CS630: Graduate Algorithms

Charalampos E. Tsourakakis

Dec. 12th, 2023

What this lecture is about ...

given a graph (network), static or dynamic

(social network, biological network, information network, ...)

find a subgraph that ...

... has many edges

... is densely connected

why I care?

what does dense mean?

review of DSP and algorithm design

Outline

- motivation
- formulation
- preliminaries: maximum flow
- exact solution
- greedy 2-approximation
- LP

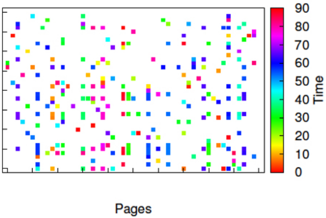
Motivation – correlation mining

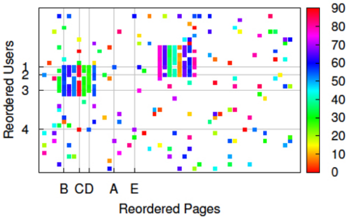
correlation mining: a general framework with many applications

- data is converted into a graph
- vertices correspond to entities
- an edge between two entities denotes strong correlation
  1. stock correlation network: data represent stock timeseries
  2. gene correlation networks: data represent gene expression
- dense subsets of vertices correspond to highly correlated entities
- applications:
  1. analysis of stock market dynamics
  2. detecting co-expression modules

Motivation – fraud detection

- dense bipartite subgraphs in page-like data reveal attempts to inflate page-like counts [Beutel et al., 2013]





source: [Beutel et al., 2013]

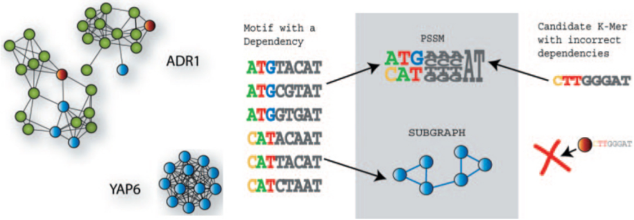
Motivation – e-commerce

e-commerce

- weighted bipartite graph $G(A \cup Q, E, w)$
- set A corresponds to advertisers
- set Q corresponds to queries
- each edge (a, q) has weight $w(a, q)$ equal to the amount of money advertiser a is willing to spend on query q

large almost bipartite cliques correspond to sub-markets

Motivation – bioinformatics



- DNA motif detection [Fratkin et al., 2006]
  - vertices correspond to k-mers
  - edges represent nucleotide similarities between k-mers
- gene correlation analysis
- detect complex annotation patterns from gene annotation data [Saha et al., 2010]

Motivation – frequent pattern mining

- given a set of transactions over items
- find item sets that occur together in a θ fraction of the transactions

issue number	heroes
1	Iceman, Storm, Wolverine
2	Aurora, Cyclops, Magneto, Storm
3	Beast, Cyclops, Iceman, Magneto
4	Cyclops, Iceman, Storm, Wolverine
5	Beast, Iceman, Magneto, Storm

e.g., {Iceman, Storm} appear in 60% of issues

Motivation – frequent pattern mining

- one of the most well-studied area in data mining
- many efficient algorithms
  Apriori, Eclat, FP-growth, Mafia, ABS, ...
- main idea: monotonicity
  a subset of a frequent set must be frequent, or
  a superset of an infrequent set must be infrequent
- algorithmically:
  start with small itemsets
  proceed with larger itemset if all subsets are frequent
- enumerate all frequent itemsets

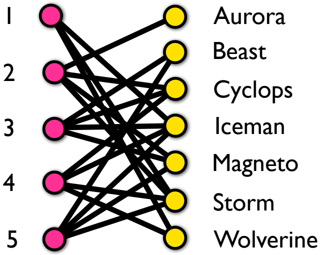
Motivation – frequent itemsets and dense subgraphs

id	heroes
1	Iceman, Storm, Wolverine
2	Aurora, Cyclops, Magneto, Storm
3	Beast, Cyclops, Iceman, Magneto
4	Cyclops, Iceman, Storm, Wolverine
5	Beast, Iceman, Magneto, Storm

$\Leftrightarrow$

	A	B	C	I	M	S	W
1	0	0	0	1	0	1	1
2	1	0	1	1	1	0	0
3	0	1	1	1	1	0	0
4	0	0	1	1	0	1	1
5	0	1	0	1	1	1	0

$\Leftrightarrow$



- transaction data $\Leftrightarrow$ binary data $\Leftrightarrow$ bipartite graphs

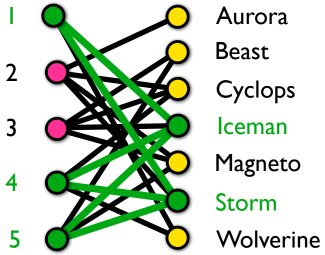
Motivation – frequent itemsets and dense subgraphs

id	heroes
1	Iceman, Storm, Wolverine
2	Aurora, Cyclops, Magneto, Storm
3	Beast, Cyclops, Iceman, Magneto
4	Cyclops, Iceman, Storm, Wolverine
5	Beast, Iceman, Magneto, Storm

$\Leftrightarrow$

	A	B	C	I	M	S	W
1	0	0	0	1	0	1	1
2	1	0	1	1	1	0	0
3	0	1	1	1	1	0	0
4	0	0	1	1	0	1	1
5	0	1	0	1	1	1	0

$\Leftrightarrow$



- transaction data $\Leftrightarrow$ binary data $\Leftrightarrow$ bipartite graphs
- frequent itemsets $\Leftrightarrow$ bi-cliques

Preliminaries, measures of density

notation

- graph $G = (V, E)$ with vertices V and edges $E \subseteq V \times V$
- degree of a node $u \in V$ with respect to $X \subseteq V$ is

$$\deg_X(u) = |\{v \in X \text{ such that } (u, v) \in E\}|$$

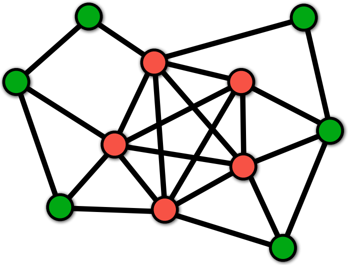
- degree of a node $u \in V$ is $\deg(u) = \deg_V(u)$
- edges between $S \subseteq V$ and $T \subseteq V$ are

$$E(S, T) = \{(u, v) \text{ such that } u \in S \text{ and } v \in T\}$$

  use shorthand $E(S)$ for $E(S, S)$
- graph cut is defined by a subset of vertices $S \subseteq V$
- edges of a graph cut $S \subseteq V$ are $E(S, \bar{S})$
- induced subgraph by $S \subseteq V$ is $G(S) = (S, E(S))$
- triangles: $T(S) = \{(u, v, w) \mid (u, v), (u, w), (v, w) \in E(S)\}$

density measures

- undirected graph $G = (V, E)$
- subgraph induced by $S \subseteq V$
- clique: all vertices in S are connected to each other



density measures

- edge density (average degree):

$$d(S) = \frac{2\,|E(S,S)|}{|S|} = \frac{2\,|E(S)|}{|S|}$$

(sometimes just drop 2)

- edge ratio:

$$\delta(S) = \frac{|E(S,S)|}{\binom{|S|}{2}} = \frac{|E(S)|}{\binom{|S|}{2}} = \frac{2\,|E(S)|}{|S|(|S|-1)}$$

- triangle density:

$$t(S) = \frac{|T(S)|}{|S|}$$

- triangle ratio:

$$\tau(S) = \frac{|T(S)|}{\binom{|S|}{3}}$$

other density measures

- k-core: every vertex in S is connected to at least k other vertices in S
- α-quasiclique: the set S has at least $\alpha \binom{|S|}{2}$ edges

  i.e., S is α-quasiclique if $E(S) \geq \alpha \binom{|S|}{2}$

reminder: min-cut and max-cut problems

min-cut problem

- source $s \in V$, destination $t \in V$
- find $S \subseteq V$, s.t.,
- $s \in S$ and $t \in \bar{S}$, and
- minimize $e(S, \bar{S})$

max-cut problem

- find $S \subseteq V$, s.t.,
- maximize $e(S, \bar{S})$

reminder: min-cut and max-cut problems

min-cut problem

- source $s \in V$, destination $t \in V$
- find $S \subseteq V$, s.t.,
- $s \in S$ and $t \in \bar{S}$, and
- minimize $e(S, \bar{S})$
- polynomially-time solvable
- equivalent to max-flow problem

max-cut problem

- find $S \subseteq V$, s.t.,
- maximize $e(S, \bar{S})$

reminder: min-cut and max-cut problems

min-cut problem

- source $s \in V$, destination $t \in V$
- find $S \subseteq V$, s.t.,
- $s \in S$ and $t \in \bar{S}$, and
- minimize $e(S, \bar{S})$
- polynomially-time solvable
- equivalent to max-flow problem

max-cut problem

- find $S \subseteq V$, s.t.,
- maximize $e(S, \bar{S})$
- **NP**-hard
- approximation algorithms
  (0.5 last time, 0.868 based on SDP)

Efficient algorithms for static graphs

Goldberg's algorithm for densest subgraph

- consider first degree density d
- on the transformed instance:
- is there a cut smaller than a certain value?

- is there a subgraph S with $d(S) \geq c$?
- transform to a min-cut instance

Goldberg's algorithm for densest subgraph

is there S with $d(S) \geq c$?

$$\begin{aligned}
\frac{2\,|E(S,S)|}{|S|} &\geq c \\
2\,|E(S,S)| &\geq c|S| \\
\sum_{u \in S} \deg(u) - |E(S,\bar{S})| &\geq c|S| \\
\sum_{u \in S} \deg(u) + \sum_{u \in \bar{S}} \deg(u) - \sum_{u \in \bar{S}} \deg(u) - |E(S,\bar{S})| &\geq c|S| \\
\sum_{u \in \bar{S}} \deg(u) + |E(S,\bar{S})| + c|S| &\leq 2\,|E|
\end{aligned}$$

Goldberg's algorithm for densest subgraph

- transformation to min-cut instance

- is there S s.t. $\sum_{u \in \bar{S}} \deg(u) + |e(S, \bar{S})| + c|S| \leq 2\,|E|$?

Goldberg's algorithm for densest subgraph

- transform to a min-cut instance

- is there S s.t. $\sum_{u \in \bar{S}} \deg(u) + |e(S, \bar{S})| + c|S| \leq 2\,|E|$?
- a cut of value $2\,|E|$ always exists, for $S = \emptyset$

Goldberg's algorithm for densest subgraph

- transform to a min-cut instance

- is there S s.t. $\sum_{u \in \bar{S}} \deg(u) + |e(S, \bar{S})| + c|S| \leq 2\,|E|$?
- $S \neq \emptyset$ gives cut of value $\sum_{u \in \bar{S}} \deg(u) + |e(S, \bar{S})| + c|S|$

Goldberg's algorithm for densest subgraph

- transform to a min-cut instance

- is there S s.t. $\sum_{u \in \bar{S}} \deg(u) + |e(S, \bar{S})| + c|S| \leq 2\,|E|$?
- YES, if min cut achieved for $S \neq \emptyset$

Goldberg's algorithm for densest subgraph

[Goldberg, 1984]

input: undirected graph $G = (V, E)$, number c
output: S, if $d(S) \geq c$

1 transform G into min-cut instance $G' = (V \cup \{s\} \cup \{t\}, E', w')$
2 find min cut $\{s\} \cup S$ on G'
3 if $S \neq \emptyset$ return S
4 else return NO

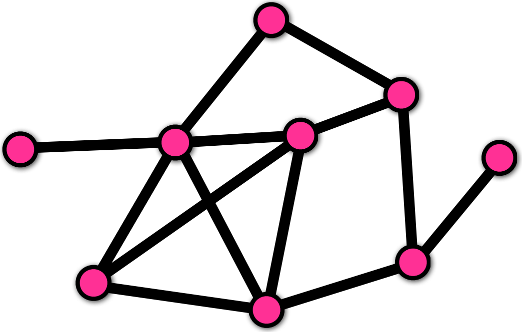
- to find the densest subgraph perform binary search on c
- logarithmic number of min-cut calls

densest subgraph problem – discussion

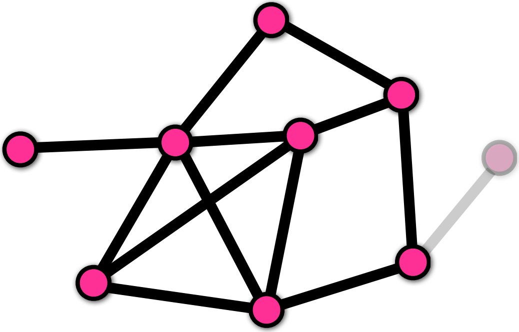
- Goldberg's algorithm polynomial algorithm, but
- $\mathcal{O}(nm)$ time for one min-cut computation
- not scalable for large graphs (millions of vertices / edges)

- faster algorithm due to [Charikar, 2000]
- greedy and simple to implement
- approximation algorithm

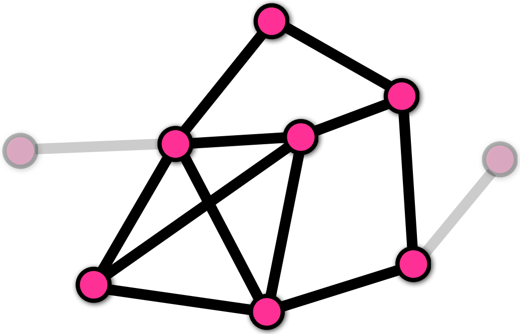
greedy algorithm for densest subgraph — example



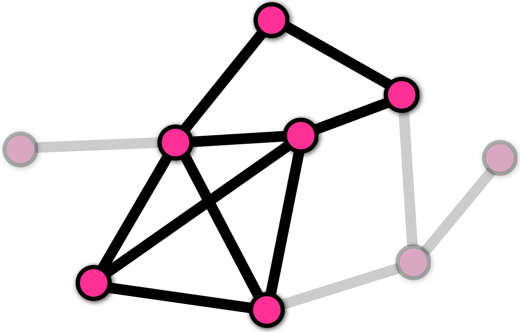
greedy algorithm for densest subgraph — example



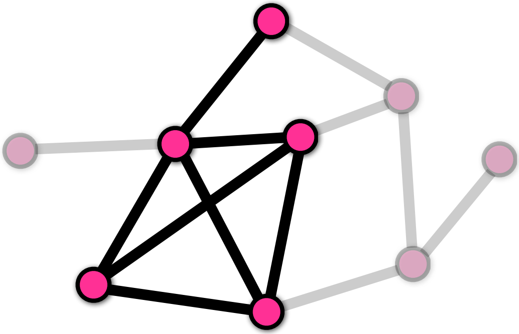
greedy algorithm for densest subgraph — example



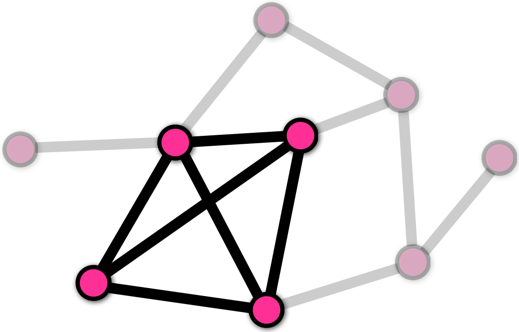
greedy algorithm for densest subgraph — example



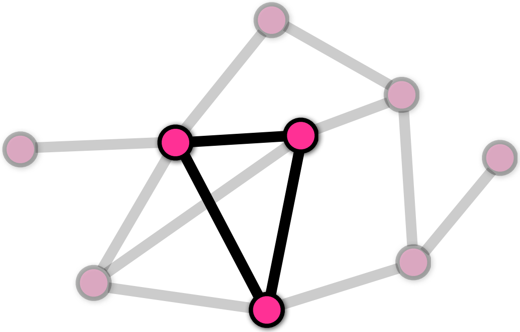
greedy algorithm for densest subgraph — example



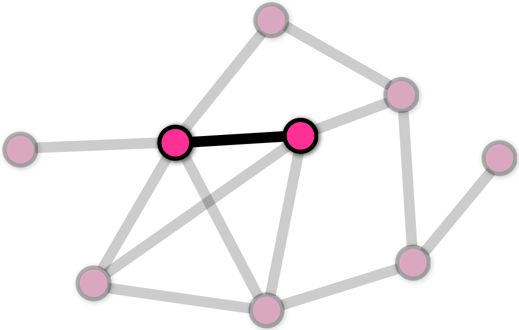
greedy algorithm for densest subgraph — example



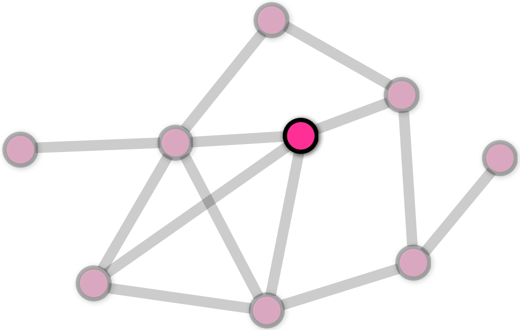
greedy algorithm for densest subgraph — example



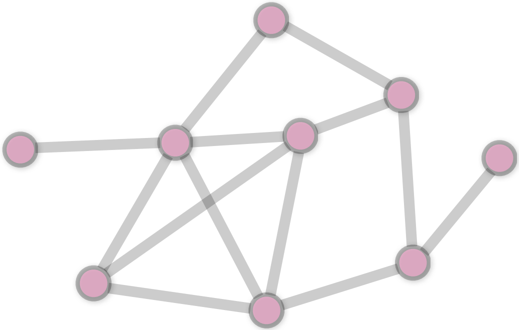
greedy algorithm for densest subgraph — example



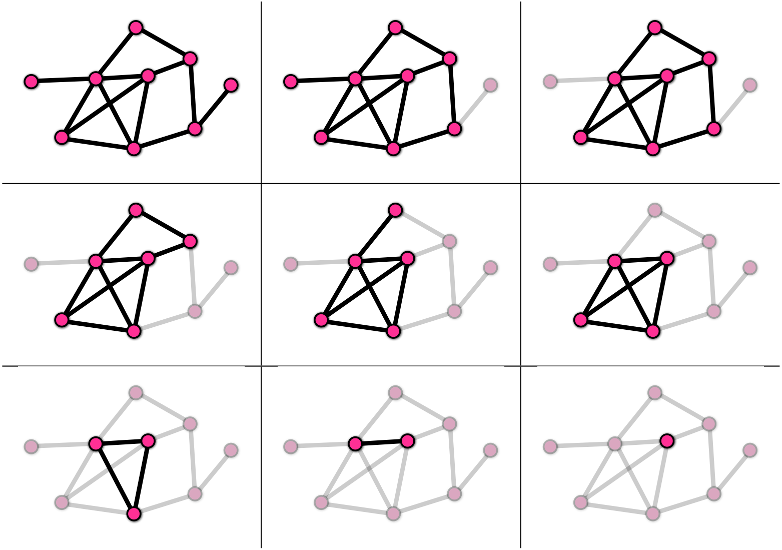
greedy algorithm for densest subgraph — example



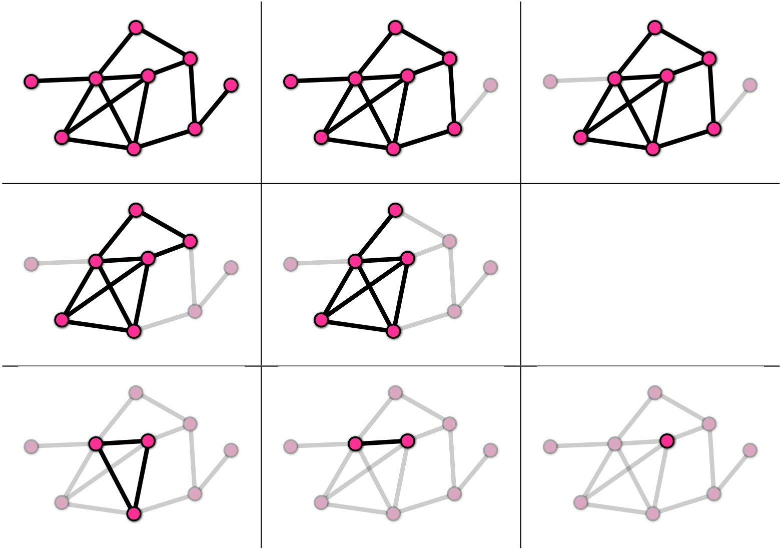
greedy algorithm for densest subgraph — example



greedy algorithm for densest subgraph — example



greedy algorithm for densest subgraph — example



greedy algorithm for densest subgraph

[Charikar, 2000]

input: undirected graph $G = (V, E)$
output: S, a dense subgraph of G

1 set $G_n \leftarrow G$
2 for $k \leftarrow n$ downto 1
2.1 let v be the smallest degree vertex in G_k
2.2 $G_{k-1} \leftarrow G_k \setminus \{v\}$
3 output the densest subgraph among $G_n, G_{n-1}, \ldots, G_1$

proof of 2-approximation guarantee

a neat argument due to [Khuller and Saha, 2009]

- let S^* be the vertices of the optimal subgraph
- let $d(S^*) = \lambda$ be the maximum degree density
- notice that for all $v \in S^*$ we have $\deg_{S^*}(v) \geq \lambda$
- (why?) by optimality of S^*

$$\frac{|e(S^*)|}{|S^*|} \geq \frac{|e(S^*)| - \deg_{S^*}(v)}{|S^*| - 1}$$

and thus

$$\deg_{S^*}(v) \geq \frac{|e(S^*)|}{|S^*|} = d(S^*) = \lambda$$

proof of 2-approximation guarantee (continued)

([Khuller and Saha, 2009])

- consider greedy when the first vertex $v \in S^* \subseteq V$ is removed
- let S be the set of vertices, just before removing v
- total number of edges before removing v is $\geq \lambda|S|/2$
- therefore, greedy returns a solution with degree density at least $\frac{\lambda}{2}$

QED

the greedy algorithm

- factor-2 approximation algorithm
- runs in linear time $\mathcal{O}(n+m)$
- for a polynomial problem ...
  but faster and easier to implement than the exact algorithm
- everything goes through for weighted graphs
  using heaps: $\mathcal{O}(m + n \log n)$
- things are not as straightforward for directed graphs

LP formulation

- Charikar gave an LP formulation as well

$$
\begin{aligned}
\text{maximize} \quad & \sum_{(i,j)\in E(G)} x_{ij} \\
\text{such that} \quad & x_{ij} \le y_i, \quad \text{for all } (i,j) \in E(G) \\
& x_{ij} \le y_j, \quad \text{for all } (i,j) \in E(G) \\
& \sum_i y_i \le 1 \\
& x_{ij}, y_i \ge 0
\end{aligned}
$$

Exercise: Prove that the optimal LP solution achieves a value at least as large as the optimal density of G.

Thank you all for the wonderful semester!

- https://tsourakakis.com/cs630-graduate-algorithms-fall23/

references I

Beutel, A., Xu, W., Guruswami, V., Palow, C., and Faloutsos, C. (2013).
Copycatch: stopping group attacks by spotting lockstep behavior in social networks.
In *Proceedings of the 22nd international conference on World Wide Web*, pages 119–130.

Charikar, M. (2000).
Greedy approximation algorithms for finding dense components in a graph.
In *APPROX*.

Fratkin, E., Naughton, B. T., Brutlag, D. L., and Batzoglou, S. (2006).
Motifcut: regulatory motifs finding with maximum density subgraphs.
Bioinformatics, 22(14):e150–e157.

references II

Goldberg, A. V. (1984).
Finding a maximum density subgraph.
Technical report, University of California at Berkeley.

Khuller, S. and Saha, B. (2009).
On finding dense subgraphs.
In *ICALP*.

Saha, B., Hoch, A., Khuller, S., Raschid, L., and Zhang, X.-N. (2010).
Dense subgraphs with restrictions and applications to gene annotation graphs.
In *Research in Computational Molecular Biology*, pages 456–472. Springer.