Cluster-based Segmentation
基于聚类的图像分割

WangRuchen

CVBIOUC
http://vision.ouc.edu.cn/~zhenghaiyong

May 19, 2015

Contents

Clustering Analysis

Clustering is unsupervised classification: no predefined classes.

Cluster: a collection of data objects

- Similar to one another within the same cluster
- Dissimilar to the objects in other clusters

Image Segmentation with Clustering

Idea:

Cluster similar pixel features together.

How to segment images by clustering?

Image Segmentation with Clustering

Image $\Longrightarrow$ Feature space

Image Segmentation with Clustering

Image $\Longrightarrow$ Feature space

Feature space:(R,G,B),(R,G,B,X,Y),(L,U,V)$\cdots$

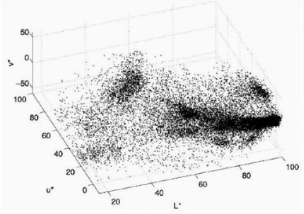
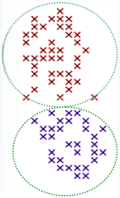


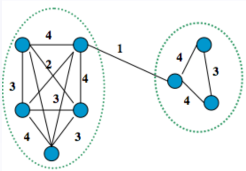
Image Segmentation with Clustering

- Vector Clustering



Each point has a vector.

- Graph Clustering



Each vertex is connected to others by edges.

Image Segmentation with Clustering

Image $\Longrightarrow$ Feature space

Feature space:(R,G,B),(R,G,B,X,Y),(L,U,V)$\cdots$

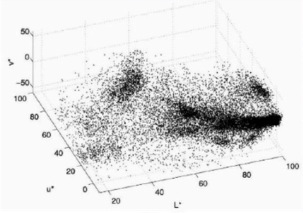


Image Segmentation with Clustering

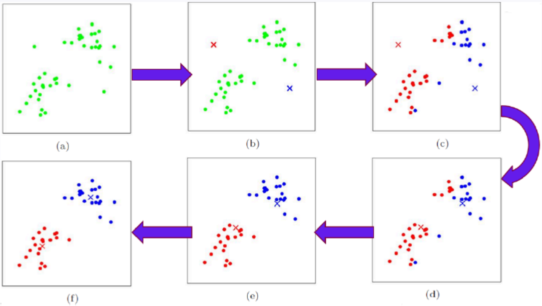
Techniques:

- K-means
- Mean Shift

K-means

Idea:

1. Randomly initialize the K cluster centers.
2. For each point, find the closest cluster centers. Put the point into the cluster.
3. Change the cluster centers.



K-means

Algorithm:

1. Choose randomly K-means $m_1, \ldots, m_k$.
2. 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.
3. Update the means to get $m_1(ic), \ldots, m_K(ic)$.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

4. Repeat steps 2 and 3 until $C_k(ic) = C_k(ic+1)$ for all k.

K-means

Pros:

- Simple and fast
- Converges to a local minimum of the error function

Cons:

- Sensitive to initialization
- Need to pick K

K-means++

Algorithm:

1. Take one center c_1, chosen uniformly at random from X.
2. Take a new center c_i, choosing $x \in X$ with probability $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$
3. Repeat Step 1, until we have taken k centers altogether.

$D(x)$ denote the shortest distance from a data point to the closest center.

K-means

Pros:

- Simple and fast
- Converges to a local minimum of the error function

Cons:

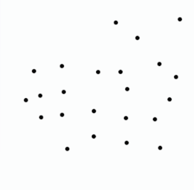
- Sensitive to initialization
- Need to pick K

Mean Shift[1]

An advanced and versatile technique for clustering-based segmentation.

Idea:

- Find the clustering center.



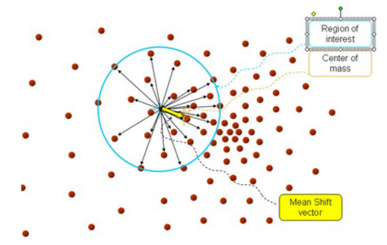
[1] D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach toward Feature Space Analysis”, PAMI, 2002.

What and How?

What and How?

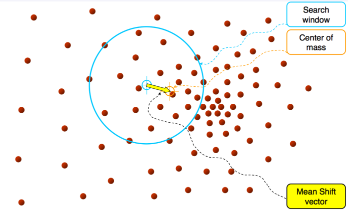
Center is maximum points of probability density function and gradient direction.

Mean Shift Vector

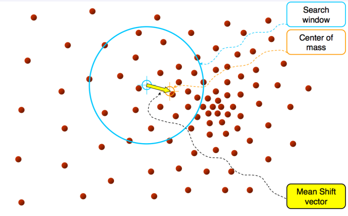


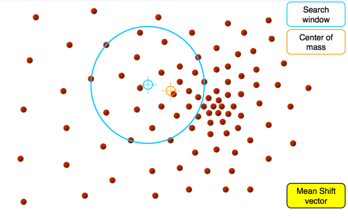
$$M_h = \frac{1}{K} \sum_{x_i S_k} (x_i - x)$$

What is Mean Shift

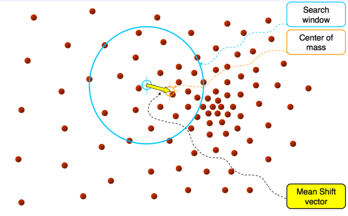


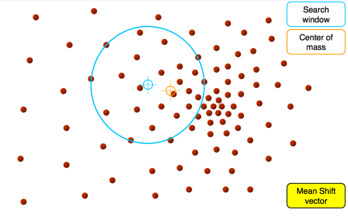
What is Mean Shift

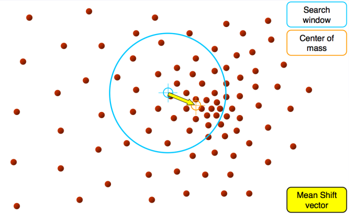




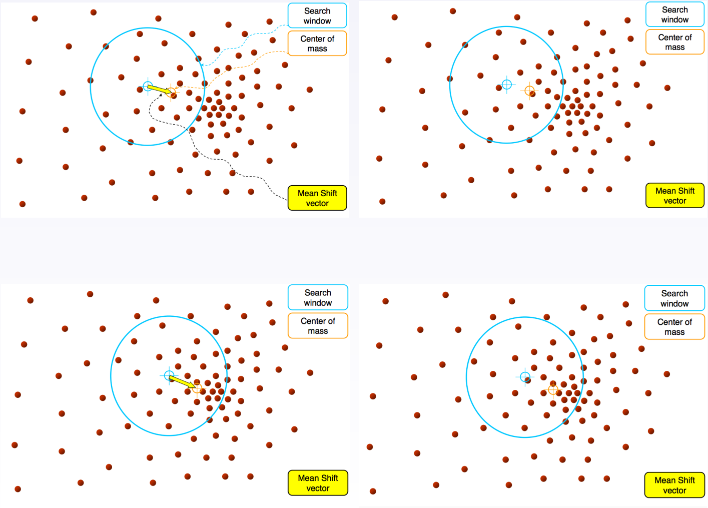
What is Mean Shift



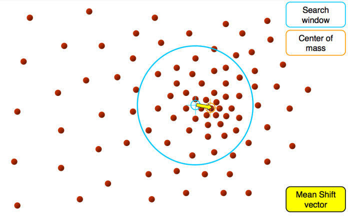




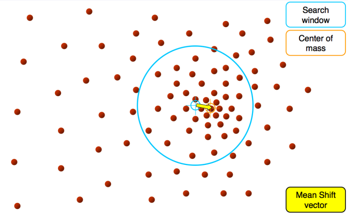
What is Mean Shift

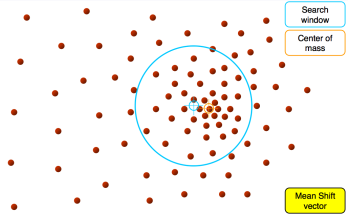


What is Mean Shift

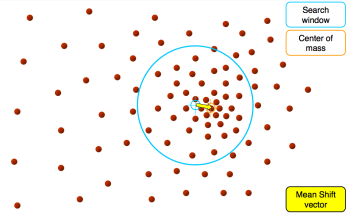


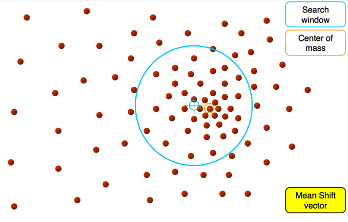
What is Mean Shift

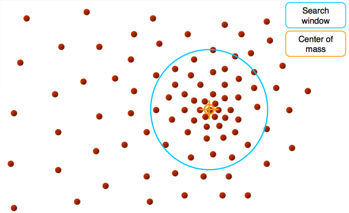




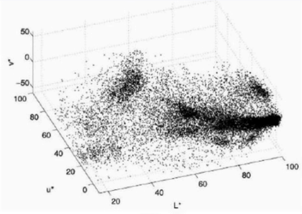
What is Mean Shift



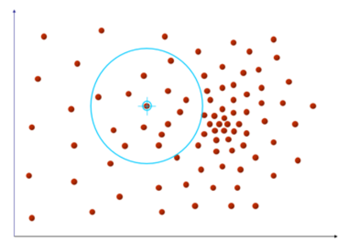




What is Mean Shift



Density Estimation



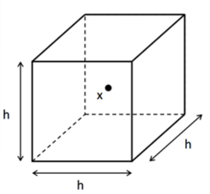
$$P = \frac{N_k}{N}$$

$$f(x) = \frac{P}{S} = \frac{N_k}{NS}$$

Kernel density estimation:

Let $(x_1, x_2, ..., x_n)$ be an independent and identically distributed sample drawn from some distribution with an unknown density $f(x)$. Its kernel density estimator is

$$\hat{f(x)}_{h,k} = \frac{1}{Nh^d} \sum_{n=1}^{N} k(\frac{x_n - x}{h})$$



Kernel function:

$$k(u) = \begin{cases} 1 & |u_i| \leqslant \frac{1}{2}, i = 1, \ldots, D \\ 0 & otherwise \end{cases}$$

Kernel density estimation

$$\hat{f}_{h,K}(x) = \frac{1}{Nh^d}\sum_{n=1}^{N} k(||\frac{x-x_n}{h}||^2)$$

Density gradient:

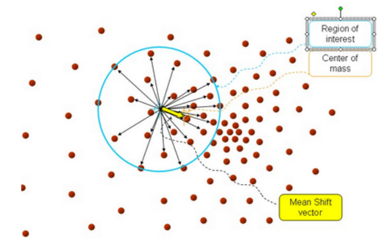
$$\hat{\nabla} f_{h,K}(x) = \frac{2}{Nh^{d+2}}\sum_{n=1}^{N}(x_n - x)[-k'(||\frac{x-x_n}{h}||^2)]$$

$$g(x) = -k'(x)$$

Kernel density estimation

$$
\begin{aligned}
\hat{\nabla} f_{h,K}(x) &= \frac{2}{Nh^{d+2}} \sum_{n=1}^{N} (x_n - x)[g(||\frac{x - x_n}{h}||^2)] \\
&= \underbrace{\frac{2}{h^2}}_{C} \underbrace{[\frac{1}{Nh^d} \sum_{n=1}^{N} g(||\frac{x_n - x}{h}||^2)]}_{f_{h,\hat{G}}(x)} \underbrace{[\frac{\sum_{n=1}^{N} x_n g(||\frac{x_n - x}{h}||^2)}{\sum_{n=1}^{N} g(||\frac{x_n - x}{h}||^2)} - x]}_{m_{h,G}(x)}
\end{aligned}
$$

Kernel density estimation

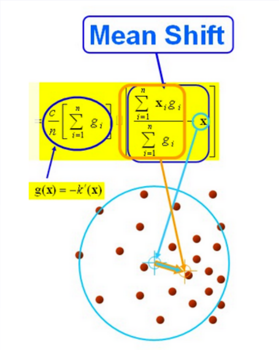


$$M_h = \frac{1}{K} \sum_{x_i S_k} (x_i - x)$$

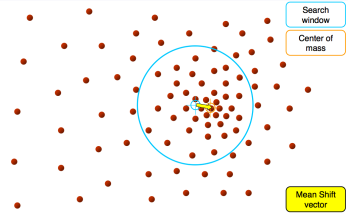
Kernel density estimation

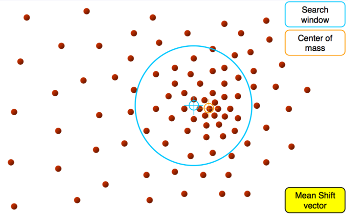
$$m_{h,G}(x) = \frac{\sum_{n=1}^{N} x_n g(||\frac{x_n - x}{h}||^2)}{\sum_{n=1}^{N} g(||\frac{x_n - x}{h}||^2)} - x$$

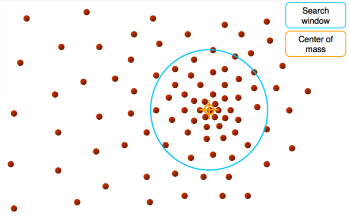
$$m_{h,G}(x) = 0$$



Kernel density estimation







Kernel density estimation

Clustering center:

$$y_{i,k+1} = \frac{\sum_{n=1}^{N} x_n g(||\frac{x_n - x}{h}||^2)}{\sum_{n=1}^{N} g(||\frac{x_n - x}{h}||^2)}$$

Mean Shift

Algorithm:

1. Find features(color, gradients, texture, etc).
2. Initialize windows at individual pixel locations.
3. Compute $y_{i,k+1}$ until convergence, $y_{i,k} = y_{i,k+1}$.
4. Assign $z_i = (x_i, y_{i,k})$.

Thanks!