Grouping and segmentation

Subhransu Maji

CMPSCI 670: Computer Vision
November 17, 2016

Overview

- Grouping and segmentation
  - Goals of segmentation
  - Clustering using k-means
  - Choice of representation
  - Two techniques:
    - Mean shift algorithm
    - Graph cuts algorithm
  - Interactive segmentation

Photoscan by Google

- Take 4 photos — stitch them together + post-processing (remove glare, crop along the boundary, remove skew)

https://www.engadget.com/2016/11/15/google-photos-photoscan-app-editing-tools/

The goals of segmentation

- Separate image into coherent “objects”

- Another way of thinking about boundary detection

Source: Lana Lazebnik
Segmentation as clustering

Depending on what we choose as the feature space, we can group pixels in different ways.

Grouping pixels based on **intensity** similarity

Feature space: intensity value (1-d)

---

Segmentation as clustering

Depending on what we choose as the feature space, we can group pixels in different ways.

Grouping pixels based on **color** similarity

Feature space: color value (3-d)

---

Segmentation as clustering

Depending on what we choose as the feature space, we can group pixels in different ways.

Grouping pixels based on **intensity** similarity

Clusters based on intensity similarity don’t have to be spatially coherent.
Depending on what we choose as the *feature space*, we can group pixels in different ways.

- **Grouping pixels based on intensity+position similarity**

  ![Intensity vs. Position Similarity](image1)

  Both regions are black, but if we also include position \((x,y)\), then we could group the two into distinct segments; way to encode both similarity & proximity.

- **Filter response (texture) similarity**

  ![Filter Bank of 24 Filters](image2)

  Feature space: filter bank responses (e.g., 24-d)

---

**Segmentation as clustering**

- Color, brightness, position alone are not enough to distinguish all regions…

- **Image segmentation example**

  ![Texture-based & Color-based Segmentation](image3)
K-means: pros and cons

Pros
- Simple, fast to compute
- Converges to local minimum of within-cluster squared error

Cons/issues
- Setting k?
- Sensitive to initial centers
- Sensitive to outliers
- Detects spherical clusters
- Assuming means can be computed

Mean shift algorithm

- The mean shift algorithm seeks modes or local maxima of density in the feature space

Feature space
(L*u*v* color values)
Mean shift

Search window
Center of mass
Mean Shift vector

Mean shift

Search window
Center of mass
Mean Shift vector

Mean shift

Search window
Center of mass
Mean Shift vector

Mean shift

Search window
Center of mass
Mean Shift vector
Mean shift

Mean Shift procedure:
For each point, repeat till convergence:
• Compute mean shift vector
• Translate the Kernel window by \( m(x) \)

Mean shift clustering

• Cluster: all data points in the attraction basin of a mode
• Attraction basin: the region for which all trajectories lead to the same mode

Mean shift clustering/segmentation

• Find features (color, gradients, texture, etc)
• Initialize windows at individual feature points
• Perform mean shift for each window until convergence
• Merge windows that end up near the same “peak” or mode
Mean shift segmentation results

http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

Images as graphs

*Image graph*
- node (vertex) for every pixel
- link between pair of pixels, \( p,q \)
- affinity weight \( w_{pq} \) for each link (edge)
  - \( w_{pq} \) measures similarity
  - similarity is inversely proportional to difference (in color and position...)
- In practice only connect nodes within a neighborhood of each pixel

Mean shift clustering results

Images as graphs

Segmentation by graph cuts

*Break graph into segments*
- Want to delete links that cross *between* segments
- Easiest to break links that have low similarity (low weight)
  - similar pixels should be in the same segments
  - dissimilar pixels should be in different segments
Cuts in a graph: Min cut

Link Cut
- set of links whose removal makes a graph disconnected
- cost of a cut:

\[ \text{cut}(A,B) = \sum_{p \in A, q \in B} w_{p,q} \]

Find minimum cut
- gives you a segmentation
- fast algorithms exist for doing this (max flow/min cut algorithms)
- faster implementations exist that exploit the grid-structure of the graph (e.g., Boykov and Jolly 2001)

Problem with minimum cut:
- Weight of cut proportional to number of edges in the cut;
- tends to produce small, isolated components.

Cuts in a graph: Normalized cut

Normalized Cut
- fix bias of Min Cut by normalizing for size of segments:

\[ N\text{cut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)} \]

assoc(A,V) = sum of weights of all edges that touch A
- ncut value is small when we get two clusters with many edges with high weights, and few edges of low weight between them
- NP-hard to compute, but approximate solution for minimizing the ncut value: generalized eigenvalue problem

Example results
Normalized cuts: pros and cons

Pros:
- Generic framework, flexible to choice of function that computes weights ("affinities") between nodes
- Does not require model of the data distribution

Cons:
- Time complexity can be high
  - Dense, highly connected graphs → many affinity computations
  - Solving eigenvalue problem
- Preference for balanced partitions

Image segmentation with priors

- Often we want to incorporate prior information
  - User input in interactive applications
  - Shape priors, e.g., we want a round object

Constrains the set of possible segmentations

“Grabcut" algorithm

Construct a color model of foreground and background

C. Rother, V. Kolmogorov, A. Blake. GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts. ACM Transactions on Graphics (SIGGRAPH’04), 2004

Gaussian mixture model (5-8 components)
(probabilistic version of k-means)
Solution using min-cut

Cost to assign to 0

Source (Label 0)

Cost of splitting nodes

Sink (Label 1)

Pairwise similarity

Cost to assign to 1

Background similarity

Foreground similarity

Moderately straightforward examples

… GrabCut completes automatically

Difficult examples

Camouflage & Low Contrast

Fine structure

Harder Case

Grabcut algorithm

Pros

• Globally optimal solution using min-cut/max-flow algorithms
• Fast algorithms exist for grid-graphs
• Works well in many cases

Cons

• Color similarity does not work when contrast is low, or when the image has fine-structures
Further thoughts and readings ..

- Chapter 5, Richard Szeliski’s book
  - [Berkeley segmentation database and benchmark](http://www.cis.upenn.edu/~jshi/GraphTutorial/)
  - Also read about the Berkeley boundary detector

- [Image segmentation via. graph cuts](http://www.cs.berkeley.edu/~malik/papers/SM-ncut.pdf)
  - Boykov and Jolly, *Interactive graph cuts for optimal boundary & region segmentation of objects in ND images*, ICCV 2001

- Normalized cuts for image segmentation (Shi and Malik)
  - Biased normalized cuts

Further thoughts and readings ..