Lecture:
k-means & mean-shift clustering

Juan Carlos Niebles and Ranjay Krishna
Stanford Vision and Learning Lab
Recap: Image Segmentation

- Goal: identify groups of pixels that go together
Recap: Gestalt Theory

• Gestalt: whole or group
  – Whole is greater than sum of its parts
  – Relationships among parts can yield new properties/features

• Psychologists identified series of factors that predispose set of elements to be grouped (by human visual system)

“I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have "327"? No. I have sky, house, and trees.”

Max Wertheimer
(1880-1943)

Untersuchungen zur Lehre von der Gestalt,
Psychologische Forschung, Vol. 4, pp. 301-350, 1923
http://psy.ed.asu.edu/~classics/Wertheimer/Forms/forms.htm
Recap: Gestalt Factors

- These factors make intuitive sense, but are very difficult to translate into algorithms.
What will we learn today?

- K-means clustering
- Mean-shift clustering

Reading: [FP] Chapters: 14.2, 14.4

What will we learn today?

• K-means clustering
• Mean-shift clustering

Reading: [FP] Chapters: 14.2, 14.4

Image Segmentation: Toy Example

- These intensities define the three groups.
- We could label every pixel in the image according to which of these primary intensities it is.
  - i.e., segment the image based on the intensity feature.
- What if the image isn’t quite so simple?
Input image

Input image

Slide credit: Kristen Grauman
Now how to determine the three main intensities that define our groups?

We need to cluster.

Slide credit: Kristen Grauman
Goal: choose three “centers” as the representative intensities, and label every pixel according to which of these centers it is nearest to.

Best cluster centers are those that minimize Sum of Square Distance (SSD) between all points and their nearest cluster center $c_i$:

$$SSD = \sum_{cluster \ i} \sum_{x \in cluster \ i} (x - c_i)^2$$
Clustering for Summarization

Goal: cluster to minimize variance in data given clusters

– Preserve information

\[ c^*, \delta^* = \arg \min_{c, \delta} \frac{1}{N} \sum_{j} \sum_{i} \delta_{ij} (c_i - x_j)^2 \]

Whether \( x_j \) is assigned to \( c_i \)
Clustering

• With this objective, it is a “chicken and egg” problem:
  – If we knew the \textit{cluster centers}, we could allocate points to groups by assigning each to its closest center.
  
  
  – If we knew the \textit{group memberships}, we could get the centers by computing the mean per group.
K-means clustering

1. Initialize \((t = 0)\): cluster centers \(c_1, \ldots, c_K\)

2. Compute \(\delta^t\): assign each point to the closest center
   - \(\delta^t\) denotes the set of assignment for each \(x_j\) to cluster \(c_i\) at iteration \(t\)
     \[
     \delta^t = \underset{\delta}{\text{argmin}} \frac{1}{N} \sum_j \sum_i \delta_{ij}^t (c_i^{t-1} - x_j)^2
     \]

1. Computer \(c^t\): update cluster centers as the mean of the points
   \[
   c^t = \underset{c}{\text{argmin}} \frac{1}{N} \sum_j \sum_i \delta_{ij}^t (c_i^{t-1} - x_j)^2
   \]

1. Update \(t = t + 1\), Repeat Step 2-3 till stopped
K-means clustering

1. Initialize \((t = 0)\): cluster centers \(c_1, \ldots, c_K\)
   - Commonly used: random initialization
   - Or greedily choose \(K\) to minimize residual

2. Compute \(\delta_t\): assign each point to the closest center
   - Typical distance measure:
     - Euclidean \(sim(x, x') = x^T x'\)
     - Cosine \(sim(x, x') = x^T x'/(\|x\| \cdot \|x'\|)\)
     - Others

1. Compute \(c^t\): update cluster centers as the mean of the points

\[
c^t = \arg\min_c \frac{1}{N} \sum_{j} \sum_{i} \delta_{ij} (c^t_{i-1} - x_j)^2
\]

2. Update \(t = t + 1\), Repeat Step 2-3 till stopped
   - \(C^t\) doesn’t change anymore.
K-means clustering

1. Initialize Cluster Centers
2. Assign Points to Clusters
3. Re-compute Means

Repeat (2) and (3)

• Java demo:
  
  http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html
K-means clustering

- Converges to a *local minimum* solution
  - Initialize multiple runs

- Better fit for spherical data

- Need to pick $K$ (# of clusters)
Segmentation as Clustering

Original image

2 clusters

3 clusters
K-Means++

- Can we prevent arbitrarily bad local minima?

1. Randomly choose first center.
2. Pick new center with prob. proportional to \((x - c_i)^2\)
   - (Contribution of \(x\) to total error)
3. Repeat until \(K\) centers.

- Expected error = \(O(\log K)\) * optimal

Arthur & Vassilvitskii 2007
Feature Space

• 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 (1D)

Slide credit: Kristen Grauman
Feature Space

- 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 (3D)

Slide credit: Kristen Grauman
Feature Space

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

- Grouping pixels based on *texture* similarity

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

Slide credit: Kristen Grauman
Smoothing Out Cluster Assignments

• Assigning a cluster label per pixel may yield outliers:
  - Original
  - Labeled by cluster center’s intensity

• How can we ensure they are spatially smooth?

Slide credit: Kristen Grauman
Segmentation as Clustering

• Depending on what we choose as the \textit{feature space}, we can group pixels in different ways.

• Grouping pixels based on \textit{intensity+position} similarity

⇒ Way to encode both \textit{similarity} and \textit{proximity}.
K-Means Clustering Results

• K-means clustering based on intensity or color is essentially vector quantization of the image attributes
  – Clusters don’t have to be spatially coherent
K-Means Clustering Results

• K-means clustering based on intensity or color is essentially vector quantization of the image attributes
  – Clusters don’t have to be spatially coherent

• Clustering based on (r,g,b,x,y) values enforces more spatial coherence
How to evaluate clusters?

- Generative
  - How well are points reconstructed from the clusters?

- Discriminative
  - How well do the clusters correspond to labels?
    - Can we correctly classify which pixels belong to the panda?
  - Note: unsupervised clustering does not aim to be discriminative as we don’t have the labels.
How to choose the number of clusters?

Try different numbers of clusters in a validation set and look at performance.

We can plot the objective function values for $k$ equals 1 to 6...

The abrupt change at $k = 2$, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as “knee finding” or “elbow finding”.

![Graph showing the objective function values for different numbers of clusters.](image)
K-Means pros and cons

- **Pros**
  - Finds cluster centers that minimize conditional variance (good representation of data)
  - Simple and fast, Easy to implement

- **Cons**
  - Need to choose K
  - Sensitive to outliers
  - Prone to local minima
  - All clusters have the same parameters (e.g., distance measure is non-adaptive)
  - *Can be slow: each iteration is $O(KN^d)$ for $N$ $d$-dimensional points

- **Usage**
  - Unsupervised clustering
  - Rarely used for pixel segmentation
What will we learn today?

- K-means clustering
- Mean-shift clustering

Reading: [FP] Chapters: 14.2, 14.4
Mean-Shift Segmentation

- An advanced and versatile technique for clustering-based segmentation

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

Mean-Shift Algorithm

- Iterative Mode Search
  1. Initialize random seed, and window W
  2. Calculate center of gravity (the "mean") of W: $\sum_{x \in W} x H(x)$
  3. Shift the search window to the mean
  4. Repeat Step 2 until convergence
Mean-Shift

Region of interest
Center of mass
Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean-Shift

Region of interest

Center of mass

Mean Shift vector
Mean-Shift

Region of interest
Center of mass

Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean-Shift

Region of interest
Center of mass

Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean-Shift

Region of interest
Center of mass

Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean-Shift

Region of interest
Center of mass
Mean Shift vector

Slide by Y. Ukrainitz & B. Sarel
Mean-Shift

Region of interest

Center of mass

Slide by Y. Ukrainitz & B. Sarel
Real Modality Analysis

Tessellate the space with windows

Run the procedure in parallel
The blue data points were traversed by the windows towards the mode.
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 pixel locations
- 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
More Results
More Results
Problem: Computational Complexity

- Need to shift many windows...
- Many computations will be redundant.
1. Assign all points within radius $r$ of end point to the mode.
2. Assign all points within radius $r/c$ of the search path to the mode $\rightarrow$ reduce the number of data points to search.
Technical Details

Given $n$ data points $\mathbf{x}_i \in \mathbb{R}^d$, the multivariate kernel density estimate using a radially symmetric kernel \(^1\) (e.g., Epanechnikov and Gaussian kernels), $K(\mathbf{x})$, is given by,

$$\hat{f}_K = \frac{1}{nh^d} \sum_{i=1}^{n} K \left( \frac{\mathbf{x} - \mathbf{x}_i}{h} \right), \quad (1)$$

where $h$ (termed the bandwidth parameter) defines the radius of kernel. The radially symmetric kernel is defined as,

$$K(\mathbf{x}) = c_k k(\|\mathbf{x}\|^2), \quad (2)$$

where $c_k$ represents a normalization constant.

Comaniciu & Meer, 2002
Technical Details

\[ \nabla \hat{f}(x) = \frac{2c_{k,d}}{nh^{d+2}} \left[ \sum_{i=1}^{n} g \left( \frac{\|x - x_i\|}{h} \right) \right] \left[ \sum_{i=1}^{n} x_i g \left( \frac{\|x - x_i\|}{h} \right) \right] - x, \tag{3} \]

where \( g(x) = -k'(x) \) denotes the derivative of the selected kernel profile.

- Term1: this is proportional to the density estimate at \( x \) (similar to equation 1 from the previous slide).
- Term2: this is the mean-shift vector that points towards the direction of maximum density.

Comaniciu & Meer, 2002
Finally, the mean shift procedure from a given point $x_t$ is:

1. Computer the mean shirt vector $m$:

\[
\begin{bmatrix}
\sum_{i=1}^{n} x_i g \left( \frac{\| x - x_i \|}{h} \right) \\
\sum_{i=1}^{n} g \left( \frac{\| x - x_i \|}{h} \right)
\end{bmatrix}
\frac{1}{n} \left[ \sum_{i=1}^{n} x_i g \left( \frac{\| x - x_i \|}{h} \right) - x \right]
\]

2. Translate the density window:

\[
x_{i}^{t+1} = x_{i}^{t} + m(x_{i}^{t}).
\]

3. Iterate steps 1 and 2 until convergence.

\[
\nabla f(x_i) = 0.
\]
Summary Mean-Shift

• **Pros**
  – General, application-independent tool
  – Model-free, does not assume any prior shape (spherical, elliptical, etc.) on data clusters
  – Just a single parameter (window size $h$)
    • $h$ has a physical meaning (unlike k-means)
  – Finds variable number of modes
  – Robust to outliers

• **Cons**
  – Output depends on window size
  – Window size (bandwidth) selection is not trivial
  – Computationally (relatively) expensive ($\sim 2s/image$)
  – Does not scale well with dimension of feature space
What will we have learned today

- K-means clustering
- Mean-shift clustering

Reading: [FP] Chapters: 14.2, 14.4