Combinatorial Optimization in Computer Vision

Chapter 2: Graph Cuts

WS 2011/12

Ulrich Schlickewei
Computer Vision and Pattern Recognition Group
Technische Universität München

Plan for Today

1. From Image Segmentation to Graph Cuts
2. Min-Cut and Max-Flow Problem
3. Augmenting Path Algorithm by Ford and Fulkerson

Image Segmentation

Decompose an image into foreground (lion) and background (forest)

image courtesy of C. Nieuwenhuis

Reminder of Last Lecture

We have seen that MAP inference in a Markov Random Field (MRF) model for binary image segmentation leads to an optimization problem

$$\min_{x \in \{0,1\}^{m \cdot n}} \underbrace{\sum_{i,j} w_{ij}^{\text{fg}} x_{ij} + w_{ij}^{\text{bg}} (1 - x_{ij})}_{\text{data term}} + \underbrace{\sum_{(i,j) \sim (k,l)} w_{ij,kl}^{\text{prior}} \delta(x_{ij}, x_{kl})}_{\text{regularizer}}$$

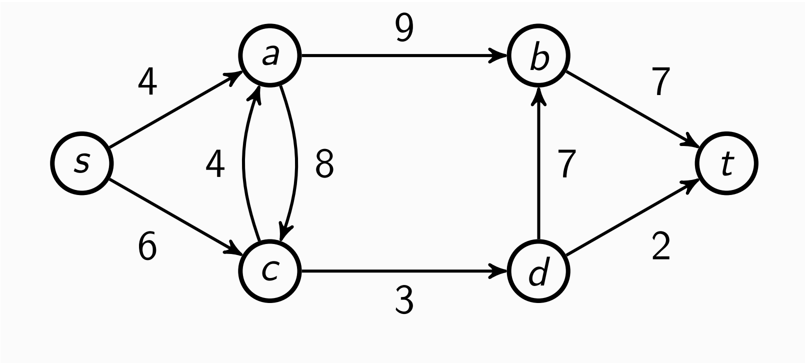
where

$$\delta(x_{ij}, x_{kl}) = \begin{cases} 1 & \text{if } x_{ij} = x_{kl} \\ 0 & \text{else.} \end{cases}$$

Networks

Definition: A network N = (V, E, s, t, c) consists of

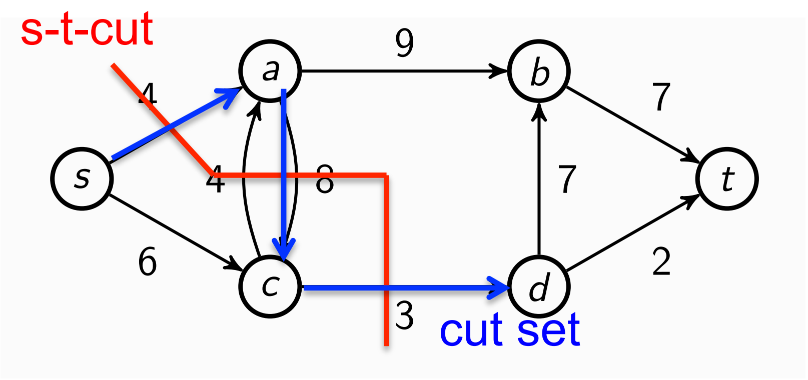
- a directed graph (V,E)
- a source $s \in V$ and a sink $t \in V$ (meaning that all edges adjacent to s are outgoing edges and all edges adjacent to t are incoming edges)
- a non-negative capacity function $c : E \rightarrow \mathbf{R}_{\geq 0}$



Cut

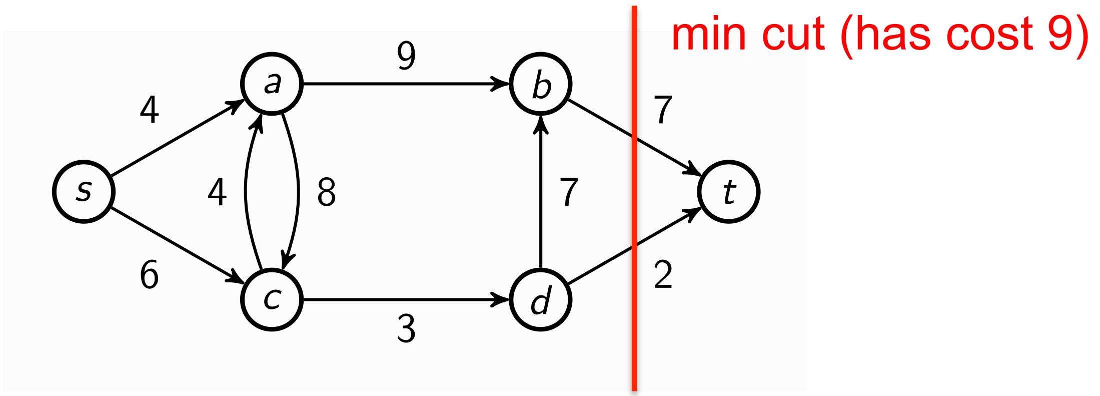
Definition: Let N = (V, E, s, t, u) be a network.

- An s-t-cut on N is a partition of V into two disjoint subsets S and T such that s ∈ S and t ∈ T.
- The cut set C of an s-t-cut is the set of edges (p,q) with p ∈ S and q ∈ T

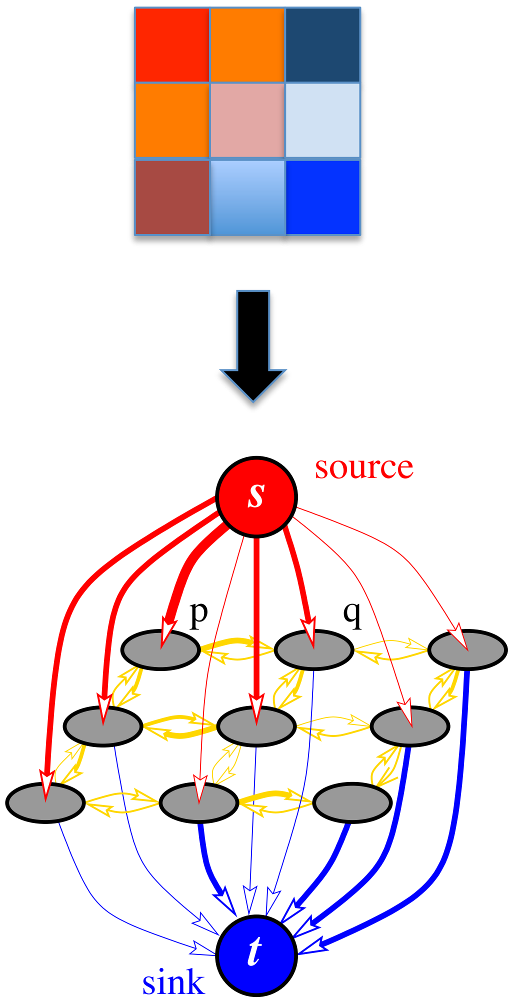


Minimal Cut

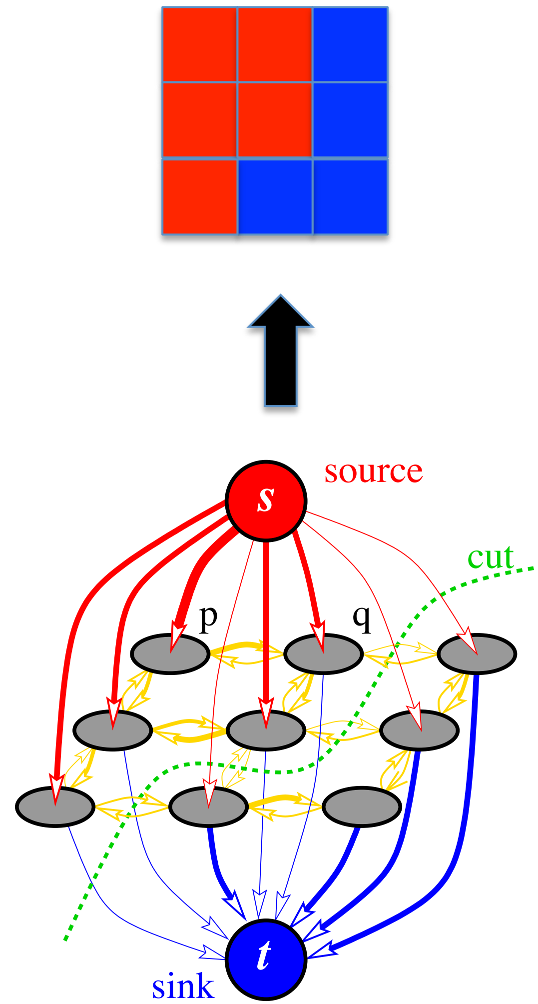
- The cost of an s-t-cut (S,T) with cut set C is given by $c(S,T) = \sum_{e \in \mathcal{S}} u(e)$.
- A cut is minimal if it has minimal cost among all cuts.



From Image Segmentation to Graph Cuts I



(a) A graph $\mathcal{G}$



(b) A cut on $\mathcal{G}$

From Image Segmentation to Graph Cuts II

More formally, given an image and the corresponding (4-neighborhood-) graph, we construct a network N = (V, E, s, t, u) by introducing

- a source s (representing the object/foreground) and a sink t (representing the background)
- directed edges starting in s ending in p and starting in p and ending in t for all pixels p (t-links)
- directed edges starting in p and ending in q for all pixels p and all neighbors q of p (n-links)
- appropriate capacities u (to be defined below)

Optimization Problem

- Given a minimum cut in the network described above, we can define a segmentation x by setting

$$x_{ij} = \begin{cases} 1 & \text{if } (i,j) \in S \\ 0 & \text{if } (i,j) \in T \end{cases}$$

- Recall that we aim at solving the optimization problem

$$\min_{x \in \{0,1\}^{m \cdot n}} \underbrace{\sum_{i,j} w_{ij}^{\text{fg}} x_{ij} + w_{ij}^{\text{bg}} (1 - x_{ij})}_{\text{data term}} + \underbrace{\sum_{(i,j) \sim (k,l)} w_{ij,kl}^{\text{prior}} \delta(x_{ij}, x_{kl})}_{\text{regularizer}}$$

Optimal Segmentation = min-cut

Theorem (Boykov, Jolly): Let V = S \cup T be a minimal cut in the image graph constructed as above. Then the induced segmentation x is a global minimizer of the optimization problem

$$\min_{x \in \{0,1\}^{m \cdot n} } \underbrace{\sum_{i,j} w_{ij}^{\text{fg}} x_{ij} + w_{ij}^{\text{bg}} (1 - x_{ij})}_{\text{data term}} + \underbrace{\sum_{(i,j) \sim (k,l)} w_{ij,kl}^{\text{prior}} \delta(x_{ij}, x_{kl})}_{\text{regularizer}} .$$

Plan for Today

1. From Image Segmentation to Graph Cuts
2. Min-Cut and Max-Flow Problem
3. Augmenting Path Algorithm by Ford and Fulkerson

Network Flow

Definition: Let N = (V, E, s, t, c) be a network. A flow in N is a map $f : E \to \mathbf{R}_{\geq 0}$ which satisfies

- (capacity constraints) for all $e \in E$: $f(e) \leq c(e)$,
- (flow conservation) for all nodes $v \in V \setminus \{s,t\}$:

$$\sum_{(u,v)\in E} f(u,v) - \sum_{(v,w)\in E} f(v,w) = 0.$$

Value of Flow

- **Intuition**: We pump water from the source to the sink. The value f(e) is the amount of water wich flows through edge e.
- The value of a flow f is the amount of water being transported from the source to the sink:

$$|f| = \sum_{(s,v)\in E} f(s,v).$$

Max-Flow-Problem

Problem: Find the maximal flow f through N.

It turns out that the max-flow problem is equivalent to the min-cut problem.

Weak Duality

Lemma: Let f be a flow in N, let (S,T) be a cut. Then

$$|f| \leq c(S,T).$$

Proof: We have

$$\begin{aligned} |f| &= \sum_{(u,v)\in E, u\in S, v\in T} f(u,v) - \sum_{(v,w)\in E, v\in T, w\in S} f(v,w) \\ &\leq \sum_{(u,v), v\in T, w\in S} c(u,v) \\ &= c(S,T). \end{aligned}$$

The Lemma implies that

$$\max_{f \text{ flow}} |f| \leq \min_{(S,T) \text{cut}} c(S,T).$$

Strong Duality

Theorem (Ford-Fulkerson ´56, Elias-Feinstein-Shannon ´56): Let f be a maximum flow in N, let (S,T) be a minimal cut. Then

$$|f| = c(S,T).$$

This theorem is usually called min-cut-max-flow-theorem. We will see the proof below.

Plan for Today

1. From Image Segmentation to Graph Cuts
2. Min-Cut and Max-Flow Problem
3. Augmenting Path Algorithm by Ford and Fulkerson

Residual Network

Definition: Let N = (V, E, s, t, c) be a network and f a flow through N. The residual network $N_f = (V_f, E_f, s_f, t_f, c_f)$ is defined by

- the set of vertices $V_f = V$, the source s_f and the sink t_f,
- the set of edges

$$E_f = \{e \in E \mid f(e) < c(e)\} \cup \{e^{-1} \mid e \in E \text{ and } f(e) > 0\},$$

- the capacities

$$c_f(e) = \begin{cases} c(e) - f(e) & \text{if } f(e) < c(e) \\ f(e^{-1}) & \text{if } e^{-1} \in E \text{ and } f(e^{-1}) > 0. \end{cases}$$

Augmenting Path

Definition: Let N = (V, E, s, t, c) be a network with flow f. An augmenting path for f is a path with no cycles connecting s to t in the residual network N_f.

Key Observation

- Let $P \subset E$ be an augmenting path for a flow f.
- Let $m_P = \min_{e \in P} c_f(e) > 0$.
- Then we can define a flow f_P with $|f_P| = |f| + m_P$ by setting for $e \in E$

$$f_P(e) = \begin{cases} f(e) & \text{if } e \notin P \\ f(e) + m_P & \text{if } e \in E \cap P \\ f(e) - m_P & \text{if } e^{-1} \in E_f \cap P \end{cases}$$

Augmenting Path Algorithm

Based on this observation, the following algorithm, proposed by Ford and Fulkerson is very natural:

1. Set $f_0 = 0$.
2. As long as there exists an augmenting path P in N_f
   - replace f with f_P
   - search an augmenting path P in N_{f_P}
3. Return f_P

Ford-Fulkerson Theorem

Theorem (Ford-Fulkerson ´56): The following are equivalent for a flow f in N:

a) f is maximal.

b) There is no augmenting path in N_f.

c) There exists a cut (S,T) in N of cost $c(S,T) = |f|$.

Strong Duality

Corollary 1 (Strong duality)**:** We have

$$\max_{f \text{ flow in } N} |f| = \min_{(S,T) \text{ cut in } N} c(S,T).$$

Corollary 2 (Partial correctness of Ford-Fulkerson algorithm)**:** If the augmenting path algorithm halts, it returns a maximal flow.

Does the Augmenting Path Algorithm Halt?

- If N has integer capacities, then the flow increases in each iteration by at least 1. Thus, the algorithm halts in this case.
- This implies that it also halts in the case of rational coefficients.
- In general, there are examples for which it does not halt.

Dinic and Edmonds-Karp Algorithm

- A variant of the augmenting path algorithm has been proposed by independently by Dinic and by Edmonds and Karp.
- The key issue is to search for the shortest augmenting path (e.g. using a breadth-first search).

Polynomial Time

Theorem (Dinic ´70, Edmonds-Karp ´72): If in each iteration the augmenting path of shortest length (=number of edges) is chosen, then the algorithm terminates in polynomial time.

Remarks

- Boykov and Kolomogorov have proposed an alternative augmenting-path-style algorithm which is not guaranteed to finish in polynomial time but which is typically more efficient for image processing graphs.
- Kolmogorov and Zabih have studied which quadratic pseudo-boolean energies can be solved via graph cuts other than image segmentation. It turns out that the energy has to be submodular. We will come to this in a later lecture.

Summary

- Binary image segmentation can be reformulated as a graph cut problem.
- Finding a minimal cut is equivalent to finding a maximum flow.
- The augmenting path algorithm tries explores iteratively new paths to pump more flow through a network.
- If shortest augmenting paths are chosen, the algorithm returns a maximum flow in polynomial time.