Introduction to Machine Learning Clustering

Prof. Chang-Chieh Cheng

Information Technology Service Center

National Chiao Tung University

Unsupervised Learning

- Unlabeled or uncategorized training data
  - The training data without target information
- Example:
  - Data analysis from questionnaires
    - How many groups can be divided from the results of questionnaires
  - How many visitor types visited your web site.
  - Object recognition from a image database
    - You don't know how many objects and what kind of object in each image of the database.

K-Means Clustering

- Clustering n data points **X** into k disjoint subsets S_i containing n_i data points so as to minimize the sum-of-squares criterion.

$$\mathbf{X} = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}\}, \mathbf{S} = \{\boldsymbol{S_1}, \boldsymbol{S_2}, \ldots, \boldsymbol{S_k}\}$$

$$\underset{\boldsymbol{S}}{\operatorname{argmin}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{k}} \sum_{\boldsymbol{x \in S_i}} \|\boldsymbol{x} - \boldsymbol{\mu_i}\|^{\mathbf{2}}$$

- where $\boldsymbol{\mu_i}$ is the center of $\boldsymbol{S_i}$
- *K*-means clustering is a type of **unsupervised learning**, which is used when data without defined categories.
- References:
  - https://en.wikipedia.org/wiki/K-means_clustering
  - http://mathworld.wolfram.com/K-MeansClusteringAlgorithm.html
  - https://www.datascience.com/blog/k-means-clustering
  - http://www.saedsayad.com/clustering_kmeans.htm

K-Means Clustering

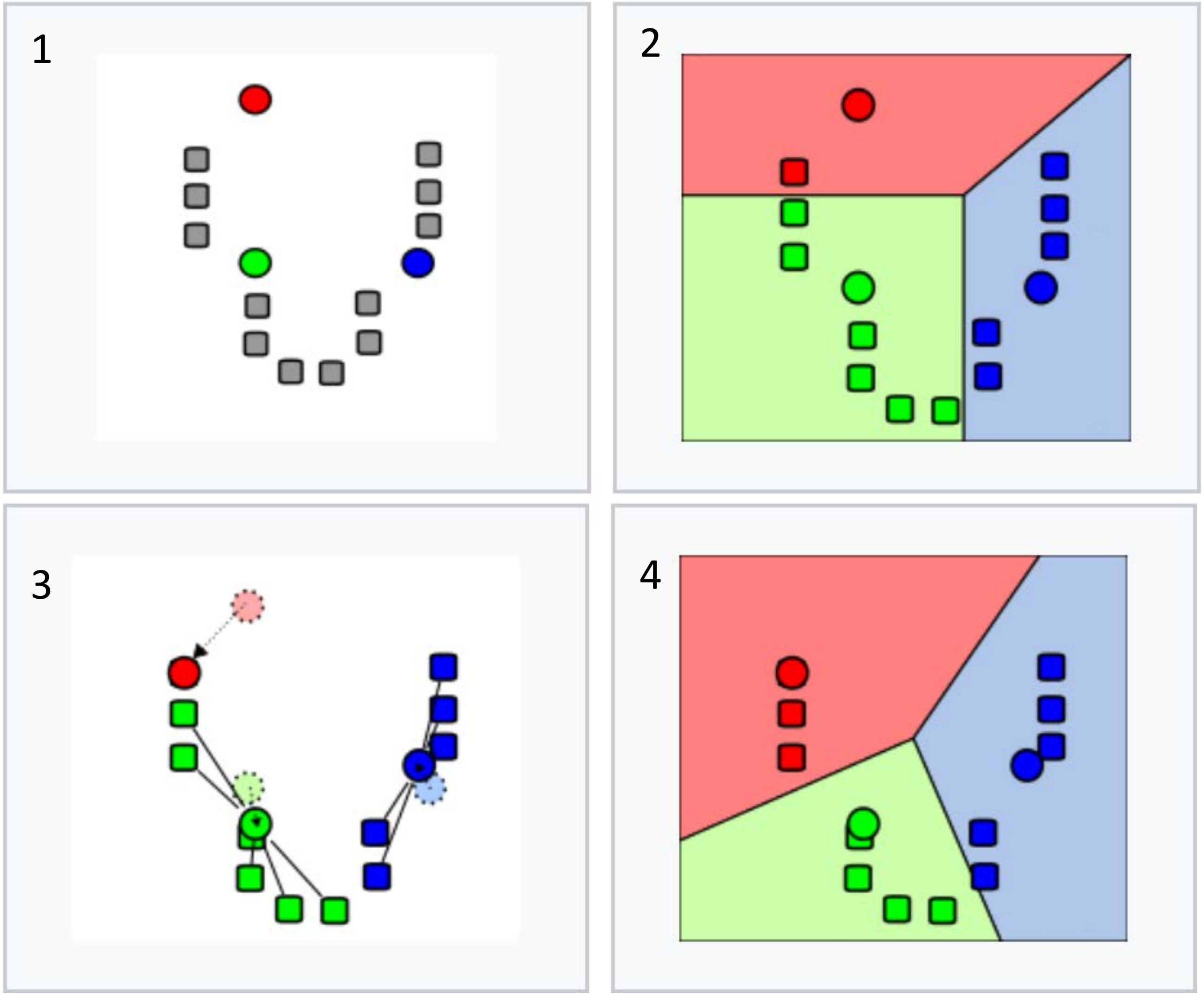
- **EM algorithm**

  Input: **X** and *k.*

  1. Select *k* points at random as cluster centers.
     - These *k* points may not ∈ **X**
  2. **E step (expectation)**
     Assign data instances to their closest cluster center according to the Euclidean distance function.
     - Generating **S**
  3. **M step (maximization)**
     Updating the cluster center by the mean of data instances in each cluster.
  4. Repeat steps 2 and 3 until a stopping criteria is met:
     - No data points change clusters.
     - The sum of the distances is minimized.
     - Some maximum number of iterations is reached.

K-Means Clustering

- Standard algorithm



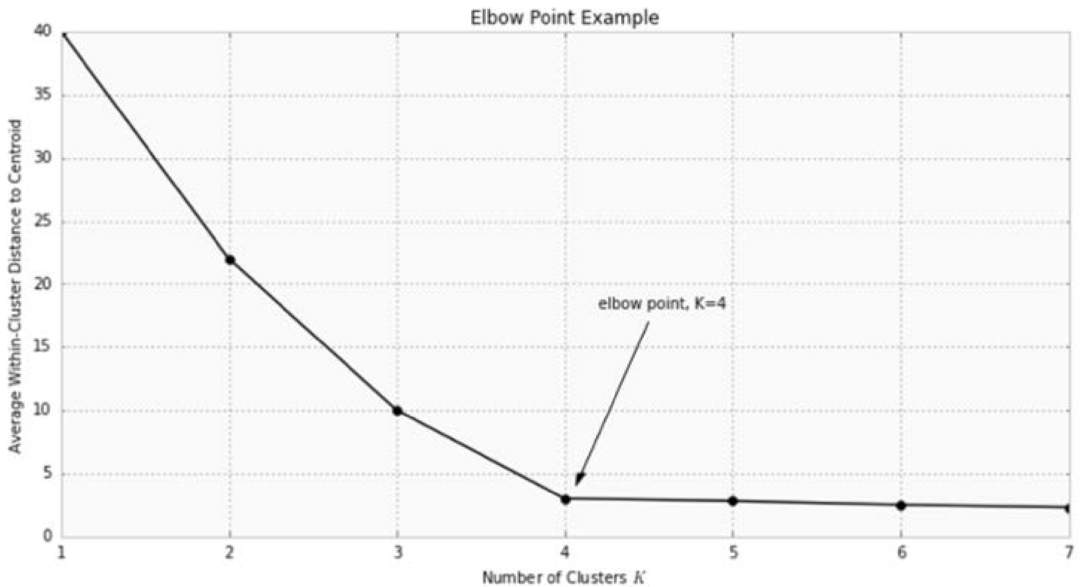
Figuring by: Nathan Landman, Hannah Pang, Christopher Williams, https://brilliant.org/wiki/k-means-clustering/

K-Means Clustering

- How to decide *k*?
  - There is no perfect method for determining exact value of *K.*
  - An approximate algorithm
    - Increasing the *k* will always reduce the distance to data points
    - Calculating

$$\boldsymbol{\alpha_k = \frac{1}{k}\sum_{i=1}^{k}\sum_{x\in S_i}\|x-\mu_i\|^2}$$

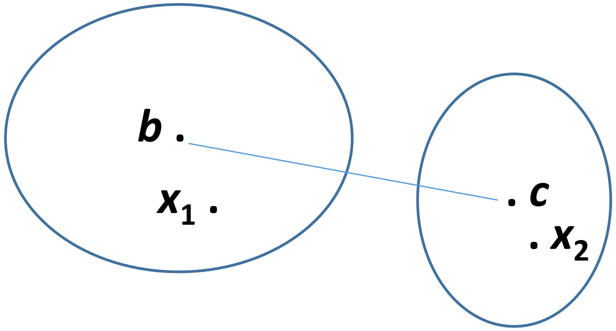
    - Select *k* while $\frac{\mathrm{d}\alpha_k}{\mathrm{d}k}$ less than a small number



Figuring by: Andrea Trevino, https://www.datascience.com/blog/k-means-clustering

K-Means Clustering

- Acceleration
  - Charles Elkan, "**Using the triangle inequality to accelerate k-means**," *Proceedings of the Twentieth International Conference on International Conference on Machine Learning (ICML'03)*, pp.147-153, 2003.
  - Let $\boldsymbol{x}$ be a point and let $\boldsymbol{b}$ and $\boldsymbol{c}$ be centers.
    If $d(\boldsymbol{b}, \boldsymbol{c}) \geq 2d(\boldsymbol{x}, \boldsymbol{b})$ then $d(\boldsymbol{x}, \boldsymbol{c}) \geq 2d(\boldsymbol{x}, \boldsymbol{b})$
  - Let $\boldsymbol{x}$ be a point and let $\boldsymbol{b}$ and $\boldsymbol{c}$ be centers.
    $d(\boldsymbol{x}, \boldsymbol{c}) \geq \max(0, d(\boldsymbol{x}, \boldsymbol{b}) - d(\boldsymbol{b}, \boldsymbol{c}))$



K-Means Clustering

- KMeans in sklean

```
from sklearn.cluster import KMeans
import numpy as np

X = np.array([[0.5, 2], [1, 4.5], [1, 0.25],
             [4, 2], [4, 4], [4, 0]])

kmeans = KMeans(n_clusters=2, random_state=0).fit(X)
print(kmeans.labels_)                      # [0 0 0 1 1 1]
print(kmeans.cluster_centers_)             #[[0.833 2.25]
                                           # [4.  2. ]]
targets = kmeans.predict([[0, 0], [4, 3]])
print(targets)                             # [0 1]
```

Gaussian Mixture Models, GMM

- 1D Gaussian

$$N(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- Multivariate normal distribution (*n*-dimensional space)

$$N(\mathbf{x}, \boldsymbol{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

  - where $\mathbf{x} = \{x_1, x_2, \ldots x_n\}$, $\boldsymbol{\mu} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_i$

Σ is the covariance matrix, $\Sigma = E[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}}]$

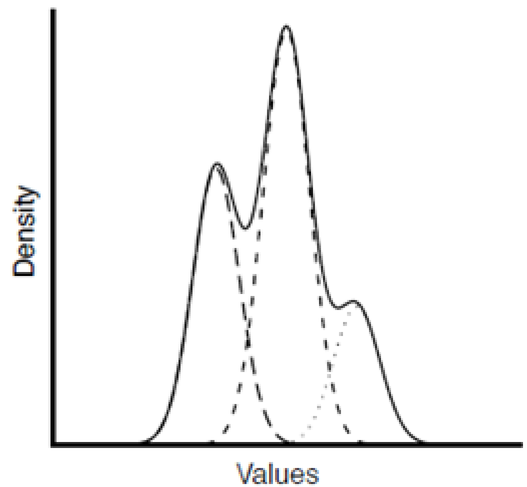
and $|\Sigma|$ is the determinant of Σ

Gaussian Mixture Models, GMM

- Gaussian mixture models
  - 1D Gaussian:

$$N(x, \mathbf{u}, \boldsymbol{\sigma}, \mathbf{w}) = \sum_{s=1}^{k} w_s N(x, \mu_s, \sigma_s)$$

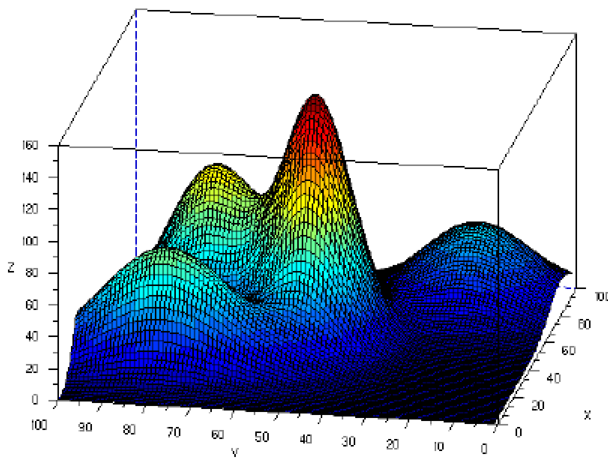
    - where $\mathbf{w} = \{w_1, w_2, \dots w_k\}$



Gaussian Mixture Models, GMM

- Gaussian mixture models
  - Multivariate Gaussian mixture models :

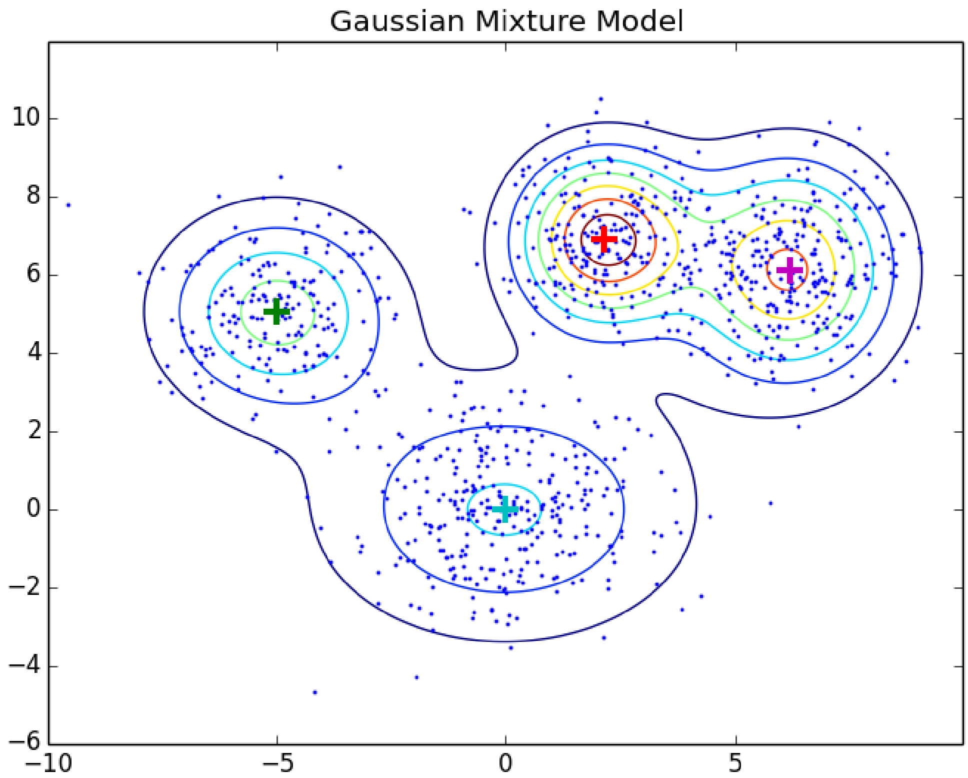
$$N(\mathbf{x}, \mathbf{u}, \mathbf{\Sigma}, \mathbf{w}) = \sum_{s=1}^{k} w_s N(\mathbf{x}, \boldsymbol{\mu}_s, \Sigma_s)$$



https://www.researchgate.net/figure/3D-view-of-a-4D-Gaussian-Mixture-Model-used-in-our-experiments_fig1_224105715

Gaussian Mixture Models, GMM

- Finding a Gaussian mixture model to fit the data distribution



http://yulearning.blogspot.com/2014/11/einsteins-most-famous-equation-is-emc2.html

Gaussian Mixture Models, GMM

- E step: Expectation
  - For each datum $\mathbf{x}_i$
  - Compute r_{is}, the probability that is belongs to cluster *s*

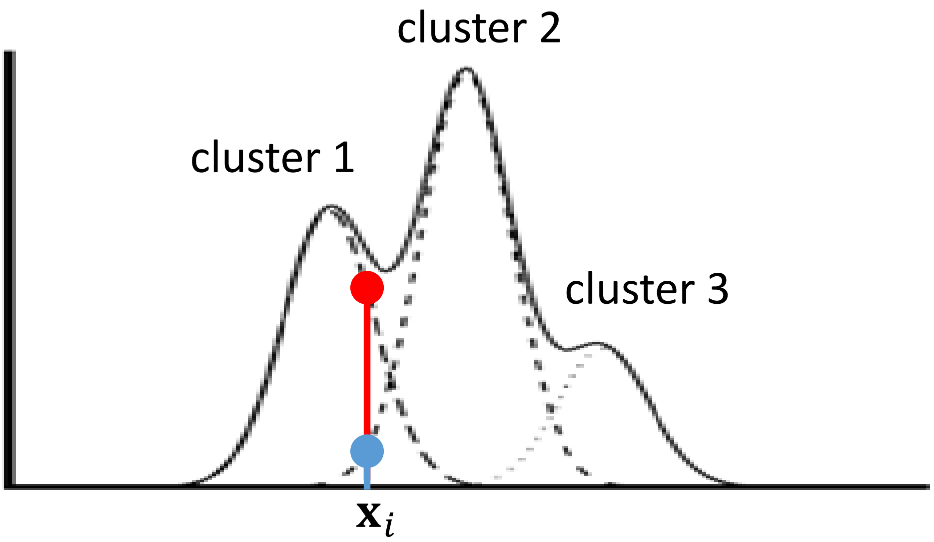
The probability of $\mathbf{x}_i$ under *s*

$$r_{is} = \frac{w_s N(\mathbf{x}_i, \boldsymbol{\mu}_s, \Sigma_s)}{\sum_{j=1}^{k} w_j N(\mathbf{x}_i, \boldsymbol{\mu}_j, \Sigma_j)}$$

Normalization

Gaussian Mixture Models, GMM

- Example:
  - $k = 3$
  - $r_{i1} = 0.8$
  - $r_{i2} = 0.2$
  - $r_{i3} = 0.0$
  - So, we can say that $\mathbf{x}_i$ belongs to cluster 1



Gaussian Mixture Models, GMM

- M step: Maximization
  - Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_m\}$ (Training data set)
  - Log likelihood

$$\log p(\mathbf{X}) = \sum_{i}^{m} \log[N(\mathbf{x}_i, \mathbf{u}, \mathbf{\Sigma}, \mathbf{w})]$$

$$= \sum_{i}^{m} \log\left[\sum_{s=1}^{k} w_s N(\mathbf{x}_i, \boldsymbol{\mu}_s, \Sigma_s)\right]$$

  - Compute w_s, $\boldsymbol{\mu}_s$, and Σ_s such that $\log p(\mathbf{X})$ is maximum

If any datum can be clustered with 100% accuracy, $\log p(\mathbf{X})$ is zero; Otherwise, $\log p(\mathbf{X})$ is a negative number
EX: One datum can be clustered to c_1 and c_2 with 50% probabilities respectively.
And $w_1 = w_2 = 0.5$. Then, $\log_2 p(\mathbf{X}) = \log_2(0.5 \times 0.5 + 0.5 \times 0.5) = -2$

Gaussian Mixture Models, GMM

- Given r_{is} for data point $\mathbf{x}_i$ and Gaussian N_s.
  - Let

$$\alpha_s = \sum_{i=1}^{m} r_{is}$$

  - Then,

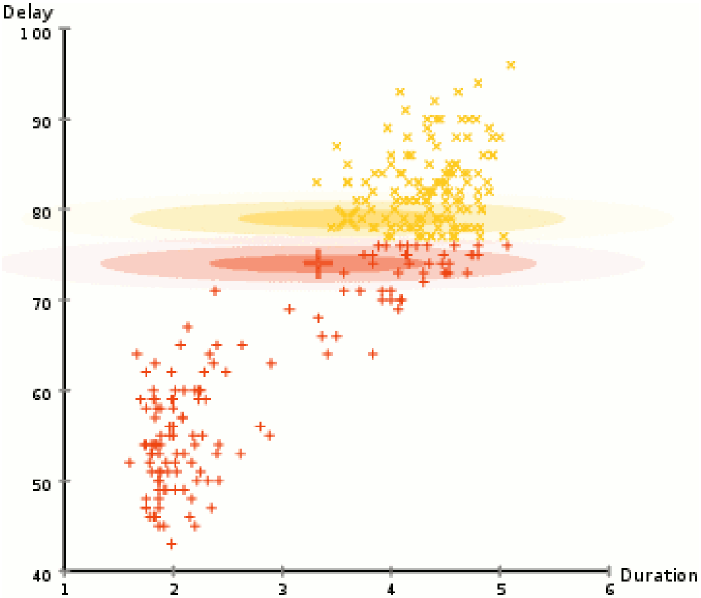
$$w_s = \frac{\alpha_s}{m}$$

$$\boldsymbol{\mu}_s = \frac{1}{\alpha_s} \sum_{i=1}^{m} r_{is} \mathbf{x}_i$$

$$\Sigma_s = \frac{1}{\alpha_s} \sum_{i=1}^{m} r_{is} (\mathbf{x}_i - \boldsymbol{\mu}_s)(\mathbf{x}_i - \boldsymbol{\mu}_s)^{\mathrm{T}}$$

Gaussian Mixture Models, GMM

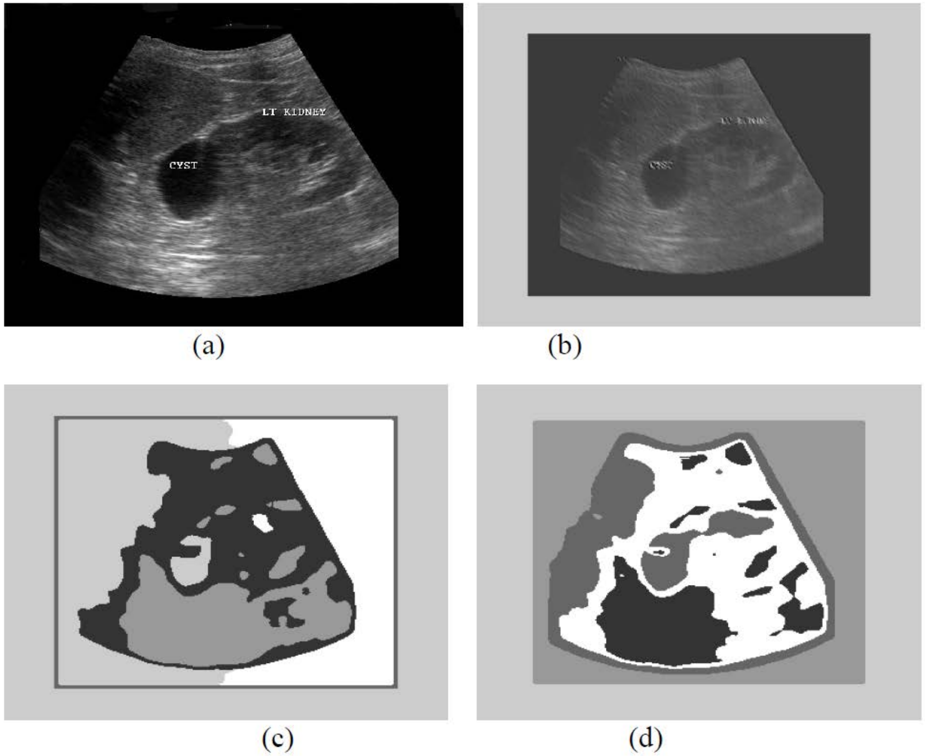
- Algorithm
  1. Given *k*
  2. Initial guess of w_s, $\boldsymbol{\mu}_s$, and Σ_s
  3. E step, compute r_{is}
  4. M step, compute w_s, $\boldsymbol{\mu}_s$, and Σ_s
  5. Repeat Step. 3 and 4. until $\log p(\mathbf{X})$ is larger than a threshold or the change of $\log p(\mathbf{X})$ is smaller that a constant.



https://en.wikipedia.org/wiki/Expectation-maximization_algorithm

Gaussian Mixture Models, GMM

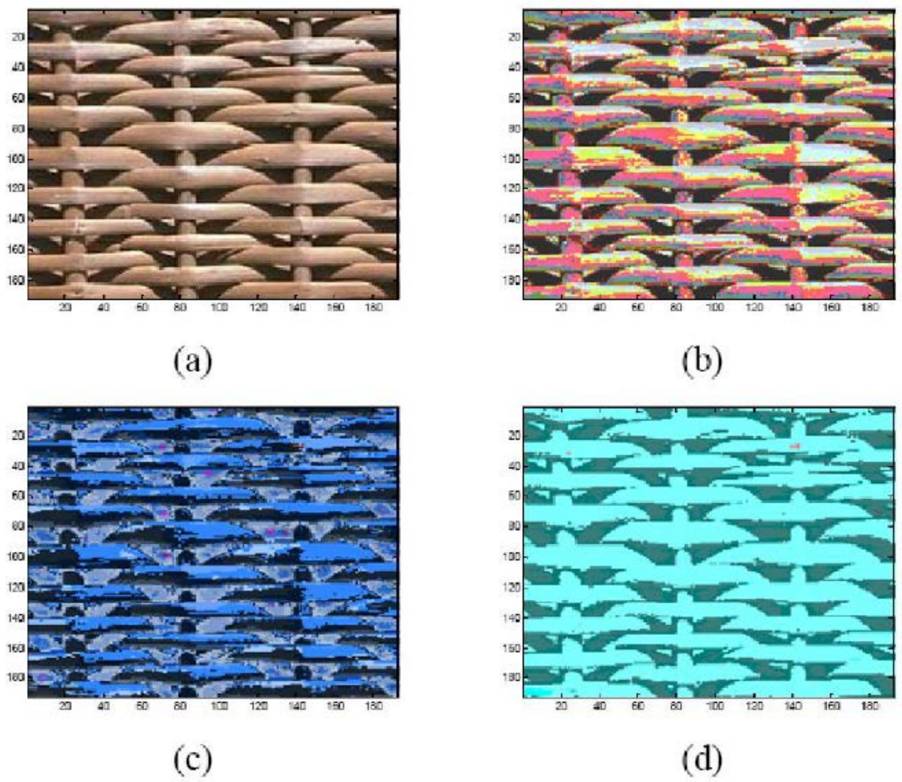
- A. Khanna et al, "US Image Segmentation Based on Expectation Maximization and Gabor Filter," *International Journal of Modeling and Optimization*, vol. 2, no. 3, pp. 230-233, Jun. 2012.



(a). Original image (b): one of the gabor filtered image
(c): result using K-means clustering (d): segmentation result using EM algorithm

Gaussian Mixture Models, GMM

- Z. Huang and D. Liu, "Segmentation of Color Image Using EM algorithm in HSV Color Space," *2007 International Conference on Information Acquisition*, Seogwipo-si, 2007, pp. 316-319.



The clusters number is bfive (C=5).

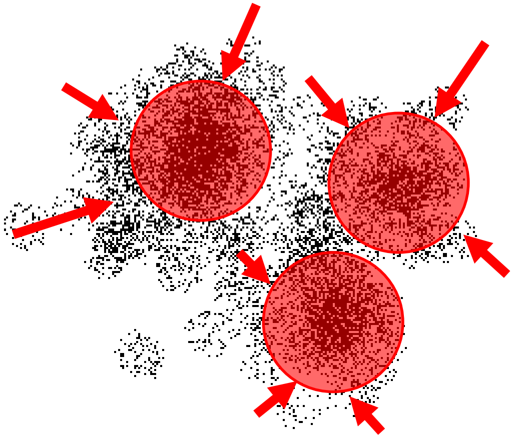
(a) is original color image;(b) kmeans in RGB color space;(c) k-means in HSV color (d). EM

Gaussian Mixture Models, GMM

- GMM in scikit-learn
- https://github.com/jameschengcs/ml/blob/master/EM_iris.py

Mean Shift Clustering

- D. Comaniciu and P. Meer, "**Mean shift: a robust approach toward feature space analysis**," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, May 2002.
- Overview
  1. Finding the trend of each point
  2. Shift each point by the trend
  3. Repeat 1 and 2 until the trend is near to zero ➔ the point is shifted to a region of a cluster



Mean Shift Clustering

- A clustering method that doesn't require specifying the number of clusters
- **The kernel density estimator**
  - Given *m* data points $\mathbf{x}_i$, $i = 1, 2, ..., m$ on a *n*-dimensional space $\boldsymbol{R}^n$, the multivariate kernel **density** estimate obtained with kernel $K(\mathbf{x})$ and window radius *h* is

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

  - where $\int_{R^n} K(\mathbf{x}) = 1$ and $K(\mathbf{x}) \geq 0$
  - $K(\mathbf{x}) = \varphi(\|\mathbf{x}\|^2)$
  - Two frequently used φ for mean shift are:
    - $\varphi(s) = \begin{cases} 1 & if\ s \leq \tau \\ 0 & if\ s > \tau \end{cases}$, where τ is a threshold.
    - $\varphi(s) = e^{-\frac{s}{2\sigma^2}}$

Mean Shift Clustering

- Shifting the positions of *m* data points that belong to a region, such the density is the highest

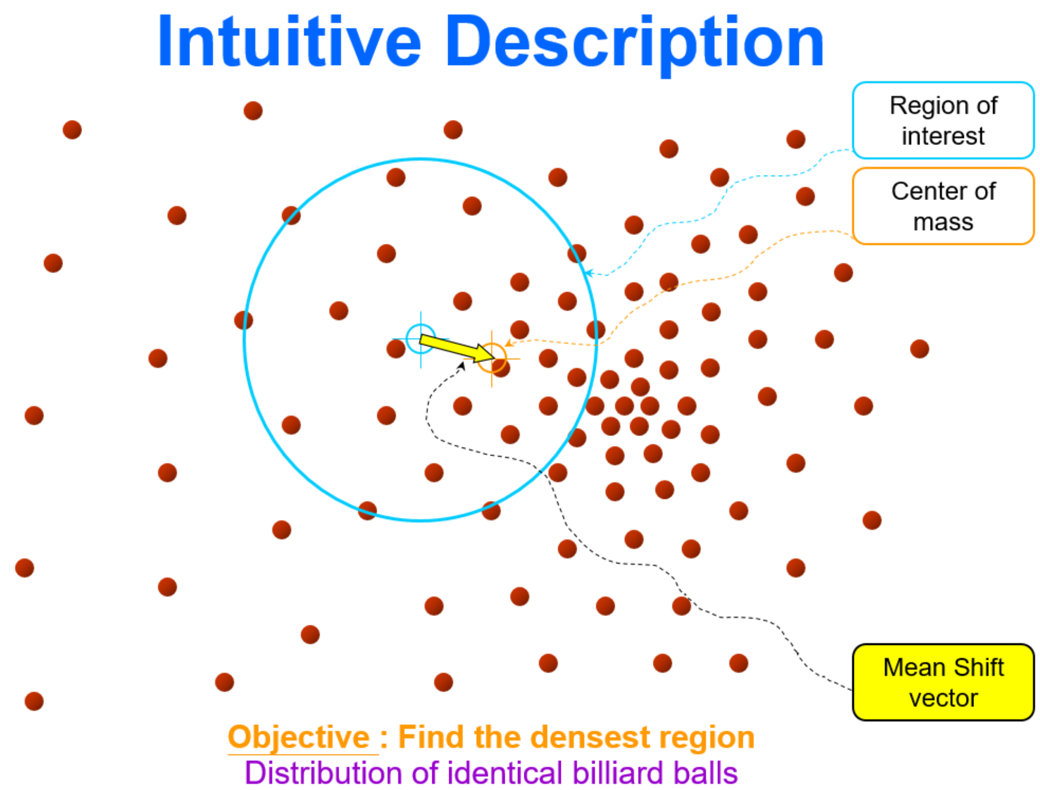


Figure: Yaron Ukrainitz & Bernard Sarel, "Mean Shift Theory and Applications"

Mean Shift Clustering

- Finding $\Delta\mathbf{x}$ to shift *m* data points of a region, and $f'(\mathbf{x}) = 0$

$$f'(\mathbf{x}) = \frac{1}{mh^n}\sum_{i=1}^{m} K'\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$$

$$= \frac{1}{mh^n}\sum_{i=1}^{m} \varphi'\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$$

$$= \frac{C}{mh^{n+2}}\sum_{i=1}^{m} (\mathbf{x}-\mathbf{x}_i)\varphi'\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)$$

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

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

$$F(x) = f\big(g(x)\big)$$
$$F'(x) = f'\big(g(x)\big)g'(x)$$
$$\frac{df(x)^2}{dx} = 2f(x)\frac{df(x)}{dx}$$

let $g(s) = -\varphi'(s)$

The 1st term is > 0 and proportional to the density estimate as x computed with the kernel

The 2nd is the mean shift

Mean Shift Clustering

- The mean shift

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

- let $\mathbf{y} = \frac{\sum_{i=1}^{m} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{m} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}$ $\quad \Delta \mathbf{x}^t = \mathbf{y}^t - \mathbf{x}^t$

- computation of the mean shift vector $\mathbf{y}^t$
- translation of the region $\mathbf{x}^{t+1} = \mathbf{y}^t$
  - because $\mathbf{x}^{t+1} = \mathbf{x}^t + \Delta \mathbf{x}^t = \mathbf{x}^t + \mathbf{y}^t - \mathbf{x}^t$
- until $\|\Delta \mathbf{x}^t\|$ is closed to zero

Mean Shift Clustering

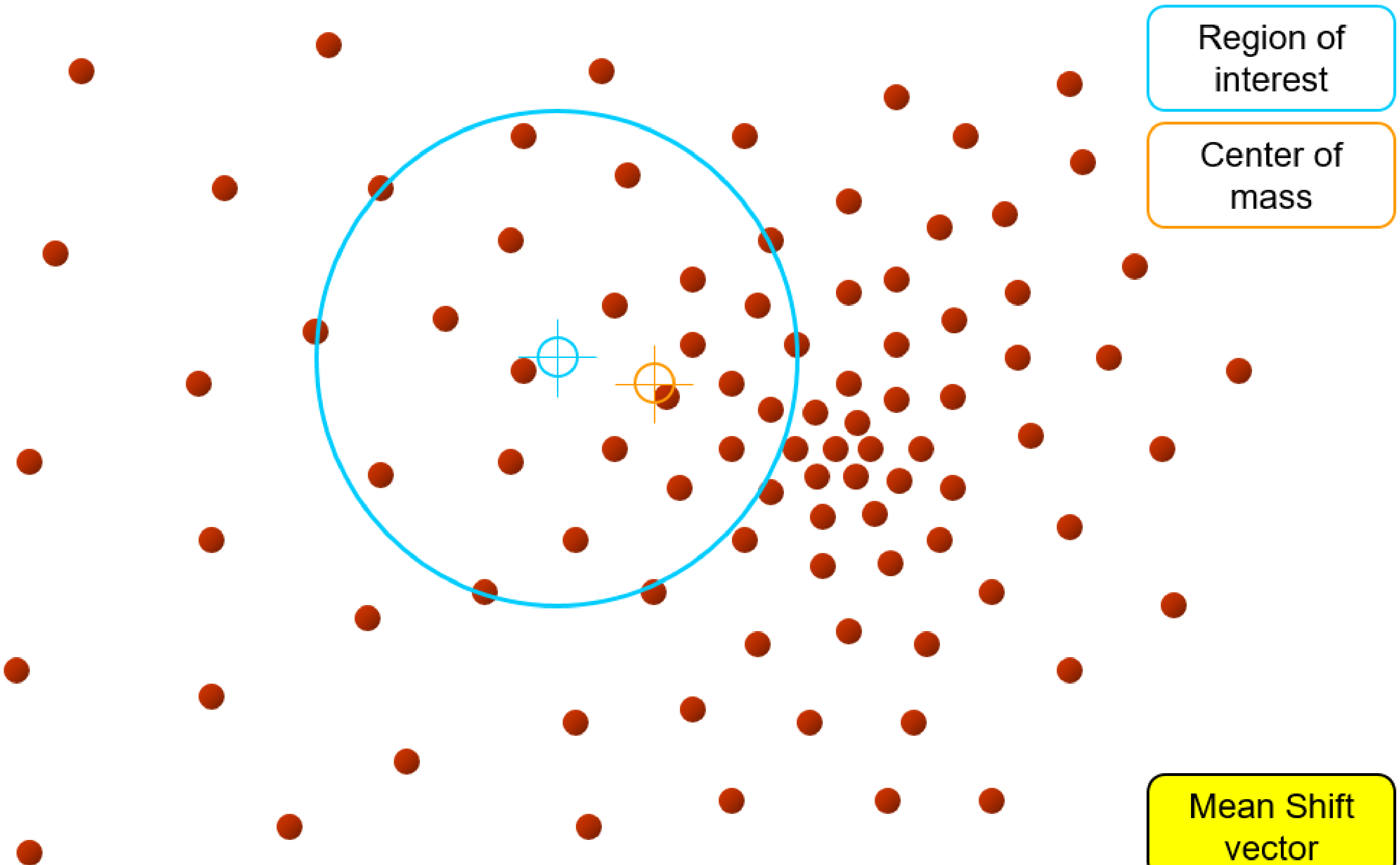
- For example,

$$g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right) = e^{-\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2}$$

$$\mathbf{y}^0 = \mathbf{x}$$

$$\mathbf{y}^t = \frac{\sum_{i=1}^{m} \mathbf{x}_i g\left(\left\|\frac{\mathbf{y}^{t-1}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{m} g\left(\left\|\frac{\mathbf{y}^{t-1}-\mathbf{x}_i}{h}\right\|^2\right)}$$

Mean Shift Clustering

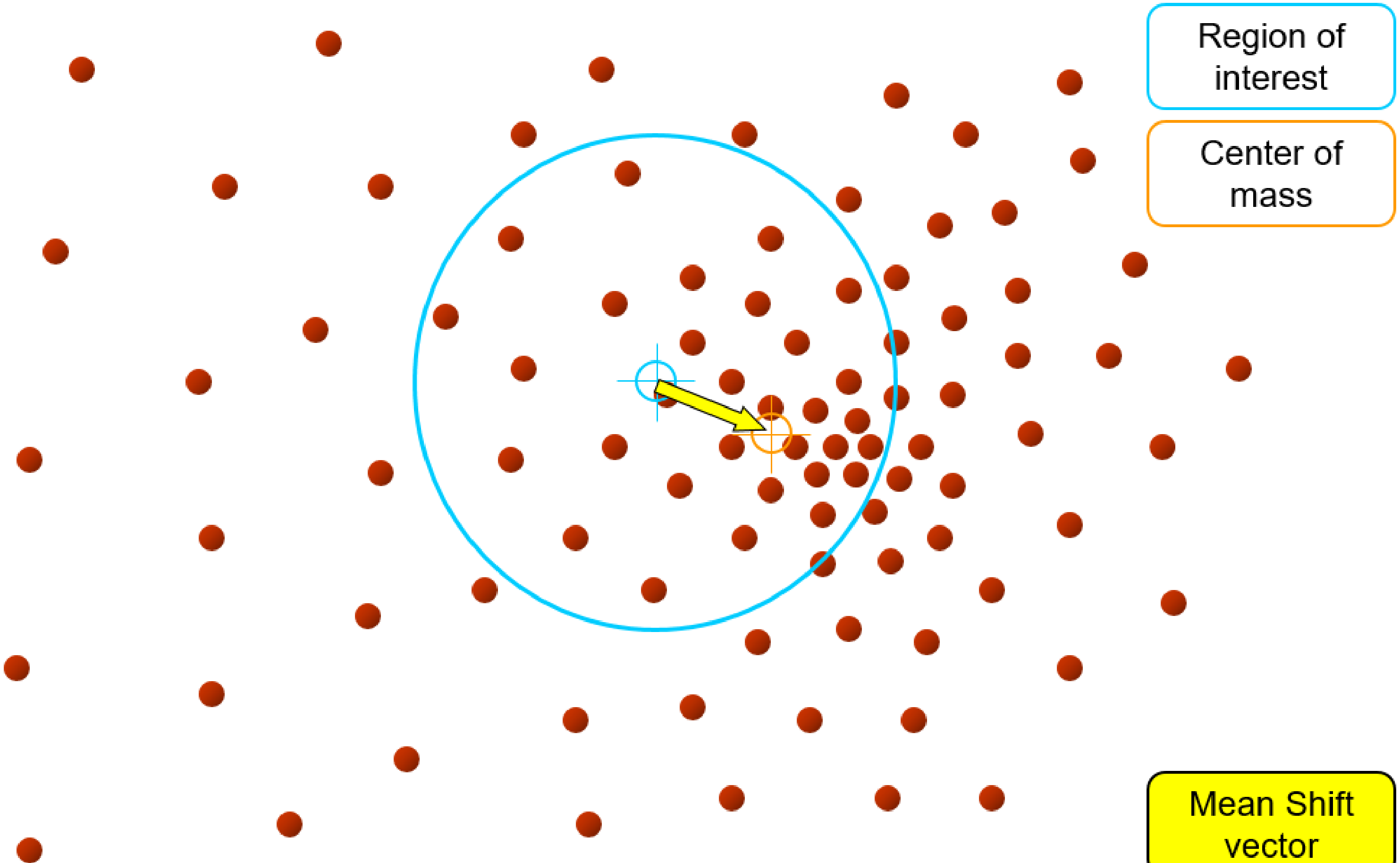


Objective : Find the densest region

Distribution of identical billiard balls

Figure : Yaron Ukrainitz & Bernard Sarel, "Mean Shift Theory and Applications"

Mean Shift Clustering

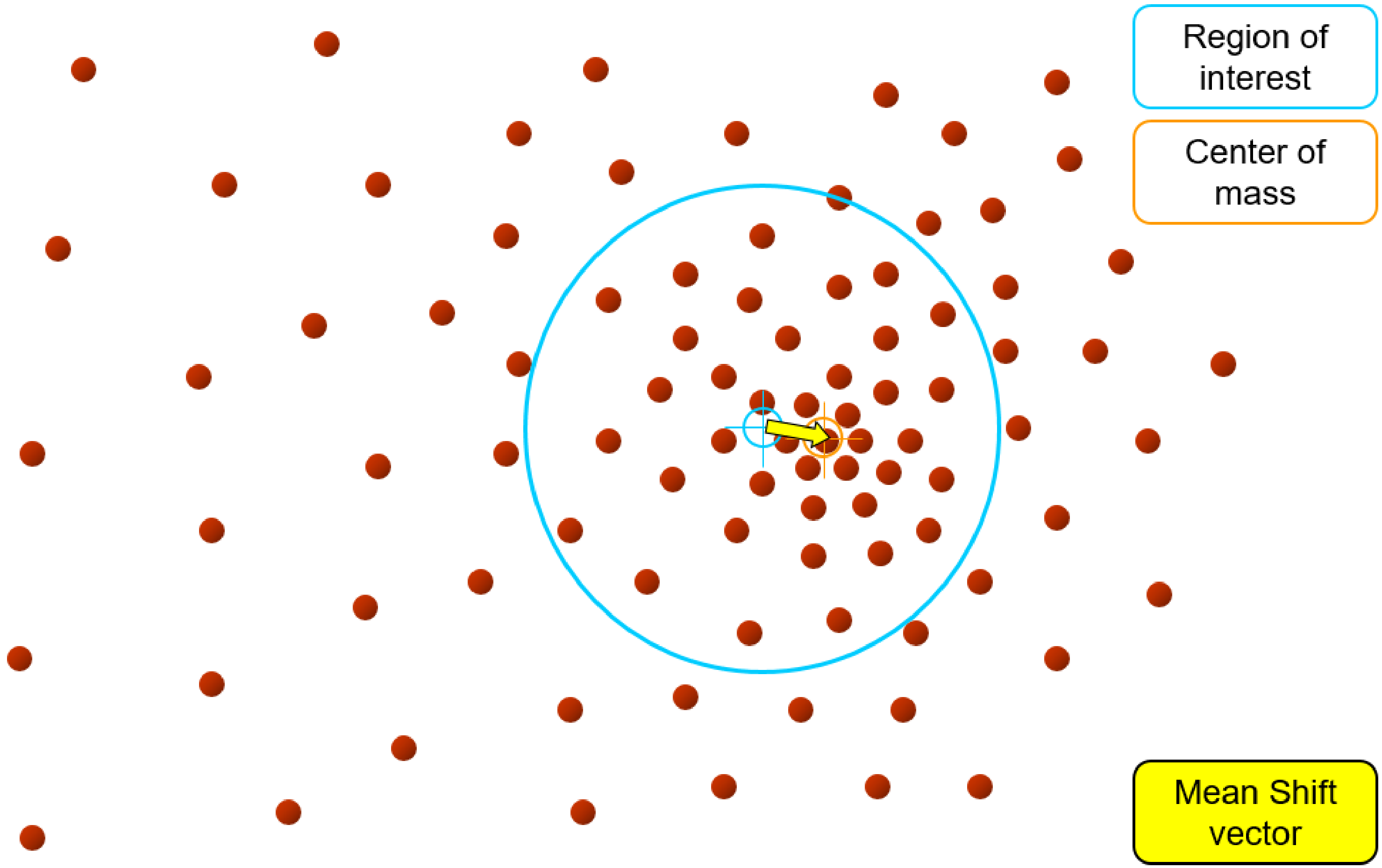


Objective : Find the densest region

Distribution of identical billiard balls

Figure : Yaron Ukrainitz & Bernard Sarel, "Mean Shift Theory and Applications"

Mean Shift Clustering

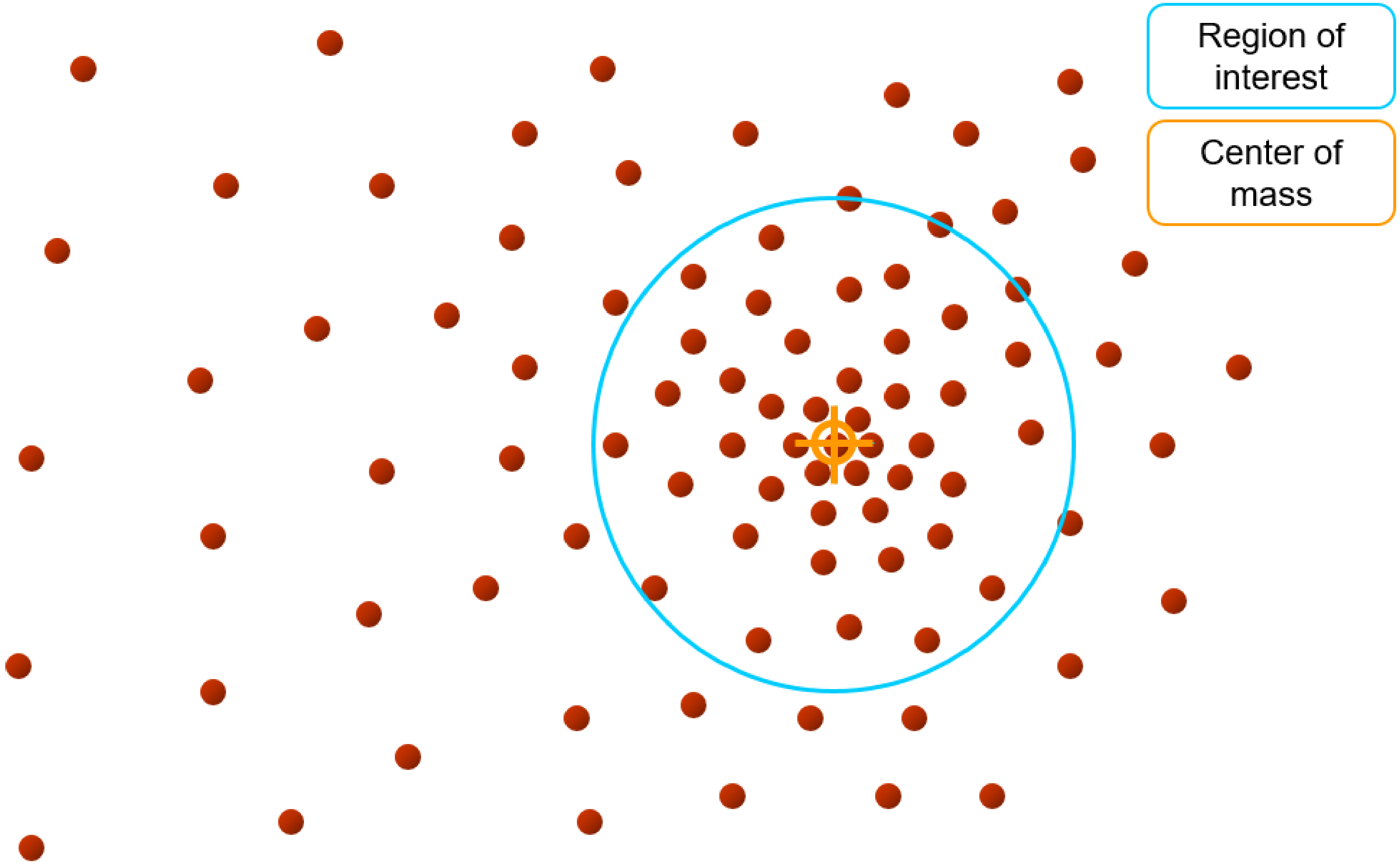


Objective : Find the densest region

Distribution of identical billiard balls

Figure: Yaron Ukrainitz & Bernard Sarel, "Mean Shift Theory and Applications"

Mean Shift Clustering



Objective : Find the densest region

Distribution of identical billiard balls

Figure: Yaron Ukrainitz & Bernard Sarel, "Mean Shift Theory and Applications"

Mean Shift Clustering

- D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, May 2002.

Mean Shift Clustering

- Mean shift in Scikit-learn
  - https://github.com/jameschengcs/ml/blob/master/MeanShift_iris.py