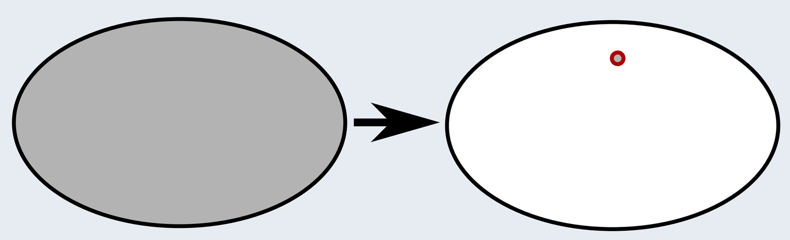
Local-Access Generators

Amartya Shankha Biswas, Ronitt Rubinfeld, Anak Yodpinyanee

CSAIL, MIT

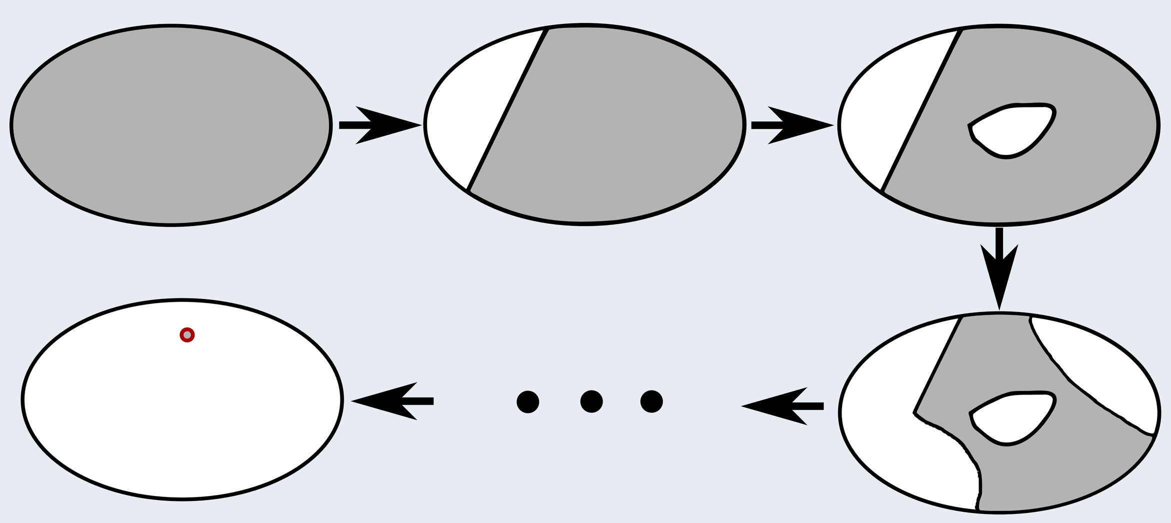
Partial Sampling from a Distribution

Full Sampling in $\mathcal{O}(T)$ time



N steps of *Partial Sampling*

Each partial step should take $\tilde{\mathcal{O}}(T/N)$ time.



Trivial Example - Sampling $G(n,p)$

Model: Undirected graph on N vertices (numbered $\langle 1,2,3,\cdots,N\rangle$), where each edge exists with prob. p.

Partial Sample: Given vertices u, v, is $(u,v) \in E$?

- Just a collection of $\frac{n(n-1)}{2}$ Bernoulli RVs with bias p.
- Given u, v, sample a single Bernoulli if neither (u,v) not (v,u) has been sampled yet. Otherwise return the previously sampled value.

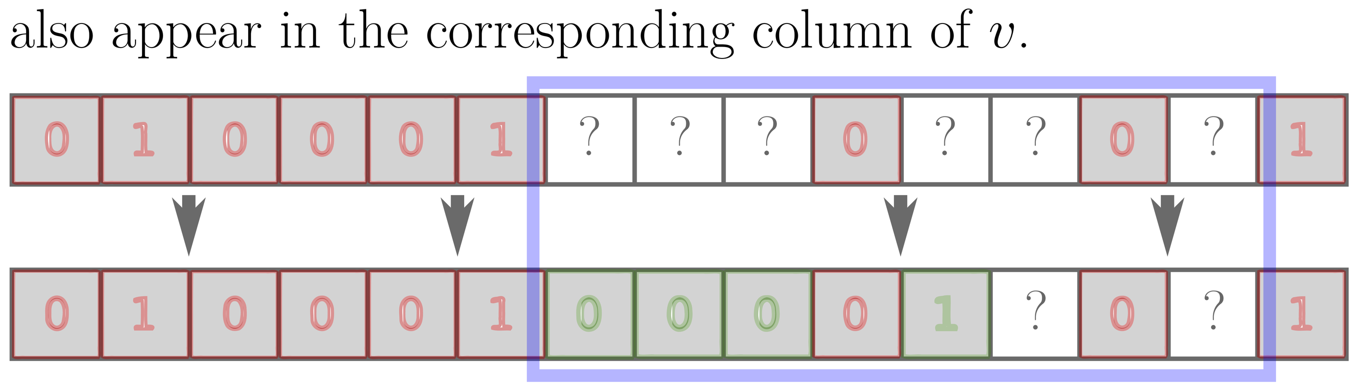
Find Next-Neighbor (*skip-sampling*)

Adjacency List query: Return neighbors of v in order.

$$\mathbb{P}[k \text{ non-neighbors before next-neighbor}] = (1-p)^k \cdot p$$

We can sample from this distribution in $\tilde{\mathcal{O}}(1)$ time [cite]. Essentially, we avoid sampling each 0 separately.

Issue: Adjacency matrix is symmetric. So, each zero must also appear in the corresponding column of v.



If the sampled neighbor is a 0, discard and resample.

Cannot afford too many re-samplings.

Bucketing Approach & *Random-Neighbor* Queries

- Divide each row of the adjacency matrix into contiguous buckets
- Expected number of neighbors in a bucket is $\Theta(\log n)$
- Each vertex v is associated with buckets $\langle B_1^v, B_2^v, B_3^v, \cdots\rangle$
- An **unfilled** bucket may contain some indirectly exposed neighbors
- A **filled** bucket will contain every possible sampled neighbor

Filling the i^{th} bucket B_i^v of vertex v

- Use skip-sampling to produce a **potential** *next-neighbor* u of v in B_i^v
- Check if (u,v) was set to 0, by looking at bucket $\mathcal{B}$ of u containing v
- If so, re-sample. Otherwise, mark u as a neighbor of v, and update $\mathcal{B}$
- W.h.p, only $\mathcal{O}(\log^2 n)$ **potential** neighbors are generated in B_i^v

Random-Neighbor(v)

- Choose a random bucket $\mathcal{B}$ of v. If the $\mathcal{B}$ is **unfilled**, fill it.
- If k neighbors found in $\mathcal{B}$, start over (reject) with probability $1 - k/M$.
- If accepted, return an uniformly random neighbor found in $\mathcal{B}$.

For $M = \mathcal{O}(\log^2 N)$, the max number of neighbors in any bucket is $< M$. So, the number of rejection sampling rounds is $\mathcal{O}(\log^2 N)$ in expectation.

Stochastic Block Model

- Each vertex is assigned to some community $C_i \subseteq V$ for $i \in [r]$
- Communities $\{C_i\}_{i\in[r]}$ partition V: if $u \in C_i, v \in C_j$, then $\mathbb{P}_{(u,v)\in E} = p_{ij}$

Given sizes of each comunity C_i and a range of length ℓ

- Count number of occurrences of each community in any contiguous range
- Sample from *Multivariate Hypergeometric Distribution*

$$\Pr[\mathbf{S}_\ell^{\mathbf{C}} = \langle s_1,\ldots,s_r\rangle] = \frac{\binom{C_1}{s_1}\cdot\binom{C_2}{s_2}\cdots\binom{C_r}{s_r}}{\binom{B}{\ell}} \quad \text{where } \ell = \sum_{i=1}^r s_i \text{ and } B = \sum_{i=1}^r C_i$$

Sampling the Multivariate Hypergeometric Distribution

[cite] solves the special case of $r = 2$ and $B = 2\ell$.

- **Extending to** $B \neq 2\ell$: Divide ℓ into $\mathcal{O}(\log n)$ dyadic segments.
- **Extending to** $r > 2$: Make a tree with r leaves (one for each C_i). Every branch down the tree is equivalent to a $r = 2$ splitting.

Work in Progress

Dimer Model: Domino Tiling

Dimer Model: Domino Tiling

Glossary

Some Necessary and Useful Vocabulary