# Scaling-up Split-Merge MCMC with Locality Sensitive Sampling (LSS)

**Chen Luo**, Anshumali Shrivastava
Rice University
Houston, TX
Jan. 29th, 2019

**The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)**

# The Good Old Metropolis Hastings

- Goal: Sample from Target Distribution $P(x)$ is hard to sample from.
  - Come up with a proposal state transition $Q(x'|x)$ given current state $x$.
- The MH Algorithm
  - Draw a sample from $x' \sim Q(x'|x)$
  - Accept this sample with probability: $\min(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)})$
- Forms a Markov Chain that converges to $P(x)$

# Key Question for Ages

- The MH Algorithm
  - Draw a sample from $x' \sim Q(x'|x)$
  - Accept this sample with probability: $\min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$
- How to design a good $Q(x'|x)$?
  - Random and efficient choice ⇒ Low acceptance (poor P(x'))
    - More Iteration but each iteration is fast!
  - Informed Choice ⇒ Needs to correlate Q and P ⇒ Expensive
    - Less iterations but each iteration very expensive.

Is there a sweet spot?

# Existing Understanding

- The Good
  - It will eventually converge to the right distribution (The Math)
- The Bad
  - Expensive: There is a tradeoff between per iteration cost vs number of iterations.
- The Ugly
  - Prior work ignored per iteration cost. Number of iteration was the gold standard metric instead of running time (or total cost of convergence)
  - Essentially focus on faster convergence even if each iteration is itself another MCMC chain.
  - No significant progress on getting around the tradeoff.

# Focus: Split-Merge MCMC

- Split-Merge MCMC
  - Clustering with number of clusters unknown.
  - Useful for dealing with clustering or topic modeling.
- Mainly Two steps for State Transition:
  - Split: Split a cluster (component) to two clusters (components)
  - Merge: Choose two cluster and merge them to one

**Same Problem: Slow convergence!**

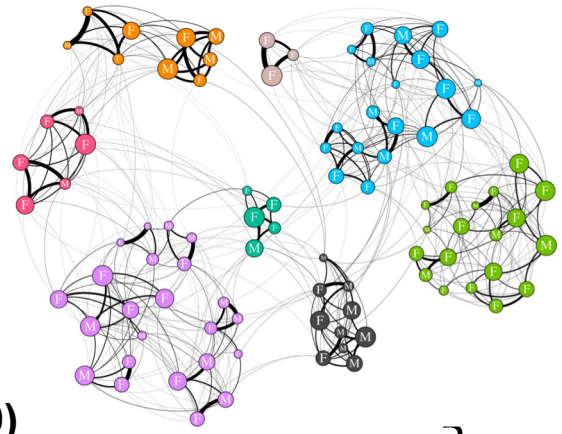


Image credit: http://rspb.royalsocietypublishing.org/content/284/1865/20171313/F1

# Existing Works on Split-Merge MCMC

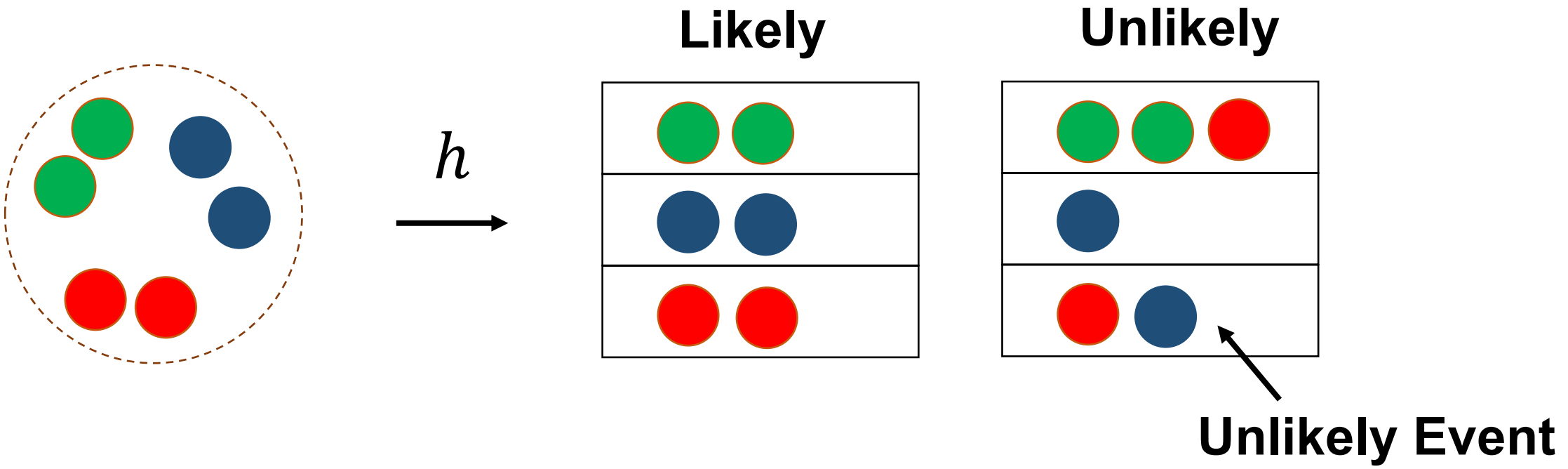
- RGSM: Restricted Gibbs split-merge.
  - Using restricted Gibbs sampling to generate proposals with higher likelihood. (A Gibbs sampling in each iteration)!!
  - Cost of Gibbs sampling is very high
- SDDS: Smart-Dumb/Dumb-Smart Algorithm
  - Combine Smart proposal with dumb proposal
  - Smart proposal obtained using greedy search ⇒ **Expensive iterations.**

**Need to find a informative proposal but cheap to compute.**

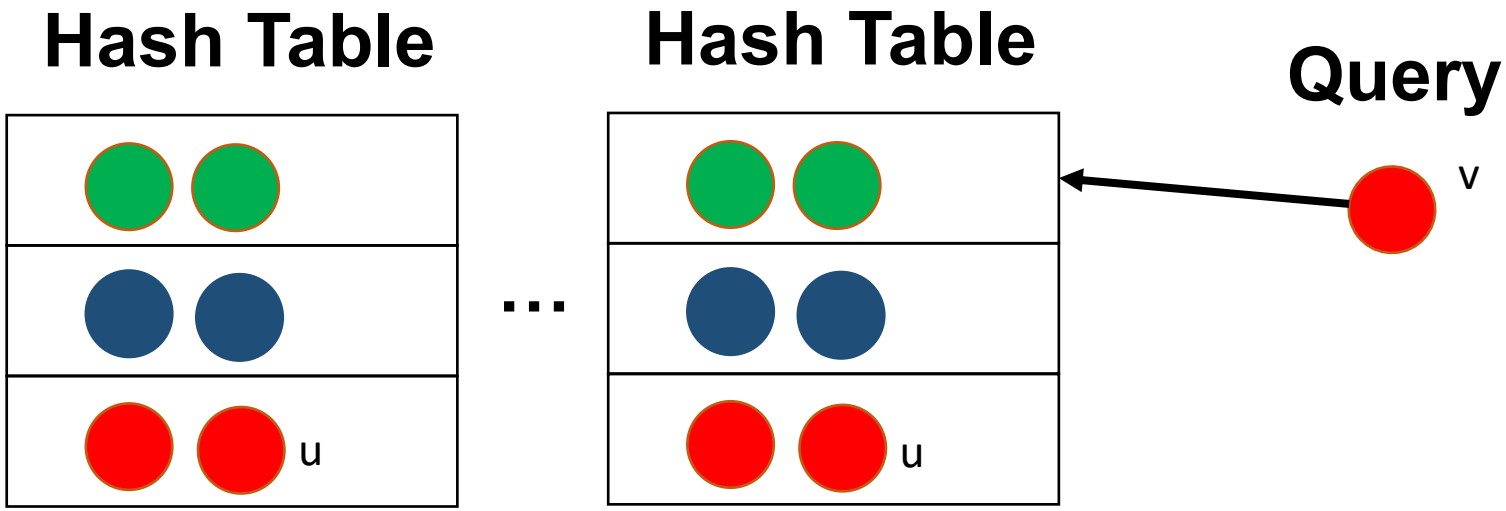
# Intuition of Our work: Leveraging Similarity for Smart Proposals

- Idea:
  - Similar entities are more likely to go to the same cluster than non-similar ones.
- Challenge:
  - $O(N^2)$ computations.
  - Dealing with very large-scale data set is prohibitive.
- Can we use LSH
  - YES there is a rare sweet spot
  - Only weighted minhash works so far!

# Locality Sensitive Hashing



# Locality Sensitive Sampling




Recent Advances in Sampling using LSH:


- B. Chen, A. Shrivastava, and R. C. Steorts. Unique entity estimation with application to the syrian conflict. *arXiv preprint arXiv:1710.02690*, 2017.
- C. Luo and A. Shrivastava. Arrays of (locality- sensitive) count estimators (ace): Anomaly detection on the edge. In *Proceedings of the 2018 World Wide Web Conference*, WWW'18, pages 1439–1448, 2018.
- R. Spring and A. Shrivastava. A new unbiased and ef- ficient class of lsh-based samplers and estimators for partition function computation in log-linear models. *arXiv preprint arXiv:1703.05160*, 2017.
- M. Charikar and P. Siminelakis. Hashing-based- estimators for kernel density in high dimensions. FOCS, 2017.

# LSHSM: First try on LSS based Proposal Design

- LSH Smart-split
  - Randomly selecting an element u in the dataset. Use LSS to sample points likely to be dissimilar to u.
  - If u and v belong to the same cluster C, we randomly split the cluster.
- Proposal probability:

$$q(x'|x) = \left(\frac{1}{2}\right)^{|C_u|+|C_v|-2} \sum_{u}^{C_u} \sum_{v}^{C_v} \left(\frac{1}{n}\left(1-\left(1-Pr(-u,v)^K\right)^L\right)\frac{|C_v \cap S_{-u}|}{|S_{-u}|}\right).$$

$$= \frac{\sum_{u}^{C_u}\sum_{v}^{C_v}\left(\frac{1}{n}\left(1-\left(1-Pr(-u,v)^K\right)^L\right)\frac{|C_v \cap S_{-u}|}{|S_{-u}|}\right)}{2^{|C_u|+|C_v|-2}}$$

# Dead End

- To split a cluster
  - Variable probability to every pair ⇒ Cannot avoid quadratic summation for exact probability computation
  - Same probability to every pair ⇒ Has to be random and hence uninformative

- We need a magic to somehow get a nontrivial probability expressions!
  - Magic do happen in mathematics.

# Unique properties of Minhash

- Given a cluster we can split them into two using minhash
  - Similar points in same cluster
  - Probability of split is exactly given as the collision probability:

$$Prob = \frac{\sum_j^{2D} \max\{0, (x^j_{\min} - x^j_{\max})\}}{\sum_j^{2D} x^j_{all}},$$

  - Needs very special choices of K =1
  - Not other hash function known to achieve this!

  - The probability expression is also unique, but can be computed in linear time. (Show in later slides)

# MinSM: Minhash based Proposal Design

- MinHash Smart-split
  - Randomly selecting an element u in the dataset.
  - Use LSS (Locality-sensitive Sampler) to sample a set of points Su that are likely to be similar to u.
  - We now split the component Cu into two components: Cu ∩ Su, Cu − Su.
- Proposal probability for Smart Split:

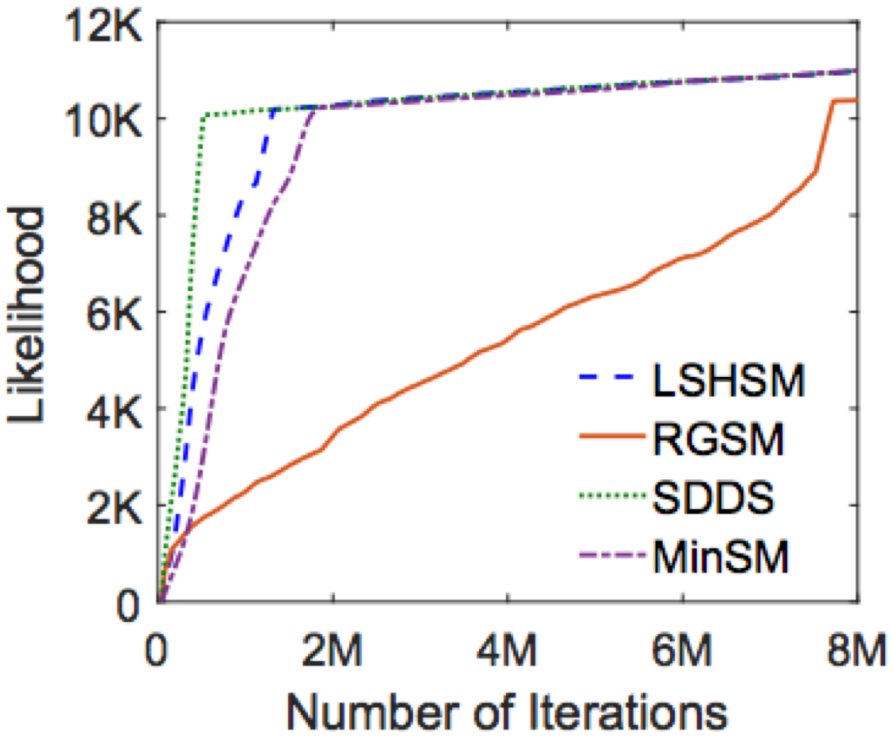
$$q(x'|x) = \frac{|S_u|}{n} \times Prob, \quad Prob = \frac{\sum_j^{2D} \max\{0, (x^j_{\min} - x^j_{\max})\}}{\sum_j^{2D} x^j_{all}}$$

# Experimental Setup

- **Dataset**
  - PubMed Dataset (abstractions that extracted from the PubMed)
    - Dimension: 141043 (Sparse)
    - Datasize: 8200000
  - KDDCUP (the KDD Cup 2004 data mining competition)
    - Dimension: 74
    - Datasize: 145751
- **Comparison Algorithms**
  - SDDS
  - MRGSM
- **Report**
  - Iteration vs Likelihood
  - Time vs. Likelihood
  - Clustering Accuracy (Accuracy & NMI)

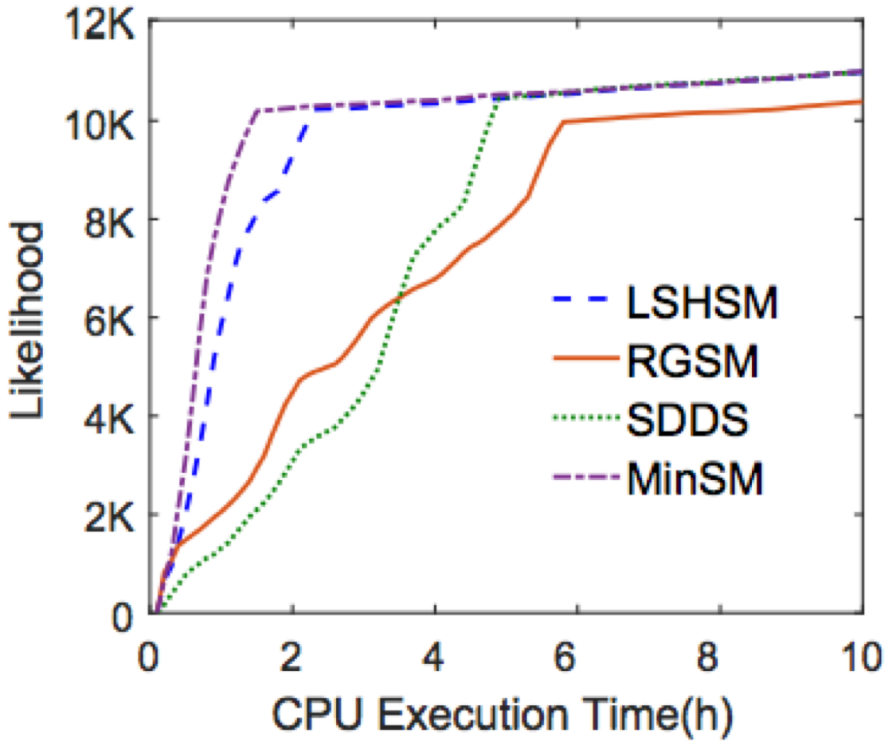
# Time/Iteration vs Likelihood


(c) PubMed Dataset

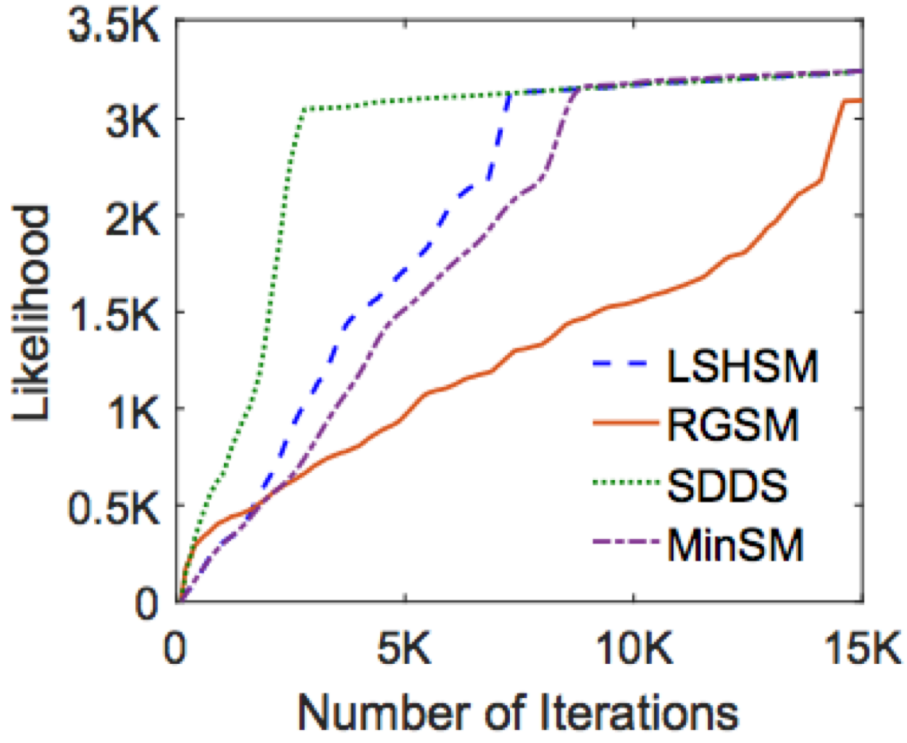



(d) PubMed Dataset

# Time/Iteration vs Likelihood




(a) KDDCUP Dataset




(b) KDDCUP Dataset

# Clustering Accuracy

Table 1: Clustering Accuracy for Different Methods

| Methods | Metric | S1 | S2 | S3 | KDD | Pub |
|---|---|---|---|---|---|---|
| RGSM | NMI | 0.96 | 0.93 | 0.88 | 0.74 | 0.63 |
| | Accuracy | 0.95 | 0.92 | 0.87 | 0.68 | 0.62 |
| SDDS | NMI | **0.97** | **0.96** | 0.95 | **0.86** | **0.80** |
| | Accuracy | **0.98** | **0.97** | 0.94 | **0.85** | **0.77** |
| LSHSM | NMI | 0.96 | 0.95 | **0.96** | 0.84 | 0.77 |
| | Accuracy | 0.97 | 0.94 | 0.96 | 0.83 | 0.75 |
| MinSM | NMI | 0.96 | 0.94 | **0.96** | 0.83 | 0.75 |
| | Accuracy | 0.97 | 0.94 | **0.97** | 0.84 | 0.74 |

# A Note on Parallel MCMC

- Parallelism
  - Running parallel MCMC chains.
  - Merge the Result
- MinSM reduces the overall cost of split-merge MCMC in each chain.

- Parallelizing MCMC is complementary to MinSM.

# Summary

- Magics do happen in Math.
- Minhash (more than 2-decade old) still has several magic left to be explored!
- LSH can mitigate computational challenges in Bayesian Inference.

RICE
Unconventional Wisdom

rushlab

# Thanks

- Questions!