Unsupervised Learning

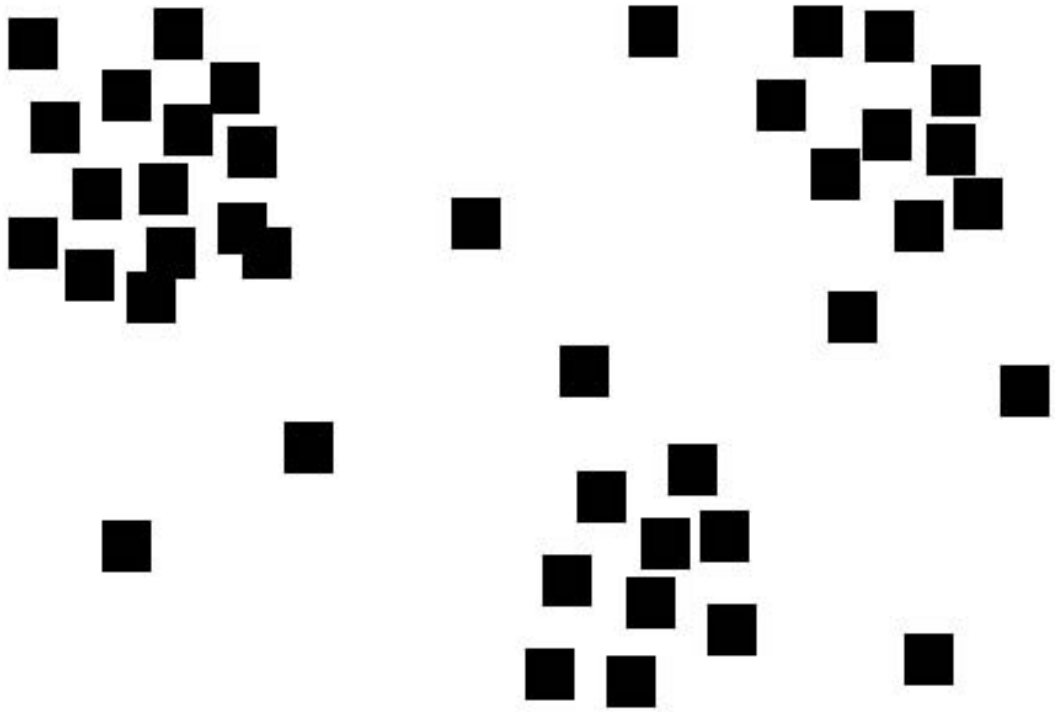
Zhirong Yang
Department of Computer Science NTNU

Lecture outline

- To introduce cluster analysis
  - "definition"
  - classical clustering algorithms
  - major difficulty and potential solution
- Other types of unsupervised learning
  - dimensionality reduction
  - anomaly detection
  - generation
- Brief introduction to Self-Supervised Learning

Cluster Analysis

How many clusters do you see?



“Definition” of Cluster Analysis

Cluster analysis or clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters.

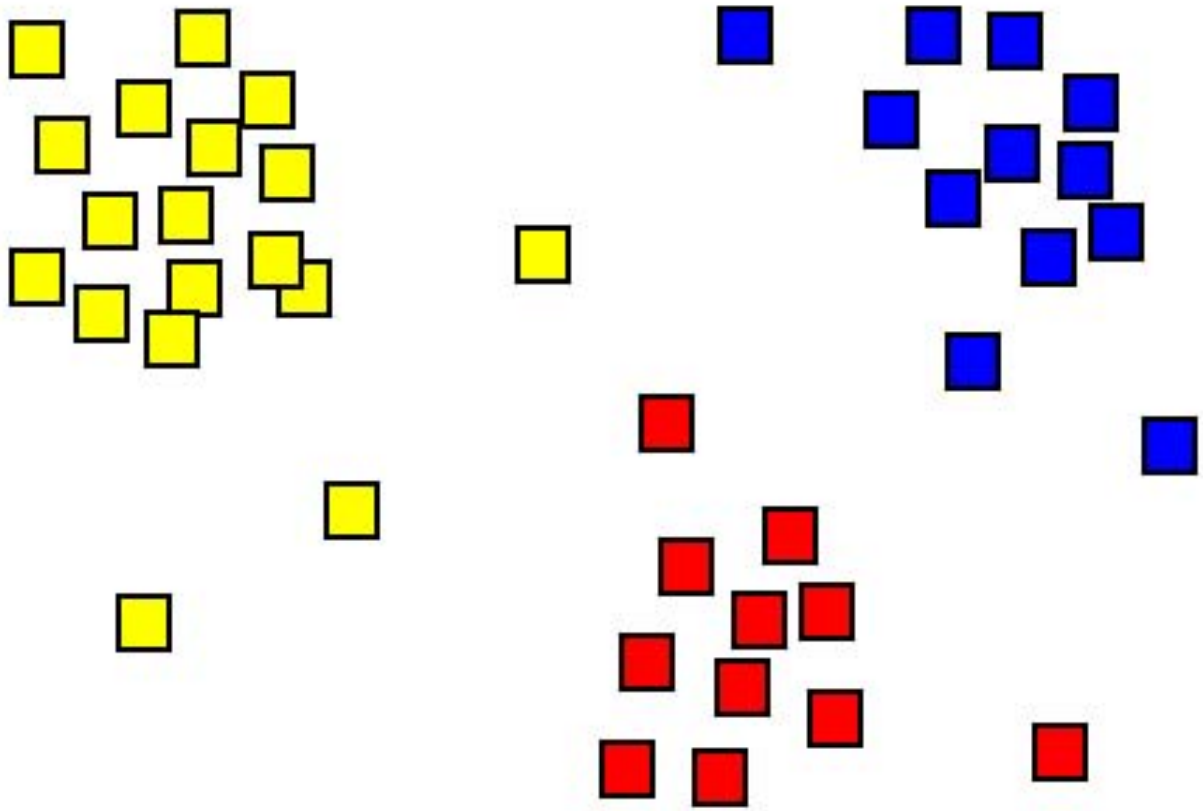
— Wikipedia

"Definition" of Cluster Analysis

Cluster analysis or clustering is the task of assigning a set of objects into groups (called clusters) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters.

— Wikipedia

How many clusters? A plausible answer

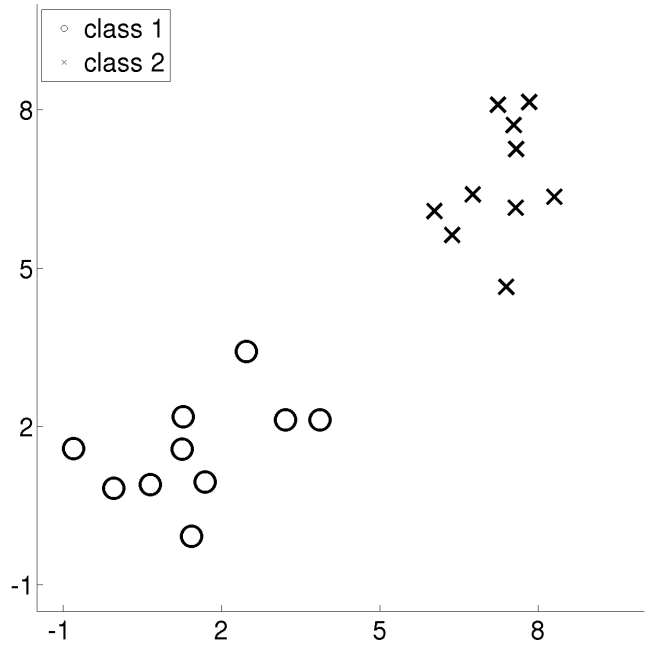


Clustering vs. Classification

- Common: multivariate input and categorical output

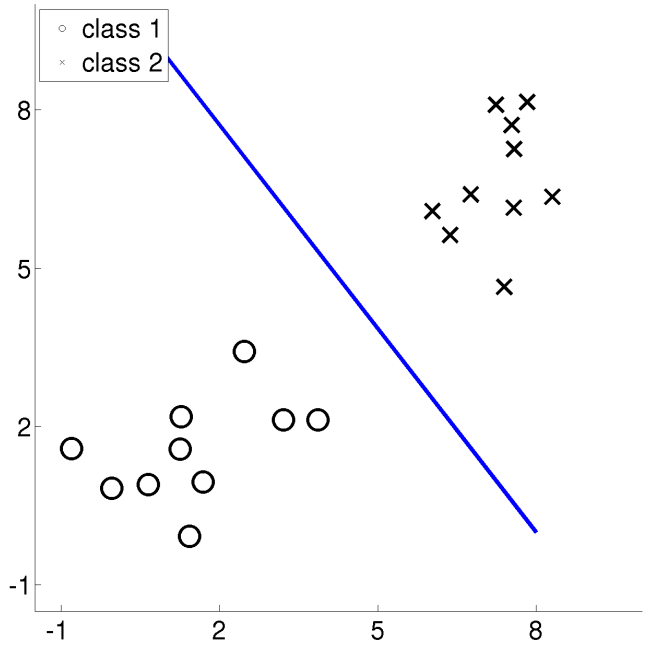
Clustering vs. Classification

- Common: multivariate input and categorical output
- Difference:
  - Classification is supervised learning
    - Training data: $\{\mathbf{x}^{(t)}, y^{(t)}\}_{t=1}^{N}$
    - Task: predict y for new data vectors $\mathbf{x}$



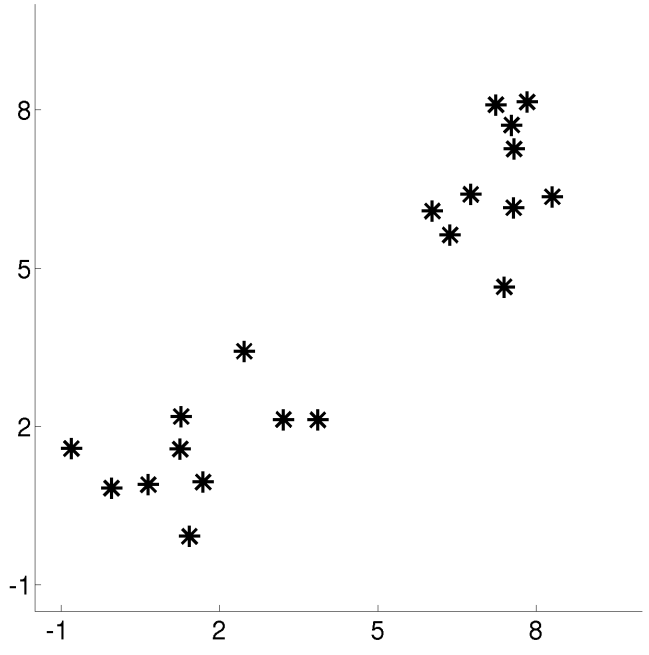
Clustering vs. Classification

- Common: multivariate input and categorical output
- Difference:
  - Classification is supervised learning
    - Training data: $\{\mathbf{x}^{(t)}, y^{(t)}\}_{t=1}^{N}$
    - Task: predict y for new data vectors $\mathbf{x}$



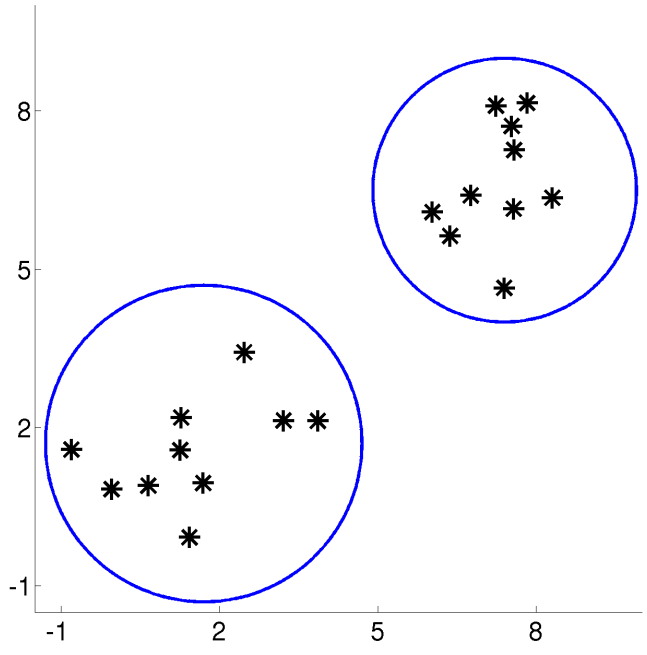
Clustering vs. Classification

- Common: multivariate input and categorical output
- Difference:
  - Classification is supervised learning
    - Training data: $\{\mathbf{x}^{(t)}, y^{(t)}\}_{t=1}^{N}$
    - Task: predict y for new data vectors $\mathbf{x}$
  - Clustering is unsupervised learning
    - Data: $\{\mathbf{x}^{(t)}\}_{t=1}^{N}$
    - Task: assign $\{\mathbf{x}^{(t)}\}_{t=1}^{N}$ into groups.



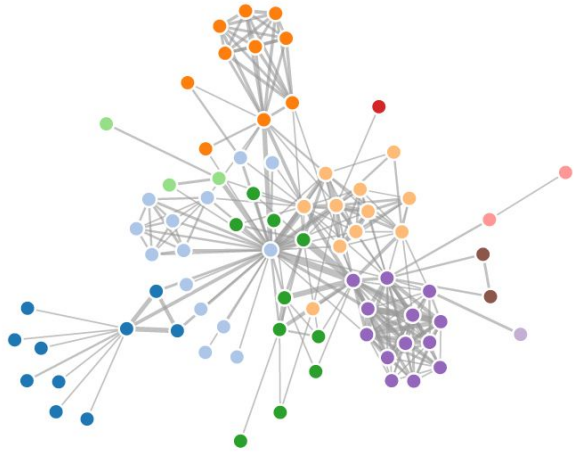
Clustering vs. Classification

- Common: multivariate input and categorical output
- Difference:
  - Classification is supervised learning
    - Training data: $\{\mathbf{x}^{(t)}, y^{(t)}\}_{t=1}^{N}$
    - Task: predict y for new data vectors $\mathbf{x}$
  - Clustering is unsupervised learning
    - Data: $\{\mathbf{x}^{(t)}\}_{t=1}^{N}$
    - Task: assign $\{\mathbf{x}^{(t)}\}_{t=1}^{N}$ into groups.

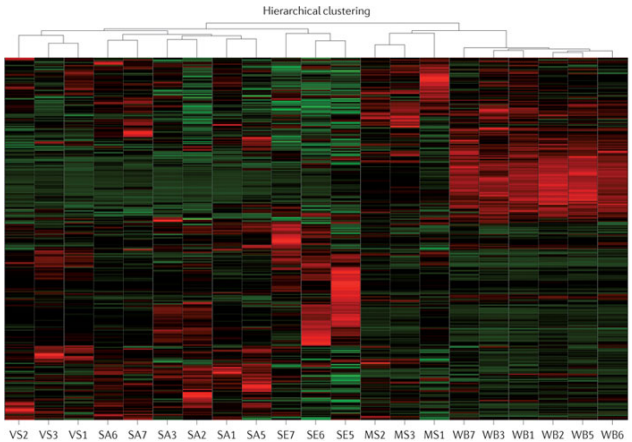


Applications of Clustering

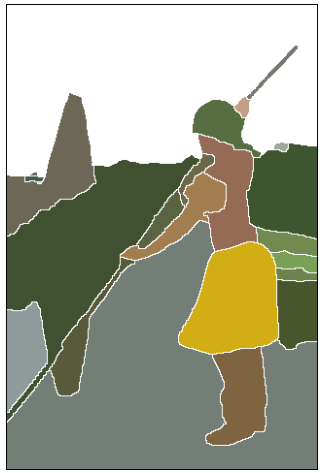
market segmentation



social network



gene expressions



computer vision

K-means clustering

- Each cluster is represented by its center point (mean vector)
- $[\mathcal{C}, \mathcal{M}] = \text{K-means}(\mathcal{X}, K)$
  - Input:
    - data points $\mathcal{X} = \{\mathbf{x}^{(t)}\}_{t=1}^{N}$, where $\mathbf{x}^{(t)} \in \mathbb{R}^D$, for $t = 1, \ldots, N$
    - number of clusters K
  - Output:
    - cluster assignment $\mathcal{C} = \{c^{(t)}\}_{t=1}^{N}$, where $\mathbf{x}^{(t)} \in \{1, \ldots, K\}$, for $t = 1, \ldots, N$
    - cluster centers $\mathcal{M} = \{\mathbf{m}^{(i)}\}_{i=1}^{K}$, where $\mathbf{m}^{(i)} \in \mathbb{R}^D$, for $i = 1, \ldots, K$

K-means algorithm (Lloyd's algorithm)

function $[\mathcal{C}, \mathcal{M}] =$ K-means($\mathcal{X}$,K)
Initialize the centers $\mathcal{M} = \{\mathbf{m}^{(i)}\}_{i=1}^{K}$.
repeat
(a) for each data point $\mathbf{x}^{(t)}$
calculate d_{ti}=dist($\mathbf{x}^{(t)}$, $\mathbf{m}^{(i)}$) for all $\mathbf{m}^{(i)}$
$c^{(t)} \leftarrow$the i with smallest d_{ti}
(b) for each cluster i
$\mathbf{m}^{(i)} \leftarrow$the mean of the cluster
until $\mathcal{C}$ and $\mathcal{M}$ do not change

e.g.

$$\text{dist}\left((a,b),(c,d)\right) = \sqrt{(a-c)^2 + (b-d)^2}$$

K-means algorithm (Lloyd's algorithm)

function $[\mathcal{C}, \mathcal{M}] =$ K-means($\mathcal{X}$,K)
Initialize the centers $\mathcal{M} = \{\mathbf{m}^{(i)}\}_{i=1}^{K}$.
repeat
(a) for each data point $\mathbf{x}^{(t)}$
calculate d_{ti}^2=dist^2($\mathbf{x}^{(t)}$, $\mathbf{m}^{(i)}$) for all $\mathbf{m}^{(i)}$
$c^{(t)} \leftarrow$the i with smallest d_{ti}^2
(b) for each cluster i
$\mathbf{m}^{(i)} \leftarrow$the mean of the cluster
until $\mathcal{C}$ and $\mathcal{M}$ do not change

e.g.

$$\text{dist}^2\left((a,b),(c,d)\right) = (a-c)^2 + (b-d)^2$$

K-means algorithm (Lloyd's algorithm)

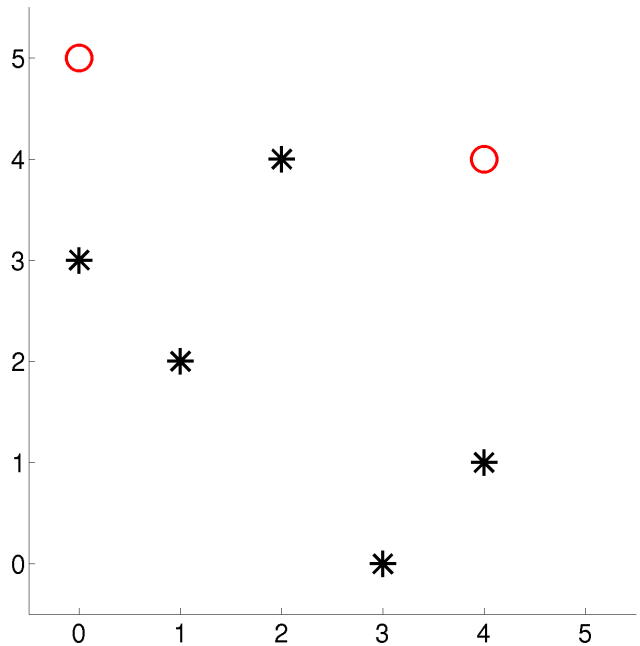
function $[\mathcal{C}, \mathcal{M}] = \text{K-means}(\mathcal{X}, K)$
Initialize the centers $\mathcal{M} = \{\mathbf{m}^{(i)}\}_{i=1}^{K}$.
repeat
(a) for each data point $\mathbf{x}^{(t)}$
calculate $d_{ti}^2 = \text{dist}^2(\mathbf{x}^{(t)}, \mathbf{m}^{(i)})$ for all $\mathbf{m}^{(i)}$
$c^{(t)} \leftarrow$ the i with smallest d_{ti}^2
(b) for each cluster i
$\mathbf{m}^{(i)} \leftarrow$ the mean of the cluster
until $\mathcal{C}$ and $\mathcal{M}$ do not change

e.g. cluster 2 has $\mathbf{x}^{(3)}$, $\mathbf{x}^{(4)}$, $\mathbf{x}^{(5)}$

$$\mathbf{m}^{(2)} \leftarrow \frac{1}{3}\left[\mathbf{x}^{(3)} + \mathbf{x}^{(4)} + \mathbf{x}^{(5)}\right]$$

K-means algorithm illustration

iteration=0



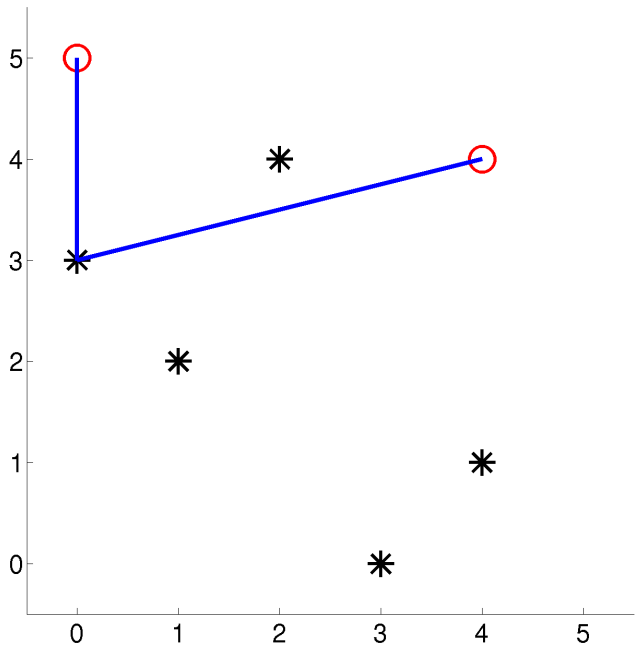
t	$\mathbf{x}^{(t)}$
1	(0.0, 3.0)
2	(1.0, 2.0)
3	(2.0, 4.0)
4	(3.0, 0.0)
5	(4.0, 1.0)

t	$c^{(t)}$
1	?
2	?
3	?
4	?
5	?

i	$\mathbf{m}^{(i)}$
1	(0.0, 5.0)
2	(4.0, 4.0)

K-means algorithm illustration

iteration=1a



t	$\mathbf{x}^{(t)}$
1	(0.0, 3.0)
2	(1.0, 2.0)
3	(2.0, 4.0)
4	(3.0, 0.0)
5	(4.0, 1.0)

t	$c^{(t)}$
1	?
2	?
3	?
4	?
5	?

i	$\mathbf{m}^{(i)}$
1	(0.0, 5.0)
2	(4.0, 4.0)

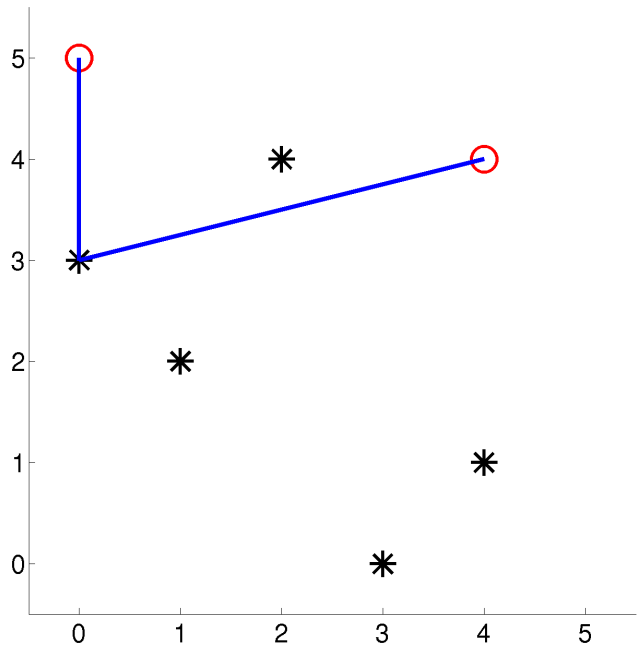
$\text{dist}^2(\mathbf{x}^{(t)}, \mathbf{m}^{(i)})$	$i = 1$	$i = 2$
$t = 1$	4	17
$t = 2$	?	?
$t = 3$	?	?
$t = 4$	?	?
$t = 5$	?	?

$$\text{dist}^2(\mathbf{x}^{(1)}, \mathbf{m}^{(1)}) = (0-0)^2 + (3-5)^2 = 4$$

$$\text{dist}^2(\mathbf{x}^{(1)}, \mathbf{m}^{(2)}) = (0-4)^2 + (3-4)^2 = 17$$

K-means algorithm illustration

iteration=1a



t	$\mathbf{x}^{(t)}$
1	(0.0, 3.0)
2	(1.0, 2.0)
3	(2.0, 4.0)
4	(3.0, 0.0)
5	(4.0, 1.0)

t	$c^{(t)}$
1	1
2	?
3	?
4	?
5	?

i	$\mathbf{m}^{(i)}$
1	(0.0, 5.0)
2	(4.0, 4.0)

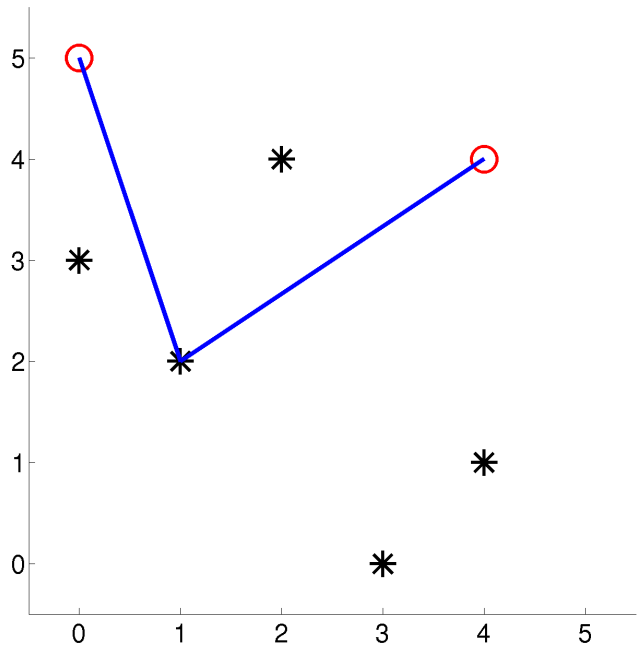
$\text{dist}^2(\mathbf{x}^{(t)}, \mathbf{m}^{(i)})$	$i = 1$	$i = 2$
$t = 1$	4	17
$t = 2$	?	?
$t = 3$	?	?
$t = 4$	?	?
$t = 5$	?	?

$$\text{dist}^2(\mathbf{x}^{(1)}, \mathbf{m}^{(1)}) = (0-0)^2 + (3-5)^2 = 4$$

$$\text{dist}^2(\mathbf{x}^{(1)}, \mathbf{m}^{(2)}) = (0-4)^2 + (3-4)^2 = 17$$

K-means algorithm illustration

iteration=1a



t	$\mathbf{x}^{(t)}$
1	(0.0, 3.0)
2	(1.0, 2.0)
3	(2.0, 4.0)
4	(3.0, 0.0)
5	(4.0, 1.0)

t	$c^{(t)}$
1	1
2	1
3	?
4	?
5	?

i	$\mathbf{m}^{(i)}$
1	(0.0, 5.0)
2	(4.0, 4.0)

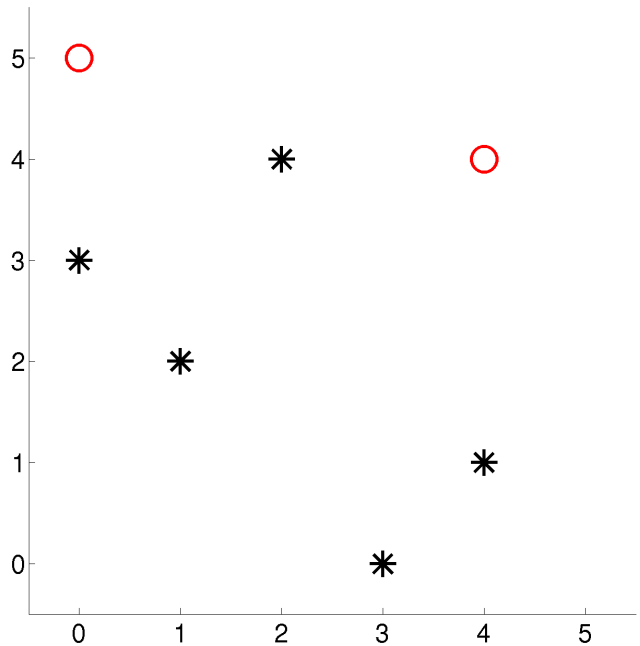
$\text{dist}^2(\mathbf{x}^{(t)}, \mathbf{m}^{(i)})$	$i = 1$	$i = 2$
$t = 1$	4	17
$t = 2$	10	13
$t = 3$	?	?
$t = 4$	?	?
$t = 5$	?	?

$$\text{dist}^2(\mathbf{x}^{(2)}, \mathbf{m}^{(1)}) = (1-0)^2 + (2-5)^2 = 10$$

$$\text{dist}^2(\mathbf{x}^{(2)}, \mathbf{m}^{(2)}) = (1-4)^2 + (2-4)^2 = 13$$

K-means algorithm illustration

iteration=1a



t	$\mathbf{x}^{(t)}$
1	(0.0, 3.0)
2	(1.0, 2.0)
3	(2.0, 4.0)
4	(3.0, 0.0)
5	(4.0, 1.0)

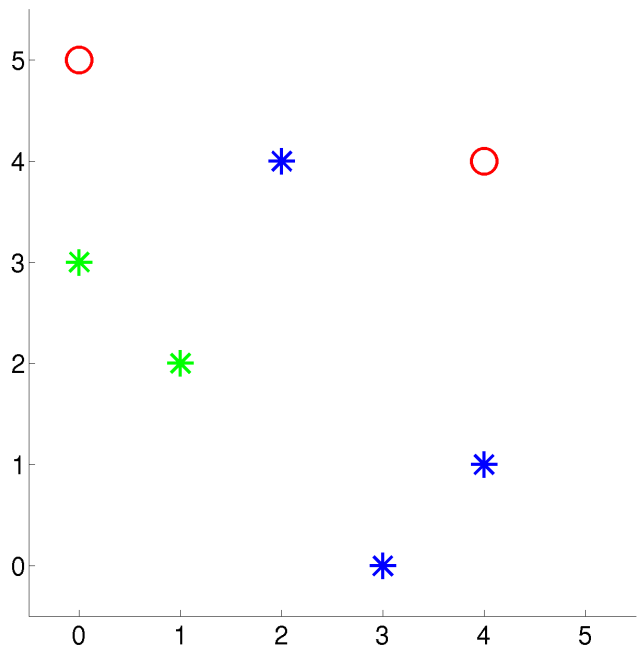
t	$c^{(t)}$
1	1
2	1
3	2
4	2
5	2

i	$\mathbf{m}^{(i)}$
1	(0.0, 5.0)
2	(4.0, 4.0)

$\text{dist}^2(\mathbf{x}^{(t)}, \mathbf{m}^{(i)})$	$i = 1$	$i = 2$
$t = 1$	4	17
$t = 2$	10	13
$t = 3$	5	4
$t = 4$	34	17
$t = 5$	32	9

K-means algorithm illustration

iteration=1b



t	$\mathbf{x}^{(t)}$
1	(0.0, 3.0)
2	(1.0, 2.0)
3	(2.0, 4.0)
4	(3.0, 0.0)
5	(4.0, 1.0)

t	$c^{(t)}$
1	1
2	1
3	2
4	2
5	2

i	$\mathbf{m}^{(i)}$
1	(0.5,2.5)
2	(3,5/3)

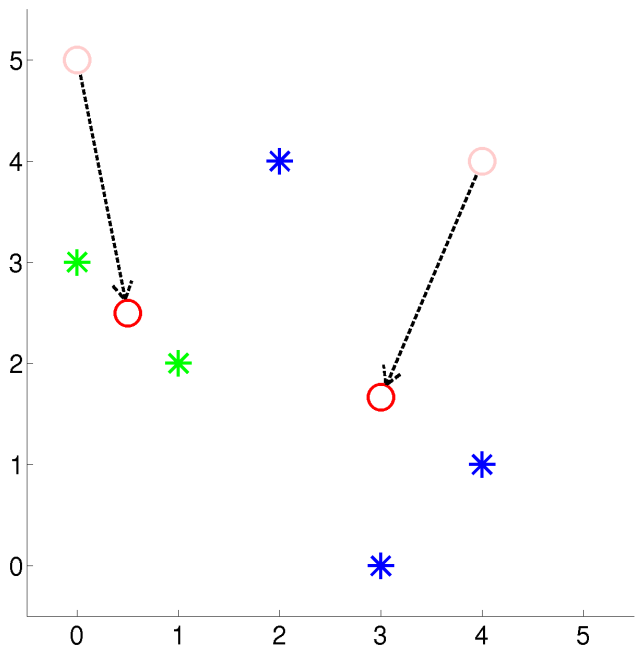
$\text{dist}^2(\mathbf{x}^{(t)}, \mathbf{m}^{(i)})$	$i=1$	$i=2$
$t=1$	4	17
$t=2$	10	13
$t=3$	5	4
$t=4$	34	17
$t=5$	32	9

$\mathbf{m}^{(1)} \leftarrow \left(\frac{1}{2}(0+1), \frac{1}{2}(3+2)\right) = (0.5, 2.5)$

$\mathbf{m}^{(2)} \leftarrow \left(\frac{1}{3}(2+3+4), \frac{1}{3}(4+0+1)\right) = (3, \frac{5}{3})$

K-means algorithm illustration

iteration=1b



t	$\mathbf{x}^{(t)}$
1	(0.0, 3.0)
2	(1.0, 2.0)
3	(2.0, 4.0)
4	(3.0, 0.0)
5	(4.0, 1.0)

t	$c^{(t)}$
1	1
2	1
3	2
4	2
5	2

i	$\mathbf{m}^{(i)}$
1	(0.5,2.5)
2	(3.0,5/3)

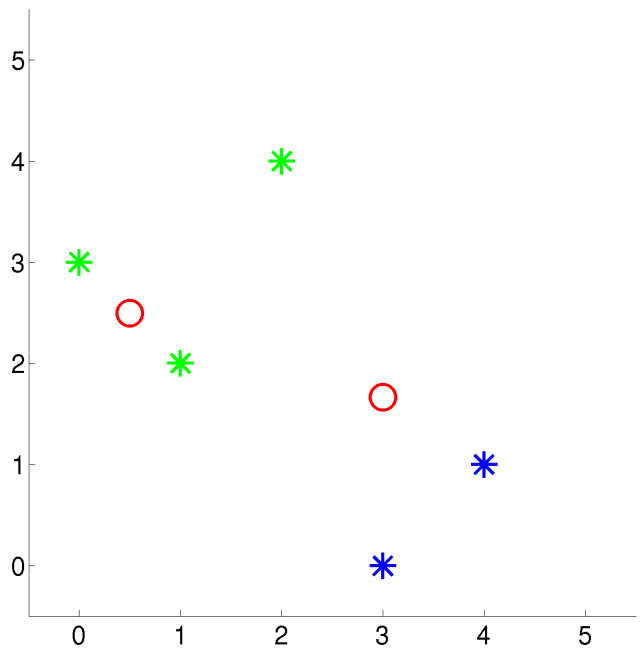
$\text{dist}^2(\mathbf{x}^{(t)}, \mathbf{m}^{(i)})$	$i=1$	$i=2$
$t=1$	4	17
$t=2$	10	13
$t=3$	5	4
$t=4$	34	17
$t=5$	32	9

$\mathbf{m}^{(1)} \leftarrow \left(\frac{1}{2}(0+1), \frac{1}{2}(3+2)\right) = (0.5, 2.5)$

$\mathbf{m}^{(2)} \leftarrow \left(\frac{1}{3}(2+3+4), \frac{1}{3}(4+0+1)\right) = (3, \frac{5}{3})$

K-means algorithm illustration

iteration=2a



t	$\mathbf{x}^{(t)}$
1	(0.0, 3.0)
2	(1.0, 2.0)
3	(2.0, 4.0)
4	(3.0, 0.0)
5	(4.0, 1.0)

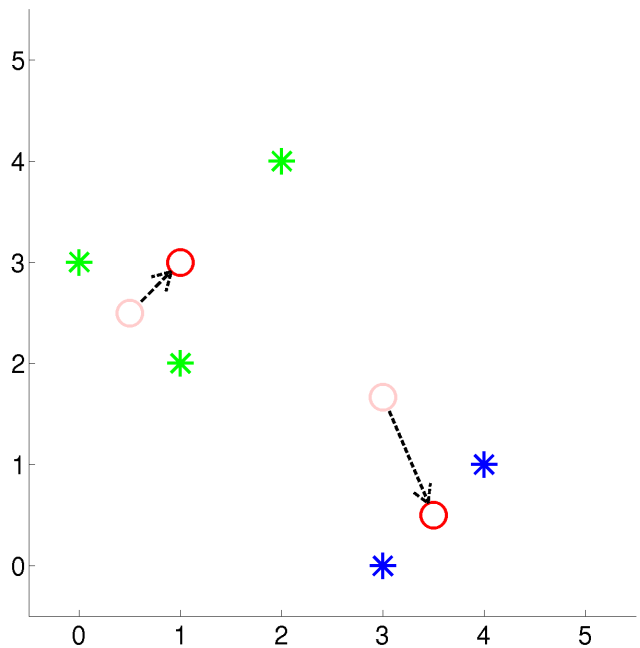
t	$c^{(t)}$
1	1
2	1
3	1
4	2
5	2

i	$\mathbf{m}^{(i)}$
1	(0.5,2.5)
2	(3.0,5/3)

$\text{dist}^2(\mathbf{x}^{(t)}, \mathbf{m}^{(i)})$	$i=1$	$i=2$
$t=1$	0.5	10.8
$t=2$	0.5	4.1
$t=3$	4.5	6.4
$t=4$	12.5	2.8
$t=5$	14.5	1.4

K-means algorithm illustration

iteration=2b



t	$\mathbf{x}^{(t)}$
1	(0.0, 3.0)
2	(1.0, 2.0)
3	(2.0, 4.0)
4	(3.0, 0.0)
5	(4.0, 1.0)

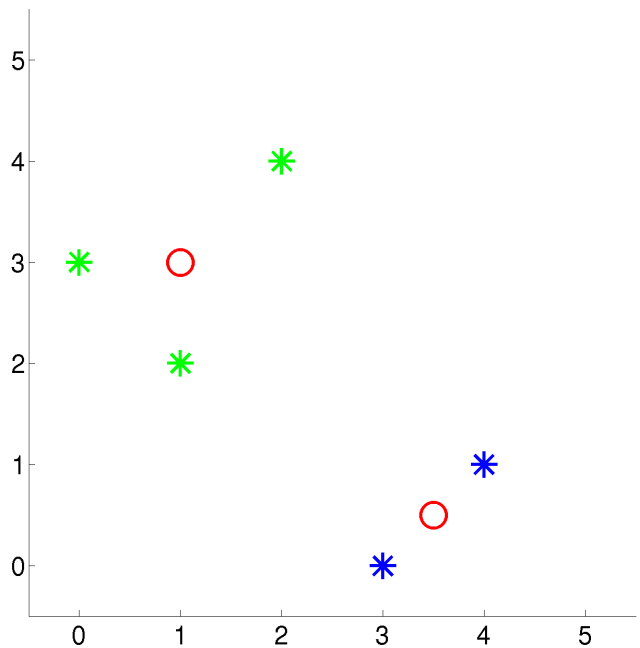
t	$c^{(t)}$
1	1
2	1
3	1
4	2
5	2

i	$\mathbf{m}^{(i)}$
1	(1.0,3.0)
2	(3.5,0.5)

$\text{dist}^2(\mathbf{x}^{(t)}, \mathbf{m}^{(i)})$	$i = 1$	$i = 2$
$t = 1$	0.5	10.8
$t = 2$	0.5	4.1
$t = 3$	4.5	6.4
$t = 4$	12.5	2.8
$t = 5$	14.5	1.4

K-means algorithm illustration

iteration=3 (converged)



t	$\mathbf{x}^{(t)}$
1	(0.0, 3.0)
2	(1.0, 2.0)
3	(2.0, 4.0)
4	(3.0, 0.0)
5	(4.0, 1.0)

t	$c^{(t)}$
1	1
2	1
3	1
4	2
5	2

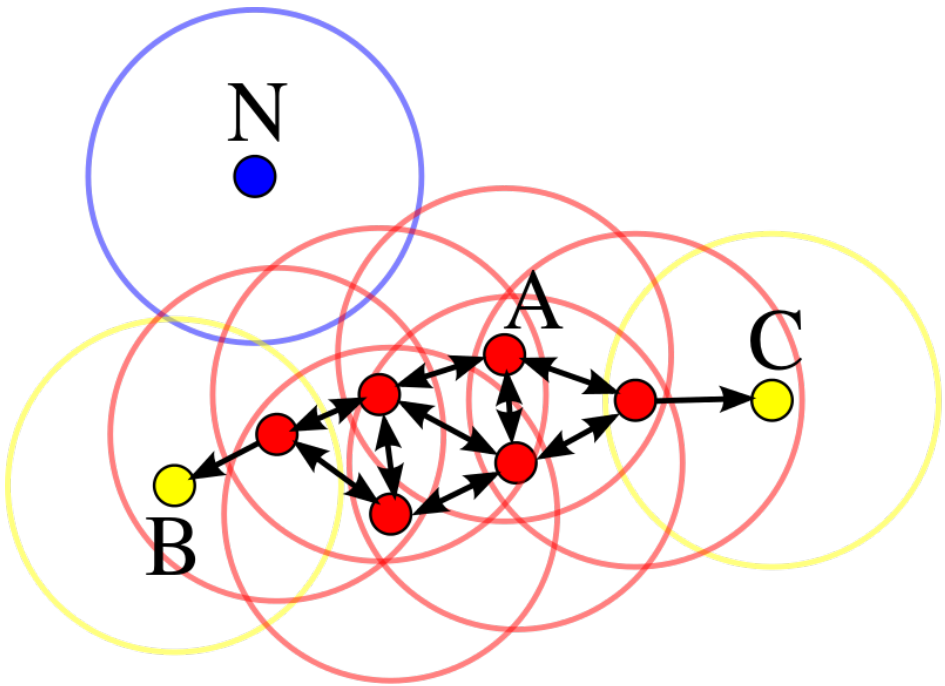
i	$\mathbf{m}^{(i)}$
1	(1.0,3.0)
2	(3.5,0.5)

$\text{dist}^2(\mathbf{x}^{(t)}, \mathbf{m}^{(i)})$	$i=1$	$i=2$
$t=1$	0.5	10.8
$t=2$	0.5	4.1
$t=3$	4.5	6.4
$t=4$	12.5	2.8
$t=5$	14.5	1.4

DBSCAN

Density-Based Spatial Clustering of Applications with Noise

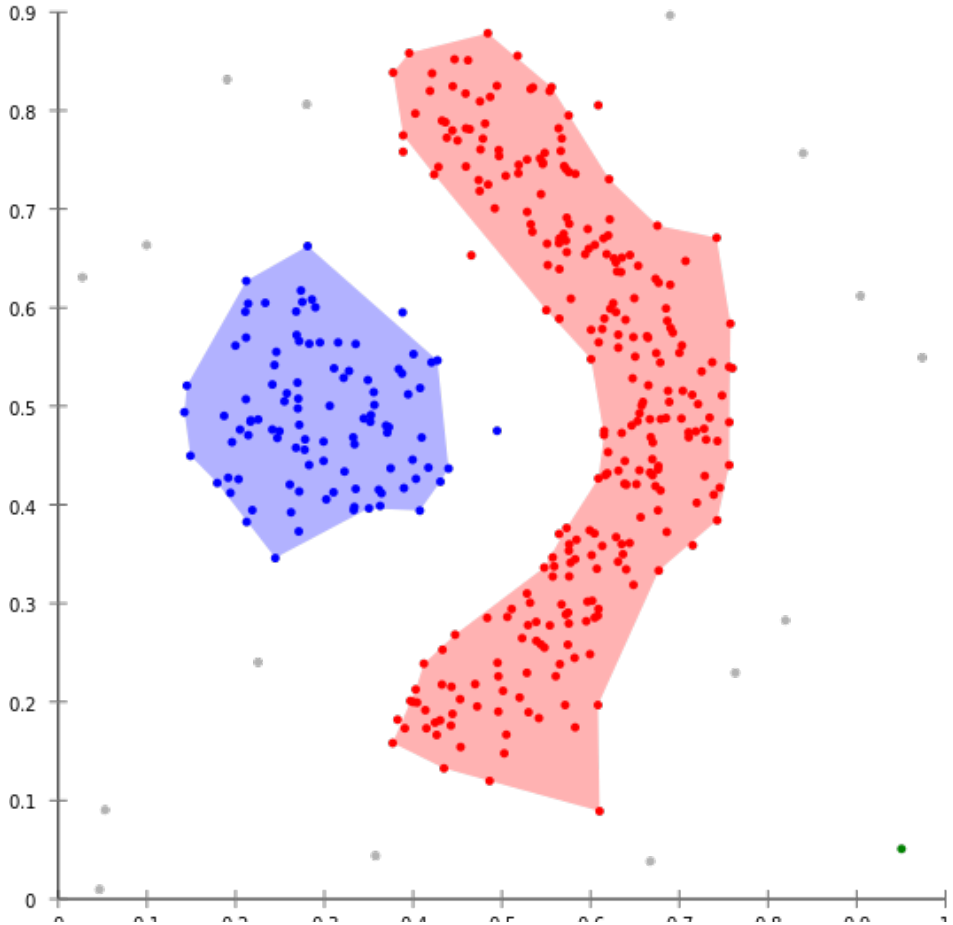
- Density = number of points within ϵ
- core point: density$\geq$minPts
- border point: density$<$minPts, but with a core-point neighbor
- noise point: otherwise



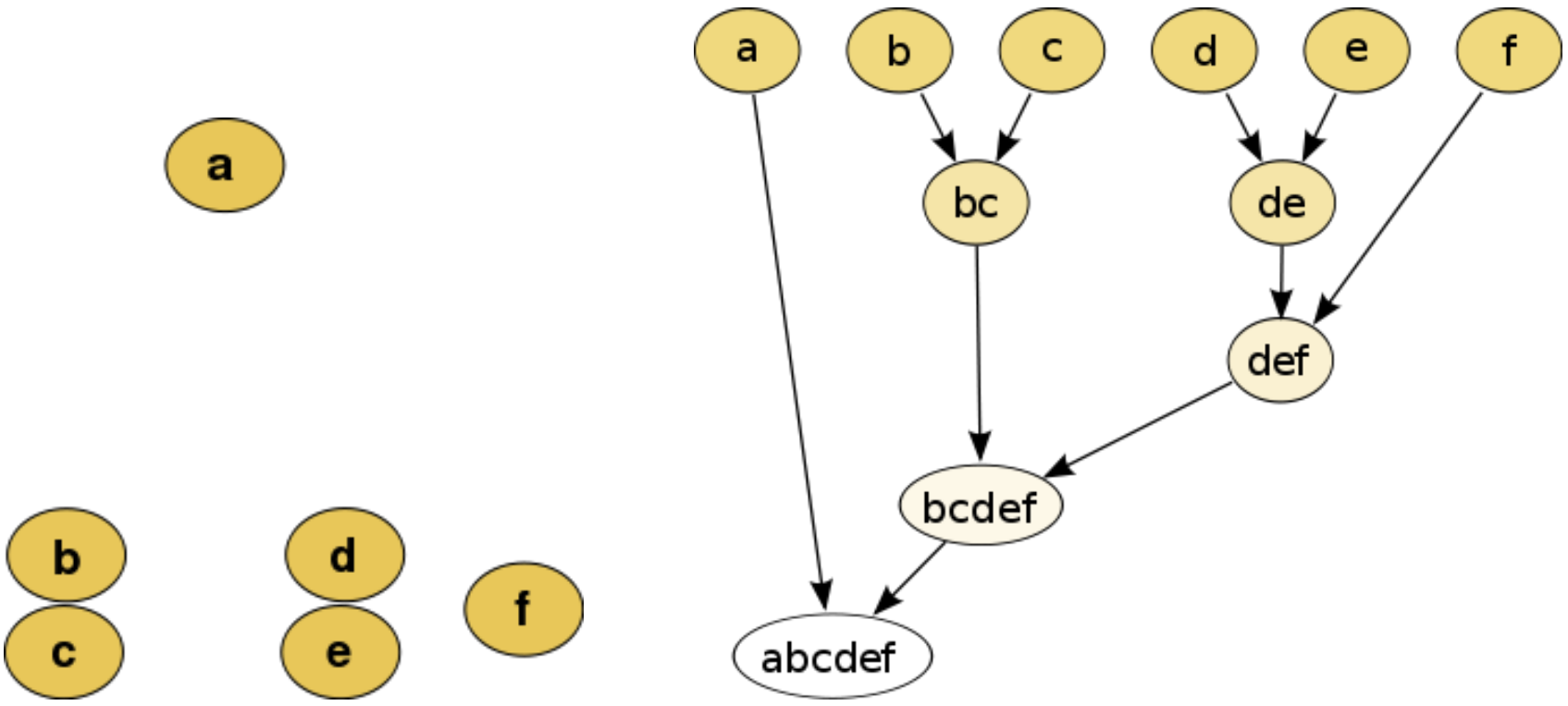
DBSCAN pseudocode

```
DBSCAN(DB, distFunc, eps, minPts) {
    C := 0                                                  /* Cluster counter */
    for each point P in database DB {
        if label(P) ≠ undefined then continue               /* Previously processed in inner loop */
        Neighbors N := RangeQuery(DB, distFunc, P, eps)     /* Find neighbors */
        if |N| < minPts then {                              /* Density check */
            label(P) := Noise                               /* Label as Noise */
            continue
        }
        C := C + 1                                          /* next cluster label */
        label(P) := C                                       /* Label initial point */
        SeedSet S := N \ {P}                                /* Neighbors to expand */
        for each point Q in S {                             /* Process every seed point Q */
            if label(Q) = Noise then label(Q) := C          /* Change Noise to border point */
            if label(Q) ≠ undefined then continue           /* Previously processed (e.g., border point) */
            label(Q) := C                                   /* Label neighbor */
            Neighbors N := RangeQuery(DB, distFunc, Q, eps) /* Find neighbors */
            if |N| ≥ minPts then {                          /* Density check (if Q is a core point) */
                S := S ∪ N                                  /* Add new neighbors to seed set */
            }
        }
    }
}
```

DBSCAN example



Hierarchical Clustering (agglomerative)



Hierarchical Clustering: distance between clusters

- minimum distance between elements of each cluster
- maximum distance between elements of each cluster
- mean distance between elements of each cluster
- median distance between elements of each cluster
- distance between centroids
- etc.
- Different distances give different clusterings
- No universally best

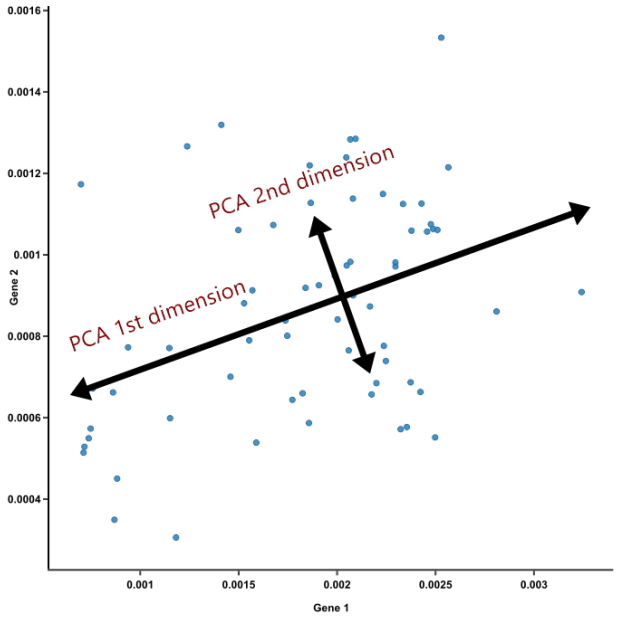
Other Unsupervised Learning Problems

Linear Dimensionality Reduction

Principal Component Analysis (PCA)

- Many real-world raw data has correlated features $\Longrightarrow$ high redundancy
- PCA finds a linear subspace with
  - uncorrelated components
  - least reduction error $\sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{W}\mathbf{W}^T\mathbf{x}_i\|^2$
- Can be solved by truncated eigendecomposition:
  1. Centering: $\mathbf{x}_i \leftarrow \mathbf{x}_i - \text{mean}\left(\{\mathbf{x}_i\}_{i=1}^{N}\right)$
  2. Compute covariance matrix Σ =Covariance$(\{\mathbf{x}_i\}_{i=1}^{N})$
  3. Eigendecomposition: $[\mathbf{W}, \Lambda]$=eigs(Σ, k), where Λ contains the largest k eigenvalues of Σ

PCA example



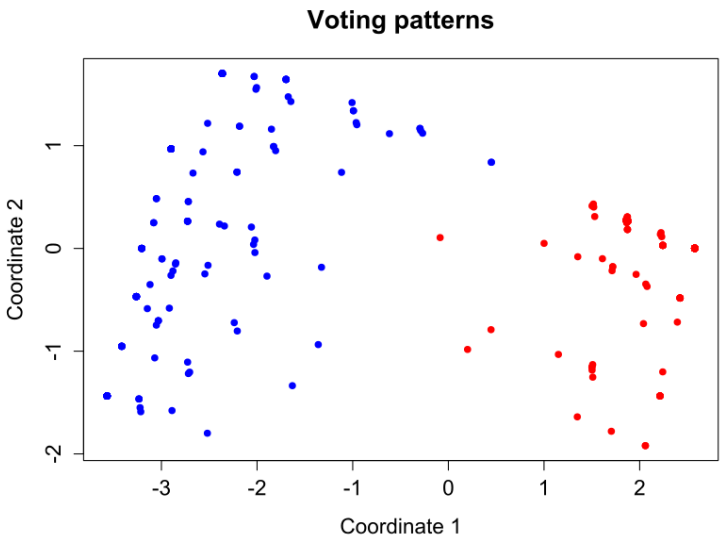
PCA illustration

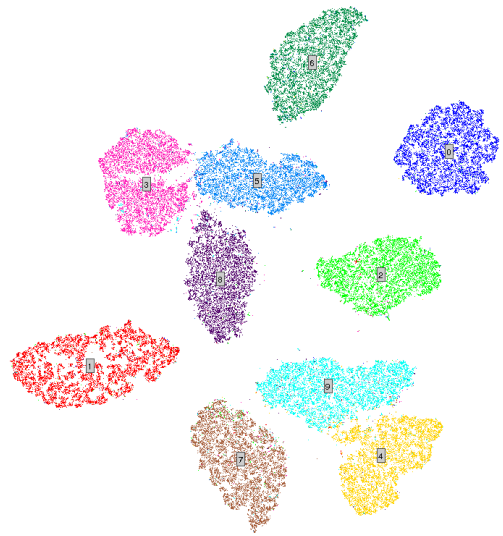
PCA reconstruction

$D = 92 \times 112 \mapsto \text{max. } d = 300$

Nonlinear Dimensionality Reduction (NLDR)

- Multidimensional scaling (MDS)
  - mainly used for visualization
  - tries to preserve (some) original distances
    - D_{ij}=distance between i and j in high-dimensional space
    - d_{ij}=distance between i and j in low-dimensional space
    - $D_{ij} \approx d_{ij}$ for all or some i, j's
- Other NLDR methods for visualizations include t-SNE, UMAP, etc.





MNIST images t-SNE demo

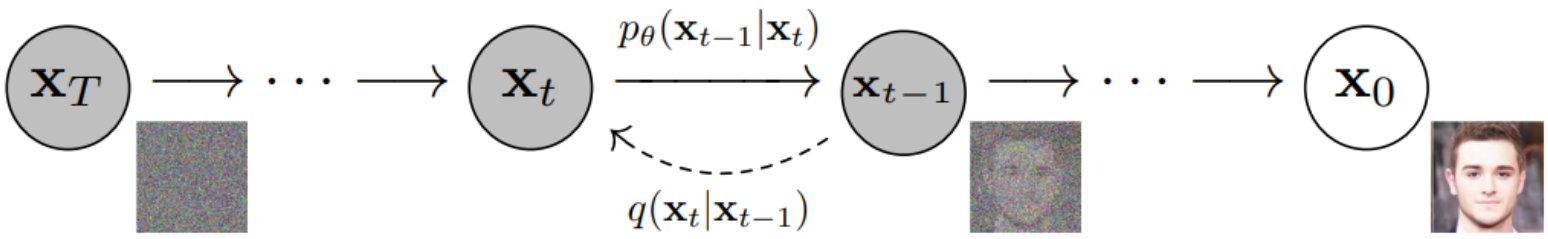
Generation

- $\mathbf{x}$: data in original space
- $\mathbf{z}$: data in latent space
- We have seen tools (e.g., PCA and t-SNE) for $\mathbf{x} \mapsto \mathbf{z}$
- Generation is for $\mathbf{z} \mapsto \mathbf{x}$
  - Use $\{\mathbf{x}_i\}_{i=1}^N$ to train a generator $\mathcal{G}$
  - generate $\mathbf{x} \leftarrow \mathcal{G}(\mathbf{z})$
  - $\mathbf{z}$ is often just noise in unsupervised learning
  - Goal: the generated and the training have similar distributions

Examples

- This person does not exist
- Midjourney showcase
- Nightcafe (a free text2image generator)
- Mubert music generator
- Synthesia video generator [Demo]

Diffusion model for genration



Self-Supervised Learning

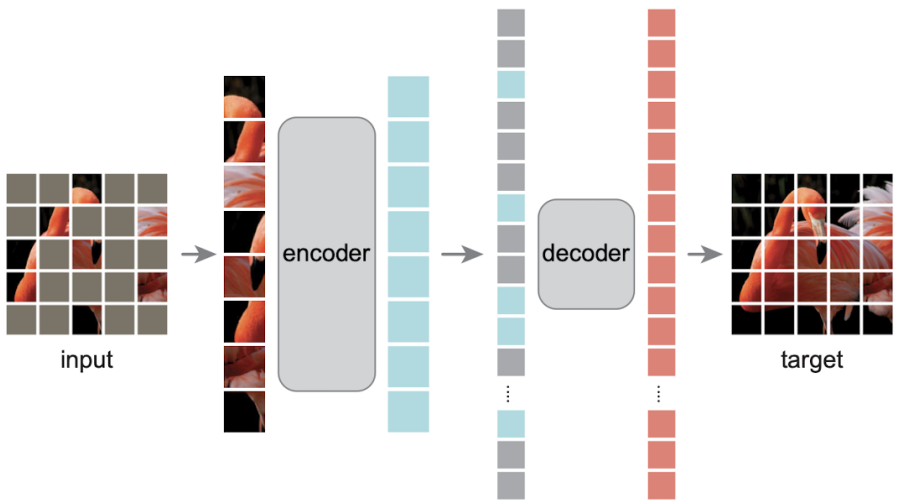
Self-Supervised Learning (SSL)

- is a type of unsupervised learning (uses only $\{x_i\}_{i=1}^N$)
- organized as a supervised form (we will see different forms)
  - supervised: $y_i \approx \hat{y} = f(x_i)$ for $i = 1, \ldots, N$
  - SSL replaces y_i's with some information from $\{x_i\}_{i=1}^N$.
- Goals:
  - to pretrain (to get common senses)
  - to learn a better representation of x_i's
- Almost all winners use SSL (in vision, text, speech, music, etc.)

Masked learning: a popular approach

Consider a sequence $x_1, x_2, \ldots, x_T$ for example

- Causal: $x_i \approx \hat{x}_i = f(x_1, \ldots, x_{i-1})$
  - e.g., `Oslo is Norway's _____.`
- Non-causal: $x_i \approx \hat{x}_i = f(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_T)$
  - e.g., `Oslo is the _____ of Norway.`
  - e.g., for an image



TDT05 Self-Supervised Learning course

- A seminar course dedicated to SSL
- Course webpage
- Registration link