gphist Documentation

Release 0.2

David Kirkby

Sep 29, 2017

Contents

The gphist package performs Bayesian inference on the cosmological expansion history using Gaussian process priors. The code is written in python and includes driver programs to run inference calculations and plot the results.

CHAPTER 1

Programs

install

Programs can be run directly from the top-level directory without needing to set *PYTHONPATH* as long as you have the required packages already installed, e.g.:

```
git clone ... gphist
cd gphist
./infer.py --help
```

Required Packages

This package was developed under python 2.7. Please file an issue (or, even better, a PR) if python 3.x support is needed.

The following python packages are required by this package:

- numpy (linalg,random)
- scipy (interpolate,stats)
- matplotlib (pyplot)
- astropy (constants,units,cosmology)

The recommended way to obtain these packages is to install a recent anaconda distribution.

To create a suitable minimal conda environment, use:

```
conda create -n gphist python=2.7 numpy scipy astropy matplotlib
```

infer

Infer the cosmological expansion history using a Gaussian process prior.

Memory Usage

Gaussian process realizations = (8+8+4)*num_evol_samples*num_steps where 8+8+4 combines DH(8) and DA(8) samples and the calculated bin_indices(4).

Histograms = (8+8)*num_bins*num_hist_steps*2**npost where 8+8 combines DH(8) and DA(8) histograms.

The histograms are written to disk, so this is also the output file size.

combine

The combine program reads the ouputs from a set of compatible infererences (produced by *infer*), performs some checks that they are indeed compatible, and generates a single combined output file suitable for use with the *plot* command.

Compatilibity Checks

All inputs must use the same:

- reference cosmology,
- redshift values for histogramming,
- histogram binning,
- number of generated samples,
- hyperparameter grid,
- posteriors (only the names are actually checked).

In addition, inputs must be generate with different initial random states.

Output Format

The output format is a numpy archive (.npz) containing a subset of the arrays written by *infer* and adding an array of -log(P) values where P is the posterior probability.

plot

Plot expansion history inferences.

examples

Quick Demo

Run inference with fixed hyperparameters for quick demonstration:

```
./infer.py --hyper-h 0.1 --hyper-sigma 0.02 --num-samples 100000 --output demo
```

Plot the results with only a CMB posterior applied:

```
./plot.py --input demo.0 --posterior CMB --show --zoom
```

Parallel Inference

Use the multi package to run parallel jobs to calculate the dark-energy evolution with higher statistics at fixed hyperparameters:

```
~/multi/multi --nohup --split 0:10:1 --run "./infer.py --seed NNN --num-samples␣
↪10000000 --output de_NNN --dark-energy"
```

Combine the parallel inferences:

```
./combine.py --input 'de_*' --output de
```

Plot the dark-energy evolution with all posteriors applied:

```
./plot.py --input de --posterior H0-LRG-Lya-CMB --show --dark-energy
```

Full Marginalization Calculation

Run inferences on a grid of hyperparameter values:

```
~/multi/multi --nohup --split 0:400:20 --run "./infer.py --seed NNN --hyper-num-h 20 -
↪-hyper-num-sigma 20 --hyper-index NNN --hyper-count 20 --output hyper20_NNN --num-
↪samples 5000000"
```

Combine inferences to marginalize over hyperparameters:

```
./combine.py --input 'hyper20_*' --output combined20
```

Generate plots:

```
./plot.py --input combined20 --nlp --full --zoom --output plots20/
```

CHAPTER 2

Library Modules

gphist package

Submodules

gphist.analysis module

gphist.cosmology module

gphist.distance module

gphist.evolution module

gphist.posterior module

gphist.process module

Module contents