

1 Basic Usage

The Alamo Python module facilitates easy access to Alamo through python and provides some additional tools. The basic usage and syntax of the tools is listed below. This module was developed using Python 2, it will not be compatible with Python 3 releases.

- `import alamopy`
 - Import the `alamopy` python module. The packages import by `alamopy.py` include (`numpy,sympy,collections,os,` and `stats`)
- `results = alamopy.doalamo(xdata,zdata,xval*,zval* |options|`
 - This subroutine is the python wrapper for the Alamo executable. It builds a `.alm` file based on the argument passed to `doalamo` and reads the results. This subroutine relies on `.lst` and `.trc` files generated during the call to `alamo`. If options to save these files are specified, or older files with the same name are already present, erroneous behavior can be encountered.
 - `xdata` and `zdata` are `[n, p]` numpy arrays where `n` is the number of data points and `p` is the number of inputs or outputs respectively
 - `xval` and `zval` are validation sets, formatted in the same fashion as the training set, that can be supplied to `doalamo` to pass validation data to Alamo explicitly
 - `|options|` additional options that can be passed to `alamo` through `alamopy` are listed in the next section.
 - `results` is a dictionary that contains the results of the Alamo run. The basic output options are:
 - * `results['model']` - A string representation of the Alamo model
 - * `results['f(model)']` - Lambda functions are used to create a compiled version of the function that will take data in the form of the training data as input
 - * `results['sse']` - The sum of squared residuals reported from Alamo.
 - * `results['xlabels']` and `res['zlabels']` : the labels assigned the the input and output variables. Recorded so the model can be deleted and re-parsed to facilitate the use of pickle.
- `results = alamopy.almconfidence(results,xdata*,zdata*)`
 - This subroutine takes the results of `doalamo` and calculates a variance-covariance matrix for the regression coefficients, and predicts the 95% confidence intervals.
 - If Alamo samples additional data during, it will append the full data set to the results dictionary. If it does not, the `xdata` and `zdata` arrays must be provided explicitly

- results['conf_inv'] contains a list of strings containing the coefficient label, value, and 95% confidence interval, and results['covariance'] contains the variance-covariance matrix associated with the regression coefficients.
- alamopy.almpick(res, 'fname')
- results = alamopy.almunpickle('fname')
 - These two subroutines delete the lambda function in the results dictionary to facilitate pickling to 'fname', and almunpickle re-parses the model to a lambda function in results['f(model)']

2 Options for doalamo

This section contains all of the arguments that can be passed to Alamo through doalamo, and additional arguments that govern the behavior of doalamo. More detailed descriptions of the Alamo options and their effects can be found in the Alamo documentation.

- xlabel - list of strings to be used as the labels for the input variables. Labels x1,x2,... are generated if none are provided
- ylabel - list of strings to be used as the labels for the output variables. Labels z1,z2,... are generated if none are provided
- logfcns, expfcns, cosfcns, sinfcns, linfcns, intercept
 - These are '0-1' options used to activate the associated basis functions.
- monomialpower, multi2power, multi3power, ratiopower
 - List of terms to be used in the respective basis functions
- modeler - Integer 1-7 determines the choice of fitness metric
- solvemip - '0-1' option that will force the solving of the .gms file

2.1 Options specific to doalamo

These options are specific to doalamo and will not change the behavior of the underlying .alm file.

- expandoutput - '0-1' option that can be used to collect more information from the Alamo .lst and .trc files
- showalm - '0-1' option that control if the Alamo output is printed to screen
- alname - A string that will assign the name of the .alm file

- outkeys - '0-1' option that determines if the dictionary is indexed according to the output labels. This option is off by default or 1 response problems, but cannot be turned off for multiresponse problems.
- savetrace - '0-1' option that controls the status of the trace file
- savescratch - '0-1' options that will save the .alm and .lst files
- almopt - A string option that will append a text file of the same name to the end of each .alm file to facilitate advanced user access in an automated fashion.

3 Examples

Four examples are included with the current version of alamopy: linearex.py, sixhumpcamel.py, lim.py, and branin.py. There is also a Jupyter notebook associated with the sixhumpcamel example that can be used to see a demonstration of the tool. All of these examples are built for Linux machines. They can be called from the command line by calling python directly, or can be called from inside a python environment using `execfile()`.