October 21, 2024

Inter-University Research Institute Corporation / Research Organization of Information and Systems
National Institute of Informatics

Mechanisms of Machine Learning

Introduction to Intelligent Systems Science II

Mahito Sugiyama

Learning from Examples (Generalization)

[Schoelkopf, 2013]

- 1, 2, 4, 7, ... $\rightarrow$ What are succeeding numbers?

Learning from Examples (Generalization)

[Schoelkopf, 2013]

- 1, 2, 4, 7, ... $\rightarrow$ What are succeeding numbers?

  1, 2, 4, 7, 11, 16, ... $(a_n = a_{n-1} + n - 1)$

Learning from Examples (Generalization)
[Schoelkopf, 2013]

- 1, 2, 4, 7, ... $\rightarrow$ What are succeeding numbers?

  1, 2, 4, 7, 11, 16, ... $(a_n = a_{n-1} + n - 1)$
  1, 2, 4, 7, 12, 20, ... $(a_n = a_{n-1} + a_{n-2} + 1)$

Learning from Examples (Generalization)

[Schoelkopf, 2013]

- 1, 2, 4, 7, ... $\rightarrow$ What are succeeding numbers?

  1, 2, 4, 7, 11, 16, ... $(a_n = a_{n-1} + n - 1)$
  1, 2, 4, 7, 12, 20, ... $(a_n = a_{n-1} + a_{n-2} + 1)$
  1, 2, 4, 7, 13, 24, ... $(a_n = a_{n-1} + a_{n-2} + a_{n-3})$

Learning from Examples (Generalization)

[Schoelkopf, 2013]

- 1, 2, 4, 7, ... $\rightarrow$ What are succeeding numbers?

  1, 2, 4, 7, 11, 16, ... $(a_n = a_{n-1} + n - 1)$
  1, 2, 4, 7, 12, 20, ... $(a_n = a_{n-1} + a_{n-2} + 1)$
  1, 2, 4, 7, 13, 24, ... $(a_n = a_{n-1} + a_{n-2} + a_{n-3})$
  1, 2, 4, 7, 14, 28 (divisors of 28)

Learning from Examples (Generalization)

[Schoelkopf, 2013]

- 1, 2, 4, 7, ... → What are succeeding numbers?

  1, 2, 4, 7, 11, 16, ... $(a_n = a_{n-1} + n - 1)$
  1, 2, 4, 7, 12, 20, ... $(a_n = a_{n-1} + a_{n-2} + 1)$
  1, 2, 4, 7, 13, 24, ... $(a_n = a_{n-1} + a_{n-2} + a_{n-3})$
  1, 2, 4, 7, 14, 28 (divisors of 28)
  1, 2, 4, 7, 1, 1, 5, ... $(\pi = 3.1415\ldots$ and $e = 2.718\ldots)$

Learning from Examples (Generalization)

[Schoelkopf, 2013]

- 1, 2, 4, 7, ... $\rightarrow$ What are succeeding numbers?

  1, 2, 4, 7, 11, 16, ... $(a_n = a_{n-1} + n - 1)$
  1, 2, 4, 7, 12, 20, ... $(a_n = a_{n-1} + a_{n-2} + 1)$
  1, 2, 4, 7, 13, 24, ... $(a_n = a_{n-1} + a_{n-2} + a_{n-3})$
  1, 2, 4, 7, 14, 28 (divisors of 28)
  1, 2, 4, 7, 1, 1, 5, ... $(\pi = 3.1415\ldots$ and $e = 2.718\ldots)$
- More than 1385 rules! (https://oeis.org)

What is Learning?

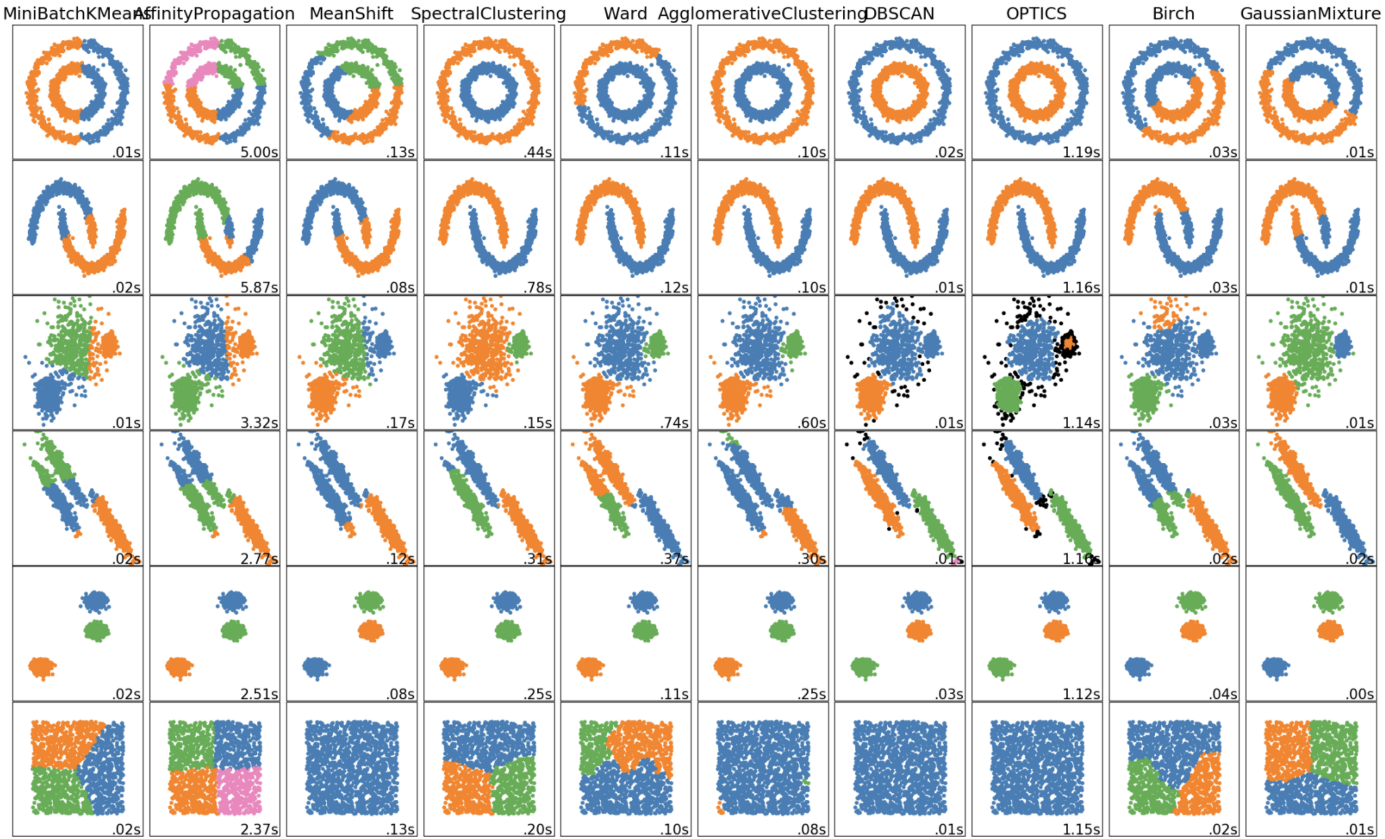
- Which is “correct” answer (generalization)?
  $\rightarrow$ No way to answer (any rule could be possible)
  - There is no “universal” answer
  - Ref: The Ugly Duckling theorem, No free lunch theorem
- Purpose of ML:
  Find rules that generalize experience (data)
  - Predicting future as well as explaining past

Components of Learning

1. What are targets of learning?
2. How to represent targets (model)?
3. How are data provided to a learner?
4. How does the learner work?
5. How to evaluate results of learning?

Clustering

- Find groups whose members are similar with each other
  - A typical problem in unsupervised learning
- Representative methods:
  - *k*-means, DBSCAN, hierarchical clustering, ...
- Assume that each data point is a d-dimensional feature vector $\boldsymbol{x} \in \mathbb{R}^d$
  - Input is a dataset $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$



A comparison of the clustering algorithms in scikit-learn

From: https://scikit-learn.org/stable/modules/clustering.html

K-means

- K-means is one of the most heavily used algorithm
- The sum of squared errors scoring function:

$$\mathrm{SSE}(\mathcal{C}) = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_k} \|\boldsymbol{x} - \boldsymbol{\mu}_k\|^2 = \sum_{k=1}^{K} \sum_{\boldsymbol{x} \in C_k} \sum_{j=1}^{d} (x^j - \mu_k^j)^2$$

  - $\boldsymbol{\mu}_k$ is the mean vector of a cluster C_k
  - Dissimilarity is measured by the squared Euclidean distance

- K-means tries to find the optimal clustering $\mathcal{C}^*$ s.t.

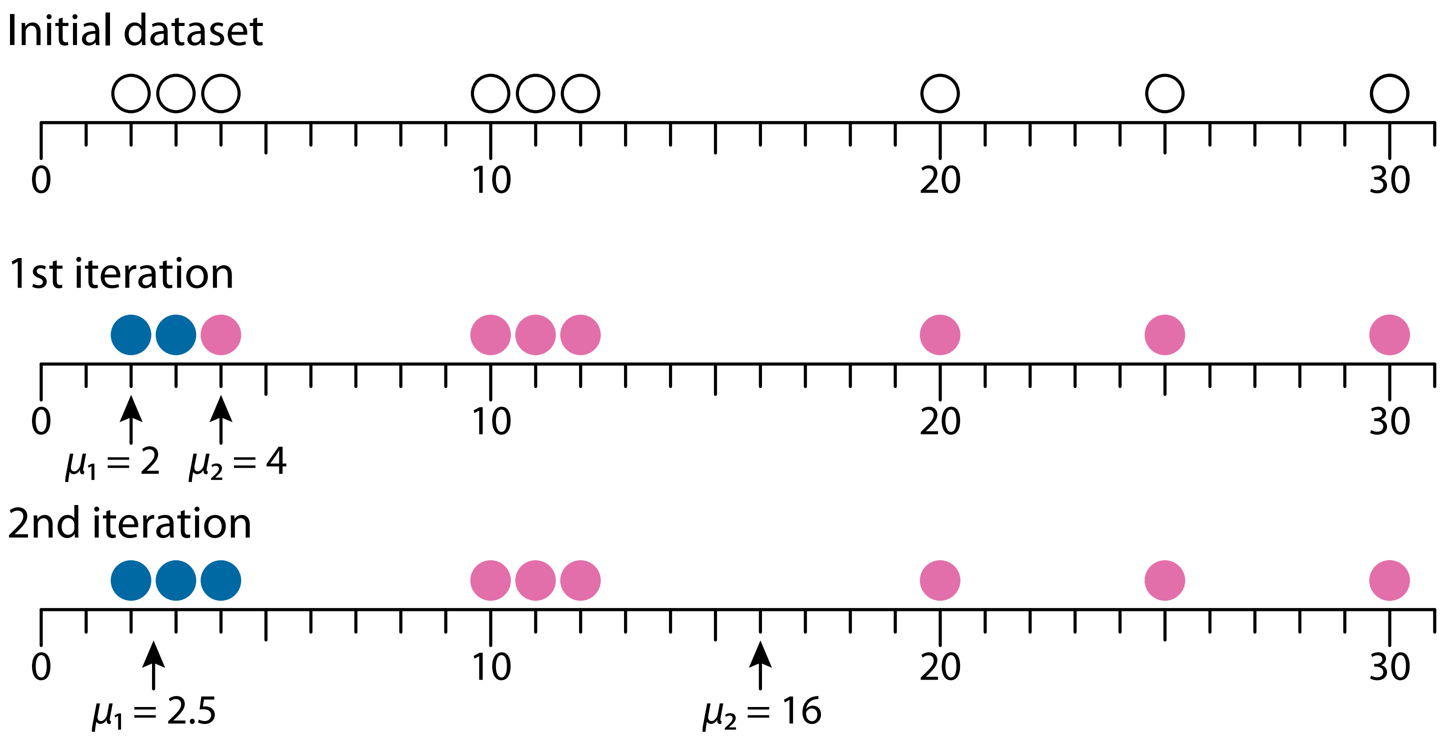
$$\mathcal{C}^* = \operatorname*{argmin}_{\mathcal{C}} \mathrm{SSE}(\mathcal{C})$$

Pseudocode of K-means

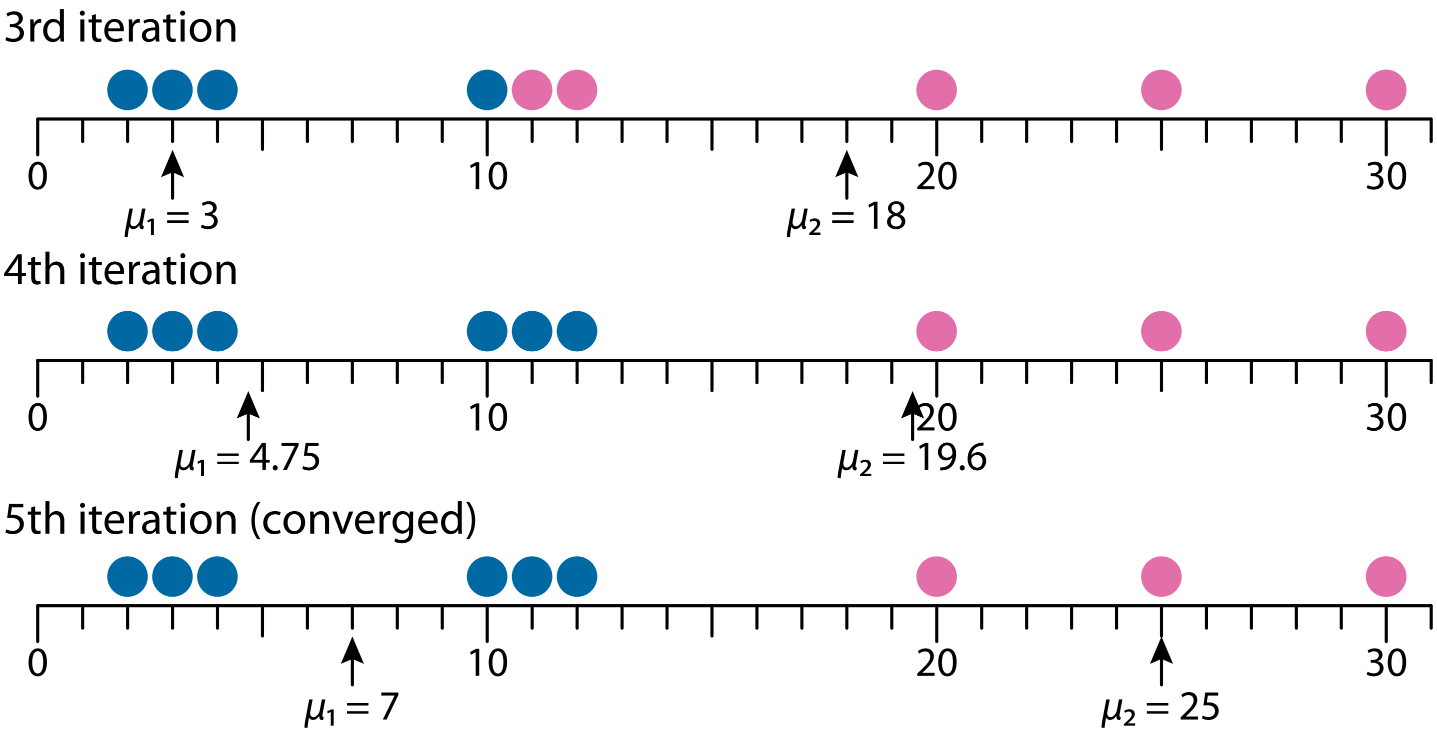
- **Input:** Dataset D, Number of clusters K
- **Output:** Clustering $\mathcal{C}$

1. Randomly initialize K centroids: $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K$
2. **repeat**
3. $\quad C_k \leftarrow \emptyset$ for all $k \in \{1, 2, \dots, K\}$
4. $\quad$ **for each** $\boldsymbol{x} \in D$ **do** // cluster assignment
5. $\quad\quad k^* \leftarrow \text{argmin}_{k \in \{1,2,\dots,K\}} ||\boldsymbol{x} - \boldsymbol{\mu}_k||^2$
6. $\quad\quad C_{k^*} \leftarrow C_{k^*} \cup \{\boldsymbol{x}\}$
7. $\quad$ **for each** $k \in \{1, 2, \dots, K\}$ **do** // centroid update
8. $\quad\quad \boldsymbol{\mu}_k \leftarrow (1/|C_k|) \sum_{\boldsymbol{x} \in C_k} \boldsymbol{x}$
9. **until** cluster assignment does not change

K-means on 1-Dimensional Data



K-means on 1-Dimensional Data



Notes on *K*-means

- K-means is a classic algorithm (proposed in 1967!), while is still the state-of-the-art
  - It is fast; its time complexity is $O(ndK)$
  - Easy to use; there is only one parameter K
- Drawbacks
  - Its result may be a local optimum, not global
  - Its result depends on initialization
  - It cannot detect non-spherical clusters

K-means++

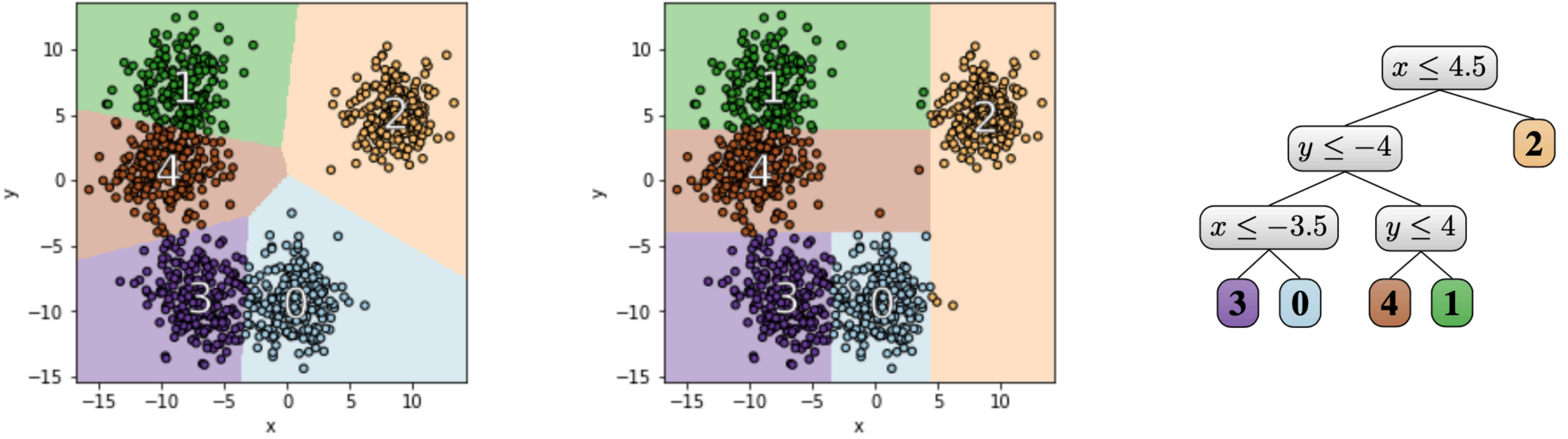
- K-means++ is an algorithm for selecting initial clustering
  - This can alleviate the problem of finding worse clustering than optimal

1. Randomly select a data point $\boldsymbol{x} \in D$ and $\mu_1 \leftarrow \boldsymbol{x}$
2. **for each** $k = \{2, 3, \ldots, K\}$ **do**
3.   **for each** $\boldsymbol{x} \in D$ **do** $D(\boldsymbol{x}) \leftarrow \min_{i \in \{1,2,\ldots,k-1\}} ||\boldsymbol{x} - \boldsymbol{\mu}_i||^2$
4.   **for each** $\boldsymbol{x} \in D$ **do** $p(\boldsymbol{x}) \leftarrow D(\boldsymbol{x}) / \sum_{\boldsymbol{s} \in D} D(\boldsymbol{s})$
5.   Select μ_k from D using the probability distribution $p(\boldsymbol{x})$ for each $\boldsymbol{x} \in D$
6. Perform K-means using $\mu_1, \mu_2, \ldots, \mu_K$ as the initial cluster centers

Recent Advances on *K*-means

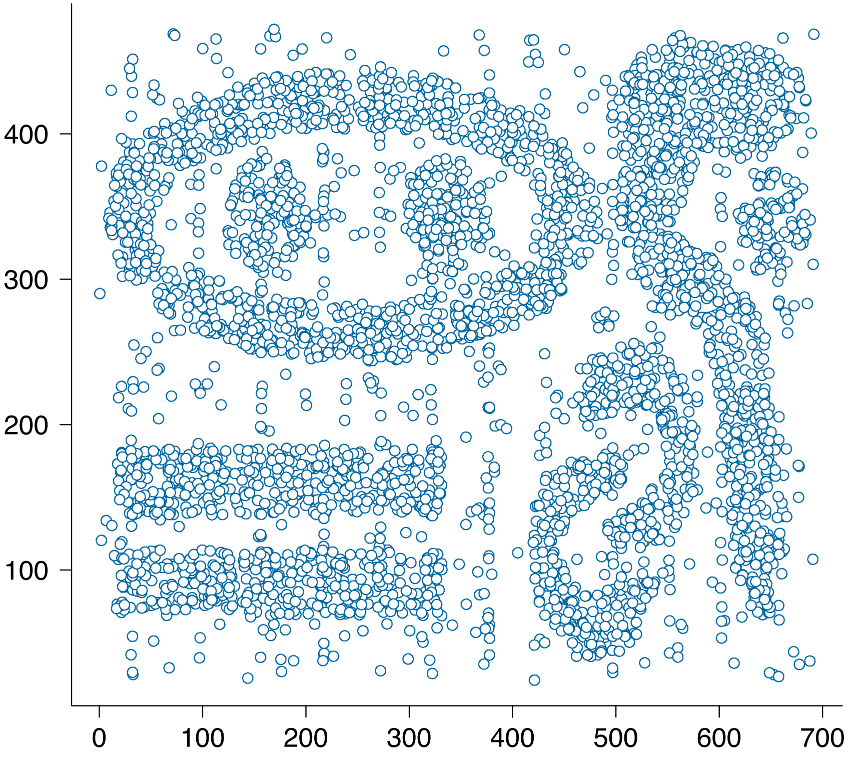
- Recent topics on K-means
  - Moshkovitz, M. et al.: **Explainable k-Means and k-Medians Clustering**, ICML2020
    - Towards interpretable clustering
  - Zhuang, Y., Chen, X., Yang, Y.: **Wasserstein K-means for clustering probability distributions**, NeurIPS2022
    - K-means for probability distributions using the Wasserstein metric
  - Cohen-Addad, V., Esfandiari, H., Mirrokni, V., Narayanan, S.: **Improved approximations for Euclidean k-means and k-median, via nested quasi-independent sets**, STOC2022
    - An improved algorithm and its theoretical analysis

Explainable *k*-Means and *k*-Medians Clustering



(a) Optimal 5-means clusters (b) Tree based 5-means clusters (c) Threshold tree

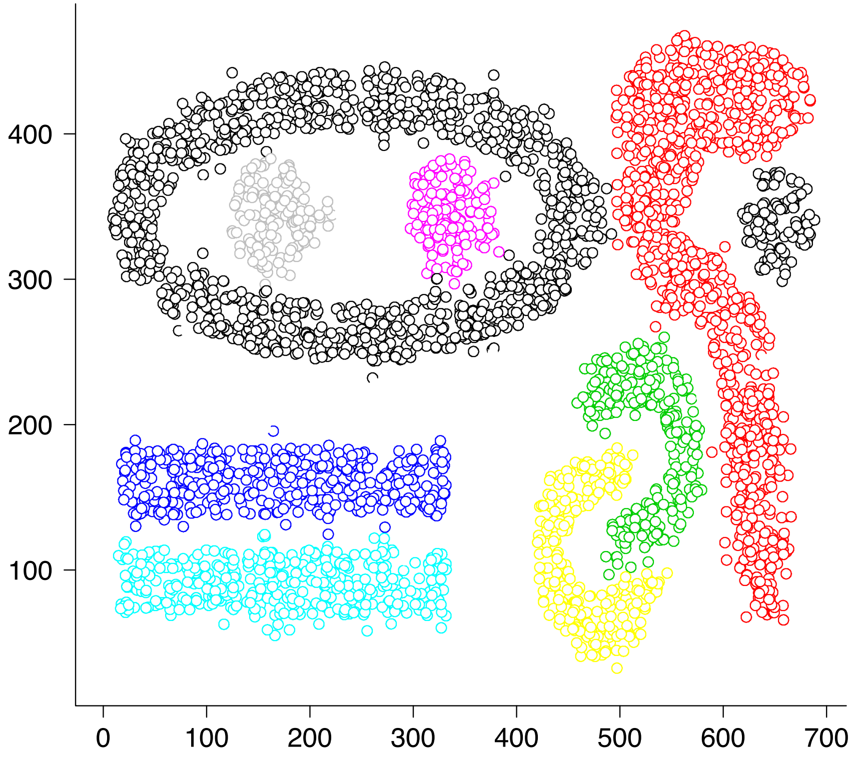
Example of Spatial Clustering



- Dataset $X \subset \mathbb{R}^2$:

1	355.60	270.21
2	549.28	351.71
3	520.08	215.48
4	575.15	166.68
⋮	⋮	⋮
4000	309.395	365.09

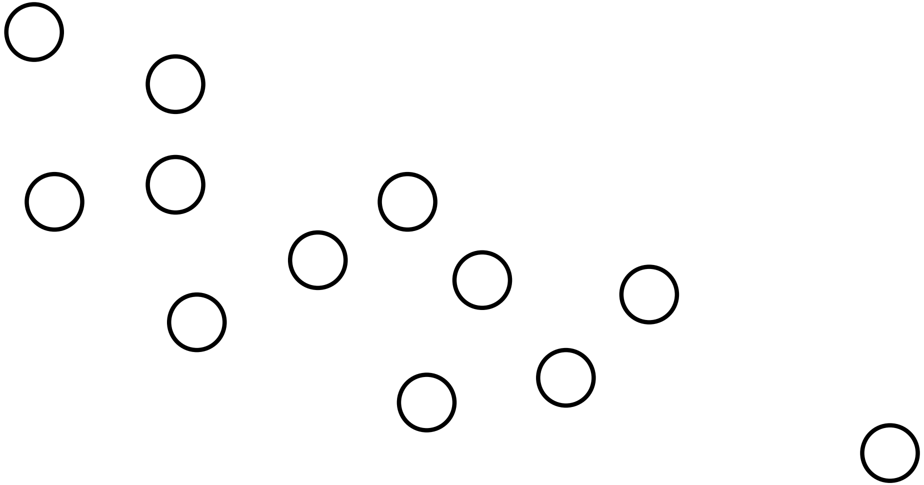
Example of Spacial Clustering Result



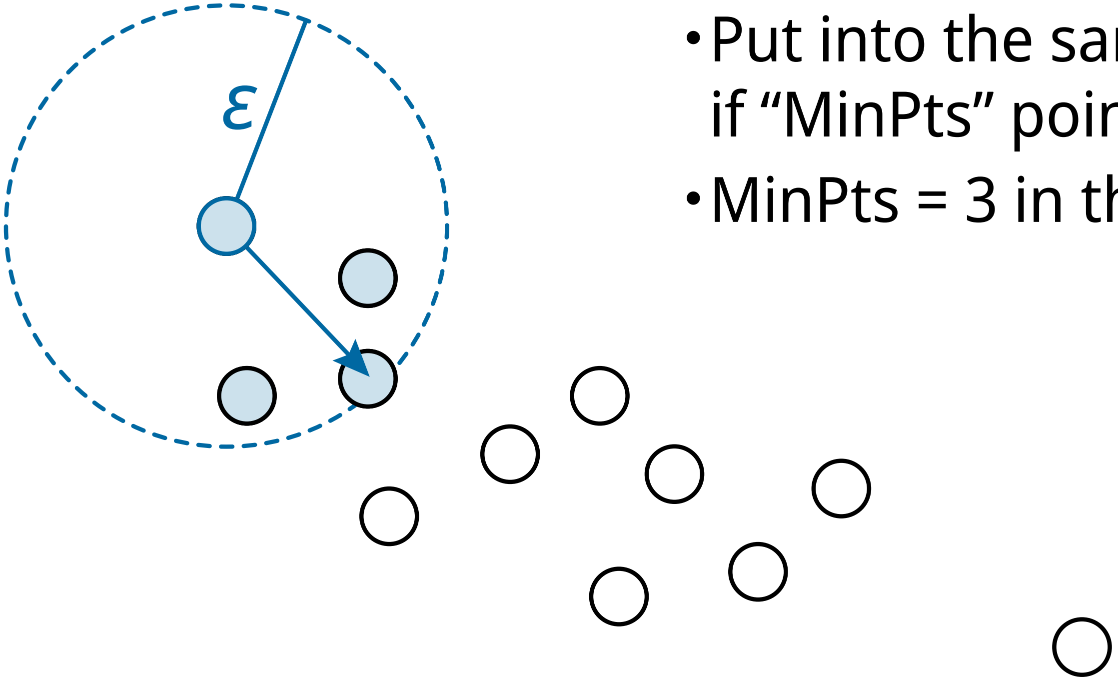
- Dataset $X \subset \mathbb{R}^2$:

1	355.60	270.21
2	549.28	351.71
3	520.08	215.48
4	575.15	166.68
$\vdots$	$\vdots$	$\vdots$
4000	309.395	365.09

DBSCAN [Ester et al., 1996]

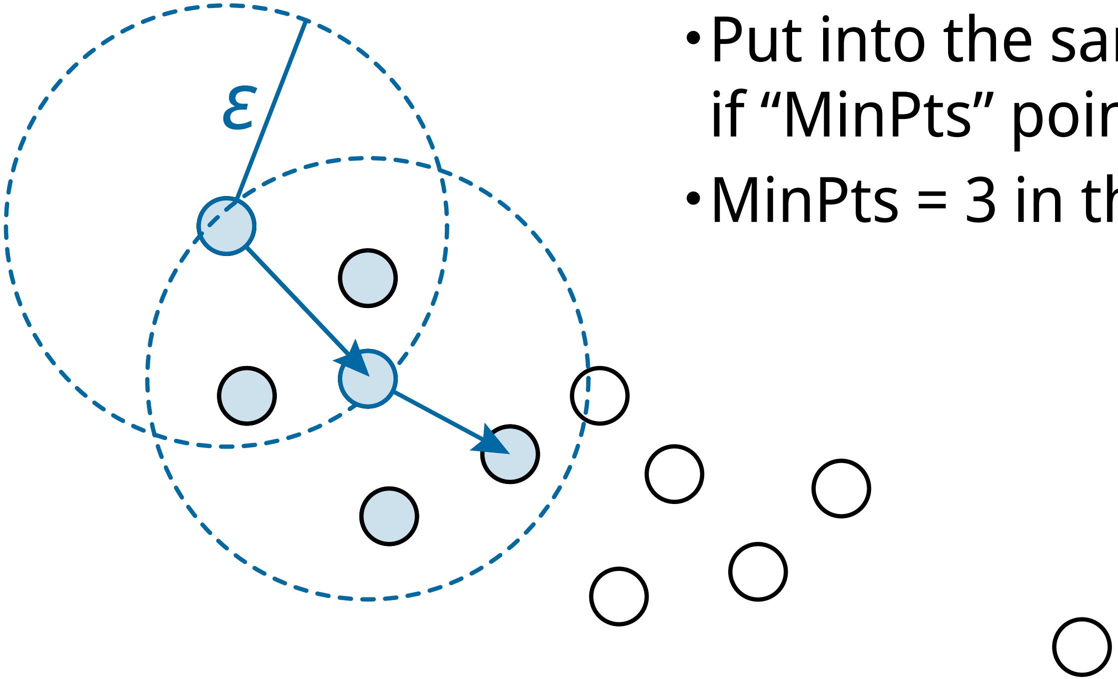


DBSCAN [Ester et al., 1996]



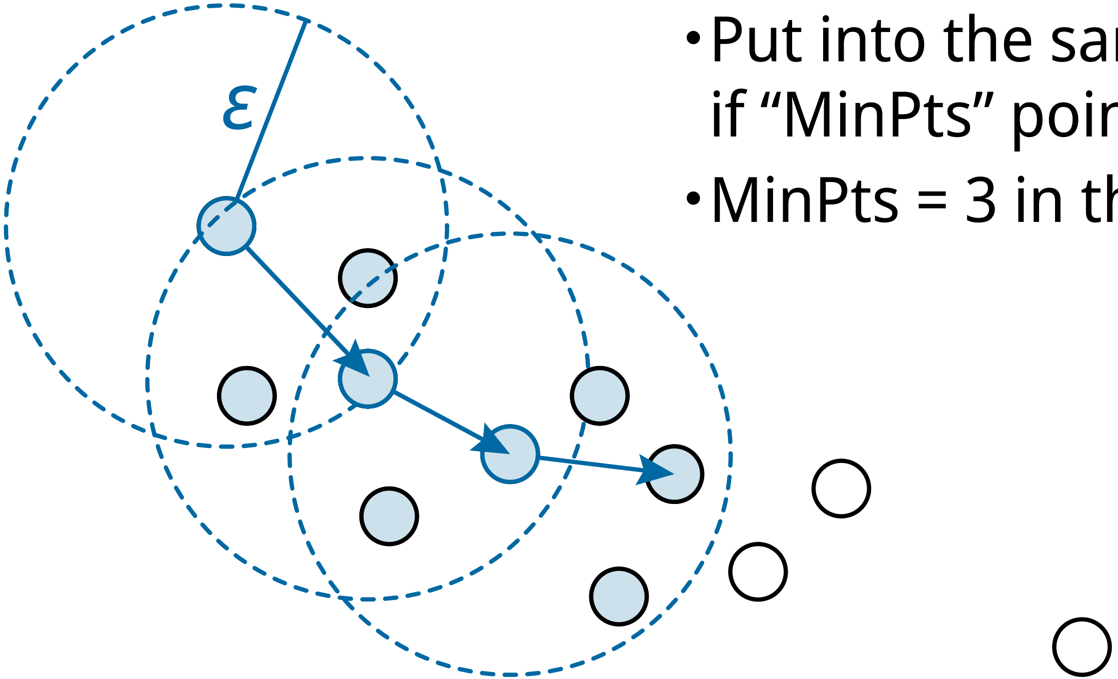
- Put into the same cluster if "MinPts" points are in the circle
- MinPts = 3 in this example

DBSCAN [Ester et al., 1996]



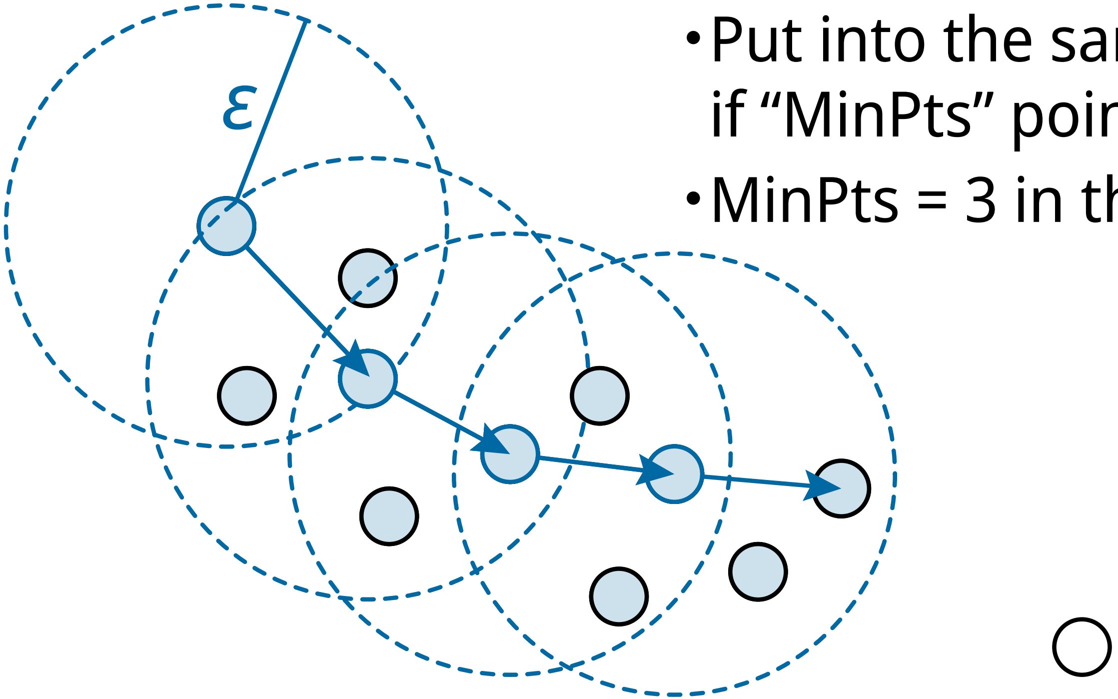
- Put into the same cluster
  if "MinPts" points are in the circle
- MinPts = 3 in this example

DBSCAN [Ester et al., 1996]



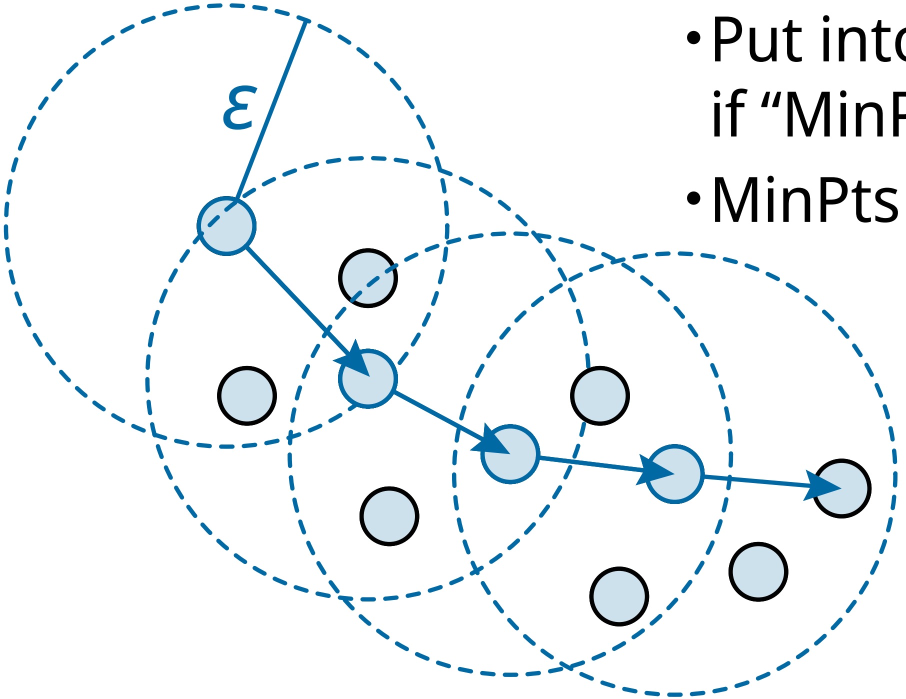
- Put into the same cluster if "MinPts" points are in the circle
- MinPts = 3 in this example

DBSCAN [Ester et al., 1996]



- Put into the same cluster if "MinPts" points are in the circle
- MinPts = 3 in this example

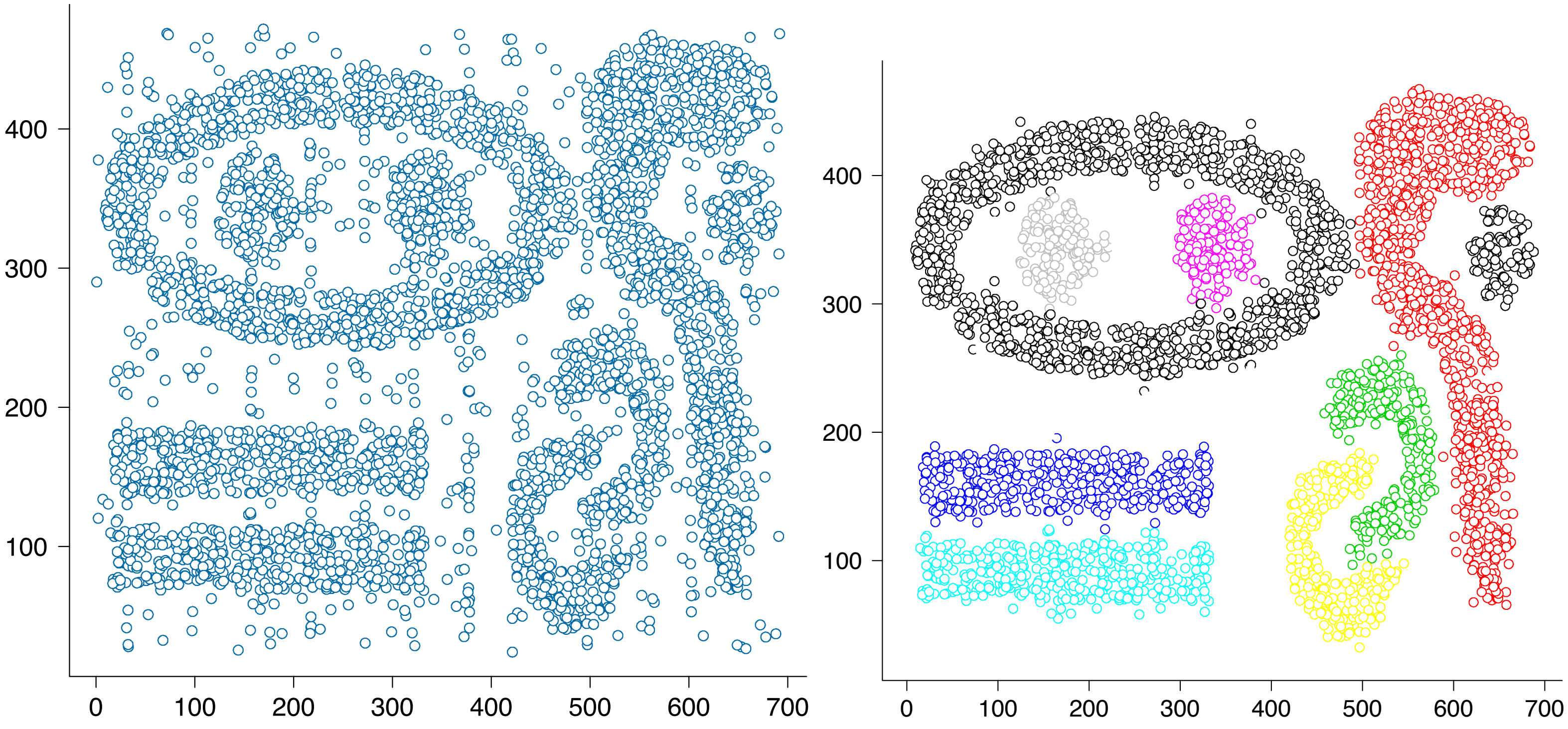
DBSCAN [Ester et al., 1996]



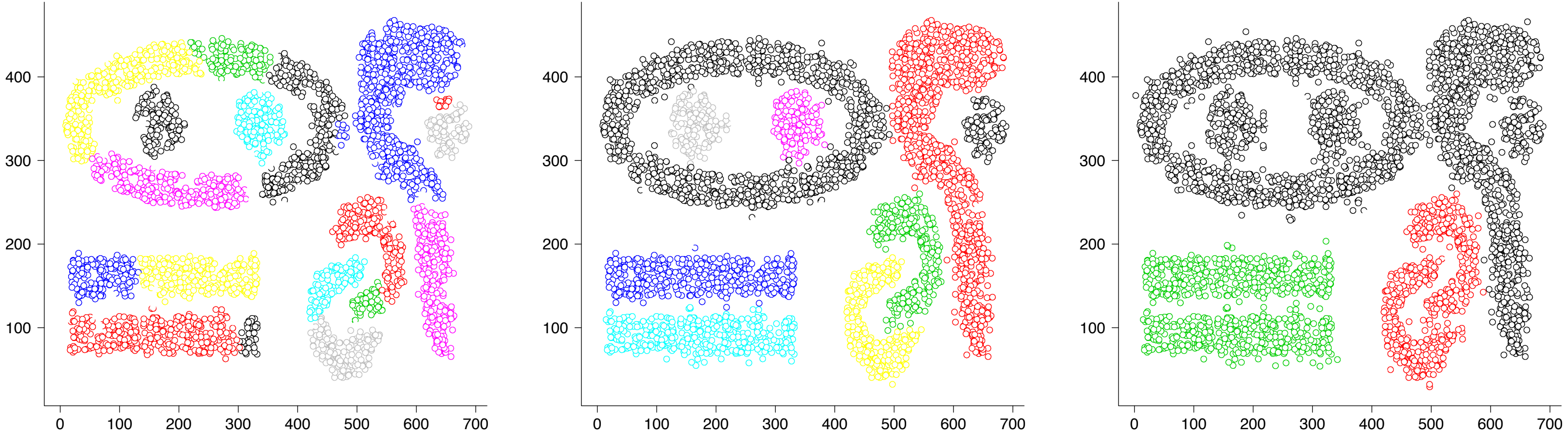
- Put into the same cluster if "MinPts" points are in the circle
- MinPts = 3 in this example

Noise if it is not reachable from any other points

Result of DBSCAN ($\epsilon = 14$, $\mathrm{MinPts} = 10$)



Clustering Results are Arbitrary



- $\epsilon = 12, 14, 16$ (from left to right), MinPts $= 10$
  - Caution needed in interpreting clustering results

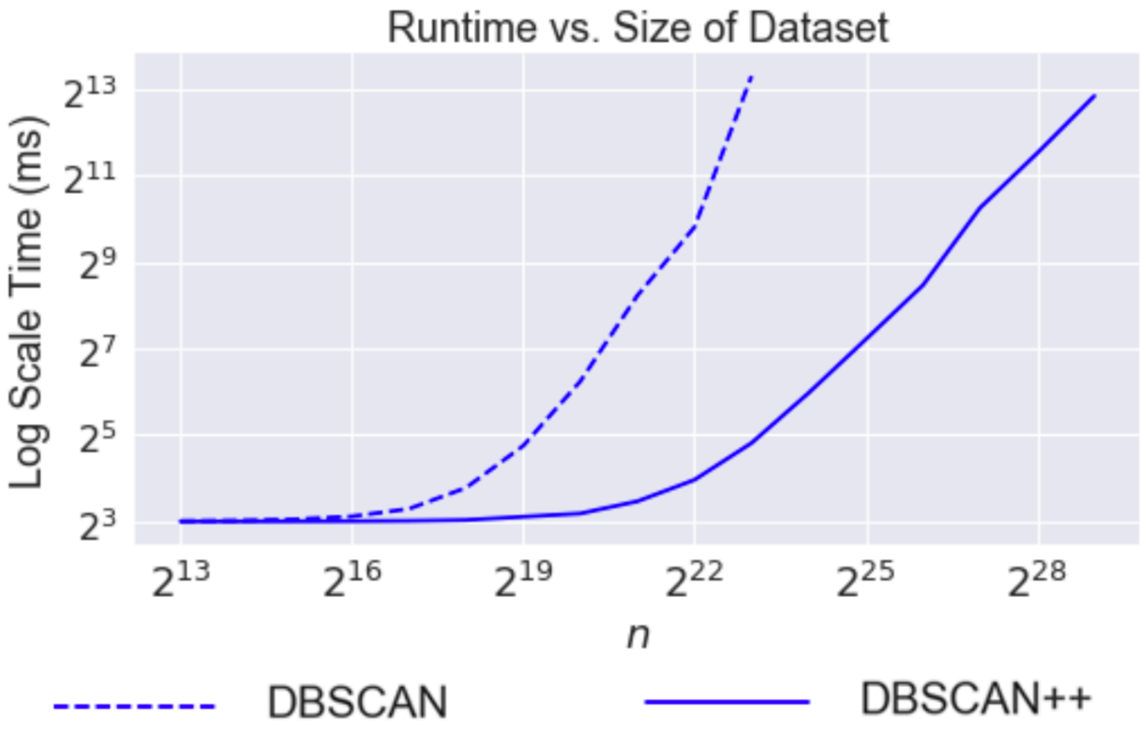
State-of-the-art of DBSCAN

- DBSCAN requires almost $O(n^2)$ for high-dimensional data
  $\rightarrow$ Can we accelerate it?
- There are approaches that address this issue using heuristics
  - It sometimes even improves the clustering performance
    as it works as regularization

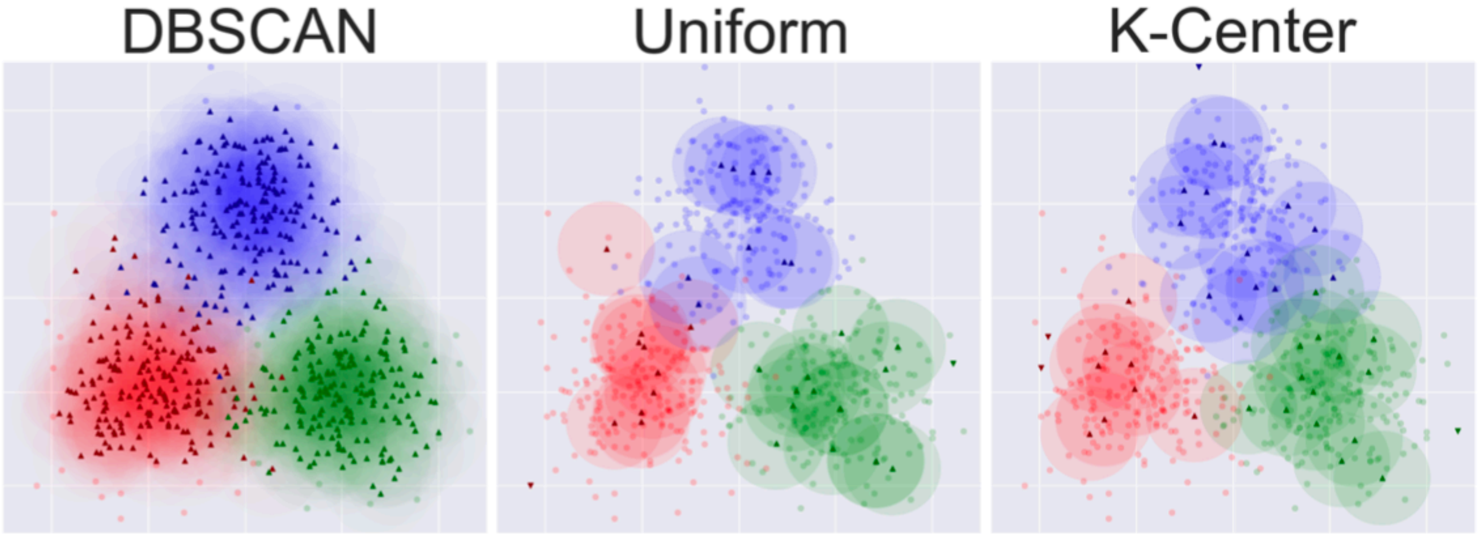
DBSCAN++ [ICML2019]

- Speed-up using sampling
  - Jang, J., Jiang, H.: **DBSCAN++: Towards fast and scalable density clustering**, ICML2019
- Sample m data points
- Compute core points among the m points
  - A point is called core if there are MinPts points in its ϵ-neighborhood
  - So, it is inexact compared to the original DBSCAN,
    but it can achieve competitive results
- The time complexity is $O(nm)$, so it is fast if m is small enough

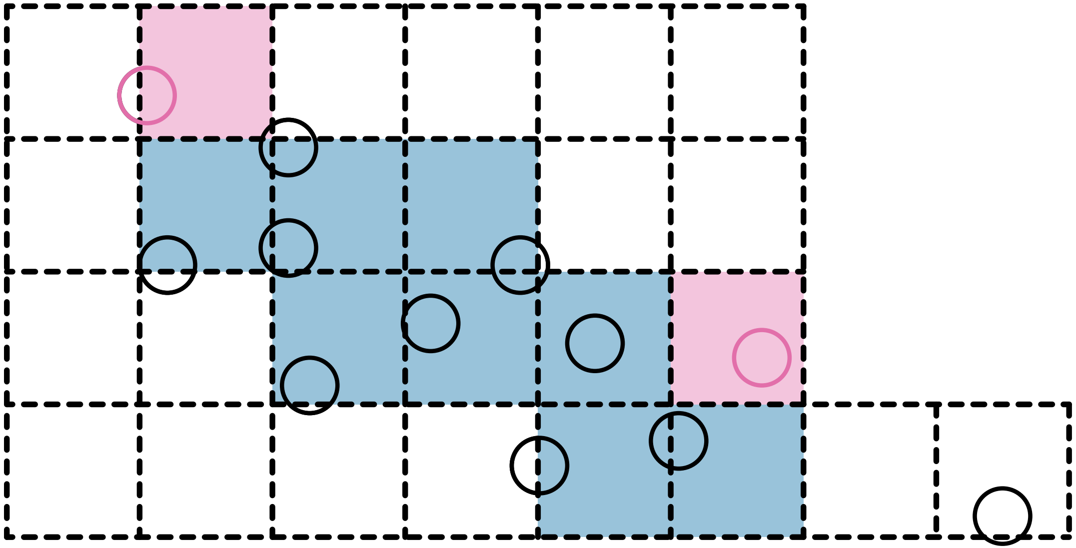
Running time of DBSCAN++ ($d = 3$)



Comparison of Sampling Strategies



BOOL [Sugiyama & Yamamoto, 2011]

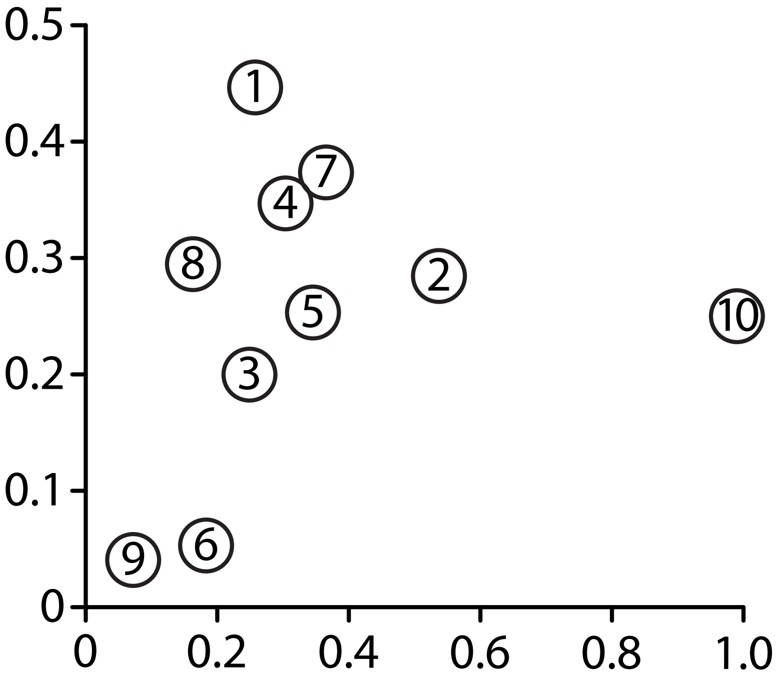


- Discretize data and connect them if contiguous
- Using radix sort, $O(n^2) \Rightarrow O(n)$
- For 10,000 points, 1,000x speedup

Outlier (Anomaly) Detection

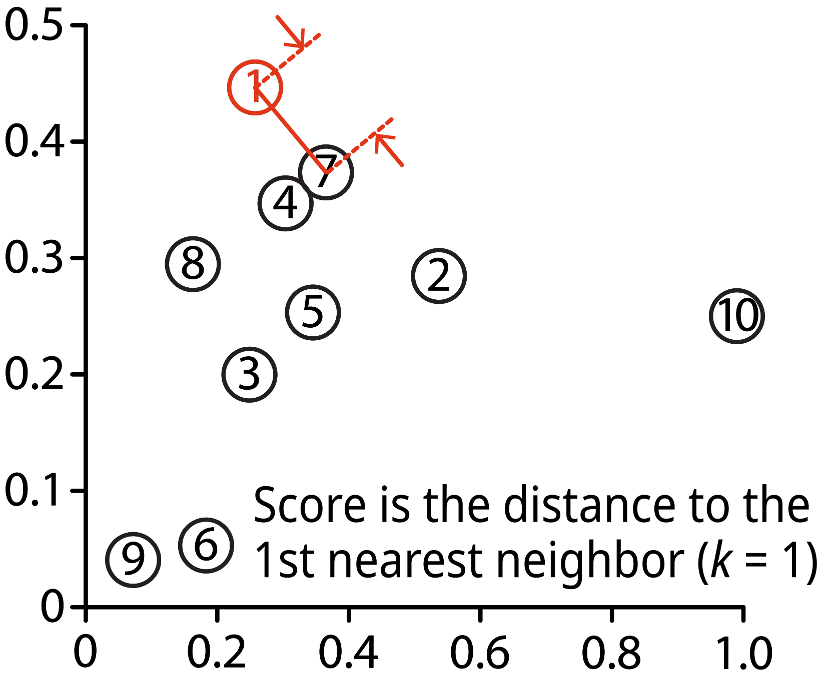
- Find outmost objects (data points)
  - They called outliers or anomalies
- Representative methods:
  - kth-NN, LOF, iForest, …
- Similar to clustering, input is a dataset $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\} \subset \mathbb{R}^d$

*k*th-NN（1/2） [Bay & Schwabacher, 2003]



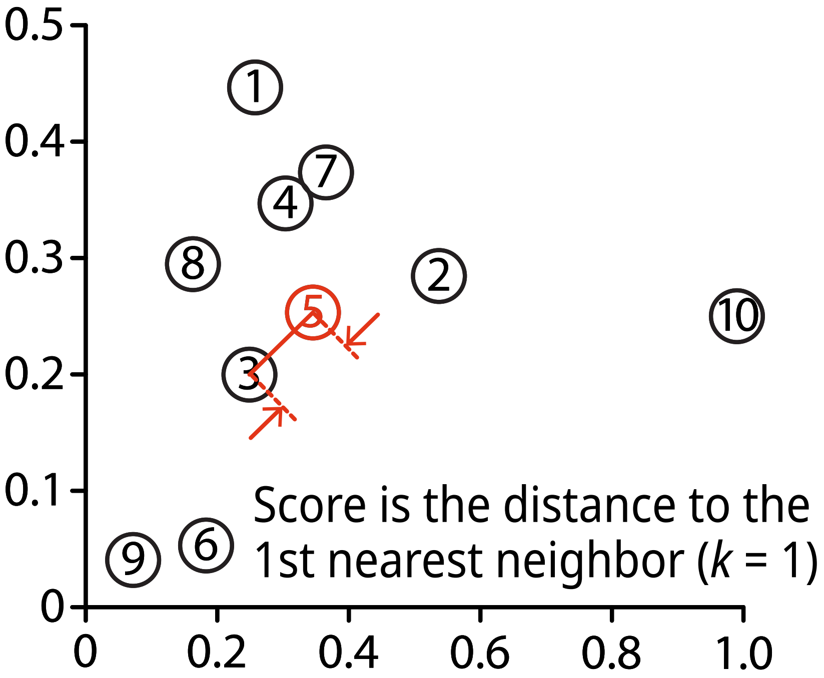
id	score

*k*th-NN（1/2） [Bay & Schwabacher, 2003]



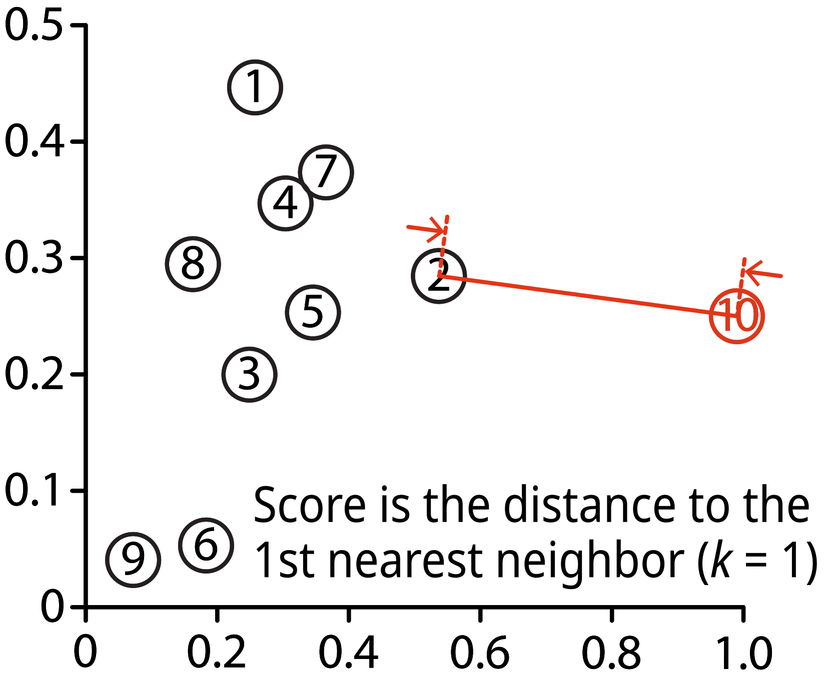
id	score
1	0.109

*k*th-NN（1/2） [Bay & Schwabacher, 2003]



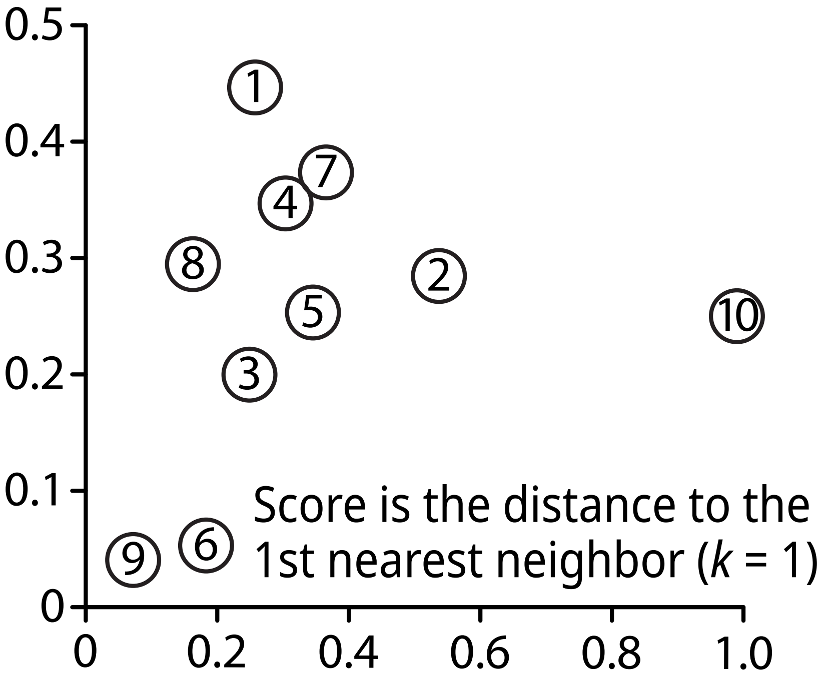
id	score
1	0.109
5	0.103

*k*th-NN（1/2） [Bay & Schwabacher, 2003]



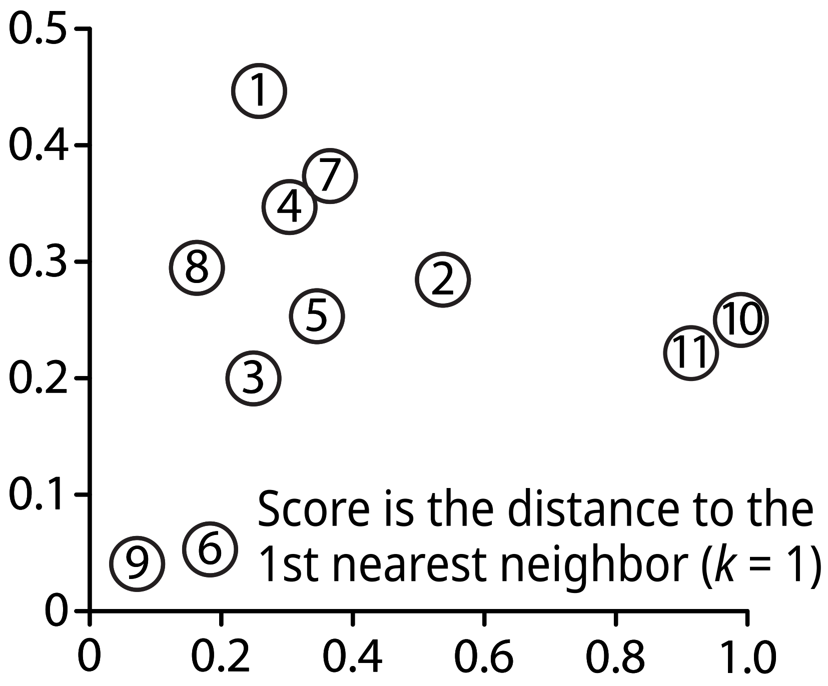
id	score
10	0.454
1	0.109
5	0.103

*k*th-NN（1/2） [Bay & Schwabacher, 2003]



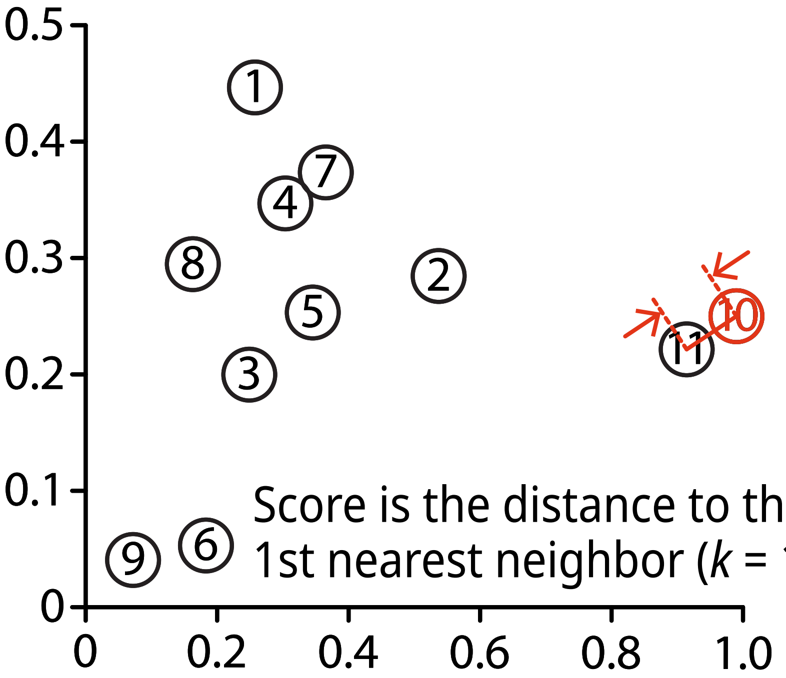
id	score
10	0.454
2	0.193
8	0.128
6	0.112
9	0.112
3	0.110
1	0.109
5	0.103
4	0.067
7	0.067

*k*th-NN（2/2） [Bay & Schwabacher, 2003]



id	score
10	0.454
2	0.193
8	0.128
6	0.112
9	0.112
3	0.110
1	0.109
5	0.103
4	0.067
7	0.067

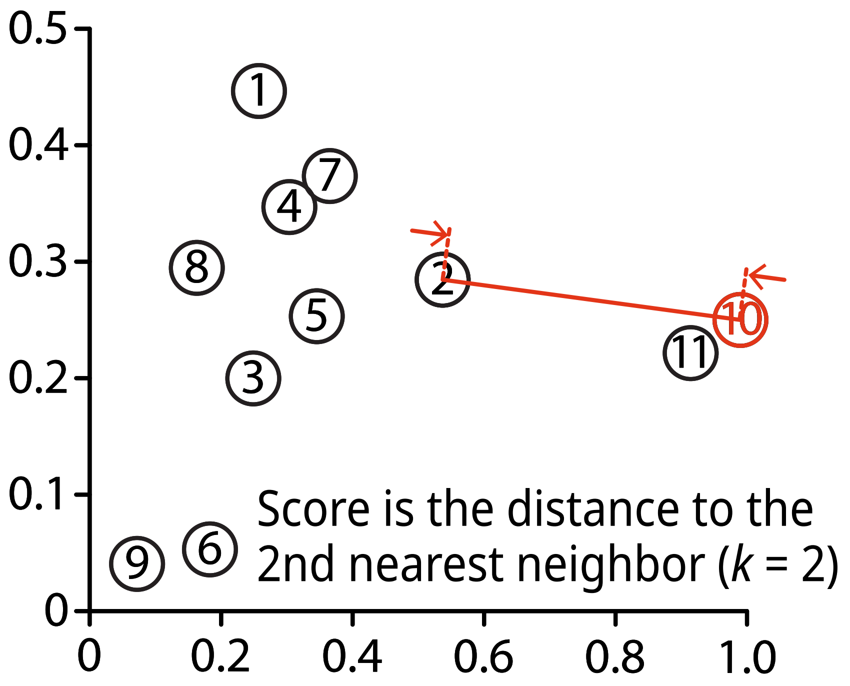
*k*th-NN（2/2） [Bay & Schwabacher, 2003]



Score is the distance to the 1st nearest neighbor (k = 1)

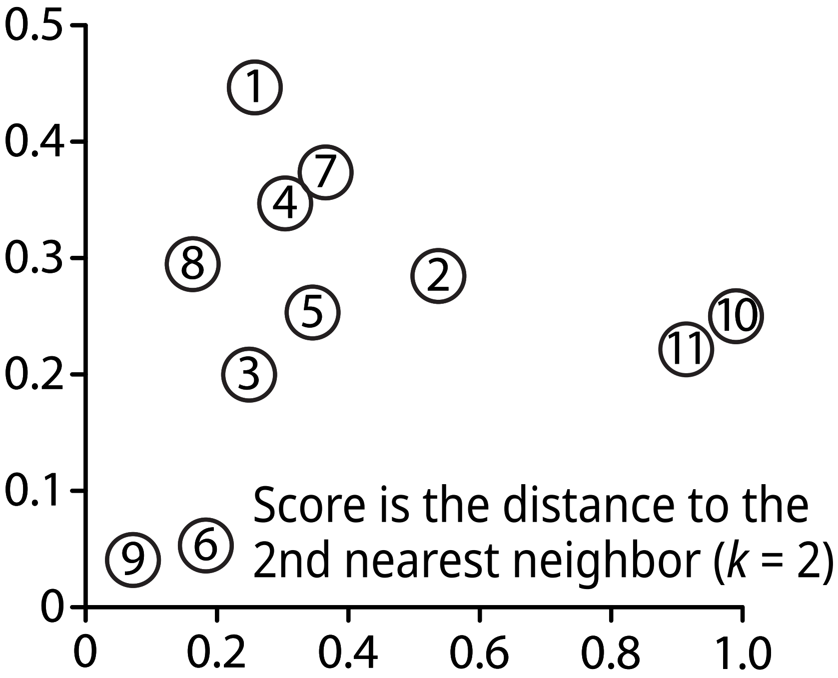
id	score
2	0.193
8	0.128
6	0.112
9	0.112
3	0.110
1	0.109
5	0.103
4	0.067
7	0.067
10	0.028
11	0.028

*k*th-NN（2/2）[Bay & Schwabacher, 2003]



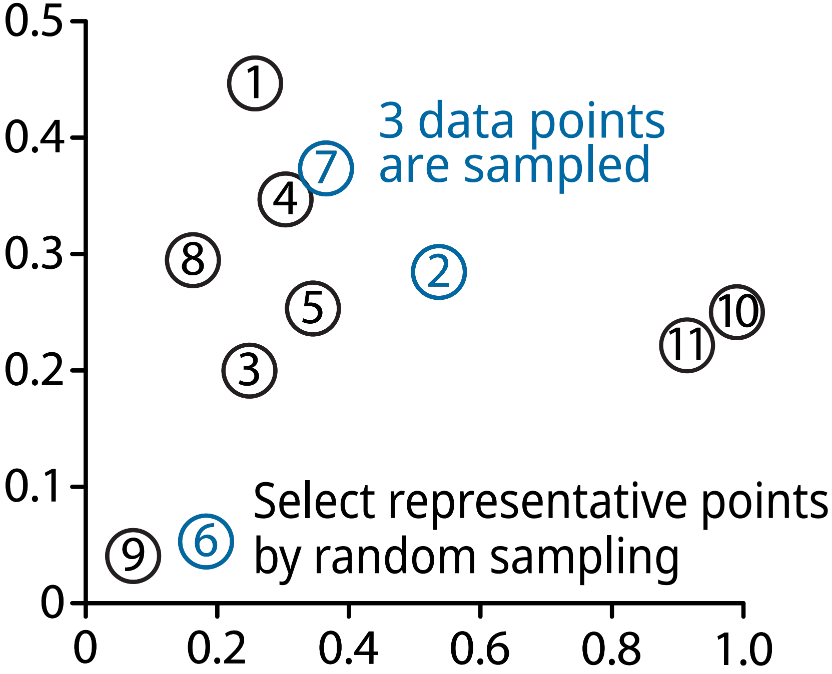
id	score
2	0.193
8	0.128
6	0.112
9	0.112
3	0.110
1	0.109
5	0.103
4	0.067
7	0.067
10	0.028
11	0.028

*k*th-NN（2/2） [Bay & Schwabacher, 2003]



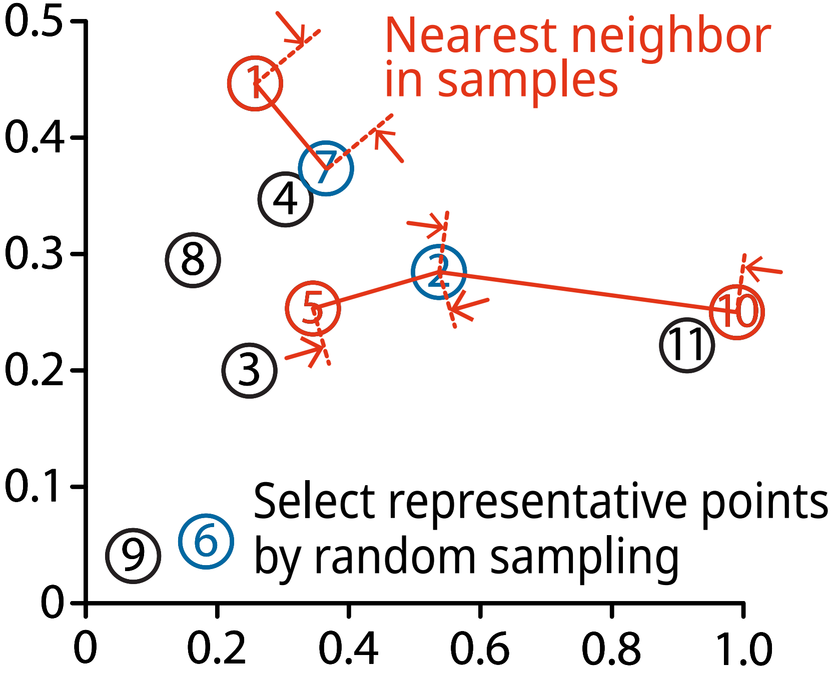
id	score
10	0.454
11	0.436
9	0.238
2	0.194
6	0.161
8	0.150
1	0.130
3	0.128
7	0.122
5	0.110
4	0.103

Sampling [Sugiyama & Borgwardt, 2013]



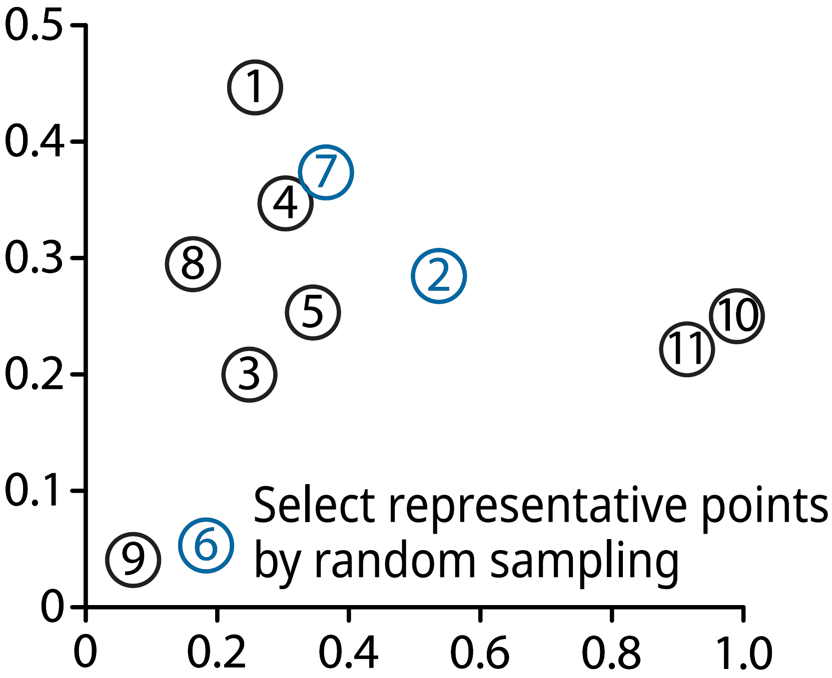
id	score

Sampling [Sugiyama & Borgwardt, 2013]



id	score
10	0.454
1	0.130
5	0.122

Sampling [Sugiyama & Borgwardt, 2013]



id	score
10	0.454
11	0.436
6	0.369
8	0.217
2	0.193
7	0.193
3	0.161
1	0.130
5	0.122
9	0.112
4	0.067

Algorithm 1: kth-NN

Initialize $M \in R^{n \times n}$, $\boldsymbol{q} \in R^n$
foreach $\boldsymbol{x}_i \in X$ **do**
    foreach $\boldsymbol{x}_j \in X$ **do**
        $m_{ij} \leftarrow d(\boldsymbol{x}_i, \boldsymbol{x}_j)$
foreach $i \in \{1, 2, \dots, n\}$ **do**
    $q_i \leftarrow k$th largest value in ith row of M
Output $\boldsymbol{q}$

Algorithm 2: Sugiyama-Borgwardt Sampling Method

$S \leftarrow$ Subsample of X, initialize $M \in R^{n \times |S|}$, $\boldsymbol{q} \in R^n$

foreach $\boldsymbol{x}_i \in X$ **do**

  foreach $\boldsymbol{s}_j \in S$ **do**

    $m_{ij} \leftarrow d(\boldsymbol{x}_i, \boldsymbol{s}_j)$

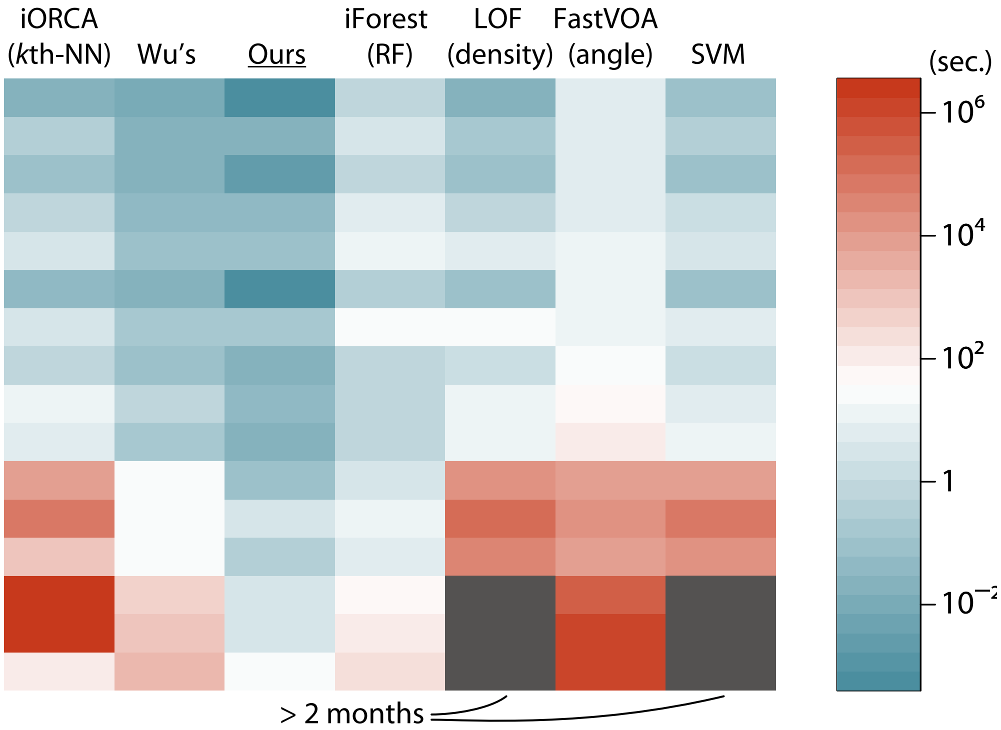
foreach $i \in \{1, 2, \ldots, n\}$ **do**

  $q_i \leftarrow$ Largest value in ith row of M

Output $\boldsymbol{q}$

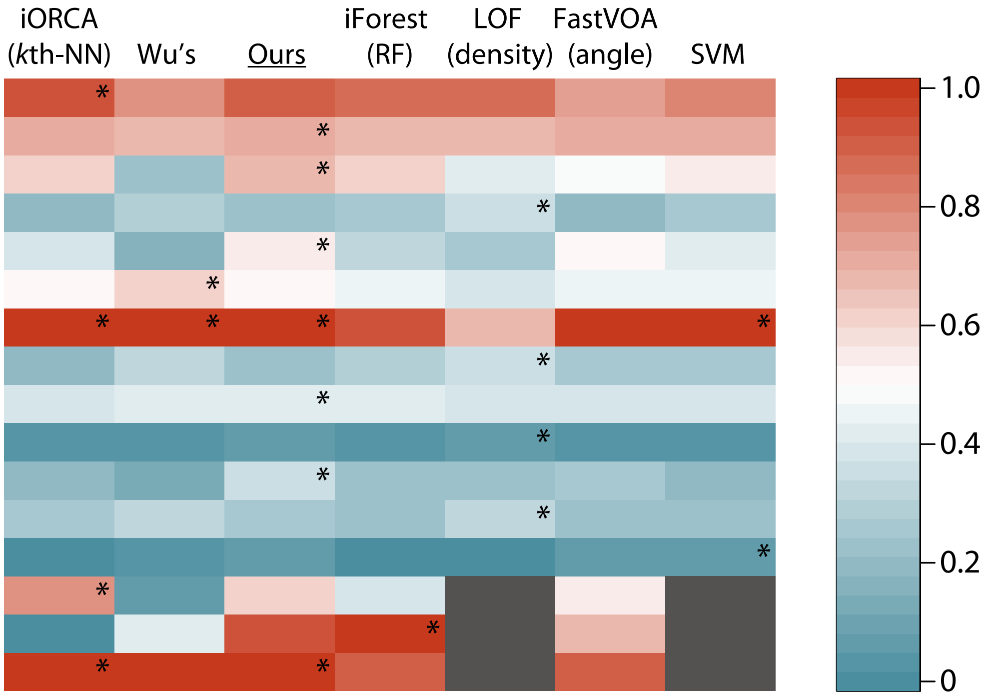
Results (Runtime)

	# of objects	# of outliers	# of dims
Ionosphere	351	126	34
Arrhythmia	452	207	274
Wdbc	569	212	30
Mfeat	600	200	649
Isolet	960	240	617
Pima	768	268	8
Gaussian*	1000	30	1000
Optdigits	1688	554	64
Spambase	4601	1813	57
Statlog	6435	626	36
Skin	245057	50859	3
Pamap2	373161	125953	51
Covtype	286048	2747	10
Kdd1999	4898431	703067	6
Record	5734488	20887	7
Gaussian*	10000000	30	20



Results (Accuracy)

	# of objects	# of outliers	# of dims
Ionosphere	351	126	34
Arrhythmia	452	207	274
Wdbc	569	212	30
Mfeat	600	200	649
Isolet	960	240	617
Pima	768	268	8
Gaussian*	1000	30	1000
Optdigits	1688	554	64
Spambase	4601	1813	57
Statlog	6435	626	36
Skin	245057	50859	3
Pamap2	373161	125953	51
Covtype	286048	2747	10
Kdd1999	4898431	703067	6
Record	5734488	20887	7
Gaussian*	10000000	30	20



*k*NN approach for Classification

- The kNN (k Nearest Neighbor) classifier predicts the label of $\boldsymbol{x}$ to the majority class among its k nearest neighbors
- Sort a given dataset D as $(\boldsymbol{x}_{(1)}, y_{(1)}), (\boldsymbol{x}_{(2)}, y_{(2)}), \dots, (\boldsymbol{x}_{(N)}, y_{(N)})$ in increasing order according to the distance from a test point $\boldsymbol{x}$
  - Euclidean distance $\|\boldsymbol{x}_i - \boldsymbol{x}\|_2 = \sqrt{\sum_{j=1}^{n} (x_i^j - x^j)^2}$ is typically used
- Take the top-k points $(\boldsymbol{x}_{(1)}, y_{(1)}), (\boldsymbol{x}_{(2)}, y_{(2)}), \dots, (\boldsymbol{x}_{(k)}, y_{(k)})$ and $\hat{y} = \operatorname{argmax}_{c \in C} |\{(\boldsymbol{x}_{(i)}, y_{(i)}) \mid i \leq k \text{ and } y_{(i)} = c\}|$
  - $|\{(\boldsymbol{x}_{(i)}, y_{(i)}) \mid i \leq k \text{ and } y_{(i)} = c\}|/k$ can be viewed as posterior $P(c \mid \boldsymbol{x})$

Summary

- Machine Learning: Science of (computational) "learning"
  - Purpose: Find rules that generalize experience (data)
- Steps of ML application:
  - Mathematically model a real world phenomenon
  - Formulate behavior of "learning"
  - Design and implement algorithms
  - Evaluate results according to the application at hand

Source of General ML

- Many books
  - K.Murphy, *Machine Learning: A Probabilistic Perspective*, The MIT Press
  - M.J.Zaki, W.Meira Jr., *Data Mining and Analysis*, Cambridge University Press
- Lecture videos like Coursera
- Competition like Kaggle

Source of ML Research

- Papers presented at conferences
  - ML and DM (data mining)
    - ICML (International Conference on Machine Learning)
    - NeurIPS (Neural Information Processing Systems)
    - ICLR (International Conference on Learning Representations)
    - KDD (Knowledge Discovery and Data Mining)
  - AI (artificial intelligence)
    - IJCAI (Inter. Joint Conference on Artificial Intelligence)
    - AAAI (Conference on Artificial Intelligence)