SE125 Machine Learning

Unsupervised Learning

Yue Ding

School of Software, Shanghai Jiao Tong University

dingyue@sjtu.edu.cn

Unsupervised Learning

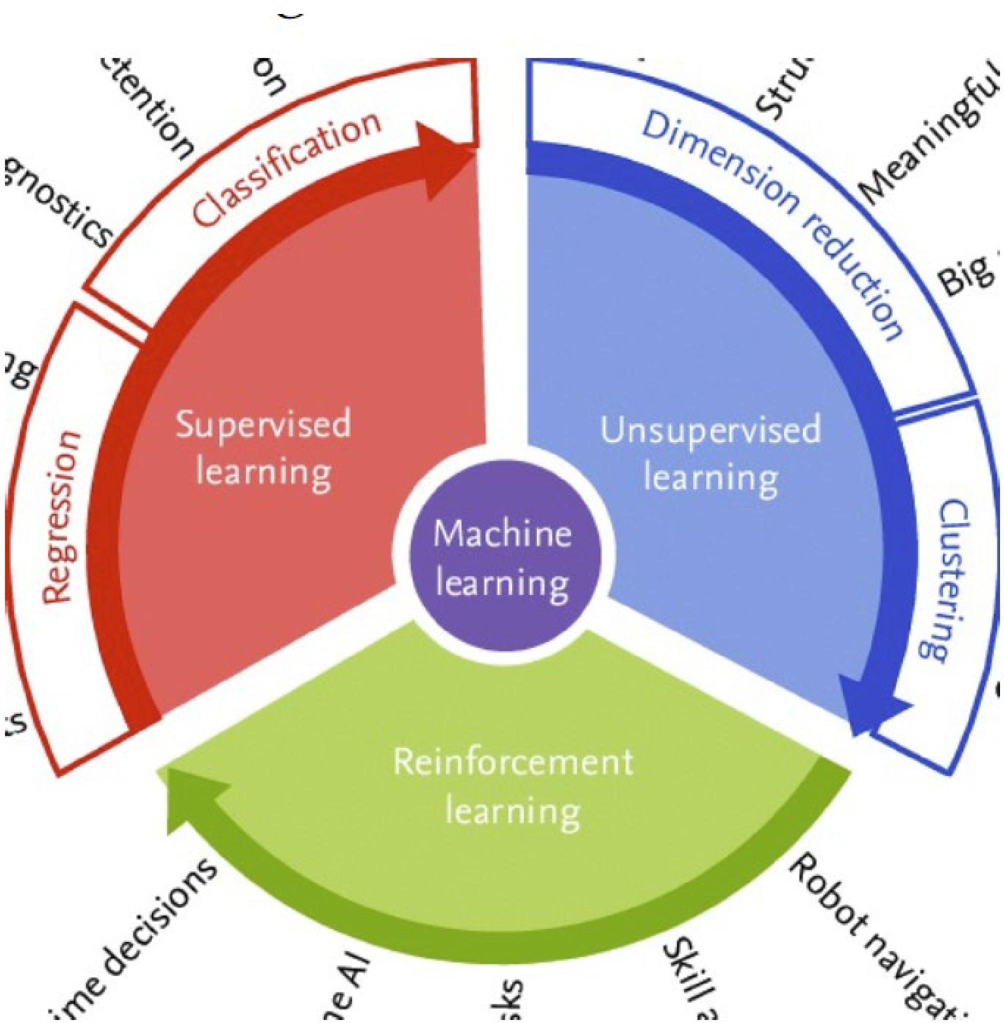
- 课程难度:

- 掌握程度:

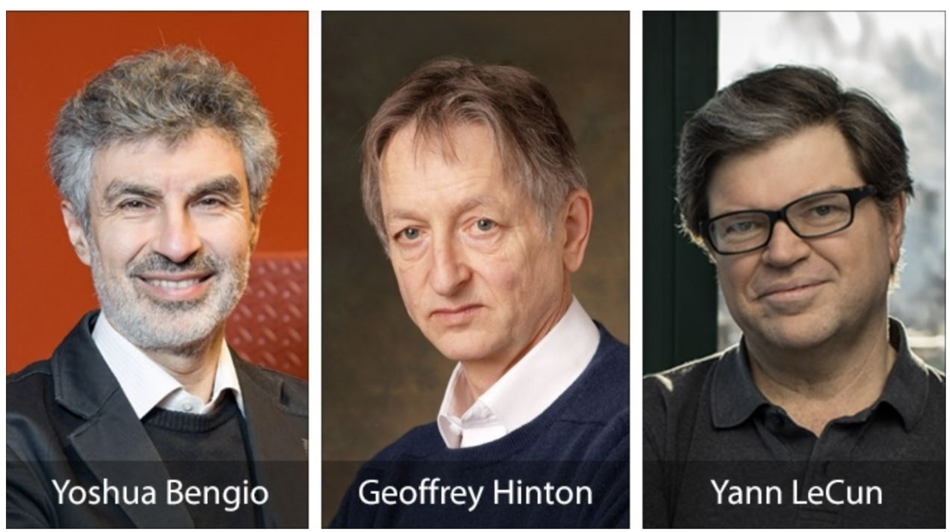
References and Acknowledgement

- CS538, unsupervised learning, University of Illinois Chicago
- http://wnzhang.net/teaching/cs420/slides/9-unsupervised-learning.pdf
- http://www.mit.edu/~9.54/fall14/slides/Class13.pdf
- Auto-Encoder, Prof. Hung-yi Lee, machine learning 2021 spring
  - https://speech.ee.ntu.edu.tw/~hylee/ml/2021-spring.html

Machine Learning Problems



"We expect unsupervised learning to become far more important in the longer term. Human and animal learning is largely unsupervised: we discover the structure of the world by observing it, not by being told the name of every object."–LeCun, Bengio, Hinton, Nature 2015

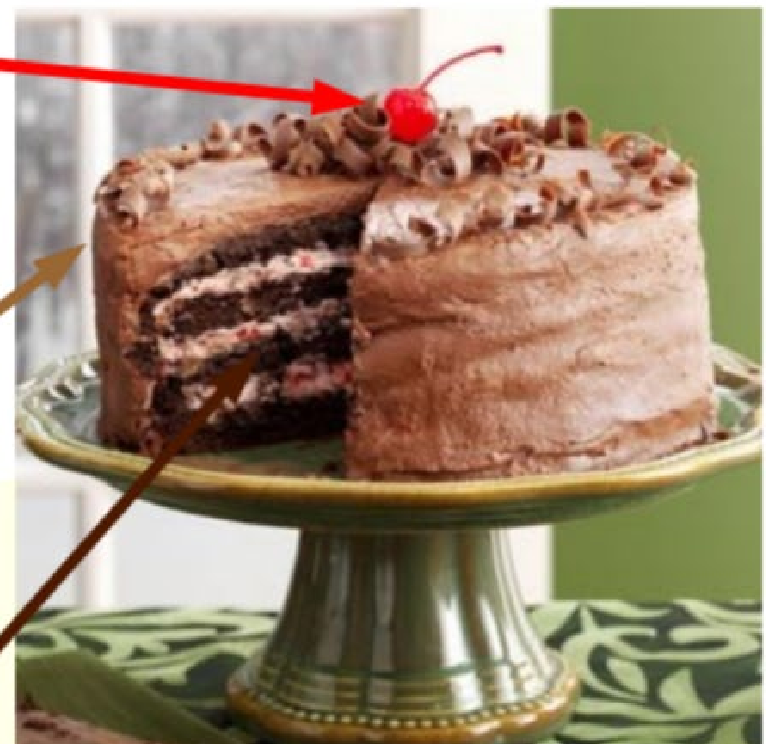


Unsupervised Learning

- Facebook AI Chief ***Yann LeCun*** introduced his now-famous “cake analogy” at NIPS2016:
  - “If intelligence is a cake, the bulk of the cake is unsupervised learning, the icing on the cake is supervised learning, and the cherry on the cake is reinforcement learning (RL).”

Unsupervised Learning

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**



- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Unsupervised/Predictive Learning (cake)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**
- (Yes, I know, this picture is slightly offensive to RL folks. But I'll make it up)

Unsupervised Learning

- LeCun updated his cake recipe at the 2019 International Solid-State Circuits Conference (ISSCC) in San Francisco, replacing “unsupervised learning” with “**self-supervised learning**,” a variant of unsupervised learning where the data provides the supervision.

Supervised Learning vs. Unsupervised Learning

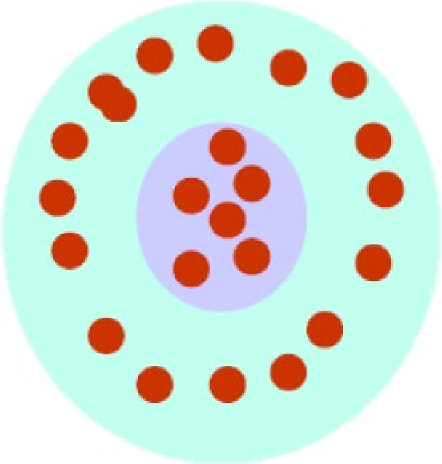
- **Supervised learning:** discover patterns in the data that relate data attributes **with a target (class)** attribute.
  - These patterns are then utilized to predict the values of the target attribute in future data instances.
- **Unsupervised learning:** The data have **no target attribute**.
  - We want to explore the data to find some intrinsic structures in them.

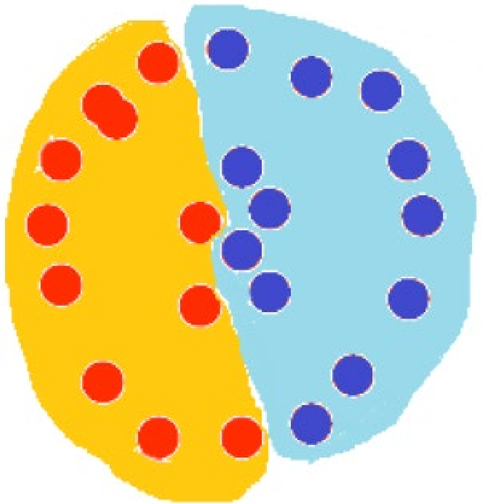
Unsupervised Learning

- Fundamentals of Unsupervised Learning
  - **K-means clustering**
  - Principal component analysis
- Probabilistic Unsupervised Learning
  - Mixture Gaussians
  - EM Methods
- Deep Unsupervised Learning
  - **Auto-encoder**

What is clustering?

- The organization of unlabeled data into similarity groups called clusters.
- A cluster is a collection of data items which are "similar" between them, and "dissimilar" to data items in other clusters.



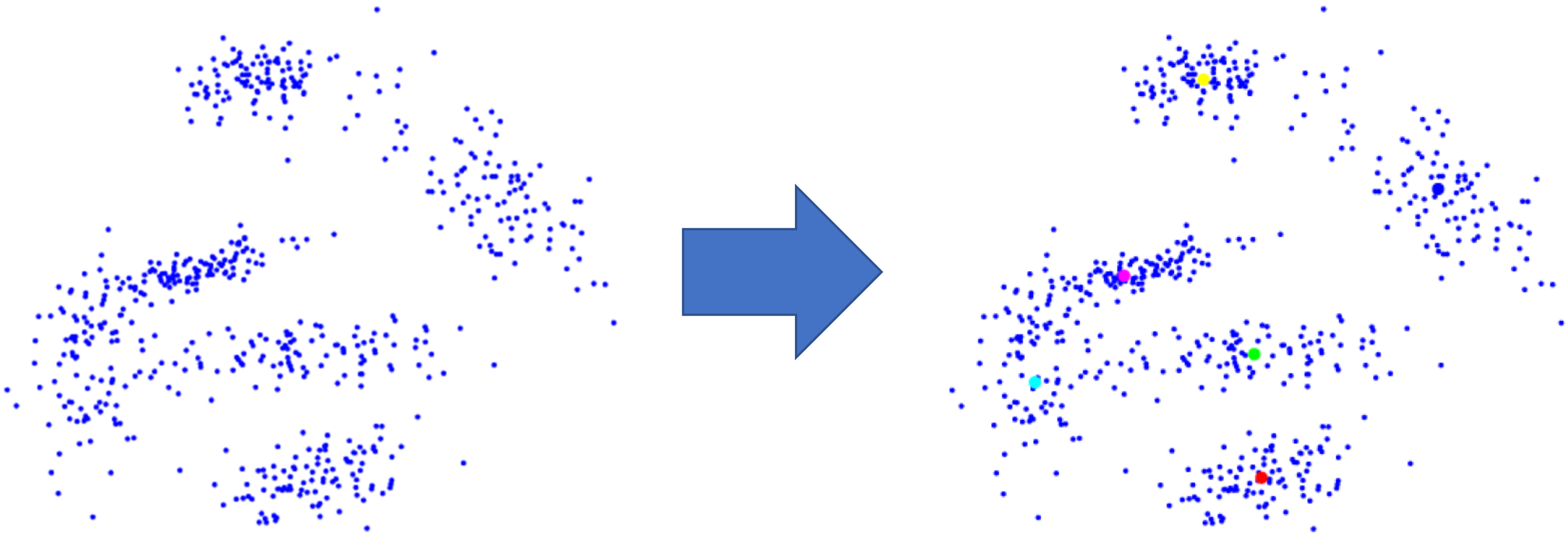


K-means Clustering

- K-means (MacQueen, 1967) is a partitional clustering algorithm.
- Let the set of data points *D* be $\{x_1, x_2, x_3, \ldots, x_n\}$,
  where $x_{i \in n}$ is a ***d*** dimensional vector.
- The *k*-means algorithm partitions the given data into ***k*** clusters:
  - Each cluster has a cluster **center**, called **centroid**.
  - k is specified by the user.

K-means Clustering

- K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.



K-means Algorithm

- Given k, the *k-means* algorithm works as follows:
  - 1.Choose k (random) data points (seeds) to be the initial centroids, cluster centers.
  - 2.Assign each data point to the closest centroid.

    Euclidean Distance:

$$L_2(x, \mu^k) = \|x - \mu^k\| = \sqrt{\sum_{m=1}^{d} (x_i - \mu_m^k)^2}$$

  - 3.Re-compute the centroids using the current cluster memberships.

$$\mu^k = \frac{1}{C_k} \sum_{x \in C_k} x$$

  - 4.If a convergence criterion is not met, repeat steps 2 and 3.

K-means convergence criterion

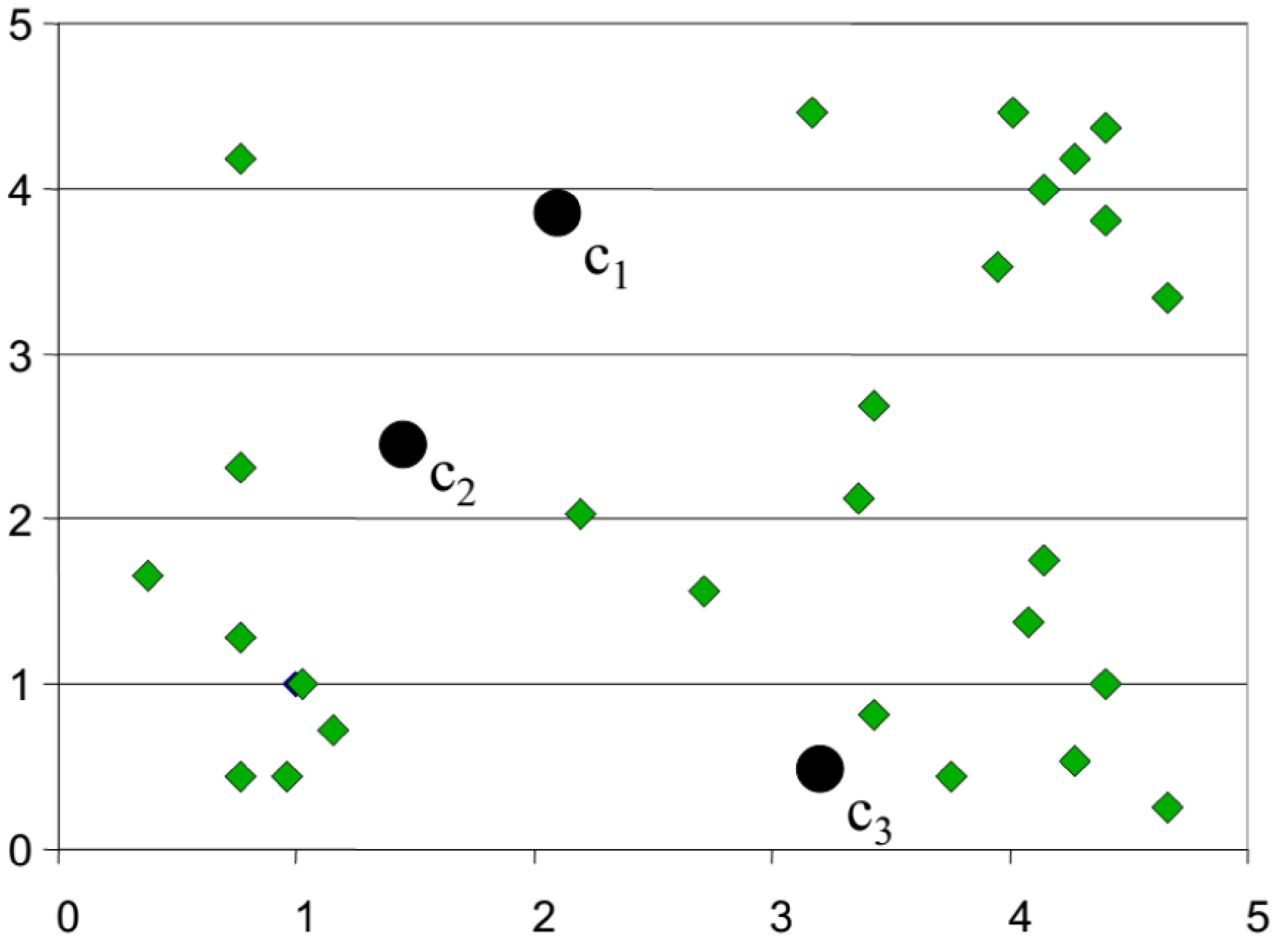
- No (or minimum) re-assignments of data points to different clusters, *or*
- No (or minimum) change of centroids, or
- Minimum decrease in the **sum of squared error** (SSE):

$$\min_{\{\mu^k\}_{k=1}^K} \sum_{k=1}^{K} \sum_{x \in C_k} L(x - \mu^k) \qquad \mu^k = \frac{1}{C_k} \sum_{x \in C_k} x$$

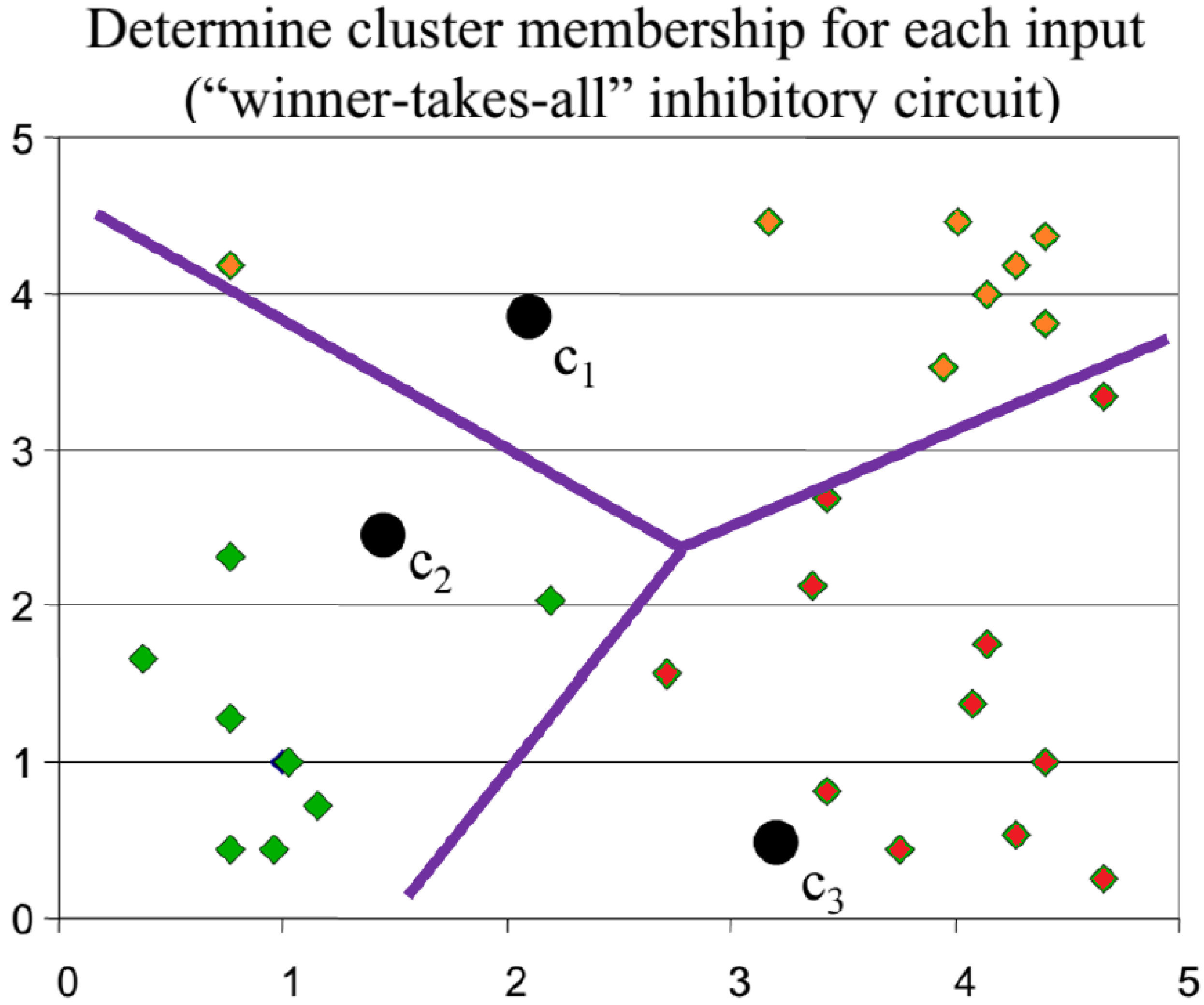
- Finding the global optimum is NP-hard.
- The *k*-means algorithm is guaranteed to converge to a local optimum.

K-means clustering example: step 1

Randomly initialize the cluster centers (synaptic weights)

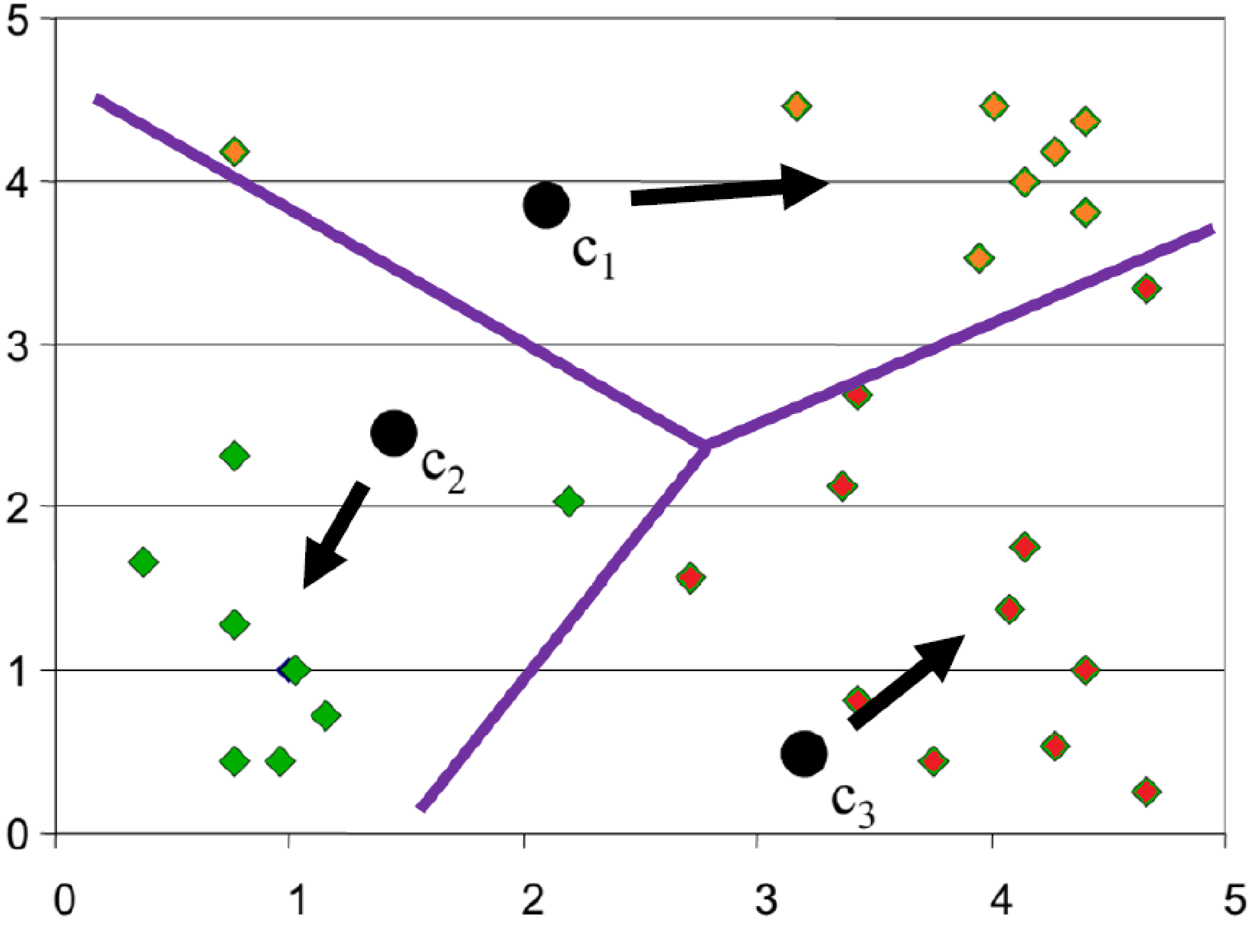


K-means clustering example: step 2

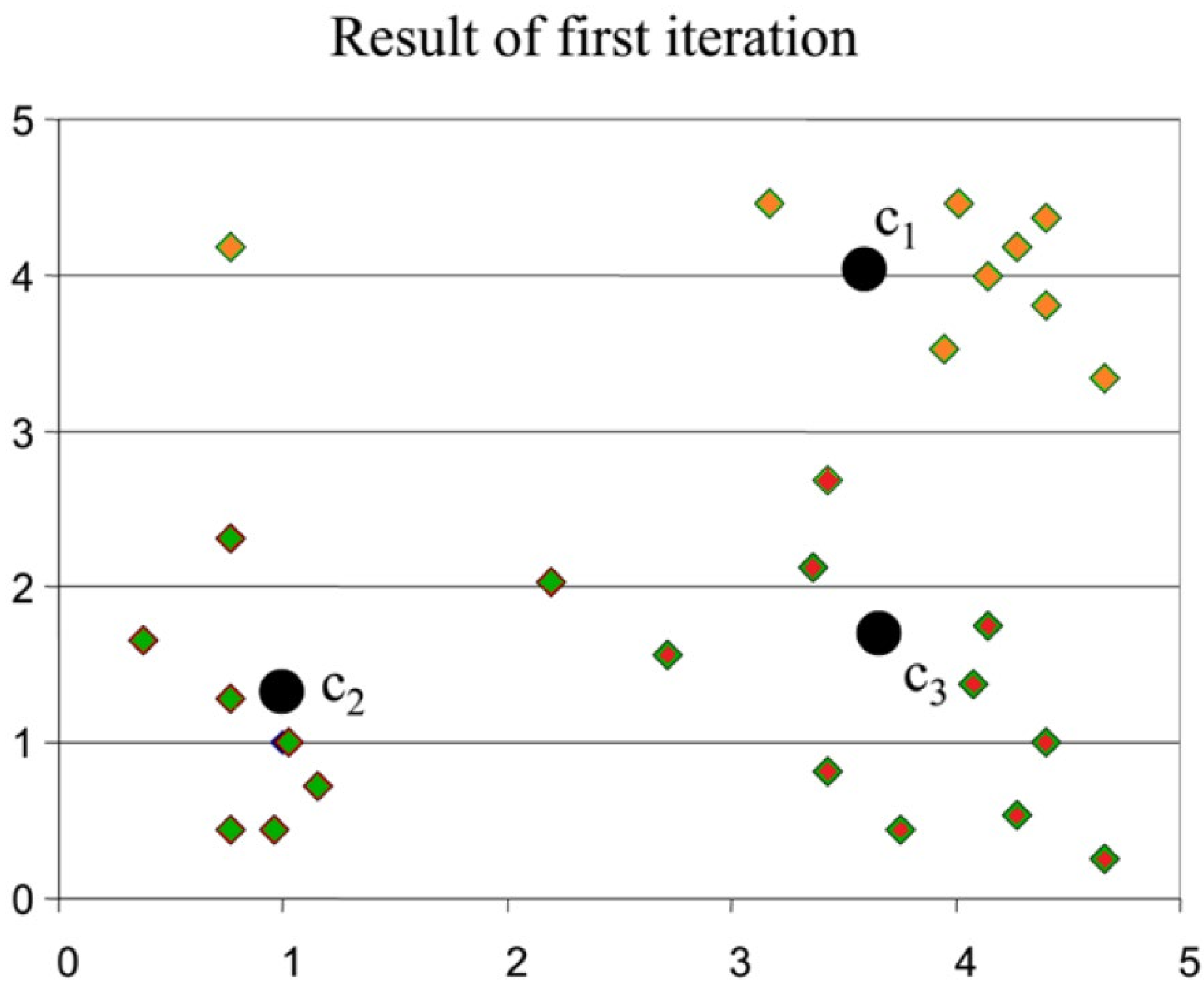


K-means clustering example: step 3

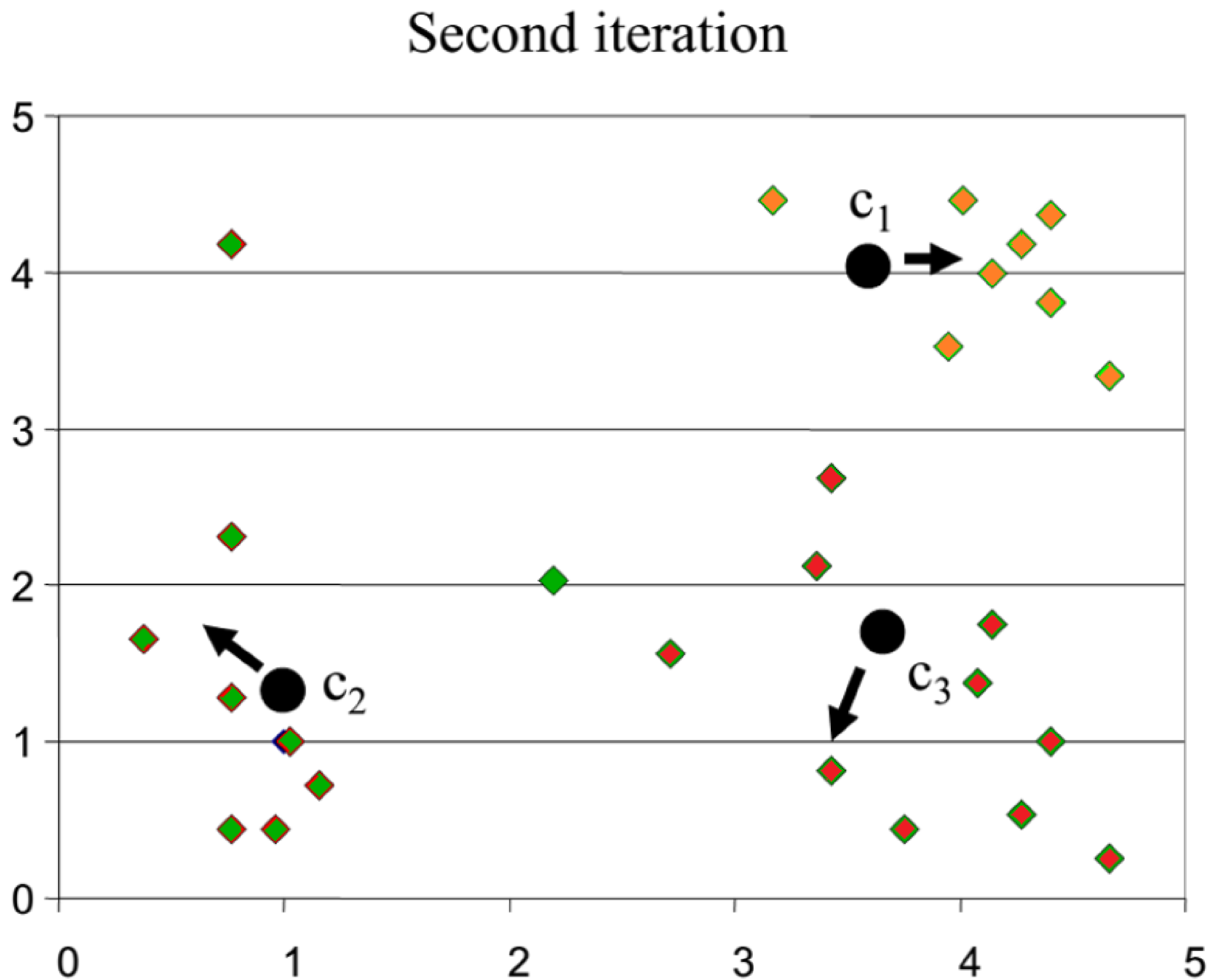
Re-estimate cluster centers (adapt synaptic weights)



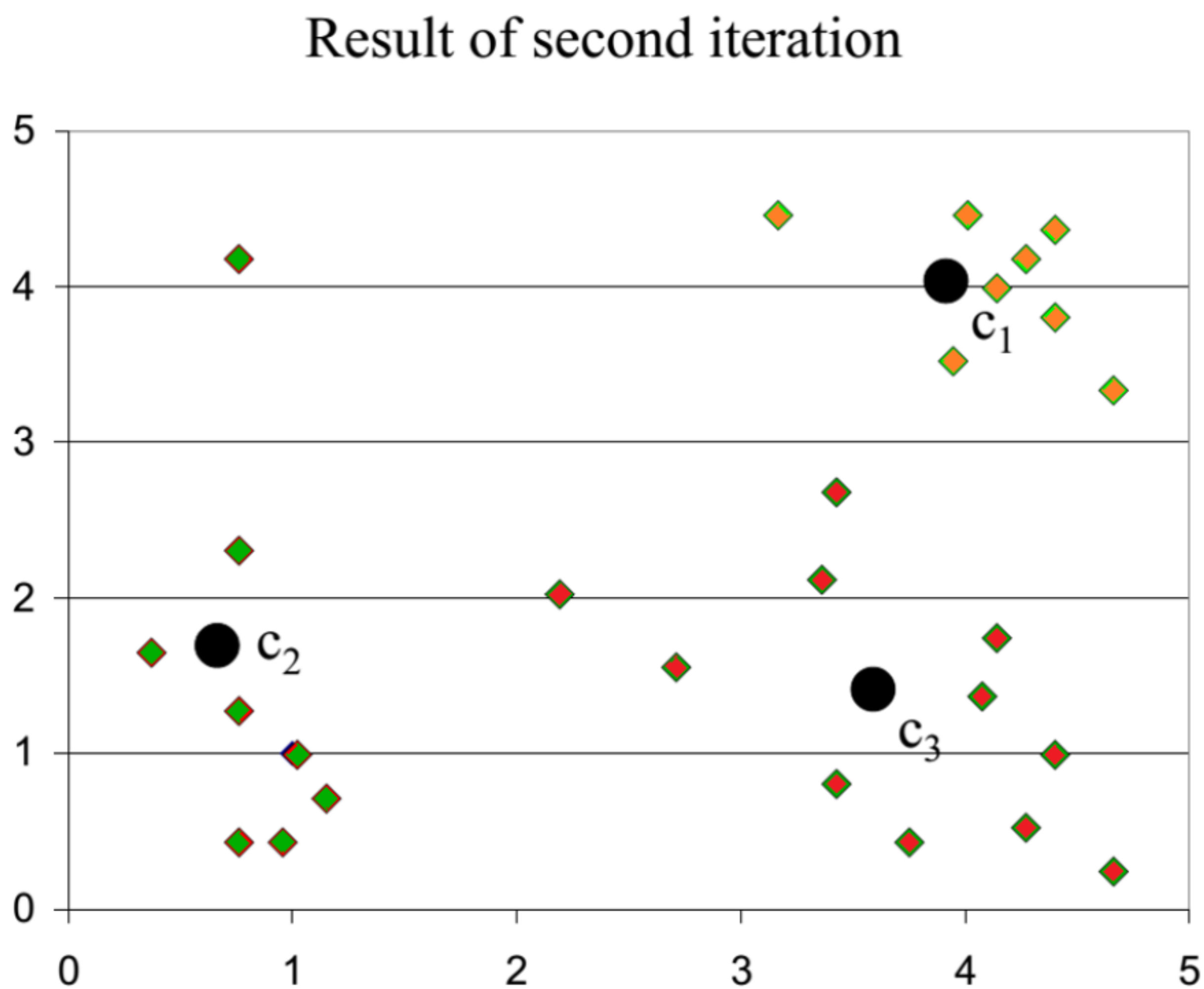
K-means clustering example



K-means clustering example



K-means clustering example



Time Complexity

- Assume computing distance between two instances is $O(d)$ where d is the dimensionality of the vectors.
- What is the time complexity of k-means clustering?

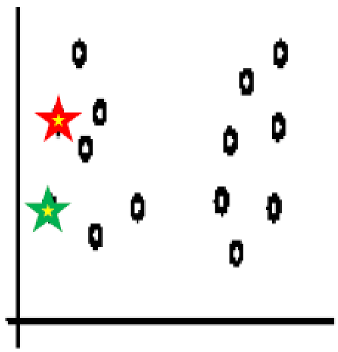
Time Complexity

- Assume computing distance between two instances is $O(d)$ where d is the dimensionality of the vectors.
- Reassigning clusters: $O(knd)$ distance computations.
- Computing centroids: Each instance vector gets added once to some centroid: $O(nd)$.
- Assume these two steps are each done once for I iterations: $O(Iknd)$

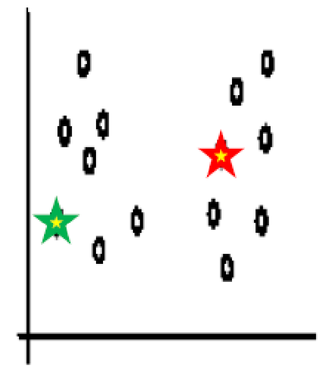
Seed Choice

- Results can vary based on random seed selection.
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings.
- Select good seeds using a heuristic or the results of another method.

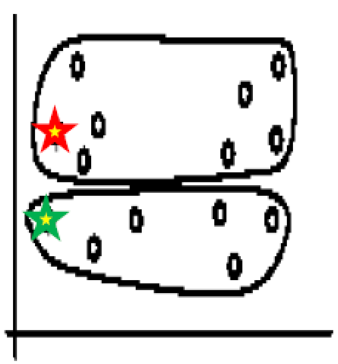
Sensitivity to Initial Seeds



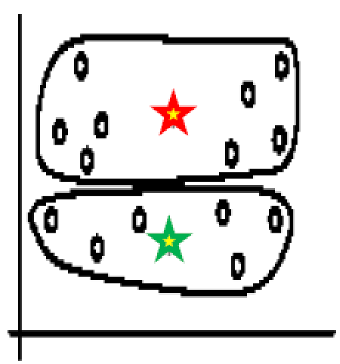
Random selection of seeds (centroids)



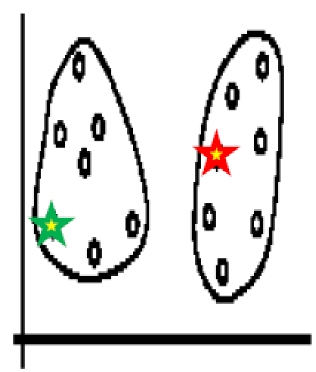
Random selection of seeds (centroids)



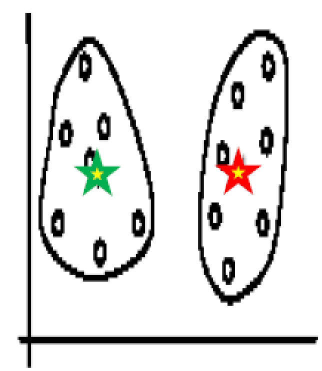
Iteration 1



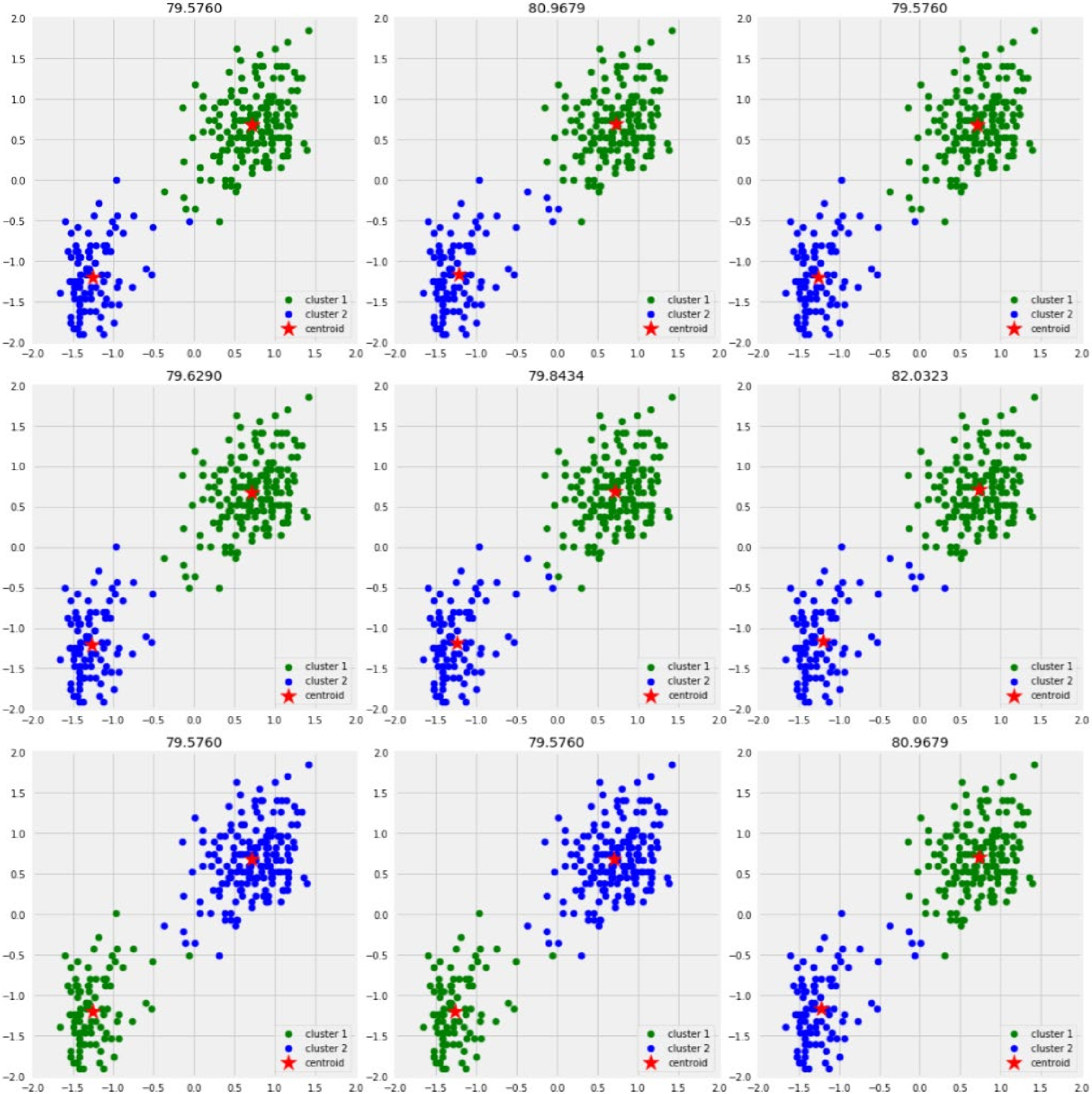
Iteration 2



Iteration 1



Iteration 2


79.5760
80.9679
79.5760
79.6290
79.8434
82.0323
79.5760
79.5760
80.9679
cluster 1
cluster 2
centroid

Silhouette Analysis

- **Silhouette analysis** can be used to determine the degree of separation between clusters. For each sample:
  - Compute the average distance from all data points in the same cluster (a^i).
  - Compute the average distance from all data points in the closest cluster (b^i).
  - Compute the coefficient:

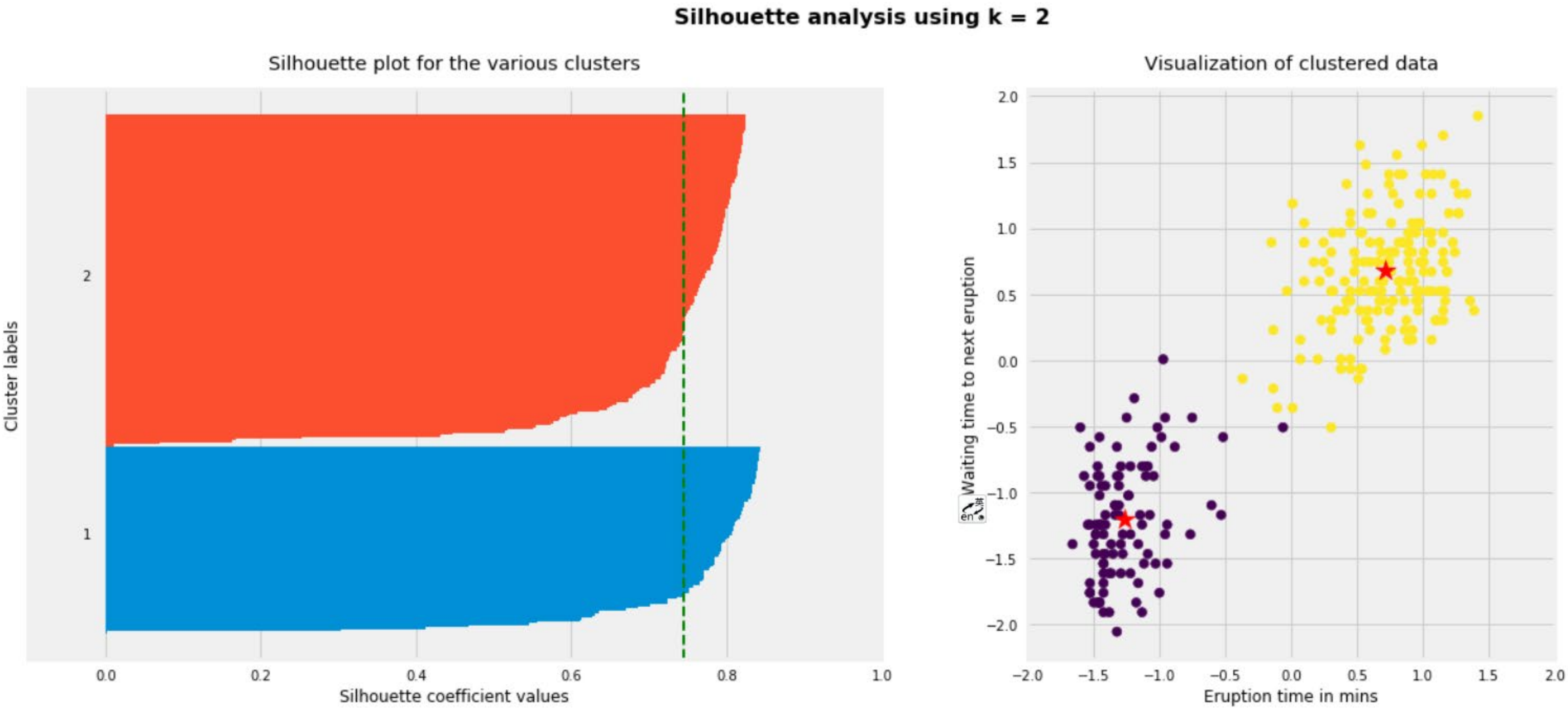
$$\frac{b^i - a^i}{\max(a^i, b^i)}$$

Silhouette Analysis

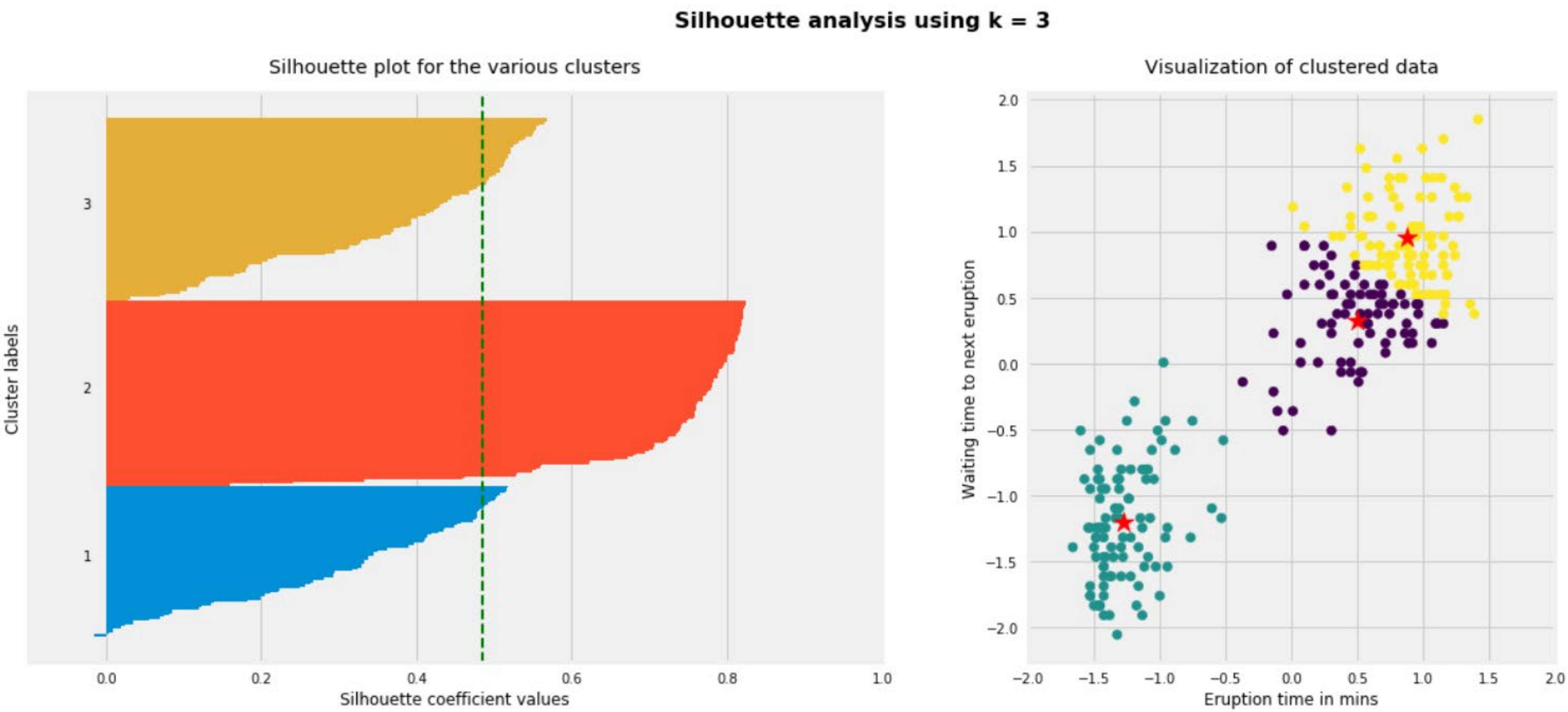
- If it is 0 : the sample is very close to the neighboring clusters.
- If it is 1 : the sample is far away from the neighboring clusters.
- If it is -1 : the sample is assigned to the wrong clusters.

- Therefore, we want the coefficients to be **as big as possible and close to 1 to have good clusters**.

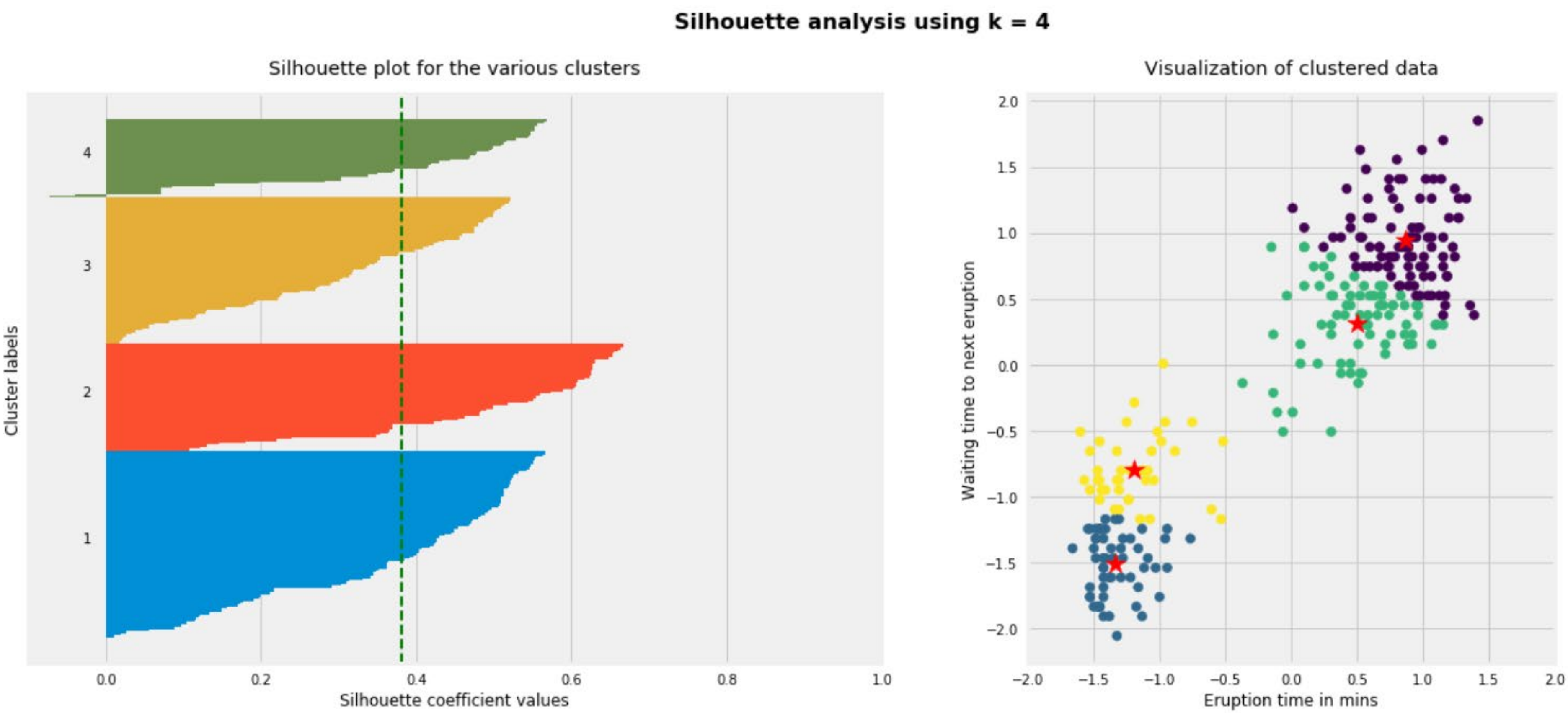
Silhouette Analysis



Silhouette Analysis



Silhouette Analysis



Silhouette Analysis

- Note that：
  - No clear evidence that any other clustering algorithm performs better in general.
  - Comparing different clustering algorithms is a difficult task. No one knows the correct clusters!

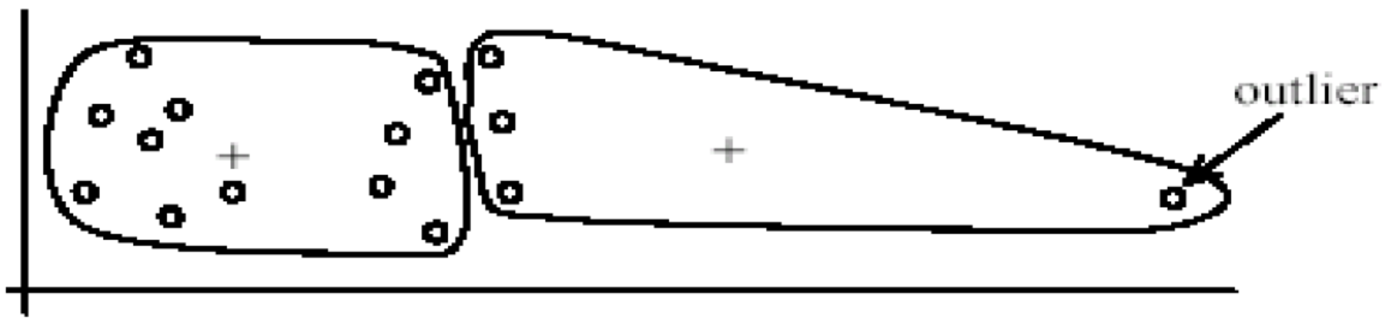
Why use K-means?

- Strengths:
  - Simple: easy to understand and to implement.
  - Efficient: *k*-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.

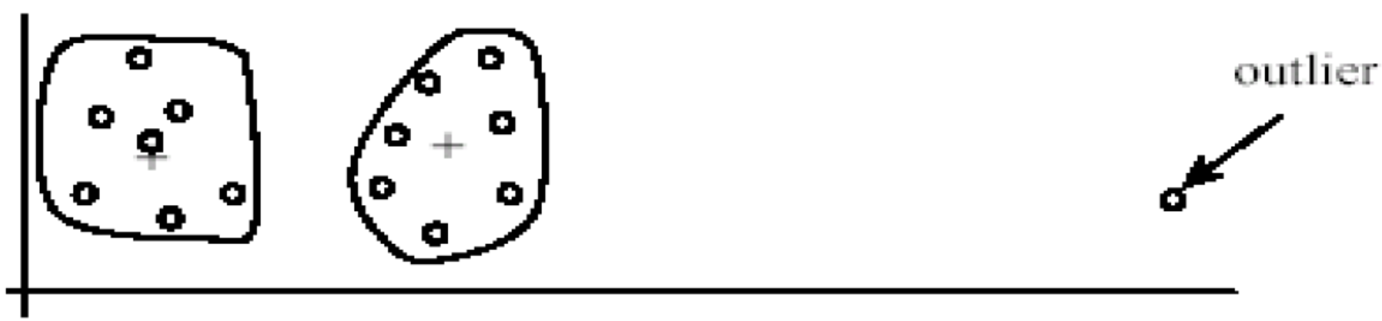
Weaknesses of K-means

- The algorithm is only applicable if the mean is defined.
- The user needs to specify *k*.
- The algorithm is sensitive to **outliers**
  - Outliers are data points that are very far away from other data points.
  - Outliers could be errors in the data recording or some special data points with very different values.

Outliers



(A): Undesirable clusters



(B): Ideal clusters

Dealing with Outliers

- Remove some data points that are much further away from the centroids than other data points.
  - To be safe, we may want to monitor these possible outliers over a few iterations and then decide to remove them.
- Perform random sampling: by choosing a small subset of the data points, the chance of selecting an outlier is much smaller.
  - Assign the rest of the data points to the clusters by distance or similarity comparison, or classification

K-means Application: Image Compression

K-means Application: Image Segmentation

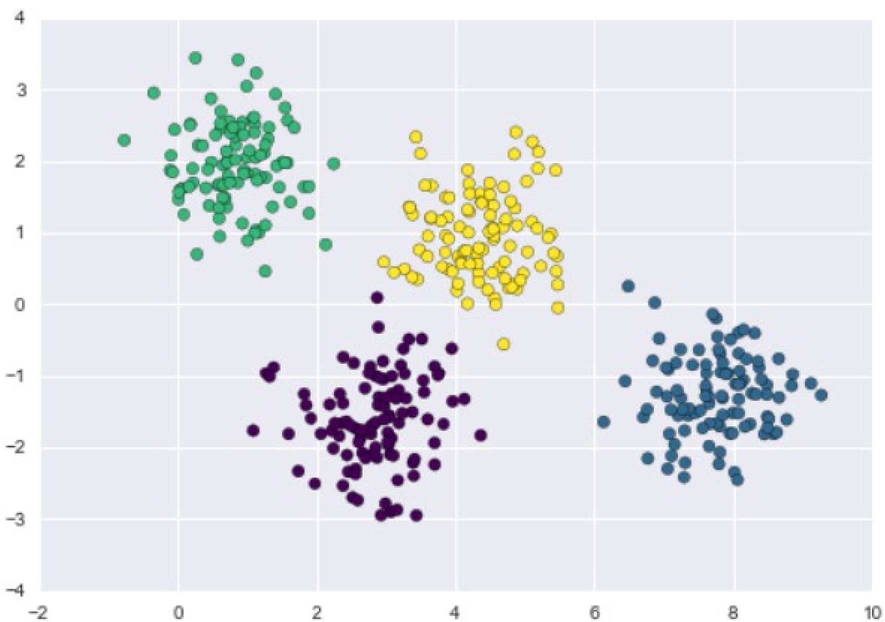
- Image segmentation is the classification of an image into *different groups*. Many kinds of research have been done in the area of image segmentation using clustering.

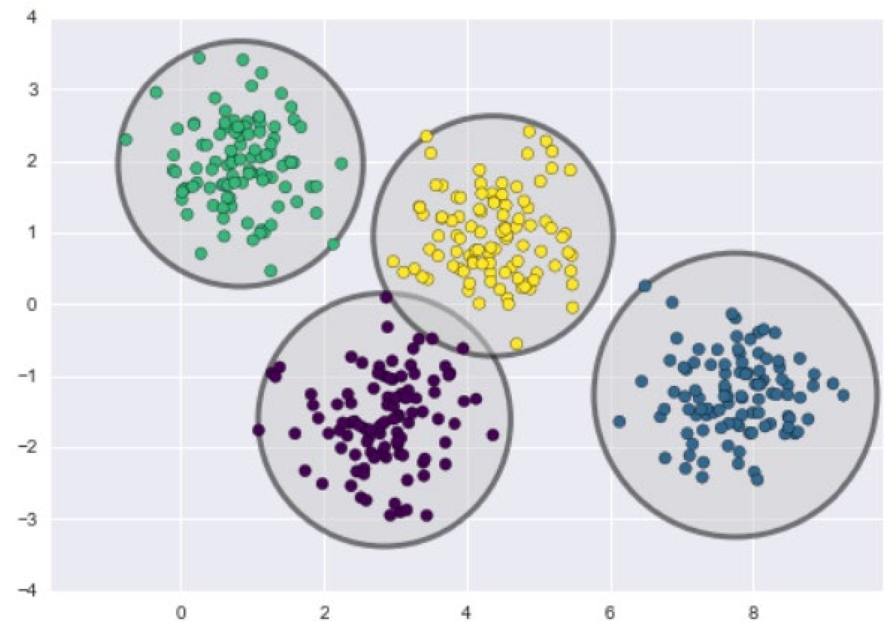
K-means Application: Image Segmentation

In Depth: Gaussian Mixture Models

（扩展内容）

- K-means algorithm is good in capturing structure of the data if clusters have a spherical-like shape. It always try to construct a nice spherical shape around the centroid.



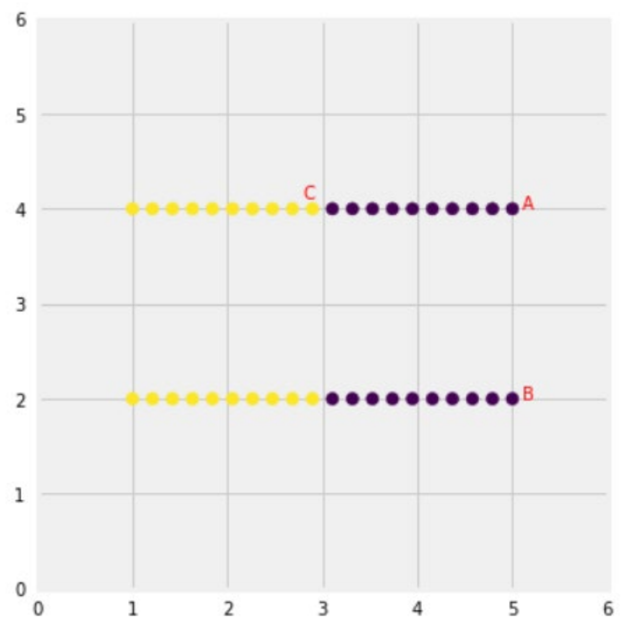


In Depth: Gaussian Mixture Models

(扩展内容)

Three cases where *k*-means will not perform well.

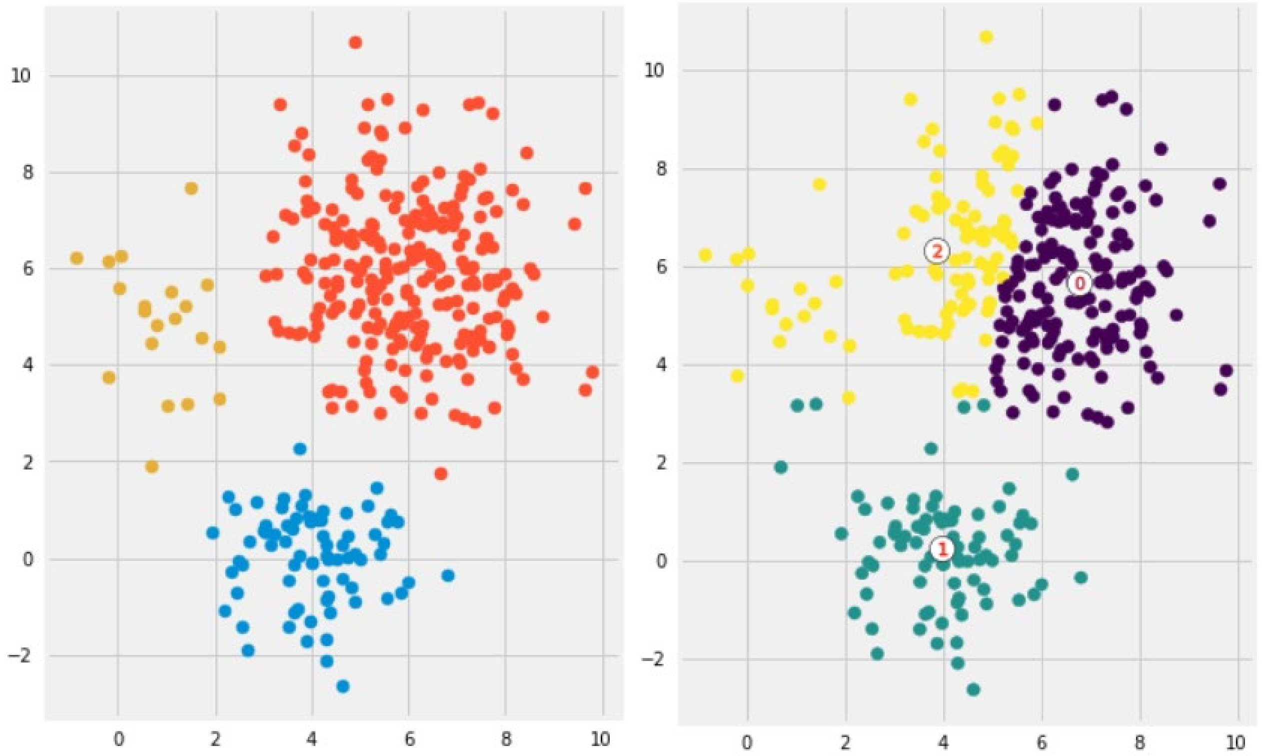
- Case 1: K-means algorithm doesn’t let data points that are far-away from each other share the same cluster even though they obviously belong to the same cluster.



In Depth: Gaussian Mixture Models

(扩展内容)

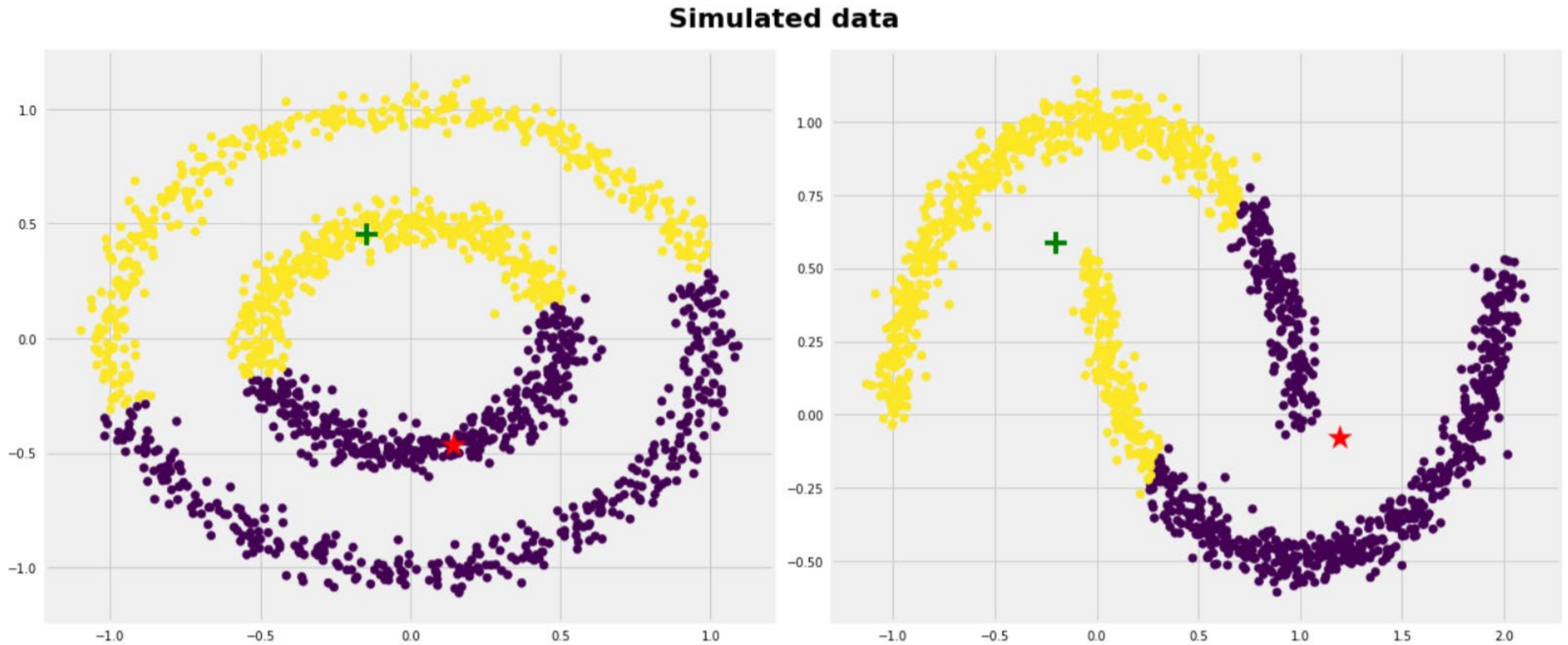
- Case 2: Data points in smaller clusters may be left away from the centroid in order to focus more on the larger cluster.



In Depth: Gaussian Mixture Models

(扩展内容)

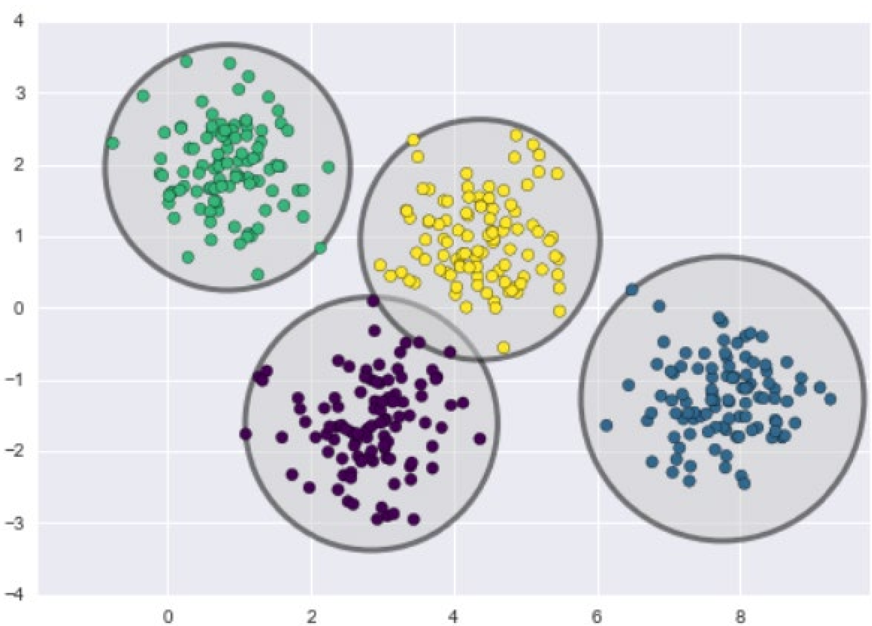
- Case 3: Data samples that have complicated geometric shape.

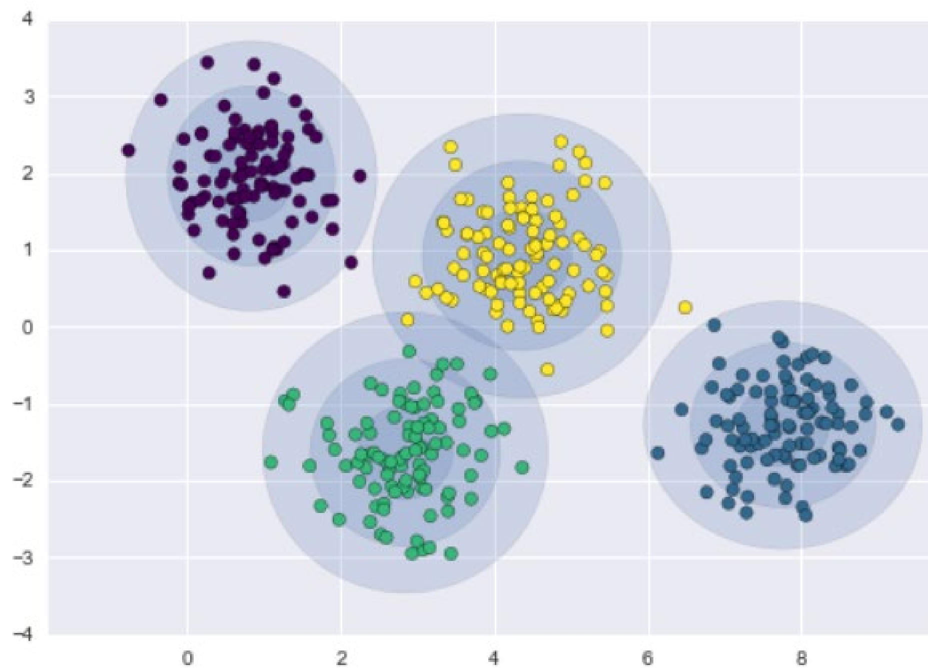


In Depth: Gaussian Mixture Models

(扩展内容)

- A Gaussian mixture model (GMM) attempts to find a mixture of multi-dimensional Gaussian probability distributions that best model any input dataset.

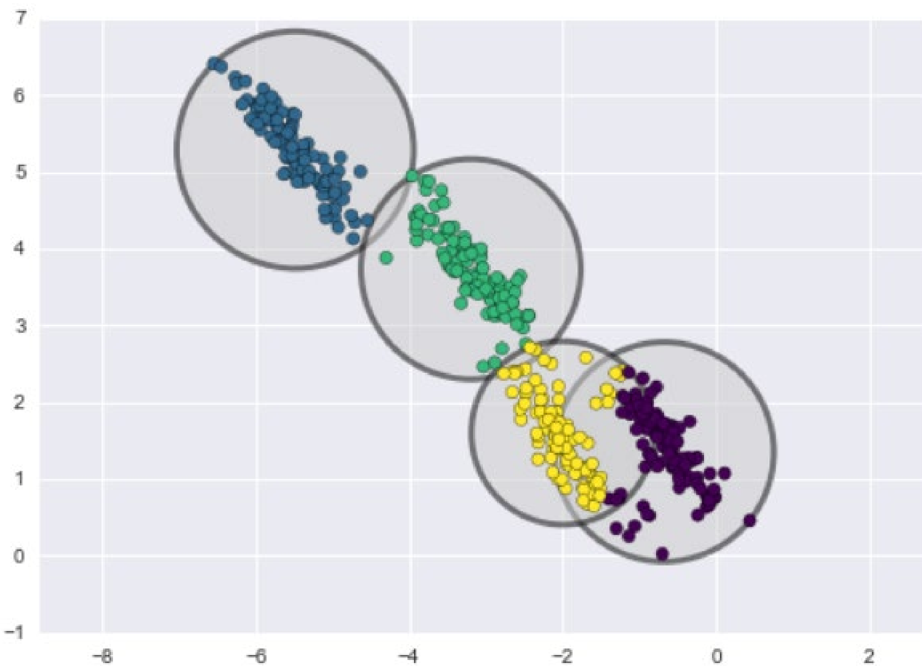


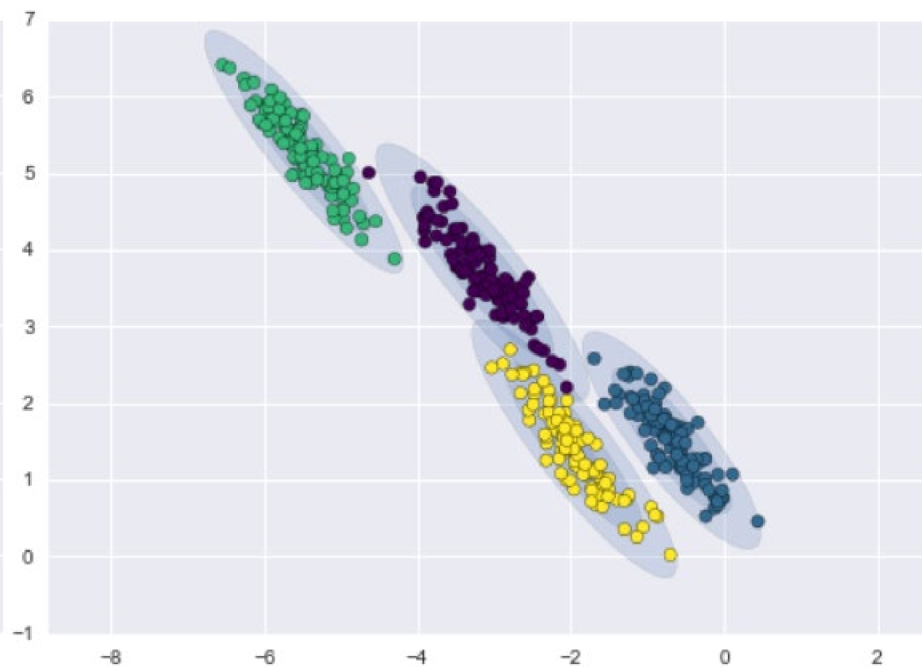


In Depth: Gaussian Mixture Models

(扩展内容)

- A Gaussian mixture model (GMM) attempts to find a mixture of multi-dimensional Gaussian probability distributions that best model any input dataset.

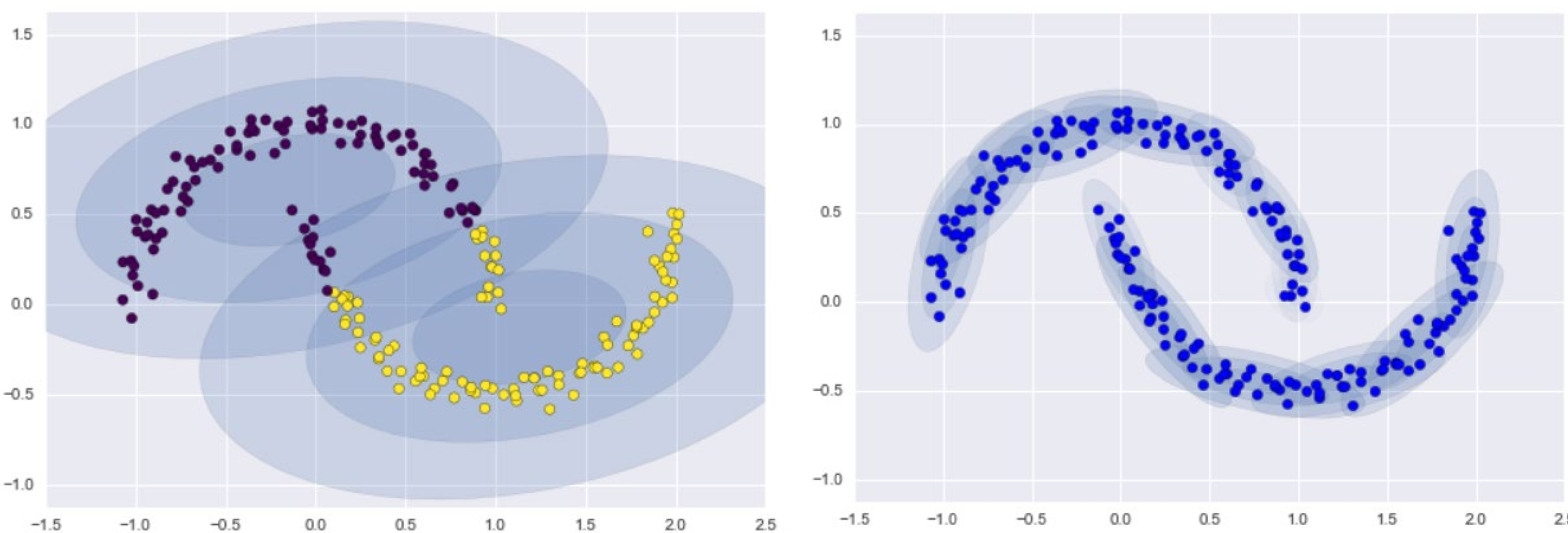




In Depth: Gaussian Mixture Models

(扩展内容)

- A Gaussian mixture model (GMM) attempts to find a mixture of multi-dimensional Gaussian probability distributions that best model any input dataset.



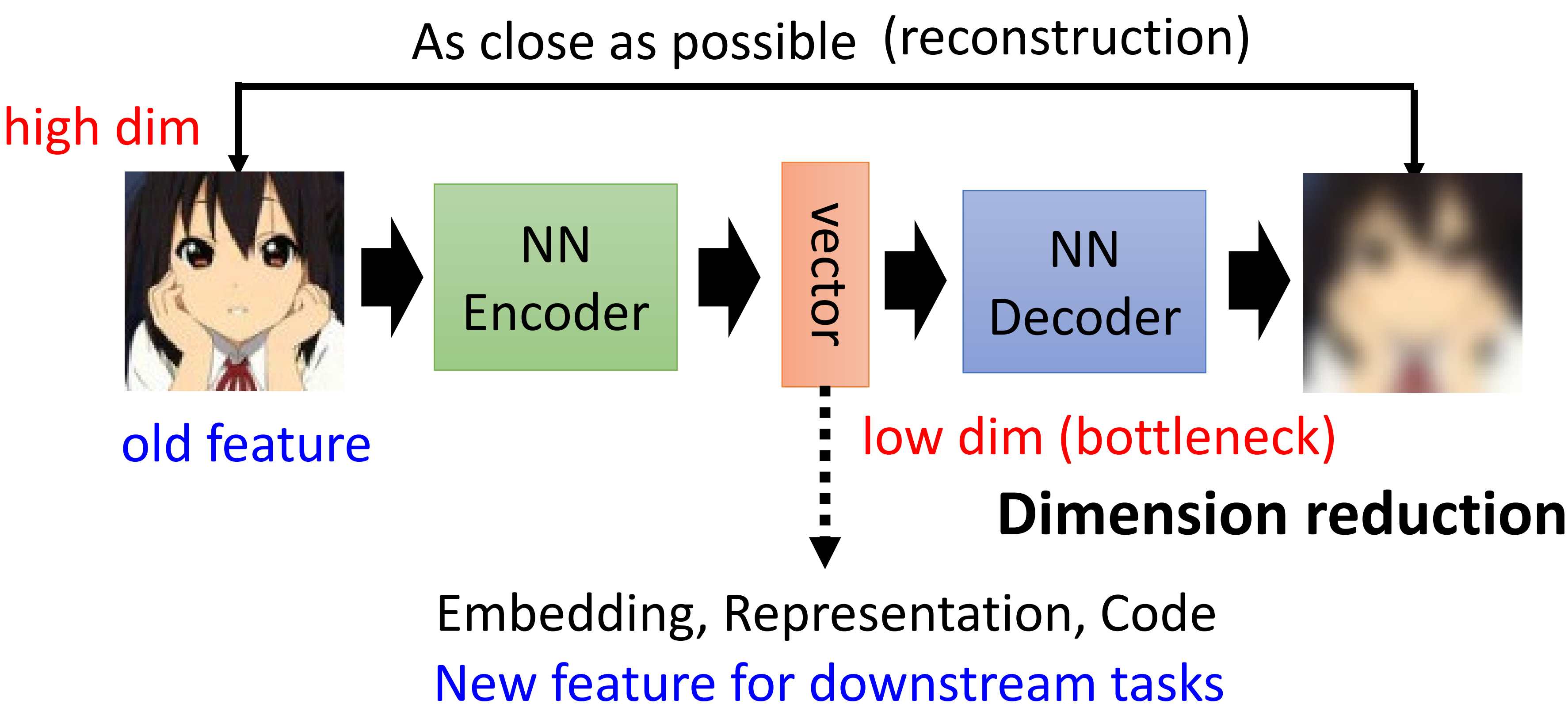
In Depth: Gaussian Mixture Models

（扩展内容）

- Expectation–Maximization (EM) algorithm
  - E-step: infer the posterior distribution of the latent variables given the model parameters.
  - M-step: tune parameters to maximize the data likelihood given the latent variable distribution
- EM methods iteratively execute E-step and M-step until convergence.

Auto-encoder

- Basic idea



More Dimension Reduction （扩展内容）

(not based on deep learning)

https://youtu.be/GBUEjkpoxXc

PCA

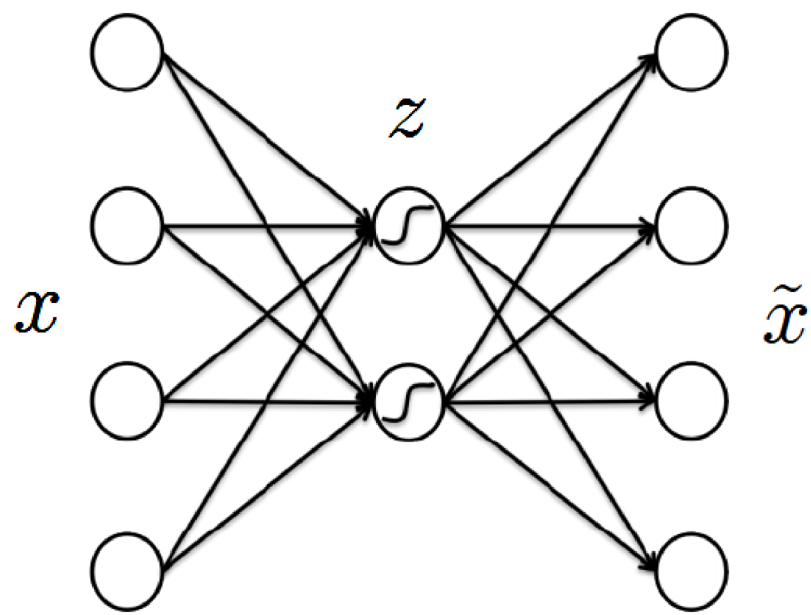
t-SNE

Auto-encoder

- An auto-encoder is an artificial neural net used for unsupervised learning of efficient codings.
  - Learn a representation (encoding) for a set of data, typically for the purpose of dimensionality reduction

$$z = \sigma(W_1 x + b_1)$$
$$\tilde{x} = \sigma(W_2 z + b_2)$$

z is regarded as the low dimensional latent factor representation of *x*



Auto-encoder

- Objective: squared difference between x and $\tilde{x}$

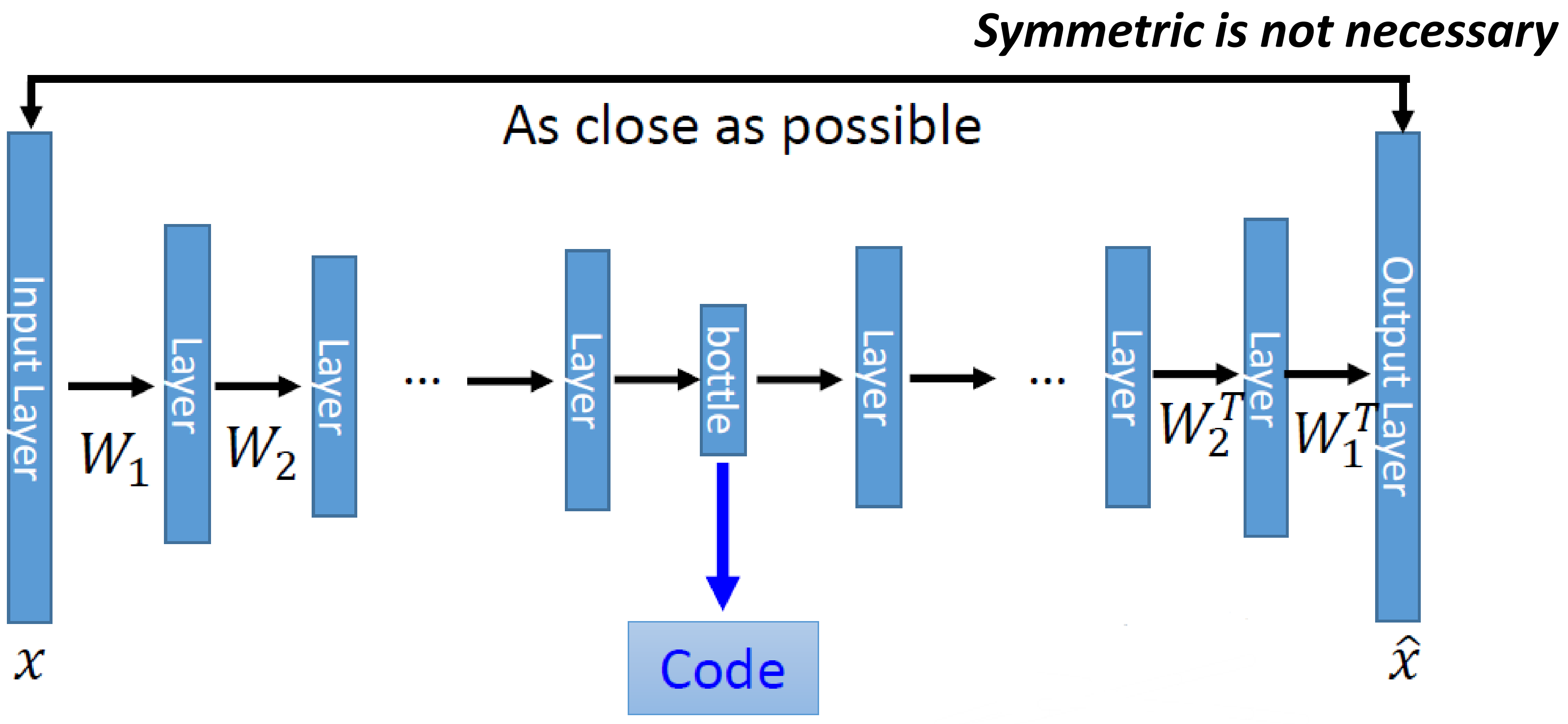
$$
\begin{aligned}
J(W_1, b_1, W_2, b_2) &= \sum_{i=1}^{m} (\tilde{x}^{(i)} - x^{(i)})^2 \\
&= \sum_{i=1}^{m} (W_2 z^{(i)} + b_2 - x^{(i)})^2 \\
&= \sum_{i=1}^{m} \left(W_2 \sigma(W_1 x^{(i)} + b_1) + b_2 - x^{(i)} \right)^2
\end{aligned}
$$

- Auto-encoder is an unsupervised learning model trained in a supervised fashion

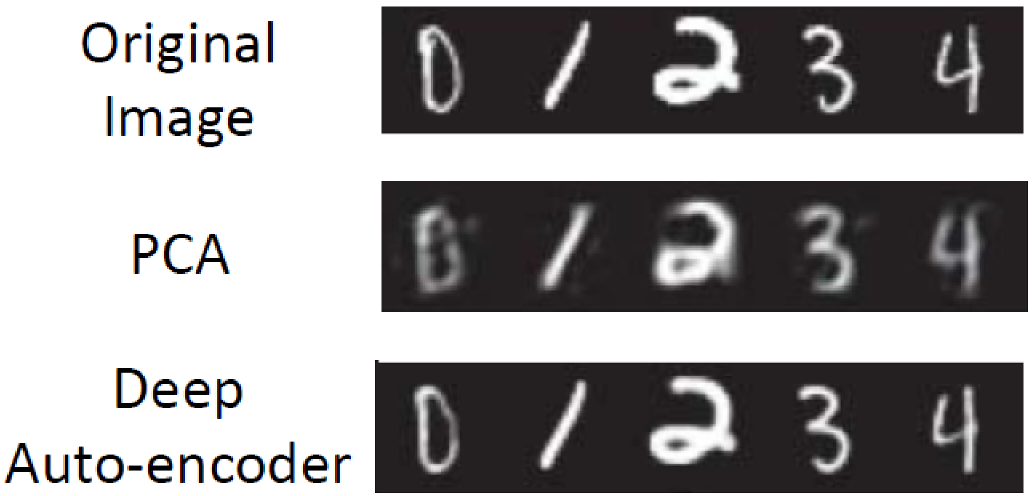
$$\theta \leftarrow \theta - \eta \frac{\partial J}{\partial \theta}$$

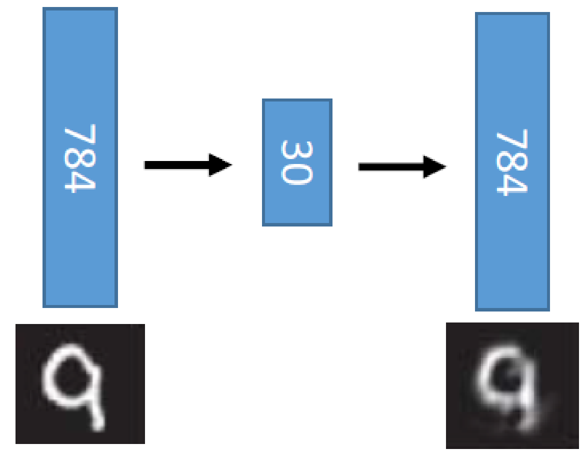
Deep Auto-encoder

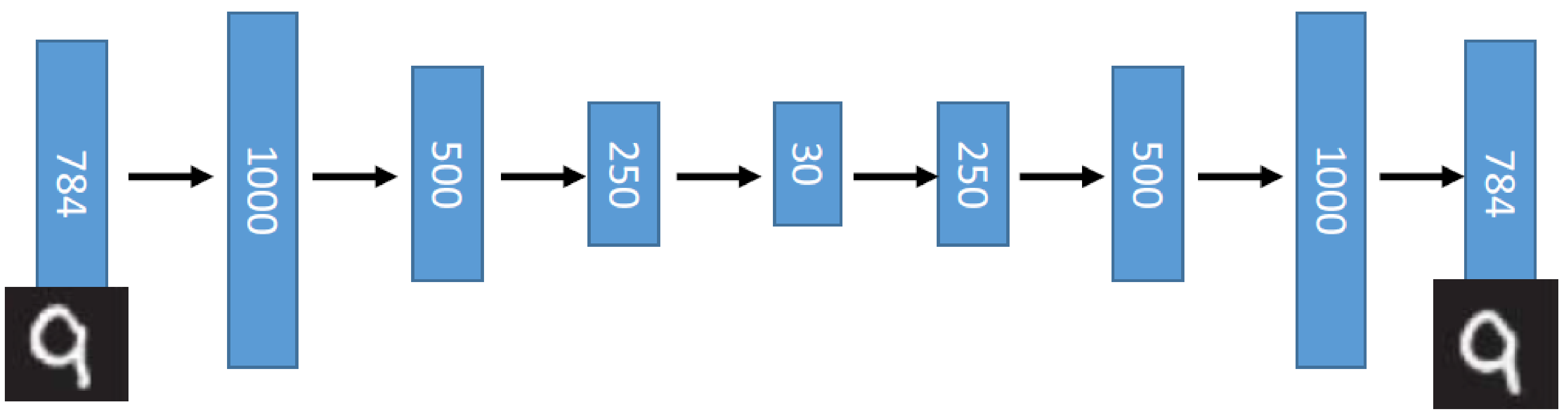
- Of course, the auto-encoder can be deep

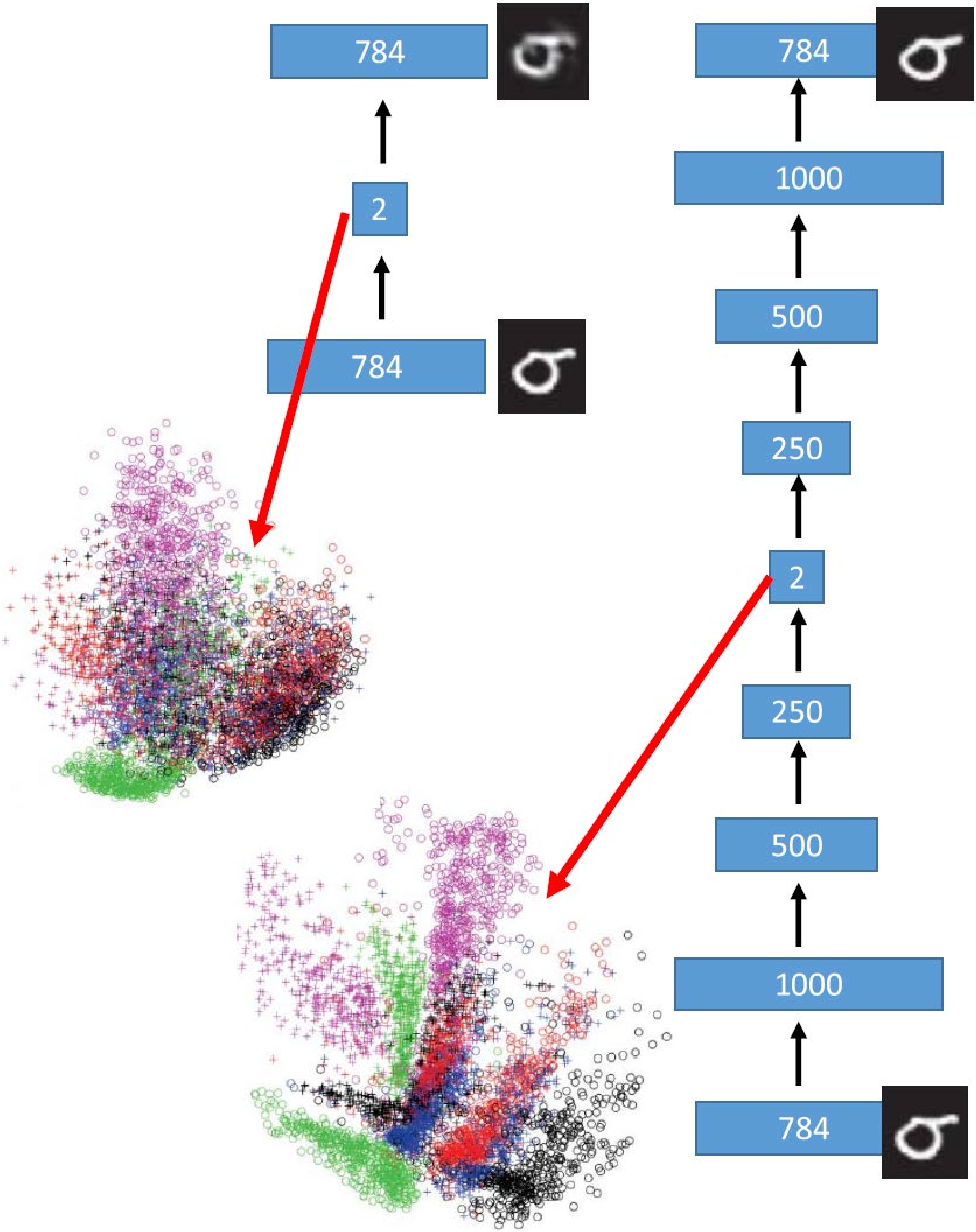


Deep Auto-encoder

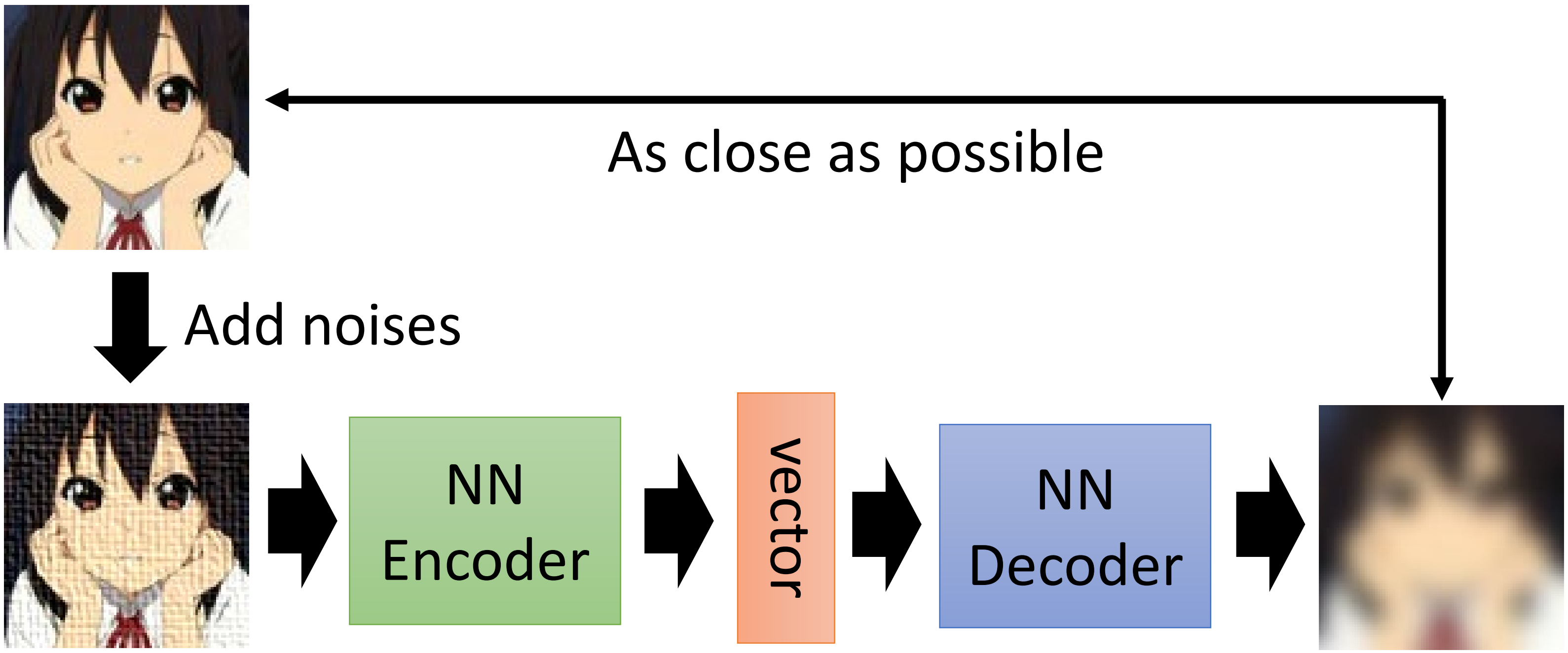






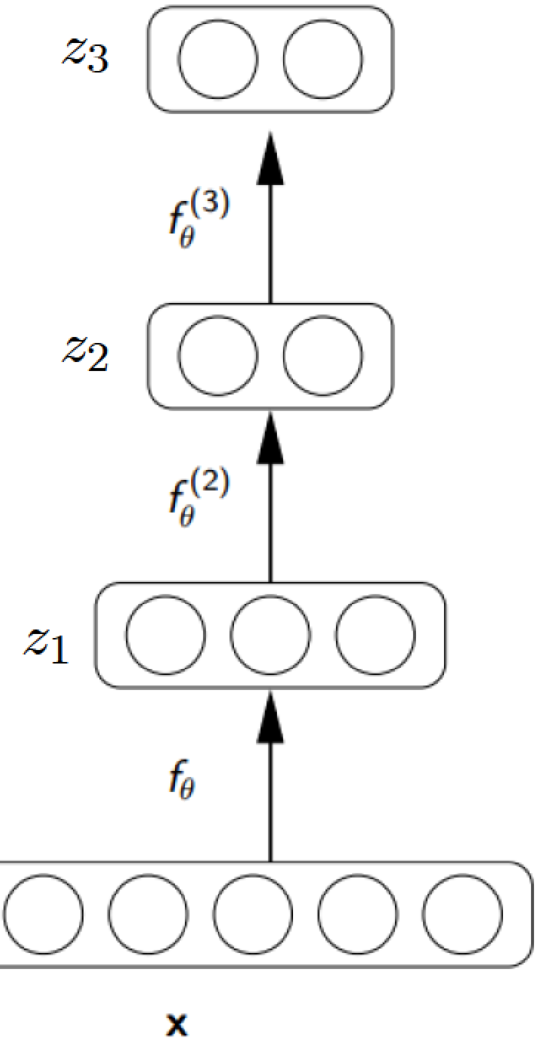

784
2
784
784
1000
500
250
2
250
500
1000
784

Denoising Auto-encoder



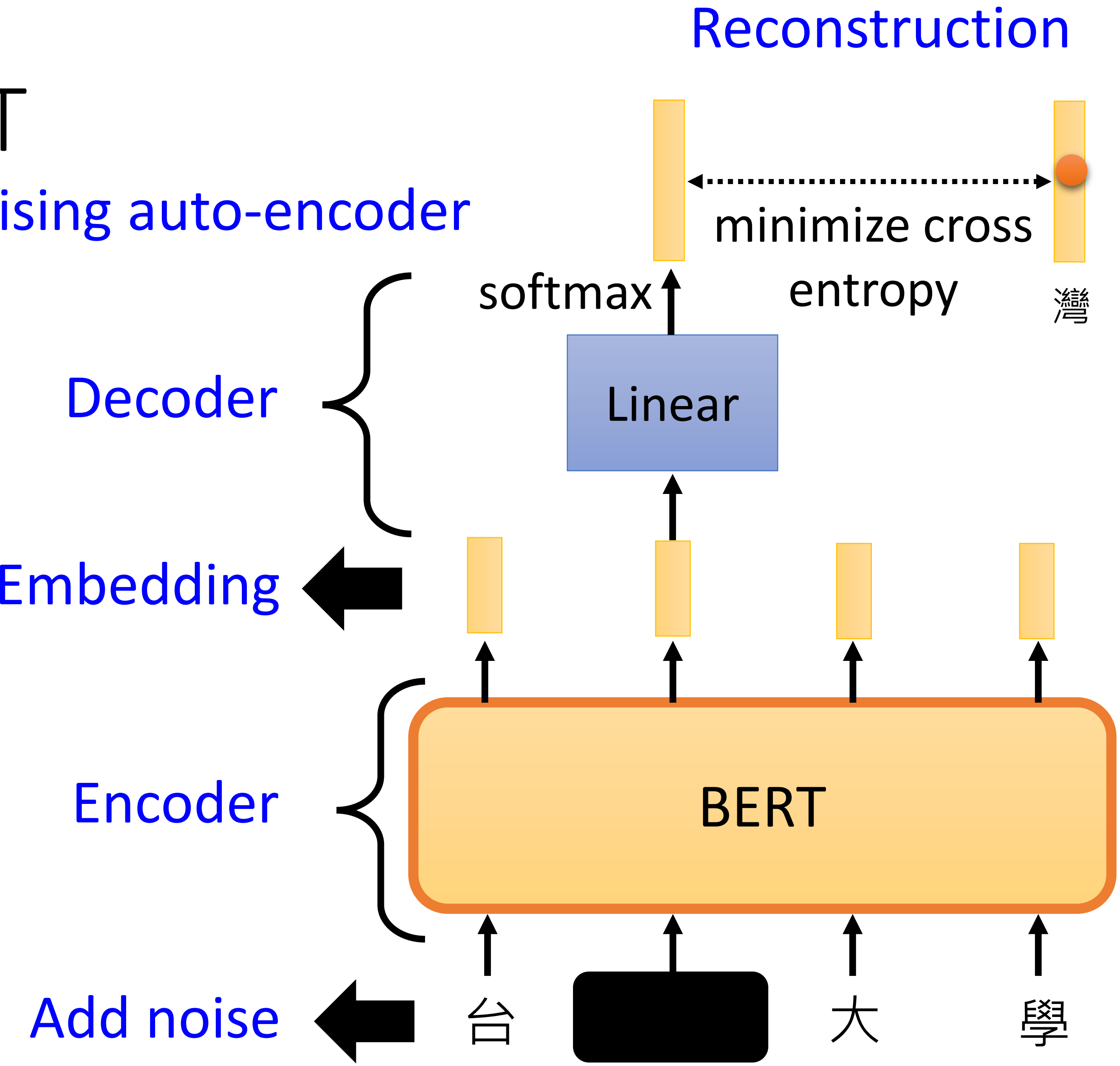
Stacked Auto-encoder

- Layer-by-layer training
  1. Train the first layer to use z_1 to reconstruct x
  2. Train the second layer to use z_2 to reconstruct z_1
  3. Train the third layer to use z_3 to reconstruct z_2



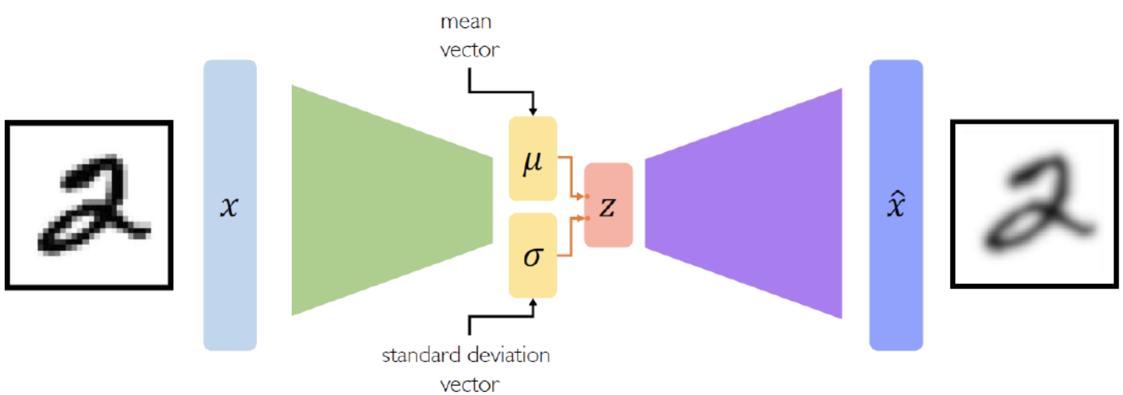
BERT

A de-noising auto-encoder



In Depth: VAE and SSL *（扩展内容）*

- The variational auto-encoder is a **generative model**.



- Self supervised learning
  - A form of unsupervised learning where the data provides the supervision.
  - In general, withhold some part of the data, and task the network with predicting it
  - The task defines a proxy loss, and the network is forced to learn what we really care about, e.g. a semantic representation, in order to solve it.