Nets and nests:
accelerated Bayesian inference for astrophysics

Mike Hobson
Astrophysics Group, Cavendish Laboratory, Cambridge

BASP Frontiers workshop: 4–9th September 2011

(see Auld, Bridges, MPH, Gull – astro-ph/0608174;
Auld, Bridges, MPH – astro-ph/0703445
Feroz, MPH – arXiv:0704.3704
Feroz, MPH, Bridges – arXiv:0809.3437, . . . ,
Bridges et al. – arXiv:1011.4306)

OVERVIEW

- Review of standard Bayesian analysis method (cosmologicial case-study)
- Fast likelihood evaluation: neural networks
- Fast and reliable parameter estimation and model selection: nested sampling
- The future: BAMBI
- Conclusions

BASICS OF BAYESIAN DATA ANALYSIS

- Collect a set of N data points D_i $(i = 1, 2, \ldots, N)$, which we denote collectively as the data vector $\boldsymbol{D}$.

- Propose some model (or hypothesis) H for the data, depending on a set of M parameters θ_j $(j = 1, \ldots, M)$, that we denote by the parameter vector $\boldsymbol{\theta}$.

- Apply Bayes' theorem

$$\Pr(\boldsymbol{\theta}|\boldsymbol{D}, H) = \frac{\Pr(\boldsymbol{D}|\boldsymbol{\theta}, H) \Pr(\boldsymbol{\theta}|H)}{\Pr(\boldsymbol{D}|H)} \quad \rightarrow \quad P(\boldsymbol{\theta}) = \frac{L(\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{E}$$

- Parameter estimation: posterior $P(\boldsymbol{\theta})$ is complete inference

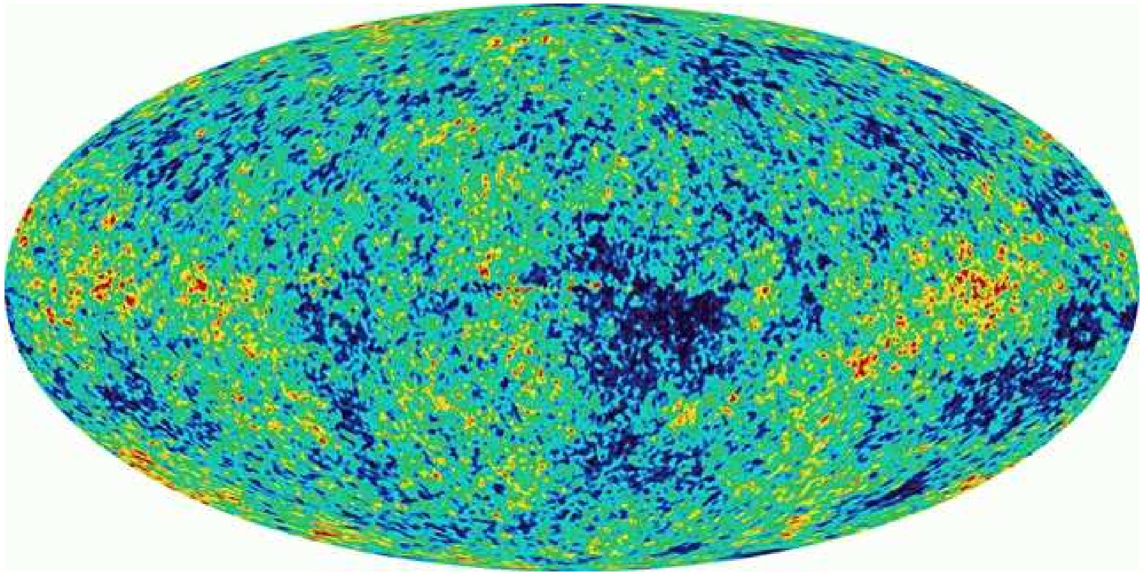
- Model selection: for H_i $(i = 0, 1)$, the probability density associated with $\boldsymbol{D}$ is

$$E_i = \int L_i(\boldsymbol{\theta})\pi_i(\boldsymbol{\theta})\, d\boldsymbol{\theta}$$

then consider ratio

$$\frac{\Pr(H_1|\boldsymbol{d})}{\Pr(H_0|\boldsymbol{d})} = \frac{E_1}{E_0} \frac{\Pr(H_1)}{\Pr(H_0)}$$

Cosmological case-study: CMB anisotropies



- Prior to recombination at $t \sim 300\,000$ yrs (or $z \approx 1100$) plasma and photons tightly coupled and transition to freely propagating photons occured quickly
  $\Rightarrow$ CMB is snapshot of primordial density fluctuations in matter at this epoch

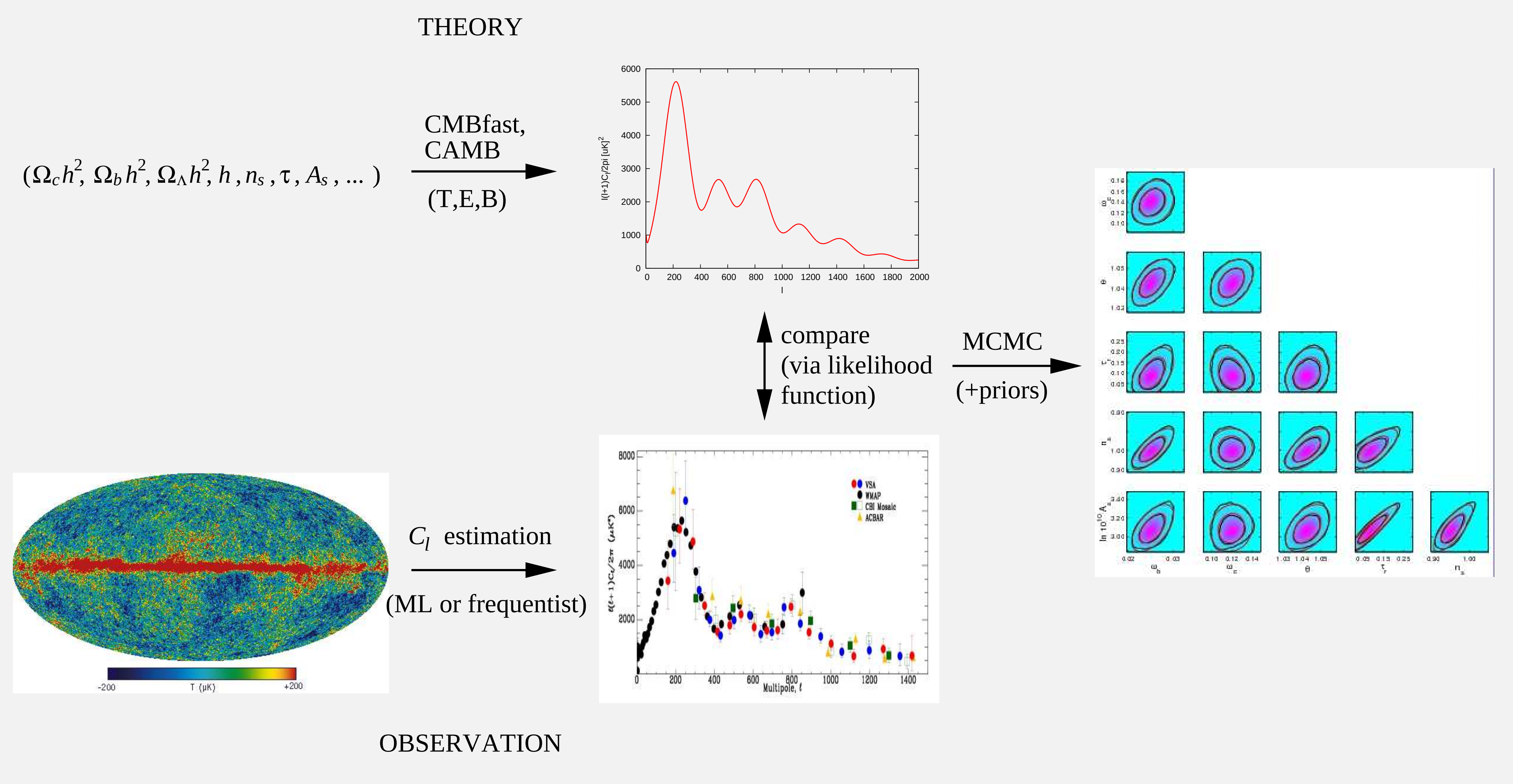
- These density fluctuations are of great interest for two reasons.

(i) These fluctuations later collapse under gravity to form all structure in the Universe

(ii) In the inflationary model, the form of these primordial density fluctuations are a powerful probe of the physics of the very early Universe

BAYESIAN STATISTICS AND COSMOLOGY

- Most obvious example: standard CMB data analysis pipeline



THEORY

$$(\Omega_c h^2,\ \Omega_b h^2,\ \Omega_\Lambda h^2,\ h\,,\ n_s\,,\ \tau\,,\ A_s\,,\ \dots\ ) \xrightarrow[\text{(T,E,B)}]{\text{CMBfast, CAMB}}$$

compare (via likelihood function)

$\xrightarrow[\text{(+priors)}]{\text{MCMC}}$

$\xrightarrow[\text{(ML or frequentist)}]{C_l \text{ estimation}}$

OBSERVATION

- But many others: signal enhancement, signal separation, object detection, . . .

PROBLEMS WITH STANDARD APPROACH

- Slow likelihood evaluation
  - C_ℓ prediction (CAMB): $\sim$ 10 secs for flat model, $\sim$ 50 secs for non-flat model
  - Likelihood function for some CMB slow: WMAP3 $\sim$ 60 secs, WMAP5 $\sim$ 10 secs
  - Likelihood function slow for some complementary datasets: 2dF, SDSS, . . .
- Slow exploration of parameter space
  - Cosmological parameter estimation typically requires $\sim 10^5$ MCMC samples

  ⇒ Full analysis requires ~ 30 days CPU time (excluding C_ℓ estimation)

  ⇒ Perform analysis in $\sim 1-4$ days on COSMOS supercomputer depending on $N_{\rm CPU}$ available ($\times 2-3$ for 'naughty user ranking', queues, etc. . .)
- AND. . . $\times \sim 10$ for cosmological model selection using MCMC thermodynamic integration

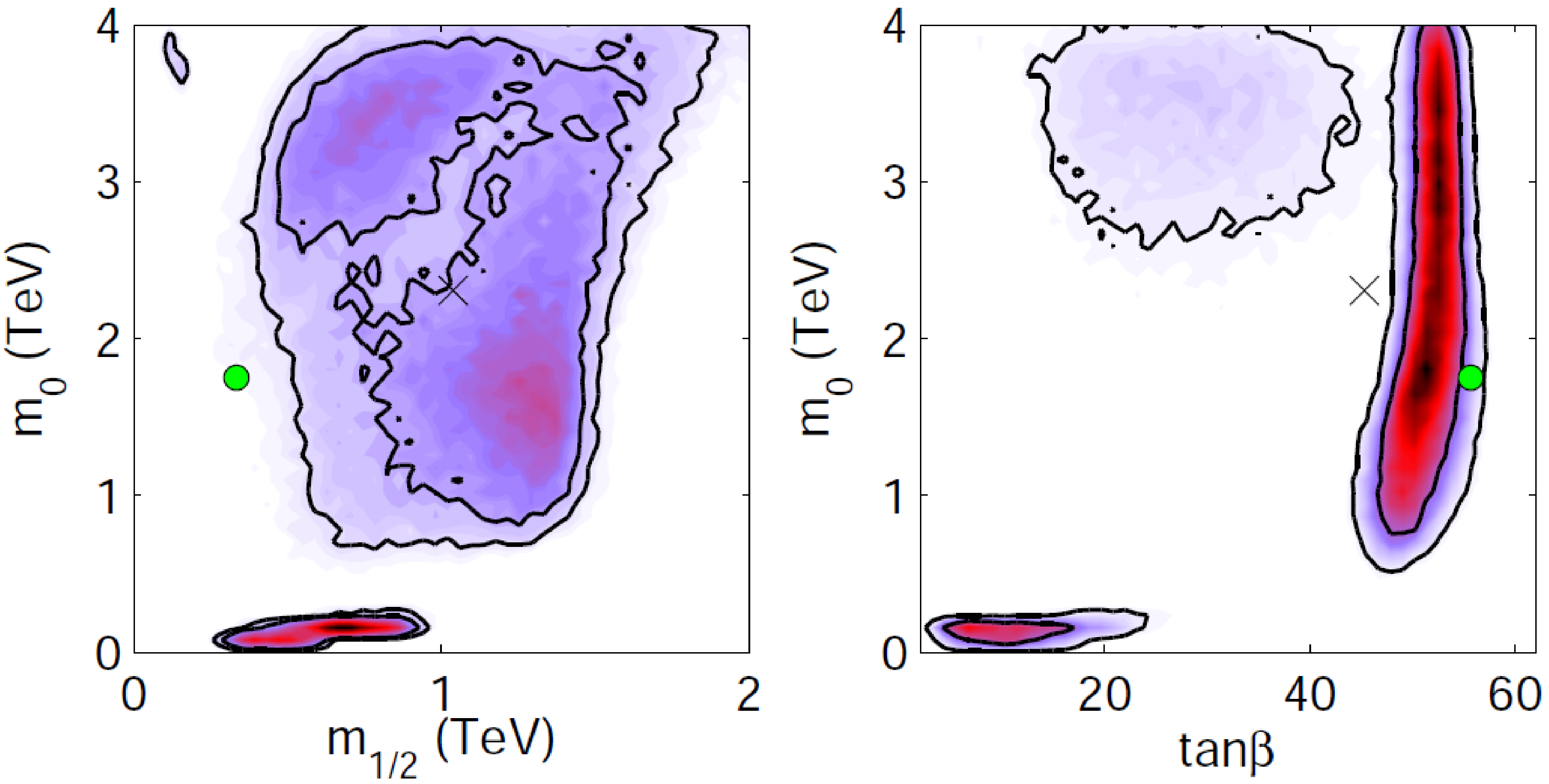
PROBLEMS WITH STANDARD APPROACH

- Incomplete exploration of parameter space

  Likelihood function of some models is complex and multimodal with narrow ridges

  ⇒ exploration with conventional MCMC methods challenging

  ⇒ low sampling efficiency and potentially incomplete exploration

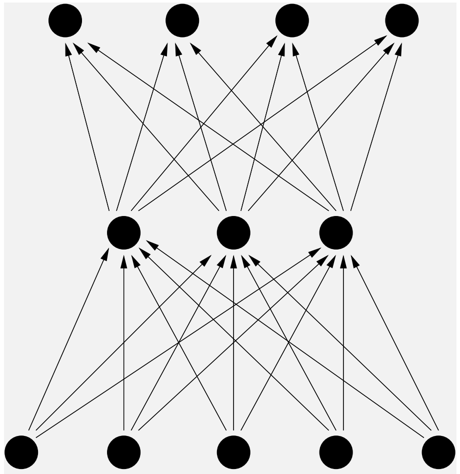


1: Neural networks: fast likelihood evaluation

MULTI-LAYER PERCEPTRON NEURAL NETWORKS

- MLP = feed-forward network composed of ordered layers of perceptrons

- Consider 3-layer MLP here: input layer, hidden layer and output layer



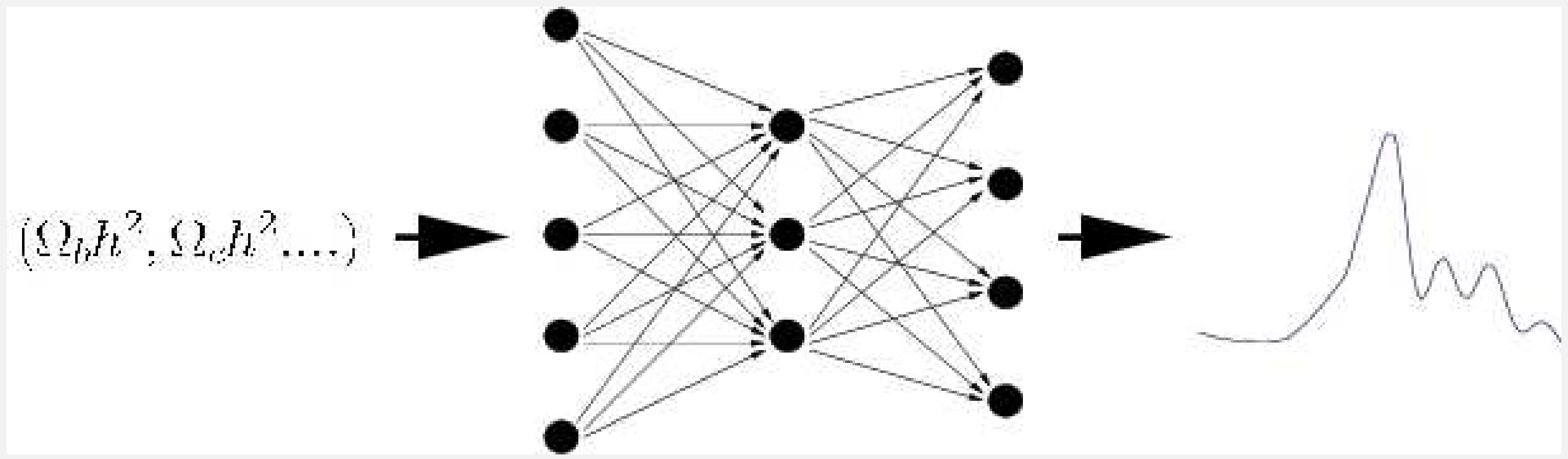
hidden layer: $h_j = g^{(1)}(f_j^{(1)});\quad f_j^{(1)} = \sum_l w_{jl}^{(1)} x_l + b_j^{(1)},$

output layer: $y_i = g^{(2)}(f_i^{(2)});\quad f_i^{(2)} = \sum_l w_{ij}^{(2)} h_j + b_i^{(2)},$

- Use non-linear activation function ($g_1(x) = \tanh x$) on outputs of all hidden layer neurons; use $g_2(x) = x$

- Any L_2-function $f : \Re^n \rightarrow \Re^m$, can be approximated to arbitrary mean square error accuracy by a 3-layer MLP

NEURAL NETWORK APPROACH TO COSMOLOGY

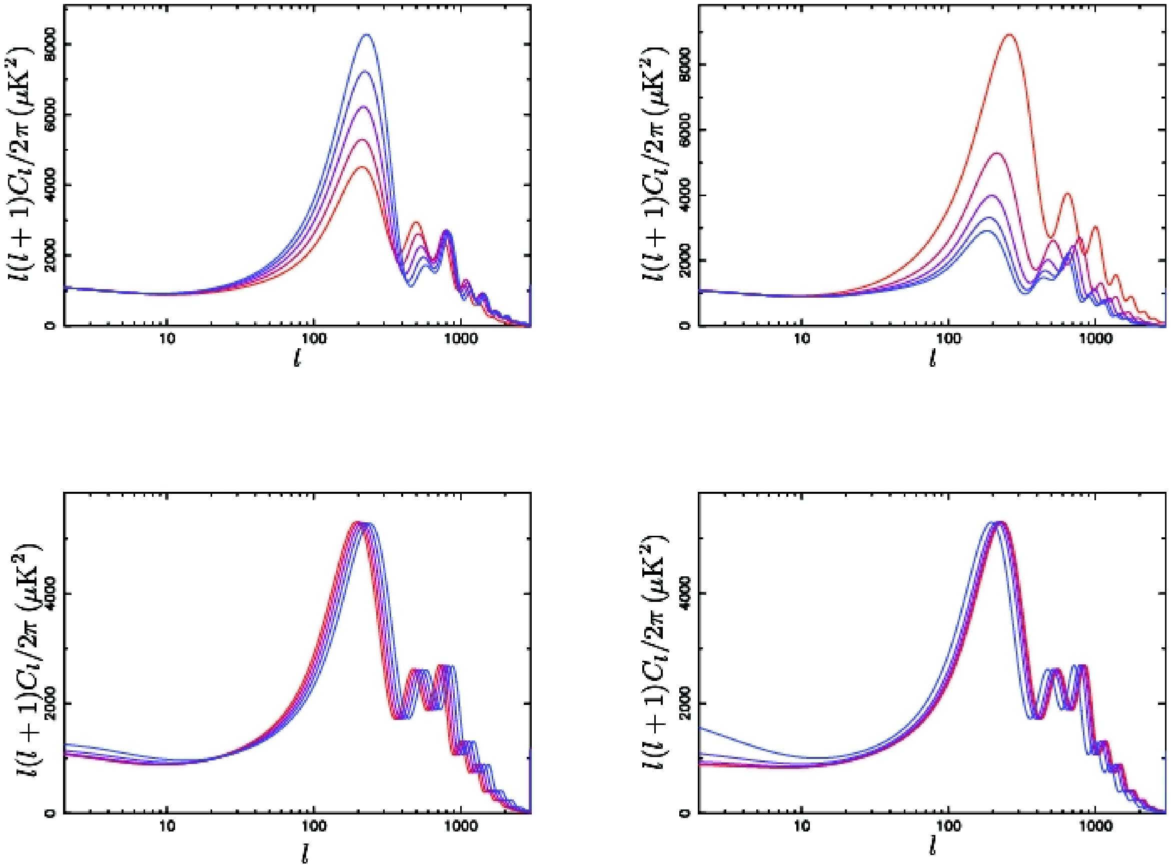


- Any analysis must relate model parameters Θ to observable quantities, such as power spectra or likelihoods directly. Can view e.g. CAMB simply as a mapping $\Theta \to C_\ell$ and engineer a computationally efficient representation of this function

- Neural networks easy: random training data, scales linearly with dimension $\Rightarrow$ train regression neural network to 'learn cosmology'

- Train separate networks outputting C_ℓ^{TT} , C_ℓ^{TE}, C_ℓ^{EE}, C_ℓ^{BB} + matter power transfer function $T(k)$ + WMAP, 2dF, SDSS likelihoods

COSMOLOGICAL MODEL ‘LEARNED’



- 7 parameter non-Flat ΛCDM model: $\{\Omega_{\mathrm{k}}, \Omega_{\mathrm{b}}h^2, \Omega_{\mathrm{c}}h^2, \theta, \tau, A_s, n_s\}$

- Parameter ranges: 8σ box around WMAP + SDSS + 2dF best-fit point

- Network(s) outputs: $C_l^{TT,TE,EE}$, $T(k)$, WMAP, 2dF, SDSS likelihoods

Neural network training

- Training data: $\mathcal{D} = (\boldsymbol{x}^k, \boldsymbol{t}^k)$
  – randomly select $\sim$ 1000s points in box in cosmological parameter space: $\boldsymbol{x}^k$
  – calculate C_ℓ and $T(k)$ spectra using CAMB (at fixed ℓ and k values)
  – calculate likelihoods using WMAP, 2dF, SDSS codes

- Minimise χ^2 with respect to network parameters $\boldsymbol{a} = (\boldsymbol{w}, \boldsymbol{b})$:

$$\chi^2(\boldsymbol{a}) = \frac{1}{2}\sum_k \sum_i \left[t_i^{(k)} - y_i(\boldsymbol{x}^{(k)}; \boldsymbol{a})\right]^2$$

- Highly non-linear function in 1000s of dimensions $\Rightarrow$ use MEMSYS optimiser on:

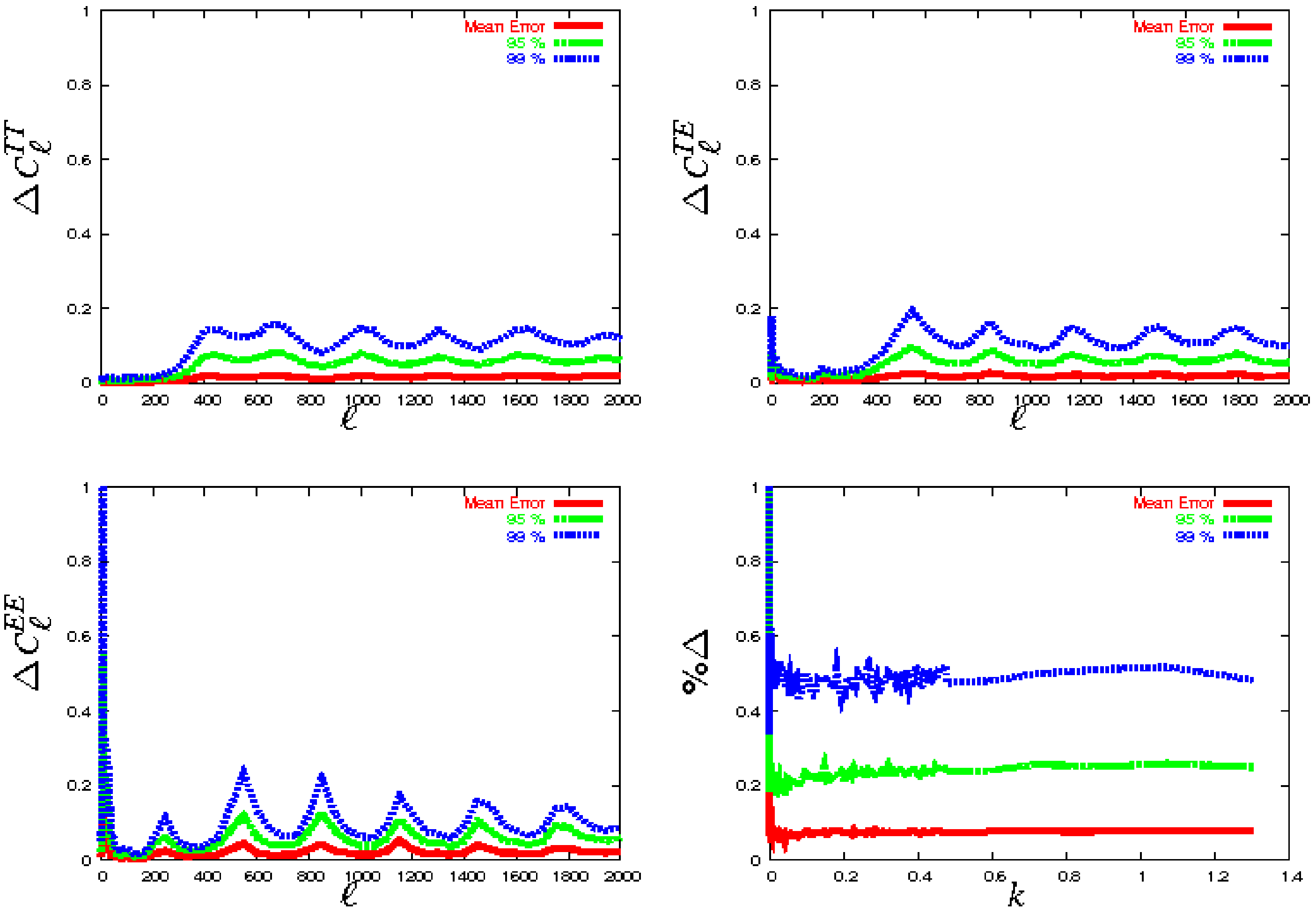
$$F(\boldsymbol{a}) = -\chi^2(\boldsymbol{a}) + \alpha S(\boldsymbol{a})$$

- Increments α down the maximum entropy trajectory (starting from $\alpha = \infty$) until the error term dominates; trains in $\sim$ 10 mins with 50 hidden nodes (max evidence)

- Create separate test data to evaluate accuracy

NN RESULTS: SPECTRA ACCURACY AND SPEED

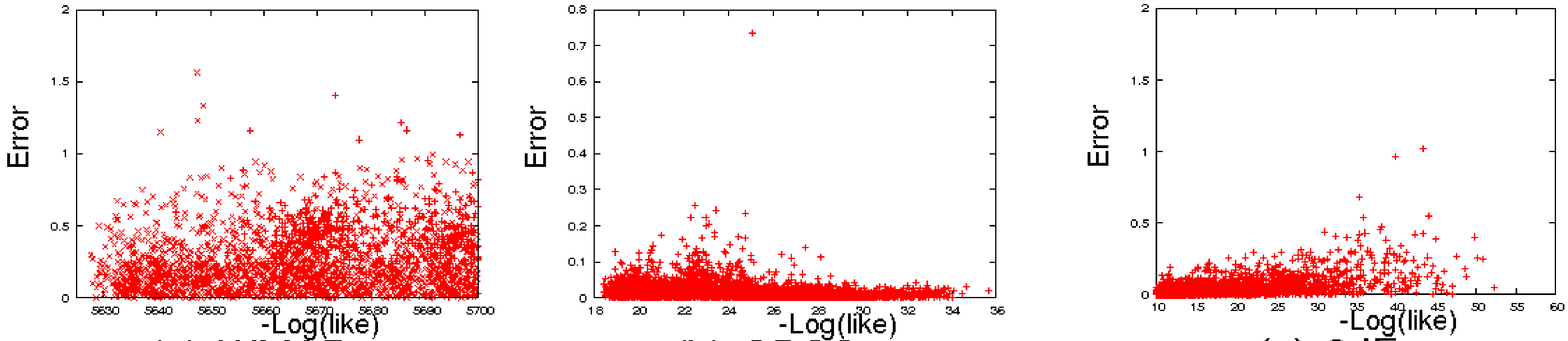
- C_ℓ and $T(k)$ accuracy in cosmic variance units (correlation on test data = 0.99998):



- CosmoNet speed of C_ℓ spectra generation $\sim 10^4$ times faster than CAMB

NN RESULTS: LIKELIHOODS ACCURACY AND SPEED

- Likelihood accuracy (correlation on test data > 0.999999)

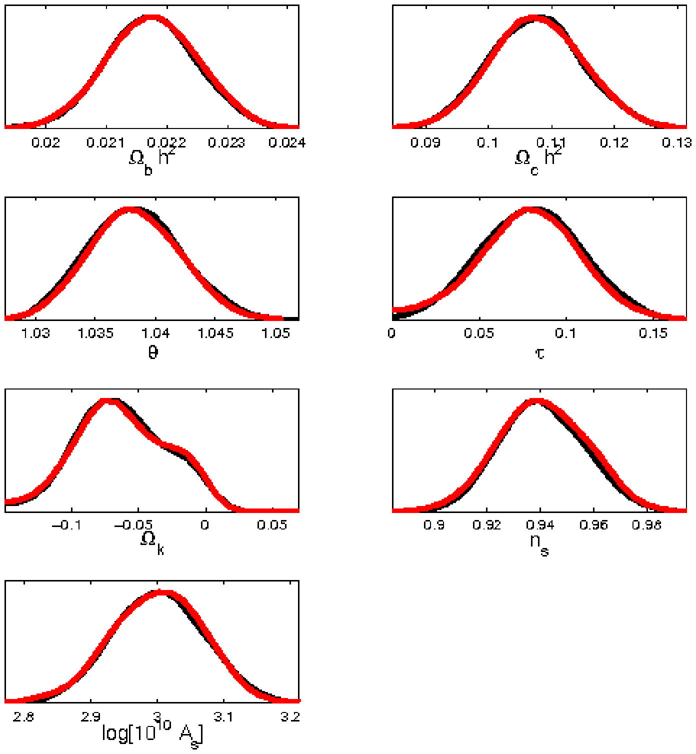


(a) WMAP (b) SDSS (c) 2dF

- CosmoNet likelihood evaluation $\sim 10^3$ times faster than WMAP code

COSMOLOGICAL PARAMETER CONSTRAINTS USING SPECTRA

- Standard method versus CosmoNet spectra → standard likelihood codes:



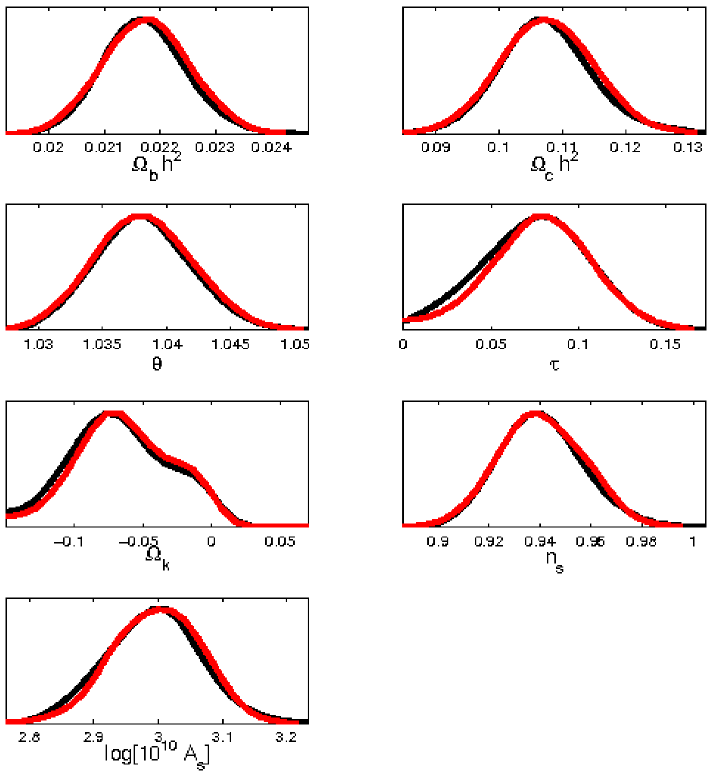
- Posteriors differ by less than inter-chain variance (20,000 samples in total)

- Standard method: ∼ 300 CPU hrs (CosmoMC)
  CosmoNet spectra + standard likelihoods: ∼ 30 CPU hrs (CosmoMC)
  CosmoNet spectra + standard likelihoods: ∼ 3 CPU hrs (MultiNest – see later!)

- Note: WMAP likelihood code is bottleneck (other experiment likelihoods fast)

COSMOLOGICAL PARAMETER CONSTRAINTS USING LIKELIHOODS

- Standard method versus CosmoNet likelihoods:

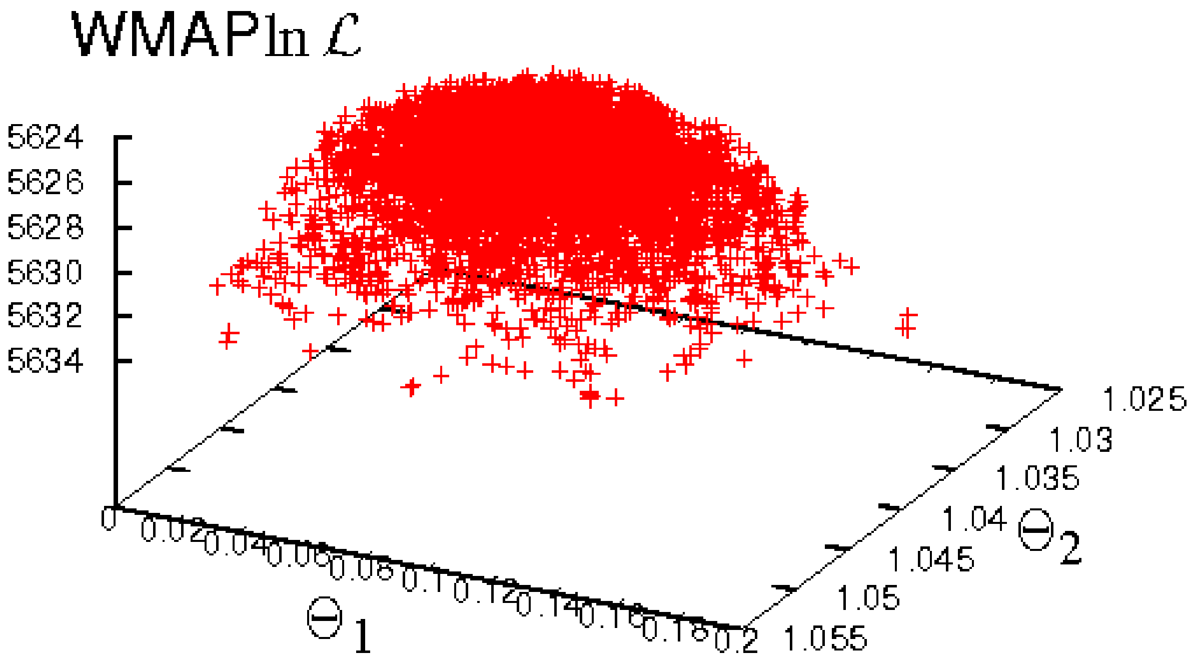


- Posteriors differ by less than inter-chain variance (20,000 samples in total)

- Standard method: $\sim$ 300 CPU hrs (CosmoMC)
  CosmoNet likelihoods: $\sim$ 2 CPU hrs (CosmoMC);
  CosmoNet likelihoods: $\sim$ 10 CPU mins (MultiNest – see later!)

INCREASING NETWORK ACCURACY FOR EVIDENCE CALCULATION

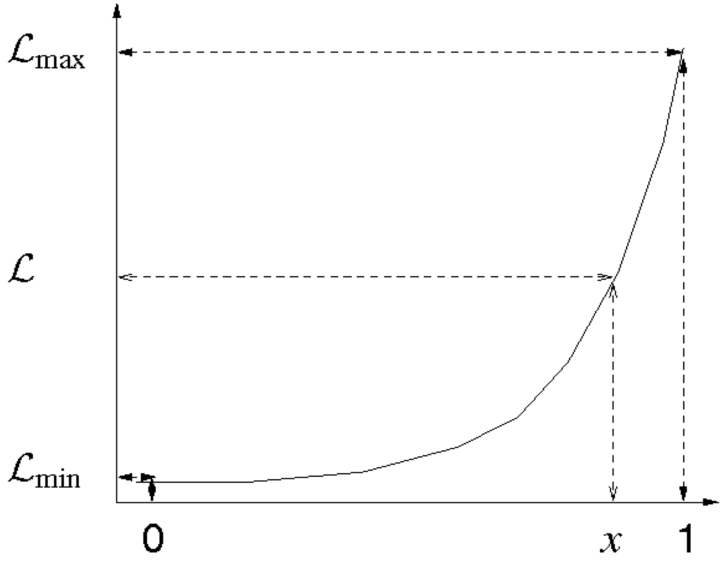
- BUT for model selection, require likelihood evaluations to greater accuracy than needed for parameter estimation, since tails of distribution are important



- Attaining sufficient accuracy in network hindered by wide variation in WMAP log-likelihood, ranging over several thousand units from peak to edge of prior

INCREASING NETWORK ACCURACY FOR EVIDENCE CALCULATION

- Transform ln L values to linear scale $[0 \rightarrow 1]$
  $\Rightarrow$ improve accuracy in wings ($\sim$ few log units)



- Include $\sim$ 50% posterior samples in training data
  $\Rightarrow$ improve accuracy near peak ($\sim$ 0.01 log units)

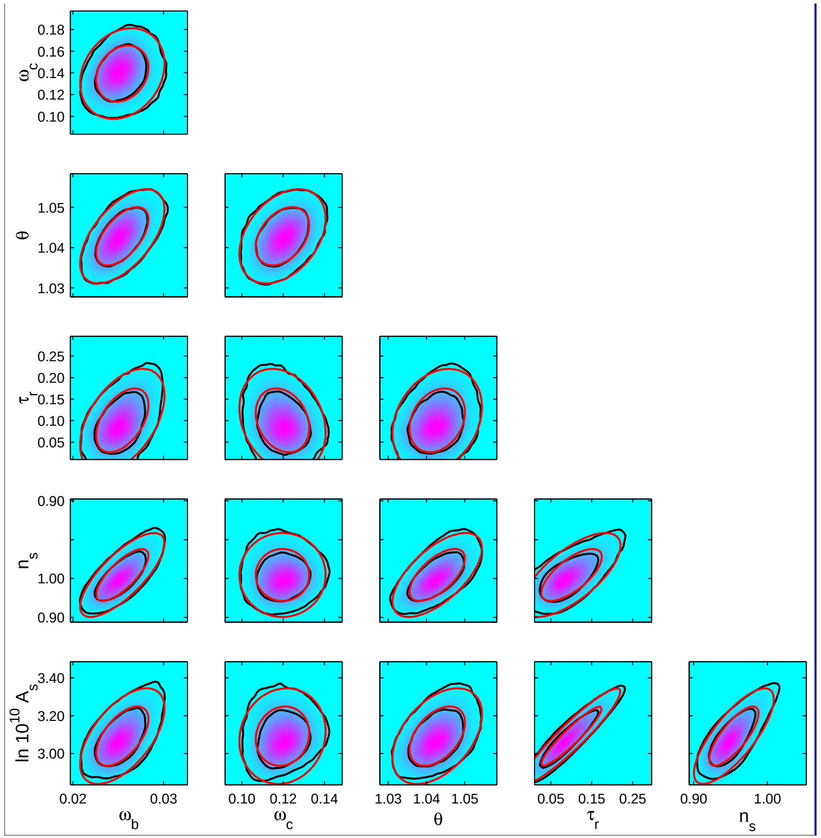
$\Rightarrow$ Network evidence estimates indistinguishable from those using CAMB

$\Rightarrow$ For cosmological model (using MCMC thermodynamic integration):
Standard+CosmoMC $E = 5636.6 \pm 0.2$ in $\sim$ 2500 CPU hrs (CosmoMC)
CosmoNet+CosmoMC $E = 5636.6 \pm 0.2$ in $\sim$ 20 CPU hrs (CosmoMC)
CosmoNet+MultiNest $E = 5636.6 \pm 0.2$ in $\sim$ 10 CPU mins (MultiNest)

2: Nested sampling: fast and reliable parameter estimation and model selection

Some cosmological posteriors

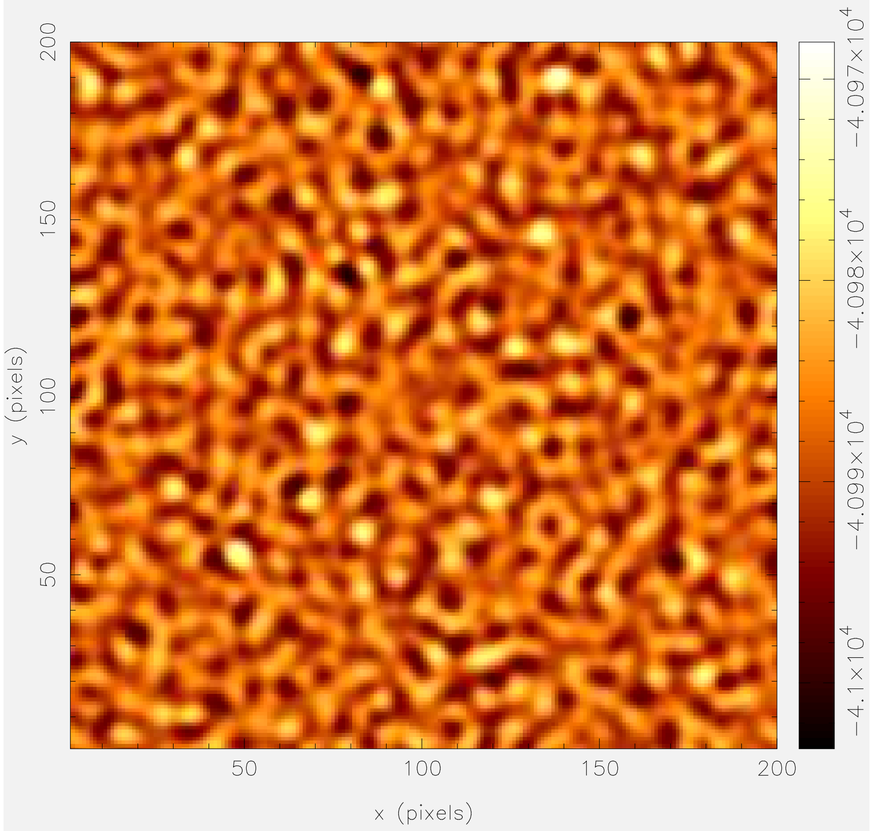
- Some cosmological posteriors are nice, others are nasty



ΛCDM: $\boldsymbol{\theta} = (\omega_b, \omega_c, \theta, \tau, \ln A, n_s)$

using CMB+SDSS+HST data

(Trotta 2004)



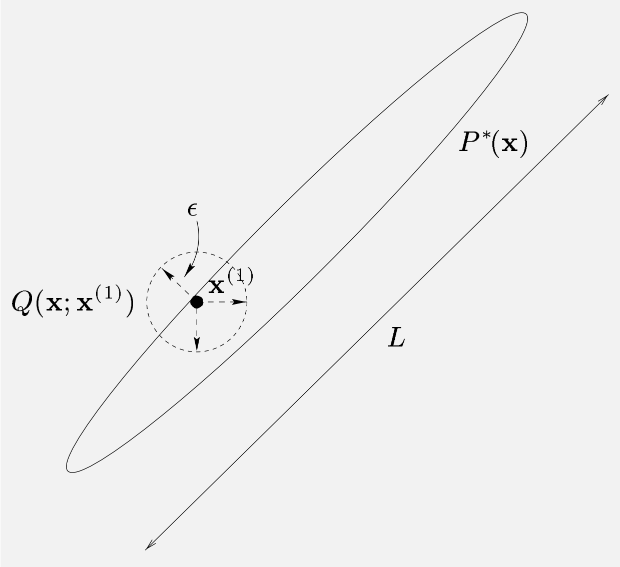
Detecting SZ clusters in CMB:

$\boldsymbol{\theta} = (X, Y, A, R)$

(Hobson & McLachlan 2003)

- Posterior exploration (parameter estimation) and integration (model selection) traditionally performed using MCMC sampling

METROPOLIS–HASTINGS ALGORITHM



- Metropolis–Hastings algorithm to sample $P(\boldsymbol{\theta})$:
  - start at arbitrary point $\boldsymbol{\theta}_0$
  - at each step draw trial point $\boldsymbol{\theta}' \leftarrow Q(\boldsymbol{\theta}'|\boldsymbol{\theta}_n)$ from proposal distribution
  - calculate ratio $r = P(\boldsymbol{\theta}')Q(\boldsymbol{\theta}_n|\boldsymbol{\theta}')/P(\boldsymbol{\theta}_n)Q(\boldsymbol{\theta}'|\boldsymbol{\theta}_n)$
  - if $r \geq 1$ accept $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}'$;
    if $r < 1$ accept with probability r, else $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n$

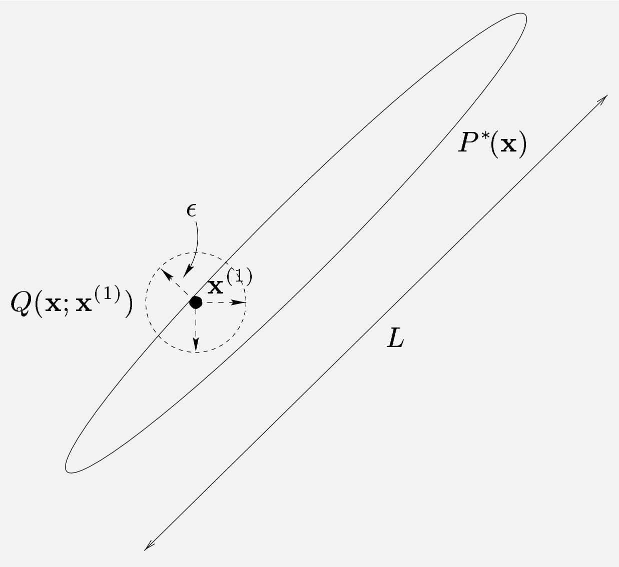
- Implementation of basic MH algorithm is trivial:

```
Initialise $\boldsymbol{\theta}_0$; set $n = 0$
Repeat [
  Sample a point $\boldsymbol{\theta}'$ from $Q(\cdot|\boldsymbol{\theta}_n)$
  Sample a uniform [0,1] random variable $U$
  If $U \leq \alpha(\boldsymbol{\theta}', \boldsymbol{\theta}_n)$ set $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}'$, else $\boldsymbol{\theta}_{n+1} = \boldsymbol{\theta}_n$
  Increment $n$]
```

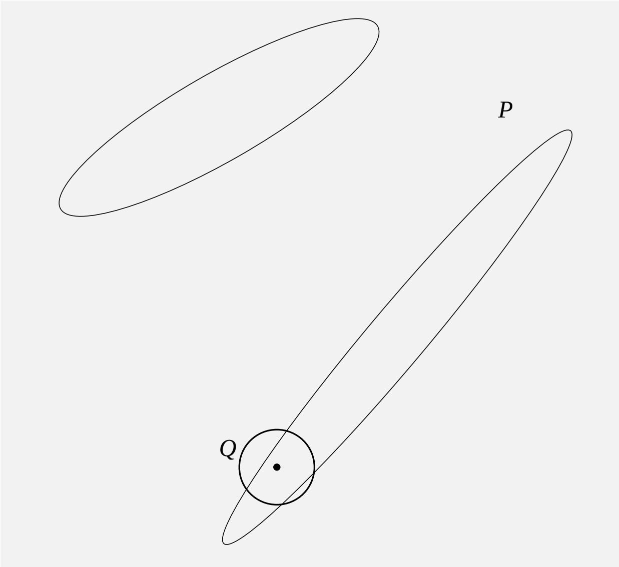
- After initial burn-in period, any (positive) proposal $Q \Rightarrow$ convergence to $P(\boldsymbol{\theta})$

- Common choice for Q is multivariate Gaussian centred on $\boldsymbol{\theta}_n$ (CosmoMC)

METROPOLIS–HASTINGS ALGORITHM: SOME PROBLEMS



- But... choice of Q strongly affects rate of convergence and sampling efficiency.
- Large proposal width $\epsilon \Rightarrow$ trial points rarely accepted
- Small proposal width $\epsilon \Rightarrow$ chain explores $P(\boldsymbol{\theta})$ by a random walk – very slow
- If largest scale of $P(\boldsymbol{\theta})$ is L
  $\Rightarrow$ typical diffusion time $t \sim (L/\epsilon)^2$
- If smallest scale of $P(\boldsymbol{\theta})$ is ℓ
  $\Rightarrow$ need $\epsilon \sim \ell \Rightarrow$ diffusion time $t \sim (L/\ell)^2$



- Particularly bad for multimodal distributions
- Transitions between distant modes very rare
- No choice of proposal width ϵ works
- Standard convergence tests will suggest converged, but actually only true in a subset of modes

METROPOLIS–HASTINGS ALGORITHM: SOME PARTIAL FIXES

- Set proposal width ϵ by trial and error to achieve acceptance ratio $\sim$ 0.5, or dynamically during burn-in, but must fix thereafter

- Multiple (non-interacting) chains sometimes useful

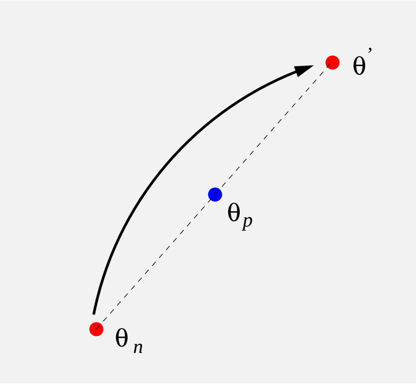
- Annealing schedules or multi-temperature chains

- Several sequential proposals: each updating only some parameters

- Innovative proposals, e.g Gibbs, Hamiltonian, slice sampling, genetic algorithms, . . .

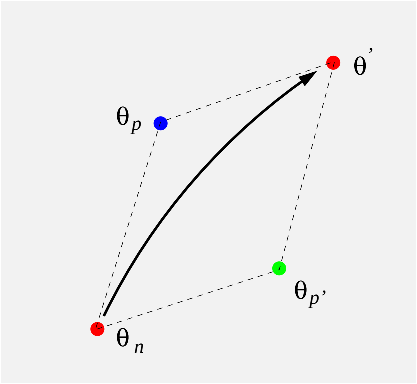
- Compound proposal: multiple proposals Q_i each chosen at random with probability p_i

- Use of multiple *interacting* chains, e.g.



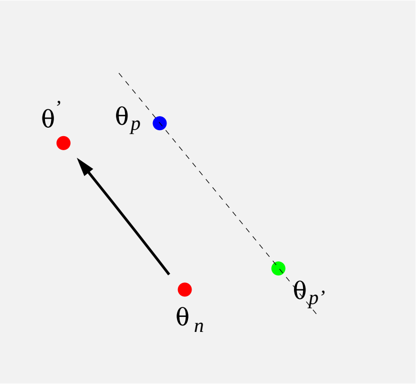
leapfrog

$$\boldsymbol{\theta}' = 2\boldsymbol{\theta}_{\rm p} - \boldsymbol{\theta}_n$$



cross-walk

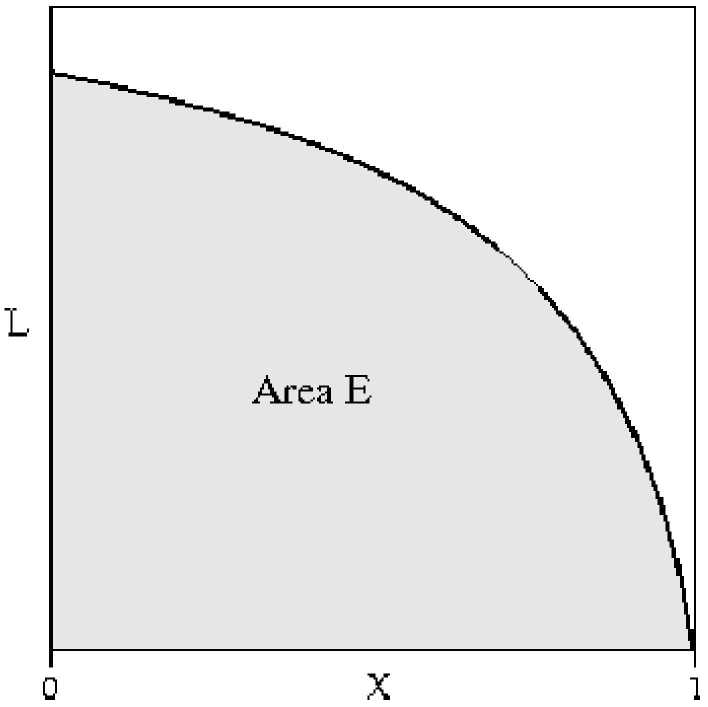
$$\boldsymbol{\theta}' = \boldsymbol{\theta}_{\rm p} + \boldsymbol{\theta}_{\rm p'} - \boldsymbol{\theta}_n$$



guided-walk

$$\boldsymbol{\theta}' = \boldsymbol{\theta}_n + (\boldsymbol{\theta}_{\rm p} - \boldsymbol{\theta}_{\rm p'})$$

Nested Sampling



- New technique for efficient evidence evaluation (and posterior samples) (Skilling 2004)

- Define $X(\lambda) = \int_{L(\boldsymbol{\theta})>\lambda} \pi(\boldsymbol{\theta})\, d\boldsymbol{\theta}$

- Write inverse $L(X)$, i.e. $L(X(\lambda)) = \lambda$

- Evidence becomes one-dimensional integral

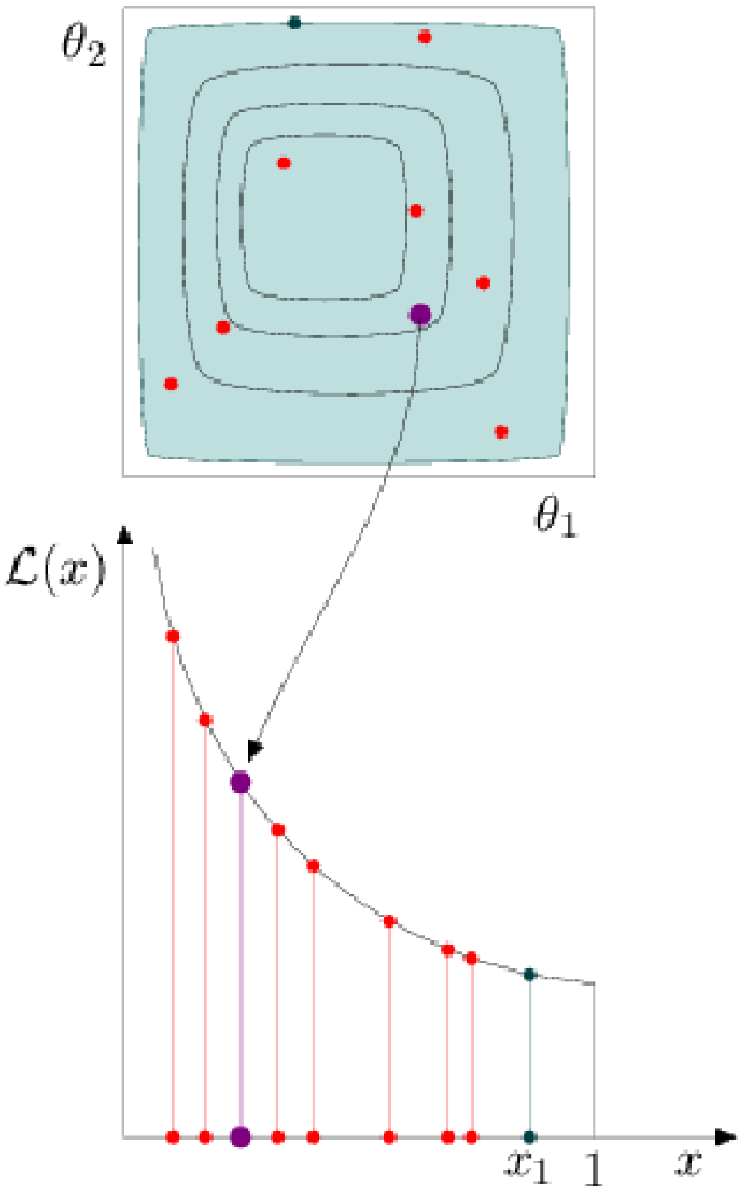
$$E = \int L(\boldsymbol{\theta})\pi(\boldsymbol{\theta})\, d\boldsymbol{\theta} = \int_0^1 L(X)\, dX$$

- Suppose can evaluate $L_j = L(X_j)$ where $0 < X_m < \cdots < X_2 < X_1 < 1$

  $\Rightarrow$ estimate E by any numerical method

$$E = \sum_{j=1}^{m} L_j w_j$$

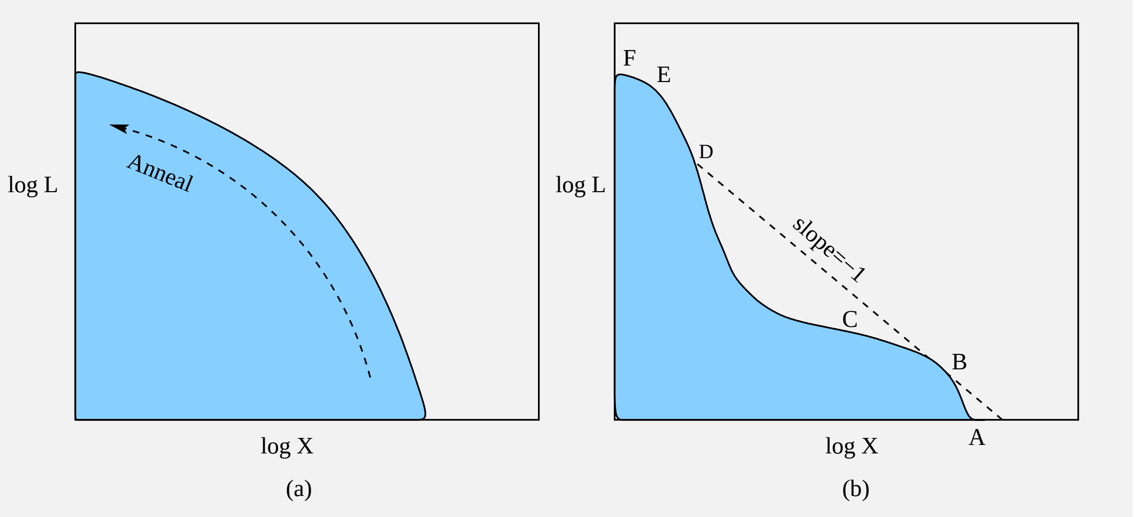
($w_j = \frac{1}{2}(X_{j-1} - X_{j+1})$ for trapezium rule)



Nested sampling approach to summation:

1. Set $i = 0$; initially $X_0 = 1$, $E = 0$
2. Sample N points $\{\boldsymbol{\theta}_j\}$ randomly from $\pi(\boldsymbol{\theta})$ and calculate their likelihoods
3. Set $i \to i + 1$
4. Find point with lowest likelihood value (L_i)
5. Remaining prior volume $X_i = t_i X_{i-1}$ where $\mathrm{Pr}(t_i|N) = N t_i^{N-1}$;
   or just use $\langle t_i \rangle = N/(N+1)$
6. Increment evidence $E \to E + L_i w_i$
7. Remove lowest point from active set
8. Replace with new point sampled from $\pi(\boldsymbol{\theta})$ within hard-edged region $L(\boldsymbol{\theta}) > L_i$
9. If $L_{\mathrm{max}} X_i < \alpha E$ (where some tolerance)
   $\Rightarrow E \to E + X_i \sum_{j=1}^{N} L(\boldsymbol{\theta}_j)/N$; stop
   else goto 3

- Advantages:
  - typically requires around few 100 times fewer samples than thermodynamic integration to calculate evidence to same accuracy (plus error estimate)
  - does not get stuck at phase changes like thermodynamic integration

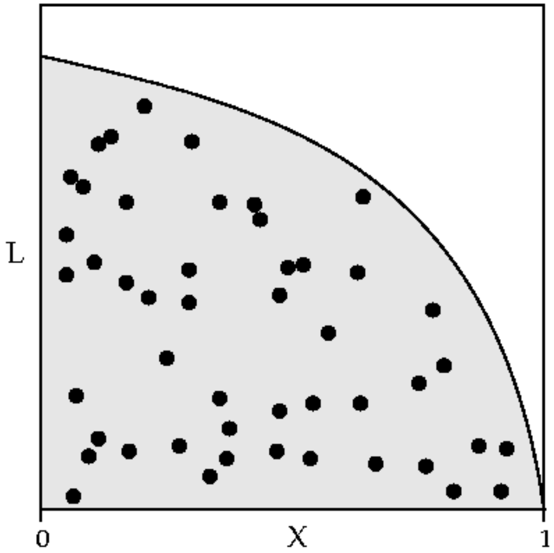


- As $\lambda : 0 \rightarrow 1$ annealing should track along curve
- But $\frac{d \log L}{d \log X} = -\frac{1}{\lambda}$, so annealing schedule cannot navigate convex regions (phase changes)

- Bonus: posterior samples easily obtained as a by-product. Simply take full sequence of sampled points $\boldsymbol{\theta}_j$ and weight jth sample by $p_j = L_j w_j / E$, e.g.

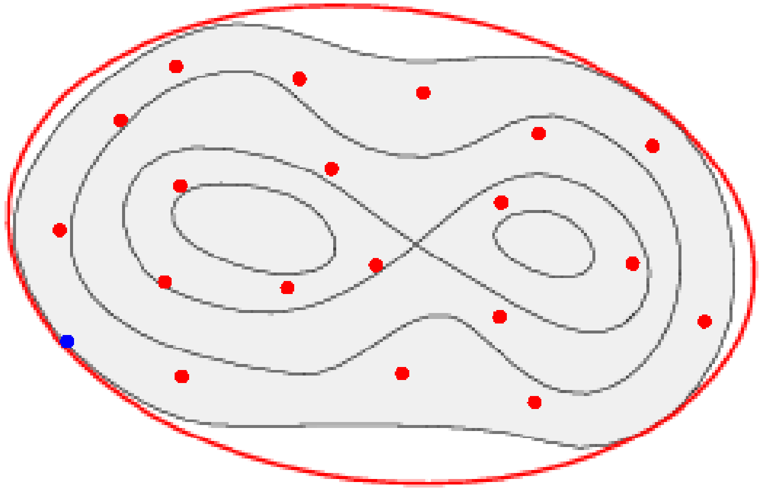
$$\mu_Q = \sum_j p_j Q(\boldsymbol{\theta}_j),$$

$$\sigma_Q^2 = \sum_j (p_j Q(\boldsymbol{\theta}_j) - \mu_Q)^2$$



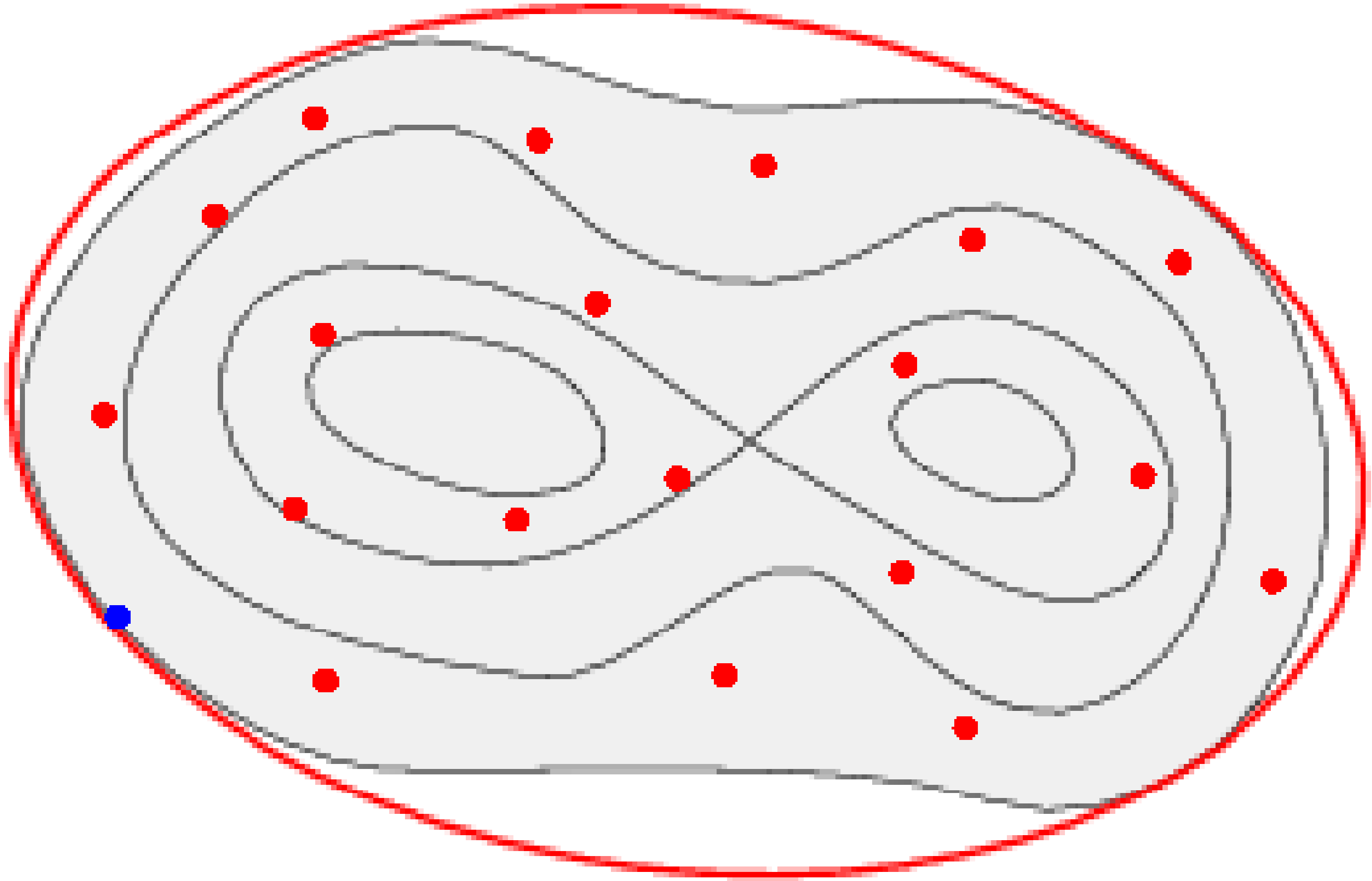
PRACTICAL CONSIDERATIONS

- Most challenging task: at each iteration i must replace removed point with one sampled from $\pi(\boldsymbol{\theta})$ within complicated, hard-edged region $L(\boldsymbol{\theta}) > L_i$
- Simple MCMC using Metropolis–Hastings possible, but can be inefficient
- Mukherjee et al. (2005) fit ellipsoid to active points, enlarge to try to account for non-ellipsoidal likelihood contour, and sample within it using simple, exact method

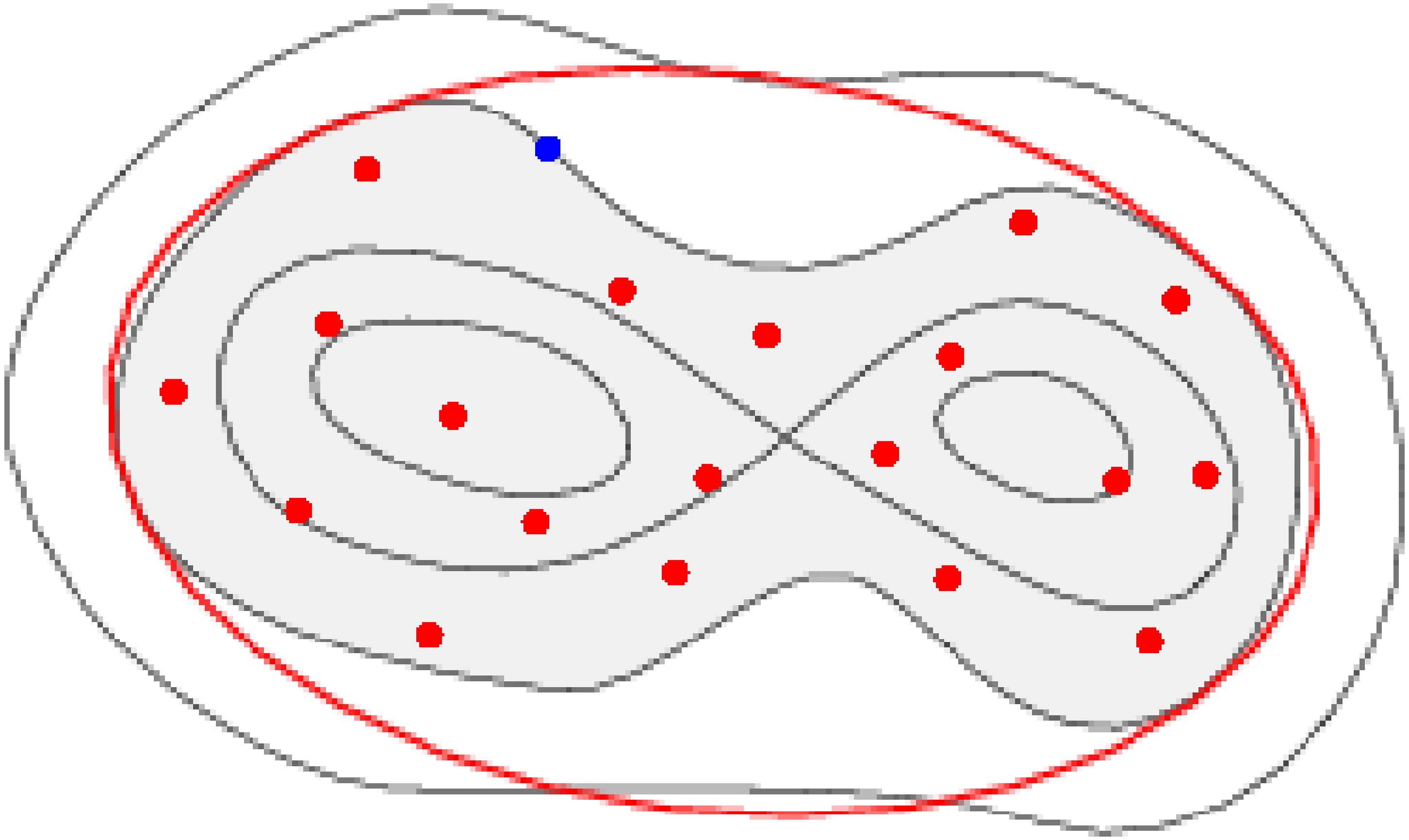


- Demonstrated high-efficiency and robustness on simple unimodal cosmological posteriors (~ 100 times faster evidence evaluation cf. thermodynamic integration)
- But. . . still problematic for multimodal/ degenerate posteriors

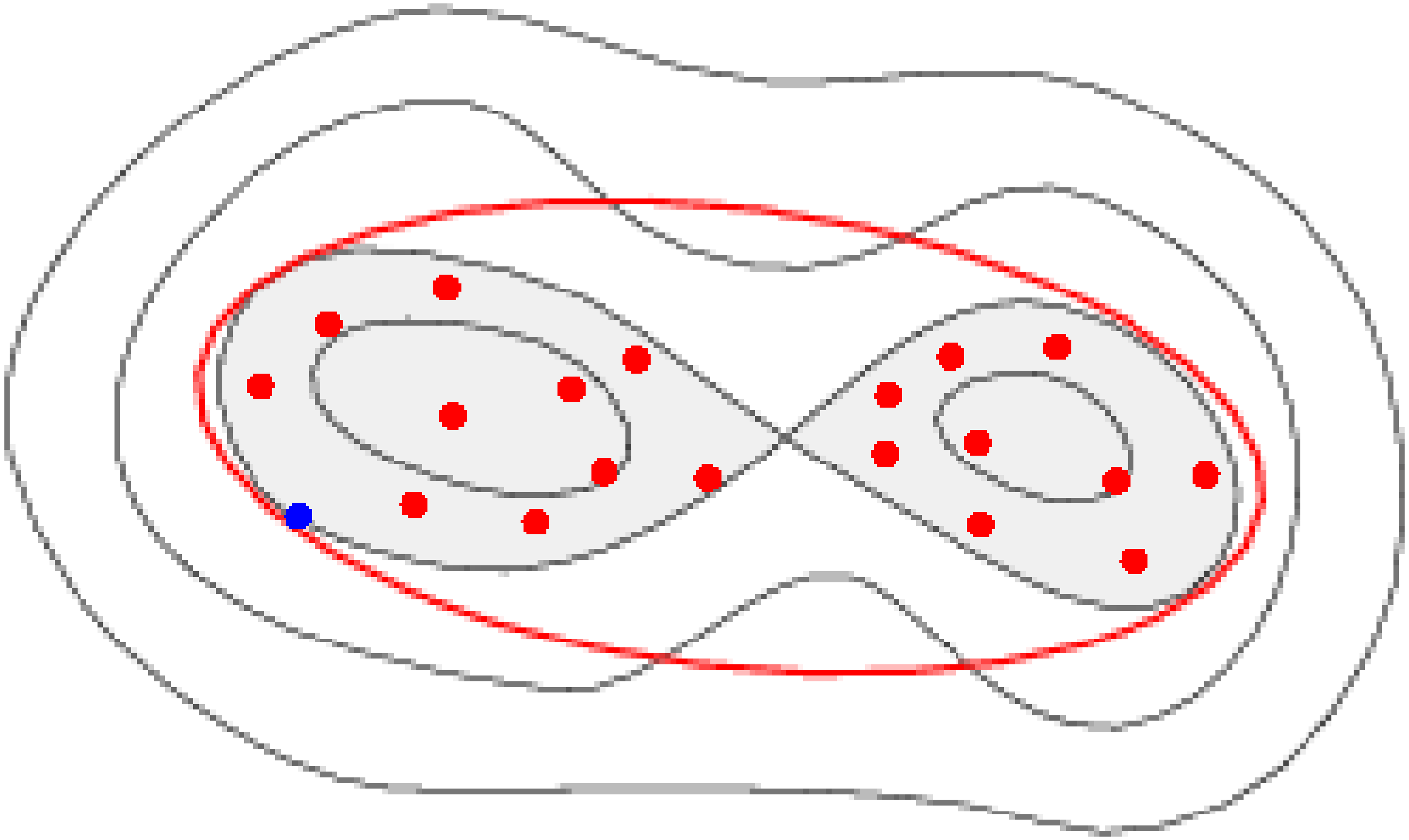
Problem with elliptical region sampling ($N = 20$):



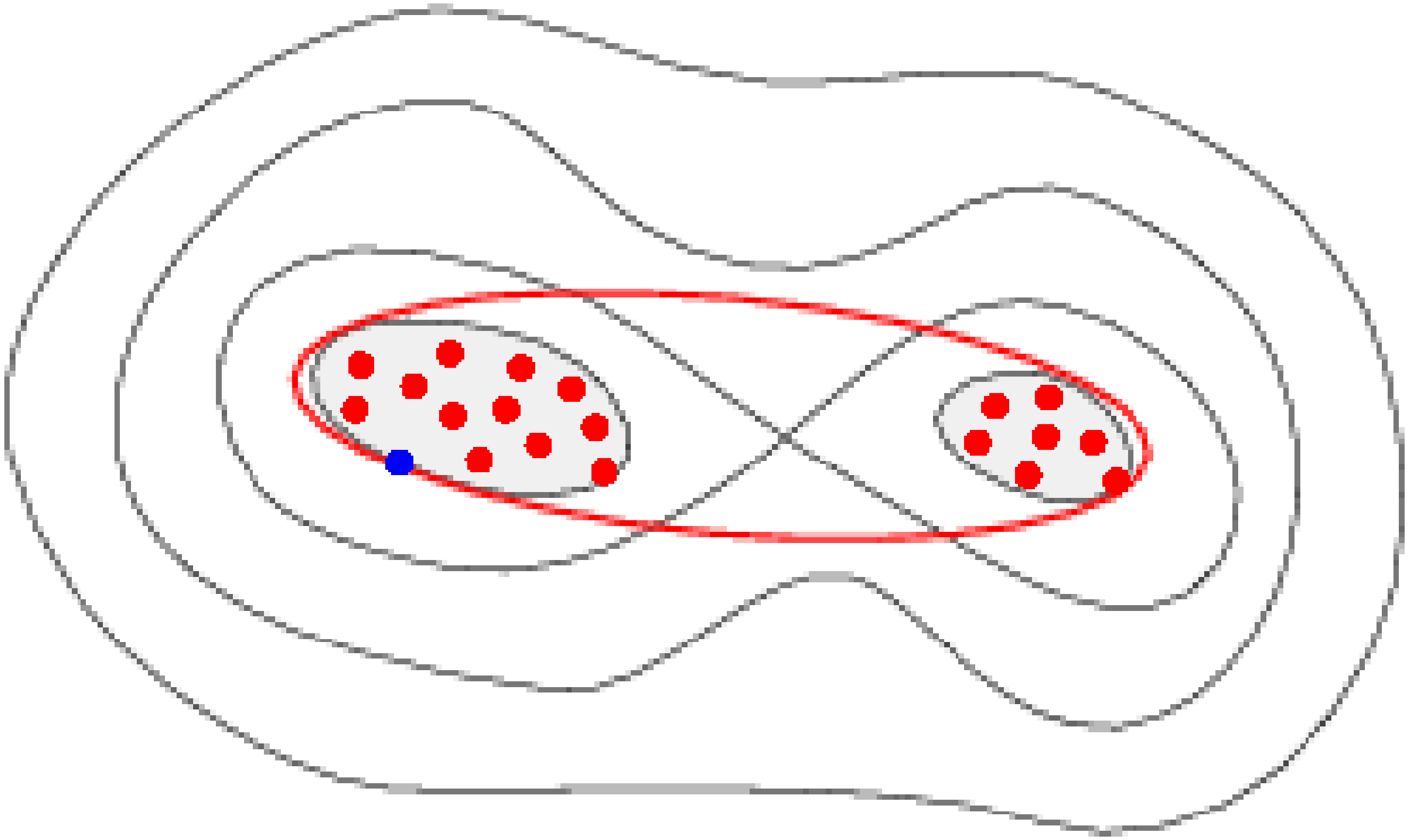
Problem with elliptical region sampling ($N = 20$):



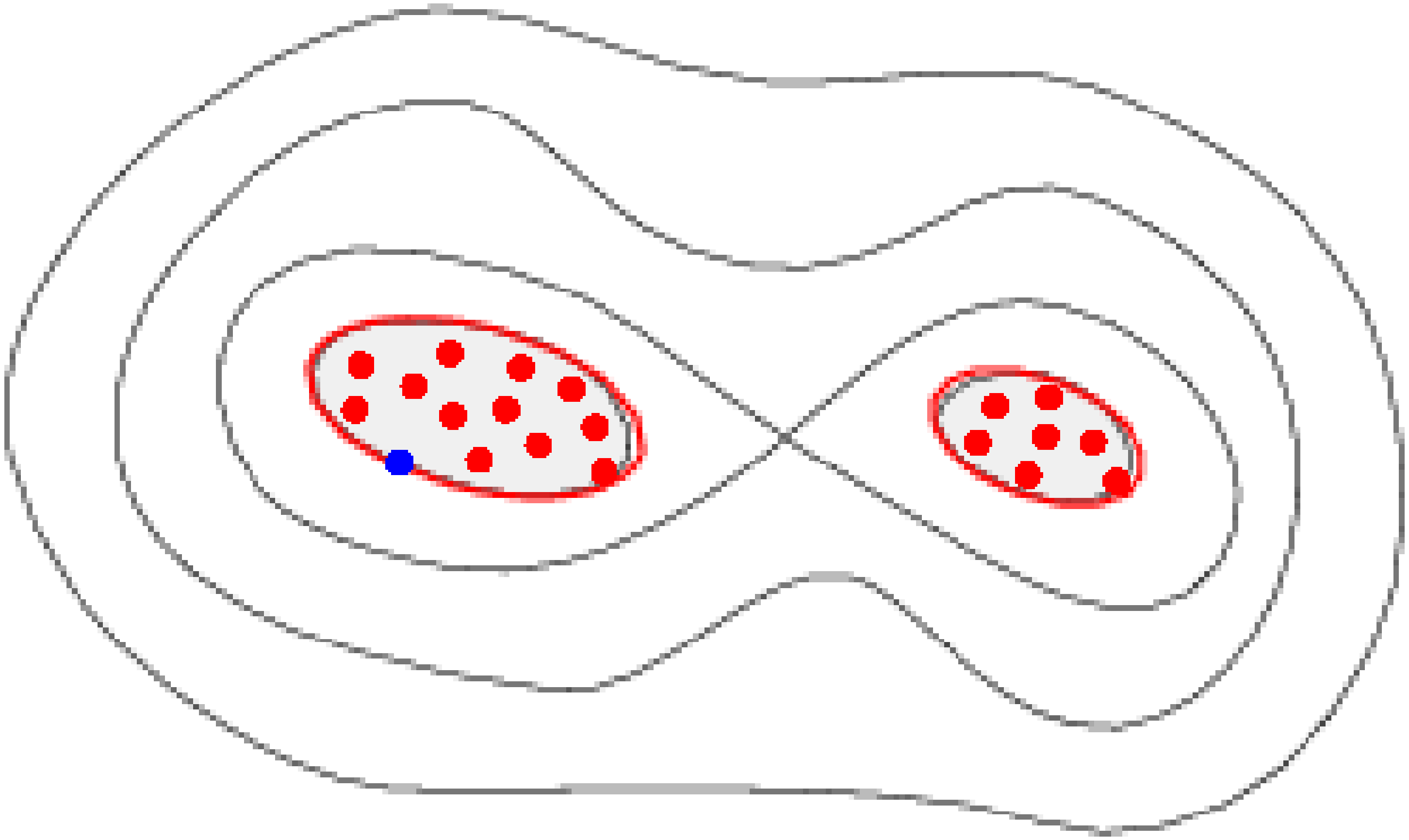
Problem with elliptical region sampling ($N = 20$):



Problem with elliptical region sampling ($N = 20$):



Problem with elliptical region sampling ($N = 20$):



MULTIMODAL NESTED SAMPLING – MULTINEST

- Introduced by Feroz & MPH (2008), refined by Feroz, MPH & Bridges (2008)

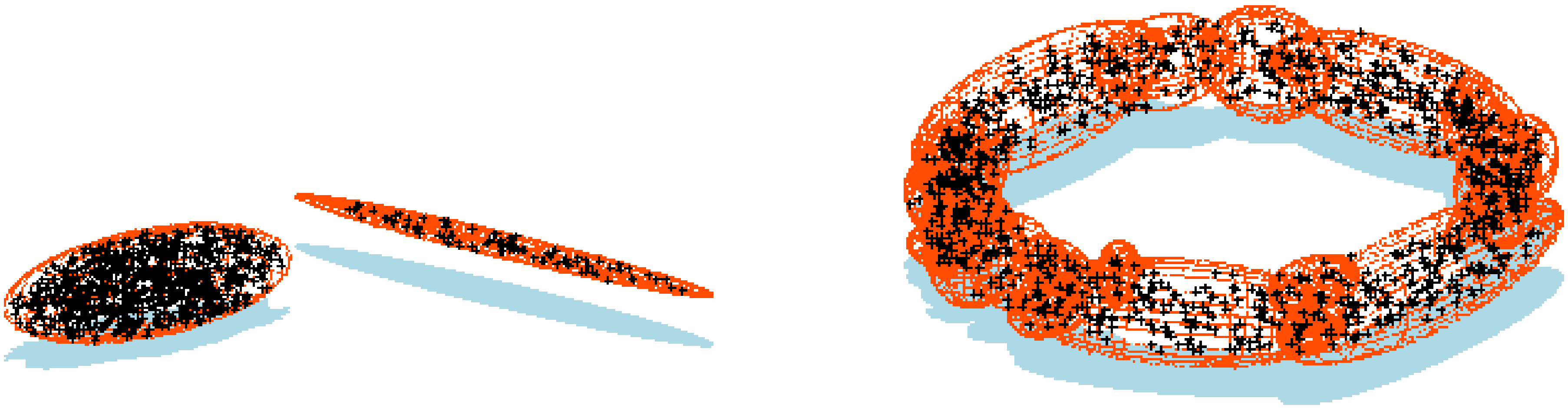
- At each nested sampling iteration i:

– construct optimal multi-ellipsoidal bound for each cluster (variable ellipsoid number), or evolve existing decomposition via scaling (fast)

– determine ellipsoid overlaps using cheap exact algorithm (Alfano et al. 2003)

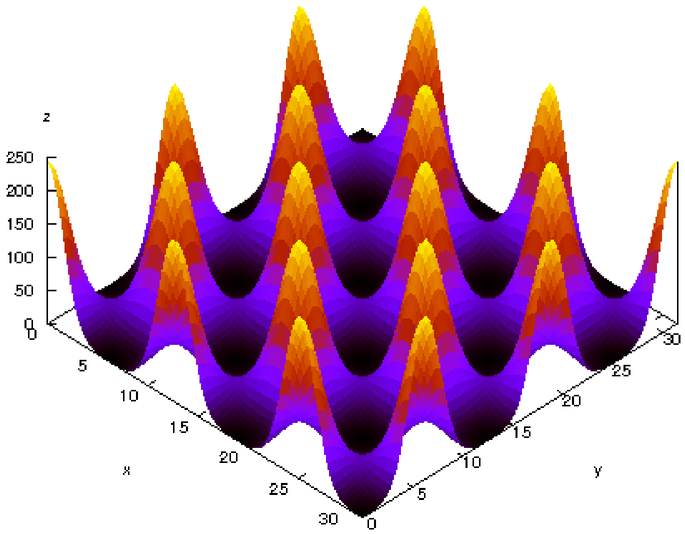
– remove point with lowest L_i from active points; increment evidence

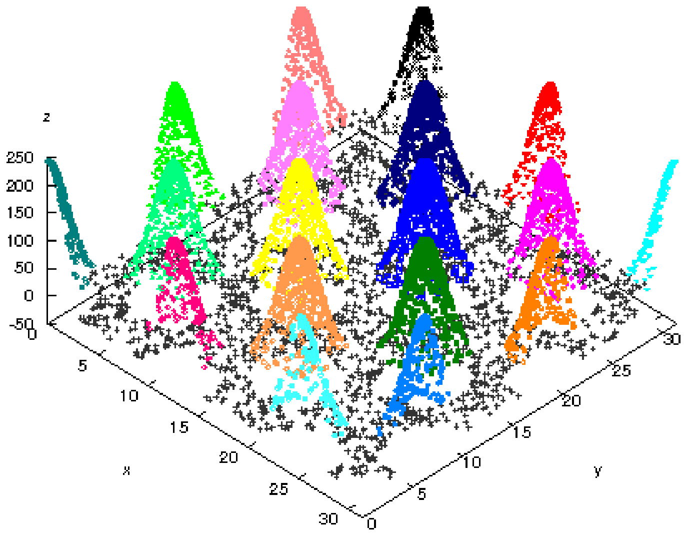
– pick ellipsoid randomly and sample new point with $L > L_i$, accounting for overlaps



- MULTINEST algorithm usefully (and easily) parallelized

IDENTIFICATION OF POSTERIOR MODES



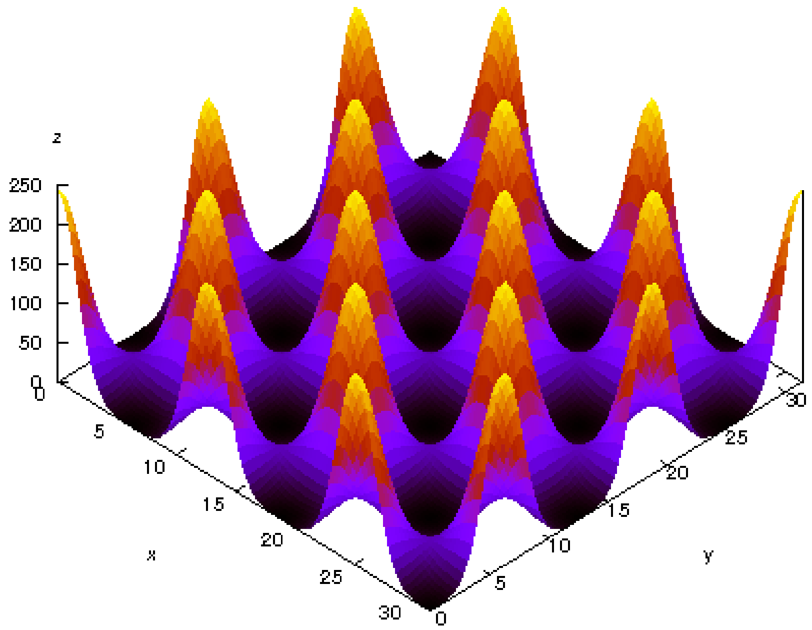


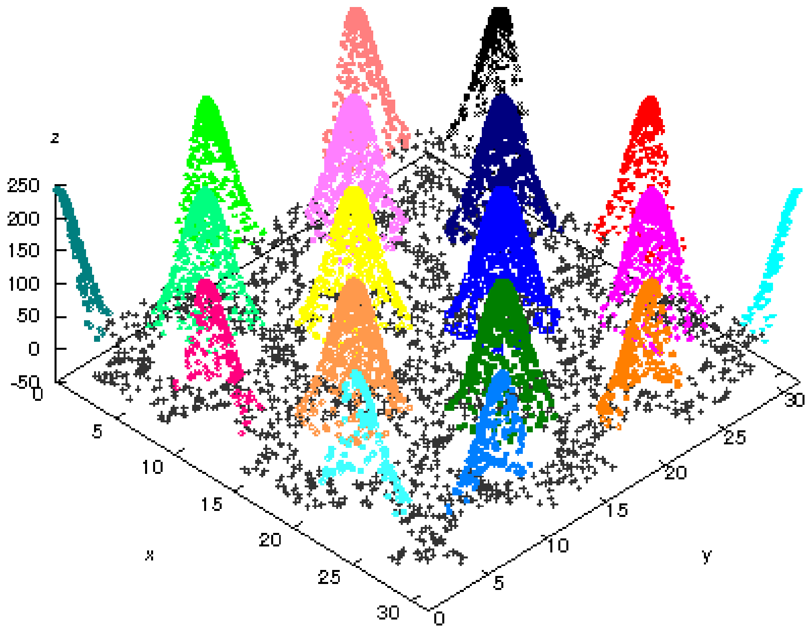
- For multimodal posteriors, useful to identify which samples ‘belong’ to which mode

- For well-defined ‘isolated’ modes:
  – can make reasonable estimate of posterior mass each contains (‘local’ evidence)
  – can construct posterior parameter constraints associated with each mode

- Partitioning and ellipsoids construction algorithm described above provides efficient and reliable method for performing mode identification
  ⇒ ‘local’ evidence and parameter constraints for each isolated mode
  ⇒ sum of local evidences equals ‘global’ evidence

Toy problem: egg-box likelihood





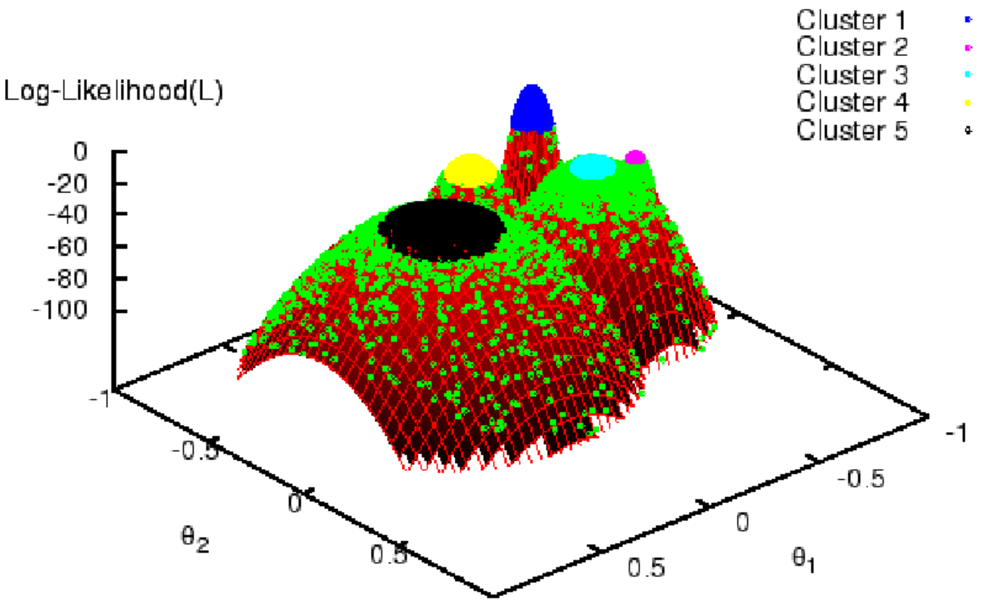
- Likelihood resembles egg-box and is given by

$$\mathcal{L}(\theta_1, \theta_2) = \exp\left[2 + \cos\left(\frac{\theta_1}{2}\right)\cos\left(\frac{\theta_2}{2}\right)\right]^5,$$

  and prior is $\mathcal{U}(0, 10\pi)$ for both θ_1 and θ_2.

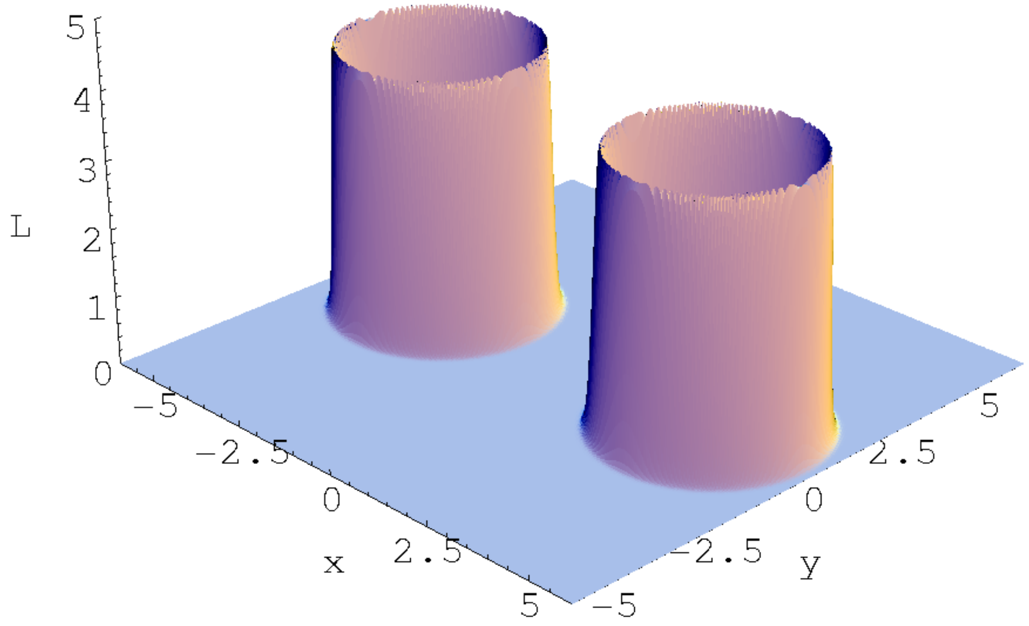
- Use 2000 active points $\Rightarrow \sim 30,000$ likelihood evaluations to obtain $\log \mathcal{Z} = 235.86 \pm 0.06$ (analytical $\log \mathcal{Z} = 235.88$)

TOY PROBLEM: MULTIPLE GAUSSIAN LIKELIHOOD



- Likelihood = five 2-D Gaussians of varying widths and amplitudes; prior = uniform
- Analytic evidence integral $\log E = -5.27$
- MULTINEST: $\log E = -5.33 \pm 0.11$, $N_{\text{like}} \approx 10^4$
- Thermodynamic integration (+ error): $\log E = -5.24 \pm 0.12$, $N_{\text{like}} \approx 4 \times 10^6$
- Typical of real applications (see later): $\sim 500\times$ efficiency of standard MCMC

TOY PROBLEM: MULTIPLE GAUSSIAN SHELLS



- Likelihood defined as

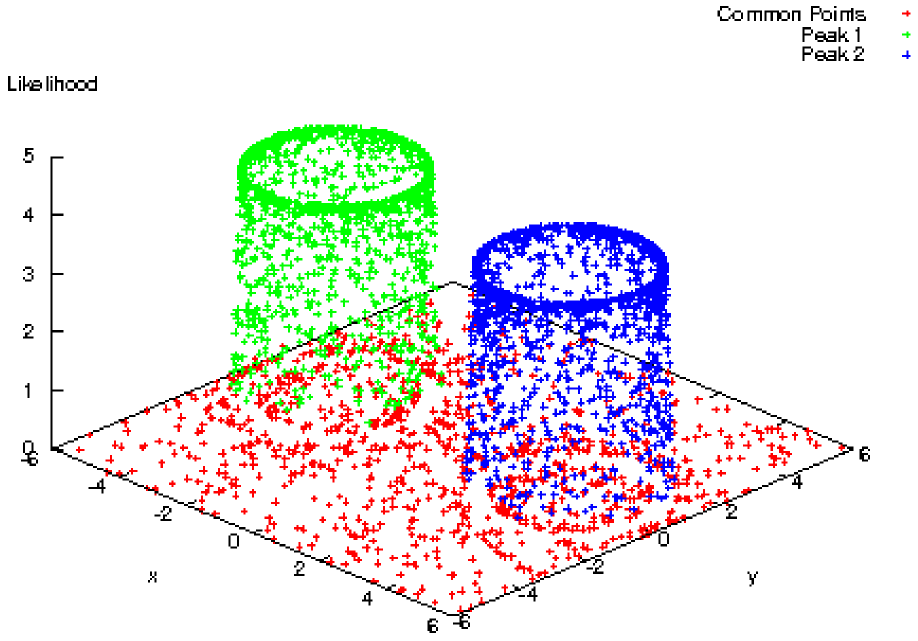
$$L(\boldsymbol{x}) = \mathrm{circ}(\boldsymbol{x}; \boldsymbol{c}_1, r_1, w_1) + \mathrm{circ}(\boldsymbol{x}; \boldsymbol{c}_2, r_2, w_2),$$

where

$$\mathrm{circ}(\boldsymbol{x}; \boldsymbol{c}, r, w) = \frac{1}{\sqrt{2\pi w^2}} \exp\left[-\frac{(|\boldsymbol{x} - \boldsymbol{c}| - r)^2}{2w^2}\right].$$

and assuming a uniform prior

- MultiNest results:

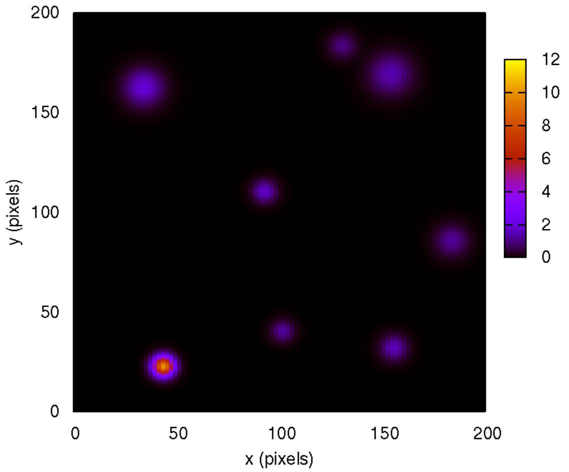


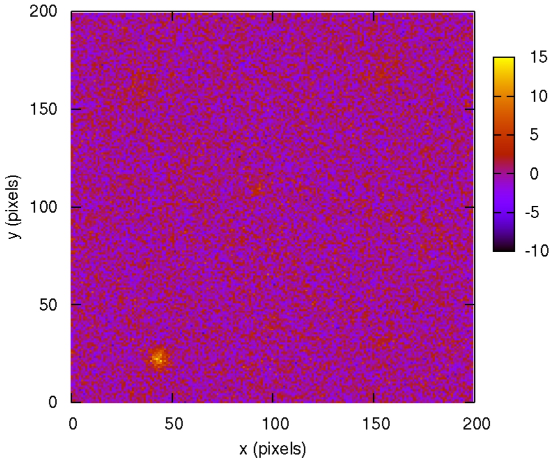
	MultiNest	
D	N_{like}	Efficiency
2	$7,370$	70.77%
5	$17,967$	51.02%
10	$52,901$	34.28%
20	$255,092$	15.49%
30	$753,789$	8.39%

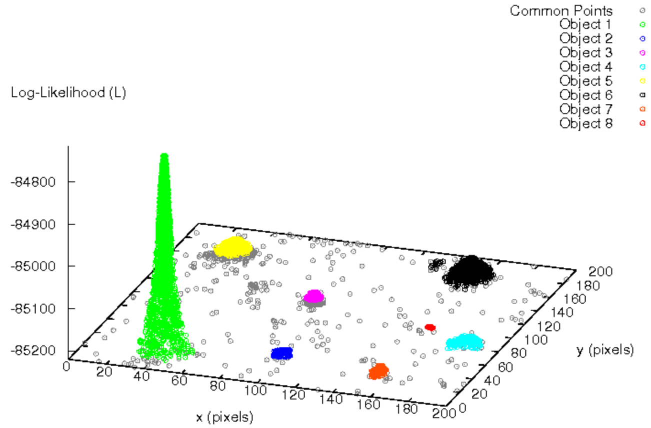
	Analytical		MultiNest		
D	$\log(\mathcal{Z})$	local $\log(\mathcal{Z})$	$\log(\mathcal{Z})$	local $\log(\mathcal{Z}_1)$	local $\log(\mathcal{Z}_2)$
2	-1.75	-2.44	-1.72 ± 0.05	-2.28 ± 0.08	-2.56 ± 0.08
5	-5.67	-6.36	-5.75 ± 0.08	-6.34 ± 0.10	-6.57 ± 0.11
10	-14.59	-15.28	-14.69 ± 0.12	-15.41 ± 0.15	-15.36 ± 0.15
20	-36.09	-36.78	-35.93 ± 0.19	-37.13 ± 0.23	-36.28 ± 0.22
30	-60.13	-60.82	-59.94 ± 0.24	-60.70 ± 0.30	-60.57 ± 0.32

- Bank sampler (MCMC): $N_{\text{like}} \sim 10^6$ in $D = 2$ for parameter estimation alone

Applications of MultiNest: toy model







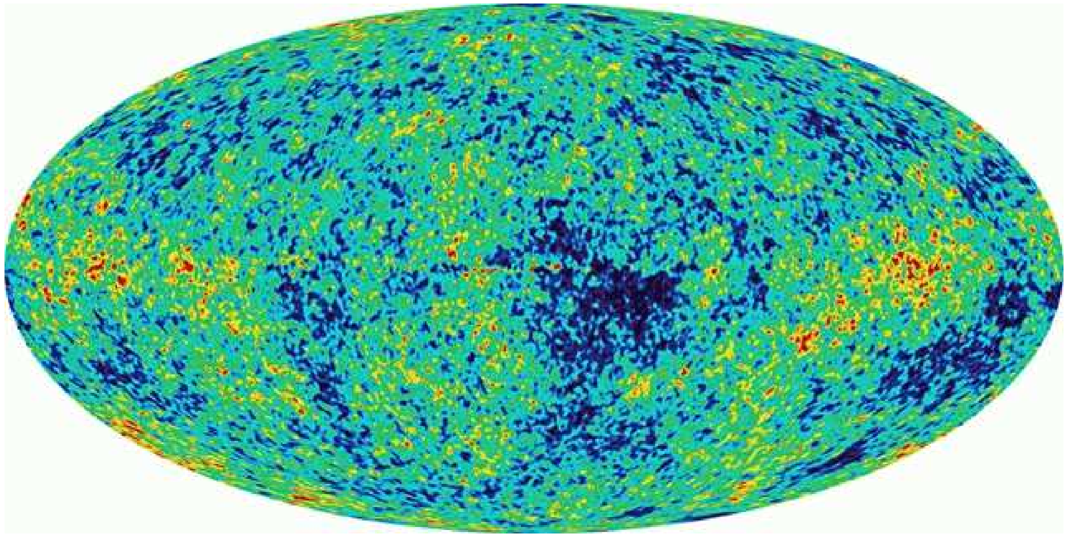
- Toy model: Gaussian objects in noise (Feroz & MPH, arXiv:0704.3704)

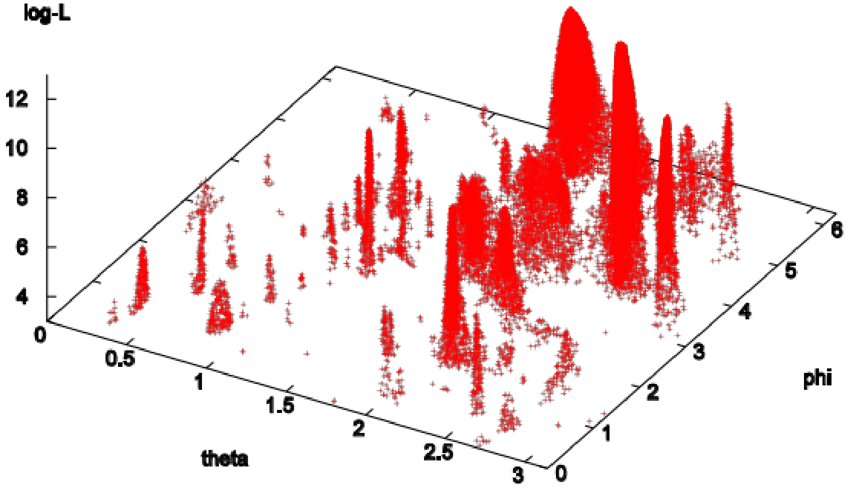
- Multinest: $N_{\rm like} \sim 10^4$, run time ~ 2 CPU mins – identified all objects correctly

- BayeSys (MCMC + thermo. int.): $N_{\rm like} \sim 5 \times 10^6$, run time ~ 16 CPU hrs
  Required several object subtraction iterations to identify all objects

APPLICATIONS OF MULTINEST: TEXTURES IN CMB

- Textures in CMB data (in preparation)

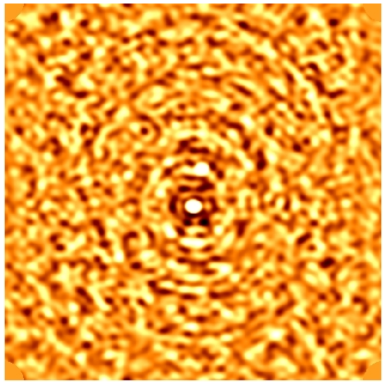


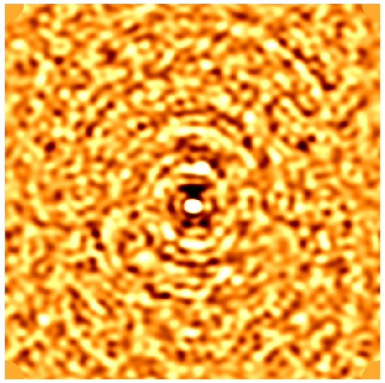


APPLICATIONS OF MULTINEST: CLUSTERS IN SZ

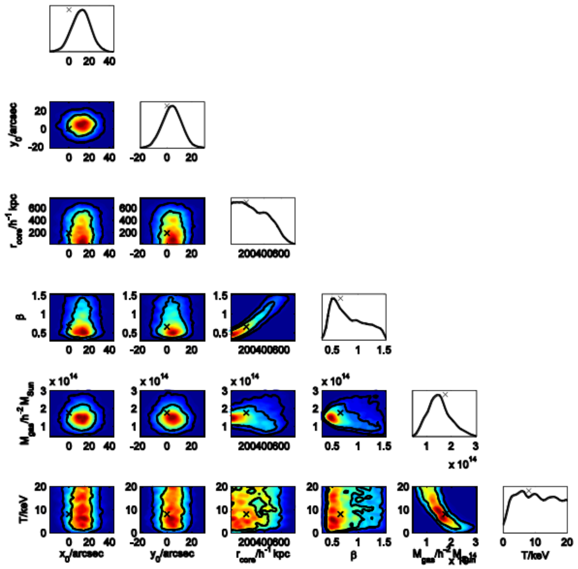
- Cluster (and point sources) in interferometric SZ data (Feroz et al., arXiv:0811.1199)

- Simulations: A (left) without cluster and B (right) with cluster (β-model), including CMB, 3 point sources, confusion noise, instrumental noise



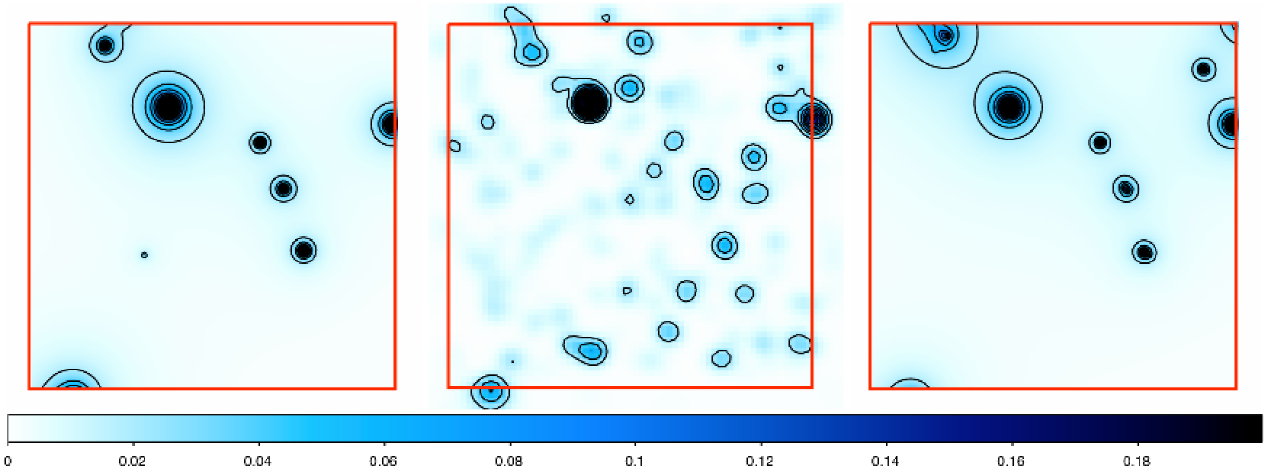


- A simulation $R = 0.35 \pm 0.05$; B simulation $R \sim 10^{33}$. Parameter constraints:

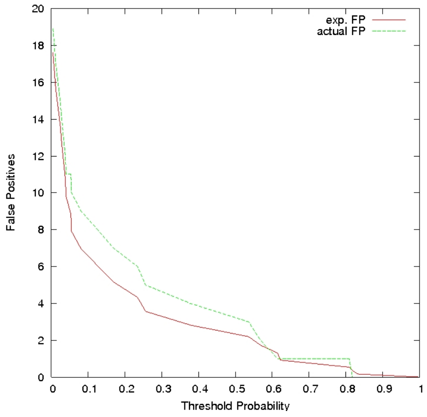


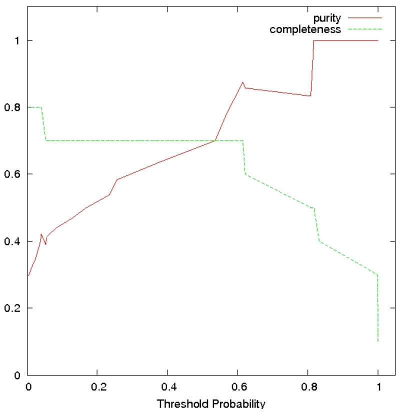
Applications of MultiNest: clusters in lensing

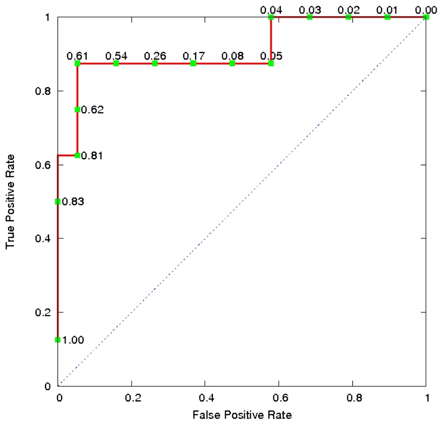
- Clusters in weak lensing surveys (Feroz, Marshall, MPH, arXiv:0810.0781)
- 0.5×0.5 deg^2 simulation (ΛCDM + Press–Schechter), 100 gal arcmin^{-2}, $\sigma = 0.3$



- Probability ith mode is true positive $p_i = R_i/(1+R_i) \Rightarrow \hat{n}_{FP} = \sum_{\substack{i=1 \\ p_i>p_{\rm th}}}^{N} (1-p_i)$

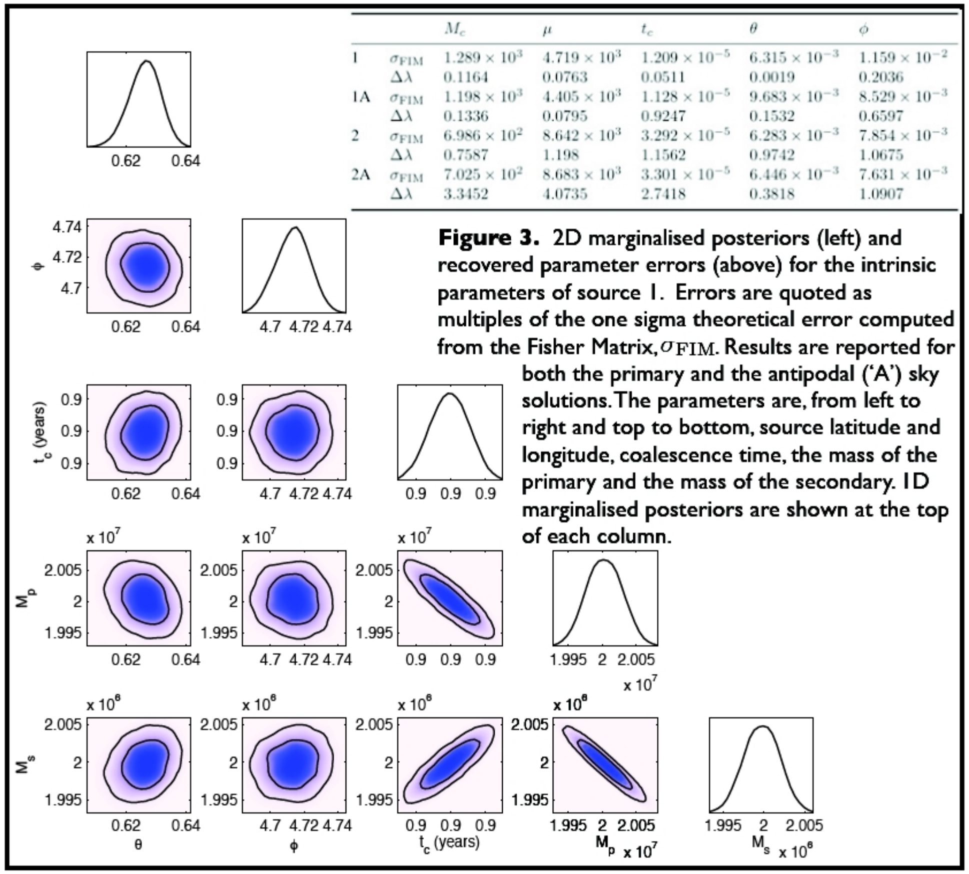






Applications of MultiNest: gravitational waves

- Simulated LISA data containing two signals from non-spinning SMBH mergers. Each source has antipodal degeneracy ⇒ at least 4 modes in posterior
- All identified and well characterized in ~ 2 CPU hrs (Feroz et al., arXiv:0904.1544)



		M_c	μ	t_c	θ	ϕ
1	$\sigma_{\rm FIM}$	1.289×10^3	4.719×10^3	1.209×10^{-5}	6.315×10^{-3}	1.159×10^{-2}
	$\Delta\lambda$	0.1164	0.0763	0.0511	0.0019	0.2036
1A	$\sigma_{\rm FIM}$	1.198×10^3	4.405×10^3	1.128×10^{-5}	9.683×10^{-3}	8.529×10^{-3}
	$\Delta\lambda$	0.1336	0.0795	0.9247	0.1532	0.6597
2	$\sigma_{\rm FIM}$	6.986×10^2	8.642×10^3	3.292×10^{-5}	6.283×10^{-3}	7.854×10^{-3}
	$\Delta\lambda$	0.7587	1.198	1.1562	0.9742	1.0675
2A	$\sigma_{\rm FIM}$	7.025×10^2	8.683×10^3	3.301×10^{-5}	6.446×10^{-3}	7.631×10^{-3}
	$\Delta\lambda$	3.3452	4.0735	2.7418	0.3818	1.0907

Figure 3. 2D marginalised posteriors (left) and recovered parameter errors (above) for the intrinsic parameters of source 1. Errors are quoted as multiples of the one sigma theoretical error computed from the Fisher Matrix, $\sigma_{\rm FIM}$. Results are reported for both the primary and the antipodal ('A') sky solutions. The parameters are, from left to right and top to bottom, source latitude and longitude, coalescence time, the mass of the primary and the mass of the secondary. 1D marginalised posteriors are shown at the top of each column.

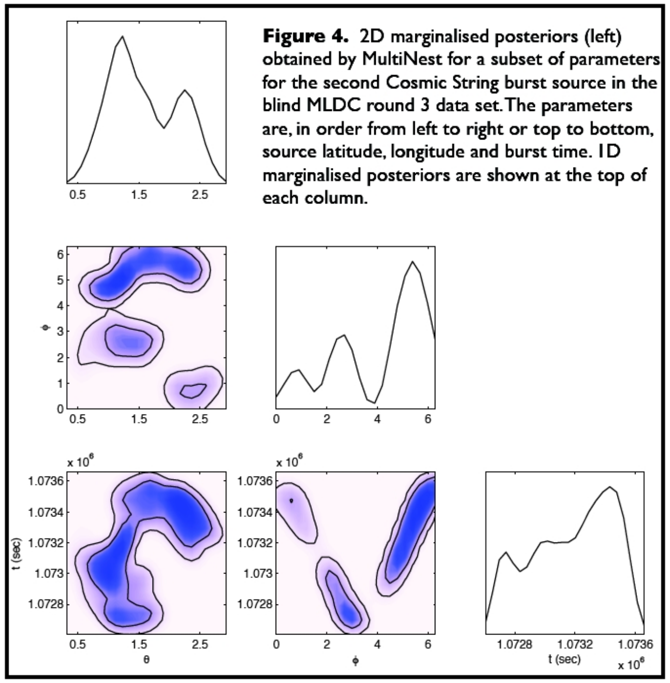
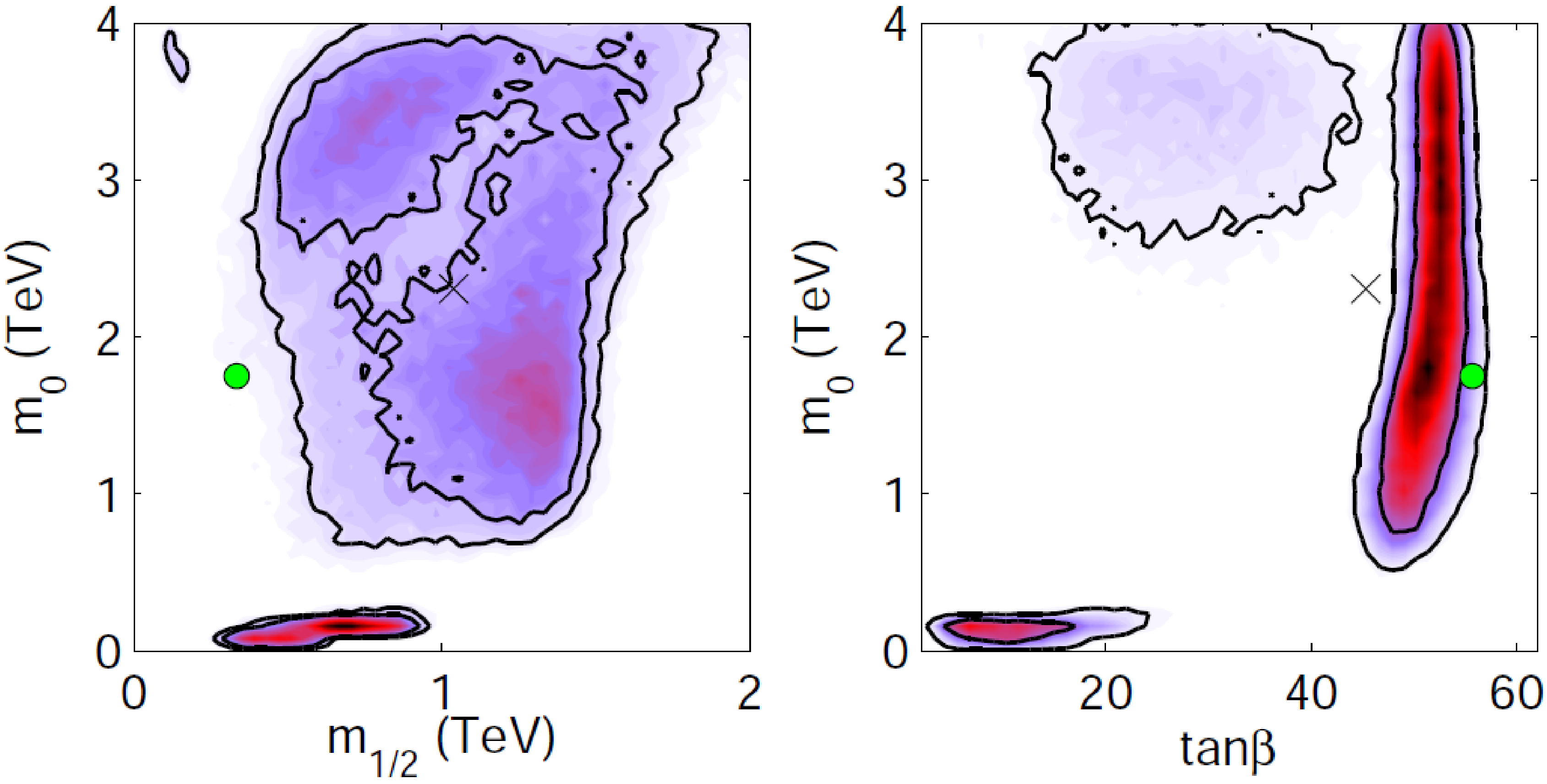


Figure 4. 2D marginalised posteriors (left) obtained by MultiNest for a subset of parameters for the second Cosmic String burst source in the blind MLDC round 3 data set. The parameters are, in order from left to right or top to bottom, source latitude, longitude and burst time. 1D marginalised posteriors are shown at the top of each column.

- Also applied successfully in Mock LISA Data Challenge Round 3 to simulations of 5 spinning BH binary inspirals and 3 cosmic strings (Feroz et al. arXiv:0911.0288)

BEYOND ASTRONOMY: MULTINEST IN PARTICLE PHYSICS

- SUSY phenomenology: MultiNest applied to cMSSM and pMSSM by us (see arXiv:0807.4512, arXiv:0809.3792, arXiv:0903.2487, arXiv0904.2548, arXiv0906.0957, arXiv:1101.3296) + and by others



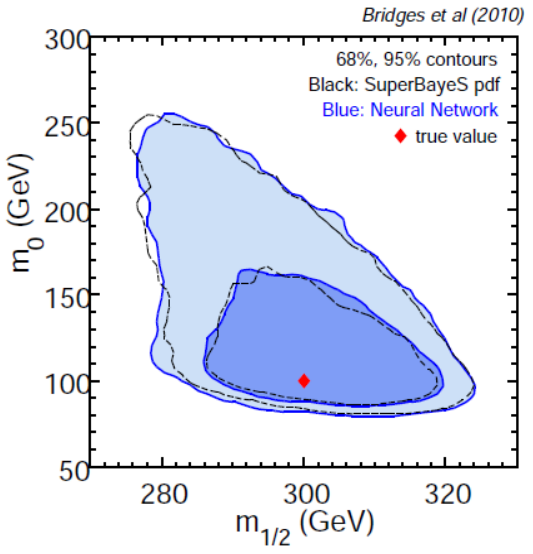
- In all cases, MULTINEST is few × 100 more efficient than MCMC

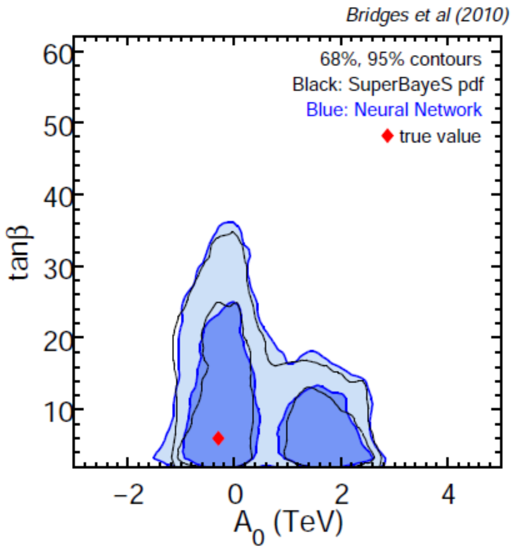
Beyond astronomy: NN and MultiNest in particle physics

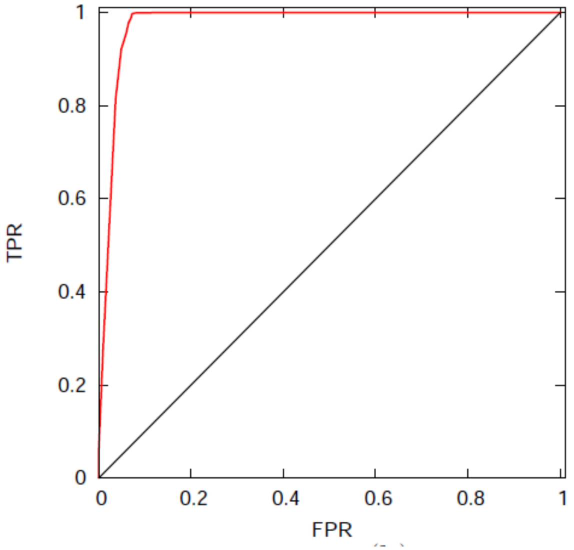
- Recently applied NN to Constrained MSSM (Bridges et al. – arXiv:1011.4306)

- SoftSusy: theory parameters $\boldsymbol{\theta} \rightarrow$ sparticle mass spectrum $\boldsymbol{m}$ by computationally expensive evolution of renormalisation group equations $\Rightarrow$ replace with NN

- Also built classification NN to partition $\boldsymbol{\theta}$-space into physical and unphysical regions







- Speeds up analysis by factor $\sim 10^4$ (MultiNest provides further factor of ~ 100) $\Rightarrow$ original SoftSusy + MCMC $=$ 720 CPU days; NN + MultiNest $=$ 1 minute

4: The future: BAMBI. . .

BLIND ACCELERATED MUTLIMODAL BAYESIAN INFERENCE (BAMBI)

- General Bayesian inference engine with wide applicability: only requires choice of priors on the parameters in model

- Combines neural networks and nested sampling in complementary manner

- Basic idea is as follows:
  - early stage (prior-driven) nested samples ⇒ (incremental) training data set
  - simultaneous training of neural network ⇒ ‘learn’ likelihood function
  - clustering in nested sampler ⇒ accelerates network training
  - once trained, network replaces likelihood code
    ⇒ completes posterior sampling and evidence evaluation extremely rapidly
  - trained likelihood network available for subsequent analyses

CONCLUSIONS

- Standard Bayesian analysis can be very computationally intensive: days–weeks on a supercomputer

- Large speed-ups possible using neural networks for model prediction

- Efficient and robust evidence evaluation and parameter estimation provided by nested sampling
  – MULTINEST allows sampling from multimodal/degenerate posteriors
  – local and global evidences and parameter constraints
  – typically few × 100 times more efficient than standard MCMC

- These methods should be useful in a wide range of physical inference problems; already applied in many areas

- COSMONET and MULTINEST code publically available from:
  `www.mrao.cam.ac.uk/software/cosmonet`
  `www.mrao.cam.ac.uk/software/multinest`

- BAMBI in development. . .

Supplementary slides

ADVANTAGES OF COSMONET

- Simplicity: provides single, simple, closed-form function for each interpolation over entire parameter space

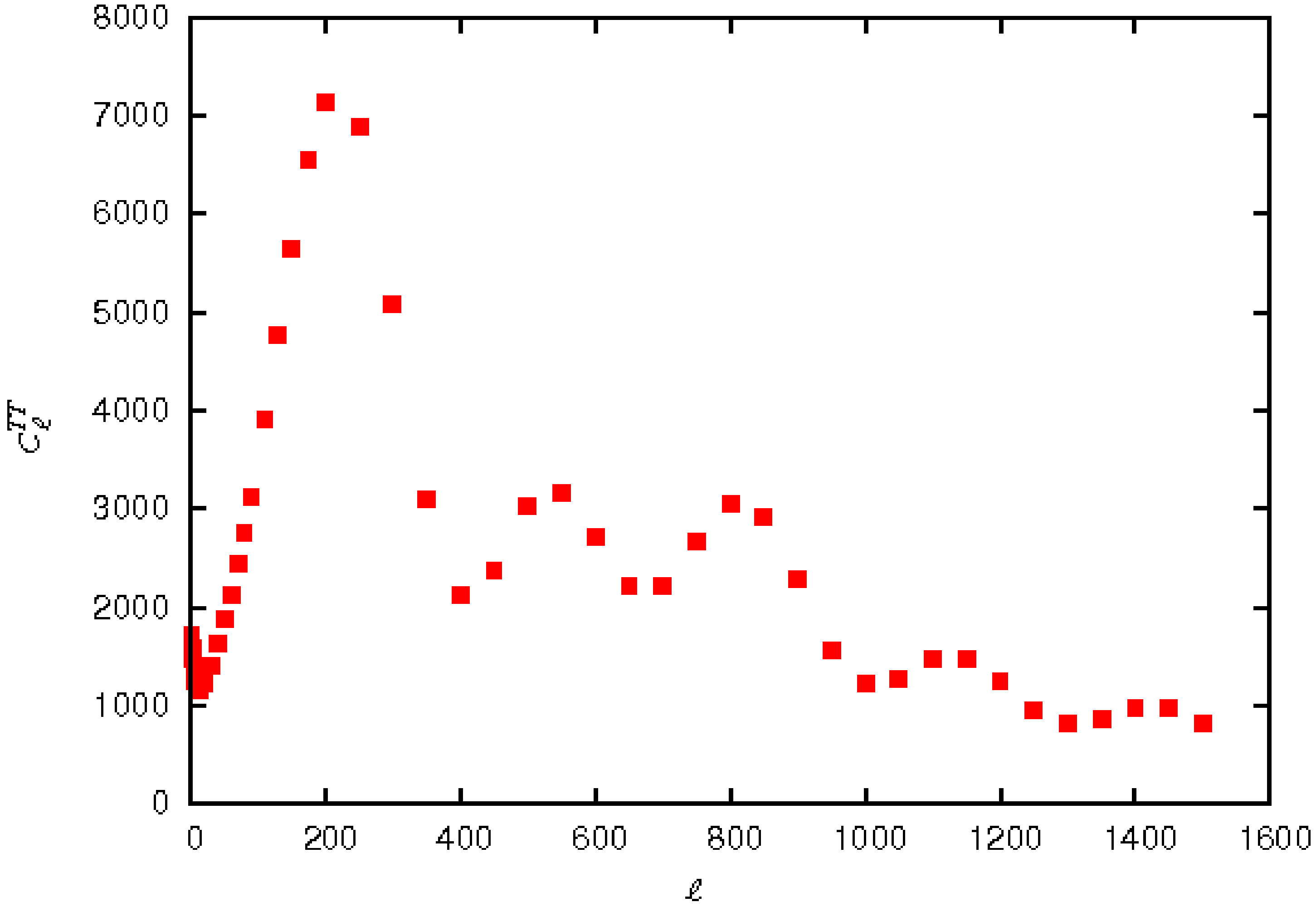
- Memory usage: a network with N_i input nodes, N_h hidden nodes and N_o output nodes has $(N_\mathrm{i}+1)N_\mathrm{h}+(N_\mathrm{h}+1)N_\mathrm{o} \approx N_\mathrm{h}N_\mathrm{o}$ parameters. For above model, requires only ~ 50 kB of parameter memory

- Accuracy: excellent after only $\sim$ few mins of training on single 2GHz CPU

- Speed: number of calculations to perform feed-forward network mapping is $2N_\mathrm{i}\,N_\mathrm{h} + 2N_\mathrm{h}N_\mathrm{o} \approx 2N_\mathrm{h}N_\mathrm{o}$. In above example, calculation of C_ℓ spectrum in $\sim$ 20 microseconds, and WMAP likelihood in $\sim$ 5 microseconds

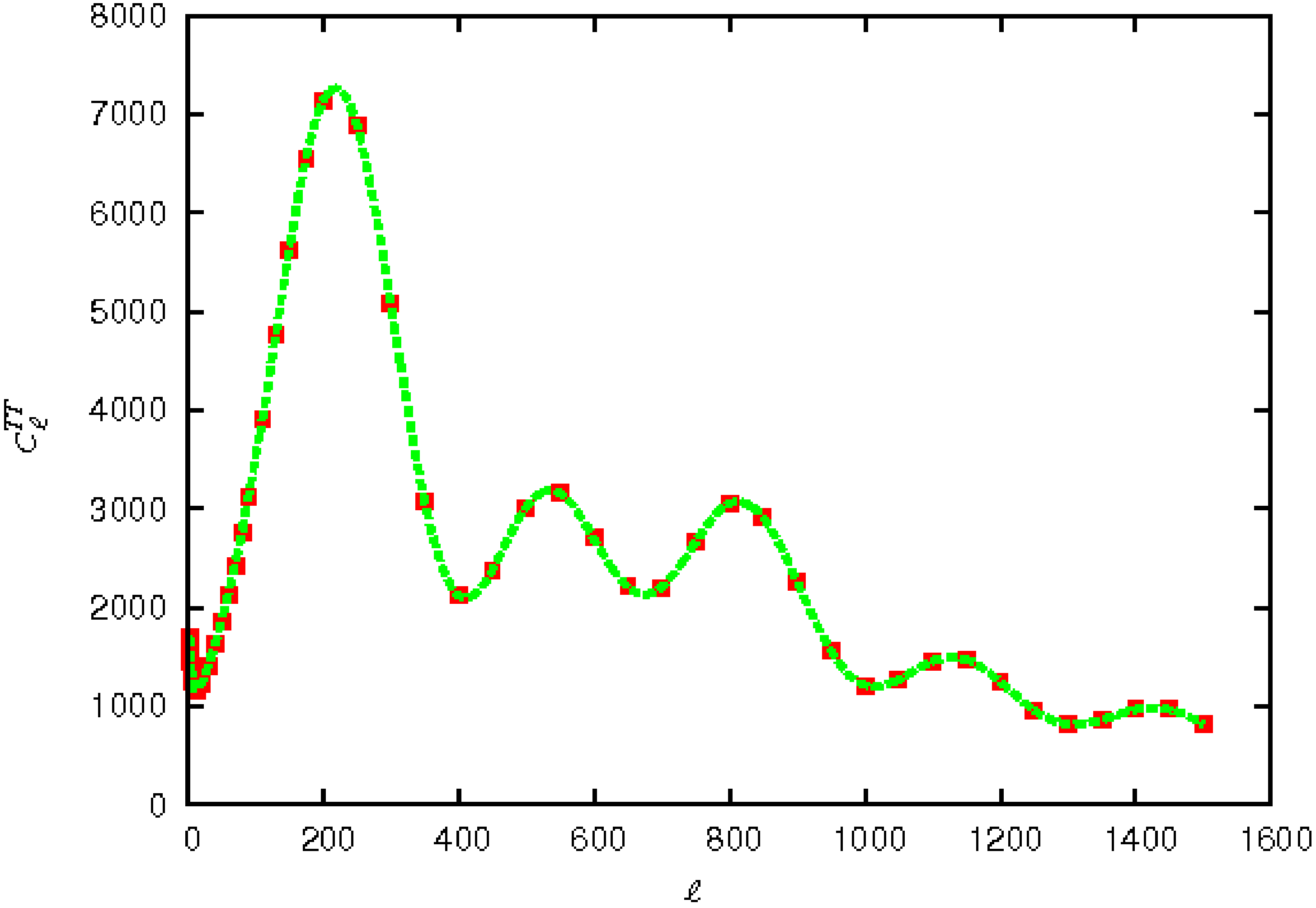
- Scaling: N_h increases at worst linearly with N_i

TRAINING DATA: C_ℓ SPECTRA



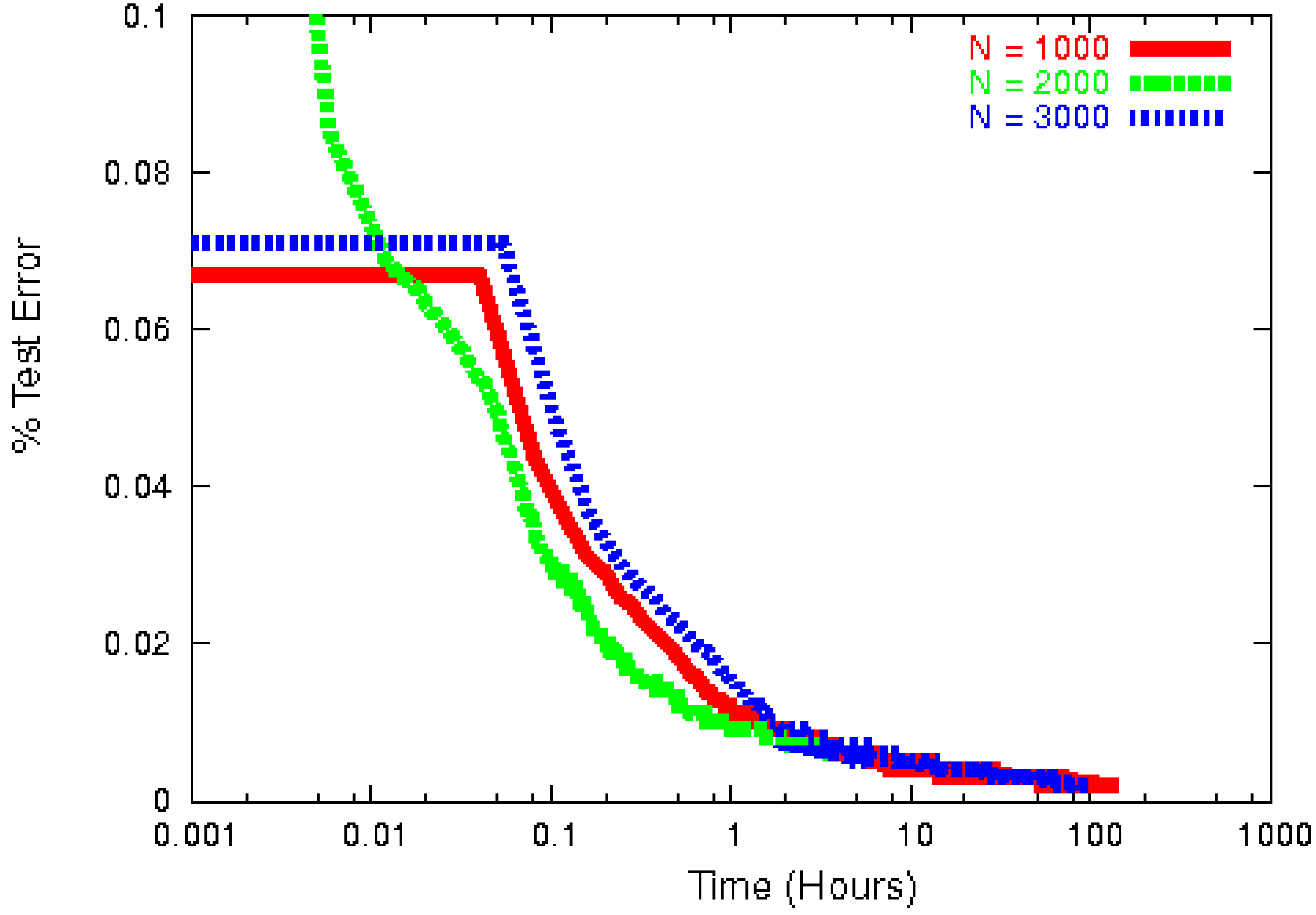
- CAMB generates C_ℓ spectra at a specified set (~ 50) of ℓ-values

TRAINING DATA: C_ℓ SPECTRA



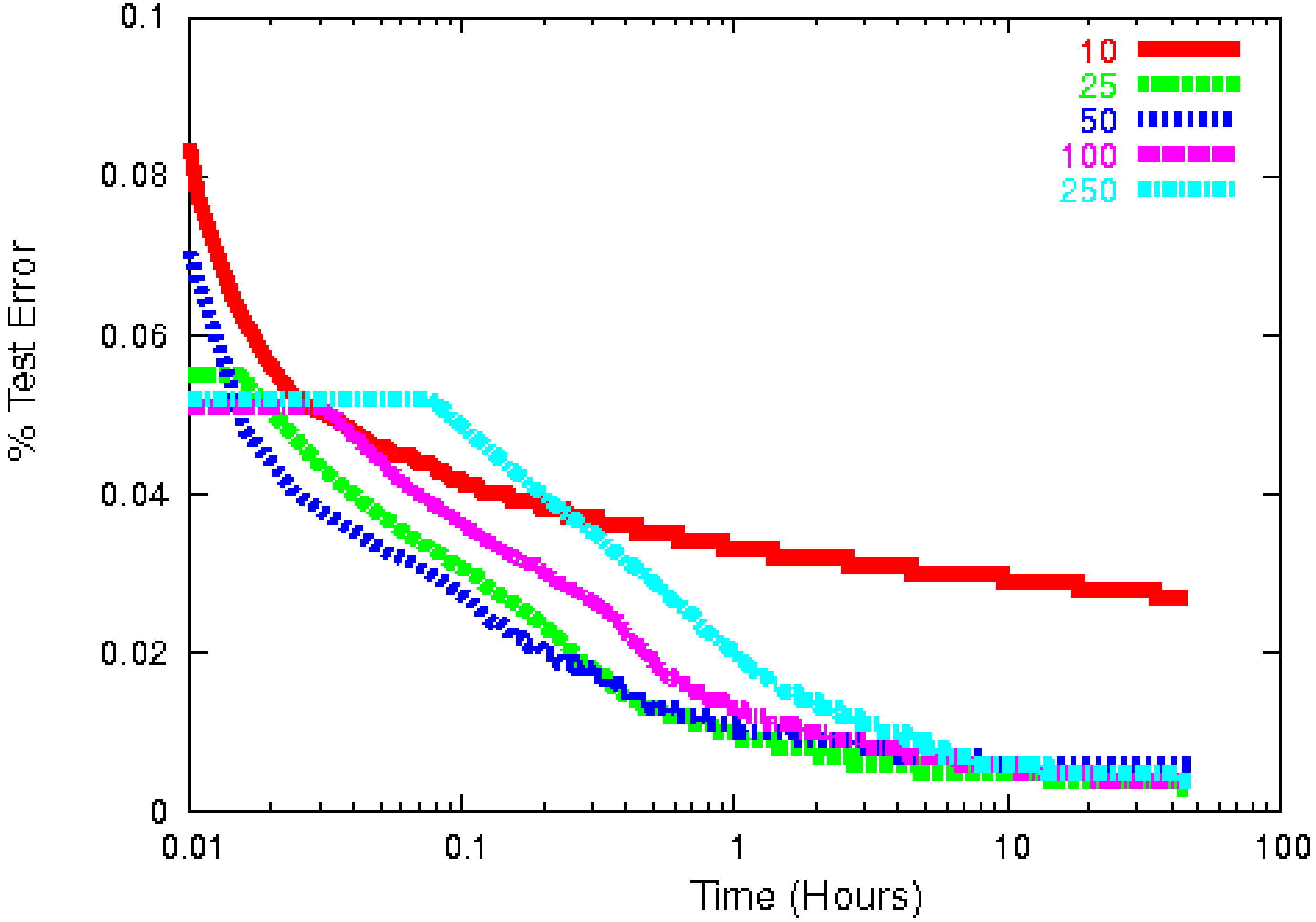
- Cubic spline interpolation used to create full set of C_ℓ values

QUANTITY OF TRAINING DATA



- Use few 1000 training data: more data simply slow training
- But can obtain usable results using few 100

Network complexity



- For cosmological application found optimum number of hidden nodes ~ 50
- Spectra with more structure would simply require more nodes
- Can find optimal number of hidden nodes by maximising evidence

UNIT HYPERCUBE SAMPLING SPACE

- Algorithm for partitioning active points into clusters and constructing ellipsoidal bounds requires uniformly distributed points

- MULTINEST 'native' space = D-dimensional unit hypercube in which samples are drawn uniformly. All operations are carried out in this space (cf. BAYESYS).

- To conserve probability mass, point $\boldsymbol{u} = (u_1, u_2, \cdots, u_D)$ in unit hypercube transformed point $\boldsymbol{\Theta} = (\theta_1, \theta_2, \cdots, \theta_D)$ in 'physical' parameter space, such that

$$\int \pi(\theta_1, \theta_2, \cdots, \theta_D)\, d\theta_1\, d\theta_2 \cdots d\theta_D = \int du_1 du_2 \cdots du_D$$

- In simple case that prior separable: $\pi(\boldsymbol{\Theta}) = \pi_1(\theta_1)\pi_2(\theta_2)\cdots\pi_D(\theta_D)$, set $\pi_j(\theta_j)d\theta_j = du_j \Rightarrow$ for given u_j, find θ_j by solving

$$u_j = \int_{-\infty}^{\theta_j} \pi_j(\theta'_j)d\theta'_j$$

- If prior $\pi(\mathbf{\Theta})$ not separable, instead write

$$\pi(\theta_1, \theta_2, \cdots, \theta_D) = \pi_1(\theta_1)\pi_2(\theta_2|\theta_1) \cdots \pi_D(\theta_D|\theta_1, \theta_2 \cdots \theta_{D-1})$$

  where

$$\pi_j(\theta_j|\theta_1, \cdots, \theta_{j-1}) = \int \pi(\theta_1, \cdots, \theta_{j-1}, \theta_j, \theta_{j+1}, \cdots, \theta_D)\, d\theta_{j+1} \cdots d\theta_D$$

- Physical point $\mathbf{\Theta}$ corresponding to point $\boldsymbol{u}$ in unit hypercube then found by using this π_j in earlier expression

- Physical parameters $\mathbf{\Theta}$ used to calculate likelihood of point $\boldsymbol{u}$
  For many problems, prior $\pi(\mathbf{\Theta})$ is uniform $\Rightarrow$ $\boldsymbol{u}$ and $\mathbf{\Theta}$-spaces coincide
  For many other problems, prior $\pi(\mathbf{\Theta})$ allows one to solve for $\mathbf{\Theta}$ point analytically

- In all cases, can solve for $\mathbf{\Theta}$ point numerically

- Alternatively... re-cast inference problem: for example, define new 'likelihood' $\mathcal{L}'(\mathbf{\Theta}) \equiv \mathcal{L}(\mathbf{\Theta})\pi(\mathbf{\Theta})$ and 'prior' $\pi'(\mathbf{\Theta}) \equiv$ constant. But potentially inefficient since lacks true prior $\pi(\mathbf{\Theta})$ to guide the sampling of active points

PARTITIONING OF POINTS AND CONSTRUCTION OF ELLIPSOIDAL BOUNDS

- At ith NS iteration, find 'optimal' ellipsoidal decomposition of N active points distributed uniformly in remaining prior volume X_i using EM approach

- Let set of N active points in unit hypercube be $S = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_N\}$ and some partitioning into K clusters be $\{S_k\}_{k=1}^{K}$, where $K \geq 1$ and $\cup_{k=1}^{K} S_k = S$.

- For cluster (or subset) S_k containing n_k points, define quasi-minimum-volume bounding ellipsoid

$$E_k = \{\boldsymbol{u} \in \mathcal{R}^{\mathsf{D}} | \boldsymbol{u}^{\mathsf{T}} (f_k \mathbf{C}_k)^{-1} \boldsymbol{u} \leq 1\},$$

where the empirical covariance matrix of the subset is

$$\mathbf{C}_k = \frac{1}{n_k} \sum_{j=1}^{n_k} (\boldsymbol{u}_j - \boldsymbol{mu}_k)(\boldsymbol{u}_j - \boldsymbol{mu}_k)^{\mathsf{T}}$$

and $\boldsymbol{mu}_k = \sum_{j=1}^{n_k} \boldsymbol{u}_j$ is its center of the mass. Enlargement factor f_k ensures E_k is a bounding ellipsoid. Note: volume of ellipsoid $V(E_k) \propto \sqrt{\mathrm{det}(f_k \mathbf{C}_k)}$

- At ith NS iteration, volume $V(S)$ from which set S uniformly sampled is unknown remaining prior volume X_i, but use expectation value $V(S) = \exp(-i/N)$

- Define objective function

$$F(S) \equiv \frac{1}{V(S)} \sum_{k=1}^{K} V(E_k)$$

and minimise $F(S)$, subject to the constraint $F(S) \geq 1$, wrt K-partitionings $\{S_k\}_{k=1}^{K}$ ⇒ 'optimal' decomposition of original sampled region into K ellipsoids

- Minimisation most easily performed using EM scheme, using result (Lu et al. 2007) that, change in $F(S)$ resulting from reassigning a point $\boldsymbol{u}$ from subset S_k to $S_{k'}$ is

$$\Delta F(S)_{k,k'} \approx \gamma \left(\frac{V(E_{k'})d(\boldsymbol{u}, S_{k'})}{V(S_{k'})} - \frac{V(E_k)d(\boldsymbol{u}, S_k)}{V(S_k)} \right)$$

where γ is a constant,

$$d(\boldsymbol{u}, S_k) = (\boldsymbol{u} - \boldsymbol{mu}_k)^{\mathsf{T}} (f_k \mathbf{C}_k)^{-1} (\boldsymbol{u} - \boldsymbol{mu}_k)$$

is 'distance' from $\boldsymbol{u}$ to centroid $\boldsymbol{mu}_k$ of ellipsoid E_k, and

$$V(S_k) = \frac{n_k V(S)}{N}$$

may be considered the volume from which subset S_k was drawn uniformly

- In fact, impose further constraint that $V(E_k) > V(S_k)$. Easily achieved by enlarging ellipsoid E_k by factor f_k, such that $V(E_k) = \max[V(E_k), V(S_k)]$, before evaluating $F(S)$ and $\Delta F(S)_{k,k'}$

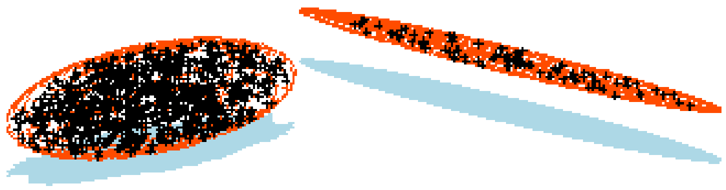
- Minimising $F(S)$ equivalent to defining

$$h_k(\boldsymbol{u}) = \frac{V(E_k) d(\boldsymbol{u}, S_k)}{V(S_k)}$$

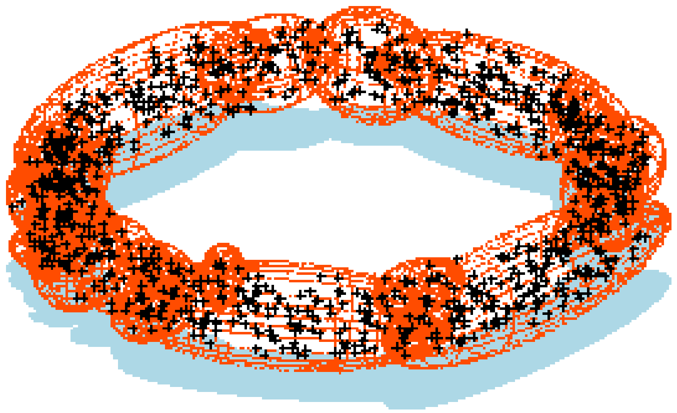
and, for all points $\boldsymbol{u} \in S$, assigning $\boldsymbol{u} \in S_k$ to $S_{k'}$ only if $h_k(\boldsymbol{u}) < h_{k'}(\boldsymbol{u})$, $\forall\, k \neq k'$, and repeating until convergence is achieved

- To find optimal number of ellipsoids, K, use recursive scheme:
– start by performing k-means partition with $K = 2$
– optimise this 2-partition as outlined above,
– recursively partition and optimise the resulting clusters

Ellipsoidal Decomposition Algorithm



1000 points drawn from two ellipsoids



1000 points drawn from a torus

1. For S, calculate bounding ellipsoid E and $V(E)$
2. Enlarge E so that $V(E) = \max[V(E), V(S)]$
3. Partition S into S_1 and S_2 containing n_1 and n_2 points using $k-$means with $K = 2$
4. Calculate E_1, E_2 and volumes $V(E_1)$, $V(E_2)$
5. Enlarge E_k $(k = 1, 2)$ so that $V(E_k) = \max[V(E_k), V(S_k)]$.
6. For all $\boldsymbol{u} \in S$, assign $\boldsymbol{u}$ to S_k such that $h_k(\boldsymbol{u}) = \min[h_1(x), h_2(x)]$
7. If no point reassigned goto 8; else goto 4
8. If $V(E_1) + V(E_2) < V(E)$ or $V(E) > 2V(S)$
   – partition S into S_1 and S_2
   – repeat entire algortihm for each subset S_1 and S_2
   else
   – return E as the optimal ellipsoid of the point set S

- EM algorithm quite computationally expensive, especially in high dimensions

- But... MultiNest need not perform full partitioning at each NS iteration

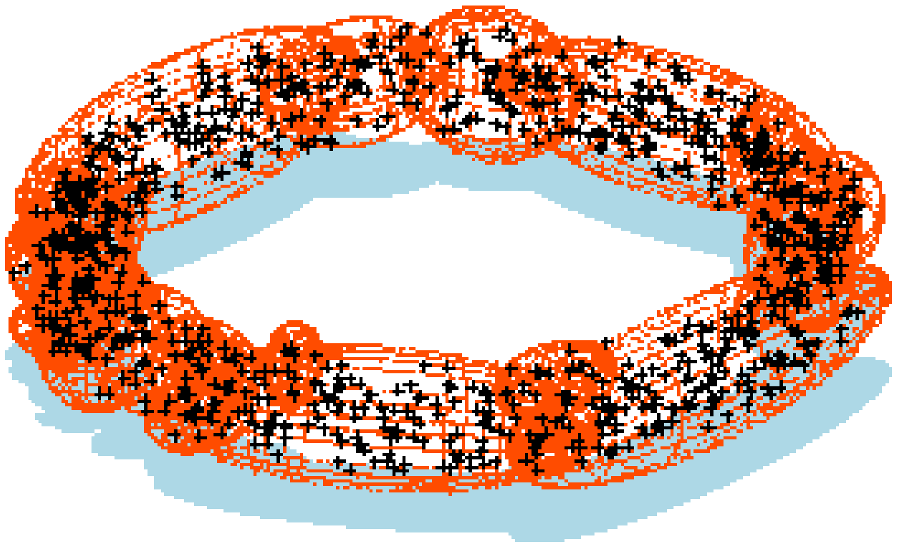
- Ellipsoids can be evolved through scaling at subsequent NS iterations $i + i'$ such that $V(E_k) = \mathrm{max}[V(E_k), X_{i+i'} n_k / N]$

- Ellipsoidal decomposition calculated at iteration i becomes less optimal as i' grows $\Rightarrow$ perform full re-partitioning of active points if $F(S) \geq h$ (typically $h = 1.1$)

- Possible that ellipsoids might not enclose the entire iso-likelihood contour, even though sum of their volumes must exceed prior volume X $\Rightarrow$ safer to set desired minimum volume as eX, where e is an enlargement factor

- Note: regardless of e-value, always ensure that E_k is a bounding ellipsoid of subset S_k.

SAMPLING FROM OVERLAPPING ELLIPSOIDS



- At each NS iteration, need to draw a new point uniformly from union of ellipsoids
- k Suppose K ellipsoids $\{E_k\}$, where kth one has volume $V(E_k)$
- Choose one ellipsoid with probability $p_k = V_k/V_{\rm tot}$
- Sample from chosen ellipsoid within hard constraint $L > L_i$
- Find number n_e of ellipsoids in which sample lies; accept with probability $1/n_e$

TRIVIAL PARALLELIZATION

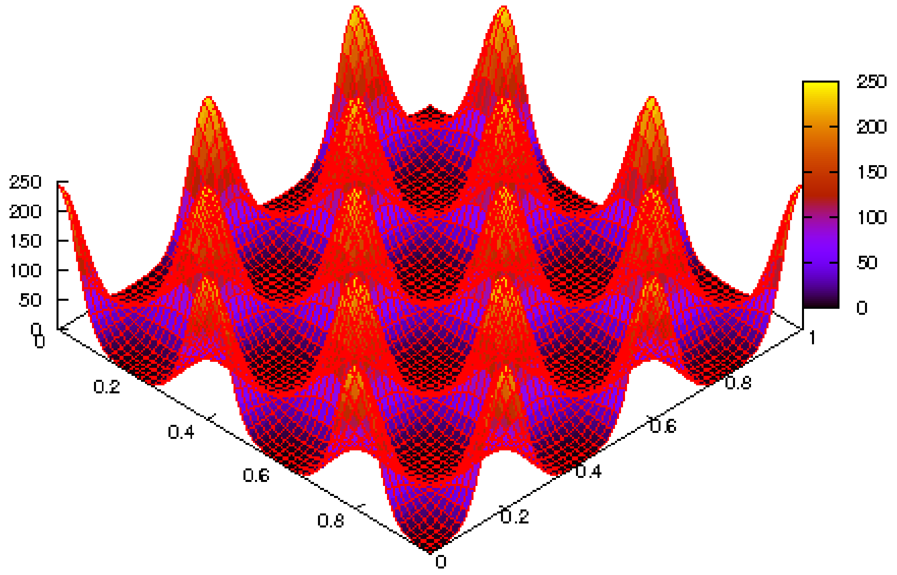
- Typical sampling efficiency less than unity since
  – ellipsoidal approximation to iso-likelihood surface not perfect
  – ellipsoids may overlap (as discussed above)

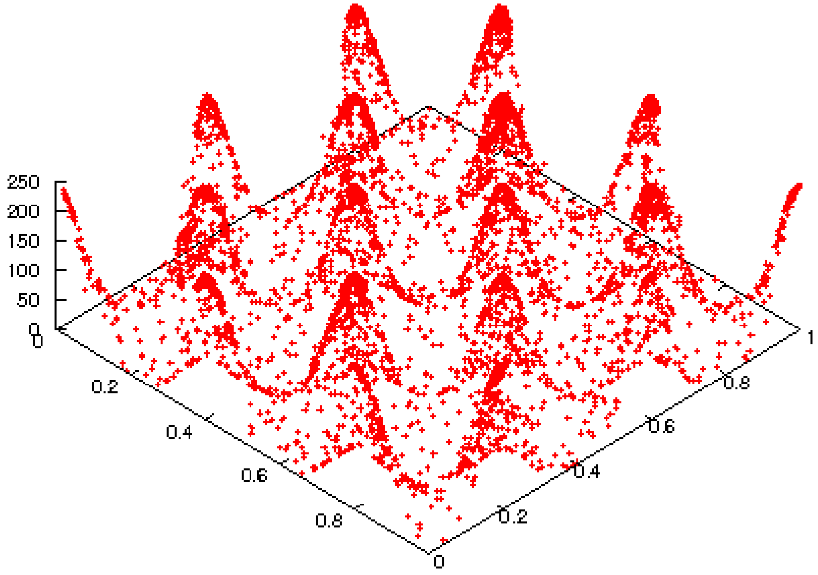
- But… MULTINEST algorithm usefully (and easily) parallelized

- At each NS iteration, draw a potential replacement point on each of $N_{\rm CPU}$ processors, where $1/N_{\rm CPU}$ is an estimate of the sampling efficiency

⇒ Effective efficiency close to unity across $N_{\rm CPU}$

IDENTIFICATION OF POSTERIOR MODES

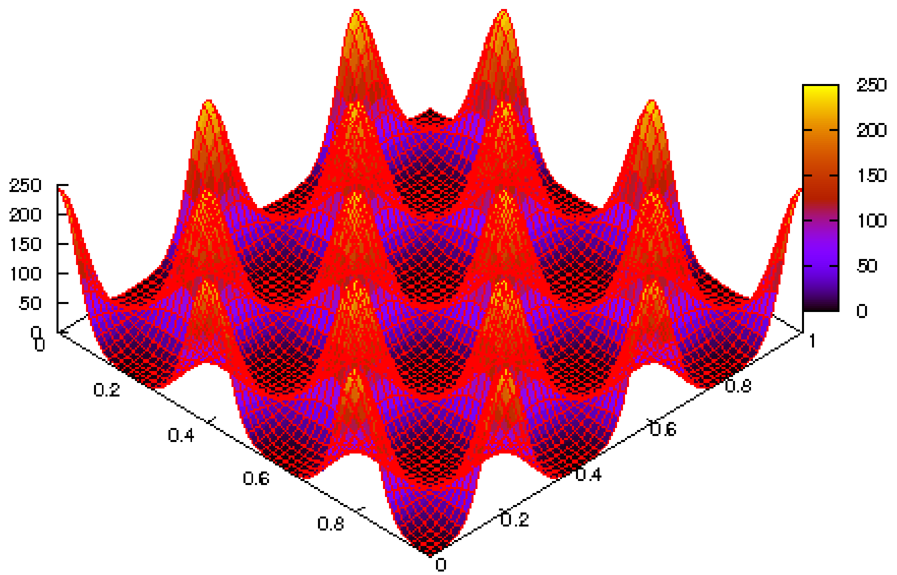


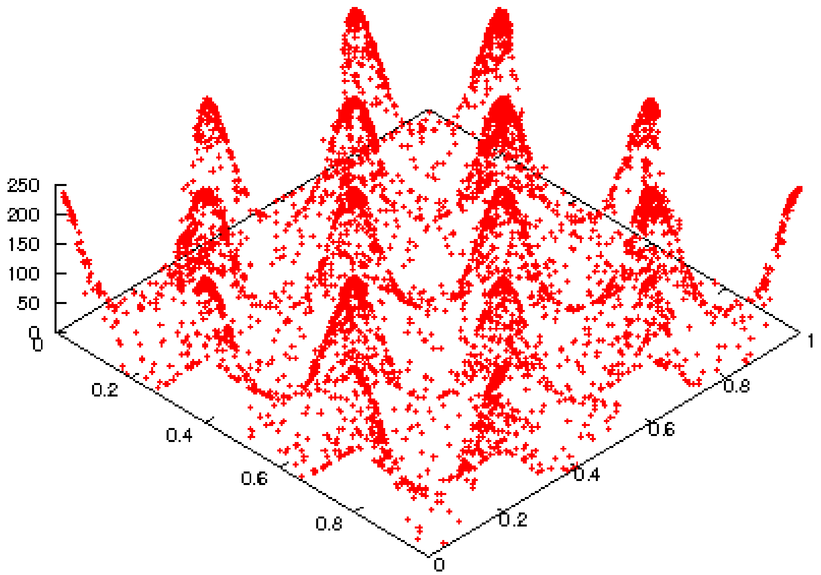


- For multimodal posteriors, useful to identify which samples ‘belong’ to which mode

- Some arbitrariness in this process: modes sit on top of some general ‘background’ of probability distribution

- Moreover, modes lying close together may only ‘separate out’ at relatively high likelihood levels



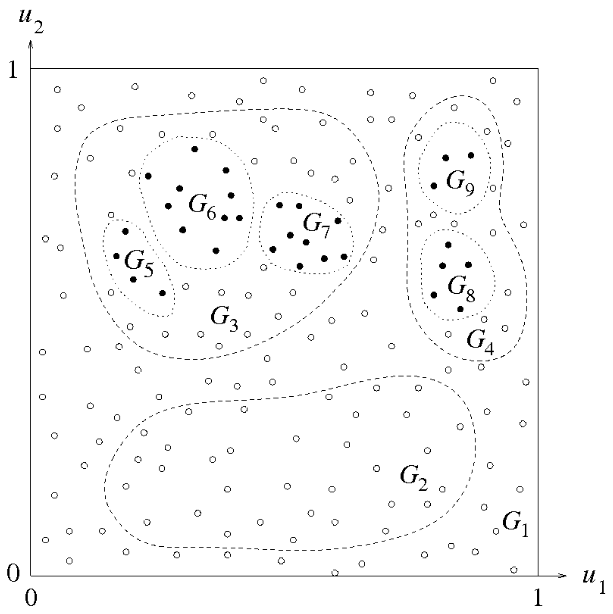


- Nonetheless, for well-defined ‘isolated’ modes:
  – can make reasonable estimate of posterior mass each contains (‘local’ evidence)
  – can construct posterior parameter constraints associated with each mode

- Once NS process reached likelihood such that ‘footprint’ of mode well-defined ⇒ identify at each subsequent iteration the points in active set belonging to mode

- Partitioning and ellipsoids construction algorithm described above provides efficient and reliable method for performing this identification

Mode identification algorithm



1. In first NS iteration, assign all active points to active group G_1

2. In subsequent NS iterations, pick subset S_k of G_1 at random:
   – S_k points become first members of 'temporary set' $\mathcal{T}$
   – E_k becomes first member of 'ellipsoid set' $\mathcal{E}$

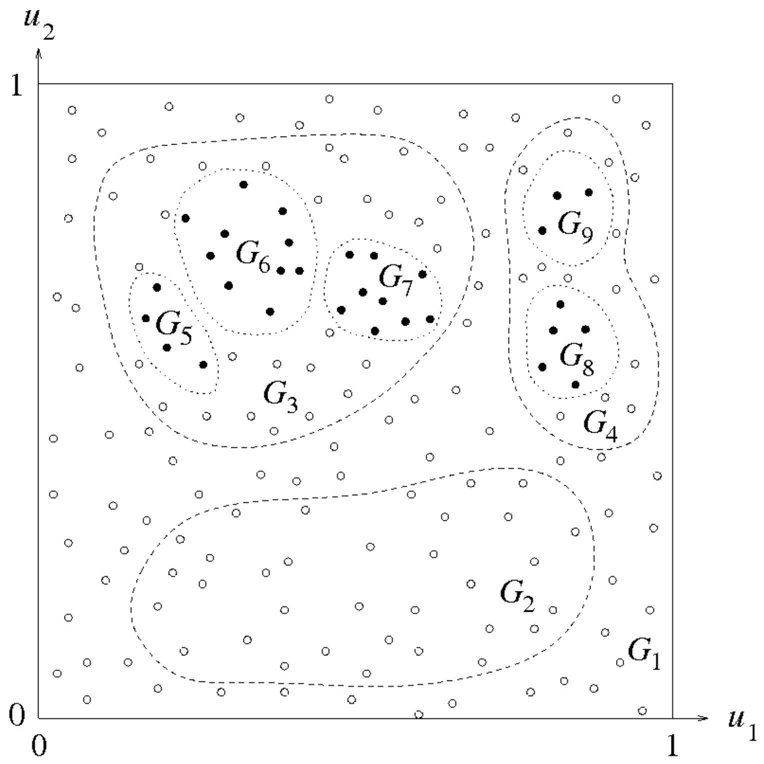
3. For all $E_{k'} \notin \mathcal{E}$, determine if $E_{k'}$ intersects any ellipsoid in $\mathcal{E}$

4. If no such intersections occur:
   – goto 5
   else, for each such intersecting ellipsoid $E_{k'}$:
   – add $S_{k'}$ points to $\mathcal{T}$ and add $E_{k'}$ to $\mathcal{E}$
   – goto 3

5. If all ellipsoids are members of $\mathcal{E}$:
   – (re)assign points in $\mathcal{T}$ to G_1
   else
   – (re)assign points in $\mathcal{T}$ to new active group G_2
   – (re)assign remaining active points to new active group G_3
   – group G_1 becomes 'inactive'

6. In current NS iteration, goto 2 and repeat algorithm for each active group until no new active groups occur

7. In subsequent NS iterations, apply algorithm to each active group

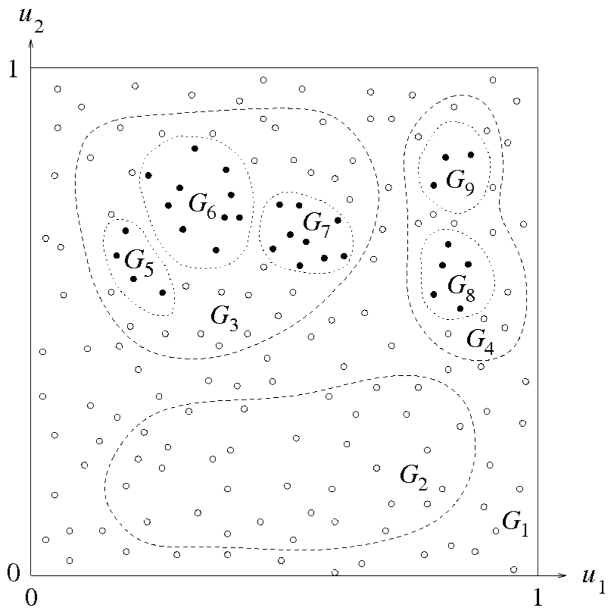


- At end of NS process ⇒ set of inactive groups and set of active groups, which together partition the full set of (inactive and active) sample points generated

- Note: as NS process reaches higher likelihoods, number of active points in any particular active group may dwindle to zero, but. . . group still considered active since it remains unsplit at the end of NS run.

- Finally, each active group is promoted to a ‘mode’, resulting in a set of L (say) such modes $\{M_l\}$.

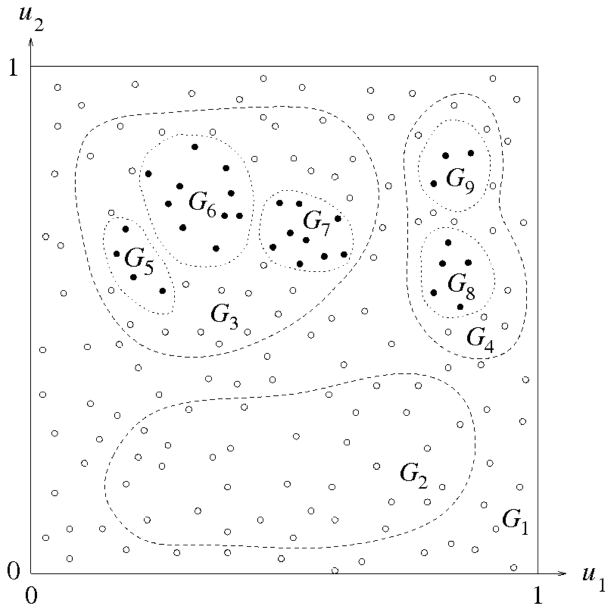
EVALUATION OF LOCAL EVIDENCES



- Suppose lth mode M_l contains the points $\{\boldsymbol{u}_j\}$ $(j = 1, n_l)$
- In simplest approach, local evidence of mode is

$$\mathcal{Z}_l = \sum_{j=1}^{n_l} \mathcal{L}_j w_j$$

where $w_j = X_M/N$ for each active point in M_l and $w_j = \frac{1}{2}(X_{i-1} - X_{i+1})$ for each inactive point (i is NS iteration when inactive point was discarded).



- Similarly, posterior inferences resulting from lth mode obtained by weighting each point in M_l by $p_j = \mathcal{L}_j w_j / Z_l$.

- But… local evidence underestimated for modes lying close together – only identified as separate regions at high likelihood values

- Overcome problem by also making use of points in the inactive groups at end of NS process

- For each mode M_l, expression local evidence now reads

$$\mathcal{Z}_l = \sum_{j=1}^{n_l} \mathcal{L}_j w_j + \sum_g \mathcal{L}_g w_g \alpha_g^{(l)},$$

  where sum over g includes all points in inactive groups, $w_g = \frac{1}{2}(X_{i-1} - X_{i+1})$ as above, and additional factors $\alpha_g^{(l)}$ are calculated as set out below.

- Similarly, posterior inferences from lth mode obtained by weighting each point in M_l by $p_j = \mathcal{L}_j w_j / Z_l$ and each point in inactive groups by $p_g = \mathcal{L}_g w_g \alpha_g^{(l)} / Z_l$

- Factors $\alpha_g^{(l)}$ most easily determined by essentially reversing the mode identification process

- Each mode M_l is simply a renamned active group G

- Identify inactive group G' that split to form G at the NS iteration i

- Assign all points in G' the factor

$$\alpha_g^{(l)} = \frac{n_G^{(A)}(i)}{n_{G'}^{(A)}(i)},$$

where $n_G^{(A)}(i)$ is number of active points in G at NS iteration i; similar for $n_{G'}^{(A)}(i)$.

- Now, G' may itself have formed when an inactive group G'' split at an eariler NS iteration $i' < i$, in which case all points in G'' are assigned the factor

$$\alpha_g^{(l)} = \frac{n_G^{(A)}(i)}{n_{G'}^{(A)}(i)} \frac{n_{G'}^{(A)}(i')}{n_{G''}^{(A)}(i')}.$$

- Process is continued until the recursion terminates

- Finally, all points in inactive groups not already assigned have $\alpha_g^{(l)} = 0$.

- Easy to show $\sum_{l=1}^{L} \mathcal{Z}_l = \mathcal{Z}$, the global evidence $\Rightarrow$ evidence exactly partitioned

- Note: can instead use mixture model to assign factors