Sequential Monte Carlo for graphical models:
Graph decompositions and Divide-and-Conquer SMC

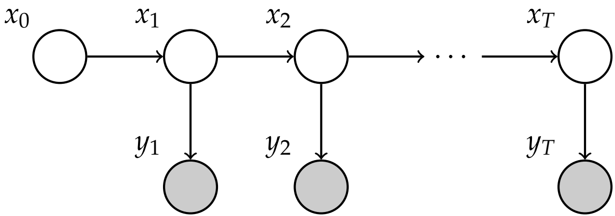
UNIVERSITY OF CAMBRIDGE

Fredrik Lindsten
Department of Engineering
The University of Cambridge
Cambridge, UK

Ongoing joint work with: **Christian A. Naesseth** (Linköping University), **Thomas Schön** (Uppsala University), **Adam M. Johansen** (University of Warwick), **Bonnie Kirkpatrick** (University of Miami), **John Aston** (University of Cambridge) and **Alexandre Bouchard-Côté** (University of British Columbia).

A **probabilistic graphical model** (PGM) is a probabilistic model where a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represents the conditional independency structure between random variables,

1. a set of **vertices** $\mathcal{V}$ (nodes) represents the random variables
2. a set of **edges** $\mathcal{E}$ containing elements $(i, j) \in \mathcal{E}$ connecting a pair of nodes $(i, j) \in \mathcal{V} \times \mathcal{V}$



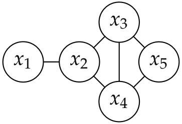
$$p(x_{0:T}, y_{1:T}) = p(x_0) \prod_{t=1}^{T} p(x_t \mid x_{t-1}) \prod_{t=1}^{T} p(y_t \mid x_t).$$

Background – graphical models (II/II)

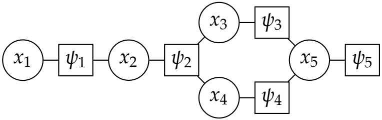
For an undirected graphical model (Markov random field), the joint PDF over all the involved random variables $X_{\mathcal{V}} := (x_i)_{i \in \mathcal{V}}$ is

$$p(X_{\mathcal{V}}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(X_C),$$

where $\mathcal{C}$ is the set of cliques in $\mathcal{G}$, and $Z = \int \prod_{C \in \mathcal{C}} \psi_C(X_C) dX_{\mathcal{V}}$.



Undirected graph



Example of a **factor graph** making interactions explicit, $p(x_{1:5}) = \frac{1}{Z} \prod_{i=1}^{5} \psi_i(\cdot)$.

Background – sequential Monte Carlo (I/II)

Approximate a **sequence** of probability distributions on a sequence of probability spaces of **increasing dimension**.

Let $\{\gamma_k(x_{1:k})\}_{k\geq 1}$ be a sequence of unnormalised densities and

$$\bar{\gamma}_k(x_{1:k}) = \frac{\gamma_k(x_{1:k})}{Z_k}$$

Approximates

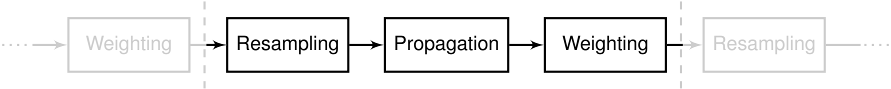
$$\bar{\gamma}_k(x_{1:k}) \approx \sum_{i=1}^{N} \frac{w_k^i}{\sum_{l=1}^{N} w_k^l} \delta_{x_{1:k}^i}(x_{1:k}).$$

Ex. (SSM)

$$\bar{\gamma}_k(x_{1:k}) = p(x_{1:k} \mid y_{1:k}), \qquad \gamma_k(x_{1:k}) = p(x_{1:k}, y_{1:k}),$$

$$Z_k = p(y_{1:k}).$$

Background – sequential Monte Carlo (II/II)



1. **Resampling:** $\{x^i_{1:k-1}, w^i_{k-1}\}_{i=1}^N \to \{\check{x}^i_{1:k-1}, 1\}_{i=1}^N$.
2. **Propagation:** $x^i_k \sim q_k(x_k \mid \check{x}^i_{1:k-1})$ and $x^i_{1:k} = \{\check{x}^i_{1:k-1}, x^i_k\}$.
3. **Weighting:** $w^i_k = W_k(x^i_{1:k}) = \frac{\gamma_k(x^i_{1:k})}{\gamma_{k-1}(x^i_{1:k-1})q_k(x^i_k | x^i_{1:k-1})}$.

$$\Rightarrow \{x^i_{1:k}, w^i_k\}_{i=1}^N$$

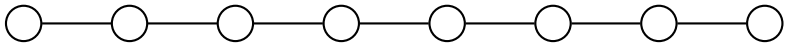
SMC samplers are used to approximate a sequence of probability distributions on a sequence of probability spaces.

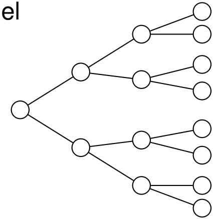
Using an artificial sequence of intermediate target distributions for an SMC sampler is a powerful (and **quite possibly underutilised**) idea.

Key idea: Perform and make use of various **decompositions** of graphical models to design SMC inference methods.

Outline

1. Example – from information theory
2. Sequential decomposition → “standard” SMC
   a) Sequential decomposition and SMC for PGMs
   b) Example – Estimating partition functions
3. Tree decomposition → *Divide-and-Conquer* with SMC
   a) Tree decomposition and D&C-SMC for PGMs
   b) Example – Hierarchical Bayesian Model





Example borrowed from:

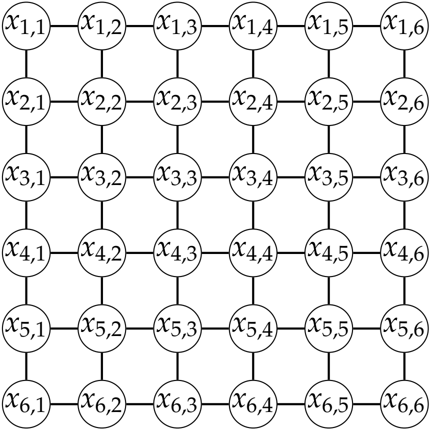
M. Molkaraie and H.-A. Loeliger, **Monte Carlo algorithms for the partition function and information rates of two-dimensional channels**, *IEEE Transactions on Information Theory*, 59(1): 495–503, 2013.

2D binary-input channel with the **constraint** that no two horizontally or vertically adjacent variables may be both be equal to 1.

$$\begin{array}{ccccc} \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & 0 & 1 & 0 & \cdots \\ \cdots & 0 & 0 & 1 & \cdots \\ \cdots & 0 & 1 & 0 & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{array}$$

“of interest in magnetic and optical storage”

The channel can be described by a square lattice **undirected graphical model**.



The variables are binary $x_{\ell,j} \in \{0,1\}$ and the interactions are pair-wise between adjacent variables. Factors:

$$\psi(x_{\ell,j}, x_{m,n}) = \begin{cases} 0, & x_{\ell,j} = x_{m,n} = 1 \\ 1, & \text{otherwise} \end{cases}$$

2D channel capacity – graphical model

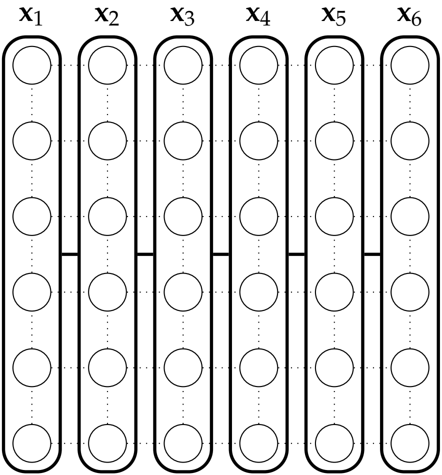
The resulting joint PDF is given by

$$p(X_{\mathcal{V}}) = \frac{1}{Z} \prod_{(\ell j, mn) \in \mathcal{E}} \psi(x_{\ell,j}, x_{m,n}),$$

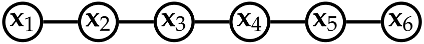
For a channel of dimension $M \times M$ we can write the finite-size **noiseless capacity** as

$$C_M = \frac{1}{M^2} \log_2 Z.$$

Unfortunately calculating Z exactly for these types of models is computationally prohibitive, since the complexity is exponential in the size of the grid M.

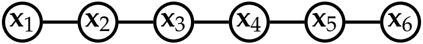


Rewrite the PGM as a high-dimensional **undirected chain** by introducing a new set of variables $\mathbf{x}_k$.



$$\phi(\mathbf{x}_k) = \prod_{j=1}^{M-1} \psi(x_{j,k}, x_{j+1,k}),$$

$$\psi(\mathbf{x}_{k-1}, \mathbf{x}_k) = \prod_{j=1}^{M} \psi(x_{j,k-1}, x_{j,k}).$$



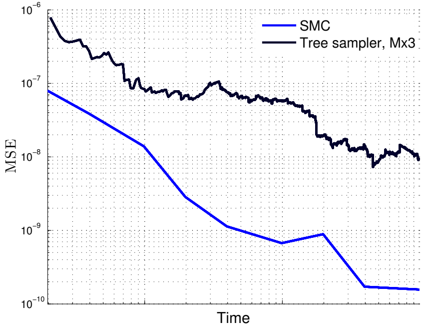
The **undirected chain** results in the following joint PDF

$$p(X_{\mathcal{V}}) = \frac{1}{Z} \prod_{k=1}^{M} \phi(\mathbf{x}_k) \prod_{k=2}^{M} \psi(\mathbf{x}_{k-1}, \mathbf{x}_k).$$

Natural **sequential decomposition**:

$$\gamma_k(\mathbf{x}_{1:k}) = \prod_{\ell=1}^{k} \phi(\mathbf{x}_\ell) \prod_{\ell=2}^{k} \psi(\mathbf{x}_{\ell-1}, \mathbf{x}_\ell).$$

2D channel capacity – 60×60 example



Our SMC sampler compared to the **tree sampler** by

F. Hamze and N. de Freitas, **From fields to trees**, *In Proceedings of the conference on Uncertainty in Artificial Intelligence (UAI)*, Banff, Canada, July, 2004.

implemented according to

M. Molkaraie and H.-A. Loeliger, **Monte Carlo algorithms for the partition function and information rates of two-dimensional channels**, *IEEE Transactions on Information Theory*, 59(1): 495–503, 2013.

For the 2D channel: **fully adapted** SMC sampler. To sample $\mathbf{x}_k$ we run a forward/backward algorithm for the kth column.

This was just a special case, the important question is, can we do this for a general graphical model?! **Yes!**

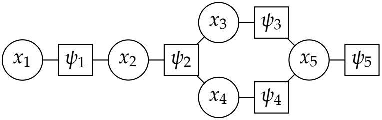
Key idea:

- Perform a **sequential decomposition** of the graphical model.
- Each **subgraph** induces an artificial target distribution.
- Apply SMC to the sequence of artificial target distributions.

Sequential decomposition of PGMs – pictures

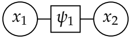
The joint PDF of the set of random variables indexed by $\mathcal{V}$, $X_{\mathcal{V}} \triangleq \{x_1, \ldots, x_{|\mathcal{V}|}\}$

$$p(X_{\mathcal{V}}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(X_C).$$

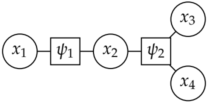


Example of a sequential decomposition of the above factor graph (the target distributions are built up by adding factors at each iteration),

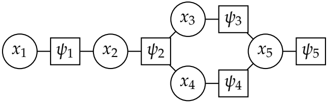
$\gamma_1(X_{\mathcal{L}_1})$



$\gamma_2(X_{\mathcal{L}_2})$



$\gamma_3(X_{\mathcal{L}_3}) \propto p(X_{\mathcal{V}})$



Let $\{\mathcal{C}_k\}_{k=1}^K$ be an **ordered partition** of $\mathcal{C}$. Define:

$$\psi_k(X_{\mathcal{I}_k}) \triangleq \prod_{C \in \mathcal{C}_k} \psi_C(X_C),$$

where $\mathcal{I}_k \subseteq \{1, \ldots, |\mathcal{V}|\}$ is the set of indices in the domain of ψ_k.

The **sequential decomposition** is based on these factors,

$$\gamma_k(X_{\mathcal{L}_k}) \triangleq \prod_{\ell=1}^{k} \psi_\ell(X_{\mathcal{I}_\ell}),$$

where $\mathcal{L}_k \triangleq \bigcup_{\ell=1}^k \mathcal{I}_\ell$.

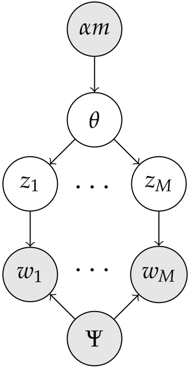
By construction, $\mathcal{L}_K = \mathcal{V}$ and the joint PDF $p(X_{\mathcal{L}_K}) \propto \gamma_K(X_{\mathcal{L}_K})$.

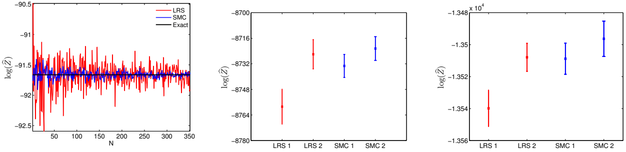
Algorithm SMC sampler for graphical models

1. **Initialise ($k = 1$):** Draw $X^i_{\mathcal{L}_1} \sim q_1(\cdot)$ and set $w^i_1 = W_1(X^i_{\mathcal{L}_1})$.
2. **For** $k = 2$ **to** K **do:**
   (a) Draw $a^i_k \sim \text{Cat}(\{w^j_{k-1}\}^N_{j=1})$.
   (b) Draw $\xi^i_k \sim q_k(\cdot | X^{a^i_k}_{\mathcal{L}_{k-1}})$ and set $X^i_{\mathcal{L}_k} = X^{a^i_k}_{\mathcal{L}_{k-1}} \cup \xi^i_k$.
   (c) Set $w^i_k = W_k(X^i_{\mathcal{L}_k})$.

- Generates samples $\{X^i_{\mathcal{L}_K}, w^i_K\}^N_{i=1} \overset{\text{approx.}}{\sim} p(X_{\mathcal{L}_K})$.
- Provides an unbiased estimate of the **partition function**!

Evaluating Latent Dirichlet Allocation models on heldout documents corresponds to estimating the partition function of a PGM.



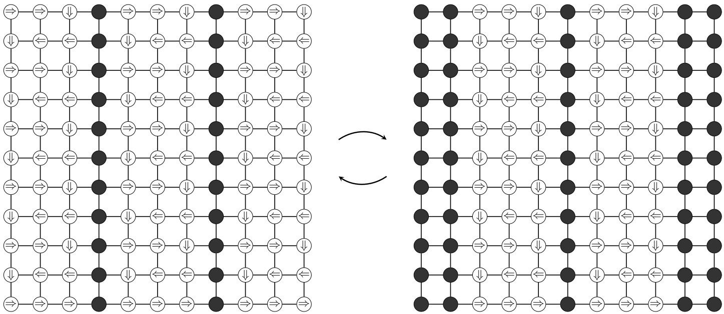


(a) Synthetic (b) PMC (c) 20 newsgroups

Estimates of the log-likelihood of heldout documents for various datasets.

Can be used for block sampling with PMCMC.

Ex) Iteratively update the white variables, conditionally on the black



Potentially useful when no "natural" sequential decomposition is available for full graph.

The sequential decomposition is basically a chain-oriented decomposition of the PGM. This naturally leads to a sequence of distributions suitable for standard SMC samplers.

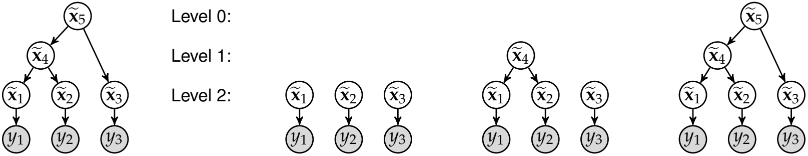
Divide-and-Conquer SMC:

Key idea:

- Consider graph decompositions organised on **trees**.
- Assign **auxiliary target distributions** to all nodes of the tree.
- Inference using a **new class** of SMC algorithms.

Tree decomposition of PGMs (I/II)

Hierarchical Bayesian network



We initialise the D&C-SMC with **independent particle populations** for each leaf in the tree decomposition. These are then merged, resampled and propagated as we move up the tree.

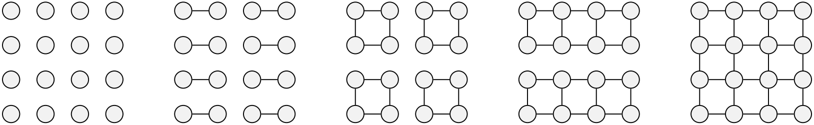
Iter 1: Initialise $(\widetilde{\mathbf{x}}_k^i, \mathbf{w}_k^i)_{i=1}^N$ for $k = 1, 2, 3$.

Iter 2: Merge populations 1 and 2 and propagate $\Rightarrow (\widetilde{\mathbf{x}}_{1,2,4}^i, \mathbf{w}_4^i)_{i=1}^N$

Iter 3: Merge populations 3 and 4 and propagate $\Rightarrow (\widetilde{\mathbf{x}}_{1,2,3,4,5}^i, \mathbf{w}_5^i)_{i=1}^N$

Tree decomposition follows naturally when the graphical model is a tree. However, the idea is more generally applicable.

Example: Lattice Markov random field



The subgraphs can be **organised on a tree!**

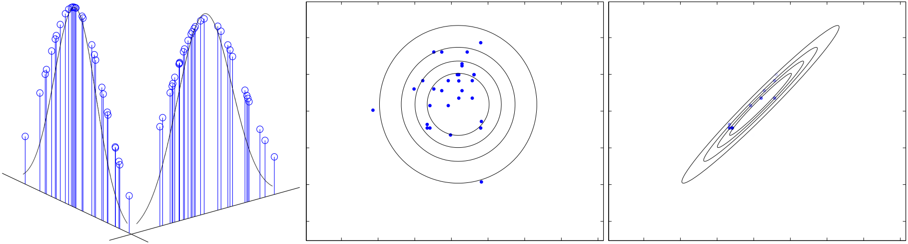
Algorithm dc_smc(t) – D&C-SMC for node $t \in T$

1. For $c \in \mathcal{C}(t)$:
   1. $(\mathbf{x}_c^i, \mathbf{w}_c^i)_{i=1}^N \leftarrow$ dc_smc(c).
   2. Resample $(\mathbf{x}_c^i, \mathbf{w}_c^i)_{i=1}^N$ to obtain the equally weighted particle system $(\check{\mathbf{x}}_c^i, 1)_{i=1}^N$.
2. For particle $i = 1, \ldots, N$:
   1. Simulate $\widetilde{\mathbf{x}}_t^i \sim q_t(\cdot \mid \check{\mathbf{x}}_{c_1}^i, \ldots, \check{\mathbf{x}}_{c_C}^i)$ from some proposal kernel on $\widetilde{\mathsf{X}}_t$, and where $(c_1, c_2, \ldots, c_C) = \mathcal{C}(t)$.
   2. Set $\mathbf{x}_t^i = (\check{\mathbf{x}}_{c_1}^i, \ldots, \check{\mathbf{x}}_{c_C}^i, \widetilde{\mathbf{x}}_t^i)$.
   3. Compute $\mathbf{w}_t^i = \dfrac{\gamma_t(\mathbf{x}_t^i)}{\prod_{c \in \mathcal{C}(t)} \gamma_c(\check{\mathbf{x}}_c^i)} \dfrac{1}{q_t(\widetilde{\mathbf{x}}_t^i \mid \check{\mathbf{x}}_{c_1}^i, \ldots, \check{\mathbf{x}}_{c_C}^i)}$.
3. Return $(\mathbf{x}_t^i, \mathbf{w}_t^i)_{i=1}^N$.

- Generalises the SMC framework (std SMC recovered if T is a chain).
- Consistent and gives an unbiased estimate of the partition function.

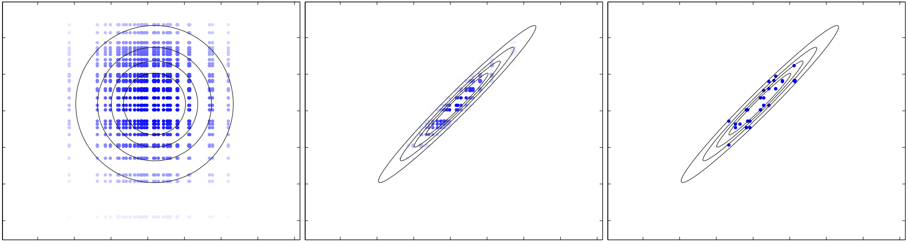
D&C-SMC extensions (I/III)

D&C "Sampling Importance Resampling"



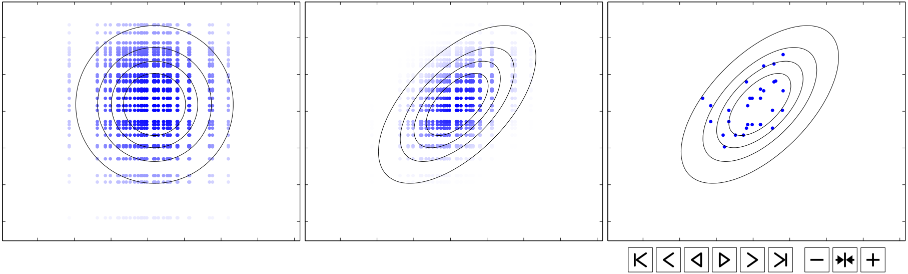
$\prod_{c\in\mathcal{C}(t)} \widehat{\gamma}_c^N(\mathbf{x}_c) \rightarrow$ Resampling $\rightarrow (\check{\mathbf{x}}_{c_1}^i,\dots,\check{\mathbf{x}}_{c_C}^i)_{i=1}^N \rightarrow$ Weighting $\rightarrow (\mathbf{x}_t^i,\mathbf{w}_t^i)_{i=1}^N$

D&C-SMC: Auxiliary mixture sampling



$\prod_{c \in \mathcal{C}(t)} \widehat{\gamma}_c^N(\mathbf{x}_c)$ $\rightarrow$ (Auxiliary) Weighting $\rightarrow$ Resampling

D&C-SMC: Auxiliary mixture sampling + Tempering



$\prod_{c \in \mathcal{C}(t)} \widehat{\gamma}_c^N(\mathbf{x}_c)$ → (Auxiliary) Weighting → Tempering

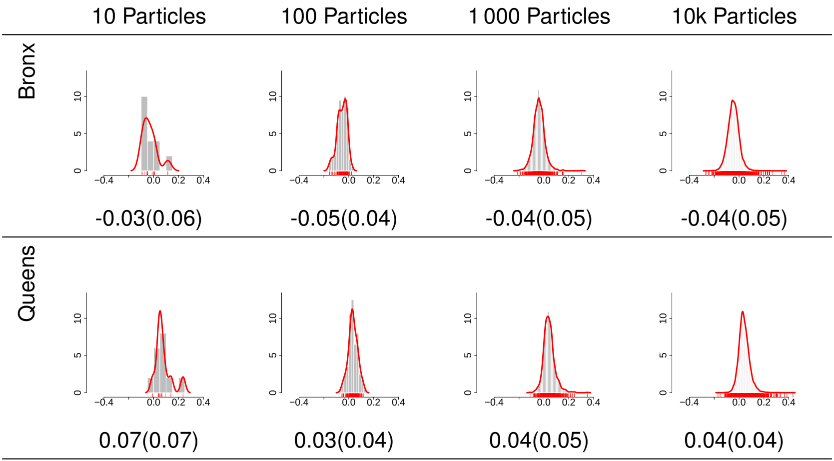
Example – NY maths test (model)

Data: Table of test results (278 399 instances), with school code, year, number of students tested in that year and school, and the number students that passed.

Structure: We organise the data into a tree with the following form: NYC (root), borough of the school district, school district, school, year.

Parameters:
- Observations at the leaf (binomial $p_t = \mathrm{logistic}(\theta_t)$).
- Parameters $\theta_{t'} = \theta_t + \Delta_e$, with $\Delta_e \sim \mathrm{N}(0, \sigma_e^2)$.
- Hyperparameters $\sigma_e^2 \sim \mathrm{Exp}(1)$.

After marginalization of internal θ-parameters, the dimensionality of the *remaining* parameters in the model is 3 555.



Posterior distribution of $\delta_e =$"difference in logistic(θ) along edge e" for two boroughs (rows) and four computational regimes (columns), with mean and std dev below each histogram.

We compare our D&C-SMC (implemented in Java) to Hamiltonian Monte Carlo (Stan, implemented in C++).

Similar posterior approximation accuracy.

Method	Iterations/Particles	Runtime
D&C-SMC	1000	39 s
HMC (Stan)	2000 (50% burn-in)	3860 s (64 min)

Node	Stan	D&C-SMC	Speedup
Manhattan	0.17	15.96	93.89
Bronx	0.05	8.12	165.69
Brooklyn	0.18	6.52	36.22
Queens	0.07	14.01	209.05
Staten Island	0.05	25.50	481.17

The effective samples per second and speedup.

Conclusions

- We have derived SMC-based inference methods for graphical models of arbitrary topologies with discrete and/or continuous random variables.
- **Key insight:** We exploit various decompositions of the graphical model to design efficient SMC samplers.
- Examples involving:
  1. estimating the partition function
  2. inferring the latent variables
  3. learning parameters.
- If you have interesting and challenging problems involving graphical models, let us know!

Some references

SMC (and PMCMC) methods for graphical models:

Christian A. Naesseth, Fredrik Lindsten and Thomas B. Schön, **Sequential Monte Carlo for Graphical Models**. *Advances in Neural Information Processing Systems (NIPS) 27*, December, 2014.

F. Lindsten, A. M. Johansen, C. A. Naesseth, B. Kirkpatrick, T. B. Schön, J. Aston and A. Bouchard-Côté, **Divide-and-Conquer with Sequential Monte Carlo**. *Preprint arXiv:1406.4993*, June, 2014.

Christian A. Naesseth, Fredrik Lindsten and Thomas B. Schön, **Capacity estimation of two-dimensional channels using Sequential Monte Carlo**. *Proceedings of the 2014 IEEE Information Theory Workshop (ITW)*, November, 2014.

Thank you!!