# Synthetically Generating Hypergraph Chung-Lu and BTER inputs

Sinan Aksov

June 1, 2018

## 1 What we need to generate

We need to synthetically generate four inputs for BTER, the first two of which serve as inputs for Chung-Lu:

- 1.  $dd_V$ : a vector of non-negative ints of length equal to the maximum vertex degree (i.e. largest row sum in the incidence matrix S).
- 2.  $dd_E$ : a vector of non-negative ints of length equal to the maximum edge cardinality (i.e. largest column sum in the incidence matrix S).
- 3.  $\operatorname{mpd}_V$ : a vector of numbers, each in the interval [0,1], of length equal to the maximum vertex degree.
- 4.  $\operatorname{mpd}_E$ : a vector of numbers, each in the interval [0,1], of length equal to the maximum edge cardinality.

#### Input constraints

- $\operatorname{length}(\operatorname{dd}_V) = \operatorname{length}(\operatorname{mpd}_V)$ , and  $\operatorname{length}(\operatorname{dd}_E) = \operatorname{length}(\operatorname{mpd}_E)$
- $\sum dd_V(i) \cdot i = \sum dd_E(i) \cdot i$ 
  - This is a necessary condition for  $dd_V$  and  $dd_E$  to be realizable by some hypergraph. This condition says that the number of vertex-hyperedges inclusions must be the same as the number of hyperedge-vertex containments (or put another way, the sum of the row-sums of the incidence matrix is the same as the sum of the column-sums of the incidence matrix).

## 2 How we generate these

We adapt the guidelines for synthetic input generation given by the creators of (non-hypergraph) BTER (link) to the hypergraph case as follows:

Step 1 By randomly sampling from a log-normal degree distribution, we generate  $dd_V$  and  $dd_E$  so that (in expectation):

$$\sum dd_V(i) \cdot i = \sum dd_E(i) \cdot i = p|V||E|,$$

where  $p = \frac{\log(|V| + |E|)}{|V| + |E|}$ . Since these  $dd_V$  and  $dd_E$  are randomly drawn, equality won't necessarily hold in the above equation. We account for this in Step 2 below.

- Step 2 Assuming, without loss of generality, that  $\sum dd_V(i) \cdot i < \sum dd_E(i) \cdot i$ , we increase the desired degree of a randomly chosen vertex from  $dd_V$  until equality holds in the above equation.
- Step 3 We generate  $\operatorname{mpd}_V$  and  $\operatorname{mpd}_E$  according to the following idealized curve:

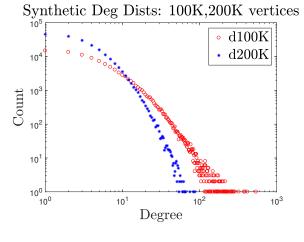
$$\mathrm{mpd}_V(d) = m_{\mathrm{max}} \exp\left(-(d-1) \cdot \xi\right),\,$$

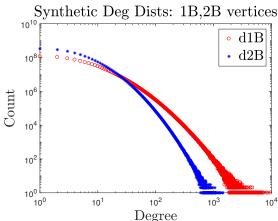
where  $2 \le d \le \text{length}(\text{dd}_V)$ , and  $m_{\text{max}}$  denotes the target maximum per-degree metamorphosis coefficient and  $\xi$  is a user specified parameter. Using the aforementioned software, (link), one can determine the optimal  $\xi$  based on  $m_{\text{max}}$  and a target global metamorphosis coefficient, which (based on values observed in real data) we set to 0.1 and 0.05, respectively.

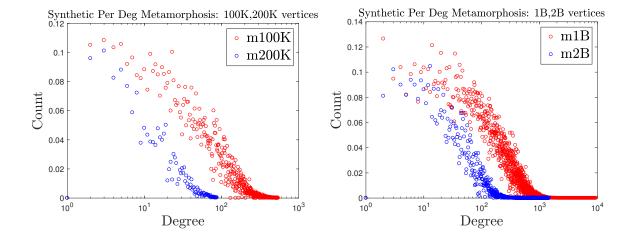
Step 4 To make these per-degree metamorphosis coefficients more realistic, we add some noise/randomness so that the final metamorphosis coefficients are chosen as

$$\mathcal{N}(\mathrm{mpd}(d), \min\{0.01, \mathrm{mpd}(d)/2\})$$

#### Plots of Some Synthetically Generated Inputs:







## 3 Input sizes and how to use them

Data Generated		
Size	Inclusions	
100K, 200K	841,088	
1M, 2M	9,940,947	
10M, 20M	114,855,148	
100M, 200M	1,301,294,319	
1B, 2B	14,548,044,386	

The degree distributions inputs are named: d100K.csv, d200K.csv, ... and the per-degree metamorphosis coefficients are named: m100K.csv, m200K.csv, ... Note that the inputs cannot be mixed and matched if they specify different numbers of inclusions, however the same input can be used for both vertices and hyperedges. For example:

#### Valid Use of Inputs

$$\begin{aligned} \mathbf{CL}(\mathrm{dd}_V,\mathrm{dd}_E) &= \mathbf{CL}(\mathrm{d}100\mathrm{K},\mathrm{d}200\mathrm{K}) \\ \mathbf{CL}(\mathrm{dd}_V,\mathrm{dd}_E) &= \mathbf{CL}(\mathrm{d}1\mathrm{M},\mathrm{d}1\mathrm{M}) \\ \mathbf{BTER}(\mathrm{dd}_V,\mathrm{dd}_E,\mathrm{mpd}_V,\mathrm{mpd}_E) &= \mathbf{BTER}(\mathrm{d}1\mathrm{B},\mathrm{d}2\mathrm{B},\mathrm{m}1\mathrm{B},\mathrm{m}2\mathrm{B}) \end{aligned}$$

#### **Invalid** Use of Inputs

$$\begin{aligned} \mathbf{CL}(\mathrm{dd}_V,\mathrm{dd}_E) &= \mathbf{CL}(\mathrm{d}100\mathrm{K},\mathrm{d}2\mathrm{M}) \\ \mathbf{BTER}(\mathrm{dd}_V,\mathrm{dd}_E,\mathrm{mpd}_V,\mathrm{mpd}_E) &= \mathbf{BTER}(\mathrm{d}1\mathrm{B},\mathrm{d}2\mathrm{B},\mathrm{m}2\mathrm{B},\mathrm{m}2\mathrm{B}) \end{aligned}$$

# 4 MATLAB run times

Model(inputs)	Runtime
$\mathbf{CL}(d100\mathrm{K}, d200\mathrm{K})$	0.4 s
$\mathbf{BTER}(\mathrm{d}100\mathrm{K},\mathrm{d}200\mathrm{K})$	$0.9 \mathrm{\ s}$
$\mathbf{CL}(\mathrm{d1M},\mathrm{d2M})$	8 s
$\mathbf{BTER}(\mathrm{d1M},\mathrm{d2M})$	11 s
$\mathbf{CL}(\mathrm{d}10\mathrm{M},\mathrm{d}20\mathrm{M})$	190 s
$\mathbf{BTER}(\mathrm{d}10\mathrm{M},\mathrm{d}20\mathrm{M})$	$169 \mathrm{\ s}$