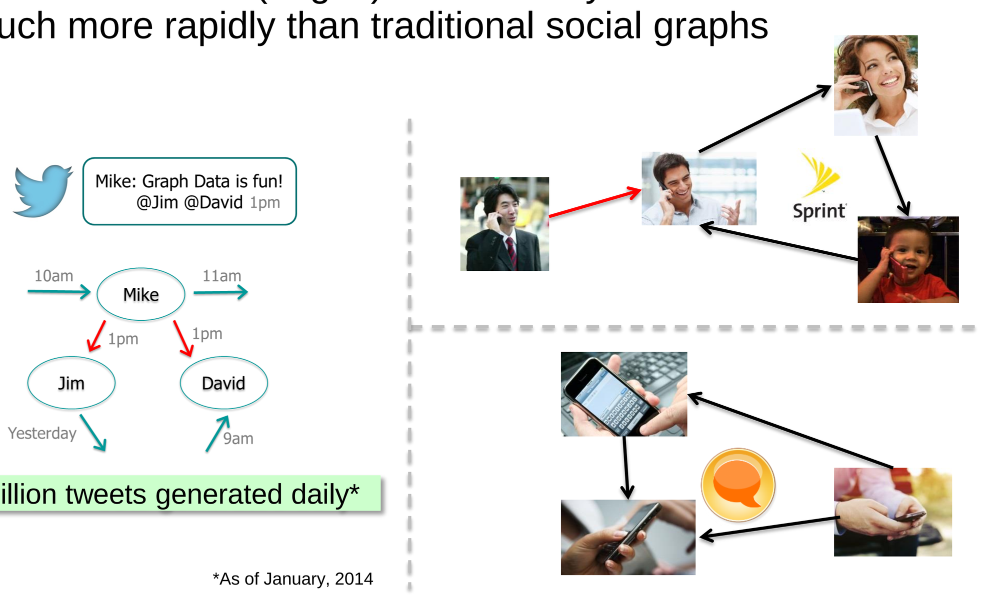
Dynamic Interaction Graphs with Probabilistic Edge Decay

Wenlei Xie[1], Yuanyuan Tian[2], Yannis Sismanis[3], Andrey Balmin[4], Peter J. Haas[2]
[1]Cornell University [2]IBM Almaden Research Center [3]Google, Inc. [4]Platfora, Inc.

Dynamic Interaction Graphs

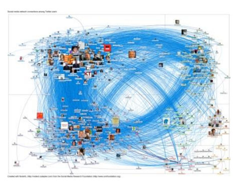
Social interactions can be modeled as graphs
- New interactions (edges) continuously added
- Much more rapidly than traditional social graphs



58 million tweets generated daily*

*As of January, 2014

Goal: Extract insight from data stream of interactions



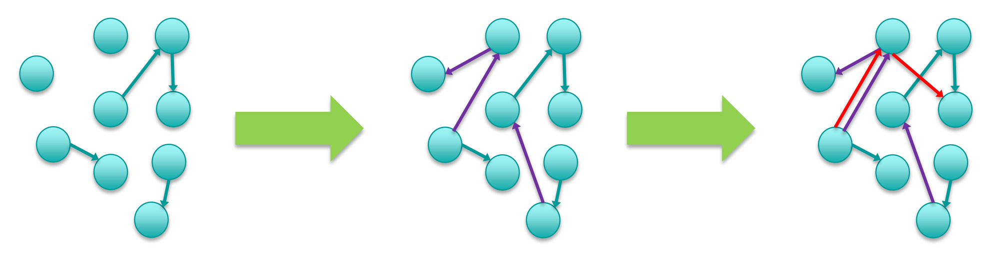
"Who are influencers on Twitter **now**?"

"What is the community structure on Twitter **now**?"

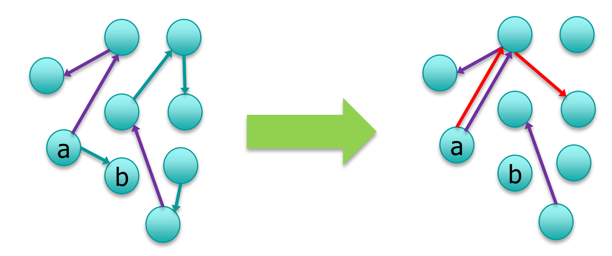
Challenge: Most graph mining algorithms assume static graph structures

Existing Models

Snapshot Model
- Consider all interactions seen so far
- Problem: Does not emphasize recent interactions (no recency)

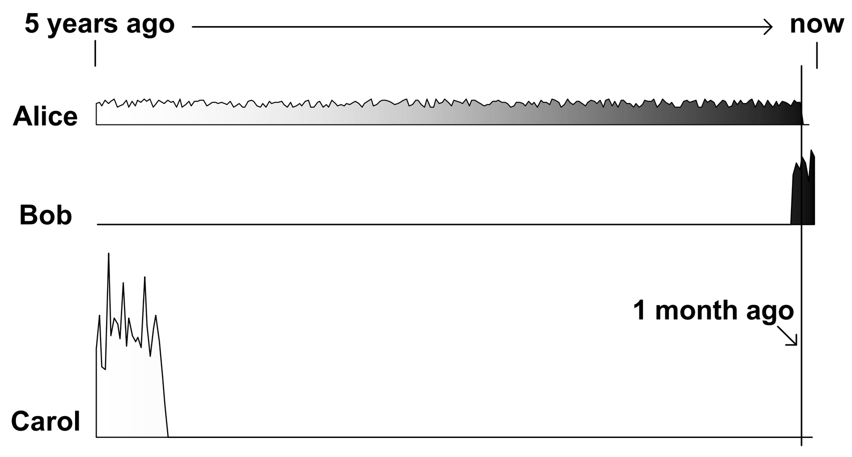


Sliding Window Model
- Consider recent interactions within a small time window
- Problem: Abruptly forgets past interactions (no continuity)



E.g., relationship between a and b is forgotten

Example: Influence Analysis



Alice: Temporarily dormant influencer
- Missed by Sliding Window Model

Bob: Rising star influencer
- Missed by Snapshot Model

Twitter data experiment: Either approach would miss ~25% of top influencers

Carol: Active in remote past, not an influencer at present
- Should she be totally forgotten?

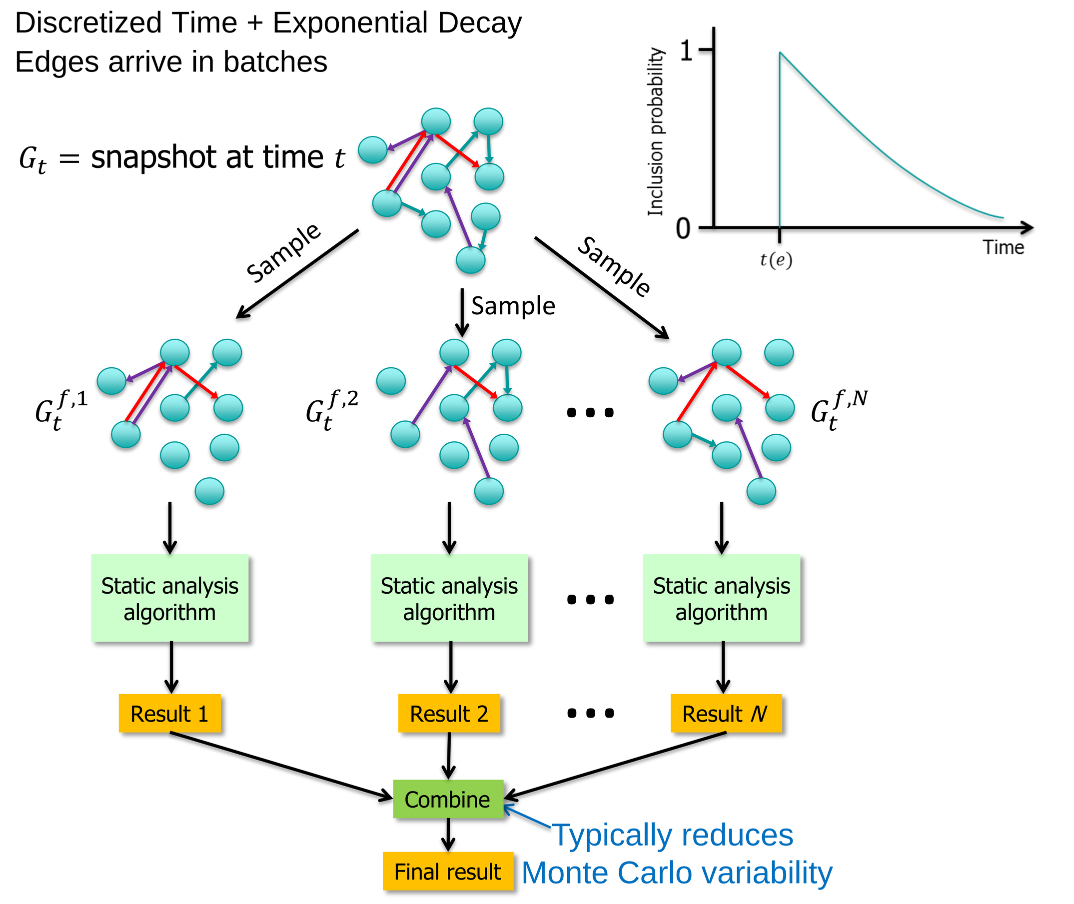
Problem: Binary View of an Edge's Role
- Included edges all have same importance regardless of how outdated they are
- *Impossible* to satisfy both recency and continuity

The Probabilistic Edge Decay Model

Key Idea: Temporally Biased Sampling
- Sample data items according to a probability that decreases over time
- Sample contains a relatively high proportion of recent interactions

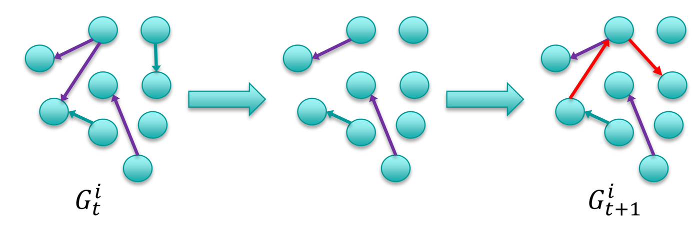
Probabilistic View of an Edge's Role — Breaking the Binary View of an Edge's Role
- All edges have chance to be considered (continuity)
- Outdated edges are less likely to be used (recency)
- Can systematically trade off recency and continuity
- Can use existing static-graph algorithms

TIDE: A distributed system for dynamic graph analysis
- Discretized Time + Exponential Decay
- Edges arrive in batches

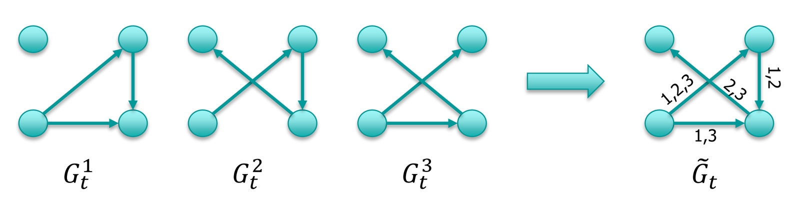


Maintaining Sample Graphs

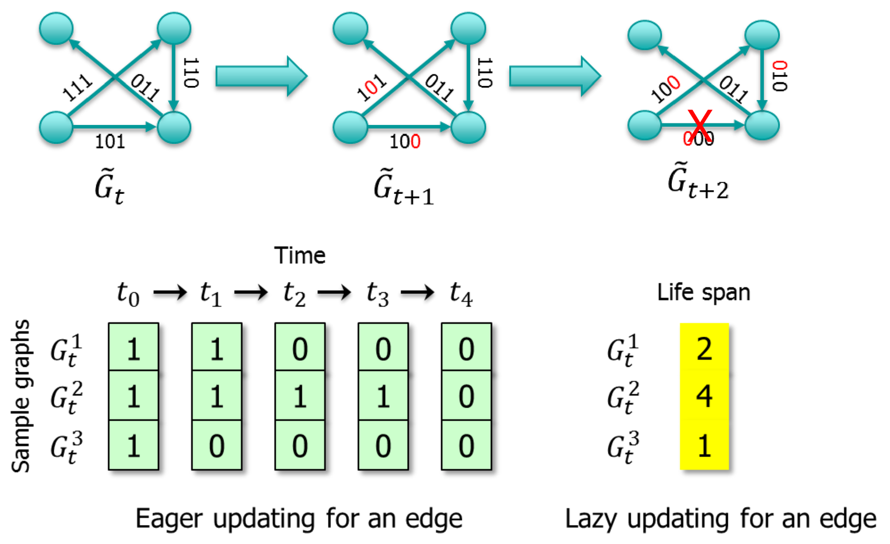
Idea #1: Exploit overlaps at successive time points



Idea #2: Exploit overlap between sample graphs at each time point [from $O(MN)$ to $O(M \log N)$ space bound]



Eager and Lazy Incremental Updating



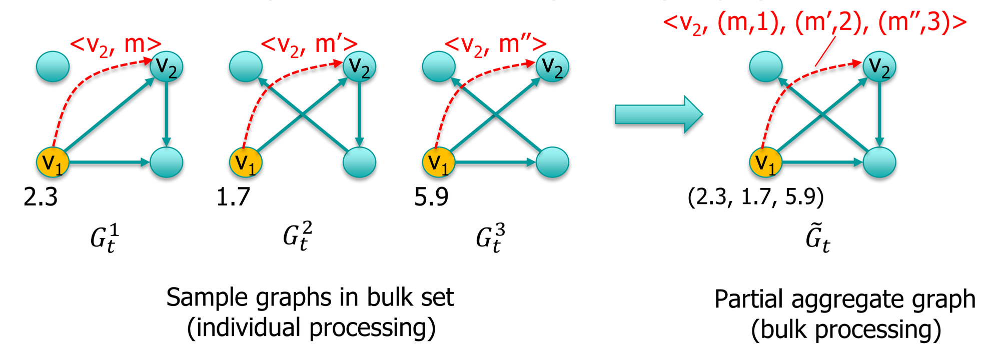
Efficient Analysis of Sample Graphs

Bulk Graph Execution Model

Think-as-a-vertex Model (Pregel, GraphLab. Trinity, GRACE, …)

High overlap between sample graphs → Similar topologies of sample graphs → Similar vertex and edge states during execution

Bulk execution: Compute results for multiple sample graphs simultaneously



Benefits via amortized extraction costs, memory locality, compression

Incremental Graph Analysis

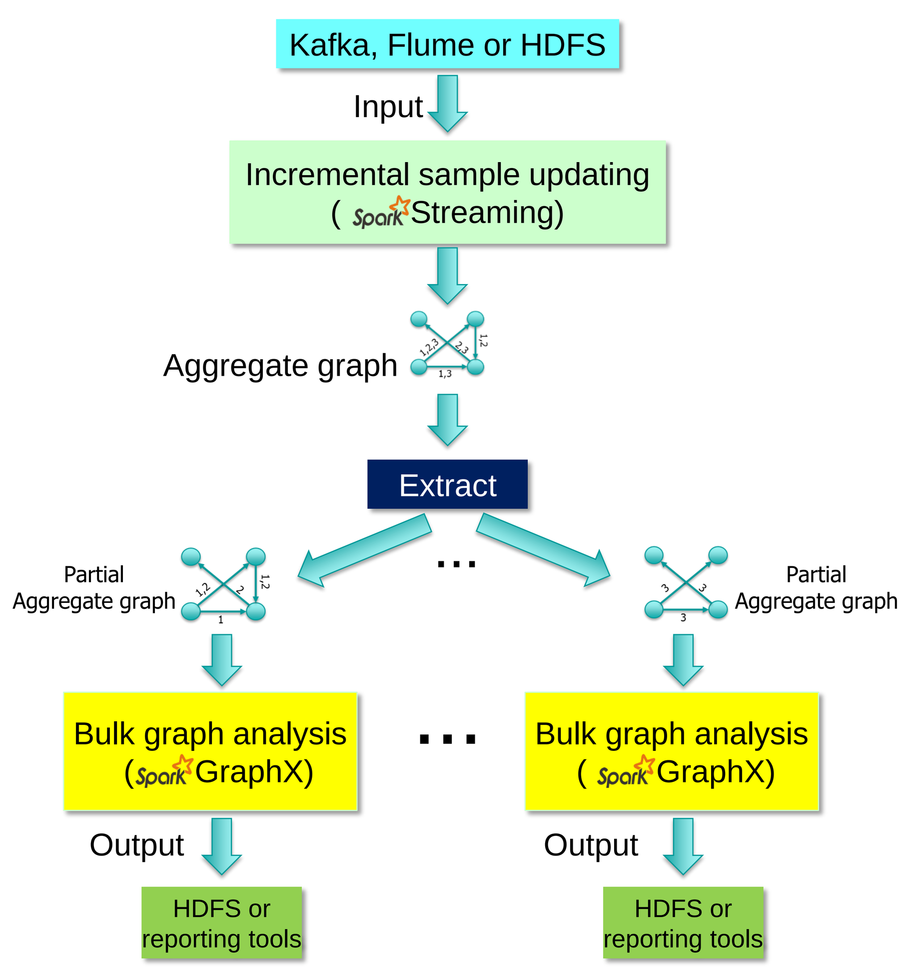
High edge overlap (p) of a sample graph at t and $t+1$ → Similar vertex and edge states during computation

Use final states at *t* as the starting states for computation at *t+1*

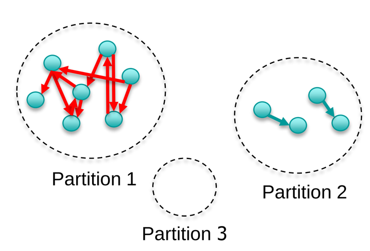
Example: Katz centrality for a random sample graph (*t* = 40)
- Computing from scratch: 28 iterations until convergence
- Initializing with final values from *t* = 39: 4 iterations

Caveat: Not applicable to all algorithms
- Same issue as in other dynamic graph processing systems

Implementation on Spark



Implementation Issues
- Maintaining the Aggregate Graph (Spark in-memory immutable RDD)
- In-place updates
- Location-aware balancing coalesce

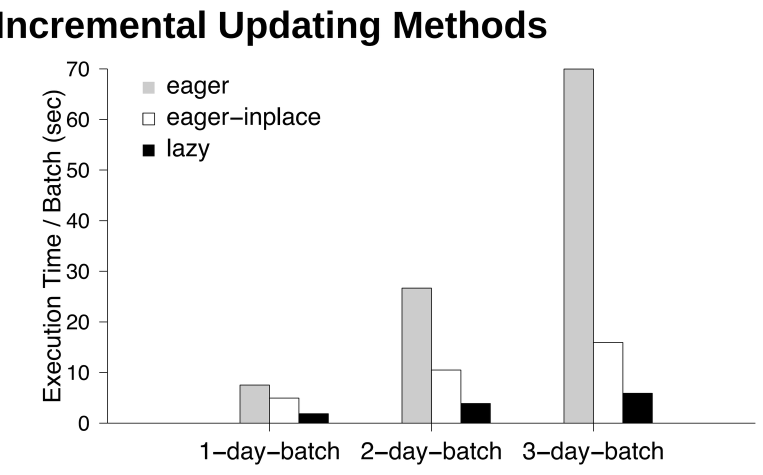


	Shuffle-based	Merge	Location-Aware
skewness	1.01	8.64	1.08
Time (sec)	120.62	0.84	1.84

Experiments

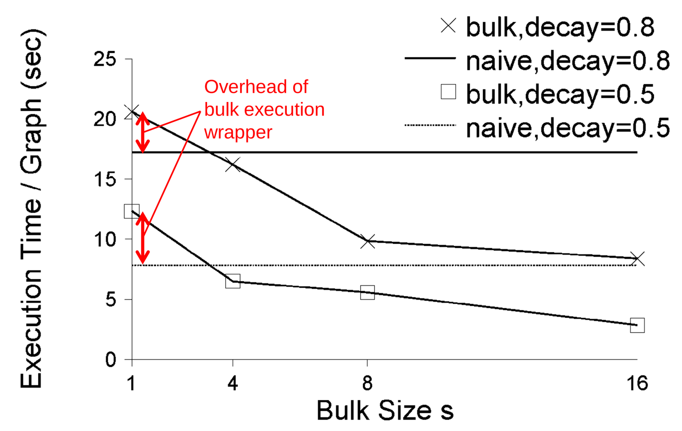
Setup
17 IBM System x iDataPlex DX 340 Servers
Twitter mention interactions: 10% of Sep 2011 to Feb 2012
- **Used 1-day, 2-day and 3-day batches to keep data large**
- **13.9 million interactions per day on average**

Incremental Updating Methods



Average per-batch time for incremental updating (After aggregate-graph stabilization at 30th batch)

Bulk Graph Execution

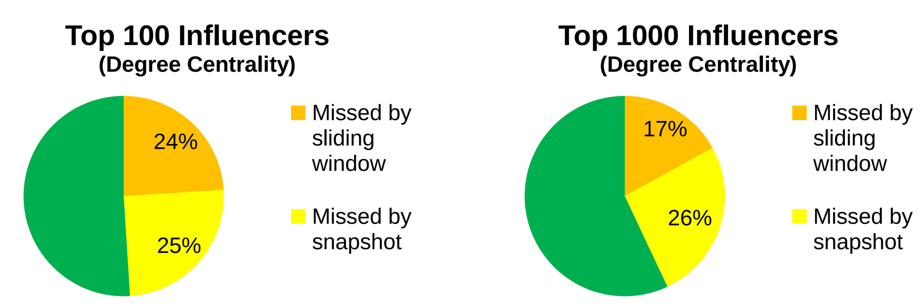


Average running time per iteration for Katz centrality

Conclusions

Novel probabilistic decay model (PED)
- Extends temporally biased sampling to graphs
  - Generalizes existing snapshot and sliding-window model
  - Allows controlled trade off between recency and continuity
- Allows direct application of static algorithms to dynamic setting

Benefit of PED Approach (Empirical Twitter Data)



Missed by the sliding window model:
- Many temporarily dormant influencers (like Alice)

Missed by snapshot model:
- Many rising star influencers (like Bob)

Methods to efficiently maintain and compute the results
- Exploit overlaps between sample graphs at each time t
- Exploit overlaps of a given sample graph at different time points

TIDE
- An end-to-end distributed system for analyzing dynamic graphs
- Prototype implementation on Spark

Future work
- General decay functions (some results already extend)
- Extend techniques for analyzing sample graphs