Graph Stream Mining and Distributed Processing
(Research Progress Report)

Rohit Kumar
Advisor- Toon Calders (ULB)
Co- Advisor – Alberto Abelló (UPC)
CPC- Torben Bach Pedersen (AAU)

Outline

- Project Statement
- Work done so far
- On going work
- Future work planned

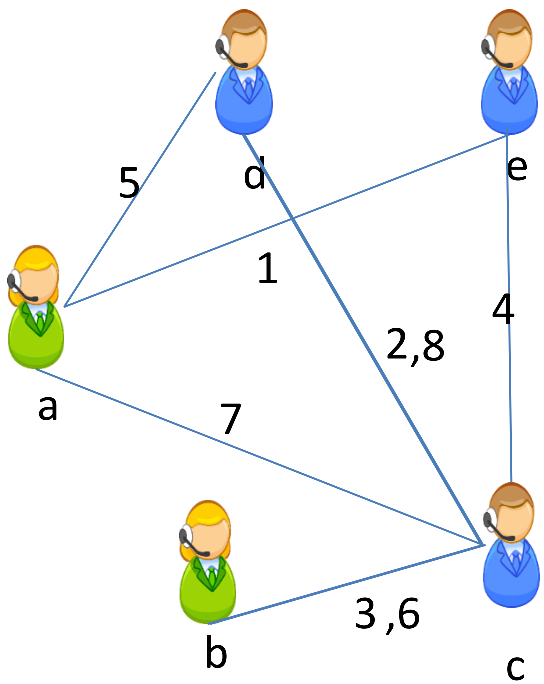
Graphs are everywhere

– Social Network
– Collaboration network
– Communication network
– Road network
– Protein interaction network
– Web graph
– Sensor Network

Types of graph

- **Static Graph :** classical graphs
- **Dynamic Graphs:** Graphs which evolve over time due to insert of new edges or nodes or deletion of edges or nodes.
- **Temporal/Interaction Networks:** Time dependent graphs.

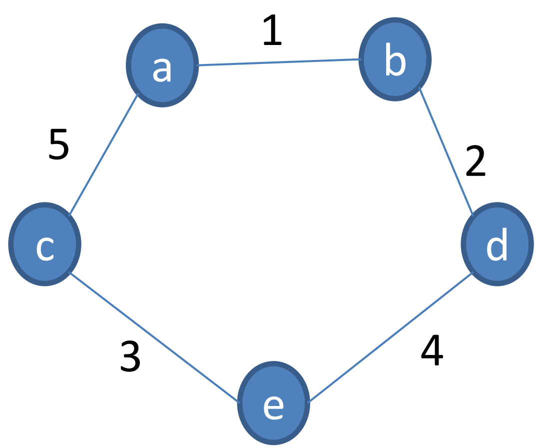
Example



1, (a , e)
2, (d, c)
3, (b , c)
4, (e , c)
5, (a , d)
6, (b , c)
7, (a , c)
8, (d , c)

.
.
.
.

Temporal Graph in sliding window

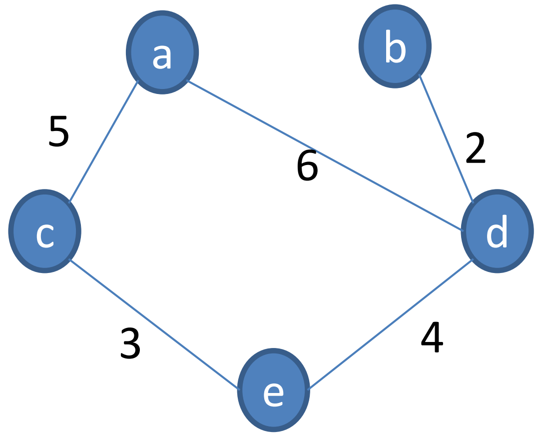


1, (a , b) 2, (b,d) 3, (c , e) 4, (d , e) 5, (a , c) 6, (a , d) 7, (a , c)

Current data in window Future edges

Consider a window length of size 5.

Temporal Graph in sliding window

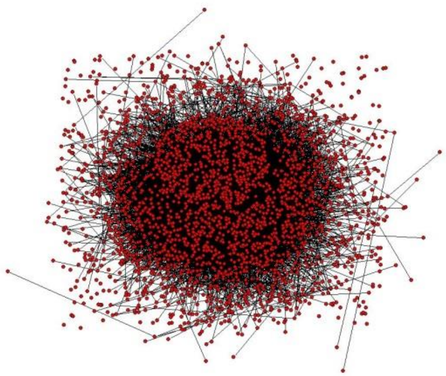


1, (a , b) | 2, (b, d) 3, (c , e) 4, (d , e) 5, (a , c) 6, (a, d) | 7, (a , c) ....

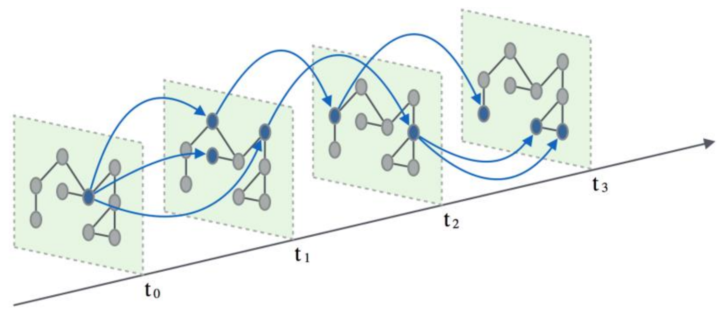
Expired edges | Current data in window | Future edges

Consider a window length of size 5.

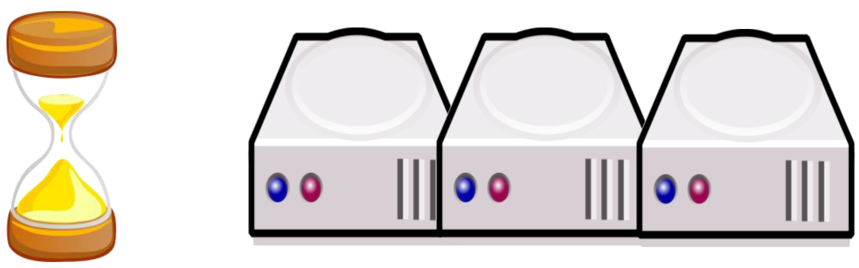
Problem!



Complexity



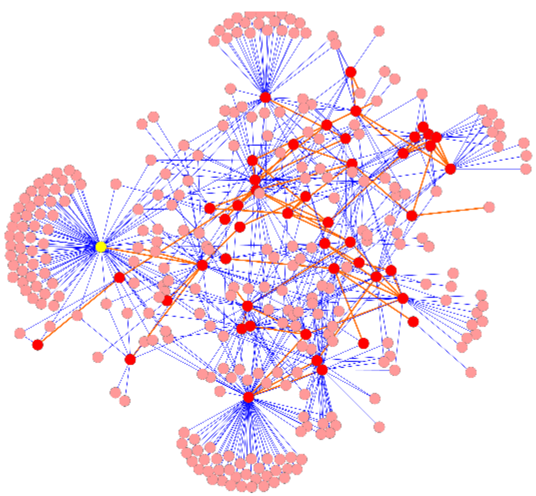
Rapidly Evolving with Time

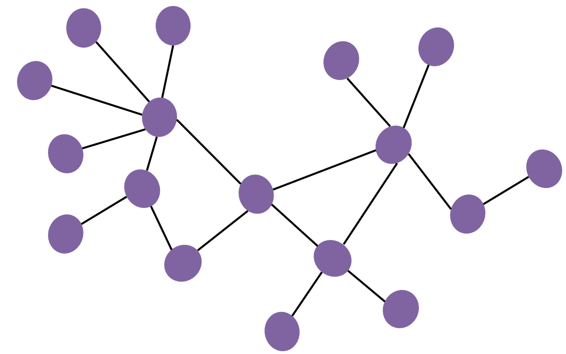


Graph analytics take lot of time and memory

https://lts2.epfl.ch/blog/gbr/category/graph/

Two approaches to solve the problem.





1. Creating scalable algorithms for temporal networks.

GraphX

2. Distributed Graph Processing

Project Milestones

M1: Algorithms

- Density Profile (T1)
- Information Flow (T2)

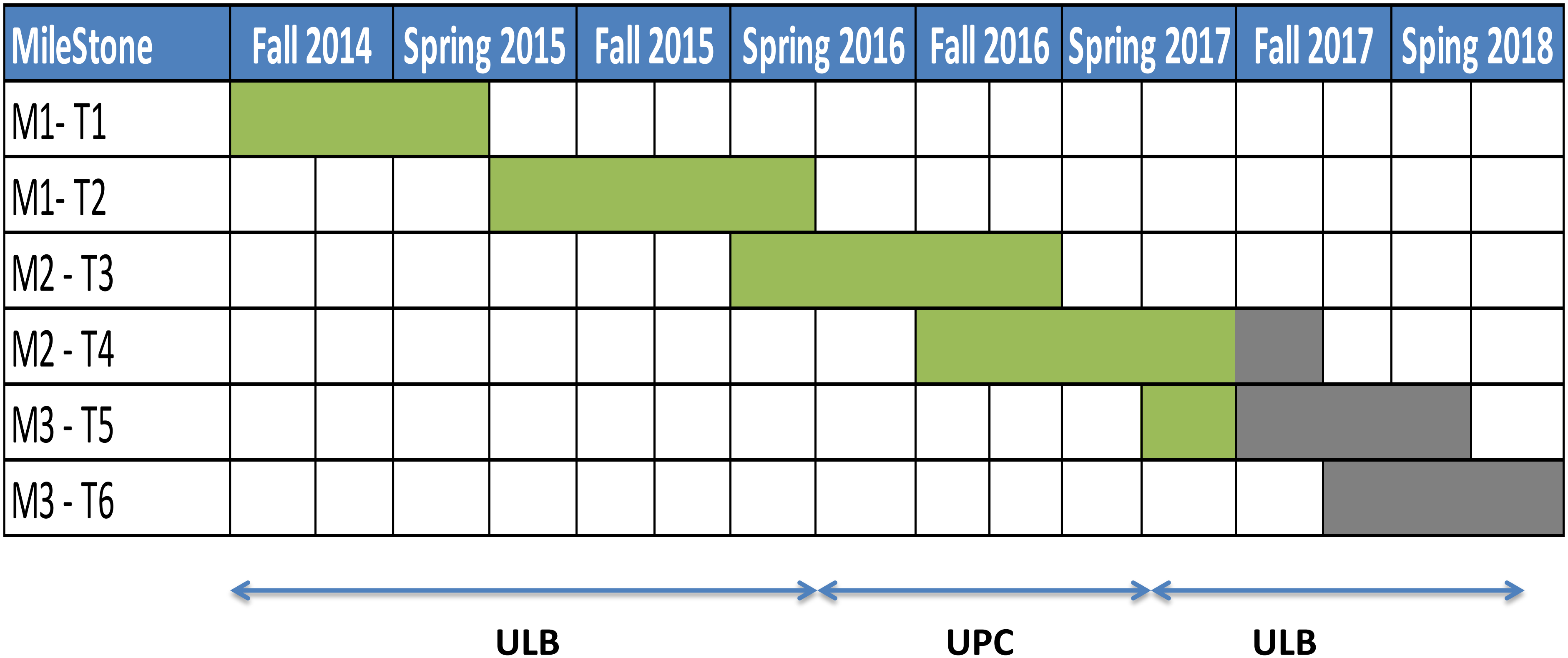
M2: System

- Pregel Cost Model (T3)
- Rule based partitioning (T4)

M3: Extension

- Cycle Detection(T5)
- Dynamic Repartitions (T6) (optional)

Gantt Chart



Fall: Aug – Jan
Spring Feb - July

Publications Status

- Milestone 1
  - 1 paper accepted in ECML/PKDD 2015(T1)
  - 1 paper accepted in WSDM 2017 (T2) (2nd author)
  - 1 paper accepted in EDBT 2017 (T2)
  - Nectar Track paper in ECML/PKDD 2017 (T2)
  - Submitted Demo Paper for CIKM (19^{th} Aug, 2017) (T2) (2^{nd} author)
  - Journal paper ready for submission in Knowledge and Information Systems (KAIS) (T2) (2^{nd} author)
- Milestone 2
  - 1 paper accepted in ADBIS 2017 (T3)
  - Journal paper (work in progress) for Information Systems Journal– By Sep 2017.(T4)
- Milestone 3
  - 1 workshop paper submitted in TDLSG-ECML/PKDD 2017(T5)
  - Paper for WWW 2017 (work in progress) – By Oct 2017. (T5)
  - Conference paper (venue not decided) (T6) - optional

*Code at: https://github.com/rohit13k

Outline

- Project Background
- Work done so far
- On going work
- Future work planned

Milestone 1 Topic 1

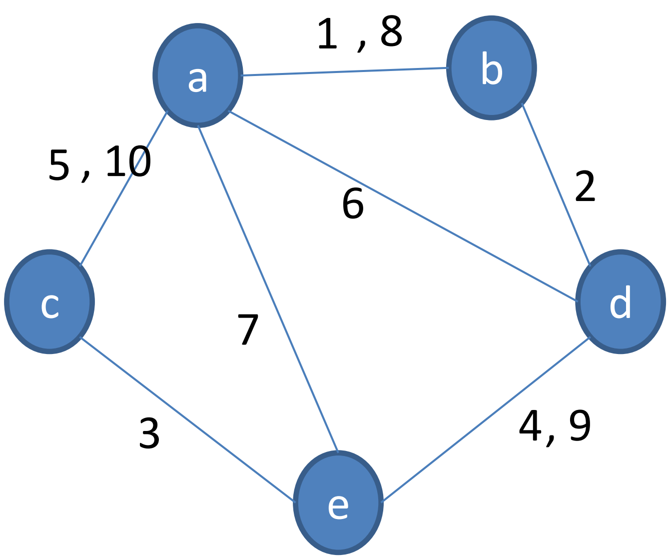
Kumar, R., Calders, T., Gionis, A., & Tatti, N.. ***Maintaining Sliding-Window Neighborhood Profiles in Interaction Networks***. **Published in *ECML/PKDD 2015***

1. Maintaining sliding-window neighborhood profiles in interaction networks*

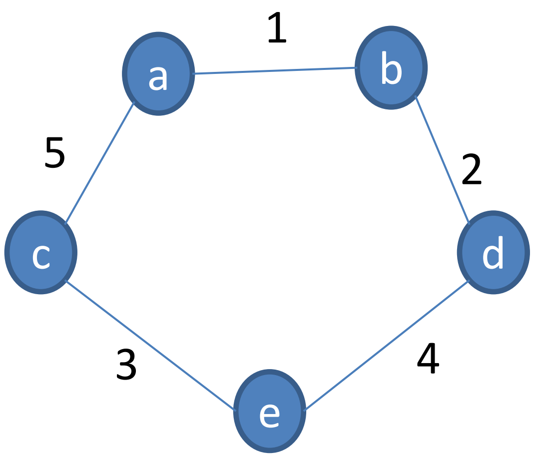
Query: How many nodes are within distance r from node v at time t?

- We call it Neighborhood profile of a node!
- For example How many nodes within distance 2 from ***a*** at different time ***t***

Node a	r = 1		r = 2	
t=3	1	b	1	d
t=5	2	c, b	2	d, e
t=7	4	c, e, d, b	0	-
t=10	4	c, e, d, b	0	-



Neighborhood profile in sliding window



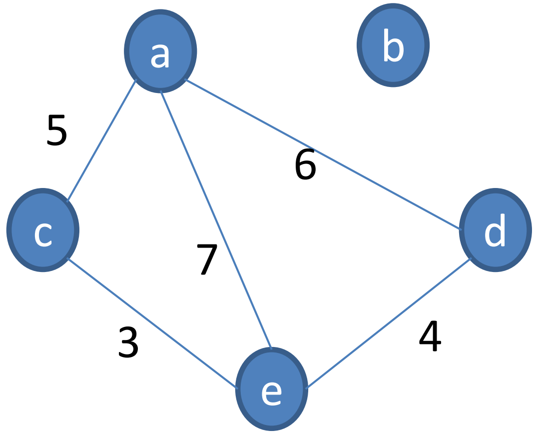
Node a	**r=1**		**r=2**	
t=5	2	b,c	2	d,e

1, (a , b) 2, (b,d) 3, (c , e) 4, (d , e) 5, (a , c) 6, (a , d) 7, (a , e)

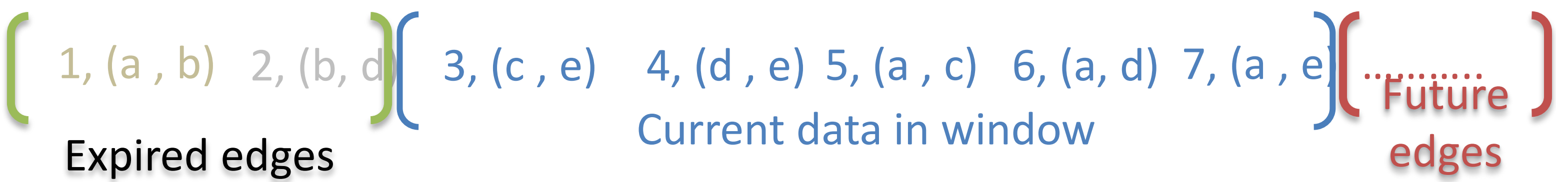
Current data in window Future edges

Consider a window length of size 5.

Neighborhood profile in sliding window

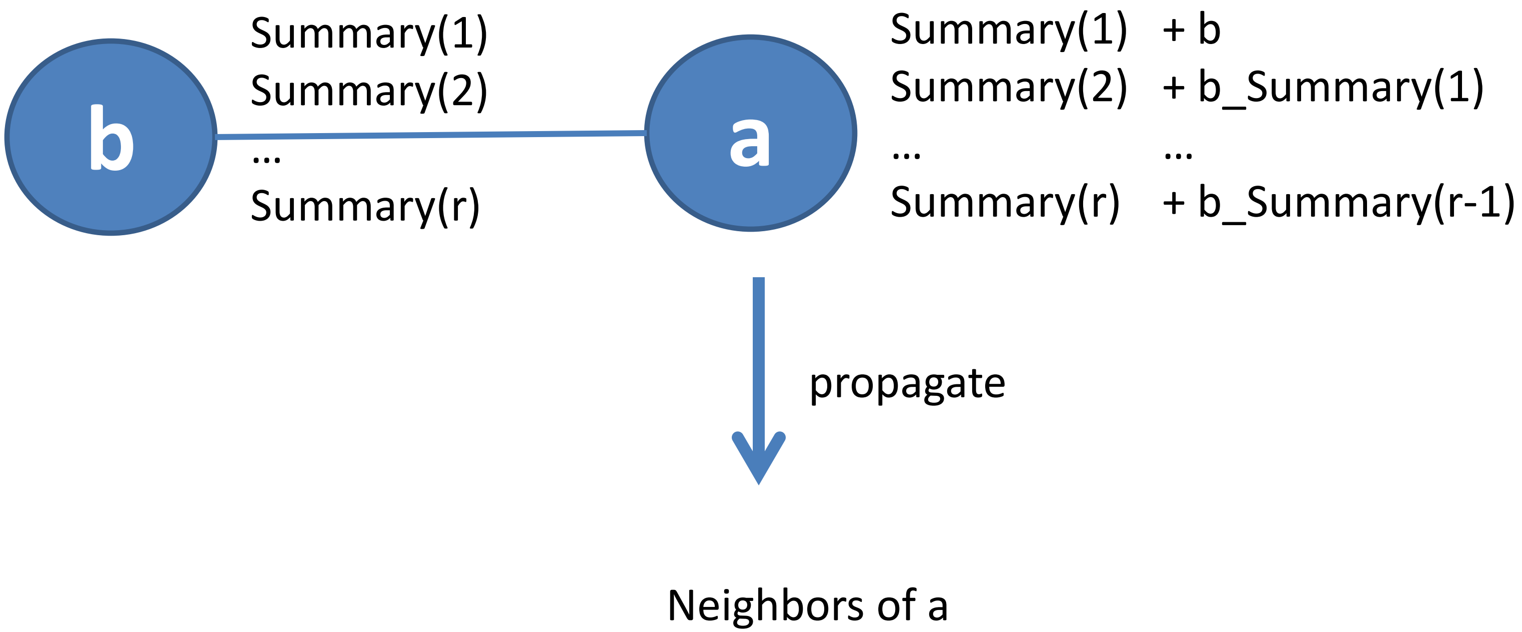


Node a	r=1		r=2	
t=5	2	b,c	2	d,e
t=7	3	c,d,e	0	



Consider a window length of size 5.

Neighborhood profile in sliding window



Complexity Analysis

n nodes and m interactions

Exact Algorithm:

Time Complexity :

– $O(r\ m\ \mathbf{n}\ \boldsymbol{log}(\mathbf{n}))$

Memory Complexity:

– $O(r\ \mathbf{n^2})$

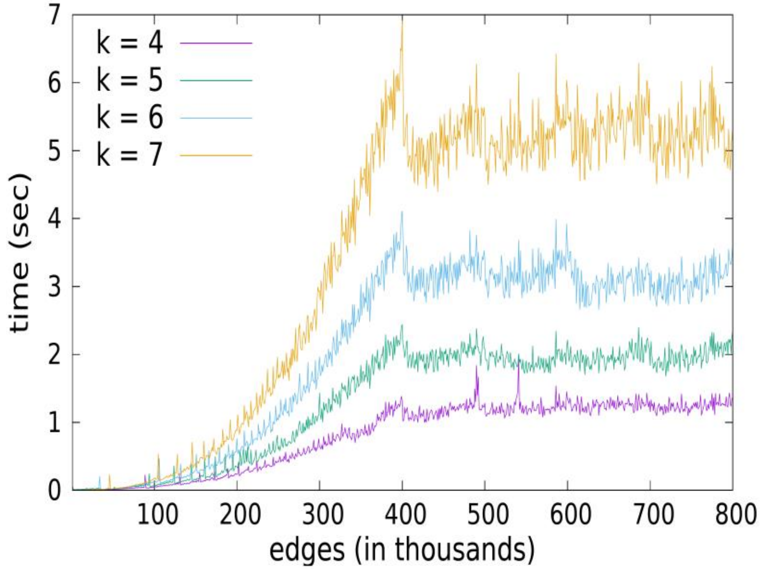
Sketch based approach using our extension of HLL:

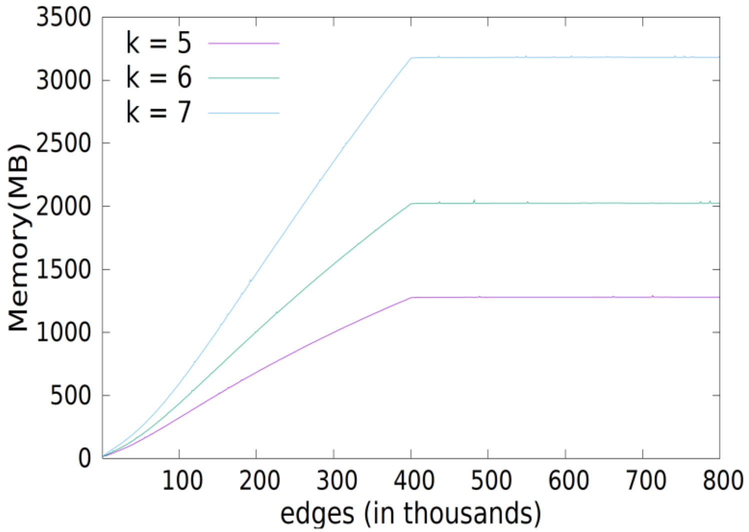
Time Complexity : $O(r\ m\ \mathbf{2^k}\boldsymbol{loglog}(\mathbf{n}))$

Memory Complexity: $O(r\ \mathbf{n2^k loglog(n)})$

$k=6$, $\omega \ll n$

Neighborhood profile in sliding window



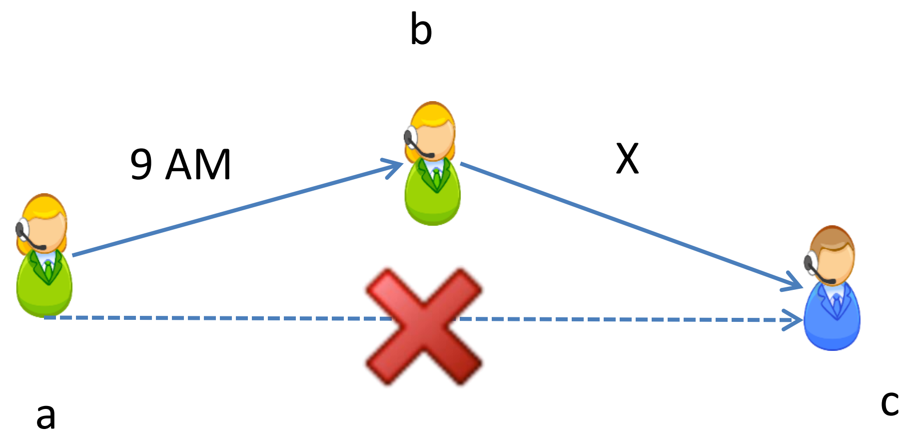


Milestone 1 Topic 2

Influence Propagation in temporal Networks

- **Kumar, R.,** & Calders, T. Information Propagation in Interaction Networks. **Published in *EDBT* 2017.**
- Saleem, M. A., **Kumar, R.,** Calders, T., Xie, X., & Pedersen, T. B. Location Influence in Location-based Social Networks. **Published in *WSDM 2017* ACM.**

1. Information Propagation in Interaction Networks



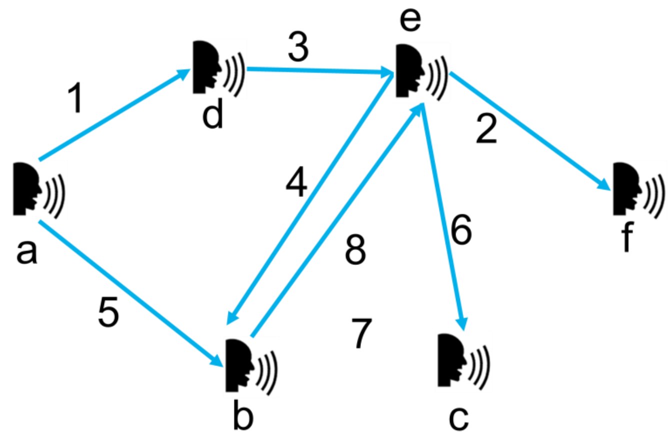
If X= 9:10 AM

If X= 8:10 AM or After 10 days!!

Influence Reachability Set

The set of users in the network which could be reached by user **a** in given time window is it’s influence reachability set.

Information Propagation in Interaction Networks*



Node (v)	IRS	
	window = 2	window = 3
a	(d, b)	**(d, b, e, c)**
b	(e, c)	(e, c)
e	**(b, c, f)**	(b, c, f)

Influential Node changes with window length

What we want to study!

- Given a set of initial users and a time window identify the number of users who will get influenced. **(Influence oracle problem)**

- Find top k influential users in the given interaction network under a time constrained information propagation. **(Influence Maximization Problem)**

Algorithm

- **One pass algorithm!**

1, (a,d)
2, (e,f)
3, (d,e)
4, (e,b)
5, (a,b)
6, (b,e)
7, (e,c)
8, (b,c)

S(u) = {(v, ʌ(u, v)) }

ʌ(u, v) is defined as the end time of the earliest information channel of length ω or less from u to v.

```
For an entry t,u,v:
   Add (S(u) , (v, t))
   S(u)=Merge(S(u),S(v))

Merge:
   For All (x, t' ) ∈ S (v)
      If (t – t' ) < ω
         Add (S(u), (x, t' ))
```

Complexity Analysis

n nodes and m interactions

Exact Algorithm:

Time Complexity :

- O (mn)

Memory Complexity:

- O (n^2)

Sketch based approach using our extension of HLL:

Time Complexity : O ($m2^k \log(\omega)^2$)

Memory Complexity: O ($n2^k \log(\omega)^2$)

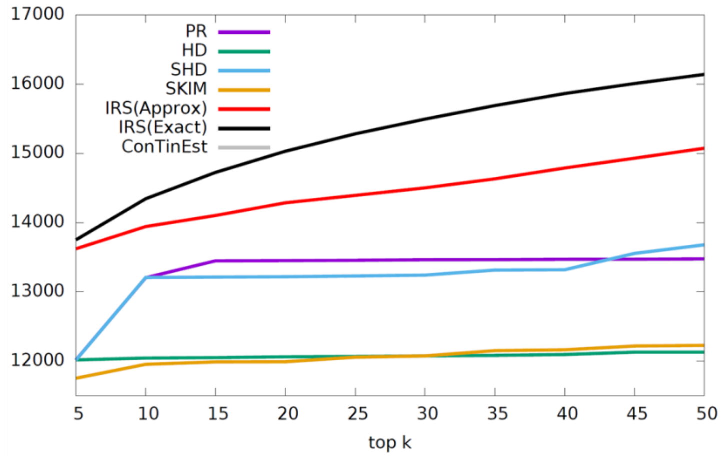
$k=6$, $\omega << n$

Efficiency Results

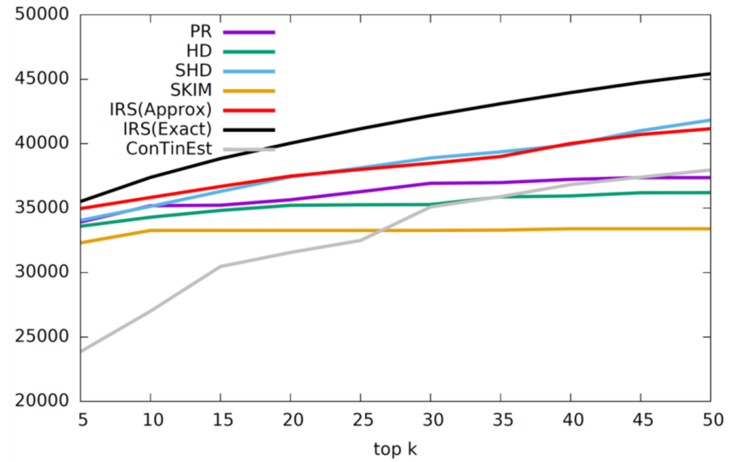
45 Million interactions in ~9 min in this laptop!!

Data Set	#Nodes(10^3)	Edges(10^3)	IRS	SIKM	PageRank	SHD	ConTinEst
twitter-US 2016	4,468	44,638	498	23.6	4261	3338	-
Enron	87	1148	93.7	2.2	49.4	8.1	1349
lkml-reply	27	1048	117.9	1.7	29.8	22.9	733
Facebook wall posts	47	877	10.3	1.1	35.6	2.9	790
twitter-higgs	304	526	2.2	4.3	29.8	1.5	3802
Slashdot threads	51	141	1.1	1.2	21.9	2.1	694

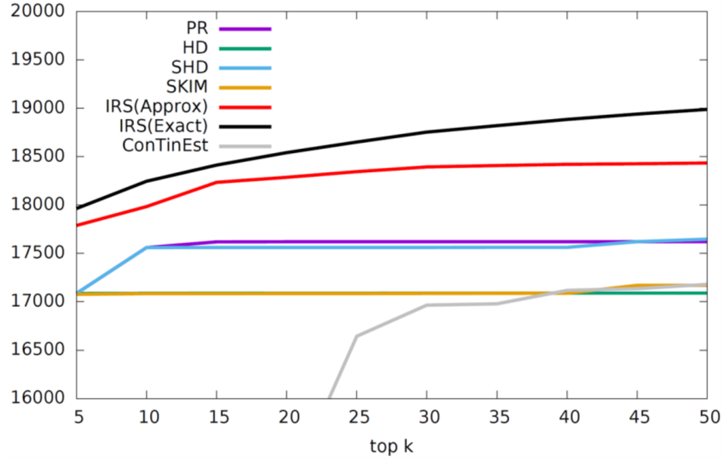
Effectiveness Results Using Time Constrained IC model.



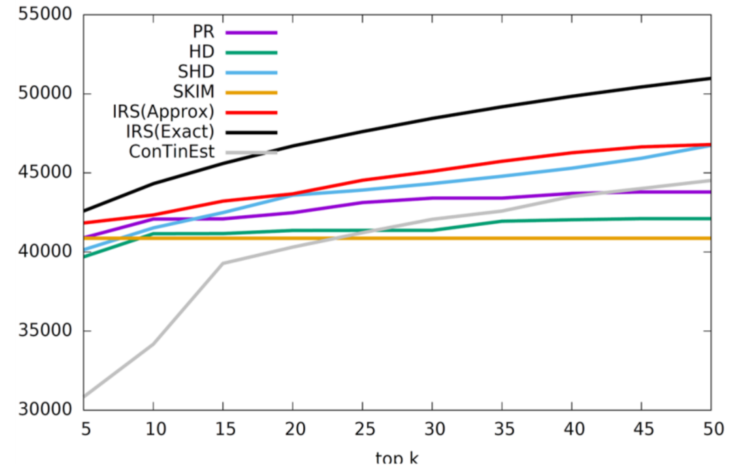
Lkml-reply (ω=1%)



Enron (ω=1%)



Lkml-reply (ω=20%)



Enron (ω=20%)

2. Towards Location Influence in Location-based Social Networks*

Check-ins

u, L , t1

..

.

.

v, L , t2

..

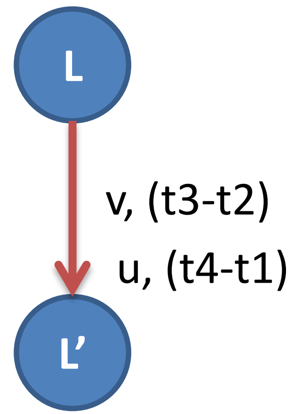
.

v, L’ , t3

..

.

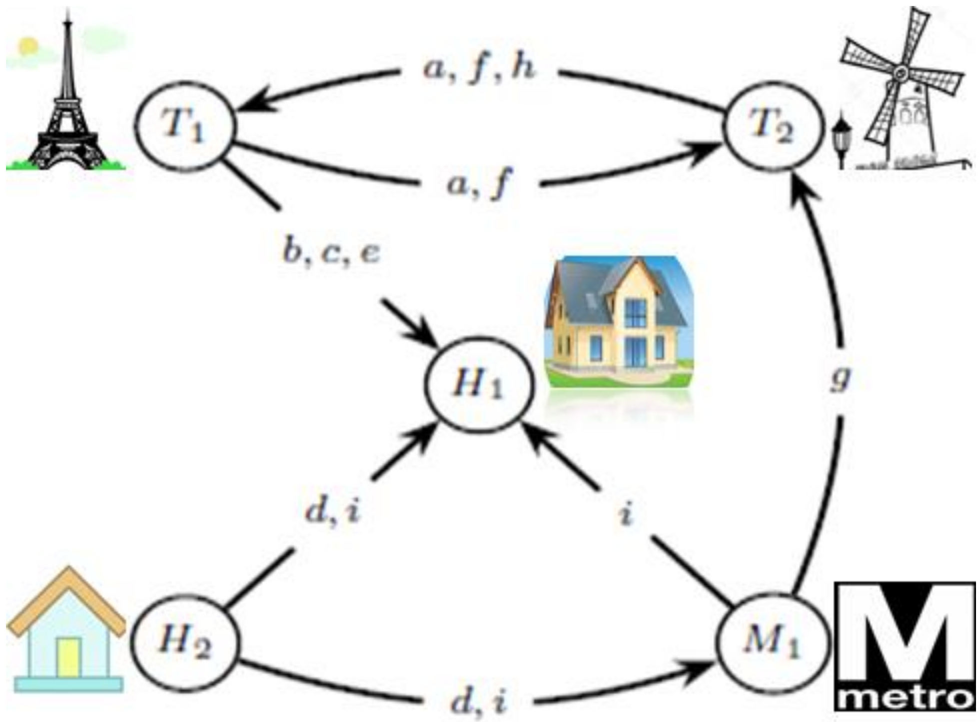
u, L’ , t4



*Joint work with Muhammad Aamir Saleem

3. Towards Location Influence in Location-based Social Networks*

Find top k locations.



*Joint work with Muhammad Aamir Saleem

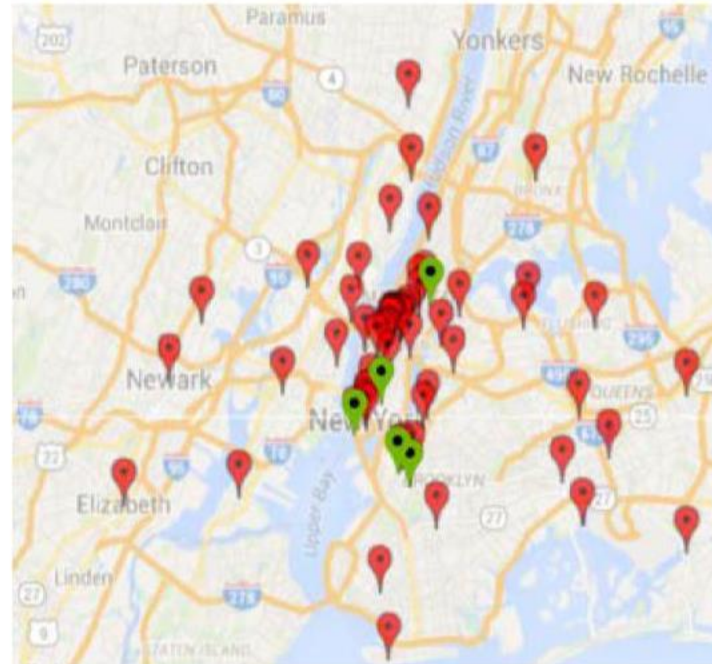
Modeling influence among locations

Influence Strength: Number of users travelling between the locations.

- **Absolute Influence Model:**
  - Influence exists if bridging visitors within a given time are greater than threshold
  - Example: T1 => T2 := |VB(T1,T2)| >= 2
- **Relative Influence Model:**
  - Biasness of popular locations, consider relative influence
  - Example: T1=> H1 := |b,c,e| / | b,c,e,i,d | >=0.4
- **Friendship-based Influence:**
  - Handle sparsity.
  - Predict future influence.

Influence spread

Naive BrightKite
(16 locations)

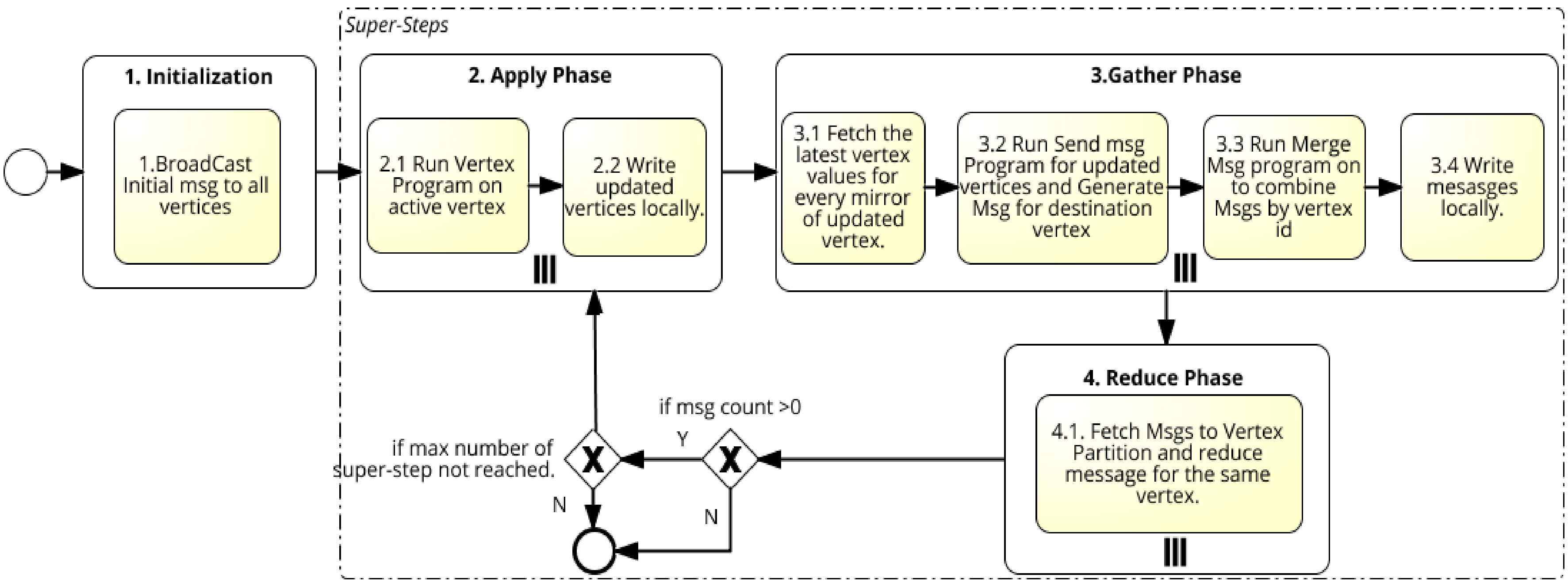


Our BrightKite
(72 locations)

Milestone 2 Topic 3

Rohit Kumar, Alberto Abello, and Toon Calders. Cost Model for Pregel on GraphX. **Accepted In ADBIS 2017.**

Cost Model for Pregel on GraphX



Cost Model

$$cPregel(V, E, s, A, P_e, P_v) := cInit(V, A, |P_v|) + \sum_{i=1}^{s} cSuperStep(V_i, E_i, A, M_{i-1}, P_e, P_v)$$

$$cSuperStep(V_i, E_i, A, M_{i-1}, P_e, P_v) := \max_{0 \le q \le |P_v|} \{cApply(V_i^q, M_{i-1}^q, A_v, P_e, P_v)\} + \max_{0 \le k \le |P_e|} \{cGather(E_i^k, M_i^k, V_i^k, A_s, A_m, P_e)\} + \max_{0 \le q \le |P_v|} \{cReduce(M_i^q, V_i^q, A_m, P_e, P_v)\}$$

Cost Model (Continued..)

$$cApply(V_i^q, M_{i-1}^q, A_v, P_v) := \sum_{v \in V_i^q} cVertexProg(v, M_{i-1}^q(v), A_v)$$
$$+ \beta_w \times \left\lceil \frac{\sum_{v \in V_i^{*q}} sizeOf(v) \times replication(v)}{B_s} \right\rceil + \alpha_1$$

Cost Model (Continued..)

$$\begin{aligned} cGather(E_i^k, M_i^k, V_i^k, A_s, A_m, P_e) := \beta_r \times & \sum_{v \in V_i^k \cap V_i^*} sizeOf(v) \\ &+ \sum_{(u,v) \in E_i^k} cSendProg(u, v, A_s) \\ &+ cProcess(M_i^k, V_i^k, A_m) \\ &+ \beta_w \times \left\lceil \frac{\sum_{m \in \widehat{M_i^k}} sizeOf(m)}{B_s} \right\rceil + \alpha_2 \end{aligned} \tag{4}$$

Where,

$$cProcess(M_i^k, V_i^k, A_m) := \sum_{v \in V_i^k} \left(|M_i^k(v)| - 1 \right) \times cMergeProg(A_m) \tag{5}$$

$$\begin{aligned} cReduce(M_i^q, V_i^q, A_m, P_e, P_v) := \gamma \times |M_i^q| \\ + cProcess(M_i^q, V_i^q, A_m) + \alpha_3 \end{aligned}$$

Cost model accuracy test

Used **Connected Component** algorithm with **CRVC** partitioning on **Twitter Euro** dataset to get the constants

Dataset	Algorithm	Partition Strategy		
		EdgePartition2D	CRVC	DBH
CollegeMsg	PageRank	96.4	97.9	97.7
	CC	97.6	96.1	96.7
twitter	PageRank	97.7	-	99.3
	CC	98.9	98.7	97.1
Higgs	PageRank	94.6	97.2	99.8
	CC	97.9	95.9	94.9

Outline

- Project Background
- Work done so far
- On going work
- Future work planned

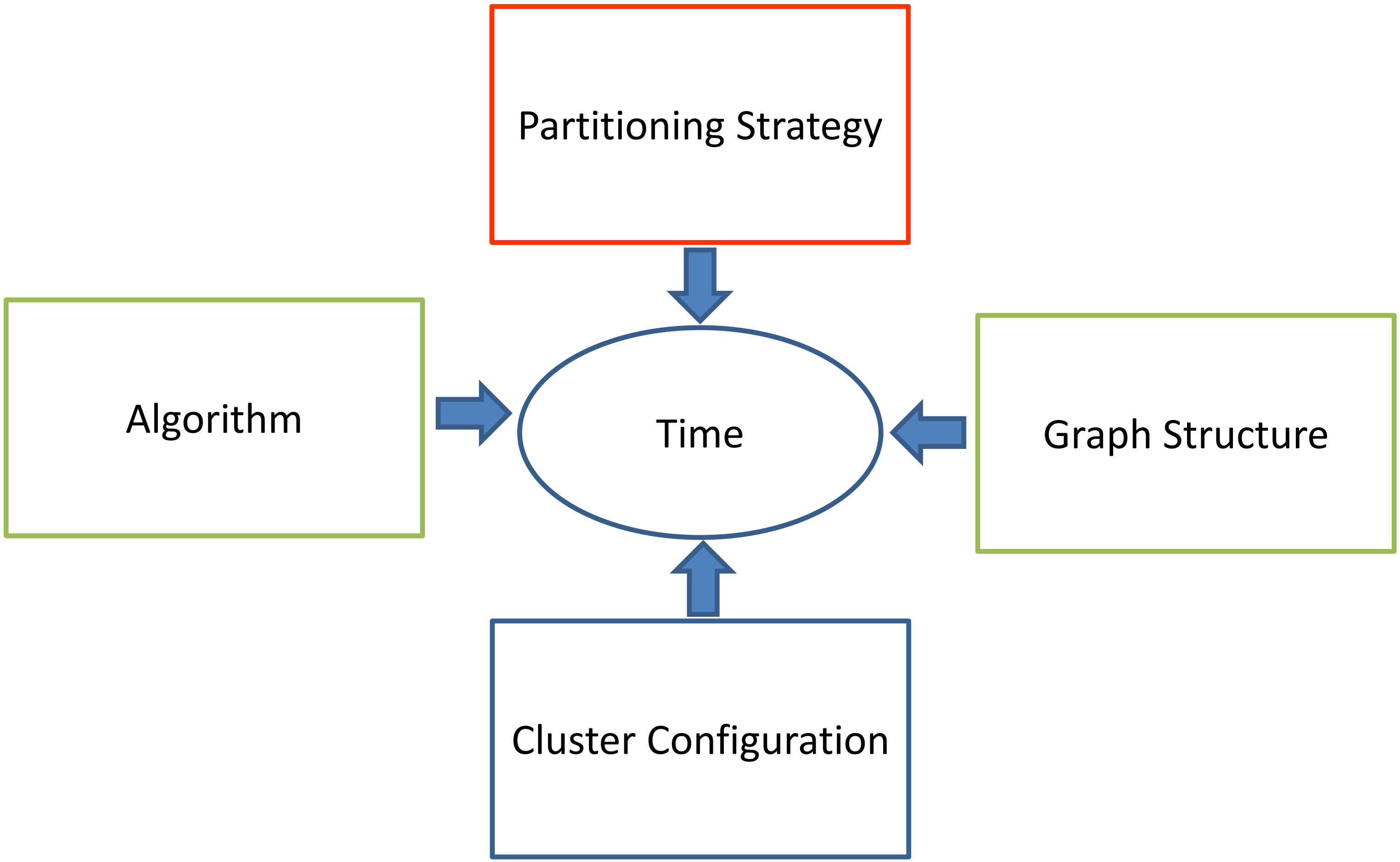
Milestone 2 Topic 4

Rule based graph partitioner for large distributed graph processing in Apache-GraphX.

Rohit Kumar, Alberto Abello, and Toon Calders.

In Journal version.

Rule derivation



Rule 1

Graph property – Low degree

Algo Property- High Communication (PageRank)

All function: same weight and is very fast

DBH is better

1) Writing messages in 2nd phase is less
2) Merge messages in 3rd phase is very less

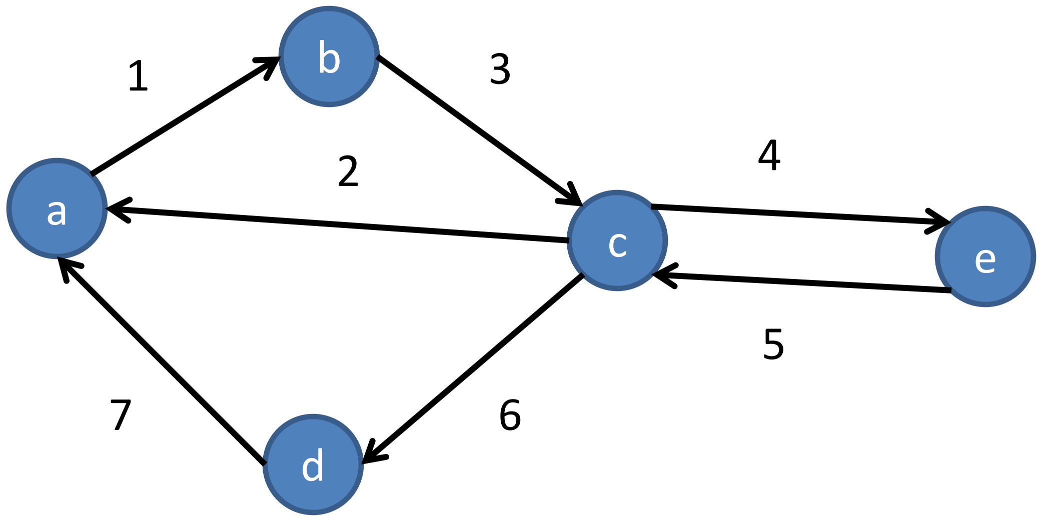
But if **mergeMsg** function is heavy

CRVC will be better

Milestone 3 Topic 5

- Rohit Kumar and Toon Calders. Finding simple temporal cycles in an interaction network. Submitted in TDLSG-ECML/PKDD 2017 (workshop).
- Rohit Kumar and Toon Calders. Efficient two phase approach to find simple temporal cycles in an interaction network. Planned to submit in Oct 2017 for WWW 2018 conference.

Temporal simple cycle

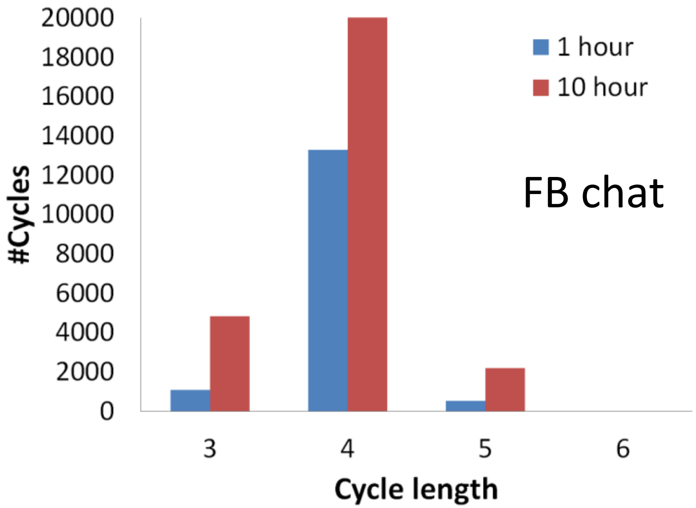


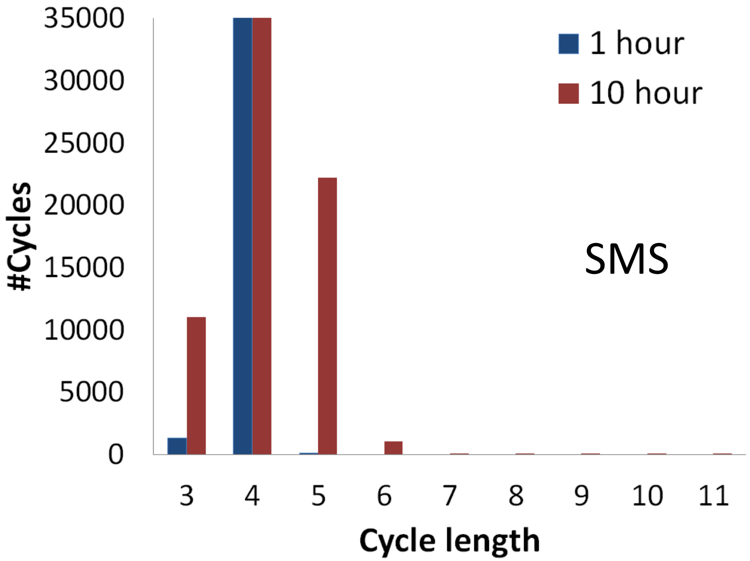
a->b->c->a   a->b->c->d->a   a->b->c->e->c->d->a

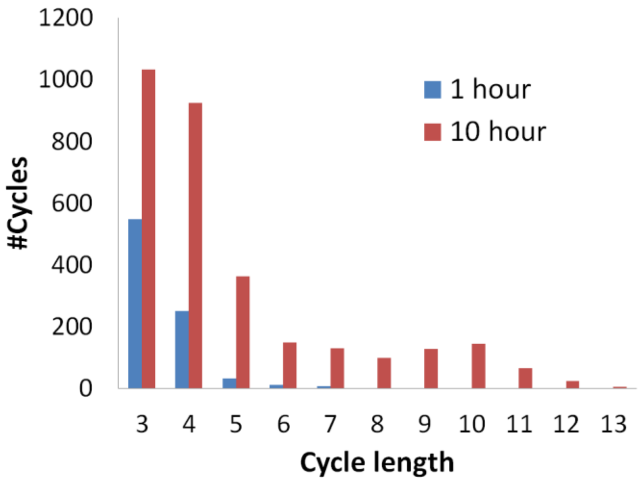
Why Cycles?

- Cyclic transactions are indication of financial frauds.
- In stock market trading cyclic patterns may indicate attempts to artificially create high trading volumes.
- To study information flow pattern in communication network.

Cycle frequency distribution







Retweet network

Outline

- Project Background
- Work done so far
- On going work
- Future work planned

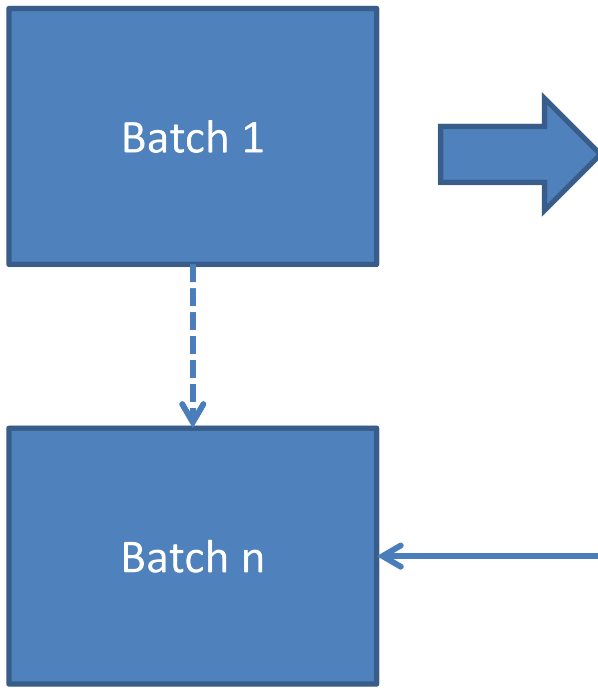
Milestone 3 Topic 6

Dynamic Repartition in GraphX for streaming graph. Rohit Kumar, Alberto Abello, Toon Calders.

Batch wise streaming.

Dynamic Repartition for batches

1. Use Rule based strategy to determine Partitioning strategy.



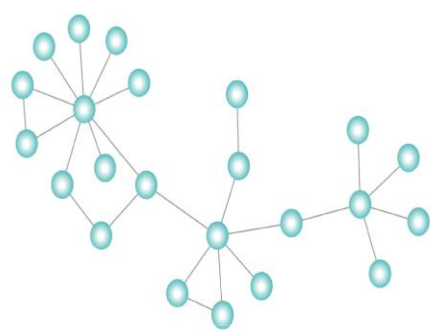
1. Calculate repartition cost.(C_R)
2. Estimate cost for new partitioner (C_{new})
3. Estimate cost using old partitioner. (C_{old})

If(C_{new} - C_{old})> $C_R + \varepsilon$
Use New partitioning.

Conclusion

- Analytical queries on temporal network need different approach then classical graph mining algorithm.
- Using window based snapshots on temporal graphs opens up interesting analytics problems.
- Using sketch based approximate solution are most of the time sufficient to solve the problem.
- A distributed system for evolving graph is missing and need to be addressed.

Thank you!



Graph Stream mining and processing

- Most of the existing graph mining algorithms require multiple pass over the graph data.
  - The method of multiple pass is not scalable for large graphs or graph stream.
- The graphs are becoming so large that it is difficult to store them in one single machine.
  - Traditional graph partitioning methods are one time partitioned and do not adopt to the changes in the graph.
- Proposed Approach
  - Study graph stream mining using the approximated graph sketch approach to create single pass graph mining algorithms.
  - Create a distributed graph processing framework which supports dynamic updates and adapts the partitioning with changes in graph.

Maintaining sliding-window neighborhood profiles in interaction networks

- In this paper we paper we presented a real-time monitoring of Neighborhood Profile of a node for a given time window in an interaction network. To address 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?
- We presented an online Algorithm to maintain Neighborhood profile of every node in the graph approximately.
- Working on the distributed version of the algorithm.
- Accepted in ECML/PKDD 2015
- Published source code at github
  - https://github.com/rohit13k/NeighborhoodProfile

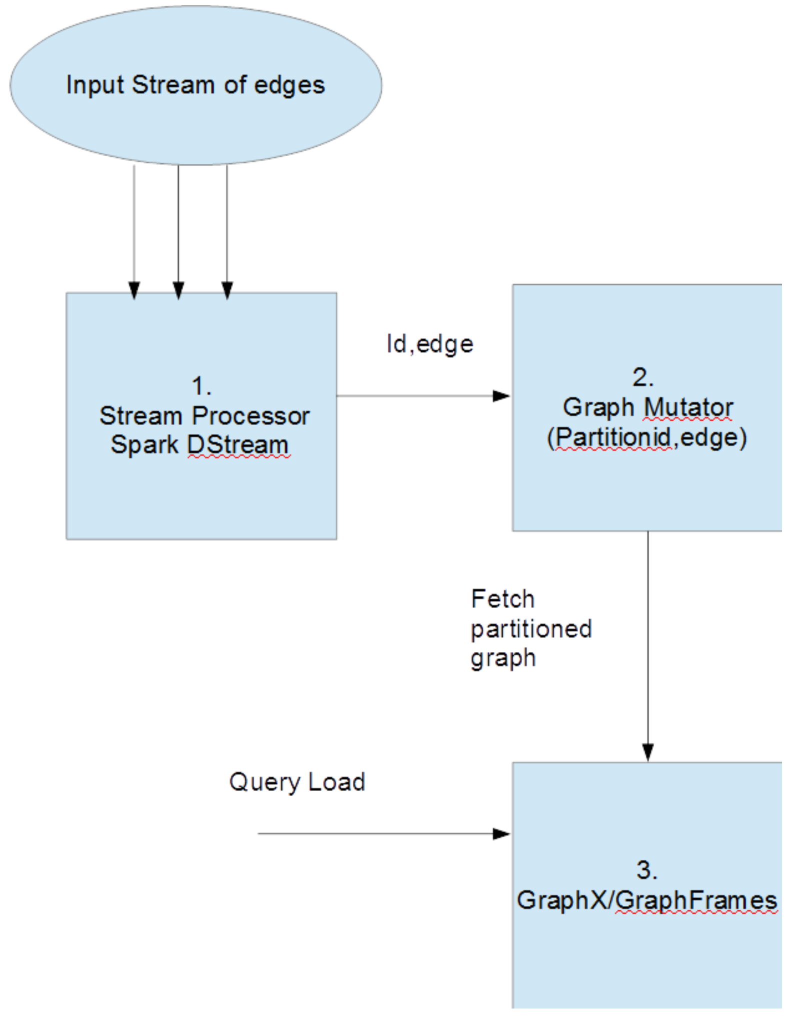
Information Propagation in Interaction Networks

- The main focus of this study is that given a interaction network and a life span of the information or topic :
  - Find out the top k influential users.
  - Find out the spread of influence given a starting set of nodes or users.
  - If the information or influence has reached a particular set of users or nodes find out the possible initiators.
- We presented an offline one pass algorithm to create Influence reachability set for every node in a interaction network to answer above queries.
- Submitted in KDD got rejected ☹!
- Working on addressing review comments.
- Planned to submit in EDBT in September.
- Published source code at
  - https://github.com/rohit13k/InfluencePropagation

Towards Location Influence in Location-based Social Networks

- In this study we analysis the user checkin activity stream generated by Location based social network to generate a Location to Location Interaction network. The type of queries we want to answer are:
  - Top K locations to maximize Influence Spread for Outdoor marketing.
  - Given a set of target locations find the minimum set of Source location to advertise so that all target locations are covered.
- We present an online incremental algorithm to generate location influence summary of each location to answer the above queries.
- We also present an offline one pass algorithm for a special case, which uses the data structure proposed in the previous paper.
- Paper ready for submission.
- Published source code at
  - https://github.com/rohit13k/LBSNAnalysisC
- Planning a demo version of this paper in ICDM 2016.

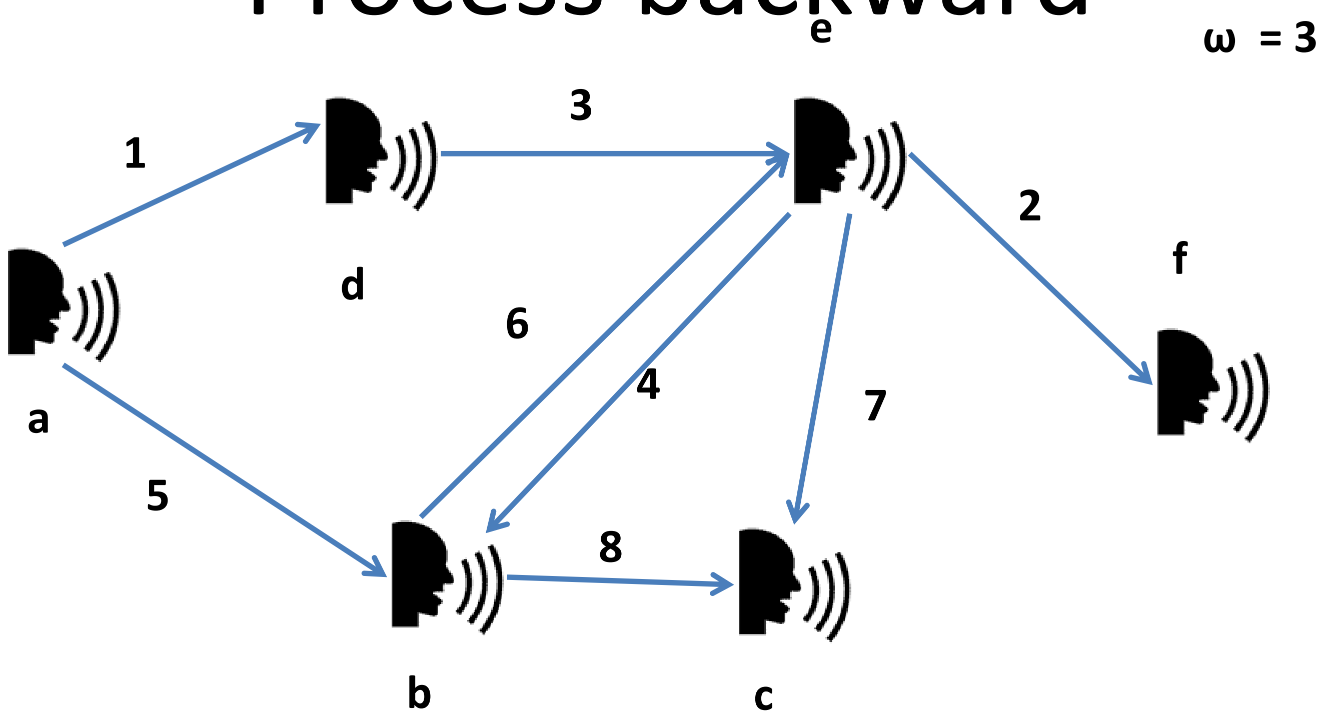
- Working on the platform for distributed graph processing for Dynamic graphs.



Information Propagation in Interaction Networks

- Algorithm details

Process backward



S(a) = b,5 e,6 c,7 d,1 e,3 b,4
S(b) = c,8 e,6 c,7
S(c) =
S(d) = e,3 b,4
S(e) = c,7 b,4 f,2
S(f) =

σ(a) = b,e,c,d
σ(b) = e,c
σ(c) =
σ(d) = e,b
σ(e) = b,c,f
σ(f) =

Algorithm

One pass algorithm!

1, (a,d)
2, (e,f)
3, (d,e)
4, (e,b)
5, (a,b)
6, (b,e)
7, (e,c)
8, (b,c)

$S(u) = \{(v, \lambda(u, v))\}$

$\lambda(u, v)$ is defined as the end time of the earliest information channel of length ω or less from u to v.

For an entry t,u,v:
  Add (S(u) , (v, t))
  S(u)=Merge(S(u),S(v))

Merge:
  For All $(x, t') \in S(v)$
    If $(t - t') < \omega$
      Add (S(u), (x, t'))