Cobaya – parameter estimation in Cosmology

Jesús Torrado

CosmoTools'18 – Aachen – 24th April 2018

Astronomy Centre, University of Sussex (soon RWTH Aachen!)

UNIVERSITY OF SUSSEX

erc

European Research Council

Established by the European Commission

So far, we've talked about

- Boltzmann codes $\rightarrow$ compute **observables** $\mathcal{O}$
- Samplers $\rightarrow$ characterise **PDF's**

Connect one and the other to do physics?

Given a model $\mathcal{M} = \mathcal{T} + \mathcal{E}$

- *theoretical* model $\mathcal{T} \Rightarrow$ observable $\mathcal{O}$
- *experimental* model, $\mathcal{E}$: noise, foregrounds...

the probability of some data D given a model

$$\mathcal{L}\left[D \,|\, \mathcal{M}\right] = \mathcal{L}\left[D \,|\, \mathcal{O}, \mathcal{E}\right]$$

Bayesian cosmology

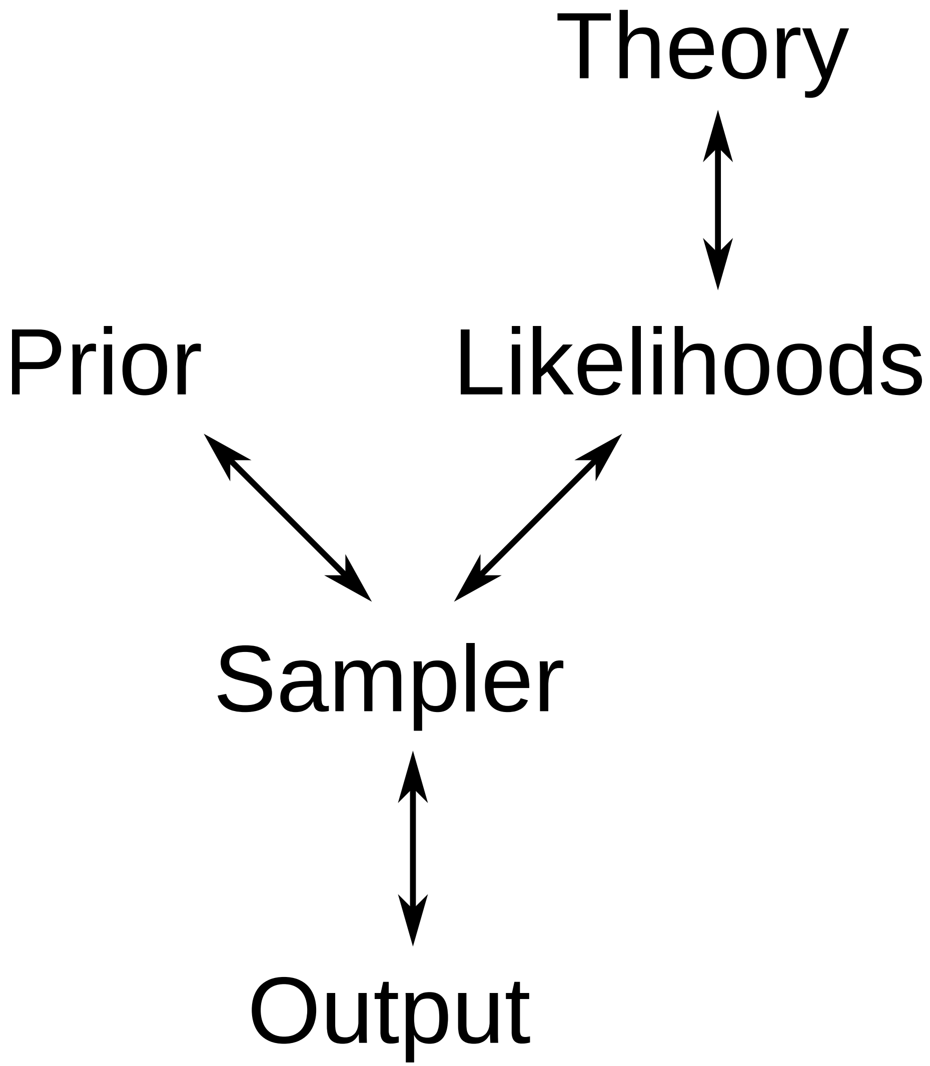
We want *posterior* probabilties of parameters (or models) given the data

$$\mathcal{P}\left[\theta \,|\, \mathcal{D}, \mathcal{M}\right] = \frac{\mathcal{L}\left[\mathcal{D} \,|\, \mathcal{M}(\theta)\right] \, \pi\left(\theta \,|\, \mathcal{M}\right)}{\mathcal{L}\left[\mathcal{D} \,|\, \mathcal{M}\right]}$$

- Choose **model + prior**: $\pi\left(\theta \,|\, \mathcal{M}\right)$
- Choose **dataset + likelihood**: $\mathcal{L}\left(\mathcal{D} \,|\, \ldots\right)$
- Choose a **sampler**: MCMC, PolyChord...

... **and get constraints of data on your model!**

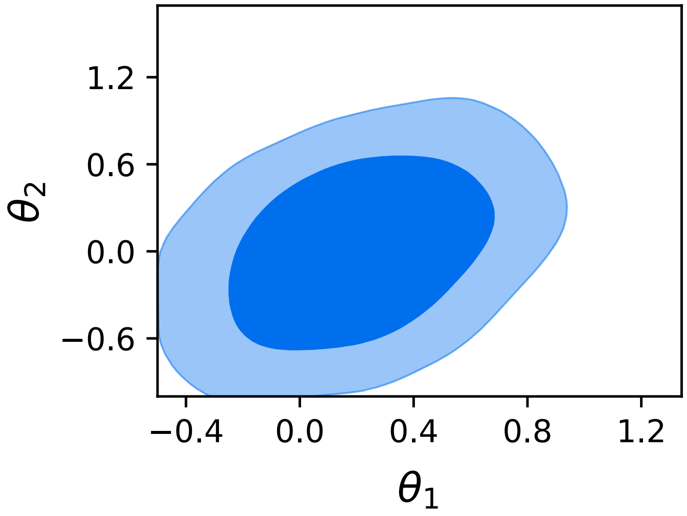
Structure of a Cosmological sampler

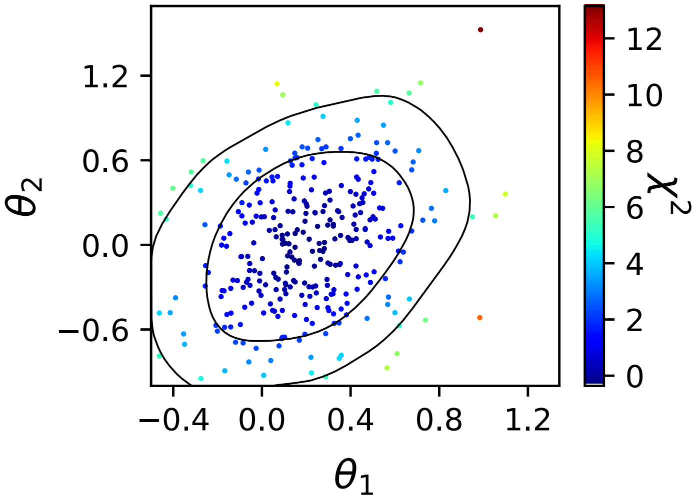


Note on computation time / speeds – I

Computations within the theory code and likelihoods are **non-analytic**, so the probability distributions aren't either! ⇒ we have to sample **numerically** and approximate

$$\mathcal{P} \propto \text{density of samples}$$





Note on computation time / speeds – II

- Theory+Likelihood computation time (e.g. CAMB/CLASS + Planck): ~ 1 sec
- Good contours: ~ 50 evaluations per dimension
- Typical # dimensions: ~ 10 with scaling: #evals$^{\sqrt{d}}$

Total: $\sim \mathcal{O}\left(1 day\right)$

That's why **parallelisation** is an intergral component of a cosmological sampler.

(and why Julien insisted on speed in his talk!)

Cosmological samplers – differences

- Language: Fotran, Python (slower, but a *glue* programming language, makes sense for connecting different codes)
- Compatibility with theory codes and likelihoods
- Design choices (personal tastes/needs): input/output format, physics/statistics focus, etc.
- As Julien said about Boltzmann codes, **it's good to have >1!** (cross testing, different needs for different people/projects)

In particular, cobaya:

- Written in Python, in a modular way
- Pythonic, structured input: dictionaries/YAML
- Statistics-oriented: modeling probabilities is simpler, and more freedom to modify external codes.
  (BUT, you cannot ask Cobaya e.g. for a C_ℓ *directly*.)
- *Dynamical* parameters, priors, likelihoods... No need to touch cobaya's source
- Almost nothing packaged, but easy to install
- Have an idea? Test it ASAP.
- *Don't re-invent the wheel*: yaml, scipy, pandas, getdist...

Currently in **BETA**: missing features, but can do science!

Input example – YAML

```
block:
   option1: value1
   option2: [value21, value22, value23]
   sub_block:
     option3: value3
     option4: value4

# This is a comment
```

The same, in Python:

```
{"block": {
    "option1": value1,
    "option2": [value21, value22, value23],
    "sub_block": {
        "option3": value3,
        "option4": value4
}}}

# This is a comment
```

Input blocks – I

```
theory:
   classy:
     path: [path to classy]  # if using your own
     # [options: values]

likelihood:
   planck_2015_lowTEB:
   planck_2015_plikHM_TTTEEE:
   planck_2015_lensing:
   bicep_keck_2015:
   custom1: "lambda omega_c: np.log(-0.5*(omega_c-0.11)/0.01)**2)"
   custom2: "import_module(my_file).my_function"
```

- Internal likelihoods need just a mention.
- Custom likelihoods (just Python functions) can be defined on the fly or loaded from a file.

Input blocks – II

```
params:
  # Sampled
  omega_c:
    prior:
      min: 0
      max: 0.5
    latex: \omega_c
  H0:
    prior:
      dist: norm
      loc: 70
      scale: 20
  # Fixed
  m_ncdm: 0.06
  # Derived
  YHe:
```

Input blocks – III

```
prior:
  myprior1: "lambda omega_c, H0: np.log(omega_c/H0 < 100)"

sampler:
  # Just one of these!
  mcmc:
    burn_in: 100
    learn_proposal: True
    [...]
  polychord:
    nlive: 100
```

- Custom, multidimensional priors (same as likelihoods)
- Samplers: MCMC (covmat learn, fast-dragging) and PolyChord

Input blocks – IV

Finally, where to put the chains and where to load codes, likelihoods, etc from:

```
output_prefix: [chains_folder]/[chain_prefix]

path_to_modules: [installation folder]
```

Get everything you need for this run:

```
$ cobaya-install [input.yaml] --path [installation folder]
```

1 demo > 1e3 words

- Basic cobaya run – (no cosmo yet)
- Cosmology: input generator
- Citation tool

Exercises – I

FIRST: let's update!

Follow this instructions.

[If PyPI fails, ignore otherwise]

```
$ pip install git+https://github.com/JesusTorrado/getdist/\#egg=getdist --upgrade
$ pip install git+https://github.com/JesusTorrado/cobaya/\#egg=cobaya --upgrade
```

[Easy] Modify the gaussian ring example to sample from $\rho,\ \theta$ directly

Check this instructions.

[Laptop] Neutrino mass hierarchy Bayes ratio: 1703.03425

- Needs PolyChord, instructions here
- Prior $\log_{10}\nu \sim \mathcal{U}[-3, -0.155]$
- Likelihood: mass differences (eqs. 3.1-3.2) and $\sum m_\nu < 0.12$ (68% c.l).
- Sampler? (we want evidences). Elongated pdf: what to tune?
- Reproduce fig. 1

Exercises – II

[Cluster] Features in the primordial spectrum

- Take Matteo's CAMB's modification (or do the same modification in CLASS, if you prefer, using the external_Pk module.
- Generate a LCDM+Planck input file: `$ cobaya-cosmo-generator`
- Add path to theory code, and priors for feature parameters (maybe pop up a python notebook and check for which parameter ranges you can still see the feature in $P(k)$ within $k \in [10^{-4}, 0.1]$.
- Fix LCDM cosmo+nuisance parameters to the best fit (faster)
- Choose a sampler (posterior will likely be multi-modal)
- Analyse the results: confidence intervals for different modes? plots?