Clustering

Faisal Qureshi
faisal.qureshi@uoit.ca

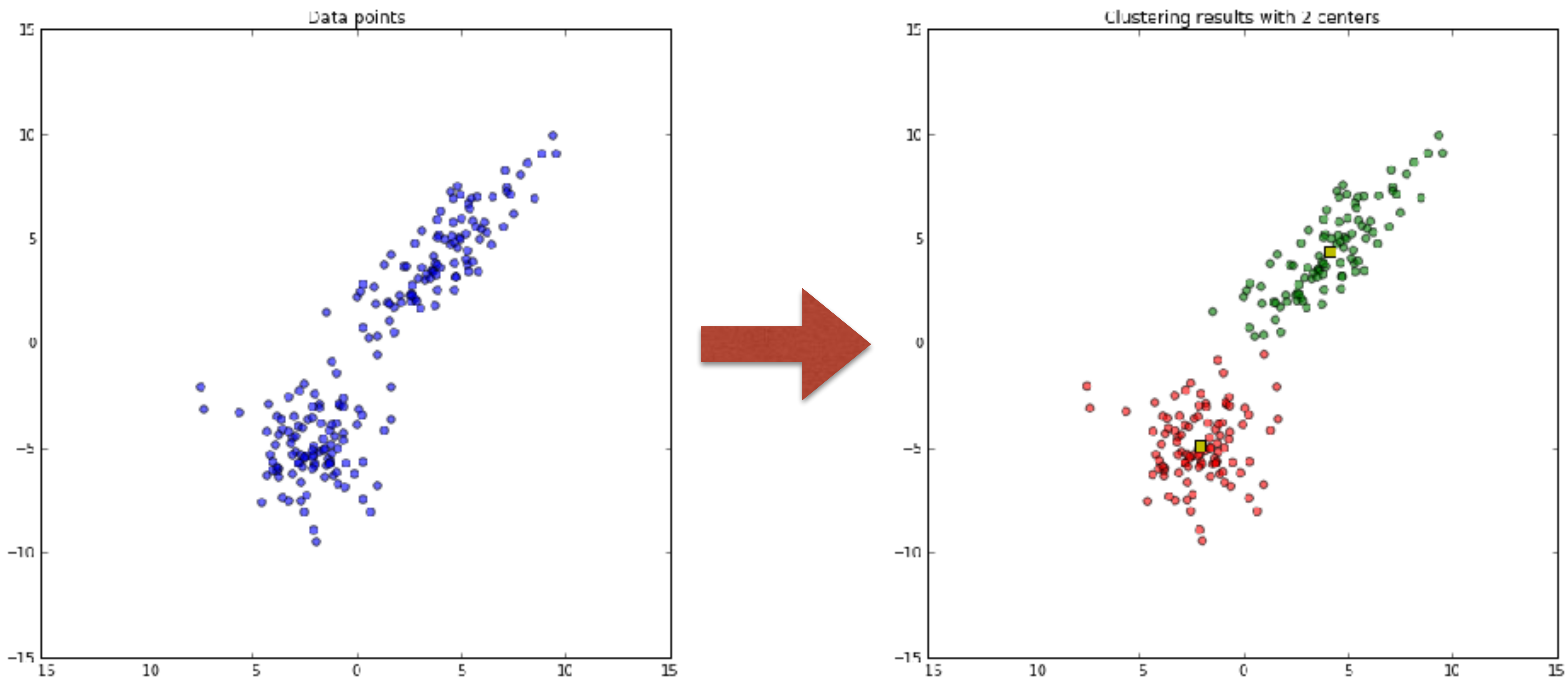
UNIVERSITY OF ONTARIO INSTITUTE OF TECHNOLOGY

Types of Learning

	Supervised learning	**Unsupervised learning**
Discrete	Classification or Categorization	Clustering
Continuous	Regression	Dimensionality reduction

Clustering

- Group together similar points (items), and represent them with a single token



Clustering

- Group together similar points (items), and represent them with a single token
- Key challenges
  - What makes two points/images/patches/items similar?
  - How can we compute an overall grouping from pairwise similarities?

Slide credit: Derek Hoiem

Why perform clustering?

- Summarizing data
- Counting
- Segmentations
- Prediction

- Look at large amounts of data
- Represent high-dimensional vectors with a cluster number

- Histograms (texture, SIFT vectors, color, etc.)

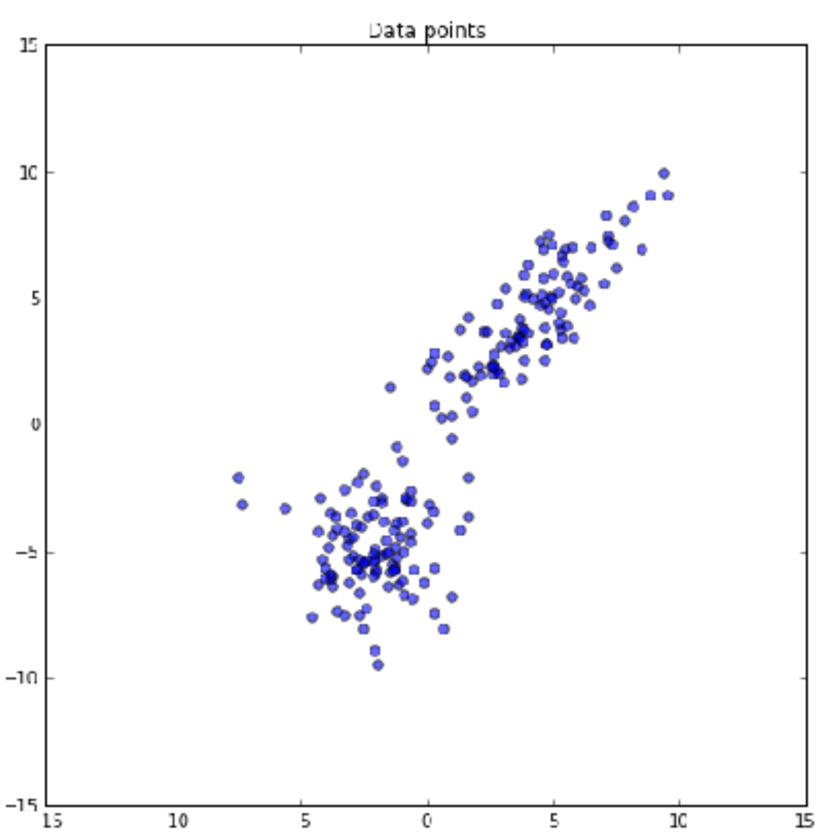
- Separate image into different regions

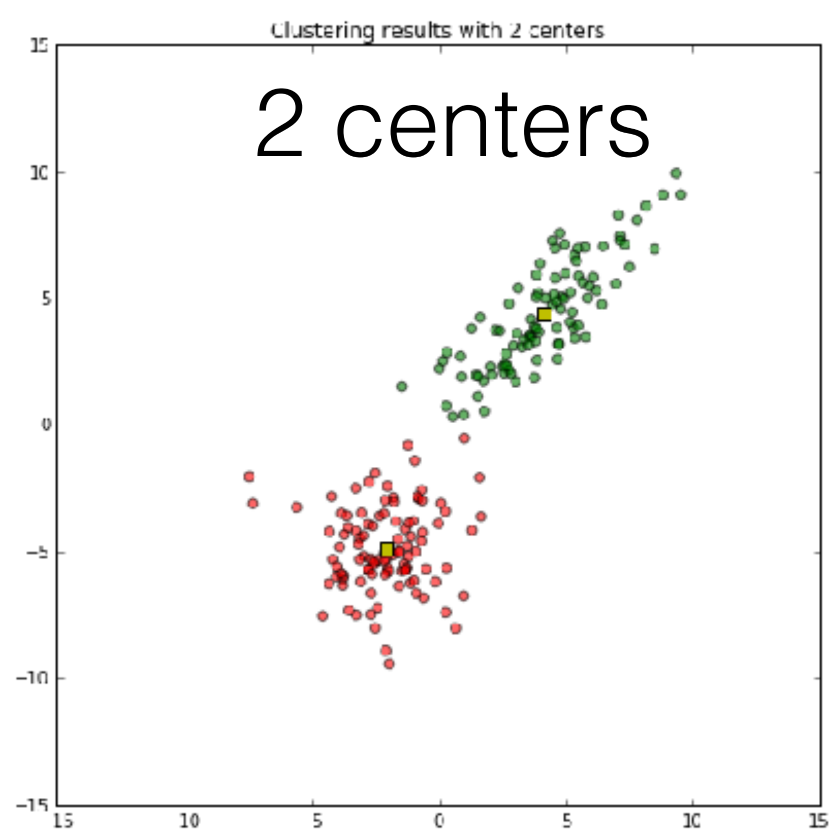
- Images recognition: image in the same cluster may have the same labels

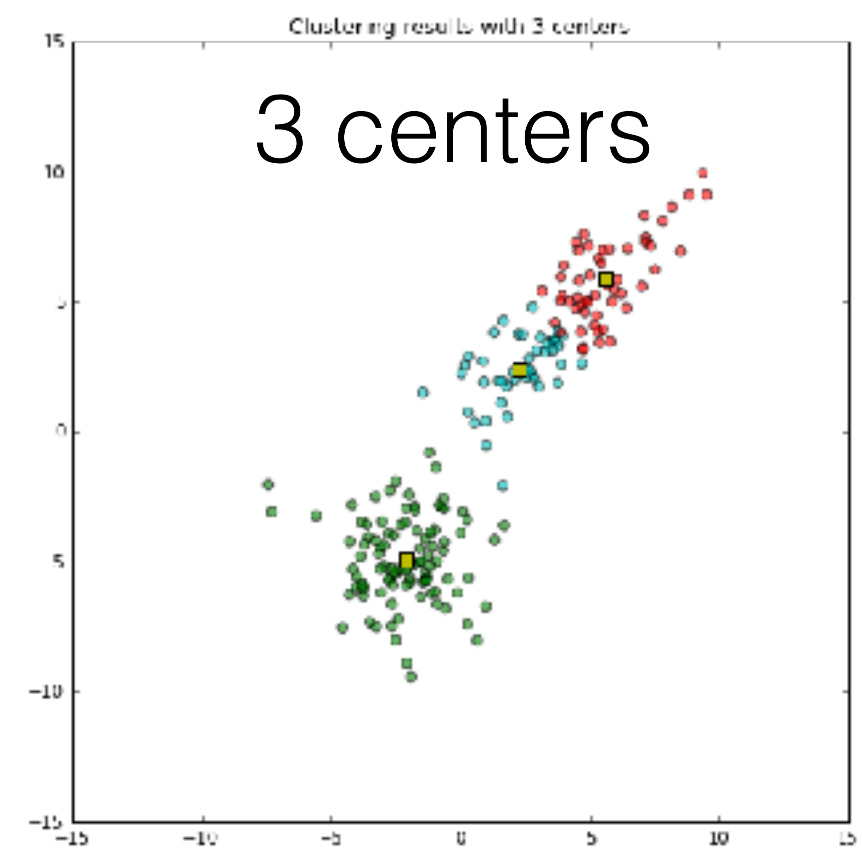
Clustering methods

- K-means
- Agglomerative clustering
- Mean-shift clustering
- Spectral clustering

K-means Clustering







K-means Clustering

- Objective: cluster to minimize variance in data given clusters
  - Preserve information

Given n data points: $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_n$

Find k cluster centers

cluster center i

$$\mathbf{c}^*, \delta^* = \underset{\mathbf{c},\delta}{\operatorname{argmin}} \frac{1}{n} \sum_j^n \sum_i^k \delta_{ij} \left(\mathbf{c}_i - \mathbf{x}_j\right)^2$$

data point j

whether or not data point j is assigned to cluster i

K-means Clustering

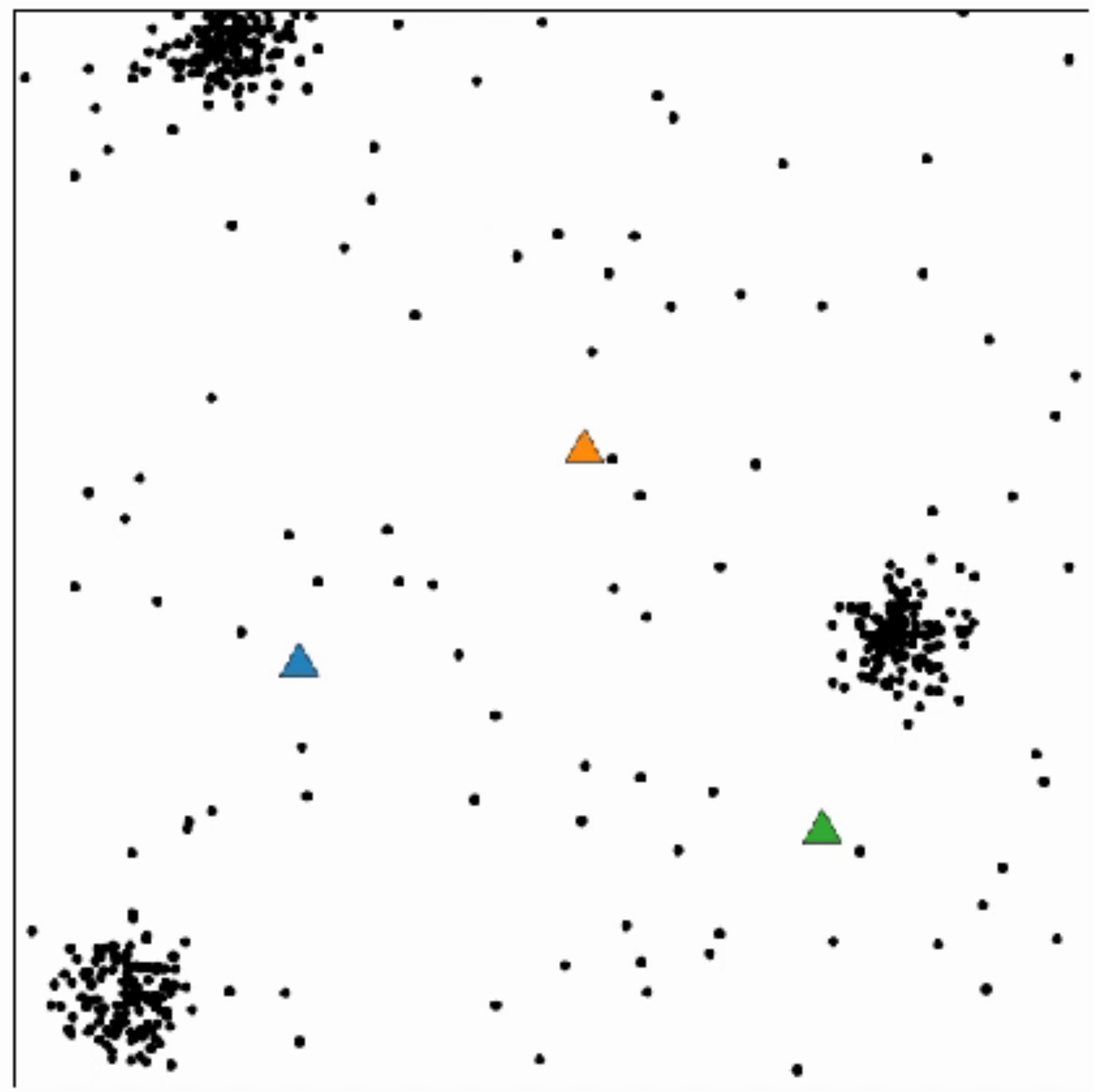
Algorithm

Initialize (randomly) centroids

Find closest centroid to each point. Group points that share the same centroid

Update each centroid to be the mean of the points in its group

Loop until convergence (number of iterations reached or centroids don't move)



http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html

K-means Clustering

1. Initialize cluster centers at time t=0: $\mathbf{c}^0$

2. Assign each point to the closest center

$$\delta^t = \underset{\delta}{\operatorname{argmin}} \frac{1}{n} \sum_j^n \sum_i^k \delta_{ij} \left(\mathbf{c}_i^{t-1} - \mathbf{x}_j\right)^2$$

3. Update the cluster enters as the mean of the points that belong to it

$$\mathbf{c}^t = \underset{\mathbf{c}}{\operatorname{argmin}} \frac{1}{n} \sum_j^n \sum_i^k \delta_{ij}^t \left(\mathbf{c}_i - \mathbf{x}_j\right)^2$$

4. Repeat steps 2 and 3, until convergence is achieved

K-means Clustering

- Initialization
  - Randomly select k points as initial cluster centers
  - Greedily select k points to minimize residual
  - What if a cluster center sits on a data point?
- Distance/similarity measures
  - Euclidean, others …
- Optimization
  - Cannot guarantee that it will converge to *global minima*
  - Multiple restarts
- Choice of K?

Image Segmentation

K-means clustering using intensity or color

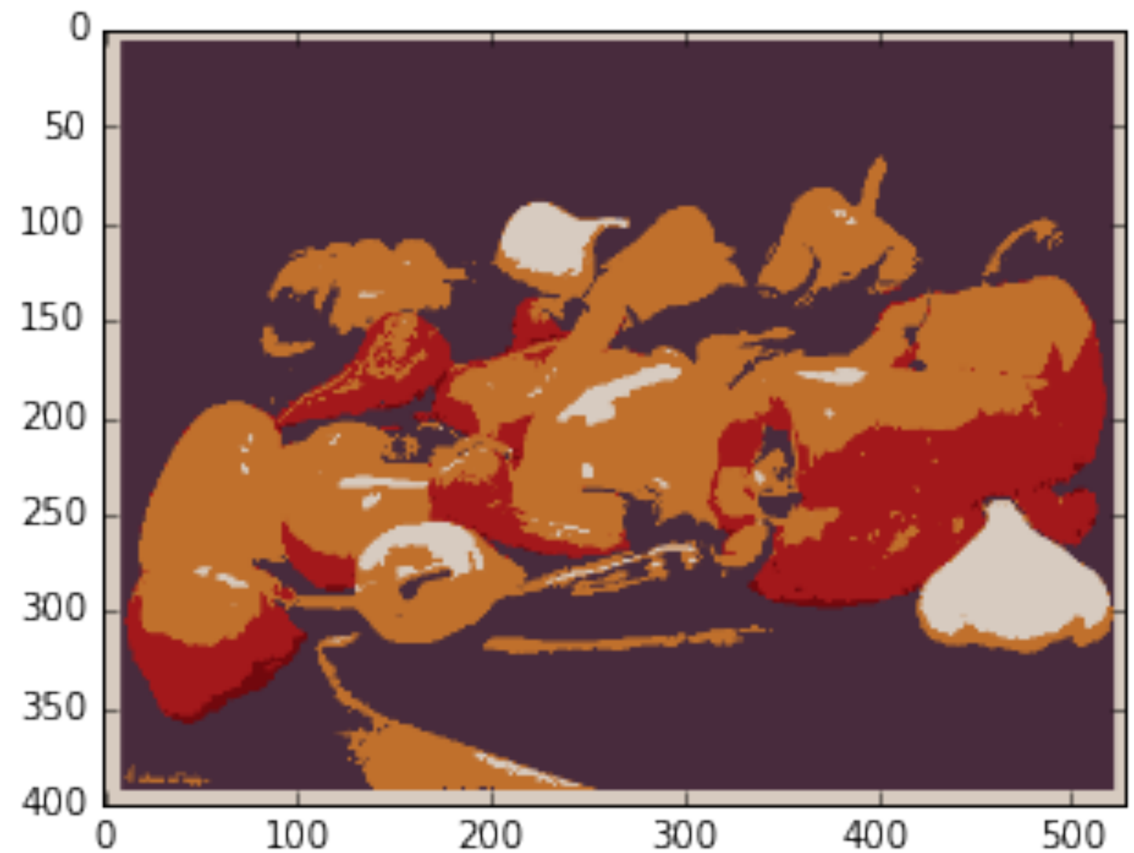
Image

Clusters on intensity

Clusters on color

Image Segmentation

Image



Each pixel is replaced by its cluster centre. The number of cluster is set to 5. Using RGB values.

K-means Clustering

- Pros
  - Find cluster centres that are good representation of data (reduces conditional variance)
  - Simple, fast* and easy to implement
- Cons
  - Need to select the number of clusters
  - Sensitive to outliers
  - Can get stuck in local minima
  - All clusters have the same parameters, i.e., distance/similarity measure is non-adaptive
  - *Each iteration is O(knd) for n, d-dimensional points, so it can be slow
- K-means is rarely used to image segmentation (pixel segmentation)

Commonly used distance/ similarity measures

- P-norms
  - City block (L1)
  - Euclidean (L2)
  - L-infinity

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}.$$

$$\|\boldsymbol{x}\|_1 := \sum_{i=1}^{n} |x_i|.$$

$$\|\boldsymbol{x}\| := \sqrt{x_1^2 + \cdots + x_n^2}.$$

$$\|\mathbf{x}\|_\infty := \max\left(|x_1|, \ldots, |x_n|\right).$$

Here x_i is the distance between two points

- Mahalanobis distance
  - Scaled Euclidean

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{\sigma_i^2}},$$

- Cosine Distance

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Slide credit: James Hayes

How many cluster centers?

- Validation set
  - Try different numbers of clusters and look at performance

Evaluating Clusters

- Generative
  - How well are points reconstructed from the clusters?
- Discriminative
  - How well do the clusters correspond to labels? This is often termed as *purity*.
  - Unsupervised clustering doesn’t aim to be discriminative

K-mediods Clustering

- Similar to K-means
  - Represent a cluster center with one of its members (data points), rather than the mean of its members
  - Choose the member (data point) that minimizes cluster similarity
- Applicable in situations where *mean* is not meaningful
  - Clustering hue values
  - Using L-infinity norm for similarity

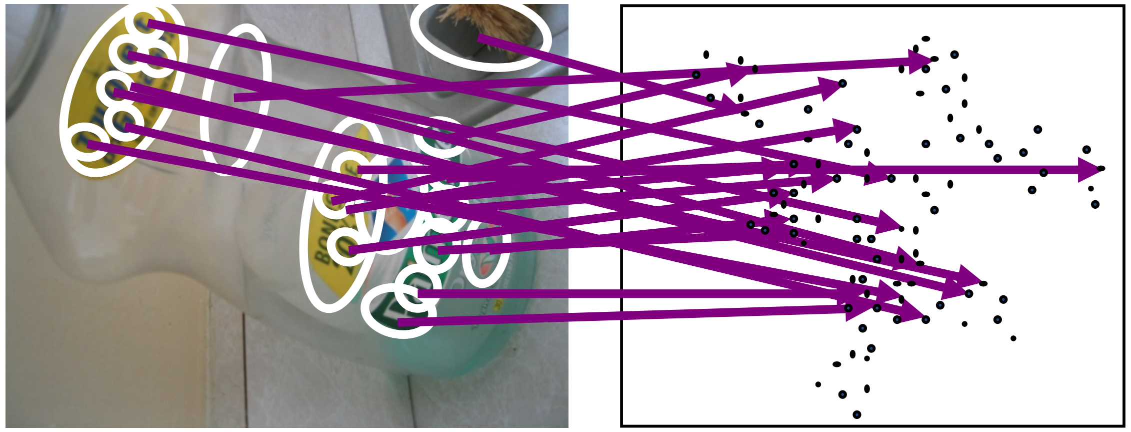
Slide credit: James Hayes

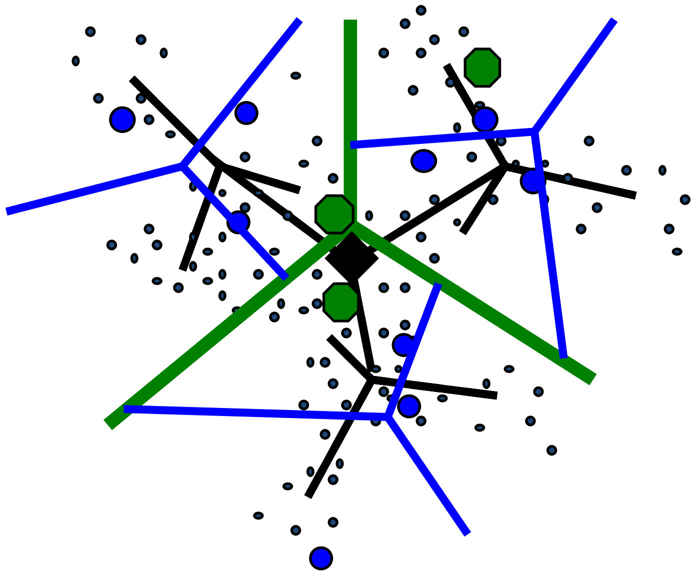
Building Visual Dictionaries

- Sample patches from a database
  - E.g., 128-dimensional SIFT features

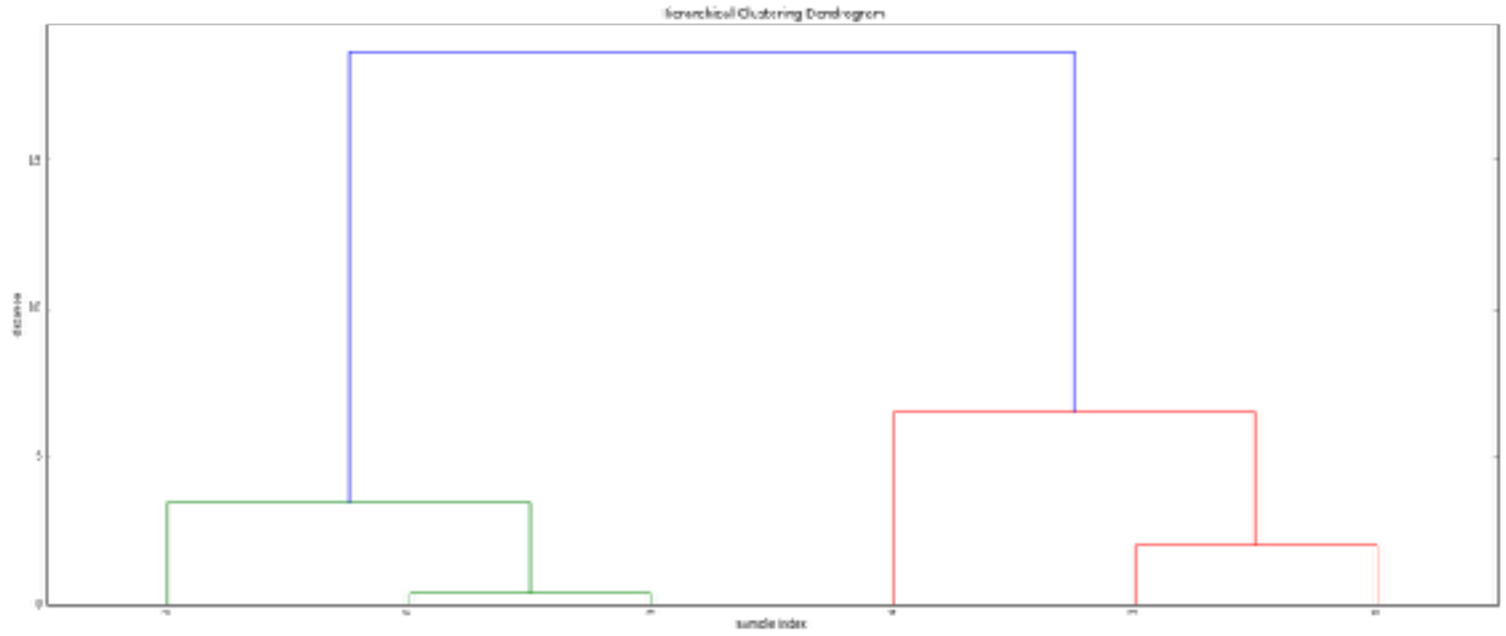
- Cluster these patches
  - Clusters centers comprise (visual) dictionary

- Assign a codeword (number, cluster center) to each new patch (say 128-dimensional SIFT feature) according to the nearest cluster

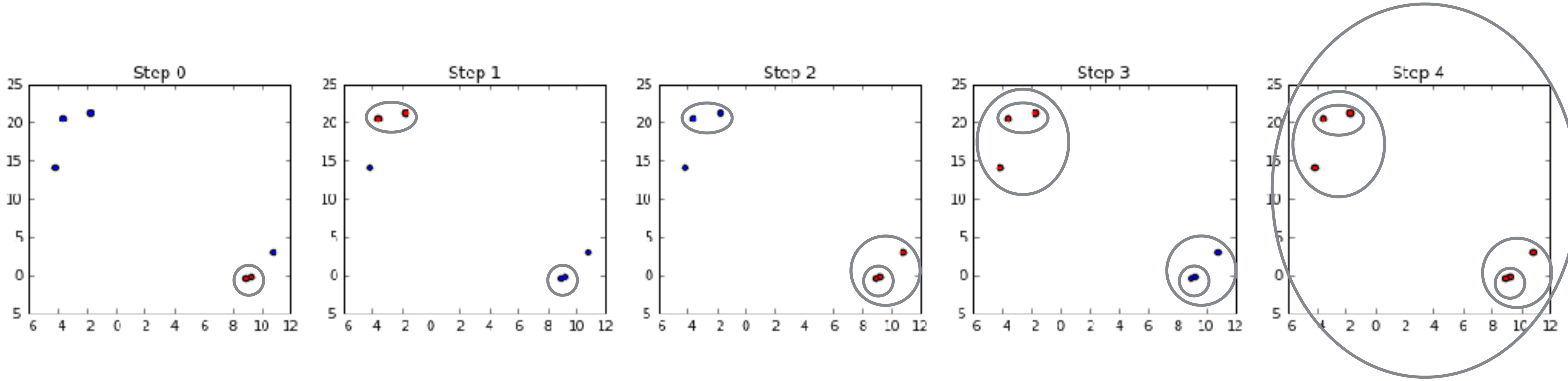




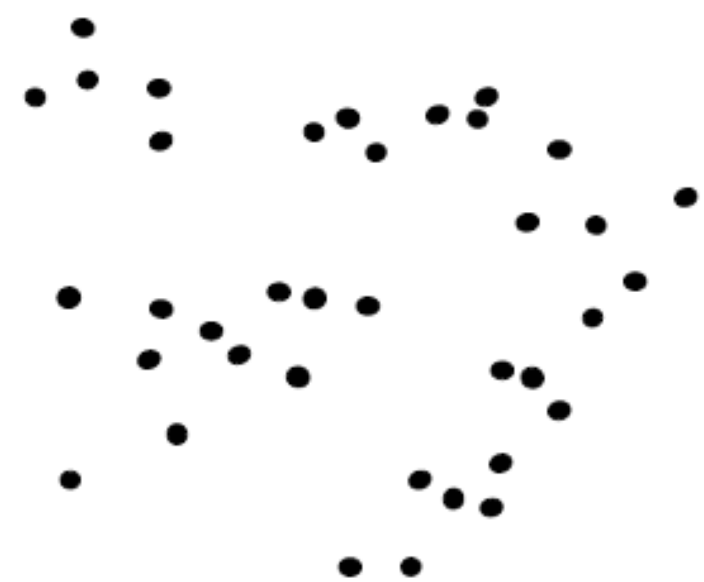
Slide credit: James Hayes



Agglomerative Clustering



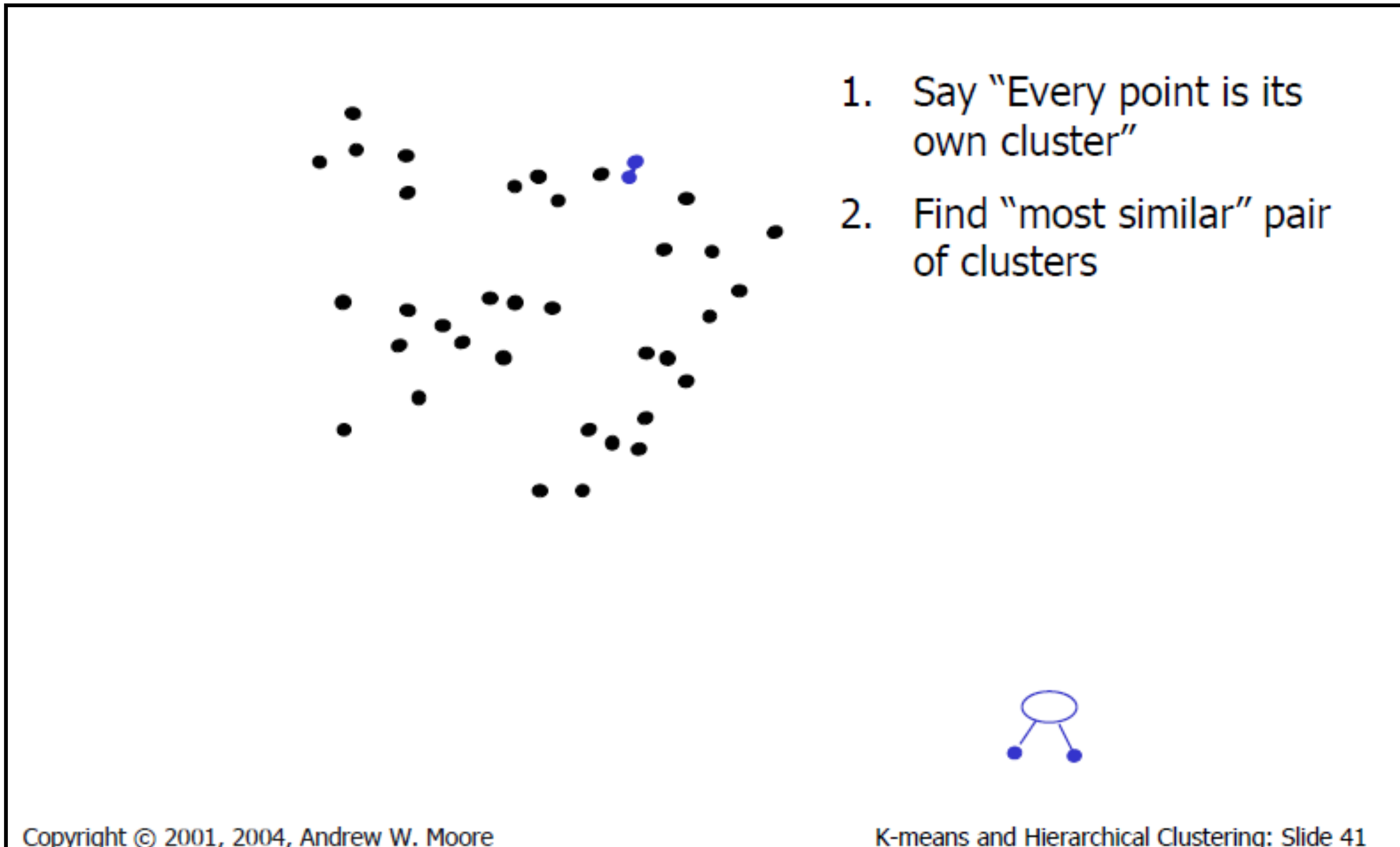
Agglomerative Clustering



1. Say "Every point is its own cluster"

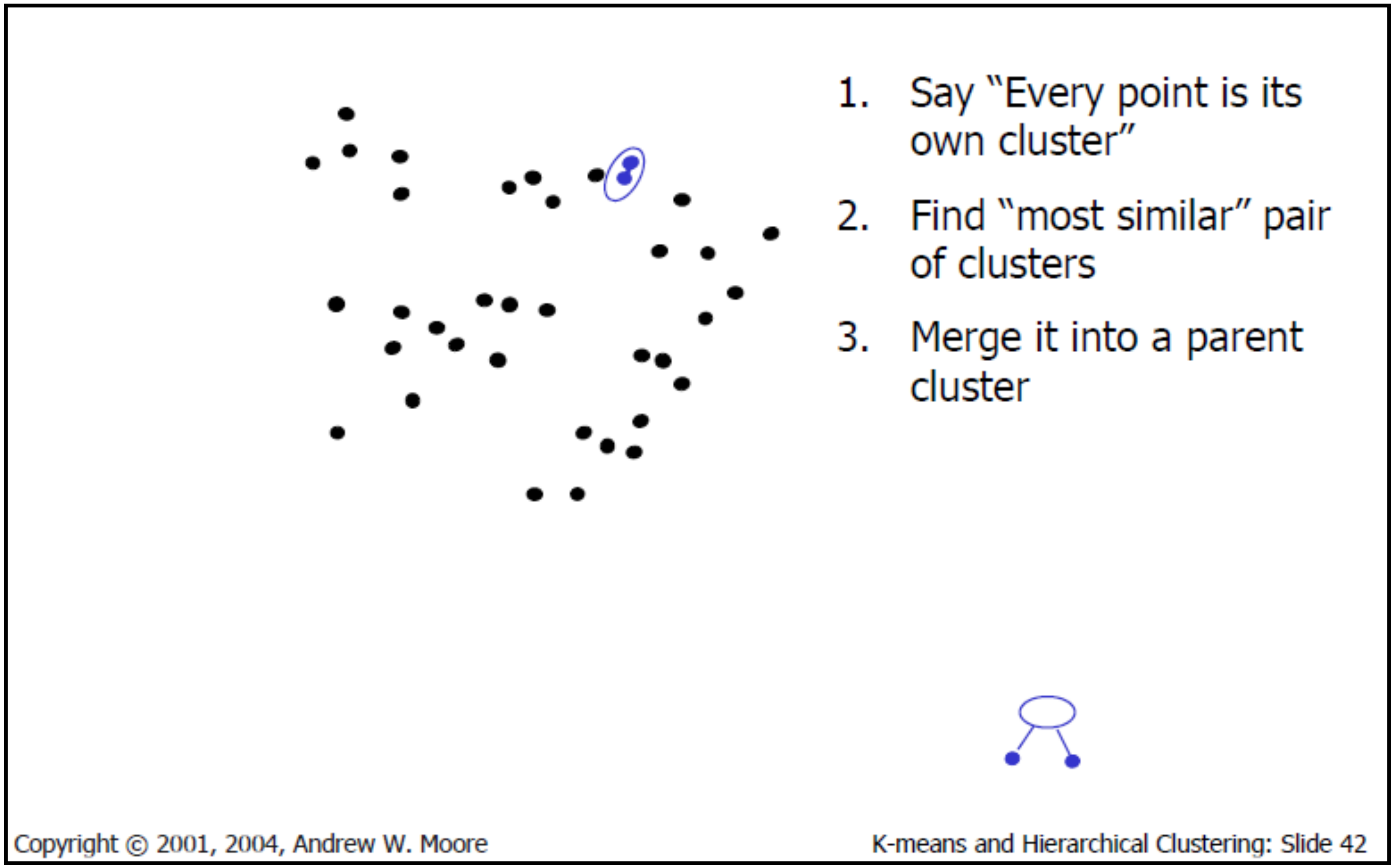
Slide credit: James Hayes

Agglomerative Clustering



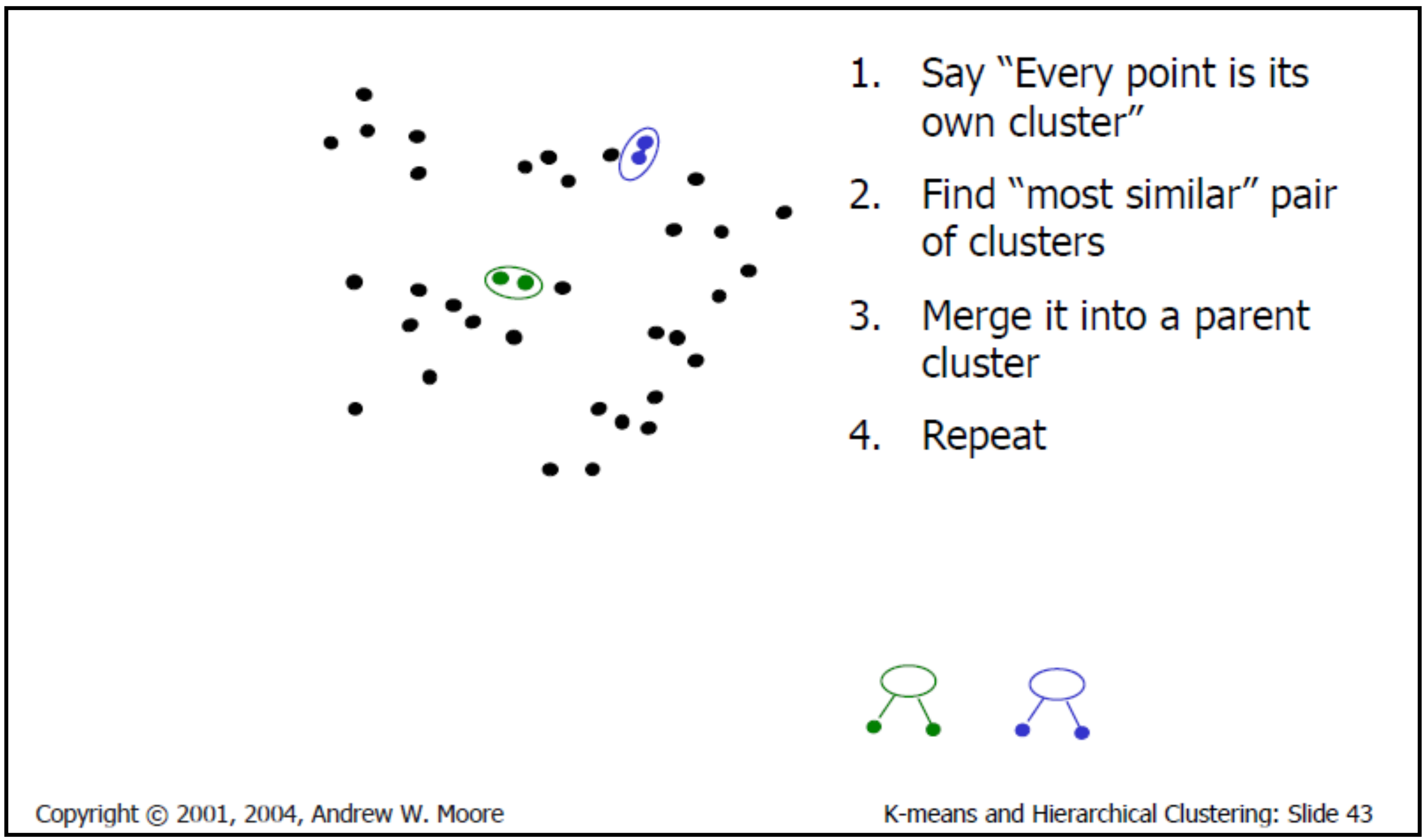
Slide credit: James Hayes

Agglomerative Clustering



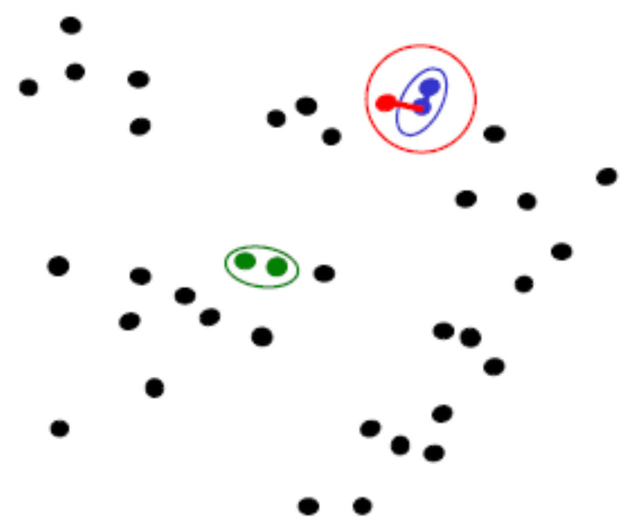
Slide credit: James Hayes

Agglomerative Clustering

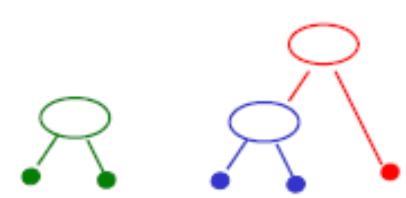


Slide credit: James Hayes

Agglomerative Clustering



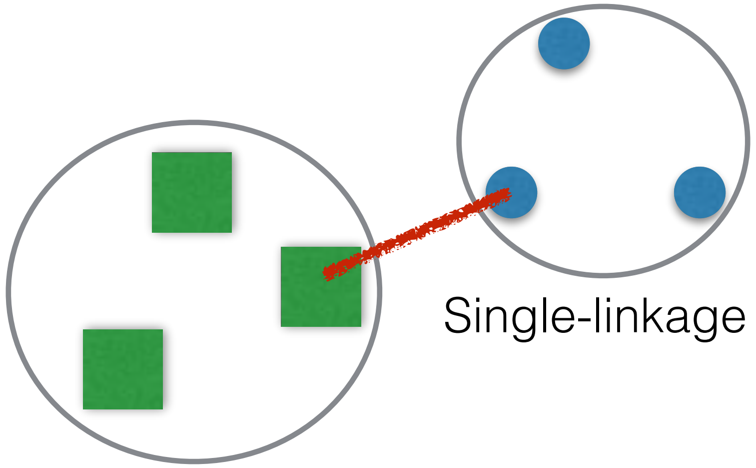
1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat

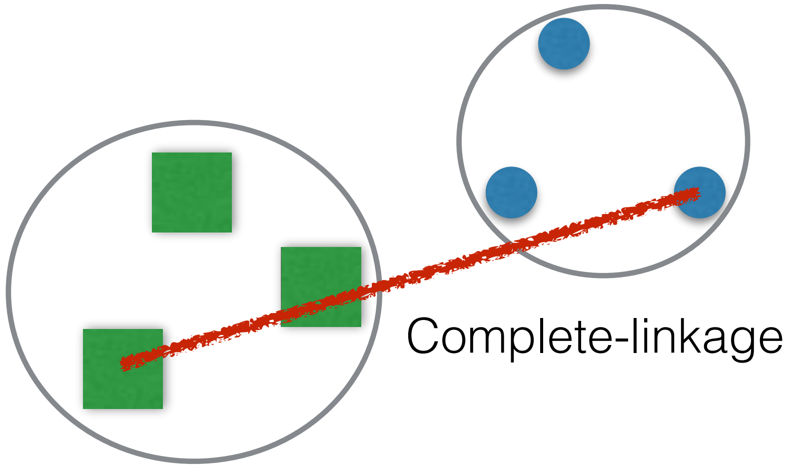


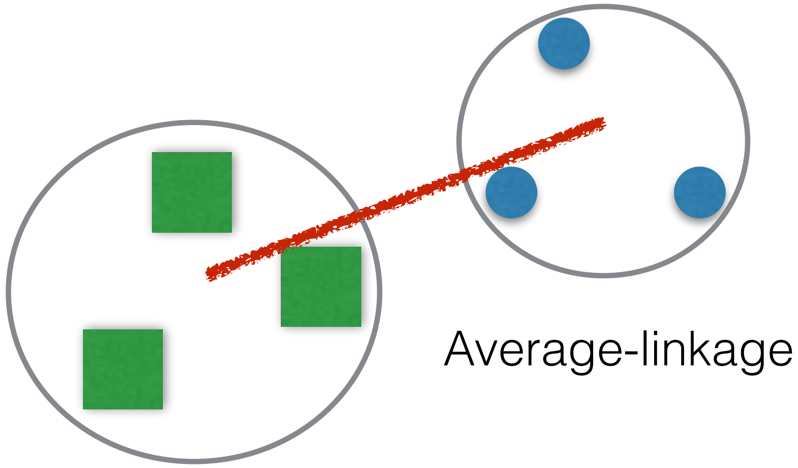
Slide credit: James Hayes

Agglomerative Clustering: Defining Cluster Similarity

- Single-linkage clustering (also called the connectedness or minimum method),
    - We consider the distance between one cluster and another cluster to be equal to the shortest distance from any member of one cluster to any member of the other cluster.
    - If the data consist of similarities, we consider the similarity between one cluster and another cluster to be equal to the greatest similarity from any member of one cluster to any member of the other cluster.
- Complete-linkage clustering (also called the diameter or maximum method)
    - We consider the distance between one cluster and another cluster to be equal to the greatest distance from any member of one cluster to any member of the other cluster.
- Average-linkage clustering
    - We consider the distance between one cluster and another cluster to be equal to the average distance from any member of one cluster to any member of the other cluster.
    - A variation on average-link clustering uses the median distance, which is much more outlier-proof than the average distance.

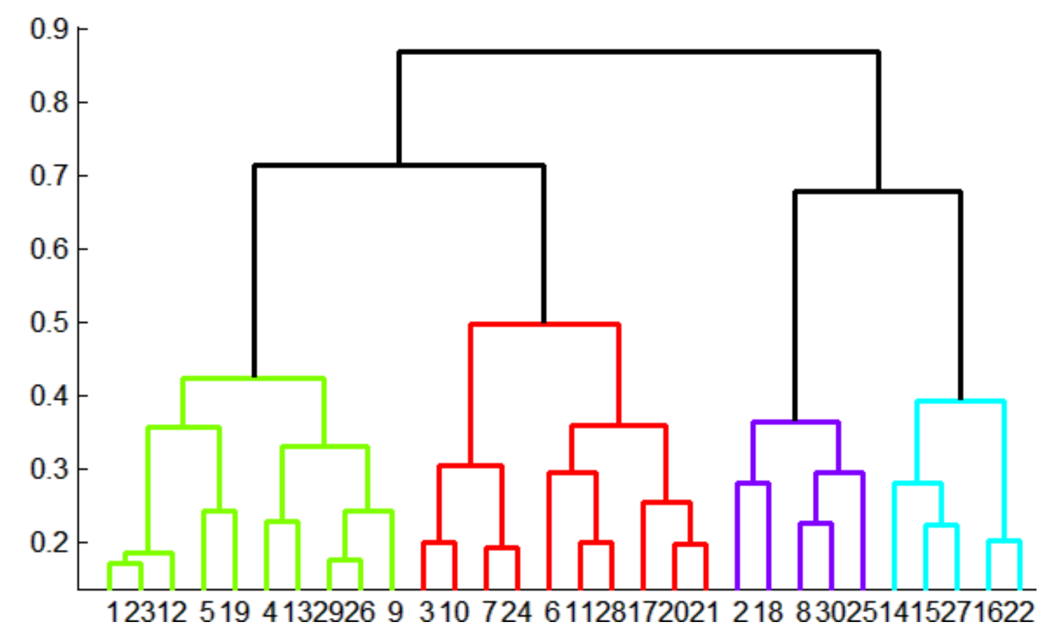






Agglomerative Clustering

- How many clusters?
  - Agglomerative clustering creates a tree (commonly referred to as a *dendrogram*)
  - Threshold based upon the maximum number of clusters
  - Threshold based upon distance of merges



dendrogram

Agglomerative Clustering

Single-linkage clustering (Johnson's algorithms)

1. *Begin with the disjoint clustering having level L(0) = 0 and sequence number m = 0.*
2. *Find the least dissimilar pair of clusters in the current clustering, say pair (r), (s), according to*

   d[(r),(s)] = min d[(i),(j)]

   where the minimum is over all pairs of clusters in the current clustering.
3. *Increment the sequence number : m = m +1. Merge clusters (r) and (s) into a single cluster to form the next clustering m. Set the level of this clustering to*

   L(m) = d[(r),(s)]
4. *Update the proximity matrix, D, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted (r,s) and old cluster (k) is defined in this way:*

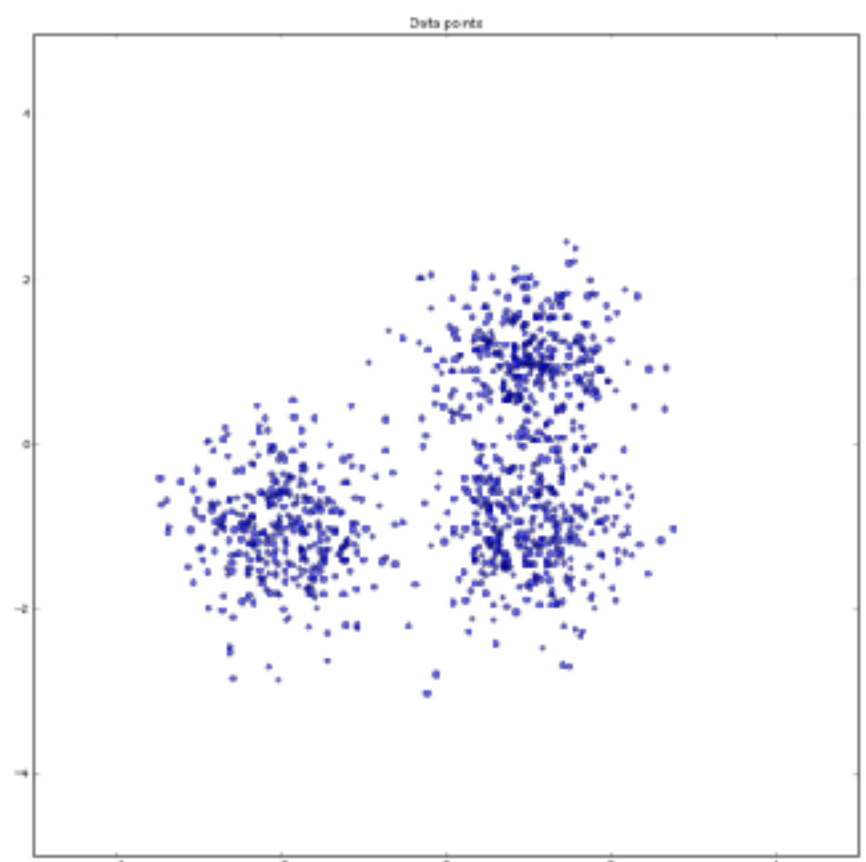
   d[(k), (r,s)] = min d[(k),(r)], d[(k),(s)]
5. *If all objects are in one cluster, stop. Else, go to step 2.*

http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html

Agglomerative Clustering

- Pros
  - Simple to implement
  - Clusters have adaptive shapes
  - Provides a hierarchy of clusters
- Bad
  - These do not scale well. Time complexity is $O(n^2)$
  - May have imbalanced clusters
  - They cannot undo what was done previously
  - Need to choose the number of clusters
  - Needs to use an “ultrametric” to get meaningful hierarchy
    - Ultrametric space is special kind of metric space in which the triangle inequality is replaced with $d(x,z) \leq \max\left(d(x,y), d(y,z)\right)$

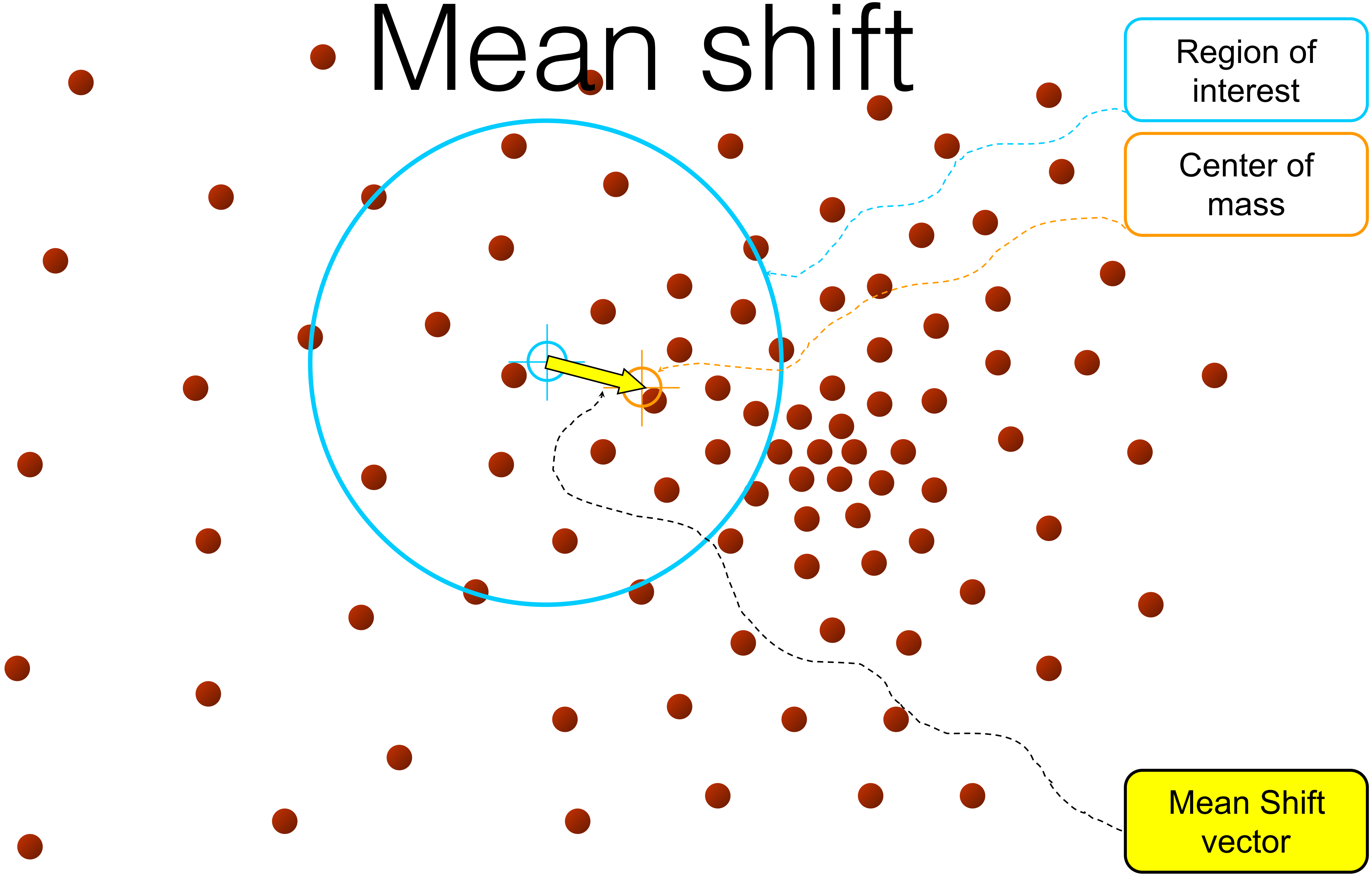
Mean Shift Clustering



Mean shift Clustering

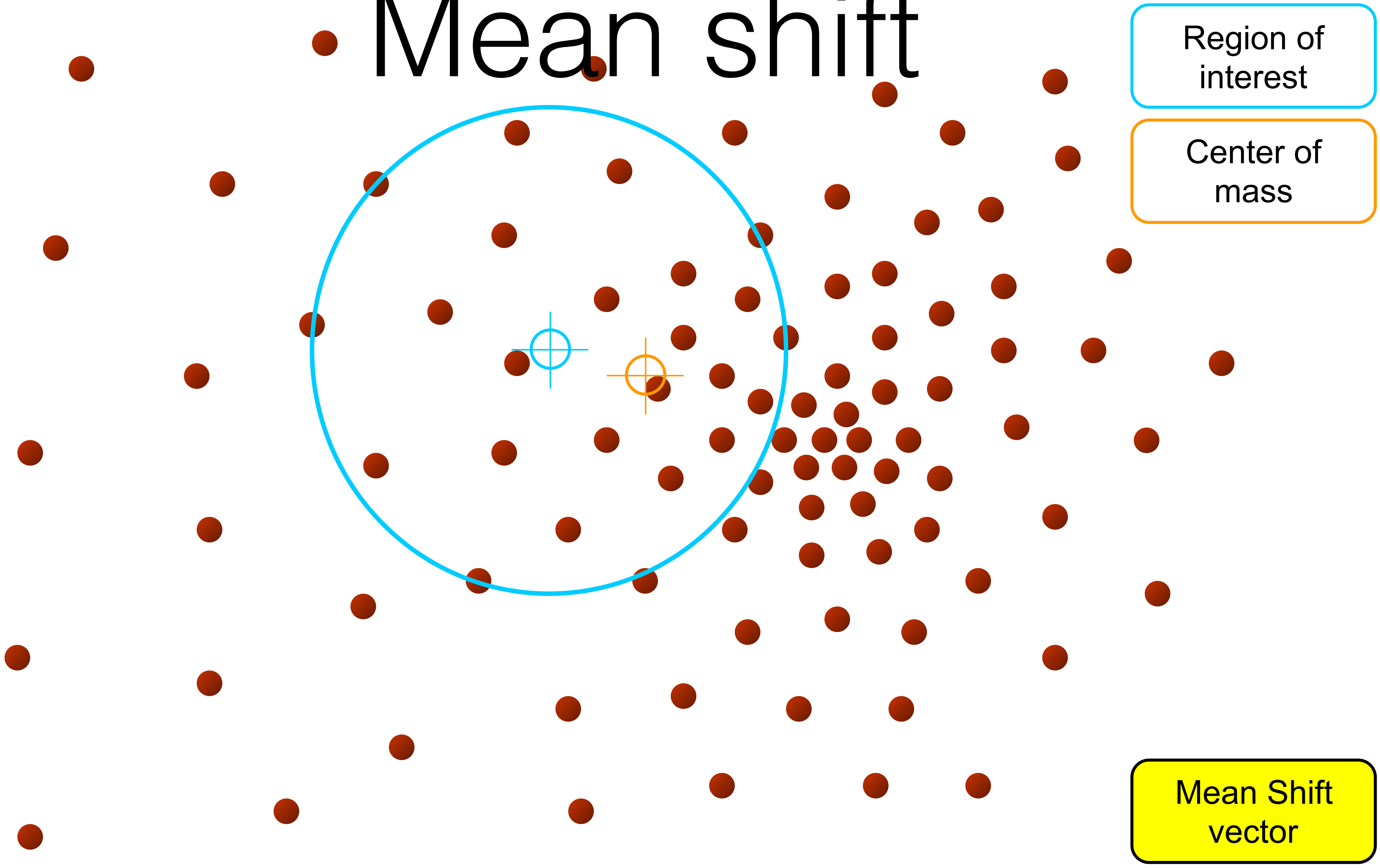
- The mean shift algorithm seeks *modes* of a given set of points
- Algorithm outline
  1. Choose kernel and bandwidth
  2. For each point
     a. Center a window on that point
     b. Compute the mean of the data in the search window
     c. Center the search window at the new mean location
     d. Repeat steps b,c above until convergence
  3. Assign points that lead to nearby modes to the same cluster

Mean shift



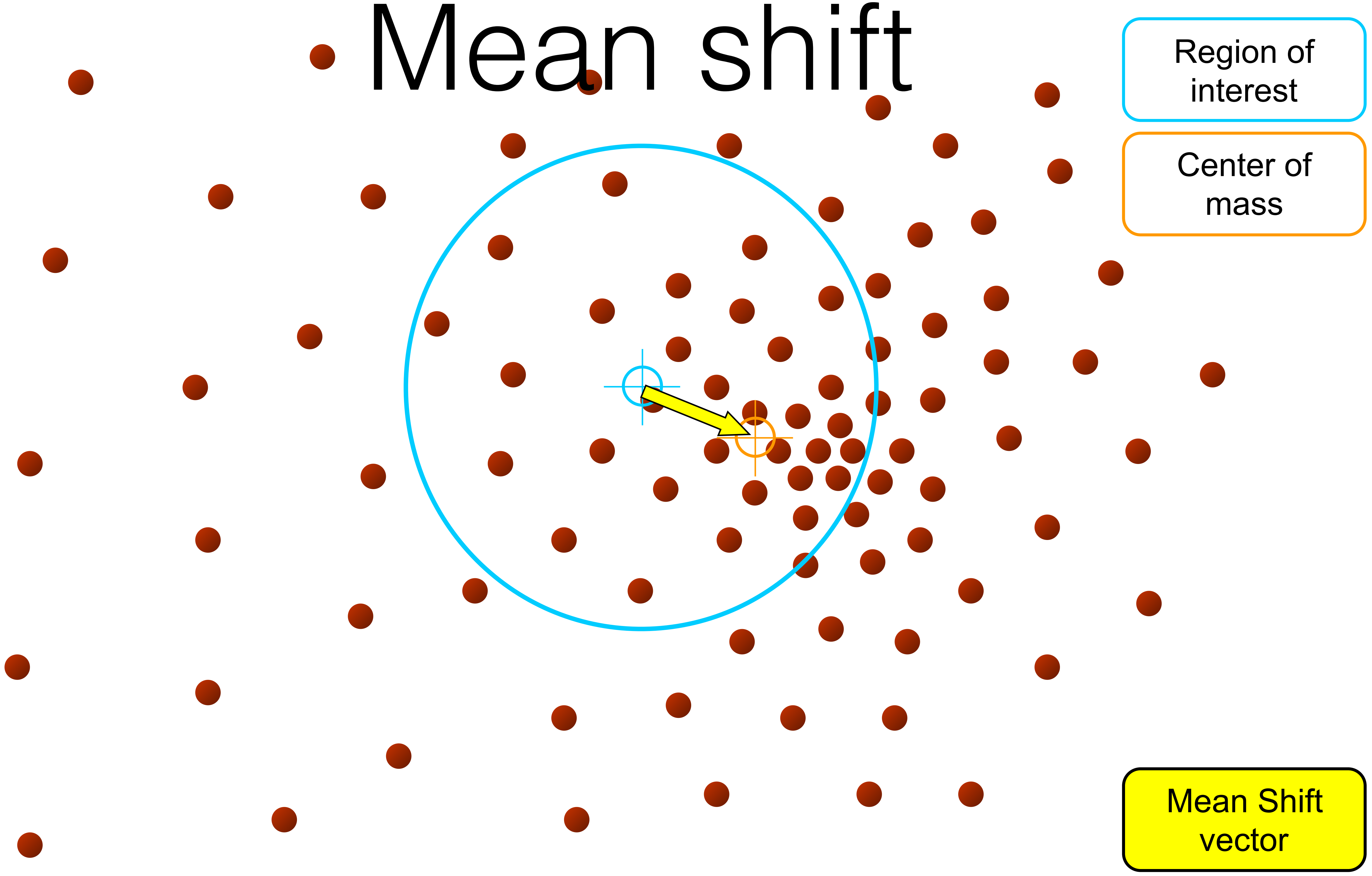
Slide by Y. Ukrainitz & B. Sarel

Mean shift



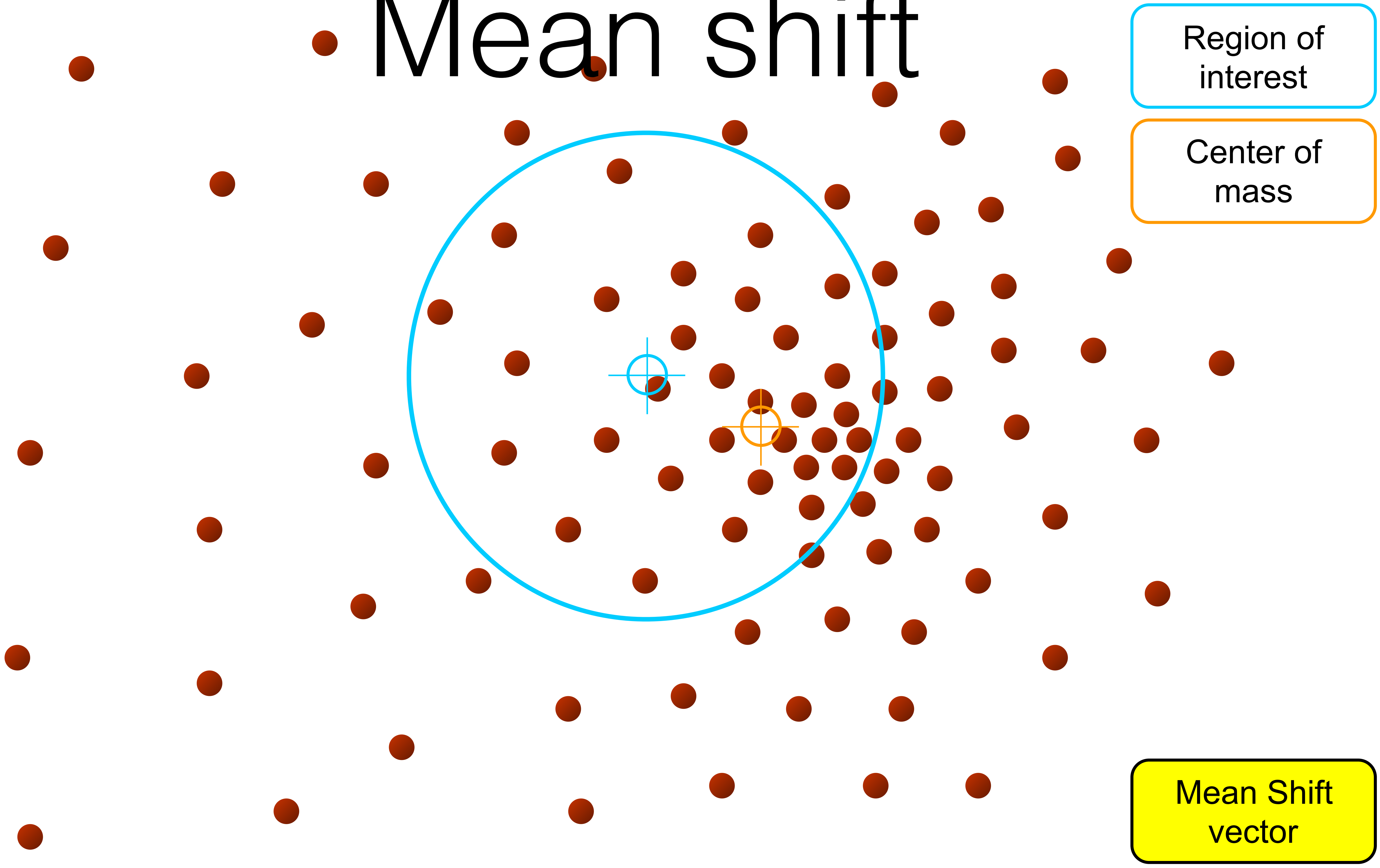
Slide by Y. Ukrainitz & B. Sarel

Mean shift



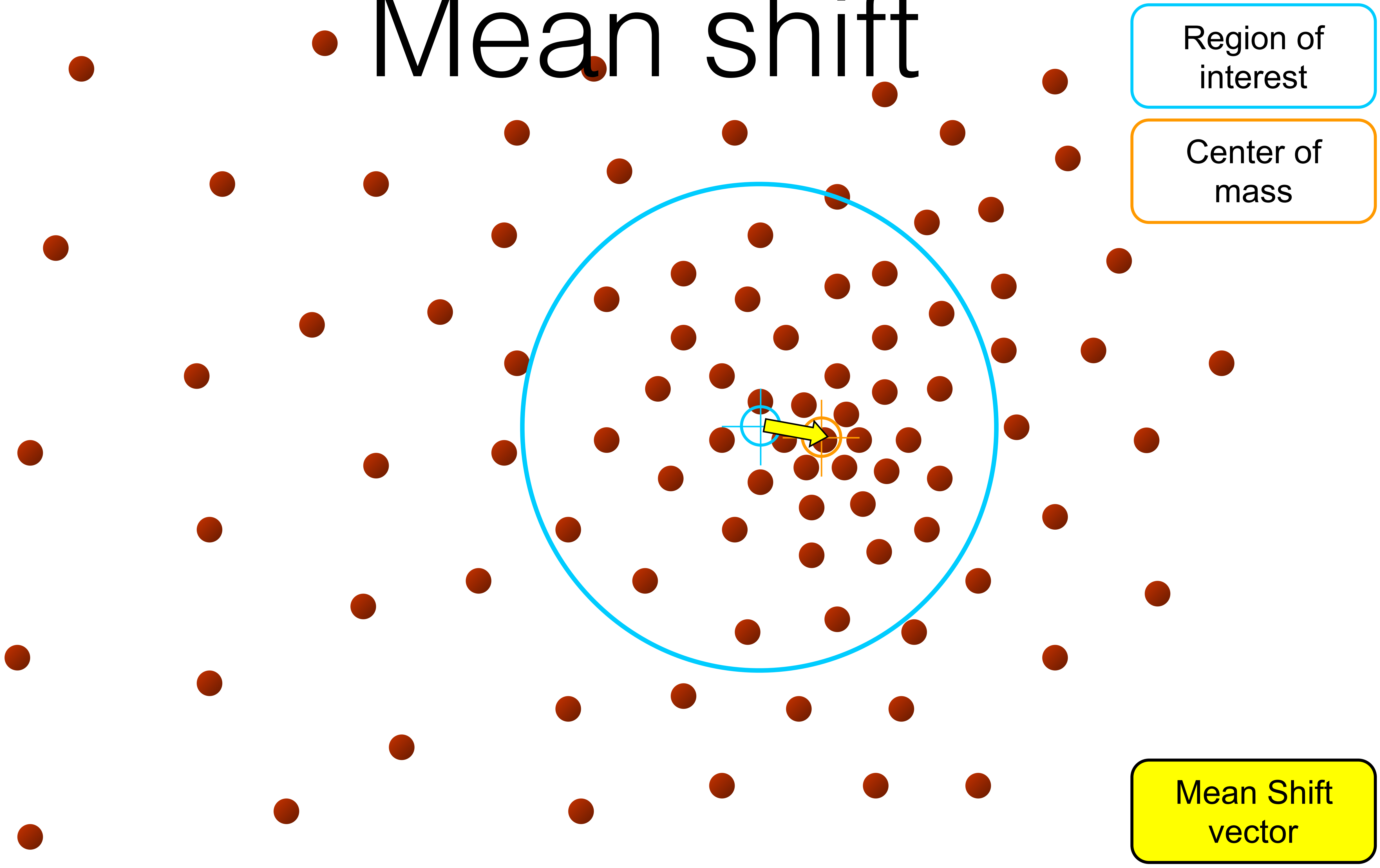
Slide by Y. Ukrainitz & B. Sarel

Mean shift



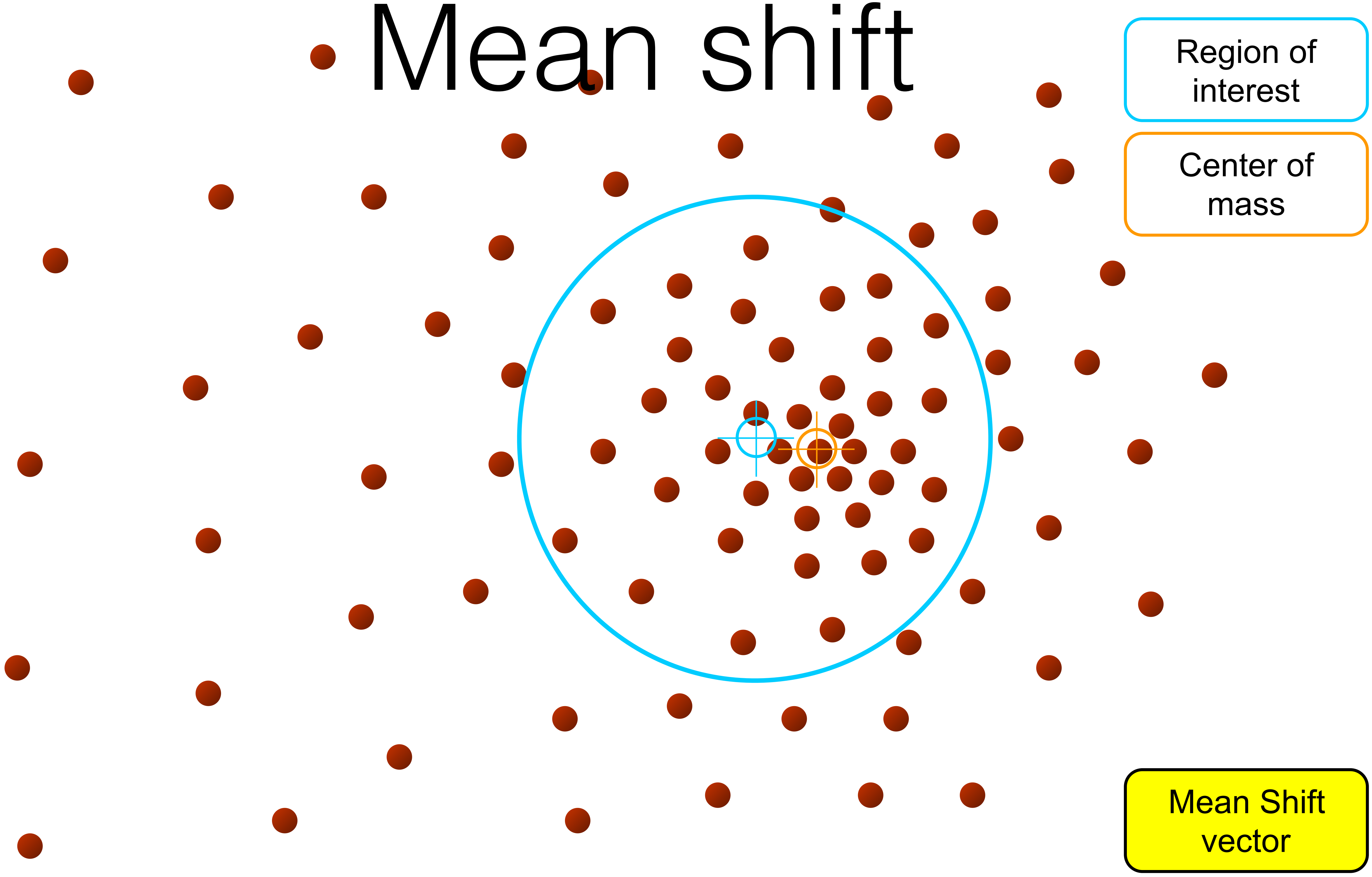
Slide by Y. Ukrainitz & B. Sarel

Mean shift



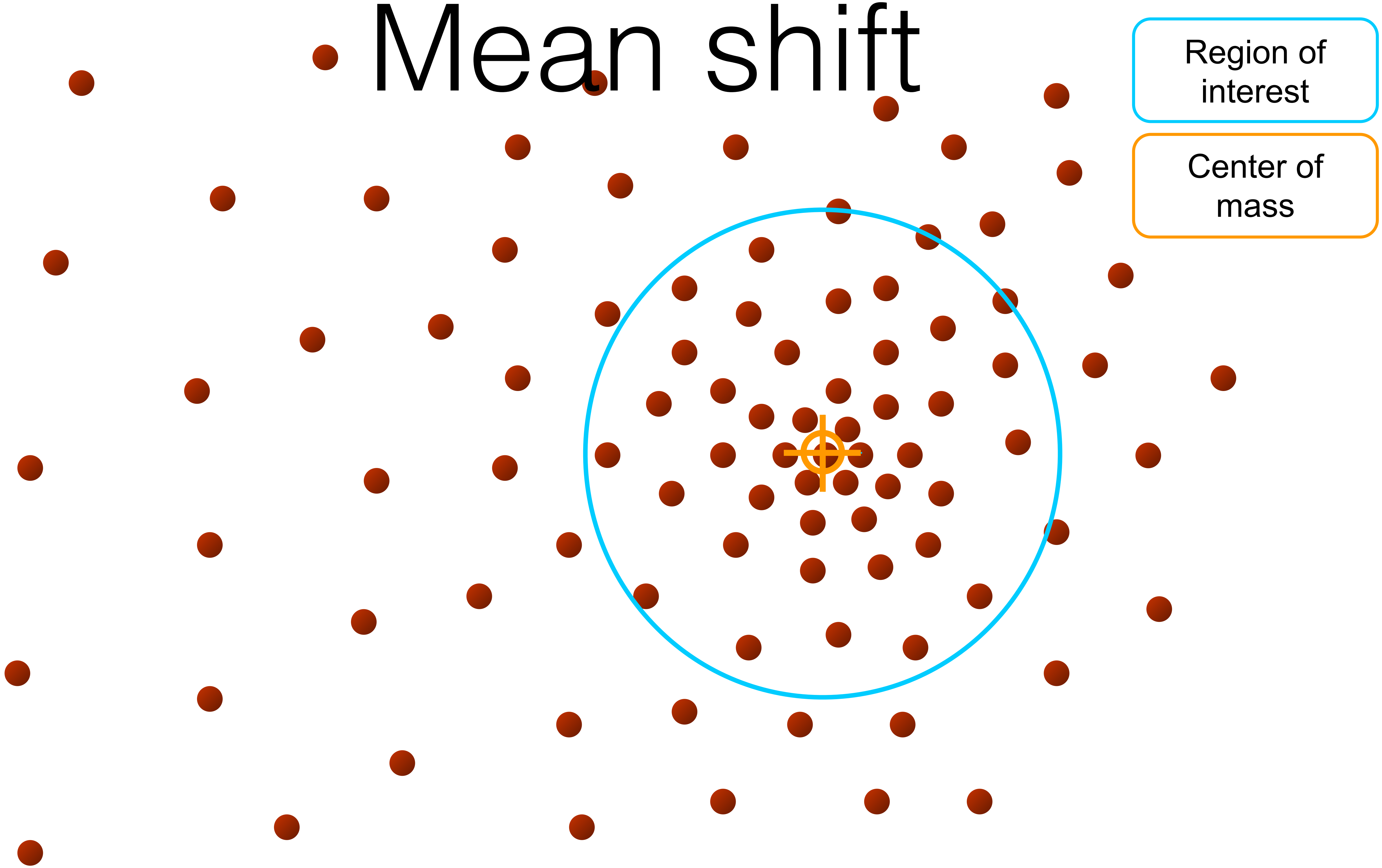
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Kernel density estimation

- Kernel density estimation function

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

- Gaussian kernel

$$K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}-\mathbf{x}_i)}{2h^2}}$$

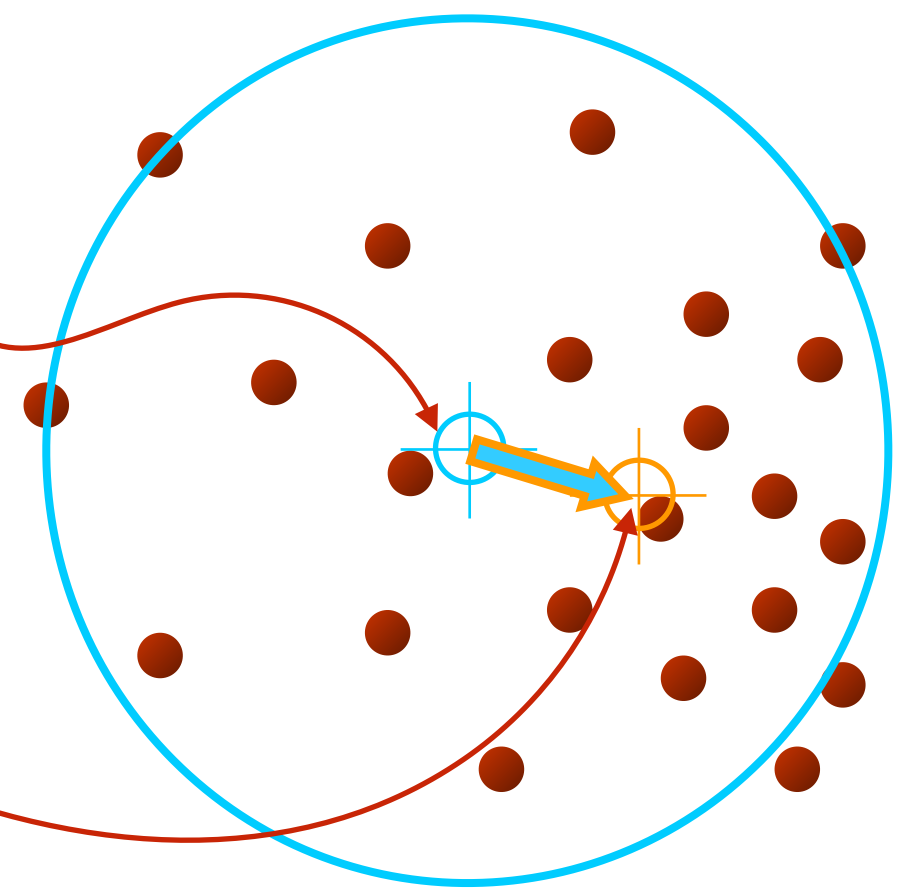
Slide credit: James Hayes

Computing Mean Shift

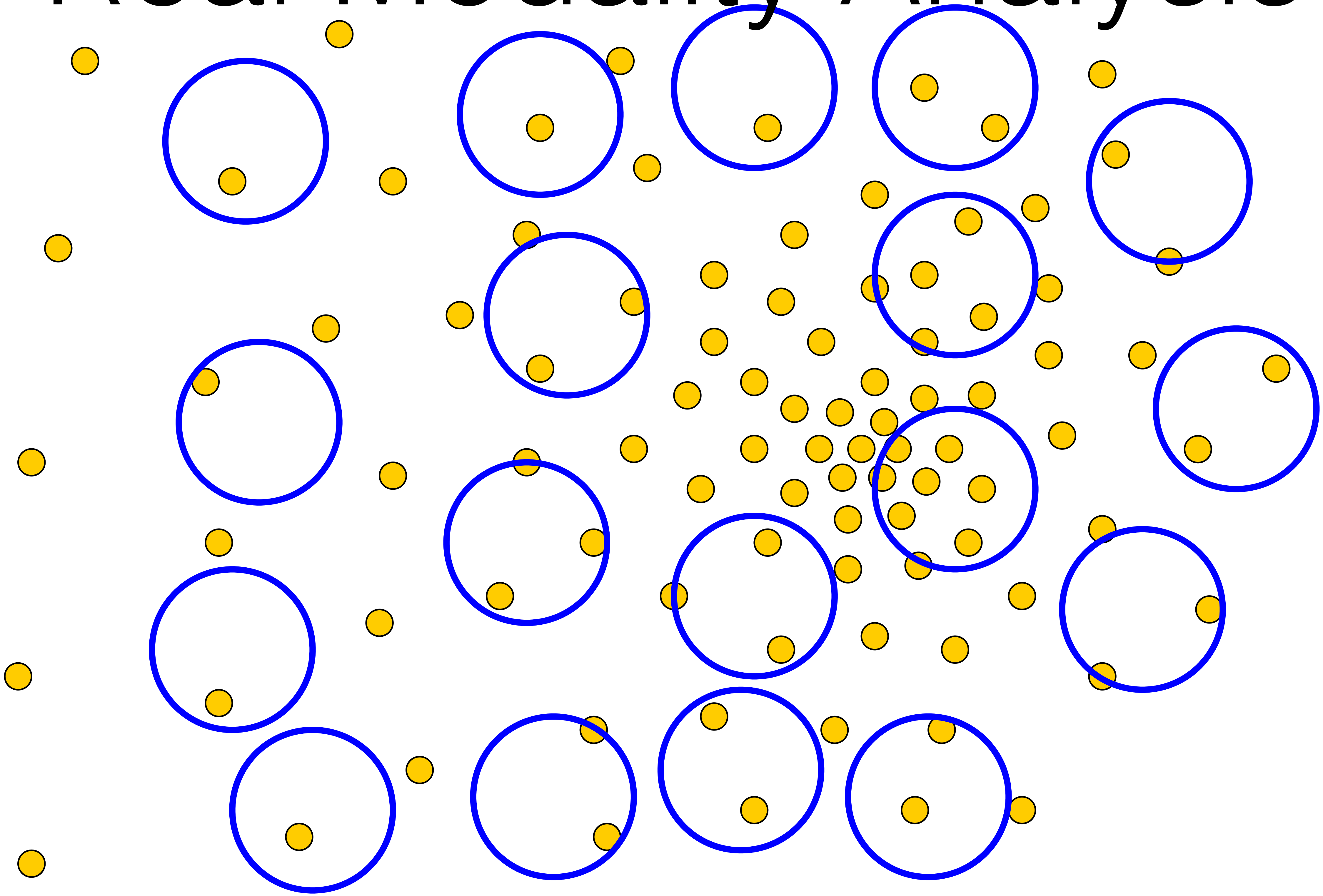
- Compute mean shift vector

- Shift the kernel window

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



Real Modality Analysis

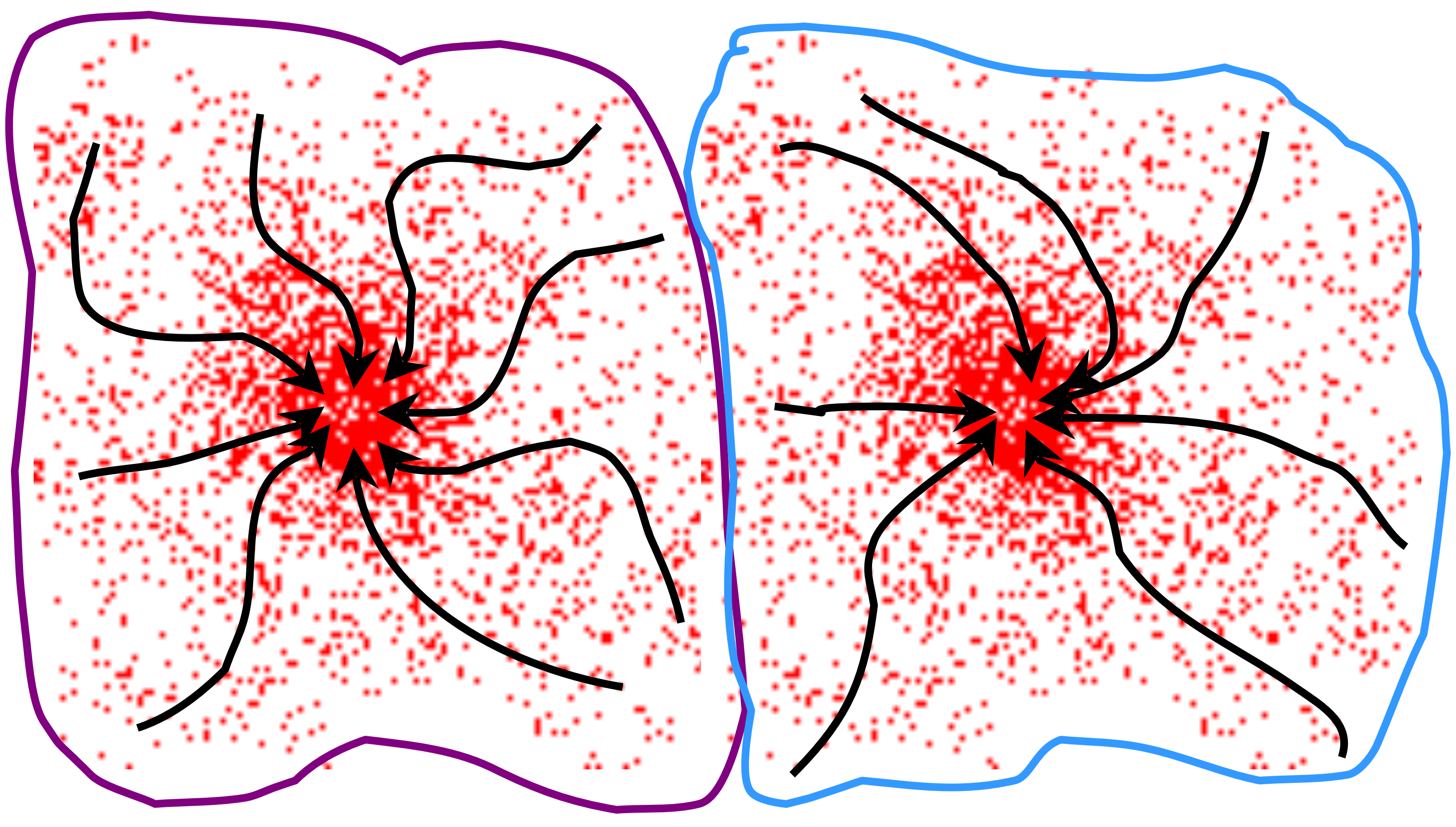


Slide credit: James Hayes

Attraction basin

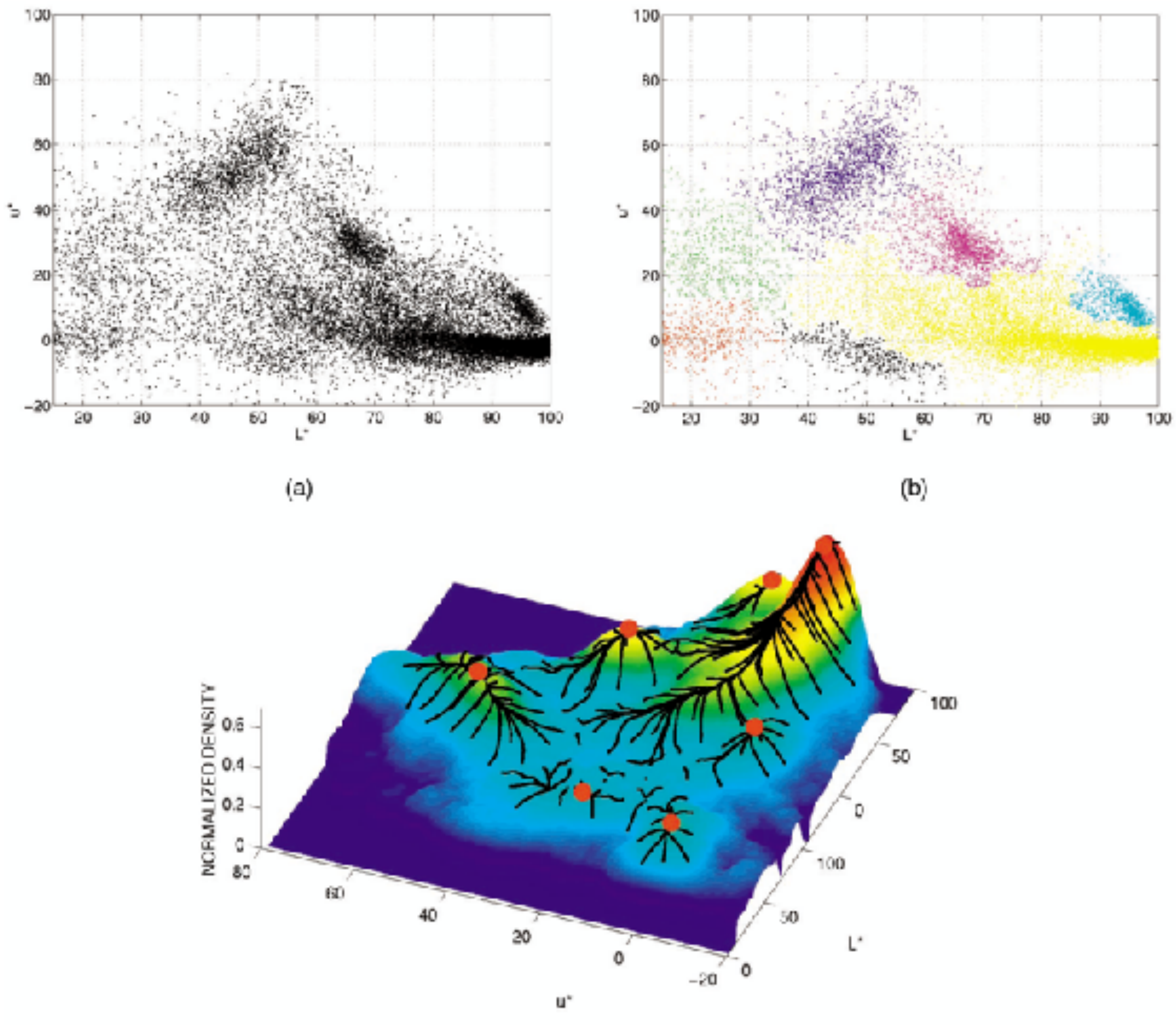
- Attraction basin: the region for which all trajectories lead to the same mode

- Cluster: all data points in the attraction basin of a mode



Slide by Y. Ukrainitz & B. Sarel

Attraction basin



(c)

Slide credit: James Hayes

Image segmentation using Mean Shift

- Compute features for each pixel (color, gradient, texture, etc.)

- Set kernel size for features (K_f) and position (K_s)

- Initialize windows at individual pixel locations

- Perform mean shift for each window until convergence is reached

- Merge windows that are within width of K_f and K_s

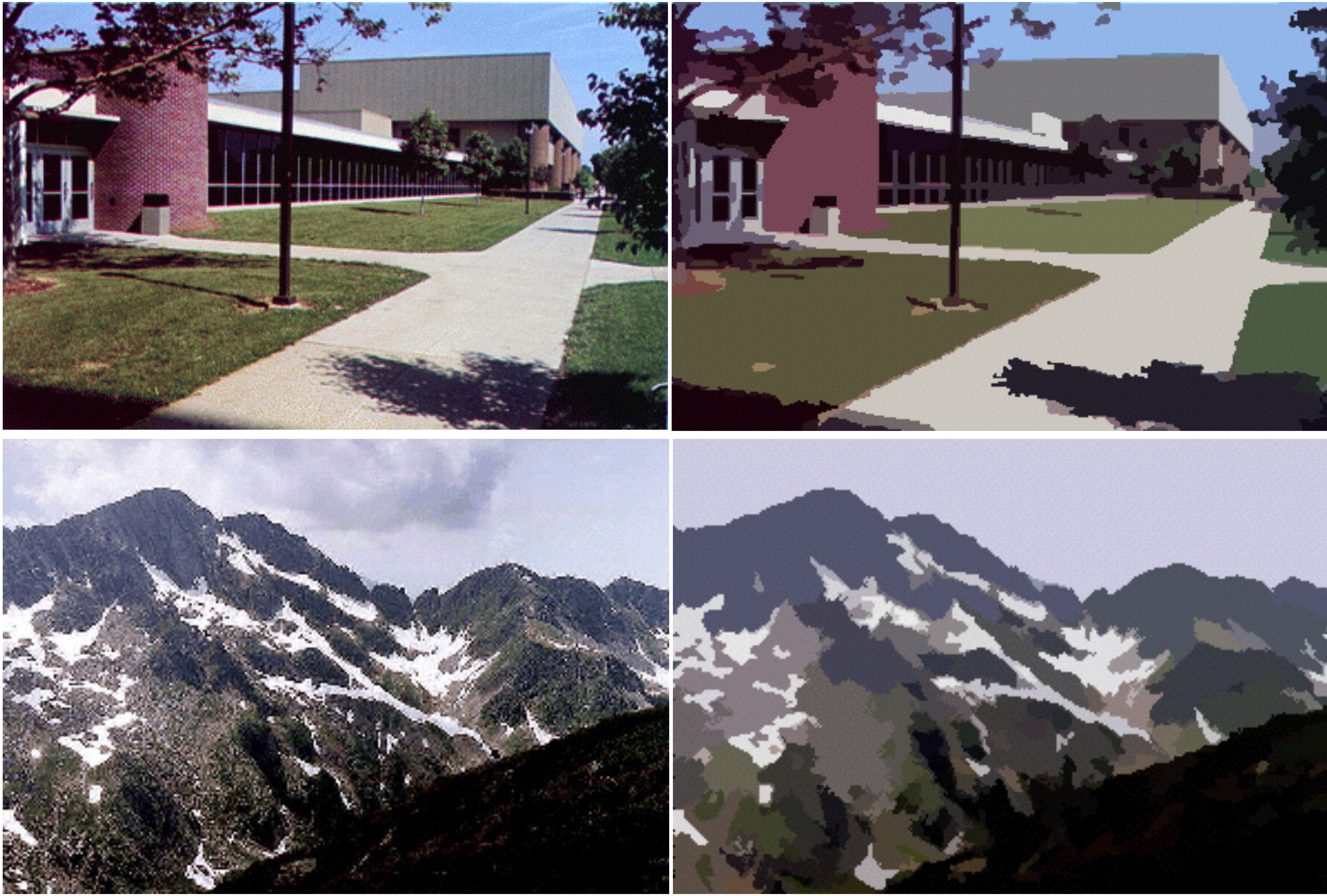
Mean shift

- Speed up
  - Binned estimation
  - Fast neighbour search
  - Update each window at each iteration
- Other tricks
  - Use kNN to determine window sizes adaptively

D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.

Mean shift

- Pros
  - Good general purpose segmentation
  - Flexible in number and shapes of regions
  - Robust to outliers
- Cons
  - Have to choose kernel size in advance
  - Not suitable for high-dimensional features (i.e., data points)
- When to use it?
  - Oversegmentation
  - Multiple segmentations
  - **Tracking**, clustering and filtering applications



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

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

Summary

- K-means clustering
- K-mediods clustering
- Agglomerative clustering
- Mean-shift clustering