CS 577- Intro to Algorithms

Network Flow (Part 4)

Dieter van Melkebeek

November 5, 2020

Outline

Outline

Applications of min cut

- Image segmentation
- Project selection

Recap

Recap

Network

- a digraph (V, E)
- edge capacities $c : E \to [0, \infty)$
- the source $s \in V$, which has indegree 0, and
- the sink $t \in V$, which has outdegree 0.

Recap

Network

- a digraph (V, E)
- edge capacities $c : E \to [0, \infty)$
- the source $s \in V$, which has indegree 0, and
- the sink $t \in V$, which has outdegree 0.

st-Cut

A partition (S, T) of V such that $s \in S$ and $t \in T$.

Recap

Network

- a digraph (V, E)
- edge capacities $c : E \to [0, \infty)$
- the source $s \in V$, which has indegree 0, and
- the sink $t \in V$, which has outdegree 0.

st-Cut

A partition (S, T) of V such that $s \in S$ and $t \in T$.

Min cut problem

Input: network $N = (V, E, c, s, t)$

Output: *st*-cut (S, T) such that $c(S, T) \doteq \sum_{e \in S \times T} c(e)$ is minimized

Recap

Network

- a digraph (V, E)
- edge capacities $c : E \to [0, \infty)$
- the source $s \in V$, which has indegree 0, and
- the sink $t \in V$, which has outdegree 0.

st-Cut

A partition (S, T) of V such that $s \in S$ and $t \in T$.

Min cut problem

Input:	network $N = (V, E, c, s, t)$
Output:	*st*-cut (S, T) such that $c(S, T) \doteq \sum_{e \in S \times T} c(e)$ is minimized
Complexity:	time $O(nm)$

Image segmentation

Image segmentation

Computational problem

Input: grid of pixels $i \in [n]$

Image segmentation

Computational problem

Input: grid of pixels $i \in [n]$

$f_i \in [0, \infty)$: "likelihood" that i is foreground

Image segmentation

Computational problem

Input: grid of pixels $i \in [n]$
$f_i \in [0, \infty)$: "likelihood" that i is foreground
$b_i \in [0, \infty)$: "likelihood" that i is background

Image segmentation

Computational problem

Input: grid of pixels $i \in [n]$
$f_i \in [0, \infty)$: "likelihood" that i is foreground
$b_i \in [0, \infty)$: "likelihood" that i is background

Output: partition of $[n]$ into foreground F and background B

Image segmentation

Computational problem

Input: grid of pixels $i \in [n]$
$f_i \in [0, \infty)$: "likelihood" that i is foreground
$b_i \in [0, \infty)$: "likelihood" that i is background
$c \in [0, \infty)$: penalty for separating neighboring pixels

Output: partition of $[n]$ into foreground F and background B

Image segmentation

Computational problem

Input: grid of pixels $i \in [n]$
$f_i \in [0, \infty)$: "likelihood" that i is foreground
$b_i \in [0, \infty)$: "likelihood" that i is background
$c \in [0, \infty)$: penalty for separating neighboring pixels

Output: partition of $[n]$ into foreground F and background B maximizing

$$\sum_{i \in F} f_i + \sum_{j \in B} b_j - c \cdot |\{(i, j) \in F \times B : i \sim j\}|$$

Image segmentation

Computational problem

Input: grid of pixels $i \in [n]$
$f_i \in [0, \infty)$: "likelihood" that i is foreground
$b_i \in [0, \infty)$: "likelihood" that i is background
$c \in [0, \infty)$: penalty for separating neighboring pixels

Output: partition of $[n]$ into foreground F and background B maximizing

$$\sum_{i \in F} f_i + \sum_{j \in B} b_j - c \cdot |\{(i,j) \in F \times B : i \sim j\}|$$

Model

Image segmentation

Computational problem

Input: grid of pixels $i \in [n]$
$f_i \in [0, \infty)$: "likelihood" that i is foreground
$b_i \in [0, \infty)$: "likelihood" that i is background
$c \in [0, \infty)$: penalty for separating neighboring pixels

Output: partition of $[n]$ into foreground F and background B maximizing

$$\sum_{i \in F} f_i + \sum_{j \in B} b_j - c \cdot |\{(i, j) \in F \times B : i \sim j\}|$$

Model

- Vertex for each pixel, source s, sink t

Image segmentation

Computational problem

Input: grid of pixels $i \in [n]$
$f_i \in [0, \infty)$: "likelihood" that i is foreground
$b_i \in [0, \infty)$: "likelihood" that i is background
$c \in [0, \infty)$: penalty for separating neighboring pixels

Output: partition of $[n]$ into foreground F and background B maximizing

$$\sum_{i \in F} f_i + \sum_{j \in B} b_j - c \cdot |\{(i,j) \in F \times B : i \sim j\}|$$

Model

- Vertex for each pixel, source s, sink t
- $S = F \cup \{s\}$ and $T = B \cup \{t\}$

Image Segmentation – rewriting objective

Image Segmentation – rewriting objective

$$\max_{F,B}\left(\sum_{i\in F} f_i + \sum_{j\in B} b_j - \sum_{\substack{i\sim j\\(i,j)\in F\times B}} c\right)$$

$$= -\min_{F,B}\left(\sum_{\substack{i\sim j\\(i,j)\in F\times B}} c - \sum_{i\in F} f_i - \sum_{j\in B} b_j\right)$$

$$= -\min_{F,B}\left(\sum_{\substack{i\sim j\\(i,j)\in F\times B}} c - \sum_{i\in[n]} f_i + \sum_{i\in B} f_i - \sum_{j\in[n]} b_j + \sum_{j\in F} b_j\right)$$

$$= \sum_{i\in[n]} f_i + \sum_{j\in[n]} b_j - \min_{F,B}\left(\sum_{\substack{i\sim j\\(i,j)\in F\times B}} c + \sum_{i\in B} f_i + \sum_{j\in F} b_j\right)$$

Image Segmentation – reduction to min cut

Image Segmentation – reduction to min cut

- Need to find partition of $[n]$ into F and B minimizing

$$(*) \doteq \sum_{\substack{i \sim j \\ (i,j) \in F \times B}} c + \sum_{i \in B} f_i + \sum_{j \in F} b_j .$$

Image Segmentation – reduction to min cut

- Need to find partition of $[n]$ into F and B minimizing

$$(*) \doteq \sum_{\substack{i \sim j \\ (i,j) \in F \times B}} c + \sum_{i \in B} f_i + \sum_{j \in F} b_j .$$

- Construct network N such that $(*) = c(S, T)$ where $S = F \cup \{s\}$ and $T = B \cup \{t\}$.

Image Segmentation – reduction to min cut

- Need to find partition of $[n]$ into F and B minimizing

$$(*) \doteq \sum_{\substack{i \sim j \\ (i,j) \in F \times B}} c + \sum_{i \in B} f_i + \sum_{j \in F} b_j.$$

- Construct network N such that $(*) = c(S, T)$ where $S = F \cup \{s\}$ and $T = B \cup \{t\}$.

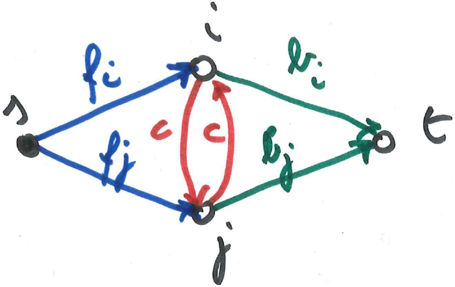
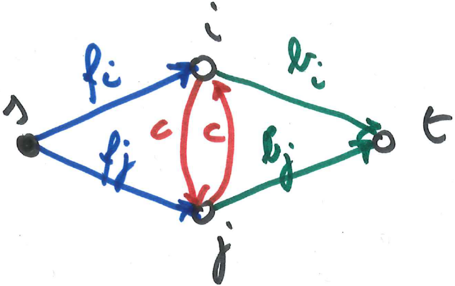


Image Segmentation – reduction to min cut

- Need to find partition of $[n]$ into F and B minimizing

$$(*) \doteq \sum_{\substack{i\sim j\\(i,j)\in F\times B}} c + \sum_{i\in B} f_i + \sum_{j\in F} b_j.$$

- Construct network N such that $(*) = c(S,T)$ where $S = F\cup\{s\}$ and $T = B\cup\{t\}$.



- Resulting algorithm runs in time $O(n^2)$.

Project Selection

Project Selection

Computational problem

Project Selection

Computational problem

Input: n projects $i \in [n]$

Project Selection

Computational problem

Input: n projects $i \in [n]$
m tools $j \in [m]$

Project Selection

Computational problem

Input: n projects $i \in [n]$
m tools $j \in [m]$
$T_i \subseteq [m]$: tools needed to realize project $i \in [n]$

Project Selection

Computational problem

Input: n projects $i \in [n]$
m tools $j \in [m]$
$T_i \subseteq [m]$: tools needed to realize project $i \in [n]$
$v_i \in [0, \infty)$: value of project $i \in [n]$ if realized

Project Selection

Computational problem

Input: n projects $i \in [n]$
m tools $j \in [m]$
$T_i \subseteq [m]$: tools needed to realize project $i \in [n]$
$v_i \in [0, \infty)$: value of project $i \in [n]$ if realized
$c_j \in [0, \infty)$: one-time cost of tool $j \in [m]$ if bought

Project Selection

Computational problem

Input: n projects $i \in [n]$
m tools $j \in [m]$
$T_i \subseteq [m]$: tools needed to realize project $i \in [n]$
$v_i \in [0, \infty)$: value of project $i \in [n]$ if realized
$c_j \in [0, \infty)$: one-time cost of tool $j \in [m]$ if bought

Output: Set of projects $I \subseteq [n]$ to realize and set of tools $J \subseteq [m]$ to buy maximizing $\sum_{i \in I} v_i - \sum_{j \in J} c_j$ such that $(\forall i \in I)\ T_i \subseteq J$.

Project Selection

Computational problem

Input: n projects $i \in [n]$
m tools $j \in [m]$
$T_i \subseteq [m]$: tools needed to realize project $i \in [n]$
$v_i \in [0, \infty)$: value of project $i \in [n]$ if realized
$c_j \in [0, \infty)$: one-time cost of tool $j \in [m]$ if bought

Output: Set of projects $I \subseteq [n]$ to realize and set of tools $J \subseteq [m]$ to buy maximizing $\sum_{i \in I} v_i - \sum_{j \in J} c_j$ such that $(\forall i \in I)\ T_i \subseteq J$.

Model

- Vertex for each project $i \in [n]$ & tool $j \in [m]$; source s, sink t

Project Selection

Computational problem

Input: n projects $i \in [n]$
m tools $j \in [m]$
$T_i \subseteq [m]$: tools needed to realize project $i \in [n]$
$v_i \in [0, \infty)$: value of project $i \in [n]$ if realized
$c_j \in [0, \infty)$: one-time cost of tool $j \in [m]$ if bought

Output: Set of projects $I \subseteq [n]$ to realize and set of tools $J \subseteq [m]$ to buy maximizing $\sum_{i \in I} v_i - \sum_{j \in J} c_j$ such that $(\forall i \in I)\ T_i \subseteq J$.

Model

- Vertex for each project $i \in [n]$ & tool $j \in [m]$; source s, sink t
- Project $i \in [n]$ is realized iff $i \in S$.

Project Selection

Computational problem

Input: n projects $i \in [n]$
m tools $j \in [m]$
$T_i \subseteq [m]$: tools needed to realize project $i \in [n]$
$v_i \in [0, \infty)$: value of project $i \in [n]$ if realized
$c_j \in [0, \infty)$: one-time cost of tool $j \in [m]$ if bought

Output: Set of projects $I \subseteq [n]$ to realize and set of tools $J \subseteq [m]$ to buy maximizing $\sum_{i \in I} v_i - \sum_{j \in J} c_j$ such that $(\forall i \in I)\ T_i \subseteq J$.

Model

- Vertex for each project $i \in [n]$ & tool $j \in [m]$; source s, sink t
- Project $i \in [n]$ is realized iff $i \in S$.
- Side of st-cut determines whether tool $j \in [m]$ is bought.

Project Selection – rewriting objective

Project Selection – rewriting objective

$$
\begin{aligned}
&\max_{I,J} \left(\sum_{i \in I} v_i - \sum_{j \in J} c_j \right) \\
&= -\min_{I,J} \left(\sum_{j \in J} c_j - \sum_{i \in I} v_i \right) \\
&= -\min_{I,J} \left(\sum_{j \in J} c_j - \sum_{i \in [n]} v_i + \sum_{i \in [n] \setminus I} v_i \right) \\
&= \sum_{i \in [n]} v_i - \min_{I,J} \left(\sum_{j \in J} c_j + \sum_{i \in [n] \setminus I} v_i \right)
\end{aligned}
$$

Project Selection – reduction to min cut

Project Selection – reduction to min cut

- Need to find $I \subseteq [n]$ and $J \subseteq [m]$ with $(\forall i \in I)\ T_i \subseteq J$ minimizing
$$(*) \doteq \sum_{j \in J} c_j + \sum_{i \in [n] \setminus I} v_i .$$

Project Selection – reduction to min cut

- Need to find $I \subseteq [n]$ and $J \subseteq [m]$ with $(\forall i \in I)\ T_i \subseteq J$ minimizing
$$(*) \doteq \sum_{j \in J} c_j + \sum_{i \in [n] \setminus I} v_i .$$
- Construct network N such that $(*) = c(S, T)$ where I are the projects in S.

Project Selection – reduction to min cut

- Need to find $I \subseteq [n]$ and $J \subseteq [m]$ with $(\forall i \in I)\ T_i \subseteq J$ minimizing

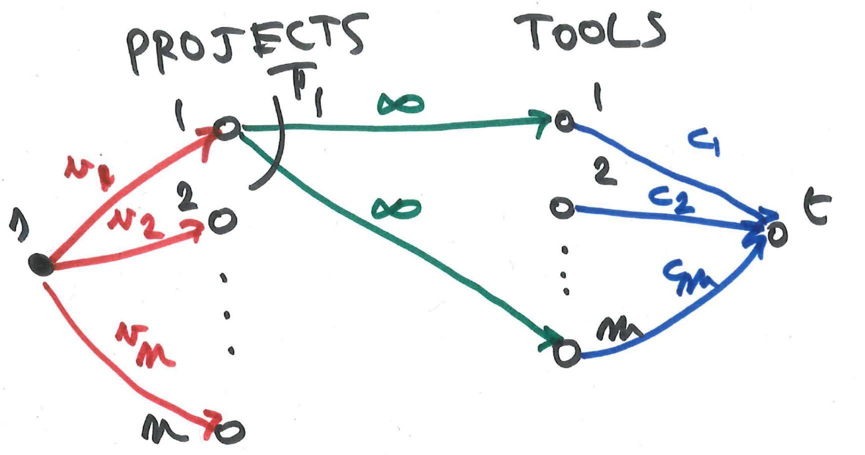
$$(*) \doteq \sum_{j \in J} c_j + \sum_{i \in [n] \setminus I} v_i .$$

- Construct network N such that $(*) = c(S, T)$ where I are the projects in S.
- Enforce condition $(\forall i \in I)\ T_i \subseteq J$ by including edges (i, j) with $c(i, j) = \infty$ for each $i \in [n]$ and $j \in T_i$, and let J be the tools in S.

Project Selection – reduction to min cut

- Need to find $I \subseteq [n]$ and $J \subseteq [m]$ with $(\forall i \in I)\ T_i \subseteq J$ minimizing
$$(*) \doteq \sum_{j \in J} c_j + \sum_{i \in [n] \setminus I} v_i .$$
- Construct network N such that $(*) = c(S, T)$ where I are the projects in S.
- Enforce condition $(\forall i \in I)\ T_i \subseteq J$ by including edges (i, j) with $c(i, j) = \infty$ for each $i \in [n]$ and $j \in T_i$, and let J be the tools in S.
- Resulting algorithm runs in time $O((n + m)(n + m + \sum_{i \in [n]} |T_i|))$.

Project Selection – reduction to min cut



$T_1 = \{1, m\}$