Divide-and-Conquer Monte Carlo Fusion
A method of unifying distributed analyses

Ryan Chan [1,3], Murray Pollock [1,2], Adam Johansen [1,3], Gareth Roberts [1,3]

[1] University of Warwick, [2] Newcastle University, [3] The Alan Turing Institute

Generalised Monte Carlo Fusion

Provides theory to carry out perfect inference for the target:

$$\pi(\boldsymbol{x}) \propto f_1(\boldsymbol{x}) \cdots f_C(\boldsymbol{x}) = \prod_{c=1}^{C} f_c(\boldsymbol{x}) \tag{1}$$

Uses *importance sampling* on the extended target distribution:

$$g\left(\boldsymbol{x}^{(1:C)}, \boldsymbol{y}\right) \propto \prod_{c=1}^{C} \left[f_c^2\left(\boldsymbol{x}^{(c)}\right) \cdot p_c\left(\boldsymbol{y} \mid \boldsymbol{x}^{(c)}\right) \cdot \frac{1}{f_c(\boldsymbol{y})} \right] \tag{2}$$

Let $p_c\left(\boldsymbol{y} \mid \boldsymbol{x}^{(c)}\right)$ be a transition density of a stochastic process with stationary distribution $f_c^2(\boldsymbol{x})$, then (2) admits π as a marginal for $\boldsymbol{y}$.

Proposal distribution:

$$h\left(\boldsymbol{x}^{(1:C)}, \boldsymbol{y}\right) \propto \prod_{c=1}^{C} \left[f_c\left(\boldsymbol{x}^{(c)}\right) \right] \cdot \exp\left\{ -\frac{1}{2T} (\boldsymbol{y} - \tilde{\boldsymbol{x}})^{\intercal} \Lambda^{-1} (\boldsymbol{y} - \tilde{\boldsymbol{x}}) \right\} \tag{3}$$

where

$$\tilde{\boldsymbol{x}} = \left(\sum_{c=1}^{C} \Lambda_c^{-1} \right)^{-1} \left(\sum_{c=1}^{C} \Lambda_c^{-1} \boldsymbol{x}^{(c)} \right) \tag{4}$$

$$\Lambda^{-1} = \sum_{c=1}^{C} \Lambda_c^{-1} \tag{5}$$

and Λ_c is the *preconditioning matrix* associated to sub-posterior $f_c(\boldsymbol{x})$ for $c = 1, \dots, C$.

If p_c transition probability of a d-dimensional double Langevin diffusion process, then under certain mild conditions,

$$\frac{g\left(\boldsymbol{x}^{(1:C)}, \boldsymbol{y}\right)}{h\left(\boldsymbol{x}^{(1:C)}, \boldsymbol{y}\right)} \propto \rho_0 \cdot \rho_1 =: w'\left(\boldsymbol{x}^{(1:C)}, \boldsymbol{y}\right) \tag{6}$$

where ρ_0 and ρ_1 are two un-normalised weights.

Algorithm 1 Generalised Monte Carlo Fusion

1. Initialise a value for $T > 0$
2. Simulate a proposal $\boldsymbol{y}$ from h:
   a) For $c = 1, \dots, C$, simulate $\boldsymbol{x}^{(c)} \sim f_c(\boldsymbol{x})$ and calculate $\tilde{\boldsymbol{x}}$
   b) Simulate $\boldsymbol{y} \sim \mathcal{N}_d(\tilde{\boldsymbol{x}}, T\Lambda)$
3. Assign unnormalised weight $w'\left(\boldsymbol{x}^{(1:C)}, \boldsymbol{y}\right)$

Why use a preconditioning matrix?

Let $\pi \propto f_1 f_2$, where $f_c \sim \mathcal{N}_2(\boldsymbol{0}, \Sigma)$ and $\Sigma = \begin{pmatrix} 1.0 & \rho_{\text{corr}} \\ \rho_{\text{corr}} & 1.0 \end{pmatrix}$

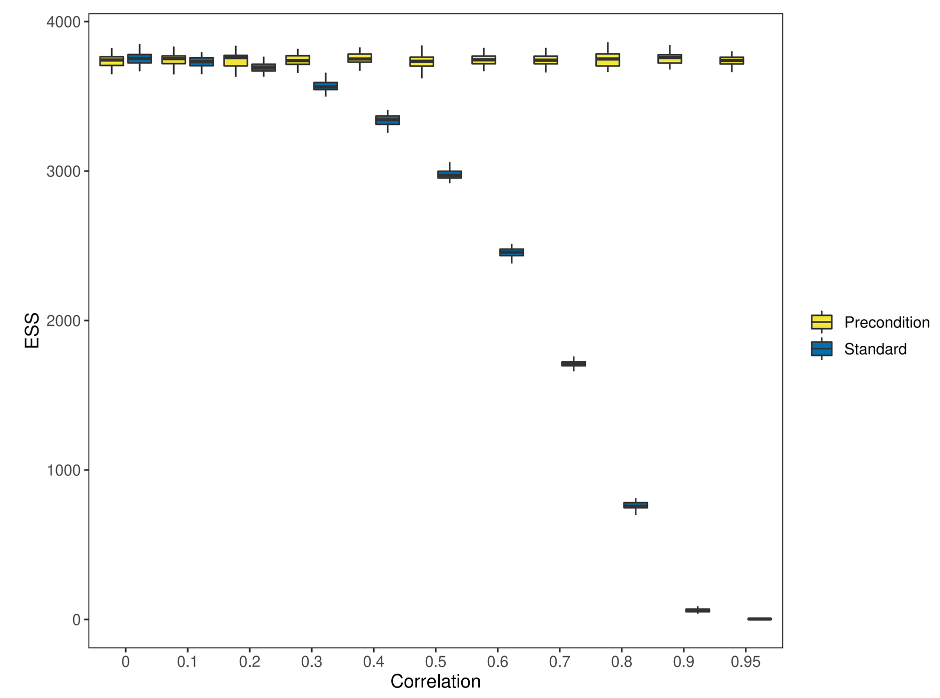

Figure 1: The effect on ESS with varying correlation

Divide-and-Conquer Monte Carlo Fusion

<u>Problem:</u> Fusion becomes inefficient as the number of sub-posteriors increases.

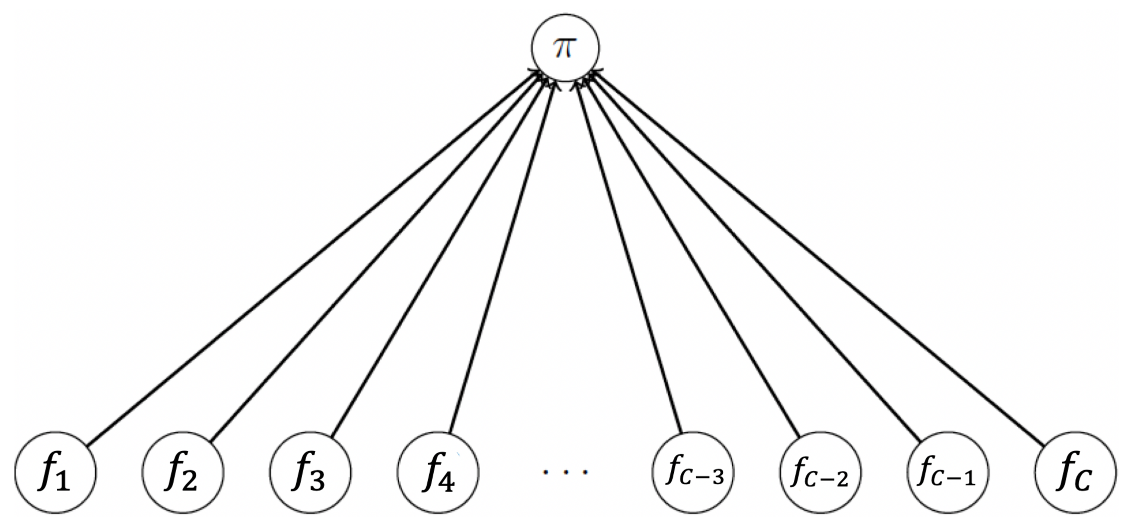

Figure 2: The 'fork-and-join' approach

However, we can adopt a *divide-and-conquer* approach:

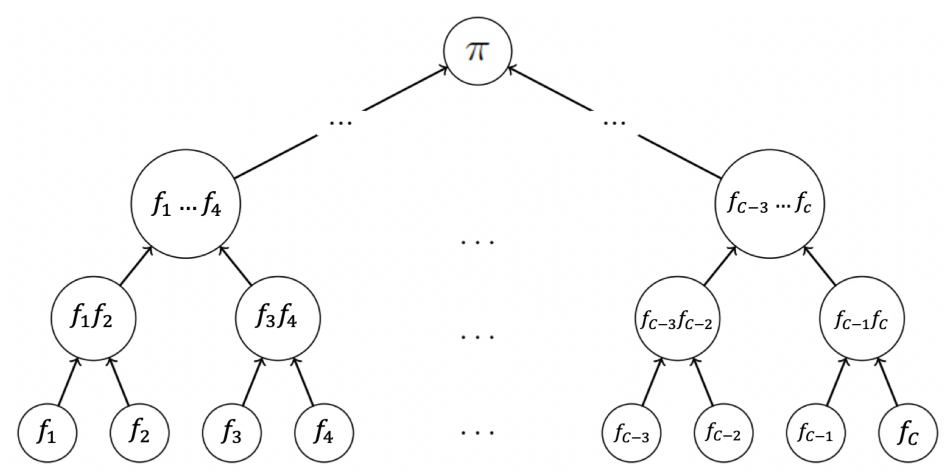

Figure 3: The balanced approach

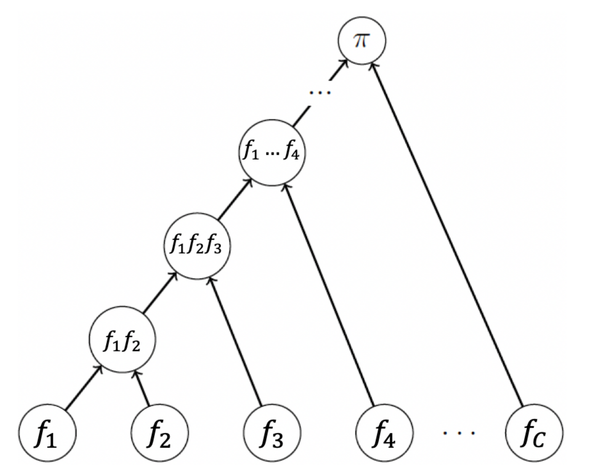

Figure 4: The sequential approach

Toy Example

- Target: $\pi(x) \propto e^{-\frac{x^4}{2}}$
- Sub-posteriors: $f_c(x) \propto e^{-\frac{x^4}{8}}$ for $c = 1, \dots, 4$
- $N = 20,000$

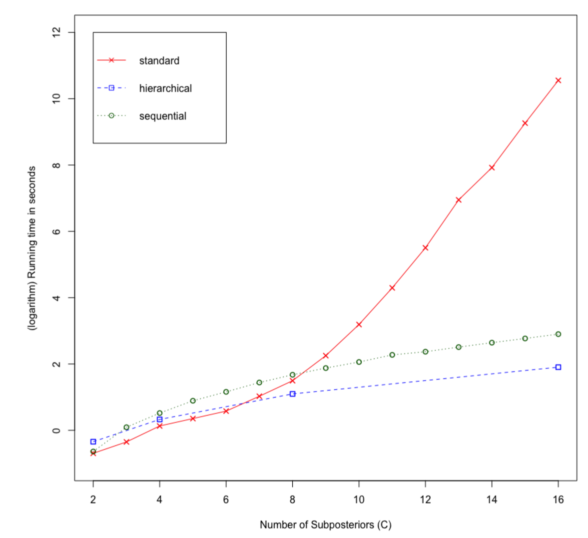

Figure 5: Run-times of MC fusion for $f_c \propto e^{-\frac{x^4}{2C}}$ for $c = 1, \dots, C$ for varying C

Logistic regression example

- Logistic regression model with credit card data to predict default on loans
- $n = 50,000$, $d = 5$
- To compare methods, use integrated absolute distance:

$$IAD = \frac{1}{2d} \sum_{j=1}^{d} \int \left| \hat{f}(\boldsymbol{x}_j) - f(\boldsymbol{x}_j) \right| \mathrm{d}x_j \in [0, 1]$$

where $\hat{f}(\boldsymbol{x}_j)$ is baseline marginal density (obtained using NUTS with Stan)

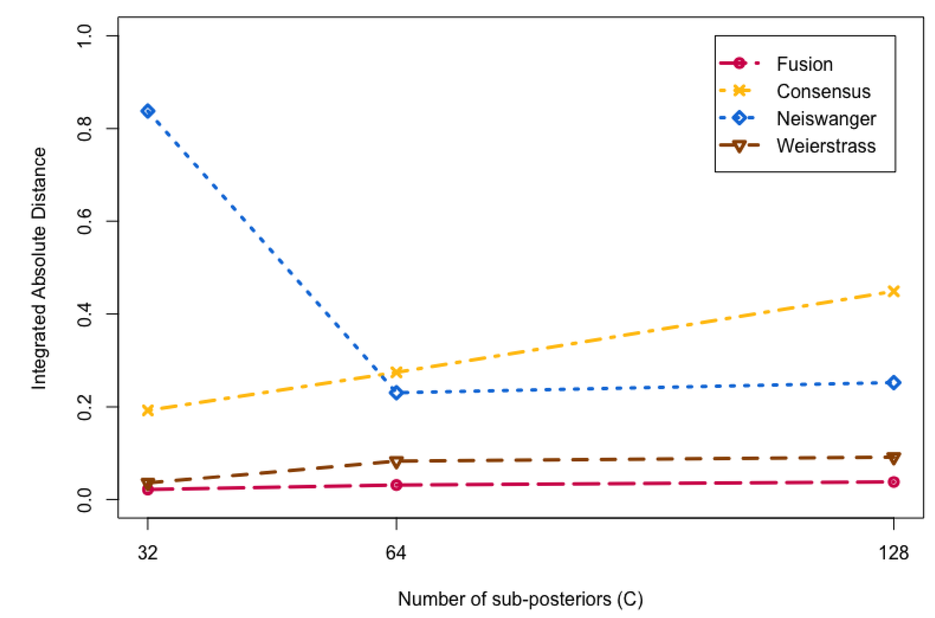

Figure 6: Integrated absolute distance for different number of sub-posteriors for different methods for unifying distributed analyses

Acknowledgements: RC is funded by The Alan Turing Institute Doctoral Studentship, under the EPSRC grant EP/N510129/1.