Region Segmentation
Readings: Chapter 10: 10.1
Additional Materials Provided

• K-means Clustering (text)
• EM Clustering (paper)
• Graph Partitioning (text)
• Mean-Shift Clustering (paper)
Image Segmentation

Image segmentation is the operation of partitioning an image into a collection of connected sets of pixels.

1. into regions, which usually cover the image

2. into linear structures, such as
   - line segments
   - curve segments

3. into 2D shapes, such as
   - circles
   - ellipses
   - ribbons (long, symmetric regions)
Example: Regions
Main Methods of Region Segmentation

1. Region Growing
2. Split and Merge
3. Clustering
Clustering

• There are $K$ clusters $C_1, \ldots, C_K$ with means $m_1, \ldots, m_K$.

• The least-squares error is defined as

$$D = \sum_{k=1}^{K} \sum_{x_i \in C_k} \| x_i - m_k \|^2.$$

• Out of all possible partitions into $K$ clusters, choose the one that minimizes $D$.

Why don’t we just do this?
If we could, would we get meaningful objects?
K-Means Clustering

Form K-means clusters from a set of n-dimensional vectors

1. Set ic (iteration count) to 1

2. Choose randomly a set of K means $m_1(1), \ldots, m_K(1)$.

3. For each vector $x_i$ compute $D(x_i, m_k(ic))$, $k=1,\ldots,K$ and assign $x_i$ to the cluster $C_j$ with nearest mean.

4. Increment ic by 1, update the means to get $m_1(ic),\ldots,m_K(ic)$.

5. Repeat steps 3 and 4 until $C_k(ic) = C_k(ic+1)$ for all $k$. 
K-Means Example 1
K-Means Example 2
K-Means Example 3

1. Select an image:  
2. Select a processor:  
3. Click

Options:
Init Method 0

Process done!

640*480 (607,118): RGB(20,22,1)

(228,26): RGB(255,170,0)
K-means Variants

• Different ways to initialize the means
• Different stopping criteria
• Dynamic methods for determining the right number of clusters (K) for a given image

• The EM Algorithm: a probabilistic formulation of K-means
K-Means

- **Boot Step:**
  - Initialize $K$ clusters: $C_1, \ldots, C_K$
    
    Each cluster is represented by its mean $m_j$

- **Iteration Step:**
  - Estimate the cluster for each data point
    
    $x_i \mapsto C(x_i)$
  - Re-estimate the cluster parameters

\[ m_j = \text{mean}\{x_i \mid x_i \in C_j\} \]
K-Means Example
K-Means Example

Where do the red points belong?
<table>
<thead>
<tr>
<th>Cluster Representation</th>
<th>K-means</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Representation</td>
<td>mean</td>
<td>mean, variance, and weight</td>
</tr>
<tr>
<td>Initialization</td>
<td>randomly select K means</td>
<td>initialize K Gaussian distributions</td>
</tr>
<tr>
<td>Expectation</td>
<td>assign each point to closest mean</td>
<td>soft-assign each point to each distribution</td>
</tr>
<tr>
<td>Maximization</td>
<td>compute means of current clusters</td>
<td>compute new params of each distribution</td>
</tr>
</tbody>
</table>
Notation

$N(\mu, \sigma)$ is a 1D normal (Gaussian) distribution with mean $\mu$ and standard deviation $\sigma$ (so the variance is $\sigma^2$).
$N(\mu, \Sigma)$ is a multivariate Gaussian distribution with mean $\mu$ and covariance matrix $\Sigma$.

**What is a covariance matrix?**

<table>
<thead>
<tr>
<th></th>
<th>R</th>
<th>G</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>$\sigma_R^2$</td>
<td>$\sigma_{RG}$</td>
<td>$\sigma_{RB}$</td>
</tr>
<tr>
<td>G</td>
<td>$\sigma_{GR}$</td>
<td>$\sigma_G^2$</td>
<td>$\sigma_{GB}$</td>
</tr>
<tr>
<td>B</td>
<td>$\sigma_{BR}$</td>
<td>$\sigma_{BG}$</td>
<td>$\sigma_B^2$</td>
</tr>
</tbody>
</table>

variance($X$): $\sigma_X^2 = \sum (x_i - \mu)^2 (1/N)$

cov($X,Y$) = $\sum (x_i - \mu_x)(y_i - \mu_y) (1/N)$
1. Suppose we have a set of clusters: $C_1, C_2, \ldots, C_K$ over a set of data points $X = \{x_1, x_2, \ldots, x_N\}$.

   - $P(C_j)$ is the probability or weight of cluster $C_j$.
   - $P(C_j | x_i)$ is the probability of cluster $C_j$ given point $x_i$.
   - $P(x_i | C_j)$ is the probability of point $x_i$ belonging to cluster $C_j$.

2. Suppose that a cluster $C_j$ is represented by a Gaussian distribution $N(\mu_j, \sigma_j)$. Then for any point $x_i$:

$$P(x_i | C_j) = \frac{1}{\sqrt{2\pi \sigma_j}} e^{-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}}$$
EM: Expectation-Maximization

• **Boot Step:**
  – Initialize $K$ clusters: $C_1, \ldots, C_K$
  $$(\mu_j, \Sigma_j) \text{ and } P(C_j) \text{ for each cluster } j.$$  

• **Iteration Step:**
  – Estimate the cluster of each data point $p(C_j \mid x_i)$
  – Re-estimate the cluster parameters $$(\mu_j, \Sigma_j), p(C_j) \text{ For each cluster } j$$
1-D EM with Gaussian Distributions

• Each cluster $C_j$ is represented by a Gaussian distribution $N(\mu_j, \sigma_j)$.
• Initialization: For each cluster $C_j$ initialize its mean $\mu_j$, variance $\sigma_j^2$, and weight $\alpha_j$.

$N(\mu_1, \sigma_1)$
$\alpha_1 = P(C_1)$

$N(\mu_2, \sigma_2)$
$\alpha_2 = P(C_2)$

$N(\mu_3, \sigma_3)$
$\alpha_3 = P(C_3)$
Expectation

• For each point $x_i$ and each cluster $C_j$ compute $P(C_j | x_i)$.

• $P(C_j | x_i) = \frac{P(x_i | C_j) \cdot P(C_j)}{P(x_i)}$

• $P(x_i) = \sum_j P(x_i | C_j) \cdot P(C_j)$

• Where do we get $P(x_i | C_j)$ and $P(C_j)$?
1. Use the pdf for a normal distribution:

\[
P(x_i \mid C_j) = \frac{1}{\sqrt{2\pi \sigma_j}} e^{-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}}
\]

2. Use \( \alpha_j = P(C_j) \) from the current parameters of cluster \( C_j \).
Maximization

- Having computed $P(C_j | x_i)$ for each point $x_i$ and each cluster $C_j$, use them to compute new mean, variance, and weight for each cluster.

\[
\mu_j = \frac{\sum_i p(C_j | x_i) \cdot x_i}{\sum_i p(C_j | x_i)}
\]

\[
\sigma_j^2 = \sum_j \frac{\sum_i p(C_j | x_i) \cdot (x_i - \mu_j) \cdot (x_i - \mu_j)^T}{\sum_i p(C_j | x_i)}
\]

\[
p(C_j) = \frac{\sum_i p(C_j | x_i)}{N}
\]
Multi-Dimensional Expectation Step for Color Image Segmentation

\[
p(C_j | x_i) = \frac{p(x_i | C_j) \cdot p(C_j)}{p(x_i)} = \frac{\sum_j p(x_i | C_j) \cdot p(C_j)}{p(x_i)}
\]
Multi-dimensional Maximization Step for Color Image Segmentation

\[ \mu_j = \frac{\sum_i p(C_j | x_i) \cdot x_i}{\sum_i p(C_j | x_i)} \]

\[ \Sigma_j = \frac{\sum_i p(C_j | x_i) \cdot (x_i - \mu_j) \cdot (x_i - \mu_j)^T}{\sum_i p(C_j | x_i)} \]

\[ p(C_j) = \frac{\sum_i p(C_j | x_i)}{N} \]

Input (Known)
\[ x_1 = \{r_1, g_1, b_1\} \]
\[ x_2 = \{r_2, g_2, b_2\} \]
\[ \ldots \]
\[ x_i = \{r_i, g_i, b_i\} \]
\[ \ldots \]

Input (Estimation)

Classification Results
\[ p(C_1 | x_1) \]
\[ p(C_j | x_2) \]
\[ \ldots \]
\[ p(C_j | x_i) \]
\[ \ldots \]

Output
Cluster Parameters
\((\mu_1, \Sigma_1), p(C_1)\) for \(C_1\)
\((\mu_2, \Sigma_2), p(C_2)\) for \(C_2\)
\[ \ldots \]
\((\mu_k, \Sigma_k), p(C_k)\) for \(C_k\)
Full EM Algorithm
Multi-Dimensional

• **Boot Step:**
  – Initialize $K$ clusters: $C_1, \ldots, C_K$

  $(\mu_j, \Sigma_j)$ and $P(C_j)$ for each cluster $j$.

• **Iteration Step:**
  – Expectation Step

  $$p(C_j \mid x_i) = \frac{p(x_i \mid C_j) \cdot p(C_j)}{p(x_i)} = \frac{p(x_i \mid C_j) \cdot p(C_j)}{\sum_j p(x_i \mid C_j) \cdot p(C_j)}$$

  – Maximization Step

  $$\mu_j = \frac{\sum_i p(C_j \mid x_i) \cdot x_i}{\sum_i p(C_j \mid x_i)}$$

  $$\Sigma_j = \frac{\sum_i p(C_j \mid x_i) \cdot (x_i - \mu_j) \cdot (x_i - \mu_j)^T}{\sum_i p(C_j \mid x_i)}$$

  $$p(C_j) = \frac{\sum_i p(C_j \mid x_i)}{N}$$
Visualizing EM Clusters

ellipses show one, two, and three standard deviations

http://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm
EM Applications

• Blobworld: Image segmentation using Expectation-Maximization and its application to image querying

• Yi’s Generative/Discriminative Learning of object classes in color images
Blobworld: Sample Results
Jianbo Shi’s Graph-Partitioning

- An image is represented by a graph whose nodes are pixels or small groups of pixels.

- The goal is to partition the vertices into disjoint sets so that the similarity within each set is high and across different sets is low.
Minimal Cuts

• Let $G = (V,E)$ be a graph. Each edge $(u,v)$ has a weight $w(u,v)$ that represents the similarity between $u$ and $v$.

• Graph $G$ can be broken into 2 disjoint graphs with node sets $A$ and $B$ by removing edges that connect these sets.

• Let $\text{cut}(A,B) = \sum_{u \in A, v \in B} w(u,v)$.

• One way to segment $G$ is to find the minimal cut.
Cut(A,B)

cut(A,B) = \sum_{u \in A, v \in B} w(u,v)
Minimal cut favors cutting off small node groups, so Shi proposed the **normalized cut**.

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

\[
\text{asso}(A,V) = \sum_{u \in A, t \in V} w(u,t)
\]

How much is A connected to the graph as a whole.
Example Normalized Cut

\[ \text{Ncut}(A,B) = \frac{3}{21} + \frac{3}{16} \]
Shi turned graph cuts into an eigenvector/eigenvalue problem.

- Set up a weighted graph \( G = (V, E) \)
  - \( V \) is the set of \( (N) \) pixels
  - \( E \) is a set of weighted edges (weight \( w_{ij} \) gives the similarity between nodes \( i \) and \( j \))
  - Length \( N \) vector \( d \): \( d_i \) is the sum of the weights from node \( i \) to all other nodes
  - \( N \times N \) matrix \( D \): \( D \) is a diagonal matrix with \( d \) on its diagonal
  - \( N \times N \) symmetric matrix \( W \): \( W_{ij} = w_{ij} \)
• Let $x$ be a characteristic vector of a set $A$ of nodes
  – $x_i = 1$ if node $i$ is in a set $A$
  – $x_i = -1$ otherwise

• Let $y$ be a continuous approximation to $x$

\[
y = (1 + x) - \frac{\sum_{x_i > 0} d_i}{\sum_{x_i < 0} d_i} (1 - x).
\]

• Solve the system of equations

\[(D - W) y = \lambda \ D \ y\]

for the eigenvectors $y$ and eigenvalues $\lambda$

• Use the eigenvector $y$ with second smallest eigenvalue to bipartition the graph ($y \Rightarrow x \Rightarrow A$)

• If further subdivision is merited, repeat recursively
How Shi used the procedure

Shi defined the edge weights \( w(i,j) \) by

\[
 w(i,j) = e^{-\|F(i)-F(j)\|_2 / \sigma} \left\{ \begin{array}{ll}
 e^{-\|X(i)-X(j)\|_2 / \sigma} & \text{if } \|X(i)-X(j)\|_2 < r \\
 0 & \text{otherwise}
\end{array} \right.
\]

where \( X(i) \) is the spatial location of node \( i \)
\( F(i) \) is the feature vector for node \( I \)
which can be intensity, color, texture, motion…

The formula is set up so that \( w(i,j) \) is 0 for nodes that are too far apart.
Examples of Shi Clustering

See Shi’s Web Page
http://www.cis.upenn.edu/~jshi/
Problems with Graph Cuts

- Need to know when to stop
- Can be slow.

Problems with EM

- Local minima
- Need to know number of segments
- Need to choose generative model
Mean-Shift Clustering

- Simple, like K-means
- But you don’t have to select K
- Statistical method
- Guaranteed to converge to a fixed number of clusters.
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 $W$: $\sum_{x \in W} xH(x)$
  3. Translate the search window to the mean
  4. Repeat Step 2 until convergence
Numeric Example
Must Use Normalized Histogram!

window \( W \) centered at 12

\[
\sum x N(x) = 10(5/15)+11(4/15)+12(3/15)+13(2/15)+14(1/15) \\
= 11.33
\]
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 trajectories
Segmentation Algorithm

• First run the mean shift procedure for each data point x and store its convergence point z.

• Link together all the z’s that are closer than .5 from each other to form clusters

• Assign each point to its cluster

• Eliminate small regions
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
References

