

# Package ‘IDE’

October 12, 2022

**Type** Package

**Title** Integro-Difference Equation Spatio-Temporal Models

**Version** 0.3.1

**Date** 2022-05-30

**Maintainer** Andrew Zammit-Mangion <andrewzm@gmail.com>

**Description** The Integro-Difference Equation model is a linear, dynamical model used to model phenomena that evolve in space and in time; see, for example, Cressie and Wikle (2011, ISBN:978-0-471-69274-4) or Dewar et al. (2009) <[doi:10.1109/TSP.2008.2005091](https://doi.org/10.1109/TSP.2008.2005091)>. At the heart of the model is the kernel, which dictates how the process evolves from one time point to the next. Both process and parameter reduction are used to facilitate computation, and spatially-varying kernels are allowed. Data used to estimate the parameters are assumed to be readings of the process corrupted by Gaussian measurement error. Parameters are fitted by maximum likelihood, and estimation is carried out using an evolution algorithm.

**Imports** methods, ggplot2, Matrix, sp, spacetime, parallel, dplyr, tidy, FRK, DEoptim, stats, utils, sparseinv

**Suggests** knitr

**BugReports** <https://github.com/andrewzm/IDE/issues/>

**Depends** R (>= 3.6.0)

**Encoding** UTF-8

**VignetteBuilder** knitr

**NeedsCompilation** no

**License** GPL (>= 2)

**RoxygenNote** 7.2.0

**Author** Andrew Zammit-Mangion [aut, cre]

**Repository** CRAN

**Date/Publication** 2022-05-30 12:10:14 UTC

## R topics documented:

IDE-package . . . . .	2
coef.IDE . . . . .	2
constant_basis . . . . .	3
IDE . . . . .	3
show_kernel . . . . .	5
simIDE . . . . .	6

<b>Index</b>	<b>8</b>
--------------	----------

---

IDE-package	<i>Integro-difference equation</i>
-------------	------------------------------------

---

### Description

The Integro-Difference Equation model is a linear, dynamical model used to model phenomena that evolve in space and in time. At the heart of the model is the kernel, which dictates how the process evolves from one time point to the next. Both process and parameter reduction are used to facilitate computation, and spatially-varying kernels are allowed. Data used to estimate the parameters are assumed to be readings of the process corrupted by Gaussian measurement error. Parameters are fitted by maximum likelihood, and estimation is carried out using an evolution algorithm.

---

coef.IDE	<i>Retrieve estimated regression coefficients</i>
----------	---------------------------------------------------

---

### Description

Takes an object of class IDE and returns a numeric vector with the estimated regression coefficients.

### Usage

```
## S3 method for class 'IDE'
coef(object, ...)
```

### Arguments

object	object of class IDE
...	currently unused

### See Also

[IDE](#) for more information on how to construct and fit an IDE model.

### Examples

```
SIM1 <- simIDE(T = 5, nobs = 100, k_spat_invariant = 1)
coef(SIM1$IDEmodel)
```

---

constant_basis	<i>Create a single, constant basis function</i>
----------------	-------------------------------------------------

---

**Description**

Constructs an object of class Basis as defined in FRK that is constant over the entire spatial domain.

**Usage**

```
constant_basis()
```

**Value**

Object of class Basis

**See Also**

[IDE](#) for how to use basis functions to construct the IDE kernel

**Examples**

```
basis1 <- constant_basis()
```

---

IDE	<i>Construct IDE object, fit and predict</i>
-----	----------------------------------------------

---

**Description**

The integro-difference equation (IDE) model is constructed using the function `IDE`, fitted using the function `IDE.fit` and used for prediction using the function `predict`.

**Usage**

```
IDE(  
  f,  
  data,  
  dt,  
  process_basis = NULL,  
  kernel_basis = NULL,  
  grid_size = 41,  
  forecast = 0,  
  hindcast = 0  
)  
  
fit.IDE(object, method = "DEoptim", fix = list(), ...)  
  
## S3 method for class 'IDE'  
predict(object, newdata = NULL, covariances = FALSE, ...)
```

**Arguments**

f	R formula relating the dependent variable (or transformations thereof) to covariates
data	data object of class STIDF
dt	object of class difftime identifying the temporal discretisation used in the model
process_basis	object of class Basis as defined in the package FRK
kernel_basis	a list of four objects of class Basis as defined in the package FRK. The first corresponds to the spatial decomposition of the kernel amplitude, the second to the kernel aperture, the third to the kernel horizontal offset, and the fourth to the kernel vertical offset. If left NULL, a spatially-invariant kernel is assumed
grid_size	an integer identifying the number of grid points to use (in one dimension) for numerical integrations
forecast	an integer indicating the number of steps to forecast (where each step corresponds to one difftime)
hindcast	an integer indicating the number of steps to hindcast (where each step corresponds to one difftime)
object	object of class IDE to for fitting or predicting
method	method used to estimate the parameters. Currently only "DEoptim" is allowed, which calls an evolution algorithm from the package DEoptim
fix	list of parameters which are fixed and not estimated (e.g., list(sigma2_eps = 0.01)). Currently only the measurement-error variance (sigma2_eps) can be fixed
...	other parameters passed to DEoptim or predict
newdata	data frame or object of class STIDF containing the spatial and temporal points at which to predict
covariances	a flag indicating whether prediction covariances should be returned or not when predicting

**Details**

The first-order spatio-temporal IDE process model used in the package IDE is given by

$$Y_t(s) = \int_{D_s} m(s, x; \theta_p) Y_{t-1}(x) dx + \eta_t(s); \quad s, x \in D_s,$$

for  $t = 1, 2, \dots$ , where  $m(s, x; \theta_p)$  is a transition kernel, depending on parameters  $\theta_p$  that specify "redistribution weights" for the process at the previous time over the spatial domain,  $D_s$ , and  $\eta_t(s)$  is a time-varying (but statistically independent in time) continuous mean-zero Gaussian spatial process. It is assumed that the parameter vector  $\theta_p$  does not vary with time. In general,  $\int_{D_s} m(s, x; \theta_p) dx < 1$  for the process to be stable (non-explosive) in time.

The redistribution kernel  $m(s, x; \theta_p)$  used by the package IDE is given by

$$m(s, x; \theta_p) = \theta_{p,1}(s) \exp \left( -\frac{1}{\theta_{p,2}(s)} \left[ (x_1 - \theta_{p,3}(s) - s_1)^2 + (x_2 - \theta_{p,4}(s) - s_2)^2 \right] \right),$$

where the spatially-varying kernel amplitude is given by  $\theta_{p,1}(s)$  and controls the temporal stationarity, the spatially-varying length-scale (variance) parameter  $\theta_{p,2}(s)$  corresponds to a kernel scale (aperture) parameter (i.e., the kernel width increases as  $\theta_{p,2}$  increases), and the mean (shift) parameters  $\theta_{p,3}(s)$  and  $\theta_{p,4}(s)$  correspond to a spatially-varying shift of the kernel relative to location  $s$ . Spatially-invariant kernels (i.e., where the elements of  $\theta_p$  are not functions of space) are assumed by default. The spatial dependence, if present, is modelled using a basis-function decomposition.

`IDE.fit()` takes an object of class `IDE` and estimates all unknown parameters, namely the parameters  $\theta_p$  and the measurement-error variance, using maximum likelihood. The only method currently used is the genetic algorithm in the package `DEoptim`. This has been seen to work well on several simulation and real-application studies on multi-core machines.

Once the parameters are fitted, the `IDE` object is passed onto the function `predict()` in order to carry out optimal predictions over some prediction spatio-temporal locations. If no locations are specified, the spatial grid used for discretising the integral at every time point in the data horizon are used. The function `predict` returns a data frame in long format. Change-of-support is currently not supported.

### Value

Object of class `IDE` that contains `get` and `set` functions for retrieving and setting internal parameters, the function `update_alpha` which predicts the latent states, `update_beta` which estimates the regression coefficients based on the current predictions for alpha, and `negloglik`, which computes the negative log-likelihood.

### See Also

[show\\_kernel](#) for plotting the kernel

### Examples

```
SIM1 <- simIDE(T = 5, nobs = 100, k_spat_invariant = 1)
IDEmodel <- IDE(f = z ~ s1 + s2,
               data = SIM1$z_STIDF,
               dt = as.difftime(1, units = "days"),
               grid_size = 41)

#fit_results_sim1 <- fit.IDE(IDEmodel,
#                             parallelType = 1)
#ST_grid_df <- predict(fit_results_sim1$IDEmodel)
```

---

show\_kernel

*Show IDE kernel*

---

### Description

Plotting function for visualising the IDE kernel.

**Usage**

```
show_kernel(IDEmodel, scale = 1)
```

**Arguments**

IDEmodel	object of class IDE
scale	factor by which to scale the arrows when visualising a spatially-varying kernel

**Details**

The function `show_kernel` adapts its behaviour to the type of kernel. If the kernel is spatially-invariant, then the kernel with  $s$  evaluated at the origin is plotted. If spatially-variant, then arrows on a regular grid over the domain are plotted. The direction of the arrow is indicative of the transport direction at a specific location, while the length of the arrow is indicative of the transport intensity.

**See Also**

[IDE](#) for details on the IDE model.

**Examples**

```
SIM1 <- simIDE(T = 5, nobs = 100, k_spat_invariant = 0)
show_kernel(SIM1$IDEmodel)
```

---

 simIDE

---

*Simulate datasets from the IDE model*


---

**Description**

Generates simulations that are then used to evaluate the fitting and prediction of an IDE model.

**Usage**

```
simIDE(T = 9, nobs = 100, k_spat_invariant = 1, IDEmodel = NULL)
```

**Arguments**

T	number of time points to simulate
nobs	number of observations randomly scattered in the domain and fixed for all time intervals
k_spat_invariant	flag indicating whether to simulate using a spatially-invariant kernel or a spatially-variant one
IDEmodel	object of class IDE to simulate form (optional)

**Details**

The domain considered is  $[0,1] \times [0,1]$ , and an IDE is simulated on top of a fixed effect comprising of an intercept, a linear horizontal effect, and a linear vertical effect (all with coefficients 0.2). The measurement-error variance and the variance of the additive disturbance are both 0.0001. When a spatially-invariant kernel is used, the following parameters are fixed:  $\theta_{p,1} = 150$ ,  $\theta_{p,2} = 0.002$ ,  $\theta_{p,3} = -0.1$ , and  $\theta_{p,4} = 0.1$ . See [IDE](#) for details on these parameters. When a spatially-varying kernel is used,  $\theta_{p,1} = 200$ ,  $\theta_{p,2} = 0.002$ , and  $\theta_{p,3}(s), \theta_{p,4}(s)$  are smooth spatial functions simulated on the domain.

**Value**

A list containing the simulated process in `s_df`, the simulated data in `z_df`, the data as STIDF in `z_STIDF`, plots of the process and the observations in `g_truth` and `g_obs`, and the IDE model used to simulate the process and data in `IDEmodel`.

**See Also**

[show\\_kernel](#) for plotting the kernel and [IDE](#)

**Examples**

```
SIM1 <- simIDE(T = 5, nobs = 100, k_spat_invariant = 1)
SIM2 <- simIDE(T = 5, nobs = 100, k_spat_invariant = 0)
```

# Index

coef.IDE, [2](#)

constant\_basis, [3](#)

fit.IDE (IDE), [3](#)

IDE, [2](#), [3](#), [3](#), [6](#), [7](#)

IDE-package, [2](#)

predict.IDE (IDE), [3](#)

show\_kernel, [5](#), [5](#), [7](#)

simIDE, [6](#)