A Statistician's Guide to the Galaxy (Fitting Zoo)

An Introduction to the Statistical Foundations of SED Fitting

Will Handley
<wh260@cam.ac.uk>

Royal Society University Research Fellow
Institute of Astronomy, University of Cambridge
Kavli Institute for Cosmology, Cambridge
Gonville & Caius College
willhandley.co.uk/talks

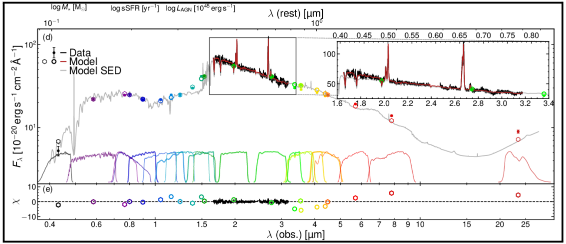
8 July 2025

The Goal: From Photons to Physics

Why SED fitting is a statistical inference problem

The Data D

We observe a galaxy's light (photometry or spectra) across different wavelengths. This is our dataset, D.



The Model $\theta|M$

We want to infer the underlying physical properties, our parameters, θ: Stellar Mass (M_*), Star Formation History (SFH), Dust content (A_V), Metallicity (Z), ...

The Challenge

The parameter space is often:

- **High-dimensional**: Many parameters to fit.
- **Degenerate**: Different combinations of parameters can produce similar SEDs.

The Language of Inference:
How we quantify what we learn from data

Prior $\pi(\theta)$

What we believe about the parameters *before* we see the data. Our physical assumptions.

Evidence $\mathcal{Z}(D)$

How we update our belief in the model using the data.

$$\underbrace{\mathcal{P}(\theta|D)}_{\text{Posterior}} = \frac{\overbrace{\mathcal{L}(D|\theta)}^{\text{Likelihood}} \times \overbrace{\pi(\theta)}^{\text{Prior}}}{\underbrace{\mathcal{Z}(D)}_{\text{Evidence}}}$$

Likelihood $\mathcal{L}(D|\theta)$

How we update our belief in the parameters using the data.

Posterior $\mathcal{P}(\theta|D)$

What we know about the parameters *after* seeing the data. It's our updated state of knowledge.

The Simplest Approach: Optimization (e.g., χ^2 Minimization)

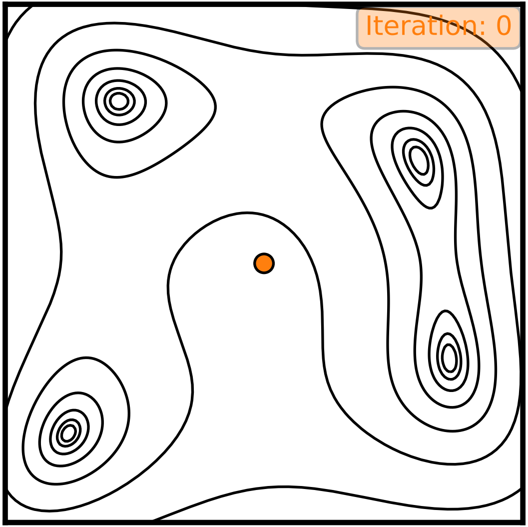
How it Works: Hill Climbing

Imagine the parameter space is a landscape where lower χ^2 (or higher likelihood) is "downhill".

- Start somewhere.
- Follow the steepest gradient downhill.
- Stop when you reach the bottom of a valley.

Advantages

- **Fast** and computationally cheap.
- Good for a quick first look.



The Simplest Approach: Optimization (e.g., χ^2 Minimization)

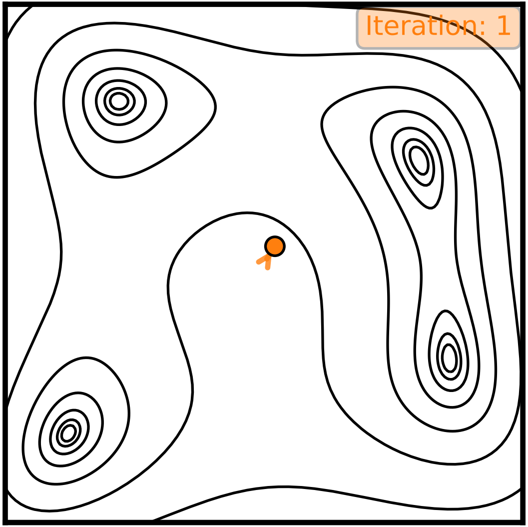
How it Works: Hill Climbing

Imagine the parameter space is a landscape where lower χ^2 (or higher likelihood) is "downhill".

- Start somewhere.
- Follow the steepest gradient downhill.
- Stop when you reach the bottom of a valley.

Advantages

- **Fast** and computationally cheap.
- Good for a quick first look.



The Simplest Approach: Optimization (e.g., χ^2 Minimization)

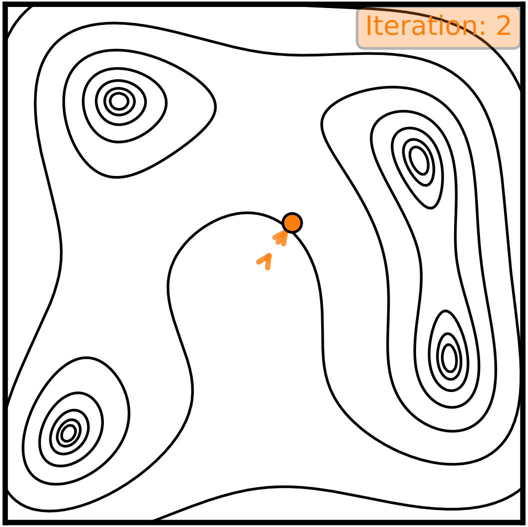
How it Works: Hill Climbing

Imagine the parameter space is a landscape where lower χ^2 (or higher likelihood) is "downhill".

- Start somewhere.
- Follow the steepest gradient downhill.
- Stop when you reach the bottom of a valley.

Advantages

- **Fast** and computationally cheap.
- Good for a quick first look.



The Simplest Approach: Optimization (e.g., χ^2 Minimization)

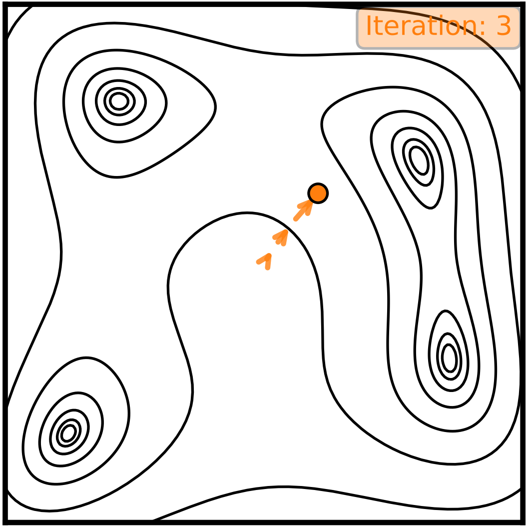
How it Works: Hill Climbing

Imagine the parameter space is a landscape where lower χ^2 (or higher likelihood) is "downhill".

- Start somewhere.
- Follow the steepest gradient downhill.
- Stop when you reach the bottom of a valley.

Advantages

- **Fast** and computationally cheap.
- Good for a quick first look.



The Simplest Approach: Optimization (e.g., χ^2 Minimization)

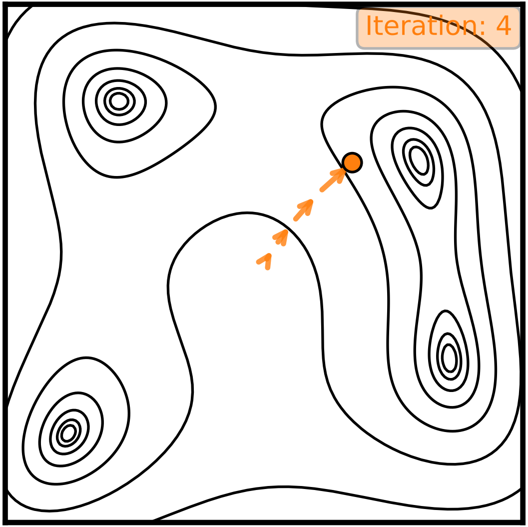
How it Works: Hill Climbing

Imagine the parameter space is a landscape where lower χ^2 (or higher likelihood) is "downhill".

- Start somewhere.
- Follow the steepest gradient downhill.
- Stop when you reach the bottom of a valley.

Advantages

- **Fast** and computationally cheap.
- Good for a quick first look.



The Simplest Approach: Optimization (e.g., χ^2 Minimization)

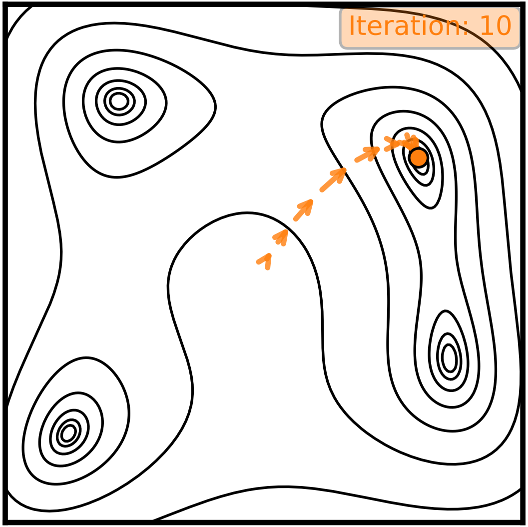
How it Works: Hill Climbing

Imagine the parameter space is a landscape where lower χ^2 (or higher likelihood) is "downhill".

- Start somewhere.
- Follow the steepest gradient downhill.
- Stop when you reach the bottom of a valley.

Advantages

- **Fast** and computationally cheap.
- Good for a quick first look.



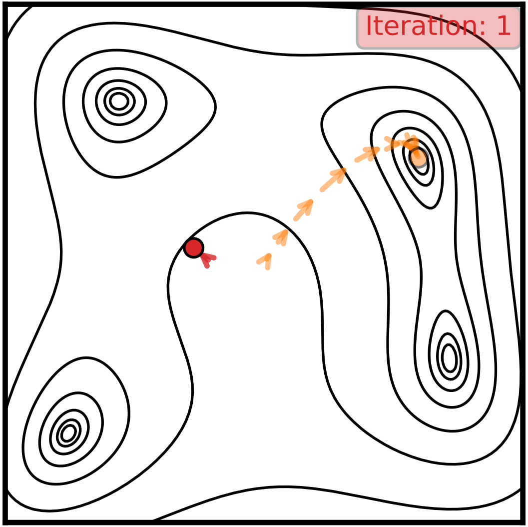
The Simplest Approach: Optimization (e.g., χ^2 Minimization)

Limitations

- Only gives a single **point estimate** (the "best fit").
- **No uncertainty quantification!** Where are the error bars?
- Can easily get stuck in a **local minimum**, missing the true global best fit.

Key Message

Optimization is fast but gives an incomplete and potentially misleading picture. Science needs error bars.



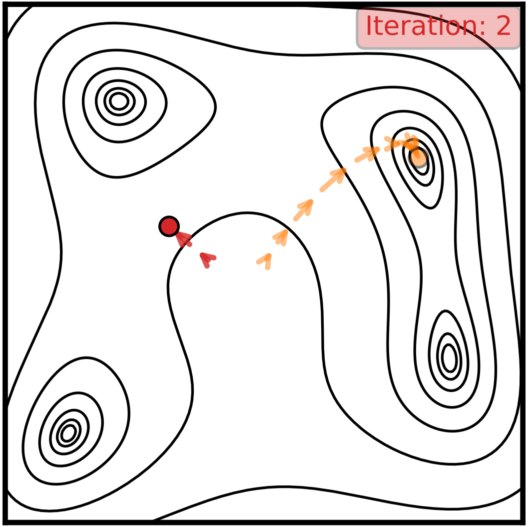
The Simplest Approach: Optimization (e.g., χ^2 Minimization)

Limitations

- Only gives a single **point estimate** (the "best fit").
- **No uncertainty quantification!** Where are the error bars?
- Can easily get stuck in a **local minimum**, missing the true global best fit.

Key Message

Optimization is fast but gives an incomplete and potentially misleading picture. Science needs error bars.



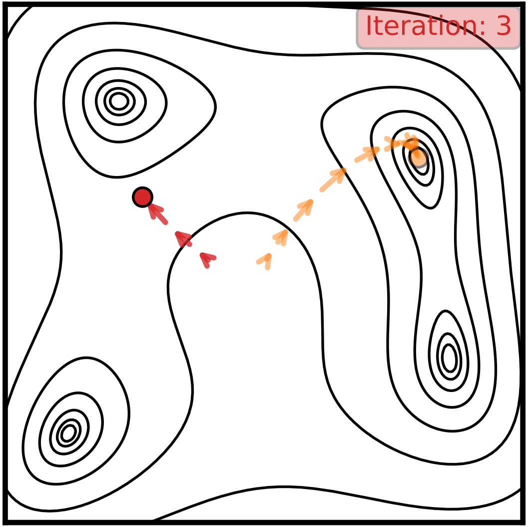
The Simplest Approach: Optimization (e.g., χ^2 Minimization)

Limitations

- Only gives a single **point estimate** (the "best fit").
- **No uncertainty quantification!** Where are the error bars?
- Can easily get stuck in a **local minimum**, missing the true global best fit.

Key Message

Optimization is fast but gives an incomplete and potentially misleading picture. Science needs error bars.



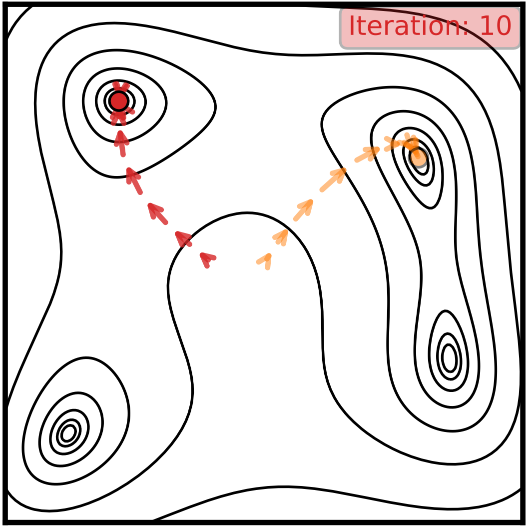
The Simplest Approach: Optimization (e.g., χ^2 Minimization)

Limitations

- Only gives a single **point estimate** (the "best fit").
- **No uncertainty quantification!** Where are the error bars?
- Can easily get stuck in a **local minimum**, missing the true global best fit.

Key Message

Optimization is fast but gives an incomplete and potentially misleading picture. Science needs error bars.

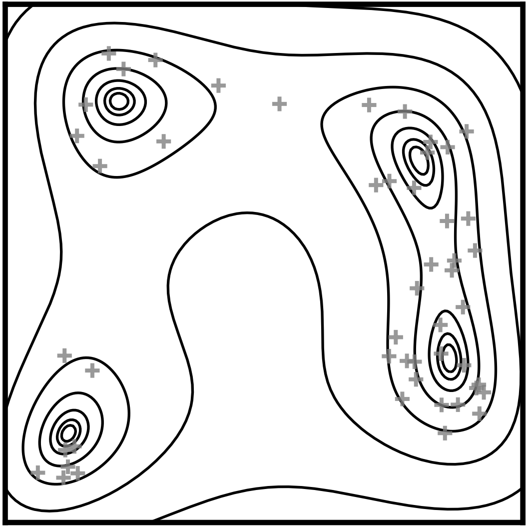


Why do sampling?

- The cornerstone of numerical Bayesian inference is working with **samples**.
- Generate a set of representative parameters drawn in proportion to the posterior $\theta \sim \mathcal{P}$.
- The magic of marginalisation $\Rightarrow$ perform usual analysis on each sample in turn.
- The golden rule is **stay in samples** until the last moment before computing summary statistics/triangle plots because

$$\boxed{f(\langle X \rangle) \ne \langle f(X) \rangle}$$

- Generally need $\sim \mathcal{O}(12)$ independent samples to compute a value and error bar.

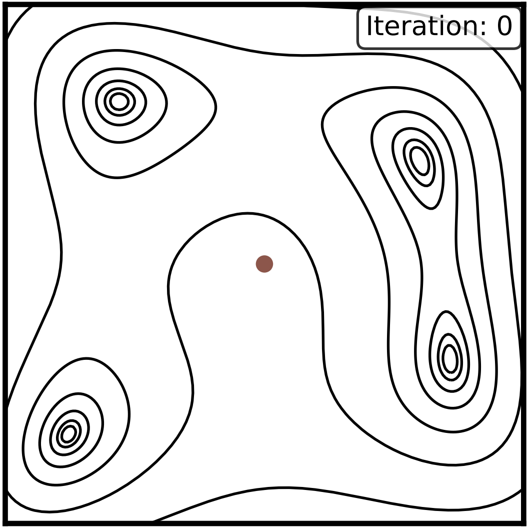


The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

How it Works (Metropolis-Hastings)

Imagine a "walker" exploring the parameter landscape.

1. Take a random step to a new position.
2. If the new spot is "higher" (better likelihood), move there.
3. If it's "lower", maybe move there anyway (with probability proportional to how much lower it is).
4. Repeat millions of times. The path the walker takes traces the posterior distribution.

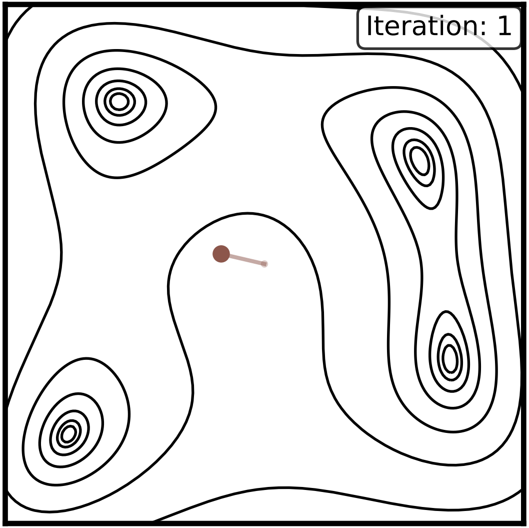


The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

How it Works (Metropolis-Hastings)

Imagine a "walker" exploring the parameter landscape.

1. Take a random step to a new position.
2. If the new spot is "higher" (better likelihood), move there.
3. If it's "lower", maybe move there anyway (with probability proportional to how much lower it is).
4. Repeat millions of times. The path the walker takes traces the posterior distribution.

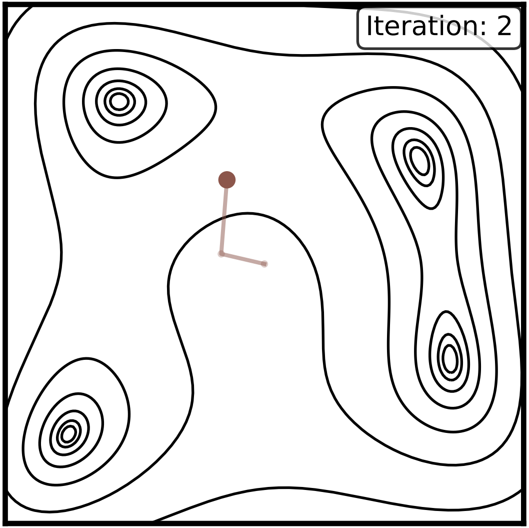


The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

How it Works (Metropolis-Hastings)

Imagine a "walker" exploring the parameter landscape.

1. Take a random step to a new position.
2. If the new spot is "higher" (better likelihood), move there.
3. If it's "lower", maybe move there anyway (with probability proportional to how much lower it is).
4. Repeat millions of times. The path the walker takes traces the posterior distribution.

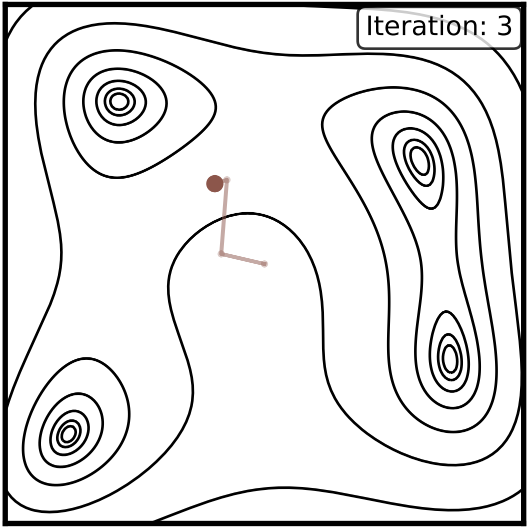


The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

How it Works (Metropolis-Hastings)

Imagine a "walker" exploring the parameter landscape.

1. Take a random step to a new position.
2. If the new spot is "higher" (better likelihood), move there.
3. If it's "lower", maybe move there anyway (with probability proportional to how much lower it is).
4. Repeat millions of times. The path the walker takes traces the posterior distribution.

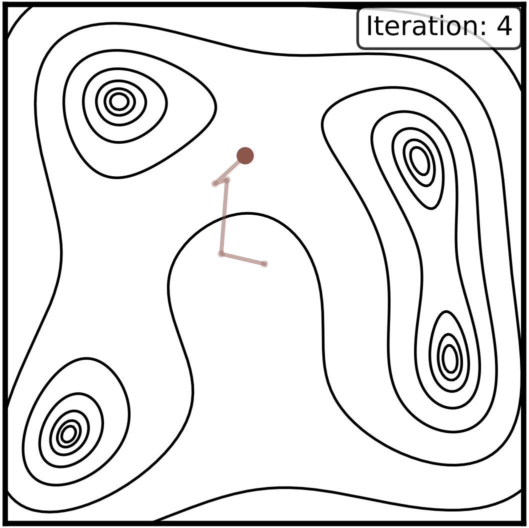


The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

How it Works (Metropolis-Hastings)

Imagine a "walker" exploring the parameter landscape.

1. Take a random step to a new position.
2. If the new spot is "higher" (better likelihood), move there.
3. If it's "lower", maybe move there anyway (with probability proportional to how much lower it is).
4. Repeat millions of times. The path the walker takes traces the posterior distribution.



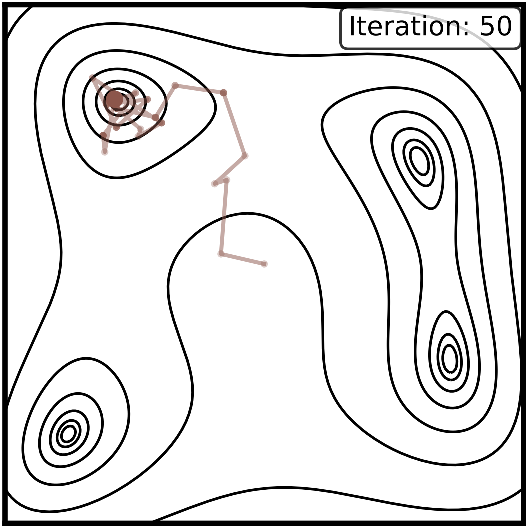
The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

Advantages & Limitations

- Explores the full posterior and gives uncertainties.
- Limitation: The walker can be inefficient. It can get "stuck" in a local high-likelihood region and fail to find other, separate modes.
- Limitation: Can be slow to explore highly correlated ("banana-shaped") posteriors.

Key Message

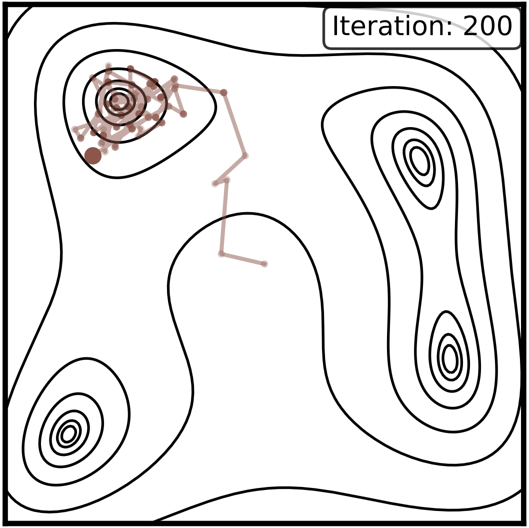
MCMC is a foundational sampling method, but its simple "random walk" can be inefficient in the complex parameter spaces of SED fitting.



The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

Advantages & Limitations

- Explores the full posterior and gives uncertainties.
- Limitation: The walker can be inefficient. It can get "stuck" in a local high-likelihood region and fail to find other, separate modes.
- Limitation: Can be slow to explore highly correlated ("banana-shaped") posteriors.



Key Message

MCMC is a foundational sampling method, but its simple "random walk" can be inefficient in the complex parameter spaces of SED fitting.

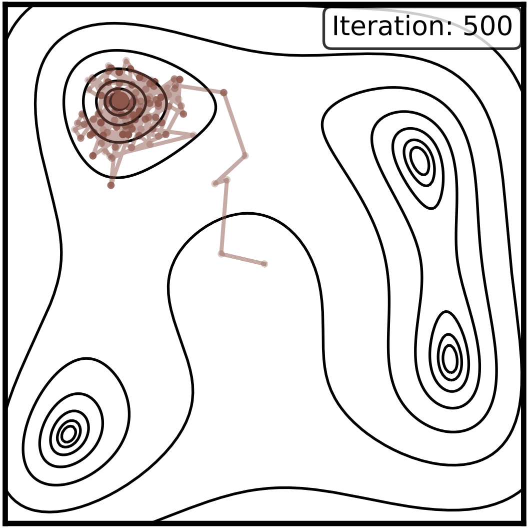
The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

Advantages & Limitations

- Explores the full posterior and gives uncertainties.
- Limitation: The walker can be inefficient. It can get "stuck" in a local high-likelihood region and fail to find other, separate modes.
- Limitation: Can be slow to explore highly correlated ("banana-shaped") posteriors.

Key Message

MCMC is a foundational sampling method, but its simple "random walk" can be inefficient in the complex parameter spaces of SED fitting.



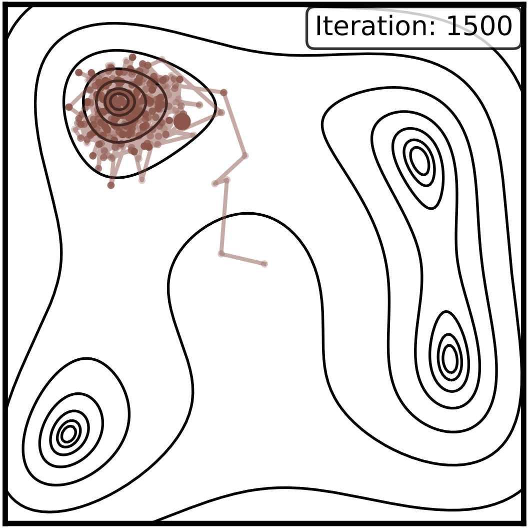
The Classic Workhorse: Markov Chain Monte Carlo (MCMC)

Advantages & Limitations

- Explores the full posterior and gives uncertainties.
- Limitation: The walker can be inefficient. It can get "stuck" in a local high-likelihood region and fail to find other, separate modes.
- Limitation: Can be slow to explore highly correlated ("banana-shaped") posteriors.

Key Message

MCMC is a foundational sampling method, but its simple "random walk" can be inefficient in the complex parameter spaces of SED fitting.

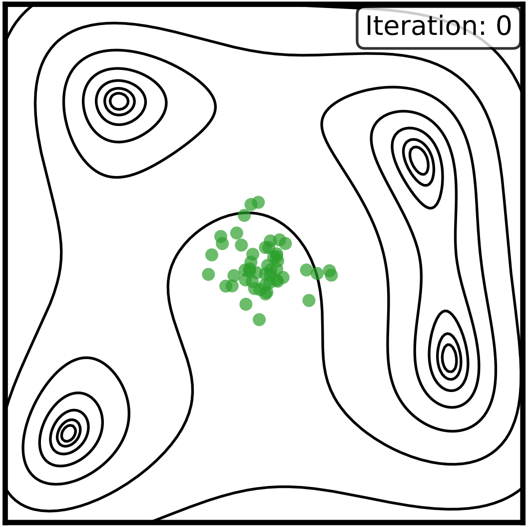


A Better Way: Ensemble Sampling (e.g., `emcee`)

How it Works

Instead of one walker, we use an **ensemble** of hundreds of walkers.

- The walkers don't move completely randomly.
- They propose new steps based on the positions of *other* walkers in the ensemble.
- This allows the whole group to learn about the shape of the posterior (e.g., its correlations) and explore it more efficiently.

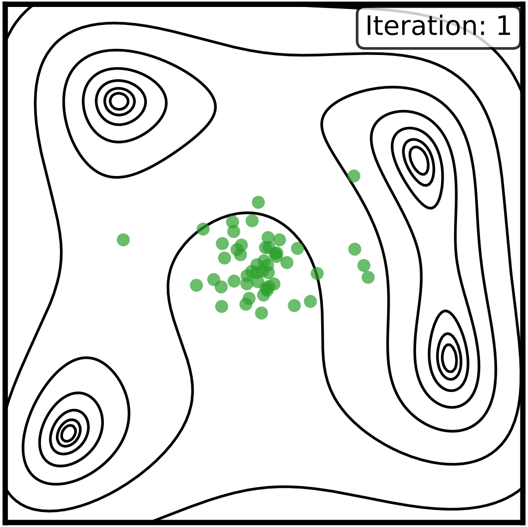


A Better Way: Ensemble Sampling (e.g., `emcee`)

How it Works

Instead of one walker, we use an **ensemble** of hundreds of walkers.

- The walkers don't move completely randomly.
- They propose new steps based on the positions of *other* walkers in the ensemble.
- This allows the whole group to learn about the shape of the posterior (e.g., its correlations) and explore it more efficiently.

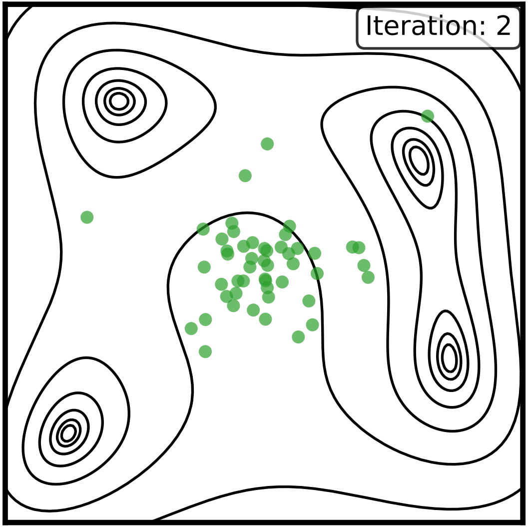


A Better Way: Ensemble Sampling (e.g., `emcee`)

How it Works

Instead of one walker, we use an **ensemble** of hundreds of walkers.

- The walkers don't move completely randomly.
- They propose new steps based on the positions of *other* walkers in the ensemble.
- This allows the whole group to learn about the shape of the posterior (e.g., its correlations) and explore it more efficiently.

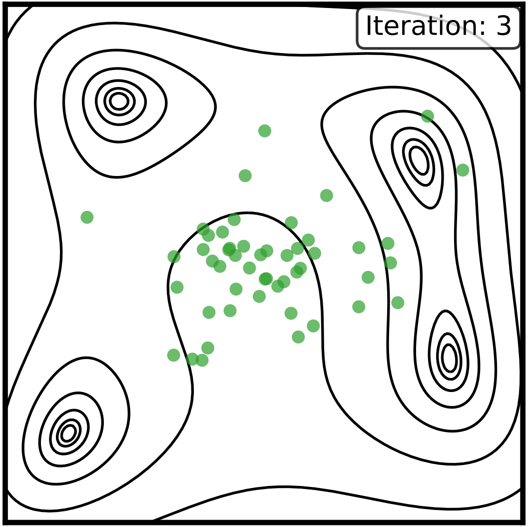


A Better Way: Ensemble Sampling (e.g., `emcee`)

How it Works

Instead of one walker, we use an **ensemble** of hundreds of walkers.

- The walkers don't move completely randomly.
- They propose new steps based on the positions of *other* walkers in the ensemble.
- This allows the whole group to learn about the shape of the posterior (e.g., its correlations) and explore it more efficiently.

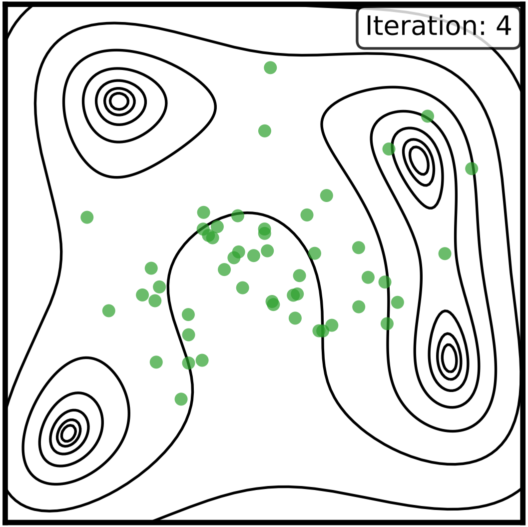


A Better Way: Ensemble Sampling (e.g., `emcee`)

How it Works

Instead of one walker, we use an **ensemble** of hundreds of walkers.

- The walkers don't move completely randomly.
- They propose new steps based on the positions of *other* walkers in the ensemble.
- This allows the whole group to learn about the shape of the posterior (e.g., its correlations) and explore it more efficiently.



A Better Way: Ensemble Sampling (e.g., emcee)

Advantages

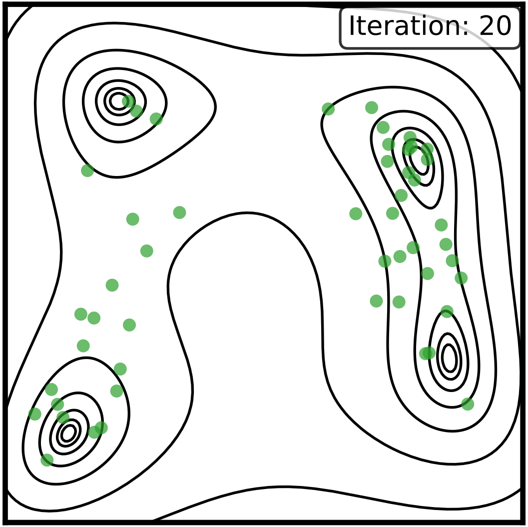
- Much better at exploring correlated, "banana-shaped" parameter spaces.
- More efficient "mixing" than a single chain.
- Easy to parallelize (one walker per CPU).

Limitation

- Ensemble can still get trapped in one mode if other modes are very far away.

Key Message

Ensemble samplers like emcee are a major improvement for many problems, especially those with parameter degeneracies.



A Better Way: Ensemble Sampling (e.g., emcee)

Advantages

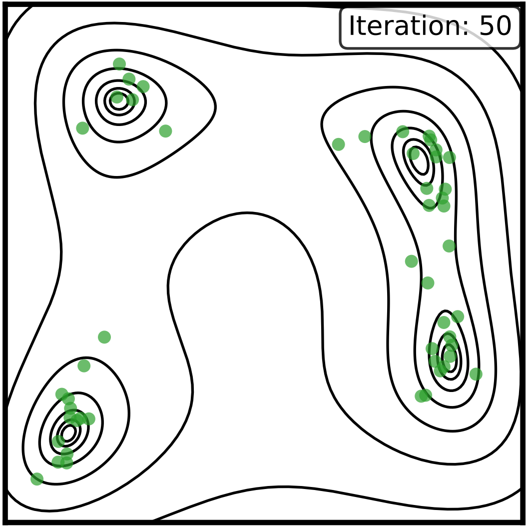
- Much better at exploring correlated, "banana-shaped" parameter spaces.
- More efficient "mixing" than a single chain.
- Easy to parallelize (one walker per CPU).

Limitation

- Ensemble can still get trapped in one mode if other modes are very far away.

Key Message

Ensemble samplers like emcee are a major improvement for many problems, especially those with parameter degeneracies.



A Better Way: Ensemble Sampling (e.g., emcee)

Advantages

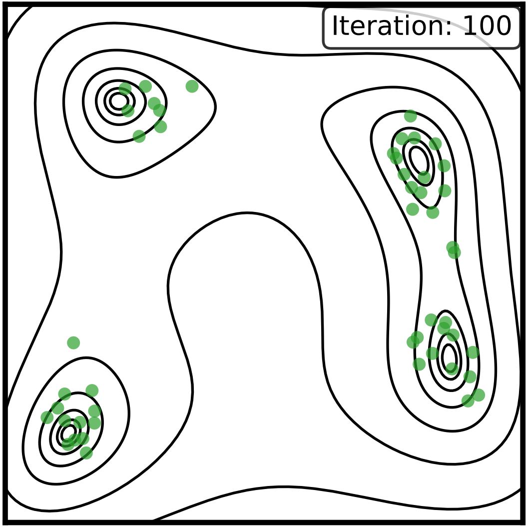
- Much better at exploring correlated, "banana-shaped" parameter spaces.
- More efficient "mixing" than a single chain.
- Easy to parallelize (one walker per CPU).

Limitation

- Ensemble can still get trapped in one mode if other modes are very far away.

Key Message

Ensemble samplers like emcee are a major improvement for many problems, especially those with parameter degeneracies.



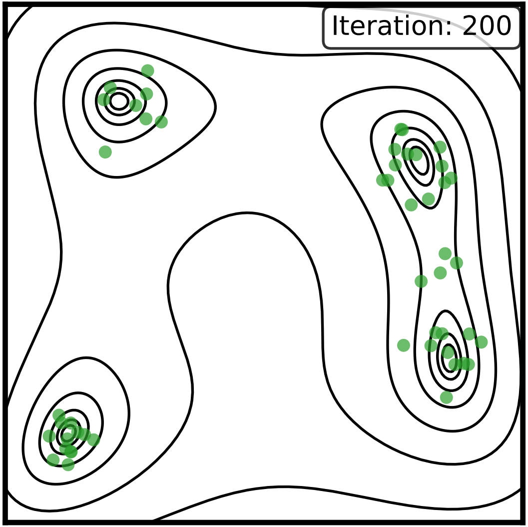
A Better Way: Ensemble Sampling (e.g., `emcee`)

Advantages

- Much better at exploring correlated, "banana-shaped" parameter spaces.
- More efficient "mixing" than a single chain.
- Easy to parallelize (one walker per CPU).

Limitation

- Ensemble can still get trapped in one mode if other modes are very far away.



Key Message

Ensemble samplers like `emcee` are a major improvement for many problems, especially those with parameter degeneracies.

The Missing Piece: Why Evidence Calculation Matters

And why it's so hard to compute

Why Evidence is Important

- **Model Comparison**: Bayes model theorem:

$$\mathcal{P}(M|D) \propto \mathcal{Z}(D|M)\mathcal{P}(M)$$

  For SED fitting: Which stellar population model best explains the galaxy photometry?
- **Occam's Razor**: Automatic complexity penalty

$$\log \mathcal{Z} = \langle \log \mathcal{L} \rangle_{\mathcal{P}} - \mathcal{D}_{\mathrm{KL}}(\mathcal{P}||\pi)$$

- **Bayesian Model Averaging**: Weighted model combinations

Why Evidence is Hard

- The high-dimensional evidence integral:

$$\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta$$

$$\left(\text{from Bayes theorem} : \mathcal{P}(\theta|D) = \frac{\mathcal{L}(\theta)\pi(\theta)}{\mathcal{Z}}\right)$$

The Missing Piece: Why Evidence Calculation Matters

And why it's so hard to compute

Why Evidence is Important

- **Model Comparison**: Bayes model theorem:

$$\mathcal{P}(M|D) \propto \mathcal{Z}(D|M)\mathcal{P}(M)$$

  For SED fitting: Which stellar population model best explains the galaxy photometry?

- **Occam's Razor**: Automatic complexity penalty

$$\log \mathcal{Z} = \langle \log \mathcal{L} \rangle_{\mathcal{P}} - \mathcal{D}_{\mathrm{KL}}(\mathcal{P}||\pi)$$

- **Bayesian Model Averaging**: Weighted model combinations

Why Evidence is Hard

- The high-dimensional evidence integral:

$$\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta$$

$$\left(\text{from Bayes theorem} : \mathcal{P}(\theta|D) = \frac{\mathcal{L}(\theta)\pi(\theta)}{\mathcal{Z}}\right)$$

- The difficulty is **not** that most of parameter space has $\mathcal{L} \approx 0$...

The Missing Piece: Why Evidence Calculation Matters

And why it's so hard to compute

Why Evidence is Important

- **Model Comparison**: Bayes model theorem:

$$\mathcal{P}(M|D) \propto \mathcal{Z}(D|M)\mathcal{P}(M)$$

  For SED fitting: Which stellar population model best explains the galaxy photometry?

- **Occam's Razor**: Automatic complexity penalty

$$\log \mathcal{Z} = \langle \log \mathcal{L} \rangle_{\mathcal{P}} - \mathcal{D}_{\mathrm{KL}}(\mathcal{P}||\pi)$$

- **Bayesian Model Averaging**: Weighted model combinations

Why Evidence is Hard

- The high-dimensional evidence integral:

$$\mathcal{Z} = \int \mathcal{L}(\theta)\pi(\theta)d\theta$$

$$\left(\text{from Bayes theorem} : \mathcal{P}(\theta|D) = \frac{\mathcal{L}(\theta)\pi(\theta)}{\mathcal{Z}}\right)$$

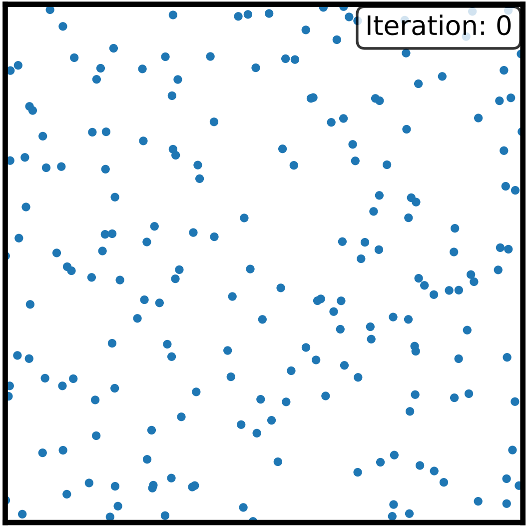
- The difficulty is **not** that most of parameter space has $\mathcal{L} \approx 0$...
- The difficulty is that we can't estimate **volume** $d\theta$ in high dimensions!

The State of the Art: Nested Sampling (e.g., `dynesty`)

A Radically Different Approach

Instead of random walking, nested sampling attacks the problem from the outside-in.

1. Start with a set of "live points" scattered across the entire **prior**.
2. At each step: find the point with the *worst* likelihood and discard it.
3. Replace it with a new point drawn from the prior, but with a likelihood *better* than the point you just discarded.
4. This forces the set of live points to continuously "shrink" into regions of higher and higher likelihood.

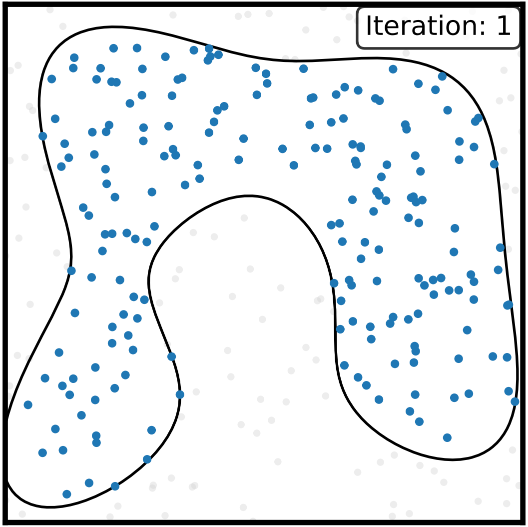


The State of the Art: Nested Sampling (e.g., dynesty)

A Radically Different Approach

Instead of random walking, nested sampling attacks the problem from the outside-in.

1. Start with a set of "live points" scattered across the entire **prior**.
2. At each step: find the point with the *worst* likelihood and discard it.
3. Replace it with a new point drawn from the prior, but with a likelihood *better* than the point you just discarded.
4. This forces the set of live points to continuously "shrink" into regions of higher and higher likelihood.

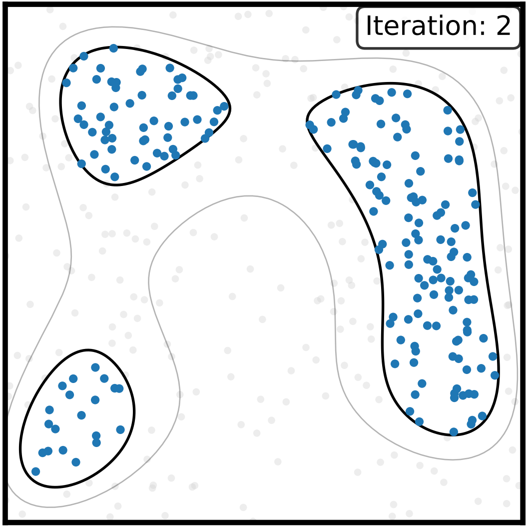


The State of the Art: Nested Sampling (e.g., `dynesty`)

A Radically Different Approach

Instead of random walking, nested sampling attacks the problem from the outside-in.

1. Start with a set of "live points" scattered across the entire **prior**.
2. At each step: find the point with the *worst* likelihood and discard it.
3. Replace it with a new point drawn from the prior, but with a likelihood *better* than the point you just discarded.
4. This forces the set of live points to continuously "shrink" into regions of higher and higher likelihood.

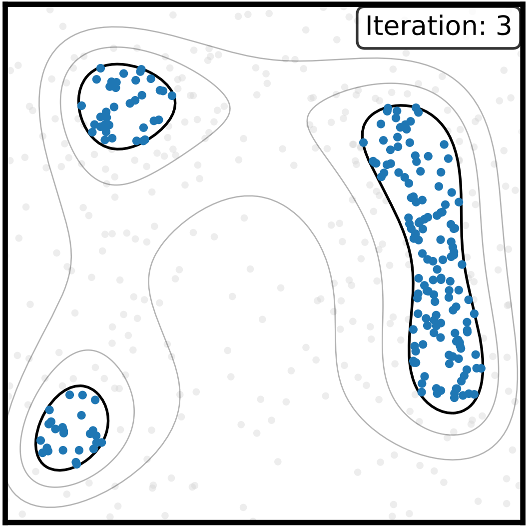


The State of the Art: Nested Sampling (e.g., `dynesty`)

A Radically Different Approach

Instead of random walking, nested sampling attacks the problem from the outside-in.

1. Start with a set of "live points" scattered across the entire **prior**.
2. At each step: find the point with the *worst* likelihood and discard it.
3. Replace it with a new point drawn from the prior, but with a likelihood *better* than the point you just discarded.
4. This forces the set of live points to continuously "shrink" into regions of higher and higher likelihood.



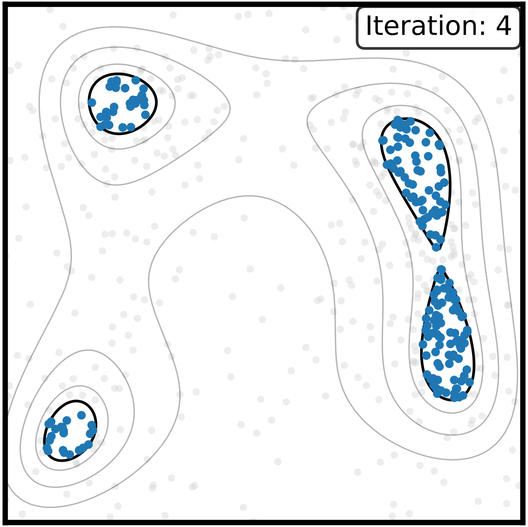
The State of the Art: Nested Sampling (e.g., `dynesty`)

Key Advantages

- Naturally handles **multimodality**. The shrinking cloud of points will find and explore all modes simultaneously.
- It calculates the **Bayesian Evidence** ($\mathcal{Z}$) as a primary output. This is essential for model comparison!

Key Message

Nested sampling excels at exploring complex, multimodal posteriors and is the go-to method for Bayesian model comparison.



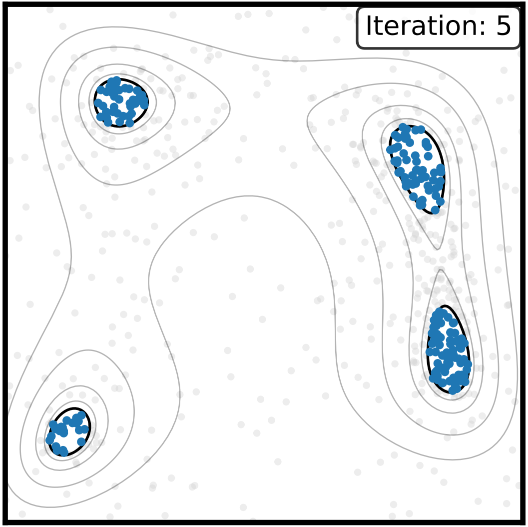
The State of the Art: Nested Sampling (e.g., `dynesty`)

Key Advantages

- Naturally handles **multimodality**. The shrinking cloud of points will find and explore all modes simultaneously.
- It calculates the **Bayesian Evidence** ($\mathcal{Z}$) as a primary output. This is essential for model comparison!

Key Message

Nested sampling excels at exploring complex, multimodal posteriors and is the go-to method for Bayesian model comparison.



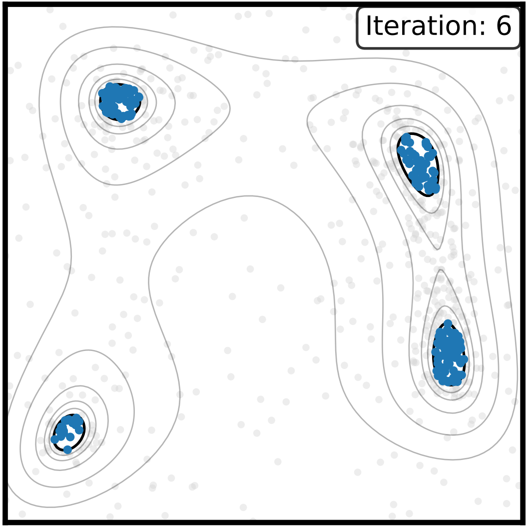
The State of the Art: Nested Sampling (e.g., `dynesty`)

Key Advantages

- Naturally handles **multimodality**. The shrinking cloud of points will find and explore all modes simultaneously.
- It calculates the **Bayesian Evidence** ($\mathcal{Z}$) as a primary output. This is essential for model comparison!

Key Message

Nested sampling excels at exploring complex, multimodal posteriors and is the go-to method for Bayesian model comparison.



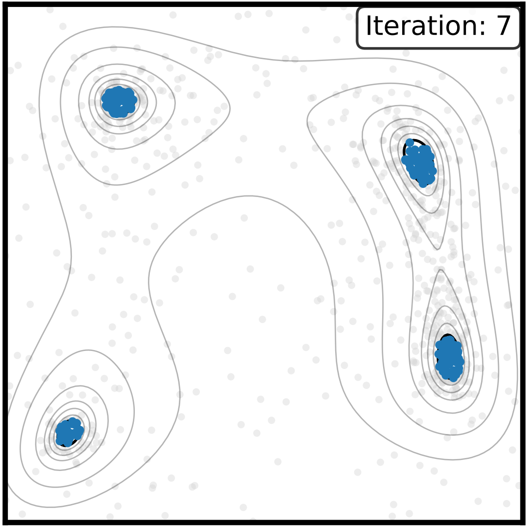
The State of the Art: Nested Sampling (e.g., `dynesty`)

Key Advantages

- Naturally handles **multimodality**. The shrinking cloud of points will find and explore all modes simultaneously.
- It calculates the **Bayesian Evidence** ($\mathcal{Z}$) as a primary output. This is essential for model comparison!

Key Message

Nested sampling excels at exploring complex, multimodal posteriors and is the go-to method for Bayesian model comparison.



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

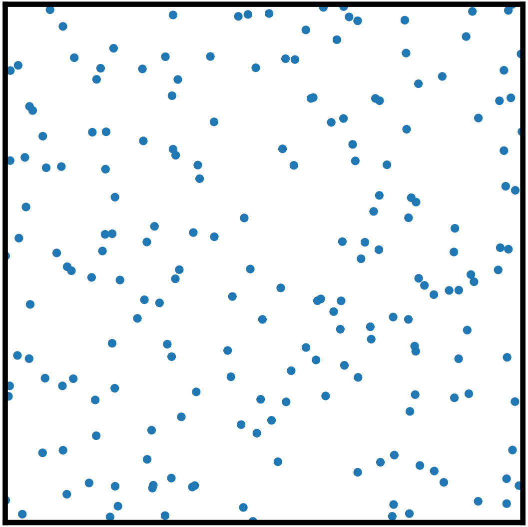
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

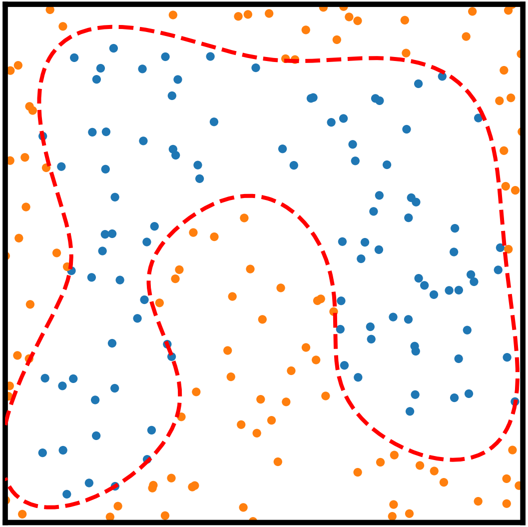
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

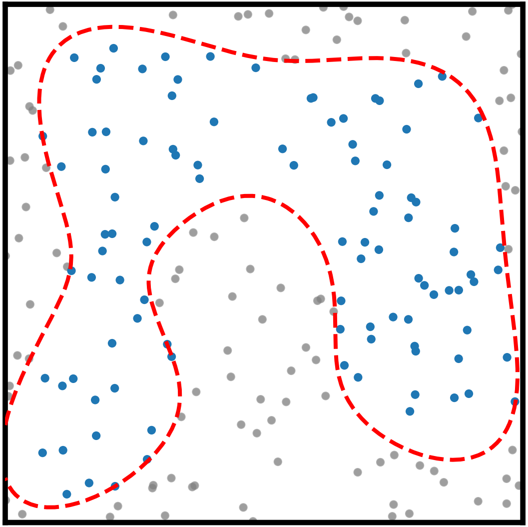
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

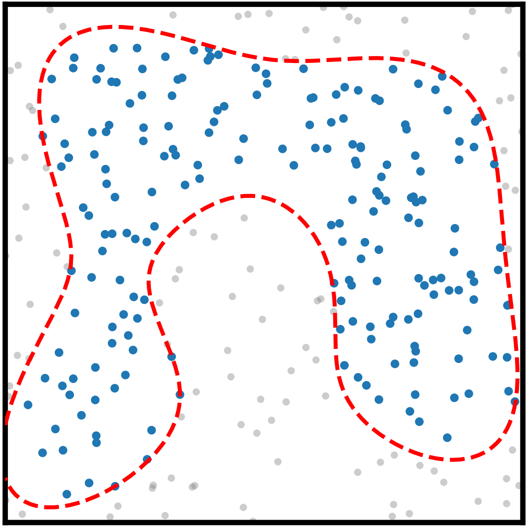
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

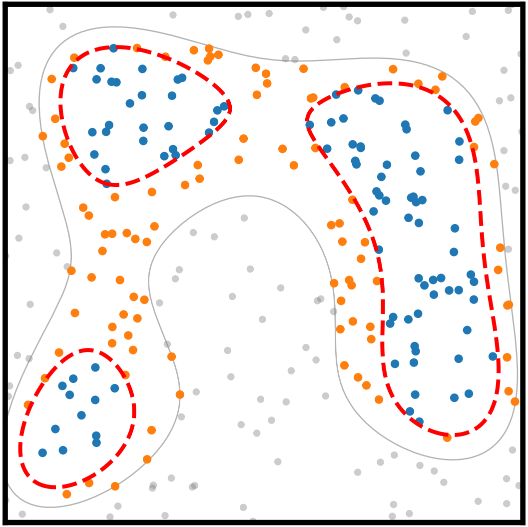
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

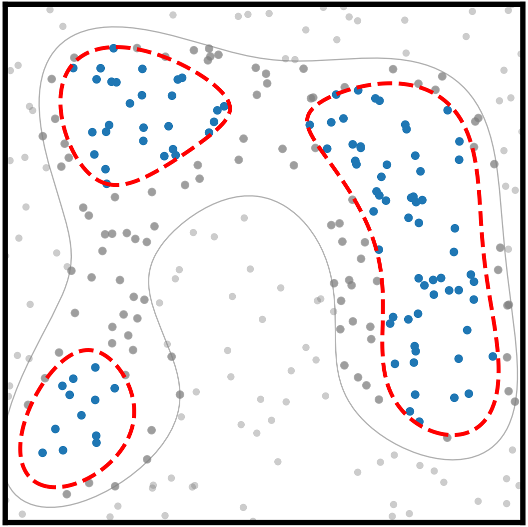
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

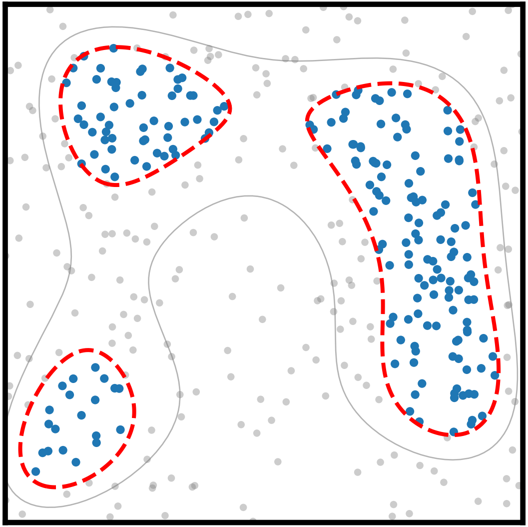
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

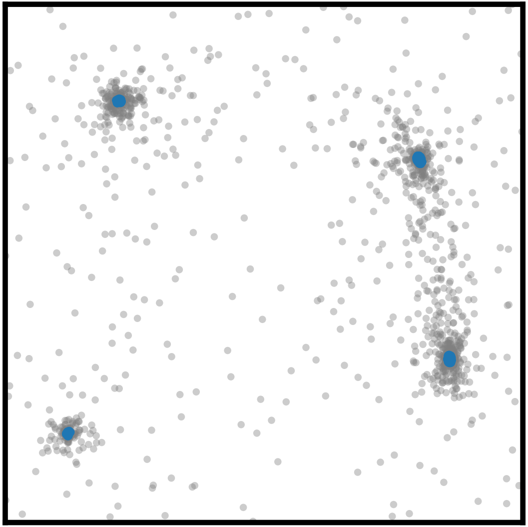
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



How Nested Sampling Estimates Volumes: The Counting Trick

Volume Contraction

At each step, the volume contracts predictably:

$$V_{i+1} = V_i \times \frac{n_{\text{inside}}}{n_{\text{total}}}$$

indep. of dimensionality, geometry or topology

Evidence

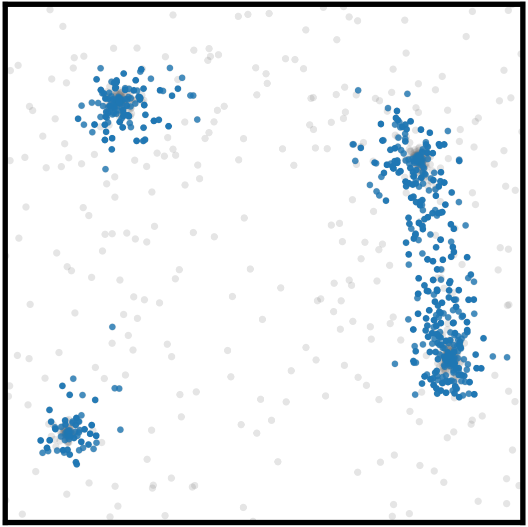
The evidence is computed as:

$$\mathcal{Z} = \sum \mathcal{L}_i \Delta V_i$$

Posterior

Each sample gets importance weight:

$$w_i = \mathcal{L}_i \times \Delta V_i$$



Practical Guidance: How to Use Nested Sampling

Understanding resolution and reliability parameters

Rejection Samplers

- e.g. `MultiNest`, `UltraNest`, `nessai`
- Construct bounding regions, reject invalid points
- Efficient in low dimensions ($d \lesssim 10$)
- Exponentially inefficient in high dimensions

Chain-based Samplers

- e.g. `PolyChord`, `dynesty`, `blackjax`
- Run Markov chains from live points
- Linear $\sim \mathcal{O}(d)$ scaling penalty
- Better for high-dimensional problems

Key Parameters

- **Resolution parameter** n_{live}: Improves results as $\sim \mathcal{O}(n_{\text{live}}^{-1/2})$
- **Reliability parameters**: Don't improve results if set arbitrarily high, but introduce systematic errors if set too low
  - `MultiNest` efficiency `eff`, `PolyChord` chain length `n_repeats`, `dynesty` `slices`

Choosing Your Tool: A Summary

No single best method, only the right tool for the job

Method	Speed	Uncertainties?	Handles Multimodality?	Evidence?
Optimization (χ^2)	Very Fast	No	No	No
MCMC (`pymc` etc)	Medium	Yes	Poorly	No
Ensemble (`emcee`)	Medium	Yes	Okay	No
Nested (`dynesty`)	Slower	Yes	Excellently	Yes!

Practical Guidance

- **Quick exploration / Sanity check?** → Use Optimization.
- **Simple, well-behaved posterior?** → `emcee` is a great choice.
- **Complex, possibly multimodal posterior?** → Use `dynesty`.
- **Need to compare different physical models?** → You *must* use Nested Sampling.

The Future: AI in Scientific Code Development

How these tools themselves are evolving

The Real AI Revolution: LLMs

The biggest impact of AI will not be in analyzing data, but in helping us write the code to do it.

- **Automated code translation**: LLMs can help port legacy Fortran/C++ models to modern, GPU-friendly & differentiable frameworks like JAX or PyTorch.

The 80/20 Rule of Scientific Work

- **80% "boring" tasks**: Writing code, debugging, drafting & reviewing papers, munging data, organising meetings...
- **20% "hard thinking"**: The actual scientific insight.

AI's biggest immediate impact is automating and accelerating the 80%, freeing up human time for the 20%.

Key Message

AI is not just a tool for analysis; it's about to fundamentally change how we develop, optimize, and deploy our science

Conclusions & What's Next

github.com/handley-lab/group

Key Takeaways

- SED fitting is a problem of **statistical inference**, not just optimization.
- The goal is the full **posterior distribution**, which gives us parameters *and* their uncertainties.
- **Sampling** methods are the tools we use to map out the posterior.
- The choice of sampler—from MCMC to Ensemble to Nested—depends on the complexity of your problem and whether you need to do **model comparison**.

Next Up: David Yallup on "GPU Accelerated Nested Sampling"

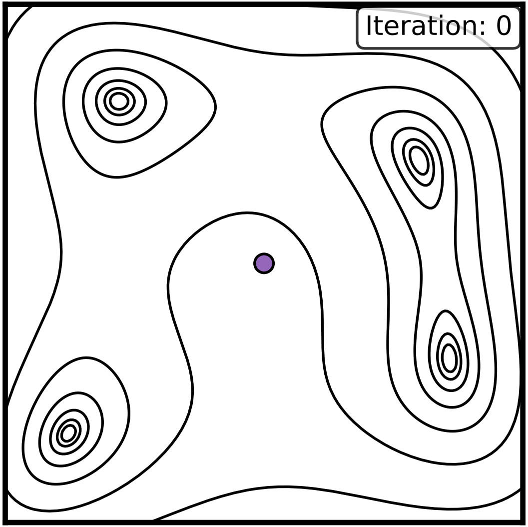
Now that we know *why* nested sampling is so powerful, we'll hear about how to make it *fast*!

Appendix: Hamiltonian Monte Carlo (HMC)

How it Works

Uses gradients to guide exploration more efficiently than random walks.

1. Treat parameters as "particles" with position and momentum.
2. Use gradient of log-likelihood as "force" to guide movement.
3. Propose coherent moves along gradient directions.
4. Accept/reject using Metropolis criterion.

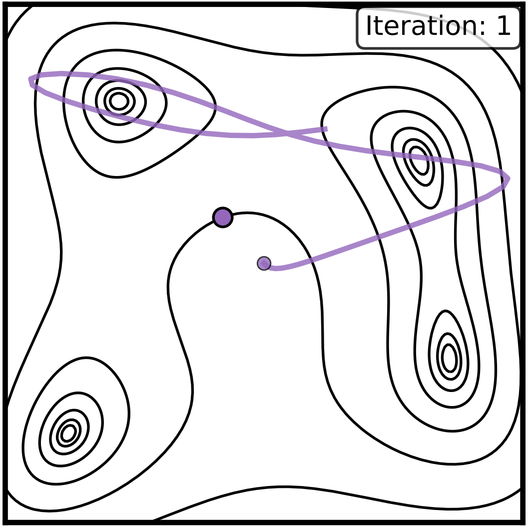


Appendix: Hamiltonian Monte Carlo (HMC)

How it Works

Uses gradients to guide exploration more efficiently than random walks.

1. Treat parameters as "particles" with position and momentum.
2. Use gradient of log-likelihood as "force" to guide movement.
3. Propose coherent moves along gradient directions.
4. Accept/reject using Metropolis criterion.

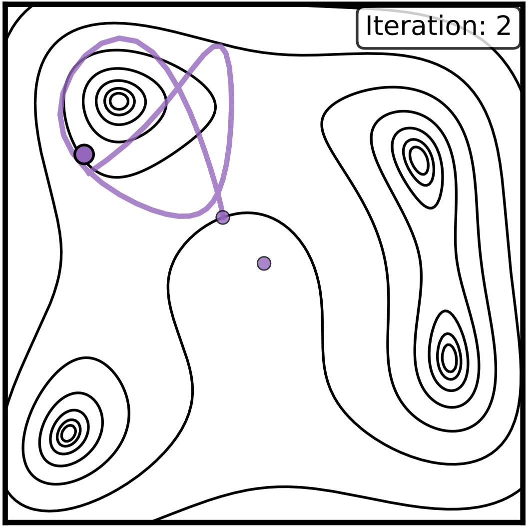


Appendix: Hamiltonian Monte Carlo (HMC)

How it Works

Uses gradients to guide exploration more efficiently than random walks.

1. Treat parameters as "particles" with position and momentum.
2. Use gradient of log-likelihood as "force" to guide movement.
3. Propose coherent moves along gradient directions.
4. Accept/reject using Metropolis criterion.

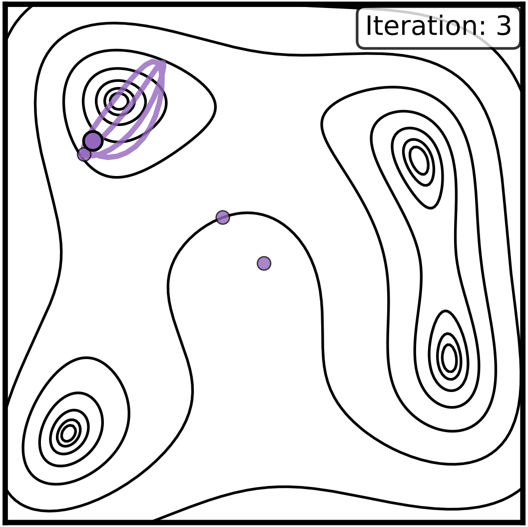


Appendix: Hamiltonian Monte Carlo (HMC)

How it Works

Uses gradients to guide exploration more efficiently than random walks.

1. Treat parameters as "particles" with position and momentum.
2. Use gradient of log-likelihood as "force" to guide movement.
3. Propose coherent moves along gradient directions.
4. Accept/reject using Metropolis criterion.

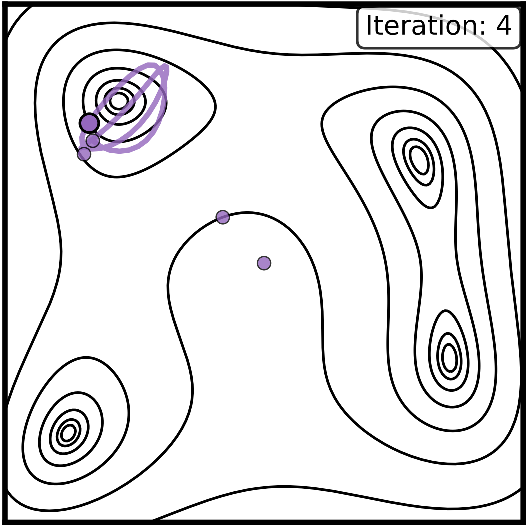


Appendix: Hamiltonian Monte Carlo (HMC)

How it Works

Uses gradients to guide exploration more efficiently than random walks.

1. Treat parameters as "particles" with position and momentum.
2. Use gradient of log-likelihood as "force" to guide movement.
3. Propose coherent moves along gradient directions.
4. Accept/reject using Metropolis criterion.



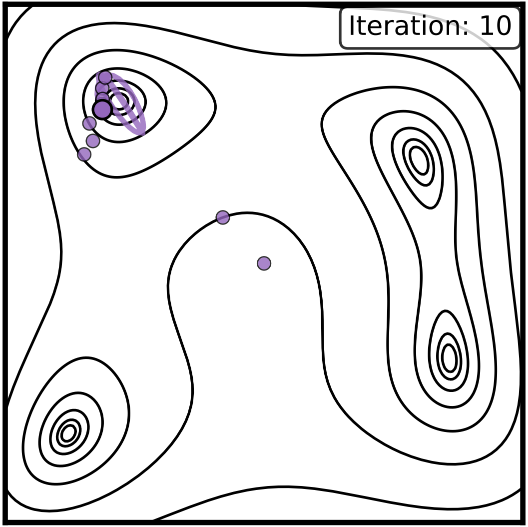
Appendix: Hamiltonian Monte Carlo (HMC)

Advantages & Requirements

- Much more efficient than random walk for smooth posteriors.
- Requires gradients of the likelihood function.
- Can traverse parameter space much faster.
- Less likely to get stuck in local regions.

Key Message

HMC leverages gradient information for efficient sampling, but requires differentiable models.



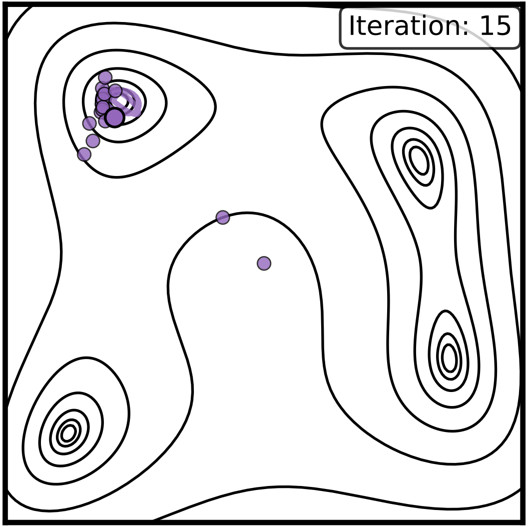
Appendix: Hamiltonian Monte Carlo (HMC)

Advantages & Requirements

- Much more efficient than random walk for smooth posteriors.
- Requires gradients of the likelihood function.
- Can traverse parameter space much faster.
- Less likely to get stuck in local regions.

Key Message

HMC leverages gradient information for efficient sampling, but requires differentiable models.



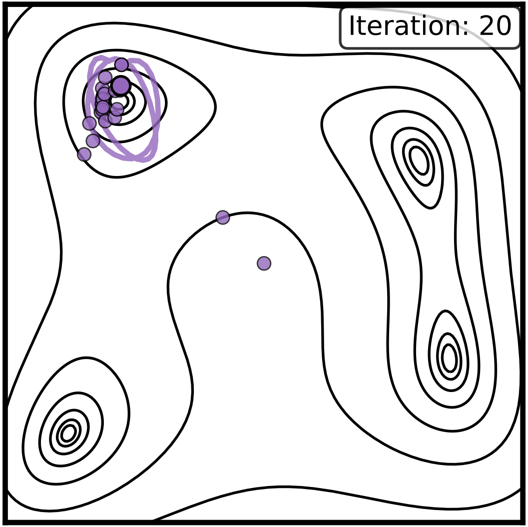
Appendix: Hamiltonian Monte Carlo (HMC)

Advantages & Requirements

- Much more efficient than random walk for smooth posteriors.
- Requires gradients of the likelihood function.
- Can traverse parameter space much faster.
- Less likely to get stuck in local regions.

Key Message

HMC leverages gradient information for efficient sampling, but requires differentiable models.



Appendix: Hamiltonian Monte Carlo (HMC)

Advantages & Requirements

- Much more efficient than random walk for smooth posteriors.
- Requires gradients of the likelihood function.
- Can traverse parameter space much faster.
- Less likely to get stuck in local regions.

Key Message

HMC leverages gradient information for efficient sampling, but requires differentiable models.