Announcements

- Project 2
- more signup slots
- questions
- Picture taking at end of class

Today’s Readings

- Forsyth chapter 14, 16

From images to objects

What Defines an Object?
- Subjective problem, but has been well-studied
- Gestalt Laws seek to formalize this
  - proximity, similarity, continuation, closure, common fate
  - see notes by Steve Joordens, U. Toronto

Image Segmentation

We will consider different methods

Already covered:
- Intelligent Scissors (contour-based)
- Hough transform (model-based)

This week:
- K-means clustering (color-based)
- EM
- Mean-shift
- Normalized Cuts (region-based)
Image histograms

How many “orange” pixels are in this image?
• This type of question answered by looking at the histogram
• A histogram counts the number of occurrences of each color
  – Given an image \( f[x, y] \rightarrow RGB \)
  – The histogram is defined to be
  \[ H_f[c] = |\{(x, y) \mid f[x, y] = c\}| \]

How many “orange” pixels are in this image?

What do histograms look like?

How Many Modes Are There?
• Easy to see, hard to compute

Histogram-based segmentation

Goal
• Break the image into K regions (segments)
• Solve this by reducing the number of colors to K and mapping each pixel to the closest color
  – photoshop demo

Here’s what it looks like if we use two colors

Histogram-based segmentation

Goal
• Break the image into K regions (segments)
• Solve this by reducing the number of colors to K and mapping each pixel to the closest color
  – photoshop demo

Here’s what it looks like if we use two colors
Clustering

How to choose the representative colors?

- This is a clustering problem!

Objective

- Each point should be as close as possible to a cluster center
  - Minimize sum squared distance of each point to closest center

\[ \sum_{\text{clusters } i} \sum_{\text{points } p \in \text{cluster } i} ||p - c_i||^2 \]

K-means clustering

K-means clustering algorithm

1. Randomly initialize the cluster centers, \( c_1, \ldots, c_K \)
2. Given cluster centers, determine points in each cluster
   - For each point \( p \), find the closest \( c_i \). Put \( p \) into cluster \( i \)
3. Given points in each cluster, solve for \( c_i \)
   - Set \( c_i \) to be the mean of points in cluster \( i \)
4. If \( c_i \) have changed, repeat Step 2


Properties

- Will always converge to some solution
- Can be a "local minimum"
- does not always find the global minimum of objective function:

\[ \sum_{\text{clusters } i} \sum_{\text{points } p \in \text{cluster } i} ||p - c_i||^2 \]

Break it down into subproblems

Suppose I tell you the cluster centers \( c_i \)

- Q: how to determine which points to associate with each \( c_i \)?
- A: for each point \( p \), choose closest \( c_i \)

Suppose I tell you the points in each cluster

- Q: how to determine the cluster centers?
- A: choose \( c_i \) to be the mean of all points in the cluster

Probabilistic clustering

Basic questions

- what’s the probability that a point \( x \) is in cluster \( m \)?
- what’s the shape of each cluster?

K-means doesn’t answer these questions

Basic idea

- instead of treating the data as a bunch of points, assume that they are all generated by sampling a continuous function
- This function is called a generative model
  - defined by a vector of parameters \( \theta \)
**Mixture of Gaussians**

One generative model is a mixture of Gaussians (MOG):

- K Gaussian blobs with means $\mu_b$, covariance matrices $V_b$, dimension $d$
  - blob $b$ defined by:
    $$r(x|\mu_b, V_b) = \frac{1}{\sqrt{(2\pi)^d |V_b|}} e^{-\frac{1}{2} (x-\mu_b)^T V_b^{-1} (x-\mu_b)}$$
  - blob $b$ is selected with probability $\omega_b$
  - the likelihood of observing $x$ is a weighted mixture of Gaussians
    $$P(x|\theta) = \sum_{b=1}^{K} \omega_b r(x|\mu_b, V_b)$$
  - where $\theta = \{\mu_1, ..., \mu_K, V_1, ..., V_K\}$

**Expectation maximization (EM)**

Goal:

- find blob parameters $\theta$ that maximize the likelihood function:
  $$P(\text{data}|\theta) = \prod_x P(x|\theta)$$

Approach:

1. E step: given current guess of blobs, compute ownership of each point
2. M step: given ownership probabilities, update blobs to maximize likelihood function
3. repeat until convergence

**EM details**

**E-step**

- compute probability that point $x$ is in blob $i$, given current guess of $\theta$
  $$P(b|x, \mu_b, V_b) = \frac{\omega_b r(x|\mu_b, V_b)}{\sum_{i=1}^{K} \omega_i r(x|\mu_i, V_i)}$$

**M-step**

- compute probability that blob $b$ is selected
  $$\omega_b^{\text{new}} = \frac{1}{N} \sum_{x \in \text{data}} P(b|x, \mu_b, V_b)$$
  - N data points
- mean of blob $b$
  $$\mu_b^{\text{new}} = \frac{\sum_{x \in \text{data}} x P(b|x, \mu_b, V_b)}{\sum_{x \in \text{data}} P(b|x, \mu_b, V_b)}$$
- covariance of blob $b$
  $$V_b^{\text{new}} = \frac{\sum_{x \in \text{data}} (x - \mu_b^{\text{new}})(x - \mu_b^{\text{new}})^T P(b|x, \mu_b, V_b)}{\sum_{x \in \text{data}} P(b|x, \mu_b, V_b)}$$

**EM demos**

Applications of EM

Turns out this is useful for all sorts of problems

- any clustering problem
- any model estimation problem
- missing data problems
- finding outliers
- segmentation problems
  - segmentation based on color
  - segmentation based on motion
  - foreground/background separation
- ...

Problems with EM

Local minima

Need to know number of segments

Need to choose generative model

Finding Modes in a Histogram

How Many Modes Are There?

- Easy to see, hard to compute

Mean Shift [Comaniciu & Meer]

Iterative Mode Search

1. Initialize random seed, and window W
2. Calculate center of gravity (the "mean") of W: \[ \mu = \frac{\sum_{x \in W} x H(x)}{\sum_{x \in W} H(x)} \]
3. Translate the search window to the mean
4. Repeat Step 2 until convergence
**Mean-Shift**

*Approach*

- Initialize a window around each point
- See where it shifts—this determines which segment it’s in
- Multiple points will shift to the same segment

**Mean-shift for image segmentation**

*Useful to take into account spatial information*

- Instead of (R, G, B), run in (R, G, B, x, y) space
- D. Comaniciu, P. Meer, Mean-shift analysis and applications, 7th International Conference on Computer Vision, Kerkyra, Greece, September 1999, 1197-1203.


**Region-based segmentation**

Color histograms don’t take into account spatial info

- Gestalt laws point out importance of spatial grouping
  - proximity, similarity, continuation, closure, common fate
- Suggests that regions are important

**Images as graphs**

*Fully-connected graph*

- node for every pixel
- link between every pair of pixels, p.q
- cost $c_{pq}$ for each link
  - $c_{pq}$ measures similarity
  - similarity is inversely proportional to difference in color and position
  - this is different than the costs for intelligent scissors
Segmentation by Graph Cuts

Break Graph into Segments
- Delete links that cross between segments
- Easiest to break links that have high cost
  - similar pixels should be in the same segments
  - dissimilar pixels should be in different segments

Cuts in a graph

Link Cut
- set of links whose removal makes a graph disconnected
- cost of a cut:
  \[ \text{cut}(A, B) = \sum_{e \in A \cup B} c_{AB} \]

Find minimum cut
- gives you a segmentation
- fast algorithms exist for doing this

But min cut is not always the best cut...

Better cut

Cuts in a graph

Normalized Cut
- a cut penalizes large segments
- fix by normalizing for size of segments
  \[ N\text{cut}(A, B) = \frac{\text{cut}(A, B)}{\text{volume}(A)} + \frac{\text{cut}(A, B)}{\text{volume}(B)} \]
- volume(A) = sum of costs of all edges that touch A
Interpretation as a Dynamical System

Treat the links as springs and shake the system
• elasticity proportional to cost
• vibration “modes” correspond to segments

Color Image Segmentation

Normalize Cut in Matrix Form

\[ W \text{ is the cost matrix : } W(i, j) = c_{i,j}; \]
\[ D \text{ is the sum of costs from node } i : D(i, i) = \sum_j W(i, j); \]
\[ D(i, j) = 0 \]

Can write normalized cut as:
\[ Ncut(A, B) = \frac{y^T(D - W)y}{y^TDy}, \text{ with } y \in \{1, -1\}, \text{ and } y^T D y = 0. \]
• Solution given by “generalized” eigenvalue problem:
\[ (D - W)y = \lambda Dy \]
• Solved by converting to standard eigenvalue problem:
\[ D^{-1}(D - W)D^{-1}z = \lambda z, \text{ where } z = D^iy \]
• optimal solution corresponds to second smallest eigenvector
• for more details, see
  Computer Vision and Pattern Recognition (CVPR), 1997
Cleaning up the result

Problem:
- Histogram-based segmentation can produce messy regions
  - segments do not have to be connected
  - may contain holes

How can these be fixed?

Dilation operator: \(G = H \oplus F\)

Dilation: does \(H\) "overlap" \(F\) around \([x,y]\)?
- \(G(x,y) = 1\) if \(H(u,v)\) and \(F(x+u-1,y+v-1)\) are both 1 somewhere
- 0 otherwise

\(F[x,y]\)

\(H[u,v]\)

Erosion: is \(H\) "contained in" \(F\) around \([x,y]\)?
- \(G(x,y) = 1\) if \(F(x+u-1,y+v-1)\) is 1 everywhere that \(H(u,v)\) is 1
- 0 otherwise

\(F[x,y]\)

\(H[u,v]\)

\(G = H \ominus F\)

Dilation demo

Photoshop demo

Erosion demo

http://www.cs.bris.ac.uk/~majid/mengine/morph.html
**Erosion operator**

Demo

- [http://www.cs.bris.ac.uk/~majid/mengine/morph.html](http://www.cs.bris.ac.uk/~majid/mengine/morph.html)

**Nested dilations and erosions**

What does this operation do?

\[ G = H \ominus (H \oplus F) \]

- this is called a **closing** operation

**Nested dilations and erosions**

What does this operation do?

\[ G = H \ominus (H \oplus F) \]

- this is called a **closing** operation

You can clean up binary pictures by applying combinations of dilations and erosions.

Dilations, erosions, opening, and closing operations are known as **morphological operations**.

- see [http://www.dai.ed.ac.uk/HIPR2/morops.htm](http://www.dai.ed.ac.uk/HIPR2/morops.htm)