



Lecture: k-means & mean-shift clustering

Juan Carlos Niebles and Ranjay Krishna

Stanford Vision and Learning Lab



CS 131 Roadmap

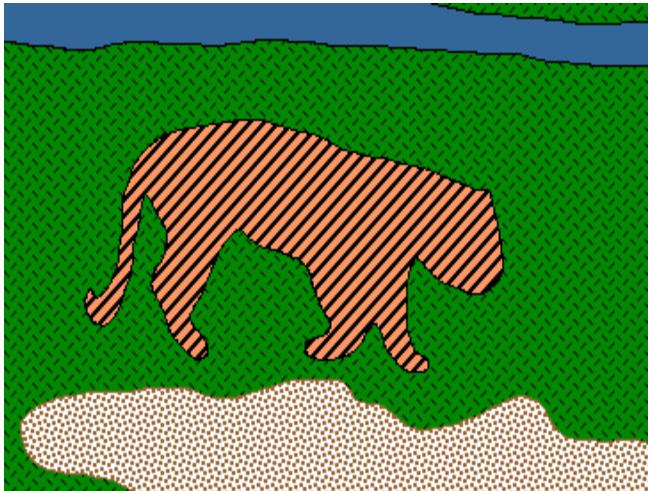
Clustering

22-Oct-2019



Recap: Image Segmentation

- Goal: identify groups of pixels that go together



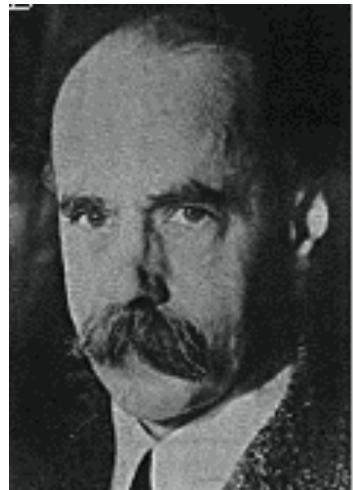


Recap: Gestalt Theory

- 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)



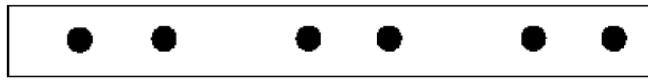
Untersuchungen zur Lehre von der Gestalt,
Psychologische Forschung, Vol. 4, pp. 301-350, 1923
<http://psy.ed.asu.edu/~classics/Wertheimer/Forms/forms.htm>



Recap: Gestalt Factors



Not grouped



Proximity



Similarity



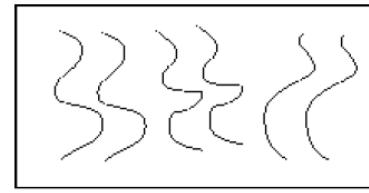
Similarity



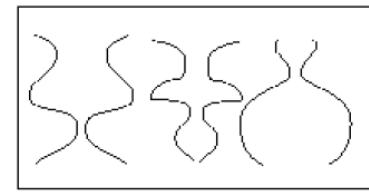
Common Fate



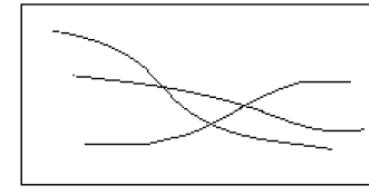
Common Region



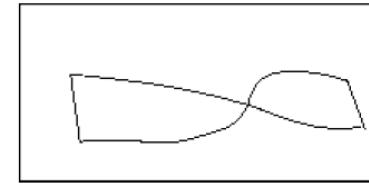
Parallelism



Symmetry



Continuity



Closure

- 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

D. Comaniciu and P. Meer, [Mean Shift: A Robust Approach toward Feature Space Analysis](#), PAMI 2002.



What will we learn today?

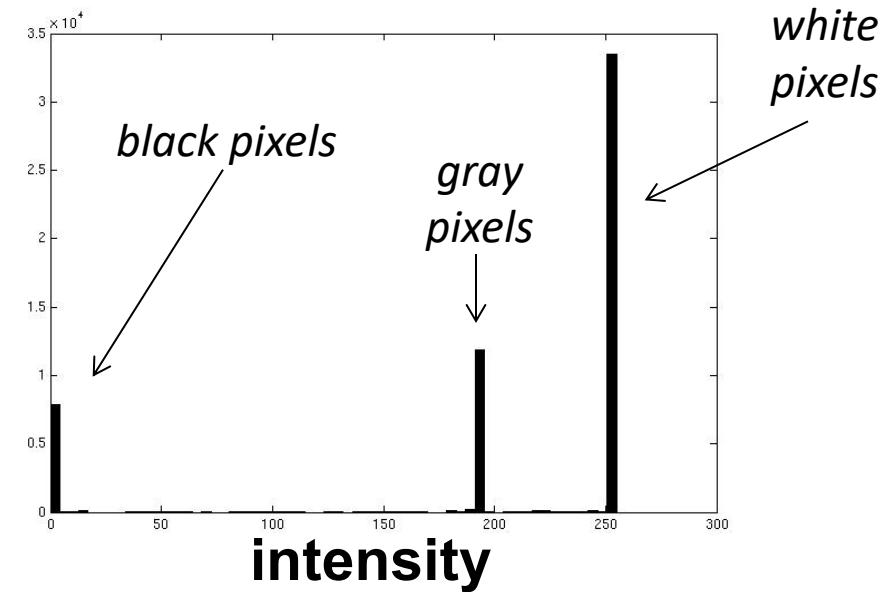
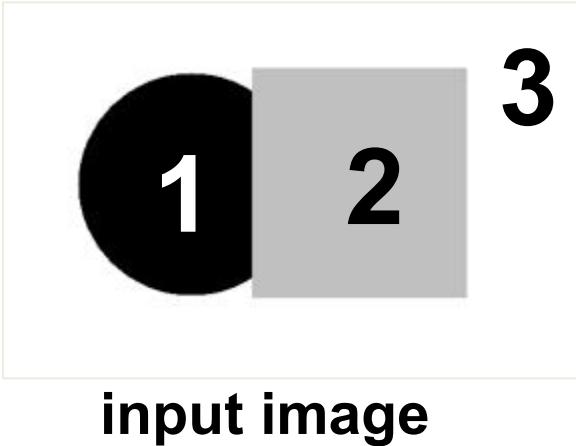
- K-means clustering
- Mean-shift clustering

Reading: [FP] Chapters: 14.2, 14.4

D. Comaniciu and P. Meer, [Mean Shift: A Robust Approach toward Feature Space Analysis](#), PAMI 2002.

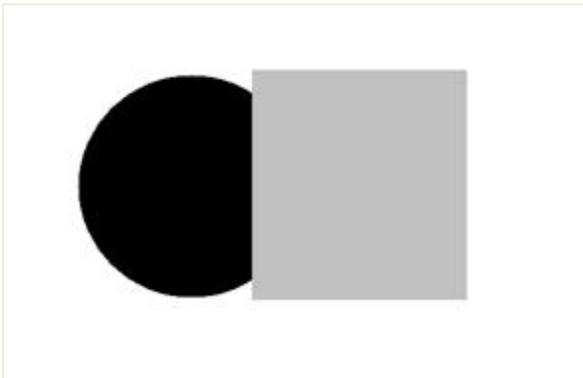


Image Segmentation: Toy Example

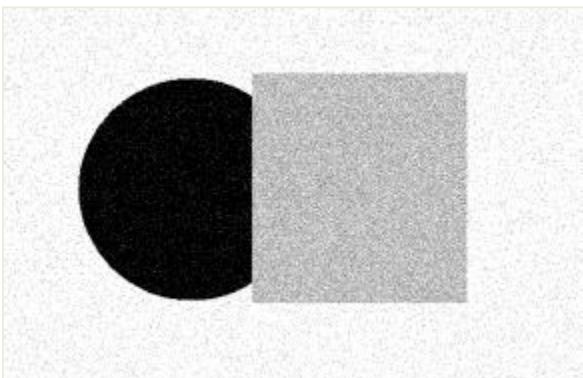
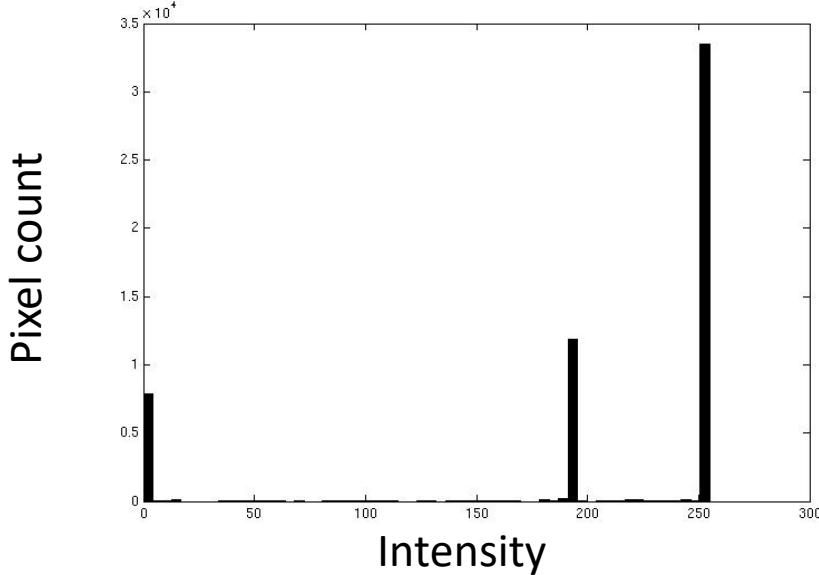


- These intensities define the three groups.
- We could label every pixel in the image according to which of these primary intensities it is.
 - i.e., segment the image based on the intensity feature.
- What if the image isn't quite so simple?

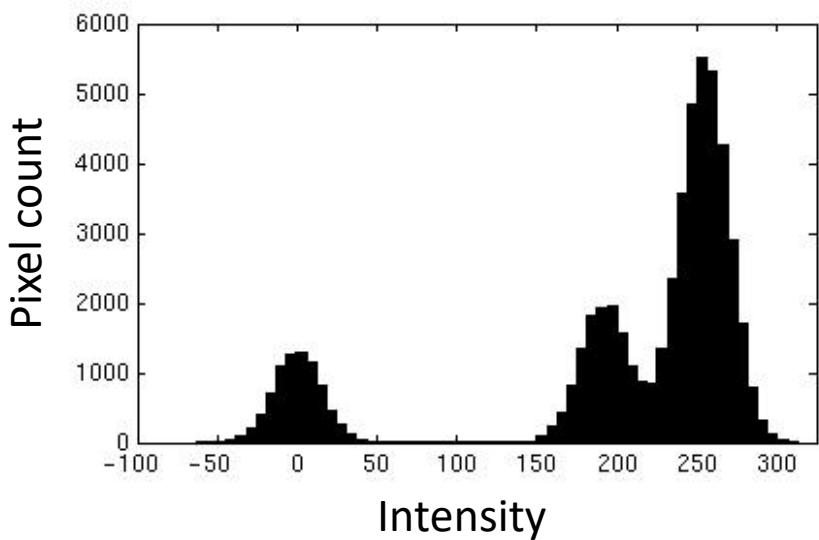
Slide credit: Kristen Grauman



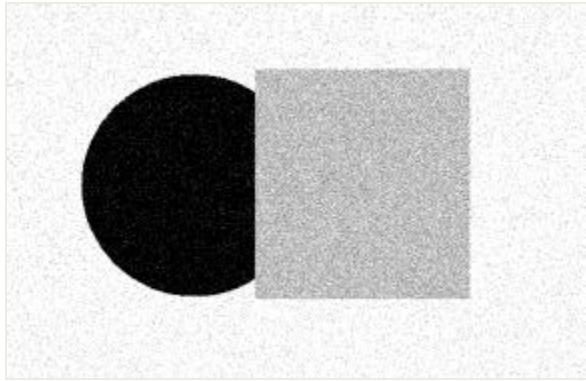
Input image



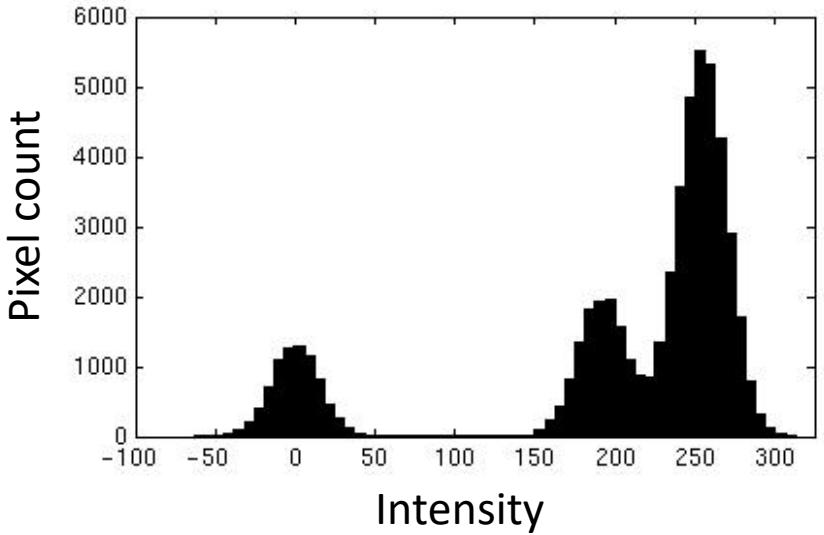
Input image



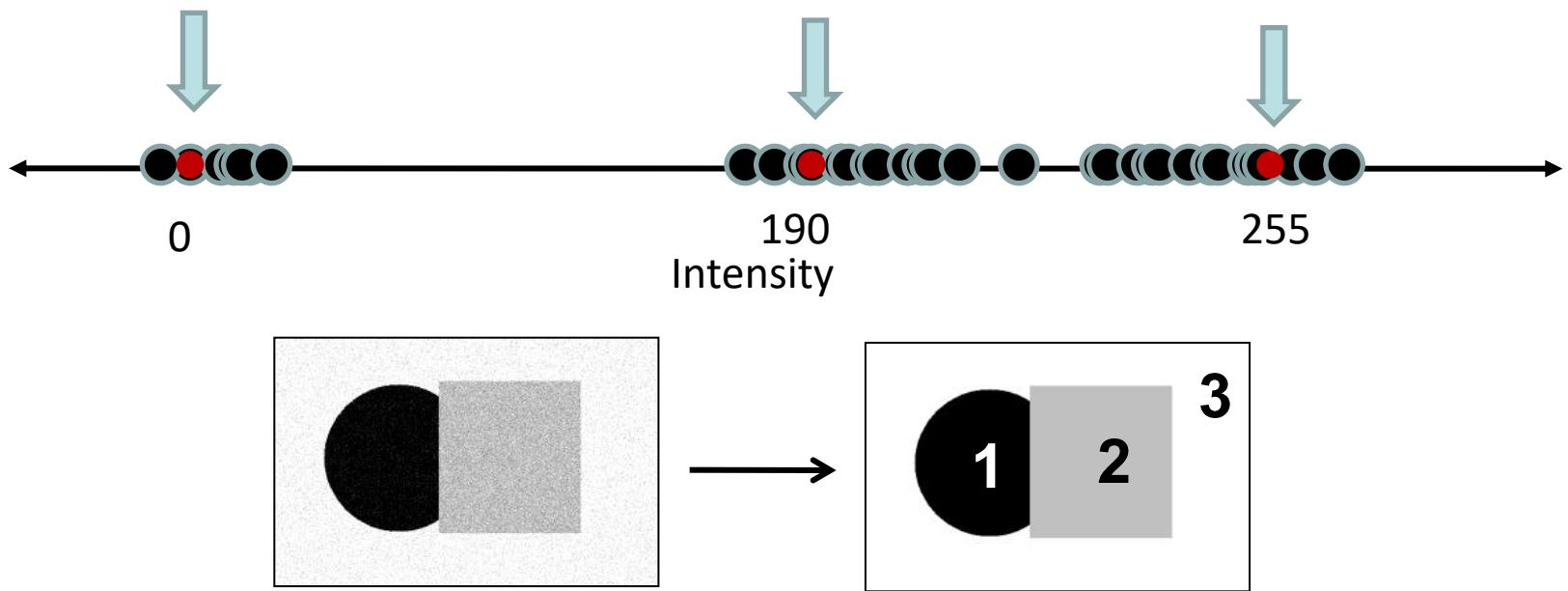
Slide credit: Kristen Grauman



Input image



- Now how to determine the three main intensities that define our groups?
- We need to cluster.



- 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 :

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

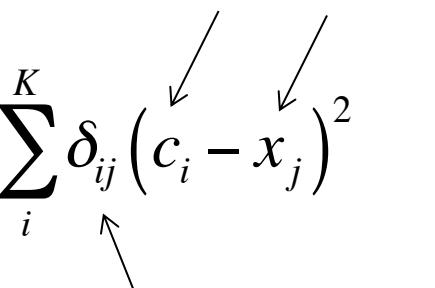


Clustering for Summarization

Goal: cluster to minimize variance in data given clusters

- Preserve information

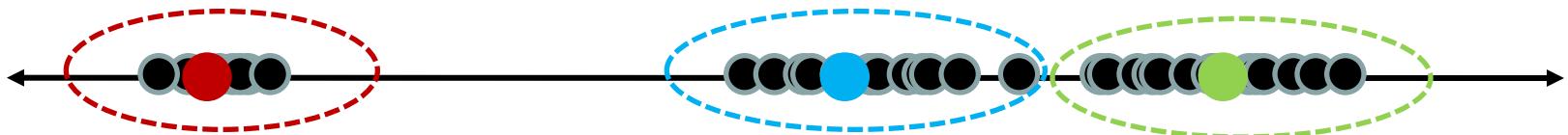
$$c^*, \delta^* = \arg \min_{c, \delta} \frac{1}{N} \sum_j^K \sum_i \delta_{ij} (c_i - x_j)^2$$

Cluster center Data

Whether x_j is assigned to c_i

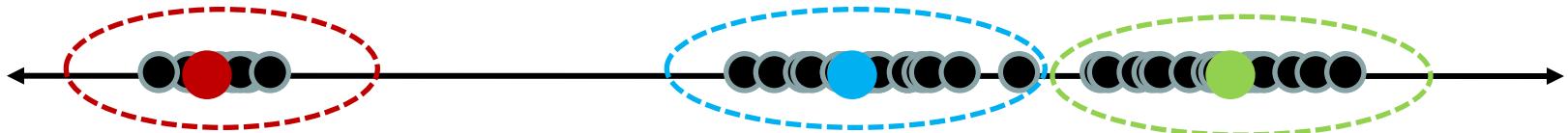


Clustering

- With this objective, it is a “chicken and egg” problem:
 - If we knew the *cluster centers*, we could allocate points to groups by assigning each to its closest center.



- If we knew the *group memberships*, we could get the centers by computing the mean per group.





K-means clustering

1. Initialize ($t = 0$): cluster centers c_1, \dots, c_K
2. Compute δ^t : assign each point to the closest center

– δ^t denotes the set of assignment for each x_j to cluster c_i at iteration t

$$\delta^t = \operatorname{argmin}_{\delta} \frac{1}{N} \sum_j^K \sum_i^K \delta_{ij}^{t-1} (c_i^{t-1} - x_j)^2$$

1. Computer c^t : update cluster centers as the mean of the points

$$c^t = \operatorname{argmin}_c \frac{1}{N} \sum_j^K \sum_i^K \delta_{ij}^t (c_i^{t-1} - x_j)^2$$

1. Update $t = t + 1$, Repeat Step 2-3 till stopped





K-means clustering

1. Initialize ($t = 0$): cluster centers c_1, \dots, c_K
 - Commonly used: random initialization
 - Or greedily choose K to minimize residual
2. Compute δ^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

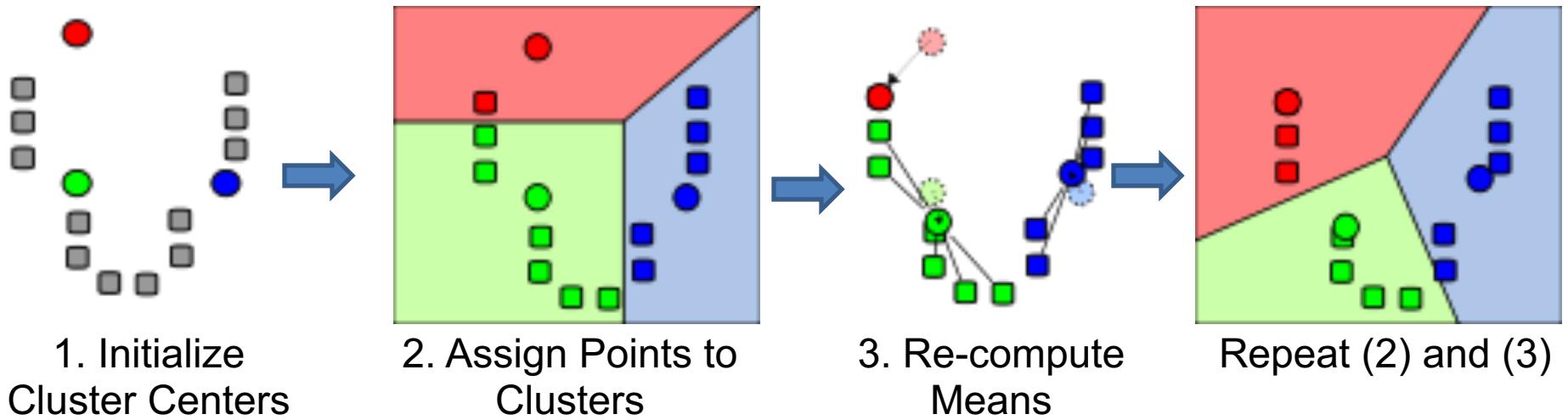
1. Computer c^t : update cluster centers as the mean of the points

$$c^t = \operatorname{argmin}_c \frac{1}{N} \sum_j^K \sum_i \delta_{ij}^t (c_i^{t-1} - x_j)^2$$

2. Update $t = t + 1$, Repeat Step 2-3 till stopped
 - c^t doesn't change anymore.



K-means clustering



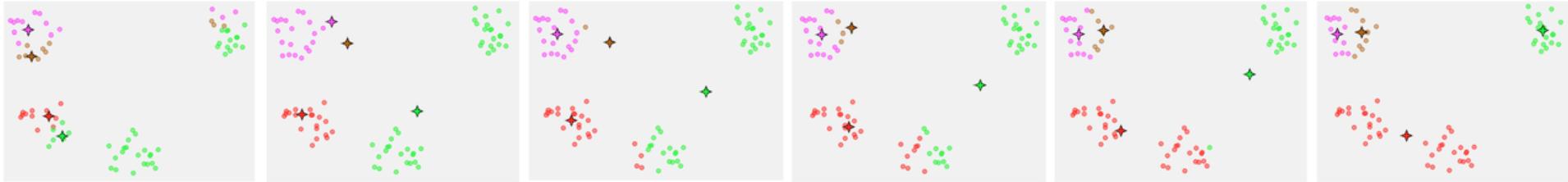
- Java demo:

http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html

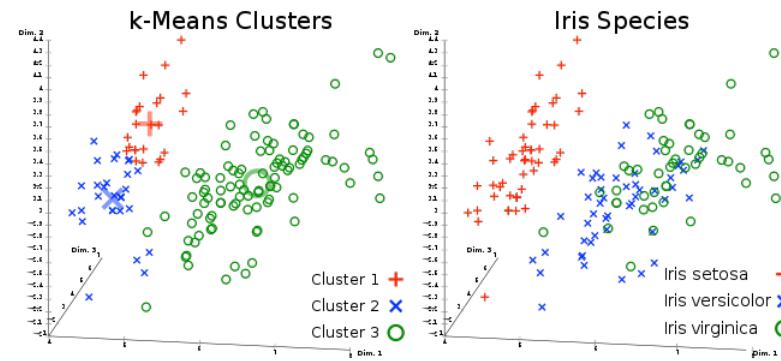


K-means clustering

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



- Better fit for spherical data



- Need to pick K (# of clusters)

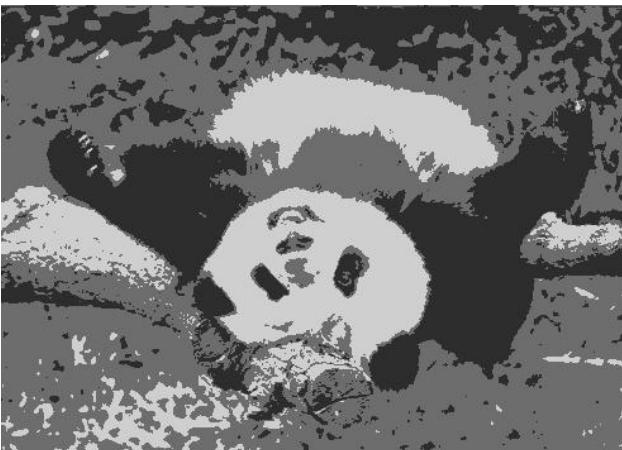
Segmentation as Clustering



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



Feature Space

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



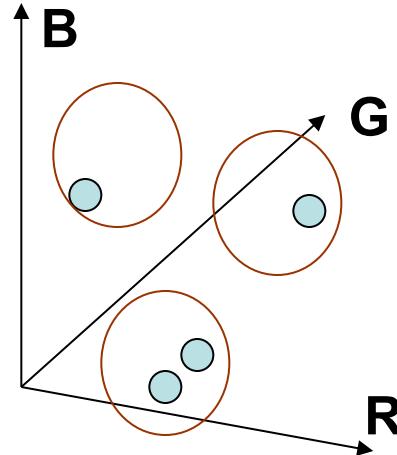
- Feature space: intensity value (1D)

Slide credit: Kristen Grauman



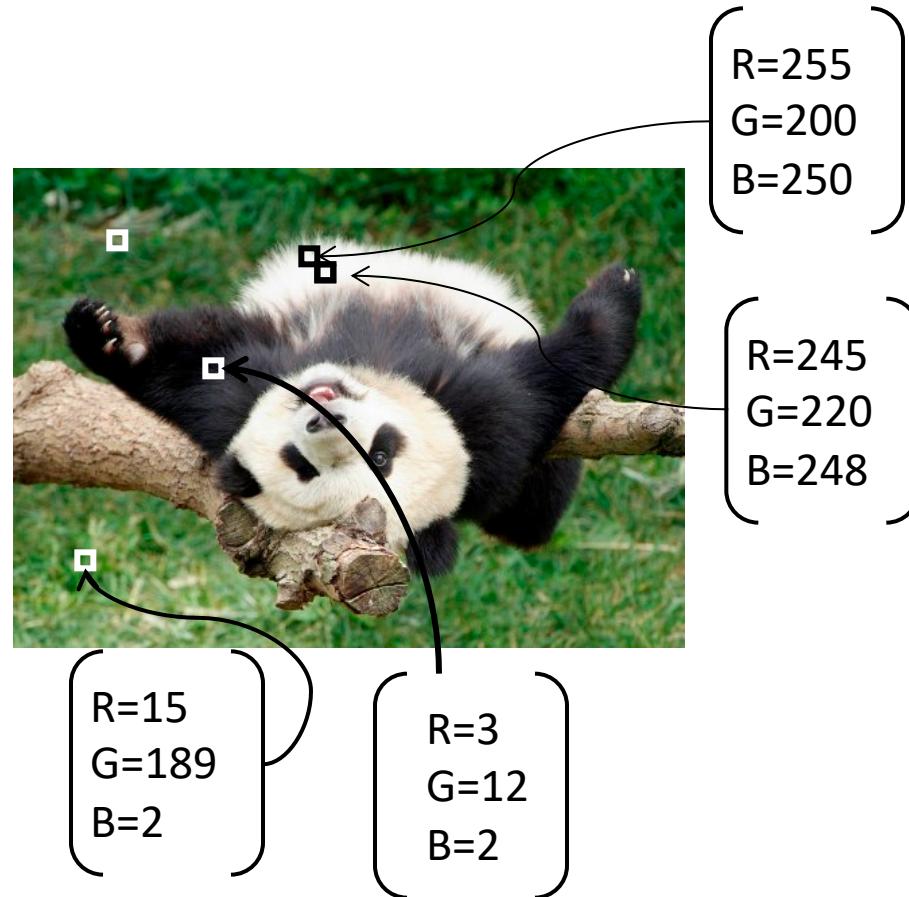
Feature Space

- 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)

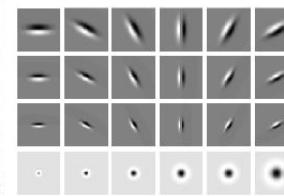
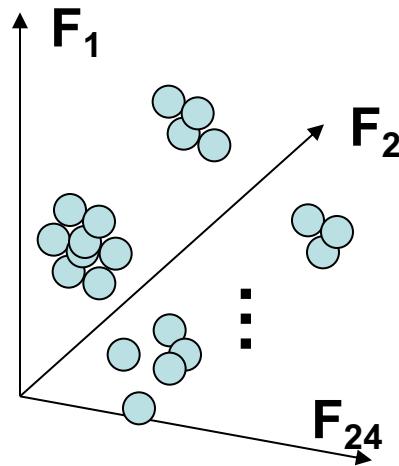
Slide credit: Kristen Grauman





Feature Space

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



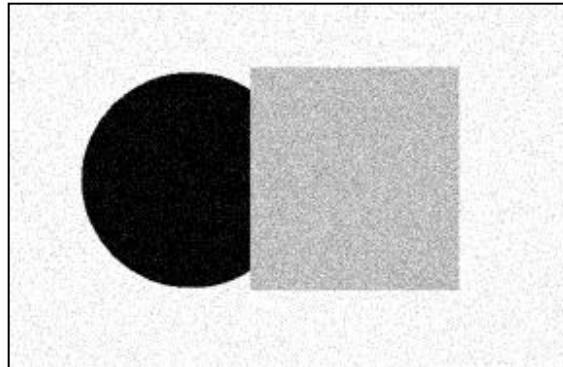
Filter bank of
24 filters

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

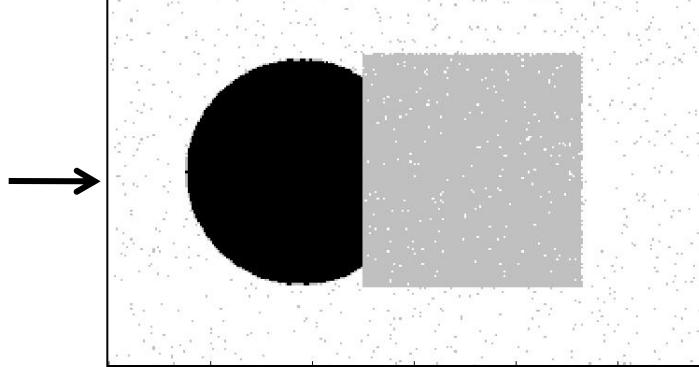


Smoothing Out Cluster Assignments

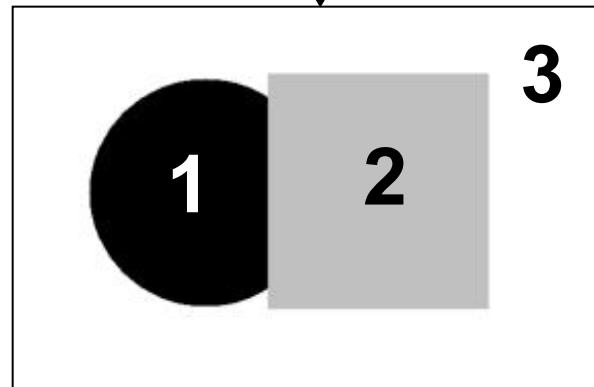
- Assigning a cluster label per pixel may yield outliers:



Original



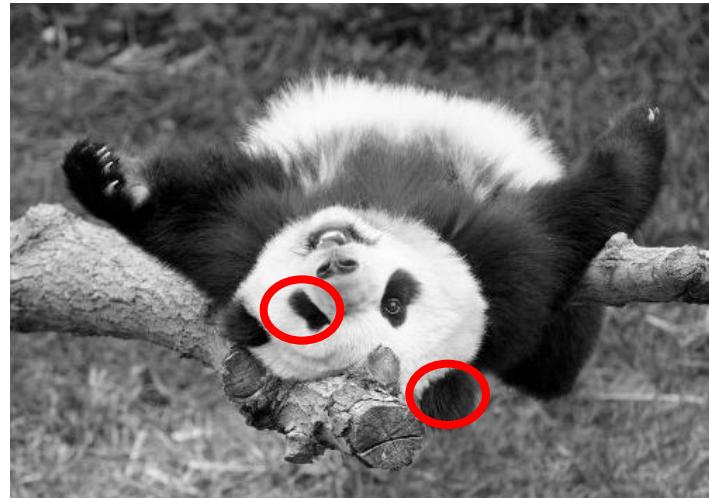
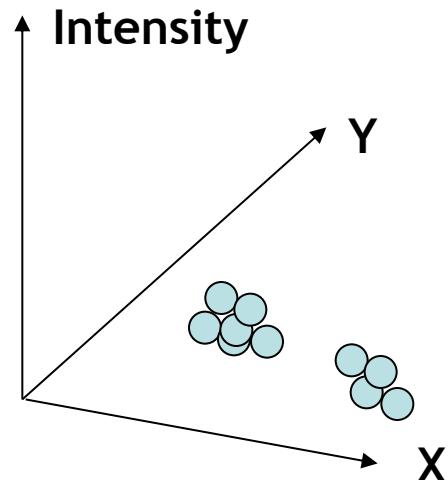
Labeled by cluster center's intensity



- How can we ensure they are spatially smooth?

Segmentation as Clustering

- Depending on what we choose as the *feature space*, we can group pixels in different ways.
- Grouping pixels based on *intensity+position* similarity



⇒ Way to encode both *similarity* and *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

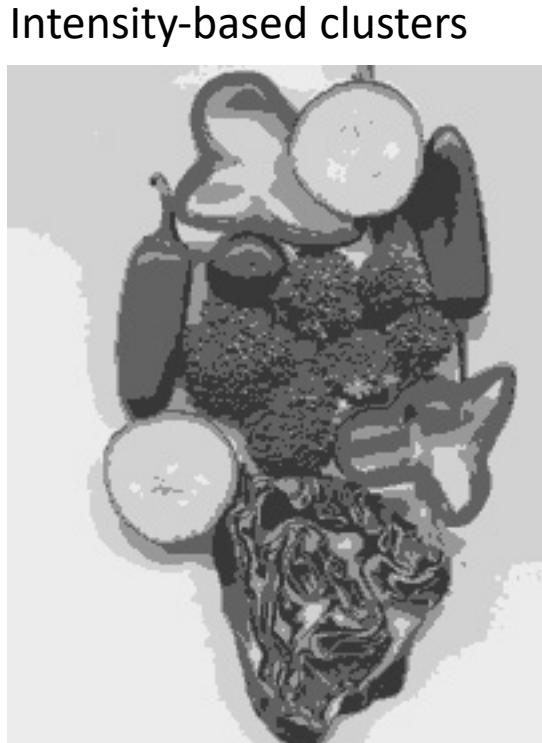


Image source: Forsyth & Ponce

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



Image source: Forsyth & Ponce



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.

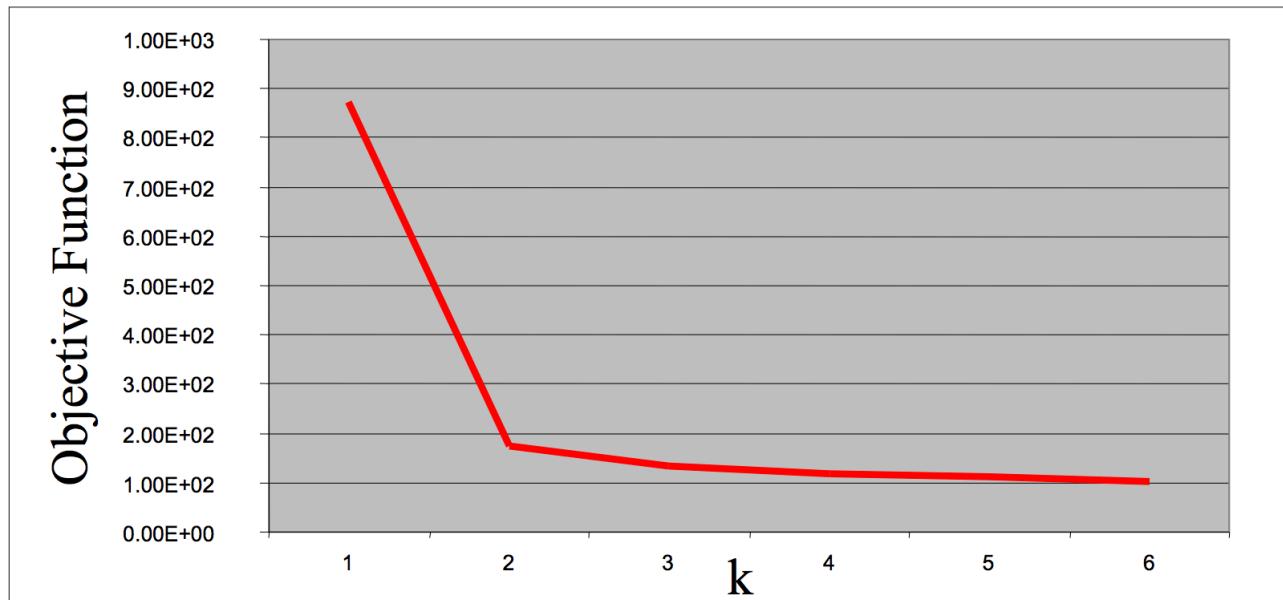


How to choose the number of clusters?

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”.



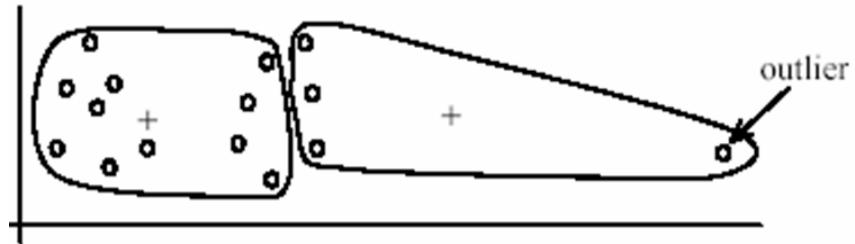


K-Means pros and cons

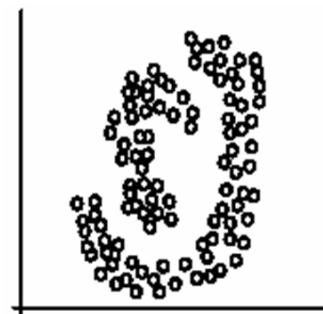
- Pros
 - Finds cluster centers that minimize conditional variance (good representation of data)
 - Simple and fast, Easy to implement
- Cons
 - Need to choose K
 - Sensitive to outliers
 - Prone to local minima
 - All clusters have the same parameters (e.g., distance measure is non-adaptive)
 - *Can be slow: each iteration is $O(KNd)$ for N d-dimensional points
- Usage
 - Unsupervised clustering
 - Rarely used for pixel segmentation



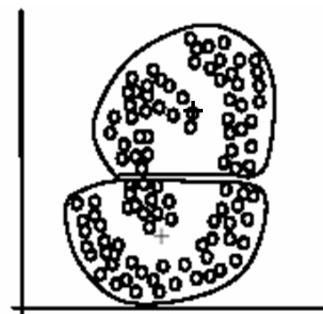
(B): Ideal clusters



outlier



(A): Two natural clusters

(B): k -means clusters



What will we learn today?

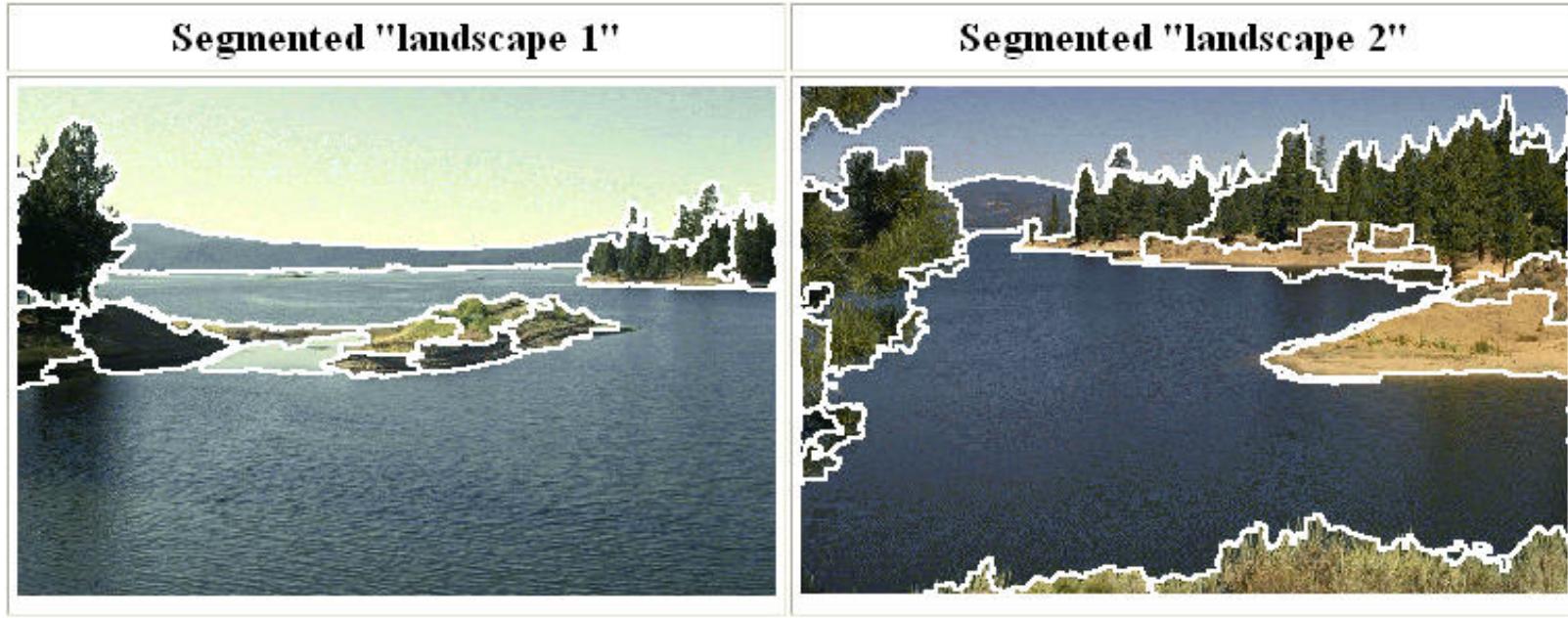
- K-means clustering
- Mean-shift clustering

Reading: [FP] Chapters: 14.2, 14.4

D. Comaniciu and P. Meer, [Mean Shift: A Robust Approach toward Feature Space Analysis](#), PAMI 2002.

Mean-Shift Segmentation

- An advanced and versatile technique for clustering-based segmentation

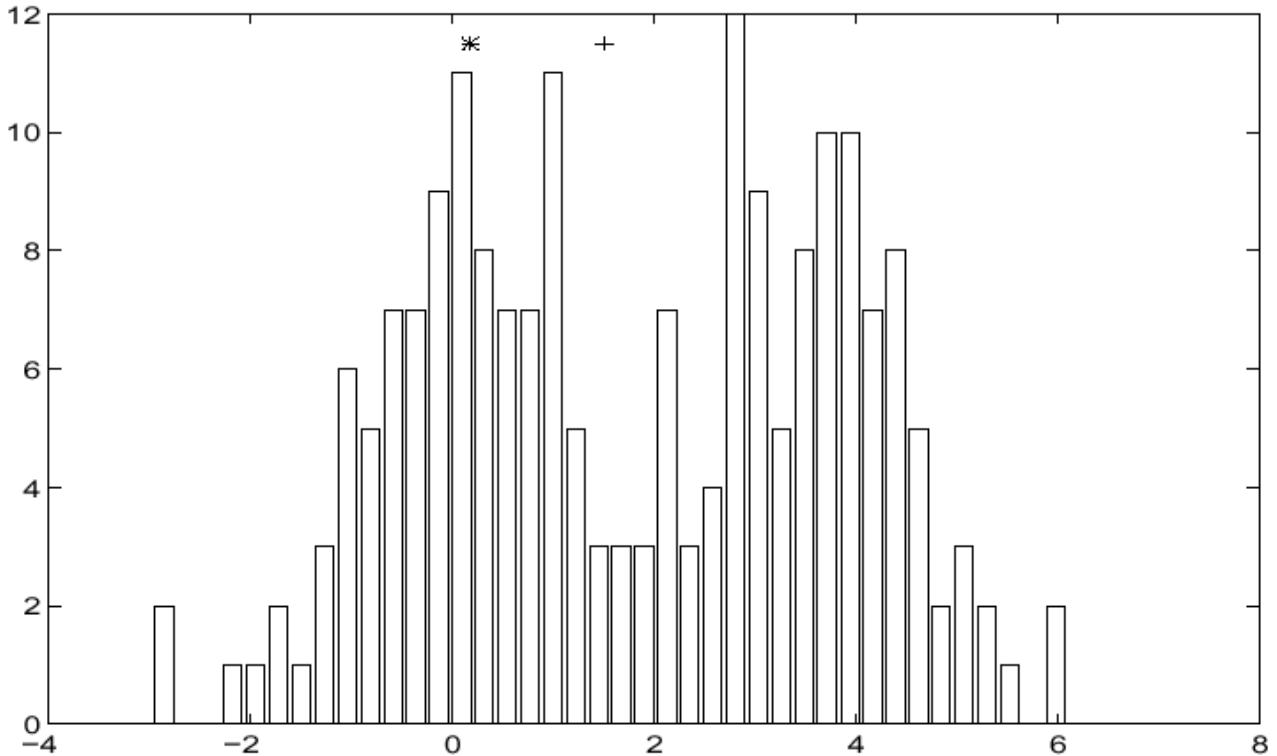


<http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html>

D. Comaniciu and P. Meer, [Mean Shift: A Robust Approach toward Feature Space Analysis](#), PAMI 2002.



Mean-Shift Algorithm



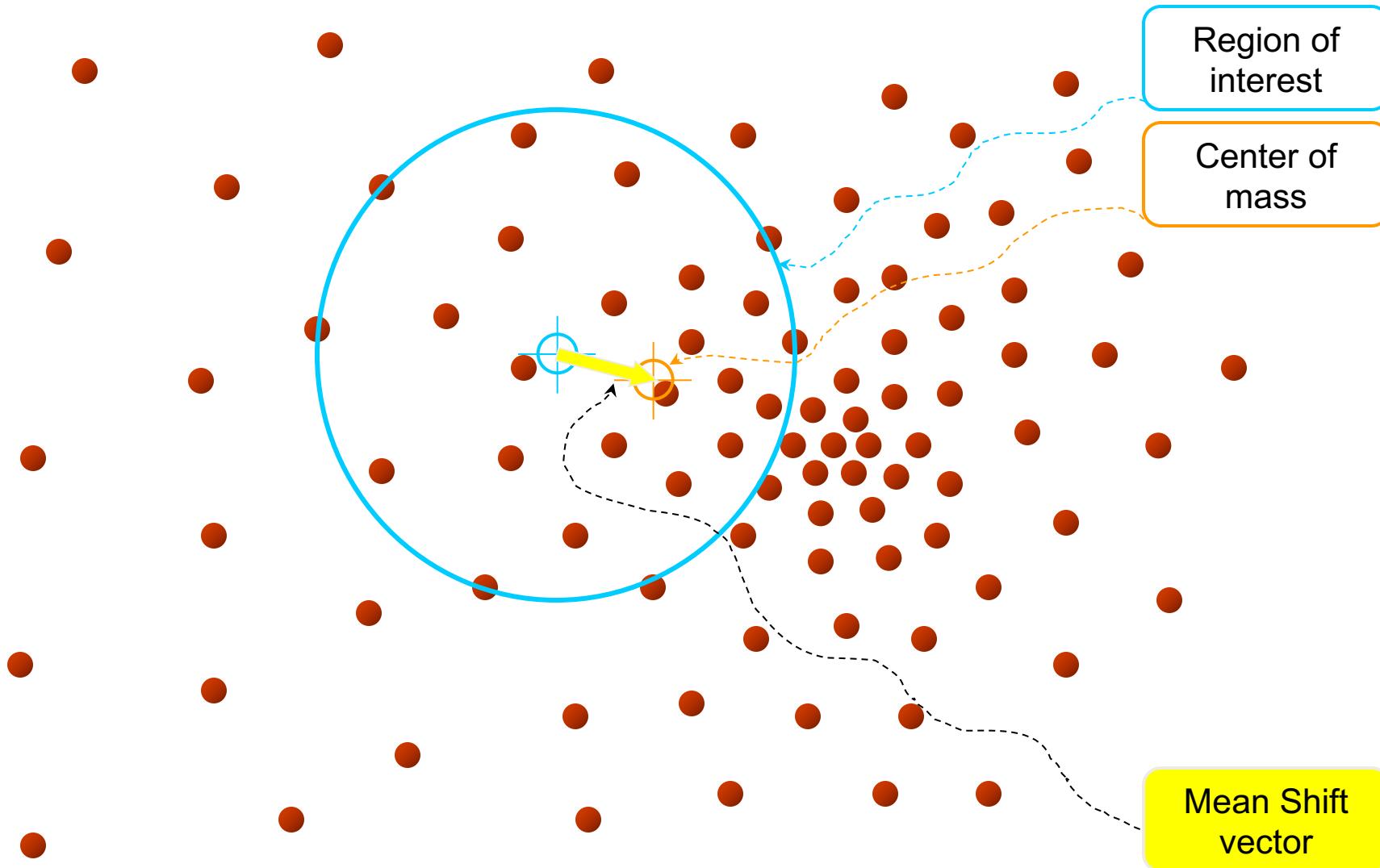
$$\sum_{x \in W} x H(x)$$

- Iterative Mode Search

1. Initialize random seed, and window W
2. Calculate center of gravity (the “mean”) of W :
3. Shift the search window to the mean
4. Repeat Step 2 until convergence

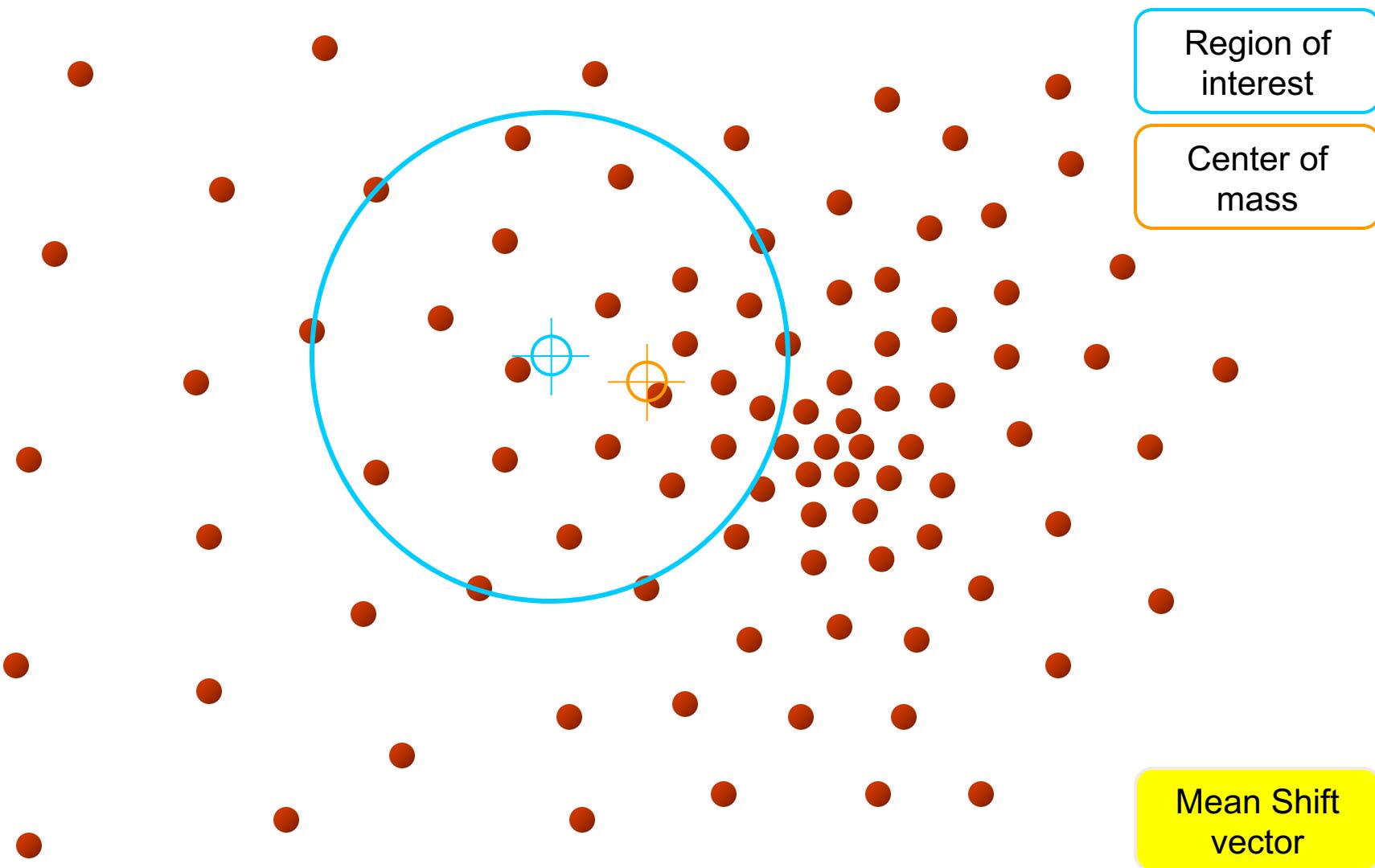


Mean-Shift



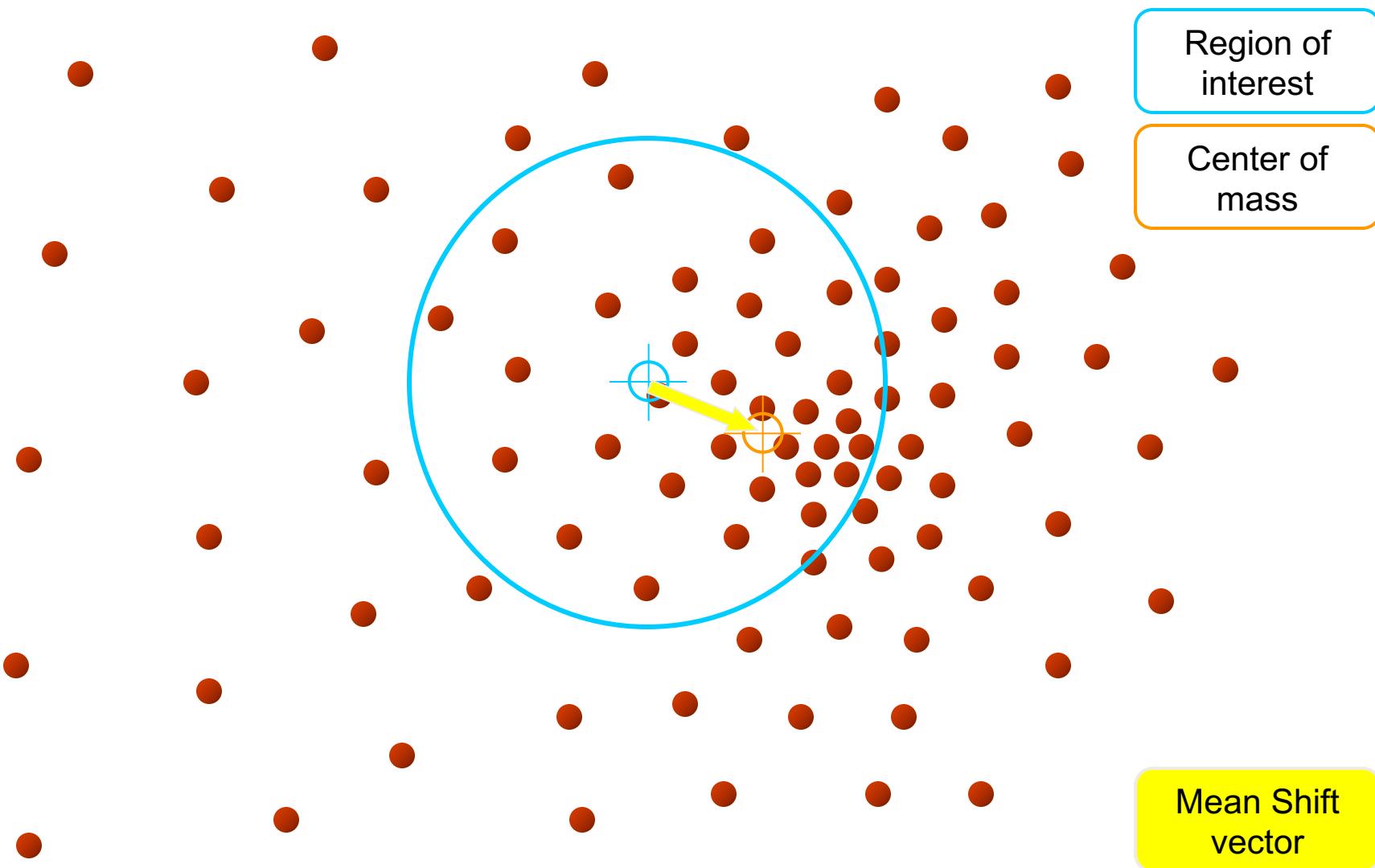


Mean-Shift



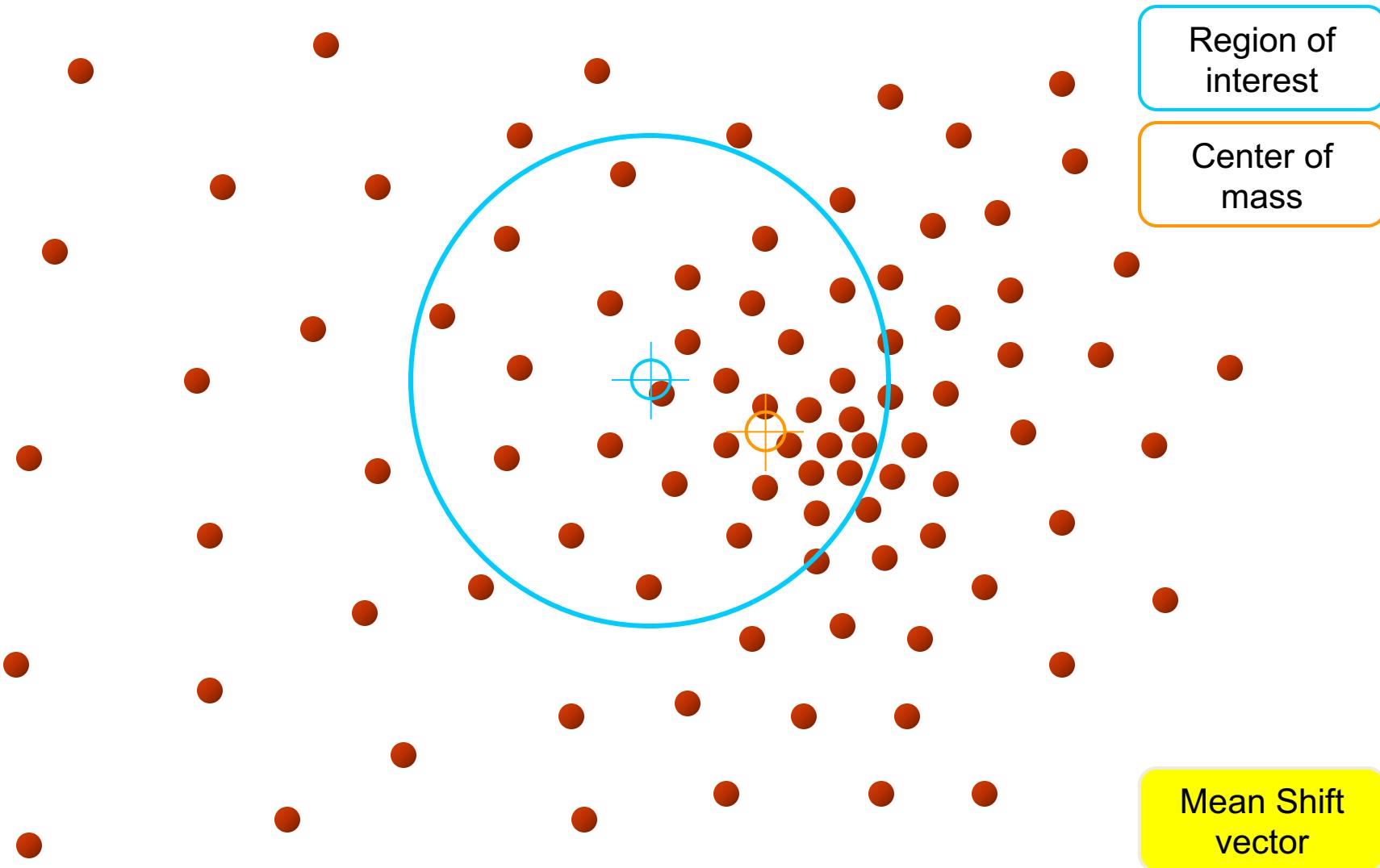


Mean-Shift



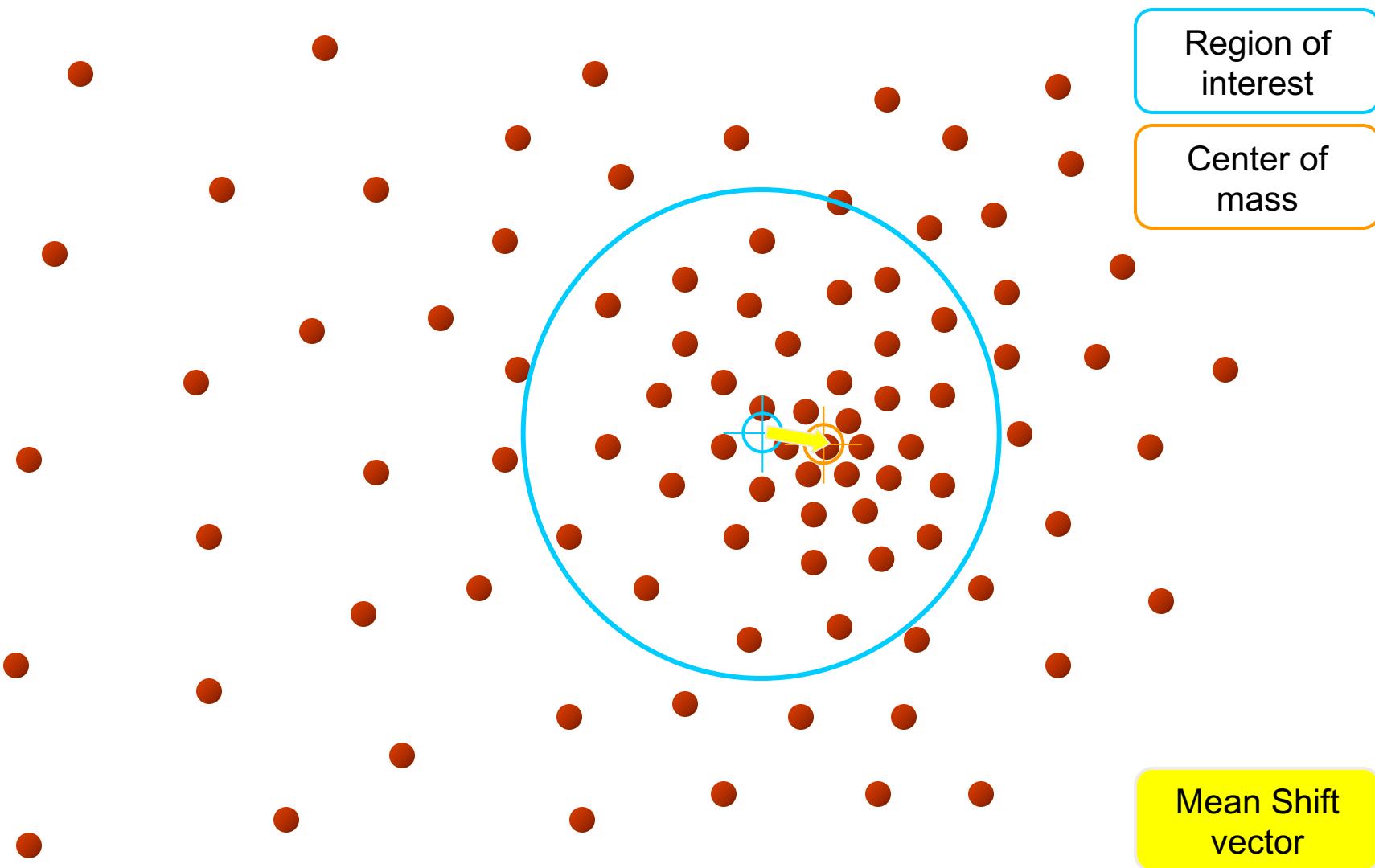


Mean-Shift

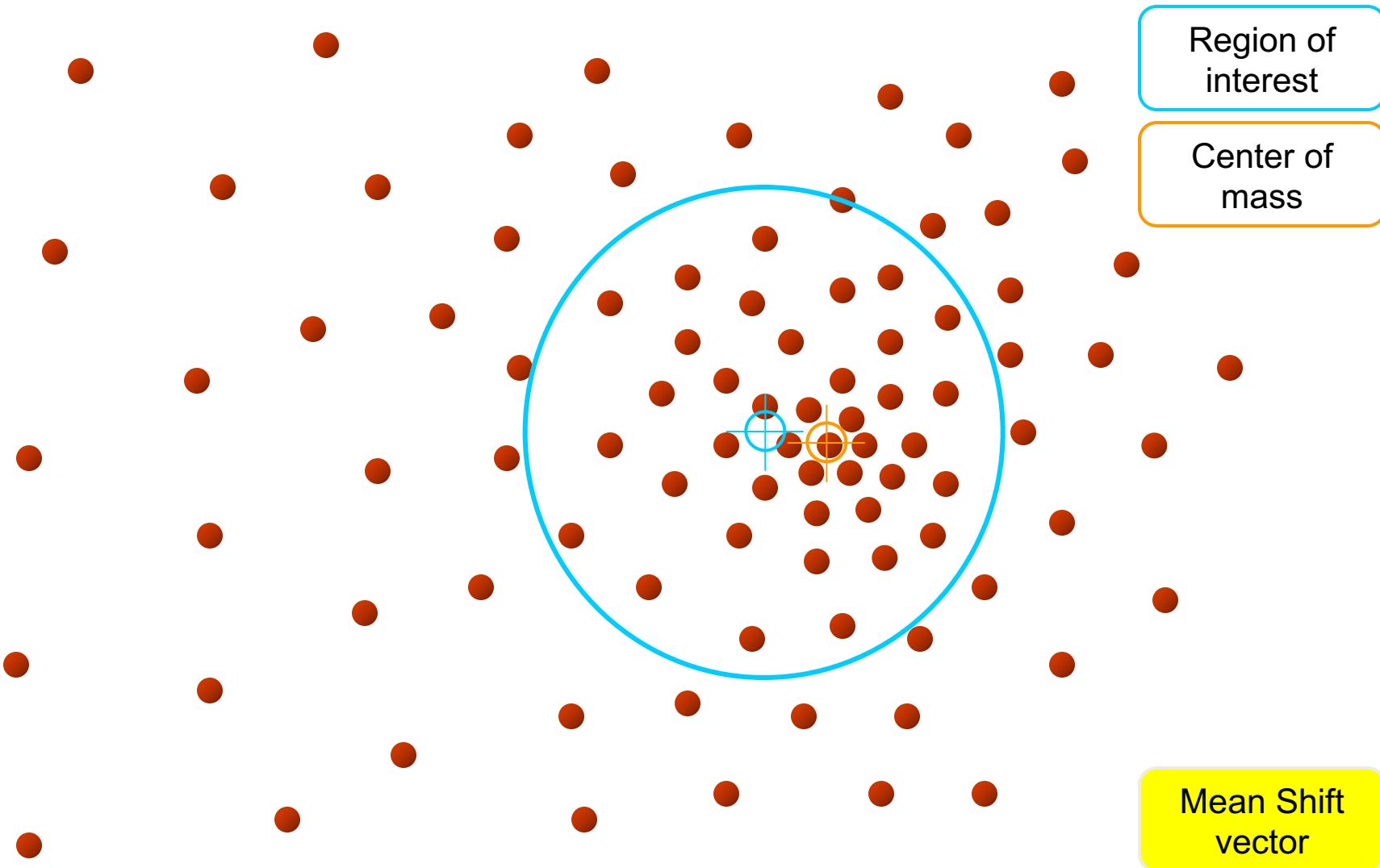




Mean-Shift

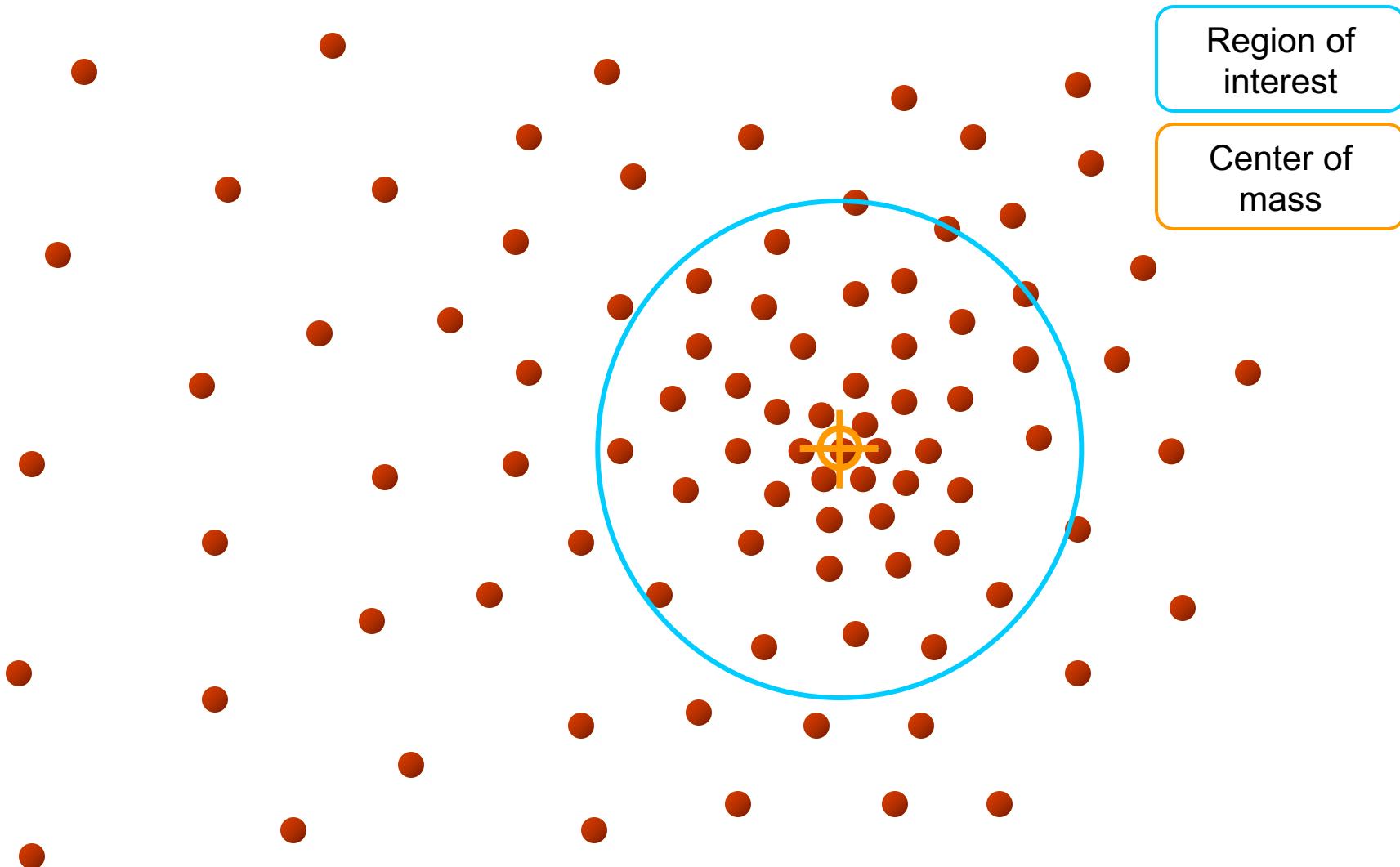


Mean-Shift



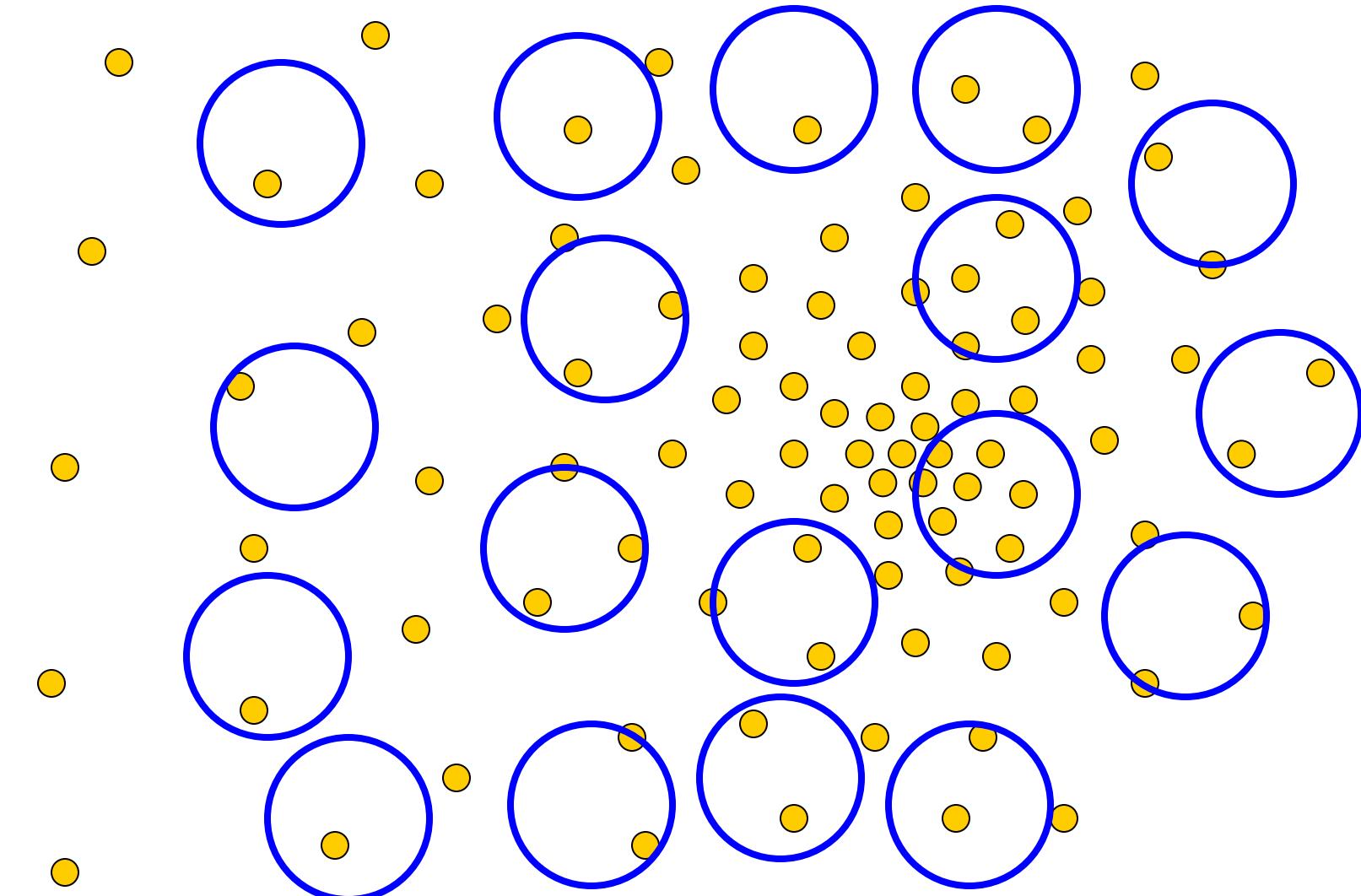


Mean-Shift



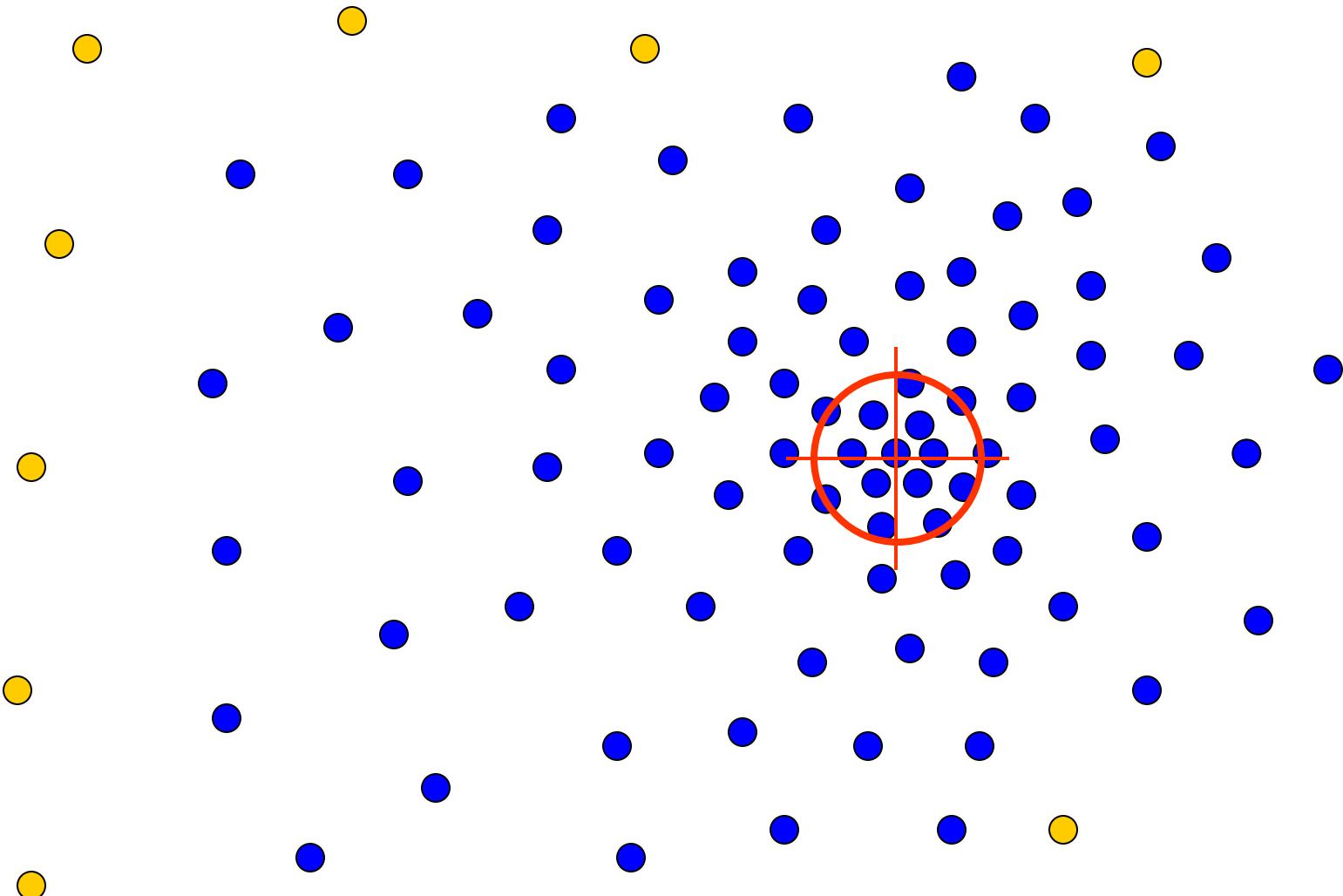


Real Modality Analysis





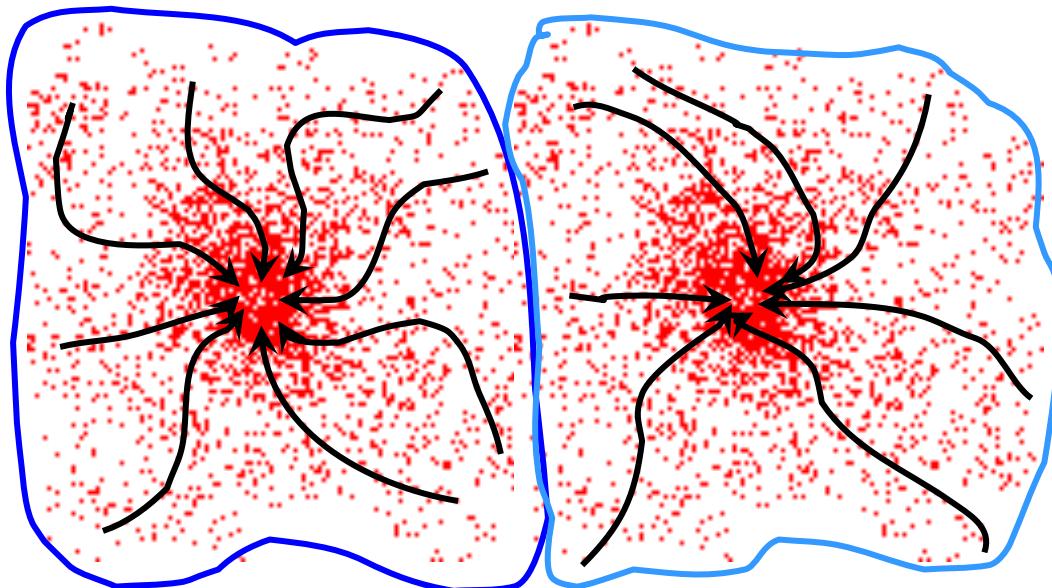
Real Modality Analysis



The **blue** data points were traversed by the windows towards the mode.

Mean-Shift Clustering

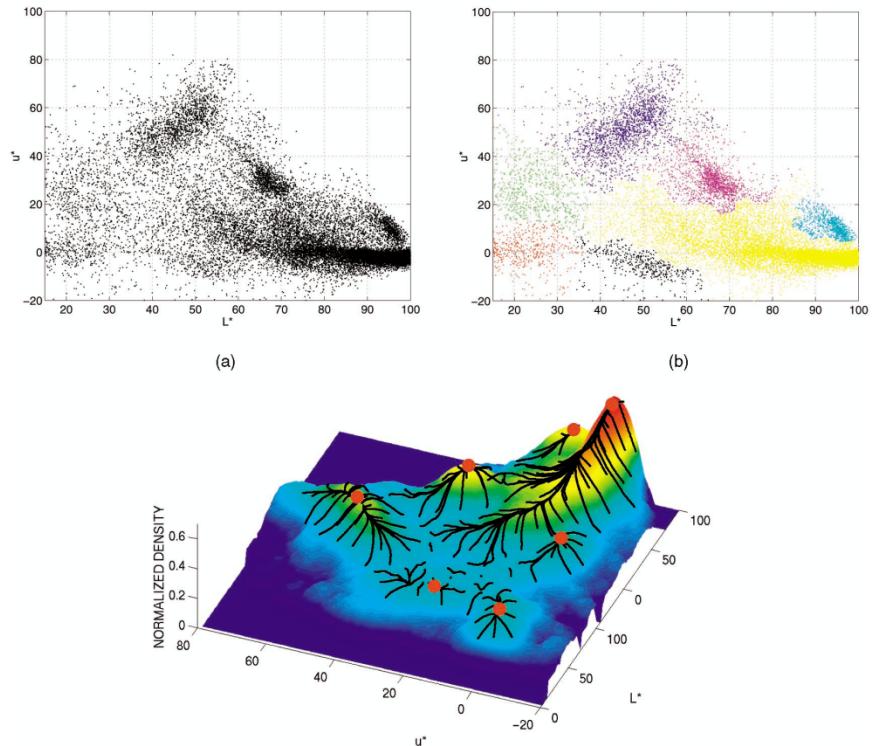
- Cluster: all data points in the attraction basin of a mode
- Attraction basin: the region for which all trajectories lead to the same mode





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

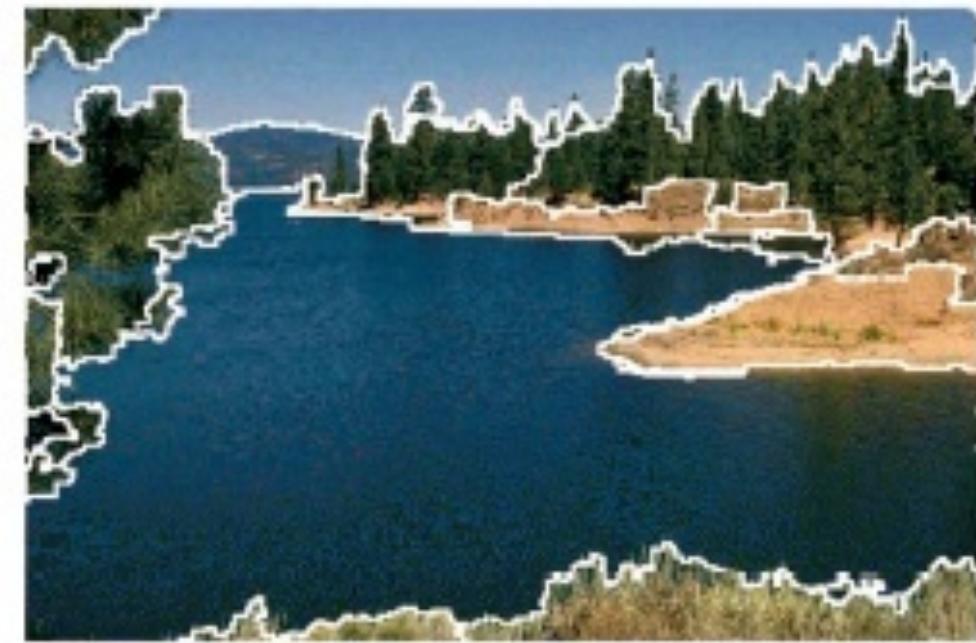


<http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html>

Slide credit: Svetlana Lazebnik

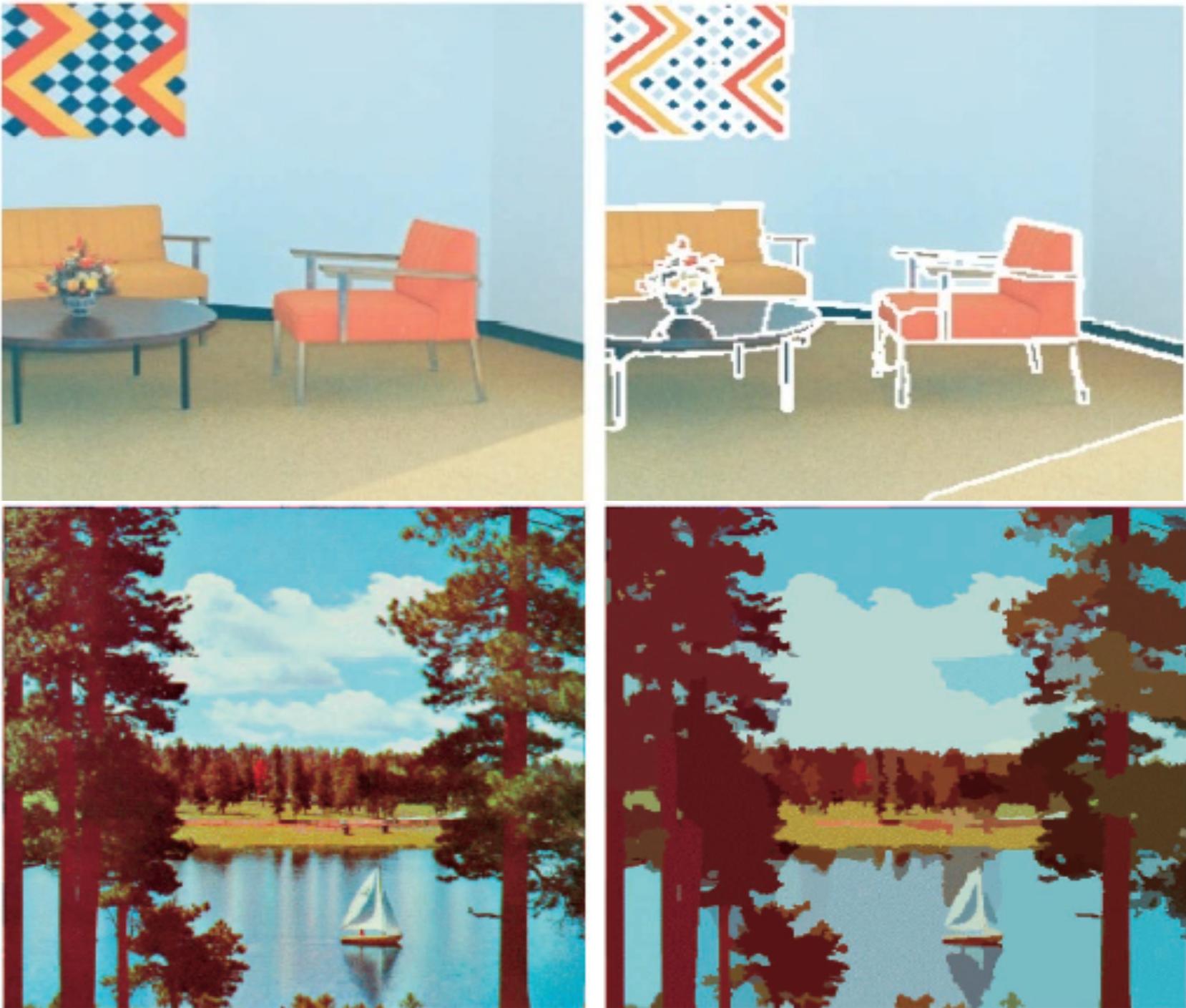
More Results

Slide credit: Svetlana Lazebnik



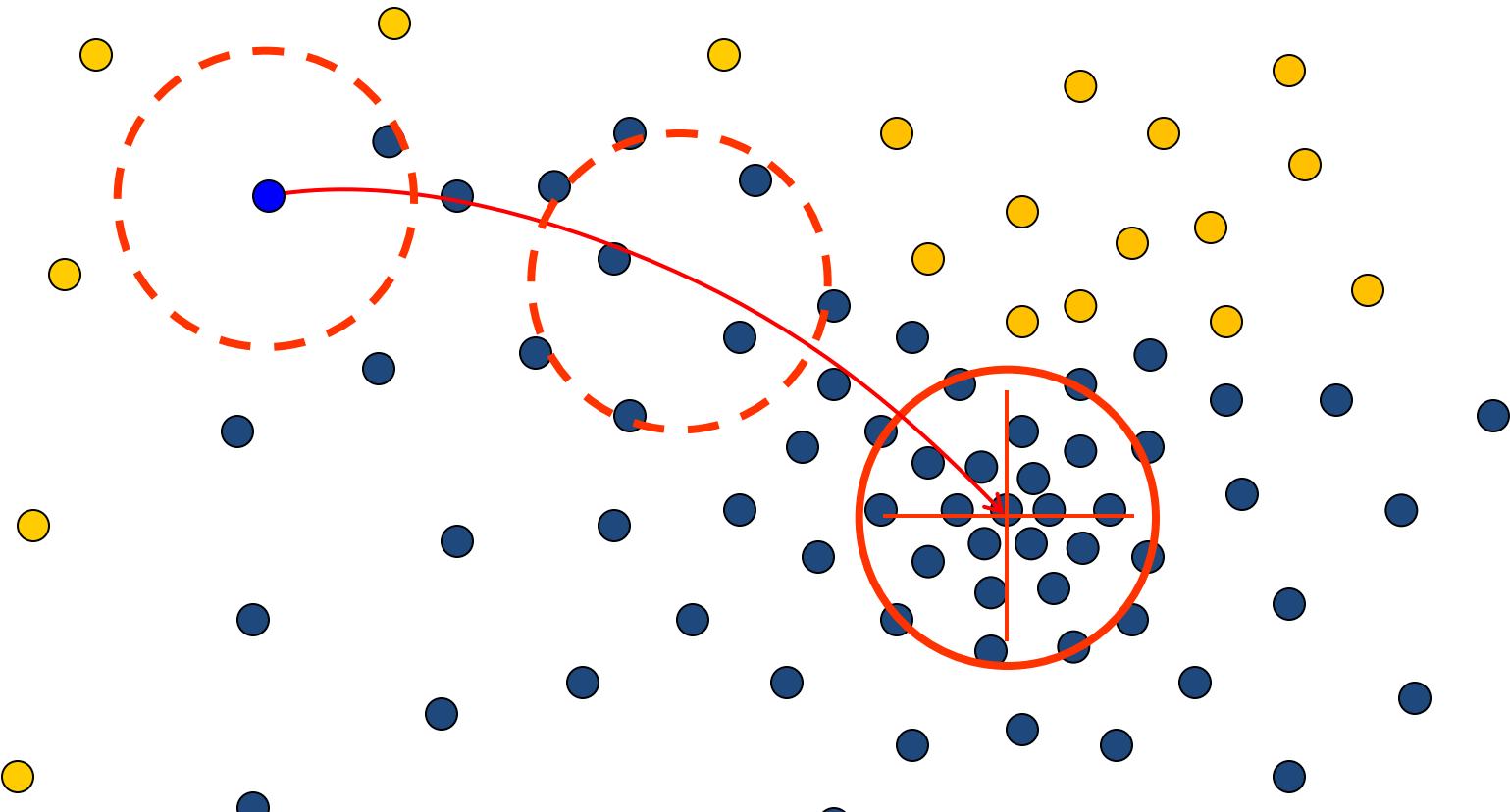


More Results





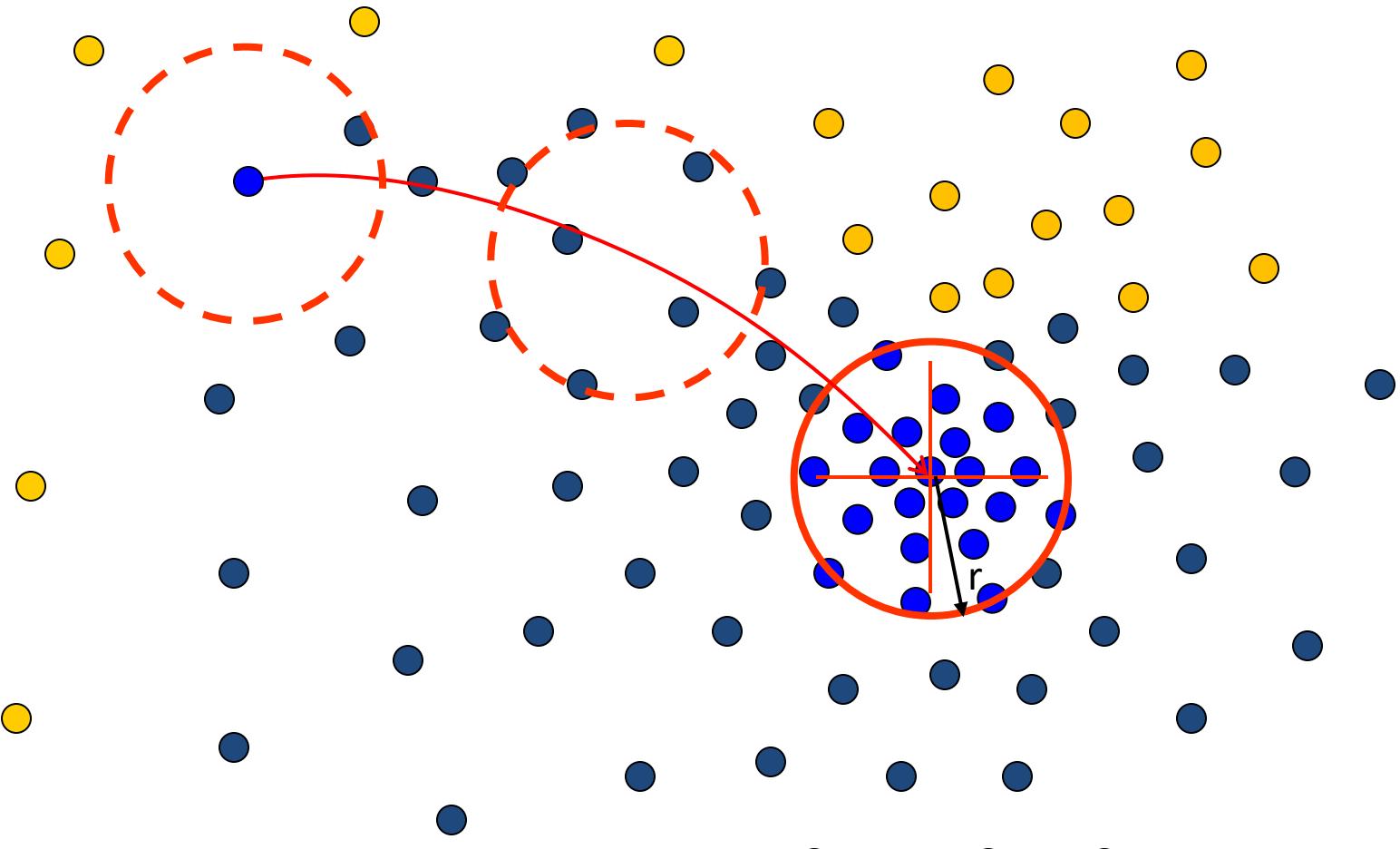
Problem: Computational Complexity



- Need to shift many windows...
- Many computations will be redundant.



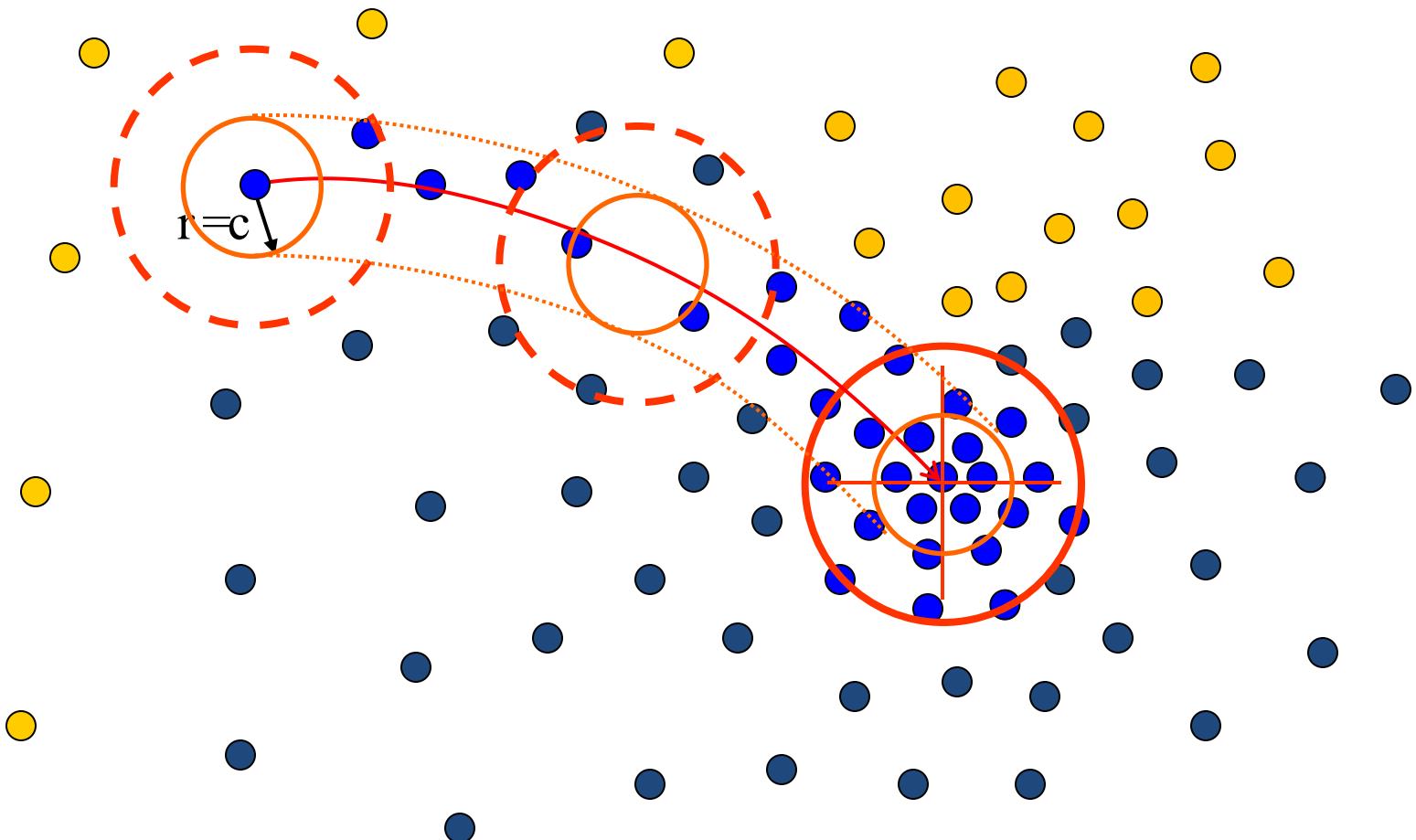
Speedups: Basin of Attraction



1. Assign all points within radius r of end point to the mode.



Speedups



2. Assign all points within radius r/c of the search path to the mode -> 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 kernel¹ (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.



Other Kernels

A kernel is a function that satisfies the following requirements :

$$1. \int_{R^d} \phi(x) = 1$$

$$2. \phi(x) \geq 0$$

Some examples of kernels include :

$$1. \text{Rectangular } \phi(x) = \begin{cases} 1 & a \leq x \leq b \\ 0 & \text{else} \end{cases}$$

$$2. \text{Gaussian } \phi(x) = e^{-\frac{x^2}{2\sigma^2}}$$

$$3. \text{Epanechnikov } \phi(x) = \begin{cases} \frac{3}{4}(1 - x^2) & \text{if } |x| \leq 1 \\ 0 & \text{else} \end{cases}$$

[source](#)



Technical Details

Taking the derivative of: $\hat{f}_K = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$

$$\nabla \hat{f}(\mathbf{x}) = \underbrace{\frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right) \right]}_{\text{term 1}} \underbrace{\left[\frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \right]}_{\text{term 2}}, \quad (3)$$

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

- Term1: this is proportional to the density estimate at \mathbf{x} (similar to equation 1 from two slides ago).
- Term2: this is the mean-shift vector that points towards the direction of maximum density.



Technical Details

Finally, the mean shift procedure from a given point \mathbf{x}_t is:

1. Computer the mean shift vector \mathbf{m} :

$$\left[\frac{\sum_{i=1}^n \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x} \right]$$

2. Translate the density window:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{m}(\mathbf{x}_i^t).$$

3. Iterate steps 1 and 2 until convergence.

$$\nabla f(\mathbf{x}_i) = 0.$$



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 ($\sim 2s/\text{image}$)
- 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

D. Comaniciu and P. Meer, [Mean Shift: A Robust Approach toward Feature Space Analysis](#), PAMI 2002.