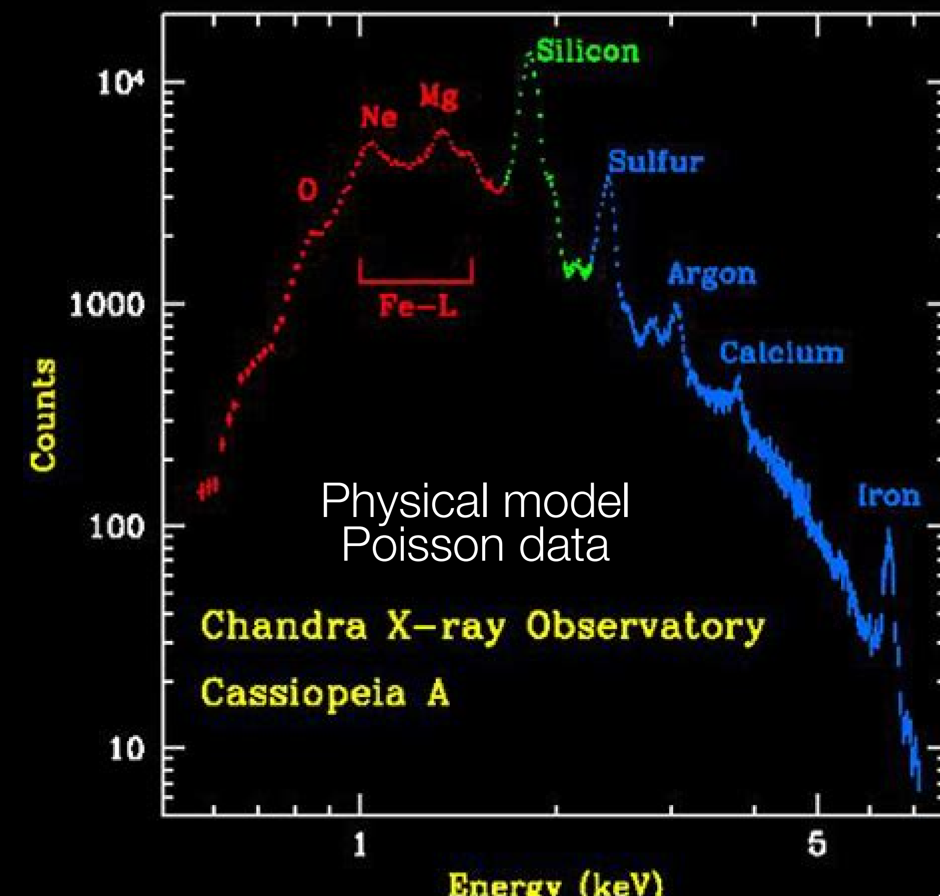
Nested Sampling

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

Astronomy & Astrophysics

Understanding individual objects



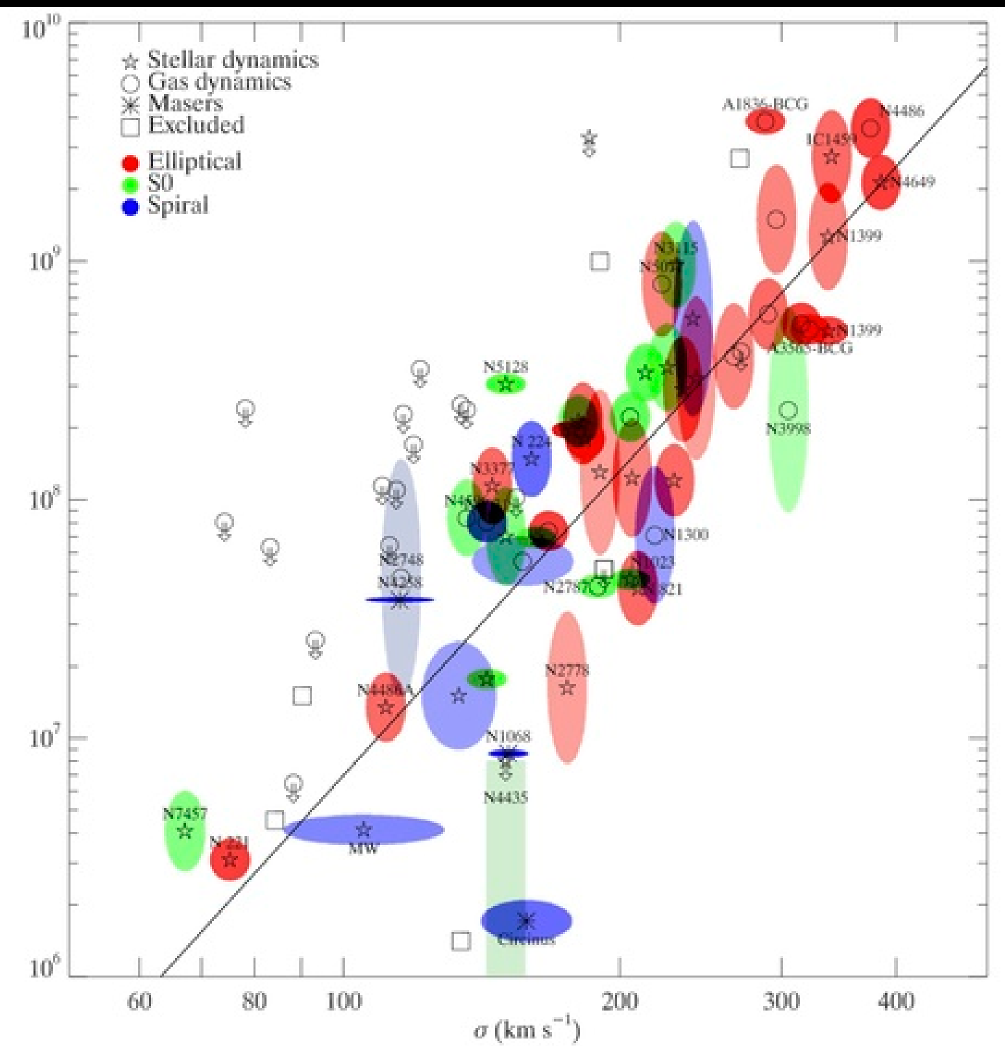
... 10^{20}eV

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

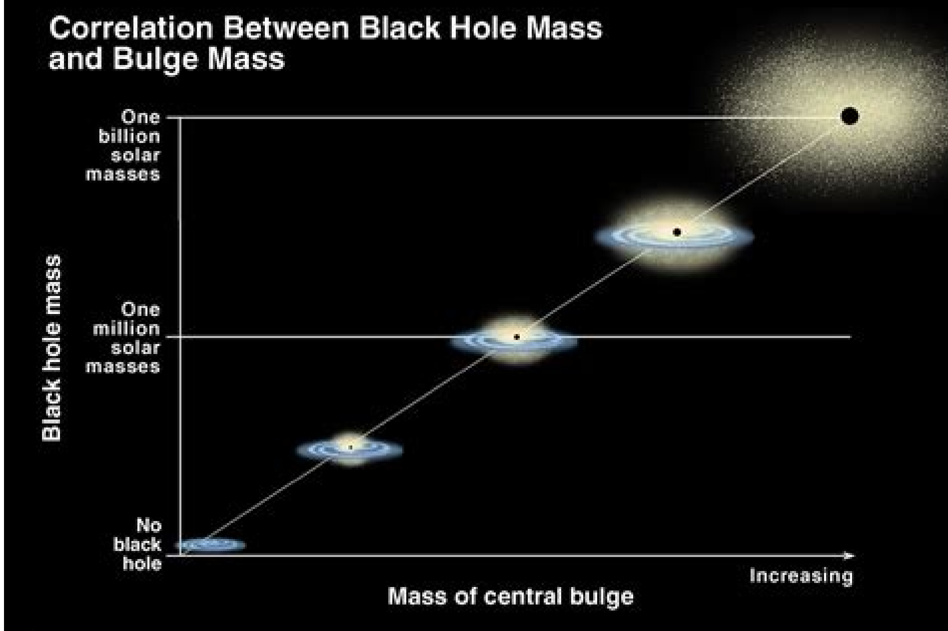
Astronomy & Astrophysics

Understanding samples

Blac Hole Mass



Velocity dispersion of stars



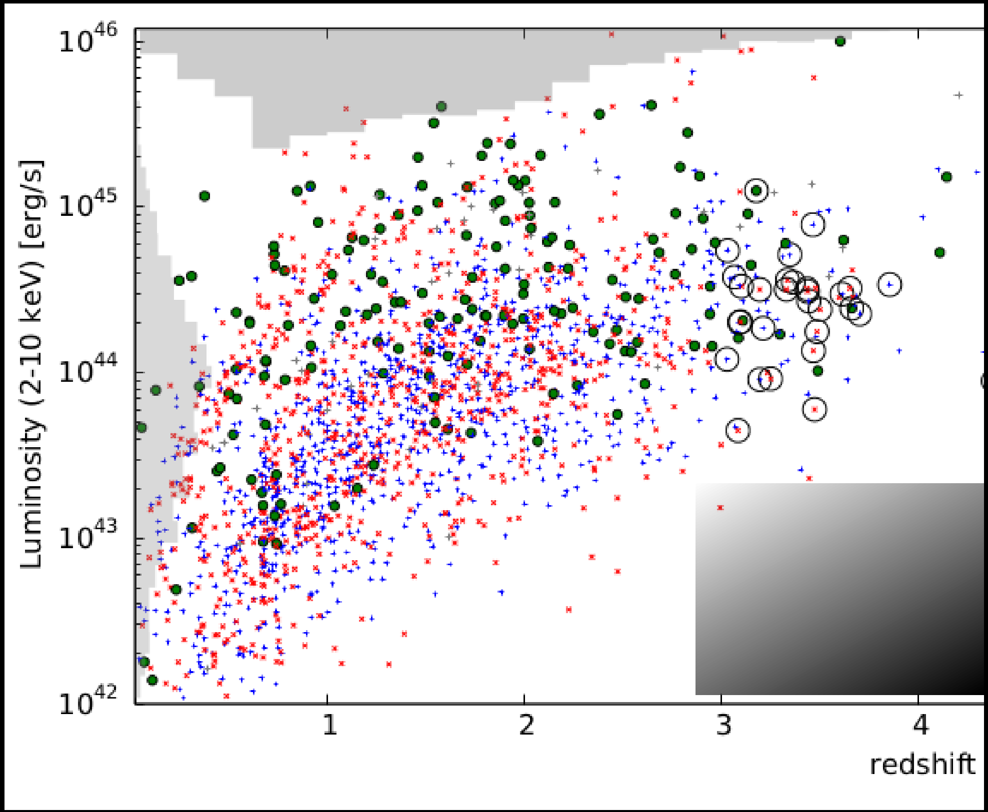
Empirical models, correlations

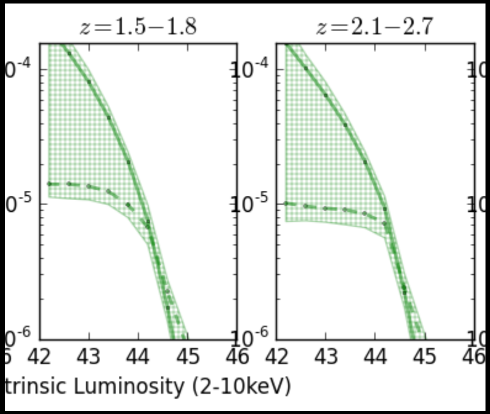
Physical interpretation

reproduction with galaxy simulations

Astronomy & Astrophysics

Understanding underlying populations



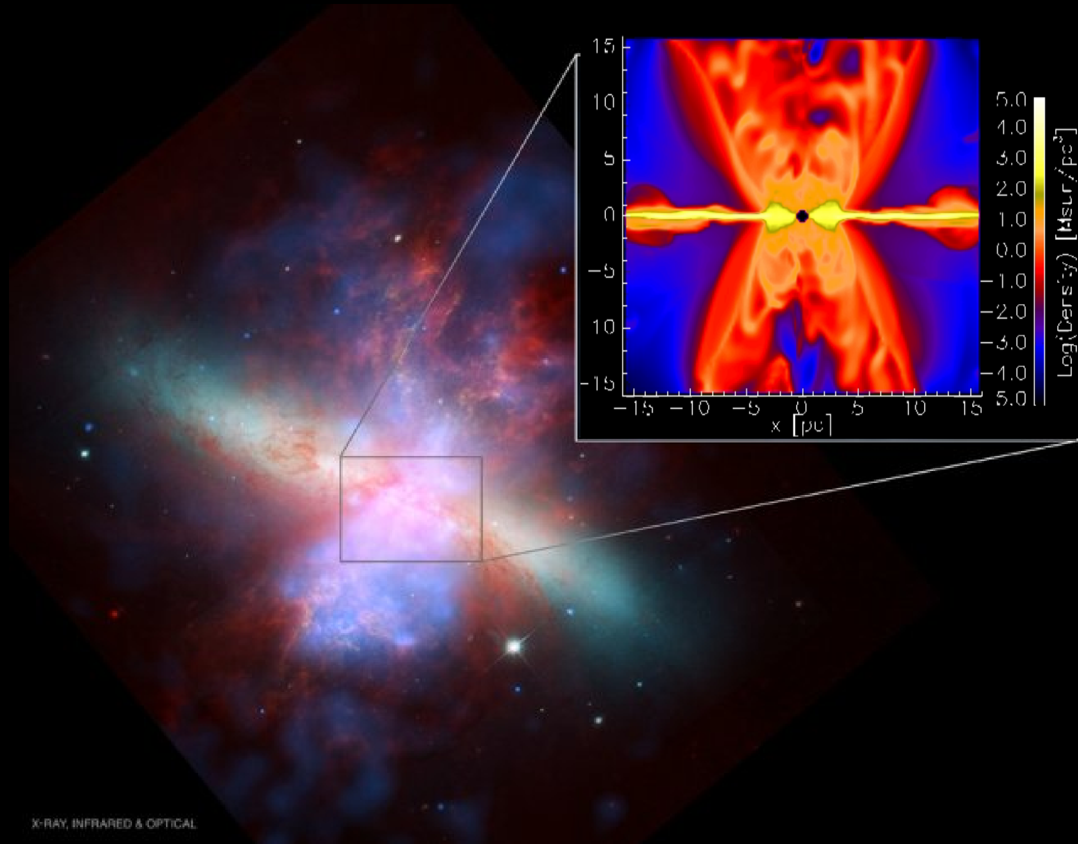


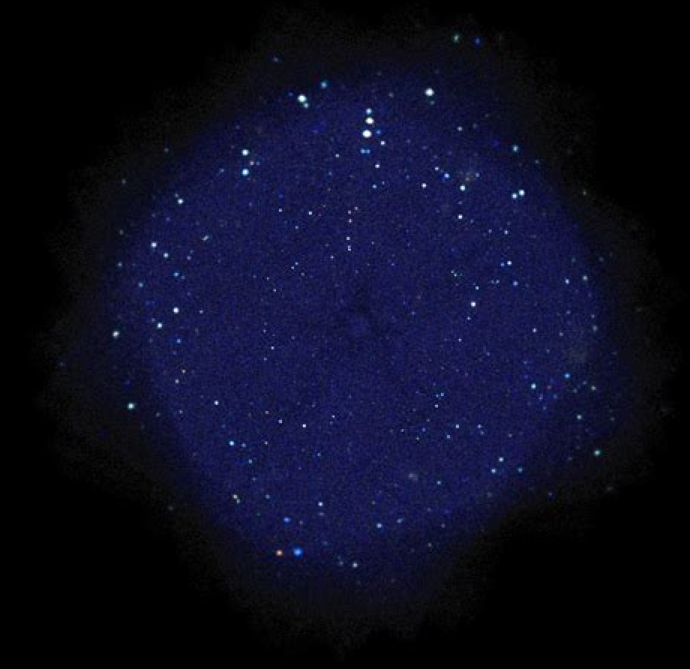
Censorship (but understood)

heavily heterogeneous uncertainties (but understood)

Hierarchical Bayesian Model with censorship (1983)

(N+1)*nested sampling + importance sampling
N * nested sampling + Stan





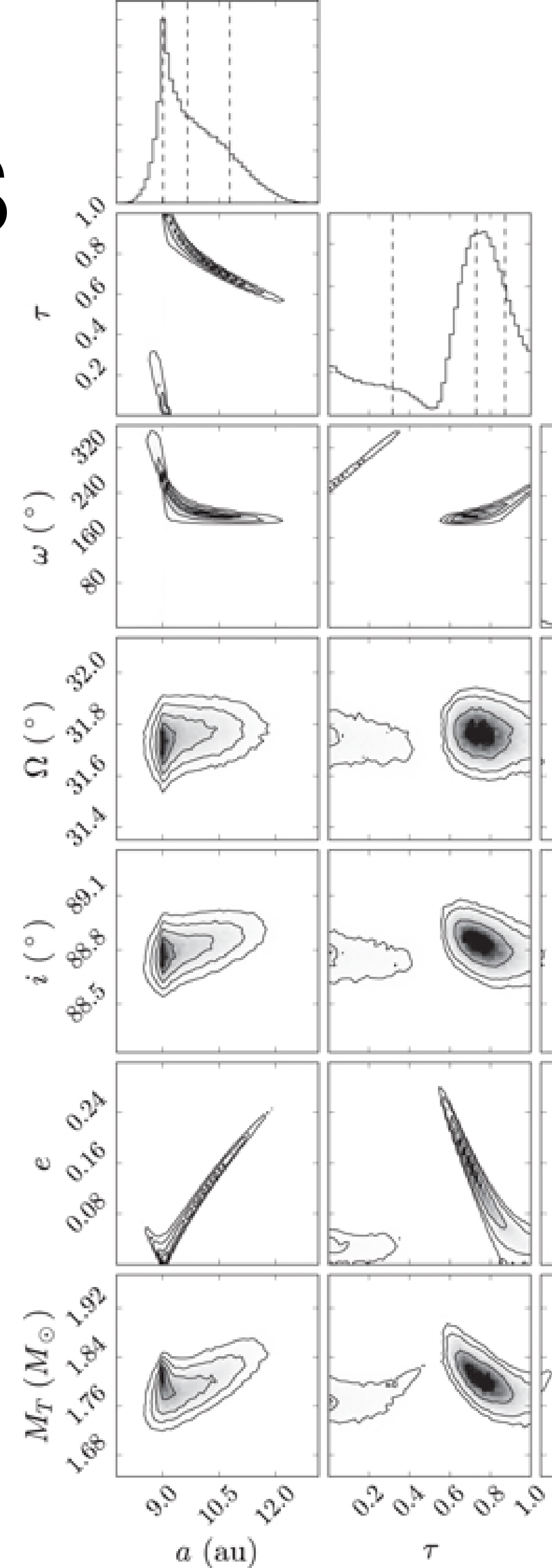
Galaxy clusters for Dark Energy
3 million Active Galactic Nuclei

Johannes Buchner

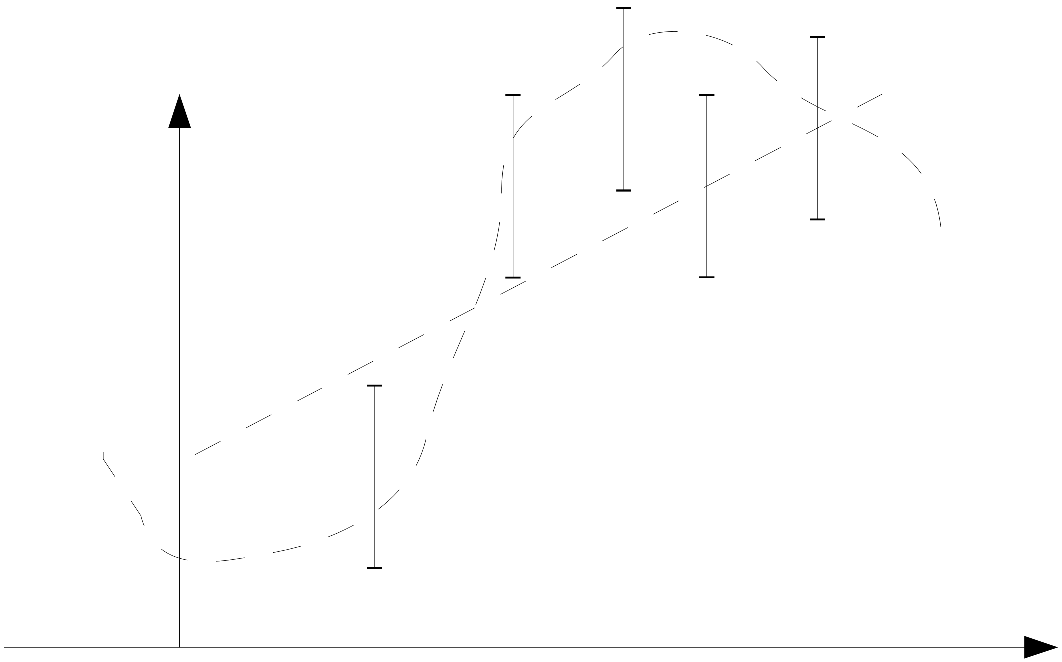
http://astrost.at/istics/
Max Planck Institute for Extraterrestrial Physics
Garching, Munich

Inference requirements

- Forward modeling with Bayesian inference is popular
- Physical models: can be complex & expensive Monte Carlo simulations
- Competing physical effects
- “degenerate” parameter constraints **P**
- multiple solutions **M**
- Sometimes little data, sometimes lots **I**
- Often 2-20d (or 1e6d) **D**



Bayesian inference is integral



- What parameters are probable under a given model?
- What models are probable?

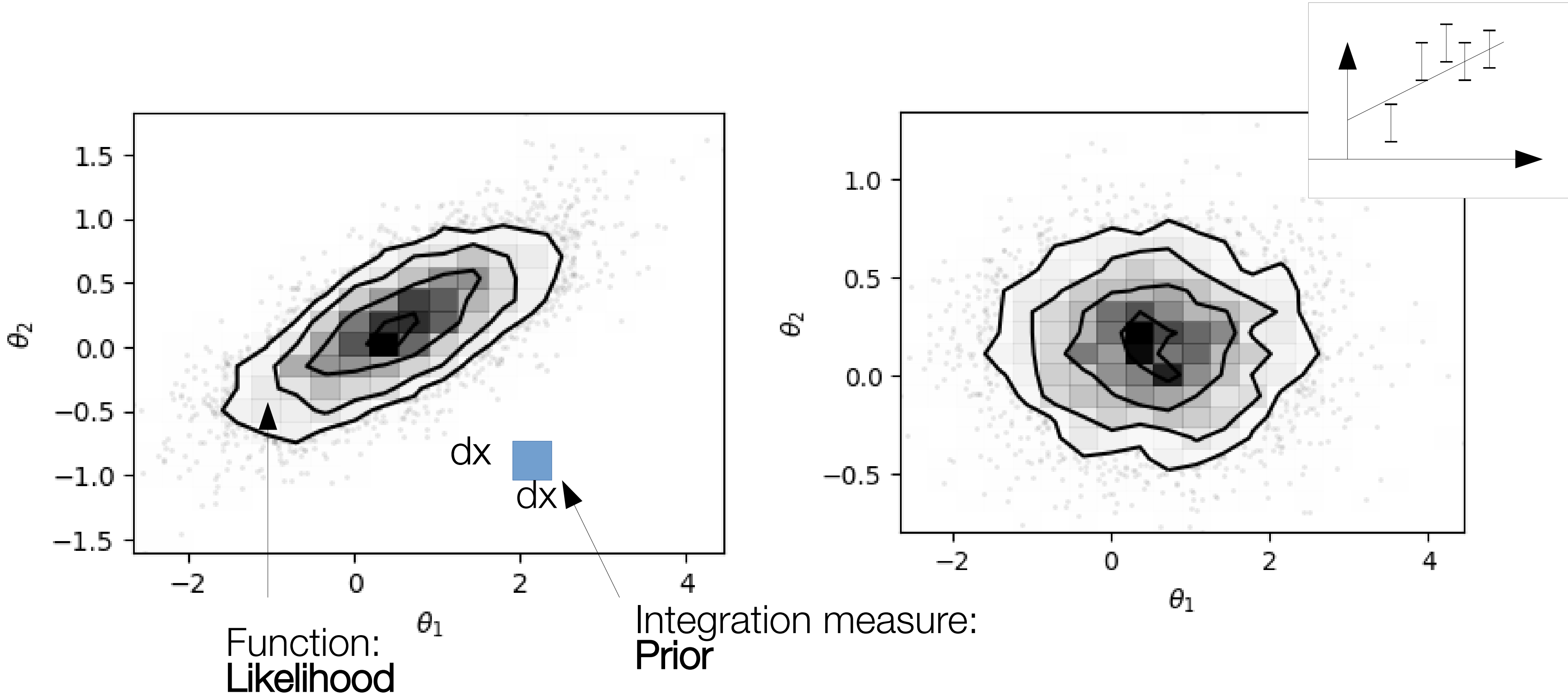
$$M_1 : y_{\text{pred}} = \theta_1 + x \cdot \theta_2$$

“physical model prediction”
non-linear, complex, slow

$$L = \prod_i \frac{(y_i - y_{\text{pred}}(x_i))^2}{2 \cdot \sigma_i^2}$$

statistical measurement process
data vs. model

Bayesian inference is integral



- Parameter estimation: identify parameter space subset
- Model comparison: identify model subset

containing 99% of probability mass

Bayesian tools

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

(& have a life beyond convergence criteria)

	Parameter estimation	Model comparison
low-d:	Nested sampling	Nested sampling
(~20d) high-d:	HMCMC with multiple chains	open research problem
	→ posterior samples	→ Bayes factors

Nested sampling theory

- Idea — Skilling 2004, 2006
- Convergence — Evans 2007; Chopin & Robert 2010; Skilling 2009
- Point Process-based Monte Carlo — Walter (2014)
- Sequential Monte Carlo — Salomone (2018)
  Superset of SMC & NS algorithms with strong theoretical foundations

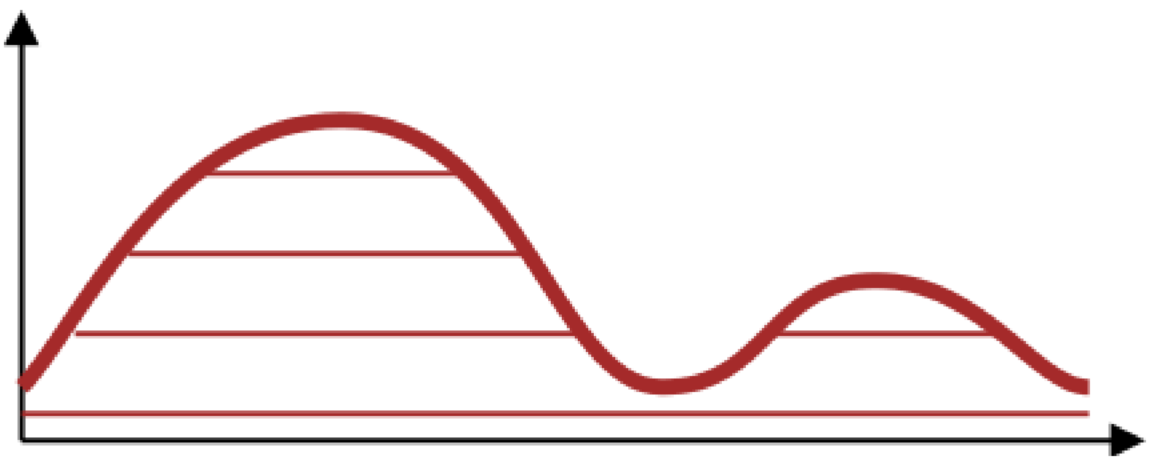
Birge (2012)
Polson & Scott (2014)

Nested sampling in practice

- Inputs
  - Dimensionality of the problem
  - Prior density function
  - Likelihood function
- Outputs
  - Posterior samples (like in MCMC)
  - ln(Z) with uncertainties

MultiNest, PyMultiNest, nestle, dynesty, polychord, UltraNest

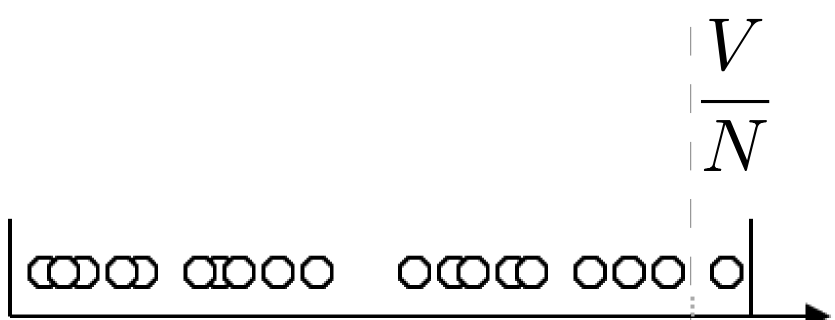
Nested sampling idea (1)



Lebegue integral:
Height * dV

$$Z \approx \sum_i \Delta V_i \cdot L_i$$

Nested sampling idea (2)

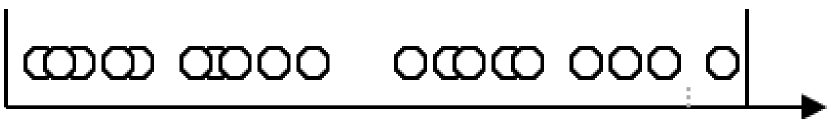


$$V' = V \times \left(1 - \frac{1}{N}\right)$$

Keeping track of volume

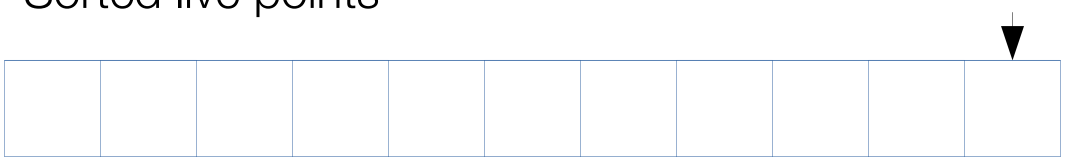
- Exponential shrinkage (~1/N)

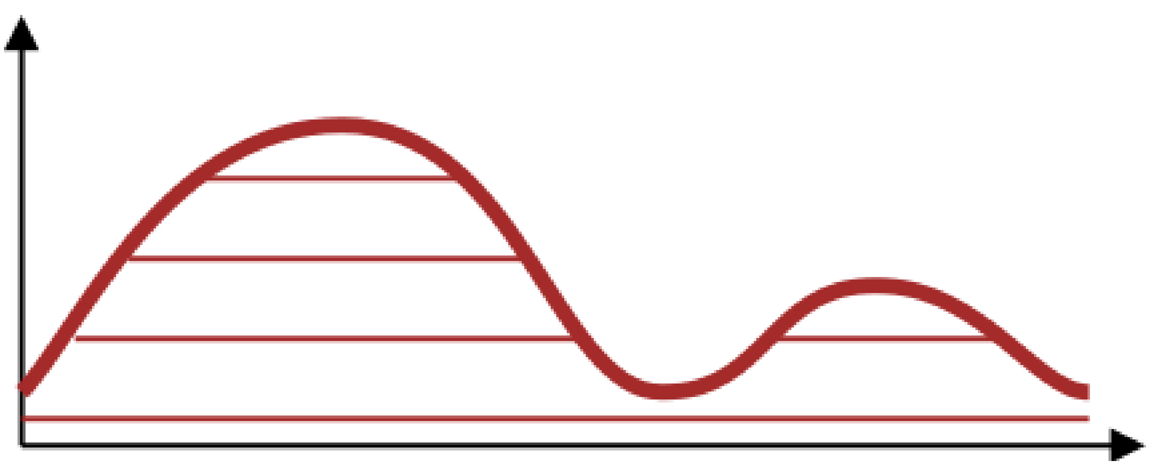
Nested sampling idea (3)



Sorted live points

To be replaced





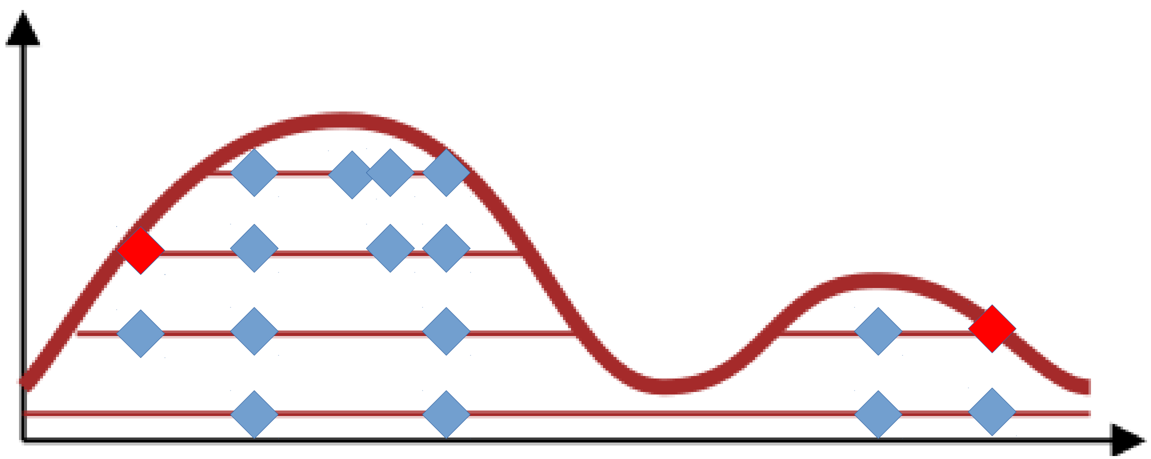
$$Z \approx \sum_i \Delta V_i \cdot L_i$$

- Keeping track of volume
    - Exponential shrinkage (~1/N)
- Keeping track of height

Skilling ‘04,06,09
Evans ‘07
Chopin&Robert ‘07,10
Walter ‘14

Nested sampling idea (4)

Integration from low to high likelihoods



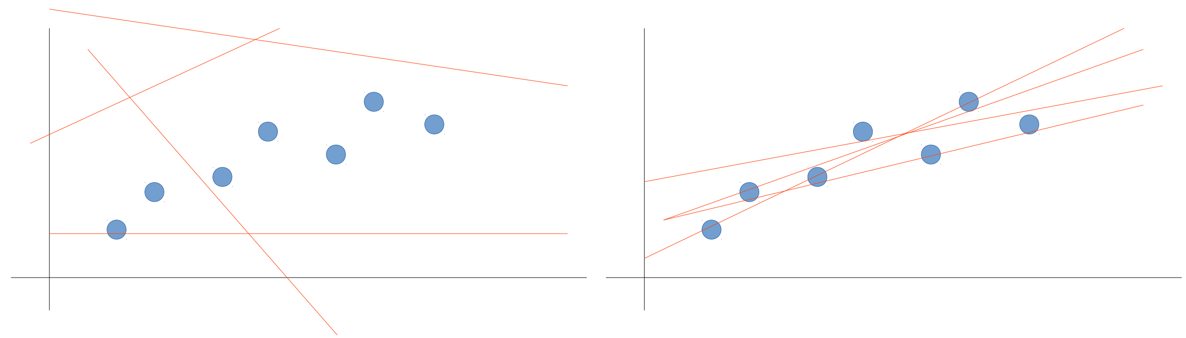
$$Z \approx \sum_i \Delta V_i \cdot L_i$$

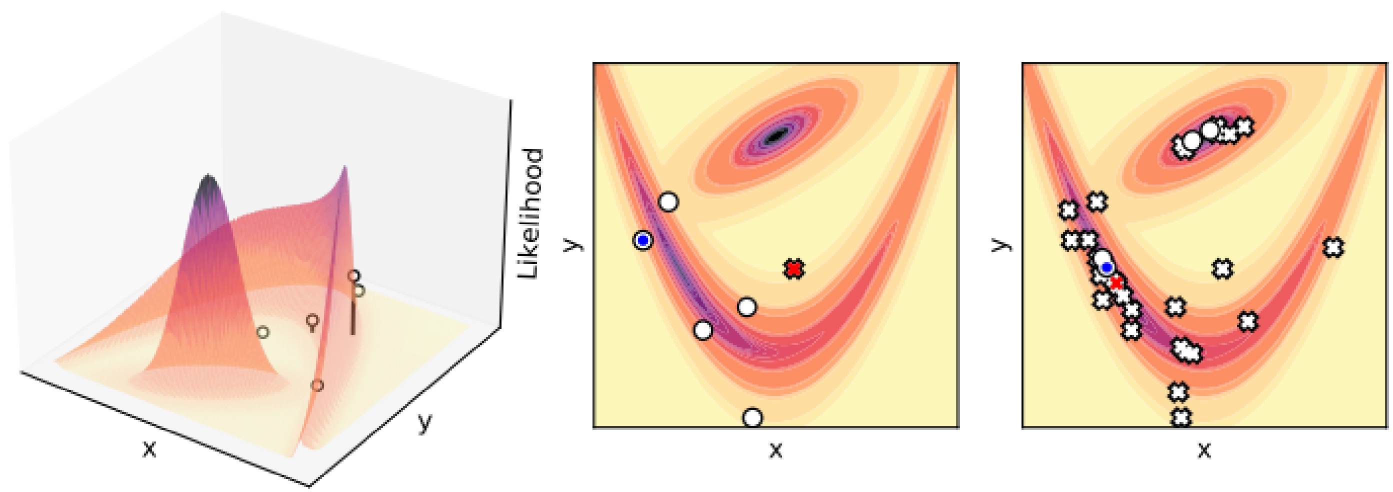
Skilling '04,06,09
Evans '07
Chopin&Robert '07,10
Walter '14

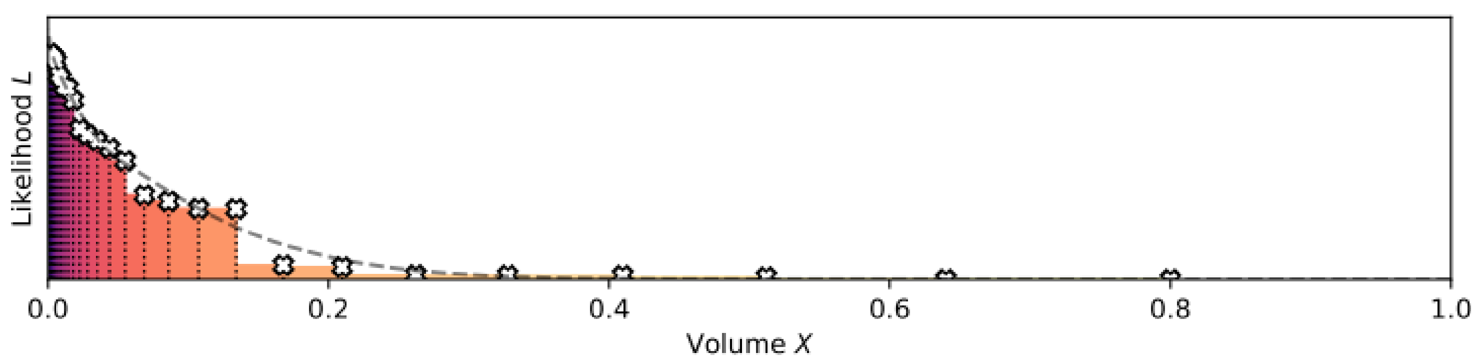
NS algorithm

- Generate initial live points from prior, evalute L
- Set V_1=1
- Loop
  - Remove lowest L point = Lmin
  - Dead point posterior weight: $w_i = V_i \times 1/N \times L_i$
  - $V_{i+1} = V_i \times (1 - 1/N)$
  - sample new point, subject to L≥Lmin
- $Z = \Sigma_i w_i$

A cooling scheme







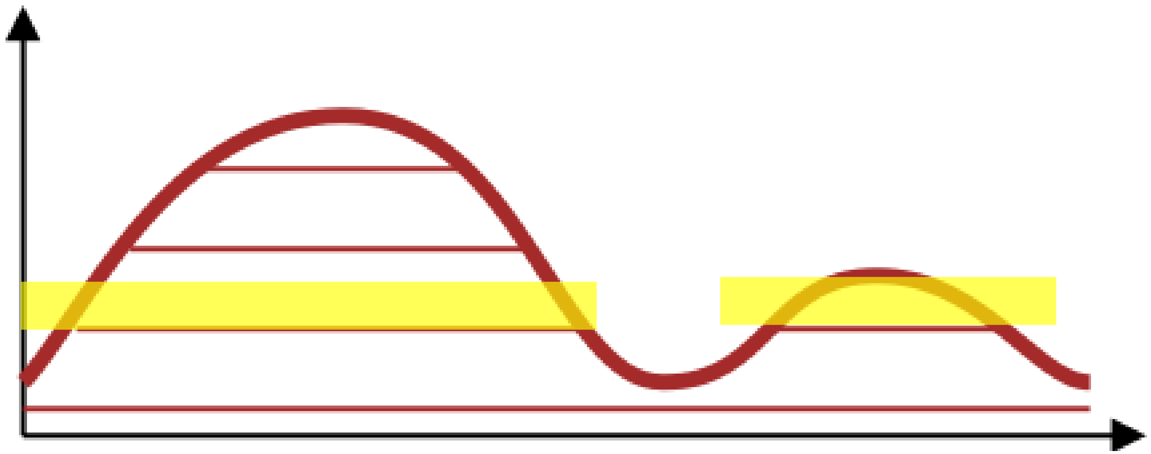
Nested sampling theory

- Idea — Skilling 2004, 2006
- Convergence — Evans 2007; Chopin & Robert 2010; Skilling 2009
- Point Process-based Monte Carlo — Walter (2014)
- Sequential Monte Carlo — Salomone+ 2018

  Superset of SMC & NS algorithms with strong theoretical foundations

Birge (2012)
Polson & Scott (2014)

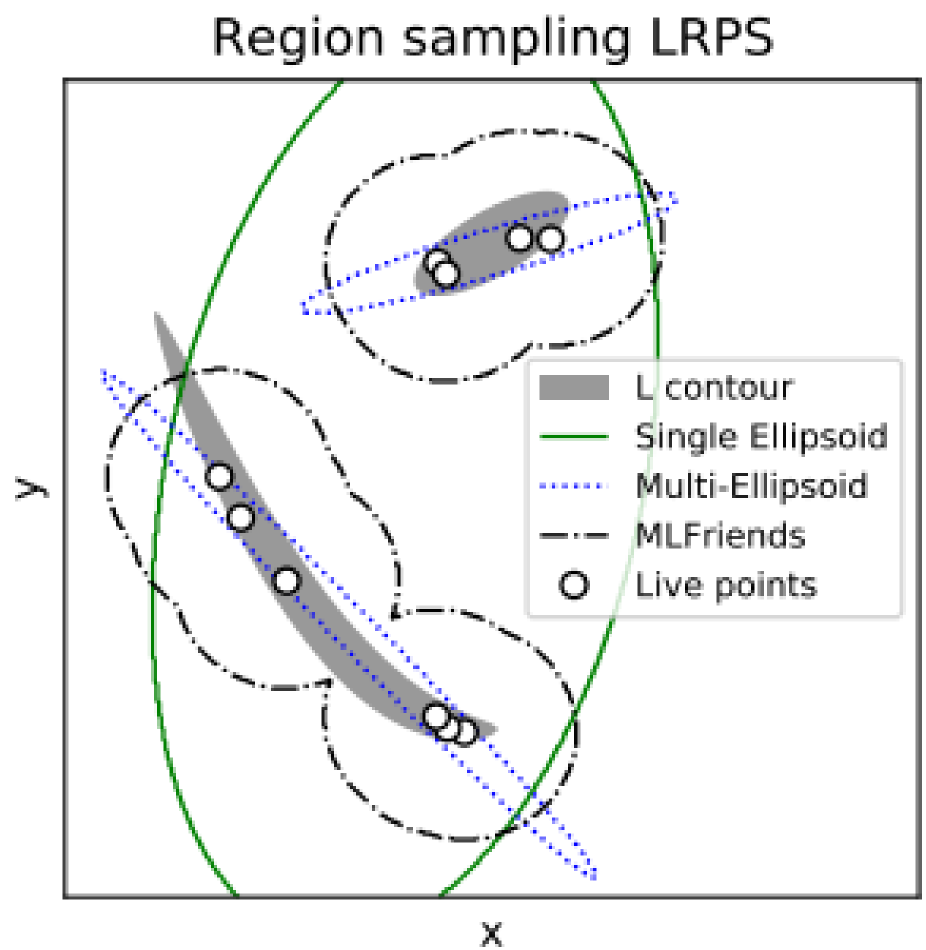
Constrained sampling

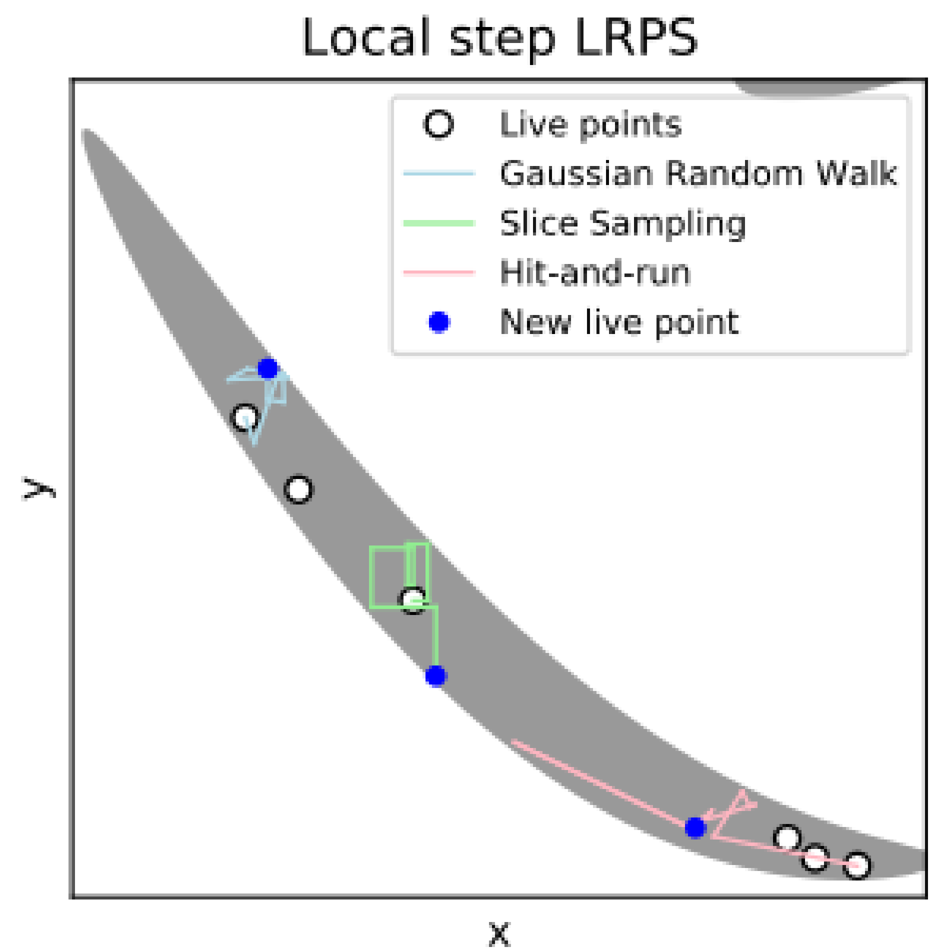


$$Z \approx \sum_i \Delta V_i \cdot L_i$$

Skilling '04,06,09
Evans '07
Chopin&Robert '07,10
Walter '14

L-restricted prior sampling



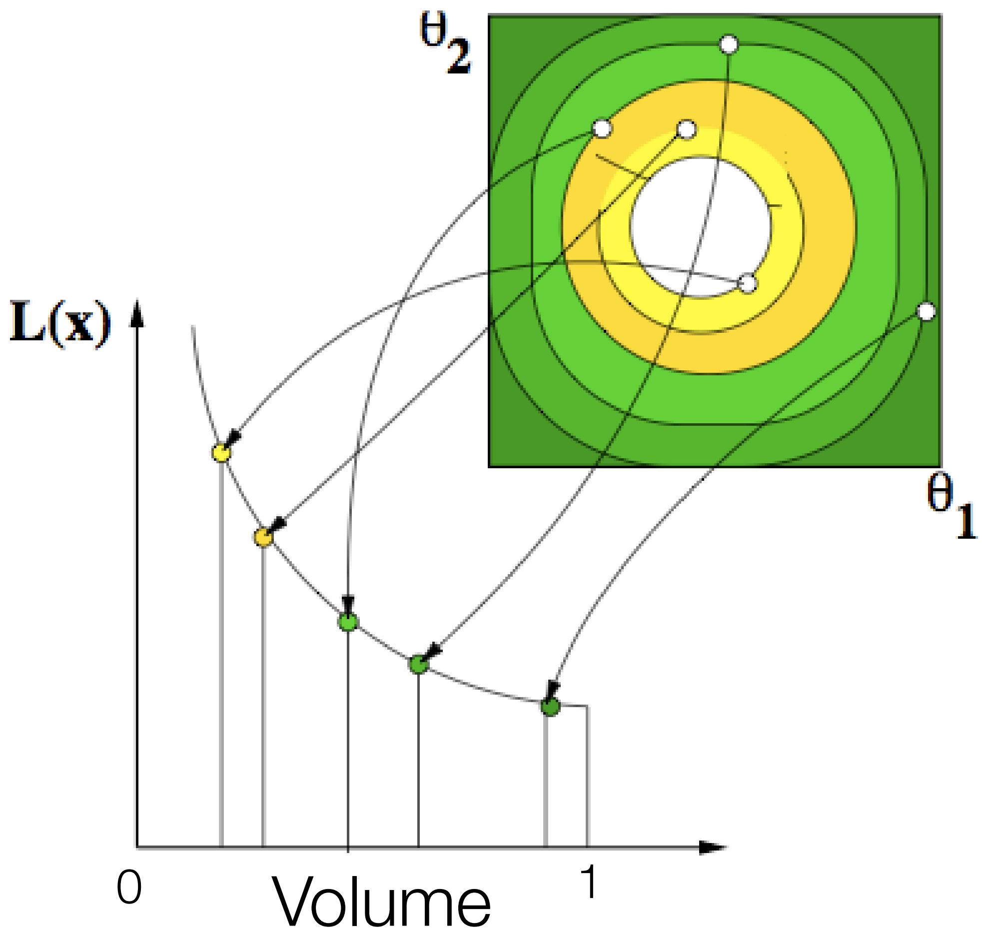


Slice sampling: Handley+2015

Non-volume preserving flows
Moss+2019

Billiard walks with gradients:
Betancourt (2011), Skilling (2012)

Multi-dimensional case



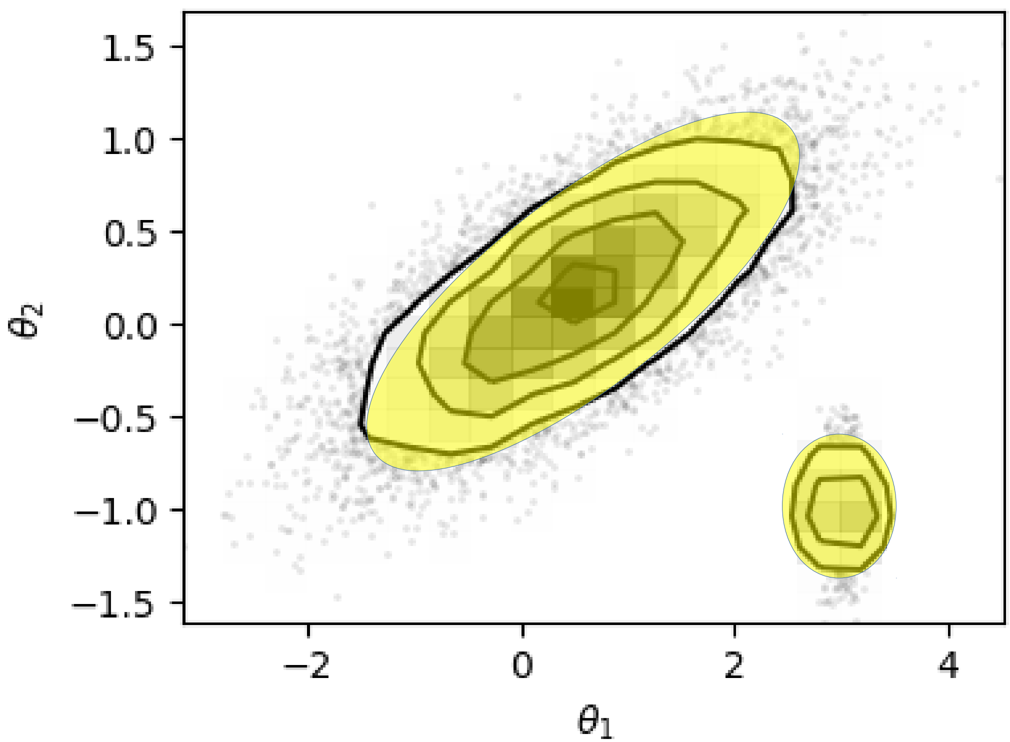
Mukherjee+06

- Ordering of samples well-defined by L
- "Indep. of dim"

  depends primarily on constrained sampler!

- Not limited to continuous spaces of fixed dimensions

Region sampling



- Ellipsoidal sampling

  Mukherjee+06
- Multi-ellipsoidal sampling

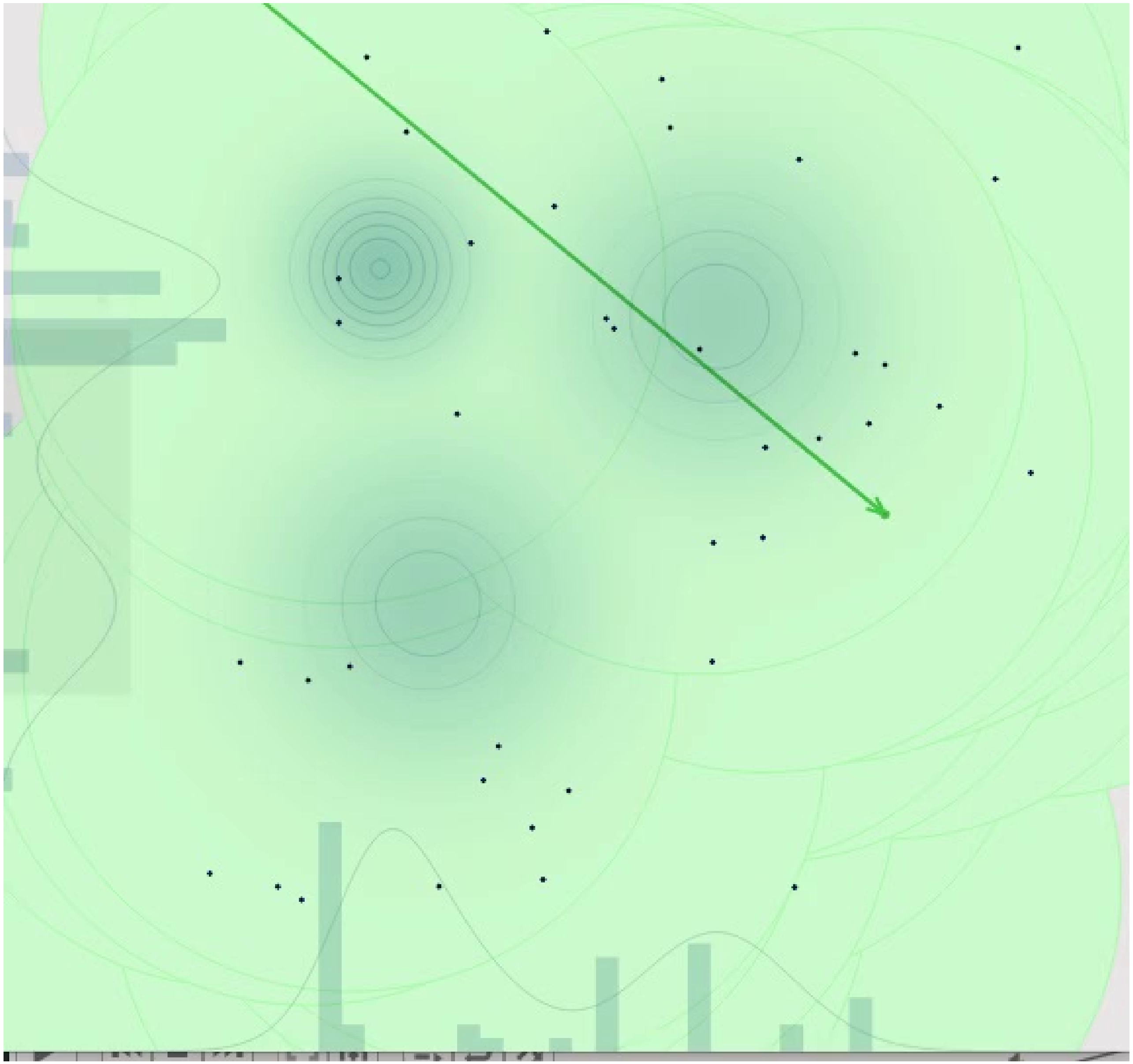
  Shaw+07, Feroz&Hobson08, Feroz+09
  - MultiNest (pymultinest)
- Very popular!

Clustering live points:
x-means (Feroz&Hobson 2008)
Iterative Jarvis-Patrick (Handley+2015)

uniformly sampled, pure, slowly changing

Wilks' theorem
Elliptical distributions

Animation of MLFriends NS



For efficiency:

Reconstruct a region

Sample uniformly

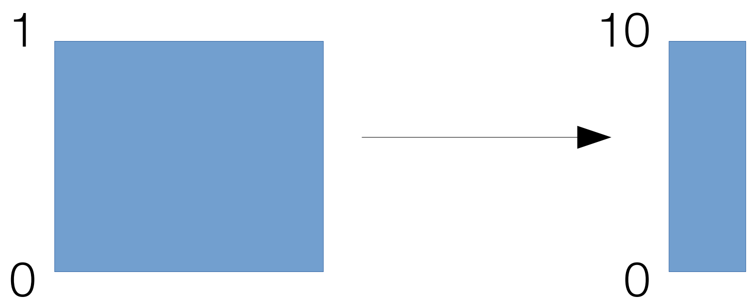
RadFriends: ellipsoid around each point
Circle size cross-validated to recover current points

Alternative: MCMC

https://chi-feng.github.io/mcmc-demo/app.html

Specifying priors

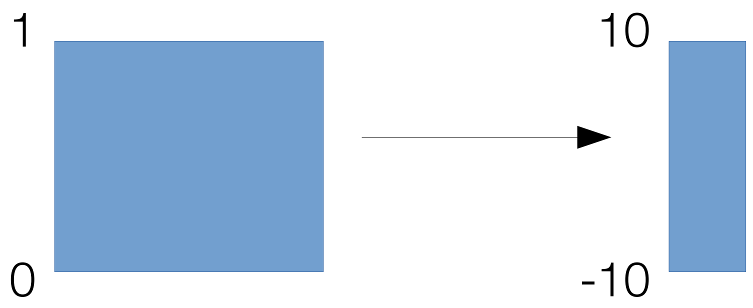
Transform uniform cube to physical parameters




```
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

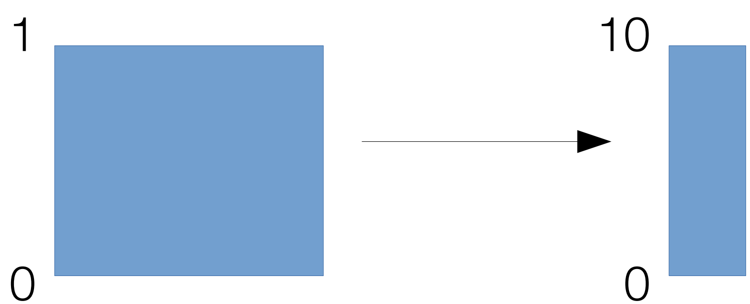
Transform uniform cube to physical parameters



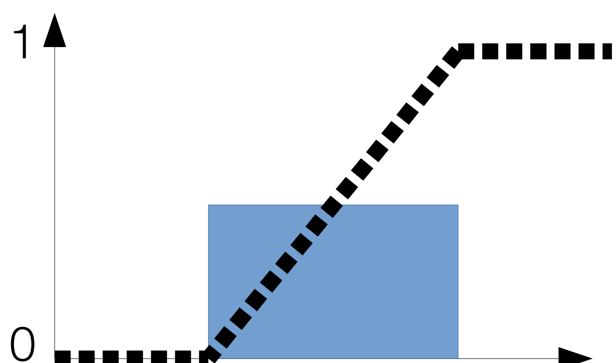

```
def my_prior_transform(cube):
    # cube is a d-dimensional array
    params = cube.copy()
    # from -10 to +10
    params[0] = cube[0] * 20 - 10
    return params
```

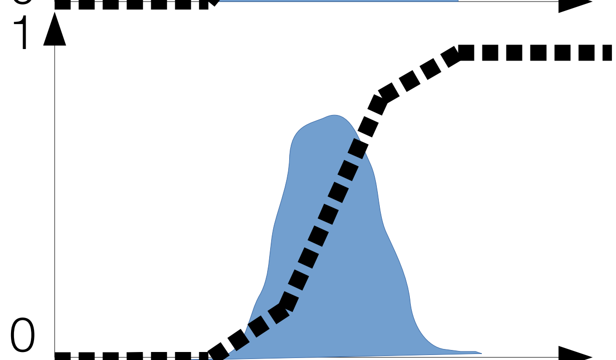
Specifying priors

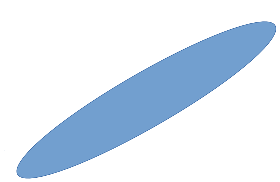
Transform uniform cube to physical parameters




```
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 10^10
    params[1] = 10**(cube[1] * 10)
    # via inverse CDF
    params[2] = rv.ppf(cube[2])
    return params

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

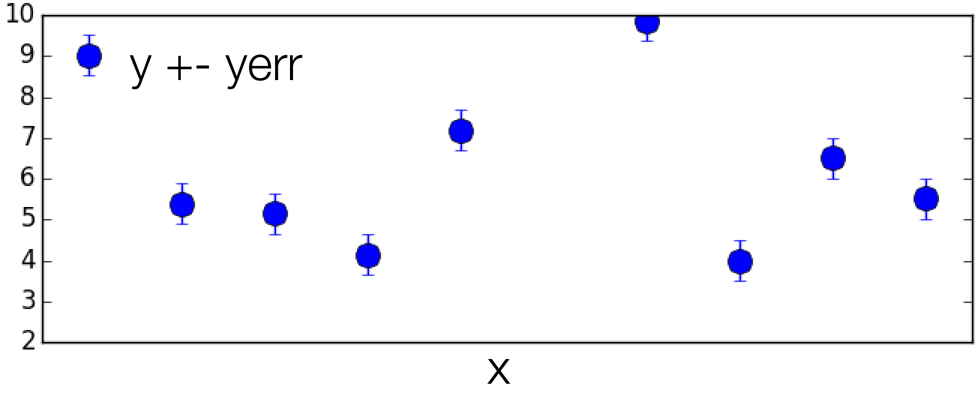






Specifying likelihoods

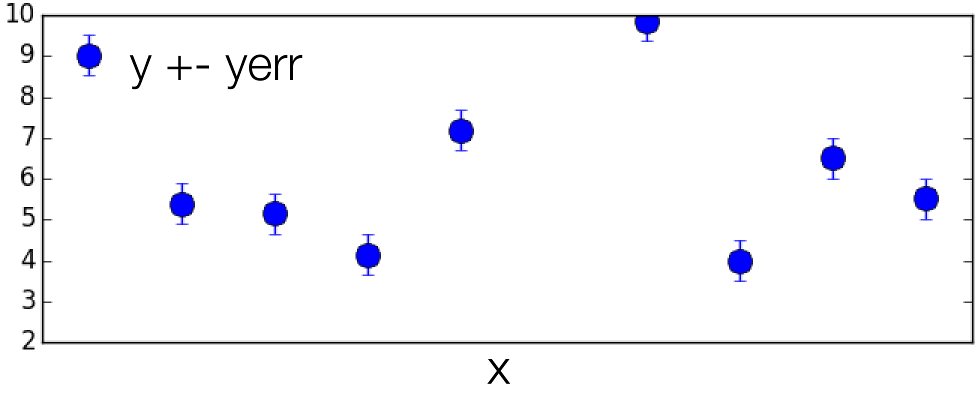
Gaussian example




```
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 -0.5*(((m - y)/yerr)**2).sum()
```

Specifying likelihoods

Gaussian example



Arbitrarily complex model calculation →

Arbitrarily complex data uncertainties →

```
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 -0.5*(((m – y)/yerr)**2).sum()


result = solve(
    LogLikelihood=my_likelihood,
    Prior=my_prior_transform,
    n_dims=4,
    outputfiles_basename='mysine_')
```

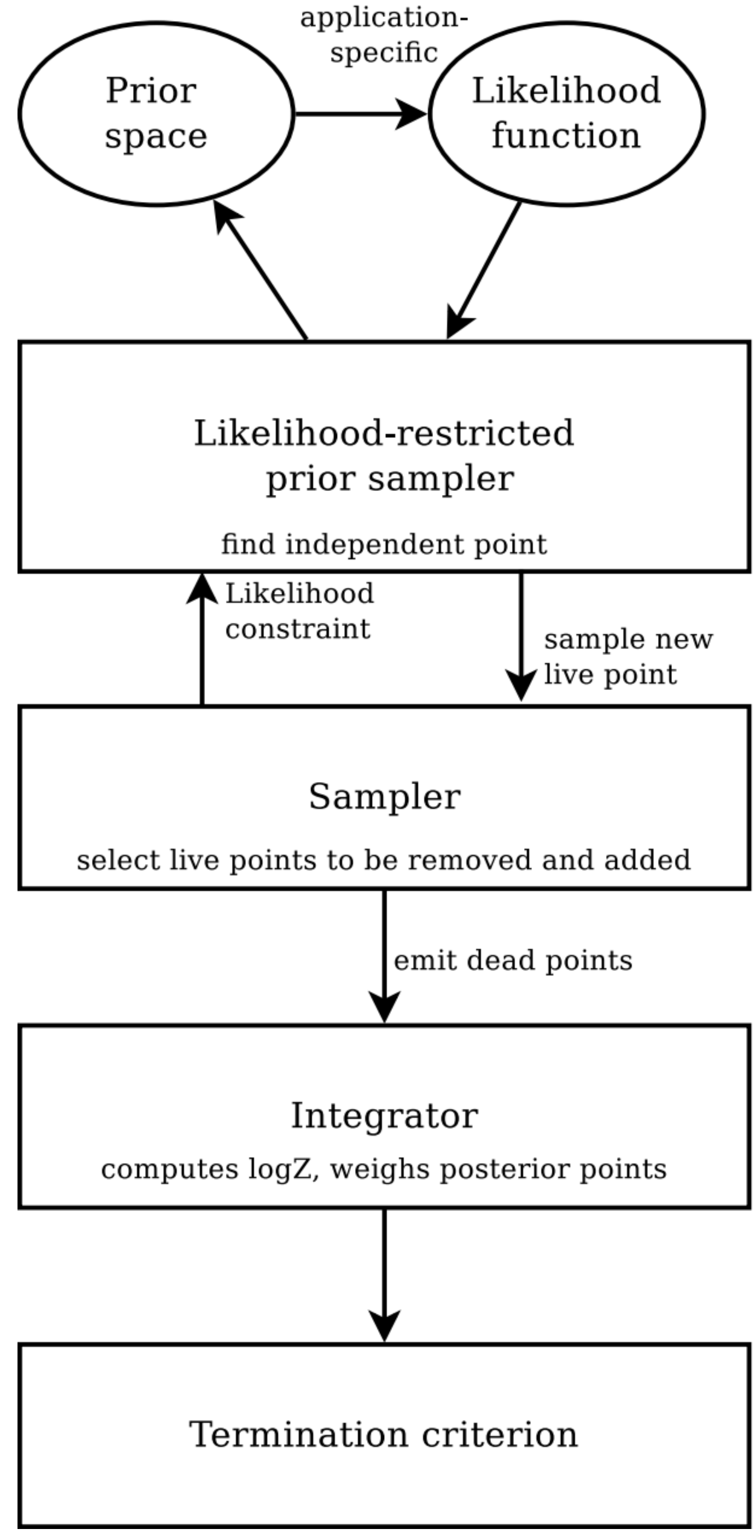
--- Tutorial session ---

https://johannesbuchner.github.io/UltraNest/example-sine-line.html

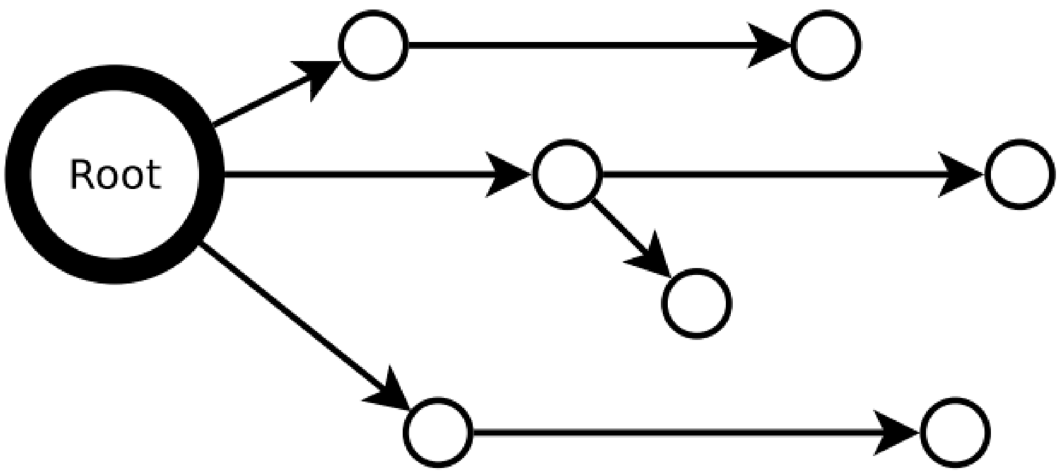
- Fit a time series with a sine
- Parameter estimation and Z computation
- Visualisations

Components

- Advanced NS topics
  - Integrators
  - Error estimates
  - Diagnostics: Tests & Visualisations
  - LRPS
    - Local steps
    - Region based
    - Hybrid
  - Termination criteria
  - Parallelisation
  - Extensions



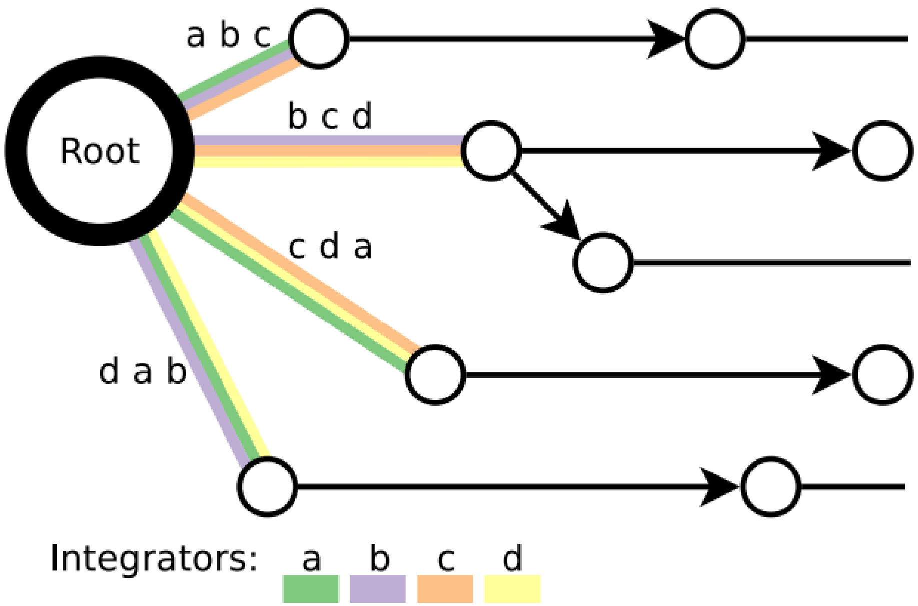
Tree search view



- Root = prior Volume
- Outgoing edges: split volume
- # parallel edges = # of live points

```
Assume queue, sorted by likelihood value
add all children of root node to queue

while queue is not empty:
    Nlive = length of queue
    obtain and remove next node from queue

    store dead node with weight L * V
    shrink volume by 1 / Nlive

    if not terminating:
        expand node

    add all children of node to Q
```

Error estimation



Estimates L noise.

To estimate V noise, change volume shrinkage

dV=1/N → beta(N, 1) randomly

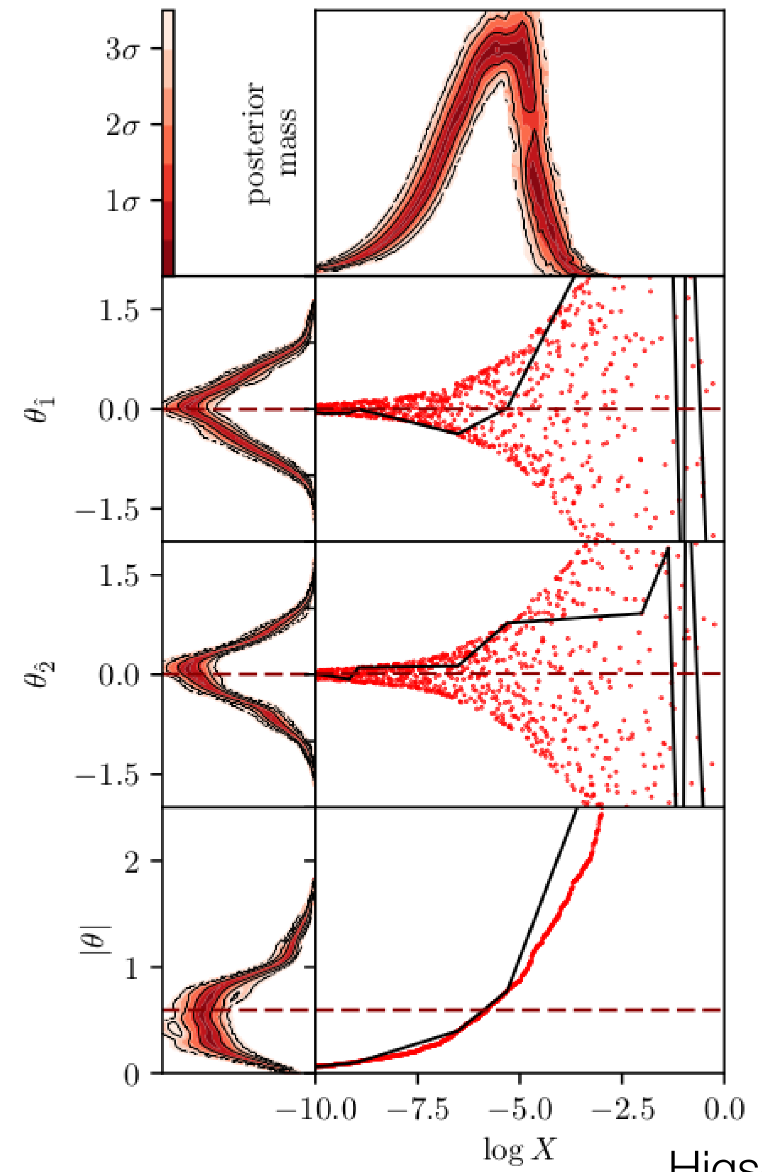
- Multiple integrators
- Blinded to some threads →
- LRPS validation
- Integration validation

Diagnostics: Visualisations

- Each leave-k-out integrator gives results

  → uncertainty in posterior

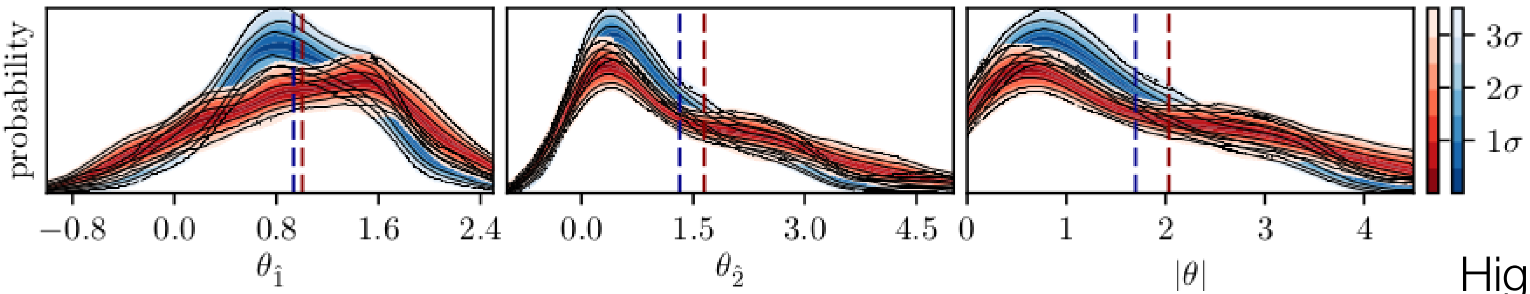
  (+uncertainty in Z)



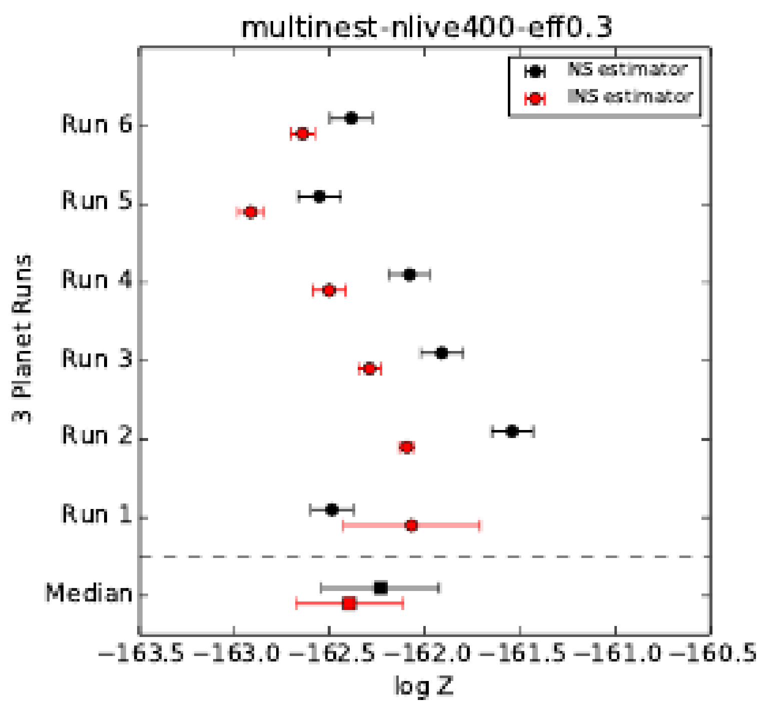
Higson+19

Diagnostics: Visualisations

- Comparison of multiple runs



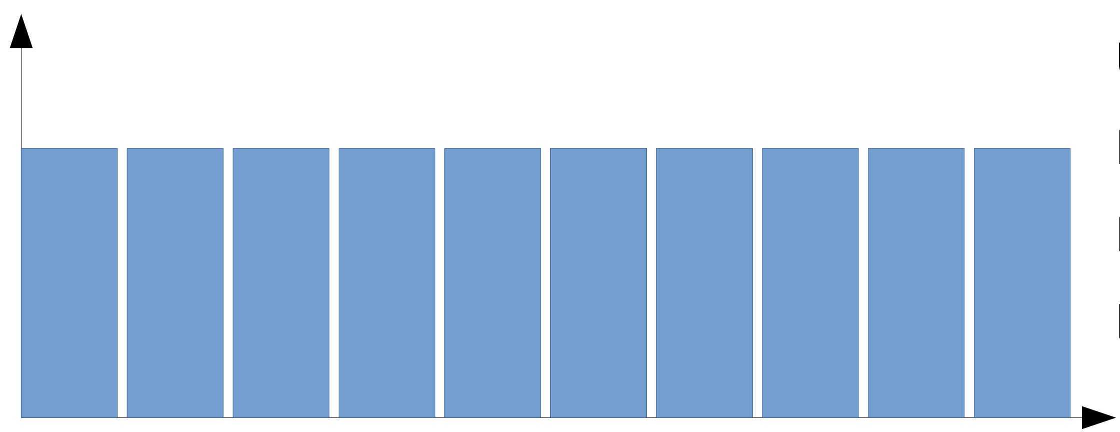
Higson+19



Nelson+20

Diagnostics: tests

Insertion order



Uniform expectation

KS test

Fowlie+2020

MWW test

Parallelisation

- Within the likelihood
- LRPS farmed out
- Removing multiple points
- Merging independent runs
- Analyzing multiple data sets simultaneously

Extensions

- Dynamic NS — Higson+17
- Using thrown-away points
    Importance NS — Chopin & Roberts 07, 08,10
- No hard borders:
    - Diffusive NS
    - Daemonic NS — Brewer+11
- HMC integrations — Habeck15
    -
    - Daemonic NS for smooth border
    - Thermometer as diagnostics

Related algorithms

- Simulated annealing
  - Special cooling schedule
- Sequential Monte Carlo https://arxiv.org/abs/1805.03924
  - Population is reused, special proposal
- Bridge, path sampling
- Importance sampling

Future

- Theory: SMC–NS
- Deeper HMC-NS blends
- More implementations in other languages

Parallelisation
Low latency
Resuming
Easy to install
High-d
Reliable

Review on Nested Sampling methods in prep.