Constrained MCMC Algorithms for ERG models

Duy Vu and David Hunter

Constraints

- ergm uses MCMC to handle the normalization constant in ML estimation of ERG models.
- The need of generating graphs randomly conditioning on some network statistics:
  - implicitly such as the number of nodes
  - explicitly by specifying the constraint option
- Our current focus is on such explicit constraints:
  - conditioning on the vertex-degrees
  - conditioning on the degree distribution
  - conditioning on some soft constraints

Conditioning on the vertex-degrees

- Snijder (1991), Rao et al. (1996), Roberts (2000), McDonald et al. (2007), Verhelst (2008): randomly select an alternating rectangle (tetrad) or a compact alternating hexagon (hexad) and form a proposed network by toggling the edges on the rectangle or the hexagon.

1	0
0	1

0	1
1	0

Tetrad

X	0	1
1	X	0
0	1	X

Hexad

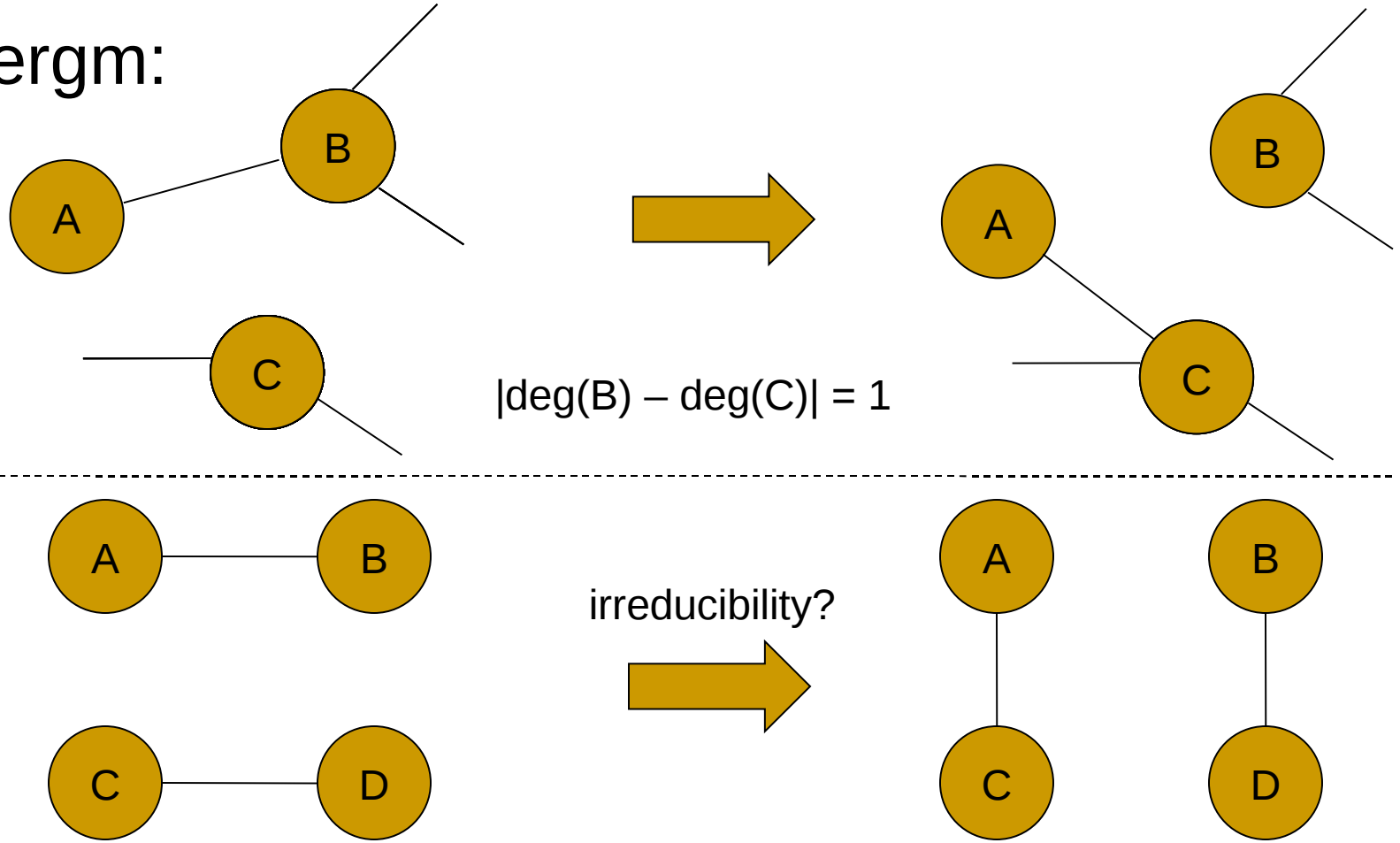
X	1	0
0	X	1
1	0	X

Conditioning on the vertex-degrees

- ergm implements McDonald et al. (2007) which works on both directed and undirected graphs.
- Verhelst (2008) claims to have uniform stationary distribution and faster convergence by combining
  - bigger moves, i.e. more complicated transformation, through the sample space.
  - importance sampling on selecting moves from the neighborhood.
- TO-DO: check the current implementation and add Verhelst's proposal to ergm.

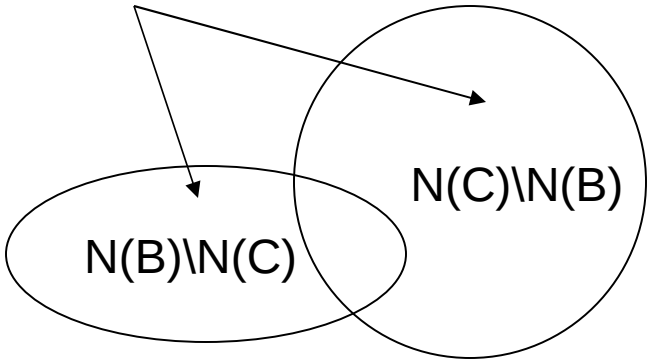
Conditioning on the degree distribution

- ergm:



Conditioning on the degree distribution

- Some potential suggestions:
  - combine tetrad + hexad toggles with switching degrees by swapping all neighbors.
  - combine tetrad + hexad toggles with switching degrees by swapping some neighbors.



- TO-DO: check their irreducibility, efficiency, and figure out their stationary distributions.

Conditioning on some soft constraints

- The fixed vertex-degrees and degree distributions are hard constraints which can be implemented by direct MH proposals above.
- How about some *soft* constraints such as the triangles, nodematch("Grade"), or nodematch("Sex")?
  - The main goal is to search for such graphs satisfying the constraints. We are not try to draw those graphs uniformly.
- We can combine a simulated annealing search with the above MH proposals so that only proposals whose constrained statistics are *close* to the target values are returned.

Conditioning on some soft constraints

```
MCMCSample() {
…
proposed_net = MHp.propose(current_net, constraints)
…
new_net = accept_reject(proposed_net)
…
}
```

```
proposed_net = SA.search(current_net, MHp,
   constraints, targets)
```

- TO-DO: devise temperature schedules, check the quality of constraint satisfaction and the efficiency.

Suggestion and Questions

Thank you!