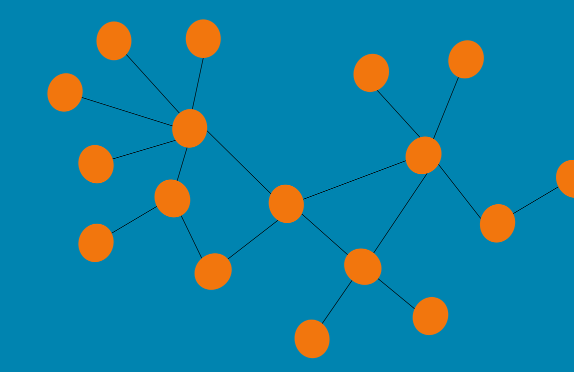
Maintaining sliding-window neighborhood profiles in interaction networks

Rohit Kumar[1], Toon Calders[1], Aristides Gionis[2] and Nikolaj Tatti[2]

[1]Department of Computer and Decision Engineering
Université Libre de Bruxelles, Belgium
rohit.kumar@ulb.ac.be

[2]Helsinki Institute for Information Technology and
Department of Computer Science
Aalto University, Finland



1. Problem / Query

Real-time monitoring of Neighborhood Profile of a node for a given time window in an interaction network.

For example queries like

- How many distinct nodes are at shortest distance r from a node v at time t?
- How many distinct nodes were at shortest distance r from a node v at time t and t-w?

2. Interaction Network/Graph?

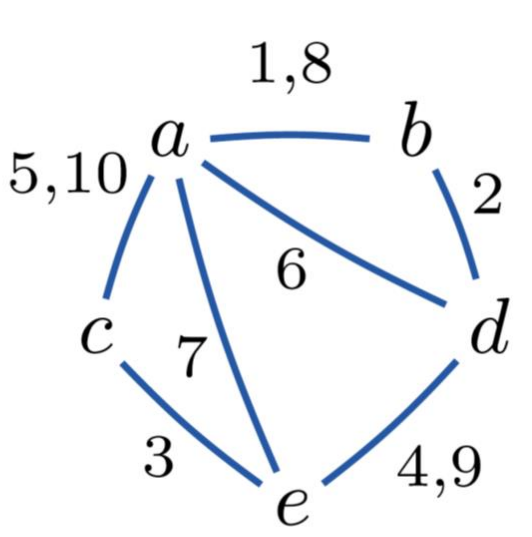
An interaction network is defined as a sequence of timestamped interactions є over edges of a static graph G = (V,E).

For example:

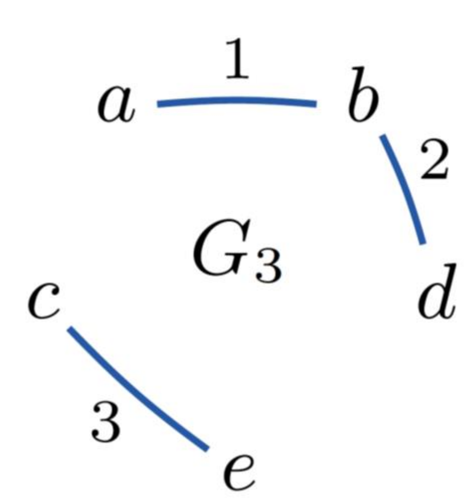
- Social interaction in social network.
- Email/ Message or call interaction in communication network.
- Data exchange between computer network.

In a sliding window model only the edges falling under the window length is considered.

Below is a toy network and three snapshot graphs with a window size 3.

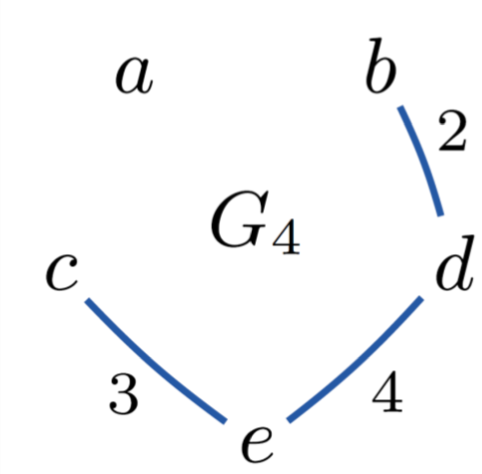


Node (v)	Snapshot 1	
	r = 1	r = 2
a	1	1
b	2	0
c	1	0
d	1	1
e	1	0



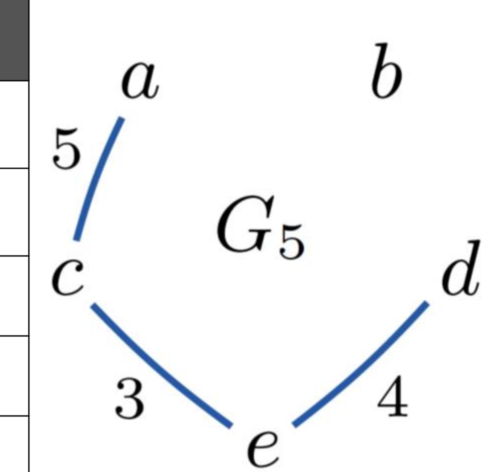
Snapshot 1

Node (v)	Snapshot 2	
	r = 1	r = 2
a	0	0
b	1	1
c	1	1
d	2	1
e	2	1



Snapshot 2

Node (v)	Snapshot 3	
	r = 1	r = 2
a	1	1
b	0	0
c	2	1
d	1	1
e	2	1

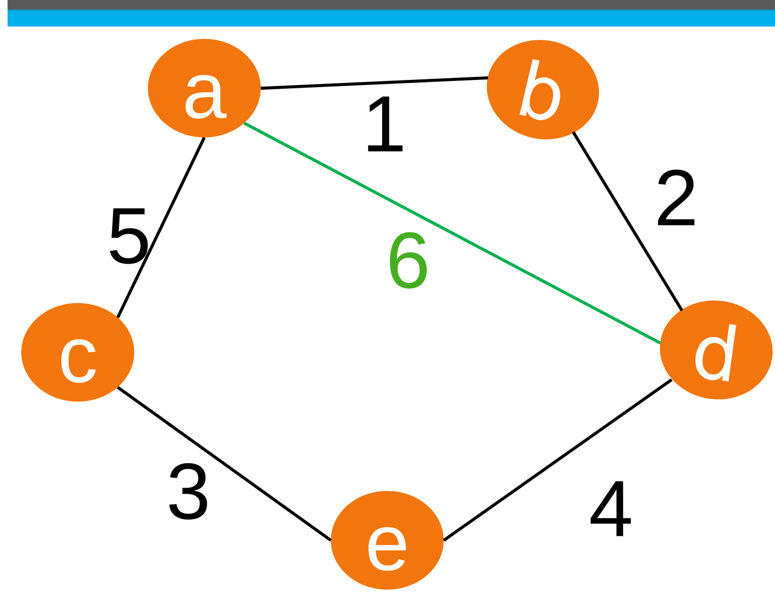


Snapshot 2

3. Challenges!

- Dynamic graph : Edges coming as stream and old edges going out of window so every snapshot is a new graph.
- No online real-time system available
- Existing Solutions (HyperANF) : Iterative and non scalable for real time query.

4. Exact Summary to answer Neighborhood queries



S^a					
r	a	b	c	d	e
1	∞	1	5	6	
2	∞	2		~~1~~	~~3~~ 4
3	∞			~~3~~	

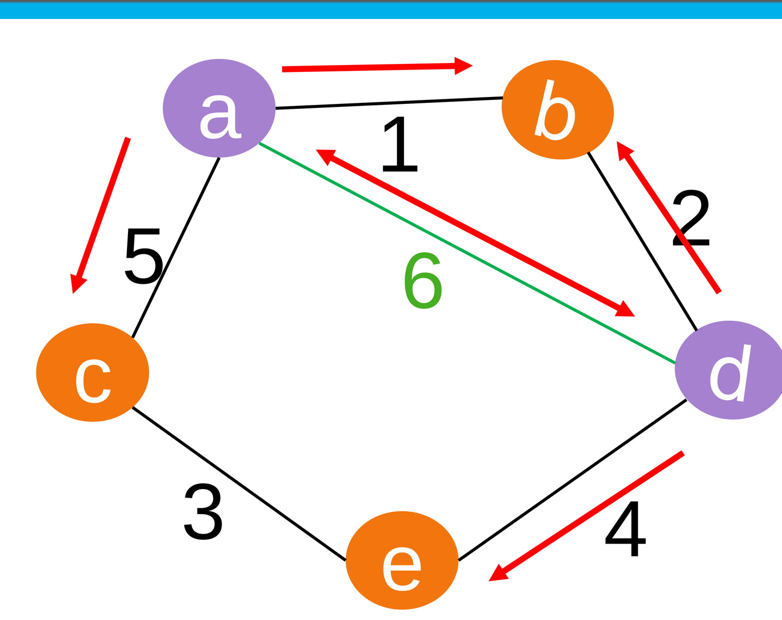
S^d					
r	a	b	c	d	e
1	6	2		∞	4
2	~~1~~		~~3~~ 5	∞	
3	~~3~~			∞	

5.Intituion behind the algorithm

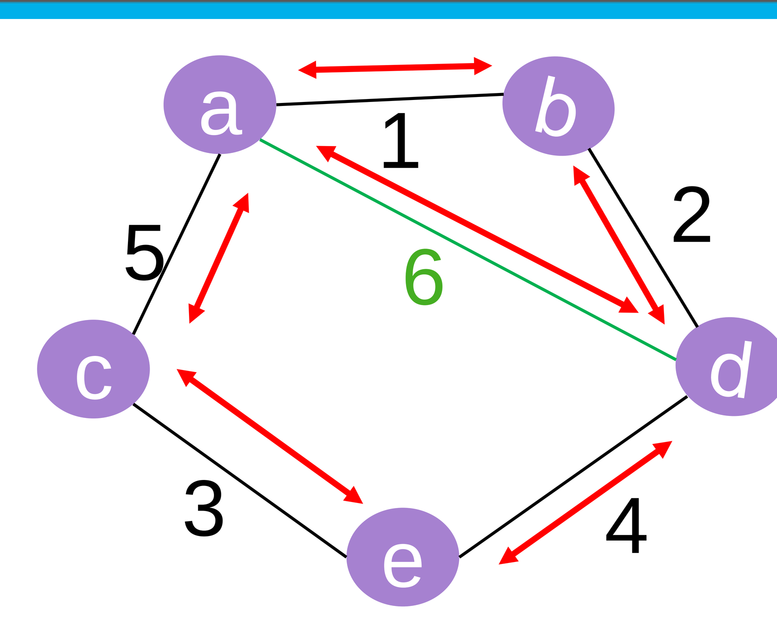
- Maintaining a list of promising paths and path horizons of length less then or equal to ***r*** of every node.
- Propagating the changes to neighbors as follows

  $S^u[r] = \text{merge}(S^v[r-1], S^u[r])\ \ \forall\ u \in N_g(v)$

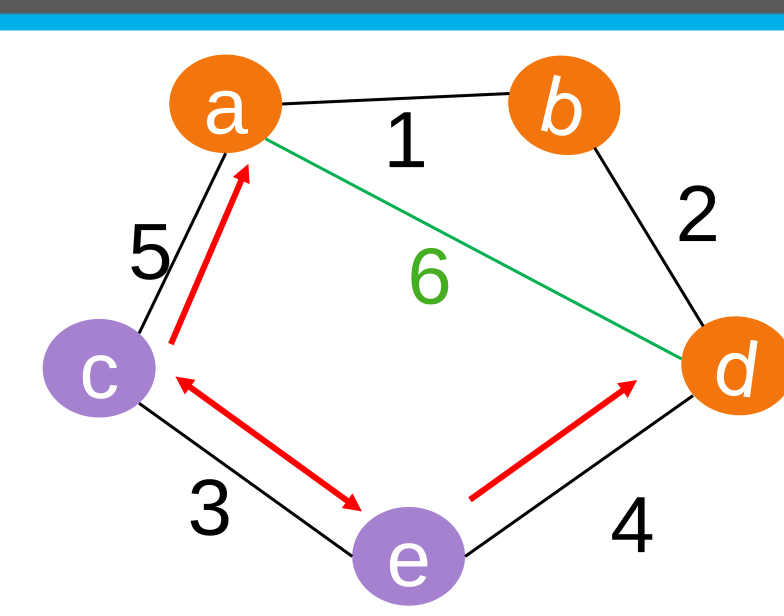
6. Distance wise propagation of summary



New edge will update S[1] of **a** and **d**
1st Super Step:
a and **d** propagate their S[1] to their neighbors to update S[2]



2nd Super Step:
All nodes had changes in their S[2] at last step so this step they propagate S[2] to their neighbors to update S[3]



3rd Super Step:
Only node **c** and **e** had a change in their summary in last step so they will only propagate to their neighbors

7. Time and Space Complexity

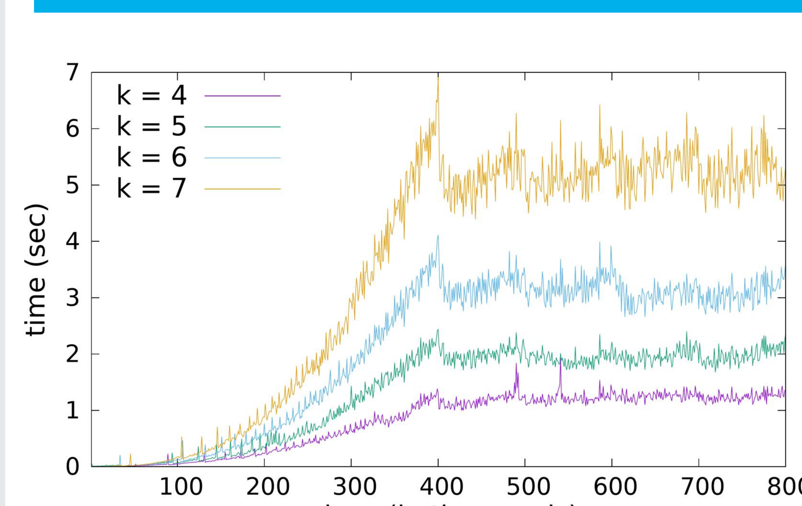
For a graph G with **n** nodes and **m** distinct edges and **r** be the upper bound on the distances

- For the exact algorithm (using Set)
  - Time Complexity of Adding new edge

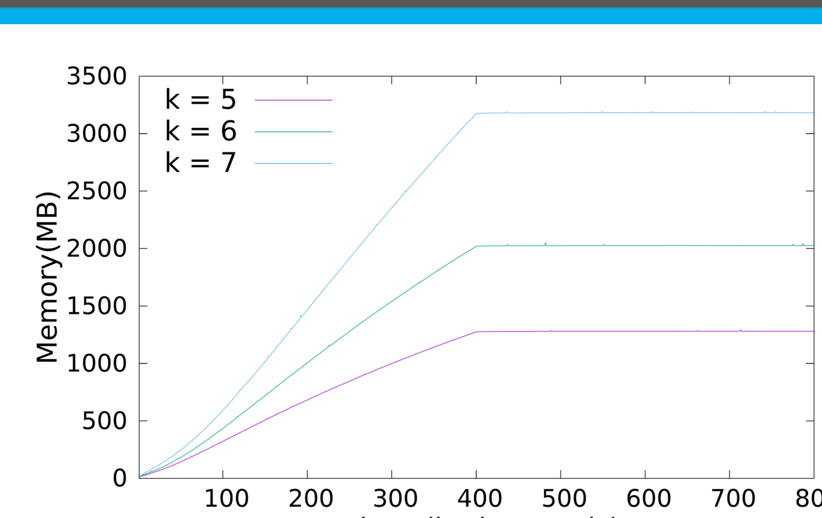
    O(r m n log(n))
  - Space constraint to maintain the summary is

    O(r n²)
- For the approx. algorithm (using Sliding HLL)
  - Time complexity : O(r m **2ᵏ*loglog*(n)**)
  - Space complexity : O(r **n2ᵏloglog(n)**)
  - Standard error : ~ 1.04 / √ ***b*** ; ***b*** #buckets in the Sliding HLL sketch

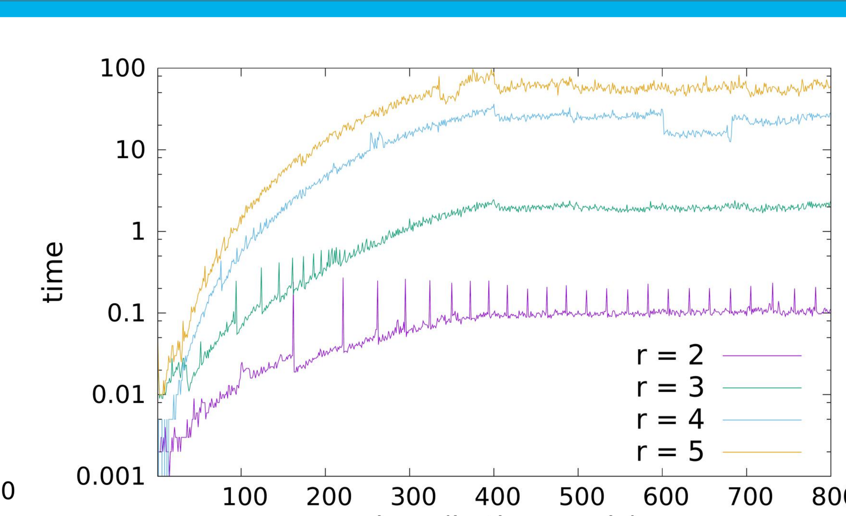
8. Results : DBLP data set (nodes : 192357 , edges : 400,000 , interactions : 800,000)



Time to process 1000 edges for different k



Memory Utilization as function of k



Time(log scale) to process 1000 edges for different distance r

Some useful background related papers

1. Boldi, P., Rosa, M., Vigna, S.: Hyperanf: Approximating the neighbourhood function of very large graphs on a budget. In: WWW. pp. 625-634 (2011)
2. Chabchoub, Y., Hebrail, G.: Sliding hyperloglog: Estimating cardinality in a data stream over a sliding window. In: ICDM Workshops (2010)
3. Palmer, C., Gibbons, P., Faloutsos, C.: ANF: A fast and scalable tool for data mining in massive graphs. In: KDD. pp. 81-90 (2002)
4. Rozenshtein, P., Tatti, N., Gionis, A.: Discovering dynamic communities in interaction networks. In: ECML PKDD. pp. 678-693 (2014)