The Potts Model and the Multi-Way Cut Problem

Scribe: Remik Ziemlinski **Lecturer: Prof. R. Zabih**
CS664 Lecture #8 **Machine Vision** **9.27.2001**

1. Motivation

Recall the energy function $E(f)$. In terms of the Potts model we may informally write the energy function as

$$E(f) = \sum_{p \in P} C(p, f(p)) + \lambda\left(\#\text{ discontinuities}\right).$$

The number of discontinuities may also be thought of as the number of disconnected components. This is what we strive for in the Multi-Way Cut (MWC) problem. Our goal is to separate terminal [label] nodes from pixel nodes through inspection of wieghts (Fig. 1). We are therefore looking for the cheapest multiway cut.

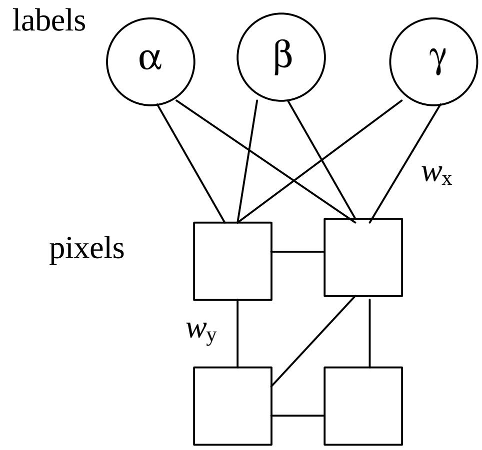


Figure 1

We know that solving the Potts problem is NP-hard, but so is MWC. What we want is to turn an instance of the Potts problem into a MWC, and vice-versa. If that is the case, then a 2-Approximation algorithm (2-APPR) for MWC would suggest $\exists$ 2-APPR for the Potts problem. Before we derive this, let's examine MWC some more.

The weights on the arcs in the graph, such as w_x (Fig. 1) arise from the "smoothness" term in $E(f)$. The weights among the pixels, i.e. w_y, are from the "data" term λ in $E(f)$. These pixel-pixel arcs are also known as "n-links", whereas the label-pixel arcs are "d-links".

MWC is defined by the minimum cut property. So for a graph $G = (V, E)$, we look for a set $C \subset E$ which separates the vertex sets S and T such that $V = S \cup T$. It is important to realize that C has no subsets that can do this as well.

But suppose we are given the graph in Fig. 2. This would not be a MWC! The reason is that we can't have $(p,q) \in C \wedge (q,r) \in C$. If they both belonged to C then $\exists$ path $\langle \alpha, q, r, \beta \rangle$. In the end, if we do have a MWC then a pixel $p \in S = P$ is connected to no more than 1 label (terminal) $l \in T$. Conceptually, one may envision this as each terminal grabbing a piece of an image.

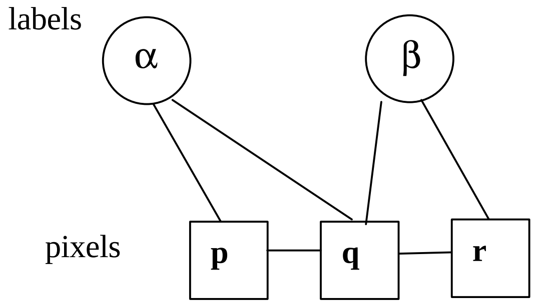


Figure 2

In order to make our graph behave correctly we want to make all the arcs (edges) non-negative. For a d-link we have some value $K - c(p, \alpha)$ where c is a cost function. To ensure this we choose some big value for K. One example is $K > \max(c(p, \alpha))\ \forall p, \alpha$. This is commonly overlooked because we wish to cut the light weights, and not vice versa.

2. Approximation for the Potts Model

From now on let us write $|C|$ as the summation of the weights in the multiway cut C. We then have the sum of n-links and d-links to equal the energy function (plus a constant). In other words,

$$|C| = E(f^C) + const$$

where f^C denotes the function that assigns a label l to all its pixels. Now consider Fig. 3 as a snapshot of a much larger net of pixels. Calculating the cost of d-links out of p procedes as follows:

$$\begin{aligned}
|C| &= \sum_{p \epsilon P} \left[\sum_{L \neq f^C(p)} K - c(p, f^C(p)) \right] \\
&= \sum_{p \epsilon P} (m-1)K - \sum_{p \epsilon P} c(p, f^C(p)) \\
&= \sum_{p \epsilon P} (m-1)K + \underbrace{c(p, f^C(p)) - \sum_{l \in L} c(p, l)}\quad \text{(cost of labels)} \\
&= \sum_{p \epsilon P} c(p, f^C(p)) + const
\end{aligned}$$

We now have an approximation for the Potts model. However, we are looking for a 2-APPR to the MWC, so we can take advantage of a key notion, that of an *isolating cut*.

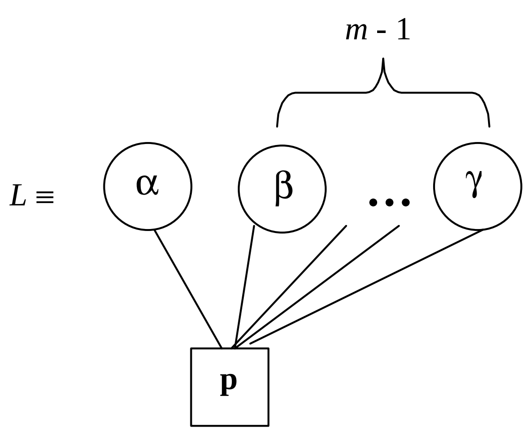


Figure 3

A method to generate such a cut can be described in 3 steps. $\forall$ terminals do:

(1) pick a terminal α as a source

(2) link all others to vertex (super-sink) t with ∞ cost arcs (i.e. can't be cut)

(3) run the MINCUT algorithm

Note that 2 things may go wrong with the above procedure.

(1) Dangling nodes (pixels) may exist. SOLUTION: Pick a [random] terminal for it.

(2^*) Overlapping of the labeled regions may occur (Fig. 4). SOLUTION: (Easy) Can't really happen. Keep reading for the real problem.

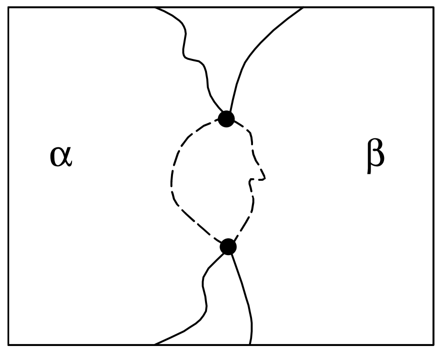


Figure 4

This *isolation heuristic* gives a 2-APPR for MWC. But $E(f^C) \leq E(f^{C^*})$ since $|C| \leq 2|C^*|$, and we see that $E(f^C) \leq 2E(f^{C^*}) + \text{const}$ given that C^* is the minimum multiway cut. That means we can't get an optimal labeling within a constant energy factor. This is actually the 2^{nd} problem, not 2^* as listed above.

3. Swap Moves

Maybe we can get a good APPR using *swap moves*. Specifically we're asking the following:

> Given a labeling F and α,β pairs, can we find an average minimization $E(f')$, with f' near f within1 $\alpha - \beta$ swap move?

A swap move is a mapping $f \mapsto f'$ if $f(p) \notin \{\alpha,\beta\} \rightarrow f'(p) = f(p)$. A special case of this is a standard Simulated Annealing move, which happens to be just a degenerate swap move. In an algorithm we ignore all but the (α,β) pair. This is when the two labels "fight it out." So to minimize f we define the procedure:

$$\text{MIN}(f) \equiv \left.\begin{array}{ll} \forall(\alpha,\beta) & \text{find cheapest } f', \\ & \underbrace{\text{if } E(f) > E(f') \text{ then } f = f'}_{\text{iteration}} \end{array}\right\} \text{cycle}$$

4. Expansion Moves

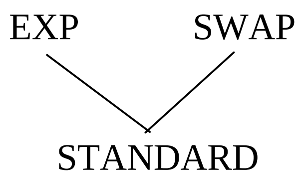
But this is not the APPR we want due to the choice of arbitrary swaps. Instead we will utilize *expansion moves*. These will lead us to a 2-APPR.

EXPANSION MOVE (α: expansion) $\equiv$ if $\forall p \quad f(p) = \alpha \rightarrow f'(p) = \alpha$

$$f'(p) \neq \alpha \rightarrow f'(p) = f(p)$$

5. Concluding Remarks

a) The 3 moves covered can be categorized as



b) These moves are powerful – with EXPANSION we can get to the global minimum $f \longmapsto f' (\longmapsto f^*)$ in m EXP moves irrespective of the huge label space.