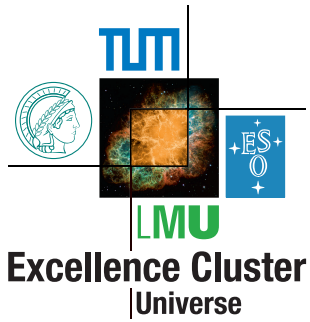
Population Monte Carlo

Frederik.Beaujean@lmu.de

Helmholtz school on
Monte Carlo methods in advanced statistics applications and data analysis
Nov 22, 2013



Plan

SUMMARY

- population Monte Carlo (PMC) ≡ adaptive importance sampling
- complicated target $P \Rightarrow$ need flexible proposal function q

$$q(\vec{x}) = \sum_{j=1}^{K} \alpha_j q_j(\vec{x} \,|\, \vec{\mu}_j, \Sigma_j)$$

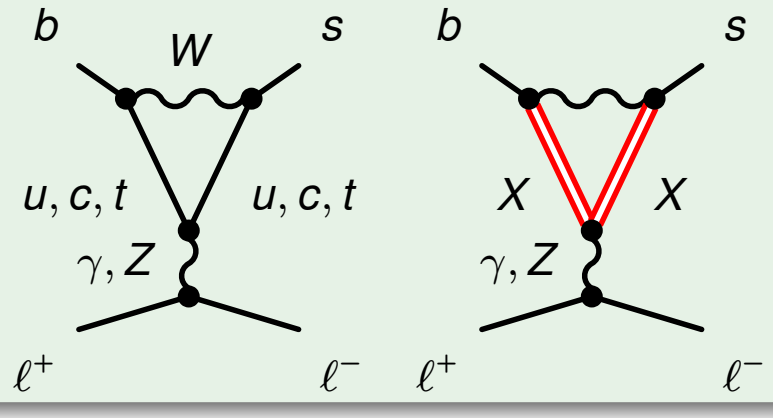
- adapt q iteratively to match P
- big issue: initialization

Plan

New physics in rare B-meson decays

Flavor changing neutral currents in SM

- quark transition $b \to s$ at loop level
- Rare decay: $\mathcal{B}\left(B \to K\mu^+\mu^-\right) \sim 10^{-7}$
- indirect search for heavy particles up to $\mathcal{O}(\mathrm{TeV})$

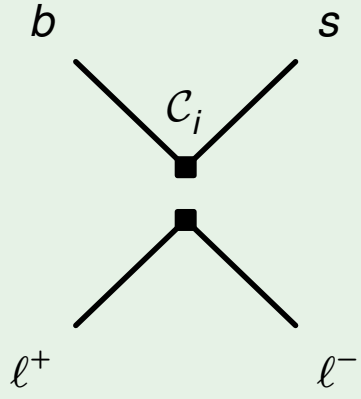


Effective theory

- high energy: effective coupling (Wilson coefficient) $\mathcal{C}_i$
- low energy: effective operator $\mathcal{O}_i$

$$\mathcal{L}_{\text{eff}} \sim \sum_i (\mathcal{C}_i^{\text{SM}} + \Delta\mathcal{C}_i)\mathcal{O}_i$$

$\Rightarrow$ Extract $\vec{\mathcal{C}}$ in global fit



Global fit

The goal

1. Compare SM with new physics: Bayes factor
2. Extract marginal distribution of $\vec{\mathcal{C}}, \dim \vec{\mathcal{C}} = 2 \ldots 6$

Bayes' theorem

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

$$P(\vec{\mathcal{C}}, \vec{\nu}|D) = \frac{P(D|\vec{\mathcal{C}}, \vec{\nu}) P(\vec{\mathcal{C}}) P(\vec{\nu})}{\int \mathrm{d}\vec{\mathcal{C}} \mathrm{d}\vec{\nu} P(D|\vec{\mathcal{C}}, \vec{\nu}) P(\vec{\mathcal{C}}) P(\vec{\nu})}$$

$$\text{Bayes factor} = \frac{\text{evidence|SM}}{\text{evidence|NP}}$$

Inputs

- theory uncertainty: quark masses, nonperturbative QCD ... $\Rightarrow P(\vec{\nu}), \dim \vec{\nu} \approx 30$
- over 50 results from Belle, LHCb ... experiments $\Rightarrow P(D|\vec{\mathcal{C}}, \vec{\nu})$

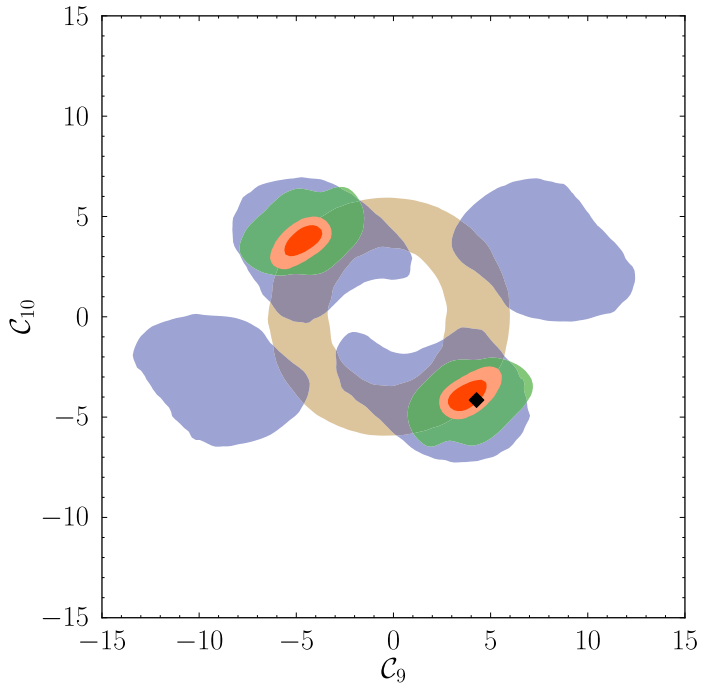
Results of adaptive importance sampling arXiv:1205.1838

Combinations of input

1. all data: 1σ 2σ
2. Standard Model: ◆
3. data subsets . . .

Challenges

- Multiple isolated maxima: observable $X \sim \mathcal{C}_i \mathcal{C}_j = (-\mathcal{C}_i)\,(-\mathcal{C}_j)$
- Theory prediction slow: $\sim 1\,\mathrm{s}$ for one sample, but need $\mathcal{O}(10^6)$ samples $\Rightarrow$ parallel evaluation
- 30 dimensions

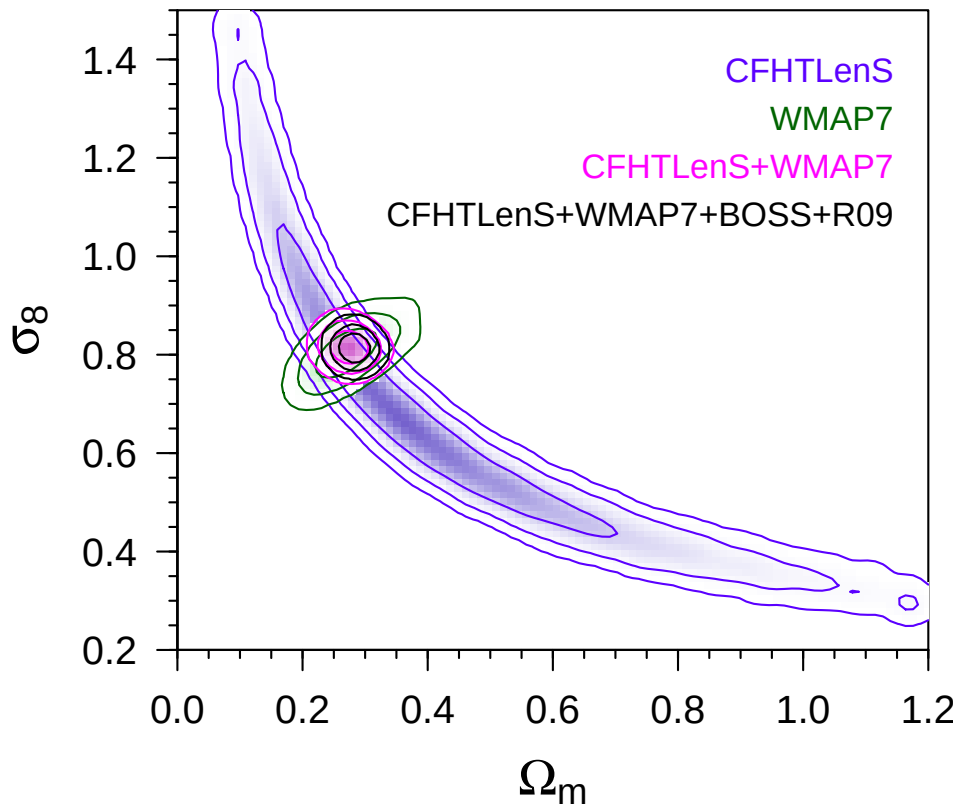


Bayes factor

$$\frac{\text{evidence}|\text{SM}}{\text{evidence}|\text{NP}} = 200 \ldots 1500$$

COSMOLOGY

- Kilbinger, Cappé, et al. develop cosmoPMC to constrain cosmological parameters in global fits
- cosmoPMC contains `pmclib`, C implementation of PMC with MPI parallelization



Kilbinger et al. arXiv:1212.3338

Black box — holy grail of MC sampling

Target

- just code that returns the value at $\vec{x}$
- any number of modes, curvature or degeneracy
- any dimension

Sampler

- use any number of cores effectively
- no manual tuning required
- output always reliable

Plan

PMC BASICS I

Regular importance sampling: $\vec{x} \in V \subseteq \mathbb{R}^d$

$$\mathrm{I} = \int \mathrm{d}\vec{x} P(\vec{x}) = \int \mathrm{d}\vec{x} \frac{P(\vec{x})}{q(\vec{x})} q(\vec{x}) \tag{1}$$

$$\Rightarrow \hat{\mathrm{I}} = \frac{1}{N} \sum_{i=1}^{N} \frac{P(\vec{x}^i)}{q(\vec{x}^i)} = \frac{1}{N} \sum_{i=1}^{N} w^i, \vec{x} \sim q \tag{2}$$

Terminology:

$$q : \text{proposal density} \tag{3}$$

$$P : \text{target density (often: posterior)} \tag{4}$$

$$w \equiv \frac{P}{q} \text{ importance weight} \tag{5}$$

Self-normalized weights

$$\overline{w}^k \equiv \frac{w^k}{\sum_i w^i} \tag{6}$$

Good for arbitary expectation

$$E_P[f] \equiv \int \mathrm{d}\vec{x} P(\vec{x}) f(\vec{x}) \approx \sum_{i=1}^{N} \overline{w}^i f(\vec{x}^i) \tag{7}$$

Law of large numbers: converges as

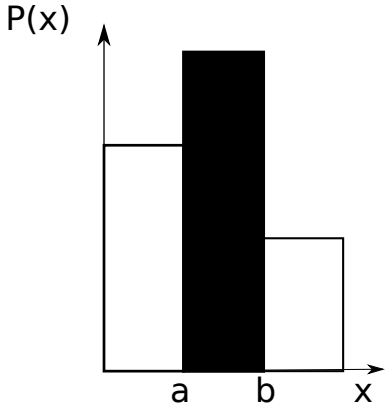
$$\sim 1/\sqrt{N} \tag{8}$$

MARGINAL DISTRIBUTIONS

Main application: marginal distributions, example in 1D

$$P(a \leq x_1 \leq b) = \int_a^b \mathrm{d}x_1 \int \mathrm{d}x_j P(\vec{x}) \tag{9}$$

$$P(a \leq x_1 \leq b) \approx \sum_{i=1}^{N} w^i f(\vec{x}^i) \tag{10}$$

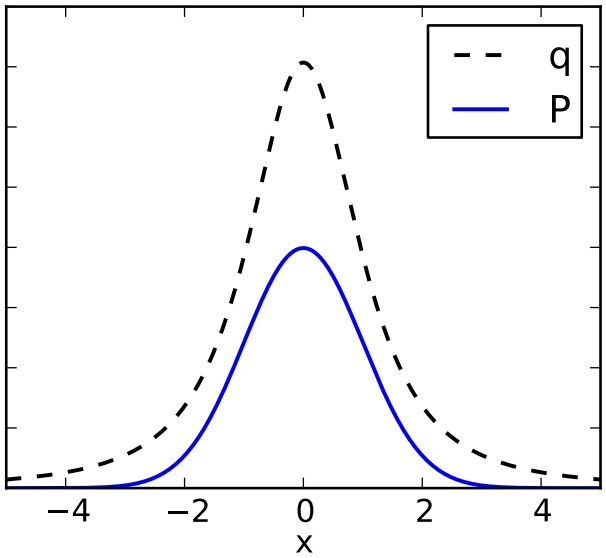


with the indicator function

$$f(\vec{x}) = \begin{cases} 1, x_1 \in [a, b] \\ 0, \text{else} \end{cases} \tag{11}$$

Note: For Markov chains, $\vec{x} \sim P \Rightarrow w \equiv 1$

Proposal Density



Requirements:

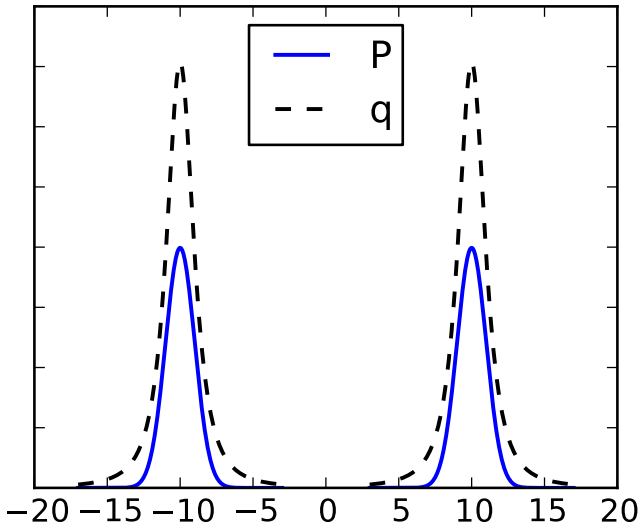
- supp $P \subseteq$ supp q
- normalized
  $\Rightarrow$ get normalization of P
- falls of slower than P for large $\vec{x}$
  $\Rightarrow$ estimators have finite variance

Desirable:

- easy to draw from
- for most applications:

$q \propto P$ optimal

Mixture Density



A very flexible class of functions:

$$q(\vec{x}) = \sum_{j=1}^{K} \alpha_j q_j(\vec{x} \mid \vec{\mu}_j, \Sigma_j) \tag{12}$$

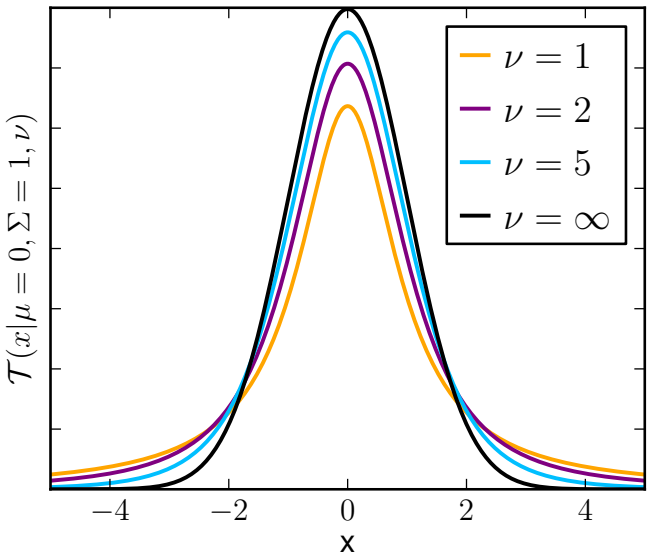
q_j = Gauss, $\mathcal{N}$ or Student's t distribution, $\mathcal{T}$

α_j = component weight, $\sum_j \alpha_j = 1$

STUDENT'S T

$$\mathcal{T}(\vec{x}|\vec{\mu},\mathbf{\Sigma},\nu) = \frac{\Gamma\left((\nu+d)/2\right)}{\Gamma\left(\nu/2\right)(\pi\nu)^{d/2}}|\mathbf{\Sigma}|^{-1/2}\left(1+\frac{1}{\nu}(\vec{x}-\vec{\mu})^T\mathbf{\Sigma}^{-1}(\vec{x}-\vec{\mu})\right)^{-(\nu+p)/2} \quad (13)$$

- mean $\vec{\mu}$, covariance $\mathbf{\Sigma}\frac{\nu}{\nu-2}$
- $\nu \equiv$ degree of freedom
- $\nu = 1 \Rightarrow$ Cauchy, Lorentz, Breit-Wigner
- $\nu = \infty \Rightarrow$ Gauss, $\mathcal{N}$



wikipedia

How to adapt q?

1. need to measure discrepancy between P, q
2. then minimize discrepancy

How to adapt q?

1. need to measure discrepancy between P, q
2. then minimize discrepancy

1. Kullback-Leibler divergence (from P to q)

$$\mathrm{KL}(P||q) = \int \mathrm{d}\vec{x} P(\vec{x}) \log \frac{P(\vec{x})}{q(\vec{x})} \tag{14}$$

- $\mathrm{KL}(P||q) \neq \mathrm{KL}(q||P) \Rightarrow$ KL not a metric
- $\mathrm{KL} \geq 0$ and $\mathrm{KL} = 0 \Leftrightarrow P \equiv q$
- KL hard to compute, even harder than $\int \mathrm{d}\vec{x} P(\vec{x})$

How to adapt q?

1. need to measure discrepancy between P, q
2. then minimize discrepancy

2. Simplification: restrict to mixture density $q = \sum_{j=1}^{K} \alpha_j q_j(\cdot|\mu_j, \mathbf{\Sigma}_j)$

   - functional optimization $\rightarrow$ ordinary parameter optimization
   - Global fit arXiv:1205.1838

$$d = 30, K = 100 \Rightarrow \dim \alpha + (\dim \vec{\mu} + \dim \mathbf{\Sigma}) \times \dim \alpha \quad (14)$$

$$100 + (30 + 30 \times 31/2) \times 100 \quad (15)$$

$$= 49600 \text{ parameters to fix} \quad (16)$$

   - too much for gradient-based methods like Minuit!
   - can't even compute $\mathrm{KL}(P||q)$!

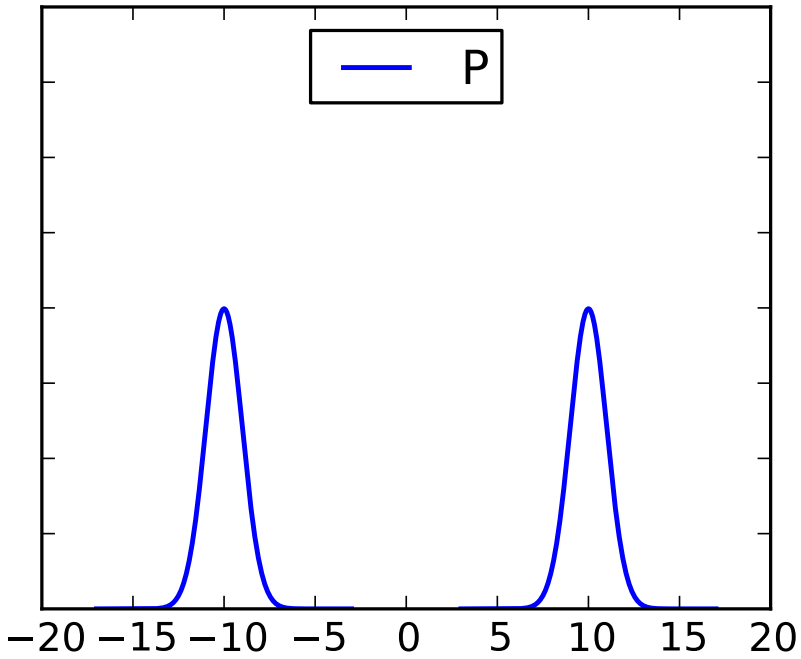
A CLEVER SOLUTION

- use analytical structure of the problem
- and expectation maximization (EM) Dempster, Laird, Rubin (1977)
- easy reading: Piater (2002) Borman (2004)

A CLEVER SOLUTION

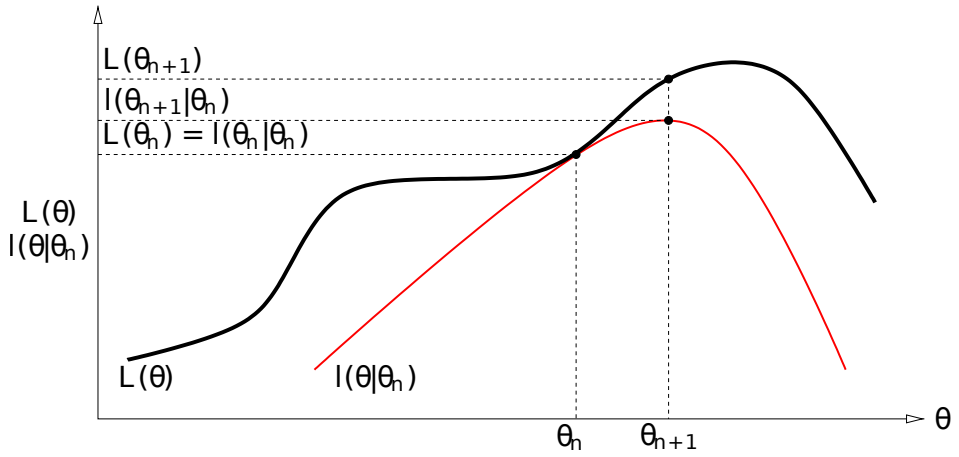
- use analytical structure of the problem
- and expectation maximization (EM) Dempster, Laird, Rubin (1977)
- easy reading: Piater (2002) Borman (2004)

EM — BASIC IDEA



- Observe $D = \{x^i : i = 1 \ldots N\}$
- Assume $x^i \sim P(\cdot) = \sum_{j=1}^{K} \alpha_j \mathcal{N}(\cdot|\mu_j, \sigma_j)$
- Maximum-likelihood fit of mixture to data
- If we knew generating component $z_i \in \{1, \ldots, K\} \Rightarrow K$ indep. problems
- Solution easy: $\mu_j = 1/N_j \sum_i x^i$, $\sigma_j^2 = 1 \Big/ N_j^2 \sum_i (x^i - \mu_j)^2$

Expectation Maximization in a nutshell



Borman (2004)

- iterative procedure to maximize L: E step $\leftrightarrow$ M step
- E step: infer missing data given current θ_n and observed data
- M step: maximize simpler function: $\theta_{n+1} = \arg\max l(\theta|\theta_n)$
- converge to *local* optimum in $\mathcal{O}(10)$ steps

THE STANDARD PMC ALGORITHM CAPPÉ ET AL. (2008) WRAITH ET AL. (2009)

1. Initial guess q^0 at $t = 0$
2. Draw N samples $\vec{x}^i \sim q^t$, compute $w^i = \dfrac{P(\vec{x}^i)}{q(\vec{x}^i)}$
3. Update q^t based on $\left\{(\vec{x}^i, w^i) : i = 1, \ldots, N\right\}$
4. Repeat steps 2 and 3 until convergence
5. If converged, regular IS with q^t

NOTE

- computing weights usually slowest part
- samples in step 2 indep. $\Rightarrow$ massive parallelization
- samples form *population* $\Rightarrow$ population Monte Carlo

Convergence — or when to stop

Optimization goal

$$\mathrm{KL}(P||q^t) = 0 \Rightarrow \exp\left[\mathrm{KL}(P||q^t)\right] = 1 \tag{17}$$

estimated by *normalized perplexity*

$$\mathcal{P}^t \equiv \frac{\exp(H^t)}{N} \in [0,1] \tag{18}$$

Shannon entropy of self-normalized weights Shannon (1948)

$$H(\{\overline{w}_i^t\}) = -\sum_{i=1}^{N} \overline{w}_i^t \log \overline{w}_i^t \tag{19}$$

Compare with quantum mechanics

$$S = -k_B \sum_i p_i \log p_i, \text{ where } p_i \equiv P(\text{state } i) \tag{20}$$

Maximum entropy for

$$q^t = P \Rightarrow \overline{w} = 1/N \Rightarrow H^t = \log N \tag{21}$$

SAMPLE QUALITY

What fraction of samples actually contributes? *effective sample size*

$$\mathrm{ESS}^t \equiv \left(\sum_{i=1}^{N} \{\overline{w}_i^t\}^2 \right)^{-1} \Big/ N \in [1/N, 1] \tag{22}$$

Examples:

$$\overline{w} = 1/N \Leftrightarrow ESS = 1 \tag{23}$$

$$\overline{w} = \{\underbrace{0, \ldots, 0}_{N_0}, \underbrace{c, \ldots, c}_{N-N_0}\} \Rightarrow \mathrm{ESS} = \frac{N - N_0}{N} \tag{24}$$

Extreme case: single outlier dominates all samples

$$N_0 = N - 1 \Rightarrow \mathrm{ESS} = 1/N \tag{25}$$

Convergence criterion

$$\mathcal{P}^t = \mathcal{P}^t\left(\overline{w}_i^t \log \overline{w}_i^t\right)$$

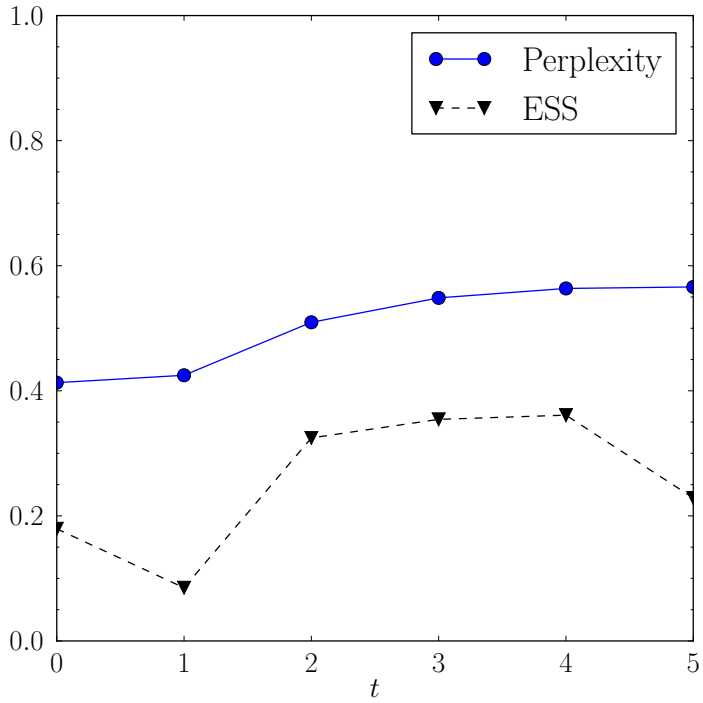
vs

$$\mathrm{ESS}^t = \mathrm{ESS}^t\left(\{\overline{w}_i^t\}^2\right)$$

- ESS better indicator of outliers
- $\mathcal{P}$ more robust indicator
- stop adapting q^t if in last two steps

$$\frac{\sqrt{V[\mathcal{P}]}}{E[\mathcal{P}]} \leq \epsilon \qquad (26)$$

with $\varepsilon \sim 10\%$



A tough example in 22D

How to start?

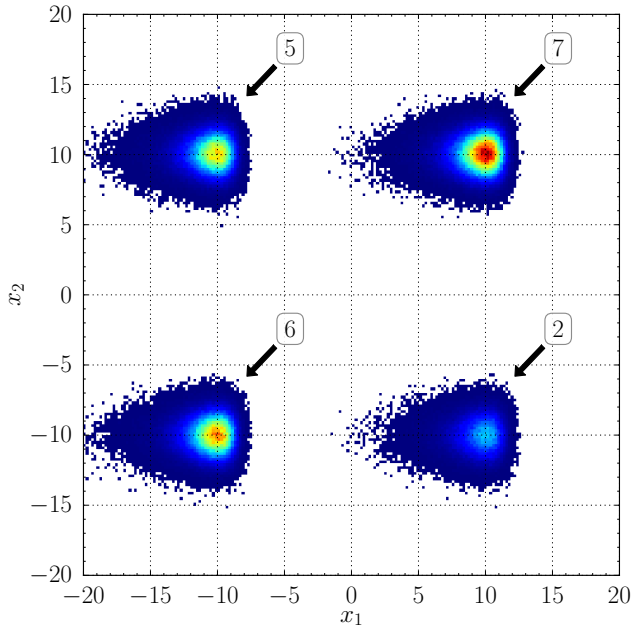
Plan

Monte Carlo sampling

Goal: draw samples from target P

1. Markov chain Monte Carlo (MCMC)

	Pro	Con
MCMC	local exploration, learns on the fly	trapped in local maxima



Four identical maxima

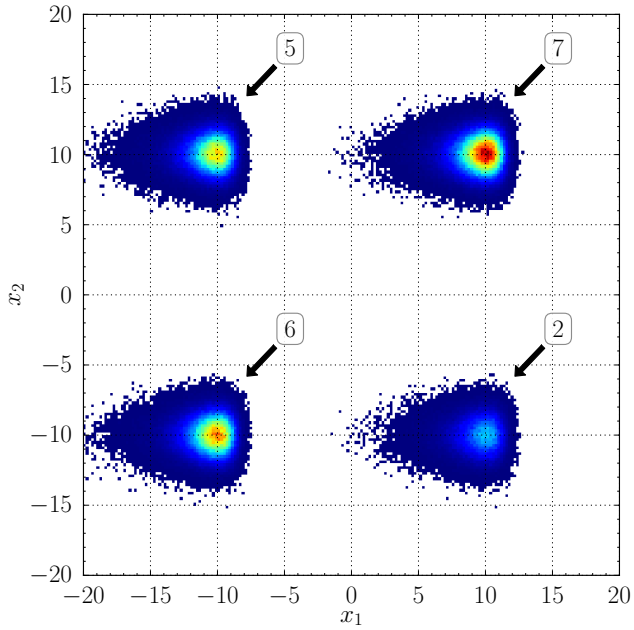
- start 20 local-random-walk chains
- each trapped in one mode

Monte Carlo sampling

Goal: draw samples from target P

1. Markov chain Monte Carlo (MCMC)
2. adaptive importance sampling = population Monte Carlo (PMC)

	Pro	Con
MCMC	local exploration, learns on the fly	trapped in local maxima
PMC	massive parallelization, yields normalization, multiple modes OK	very sensitive to initialization



Four identical maxima

- start 20 local-random-walk chains
- each trapped in one mode

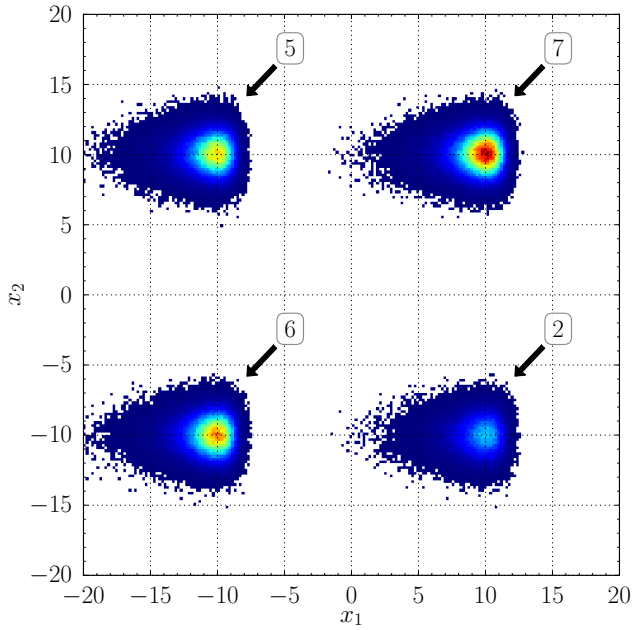
Monte Carlo sampling

Goal: draw samples from target P

1. Markov chain Monte Carlo (MCMC)
2. adaptive importance sampling = population Monte Carlo (PMC)

	Pro	Con
MCMC	local exploration, learns on the fly	trapped in local maxima
PMC	massive parallelization, yields normalization, multiple modes OK	very sensitive to initialization

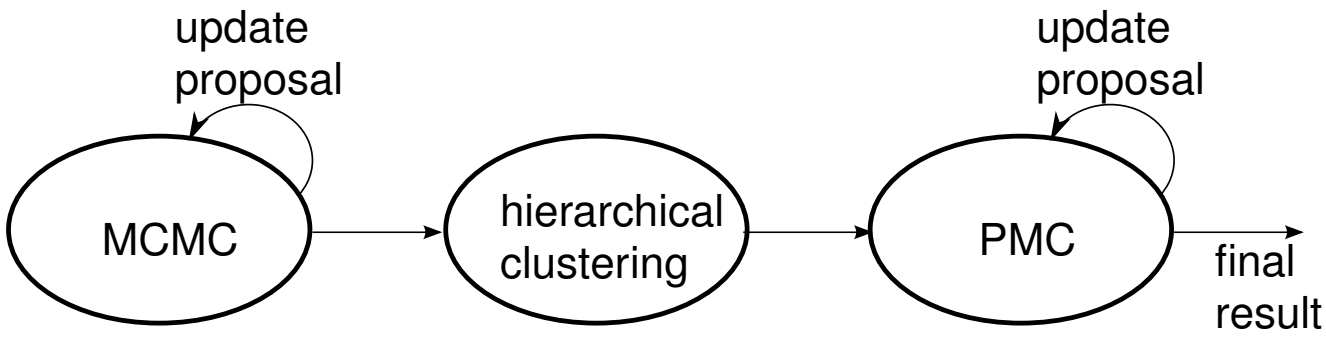
$\Rightarrow$ Combine MCMC and PMC



Four identical maxima

- start 20 local-random-walk chains
- each trapped in one mode

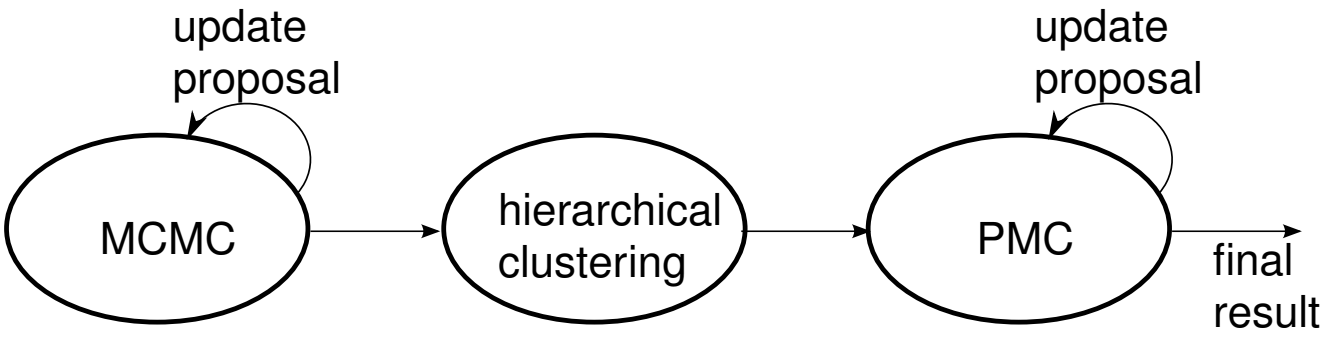
Initializing PMC Beaujean & Caldwell (2013)



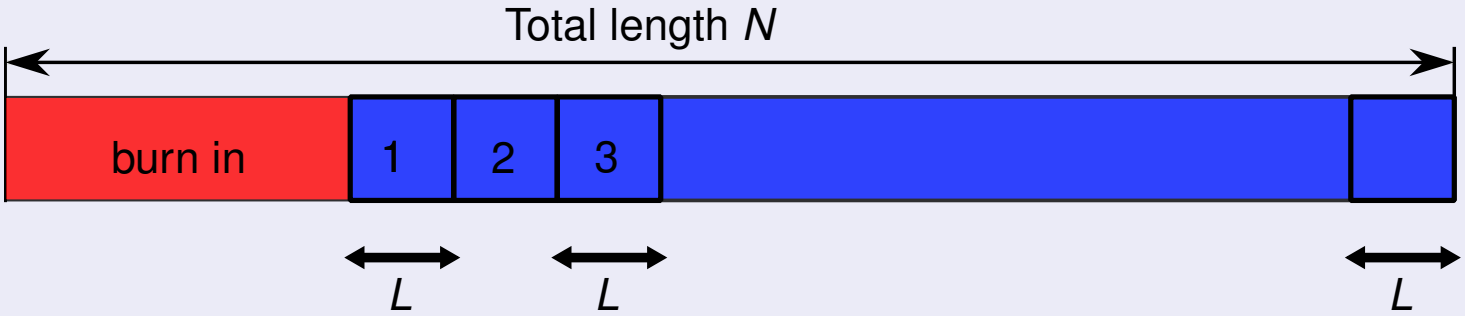
Markov chain implementation Haario et al. (2001)

- starting point drawn uniformly across parameter space
- Gaussian proposal whose covariance is
  - adapted to target, $\mathbf{\Sigma} \to \mathbf{\Sigma}_P$
  - scaled such that proposed points are accepted with $(15 - 35)\%$.
- don't have to stop adapting, quit after N steps
- chains stuck $\Rightarrow$ multiple chains

INITIALIZING PMC BEAUJEAN & CALDWELL (2013)



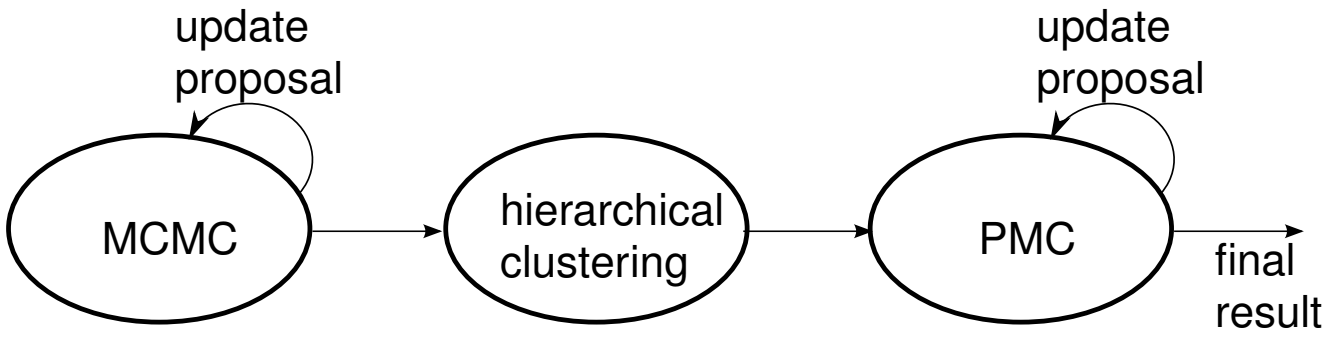
SPLIT UP THE LOCAL-RANDOM-WALK MARKOV CHAIN



patch of length $L \stackrel{!}{=}$ one component

$$\text{sample mean } \vec{\mu} = \frac{1}{N}\sum_{i=1}^{L}\vec{x}^i, \quad \text{sample cov. } \mathbf{\Sigma} = \frac{1}{N-1}\sum_{i=1}^{L}(\vec{x}^i - \vec{\mu})(\vec{x}^i - \vec{\mu})^T$$

Initializing PMC Beaujean & Caldwell (2013)

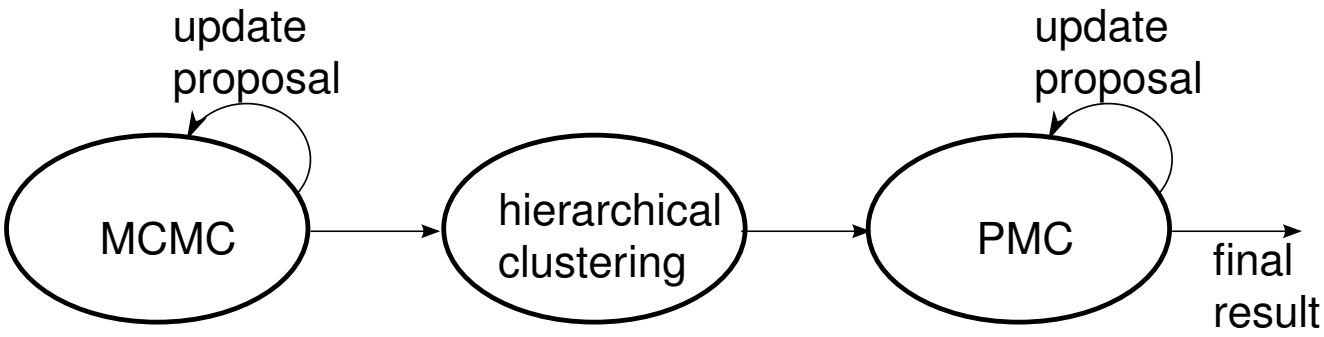


Hierarchical clustering I

Goal: condense information

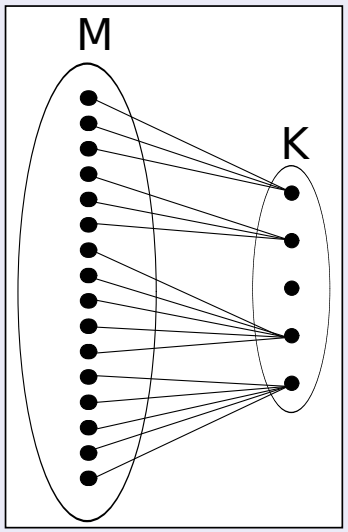
- Have *Gaussian* mixture with M components $f(\vec{x}) = \sum_{l=1}^{M} \beta_l f_l(\vec{x} \,|\, \vec{\mu}_l, \Sigma_l)$,
- Ex: 20 chains, 10000 samples, 20% burn-in, $L = 100 \Rightarrow M = 1600$ (huge!)
- Want mixture with $K \ll M$ components $q(\vec{x}) = \sum_{j=1}^{m} \alpha_j q_j(\vec{x} \,|\, \vec{\mu}_j, \Sigma_j)$, say $K = 10$
- Find q "closest" to f. Sounds familiar?

Initializing PMC Beaujean & Caldwell (2013)

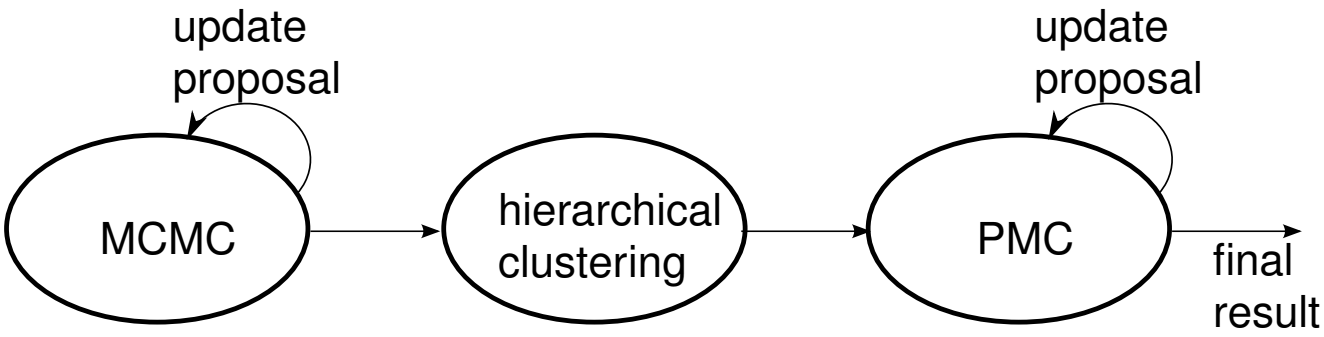


Hierarchical clustering II Goldberger & Roweis (2004)

- distance measure $d(f,q) = \sum_{l=1}^{M} \alpha_l \min_j \mathrm{KL}(f_l \| q_j)$,
- $\mathrm{KL}(f_l \| q_j)$ simple for Gaussians $\Rightarrow$ fast
- map many f_l to closest q_j, adjust q_j to minimze $d(f,q)$
- use expectation-maximization, need initial guess again

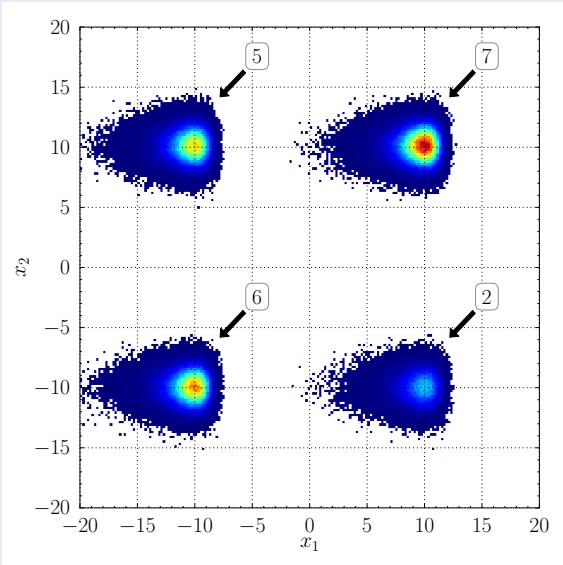


INITIALIZING PMC BEAUJEAN & CALDWELL (2013)

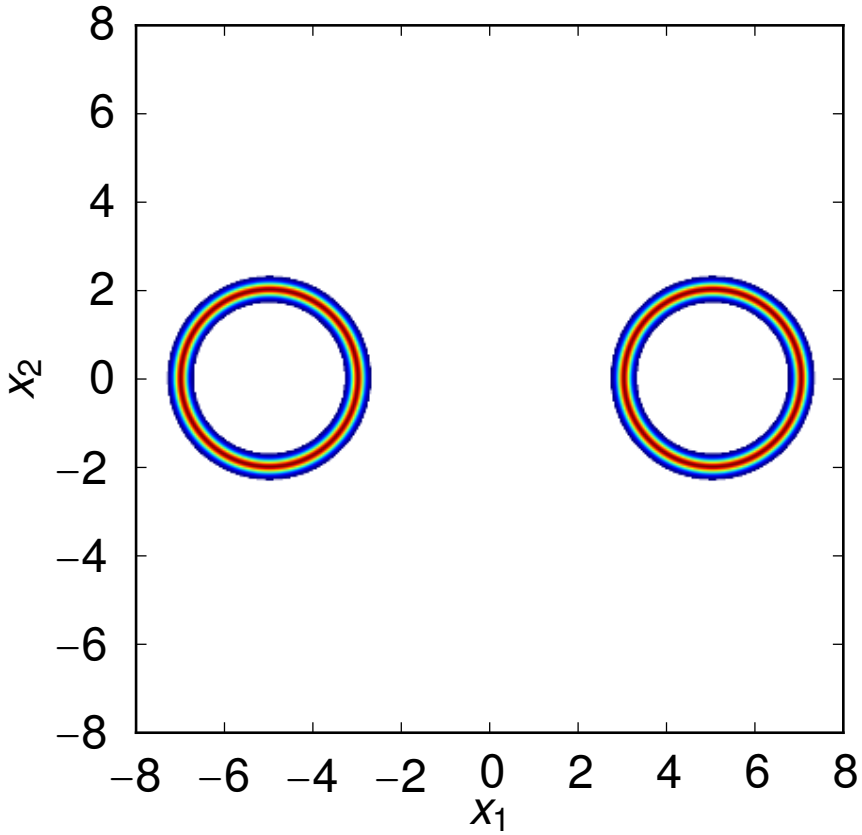


HIERARCHICAL CLUSTERING III: INITIAL GUESS

- pick K (too small $\Rightarrow$ poor proposal, too large $\Rightarrow$ wasting time)
- group chains: need similar mean and variance in *each* dimension
- same #components from each group
- sample mean, covariance from *long* chain patches



Example: Gaussian shells in 2D



$$P(\vec{x}) = \frac{1}{2}\,\mathrm{circ}(\vec{x}|\vec{c}_1, r, w) + \frac{1}{2}\,\mathrm{circ}(\vec{x}|\vec{c}_2, r, w)$$

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

$$\vec{c} = (\pm 5, 0), r = 2, w = 0.1$$

Challenges

- multimodal
- degeneracy

GETTING YOUR HANDS DIRTY

GET YOUR COPY OF THE EXERCISES

```
ssh -X USER@naf-school01.desy.de
% source /afs/desy.de/group/school/mc-school/setup_pmc.sh
% cp /afs/desy.de/user/s/school09/exercises.tar.bz ~
% tar xaf exercises.tar.bz ; ls exercises/
clustering  mcmc  pmc
```

- each folder has `exercise.pdf` with instructions
- use `evince` to open pdf files
- start with `mcmc`

Plan

INTEGRATION PRECISION BEAUJEAN & CALDWELL (2013)

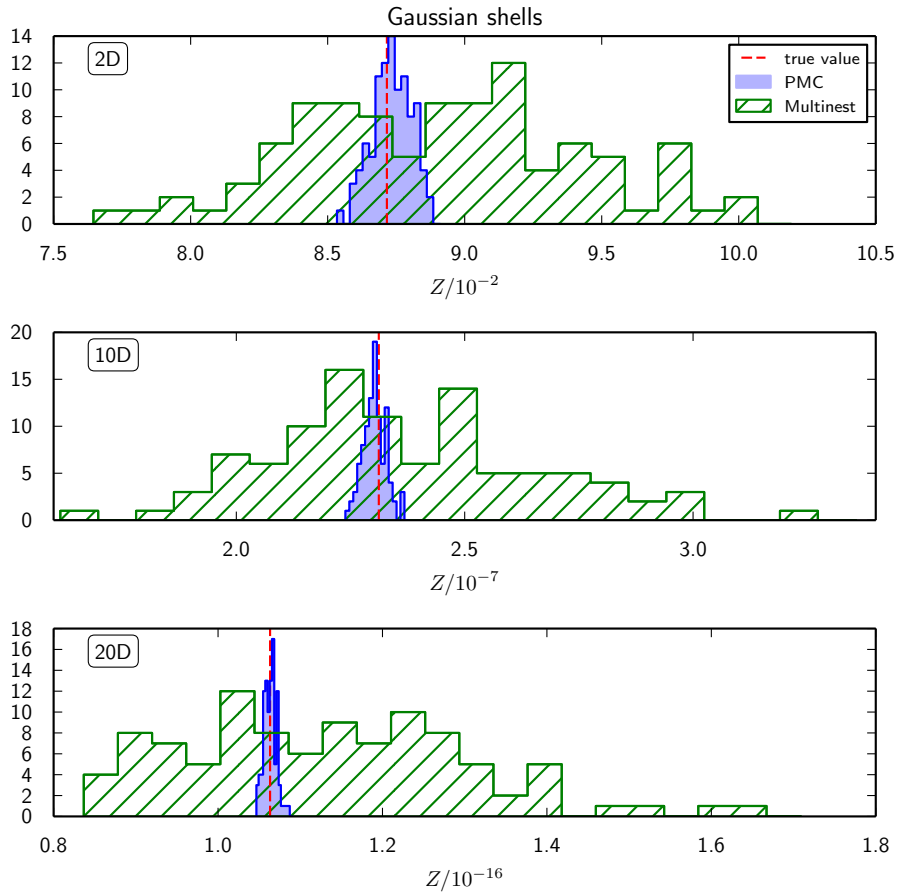
STATISTICAL UNCERTAINTY

$$\widehat{Z} = \frac{1}{N}\sum_{i=1}^{N} w^i$$

$$\widehat{\Delta Z} = \sqrt{\frac{1}{N(N-1)}\sum_{i=1}^{N}\left(w^i - \widehat{Z}\right)^2}$$

PMC VS MULTINEST V2.18

- "same number" of samples contribute
- $d = 2 : \widehat{\Delta Z} \lesssim 1\%$ (PMC) vs 5% (nest)
- variance of $\widehat{Z}$ in repetitions matches $\widehat{\Delta Z}$ very well



100 runs on Gaussian shells

Implementation

Current status

- Exercises use flavor tool `eos` (docs) and `pmclib` (backend)
- `matplotlib` (plots) and `HDF5` (I/O)

Future development

- Publish MCMC + clustering + PMC in python ⇒ `https://github.com/fredRos/pypmc.git`

Improvements

Goals

- User friendliness: less (no?) tuning parameters
- more robust results

Initialization

- Hierarchical clustering → variational Bayes
- sampling from Beta distr. ⇒ massive parallelization

PMC

- update student's t ν
- combine samples from multiple steps Cornuet et al. ⇒ soften outlier

$$w_t^i = P(\vec{x}^i) \Big/ \sum_t q^t(\vec{x}^i)$$

Conclusion

PMC

- is a powerful algorithm easily using 1000's of cores
- benefits from MCMC for initialization
- is not the holy grail