APPLICATIONS MAX-FLOW MIN-CUT

- First, recalls some facts

Relationship between Max-flow and Min-cut Capacity

The minimum s-t cut problem reduces in linear time to the maximum flow problem

Proof: Assume a residual graph G_f, and we know the max flow.

- Then we can just perform a DFS/BFS, which takes linear time from s to t.
- Since no path $s \to t$, the DFS/BFS will stop before reaching t

Let (A, B) be the cut s.t. $\begin{cases} v \in A: v \text{ is visited during DFS/BFS} \\ w \in B: w \text{ is not visited during DFS/BFS} \end{cases}$

$\Rightarrow$ The cut (A, B) is the minimum-cut that is derived from a given max-flow in $O(n)$ time

Image Segmentation Problem as Min-Cut Capacity Problem

Problem:

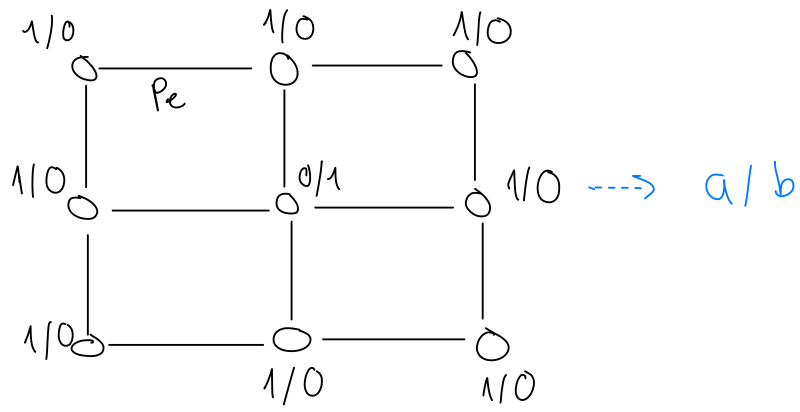
Input: A image

Goal: Segment the image into "foreground" and "background"

The setup:

Think of the image as an undirected graph G, each pixel is a vertex with 2 values $\begin{cases} a_v : v \in X: \text{foreground set} \\ b_v : v \in Y: \text{background set} \end{cases}$

each edge has a value p_e



We try to maximize this objective function:

$$f = \sum_{v \in X} a_v + \sum_{v \in Y} b_v - \sum_{e \in \delta(X)} p_e$$

where
- X: foreground set
- Y: background set
- $\delta(X)$: set of edges cut by (X, Y)

Observations:

- Function f is the maximum likelihood function
- Vertex v "earns" a prize a_v if $v \in X$, similarly
  "earns" a prize b_v if $v \in Y$
- An optimal set X and Y $\Leftrightarrow$ Minimum penalty term $\sum_{e \in \delta(X)} p_e$

$\Rightarrow$ By the last observation, we can "kinda" see that "If we partition X and Y in a way that minimize the penalty term, then the Image Segmentation problem is solved"

$\Rightarrow$ Very similar to min-cut capacity problem

But there is a problem: How to cast this problem as a Min-cut capacity problem

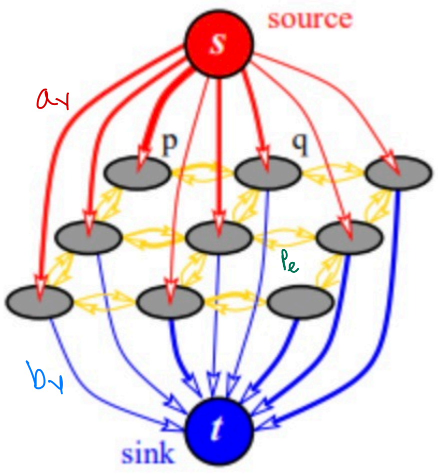
Some differences we can see in Image Segmentation problem and Min-Cut Capacity problem are:

1. No source (s) and sink (t), undirected edge
2. Each vertex in Image Segmentation problem has 2 values, compare to 1 value in Min-Cut Capacity problem.
3. Maximize problem vs Minimizing problem

Lets address this one by one:

1. No source (s) and sink (t), undirected edge
   $\Rightarrow$ Add source (s) and sink (t), convert undirected edge to two directed edges
2. Each vertex in Image Segmentation problem has 2 values, compare to 1 value in Min-Cut Capacity problem.
   $\Rightarrow$ edge (s, v) has capacity $c(s, v) = a_v$
   edge (v, t) has capacity $c(v, t) = b_v$

Base on those description, we can construct a graph G' as follows:



3. Maximize objective function in G vs Minimize cut capacity in G'

We know: Objective function f in G

$$f = \sum_{v \in X} a_v + \sum_{v \in Y} b_v - \sum_{e \in \delta(X)} p_e$$

And: $\quad \operatorname{argmax}(f) = \operatorname{argmin}(-f)$

So, the corresponding objective function in G' has the form

$$f' = -f$$

$$= -\sum_{v \in X} a_v - \sum_{v \in Y} b_v + \sum_{e \in \delta(X)} p_e$$

We can get rid of the negative values by using this fact:

$$\operatorname{argmin}(f') = \operatorname{argmin}(f' + C)$$

Let $C = \sum_{v \in V} a_v + \sum_{v \in Y} b_v$

Then: $\quad f' + C = \sum_{v \in Y} a_v + \sum_{v \in X} b_v + \sum_{e \in \delta(X)} p_e$

objective function of G'

Claim: $\exists$ bijection (one-to-one) between

Partition (X, Y) of $G \longleftrightarrow (S, T)$ cut of G'

such that Objective function f is preserved

We prove 2 things:

- $(X, Y) \longleftrightarrow (S, T)$, Uniqueness, one-to-one
- Cut capacity = objective function f', Preservation

• First: $(X, Y) \longleftrightarrow (S, T)$

By construction of G': $\begin{cases} S = X + (s) \\ T = Y + (t) \end{cases}$

$\Rightarrow$ Proven

- Second: Cut capacity = objective function f'

For a cut $(X + \textcircled{s}, Y + \textcircled{t})$, there are 3 cases:

- $(s, v)\ \forall\ v \in Y \iff c(s,v) = a_v$
- $(v, t)\ \forall\ v \in X \iff c(v, t) = b_v$
- $(u, v)\ \forall \begin{cases} u \in X \\ v \in Y \end{cases} \iff c(u, v) = p_e$

Easy to see that the Cut capacity for any cut $(X + \textcircled{s}, Y + \textcircled{t})$ is:
$$\sum_{v \in Y} a_v + \sum_{v \in X} b_v + \sum_{e \in \delta(X)} p_e$$

$\Rightarrow$ which is the same as objective function f' in G'

Since: $\underset{(X,Y)}{\text{argmax}}\ f$ in $G \longleftrightarrow \underset{(S,T)}{\text{argmin}}\ f'$ in G'

Conclude that: $\exists$ bijection between 2 sets G and G'

such that: $\underset{(X,Y)}{\text{argmax}}\ f$ in $G \leftrightarrow \underset{(S,T)}{\text{argmin}}\ f'$ in G'

$\Rightarrow$ Proven Claim

Bipartite Matching as Max-flow

The problem:

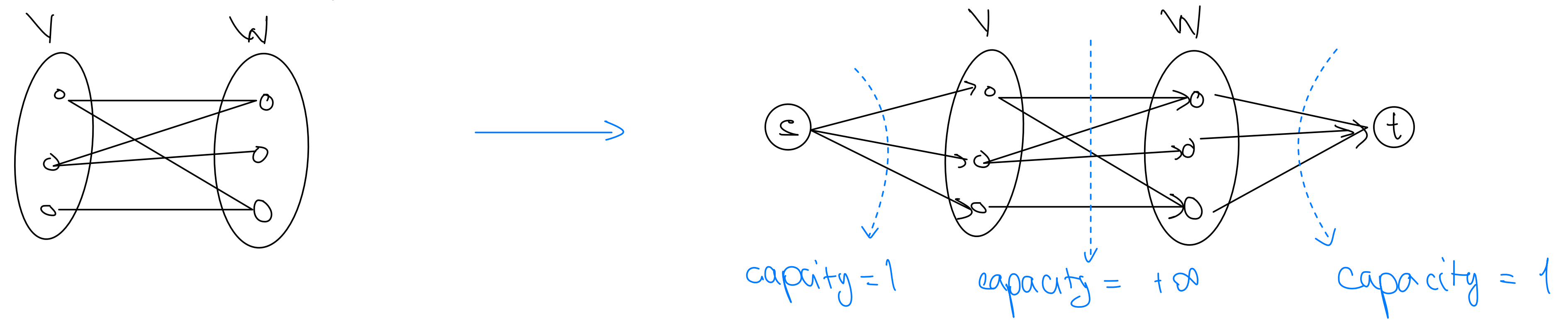
Input: Bipartite graph $G = (V, W, E)$

Matching: Subset of edges $M \subseteq E$ with no shared endpoints

Goal: Find max-cardinality subset M

Claim: Bipartite Matching Problem reduces to Max-flow problem in $O(n)$

Convert bipartite graph G to flow network G'



Prove the claim by proving: $\exists$ bijection between

Matchings in G $\longleftrightarrow$ Integral flows in G'

(such that preserves Objective function)

Proof:

- $\forall$ edge (v, w) s.t. $v \in V$, $w \in W$ that is a match in G $\Leftrightarrow$ $0 < f(v, w) < c(u, w)$

  $\Rightarrow$ edges $\begin{cases} (s, v) = 1 \\ (w, t) = 1 \end{cases}$

  $\Rightarrow$ create a flow in G': $s \rightarrow v \rightarrow w \rightarrow t$ (1)
- $\forall$ vertex v of a matching edge (v, w) cannot belongs to another matching edge (2)
- $\forall$ vertex v belongs to a flow $s \Rightarrow v \rightarrow w \rightarrow t$ cannot appear in an augmenting path again (3)

(1), (2) and (3) $\Rightarrow$ matchings in G $\longleftrightarrow$ flows in G'

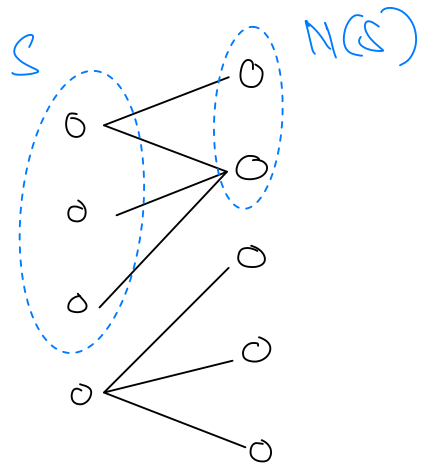
Hall's theorem

Assume V is the LHS set, and $|LHS| < |RHS|$:

$\exists$ perfect matching $\Leftrightarrow$ $\forall S \subseteq V$, $|S| \leq |N(S)|$

In words:

If you can find a "restricting subset" in your bipartite graph, meaning a set S in the LHS (assume $|LHS| < |RHS|$) such that $|S| > |N(S)|$, then there are some vertices in S without a match



$\Rightarrow$ Can't match all of S

$\Rightarrow$ No perfect-matching

Proof: Hall's theorem

- Case 1: Perfect matching (i.e. all V matches all W)

  $\Rightarrow$ Trivially true

- Case 2: Non-perfect matching (Given 100 edges between V and W, and 40 matchings. How do we know we have the max-cardinality matching?)