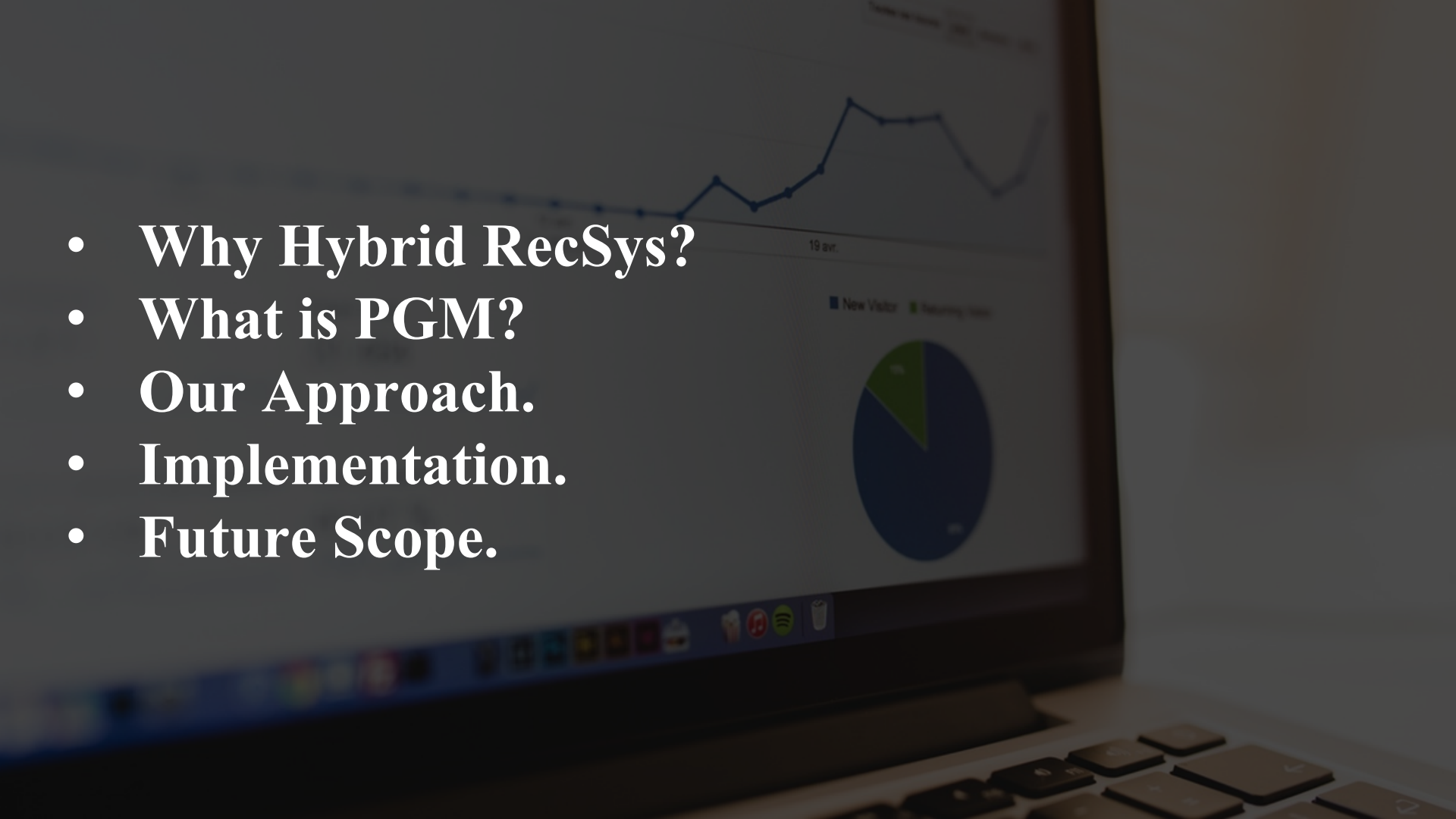
Hybrid RecSys using Probabilistic Graphical Model

Tuhin Sharma
Impel Labs
tuhinsharma121@gmail.com



- **Why Hybrid RecSys?**
- **What is PGM?**
- **Our Approach.**
- **Implementation.**
- **Future Scope.**

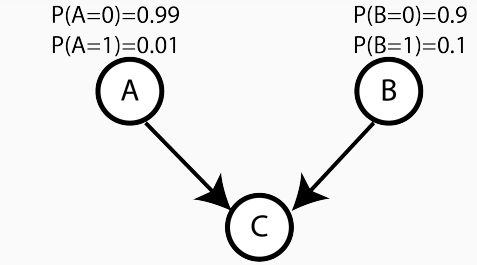
Why Hybrid Recommender System?

- Content Based
  - No scope of serendipity
  - Over-specialization
  - Does not use the interaction information between users

- Collaborative Filtering
  - Content is *unpopular*.
  - Cold start problem.

What is PGM?

Probabilistic Graphical Model (PGM) expresses the conditional dependency structure between random variables.



Conditional Probabilities

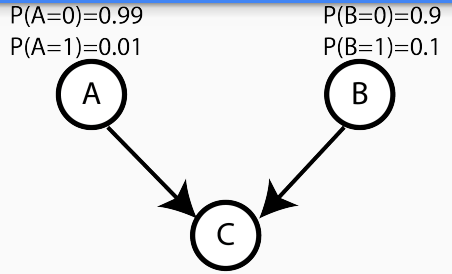
	A = 0		A=1	
	B=0	B=1	B=0	B=1
C = 0	0.9	0.5	0.4	0.1
C = 1	0.1	0.5	0.6	0.9

Use to infer

Unconditional Probabilities

A	B	C	prob
0	0	0	0.8019
0	0	1	0.0891
0	1	0	0.0495
0	1	1	0.0495
1	0	0	0.0036
1	0	1	0.0054
1	1	0	0.0001
1	1	1	0.0009

Probabilistic Graphical Model



Conditional Probabilities

	A = 0		A=1	
	B=0	B=1	B=0	B=1
C = 0	0.9	0.5	0.4	0.1
C = 1	0.1	0.5	0.6	0.9

Use to infer

Unconditional Probabilities

A	B	C	prob
0	0	0	0.8019
0	0	1	0.0891
0	1	0	0.0495
0	1	1	0.0495
1	0	0	0.0036
1	0	1	0.0054
1	1	0	0.0001
1	1	1	0.0009

$$P(A,B,C) = P(A) * P(B) * P(C|A,B)$$

$$P(B=1|C=1)$$

$$= \frac{P(B=1,C=1)}{P(C=1)}$$

$$= \frac{P(A=0,B=1,C=1)+P(A=1,B=1,C=1)}{P(A=0,B=0,C=1)+P(A=1,B=0,C=1)+P(A=0,B=1,C=1)+P(A=1,B=1,C=1)}$$

$$= \frac{0.0495+0.0009}{0.0891+0.0054+0.0495+0.0009} = 0.3478$$

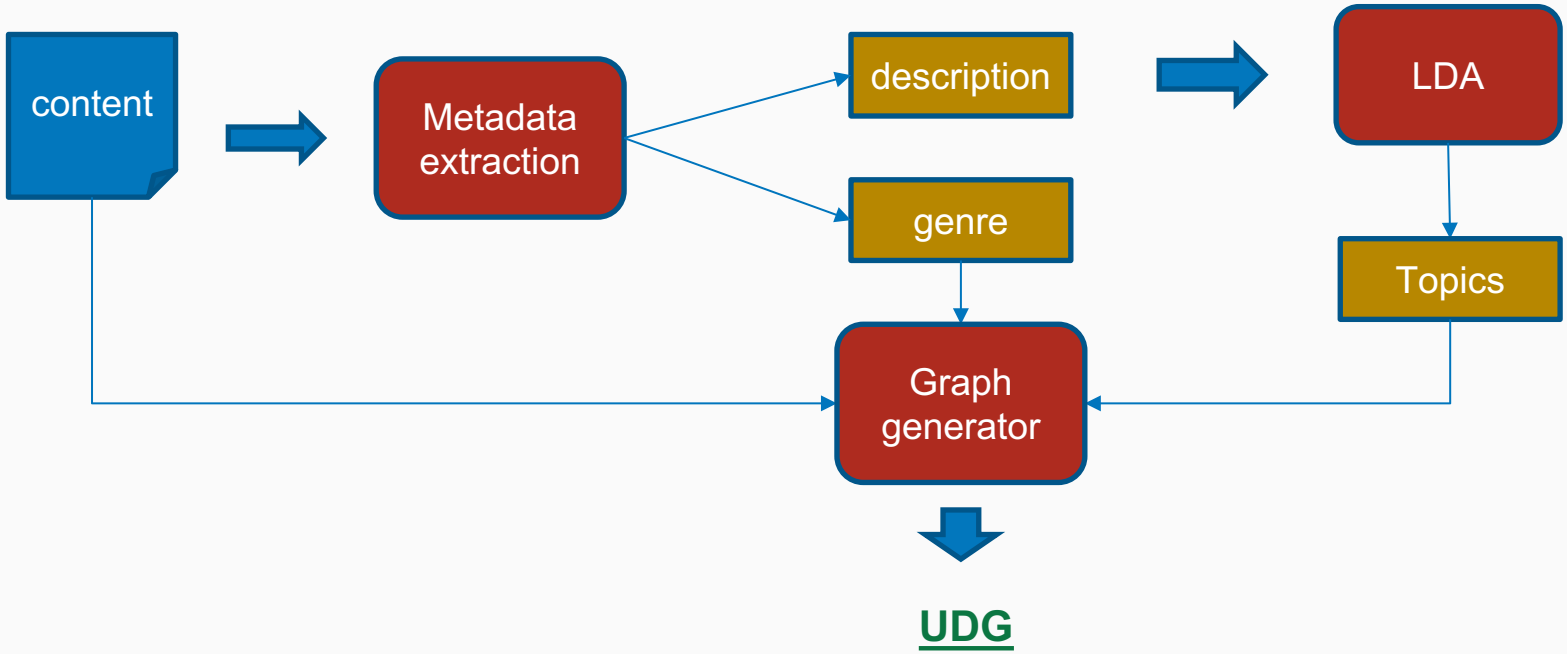
Why we need PGM?

- Latent similarity and dependency between genres/categories.
- Easy to explain recommendation.
- Handles cold-start problem.
- Easy to add new items/contents.

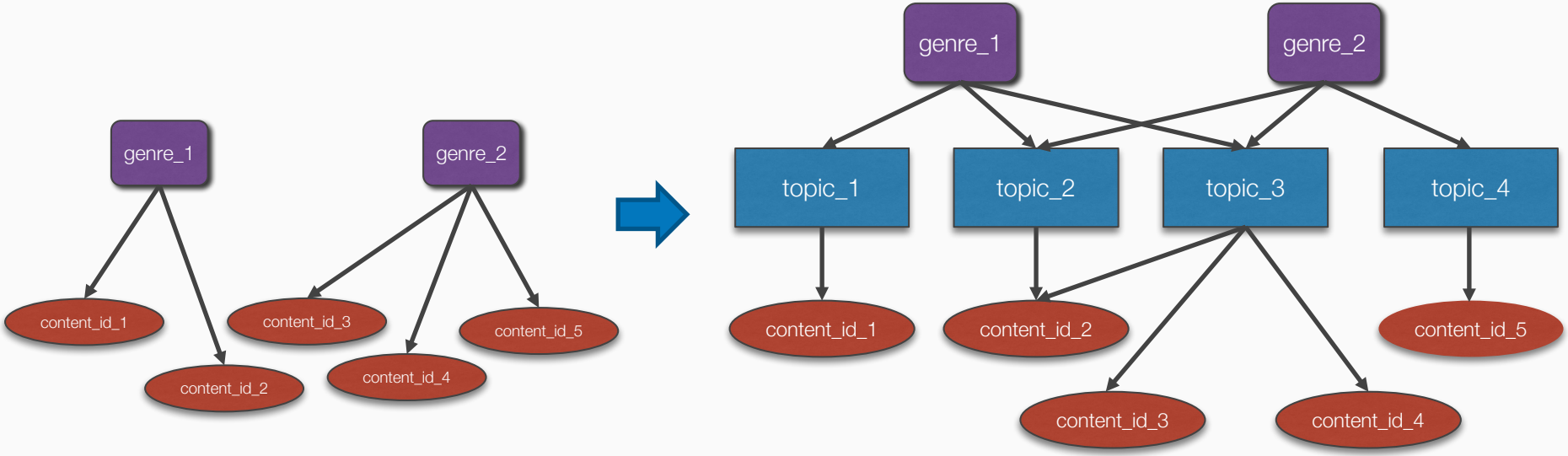
Approach

- Unweighted dependency graph (UDG)
  - LDA (Latent Dirichlet Allocation)
  - Content Based
- Probabilistic Graphical Model
  - Co-occurrence Matrix
  - Bayesian Network
  - Collaborative Based

Unweighted dependency graph (UDG)



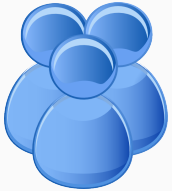
UDG and PGM (Approach contd.)



Explicit Connections

Derived Connections

Real Life example



Users

News Videos

John

Watched Videos

John

Implementation

- Vertically Scalable
  - Pgmpy
  - Pymc
  - Libpgm
  - ***Pomegranate***
- Horizontally Scalable
  - Edward on Tensorflow.
  - Pymc3 on Theano

Available at https://github.com/tuhinsharma/recsys-pgm/blob/master/hybrid-pgm-recsys.ipynb

Performance of Pomegranate

content count (number of nodes in PGM)	Number of user viewership	Model Size	Training Time	Prediction Time
20K	5K	18.3 MB	508 sec	4 sec
30K	5K	27.4 MB	749 sec	9 sec
40K	5K	36.6 MB	1049 sec	16 sec
50K	5K	45.8 MB	1347 sec	26 sec
200K	5K	183.4 MB	5833 sec / 1.6 hrs	~2 min
300K	5K	274.4 MB	10732 sec / 2.9 hrs	~3 min
400K	5K	366.5 MB	18901 sec / 5.3 hrs	~5 min

CPU – 8, Memory – 16 GB

Summary

- Size of CPT for node *n* with *m* number of parents is 2^{m-1}.
  - Ontology creation should be ***smart*** enough.
- DAG (Directed Acyclic Graph)
  - Bayesian Belief Propagation. (**pomegranate**)
- Scalability
  - Variational inference. (**edward**)

Thank You

Tuhin Sharma
Impel Labs
tuhinsharma121@gmail.com