GaussPy Documentation

Release 1.0

Robert Lindner, Carlos Vera-Ciro

February 05, 2016

CONTENTS

Contents:

CHAPTER

ONE

INTRODUCTION

When interpreting data, it is useful to fit models made up of parametrized functions. Gaussian functions, described simply by three parameters and often physically well-motivated, provide a convenient basis set of functions for such a model. However, any set of Gaussian functions is not an orthogonal basis, and therefore any solution implementing them will not be unique. Furthermore, determining fits to spectra involving more than one Gaussian function is complex, and the crucial step of guessing the number of functions and their parameters is not straightforward. Typically, reasonable fits can be determined iteratively and by-eye. However, for large sets of data, analysis by-eye requires an unreasonable amount of processing time and is unfeasible.

This document describes the installation and use of a solution to this problem, called Autonomous Gaussian Decomposition (AGD). AGD is an algorithm designed to automatically decompose spectral line data into Gaussian functions by using computer vision and machine learning techniques to quickly provide optimized initial guesses for the parameters of a multi-component Gaussian model. The speed and adaptability of AGD allow it to interpret large volumes of spectral data efficiently. Although it was initially designed for applications in radio astrophysics, AGD can be used to search for one-dimensional Gaussian (or any other single-peaked spectral profile)-shaped components in any data set.

To decide how many Gaussian functions to include in the model and what their parameters are, AGD uses a technique called derivative spectroscopy. The derivatives of a spectrum can efficiently identify shapes within that spectrum corresponding to the underlying model, including gradients, curvature and edges. The details of this method are described fully in Lindner et al. (2015), AJ, 149, 138.

CHAPTER TWO

INSTALLATION

2.1 Dependencies

- Python 2.7
- Numpy
- Scipy
- h5py
- GNU Scientific Library (GSL)

If you do not already have Python 2.7, you can install the Anaconda Scientific Python distribution, which comes pre-loaded with Numpy, Scipy, and h5py.

To obtain GSL:

```
sudo apt-get install libgsl0-dev
```

2.2 Download GaussPy

Download GaussPy from...

2.3 Installing GaussPy

To install make sure that all dependences are already installed and properly linked to python –python has to be able to load them–. Then cd to the local directory containing gausspy and type

$ python setup.py install

If you don't have root access and/or wish a local installation of gausspy then use

$ python setup.py install –user

change the 'requires' statement in setup.py to include scipy and lmfit

CHAPTER

THREE

QUICKSTART TUTORIAL

After creating the data, we will train the smoothing parameter alpha to the optimal value.

```
import gausspy.GaussianDecomposer as gp

g = gp.GaussianDecomposer()

# Load the training data from the pickle file
g.load_training_data('agd_data.pickle')

# One phase training
g.set('phase', 'one')

# threshold below which Gaussian components will not be fit
g.set('SNR_thresh', 5.)

# Find the optimal alpha value with the training dataset given an initial
# guess for the alpha value
g.train(alpha1_initial = 10.)
```

Now we can see the trained value of alpha by looking at the attribute of our GaussianDecomposer instance.

```
# get the parameters attribute of g, which is a dictionary of important
# variables
print(g.p['alpha1'])
```

CHAPTER FOUR

INDICES AND TABLES

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