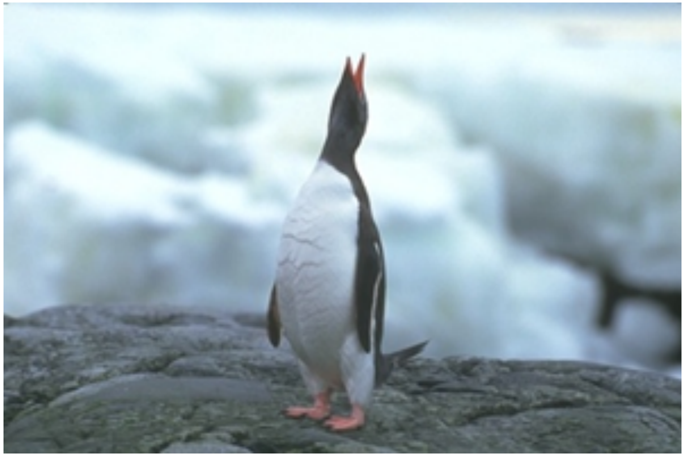
# CS4800: Algorithms & Data
# Jonathan Ullman

Lecture 20:
More Applications of Network Flow

Apr 3, 2018

# Image Segmentation

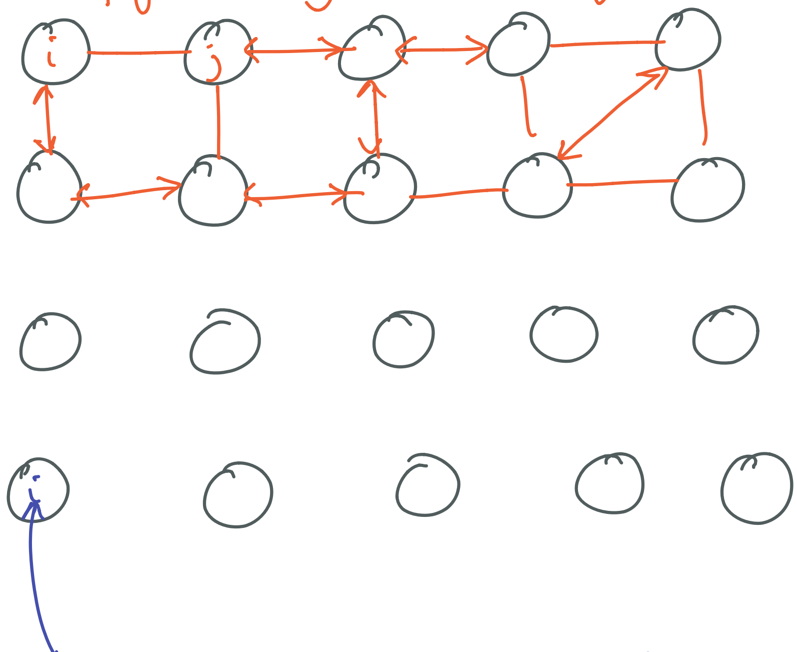
# Image Segmentation



- 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

# Image Segmentation

$p_{ij}$ : penalty for separating these two




$a_i$ : how likely to be in the foreground

$b_i$ : how likely to be in the background

# Image Segmentation

- Input:
  - a directed graph $G = (V, E)$; $V$ = "pixels", $E$ = "pairs"
  - likelihoods $a_i, b_i \geq 0$ for every $i \in V$
  - separation penalty $p_{ij} \geq 0$ for every $(i, j) \in E$
- Output:
  - a partition of $V$ into $(A, B)$ that maximizes

foreground background

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

w/o $p_{ij}$, the problem is trivial

# Reduction to MinCut

- Differences between IS and MINCUT:
  - IS asks us to maximize, MINCUT asks us to minimize

IS (maximization)

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

$\Longleftrightarrow$

IS' (minimization)

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

  - IS allows any partition, MINCUT requires $s \in A$, $t \in B$

Add nodes s, t. Now any way of partitioning the nodes in V is OK.

$$\operatorname*{argmax}_{A,B} \sum_{i\in A} a_i + \sum_{j\in B} b_j - \sum_{\substack{(i,j)\in E \\ i\in A,\, j\in B}} p_{ij}$$

$$= \operatorname*{argmin}_{A,B} -\sum_{i\in A} a_i - \sum_{j\in B} b_j + \sum_{\substack{(i,j)\in E \\ \dots}} p_{ij}$$

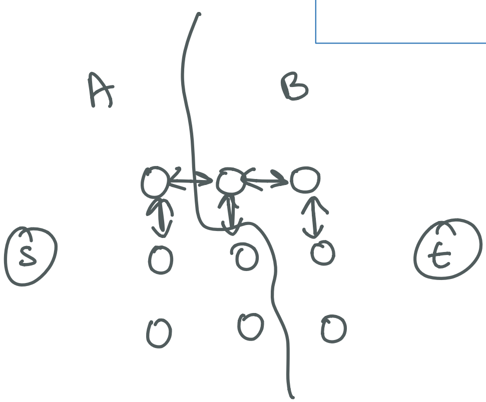
$$= \operatorname*{argmin}_{A,B} \left(\sum_{i\in V} a_i + \sum_{j\in V} b_j\right) - \sum_{i\in A} a_i - \sum_{j\in B} b_j + \sum_{\substack{(i,j)\in E \\ \dots}} p_{ij}$$

$$= \operatorname*{argmin}_{A,B} \sum_{i\in B} a_i + \sum_{j\in A} b_j + \sum_{\substack{(i,j)\in E \\ i\in A,\, j\in B}} p_{ij}$$

# Reduction to MinCut

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

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




MINCUT will be some cut A,B that minimizes $\text{cap}(A,B) = \sum_{\substack{(i,j) \in E \\ i \in A,\, j \in B}} c(i,j)$

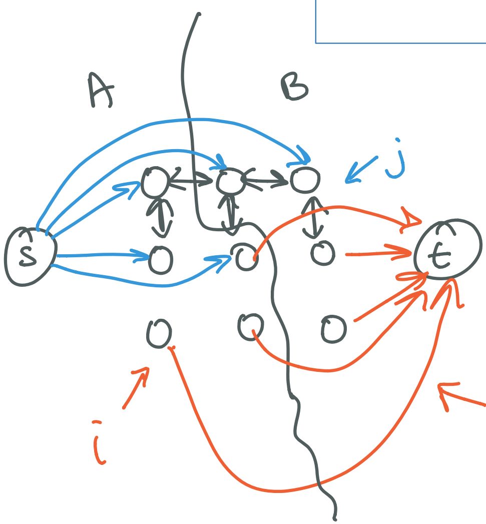
# Reduction to MinCut

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

$c(s,j) = a_i$

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

$$\min_{A,B} \sum_{\substack{(i,j) \in E \\ i \in A, j \in B}} c(i,j)$$




$c(i,t) = b_i$

① $\forall\ (i,j) \in E$, $c(i,j) = p(i,j)$

② Add edges $(s,i)$ w/ $c(s,i) = a_i$

③ Add edges $(i,t)$ w/ $c(i,t) = b_i$

# Step 1: Transform the Input

**Input G,{a,b,p} for IS** → **Input G' for MINCUT**

$G=(V,E)$ $\{a_i\}\{b_i\}\{p_{ij}\}$

$G'=(V',E')$ $\{c_{ij}\}$

$$V' = V + \{s,t\}$$

$$E' = E + \{(s,i)\} + \{(i,t)\}$$

$$c_{ij} = p_{ij} \quad \forall e \in E$$

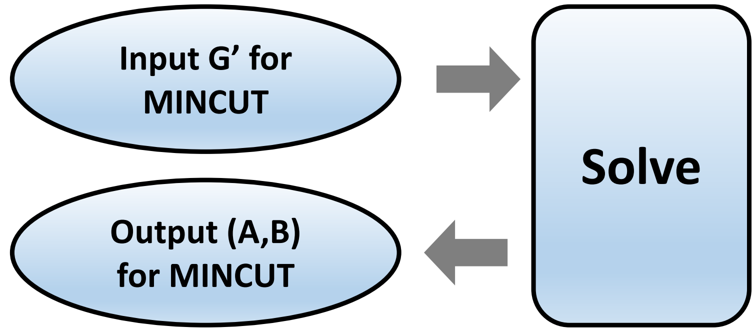
$$c_{si} = a_i, \quad c_{it} = b_i \quad \forall i \in V$$

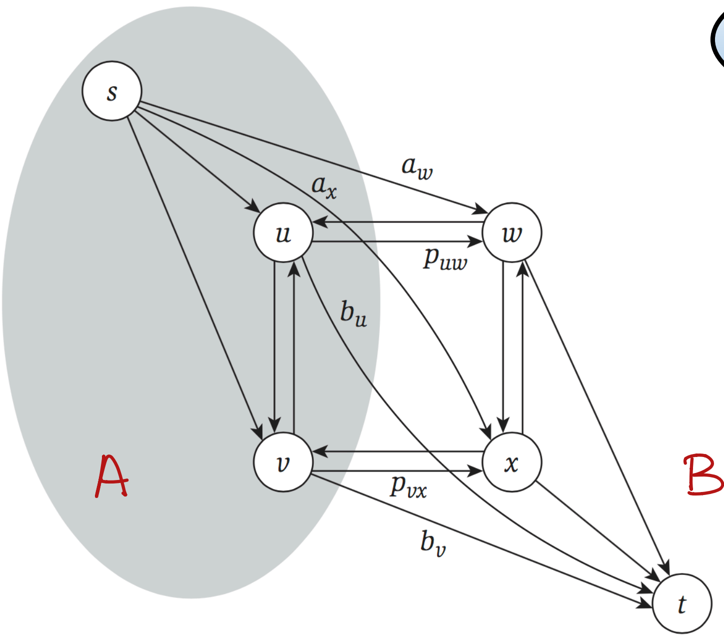
$O(m)$ time

$|V'| = |V| + 2$

$|E'| = |E| + 2n$

# Step 2: Receive the Output



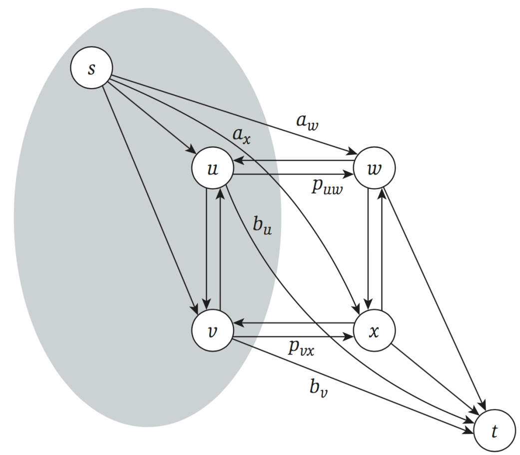
# Step 3: Transform the Output

Output (A,B) for IS ← Output (A',B') for MINCUT

$A = A' - \{s\}$

$B = B' - \{t\}$

$O(1)$ time



# Reduction to MinCut

- correctness?

The minimum cut in $G'$ is $A', B'$ minimizing

$$\sum_{i \in A} b_i + \sum_{i \in B} a_i + \sum_{\substack{(i,j) \in E \\ i \in A', j \in B'}} p_{ij}$$

- running time?

Can solve mincut in time $O(|V'||E'|)$

$$= O((n+2)(m+2n))$$

$$= O(nm)$$

The reduction is $O(m)$ $\Rightarrow$ total time is $O(nm)$

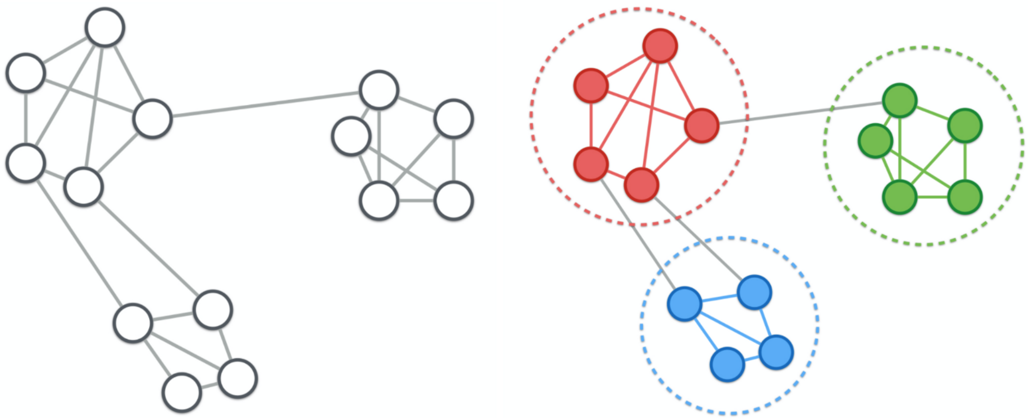
# Densest Subgraph

# Community Detection

# ~~Image Segmentation~~

$E(S,S) = 8$

$\frac{2 \cdot 8}{5} = 3.2$



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

# Densest Subgraph

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

$E(A,B) = \{ (i,j) \in E : i \in A, j \in B \}$

$E(S,S) =$ set of edges w/ both endpoints in $S$

$E(S,S^c) =$ set of edges crossing the cut $(S,S^c)$

# Reduction to MinCut

- assume $c(e)=1$
- assume there are "dummy nodes" $s,t$

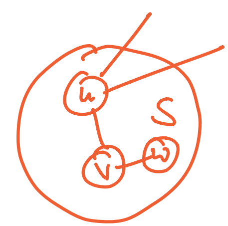
- Different objectives
  - find $(S, S^c)$ to maximize $\frac{2|E(S,S)|}{|S|}$ (DS)
  - find $(S, S^c\ )$ to minimize $|E(S, S^c)|$ (MINCUT)

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

Is there a set of density $\geq d$?

  - $2|E(S,S)| \geq d|S|$
  - $\Sigma_{v \in S} \deg(v)\ - |E(S, S^c)| \geq d|S|$
  - $\Sigma_{v \in V} \deg(v)\ - \Sigma_{v \in S^c} \deg(v) - |E(S, S^c)| \geq d|S|$
  - $2|E|\ - \Sigma_{v \in S^c} \deg(v) - |E(S, S^c)| \geq d|S|$

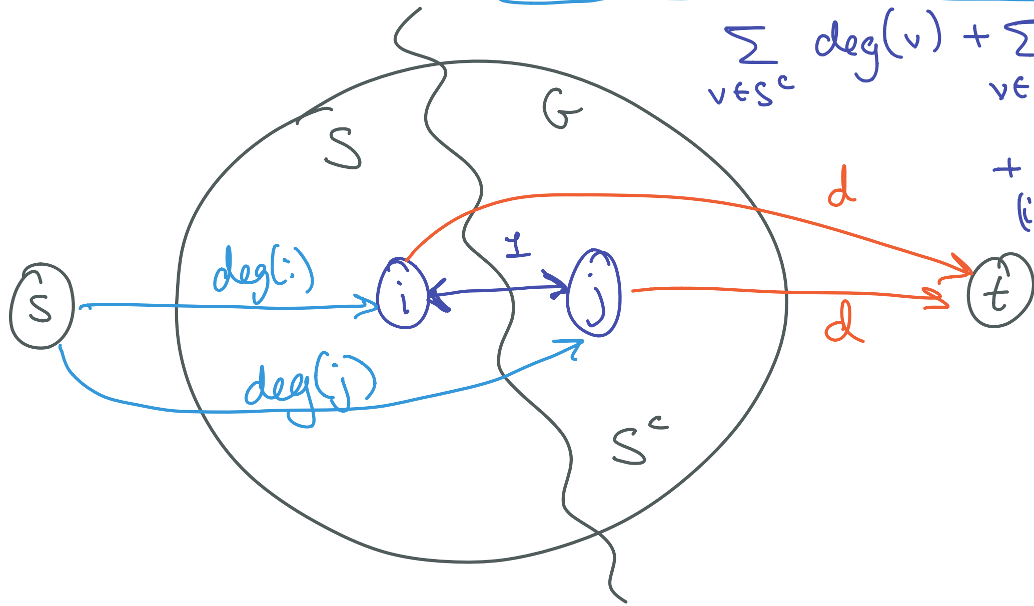
  - $\Sigma_{v \in S^c} \deg(v) + d|S| + \ |E(S, S^c)| \leq 2|E|$



# Reduction to MinCut

$\boxed{\Sigma_{v\in S^c} \deg(v) + d|S| + |E(S,S^c)|} \le 2|E|$

$\sum_{v\in S^c} \deg(v) + \sum_{v\in S} d + \sum_{(i,j)\in E(S,S^c)} 1$




mincut finds $S, S^c$ to minimize $\sum_{(i,j)\in E(S,S^c)} c(i,j)$

① $\forall (i,j) \in E$ add a bidirectional edge w/ $c(i,j) = 1$

② $\forall i$ add $(s,i)$ w/ capacity $c(s,i) = \deg(i)$

③ $\forall i$ add $(i,t)$ w/ capacity $c(i,t) = d$