Practical Bayesian Inference with PyMultiNest

Johannes Buchner
Max Planck Institute for extraterrestrial Physics
http://astrost.at/istics/
Astrophysics

Understanding individual objects

In wavelength or energy →
In time →
In space →
In multiple →
Light & beyond

Physical model
Poisson data

Chandra X-ray Observatory
Cassiopeia A
Astrophysics

Understanding samples

Correlation Between Black Hole Mass and Bulge Mass

Empirical models, correlations
Physical interpretation
reproduction with galaxy simulations

Blac Hole Mass

Velocity dispersion of stars

Stellar dynamics
Gas dynamics
Masers
Excluded
Elliptical
S0
Spiral
Astrophysics

Understanding underlying populations

Censorship (but understood)

Heavily heterogeneous uncertainties (but understood)

Hierarchical Bayesian Model with censorship (1983)

(N+1)*nested sampling + importance sampling
N* nested sampling + Stan
Galaxy clusters for Dark Energy
3 million Active Galactic Nuclei

• Johannes Buchner
Max Planck Institute for Extraterrestrial Physics
Garching, Munich

@eROSITA_SRG
Agenda

- Choice of tools
- Setting up problems, running
- Use of posterior samples
- Visualisation
- Workflow
- Comparison of model comparison methods (BF vs IC vs prediction-based)
Inference requirements

- Forward modeling with Bayesian inference pretty popular
- Physical models can be complex & expensive Monte Carlo simulations
- "degenerate" parameter constraints
- Sometimes little data
- Multiple solutions
- Mostly 2-15d (or 1e6d)
- Competing physical effects
# Bayesian tools

<table>
<thead>
<tr>
<th>Parameter estimation</th>
<th>Model comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nested sampling (MultiNest)</td>
<td>Nested sampling (MultiNest)</td>
</tr>
<tr>
<td>HMCMC with multiple chains – Stan</td>
<td>open research problem. NS+some MCMC variant. More research needed</td>
</tr>
<tr>
<td>→ posterior samples</td>
<td>→ lnZ</td>
</tr>
</tbody>
</table>

To handle multiple maxima, low state of information, peculiar posterior shapes, numerical likelihoods

(& have a life beyond convergence criteria)

GP: george, celerite
https://arxiv.org/abs/1703.09710
Other tools: emcee, PolyChord
PyMultiNest in practice

- **Inputs**
  - Dimensionality of problem
  - Number of live points (~400-1000)
  - Efficiency parameter 30%
  - Prior transformation function
  - Likelihood function

- **Outputs**
  - Posterior samples (like in MCMC)
  - ln(Z) with uncertainties

In Python
Easy to load data
complex calculations, legacy code
visualize results
automate

https://github.com/JohannesBuchner/PyMultiNest/
Specifying priors

Transform uniform cube to physical parameters

```python
def my_prior_transform(cube):
    # cube is a d-dimensional array
    params = cube.copy()
    # from 0 to 10
    params[0] = cube[0] * 10
    return params
```
Specifying priors

Transform uniform cube to physical parameters

def my_prior_transform(cube):
    # cube is a d-dimensional array
    params = cube.copy()
    # from -10 to +10
    params[0] = cube[0] * 20 - 10
    return params
Specifying priors

Transform uniform cube to physical parameters

```python
def my_prior_transform(cube):
    # cube is a d-dimensional array
    params = cube.copy()
    # from -10 to +10
    params[0] = cube[0] * 20 - 10
    # from 1 to 10
    # via inverse CDF
    params[2] = rv.ppf(cube[2])
    return params

# Gaussian prior 5+-1
rv = scipy.stats.norm(5, 1)
```
Specifying likelihoods

Gaussian example

mydata = numpy.loadtxt("mydata.txt")
x, y, yerr = mydata.transpose()

def my_likelihood(params):
    # params is a d-dimensional array
    # already transformed
    a, b, c, d = params
    # compute model prediction:
    m = (a + numpy.sin(x * c)) * d
    # compute gaussian likelihood
    return (((m - y)/yerr)**2).sum()
Specifying likelihoods

Gaussian example

mydata = numpy.loadtxt("mydata.txt")
x, y, yerr = mydata.transpose()

def my_likelihood(params):
    # params is a d-dimensional array
    # already transformed
    a, b, c, d = params
    # compute model prediction:
    m = (a + numpy.sin(x * c)) * d
    # compute gaussian likelihood
    return (((m – y)/yerr)**2).sum()

result = solve(
    LogLikelihood=my_likelihood,
    Prior=my_prior_transform,
    n_dims=4,
    outputfiles_basename='mysine_')
RadFriends / MultiNest

- **Use existing points** to guess contour
- **Expand contour a little bit**
- **Draw uniformly from contour**
- **Reject points below likelihood threshold**
- **RadFriends**: Compute distance at which every point has a neighbor. Bootstrap (Leave out) for safety.
- **MultiNest clusters and uses ellipses**
Example
Important files

- `<prefix>post_equal_weights.dat`
- `<prefix>stats.dat`
- `multinest_marginals*.py <prefix>`
Model comparison

- Empirical models
  - Information content
- Component presence
  - Regions of practical equivalence
- Physical effects
  - Bayes factors
  - Priors well-justified
  - Monte Carlo simulations (parametric bootstrap)

https://arxiv.org/abs/1506.02273
Betancourt (2015)
Workflows

- Best model
- Parameters go to extremes
- Visualisations
- Thinking
- Add to model comparison set
- New physical model
- AIC
- Visualisations (PPC)
- Visualisations (economics/political science)
- WAIC/LOO CV
- BMC
- Model set
- Best empirical model
- empirical model modification
- (physics)
data {
    int<lower=0> N;
    real[N] x;
    real[N] y;
    real[N] y_err;
}
parameters {
    real a;
    real b;
    real c;
    real d;
}
model {
    for (i in 1:N) {
        real m;
        m <- a*sin(x[i]*b + c) + d;
        y[i] ~ normal(m, y_err[i]);
    }
}
Strengths

- In general robust & reliable
- MPI capable
  
  ```
  $ pip install mpi4py
  $ mpiexec -np 4 python myprog.py
  ```
- Can resume
Failure modes

- estimators \( INS! = NS \)
  - Multiple runs → \( \ln Z \) scatter
  - Check with other techniques (RadFriends, VB+IS)
Creating difficulties

- When ellipsoids approximation is poor
- Multi-modal
- Posterior is heavy-tailed/asymmetric
Future

- Failure modes & tuning of step-based methods
  - slice sampling, NUTS
- Region reconstruction + step-based methods
- Rigorous extensions to dynamic nested sampling
- Review of nested sampling extensions

Stay tuned!
UltraNest3