Image Segmentation

- Using an expensive camera and appropriate lenses, you can get a “bokeh” effect on portrait photos in which the background is blurred and the foreground is in focus.
- But using cheap cameras in phones and appropriate software you can fake this effect...

Formulating the problem

- **Segmentation** or Foreground-Background separation.
- **Input:**
  - Set \( S \) pixels in \( n \times n \) grid with set \( N \) of neighboring pairs.
  - \( i \)th pixel has foreground score \( f_i \geq 0 \) and background score \( b_i \geq 0 \).
  - Each \((i, j)\) in \( N \) has penalty \( p_{ij} \geq 0 \) for labeling one in foreground and one in background.
- **Goal:** Partition pixels into foreground \( F \) and background \( B = S \setminus F \) to maximize
  \[
  \text{score}(F, B) = \sum_{i \in F} f_i + \sum_{j \in B} b_j - \sum_{(i, j) \in N : i \in F, j \in B} p_{ij}
  \]

Turning the problem into a network flow problem

- Define the directed graph \( G \) where
  - Pixels \( S \) are nodes of \( G \)
  - For each \((i, j)\) in \( N \), edge in each direction with capacity \( p_{ij} \)
  - Node \( s \) with an edge to each pixel \( i \) with capacity \( f_i \)
  - Node \( t \) with an edge from each pixel \( j \) with capacity \( b_j \)
- **Observe:**
  \[
  \text{score}'(F, B) = \sum_{i \in F} f_i + \sum_{j \in B} b_j + \sum_{(i, j) \in N : i \notin F, j \in B} p_{ij}
  \]
- **Define the directed graph \( G \) where**
  - Pixels \( S \), nodes of \( G \)
  - For each \((i, j)\) in \( N \), edge in each direction with capacity \( p_{ij} \)
  - Node \( s \) with an edge to each pixel \( i \) with capacity \( f_i \)
  - Node \( t \) with an edge from each pixel \( j \) with capacity \( b_j \)
- **Observe:**
  \[
  \text{score}'(F, B) = \text{cut}(F, B)
  \]
- **So finding minimum cut in \( G \) is equivalent to maximizing the image segmentation score.**
Example

Formally...

Claim. \((F^*, B^*)\) maximizes score ⇔ \((F^*, B^*)\) minimizes cut.

- Suppose \((F^*, B^*)\) maximizes score. Then for any \((F, B)\),
  \[
  0 \leq \text{score}(F^*, B^*) - \text{score}(F, B)
  = \text{score}(F, B) - \text{score}'(F^*, B^*)
  = \text{cut}(F, B) - \text{cut}'(F^*, B^*)
  
  \]
- Hence \((F^*, B^*)\) also minimizes cut.
- Other direction is analogous.

Note: You have to prove both.

Reducibility and Intractability

- Claim 1. If \(Y \leq^p X\) and \(X\) poly-time solvable, so is \(Y\).
  - Can use to design algorithms.
- Claim 2. If \(Y \leq^p X\) and \(Y\) not poly-time solvable, then \(X\) is not either.
  - Contrapositive of above.
  - Can be used to prove hardness.
  - The catch: we do not know of any problem \(Y\) that provably cannot be solved in polynomial time.

A first reduction

**Definition.** \(S \subset V\) is an independent set in a graph \(G = (V, E)\) if no nodes in \(S\) share an edge.

**Problem.** Does \(G\) have independent set of size at least \(k\)?

**Definition.** \(S \subset V\) is a vertex cover in a graph \(G = (V, E)\) if every edge adjacent to some \(v \in S\).

**Problem.** Does \(G\) have vertex cover of size at most \(k\)?

**Theorem.** \(\text{INDEPENDENTSET} \leq^p \text{VERTEXCOVER}\) and \(\text{VERTEXCOVER} \leq^p \text{INDEPENDENTSET}\).
Reduction #2: Set cover

**Problem.** Given a set $U$ of $n$ elements, subsets $S_1, \ldots, S_m \subseteq U$, and a number $k$, does there exist a collection of at most $k$ subsets $S_i$ whose union is $U$?

- Example:
  - $U$ is the set of all skills.
  - Each $S_i$ is a person.
  - Want to find a small team that has all skills.

- **Theorem.** $\text{VERTEXCOVER} \leq_p \text{SETCOVER}$

Interlude

- **Decision versus Optimization**
  - Algorithms so far have been for optimization
  - Reductions so far have been for decision
  - But can reduce optimization to decision and vice versa.
  - e.g., solve $\text{MAXINDSET}(G)$ by solving $\text{INDSET}(G, k)$ for $k = 1, \ldots, n$.
  - e.g., solve $\text{INDSET}(G, k)$ by computing $S = \text{MAXINDSET}(G)$ and output $1[|S| \geq k]$.

A bad reduction.

Given $\text{VERTEXCOVER}$ instance $(G, k)$, make $\text{SETCOVER}$ instance with $U = E$, $S_v$ is edges incident to $v$, $S_0 = U$, and integer $k$.

- If $G$ has VC of size at most $k$, then $U$ has cover of size at most $k$.
- But if $U$ has cover of size $k$, $G$ might not!

If $(G, k)$ is a No instance, the reduction does not correctly return No.

Set cover reduction

**Reduction.** Given $G = (V, E)$ make set cover instance with $U = E$, and $S_v$ is all edges incident to $v$. Keep $k$ the same.

**Proof.** $U$ covered with at most $k$ sets if and only if $E$ covered by at most $k$ vertices.

- If $v_1, \ldots, v_k$ is a VC then $S_{v_1}, \ldots, S_{v_k}$ is a SC.
- If $S_{v_1}, \ldots, S_{v_k}$ covers $U$, then every edge adjacent to one of $\{v_1, \ldots, v_k\}$.

Common Confusions

$Y \leq_p X$ means:

- $Y$ is “no harder” than $X$
- $X$ is “at least as hard” as $Y$.
- To show $Y$ is easy, show $Y \leq_p X$ for easy $Y$.
- To show $X$ is hard, show $Y \leq_p X$ for hard $Y$.

For decision problem $Y$, need to show two things.

- Correctly outputs $\text{YES}$ and $\text{NO}$.

Reduction #3: Satisfiability

- Can we determine if a boolean formula has a satisfying assignment?
  - Let $X = \{x_1, \ldots, x_n\}$ be boolean variables
    - A term or literal is $x_i$ or $\bar{x}_i$.
    - A clause is or of several terms $(t_1 \lor t_2 \lor \ldots \lor t_k)$.
    - A formula is and of several clauses
    - An assignment $v : X \rightarrow \{0, 1\}$ gives T/F to each variable.
  - $v$ satisfies formula if all clauses evaluate to True.

**Example.**

$$(x_1 \lor \bar{x}_2) \land (x_1 \lor x_4 \lor \bar{x}_3) \land (\bar{x}_1 \lor x_4) \land (x_3 \lor x_2)$$
Reduction #3: Satisfiability

SAT – Given boolean formula $C_1 \land C_2 \ldots \land C_m$ over variables $X = \{x_1, \ldots, x_n\}$, does there exist a satisfying assignment?

3-SAT – Given boolean formula $C_1 \land C_2 \ldots \land C_m$ over variables $X = \{x_1, \ldots, x_n\}$ where each $C_i$ has three literals, does there exist a satisfying assignment?

▶ Any algorithms?

**Theorem.** $3$-SAT $\leq_P$ IndependentSet.

Reduction #3: Satisfiability

$(x_1 \lor x_2 \lor \bar{x}_3) \land (\bar{x}_1 \lor \bar{x}_2 \lor \bar{x}_3)$

▶ Associate nodes in graph with literals ($\geq 2$ per variable).
▶ If $v(x_i) = 1$ in assignment, then cannot select some nodes.
▶ Associate 3 nodes per clause in a gadget.

Claim Graph has IS of size $n + m$ if and only if formula satisfiable.

▶ If formula satisfiable, select correct term on the left and one per clause on the right.
▶ If graph has IS,
  ▶ At most one node per clause on the right
  ▶ At most one node per variable on the left.
  ▶ If node selected in clause, its negation cannot be selected.