavian Journal of Statistics 2012*, Vol 40, **322–343**

`EstParamNormalGamma` *Estimation In Mixed Stochastic Differential Equations with random effects in the drift and in the diffusion coefficient*

Description

Estimation of the parameters of the mixed SDE

$dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma_j a(X_j(t))dW_j(t)$

with Normal conditional distribution of the random effects in the drift $\alpha_j, \beta_j|\sigma_j \sim N(\mu, \sigma_j^2\Omega)$ and the square root of an inverse Gamma distributed random effect in the diffusion $\sigma_j^2 \sim Gamma(a, \lambda)$.

Usage

```
EstParamNormalGamma(U, V, S, SigDelta, K, drift.random, drift.fixed = NULL)
```

Arguments

`U`	matrix of M sufficient statistics U (see UVS).
`V`	list of the M sufficient statistics matrix V (see UVS).
`S`	vector of the M sufficient statistics S (see UVS).
`SigDelta`	vector of the M constant terms of the individual likelihood (see UVS).
`K`	number of times of observations.
`drift.random`	random effects in the drift: 1 if one additive random effect, 2 if one multiplicative random effect or c(1,2) if 2 random effects.
`drift.fixed`	NULL if the fixed effect(s) in the drift is (are) estimated, value of the fixed effect(s) otherwise. Default to NULL.

Value

`mu`	estimated value of the mean of the Normal distribution
`omega`	estimated value of the standard deviation of the Normal distribution
`a`	estimated value of the shape of the Gamma distribution.
`lambda`	estimated value of the scale of the Gamma distribution.
`BIChere`	BIC indicator
`AIChere`	AIC indicator

References

Estimaton of the joint distribution of random effects for a discretely observed diffusion with random effects, M. Delattre, V. Genon-Catalot and C. Laredo, *Preprint*, hal-01446063.

`Fit.class-class` *S4 class for the estimation results in the mixed SDE with random effects in the drift, in the diffusion or both*

Description

S4 class for the estimation results in the mixed SDE with random effects in the drift, in the diffusion or both

Slots

- `model` character 'OU' or 'CIR'
- `drift.random` numeric 0, 1, 2, or c(1,2)
- `diffusion.random` numeric 0 or 1
- `gridf` matrix of values on which the estimation of the density of the random effects in the drift is done
- `gridg` matrix of values on which the estimation of the density of the random effects in the diffusion is done
- `mu` numeric estimator of the mean mu of the drift random effects
- `omega` numeric estimator of the variance of the drift random effects
- `a` numeric estimator of the shape of the Gamma distribution for the diffusion random effect
- `lambda` numeric estimator of the scale of the Gamma distribution for the diffusion random effect
- `sigma2` numeric estimated value of σ^2 if the diffusion coefficient is not random
- `index` index of the used trajectories
- `estimphi` matrix of the estimator of the drift random effects
- `estimpsi2` vector of the estimator of the diffusion random effects σ_j^2
- `estimf` estimator of the (conditional) density of ϕ, matrix form
- `estimg` estimator of the density of ϕ, matrix form
- `estim.drift.fix` 1 if the user asked for the estimation of fixed parameter in the drift
- `estim.diffusion.fix` 1 if the user asked for the estimation of fixed diffusion coefficient
- `discrete` 1 if the estimation is based on the likelihood of discrete observations, 0 otherwise
- `bic` numeric bic
- `aic` numeric aic
- `times` vector of observation times, storage of input variable
- `X` matrix of observations, storage of input variable

`likelihoodGamma` *Computation of the Euler approximation of -2 log-likelihood when there is one random effect in the diffusion coefficient and fixed effects in the drift.*

Description

Computation of the Euler approximation of -2 log-likelihood of the mixed SDE:

$dX_jj(t) = (\alpha - \beta X_j(t))dt + \sigma_j a(X_j(t))dW_j(t)$

with inverse Gamma distribution for σ_j^2 and fixed parameters α, β in the drift.

Usage

```
likelihoodGamma(a, lambda, U, V, S, SigDelta, K, drift.fixed)
```

Arguments

`a`	value of the shape of the Gamma distribution.
`lambda`	value of the scape of the Gamma distribution.
`U`	matrix of M sufficient statistics U (see `UVS`).
`V`	list of the M sufficient statistics matrix V (see `UVS`).
`S`	vector of the M sufficient statistics S (see `UVS`).
`SigDelta`	vector of the M constant terms of the individual likelihood (see `UVS`).
`K`	number of times of observations.
`drift.fixed`	values of the fixed effects in the drift.

Value

`L`	value of -2 x log-likelihood

References

Estimation of population parameters in stochastic differential equations with random effects in the diffusion coefficient, M. Delattre, V. Genon-Catalot and A. Samson, *ESAIM: Probability and Statistics 2015*, Vol 19, **671 – 688**

Parametric inference for discrete observations of diffusion processes with mixed effects, M. Delattre, V. Genon-Catalot and C. Laredo, *Preprint 2016*, hal-01332630

likelihoodMixtureNormal

Computation of the Log Likelihood in mixtures of Mixed Stochastic Differential Equations

Description

Computation of -2 log-likelihood the mixed SDE

$dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma a(X_j(t))dW_j(t)$

with random effects in the drift α_j, β_j following a mixture of Gaussian distributions.

Usage

```
likelihoodMixtureNormal(mu, omega, sigma, mixt.prop, U, V, S, K, estimphi,
  drift.random)
```

Arguments

mu	mean of the random effects. N x 2 matrix, first (resp. second) column is the mean of α_j (resp. β_j) in each mixture component if α_j (resp. β_j) is random, the fixed effect value otherwise.
omega	standard deviation of the random effects. N x 2 matrix, the components corresponding to a fixed effect should be set to 0.
sigma	value of the diffusion parameter.
mixt.prop	vector of mixture proportions.
U	matrix of M sufficient statistics U (see UVS).
V	list of the M sufficient statistics matrix V (see UVS).
S	vector of the M sufficient statistics S (see UVS).
K	number of times of observations.
estimphi	matrix of the estimators of the random effects in the drift.
drift.random	random effects in the drift: 1 if one additive random effect, 2 if one multiplicative random effect or c(1,2) if 2 random effects.

Value

L	value of -2 x loglikelihood

References

Mixtures of stochastic differential equations with random effects: application to data clustering, M. Delattre, V. Genon-Catalot and A. Samson, *Journal of Statistical Planning and Inference 2016*, Vol 173, **109–124**

`likelihoodNormal` *Computation Of The Log Likelihood In Mixed Stochastic Differential Equations with random effects in the drift and fixed effect in the diffusion.*

Description

Computation of -2 log-likelihood of the mixed SDE

$dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma a(X_j(t))dW_j(t)$

with Normal distribution of the drift random effects α_j, β_j and fixed effect in the diffusion coefficient.

Usage

```
likelihoodNormal(mu, omega, sigma, U, V, S, SigDelta = 0, K, estimphi,
  drift.random, discrete = 1)
```

Arguments

`mu`	current value of the mean of the normal distribution.
`omega`	current value of the standard deviation of the normal distribution.
`sigma`	current value of the diffusion coefficient.
`U`	vector of the M sufficient statistics U (see `UVS`).
`V`	list of the M sufficient statistics matrix V (see `UVS`).
`S`	vector of the M sufficient statistics S (see `UVS`).
`SigDelta`	vector of the M constant terms of the individual likelihood (see `UVS`). Required only if discrete = 1. Defaults to 0.
`K`	number of times of observations.
`estimphi`	matrix of estimators of the random effects.
`drift.random`	random effects in the drift: 1 if one additive random effect, 2 if one multiplicative random effect or c(1,2) if 2 random effects.
`discrete`	1 for discrete observations, 0 otherwise. If discrete = 0, the exact likelihood associated with continuous observations is discretized. If discrete = 1, the likelihood of the Euler scheme of the mixed SDE is computed. Defaults to 1.

Value

`L`	value of -2 x loglikelihood

References

Maximum likelihood estimation for stochastic differential equations with random effects, M. Delattre, V. Genon-Catalot and A. Samson, *Scandinavian Journal of Statistics 2012*, Vol 40, **322–343**

Parametric inference for discrete observations of diffusion processes with mixed effects, M. Delattre, V. Genon-Catalot and C. Laredo, *Preprint*, hal-01332630

likelihoodNormalGamma *Computation Of The Log Likelihood of the Euer scheme in Mixed Stochastic Differential Equations with random effects in the drift and in the diffusion.*

Description

Computation of -2 log-likelihood of the Euler scheme the mixed SDE

$dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma_j a(X_j(t))dW_j(t)$,

with Normal conditional distribution of the random effects in the drift $\alpha_j, \beta_j|\sigma_j \sim N(\mu, \sigma_j^2\Omega)$ and inverse Gamma distribution of the random effect in the diffusion $1/\sigma_j^2 \sim Gamma(a, \lambda)$.

Usage

```
likelihoodNormalGamma(a, lambda, mu, omega, U, V, S, SigDelta, K, drift.random)
```

Arguments

a	current value of the shape of the Gamma distribution.
lambda	current value of the scape of the Gamma distribution.
mu	current value of the mean of the normal distribution.
omega	current value of the standard deviation of the normal distribution.
U	vector of the M sufficient statistics U (see UVS).
V	list of the M sufficient statistics V (see UVS).
S	vector of the M sufficient statistics S (see UVS).
SigDelta	vector of the M constant terms of the individual likelihood (see UVS). Required only if discrete = 1. Defaults to 0.
K	number of times of observations.
drift.random	random effects in the drift: 1 if one additive random effect, 2 if one multiplicative random effect or c(1,2) if 2 random effects.

Value

L	value of -2 x loglikelihood

References

Estimation of the joint distribution of random effects for a discretely observed diffusion with random effects, M. Delattre, V. Genon-Catalot and C. Laredo, *Preprint*, hal-01446063.

`likelihoodNormalindi` *Computation of the individual Log-Likelihoods in Mixed Stochastic Differential Equations*

Description

Computation of -2 log-likelihood of individual j in the mixed SDE with Normal distribution of the drift random effects and fixed effect in the diffusion:

$dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma a(X_j(t))dW_j(t).$

Usage

```
likelihoodNormalindi(mu, omega, sigma, Uj, Vj, Sj, K, estimphij, drift.random)
```

Arguments

`mu`	vector of mean of the random effects. First (resp. second) value is the mean of α_j (resp. β_j) if α_j (resp. β_j) is random, the fixed effect value otherwise.
`omega`	standard deviation of the random effects. The components corresponding to a fixed effect should be set to 0.
`sigma`	value of the diffusion parameter.
`Uj`	vector of the sufficient statistics U for individual j (see `UVS`).
`Vj`	matrix of the sufficient statistics V for individual j (see `UVS`).
`Sj`	value of the sufficient statistic S for individual j (see `UVS`).
`K`	number of times of observations.
`estimphij`	vector of estimators of the random effects for individual j.
`drift.random`	random effects in the drift: 1 if one additive random effect, 2 if one multiplicative random effect or c(1,2) if 2 random effects.

Value

`L`	value of -2 x loglikelihood

References

Maximum likelihood estimation for stochastic differential equations with random effects, M. Delattre, V. Genon-Catalot and A. Samson, *Scandinavian Journal of Statistics 2012*, Vol 40, **322–343**

Mixture.fit.class-class

S4 class for the estimation results when the random effects in the drift follow mixture of normal distributions

Description

S4 class for the estimation results when the random effects in the drift follow mixture of normal distributions

Slots

- `model` character 'OU' or 'CIR'
- `drift.random` numeric 1, 2, or c(1,2)
- `gridf` matrix of values on which the estimation of the density of the random effects is done
- `mu` array estimated value of the mean of the drift random effects at each iteration of the EM algorithm (Niter x nb.mixt x 2)
- `omega` array estimated value of the standard deviation of the drift random effects at each iteration of the EM algorithm (Niter x nb.mixt x 2)
- `mixt.prop` matrix estimated value of the mixing proportions at each iteration of the EM algorithm (Niter x nb.mixt)
- `sigma2` numeric estimated value of σ^2
- `index` index of the used trajectories
- `estimphi` matrix of the estimator of the drift random effects
- `probindi` matrix of posterior component probabilities
- `estimf` matrix estimator of the density of the drift random effects
- `estim.drift.fix` numeric 1 if the user asked for the estimation of fixed parameter in the drift
- `bic` numeric bic
- `aic` numeric aic
- `times` vector of observation times, storage of input variable
- `X` matrix of observations, storage of input variable

`mixture.sim` *Simulation Of A Mixture Of Normal Distributions*

Description

Simulation of M random variables from a mixture of Gaussian distributions

Usage

```
mixture.sim(M, param, prob)
```

Arguments

`M`	number of simulated variables
`param`	vector of parameters with the means and standard-deviations of the normal distributions
`prob`	mixture components probabilities

Details

If the distribution is $p1N(\mu1, \sigma1^2) + (1 - p1)N(\mu2, \sigma2^2)$

param=c($\mu1, \sigma1, \mu2, \sigma2$) and prob=c(p1,1-p1)

Value

`Y`	vector of simulated variables

`msde.fit` *Estimation Of The Random Effects In Mixed Stochastic Differential Equations*

Description

Parametric estimation of the joint density of the random effects $(\alpha_j, \beta_j, \sigma_j)$ in the mixed SDE $dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma_j a(X_j(t))dW_j(t)$.

Usage

```
msde.fit(times, X, model = c("OU", "CIR"), drift.random = c(1, 2),
  drift.fixed = NULL, diffusion.random = 0, diffusion.fixed = NULL,
  mixture = 0, nb.mixt = 1, Niter = 10, discrete = 1)
```

Arguments

`times`	vector of observation times
`X`	matrix of the M trajectories (each row is a trajectory with as much columns as observations)
`model`	name of the SDE: 'OU' (Ornstein-Uhlenbeck) or 'CIR' (Cox-Ingersoll-Ross)
`drift.random`	random effects in the drift: 0 if only fixed effects, 1 if one additive random effect, 2 if one multiplicative random effect or c(1,2) if 2 random effects. Defaults to c(1,2).
`drift.fixed`	NULL if the fixed effect(s) in the drift is (are) estimated, value of the fixed effect(s) otherwise. Default to NULL
`diffusion.random`	default 0, 1 if one random effect in the diffusion, 0 if there is no random effect in the diffusion
`diffusion.fixed`	NULL if the fixed effect in the diffusion is estimated, value of the fixed effect otherwise. Default to NULL
`mixture`	1 if the random effects in the drift follow a mixture distribution, 0 otherwise. Default to 0.
`nb.mixt`	default 1, number of mixture components for the distribution of the random effects in the drift otherwise.
`Niter`	default 10, number of iterations for the EM algorithm if mixture = 1
`discrete`	default 1, 1 for discrete observations, 0 otherwise. If discrete = 0, and diffusion.random = 0, the exact likelihood associated with continuous observations is discretized. If discrete = 1, the likelihood of the Euler scheme of the mixed SDE is computed.

Details

Estimation of the random effects density from M independent trajectories of the SDE (the Brownian motions W_j are independent), with linear drift. The drift includes no, one or two random effects:

if drift.random = 0: $\alpha_j \equiv \alpha$ and $\beta_j \equiv \beta$ are fixed

if drift.random = 1: $\beta_j \equiv \beta$ is fixed and α_j is random

if drift.random = 2: $\alpha_j \equiv \alpha$ is fixed and β_j is random

if drift.random = c(1,2): α_j and β_j are random

The diffusion includes either a fixed effect or a random effect:

if diffusion.random = 0: $\sigma_j \equiv \sigma$ is fixed

if diffusion.random = 1: σ_j is random

If there is no random effect in the diffusion (diffusion.random = 0), the drift random effect follow Gaussian distributions: $\alpha_j, \beta_j \sim N(\mu, \Omega)$. If there is one random effect (σ_j) in the diffusion (diffusion.random = 1), $\sigma_j \sim Gamma(a, \lambda)$, and $\alpha_j, \beta_j | \sigma_j \sim N(\mu, \sigma_j^2 \Omega)$.

Two diffusions are implemented:

the Ornstein-Uhlenbeck model (OU) $a(X_j(t)) = 1$

the Cox-Ingersoll-Ross model (CIR) $a(X_j(t)) = \sqrt{X_j(t)}$

Validation method: For a number of trajectory numj (fixed by the user or randomly chosen) this function simulates Mrep =100 (by default) new trajectories with the value of the estimated random effect. Then it plots on the left graph the Mrep new trajectories $(Xnumj^k(t1), ...Xnumj^k(tN)), k = 1, ...Mrep$ with in red the true trajectory $(Xnumj(t1), ...Xnumj(tN))$. The right graph is a qq-plot of the quantiles of samples $(Xnumj^1(ti), ...Xnumj^{Mrep}(ti))$ for each time ti compared with the uniform quantiles. The outputs of the function are: a matrix `Xnew` dimension Mrepx N+1, vector of quantiles `quantiles` length N and the number of the trajectory for the plot `numj`

Prediction method: (A COMPLETER)

Value

`index`	is the vector of subscript in $1, ..., M$ where the estimation of phi has been done, most of the time $index = 1 : M$
`estimphi`	matrix of estimators of $\phi = \alpha, or \beta, or (\alpha, \beta)$ from the efficient statitics (see UVS), matrix of two lines if drift.random =c(1,2), numerical type otherwise
`estimpsi2`	matrix of estimators of $\psi^2 = \sigma^2$ from the efficient statistics (see UVS), matrix of one line
`gridf`	grid of values for the plots of the random effects distribution in the drift, matrix form
`gridg`	grid of values for the plots of the random effects distribution in the diffusion, matrix form
`estimf`	estimator of the density of ϕ from a kernel estimator from package: stats, function: density. Matrix form: one line if one random effect or square matrix otherwise
`estimg`	estimator of the density of ψ^2. Matrix form: one line if one random effect or square matrix otherwise
`mu`	estimator of the mean of the random effects normal density
`omega`	estimator of the standard deviation of the random effects normal density
`a`	estimated value of the shape of the Gamma distribution
`lambda`	estimated value of the scale of the Gamma distribution
`sigma2`	value of the diffusion coefficient if it is fixed
`bic`	BIC criterium
`aic`	AIC criterium
`model`	initial choice
`drift.random`	initial choice
`diffusion.random`	initial choice
`drift.fixed`	initial choice
`estim.drift.fix`	initial choice
`estim.diffusion.fixed`	initial choice

`discrete`	initial choice
`times`	initial choice
`X`	initial choice

For the 'paramMLmixture' method:

`mu`	estimated value of the mean at each iteration of the algorithm. Niter x N x 2 array.
`omega`	estimated value of the standard deviation at each iteration of the algorithm. Niter x N x 2 array.
`mixt.prop`	estimated value of the mixture proportions at each iteration of the algorithm. Niter x N matrix.
`probindi`	posterior component probabilites. M x N matrix.

References

See Maximum Likelihood Estimation for Stochastic Differential Equations with Random Effects, Delattre, M., Genon-Catalot, V. and Samson, A. *Scandinavian Journal of Statistics 40(2) 2012* **322-343**

Estimation of population parameters in stochastic differential equations with random effects in the diffusion coefficient, Delattre, M., Genon-Catalot, V. and Samson, A. *ESAIM:PS 19 2015* **671-688**

Mixtures of stochastic differential equations with random effects: application to data clustering, Delattre, M., Genon-Catalot, V. and Samson, A. *Journal of Statistical Planning and Inference 173 2016* **109-124**

Parametric inference for discrete observations of diffusion processes with mixed effects, Delattre, M., Genon-Catalot, V. and Laredo, C. *hal-01332630 2016*

Estimation of the joint distribution of random effects for a discretely observed diffusion with random effects, Delattre, M., Genon-Catalot, V. and Laredo, C. *hal-01446063 2017*

Examples

```
## Not run:
# Example 1: one random effect in the drift and one random effect in the diffusion coefficient.

# -- Simulation
M <- 100
Tmax <- 5
N <- 5000
model <- 'OU'
drift.random <- 2
diffusion.random <- 1
drift.fixed <- 0
drift.param <- c(0.5,0.5)
diffusion.param <- c(8,1/2)

sim1 <- msde.sim(M = M, T = Tmax, N = N, model = model, drift.random = drift.random,
diffusion.random = diffusion.random, drift.fixed = drift.fixed,
mixture = 0, drift.param = drift.param, diffusion.param = diffusion.param)
```

```
# -- Estimation

# -----Fixed effect in the drift estimated
res1 <- msde.fit(times = sim1$times, X = sim1$X, model = 'OU', drift.random = 2,
                 diffusion.random = 1, estim.drift.fix = 1, mixture = 0)
summary(res1)
valid(res1)
plot(res1)

# ----- Fixed effect in the drift known and not estimated
res1bis <- msde.fit(times = sim1$times, X = sim1$X, model = 'OU', drift.random = 2,
                    diffusion.random = 1, drift.fixed=0, mixture = 0)
summary(res1bis)

# Example 2: one random effect in the drift and one fixed effect in the diffusion coefficient

# -- Simulation
M <- 100
Tmax <- 5
N <- 5000
model <- 'OU'
diffusion.random <- 0
diffusion.param <- 0.5
drift.random <- 2
drift.fixed <- 10
drift.param <- c(1,sqrt(0.4/4))

sim2 <- msde.sim(M = M, T = Tmax, N = N, model = model, drift.random = drift.random,
diffusion.random = diffusion.random, drift.fixed = drift.fixed,
mixture=0, drift.param = drift.param,
diffusion.param = diffusion.param)

# -- Estimation
res2 <- msde.fit(times = sim2$times, X = sim2$X, model = 'OU', drift.random = 2,
                 diffusion.random = 0, estim.drift.fix = 1, mixture = 0)

summary(res2)
plot(res2)

# Example 3: two random effects in the drift and one random effect in the diffusion coefficient

# -- Simulation
M <- 100
Tmax <- 5
N <- 5000
model <- 'OU'
drift.random <- c(1,2)
diffusion.random <- 1
density.phi <- 'normalnormal'
drift.param <- c(1,0.5,0.5,0.5)
diffusion.param <- c(8,1/2)
```

```
sim3 <- msde.sim(M = M, T = Tmax, N = N, model = model, drift.random = drift.random,
diffusion.random = diffusion.random, mixture = 0,
drift.param = drift.param, diffusion.param = diffusion.param)

# -- Estimation

res3 <- msde.fit(times = sim3$times, X = sim3$X, model = 'OU', drift.random = c(1,2),
                 diffusion.random = 1, mixture = 0)
summary(res3)
plot(res3)

# Example 4: fixed effects in the drift and one random effect in the diffusion coefficient

# -- Simulation
M <- 100
Tmax <- 5
N <- 5000
model <- 'OU'
drift.random <- 0
diffusion.random <- 1
drift.fixed <- c(0,1)
diffusion.param <- c(5,3)

sim4 <- msde.sim(M = M, T = Tmax, N = N, model = model, drift.random = drift.random,
diffusion.random = diffusion.random, drift.fixed = drift.fixed,
diffusion.param = diffusion.param)

# -- Estimation
res4 <- msde.fit(times = sim4$times, X = sim4$X, model = 'OU', drift.random = 0,
                 diffusion.random = 1, mixture = 0, estim.drift.fix = 0,
                 drift.fixed = c(0,0), discrete = 1)

summary(res4)

# Example 5: one fixed effect and one mixture random effect in the drift, and one fixed effect in
# the diffusion coefficient

# -- Simulation
M <- 100
Tmax <- 5
N <- 5000
diffusion.random <- 0
diffusion.param <- 0.1
model <- 'OU'
drift.random <- 1
drift.fixed <- 1
nb.mixt <- 2
mixt.prop <- c(0.5,0.5)
param.ea1 <- c(0.5, 0.25, 1.8, 0.25)
param.ea2 <- c(1, 0.25, 1, 0.25)
drift.param <- param.ea1

sim5 <- msde.sim(M = M, T = Tmax, N = N, model = model, drift.random = drift.random,
```

```
                 diffusion.random = diffusion.random, drift.fixed = drift.fixed,
                 mixture=1, drift.param = drift.param,
              diffusion.param = diffusion.param, nb.mixt = nb.mixt, mixt.prop = mixt.prop)

# -- Estimation
res5 <- msde.fit(times = sim5$times, X = sim5$X, model = 'OU', drift.random = 1,
                 estim.drift.fix = 1, diffusion.random = 0, estim.diffusion.fix = 1,
                 mixture = 1, nb.mixt=2, Niter = 25)

summary(res5)
plot(res5)

## End(Not run)
```

`msde.pred` *Prediction Of Mixed Stochastic Differential Equations Trajectories*

Description

This function proposes to keep two thirds of the data (randomly chosen) to do the parametric estimation of the density of the parameters and of the fixed parameters of model $dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma_j a(X_j(t))dW_j(t)$ using the same method as in the estimationr function `msde.fit` and then to predict new trajectories from the estimated model. The plot reflect the adequation between the last third of the data and the simulated one.

Usage

```
msde.pred(times, X, model = c("OU", "CIR"), drift.random,
  drift.fixed = NULL, diffusion.random = 0, diffusion.fixed = NULL,
  mixture = 0, nb.mixt = 1, Niter = 10, discrete = 1,
  plot.pred = TRUE, level = 0.05, newwindow = FALSE)
```

Arguments

`times`	vector of observation times
`X`	matrix of the M trajectories (each row is a trajectory with as much columns as observations)
`model`	name of the SDE: 'OU' (Ornstein-Uhlenbeck) or 'CIR' (Cox-Ingersoll-Ross)
`drift.random`	random effects in the drift: 1 if one additive random effect, 2 if one multiplicative random effect or c(1,2) if 2 random effects
`drift.fixed`	default NULL, fixed effect in the drift: value of the fixed effect when there is only one random effect and it is not estimated, NULL otherwise
`diffusion.random`	default 0, 1 if one random effect in the diffusion, 0 if there is no random effect in the diffusion

`diffusion.fixed`
: default NULL, fixed effect in the diffusion: value of the fixed effect when there is no random effect in the diffusion and it is not estimated, NULL otherwise

`mixture`
: 1 if the random effects in the drift follow a mixture distribution, 0 otherwise. Default to 0.

`nb.mixt`
: default 1, number of mixture components for the distribution of the random effects in the drift

`Niter`
: default 10, number of iterations for the EM algorithm if mixture = 1

`discrete`
: default 1, 1 for discrete observations, 0 otherwise. If discrete = 0, and diffusion.random = 0, the exact likelihood associated with continuous observations is discretized. If discrete = 1, the likelihood of the Euler scheme of the mixed SDE is computed.

`plot.pred`
: logical(1), if TRUE, the results are depicted grafically

`level`
: alpha for the predicion intervals, default 0.05

`newwindow`
: logical(1), if TRUE, a new window is opened for the plot

Value

`res`
: is the vector of subscript in $1, ..., M$ where the estimation of phi has been done, most of the time $index = 1 : M$

`Xpred`
: is the vector of subscript in $1, ..., M$ where the estimation of phi has been done, most of the time $index = 1 : M$

`indexpred`
: matrix of estimators of $\phi = \alpha, or\beta, or(\alpha, \beta)$ from the efficient statitics (see `UVS`), matrix of two lines if drift.random =c(1,2), numerical type otherwise

`phipred`
: matrix of estimators of $\psi^2 = \sigma^2$ from the efficient statistics (see `UVS`), matrix of one line

References

See Maximum Likelihood Estimation for Stochastic Differential Equations with Random Effects, Delattre, M., Genon-Catalot, V. and Samson, A. *Scandinavian Journal of Statistics 40(2) 2012* **322-343**

Estimation of population parameters in stochastic differential equations with random effects in the diffusion coefficient, Delattre, M., Genon-Catalot, V. and Samson, A. *ESAIM:PS 19 2015* **671-688**

Mixtures of stochastic differential equations with random effects: application to data clustering, Delattre, M., Genon-Catalot, V. and Samson, A. *Journal of Statistical Planning and Inference 173 2016* **109-124**

Parametric inference for discrete observations of diffusion processes with mixed effects, Delattre, M., Genon-Catalot, V. and Laredo, C. *hal-01332630 2016*

Estimation of the joint distribution of random effects for a discretely observed diffusion with random effects, Delattre, M., Genon-Catalot, V. and Laredo, C. *hal-01446063 2017*

Examples

```
## Not run:
# Example 1: one random effect in the drift and one random effect in the diffusion coefficient.

## End(Not run)
```

`msde.sim` *Simulation Of A Mixed Stochastic Differential Equation*

Description

Simulation of M independent trajectories of a mixed stochastic differential equation (SDE) with linear drift

$$dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma_j a(X_j(t))dW_j(t), j = 1, ..., M.$$

There may be two random effects (α_j, β_j) in the drift and one random effect σ_j in the diffusion coefficient.

Usage

```
msde.sim(M, T, N = 100, model, drift.random, diffusion.random, mixture = 0,
  drift.param, diffusion.param, nb.mixt = 1, mixt.prop = 1, t0 = 0,
  X0 = 0.01, delta = T/N, op.plot = 0, add.plot = FALSE)
```

Arguments

`M` number of trajectories.

`T` horizon of simulation.

`N` number of simulation steps, default Tx100.

`model` name of the SDE: 'OU' (Ornstein-Uhlenbeck) or 'CIR' (Cox-Ingersoll-Ross).

`drift.random` random effects in the drift: 0 if no random effect, 1 if one additive random effect, 2 if one multiplicative random effect or c(1,2) if 2 random effects.

`diffusion.random` random effect in the diffusion coefficient: 0 if no random effect, 1 if one multiplicative random effect.

`mixture` 1 if the random effects in the drift follow a mixture of Normal distributions, 0 otherwise. Default to 0.

`drift.param` fixed effects in the drift: value of the fixed effect when there is only one random effect, 0 otherwise. If drift.random =2, fixed can be 0 but β has to be a non negative random variable for the estimation. vector (not mixture) or matrix (mixture) of parameters of the distribution of the random effects in the drift.

`diffusion.param` diffusion parameter if the diffusion coefficient is fixed, vector of parameters of the distribution of the diffusion random effect otherwise.

nb.mixt	number of mixture components if the drift random effects follow a mixture distribution, default nb.mixt=1.
mixt.prop	vector of mixture proportions if the drift random effects follow a mixture distribution, default mixt.prop=1.
t0	time origin, default 0.
X0	initial value of the process, default X0=0.001.
delta	time step of the simulation (T/N).
op.plot	1 if a plot of the trajectories is required, default 0.
add.plot	1 for add trajectories to an existing plot

Details

Simulation of M independent trajectories of the SDE (the Brownian motions W_j are independent), with linear drift. There may be one or two random effects in the drift:

If drift.random = 0, α and β are fixed effects ($\alpha_j \equiv \alpha$ and $\beta_j \equiv \beta$)

If drift.random = 1, β is a fixed effect ($\beta_j \equiv \beta$), and the drift function is written $(\alpha_j - \beta X_j(t))$

If drift.random = 2, α is a fixed effect ($\alpha_j \equiv \alpha$), and the drift function is written $(\alpha - \beta_j X_j(t))$

If drift.random = c(1,2), both effects are random, and the drift function is written $(\alpha_j - \beta_j X_j(t))$

Two diffusions are implemented:

Ornstein-Uhlenbeck model (OU): $a(X_j(t)) = 1$

Cox-Ingersoll-Ross model (CIR): $a(X_j(t)) = \sqrt{X_j(t)}$

There may be either a fixed or a random effect in the diffusion coefficient:

If diffusion.random = 0, σ is a fixed effect ($\sigma_j \equiv \sigma$). In that case, the random effects in the drift follow a Normal distribution or a mixture of Normal distributions.

If diffusion.random = 1, σ_j is a random effect. In that case, σ_j^2 follow an inverse Gamma distribution with parameters diffusion.param=c(shape,scale), and conditional on σ_j, the random effects in the drift follow a Normal distribution with mean mu and variance $Omega * \sigma_j^2$

Value

X	matrix (M x (N+1)) of the M trajectories.
phi	vector (or matrix) of the M simulated random effects.

References

This function mixedsde.sim is based on the package sde, function sde.sim. See Simulation and Inference for stochastic differential equation, S.Iacus, *Springer Series in Statistics 2008* Chapter 2

See Also

http://cran.r-project.org/package=sde

`out` *Transfers the class object to a list*

Description

Method for the S4 classes

Usage

```
out(x)
```

Arguments

`x`	Fit.class or Mixture.fit.class class

`plot,Fit.class,ANY-method` *Plot method for the estimation class object*

Description

Plot method for the S4 class Fit.class

Usage

```
## S4 method for signature 'Fit.class,ANY'
plot(x, newwindow = FALSE, ...)
```

Arguments

`x`	Fit.class class
`newwindow`	logical(1), if TRUE, a new window is opened for the plot
`...`	optional plot parameters

`plot,Mixture.fit.class,ANY-method`
Plot method for the mixture estimation class object

Description

Plot method for the S4 class Mixture.fit.class

Usage

```
## S4 method for signature 'Mixture.fit.class,ANY'
plot(x, newwindow = FALSE, ...)
```

Arguments

`x`	Mixture.fit.class class
`newwindow`	logical(1), if TRUE, a new window is opened for the plot
`...`	optional plot parameters

`probind` *Computation of the component probabilities*

Description

Computation of the individual component probabilities in the mixed SDE

$dXj(t) = (\alpha_j - \beta_j Xj(t))dt + \sigma a(Xj(t))dWj(t)$

with random effects in the drift following a mixture of Gaussian distributions, and fixed effect in the diffusion.

Usage

```
probind(mu, omega, mixt.prop, sigma, U, V, S, K, estimphi, drift.random)
```

Arguments

`mu`	mean of the random effects. N x 2 matrix, first (resp. second) column is the mean of α_j (resp. β_j) in each mixture component if α_j (resp. β_j) is random, the fixed effect value otherwise.
`omega`	standard deviation of the random effects. N x 2 matrix, the components corresponding to a fixed effect should be set to 0.
`mixt.prop`	vector of mixture proportions.
`sigma`	value of the diffusion parameter.
`U`	matrix of M sufficient statistics U (see `UVS`).

`V`	list of the M sufficient statistics matrix V (see UVS).
`S`	vector of the M sufficient statistics S (see UVS).
`K`	number of times of observations.
`estimphi`	matrix of estimators of the fixed/random effects. 2 x M matrix.
`drift.random`	random effects in the drift: 1 if one additive random effect, 2 if one multiplicative random effect or c(1,2) if 2 random effects.

Value

`probindi`	M x N matrix of individual component probabilities.

Mixtures of stochastic differential equations with random effects: application to data clustering, M. Delattre, V. Genon-Catalot and A. Samson, *Journal of Statistical Planning and Inference 2016*, Vol 173, **109–124**

`Q_EM` *Computation of the E-step of the EM algorithm for mixtures of stochastic differential equations with random effects*

Description

Computation of the E-step of the EM algorithm for parameter estimation in the mixed SDE

$dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma a(X_j(t))dW_j(t)$

with random effects in the drift following a mixture of Gaussian distributions, and a fixed effect in the diffusion.

Usage

```
Q_EM(mu, omega, sigma, probindi, U, V, S, K, estimphi, drift.random)
```

Arguments

`mu`	mean of the random effects. N x 2 matrix, first (resp. second) column is the mean of α_j (resp. β_j) in each mixture component if α_j (resp. β_j) is random, the fixed effect value otherwise.
`omega`	standard deviation of the random effects. N x 2 matrix, the components corresponding to a fixed effect should be set to 0.
`sigma`	value of the diffusion parameter.
`probindi`	M x N matrix of individual component probabilites.
`U`	matrix of M sufficient statistics U (see UVS).
`V`	list of the M sufficient statistics matrix V (see UVS).
`S`	vector of the M sufficient statistics S (see UVS).
`K`	number of times of observations.
`estimphi`	matrix of estimators of the fixed/random effects. 2 x M matrix.
`drift.random`	random effects in the drift: 1 if one additive random effect, 2 if one multiplicative random effect or c(1,2) if 2 random effects.

Value

`Q` value of the E-step.

References

Mixtures of stochastic differential equations with random effects: application to data clustering, M. Delattre, V. Genon-Catalot and A. Samson, *Journal of Statistical Planning and Inference 2016*, Vol 173, **109–124**

`summary,Fit.class-method`
Short summary of the results of class object Fit.class

Description

Method for the S4 class Fit.class

Usage

```
## S4 method for signature 'Fit.class'
summary(object)
```

Arguments

`object` Fit.class class

`summary,Mixture.fit.class-method`
Short summary of the results of class object Mixture.fit.class

---

Description

Method for the S4 class Mixture.fit.class

Usage

```
## S4 method for signature 'Mixture.fit.class'
summary(object)
```

Arguments

`object` Mixture.fit.class class

UVS *Computation Of The Sufficient Statistics*

Description

Computation of the sufficient statistics of the (approximate) likelihood of the mixed SDE $dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma_j a(X_j(t))dW_j(t)$.

Usage

```
UVS(X, model, times)
```

Arguments

`X`	matrix of the M trajectories.
`model`	name of the SDE: 'OU' (Ornstein-Uhlenbeck) or 'CIR' (Cox-Ingersoll-Ross).
`times`	times vector of observation times.

Details

Computation of the sufficient statistics of the (approximate) likelihood of the mixed SDE $dX_j(t) = (\alpha_j - \beta_j X_j(t))dt + \sigma_j a(X_j(t))dW_j(t) = (\alpha_j, \beta_j)b(X_j(t))dt + \sigma_j a(X_j(t))dW_j(t)$ with $b(x) = (1, -x)^t$:

U : $U(Tend) = \int_0^{Tend} b(X(s))/a^2(X(s))dX(s)$

V : $V(Tend) = \int_0^{Tend} b(X(s))^2/a^2(X(s))ds$

S : $S(X(t_1), ..., X(t_n)) = 1/delta \sum_{j=1}^{n}(X(t_j) - X(t_{j-1}))^2/a^2(X(t_{j-1}))$

SigDelta: $SigDelta(X(t_1), ..., X(t_n)) = nlog(delta) + \sum_{j=1}^{n} log(a(X(t_j)))$

Value

`U`	vector of the M statistics U(Tend)
`V`	list of the M matrices V(Tend)
`S`	vector of the M quadratic variations S(X(t_1),...,X(t_n))
`SigDelta`	vector of the M constant contributions to the Euler scheme approximation to the likelihood SigDelta(X(t_1),...,X(t_n))

References

See

Maximum Likelihood Estimation for Stochastic Differential Equations with Random Effects, Delattre, M., Genon-Catalot, V. and Samson, A. *Scandinavian Journal of Statistics 40(2) 2012* **322-343**

Estimation of population parameters in stochastic differential equations with random effects in the diffusion coefficient, Delattre, M., Genon-Catalot, V. and Samson, A. *ESAIM:PS 19 2015* **671-688**

Mixtures of stochastic differential equations with random effects: application to data clustering, Delattre, M., Genon-Catalot, V. and Samson, A. *Journal of Statistical Planning and Inference 173 2016* **109-124**

Parametric inference for discrete observations of diffusion processes with mixed effects, Delattre, M., Genon-Catalot, V. and Laredo, C. *hal-01332630 2016*

Estimation of the joint distribution of random effects for a discretely observed diffusion with random effects, Delattre, M., Genon-Catalot, V. and Laredo, C. *hal-01446063 2017*

`valid` *## VALIDATION Validation of the chosen model.*

Description

Validation of the chosen model. For the index numj, Mrep=100 new trajectories are simulated with the value of the estimated random effect number numj. Two plots are given: on the left the simulated trajectories and the true one (red) and one the left the corresponding qq-plot for each time.

Usage

```
valid(x, ...)
```

Arguments

`x`	Fit.class or Mixture.fit.class class
`...`	other optional parameters

`valid,Fit.class-method` *Validation of the chosen model.*

Description

Validation of the chosen model. For the index numj, Mrep=100 new trajectories are simulated with the value of the estimated random effect number numj. Two plots are given: on the left the simulated trajectories and the true one (red) and one the left the corresponding qq-plot for each time.

Usage

```
## S4 method for signature 'Fit.class'
valid(x, Mrep = 100, newwindow = FALSE,
  plot.valid = TRUE, numj, ...)
```

Arguments

`x`	Fit.class class
`Mrep`	number of trajectories to be drawn
`newwindow`	logical(1), if TRUE, a new window is opened for the plot
`plot.valid`	logical(1), if TRUE, the results are depicted grafically
`numj`	optional number of series to be validated
`...`	optional plot parameters

---

valid,Mixture.fit.class-method

Validation of the chosen model for the mixture class.

Description

Validation of the chosen model. For the index numj, Mrep=100 new trajectories are simulated with the value of the estimated random effect number numj. Two plots are given: on the left the simulated trajectories and the true one (red) and one the left the corresponding qq-plot for each time.

Usage

```
## S4 method for signature 'Mixture.fit.class'
valid(x, Mrep = 100, newwindow = FALSE,
  plot.valid = TRUE, numj, ...)
```

Arguments

`x`	Mixture.fit.class class
`Mrep`	number of trajectories to be drawn
`newwindow`	logical(1), if TRUE, a new window is opened for the plot
`plot.valid`	logical(1), if TRUE, the results are depicted grafically
`numj`	optional number of series to be validated
`...`	optional plot parameters

Index