IDL Library for affine-invariant MCMC Ensemble Sampler

API Documentation for idl-emcee

Contents

Part I

Overview

Overview

idl_emcee is an Interactive Data Language (IDL)/GNU Data Language (GDL) implementation of the `affine-invariant Markov chain Monte Carlo (MCMC) ensemble sampler`, based on sl_emcee by M. A. Nowak, an S-Lang/ISIS implementation of the MCMC Hammer proposed by Goodman & Weare (2010), and then implemented in Python (emcee) by Foreman-Mackey et al. (2013).

Dependencies

* This package requires the following packages:

- The IDL Astronomy User's Library

* To get this package with all the dependent packages, you can simply use git command as follows:

```
git clone --recursive https://github.com/mcfit/idl_emcee.git
```

GDL Installation

* The GNU Data Language (GDL) can be installed on

- Linux (Fedora):

```
sudo dnf install gdl
```

- Linux (Ubuntu):

```
sudo apt-get install gnudatalanguage
```

- OS X (brew):

```
brew tap brewsci/science
brew install gnudatalanguage
```

- OS X (macports):

```
sudo port selfupdate
sudo port upgrade libtool
sudo port install gnudatalanguage
```

- Windows: using the GNU Data Language for Win32 (Unofficial Version) or compiling the GitHub source with Visual Studio 2015 as seen in appveyor.yml.

* To setup idl_emcee in GDL, add its path to .gdl_startup in the home directory:

```
!PATH=!PATH + ':/home/idl_emcee/pro/'
```

```
Set ‘‘GDL_STARTUP‘‘ in ‘‘.bashrc‘‘ (bash):
```

```
export GDL_STARTUP=~/.gdl_startup
```

or in .tcshrc (cshrc):

```
setenv GDL_STARTUP ~/.gdl_startup
```

* This package needs GDL version 0.9.8 or later.

IDL Installation

* To install idl_emcee in IDL, add its path to your IDL path. For more information about the path management in IDL, read the IDL path management by Harris Geospatial or the IDL library installation by David Fanning.

* This package needs IDL version 7.1 or later.

Project statistics

Directories:	1
.pro files:	6
.sav files:	0
Routines:	6
Lines:	164

Part II

API

Directory: ./

Overview

emcee_func_erros.pro

EMCEE_FUNC_ERROS

This function returns the uncertainties of the function outputs
based on the confidence level.

```
result = emcee_func_erros(output, mcmc_sim, clevel [, /do_plot])
```

Returns

type=arrays. This function returns uncertainties.

Parameters

output IN REQUIRED TYPE=arrays

the output array returned by the calling function.

mcmc_sim IN REQUIRED TYPE=arrays

the results of the MCMC simulations from emcee_hammer().

clevel IN REQUIRED TYPE=float

the confidence level for the the lower and upper limits.
clevel=0.38292492 ; 0.5-sigma, clevel=0.68268949 ; 1.0-
sigma, clevel=0.86638560 ; 1.5-sigma, clevel=0.90 ; 1.645-
sigma, clevel=0.95 ; 1.960-sigma, clevel=0.95449974 ; 2.0-
sigma, clevel=0.98758067 ; 2.5-sigma, clevel=0.99 ; 2.575-
sigma, clevel=0.99730020 ; 3.0-sigma, clevel=0.99953474 ;
3.5-sigma, clevel=0.99993666 ; 4.0-sigma, clevel=0.99999320
; 4.5-sigma, clevel=0.99999943 ; 5.0-sigma, clevel=0.99999996
; 5.5-sigma, clevel=0.999999998; 6.0-sigma.

Keywords

do_plot IN OPTIONAL TYPE=boolean
set to plot a normalized histogram of the MCMC chain

Examples

For example:

```
IDL> output_error=emcee_func_erros(output, mcmc_sim, clevel)
```

Author

Ashkbiz Danehkar

Copyright

History

15/03/2017, A. Danehkar, IDL code written Adopted from chain_hist() of sl_emcee by M.A. Nowak included in isisscripts

Version

0.1.0

emcee_hammer.pro

EMCEE_HAMMER

This function runs the affine-invariant MCMC Hammer, and returns the MCMC simulations

```
result = emcee_hammer(fcn, input, input_err_m, input_err_p, output, walk_num, iteration_num
    , use_gaussian, FUNCTARGS=parameter)
```

Returns

type=arrays. This function returns the results of the MCMC simulations.

Parameters

fcn IN REQUIRED TYPE=string
the calling function name

input IN REQUIRED TYPE=float
the input parameters array used by the calling function.

input_err_m IN REQUIRED TYPE=float
: the lower limit uncertainty array of the parameters for the calling function.

input_err_p IN REQUIRED TYPE=float
: the upper limit uncertainty array of the parameters for the calling function.

output IN REQUIRED TYPE=arrays
: the output array returned by the calling function.

walk_num IN REQUIRED TYPE=integer
: the number of the random walkers

iteration_num IN REQUIRED TYPE=integer
: the number of the MCMC iteration

use_gaussian IN REQUIRED TYPE=boolean
: if sets to 1, the walkers are initialized as a gaussian over the specified range between the min and max values of each free parameter, otherwise, the walkers are initialized uniformly over the specified range between the min and max values of each free parameter.

Keywords

FUNCTARGS IN TYPE=parameter
: the function arguments (not used for MCMC)

Examples

For example:

```
IDL> mcmc_sim=emcee_hammer('myfunc', input, input_err, output, $
 IDL>                       walk_num, iteration_num, use_gaussian)
```

Author

Ashkbiz Danehkar

Copyright

History

15/03/2017, A. Danehkar, IDL code written Adopted from emcee() of sl_emcee by M.A. Nowak included in isisscripts

01/05/2020, A. Danehkar, function arguments added

Version

0.1.0

emcee_initialize.pro

EMCEE_INITIALIZE

This function returne the initialized walkers for each free parameter.

```
result = emcee_initialize(fcn, param, param_err_m, param_err_p, walk_num, output_num,
   use_gaussian, FUNCTARGS=parameter)
```

Returns

type=arrays. This function returns the initialized walker.

Parameters

fcn IN REQUIRED TYPE=string
: the calling function name

param IN REQUIRED TYPE=arrays
: the input parameters array used by the calling function.

param_err_m IN REQUIRED TYPE=arrays
: the lower limit uncertainty array of the parameters for the calling function.

param_err_p IN REQUIRED TYPE=arrays
: the upper limit uncertainty array of the parameters for the calling function.

walk_num IN REQUIRED TYPE=integer
: the number of the random walkers.

output_num IN REQUIRED TYPE=integer
: the number of the output array returned by the calling function.

use_gaussian IN REQUIRED TYPE=boolean
: if sets to 1, the walkers are initialized as a gaussian over the specified range between the min and max values of each free parameter, otherwise, the walkers are initialized uniformly over the specified range between the min and max values of each free parameter.

Keywords

FUNCTARGS IN TYPE=parameter
: the function arguments (not used for MCMC)

Examples

For example:

```
IDL> x_walk=emcee_initialize(fcn, input, input_err, walk_num, $
 IDL>                        output_num, use_gaussian))
```

Author

Ashkbiz Danehkar

Copyright

This library is released under a GNU General Public License.

History

15/03/2017, A. Danehkar, IDL code written Adopted from emcee() of sl_emcee by M.A. Nowak included in isisscripts

01/05/2020, A. Danehkar, function arguments added

Version

0.1.0

emcee_inv_tot_dist.pro

EMCEE_INV_TOT_DIST

This function returne the inverse Cumulative Distribution Function: 1/sqrt(z) if the random number generator z is between 1/z_a and z_b, is used to generate for a 1/sqrt(z) probability distribution.

```
result = emcee_inv_tot_dist(z, z_a, z_b)
```

Returns

type=arrays. This function returns the lower and higher linear histogram grids (hist_lo, hist_hi)

Parameters

z IN REQUIRED TYPE=float

the a random number generator for the probability distribution 1/sqrt(z).

z_a IN REQUIRED TYPE=float

the inverse lower limit for the random number generator z: 1/z_a <= z.

z_b IN REQUIRED TYPE=float
the higher limit for the random number generator z: z <= b.

Examples

For example:

```
IDL> z = emcee_inv_tot_dist(random_num, adjust_scale_low, adjust_scale_high);
```

Author

Ashkbiz Danehkar

Copyright

This library is released under a GNU General Public License.

History

15/03/2017, A. Danehkar, IDL code written Adopted from icdf() of sl_emcee by M.A. Nowak included in isisscripts

Version

0.1.0

emcee_linear_grid.pro

EMCEE_LINEAR_GRID

This procedure generates a linear grid of histogram bins.

```
emcee_linear_grid, x_min, x_max, nbins, hist_lo, hist_hi
```

Parameters

x_min IN REQUIRED TYPE=float
the lower limit.

x_max IN REQUIRED TYPE=float
the higher limit.

nbins IN REQUIRED TYPE=float
the bins number.

hist_lo OUT REQUIRED TYPE=arrays
returns the lower linear histogram grid,

hist_hi OUT REQUIRED TYPE=arrays
returns the higher linear histogram grid.

Examples

For example:

```
IDL> x_min=1
 IDL> x_max=20
 IDL> nbins=1000
 IDL> emcee_linear_grid, x_min, x_max, nbins, lo, hi
```

Author

Ashkbiz Danehkar

Copyright

This library is released under a GNU General Public License.

History

15/03/2017, A. Danehkar, IDL code written Adopted from
the S-Lang function linear_grid() in isis.

Version

0.1.0

emcee_update_walk.pro

EMCEE_UPDATE_WALK

This function creates the trial walker, examines whether it is acceptable, and returns the updated walker.

```
result = emcee_update_walk(fcn, random_num, x_a, x_b [, FUNCTARGS=parameter])
```

Returns

type=arrays. This function returns the updated walker.

Parameters

fcn IN REQUIRED TYPE=string
the calling function name.

random_num IN REQUIRED TYPE=integer
the random number.

x_a IN REQUIRED TYPE=arrays
the vector of the parameters for a specific walker.

x_b IN REQUIRED TYPE=arrays
the array of the walker parameters.

Keywords

FUNCTARGS IN OPTIONAL TYPE=parameter
the function arguments

Examples

For example:

```
IDL> x_output[j,*]=emcee_update_walk(fcn,a_random[random_num[j],*],$
 IDL>                                array_xwalk,x_walk[*,b_walk])
```

Author

Ashkbiz Danehkar

Copyright

History

15/03/2017, A. Danehkar, IDL code written Adopted from update_walker() of sl_emcee by M.A. Nowak included in isisscripts

01/05/2020, A. Danehkar, function arguments added

Version

0.1.0