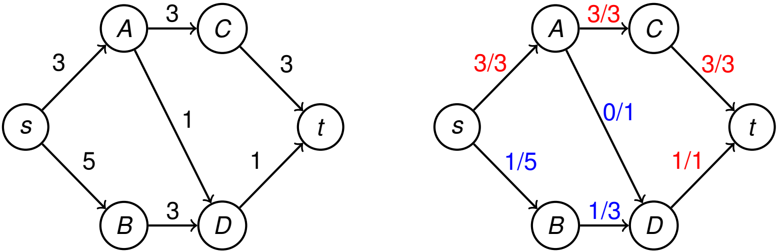
Lecture Pattern Analysis

Part 24: Recap: Max Flow and Min Cut

Christian Riess
IT Security Infrastructures Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg
July 26, 2022

Overview

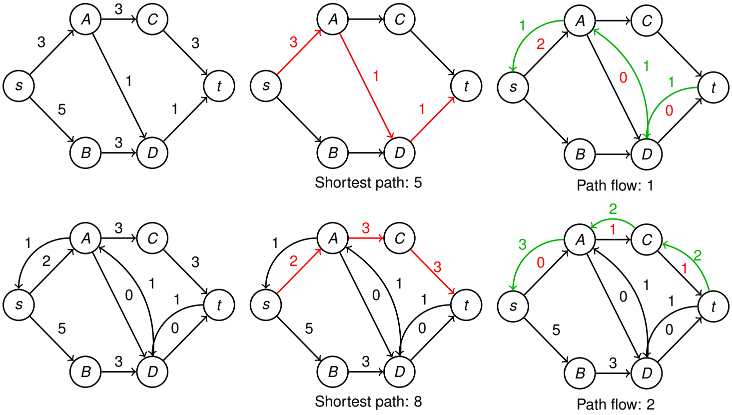
- Max flow is a combinatorial standard problem, solved in polynomial time
- Given: graph with positive edge weights, source node, sink node
- Task: If edge weights are tube capacities, then determine the maximum possible throughput of water (“flow”) from source *s* to sink *t* per time unit:



- The minimum cut task seeks the smallest sum of edges to disconnect *s* and *t*.
- Max flow and min cut are identical: a min cut is easily found, e.g., by selecting the red edges until there is no *s*-*t* path left

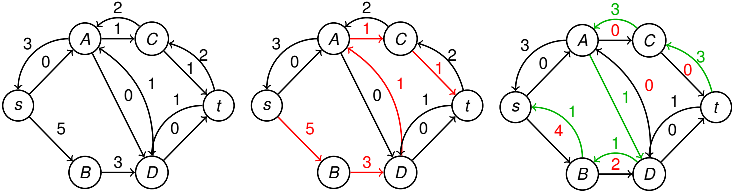
Ford Fulkerson in a Nutshell (1/2)

- Max flow algorithm by Ford and Fulkerson is probably most well-known:
  1. Greedily search shortest path
  2. Max out the flow capacity along that path, reduce edge weights
  3. Introduce backward edges to undo greedy dead ends, goto 1) if *s*-*t* path left



Ford Fulkerson in a Nutshell (2/2)

- The third shortest path uses a back link, and completes the max flow algorithm:



Shortest path: 11

Path flow: 1

- The total flow is $1 + 2 + 1 = 4$, with pipe usage as shown on slide 1
- The minimum cut includes the edge $D \to t$ and any one of the edges $(s \to A, A \to C, C \to t)$
- Hence, the four sets of edges for equivalent minimum cuts are
  - $(C \to t, D \to t)$,
  - $(A \to C, D \to t)$, and
  - $(s \to A, A \to D, D \to t)$, where $A \to D$ goes backwards and does not count

Lecture Pattern Analysis

Part 25: MRF Inference via Min Cuts

Christian Riess
IT Security Infrastructures Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg
July 25, 2022

Overview

- The MRF inference task is to find optimal label assignments

$$z_1^*, \dots, z_N^* = \operatorname*{argmax}_{z_1,\dots,z_N} \frac{1}{Z} \exp\left(-\sum_i E(x_i, z_i) - \sum_{i,j} E(z_i, z_j) \right) \tag{1}$$

  (where we limited the maximum clique size to 2, i.e., each term includes at most two hidden variables)
- This is equivalent to the minimization of the sum of energy terms

$$z_1^*, \dots, z_N^* = \operatorname*{argmin}_{z_1,\dots,z_N} \sum_i E(x_i, z_i) + \sum_{i,j} E(z_i, z_j) \tag{2}$$

- The idea of MRF inference via graph cuts[1] is to
  - encode these energy terms in a specialized graph,
  - such that that graph's minimum cut also minimizes the sum of energy terms

[1] The literature reference for this lecture is the paper by Kolmogorov and Zabih, which is uploaded to studOn

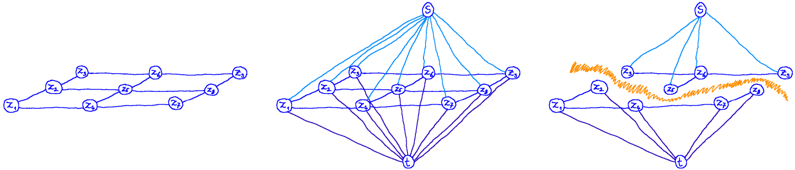
Constraints and Benefits

- The construction requires
  - binary labels (for convenience, I will just write “0” and “1”),
  - maximum clique size of 2, and
  - that pairwise energy terms satisfy the **submodularity condition**

$$E(0,0) + E(1,1) \leq E(0,1) + E(1,0) \tag{3}$$

- Under these constraints, the algorithm finds
  - a globally optimal labeling
  - in polynomial time
- The **α-expansion algorithm** extends the method to non-binary labelings
- α-expansion is only locally optimal, but within a guaranteed margin around the global optimum

Construction Idea



1 Start with the neighborhood relationship of the hidden variables z_i

2.a To encode the optimization problem, add a source s and sink t

2.b Identify s with label 0 and t with label 1

2.c Set appropriately chosen edges and edge weights between all nodes

3.a Calculate minimum cut

3.b Nodes connected with s obtain label 0, the others obtain label 1

3.c The minimal s-t cut is identical to the minimal-energy binary labeling of the MRF

Additivity of Graphs

- Let us clarify how this magic works
- Key to success is that the min cut construction is homomorphic under graph composition:
  - if G_1 encodes $\min(E_1)$ and G_2 encodes $\min(E_2)$,
  - then $(G_1 \cup G_2)$ encodes $\min(E_1 + E_2)$
- Hence, if you encode each energy term such that it is optimal under min cut, then the combination of all energy terms will also be optimal under min cut
- I think the beauty of this result speaks for itself.
- So, let us now look for optimal encodings of the unary and pairwise potentials

Encoding of Unary Energy Terms

- Graph construction for a single unary term $E(x_i, z_i)$, with $s = 0$, $t = 1$:

$$s \xrightarrow{\quad E(x_i, 1) \quad} z_i \xrightarrow{\quad E(x_i, 0) \quad} t$$

- For example, a cut between s and z_i assigns the label $z_i = t = 1$
- Hence, the cost is $E(x_i, 1)$, which relates observation x_i to label 1

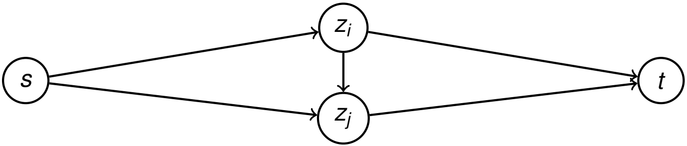
- Note that the minimum cut remains the same if we construct an equivalent smaller graph: For example, if $E(x_i, 1) > E(x_i, 0)$, the equivalent graph is

$$s \xrightarrow{\quad E(x_i, 1) - E(x_i, 0) \quad} z_i \qquad\qquad t$$

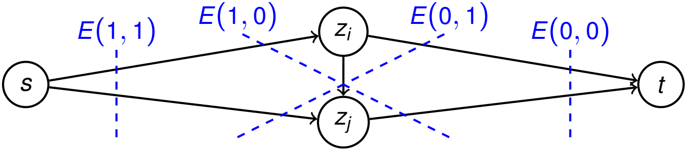
- Conversely, if $E(x_i, 1) < E(x_i, 0)$, then we can just use weight $E(x_i, 0) - E(x_i, 1)$ between z_i and t and remove the other edge

Encoding of Pairwise Energy Terms (1/2)

- Graph construction for a single pairwise term $E(z_i, z_j)$, with $s = 0$, $t = 1$:



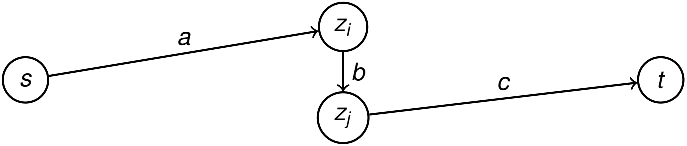
- Possible cuts and associated costs:



- This graph has 5 edges, but only 4 terms $E(0,0)$, $E(0,1)$, $E(1,0)$, $E(1,1)$
- Hence, let us use immediately write the equivalent smaller graph

Encoding of Pairwise Energy Terms (2/2)

- Most (submodular!) tasks satisfy $E(1,0) > E(0,0)$ and $E(1,0) > E(1,1)$
- These relations admit the simplified graph



with edge weights a, b, c obtained from the linear system of equations

$$a + k = E(1,1) \tag{4}$$
$$c + k = E(0,0) \tag{5}$$
$$b + k = E(0,1) \tag{6}$$
$$a + c + k = E(1,0) \tag{7}$$

with constant offset k, and without b in Eqn. 7 since this is a backward edge

Graph Cut Inference for Binary and Non-Binary Labels

- The full graph is constructed by summing the subgraphs of all energy terms
- Min cut on that full graph finds in **polynomial time** a solution that is **globally optimal** for binary labels
- Non-binary labellings can be found via α-expansion:
  1. From the set of labels, select one specific label α
  2. Fix hidden variables that already have label α
  3. Seek the lowest energy labelling that switches at least one other hidden variable to α
  4. Keep this “expanded” α labelling if its energy is lower than the current energy

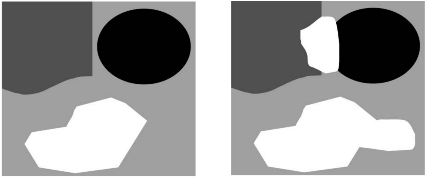


Fig. 1. An example of an expansion move. The labeling on the right is a white-expansion move from the labeling on the left.