Modeling Graphs Using a Mixture of Kronecker Models

Suchismit Mahapatra
suchismi@buffalo.edu

Varun Chandola
chandola@buffalo.edu

Department of Computer Science & Engineering

University at Buffalo The State University of New York

Generative models for graphs

- Allow us to generate synthetic graphs which closely capture the properties of real world graphs.
- Should be ideally parametric which allow for learning to be able to generate graphs of arbitrary size.
- Should be able to scale to massive graph sizes.

Why generative models for graphs ?

- Limited availability of real world graph data, mainly due to high cost and privacy concerns.
- Allow us to extrapolate/produce realistic simulations at a desired scale.
- Provide anonymity.
- Allow researchers to simulate/understand “flow”.
- Enable us to study how graphs grow over time.

Kronecker Product based Graph Models (KPGM)

- Parametric, uses seed matrices.

$$\mathcal{P}_1 = \Theta = \begin{bmatrix} \theta_{11} & \theta_{12} \\ \theta_{21} & \theta_{22} \end{bmatrix}$$

- Can effectively model the structure of real networks and model network properties.
- Multiplicative nature of the model allows for **fast sampling** of massive sized graphs.

$$\mathbf{C} = \mathbf{A} \otimes \mathbf{B} \doteq \begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \dots & a_{1,m}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \dots & a_{2,m}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\mathbf{B} & a_{n,2}\mathbf{B} & \dots & a_{n,m}\mathbf{B} \end{pmatrix}$$

$$\mathcal{P}_n = \underbrace{\mathcal{P}_1 \otimes \mathcal{P}_1 \otimes \dots \otimes \mathcal{P}_1}_{n \text{ times}}$$

note: Images taken from [12]

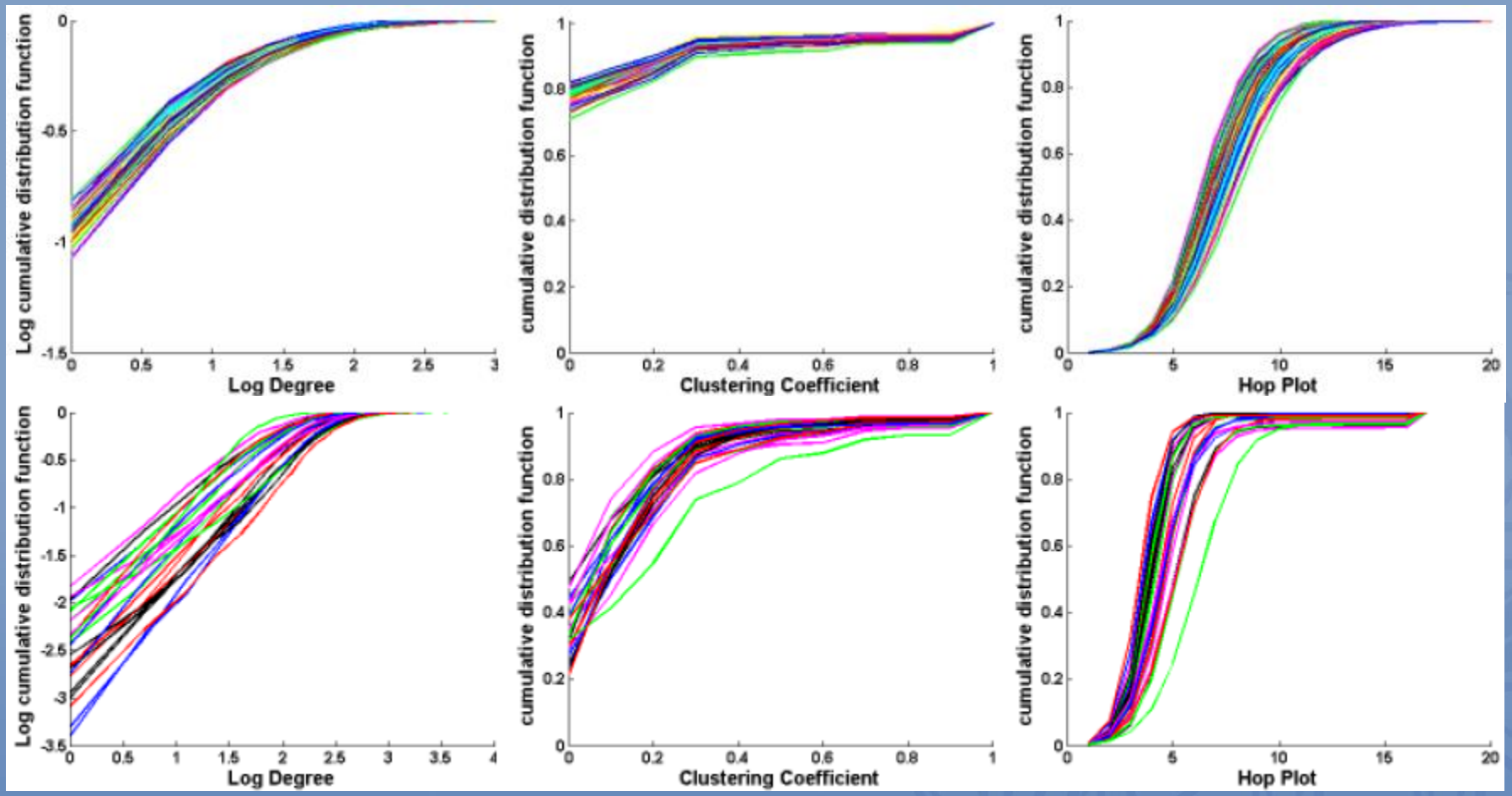
Other generative model variants

- **Erdos-Renyi (ER)** [18] – earliest model, fails to capture properties of real-world graphs.
- **Exponential Random Graph Models (ERGM)** [25] – stochastic log linear model.
- **Stochastic Block Models (SBM)** [24] – based on clusters and memberships of nodes to each.
- **Chung-Lu (CL)** [1] - captures degree distribution.
- **Block Two-Level Erdos-Renyi (BTER)** [20] - match degree distribution, clustering coefficient, not "truly" generative.

Issues with KPGM based models

- Lack the **ability to capture the natural variability observed in real world graphs**.
  - Synthetic graphs sampled from KPGM show little variation in terms of several graph properties.
- Seshadri et al. [21] have shown that graphs generated from KPGM have 50-75% isolated vertices.
- Tied-KPGM (*t*KPGM), mixed-KPGM (*m*KPGM) [14] models proposed to alleviate the issues
  - Not expressive enough.

Issues with KPGM based models



note: Images taken from [12]

Variance in population of graphs

- Real world graphs can be thought of as being generated from a natural process.
- Examples include :-
  - graphs collected at different times i.e. snapshots of graphs.
  - social networks for different groups of people (e.g., schools)
  - healthcare networks for different spatial regions.
  - road networks etc.
- **Populations of graphs generated by the same process exhibit a natural variance in terms of the structural properties.**

Variance in population of graphs

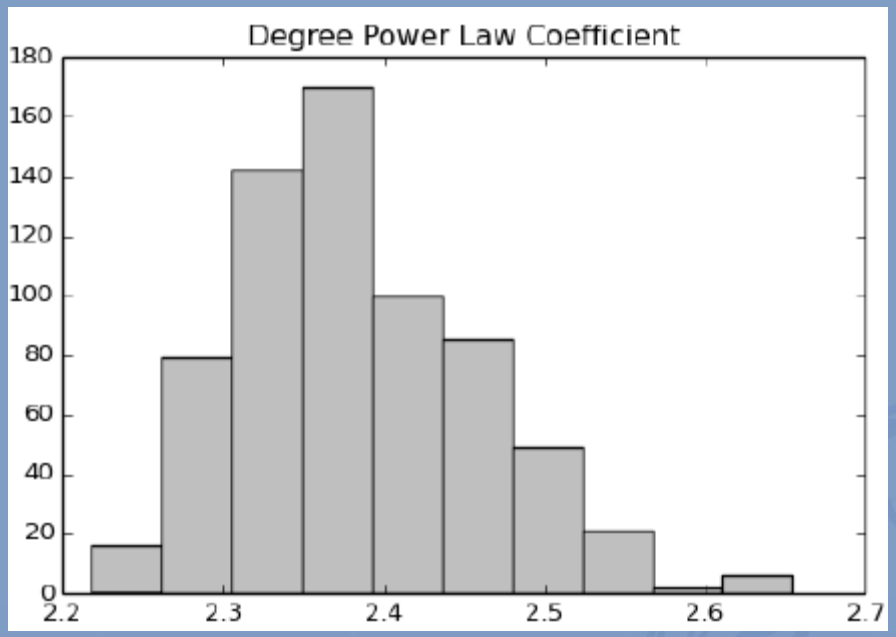


Illustration of the variance in power law coefficient for a population of over 700 Autonomous Systems (AS) graphs sampled at different time points.

The *x*KPGM model

- Employs a mixture-model based approach which allows it to capture the variance in graphs.
- Uses **two or more** initiator matrices of possibly different sizes
- A **k**-length vector **π** which defines the mixing probabilities and a level tying parameter **l**.

Algorithm 1 Graph Generation Algorithm for *xKPGM*

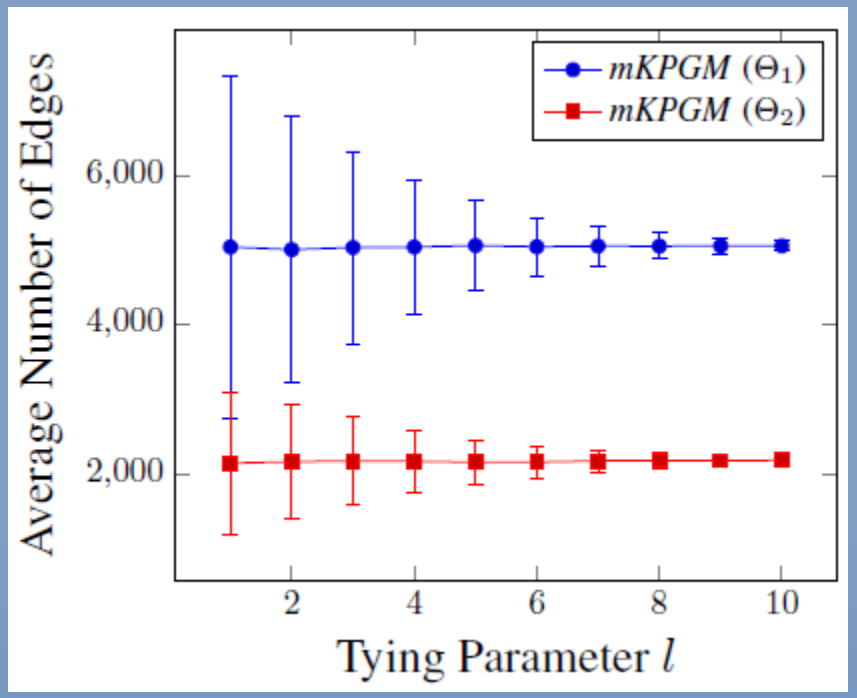
Input: $\Theta_1, \Theta_2, \ldots, \Theta_k, \pi, n, l$
Output: Adjacency matrix A

$\mathcal{P} \leftarrow 1$
`// Untied Phase`
foreach $t = 1$ *to* l **do**
  $i \sim Multinomial(\pi)$
  $\mathcal{P} \leftarrow \mathcal{P} \otimes \Theta_i$

`// Tied Phase`
foreach $t = l + 1$ *to* n **do**
  $A \leftarrow R(\mathcal{P})$; `// R - Realization`
  $i \sim Multinomial(\pi)$
  $\mathcal{P} \leftarrow A \otimes \Theta_i$

$A \leftarrow R(\mathcal{P})$
return A

The *x*KPGM model



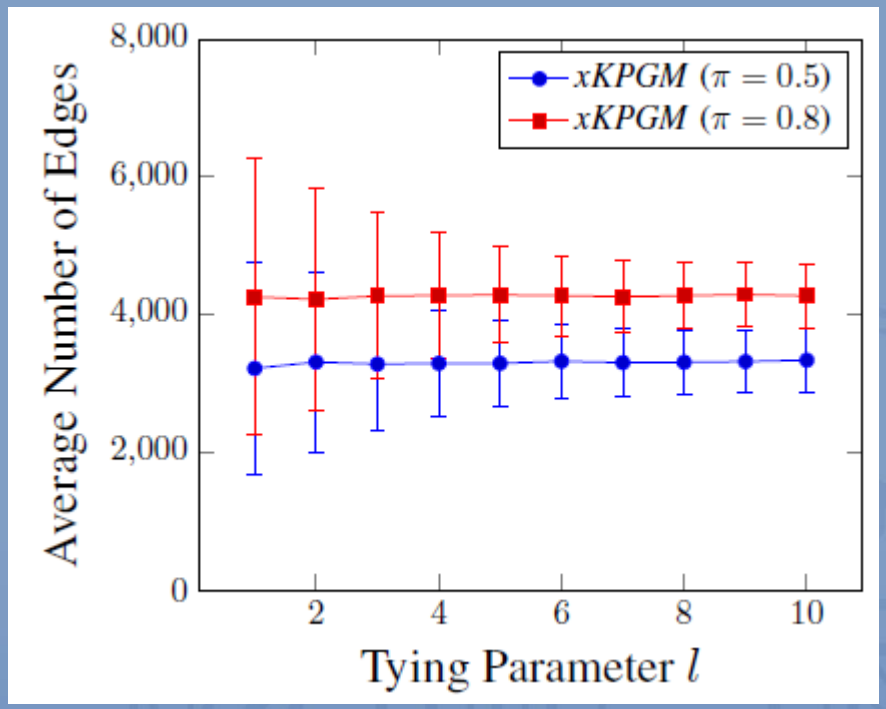


Illustration for the variation in # of edges for synthetic graphs generated by mKPGM and xKPGM for varying levels of tying (l).

*x*KPGM – A generic model

- Different KPGM based models are specific instances of the *x*KPGM model:-
  - k = 1 and l = 1, *x*KPGM reduces to ***t*KPGM**.
  - k = 1 and l = n, *x*KPGM reduces to **KPGM**.
  - k = 1 and 1 ≤ l < n, *x*KPGM reduces to ***m*KPGM**.

How the different models stack up

	PA [3]	ERGM [25]	CL [1]	*BTER* [20]	*KPGM* [12]	*mKPGM* [16]	*xKPGM*
1. *Learnable*	×	√	See note	See note	√	√	√
2. *Scalable Learning*	–	√	–	–	√	×	√
3. *Scalable Generation*	×	×	√	√	√	√	√
4. *Match Local Properties*	×	×	×	√	√	√	√
5. *Capture Variance*	×	×	×	×	×	√	√

- CL and BTER models do not allow generation of arbitrary sized synthetic graphs.

Learning Parameters

- Employs a method of moments approach to learn parameters
  - Initiator matrices and mixing probability vector
- Each graph is represented as a set of *moments*
  - Number of edges, triangles, hairpins, etc.
- We derive analytical expressions for expected value of each moment as a function of the parameters
- Find parameters that best fit the expected values

Analytical Expression for Moments

$2\mathbb{E}[\mathrm{E}]$	$=$	$\prod_{i=1}^{n}(a_i+2b_i+c_i)^{\pi_i n}-\prod_{i=1}^{n}(a_i+c_i)^{\pi_i n}$
$2\mathbb{E}[\mathrm{H}]$	$=$	$\prod_{i=1}^{n}((a_i+b_i)^2+(b_i+c_i)^2)^{\pi_i n}-2\prod_{i=1}^{n}(a_i(a_i+b_i)+c_i(c_i+b_i))^{\pi_i n}-\prod_{i=1}^{n}(a_i^2+2b_i^2+c_i^2)^{\pi_i n}+2\prod_{i=1}^{n}(a_i^2+c_i^2)^{\pi_i n}$
$6\mathbb{E}[\mathrm{T}]$	$=$	$\prod_{i=1}^{n}((a_i+b_i)^3+(b_i+c_i)^3)^{\pi_i n}-3\prod_{i=1}^{n}(a_i(a_i+b_i)^2+c_i(b_i+c_i)^2)^{\pi_i n}$
	$-$	$3\prod_{i=1}^{n}(a_i^3+c_i^3+b_i(a_i^2+c_i^2)+b_i^2(a_i+c_i)+2b_i^3)^{\pi_i n}+2\prod_{i=1}^{n}(a_i^3+2b_i^3+c_i^3)^{\pi_i n}$
	$+$	$5\prod_{i=1}^{n}(a_i^3+c_i^3+b_i^2(a_i+c_i))^{\pi_i n}+4\prod_{i=1}^{n}(a_i^3+c_i^3+b_i(a_i^2+c_i^2))^{\pi_i n}-6\prod_{i=1}^{n}(a_i^3+c_i^3)^{\pi_i n}$
$6\mathbb{E}[\Delta]$	$=$	$\prod_{i=1}^{n}(a_i^3+3b_i^2(a_i+c_i)+c_i^3)^{\pi_i n}-3\prod_{i=1}^{n}(a_i(a_i^2+b_i^2)+c_i(b_i^2+c_i^2))^{\pi_i n}+2\prod_{i=1}^{n}(a_i^3+b_i^3)^{\pi_i n}$

Moments are derived using the “permutation trick”

The "Permutation Trick"

- A graph generated using an arbitrary sequence of initialization matrices is equivalent to the following canonical sequence: $\underbrace{(\Theta_1 \otimes \Theta_1 \ldots)}_{n_1 \text{ times}} \otimes \underbrace{(\Theta_2 \otimes \Theta_2 \ldots)}_{n_2 \text{ times}} \otimes \ldots \otimes \underbrace{(\Theta_k \otimes \Theta_k \ldots)}_{n_k \text{ times}}$
- For two matrices A and B: $A \otimes B = M(B \otimes A)N$

  where M and N are permutation matrices.
- This helps in finding the exact expressions for moments

Parameter estimation for xKPGM

- The estimation method searches for parameters θ_1, θ_2, θ_3,… ,π which minimizes

$$f(\Theta, \mathbf{F}^*) = \sum_{i=1}^{|\mathbf{F}|} w_i \left(\frac{F_i^* - \mathbb{E}[F_i|\Theta]}{F_i^*} \right)^2$$

- The aim is to find model parameters for which the expected moments for the model match closely with the moments computed from the observed graph.
- Can be extended to learn from multiple graph instances.

Experimental Setup

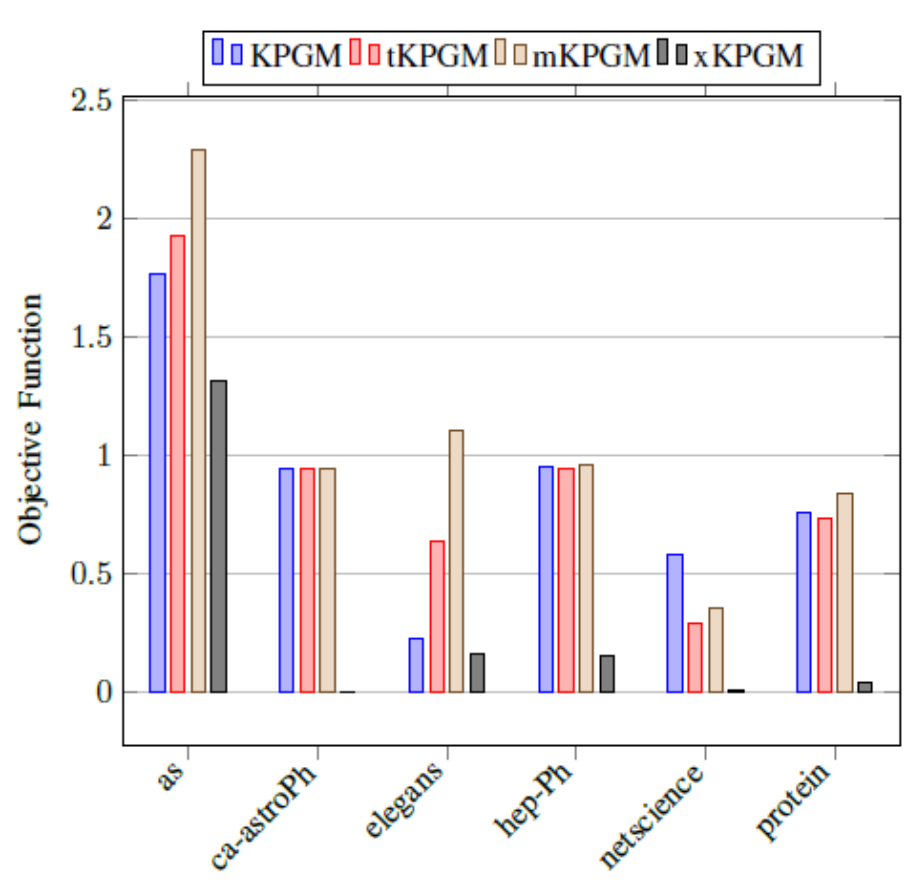
Data sets used are publicly available data graph sets :-

Name	Description	Nodes	Edges
as [23]	CAIDA AS Relationship Graph	6,474	13,233
ca-astroPh [23]	Collaboration network of Arxiv Astro Physics	18,772	396,160
elegans [6]	C. elegans metabolic network	453	4,596
hep-ph [23]	Citation network from Arxiv HEP-PH	34,546	421,578
netscience [17]	Coauthorship network of scientists	1,589	5,484
protein [10]	Protein interaction network for Yeast	1,870	4,480

Experimental Setup

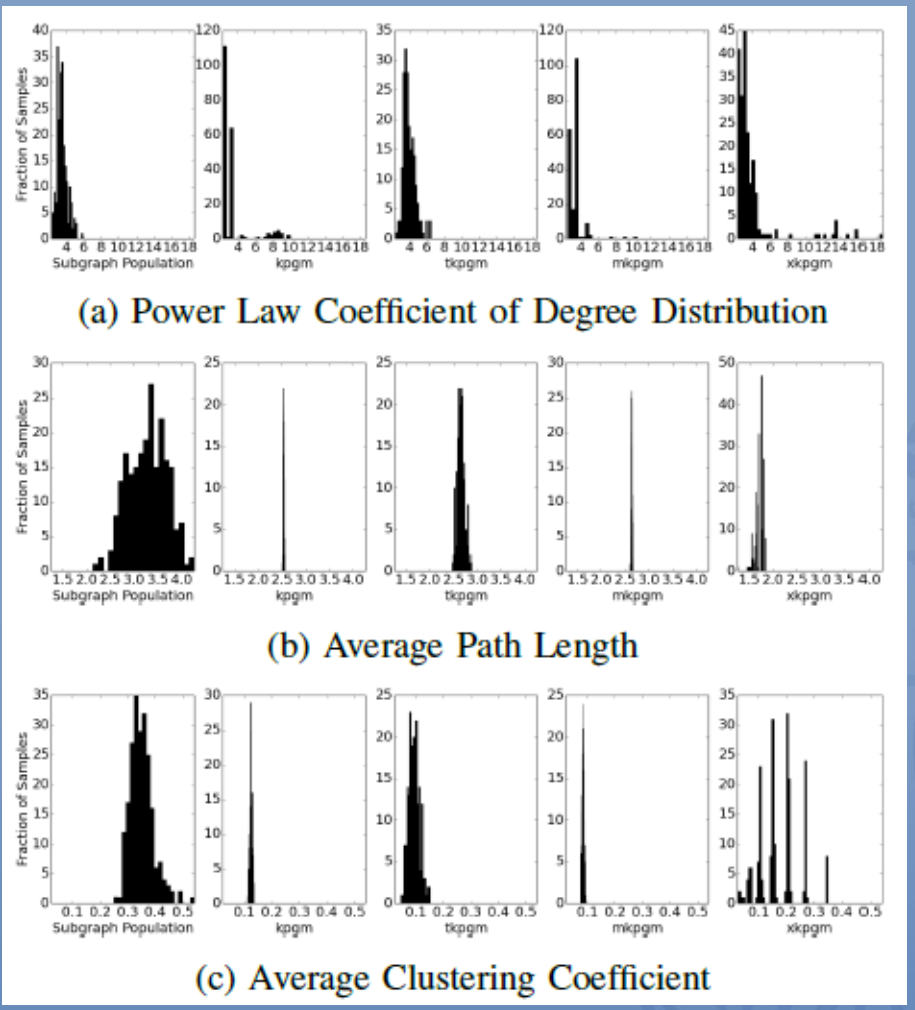
- Employed a variant of the forest fire model to generate 200 subgraphs from the real world graphs and measured characteristics of the subgraphs.
- For each model, we used the estimated parameters and generate 200 samples of appropriate sizes.
- To evaluate our model we used salient characteristics of graphs :-
  - Power law co-efficient
  - Average path length
  - Average clustering co-efficient
  - # of edges
  - # of triangles

Results - Matching moments



Comparing xKPGM with other models in terms of the objective function value obtained after training.

Results - Capturing variance

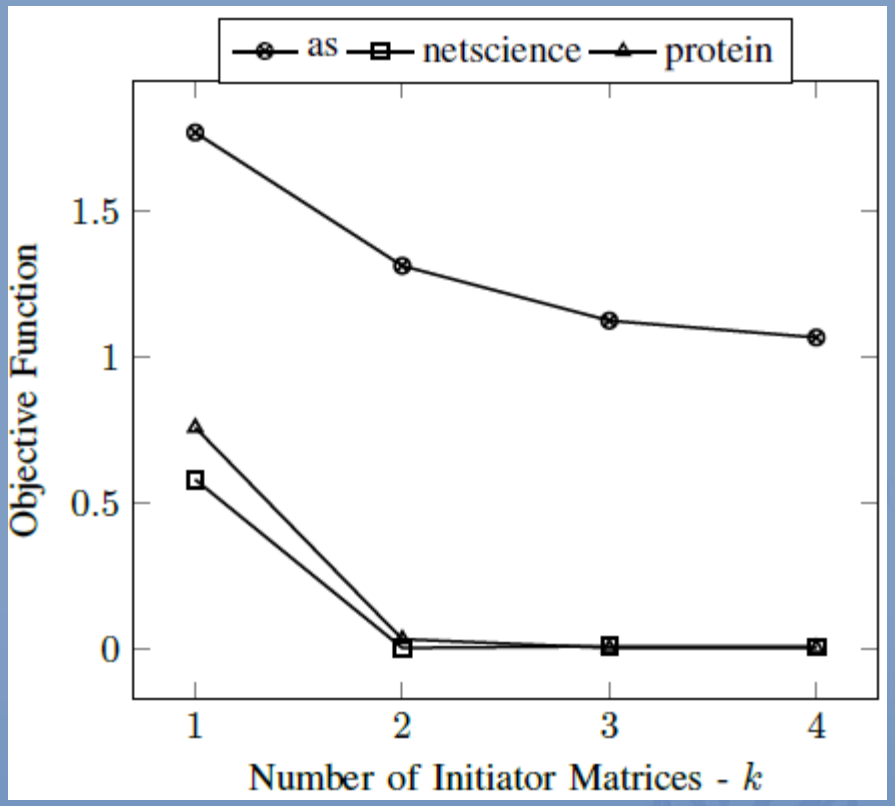


(a) Power Law Coefficient of Degree Distribution

(b) Average Path Length

(c) Average Clustering Coefficient

Results - Impact of # of seed matrices



Performance of *x*KPGM using a different number of initiator matrices for three different data sets.

Summary

- *x*KPGM, the proposed generative model induces robust variability for multiple graph features while retaining the strong capabilities of KPGM, i.e. scaling to massive graphs.
- Using the method of moments approach allows for scalable learning.
- *x*KPGM outperforms state of art methods both in terms of matching the graph properties and the variance in the population.

Future work

- Seshadri et al. [21] have demonstrated that graphs generated from KPGM have 50-75% isolated vertices.
  - Highly undesirable, need to address this.
- Currently we are using hairpins, tri-pins, triangle counts as our moments.
  - Can we find "better" moments which are more representative of substructure in graphs ?
- Kronecker products lend themselves beautifully to graph substructure clustering.
- Twitter etc. have multiple modes of operation – information and social networks within them.