Imfit Documentation

Release 1.0

Jörn Werdecker

September 27, 2013

CONTENTS

Contents:

LMFIT REFERENCE

class lmfit.**lmfit**(*func*, *xdata*, *ydata*, *p0*, *yerror=None*, *lm_options={}*, *verbose=True*, *plot=False*, *plot_options={}*)

Class handling non-linear least squares fitting of 2d datasets.

Based on scipy's leastsq function lmfit implements the Levenberg-Marquardt-Algorithm provided by the Fortran MINPACK library.

| | **Parameters** | **func** : function-type |

Testfunction to be fitted.

xdata, ydata: array-like :

Datasets with equal dimensions.

p0 : dict or list

Set of initial parameters. When passing p0 as a list the ordering of the parameters must be the same as in the function definition. So for func = lambda x, a, b: a*x + b either is possible: p0={'a':1, 'b': 2} or p0=[1, 2].

yerror : array-like, optional

Weights for individual data points. Must have the same dimensions as x/ydata arrays.

lm_options : dict, optional

Options passed to scipy.optimize.leastsq. Cf. scipy reference for possible options.

verbose, plot : bool, optional

Toggle verbose output (default: True) and plot window (default: False).

plot_options : dict, optional

Options passed to this classes plot method.

See Also:

scipy.optimize.leastsq Wrapper for MINPACK's fit functions.

Attributes

xdata, ydata		
CovMatr		
fig	class instance	Instance of matplotlib's figure class. Only available after plotting.

Chi2
> Value of Chi^2.

CovMatrix
> The Covariance Matrix.

P
> Dictionary containing the resulting fit parameters.

RMSChi2
> Root mean square value of Chi^2.

Residuals
> Array containing the Residuals.

StdDev
> Dictionary containing the standard deviations of the resulting fit parameters.

__call__ (*x*)
> Evaluates the testfunction at x with the current set of parameters

__weakref__
> list of weak references to the object (if defined)

bootstrap (*n=500*, *plot=False*)
> Performs a bootstrapping analysis of the Residuals.
>
> The Residuals are randomly resampled and superimposed on the fitted testfunction. This artificial dataset is then fitted again and the final parameters are stored. This is repeated n-times. In the end the mean values and standard deviations of the fit parameters from all fits are calculated and returned.

> > **Parameters** **n** : int, optional
> >
> > > Number of bootstrapping runs (default=500).
> >
> > **plot** : bool, optional
> >
> > > Plot all fits (default=False). Be careful! Can be slow for large values of n.
> >
> > **Returns** **Mean** : dict
> >
> > > Mean values of the fit parameters from all bootstrap fits.
> >
> > **StdDev: dict** :
> >
> > > Standard Deviations of the fit parameters determined from all bootstrap fits.
> >
> > **outlist** : dict
> >
> > > List of dictionaries containing all final parameter sets from the bootstrap fits.

fit (*p0*, *lm_options={}*, *verbose=True*, *plot=False*, *plot_options={}*)
> Carries out the non-linear fit.

> > **Parameters** **p0** : dict or list
> >
> > > Set of initial parameters. When passing p0 as a list the ordering of the parameters must be the same as in the function definition. So for func = lambda x, a, b: a*x + b either is possible: p0={'a':1, 'b': 2} or p0=[1, 2].
> >
> > **yerror** : array-like, optional
> >
> > > Weights for individual data points. Must have the same dimensions as x/ydata arrays.

> **lm_options** : dict, optional
>
>> Options passed to scipy.optimize.leastsq. Cf. scipy reference for possible options.
>
> **verbose, plot** : bool, optional
>
>> Toggle verbose output (default: True) and plot window (default: False).
>
> **plot_options** : dict, optional
>
>> Options passed to this classes plot method.
>
> **Raises   FloatingPointError** :
>
>> If the testfunction can be evaluated given the initial parameters.
>
> **Exception** :
>
>> If scipy.optimize.leastsq fails without raising an own exception.

full_results
Dictionary with all results from the fit and additional information.

func
The original testfunction.

plot (*residuals=True*, *acf=True*, *lagplot=True*, *histogramm=True*)
Creates a plot of the data and the test function using current parameters.

> **Parameters   residuals** : bool, optional
>
>> Plot the residuals
>
> **acf** : bool, optional
>
>> Plot the autocorrelogramm
>
> **lagplot** : bool, optional
>
>> Show a lagplot
>
> **histogramm** : bool, optional
>
>> Plot histogramm of residuals

report ()
Prints a report about the results of the last fitting procedure.

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

l

lmfit, 3

PYTHON MODULE INDEX

l

lmfit, 3

INDEX

Symbols

B

C

F

L

P

R

S