Hierarchical Bayesian Method with MCMC Integration on GPUs

5th GPU Workshop - The Future of Many-Core Computing in Science
Wigner Research Centre for Physics (Budapest), 20-21 May 2015

János Márk Szalai-Gindl[1] (szalai@complex.elte.hu)
István Csabai[1], László Dobos[1], Támás Budavári[2], Thomas J. Loredo[3], Brandon C. Kelly[4]

[1]Department of Physics of Complex Systems, ELTE
[2]The Johns Hopkins University, U.S.
[3]Cornell University, U.S.
[4]University of California, U.S.

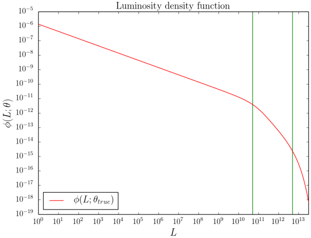
Brightness distribution of galaxies

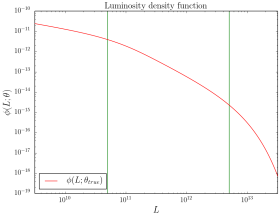
- We can measure: flux, distance
- Use formula: $L = 4\pi r^2 F$
- Luminosity function:

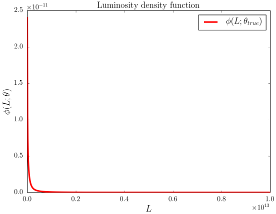
$$\phi(L;\beta,u,l) := \phi_0 \cdot \left(1 - \exp\left(-\frac{L}{l}\right)\right) \cdot \left(\frac{L}{u}\right)^{\beta} \cdot \exp\left(-\frac{L}{u}\right)$$

- β: shape p., l: lower scale p., u: upper scale p., ϕ_0: norm. const.

Luminosity function







Brightness distribution of galaxies

- We can measure: flux, distance
- Use formula: $L = 4\pi r^2 F$
- Luminosity function:

$$\phi(L;\beta,u,l) := \phi_0 \cdot \left(1 - \exp\left(-\frac{L}{l}\right)\right) \cdot \left(\frac{L}{u}\right)^{\beta} \cdot \exp\left(-\frac{L}{u}\right)$$

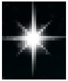
- β: shape p., l: lower scale p., u: upper scale p., ϕ_0: norm. const.
- Main goal: estimate full probability distribution of β, l and u and refine luminosity measurements

Solution

- Method: hierarchical Bayesian model
- Input: data + model
- Output: p(parameters) and $\phi(L;\beta,u,l)$

General formulation

Figure : N objects in a population



- characteristics ($\chi \in X$)
  - *e.g. distance, size, brightness...*
  - *parametrized PDF:* $p(\chi|\theta)$
- population parameters ($\theta \in \Theta$)
- measurements (D)
  - χ *with noise*
- $\boldsymbol{\chi} := \{\chi_i\}_{i=1}^N$, $\boldsymbol{D} := \{D_i\}_{i=1}^N$

Goals

- Inference of distribution of population parameters

Goals

- Inference of distribution of population parameters
- A "classic" method: MLE
  - *Likelihood function:* $\mathcal{L}(\theta|D_1,\ldots,D_N)=\prod_{i=1}^{N}p(D_i|\theta)$
  - *Finding a value of* θ *that maximizes* $\mathcal{L}$

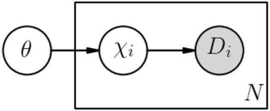
Goals

- Inference of distribution of population parameters
- A "classic" method: MLE
  - *Likelihood function:* $\mathcal{L}(\theta|D_1, \ldots, D_N) = \prod_{i=1}^{N} p(D_i|\theta)$
  - *Finding a value of* θ *that maximizes* $\mathcal{L}$
- However, suppose we have detail info about measurement errors and
- Further goals:
  - *Not only model fitting but iterative improvement for estimation of probability distribution of population parameters and characteristics*
  - *Refinement on measurements*

Goals

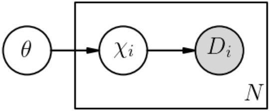
- Inference of distribution of population parameters
- A "classic" method: MLE
  - *Likelihood function:* $\mathcal{L}(\theta|D_1,\ldots,D_N)=\prod_{i=1}^{N} p(D_i|\theta)$
  - *Finding a value of* θ *that maximizes* $\mathcal{L}$
- However, suppose we have detail info about measurement errors and
- Further goals:
  - *Not only model fitting but iterative improvement for estimation of probability distribution of population parameters and characteristics*
  - *Refinement on measurements*
- $\Longrightarrow$ Hierarchical Bayesian approach:
  - *Dealing with not only probabilities but distribution functions*

Hierarchical Bayesian Model



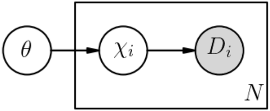
- Drawing samples from joint posterior distribution:
  - $p(\boldsymbol{\chi}, \theta | \boldsymbol{D}) \propto p(\boldsymbol{D}|\boldsymbol{\chi}) \cdot p(\boldsymbol{\chi}|\theta) \cdot p(\theta)$

Hierarchical Bayesian Model



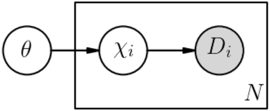
- Drawing samples from joint posterior distribution:
  - $p(\boldsymbol{\chi}, \theta | \boldsymbol{D}) \propto p(\boldsymbol{D}|\boldsymbol{\chi}) \cdot p(\boldsymbol{\chi}|\theta) \cdot p(\theta)$
- By a so-called *Metropolis-within-Gibbs* (MWG) sampling
- This is an MCMC method run over $\Theta \times X^N$ parameter space

Hierarchical Bayesian Model



- Drawing samples from joint posterior distribution:
  - $p(\boldsymbol{\chi}, \theta | \boldsymbol{D}) \propto p(\boldsymbol{D}|\boldsymbol{\chi}) \cdot p(\boldsymbol{\chi}|\theta) \cdot p(\theta)$
- By a so-called *Metropolis-within-Gibbs* (MWG) sampling
- This is an MCMC method run over $\Theta \times X^N$ parameter space
- Alternate estimation θ and $\boldsymbol{\chi}$ (Gibbs)

Hierarchical Bayesian Model

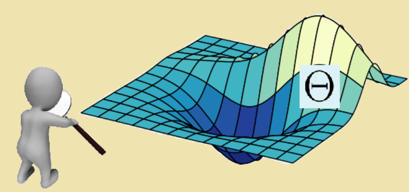


- Drawing samples from joint posterior distribution:
  - $p(\boldsymbol{\chi}, \theta | \boldsymbol{D}) \propto p(\boldsymbol{D} | \boldsymbol{\chi}) \cdot p(\boldsymbol{\chi} | \theta) \cdot p(\theta)$
- By a so-called *Metropolis-within-Gibbs* (MWG) sampling
- This is an MCMC method run over $\Theta \times X^N$ parameter space
- Alternate estimation θ and $\boldsymbol{\chi}$ (Gibbs)
- $N + 1$ Markov chains (Metropolis)

Computational approaches

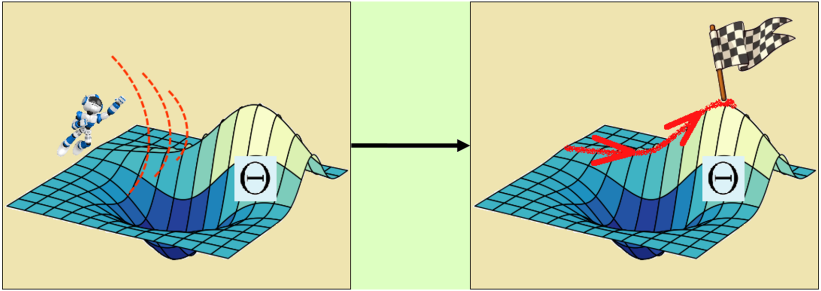
- $\theta \sim p(\theta|\boldsymbol{\chi})$ (CPU)
- $\chi_i \sim p(\chi_i|\theta, D_i)$ for $i = 1, \ldots, N$ (GPU)
- $p(\theta|\boldsymbol{\chi}) \propto \underbrace{p(\theta)}_{\text{prior}} \cdot \underbrace{\prod_{i=1}^{N} p(\chi_i|\theta)}_{\mathcal{L}(\theta|\chi_1,\ldots,\chi_N)}$
- $p(\chi_i|\theta, D_i) \propto \underbrace{p(D_i|\chi_i)}_{\text{error}} \cdot p(\chi_i|\theta)$
- Posterior mean $< \chi_i >$ directly from MCMC

Previous knowledge

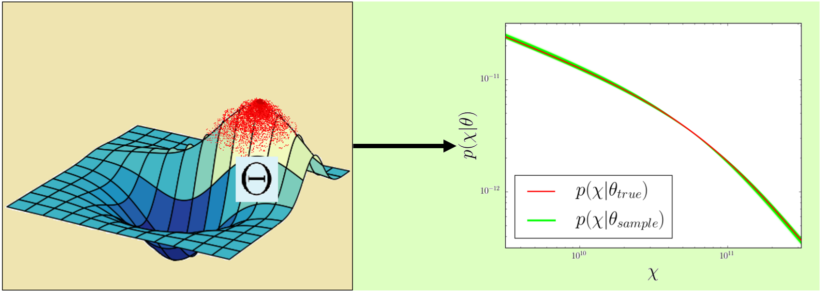


- Prior PDF
- Initial values

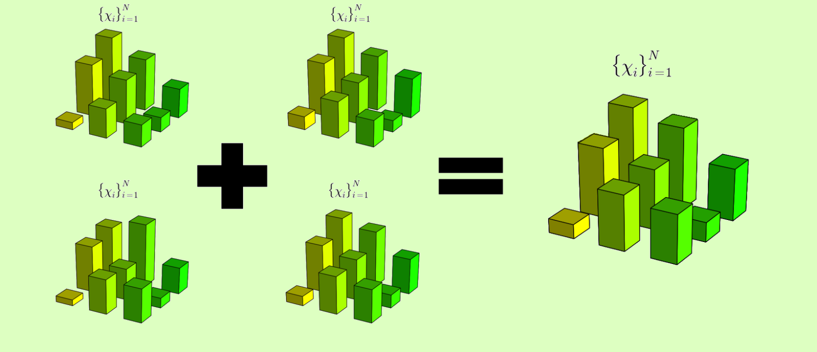
Burn-in period



Sampling result

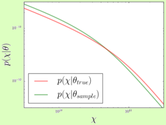


Posterior mean



Sampling

$D \rightarrow \chi_0 \rightarrow p(\theta_k|\chi_k)$

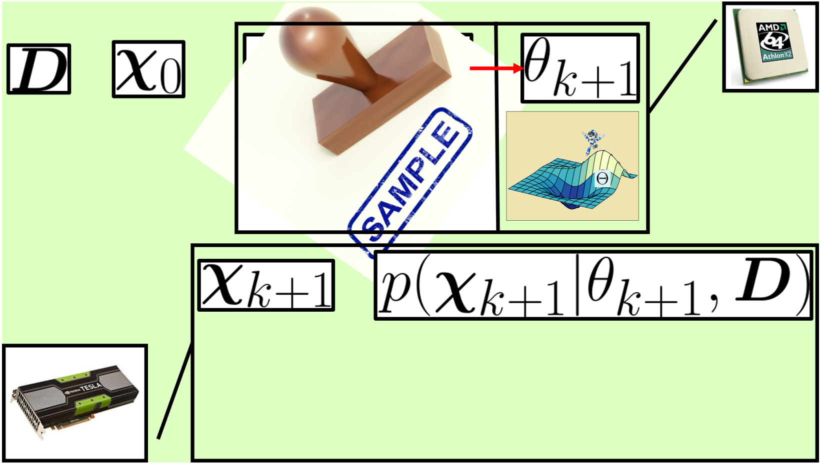


θ_{k+1}

χ_{k+1}

$p(\chi_{k+1}|\theta_{k+1}, D)$

Sampling


D
X_0
SAMPLE
θ_{k+1}
Θ
AMD 64
Athlon X2
X_{k+1}
$p(X_{k+1}|\theta_{k+1}, D)$
TESLA

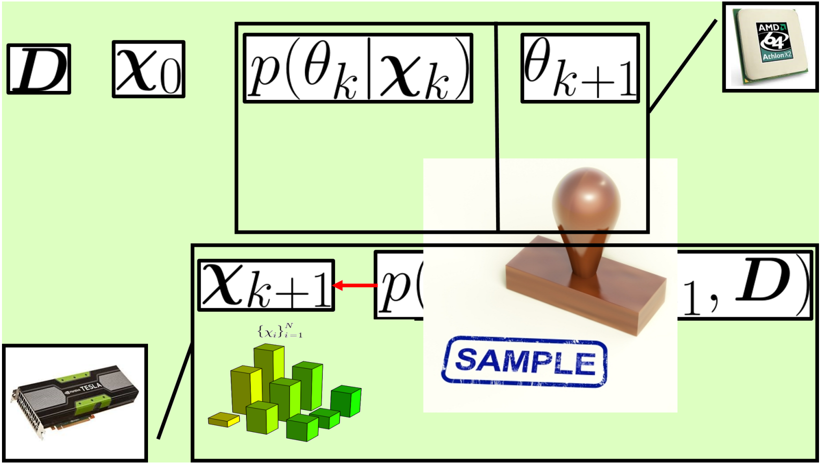
Sampling

$\boldsymbol{D}$ $\quad \boldsymbol{\chi}_0$

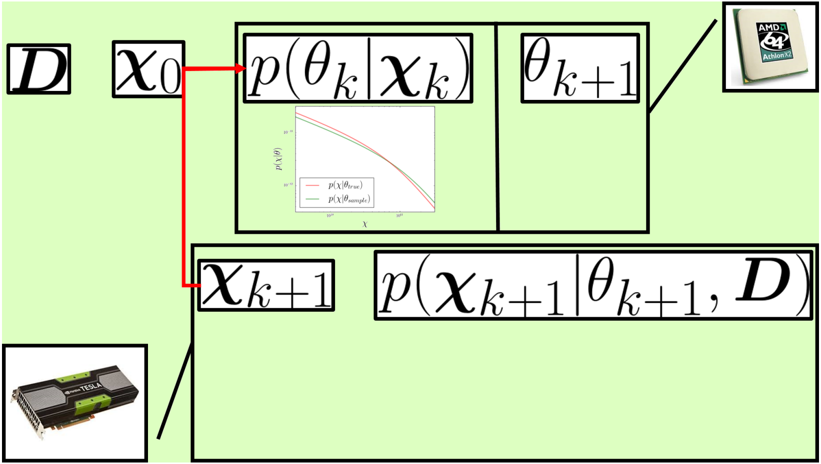
$p(\theta_k | \boldsymbol{\chi}_k)$ $\quad \theta_{k+1}$

$\boldsymbol{\chi}_{k+1}$ $\quad p(\boldsymbol{\chi}_{k+1} | \theta_{k+1}, \boldsymbol{D})$

Sampling


D
X_0
$p(\theta_k | X_k)$
θ_{k+1}
AMD 64
Athlon X2
X_{k+1}
$p($
$1, D)$
$\{x_i\}_{i=1}^N$
SAMPLE
TESLA

Sampling


D
χ_0
$p(\theta_k|\chi_k)$
θ_{k+1}
$p(\chi|\theta_{true})$
$p(\chi|\theta_{sample})$
χ
χ_{k+1}
$p(\chi_{k+1}|\theta_{k+1}, D)$
AMD 64 Athlon X2
TESLA

Tradeoff

$$\text{CPU} + \text{data move} \overset{?}{<} \text{single-threaded GPU kernel}$$

Use case: luminosity function

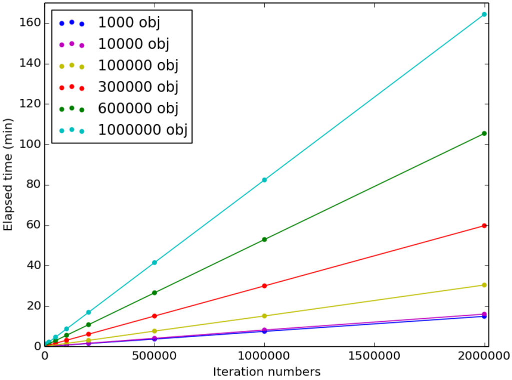
- $\theta = (\beta, l, u)$
- If each object is "visible" $\Longrightarrow$ looks pretty good
- If there is flux limit $\Longrightarrow$ difficulties
- We have implementation for both cases which is written in C/C++ with CUDA and works with simulated data.

A possible difficulty

- Complicated numerical integration, e.g.:
- $\int_0^\infty \int_0^\infty \zeta\left(\frac{L}{4\pi r^2}, T, \sigma_0\right) \cdot \phi(L;\theta) \cdot \delta(r)\,\mathrm{d}L\,\mathrm{d}r$
- $\delta(r)$: distance PDF
- T: flux limit
- $\zeta\left(\frac{L}{4\pi r^2}, T, \sigma_0\right) = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{\chi - T}{\sqrt{2}\sigma(\chi)}\right)\right)$
- erf: error function

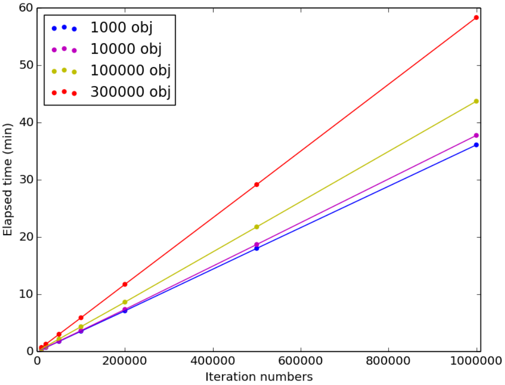
Performance tests I

- NVIDIA Tesla K40c
- Linear scaling
- 1M obj.
  2M iter.
  ca. 2.6 hours
- (For a simple model)



Performance tests II

- For more complex model
- e.g. with time-consuming num. int.



Future works

- Utilization cosmological distance
  - *The expansion of Universe hasn't been taking into account yet*
  - *More complex numerical integration etc.*
- Application on SDSS data set

Summary

- HB method for object characteristics and population-level parameters estimation
- Characteristics computation with MC on GPU cores

Acknowledgements

- The research was supported by the Hungarian OTKA grants 103244 and 114560.
- Images by courtesy of Boians Cho Joo Young, Stuart Miles, TAW4 and xedos4 at FreeDigitalPhotos.net

Thank you for your attention!