ChiSeL:

Graph Similarity Search using Chi-Squared Statistics in Large Probabilistic Graphs

Shubhangi Agarwal[†], Sourav Dutta[††], Arnab Bhattacharya[†]

sagarwal@cse.iitk.ac.in, sourav.dutta2@huawei.com, arnabb@cse.iitk.ac.in

August, 2020

VLDB 2020, Tokyo, Japan (online)

[†]Department of Computer Science and Engineering,
Indian Institute of Technology Kanpur,
INDIA
[††]Huawei Research Centre, Dublin,
IRELAND

Introduction

Subgraph querying in large real-world graphs is challenging due to *uncertainty* of data.

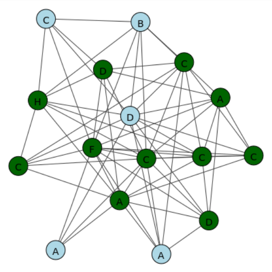


FIGURE 1: GRAPH (WITH QUERY MANIFESTATION)

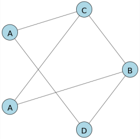


FIGURE 2: QUERY

- Subgraph isomorphism is NP-complete

Introduction

Subgraph querying in large real-world graphs is challenging due to *uncertainty* of data.

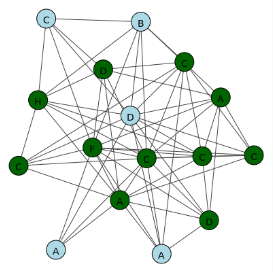


FIGURE 1: GRAPH (WITH QUERY MANIFESTATION)

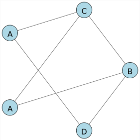


FIGURE 2: QUERY

- Subgraph isomorphism is NP-complete *Approximate Subgraph Matching*

Introduction

Subgraph querying in large real-world graphs is challenging due to *uncertainty* of data.

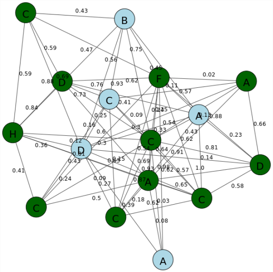


FIGURE 1: GRAPH (WITH QUERY MANIFESTATION)

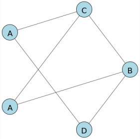


FIGURE 2: QUERY

- Subgraph isomorphism is NP-complete
  Approximate Subgraph Matching
- Automatically created knowledge bases
  e.g. StringDB, YAGO, DBPedia

Introduction

Subgraph querying in large real-world graphs is challenging due to *uncertainty* of data.

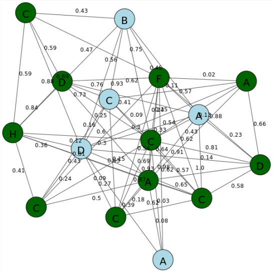


FIGURE 1: GRAPH

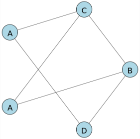


FIGURE 2: QUERY

- Subgraph isomorphism is NP-complete
  Approximate Subgraph Matching
- Automatically created knowledge bases
  e.g. StringDB, YAGO, DBPedia

Aim: Find the **top-k subgraphs** of G that are the **best approximate matches** of Q.

Challenges

- Best approximate match
  1. High existential probability
  2. High label and structural similarity

Challenges

- Best approximate match
    1. High existential probability – captured through Possible World Semantics
    2. High label and structural similarity
- $2^{|E|}$ possible worlds! ($|E|$ = #edges in G)

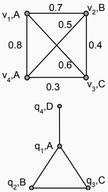
Challenges

- Best approximate match
  1. High existential probability – captured through Possible World Semantics
  2. High label and structural similarity
- $2^{|E|}$ possible worlds! ($|E|$ = #edges in G)

CHISEL (Chi-Squared Search in Large Probabilistic Graphs)

- Efficient integration of *possible world semantics (PWS) model* with label and structure match
- Similarity captured through *statistical significance*
- Return *best approximately matching subgraphs*

CHISEL



G

Vertex	Label	Neighbors
v_1	A	(A,0.8), (B,0.7), (C,0.6)
v_2	B	(A,0.7), (A,0.5), (C,0.4)
v_3	C	(A,0.6), (A,0.3), (B,0.4)
v_4	A	(A,0.8), (B,0.5), (C,0.3)

Q

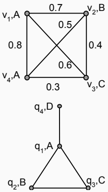
Vertex	Label	Neighbors	Triplet-list
q_1	A	B,C,D	(B,A,C), (C,A,D), (B,A,D)
q_2	B	A,C	(A,B,C)
q_3	C	A,B	(B,C,A)
q_4	D	A	(A,D,ϕ)

FIGURE 3: EXAMPLE

Steps

1. Indexes and probability computation
   (Offline for G)
2. Vertex pair construction
3. Similarity computation
4. Expand candidate vertex pairs

ChiSeL – Step 1



G

Vertex	Label	Neighbors
v_1	A	(A,0.8), (B,0.7), (C,0.6)
v_2	B	(A,0.7), (A,0.5), (C,0.4)
v_3	C	(A,0.6), (A,0.3), (B,0.4)
v_4	A	(A,0.8), (B,0.5), (C,0.3)

Q

Vertex	Label	Neighbors	Triplet-list
q_1	A	B,C,D	(B,A,C), (C,A,D), (B,A,D)
q_2	B	A,C	(A,B,C)
q_3	C	A,B	(B,C,A)
q_4	D	A	(A,D,ϕ)

FIGURE 3: EXAMPLE

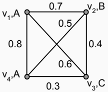
Steps

1. **Indexes and probability computation**
   (Offline for G)
2. Vertex pair construction
3. Similarity computation
4. Expand candidate vertex pairs

- Labels-Vertices Inverted Index
  $A \rightarrow v_1, v_4$
- Expected Degree
  $$\delta_{v_1} = \mathcal{E}[deg(v_1)] = 2.1$$
- Neighbor Labels Index
  $v_1 \rightarrow [\langle A, 0.8\rangle, \langle B, 0.7\rangle, \langle C, 0.6\rangle]$
- Neighbor Label Probabilities
  $Pr\Big(\#\big(A \in ne(v_1)\big) = k\Big)$
  $k \in \{0, 1, \geq 1\}$

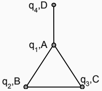
Index construction for Q on the fly

ChiSeL – Step 2



G

Vertex	Label	Neighbors
v_1	A	(A,0.8), (B,0.7), (C,0.6)
v_2	B	(A,0.7), (A,0.5), (C,0.4)
v_3	C	(A,0.6), (A,0.3), (B,0.4)
v_4	A	(A,0.8), (B,0.5), (C,0.3)



Q

Vertex	Label	Neighbors	Triplet-list
q_1	A	B,C,D	(B,A,C), (C,A,D), (B,A,D)
q_2	B	A,C	(A,B,C)
q_3	C	A,B	(B,C,A)
q_4	D	A	(A,D,ϕ)

FIGURE 3: EXAMPLE

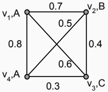
Steps

1. Indexes and probability computation
   (Offline for G)
2. **Vertex pair construction**
3. Similarity computation
4. Expand candidate vertex pairs

- Vertices with same labels

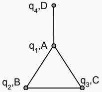
$$\mathcal{VP} = \{\underbrace{\langle v_1, q_1\rangle}_{A}, \underbrace{\langle v_4, q_1\rangle}_{A}, \underbrace{\langle v_2, q_2\rangle}_{B}, \underbrace{\langle v_3, q_3\rangle}_{C}\}$$

CHISEL – Step 3



G

Vertex	Label	Neighbors
v_1	A	(A,0.8), (B,0.7), (C,0.6)
v_2	B	(A,0.7), (A,0.5), (C,0.4)
v_3	C	(A,0.6), (A,0.3), (B,0.4)
v_4	A	(A,0.8), (B,0.5), (C,0.3)



Q

Vertex	Label	Neighbors	Triplet-list
q_1	A	B,C,D	(B,A,C), (C,A,D), (B,A,D)
q_2	B	A,C	(A,B,C)
q_3	C	A,B	(B,C,A)
q_4	D	A	(A,D,ϕ)

FIGURE 3: EXAMPLE

Steps

1. Indexes and probability computation *(Offline for G)*
2. Vertex pair construction
3. **Similarity computation**
4. Expand candidate vertex pairs

- For each *query triplet* find the *best triplet match* in each possible world
- Use of χ^2 measure
- Query Triplet - $\langle l_x, l_q, l_y \rangle$ $x, y \in$ neighbor(q)
  $q_1 \rightarrow \{\langle B, A, C\rangle, \langle C, A, D\rangle, \langle B, A, D\rangle\}$

Similarity Computation *(using χ^2 Statistic)*

Query triplet $\longrightarrow$ **Best triplet match** $\longrightarrow$ Compute χ^2 value

- s_0: no instance of l_x and l_y exist in the neighborhood
- s_1: exactly one of the labels, either l_x or l_y, exist in the neighborhood
- s_2: at least one instance of both the labels exists

Similarity Computation *(using χ^2 Statistic)*

Query triplet $\longrightarrow$ **Best triplet match** $\longrightarrow$ Compute χ^2 value

For instance, for query triplet $\langle B, A, C\rangle$ in example,

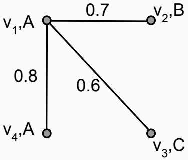


FIGURE 4: SAMPLE WORLD WITH ALL NEIGHBORS
PR(W) = $0.8 \times 0.7 \times 0.6 = 0.336$

Best Match - $\langle B, A, C\rangle$
Symbol, Pr(w) - $\langle s_2, 0.336\rangle$

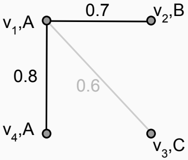


FIGURE 5: SAMPLE WORLD WITH TWO NEIGHBORS
PR(W) = $0.8 \times 0.7 \times (1 - 0.6) = 0.224$

Best Match - $\langle B, A, A\rangle$
Symbol, Pr(w) - $\langle s_1, 0.224\rangle$

Similarity Computation *(using χ^2 Statistic)*

Query triplet $\longrightarrow$ Best triplet match $\longrightarrow$ **Compute χ^2 value**

Statistical significance (χ^2 statistic) to capture similarity

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i}$$

- Captures deviation of *Observed* value from *Expected* value
- Random distribution hypothesis
- Manifestation of Q in G is a rare event
- Matching subgraph $\Rightarrow$ Higher χ^2

Similarity Computation *(using χ^2 Statistic)*

Query triplet $\longrightarrow$ Best triplet match $\longrightarrow$ **Compute χ^2 value**

Statistical significance (χ^2 statistic) to capture similarity

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i}$$

- Captures deviation of *Observed* value from *Expected* value
- Random distribution hypothesis
- Manifestation of Q in G is a rare event
- Matching subgraph $\Rightarrow$ Higher χ^2

Enumerating all possible worlds – inefficient and impractical!

Efficient Integration of PWS

Offline

Neighbor Label Probabilities, $\mathbf{P(\#l_x)}$

- Captures probability distribution of labels in neighborhood
- $P(\#l_x = 0) \rightarrow$ Probability that no instance of l_x exists as neighbor
- For $\#l_x \in 0, 1, \geq 1$

Efficient Integration of PWS

Offline

Neighbor Label Probabilities, $\mathrm{P}(\#\mathrm{l}_{\mathbf{x}})$

- Captures probability distribution of labels in neighborhood
- $P(\#l_x = 0) \rightarrow$ Probability that no instance of l_x exists as neighbor
- For $\#l_x \in 0, 1, \geq 1$

Expected Probability of Symbols

- Probability that a neighbor has label l is $\frac{1}{|\mathcal{L}|}$, assuming $|\mathcal{L}|$ labels
- $P_e(s_0) = \left(\left(1 - \frac{1}{|\mathcal{L}|}\right)^{\delta}\right)^2$
- PWS captured through δ

Efficient Integration of PWS

Offline

Neighbor Label Probabilities, $\mathrm{P}(\#\mathrm{l_x})$

- Captures probability distribution of labels in neighborhood
- $P(\#l_x = 0) \rightarrow$ Probability that no instance of l_x exists as neighbor
- For $\#l_x \in 0, 1, \geq 1$

Expected Probability of Symbols

- Probability that a neighbor has label l is $\frac{1}{|\mathcal{L}|}$, assuming $|\mathcal{L}|$ labels
- $P_e(s_0) = \left(\left(1 - \frac{1}{|\mathcal{L}|}\right)^{\delta}\right)^2$
- PWS captured through δ

Online

- $P(s_0) \rightarrow$ no label match in neighbors
- $O[s_0] = \sum_{triplets} P(s_0)$
- $O[s_1], O[s_2]$ likewise
- $E[s_0] = \sum_{triplets} P_e(s_0)$
- Similarly, $E[s_1], E[s_2]$

Similarity Computation *(using χ^2 Statistic)*

Query triplet $\longrightarrow$ **Best triplet match** $\longrightarrow$ Compute χ^2 value

- A distribution over s_0, s_1 and s_2 for each triplet across worlds
- Sum over all triplets to get observed values for the vertex pair
- Compute similarity for each vertex pair as

$$\chi^2_{\langle v,q\rangle} = \sum_{i=0}^{2} \frac{(O[s_i] - E[s_i])^2}{E[s_i]}$$

Expand Candidate Vertex Pairs – Step 4

- Greedy expansion over vertex pairs
  - Stop when no more query vertices or cannot expand
- Prefer neighbors with
  - High χ^2 value
  - Large edge probability

Expand Candidate Vertex Pairs – Step 4

- Greedy expansion over vertex pairs
  - Stop when no more query vertices or cannot expand
- Prefer neighbors with
  - High χ^2 value
  - Large edge probability

Ranking answer subgraph s_i

- $\chi^2(s_i)$ = Sum of χ^2 of constituting vertex pairs

Expand Candidate Vertex Pairs – Step 4

- Greedy expansion over vertex pairs
  - Stop when no more query vertices or cannot expand
- Prefer neighbors with
  - High χ^2 value
  - Large edge probability

Ranking answer subgraph s_i

- $\chi^2(s_i)$ = Sum of χ^2 of constituting vertex pairs
- Sort into groups by size of subgraph in descending order
- Sort groups internally by $\chi^2(s_i)$ in descending order
- Top-1 answer has highest cardinality and highest χ^2 in that group

Experimental Setup

Datasets

Dataset	# Vertices	# Edges	# Labels	Avg. Degree
PPI-complete	7.6M	1.2B	0.2M	316
PPI-small	12.0K	10.7M	2.4K	1789
YAGO	4.3M	11.5M	4.0M	5
IMDb	3.0M	11.0M	3.0M	7

FIGURE 6: CHARACTERISTICS OF DATASETS USED

- **STRING DB**: Protein-protein interaction network, v10.5 (PPI-complete)
  - PPI-small (sampled from PPI-complete)
- **YAGO**: Open source knowledge graph
  - Entities extracted from Wiki, WordNet, GeoNames
- **IMDb**: Deterministic dataset with information on actors, directors etc.
  - Random edge-probabilities assigned
- **Queries**: Exact and Noisy

Results

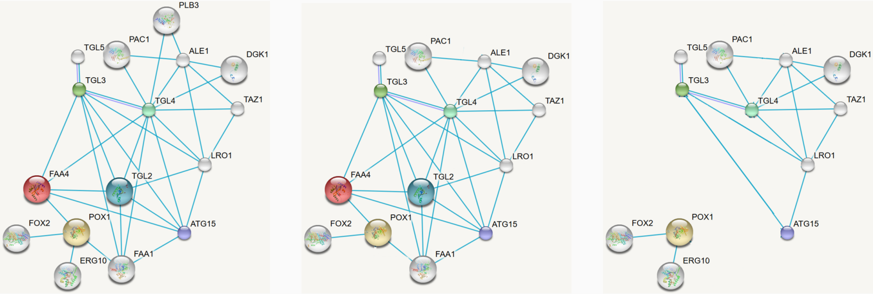
Table 1: Overall Performance Comparison for top-10 subgraphs

Method	YAGO		IMDb		PPI-small		PPI-complete	
	Acc.	Time (s)	Acc.	Time (s)	Acc.	Time (s)	Acc.	Time (s)
ChiSeL	**0.89**	**1.62**	**0.87**	**0.05**	**0.96**	**16.69**	**0.84**	**0.14**
PBound	0.26	560.92	0.57	101.95	0.29	3134.09	Time-out	
Fuzzy	0.61	1.82	0.62	2.19	0.01	13970.94	Time-out	

- ***PBound***[1] – tree decomposition based approach
- ***Fuzzy***[2] – path decomposition based approach
- Averaged across all queries

[1] Gu *et al.* 2016, WWW, pages 755-782,
[2] Li *et al.* 2019, Fuzzy Sets and Systems, 376:106-126

Real-World Use Case: String DB



(a) String DB Query (b) CHISEL (c) NAGA

FIGURE 7: EVALUATION OF REAL QUERIES ON PPI-COMPLETE

- ***NAGA***[1]: Neighbour Aware Greedy Graph Search

[1] Dutta *et al.* 2017, WWW, pages 1281-1290

Probabilistic Graph Variants

ChiSeL can be extended to incorporate following problem settings:

- Edge labeled graphs
- Noisy labels
- Uncertain vertices
- Label uncertainty
- Uncertain query graphs

Conclusions

- Approximate searching is necessary for large probabilistic graphs
- CHISEL is a threshold-free approach
  - Scales well for large probabilistic graphs
  - Efficiently integrates possible world semantics
- The χ^2 statistic is a powerful framework to capture subgraph similarity
- CHISEL is extendable to different variants of uncertain scenarios

The source code of CHISEL is available from https://github.com/Shubhangi-Agarwal/ChiSeL.

Conclusions

- Approximate searching is necessary for large probabilistic graphs
- CHISEL is a threshold-free approach
  - Scales well for large probabilistic graphs
  - Efficiently integrates possible world semantics
- The χ^2 statistic is a powerful framework to capture subgraph similarity
- CHISEL is extendable to different variants of uncertain scenarios

THANK YOU!

Questions?

Answers!

The source code of CHISEL is available from https://github.com/Shubhangi-Agarwal/ChiSeL.

Experimental Setup - Benchmarks

PBound[1]

- Performs maximal subgraph matching
- Incrementally computes similarity probabilities
- Prune using probability upper bounds

Fuzzy[2]

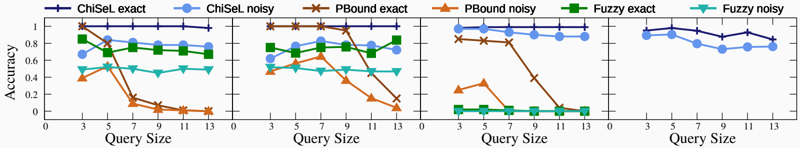
- Path-based graph decomposition
- K-partite based path-joining techniques

---

[1] Gu *et al.* 2016, International Conference on World Wide Web (WWW)
[2] Li *et al.* 2019, Fuzzy Sets and Systems

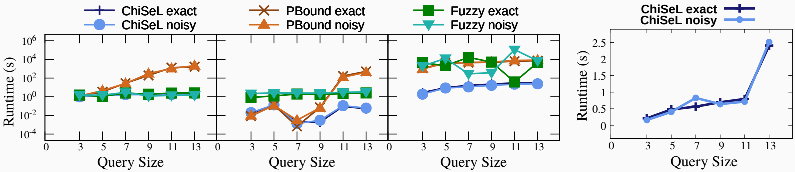
Efficiency Results

Figure 8: Accuracy comparison of different algorithms over all datasets.



(a) YAGO (b) IMDb (c) PPI-small (d) PPI-complete

Figure 9: Runtime comparison of different algorithms over all datasets.



(a) YAGO (b) IMDb (c) PPI-small (d) PPI-complete

Experimental Setup - Query dataset

Exact Queries (*exact*)

- Constructed from the datasets
- Randomly select a vertex
- Explore neighborhood through random walk
- Stop if q vertices visited
- Subgraph induced from visited vertices - *exact* query

Noisy Queries (*noisy*)

- Constructed from *exact*
- Insert noise in *exact* queries
- Randomly insert, delete or replace edges
- Stop when #operations are one-third of total edges

Per Dataset	**Other datasets**	**PPI-complete**
Query graph size	$\{3, 5, \ldots, 13\}$	$\{3, 5, \ldots, 25\}$
#Queries of each type	20	20
Total #Queries	$2 * 6 * 20 = \mathbf{240}$	$2 * 12 * 20 = \mathbf{480}$

TABLE 2: CHARACTERISTICS OF QUERIES