Nested sampling

An efficient and robust Bayesian inference tool for 21cm cosmology

Will Handley
<wh260@cam.ac.uk>

Royal Society University Research Fellow
Astrophysics Group, Cavendish Laboratory, University of Cambridge
Kavli Institute for Cosmology, Cambridge
Gonville & Caius College
`willhandley.co.uk`

21st October 2020

Bayesian inference in global 21cm cosmology

- Data analysis problem

$$\text{Data} = \text{Signal} + \text{Foreground} + \text{Noise}$$
$$T_{\text{data}}(\nu) = T_{\text{21cm}}(\nu;\theta_{\text{21cm}}) + T_{\text{fg}}(\nu;\theta_{\text{fg}}) + T_{\text{noise}}(\nu)$$

- We can only statistically describe T_{noise}
  - e.g. as an (un)correlated Gaussian random variable $P(T_{\text{noise}}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-T_{\text{noise}}^2/2\sigma^2}$
- Allows us to form a *likelihood* for the unknown parameters $\theta = (\theta_{\text{21cm}}, \theta_{\text{fg}})$

$$\mathcal{L}(\theta) = P(\text{Data}|\theta) \propto e^{-\frac{1}{2}\chi(\theta)^2}$$
$$\chi^2(\theta) = \sum_\nu \frac{1}{\sigma_\nu^2}[T_{\text{data}}(\nu) - T_{\text{21cm}}(\nu;\theta_{\text{21cm}}) - T_{\text{fg}}(\nu;\theta_{\text{fg}})]^2$$

- This misses the initial step of data compression (reduction, flagging, integration)
- In practice also include θ_{noise} as parameters as well.

The three pillars of Bayesian inference

- Frequentist approaches maximise the likelihood $\mathcal{L}(\theta) = P(D|\theta)$
  - anything using a "least squares" optimisation (e.g. `maxsmooth`, EDGES calibration)
- Bayes theorem allows us to answer science questions using probabilities:

Parameter estimation Given a model, which range of parameters best describe the data?

$$P(\theta|D,M) = \frac{P(\theta|D,M)P(\theta|M)}{P(D|M)}, \qquad \text{Posterior} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Evidence}}$$

Model comparison Which models do the data prefer?

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)}, \qquad \text{Model Probability} = \frac{\text{Evidence} \times \text{Model Prior}}{\text{Normalisation}}$$

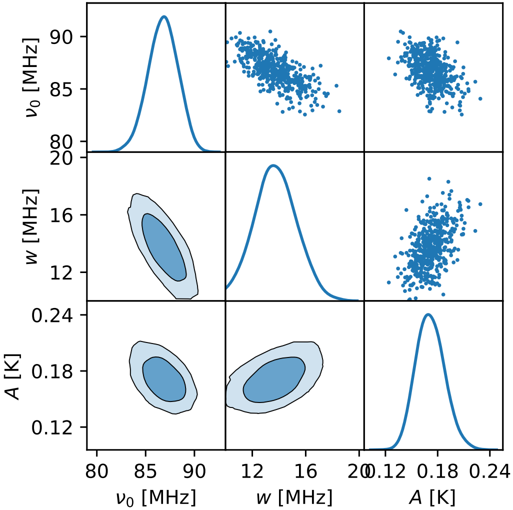
Tension quantification Are different (sub)sets of data consistent with one another?

Model comparison in 21cm cosmology

- Model comparison: Given this data, what odds would a bookmaker put on a model?
- In general in 21cm cosmology model takes the form $T_{\text{data}} = T_{\text{21cm}} + T_{\text{fg}} + T_{\text{noise}}$
- Model comparison can be applied to any combination of these pieces

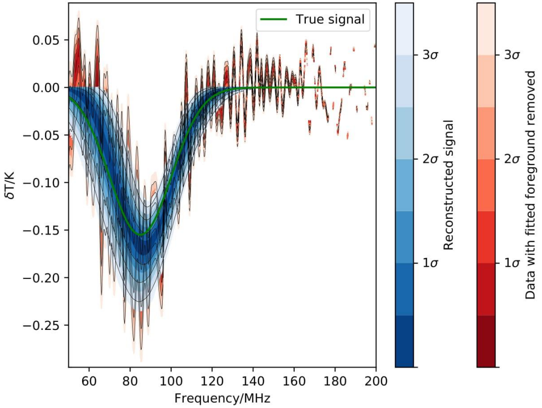
1. Does the data contain a signal?
   (Does $T_{\text{data}} = T_{\text{21cm}} + T_{\text{fg}} + T_{\text{noise}}$ have a higher evidence than $T_{\text{data}} = T_{\text{fg}} + T_{\text{noise}}$)
2. Which foreground model is better? (polynomial vs forward sky model)
3. Which noise model is better? (correlated gaussian, uncorrelated gaussian, lorentzian)
4. How many components should a foreground model have?
5. How many components should a calibration model have?

- The key concept in numerical Bayesian inference is *sampling* a distribution
- i.e. drawing parameter vectors θ in proportion to the probability density $P(\theta)$.
- Compression of high-dimensional function.
- Posterior $P(\theta|D)$ encodes multidimensional generalisation of error bars
- Sampling is traditionally accomplished using random-walk MCMC tools like Gibbs sampling, Metropolis-Hastings, emcee or Hamiltonian Monte Carlo.

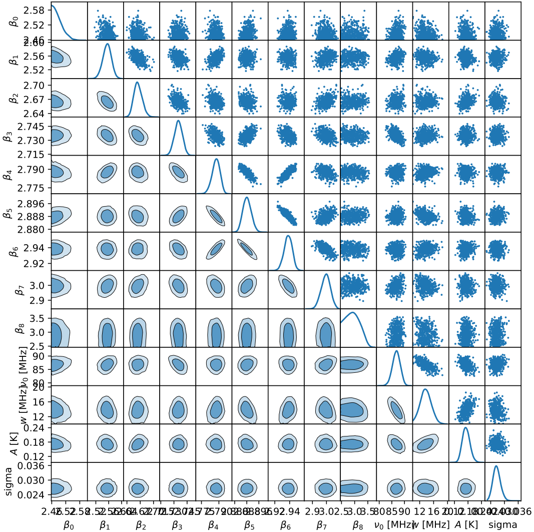


Numerical tools

- The key concept in numerical Bayesian inference is *sampling* a distribution
- i.e. drawing parameter vectors θ in proportion to the probability density $P(\theta)$.
- Compression of high-dimensional function.
- Posterior $P(\theta|D)$ encodes multidimensional generalisation of error bars
- Sampling is traditionally accomplished using random-walk MCMC tools like Gibbs sampling, Metropolis-Hastings, emcee or Hamiltonian Monte Carlo.



- The key concept in numerical Bayesian inference is *sampling* a distribution
- i.e. drawing parameter vectors θ in proportion to the probability density $P(\theta)$.
- Compression of high-dimensional function.
- Posterior $P(\theta|D)$ encodes multidimensional generalisation of error bars
- Sampling is traditionally accomplished using random-walk MCMC tools like Gibbs sampling, Metropolis-Hastings, emcee or Hamiltonian Monte Carlo.



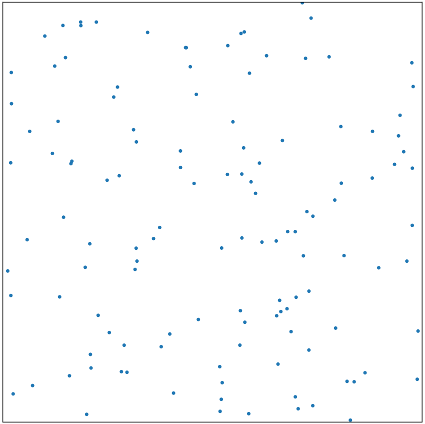
Nested sampling in action

- Nested sampling is a completely different way of sampling.
- Uses ensemble sampling to compress prior to posterior.
- Maintain a set S of n samples, which are sequentially updated:

  S_0: Generate n samples uniformly over the space (from the prior π).

  S_{n+1}: Delete the lowest likelihood sample in S_n, and replace it with a new uniform sample with higher likelihood

- Requires one to be able to uniformly within a region, subject to a *hard likelihood constraint*.



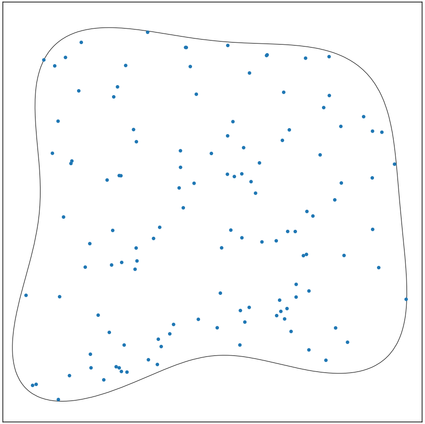
Nested sampling in action

- Nested sampling is a completely different way of sampling.
- Uses ensemble sampling to compress prior to posterior.
- Maintain a set S of n samples, which are sequentially updated:

  S_0: Generate n samples uniformly over the space (from the prior π).

  S_{n+1}: Delete the lowest likelihood sample in S_n, and replace it with a new uniform sample with higher likelihood

- Requires one to be able to uniformly within a region, subject to a *hard likelihood constraint*.

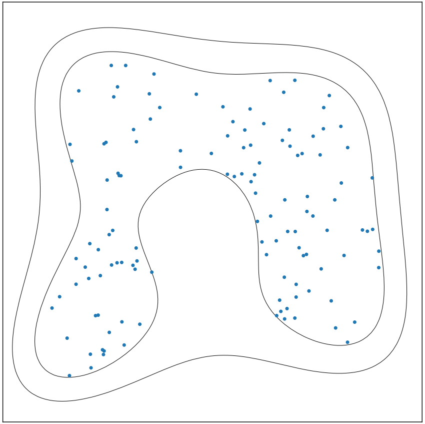


Nested sampling in action

- Nested sampling is a completely different way of sampling.
- Uses ensemble sampling to compress prior to posterior.
- Maintain a set S of n samples, which are sequentially updated:

  S_0: Generate n samples uniformly over the space (from the prior π).

  S_{n+1}: Delete the lowest likelihood sample in S_n, and replace it with a new uniform sample with higher likelihood
- Requires one to be able to uniformly within a region, subject to a *hard likelihood constraint*.

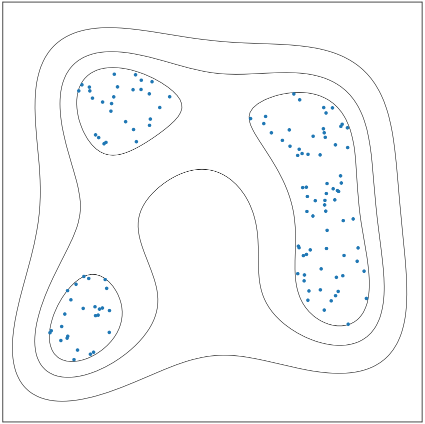


Nested sampling in action

- Nested sampling is a completely different way of sampling.
- Uses ensemble sampling to compress prior to posterior.
- Maintain a set S of n samples, which are sequentially updated:

  S_0: Generate n samples uniformly over the space (from the prior π).

  S_{n+1}: Delete the lowest likelihood sample in S_n, and replace it with a new uniform sample with higher likelihood
- Requires one to be able to uniformly within a region, subject to a *hard likelihood constraint*.



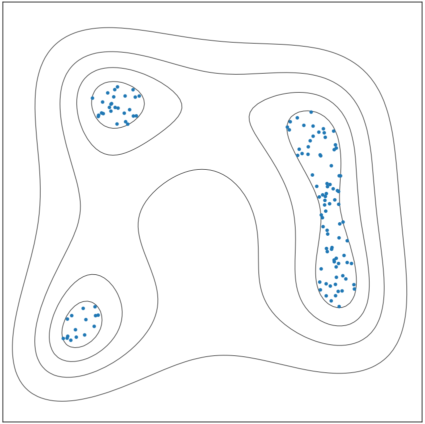
Nested sampling in action

- Nested sampling is a completely different way of sampling.
- Uses ensemble sampling to compress prior to posterior.
- Maintain a set S of n samples, which are sequentially updated:

  S_0: Generate n samples uniformly over the space (from the prior π).

  S_{n+1}: Delete the lowest likelihood sample in S_n, and replace it with a new uniform sample with higher likelihood

- Requires one to be able to uniformly within a region, subject to a *hard likelihood constraint*.



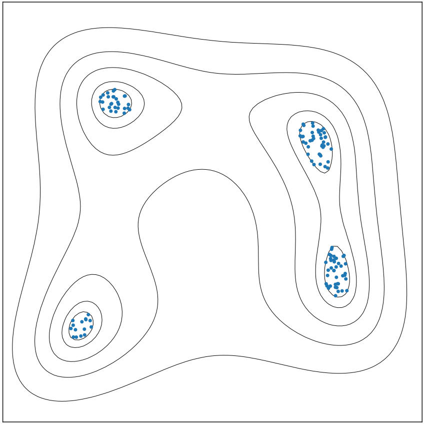
Nested sampling in action

- Nested sampling is a completely different way of sampling.
- Uses ensemble sampling to compress prior to posterior.
- Maintain a set S of n samples, which are sequentially updated:

  S_0: Generate n samples uniformly over the space (from the prior π).

  S_{n+1}: Delete the lowest likelihood sample in S_n, and replace it with a new uniform sample with higher likelihood

- Requires one to be able to uniformly within a region, subject to a *hard likelihood constraint*.



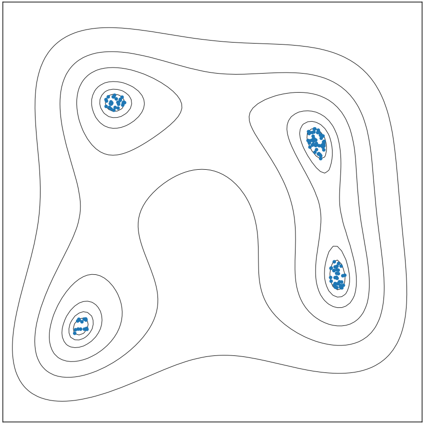
Nested sampling in action

- Nested sampling is a completely different way of sampling.
- Uses ensemble sampling to compress prior to posterior.
- Maintain a set S of n samples, which are sequentially updated:

  S_0: Generate n samples uniformly over the space (from the prior π).

  S_{n+1}: Delete the lowest likelihood sample in S_n, and replace it with a new uniform sample with higher likelihood

- Requires one to be able to uniformly within a region, subject to a *hard likelihood constraint*.



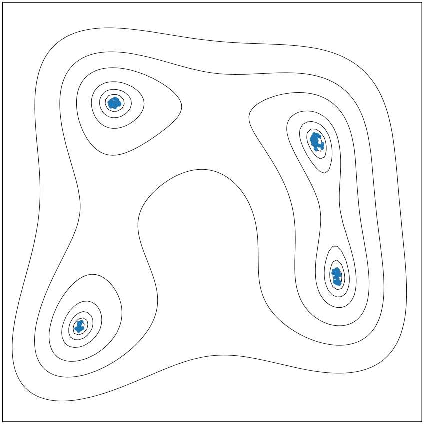
Nested sampling in action

- Nested sampling is a completely different way of sampling.
- Uses ensemble sampling to compress prior to posterior.
- Maintain a set S of n samples, which are sequentially updated:

  S_0: Generate n samples uniformly over the space (from the prior π).

  S_{n+1}: Delete the lowest likelihood sample in S_n, and replace it with a new uniform sample with higher likelihood

- Requires one to be able to uniformly within a region, subject to a *hard likelihood constraint*.



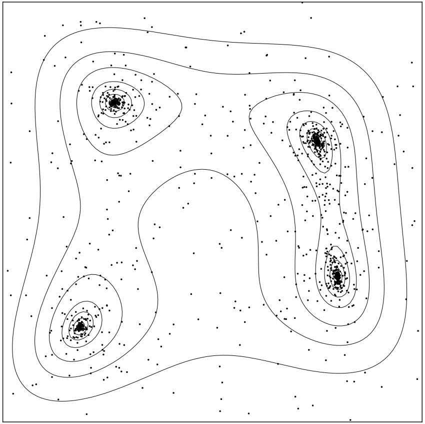
Nested sampling in action

- Nested sampling is a completely different way of sampling.
- Uses ensemble sampling to compress prior to posterior.
- Maintain a set S of n samples, which are sequentially updated:

  S_0: Generate n samples uniformly over the space (from the prior π).

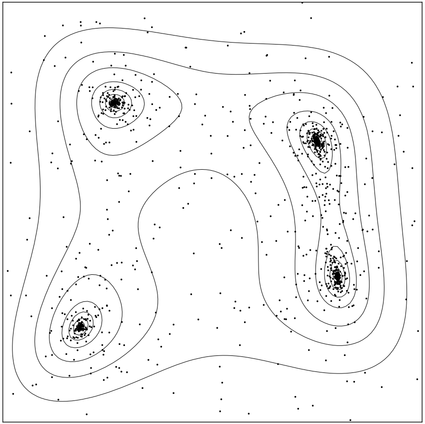
  S_{n+1}: Delete the lowest likelihood sample in S_n, and replace it with a new uniform sample with higher likelihood

- Requires one to be able to uniformly within a region, subject to a *hard likelihood constraint*.



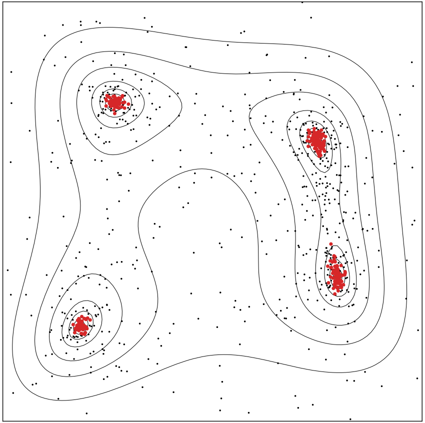
Nested sampling in action

- At the end, one is left with a set of discarded points
- These may be weighted to form posterior samples
- They can also be used to calculate the evidence
- The evolving ensemble of live points allows algorithms to perform self-tuning and mode clustering



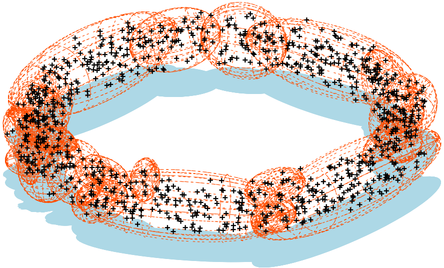
Nested sampling in action

- At the end, one is left with a set of discarded points
- These may be weighted to form posterior samples
- They can also be used to calculate the evidence
- The evolving ensemble of live points allows algorithms to perform self-tuning and mode clustering

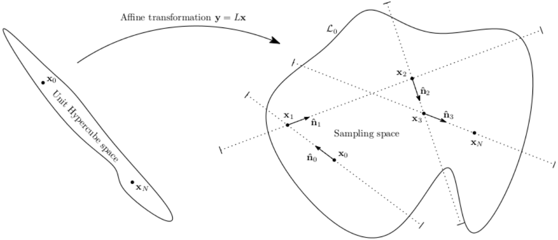


Implementations of Nested Sampling

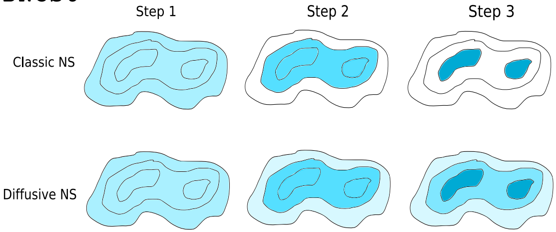
MultiNest



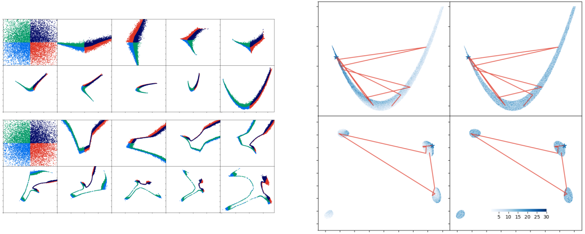
PolyChord



DNest



NeuralNest



Nested Sampling: Benefits and drawbacks

Relative to traditional numerical posterior samples (Metropolis Hastings, HMC, emcee), nested sampling:

+ Can calculate evidence (and therefore perform model comparison).
+ Can handle multi-modal distributions.
+ Requires little tuning for an a-priori unseen problem.
+ Highly parallelisable ($n_{\mathrm{cores}} \sim n_{\mathrm{live}} \gg 4$).
− Slower than a well-tuned posterior sampler.
− Run time is dependent on prior choice, and priors must be proper (some people view this as a feature rather than a bug).

1. Nested sampling is a likelihood scanner, rather than posterior explorer.
   - This means typically most of its time is spent on burn-in rather than posterior sampling
   - Changing the stopping criterion from 10^{-3} to 0.5 does little to speed up the run, but can make results very unreliable
2. The number of live points n_live is a resolution parameter.
   - Run time is linear in n_live, posterior and evidence accuracy goes as $\frac{1}{\sqrt{n_\mathrm{live}}}$.
   - Set low for exploratory runs $\sim\mathcal{O}(10)$ and increased to $\sim\mathcal{O}(1000)$ for production standard.
3. Most algorithms come with additional reliability parameter(s).
   - e.g. `MultiNest`: eff, `PolyChord`: n_repeats
   - These are parameters which have no gain if set too conservatively, but increase the reliability
   - Check that results do not degrade if you reduce them from defaults, otherwise increase.

Key tools

`anesthetic`	Nested sampling post processing [1905.04768]
`insertion`	cross-checks using order statistics [2006.03371] `github.com/williamjameshandley/anesthetic`
`nestcheck`	cross-checks using unthreaded runs [1804.06406] `github.com/ejhigson/nestcheck`
`MultiNest`	Ellipsoidal rejection sampling [0809.3437] `github.com/farhanferoz/MultiNest`
`PolyChord`	Python/C++/Fortran state of the art [1506.00171] `github.com/PolyChord/PolyChordLite`
`dynesty`	Python re-implementation of several codes [1904.02180] `github.com/joshspeagle/dynesty`