Topics in Retrospective simulation

Gareth Roberts

Department of Statistics, University of Warwick

FoCSML, January 2025

Part 5: Fusion and Football Draws

Two parts

1. Fusion Intractable likelihood problems where data is separated, either deliberately or because of constraints
2. Football draws How to do public draws for football competitions fairly.

Both involve extensive use of retrospective simulation techniques.

Overview of Fusion part

Background

Monte Carlo Fusion

Bayesian Fusion

Recent directions

Monte Carlo Fusion (Dai et al., 2019)

Bayesian Fusion (Dai et al., 2021)

Divide-and-Conquer Fusion (Chan et al., 2021)

Divide-and-conquer paradigm

Interested in carrying out Bayesian inference based on subsets of the data, each held on a separate core and then *combining* inferences. Motivation:

- Partitioning Big Data to make it manageable
- Inference under privacy constraints.

Divide-and-conquer paradigm

Interested in carrying out Bayesian inference based on subsets of the data, each held on a separate core and then *combining* inferences. Motivation:

- Partitioning Big Data to make it manageable
- Inference under privacy constraints.

For existing methods, the combination process will likely involve some kind of approximation.

Divide-and-conquer paradigm

Interested in carrying out Bayesian inference based on subsets of the data, each held on a separate core and then *combining* inferences. Motivation:

- Partitioning Big Data to make it manageable
- Inference under privacy constraints.

For existing methods, the combination process will likely involve some kind of approximation.

Errors induced by such approximations are often not well-understood.

Bayesian setting

$f(\boldsymbol{x})$ is a posterior of interest.

$$f(\boldsymbol{x}) \propto \prod_{c=1}^{C} f_c(\boldsymbol{x}) \,.$$

Assume we can draw samples from $f_c(\boldsymbol{x})$; but we cannot draw samples from $f(\boldsymbol{x})$.

There is now a substantial literature on this problem. Many useful methods exist, but all involve approximation. Eg Consensus Monte Carlo (Scott et al., 2016) which is exact when each of $f_c(\boldsymbol{x})$ are Gaussian.

Our aims for this work:

1. Provide methodology which is ”exact”.
2. Make this methodology scalable.
3. Show robustness of the methodology for instance to incompatibility of the different f_c densities.
4. Give practical algorithmic guidance for implementing the resuting algorithms (which have a number of user-specified tuning parameters).

An auxiliary variable representation

$$f(\boldsymbol{y}) \propto \prod_{c=1}^{C} f_c(\boldsymbol{y})$$

Consider $d(C+1)$-dimensional distribution

$$\begin{aligned} & g\left(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(C)}, \boldsymbol{y}\right) \\ & \propto \prod_{c=1}^{C}\left[f_c^2\left(\boldsymbol{x}^{(c)}\right) p_c\left(\boldsymbol{y} \,\middle|\, \boldsymbol{x}^{(c)}\right) \cdot \frac{1}{f_c(\boldsymbol{y})}\right] \end{aligned} \tag{1}$$

where $p_c\left(\boldsymbol{y} \,\middle|\, \boldsymbol{x}^{(c)}\right)$ is a transition density of a Markov chain/process with stationary distribution $f_c^2(\boldsymbol{x})$.

An auxiliary variable representation

$$f(\boldsymbol{y}) \propto \prod_{c=1}^{C} f_c(\boldsymbol{y})$$

Consider $d(C+1)$-dimensional distribution

$$\begin{aligned} & g\left(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(C)}, \boldsymbol{y}\right) \\ & \propto \prod_{c=1}^{C} \left[f_c^2\left(\boldsymbol{x}^{(c)}\right) p_c\left(\boldsymbol{y} \,\middle|\, \boldsymbol{x}^{(c)}\right) \cdot \frac{1}{f_c(\boldsymbol{y})} \right] \end{aligned} \tag{1}$$

where $p_c\left(\boldsymbol{y} \,\middle|\, \boldsymbol{x}^{(c)}\right)$ is a transition density of a Markov chain/process with stationary distribution $f_c^2(\boldsymbol{x})$.

Then the marginal distribution of $\boldsymbol{y}$ is f.

Choosing $p_c(y \mid x)$

Many possible choices for $p_c(\boldsymbol{y} \mid \boldsymbol{x})$.

Consider $p_c(\boldsymbol{y} \mid \boldsymbol{x})$ to be the transition density of the following double Langevin diffusion $\boldsymbol{x}_t^{(c)}$ over a pre-defined time $T > 0$:

$$\mathrm{d}\boldsymbol{x}_t^{(c)} = \nabla \log f_c(\boldsymbol{x}_t^{(c)})\,\mathrm{d}t + \mathrm{d}\boldsymbol{W}_t^{(c)},$$

$\boldsymbol{W}_t^{(c)}$: d-dimensional Brownian motion.

Such $\boldsymbol{x}_t^{(c)}$ has invariant distribution $f_c^2(\boldsymbol{x})$. From now on we shall use this choice for $p_c(\boldsymbol{y} \mid \boldsymbol{x})$.

Choosing $p_c(y \mid x)$

Many possible choices for $p_c(\boldsymbol{y} \mid \boldsymbol{x})$.

Consider $p_c(\boldsymbol{y} \mid \boldsymbol{x})$ to be the transition density of the following double Langevin diffusion $\boldsymbol{x}_t^{(c)}$ over a pre-defined time $T > 0$:

$$\mathrm{d}\boldsymbol{x}_t^{(c)} = \nabla \log f_c(\boldsymbol{x}_t^{(c)})\,\mathrm{d}t + \mathrm{d}\boldsymbol{W}_t^{(c)},$$

$\boldsymbol{W}_t^{(c)}$: d-dimensional Brownian motion.

Such $\boldsymbol{x}_t^{(c)}$ has invariant distribution $f_c^2(\boldsymbol{x})$. From now on we shall use this choice for $p_c(\boldsymbol{y} \mid \boldsymbol{x})$.

But $p_c(\boldsymbol{y} \mid \boldsymbol{x})$ is typically not tractable in this case.

Monte Carlo fusion by rejection sampling

To simulate from the extended pdf

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

use proposal density proportional to

$$\begin{aligned} & h(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(C)}, \boldsymbol{y}) \\ & = \prod_{c=1}^{C} \left[f_c\left(\boldsymbol{x}^{(c)}\right) \right] \cdot \exp\left(-\frac{C \cdot \|\boldsymbol{y} - \bar{\boldsymbol{x}}\|^2}{2T} \right) \end{aligned}$$

for some user-specified $T > 0$.

Simulating the proposal h

$$h(\ldots) = \prod_{c=1}^{C} f_c\left(\boldsymbol{x}^{(c)}\right) \cdot \exp\left(-\frac{C \cdot \|\boldsymbol{y} - \bar{\boldsymbol{x}}\|^2}{2T}\right)$$

This choice of h can be easily simulated.

- For $c = 1 \ldots C$, draw $\boldsymbol{x}_c$ from f_c.
- Compute $\bar{\boldsymbol{x}}$
- Simulate $\boldsymbol{y} \sim N(\bar{\boldsymbol{x}}, T\mathbf{I}/C)$

(For some others proposal choices of h see papers.)

Rejection sampling – acceptance probability

Simulation from h: $\boldsymbol{x}_c$ is drawn from $f_c(\cdot)$, independently; then $\boldsymbol{y}$ from $\mathcal{N}(\bar{\boldsymbol{x}}, T\mathbf{I}/C)$.

Rejection sampling ratio:

$$\frac{g(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(C)}, \boldsymbol{y})}{h(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(C)}, \boldsymbol{y})} \propto \quad \underbrace{\rho \times Q}_{\text{acceptance prob}}$$

$$\rho := \rho(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(C)}) = e^{-\frac{C\sigma^2}{2T}},$$

$$\sigma^2 = C^{-1} \sum_{c=1}^{C} \|\boldsymbol{x}^{(c)} - \bar{\boldsymbol{x}}\|^2,$$

Rejection sampling– acceptance probability

$$Q = \mathbf{E}\left(\prod_{c=1}^{C} E_c\right) = \prod_{c=1}^{C} \mathbf{E}\left(E_c\right),$$

with

$$E_c := \exp\left\{-\int_0^T \left(\phi_c\left(\boldsymbol{x}_t^{(c)}\right) - \Phi_c\right) \mathrm{d}t\right\}.$$

where $\boldsymbol{x}_t^{(1)}, \cdots, \boldsymbol{x}_t^{(C)}$ are independent Brownian bridges tied down at $\boldsymbol{x}_0^{(c)} = \boldsymbol{x}^{(c)}$ and $\boldsymbol{x}_T^{(c)} = \boldsymbol{y}$.

$$\phi_c(\boldsymbol{x}) = \frac{\Delta f_c(\boldsymbol{x})}{f_c(\boldsymbol{x})}$$

Φ_c are constants chosen so that integrand is non-negative.

Rejection sampling– acceptance probability

$$Q = \mathbf{E}\left(\prod_{c=1}^{C} E_c\right) = \prod_{c=1}^{C} \mathbf{E}\left(E_c\right),$$

with

$$E_c := \exp\left\{-\int_0^T \left(\phi_c\left(\boldsymbol{x}_t^{(c)}\right) - \Phi_c\right)\,\mathrm{d}t\right\}.$$

where $\boldsymbol{x}_t^{(1)}, \cdots, \boldsymbol{x}_t^{(C)}$ are independent Brownian bridges tied down at $\boldsymbol{x}_0^{(c)} = \boldsymbol{x}^{(c)}$ and $\boldsymbol{x}_T^{(c)} = \boldsymbol{y}$.

$$\phi_c(\boldsymbol{x}) = \frac{\Delta f_c(\boldsymbol{x})}{f_c(\boldsymbol{x})}$$

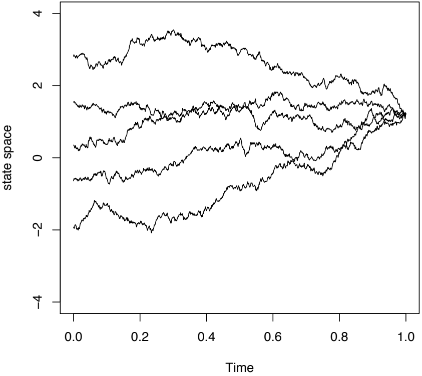
Φ_c are constants chosen so that integrand is non-negative.

So need to perform C independent tests, one for each Brownian bridge.

The Q rejection step

Given the starting and common end point, we simulate independent Brownian bridges as below.

We then assign to each bridge an acceptance probability given by E_c.



Brownian bridge rejection test

Need to simulate from event of probability

$$\exp\left\{-\int_0^T \left(\phi_c\left(\boldsymbol{x}_t^{(c)}\right) - \Phi_c\right)\,\mathrm{d}t\right\}\;.$$

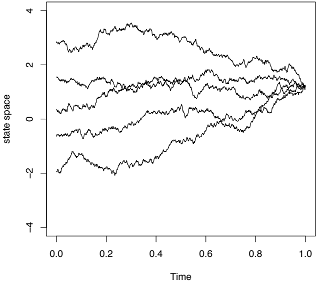
Use path space rejection sampling (Beskos et al., 2008, 2006; Beskos and Roberts, 2005).

This methodology is exact and can be carried out efficiently as long as the dimensionality d is not too large.

Can be efficient even for large d so long as T is sufficiently small (typically $O(d^{-1})$).

Summary of the algorithm

Once we have starting points and common endpoints, we propose C the Brownian bridges; acceptance probability $\rho \times Q$



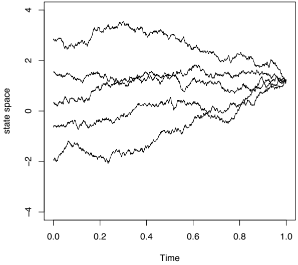
ρ looks only at the c initial points and penalises over-discrepant starting values.

Q penalises all c trajectories according to state-dependent hazard rates along the trajectory.

The Monte Carlo fusion method

A simple rejection sampler implemented as follows.

1. Draw starting values $\boldsymbol{x}_c \sim f_c$, $1 \leq c \leq C$.
2. Carry out ρ rejection step
3. Draw common terminal value $\boldsymbol{y}$.
4. Construct the C Brownian bridges
5. Carry out Q rejection step.



Illustrative example

Consider target density $\pi(u) \propto u^4(1-u)$, $u \in [0,1]$, i.e. Beta$(5,2)$.

Transform to $\mathbb{R}$: $x = \log(u/(1-u))$

$$f(x) \quad \propto \quad \left[\frac{\exp(x)}{1+\exp(x)}\right]^5 \left[\frac{1}{1+\exp(x)}\right]^2 .$$

Illustrative example

Consider target density $\pi(u) \propto u^4(1-u)$, $u \in [0,1]$, i.e. Beta$(5,2)$.

Transform to $\mathbb{R}$: $x = \log(u/(1-u))$

$$f(x) \quad \propto \quad \left[\frac{\exp(x)}{1+\exp(x)}\right]^5 \left[\frac{1}{1+\exp(x)}\right]^2 .$$

Decompose $f(x)$ into $C = 5$ components:

$$\begin{aligned} f(x) \quad &\propto \quad f_1(x)\cdots f_C(x) \\ f_c(x) \quad &= \quad \left[\frac{\exp(x)}{1+\exp(x)}\right] \left[\frac{1}{1+\exp(x)}\right]^{0.4} . \end{aligned} \tag{2}$$

Simulation results

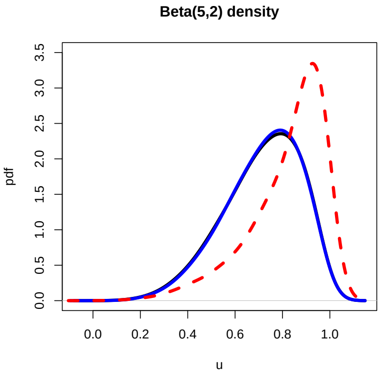


Figure: Kernel density fitting with bandwidth 0.04 for Beta$(5, 2)$: [1.]– black solid curve, standard exact MC, r command; [2.]– blue solid curve, Rejection sampling; [3.]– red dashed curve, CMC algorithm.

Limitations of the above rejection sampling algorithm

1. Scalability: the acceptance probability of Monte Carlo fusion can be small, especially when C is large and or d is large.
2. Small T: T has to be reasonably large to make ρ to be relatively large. In fact typically needs to be $O(d)$.
3. Large T. However, Q might typically decrease exponentially as T increases.

Limitations of the above rejection sampling algorithm

1. Scalability: the acceptance probability of Monte Carlo fusion can be small, especially when C is large and or d is large.
2. Small T: T has to be reasonably large to make ρ to be relatively large. In fact typically needs to be $O(d)$.
3. Large T. However, Q might typically decrease exponentially as T increases.

So method only applicable for moderate-dimensional problems.

This might be fine for many applications (eg in privacy) but we want a more generally applicable method.

Aim to construct a more scalable version of this using sequential importance sampling.

Monte Carlo fusion: main limiting factor

$$Q = \prod_{c \in C} \exp\left\{ -\int_0^T \left(\phi_c\left(\boldsymbol{x}_t^{(c)} \right) - \Phi_c \right) \mathrm{d}t \right\} .$$

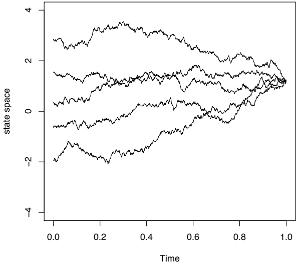
MCF algorithm:

1. Draw starting values $\boldsymbol{x}_c \sim f_c$, $1 \leq c \leq C$.
2. Carry out ρ rejection step
3. Draw common terminal value $\boldsymbol{y}$.
4. Construct the C Brownian bridges
5. Carry out Q rejection step.

The blue step is generally the limiting factor.

Can we produce a scalable sequential version of MCF?

1. Draw starting values $\boldsymbol{x}_c \sim f_c$, $1 \leq c \leq C$.
2. Carry out ρ rejection step
3. Draw common terminal value $\boldsymbol{y}$.
4. Construct the C Brownian bridges.
5. Carry out Q rejection step.



Not sequential!

A closer look at the proposal: Fusion Measure

Fusion measure:

1. $\boldsymbol{x}_c \sim f_c$, $1 \leq c \leq C$
2. Set $\boldsymbol{y} \sim MVN(\bar{\boldsymbol{x}}, \mathbf{I}_{d\times d}/T)$
3. Fill in the rest with Brownian bridges.

Conditionally Gaussian.

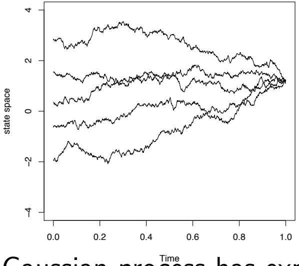
Turns out to have a tractable SDE representation.

Sequential interacting Brownian motions

We can combined steps [3] and [4] to give the sequential and equivalent formulation:

$$d\mathbf{X}_t^{(c)} = \frac{\bar{\mathbf{X}}_t - \mathbf{X}_t^{(c)}}{T - t}\,\mathrm{d}t + \mathrm{d}\mathbf{B}_t^{(c)}$$

where $\bar{\mathbf{X}}_t$ is the mean of the C processes at time t.



This Cd-dimensional Gaussian process has explicit finite-dimensional distributions:

The fusion measure

For $s < t$

$$X_t \sim MVN\left(M(s,t), V(s,t)\right),$$

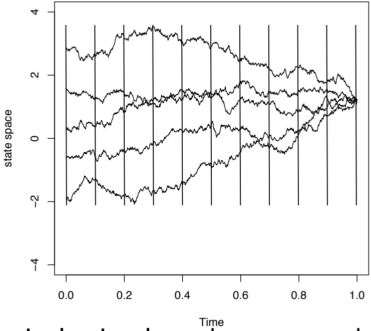
where $M(s,t) = (M_1(s,t), \dots M_C(s,t))$ with

$$M_c(s,t) = \frac{T-t}{T-s} X_{s,c} + \frac{t-s}{T-s} \bar{X}_s,$$

and where $V_{s,t} = \Sigma \otimes \mathbf{I}_{d \times d}$ with $\Sigma = (\Sigma_{ij})$ being a $C \times C$ matrix given by

$$\Sigma_{ii} = \frac{(t-s) \cdot (T-t)}{T-s} + \frac{(t-s)^2}{C(T-s)}, \qquad \Sigma_{ij} = \frac{(t-s)^2}{C(T-s)}.$$

Partition over time



..... so we can iteratively simulate the process through a discrete set of skeleton time points.

Now set up complete to use a sequential approach

Iterative treatment of the Q rejection step

Recall, in Monte Carlo fusion each strand of the coalescing Brownian motions is ultimately accepted with probability

$$E_c = \exp\left\{ -\int_0^T \left(\phi_c(\mathbf{X}_t^{(c)}) - \Phi_c \right) \mathrm{d}t \right\} .$$

Decompose according to our time partition:

$$E_c = \prod_{i=1}^{n} \exp\left\{ -\int_{t_{i-1}}^{t_i} \left(\phi_c(\mathbf{X}_t^{(c)}) - \Phi_c \right) \mathrm{d}t \right\} .$$

Iterative treatment of the Q rejection step

Recall, in Monte Carlo fusion each strand of the coalescing Brownian motions is ultimately accepted with probability

$$E_c = \exp\left\{ - \int_0^T \left(\phi_c(\mathbf{X}_t^{(c)}) - \Phi_c \right) \mathrm{d}t \right\} .$$

Decompose according to our time partition:

$$E_c = \prod_{i=1}^{n} \exp\left\{ - \int_{t_{i-1}}^{t_i} \left(\phi_c(\mathbf{X}_t^{(c)}) - \Phi_c \right) \mathrm{d}t \right\} .$$

This paves the way for the use of a Sequential Monte Carlo (SMC) approach.

Computational problem

For SMC to work we will need to evaluate weights:

$$\exp\left\{-\int_{t_{j-1}}^{t_j} \xi_c(\mathbf{X}_t^{(c)})\,\mathrm{d}t\right\} .$$

but this is not tractable. But we can easily construct unbiased non-negative estimators (Fearnhead, Papaspiliopoulos and R, 2008, 2010). Suppose (for now) that $0 \le \xi_c(\cdot) \le M$ (at least on (t_{j-1}, t_j)), then for any $\lambda > 0$

$$e^{M(t_{j-1}-t_j)} \frac{\prod_{k=1}^{\kappa}(M - U_k)}{\lambda^{\kappa}}$$

where κ is a Poisson(λ) variable is an unbiased estimator and U_k are IID from $U(0, M)$.

More general (often better) alternatives to the Poisson(λ) can be used generalised Poisson estimator.

Localisation

In our case $\phi(x) = 2\|\nabla \log \pi(x)\|^2 + \Delta \log \pi(x)$.

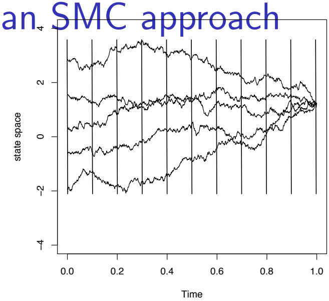
We use the layered Brownian motion framework initially developed in (Beskos et al., 2008), and now being developed into a Black box R package *Brownian motion* by Aslett and Pollock.

The approach first simulate a layer, eg $\sup\lfloor |X_s| \rfloor$ for s in the interval in question.

Then compute the localised bound $M(L)$, ie the bound conditional on the layer.

This paves the way for a random weight particle filter (Chopin 2002). Method works best if variance of weight estimator is as small as possible: eg choose λ in the above (Chopin 2004).

Bayesian Fusion – an SMC approach



Use a population of C-tuples, each one passing the initial proximity test using the rejection sampler ρ. Then iterate

1. Transition from t_{j-1} to t_j according to appropriate normal increment with weighting giving unbiased estimator of

$$\exp\left\{-\int_{t_{j-1}}^{t_j}\left(\phi_c(\mathbf{X}_t^{(c)})-\Phi_c\right)\,\mathrm{d}t\right\}\ .$$

2. Resample as appropriate

This is a random weight SMC algorithm (Chopin, 2002)

Choosing algorithm parameters

1. Choice of partition. Currently use equally spaced time points. Makes sense to ensure that the variance of the weight increments are comparable between intervals.
2. Choice of T. Unlike Monte Carlo fusion, no longer need T reasonably small to ensure practical acceptance probability.
3. In fact can make T quite large ensuring the algorithm is robust to the f_i densities being well separated. This regains polynomial complexity in d (at least in the toy examples where we can do calculations).

Detailed guidance on parameter values in paper. For example:

$$T = O\left(\frac{C^{3/2} v_{between}}{d}\right), \qquad \Delta_j = O\left(\frac{v_{within}}{C^{1/2} d^{1/2}}\right)$$

Running time comparison - discrepant densities

$$C = 2,\ f = \mathcal{N}(0, 2),\ f_1 = \mathcal{N}(-\mu, 4),\ f_2 = \mathcal{N}(\mu, 4)$$

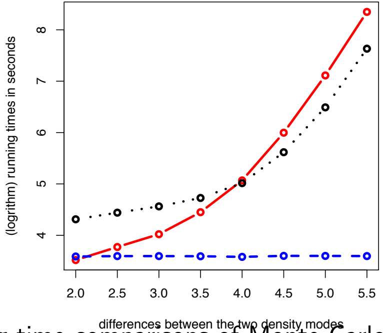


Figure: Log-running-time comparisons of Monte Carlo Fusion (red solid line), Bayesian Exact Fusion (black dotted line) and Bayesian sequential Fusion algorithms (blue dashed line). Results based on two computational cores, but sub-densities having different mode discrepancy.

Red = Monte Carlo Fusion I Black = Monte Carlo Fusion II Blue = Bayesian fusion

Running time comparison - cores

$f = \mathcal{N}(0, 0.5),\ f_c = \mathcal{N}(0, 0.5C)$

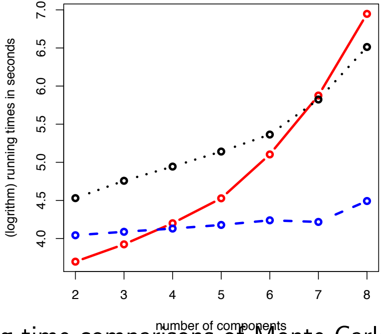


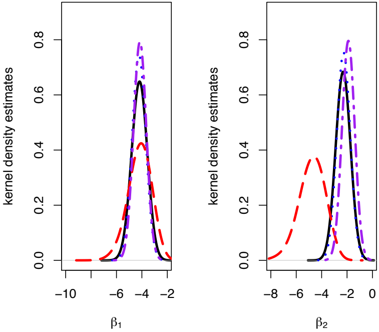
Figure: Log-running-time comparisons of Monte Carlo Fusion (red solid line), Bayesian Exact Fusion (black dotted line) and Bayesian sequential Fusion algorithms (blue dashed line). Results based on sub-densities with the same mode, but different number of computational cores.

Red = Monte Carlo Fusion I

Black = Monte Carlo Fusion II Blue = Bayesian fusion

Posterior for logistic regression model

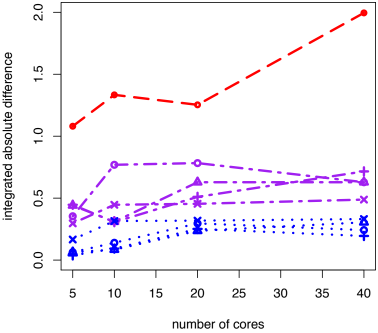
We choose sample size 1000 and $C = 40$, $T = 0.2$ and $n = 20$.



Blue = Bayesian fusion
Purple = WRS (Wang and Dunson, 2013)
Red = CMC (Scott et al., 2016)

Posterior for logistic regression model

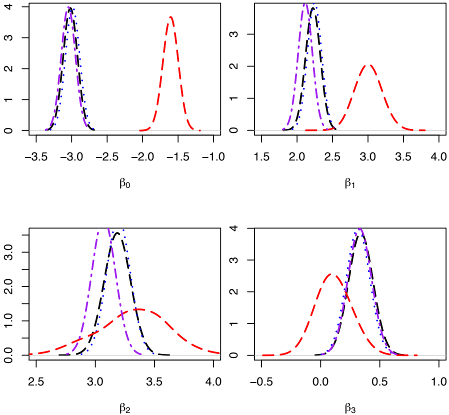
Sample size 1000 and $C = 40$ and $n = 20$, but with different $T = 0.2, 0.15, 0.1, 0.05$.



Blue = Bayesian fusion
Purple = WRS
Red = CMC

Real data example

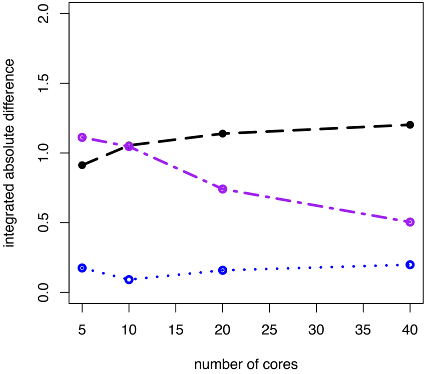
US Census Bureau data sets 1994-95. $C = 40$.



Black = truth; Blue = Bayesian fusion

Purple = WRS; Red = CMC

Error analysis

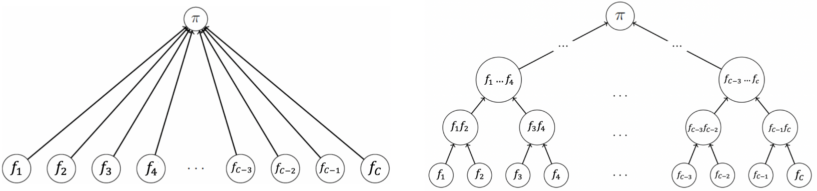


Black = truth; Blue = Bayesian fusion
Purple = WRS; Red = CMC

Divide and conquer strategies

Ideas in Ryan Chan's recent PhD thesis.

Another substantial advantage of an SMC framework is that we can use Divide-and-Conquer SMC strategies for combination of the subposteriors (Lindsten et al 2017, Kuntz et al 2021).



Left hand side is traditional BF, right hand side is D&C BF.

Generalised Bayesian Fusion

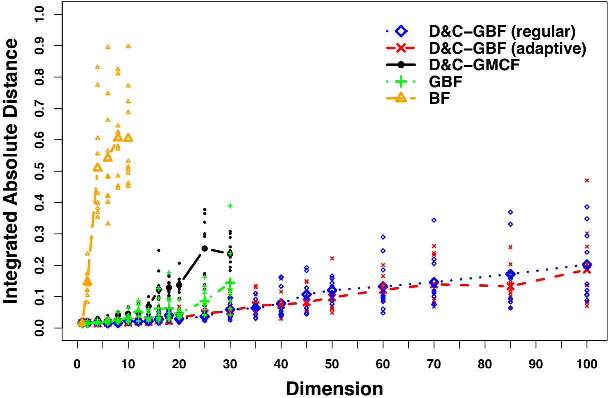
Carry out a preliminary affine transformation of the parameter space. Can lead to substantial improvements (also part of Ryan Chan's PhD thesis).

$$\mathrm{d}\boldsymbol{x}_t^{(c)} = \Lambda \nabla \log f_c(\boldsymbol{x}_t^{(c)})\,\mathrm{d}t + \Lambda^{1/2}\,\mathrm{d}\boldsymbol{W}_t^{(c)},$$

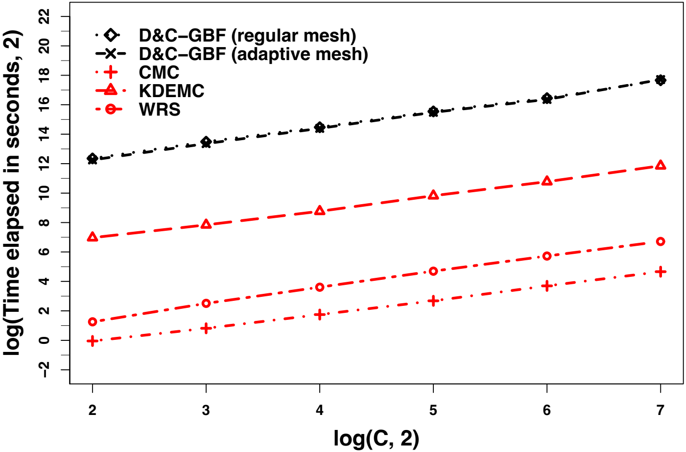
Generalised Bayesian Fusion

Carry out a preliminary affine transformation of the parameter space. Can lead to substantial improvements (also part of Ryan Chan's PhD thesis).

$$\mathrm{d}\boldsymbol{x}_t^{(c)} = \Lambda \nabla \log f_c(\boldsymbol{x}_t^{(c)})\,\mathrm{d}t + \Lambda^{1/2}\,\mathrm{d}\boldsymbol{W}_t^{(c)},$$



A running time comparison



heterogenous information

(Work also with Ryan Chan)

What if one core (core 1 say) had lots more data than the others?

Then given $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C$, we'd expect a draw from π to be much closer to $\boldsymbol{x}_1$ than the other $\boldsymbol{x}$s.

But BF, MCF in their basic form treat all the $\boldsymbol{x}$s identically. Proposal is less efficient.

heterogenous information

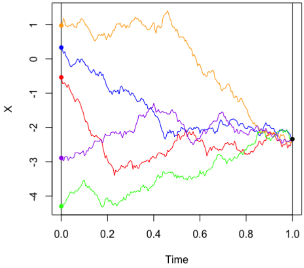
(Work also with Ryan Chan)

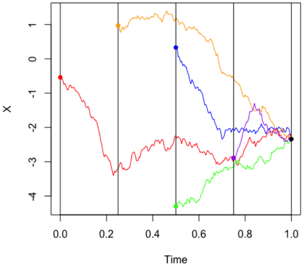
What if one core (core 1 say) had lots more data than the others?

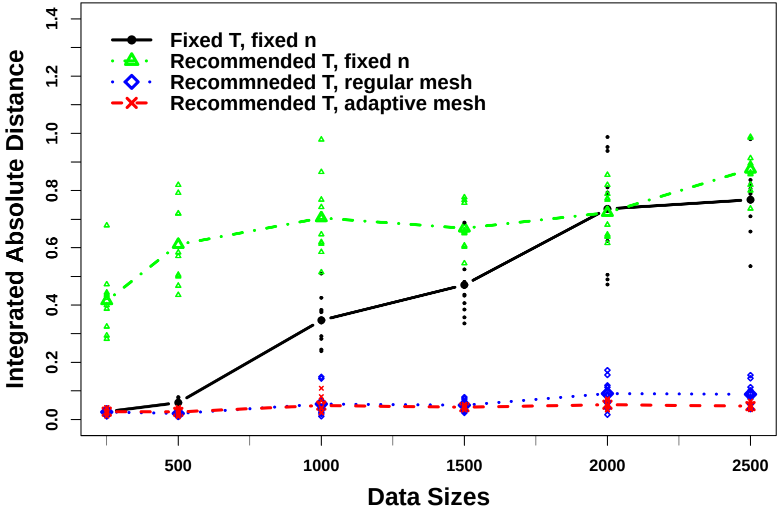
Then given $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C$, we'd expect a draw from π to be much closer to $\boldsymbol{x}_1$ than the other $\boldsymbol{x}$s.

But BF, MCF in their basic form treat all the $\boldsymbol{x}$s identically. Proposal is less efficient.

We can modify the approach to pick a different T_c for different cores. Eg $T_c \propto N_c^{-1}$.






Fixed T, fixed n
Recommended T, fixed n
Recommneded T, regular mesh
Recommended T, adaptive mesh
Integrated Absolute Distance
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
500
1000
1500
2000
2500
Data Sizes

Conclusions

- Bayesian fusion provides a scalable consistent methodology, robust to different distributions.
- It is not quite exact - uses SMC, but consistency follows from standard SMC considerations.
- Other inexpensive principled approximations are readily suggested by simplifying the complete Bayesian fusion algorithm.
- Some formal theory for scalability is being developed, with associated practical guidance on use.
- Just beginning to explore privacy applications (joint with Louis Aslett also).

References

Beskos, A., Papaspiliopoulos, O., and Roberts, G. (2008). A factorisation of diffusion measure and finite sample path constructions. *Methodology and Computing in Applied Probability*, 10:85–104.

Beskos, A., Papaspiliopoulos, O., Roberts, G., and Fearnhead, P. (2006). Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processess, with discussion. *Journal of Royal Statistical Society, B*, 68:333–382.

Beskos, A. and Roberts, G. (2005). An exact simulation of diffusions. *Annals of Applied Probability*, 15(4):2422–2444.

Chan, R. S. Y., Pollock, M., Johansen, A. M., and Roberts, G. O. (2021). Divide-and-conquer monte carlo fusion. *arXiv:2110.07265*.

Dai, H., Pollock, M., and Roberts, G. (2019). Monte Carlo fusion. *Journal of Applied Probability*, 56(1):174–191.

Dai, H., Pollock, M., and Roberts, G. (2021). Bayesian fusion. *arXiv:2102.02123*.

Scott, S., Blocker, A., and Bonassi, F. (2016). Bayes and big data: The consensus monte carlo algorithm. *International Journal of Management Science and Engineering Management*.

Wang, X. and Dunson, D. B. (2013). Parallelizing mcmc via weierstrass sampler. *arXiv preprint arXiv:1312.4605*.

Outline of Football Draws part of talk

1. Consider the practical problem of implementing a public draw unbiasedly.
2. Note the clear biases of currently used methods.
3. suggest some fixes using Retrospective Simulation.

2022 FIFA world cup draw

The 2022 FIFA World Cup took place in November/December 2022, in Qatar. The actual draw took place on 1st April 2022 using a procedure essentially identical to that of 2018.

Aims of a public draw of this type:

1. Advertising for the main event
2. Entertainment
3. Geographical diversity
4. Balanced
5. Transparency
6. Fairness

32 national teams to be partitioned into 8 groups of 4 teams each, who would all play each other in the *group stages* of the competition. Each group consists of one team from each pot:

Pot 1: Qatar[**As**], Belgium[**Eu**], Brazil[**SA**], France[**Eu**], Argentina[**SA**], England[**Eu**], Portugal[**Eu**], Spain[**Eu**].

Pot 2: Denmark[**Eu**], Netherlands[**Eu**], Germany[**Eu**], Switzerland[**Eu**], Croatia[**Eu**], Mexico[**NA**], USA[**NA**], Uruguay[**SA**].

Pot 3: Iran[**As**], Serbia[**Eu**], Japan[**As**], Senegal[**Af**], Tunisia[**Af**], Poland[**Eu**], KoreaRep[**As**], Morocco[**Af**].

Pot 4: Wales/Scot/Ukr[**Eu**], Peru/UAE/Au[**SA**,**As**], CostaRica/NZ[**NA**,**Oc**], SaudiArabia[**As**], Cameroon[**Af**], Ecuador[**SA**], Canada[**NA**], Ghana[**Af**].

Geographical constraints

1. Each group needed to include one team from each of the 4 pots.
2. Each group contains either 1 or 2 teams from **Eu** .
3. Each group contains either 0 or 1 teams from each of the other regions.

Procedure:

1. Qatar was automatically placed in Group A.
2. Then, the remaining teams from Pot 1 were selected one at a time, uniformly at random, and placed into the next group from B through H.
3. Then, the teams from Pot 2 were selected one at a time, uniformly at random, and assigned to the next feasible group.
4. Repeat for Pot 3 then Pot 4.

Biases in the FIFA draw mechanism

We define a fair draw to be one which samples uniformly among all draws which respect the constraints.

Is the FIFA mechanism fair? Or approximately fair?

For 2022 draw, for football reasons there was a particular interest in the probability that a team might end up in the *Qatar group*.

Illustrated bias: FIFA draw gives

$$\mathbf{P}(\text{Mexico gets QATAR}) = 0.125$$

whereas the *fair* probability gives

$$\mathbf{P}(\text{Mexico gets QATAR}) = 0.09\ .$$

Some might argue that this therefore unfairly advantages Mexico.

Rejection sampling

The most obvious way to simulate from the uniform distribution over all feasible draws, is to carry out rejection sampling from a proposal distribution which is uniform over all possible draws *without* the geographical constraints.

For the 2022 World Cup draw this leads to an acceptance probability of around 1 in 540.

This is clearly inefficient but fast enough to carry out 1000 draws in a fraction of a second!

However this solution lacks transparency and excitement!

Various solutions in literature involving drawing from the continental structure of the draw, allowing random allocation of teams within a continent to be carried out *live*.

But can we have a truly sequential and practical solution?

An alternative

A different approach first identifies feasible teams at each step, and then chooses uniformly from those options:

This is the approach recently adopted by UEFA: UEFA CL Last 8 2021-22

An alternative

A different approach first identifies feasible teams at each step, and then chooses uniformly from those options:

This is the approach recently adopted by UEFA: UEFA CL Last 8 2021-22

Different from FIFA *first available group* strategy. Is it unbiased?

An alternative

A different approach first identifies feasible teams at each step, and then chooses uniformly from those options:

This is the approach recently adopted by UEFA: UEFA CL Last 8 2021-22

Different from FIFA *first available group* strategy. Is it unbiased? No!

A motivating example

Simplified setting. Suppose there are two seeded pots:

Pot 1: Qatar (Q) [**Af**], France (F) [**Eu**], Brazil (B) [**SA**]
Pot 2: Mexico (M) [**NA**], Switzerland (S) [**Eu**], Uruguay (U) [**SA**]

Qatar, as hosts, are automatically assigned to Group A. Wlog France is placed in Group B, and Brazil in Group C.

The two **SA** teams, Brazil in Pot 1 and Uruguay in Pot 2, must be kept apart, so that Uruguay cannot be placed in Group C.

Possible groups:
D_1: QM, FU, and BS.
D_2: QS, FU, and BM.
D_3: QU, FM, and BS.
D_4: QU, FS, and BM.

Let $\mathbf{P}$ be the probability measure resulting from FIFA's draw.

Let $\mathbf{U}$ be the probability measure which assigns equal probability to each valid draw: $\mathbf{U}(D_i) = 1/4, \quad i = 1, 2, 3, 4.$

However $\mathbf{P}(D_1) = \mathbf{P}(D_2) = 1/3$ while $\mathbf{P}(D_3) = \mathbf{P}(D_4) = 1/6.$

In particular, if QU is the event that Qatar is paired with Uruguay, then $\mathbf{P}(QU) = 1/3$ whereas $\mathbf{U}(QU) = 1/2.$

In search of debiasing: random order sequential procedures

Can we fix this by randomising the order in which we fill the groups? Ie a *random order sequential draw* giving rise to $\mathbf{P}_{rand}$.

Here $\mathbf{P}_{rand} = \mathbf{U}$!

But in general random order sequential draws are still biased and in fact their bias can even be larger than that for a fixed order sequential draw.

In search of debiasing: multiple ball strategies

Analysing the sequential draw more carefully:

Two of the possible draws (D_3 and D_4) put Uruguay in Group A with Qatar, while only one (D_1) puts Mexico in Group A, and one (D_2) puts Switzerland in Group A.

This suggests that when selecting the team from Pot 2 to put in Group A, we could use a bowl with *two* Uruguay balls, and only one from each of Mexico and Switzerland.
The next team in the draw is then selected uniformly at random just as before.

This simple *multiple balls solution* thus achieves the correct conditional probability (in terms of $\mathbf{U}$) for the team from Pot 2 to be placed in Group A.

Towards a general multiple ball strategy

When selecting the team from Pot 2 to put in Group A, count, for each team in Pot 2, the number of valid draws which could be constructed and that have that team in that position.

In simple example above case, we have $n_U = 2$ for Uruguay, and $n_M = 1$ for Mexico, and $n_S = 1$ for Switzerland. We thus place n_U balls for Uruguay, n_M balls for Mexico, n_S balls for Switzerland, all into a bowl, and then sample one of the balls uniformly at random.

Thus for instance

$$\mathbf{P}(\text{pick Uruguay}) = \frac{n_U}{n_U + n_D + n_S} \ .$$

Can this strategy be extended to larger draws?

In principle, yes.

Towards a general multiple ball strategy

When selecting the team from Pot 2 to put in Group A, count, for each team in Pot 2, the number of valid draws which could be constructed and that have that team in that position.

In simple example above case, we have $n_U = 2$ for Uruguay, and $n_M = 1$ for Mexico, and $n_S = 1$ for Switzerland. We thus place n_U balls for Uruguay, n_M balls for Mexico, n_S balls for Switzerland, all into a bowl, and then sample one of the balls uniformly at random.

Thus for instance

$$\mathbf{P}(\text{pick Uruguay}) = \frac{n_U}{n_U + n_D + n_S} \ .$$

Can this strategy be extended to larger draws?

In principle, yes.

In practice no! Eg for World Cup 2022 draw, when choosing the Pot 2 team to play Qatar, need to put 5×10^{14} balls into the bowl.

Towards a general multiple ball strategy

When selecting the team from Pot 2 to put in Group A, count, for each team in Pot 2, the number of valid draws which could be constructed and that have that team in that position.

In simple example above case, we have $n_U = 2$ for Uruguay, and $n_M = 1$ for Mexico, and $n_S = 1$ for Switzerland. We thus place n_U balls for Uruguay, n_M balls for Mexico, n_S balls for Switzerland, all into a bowl, and then sample one of the balls uniformly at random.

Thus for instance

$$\mathbf{P}(\text{pick Uruguay}) = \frac{n_U}{n_U + n_D + n_S} \ .$$

Can this strategy be extended to larger draws?

In principle, yes.

In practice no! Eg for World Cup 2022 draw, when choosing the Pot 2 team to play Qatar, need to put 5×10^{14} balls into the bowl.

Can we make this practical?

Random discrete rational random simulation

Given a partial draw, set n_j to be the number of ways of completing the draw which put team j in the next position.

We simply want to simulate from a discrete random variable with at most 8 possible values with

$$\mathbf{P}(\text{pick } j) := p_j = \frac{n_j}{\sum_k n_k} \ .$$

But we want to do this by *discrete rational simulation*.

Random discrete rational random simulation

Given a partial draw, set n_j to be the number of ways of completing the draw which put team j in the next position.

We simply want to simulate from a discrete random variable with at most 8 possible values with

$$\mathbf{P}(\text{pick } j) := p_j = \frac{n_j}{\sum_k n_k} \ .$$

But we want to do this by *discrete rational simulation*.

However the number of balls of each type does not need to be fixed. Ie it can be random

Suppose we decide to have M balls in total. Then, letting $r_j = Mp_j$ we shall choose a number of balls of type j, R_j say, with

$$\mathbb{E}(R_j) = r_j \quad \forall j$$

How to do this?

We want M to be as small as possible.

But we also want to have at least one ball of type j, for each j for which $p_j > 0$.

Therefore choose

$$M = \lceil \max\{(1/p_j) : p_j > 0\} \rceil$$

(where $\lceil x \rceil$ = smallest integer $\geq x$).

Then we have that $r_j \geq 1$ for each j for which $p_j > 0$. So by residual sampling, we will obtain at least one ball when j can lead to a feasible solution.

An algorithm for simulating from the $\{p_j\}$.

1. For each $1 \leq j \leq J$, place $a_j := \lfloor r_j \rfloor$ balls of type j into the bowl.
2. Set $u_j = r_j - a_j$, and $v_j = \sum_{\ell=1}^{j} u_\ell$ for $1 \leq j \leq J$, with $v_0 = 0$. Also set $k = v_J$. (Thus $k = M - \sum_j a_j$ which is a non-negative integer.)
3. Simulate independent uniform random variables $U_1, \ldots, U_k$ with $U_i \sim U[i-1, i)$.
4. For each $1 \leq j \leq J$, let $b_j = |\{i : U_i \in [v_{j-1}, v_j)\}|$ be the number of random variables U_i which lie in the interval $[v_{j-1}, v_j)$, and add b_j additional balls of type j to the bowl.
5. Select a ball uniformly at random from the M balls in the bowl.

Not quite there yet ...

The above algorithm is unbiased as required.

It is completely generic, can cope with any number and type of constraints.

M will usually be small unless one of the p_js is very small. (In this case the FIFA method is hugely biased.)

Not quite there yet ...

The above algorithm is unbiased as required.

It is completely generic, can cope with any number and type of constraints.

M will usually be small unless one of the p_js is very small. (In this case the FIFA method is hugely biased.)

However it requires knowledge of the p_js at each step.

These is typically not feasible to compute exactly in a time interval short enough for a football fan to not get bored. (Could be a 15 digit number.)

Echoes of intractable likelihood problems

We wish to simulate from a distribution (eg a posterior) without having to evaluate the density of that distribution.

Ubiquitous problem in computational statistics

A simple collection of techniques for doing these simulations are known as retrospective sampling (used in diffusion simulation, Bernoulli factories, pseud-marginal methods, inference problems for latent stochastic processes, mixtures, HMMs etc....)

A simple variant is effective here .....

Unbiased estimators of p_j

We can readily acquire unbiased estimators of the p_js through the rejection sampler simulator described above.

Eg sample 1000 exact draws starting from the current stage of the draw and estimate

$$\hat{p}_j = \#\{\text{draws which include } j \text{ as the next team}\} \; .$$

It turns out that we can apply the same algorithm with the $\{\hat{p}_j\}$s without bias. Set

$$\hat{M} = \lceil \max\{(1/\hat{p}_j) : \hat{p}_j > 0\} \rceil$$

$$\hat{r}_j = \hat{M}\hat{p}_j$$

A practical algorithm for simulating from the $\{p_j\}$.

1. Produce the $\{\hat{p}_j\}$s as above.
2. For each $1 \le j \le J$, place $\hat{a}_j := \lfloor \hat{r}_j \rfloor$ balls of type j into the bowl.
3. Set $\hat{u}_j = \hat{r}_j - \hat{a}_j$, and $v_j = \sum_{\ell=1}^{j} \hat{u}_\ell$ for $1 \le j \le J$, with $v_0 = 0$. Also set $k = v_J$. (Thus $k = \hat{M} - \sum_j \hat{a}_j$ which is a non-negative integer.)
4. Simulate independent uniform random variables $U_1, \ldots, U_k$ with $U_i \sim U[i-1, i)$.
5. For each $1 \le j \le J$, let $\hat{b}_j = |\{i : U_i \in [v_{j-1}, v_j)\}|$ be the number of random variables U_i which lie in the interval $[v_{j-1}, v_j)$, and add $\hat{b}_j$ additional balls of type j to the bowl.
6. Select a ball uniformly at random from the M balls in the bowl.

A fair sequential multiple balls draw

Proposition

If the above procedure is used repeatedly for each position of a group draw, then the final group draw will have distribution **U**, *i.e.it will be equally likely to be any of the potential valid draws.*

How does it work in practice?

In theory, the algorithm does not guarantee that each team with $p_j > 0$ is represented. However in practice it is vanishingly unlikely that this is not the case.

$\hat{M}$ tends to be larger for smaller sample sizes in the estimation of p_j by $\hat{p}_j$, but not by much.

Refinements to reduce $\hat{M}$ which exploit symmetries between countries from different continental confederations can be constructed.

An MCMC solution

It is easy to construct an MCMC algorithm which proposes to swapping the groups for two teams from the same pot.

Unlike MCMC which we are familar with, such a simulation could start from stationarity using the rejection sampler.

Note that this is rather different from a sequential solution in which slots in the draw are filled successively.

However moves can still easily be simulated by drawing balls from pots.

For example we could have a seed pot containing 4 balls numbered 1-4 and a group pot containing 8 balls A-H. One seed ball and two group pot balls define a proposed swap, between the two chosen groups, of the teams currently in those groups of the chosen seed.

There is no randomness in the accept/reject step: valid moves are accepted and invalid ones are rejected.

Discussion

We also have a *multiple lives* solution to this problem, whereby some teams are only allocated to the current location in the draw if they have been selected a prescribed multiple number of times. Thus the team selected is only chosen if it loses its lives first.

Discussion

We also have a *multiple lives* solution to this problem, whereby some teams are only allocated to the current location in the draw if they have been selected a prescribed multiple number of times. Thus the team selected is only chosen if it loses its lives first.

All procedures use transparent randomness (ie drawing balls from pots etc) as well as hidden randomness. A good method should have as little hidden randomness as possible. Can quantify in terms of expected conditional variance, typically far greater than 90% for the multiple balls draw for instance (and can be made arbitrarily close to 100%).

Discussion

We also have a *multiple lives* solution to this problem, whereby some teams are only allocated to the current location in the draw if they have been selected a prescribed multiple number of times. Thus the team selected is only chosen if it loses its lives first.

All procedures use transparent randomness (ie drawing balls from pots etc) as well as hidden randomness. A good method should have as little hidden randomness as possible. Can quantify in terms of expected conditional variance, typically far greater than 90% for the multiple balls draw for instance (and can be made arbitrarily close to 100%).

However practical and fair solutions which only use transparent randomness are not available (yet at least).

Discussion

We also have a *multiple lives* solution to this problem, whereby some teams are only allocated to the current location in the draw if they have been selected a prescribed multiple number of times. Thus the team selected is only chosen if it loses its lives first.

All procedures use transparent randomness (ie drawing balls from pots etc) as well as hidden randomness. A good method should have as little hidden randomness as possible. Can quantify in terms of expected conditional variance, typically far greater than 90% for the multiple balls draw for instance (and can be made arbitrarily close to 100%).

However practical and fair solutions which only use transparent randomness are not available (yet at least).

Whether football authorities care about fairness is unclear. There are many political factors which influence the form of the draw and may be more important to many than fairness.

Discussion

We also have a *multiple lives* solution to this problem, whereby some teams are only allocated to the current location in the draw if they have been selected a prescribed multiple number of times. Thus the team selected is only chosen if it loses its lives first.

All procedures use transparent randomness (ie drawing balls from pots etc) as well as hidden randomness. A good method should have as little hidden randomness as possible. Can quantify in terms of expected conditional variance, typically far greater than 90% for the multiple balls draw for instance (and can be made arbitrarily close to 100%).

However practical and fair solutions which only use transparent randomness are not available (yet at least).

Whether football authorities care about fairness is unclear. There are many political factors which influence the form of the draw and may be more important to many than fairness.

The multiple ball sampler and the MCMC sampler programmes can be found at http://probability.ca/fdraw/