Particle methods

CS B553
Spring 2013

Basic graph cut construction

• One non-terminal vertex per pixel
  – Each pixel has edge to s, t, and neighbors
  – Edge p-s has weight $D_p(0)$, edge p-t has weight $D_p(1)$
  – Edge (p,q) has weight $V_{pq}(0,1)$

• Run graph cuts to find a min cut
  – Label pixel p 0 if connected to t, and 1 if connected to s

• Cost of cut is the cost of the entire MRF labeling
  – So min cut means we’ve found min-cost labeling!

$$E(x_1, \ldots, x_n) = \sum_p D_p(x_p) + \sum_{p,q} V_{pq}(x_p, x_q)$$

Can this be generalized for multi-label problems?

• Not easily.
  – NP-hard for even the Potts model ([K/BVZ 01]

• Two main approaches
  1. Exact solution ([Shikawa 03]
     • Large graph, convex $V$ (arbitrary $D$)
  2. Approximate solutions ([BVZ 01]
     • Solve a binary labeling problem, repeatedly
     • Expansion move algorithm

Expansion move algorithm

Input labeling $f$

- Make green expansion move that most decreases cost
  – Then make the best blue expansion move, etc
  – Done when no $\alpha$-expansion move decreases the energy, for any label $\alpha$
  – See [BVZ 01] for details

Adapted from R. Zabih’s slide

Binary sub-problem

Input labeling

Expansion move

Binary image

Adapted from R. Zabih’s slide

Announcements

• A3 posted
  – Due Friday March 8, 11:59PM
The expansion move algorithm

1. Start with an arbitrary labeling
2. Cycle through every label A in some order
   2.1 Find the lowest cost labeling that involves an A-expansion move – this is a binary subproblem!
   2.2 Make the move if its cost is lower than current labeling
3. If cost did not decrease in the cycle, we’re done
   Otherwise, go to step 2

Adapted from R. Zabih’s slide

Move examples

The swap move algorithm

1. Start with an arbitrary labeling
2. Cycle through every label pair (A,B) in some order
   2.1 Find the lowest cost labeling within a single AB-swap
   2.2 Go there if its cost is lower than the current labeling
3. If cost did not decrease in the cycle, we’re done
   Otherwise, go to step 2

Adapted from R. Zabih’s slide

Multi-label graph cuts

- The approximate algorithm works for:
  - D of any form
  - V must satisfy a (generalized) submodularity constraint:
    \[
    V(\alpha, \alpha) + V(f(p), f(q)) \leq V(f(p), \alpha) + V(\alpha, f(q))
    \]
Graph cuts properties

- Binary graph cuts is key step of inner loop
- In each iteration of graph cuts, the total cost can’t increase
  - Converges to a solution in $O(n)$ steps
  - In practice, typically converges in just a few steps
- At convergence, the solution is a local minimum
  - And, we can prove an approximation bound: The cost of the graph cuts solution is within a factor of 2 of the cost of the exact solution!

Why does graph cuts work so well?

- It’s an iterative, hill-climbing approach, but one in which every step is searching over a huge space
  - Every step searches over $O(2^n)$ labelings!
  - Starting from an arbitrary labeling, you can get to the optimal labeling in just $k$ of these steps
- Compare this to other, more obvious hill-climbing techniques, e.g. change a single pixel at a time
  - Every step searches over just $O(1)$ labelings
  - Generally yields a weak local minimum

Graph cuts vs BP

- Graph cuts typically finds slightly lower-energy solutions
  - However, lower-energy is not necessarily better...
- More theoretical results are known for graph cuts
  - On 2 label problems, graph cuts gives exact solution
  - On multilabel problems with convex cost functions, GC gives solutions in polynomial time (but not practical in practice)
- On other multilabel problems, GC has an approx. bound
- BP is more general
  - Works on any graph structure, and any pairwise cost function
  - Can choose MAP inference or compute marginals
  - Easier to implement

Comparing techniques on stereo

- Compare techniques on cost of best solution ("energy") versus time

Ground truth vs Graph cuts vs BP

- Graph cuts typically finds slightly lower-energy solutions
  - However, lower-energy is not necessarily better...
- More theoretical results are known for graph cuts
  - On 2 label problems, graph cuts gives exact solution
  - On multilabel problems with convex cost functions, GC gives solutions in polynomial time (but not practical in practice)
- On other multilabel problems, GC has an approx. bound
- BP is more general
  - Works on any graph structure, and any pairwise cost function
  - Can choose MAP inference or compute marginals
  - Easier to implement

Adapted from R. Zabih’s slide
GrabCut

Rother et al., SIGGRAPH 2004

Application: texture synthesis

“Graphcut textures” [Kwatra et al 03]

Graphcuts video textures

“Graphcut textures” [Kwatra et al 03]

Interactive Digital Photomontage

Aseem Agarwala, Mira Dontcheva, Maneesh Agrawala, Steven Drucker, Alex Colburn, Brian Curless, David Salesin, Michael Cohen

University of Washington & Microsoft Research
Image objective

0 for any label
0 if red
∞ otherwise