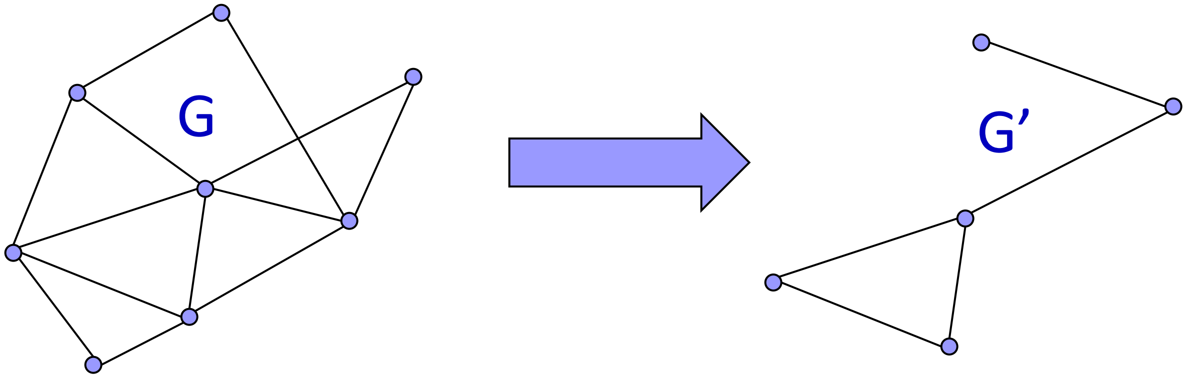
Streaming Algorithms for Matching Size in Sparse Graphs

Graham Cormode

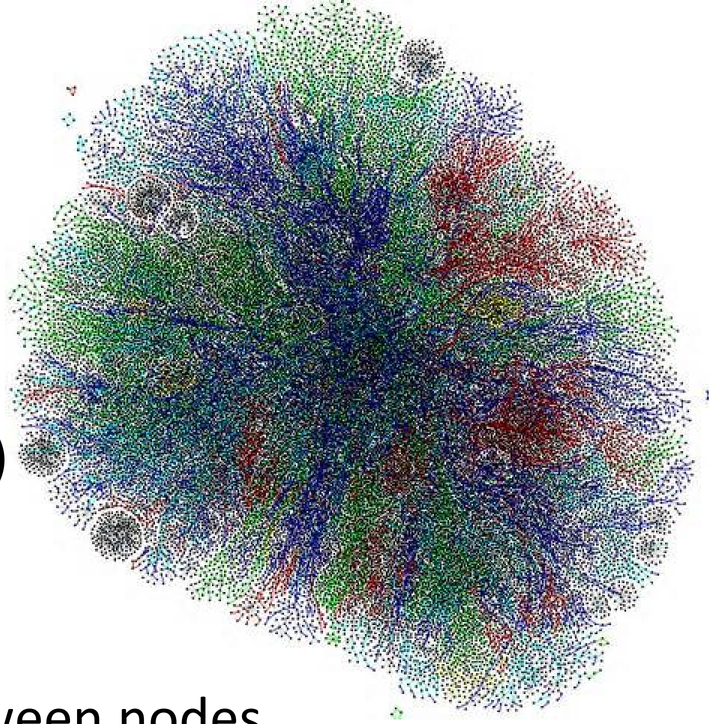
g.cormode@warwick.ac.uk

Joint work with

S. Muthukrishnan (Rutgers), Morteza Monemizadeh (Rutgers → Amazon)
Hossein Jowhari (Warwick → K. N. Toosi U. of Technology)

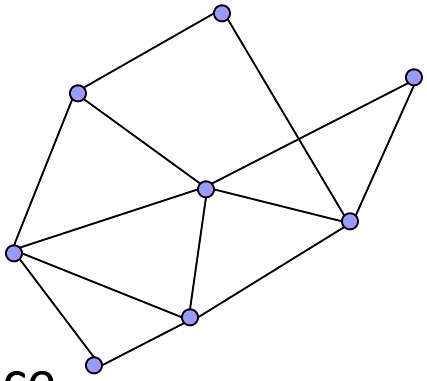


Big Graphs



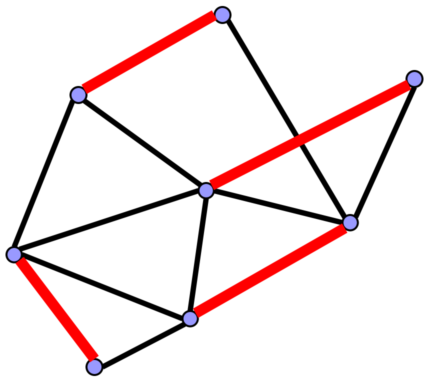
- Increasingly many "big" graphs:
  - Internet/web graph (2^{64} possible edges)
  - Online social networks (10^{11} edges)
- Many natural problems on big graphs:
  - Connectivity/reachability/distance between nodes
  - Summarization/sparsification
  - Traditional optimization goals: vertex cover, maximal matching
- Various models for handling big graphs:
  - Parallel (BSP/MapReduce): store and process the whole graph
  - Sampling: try to capture a subset of nodes/edges
  - Streaming (this work): seek a compact summary of the graph

Streaming graph model



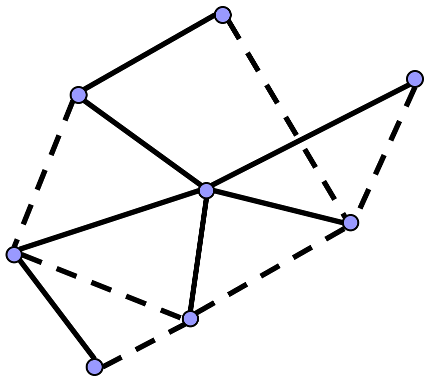
- The "you get one chance" model:
  - Vertex set [n] known, see each edge only once
  - Space used must be sublinear in the size of the input
  - Analyze costs (time to process each edge, accuracy of answer)
- Variations within the model:
  - See each edge exactly once or at least once?
    - Assume exactly once, this assumption can be removed
  - Insertions only, or edges added and deleted?
  - How sublinear is the space?
    - Semi-streaming: linear in n (nodes) but sublinear in m (edges)
    - "Strictly streaming": sublinear in n, polynomial or logarithmic
- Many problems "hard" (space lower bounds) for graph streaming

Streaming Matching



- Aim to find a matching for the input graph
  - Subgraph with maximum degree 1
- Easy linear space 2-approximation in insert-only
  - Just greedily construct a matching, $O(n)$ space
- We seek to approximate the size of the matching in $o(n)$ space
  - Kapralov, Khanna, Sudan, SODA'14: $O(\text{poly} \log n)$ approx in $O(\text{poly} \log n)$ space, assuming random order of arrivals
  - Esfandiari et al., SODA'15 : $O(c)$ approximation in $O(c\, n^{2/3})$ space, assuming graph has c-bounded arboricity
  - Bury and C. Schwiegelshohn, ESA'15: Weighted graphs
  - McGregor and Vorotnikova, APPROX'16: Improved constant factors

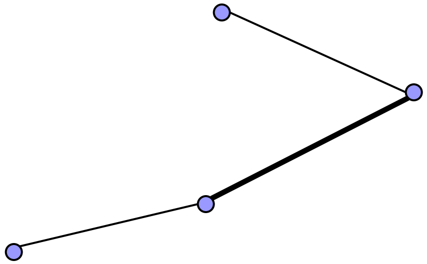
Matching under sparsity



- Many graphs (phone, web, social) are 'sparse'
  - Asymptotically fewer than $O(n^2)$ edges
- Characterize sparsity by bounded arboricity c
  - Edges can be partitioned into at most c forests
  - Equivalent to the largest local density, $|E(U)|/(|U|-1)$ for $U \subseteq V$
    - E(U) is the number of edges in the subgraph induced by U
  - E.g. planarity corresponds to 3-bounded arboricity
- Use structural properties of graph streams to give results
  - Improved poly. space algorithm for matching with deletions
  - First polylog space algorithm for matching with inserts only

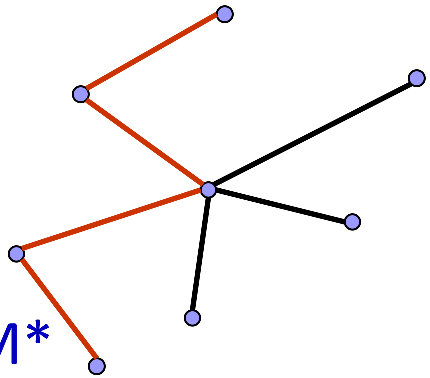
α -Goodness

- Define an edge in a stream to be α-good if neither of its endpoints appears more than α times in the suffix of the input
  - Intuition: This definition sparsifies the graph but approximately preserves the matching
- The number of α-good edges approximates the matching size
  - Edges on low degree nodes are already α-good
  - Every high degree node has at most α+1 α-good edges
  - Estimating the number of α-good edges is easier than finding the matching itself



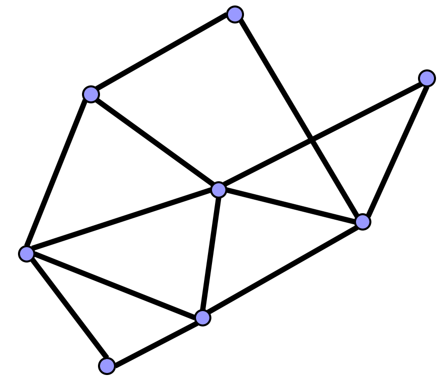
Edge is 1-good if at most 1 edge on each endpoint arrives later

Easy case: trees (c=1)



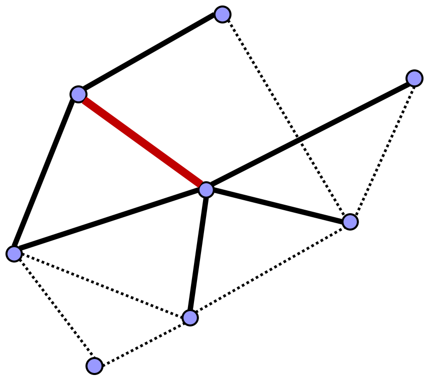
- Consider a tree T with maximum matching size M^*
- $|E_1| \le 2M^*$: The subgraph E_1 has degree at most 2, no cycles
  - So can make a matching for T from E_1 using at least half the edges
- $|E_1| \ge M^*$: Proof by induction on number of nodes n
  - Base case: $n=2$ is trivial
  - Inductive case: add an edge (somewhere in the stream) that connects a new leaf to an existing node
    - Either M^* and $|E_1|$ stay the same, or $|E_1|$ increases by 1 and M^* increases by at most 1
    - At most 1 edge is ejected from E_1, but the new edge replaces it

General case



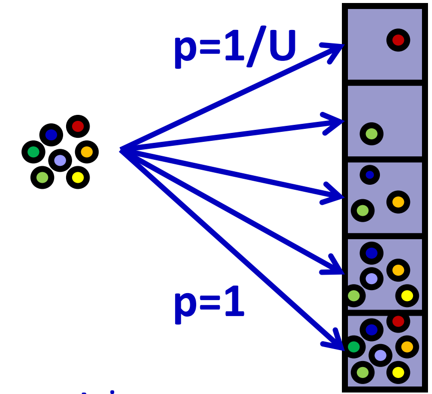
- Upper bound: $|E_{6c}| \leq (22.5c + 6)/3\ M^*$
  - E_α has degree at most $\alpha+1$, and invoke a bound on M^* [Han 08]
- Lower bound: $M^* \leq 3|E_{6c}|$
  - Break nodes into low L and high degree H classes (as before)
  - Relate the size of a maximum matching to number of high degree nodes plus edges with both ends low degree
  - Define HH: the nodes in H that only link to others in H
    - There must still be plenty of these by a counting argument
  - Use bounded arboricity to argue that half the nodes in HH have degree less than 6c (averaging argument)
  - These must all have a 6c-good edge (not too many neighbors)
- Combine these to conclude $M^* \leq 3|E_{6c}| \leq (22.5c + 6)M^*$

Testing edges for α-Goodness



- To estimate matching size, count number of α-good edges
- Follow a sampling strategy similar to L_0 sampling
  - Uniformly sample an edge (u, v) from the stream (easy to do)
  - Count number of subsequent edges incident on u and v
  - Terminate procedure if more than α incident edges
- Need to sample many times in parallel to get result
  - Sample rate too low: no edges found are α-good
  - Sample rate too high: space too high
    - But we can drop the instances that fail
- Goldilocks effect: We can find a sample rate that is just right
  - And bound the space of the over-sampling instances

Parallel guessing



- Make parallel guesses of sampling rates p_i
  - Run $1/\varepsilon \log n$ guesses with sampling rates $p_i = (1+\varepsilon)^{-i}$
  - Terminate level i if more than $O(\alpha \log(n)/\varepsilon^2)$ guesses are active
- Estimate: Use lowest non-terminated level to make estimate
- Correctness: there is a 'good' level that will not be terminated
  - E_α not monotone! Might go up and down as we see more edges
  - But the matching size only increases as the stream goes on
  - Use the previous analysis relating E_α to matching size to bound
  - Also argue that using other levels to estimate is OK
- Result: use $O(c/\varepsilon^2 \log n)$ space to $O(c)$ approximate M^*

Matching with deletions

- We assume not too many deletions: bounded by $O(\alpha n)$
- Our algorithm samples nodes into a set T with probability p
- In parallel as insertions/deletions of edges arrive, maintain:
  1. The induced subgraph on T
  2. The cut edges between T and degrees of neighbors of T
  3. A matching of size at most $1/p$
- Via arboricity assumption, nodes have expected degree $O(\alpha)$
- Matching (3) maintained via randomized algorithm in space $O(p^{-2})$
- Result: Balancing the space costs sets $p = n^{-1/3}$, total space $O(n^{2/3})$
  – Estimate matching size by #high degree nodes + #low degree edges
  – Maintained statistics are sufficient to $O(\alpha^2)$ approximate matching size based on number of surviving high degree nodes

Open Problems

- Work in progress: improve constants and simplify analysis [McGregor and Vorotnikova: connection to fractional matchings]
- Extensions to the parallel/distributed case
  - Obstacle: α-good definition seems inherently centralized
- Other notions of structure/sparsity beyond arboricity?
- Extend to the weighted matching case: some recent results here
- Connections between the streaming and online models?
- Cardinality estimation for other graph problems, e.g.:
  - Maximum Independent Set
  - Dominating Set

Thank you!