CS3000: Algorithms & Data
Jonathan Ullman

Lecture 17:
More Applications of Network Flow

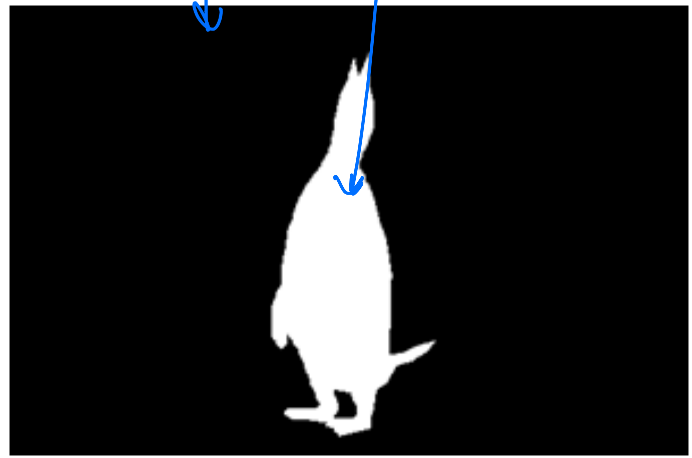
March 25, 2020

Image Segmentation

background foreground

m

n



Set of pixels $[n] \times [m]$

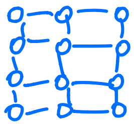
- Separate image into foreground and background
- We have some idea of:
  - whether pixel i is in the foreground or background
  - whether pair (i,j) are likely to go together

e.g. middle pixels more likely to be foreground

Image Segmentation

Assume all values in the graph were given externally

- Input:



  - an undirected graph $G = (V, E)$; V = "pixels", E = "pairs"
  - likelihoods $a_i, b_i \geq 0$ for every $i \in V$

a ≈ likelihood of foreground

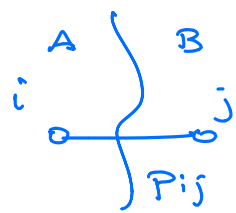
  - separation penalty $p_{ij} \geq 0$ for every $(i, j) \in E$

- Output:
  - a partition of V into (A, B) that maximizes

$$q(A, B) = \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{(i,j) \in E \\ \text{btw } A \text{ and } B}} p_{ij}$$

foreground

background



Reduction to MinCut

Short for Image Segmentation

- Differences between $\boxed{\text{SEG}}$ and MINCUT:
  - SEG asks us to maximize, MINCUT asks us to minimize

$$\max_{x} f(x) \qquad\qquad \min_{x} -f(x)$$

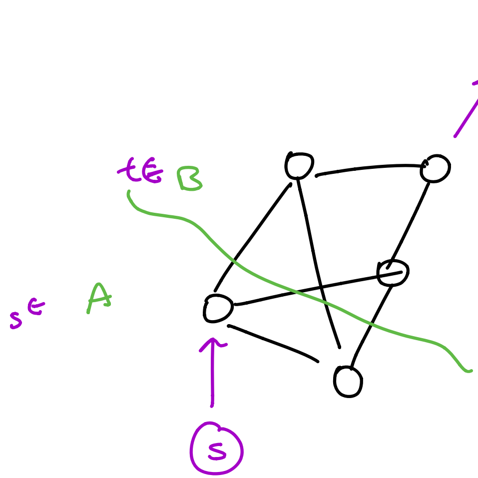
$$\max_{A,B} \sum_{i\in A} a_i + \sum_{j\in B} b_j - \sum_{\substack{(i,j)\in E \\ \text{btw } A \text{ and } B}} p_{ij} \quad \Longleftrightarrow \quad \boxed{\min_{A,B} \sum_{i\in A} b_i + \sum_{j\in B} a_j + \sum_{\substack{(i,j)\in E \\ \text{btw } A \text{ and } B}} p_{ij}}$$

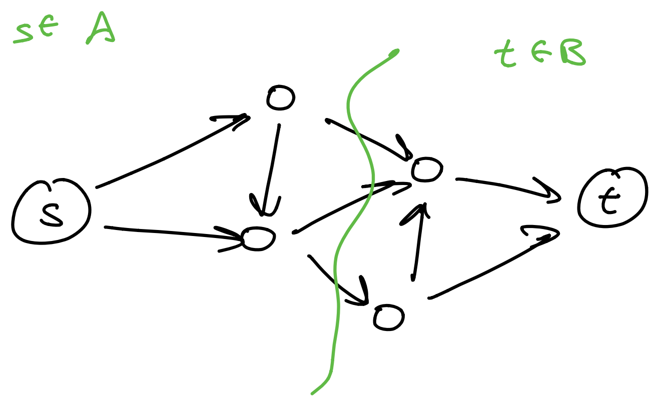
$$\min_{A,B} -\sum_{i\in A} a_i - \sum_{j\in B} b_j + \sum_{\substack{(i,j)\in E \\ \text{btw } A,B}} p_{ij}$$

$$\min_{A,B} \underline{\boxed{\sum_{i\in V} a_i + b_i}} - \sum_{i\in A} a_i - \sum_{i\in B} b_i + \sum p_{ij}$$

Reduction to MinCut

- Differences between SEG and MINCUT:
  - SEG allows any partition, MINCUT requires $s \in A, t \in B$





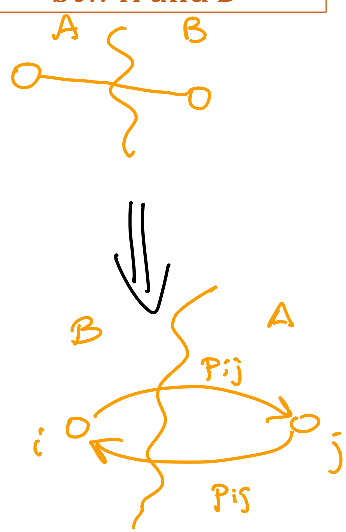
Solution: Add "dummy nodes"
s and t to the graph

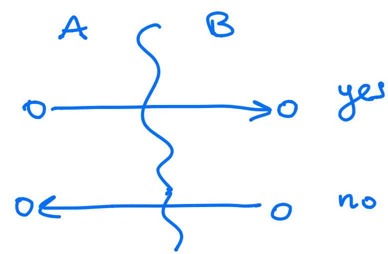
Reduction to MinCut

- Differences between SEG and MINCUT:
  - SEG has edges **between A and B**, MINCUT considers edges **from A to B**

$$\min_{A,B} \sum_{i \in A} b_i + \sum_{j \in B} a_j + \sum_{\substack{(i,j)\in E \\ \text{btw } A \text{ and } B}} p_{ij}$$

$$\min_{A,B} \sum_{\substack{(i,j)\in E \\ \text{from } A \text{ to } B}} p_{ij}$$





Solution:

Replace undirected edge (i,j) w/ $i \to j$ and $j \to i$ both with capacity p_{ij}

capacity p_{ij} in both directions

Reduction to MinCut

- Differences between SEG and MINCUT:
  - SEG has terms for each node in A,B, MINCUT only has terms for edges from A to B

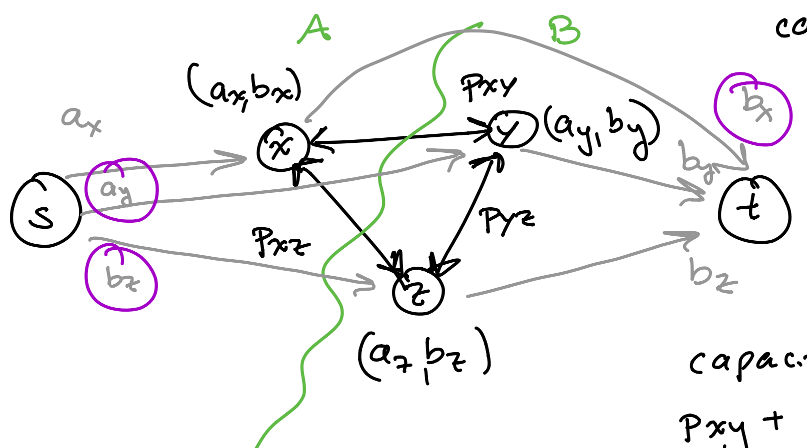
① ② ③ ③

$$\min_{A,B} \boxed{\sum_{i \in A} b_i} + \boxed{\sum_{j \in B} a_j} + \sum_{\substack{(i,j) \in E \\ \text{btw } A \text{ and } B}} p_{ij}$$

$$\min_{A,B} \sum_{\substack{(i,j) \in E \\ \text{from } A \text{ to } B}} p_{ij}$$

capacity: $p_{xy} + p_{xz}$

Solution:
Use "dummy" edges from s and t



capacity we want:
$p_{xy} + p_{xz} + b_x + a_y + a_z$

Reduction to MinCut

- How should the reduction work?
  - capacity of the cut should correspond to the function we're trying to minimize

$$\min_{A,B} \sum_{i \in A} b_i + \sum_{j \in B} a_j + \sum_{\substack{(i,j) \in E \\ \text{from } A \text{ to } B}} p_{ij}$$

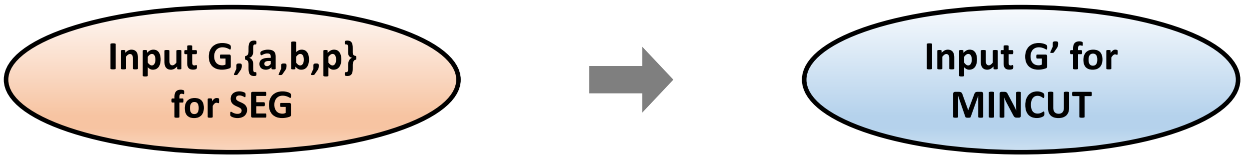
① Replace max with min

② Replace undirected edges w/ pairs of directed edges

③ Add dummy nodes s, t

④ Add dummy edges $s \xrightarrow{a_x} x$ $\quad x \xrightarrow{b_x} t$

Step 1: Transform the Input



① Replace max with min

$O(m)$ → ② Replace undirected edges w/ pairs of directed edges

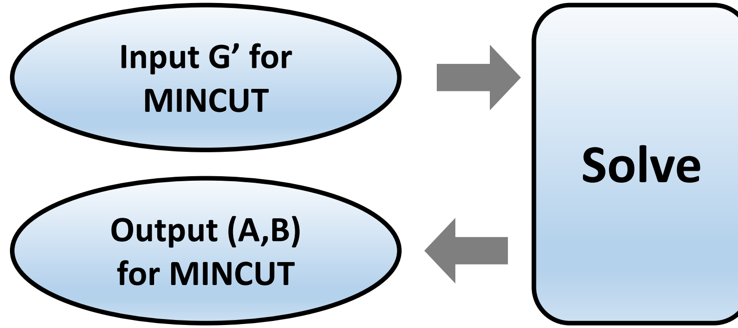
$O(1)$ → ③ Add dummy nodes s, t

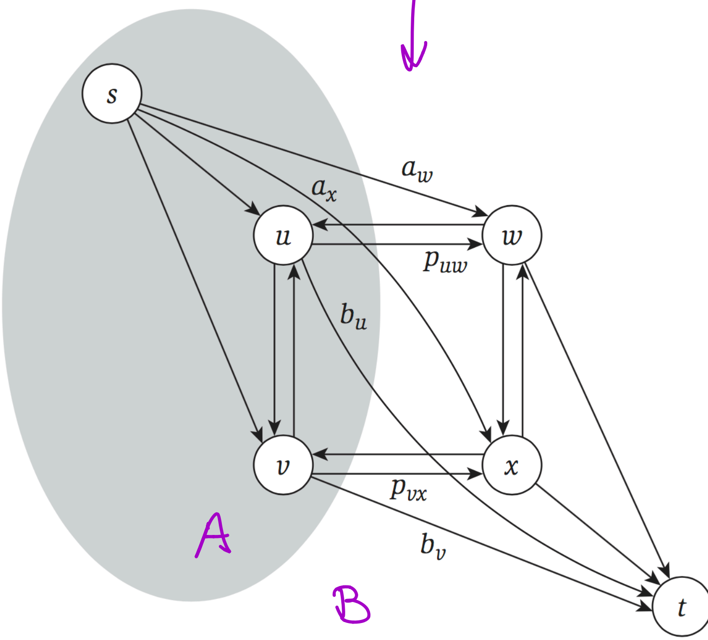
$O(n)$ → ④ Add dummy edges $s \xrightarrow{a_x} x$ $\quad$ $x \xrightarrow{b_x} t$

Total Time : $O(m+n)$

Step 2: Receive the Output

{u, w, v, x} were the original graph





(A, B) is a minimum s,t cut in G'

Running Time:

Solve mincut on a graph with n+2 nodes and 2m+2n edges

$\rightarrow$ O(mn) time

Step 3: Transform the Output

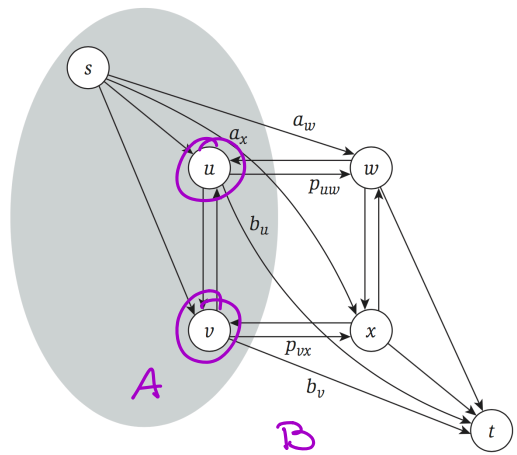
Output (A,B) for SEG

Output (A,B) for MINCUT

Return partition

$A = \{u, v\}$

$B = \{w, x\}$



Time: $O(n)$

Reduction to MinCut

- correctness?
  - Every partition (A,B) of the original nodes corresponds to an s,t cut $(A \cup \{s\}, B \cup \{t\})$
  - For every s,t cut $(A \cup \{s\}, B \cup \{t\})$ the capacity is $\boxed{\sum_{i \in A} b_i + \sum_{i \in B} a_i + \sum_{\substack{(i,j) \in E \\ i \in A\ j \in B}} p_{ij}}$

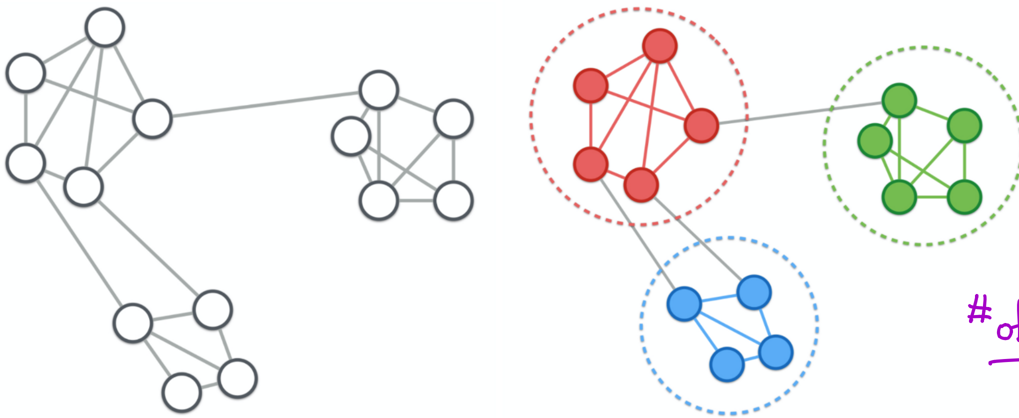
Exactly what SEG wants to minimize

- running time?

Total Time: $O(mn)$

Bottleneck is solving minimum cut

~~Image Segmentation~~ Densest Subgraph



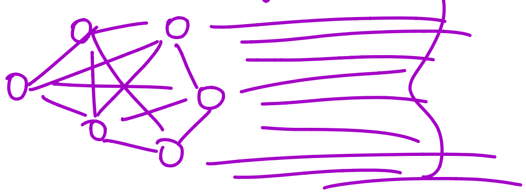
$$\frac{\#\text{ of edges inside}}{\binom{|S|}{2}}$$

1

$$\frac{\#\text{ of edges in}}{|S|}$$

- Want to identify communities in a network
  - "Community": a set of nodes that have a lot of connections inside and few outside

15 edges inside



100,000,000 nodes

1000 edges outside

15/6 = 2.5   1/2 = .5

Densest Subgraph

$$\frac{2 \cdot \#\text{of edges inside } A}{\#\text{of nodes in } A}$$

- Input:
  - an undirected graph $G = (V, E)$
- Output:
  - a subset of nodes $A \subseteq V$ that maximizes $\frac{2|E(A,A)|}{|A|}$

- $E(A,A)$ = set of edges w/ both endpoints in A
- $E(A,B)$ = set of edges w/ one endpoint in A, one in B

(1) DS uses an undirected graph

(2) DS lets us choose any set A

MINCUT uses a directed g

MINCUT lets choose s,t cut

(1) 

(2) Add "dummy" nodes s, t

Same transformations as SEG

Need to transform the objective function:

DS

$$\frac{2 \cdot |E(A,A)|}{|A|}$$

?

MINCUT

$$\sum_{\substack{(i,j) \in E \\ i \in A \;\; j \in B}} c_{i,j}$$

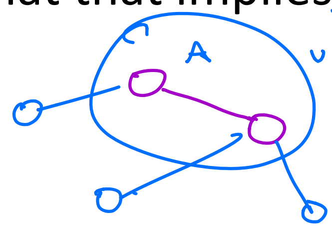
$$\Downarrow$$

$$\underbrace{\sum_{i \in A} a_i + \sum_{i \in B} b_i}_{\text{using "dummy" edges}} + \sum_{\substack{(i,j) \in E \\ \text{btw } A,B}} c_{i,j}$$

Reduction to MinCut

Clm: If I can ask yes/no questions "Is the DS dense than δ?" then I can find the densest subgraph.

- Different objectives
  - maximize $\frac{2|E(A,A)|}{|A|}$ vs. minimize $|E(A,B)|$
- Suppose $\boxed{\frac{2|E(A,A)|}{|A|}} \geq \delta$ and see what that implies



$\sum_{v \in A} \deg(v) = 2|E(A,A)| + |E(A,B)|$

$$\Leftrightarrow 2|E(A,A)| \geq \delta|A|$$

$$\Leftrightarrow \Sigma_{v\in A} \deg(v) - |E(A,B)| \geq \delta|A|$$

$$\Leftrightarrow \Sigma_{v\in V} \deg(v) - \Sigma_{v\in B} \deg(v) - |E(A,B)| \geq \delta|A|$$

$$\Leftrightarrow 2|E| - \Sigma_{v\in B} \deg(v) - |E(A,B)| \geq \delta|A|$$

$$\Leftrightarrow \Sigma_{v\in B} \deg(v) + \delta|A| + |E(A,B)| \leq 2|E|$$

$$\Leftrightarrow \Sigma_{v\in B} \deg(v) + \Sigma_{v\in A}\delta + \Sigma_{e \text{ from } A \text{ to } B}\, 1 \leq 2|E|$$

Reduction to MinCut

Clm: If I can ask yes/no questions "Is the DS denser than δ?" then I can find the densest subgraph.

- Different objectives
  - maximize $\frac{2|E(A,A)|}{|A|}$ vs. minimize $|E(A,B)|$

- Suppose $\boxed{\frac{2|E(A,A)|}{|A|}} \geq \hat{\delta}$ and see what that implies

$$\Leftrightarrow 2|E(A,A)| \geq \delta|A|$$

$$\Leftrightarrow \Sigma_{v\in A} \deg(v) - |E(A,B)| \geq \delta|A|$$

$$\Leftrightarrow \Sigma_{v\in V} \deg(v) - \Sigma_{v\in B} \deg(v) - |E(A,B)| \geq \delta|A|$$

$$\Leftrightarrow 2|E| - \Sigma_{v\in B} \deg(v) - |E(A,B)| \geq \delta|A|$$

$$\Leftrightarrow \Sigma_{v\in B} \deg(v) + \delta|A| + |E(A,B)| \leq 2|E|$$

$$\Leftrightarrow \Sigma_{v\in B} \deg(v) + \Sigma_{v\in A}\delta + \Sigma_{e \text{ from } A \text{ to } B} 1 \leq 2|E|$$

$$\sum_{v\in A} \deg(v)$$
$$= \sum_{v\in A} \deg(v) + \sum_{v\in B} \deg(v) - \sum_{v\in B} \deg(v)$$
$$= \sum_{v\in V} \deg(v) - \sum_{v\in B} \deg(v)$$

$$|E(A,B)| = \sum_{\substack{e \text{ from} \\ A\to B}} 1$$

$$\delta|A| = \sum_{v\in A} \delta$$

$$\sum_{v \in B} \deg(v) + \sum_{v \in A} \delta + \sum_{\substack{e \\ \text{from } A \text{ to } B}} 1$$

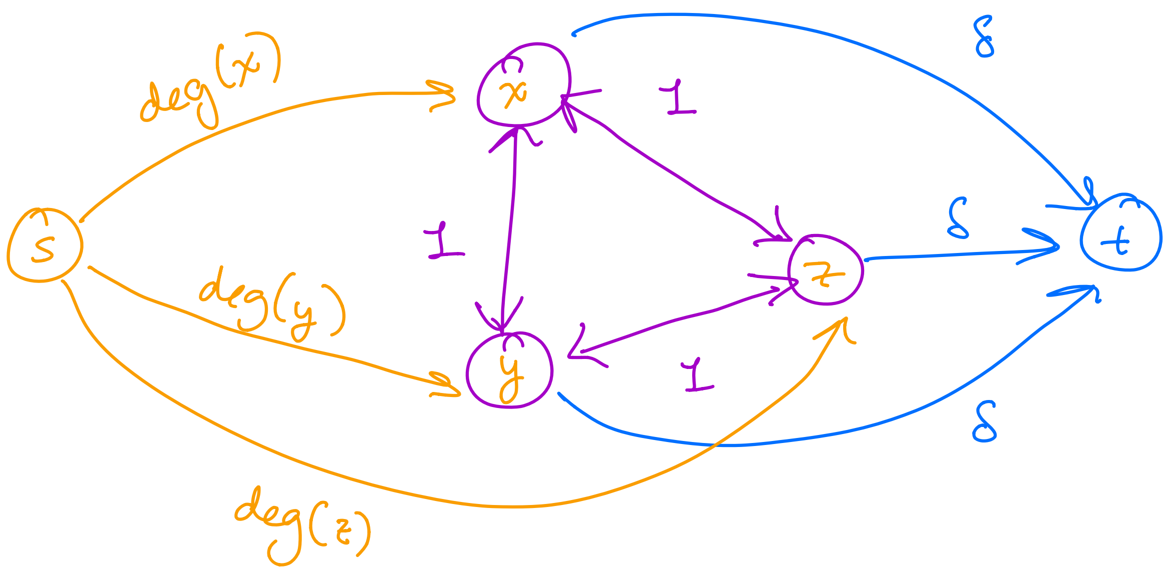
If the value is $\leq 2|E|$

then the subgraph A has

$$\frac{2 \cdot |E(A,A)|}{|A|} \geq \delta$$

Reduction to MinCut

$$\boxed{\Sigma_{v\in B} \deg(v) + \Sigma_{v\in A}\, \delta + \Sigma_{e \text{ from } A \text{ to } B}\, 1 \leq 2|E|}$$



This graph has min-cut $\leq 2|E|$ if and only if $\exists$ a subgraph of density $\geq \delta$