Karlsruher Institut für Technologie

Machine Learning 1 – Fundamentals

Unsupervised Learning

Prof. Dr. J. M. Zöllner, M.Sc. Nikolai Polley, M.Sc. Marcus Fechner

Outline

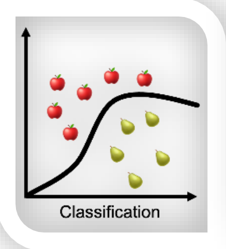
- Motivation
- Clustering
- Dimensionality Reduction / Feature Extraction
- Outlook: Advanced Methods

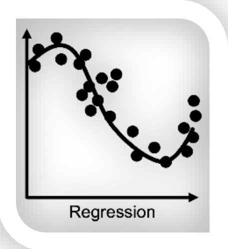
Types of Learning

Repetition

Supervised Learning

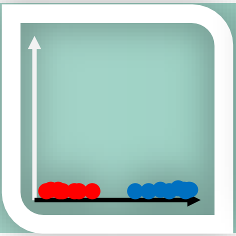
- **Data**: Examples with input and desired output data (labeled data).
- **Goal**: Learn the relationship between input and output data in order to predict the correct output for new input data.
- Seen in lectures: Decision trees, neural networks and SVM

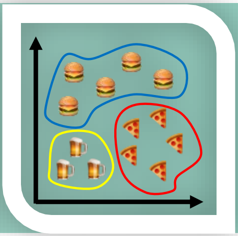




Unsupervised Learning

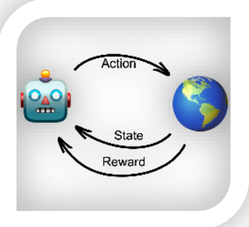
- **Data**: Examples contain only input data (unlabeled data).
- **Goal**: Find the underlying structure in the data.





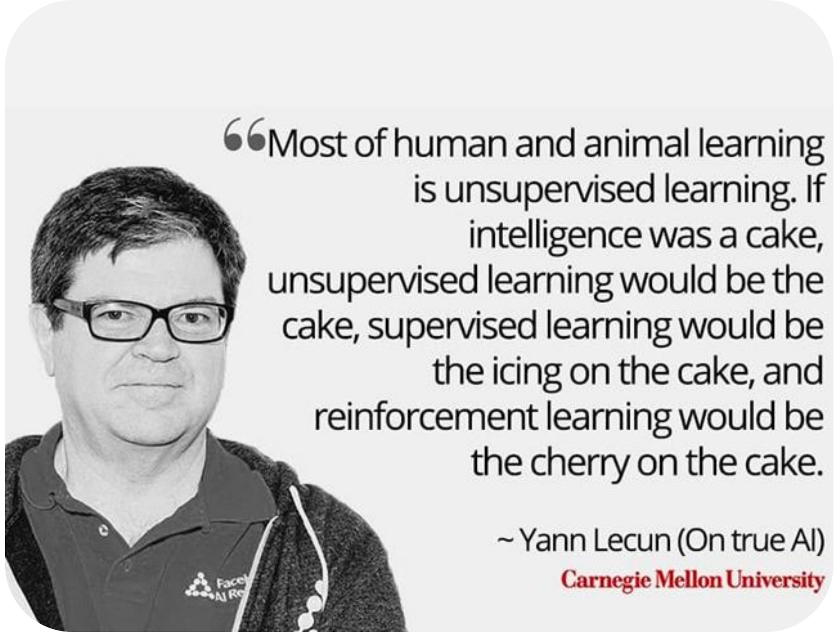
Reinforcement Learning

- **Data**: Experience from interaction with an environment and resulting reward.
- **Goal**: Learn a behavior that maximizes the reward in the long term.



Why Unsupervised Learning?

- Annotating training data is often the most expensive aspect of creating a dataset for machine learning. Obtaining vast quantities of unlabeled data, however, is relatively inexpensive
  - Download images/text from internet
  - Record sensor data while driving a car
- Unsupervised learning allows the discovery of new pattern relationships and insights into the structure of the underlying data
  - Can be used as a foundation for other methods



Unsupervised Learning - Basics

- Exploiting “similarities” in training data to:
  - agglomerate classes/clusters (iteratively)
  - extract essential characteristics, i.e. features

- Analogy to human learning

  A student:
  - gradually learns new concepts
  - observes objects/events
  - forms (hierarchical) concepts that summarize and organize the experiences
  - finds a correct representation of data, which enables faster learning

Unsupervised Learning - Basics

- **Classical Clustering Algorithms**
  - k-means-Clustering
  - Agglomerative Hierarchical Clustering 层次聚类
- **Densitiy Based Clustering Algorithms**
  - DBSCAN
  - OPTICS

密度聚类

- **Neural Network based Algorithms (Generative Models)**
  - Autoencoder, Variational Autoencoder, Self-Supervised Learning
  - Bidirectional networks (RBM) 双向网络
- **Conceptual Clustering:** CLUSTER/2
- **Concept Formation:** COBWEB, CLASSIT
- **Learning by Discorvery:** BACON, ABACUS

Outline

- Motivation
- Clustering 聚类
- Dimensionality Reduction / Feature Extraction
- Outlook: Advanced Methods

K-means-Clustering (Lloyd, 1982)

- Very simple and yet frequently used
- Divides a data set into a (usually) pre-determined number of clusters

将数据划分为 k 个聚类，每个聚类由一个中心点（质心）表示。

- **Basic Idea**:
  - Defining a center point for each cluster
  - Iterative adaptation/improvement
    - … regarding data belonging to the cluster
    - … regarding center of cluster (also called centroids)
  - **Optimality Criterion:** Minimization of distance between all data points and their respective centroids

目标是最小化所有数据点与其质心的距离平方和。

K-means-Clustering – Formal

- **Given**:
  - Set $\boldsymbol{x}$ of unlabeled training examples, each with d attributes:
    $$\boldsymbol{x} = [x_1, x_2, \ldots, x_d]$$
  - Number of clusters k

- **Objective**:
  - Partition the training set into clusters $X_1, \ldots X_k$ (with center points $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k$) such that a minimum of distance between data and centroids is reached.
    $$\mathcal{L} = \min \sum_{j=1}^{k} \sum_{\boldsymbol{x}_i \in X_j} \left\| \boldsymbol{x}_i - \boldsymbol{c}_j \right\|^2$$

K-means-Clustering – Lloyd’s Algorithm

- Place k points $\boldsymbol{c}_j$ "randomly" in the d-dimensional space to initialize the centers of the clusters / centroids
- Iterative Process:
  - **While:** $\boldsymbol{c}_j$ changes
    - **Assign** each element $\boldsymbol{x}_i$ towards its nearest centroid $\boldsymbol{c}_l$:

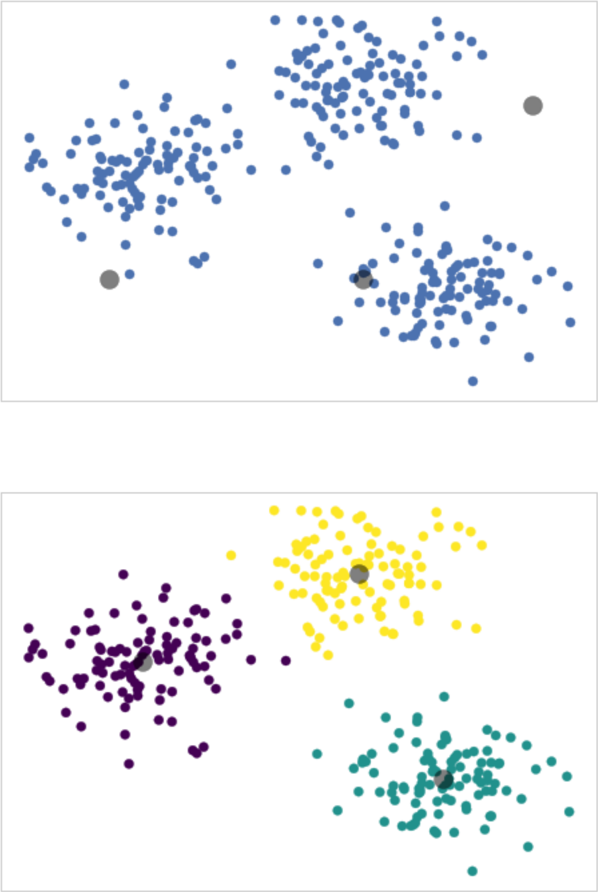
$$l = \underset{1 \leq j \leq k}{\arg\min} \left\| \boldsymbol{x}_i - \boldsymbol{c}_j \right\|^2 \Longrightarrow \boldsymbol{x}_i \in X_l$$

    - **Recalculate centroids** $\boldsymbol{c}_j$ for each resulting Cluster X_j :

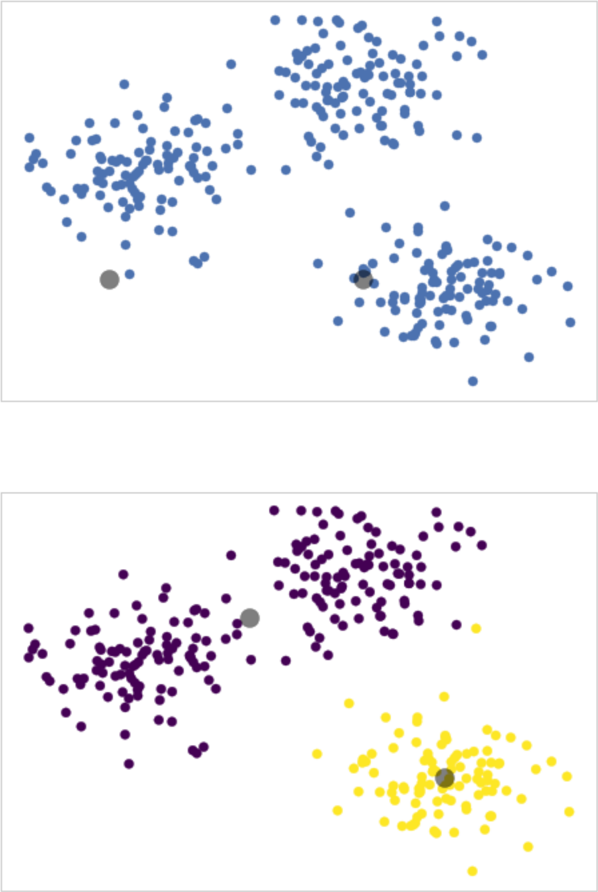
$$\boldsymbol{c}_j = \frac{\sum_{x_j \in X_j} \boldsymbol{x}_j}{|X_j|}$$

K-means-Clustering – Examples

1.1 (k=3)



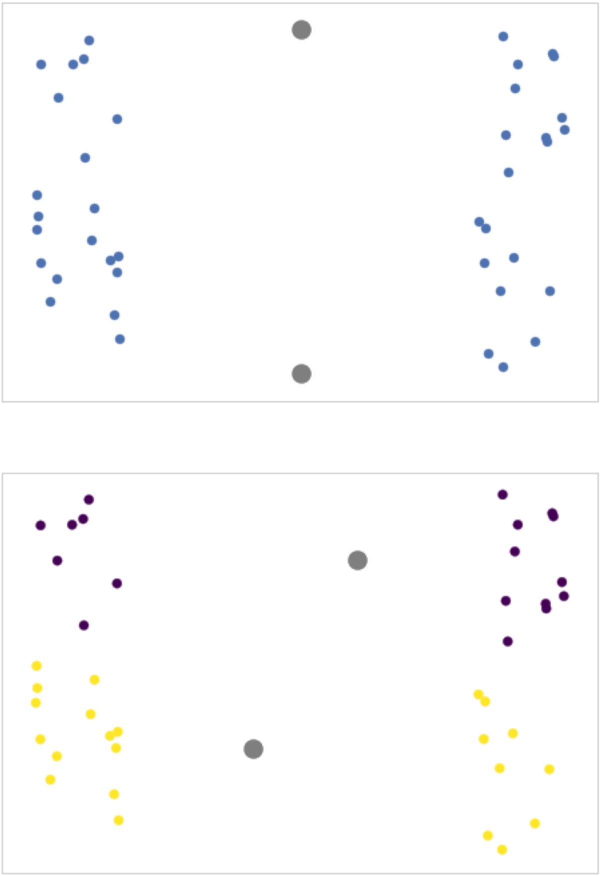
1.2 (k=2)



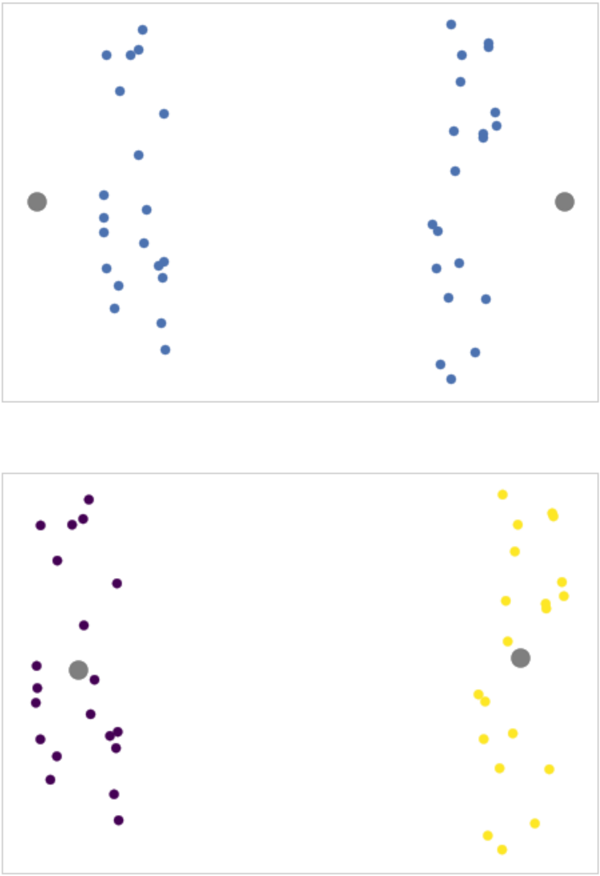
2.1(k=2)

K-means-Clustering – Examples

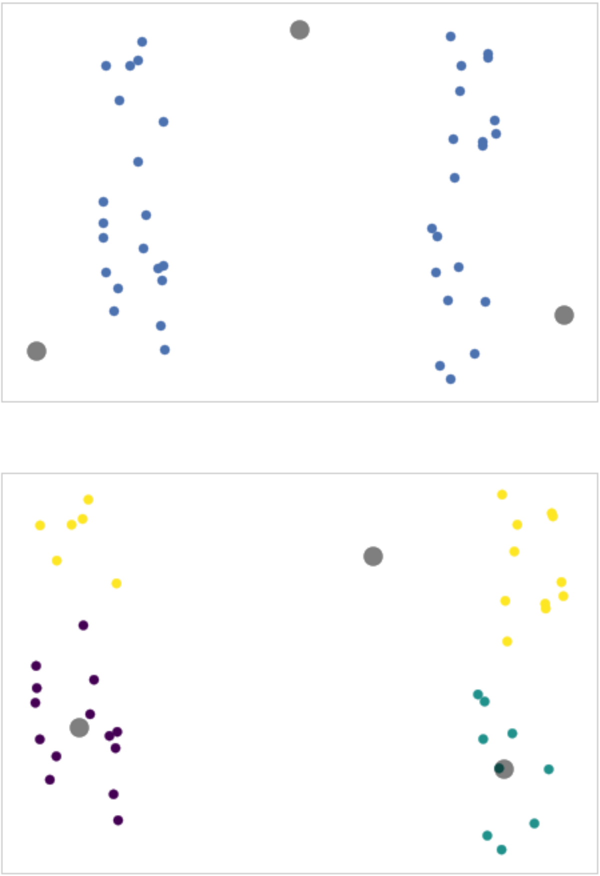
3.1 (k=2)



3.2 (k=2)



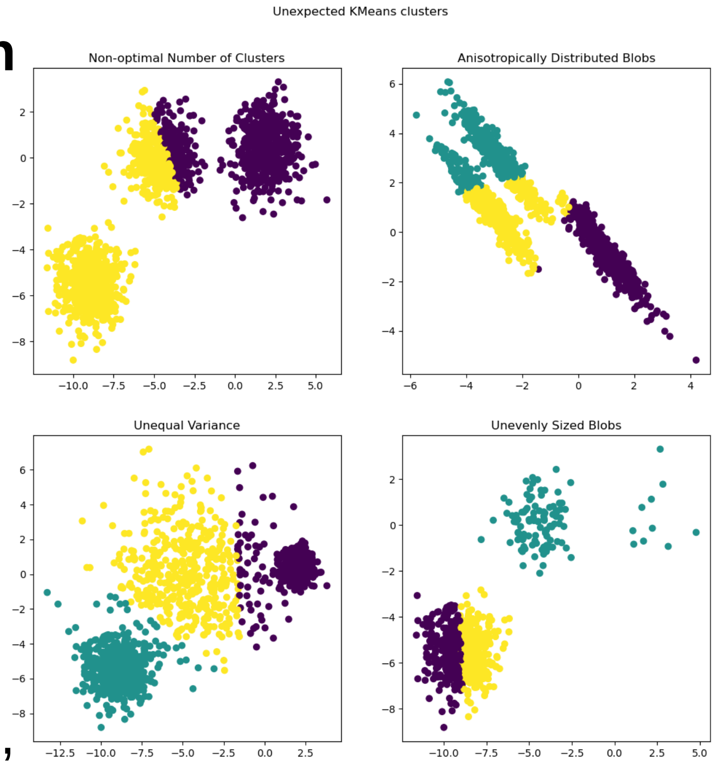
3.3 (k=3)



K-means-Clustering – Evaluation I

- Solving the problem is NP-hard, so Lloyd's algorithm is used, which converges to only a locally optimal solution
- Results are **highly dependent** on the **number** k and **initialization** of centroids $\boldsymbol{c}_j$
  - Example 1.2:
    - **Failure:** Wrong number of initial clusters
    - **Solution:** Choose correct number of cluster
  - Example 3.1 and 3.3:
    - **Possible Failure:** Potentially wrong cluster
    - **Possible Solution**: Initialize the algorithm multiple times with different centroid starting points
- Results are dependent of used distance metric $|\boldsymbol{x} - \boldsymbol{c}_j|$ 维度灾难
  - **Curse of dimensionality**! In high-dimensional representations, all data is dissimilar → Makes it harder to find clusters

在高维空间中，所有数据点可能都显得不相似，从而难以形成清晰的聚类结构。

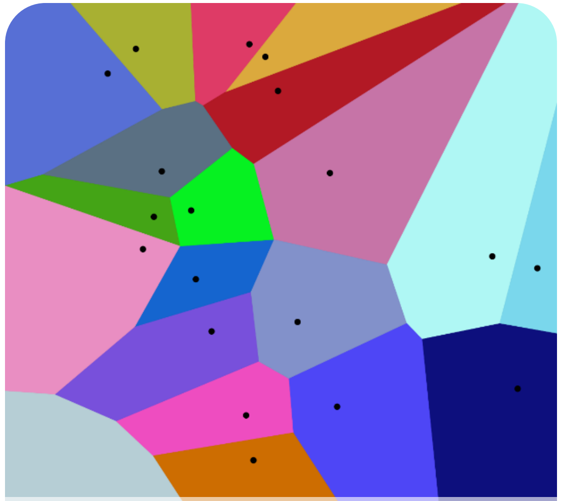


K-means-Clustering – Evaluation II

- Centroids create Voronoi cells which are used for testing and inference of method

  所以需要用先验知识来确定合适的k，规避overfitting

- Results depend on the correct choice of k
  - No sound theoretical solutions
  - Can we determine k from domain?
    - E.g. optical letter recognition $\Rightarrow k = 26$



Classification based on centroids

  - Triggering k-means method multiple times with different values for k
    - Termination if result meets a certain optimality criterion 如果结果符合特定的最优标准，则终止
      - Problem: Possibly overfitting on training data
      - Challenge: Find a good optimality criterion (e.g., minimum number of data points per cluster, maximum number of clusters...)

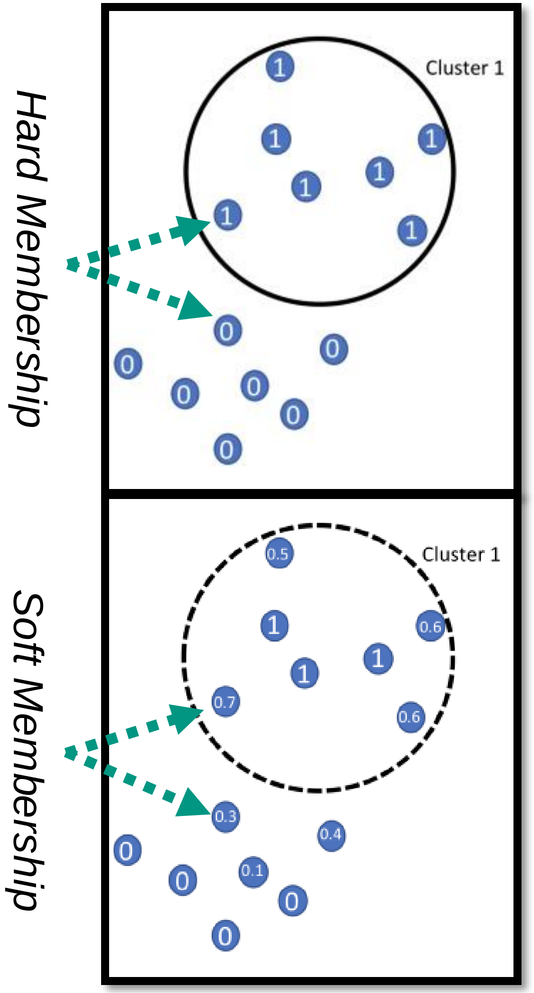
Fuzzy-k-means-Clustering I

- **Regular k-means**
  - Each instance belongs to exactly one cluster.

模糊K聚类

- **Softening**: Each instance $\boldsymbol{x}_i$ contains “soft” probabilities for a membership of cluster X_j
  - $P(X_j|\boldsymbol{x}_i)$ ”Probability measure of membership“
  - $P(X_j|\boldsymbol{x}_i) \sim 0$: Instance is far away from centroid
  - $P(X_j|\boldsymbol{x}_i) \sim 1$: Instance is close/inside the cluster
  - P is normalized over all clusters X_j : $\sum_{X_j} P(X_j \mid \boldsymbol{x}_i) = 1$

每个数据点 x_i 对所有聚类的概率总和为1



Fuzzy-k-means-Clustering II

- Cluster belonging of instance $\boldsymbol{x}_i$ towards X_j with centroid $\boldsymbol{c}_j$,is characterized by distance d_{ij} between $\boldsymbol{x}_i$ and $\boldsymbol{c}_j$

b is a hyperparameter that scales the fuzziness/softness with default value $= 2$

隶属于某个簇的隶属度公式
（其中 b 是控制模糊性的超参数，默认为2。）

$$P(X_j|\boldsymbol{x}_i) = \frac{\left(\frac{1}{d_{ij}}\right)^{\frac{1}{b-1}}}{\sum_{r=1}^{k}\left(\frac{1}{d_{ir}}\right)^{\frac{1}{b-1}}} \qquad \text{with } d_{ij} = \left\|\boldsymbol{x}_i - \boldsymbol{c}_j\right\|^2$$

- p is normalized over clusters X_j

$$\forall i = 1, \dots, n: \qquad \sum_{j=1}^{k} P(X_j|\boldsymbol{x}_i) = 1$$

Fuzzy-k-means-Clustering III

模糊隶属度影响质心更新，每次迭代根据当前隶属度计算新的质心位置：

- Iterative adaptation of $\boldsymbol{c}_j$ considers the fuzzy belongings of all data points $\boldsymbol{x}_i$

$$\boldsymbol{c}_j = \frac{\sum_{i=1}^{n} P\left(X_j|\boldsymbol{x}_i\right)^b x_i}{\sum_{i=1}^{n}\left(P(X_j|\boldsymbol{x}_i)\right)^b}$$

- Higher b-values reduce the influence of distant instances

- **Problem**: Runtime = $O(\boldsymbol{k}n)$ for each training iteration

运行时间（复杂度），k为族的数量，n为数据点数

密度聚类

DBSCAN – Density Based Spatial Clustering

- Density based method
- Iteratively assigns data points to one of three classes:
  - Core sample
  - Neighbor
  - Noise

- Clusters are sets of core samples and their respective neighbors

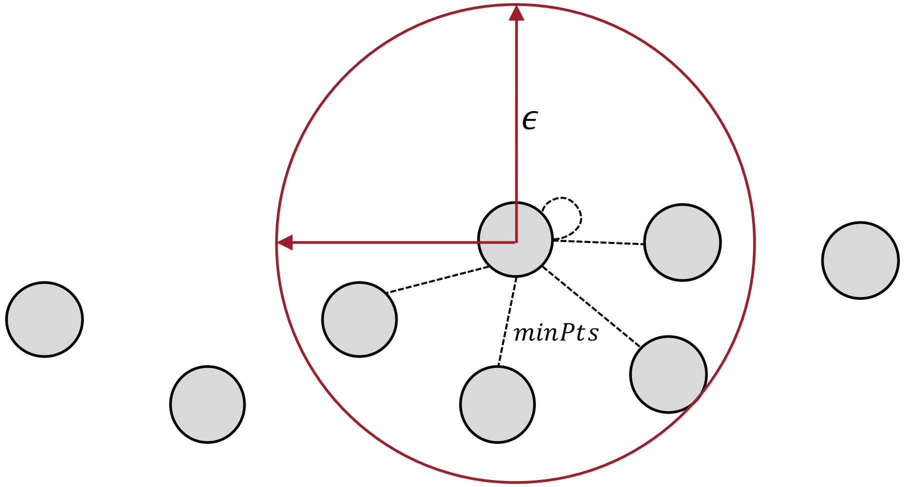
- Separates high density areas from low density areas

DBSCAN – Density Based Clustering

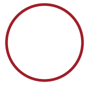
- Algorithm is parametrized by $minPts$ and a distance measure ε
- **Core Sample**: A point is a core sample if at least $minPts$ points are within distance ε
- **Neighbor**: A point is a neighbor if there are not $minPts$ points within distance ε, but at least one point within distance ε is a core sample.
- **Noise**: There is no core sample within distance ε

将点的噪声/归属于某个簇的划分归为两个参数 minPts（最小点数）+epsilon（半径参数）

这三个定义看

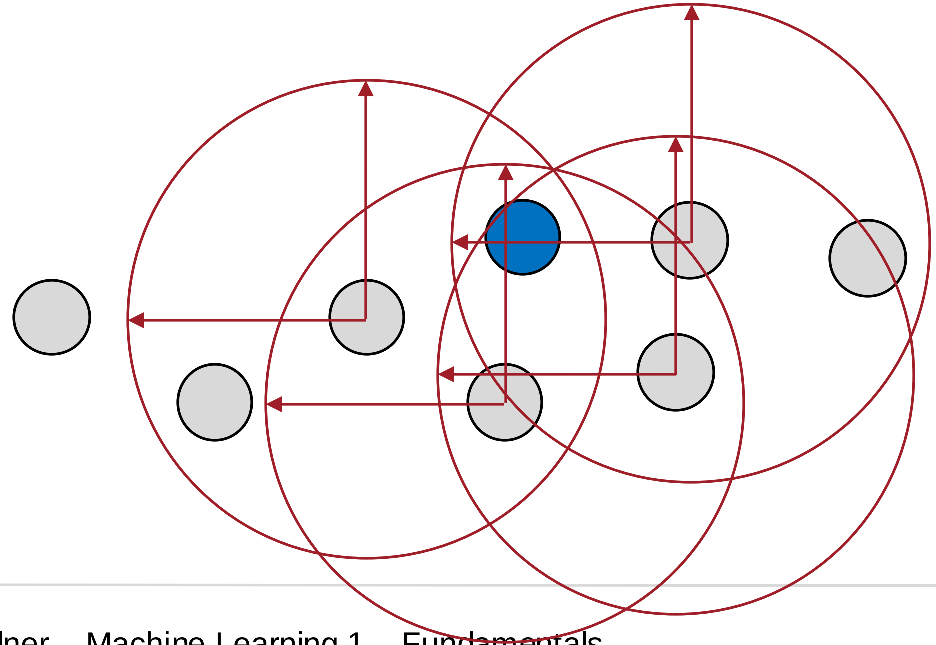


$minPts = 4$ *(including itself)*

$\epsilon =$ 

DBSCAN – Density Based Clustering

- Algorithm is parametrized by $minPts$ and a distance measure ε
- **Core Sample**: A point is a core sample if at least $minPts$ points are within distance ε
- **Neighbor**: A point is a neighbor if there are not $minPts$ points within distance ε, but at least one point within distance ε is a core sample.
- **Noise**: There is no core sample within distance ε

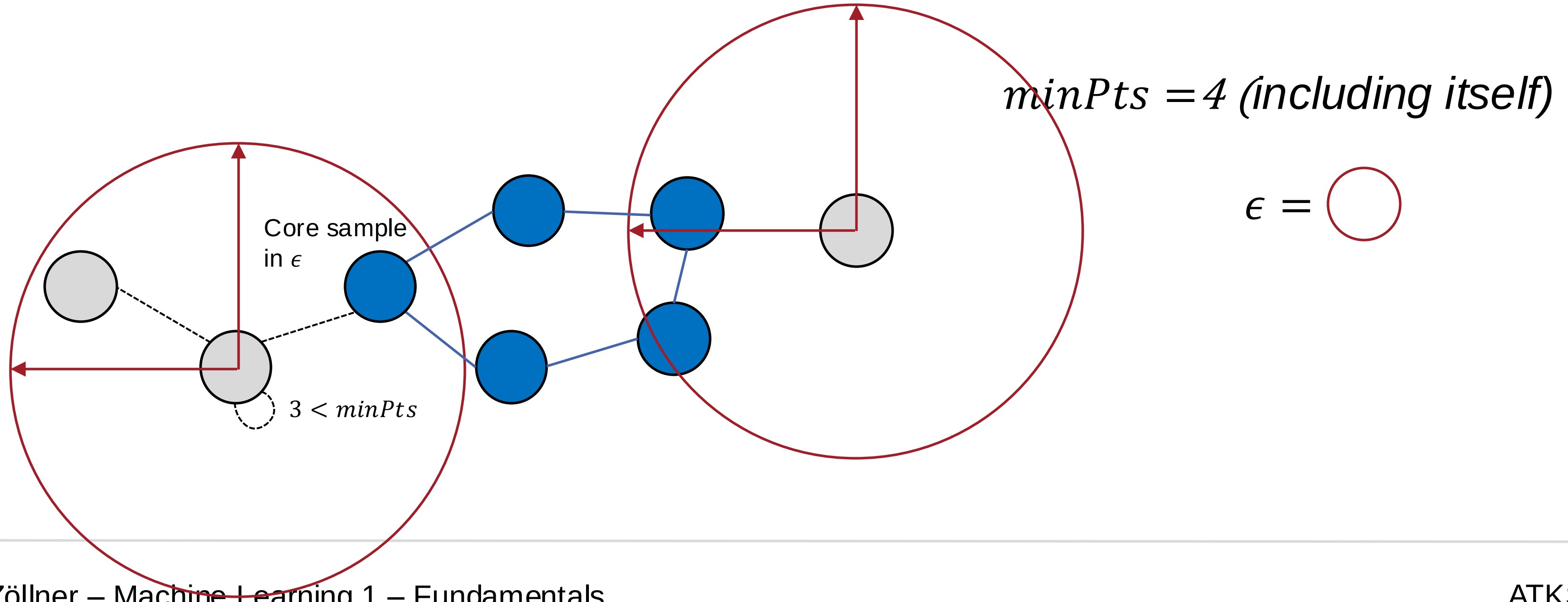


$minPts = 4$ *(including itself)*

$\epsilon =$ ◯

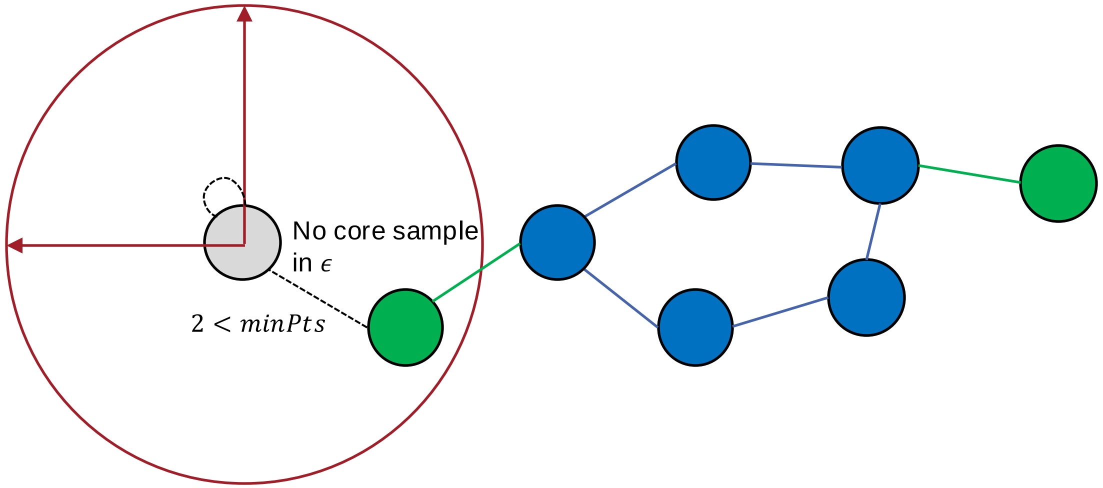
DBSCAN – Density Based Clustering

- Algorithm is parametrized by $minPts$ and a distance measure ε
- **Core Sample**: A point is a core sample if at least $minPts$ points are within distance ε
- **Neighbor**: A point is a neighbor if there are not $minPts$ points within distance ε, but at least one point within distance ε is a core sample.
- **Noise**: There is no core sample within distance ε

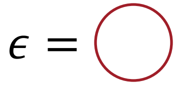


DBSCAN – Density Based Clustering

- Algorithm is parametrized by $minPts$ and a distance measure ε
- **Core Sample**: A point is a core sample if at least $minPts$ points are within distance ε
- **Neighbor**: A point is a neighbor if there are not $minPts$ points within distance ε, but at least one point within distance ε is a core sample.
- **Noise**: There is no core sample within distance ε



$minPts = 4$ *(including itself)*

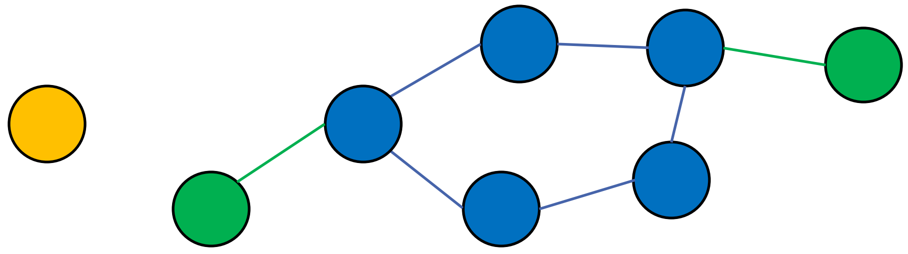


GDBSCAN: Pseudocode for Generalized DBSCAN

一种DBSCAN的泛化版本

```
GDBSCAN(D, getNeighbors, isCorePoint)
  C = 0
  for each unvisited point P in D
    mark P as visited
    N = getNeighbors(P)
    if isCorePoint(P, N)
      C = next cluster
      expandCluster(P, N, C)
   else mark P as NOISE

expandCluster(P, N, C)
  add P to cluster C
  for each point P' in N
    if P' is not visited
       mark P' as visited
       N' = getNeighbors(P‘)
       if isCorePoint(P', N‘)
         N = N joined with N‘
    if P' is not yet member of any cluster
      add P' to cluster C
```



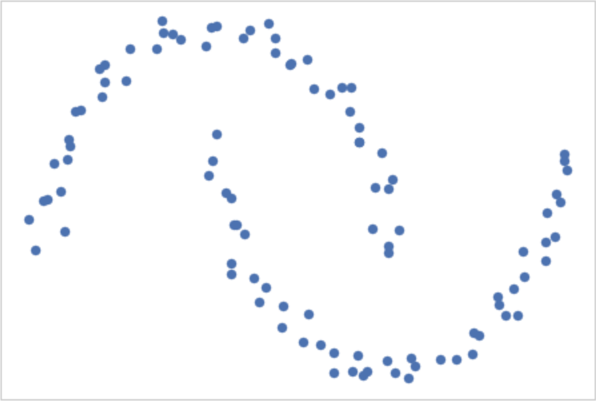
```
getNeighbors(P):
        returns Points in ε-distance of P
isCorePoint(P,N):
        densitiy estimation
```

(G)DBSCAN vs. k-means-Clustering

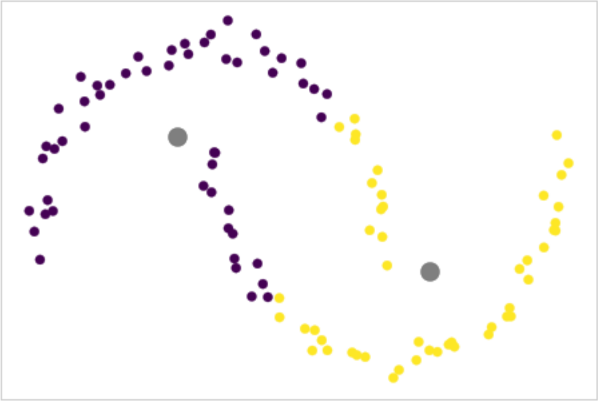
DBSCAN的优缺点

- DBSCAN is robust against noise ⟶ incorporates noisy samples into algorithm
- DBSCAN can cluster different data densities
  - Challenge: correctly define “density” with $minPts$ and ε
- DBSCAN does not require a priori knowledge about number of clusters

unclustered



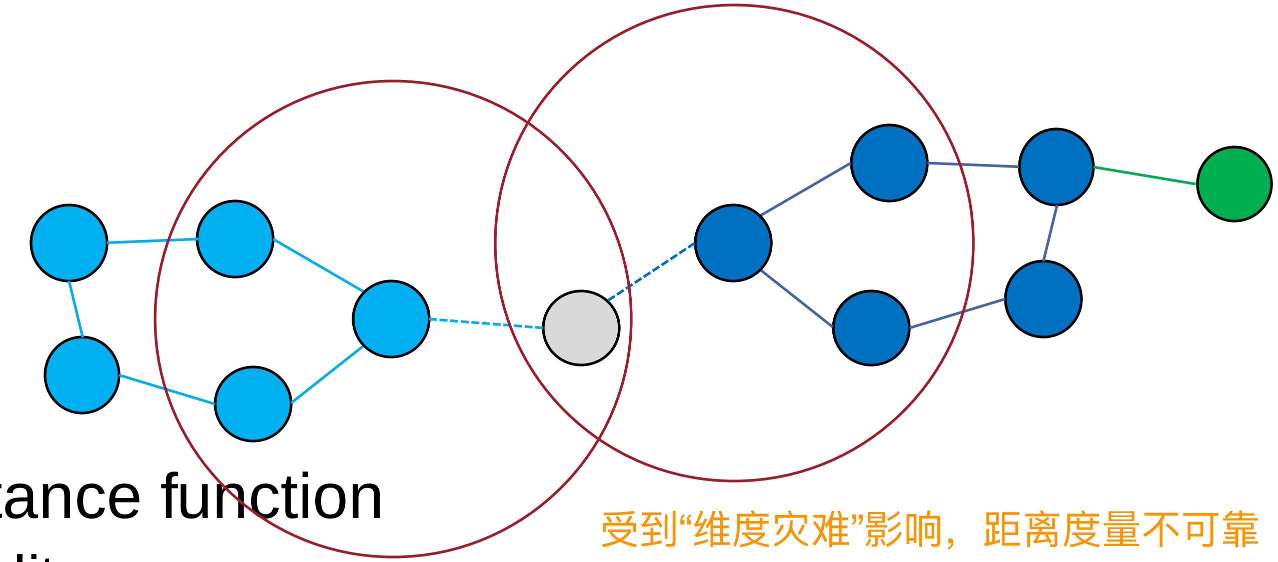
k-means



DBSCAN

(G)DBSCAN: Disadvantages

- DBSCAN non-deterministic e.g. 不确定性-点的处理顺序会影响结果
  - Grey point is neighbor of two separate core samples but no core sample itself
  - Cluster assignment depends on the order of processing



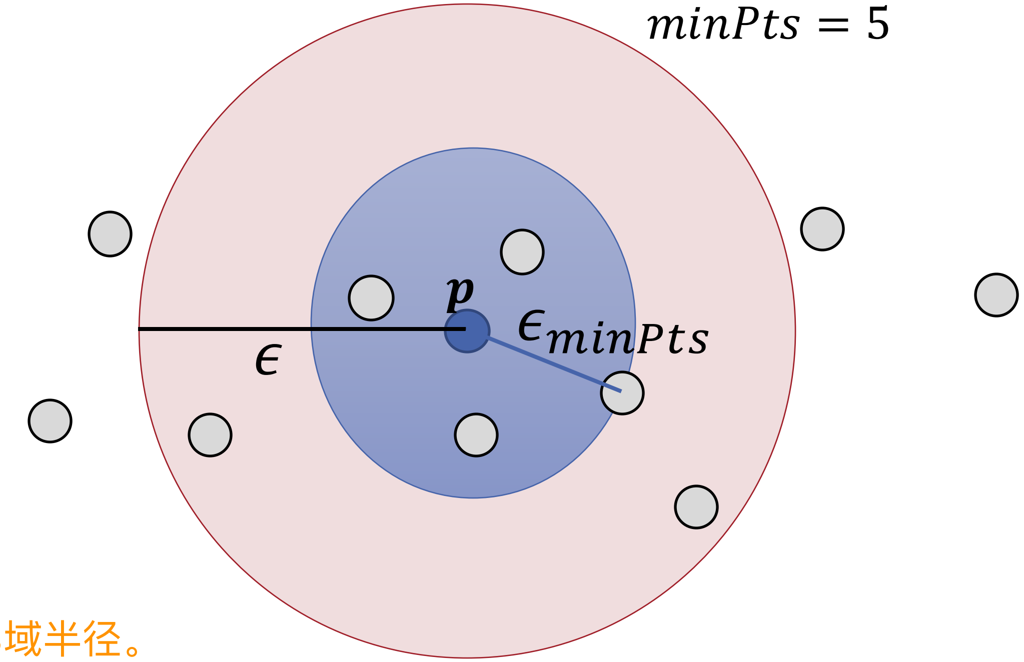
- Still dependent of distance function
  - Curse of Dimensionality

受到“维度灾难”影响，距离度量不可靠

- Fitting Selection of ε, $minPts$ can be difficult (different densities) 参数选择敏感

OPTICS – Ordering Points To Identify the Clustering Structure

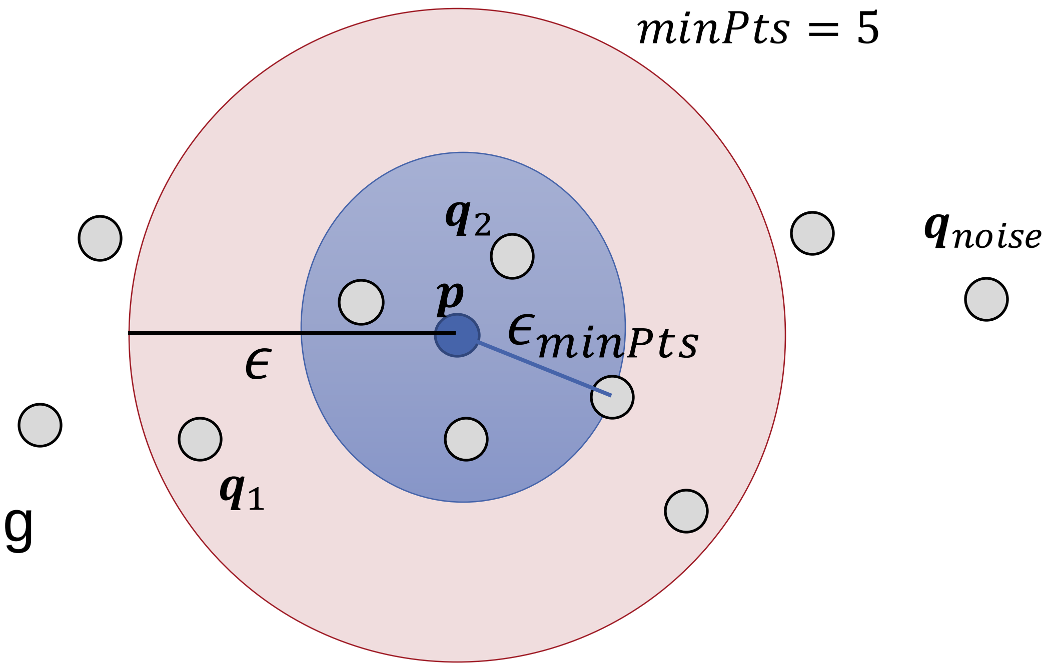
- OPTICS is extension of DBSCAN
  - Hyperparameter $minPts$ still used to categorize core samples
  - Hyperparameter ε still used but influence is reduced (upper bound)
  - Extension: Calculate distance ε_{minPts} , from which a point would qualify as core sample (i.e. $minPts$ in its vicinity)
  - ε_{minPts} is also called the *core distance* of a point



相对于DBSCAN的拓展内容：
引入了ε_minPts（核心距离）：表示一个点成为核心点所需的最小邻域半径。
通过计算点之间的reachability distance（可达距离）进行排序，以更灵活地识别聚类边界。

OPTICS – Ordering Points To Identify the Clustering Structure

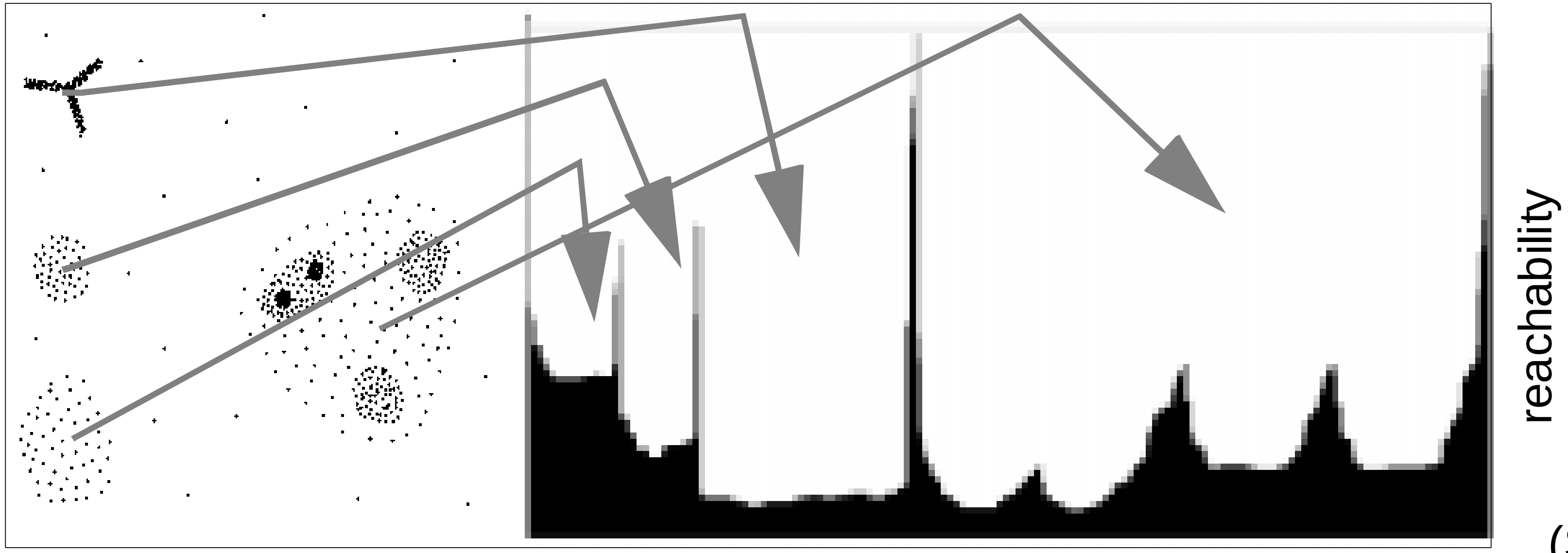
- **Definition**: $reachability_{\epsilon,minPts}(\boldsymbol{q},\boldsymbol{p}) = \begin{cases} undefinded, \ if \ |N_{\varepsilon}(\boldsymbol{p})| < \ minPts \\ \max(\varepsilon_{minPts}, distance(\boldsymbol{p},\boldsymbol{q})) \end{cases}$
  - $reachability(\boldsymbol{p},\boldsymbol{q}_{noise}) = undefined$
  - $reachability(\boldsymbol{q}_1,\boldsymbol{p}) = distance(\boldsymbol{p},\boldsymbol{q}_1)$
  - $reachability(\boldsymbol{q}_2,\boldsymbol{p}) = \ \varepsilon_{minPts}$



- Extension of DBSCAN with:
  - Iterative sorting of neighbors according to their (defined) reachability
  - Processing and assignment to clusters in sorting order (new cluster or noise depending on ε)

OPTICS – Ordering Points To Identify the Clustering Structure

- **Conclusion**:
  在reachability图中，低谷对应潜在聚类。
  - Cells with large number of objects are potential cluster centers and are visible as “valleys” in reachability/cluster-order histogram.
  - Complete processing of new clusters is possible algorithmically (see [8]) and provides the clusters (“valleys”) 可进一步创建子聚类或调整参数获得更精确的聚类结果。

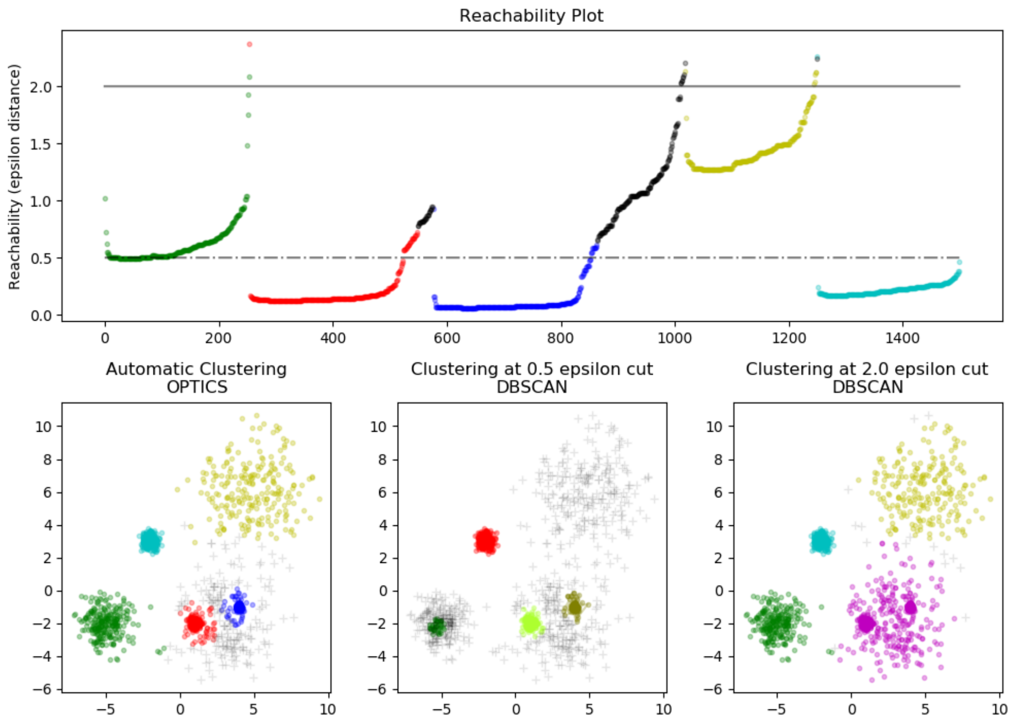


(Source: [8])

OPTICS – Ordering Points To Identify the Clustering Structure

通过可达性（reachability）定义点与点之间的关系，而非依赖单一固定的密度阈值
可达性度量了点之间的聚类紧密性

- Parametrization less dependent on different cluster densities by defining reachability
- Search for valleys in histogram:
  - Returns cluster
  - Depth of valley represents density
  - Creation of subclusters also possible
- Cluster adjustable depending on ε as upper bound of reachability



Source: https://scikit-learn.org/stable/auto_examples/cluster/plot_optics.html#sphx-glr-auto-examples-cluster-plot-optics-py

Outline

- Motivation
- Clustering
- Dimensionality Reduction / Feature Extraction
- Outlook: Advanced Methods

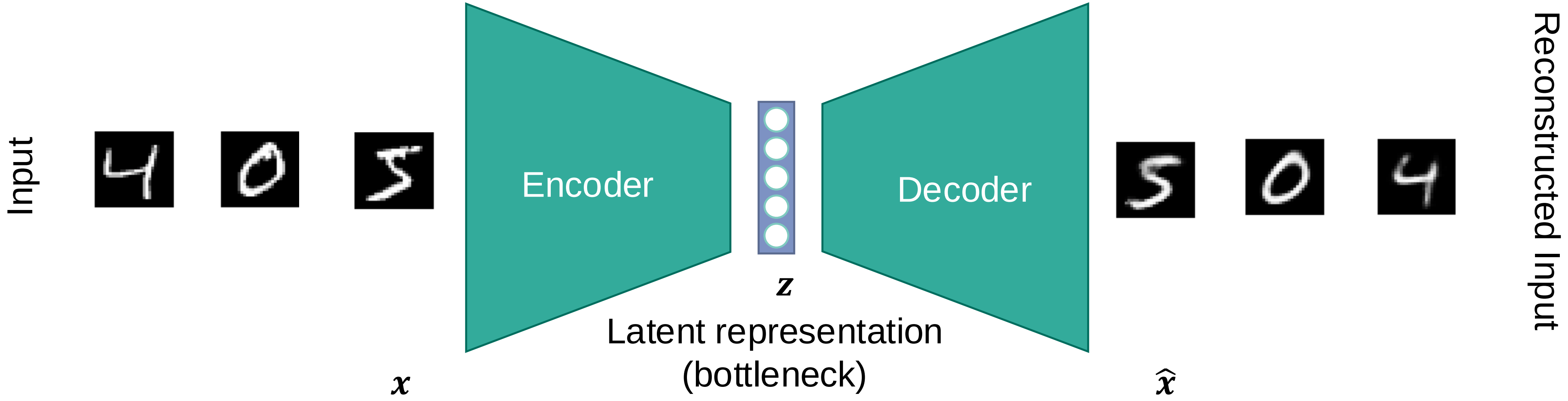
Unsupervised Learning with Neural Networks

- **Principle**: Network learns to reconstruct its input
  - with network architecture creating artificial restrictions 添加人为限制
- Network needs to learn features from dataset

- **Examples**:
  - Autoencoder
  - Restricted Boltzmann Machine
  - Deep Belief Networks

Autoencoder

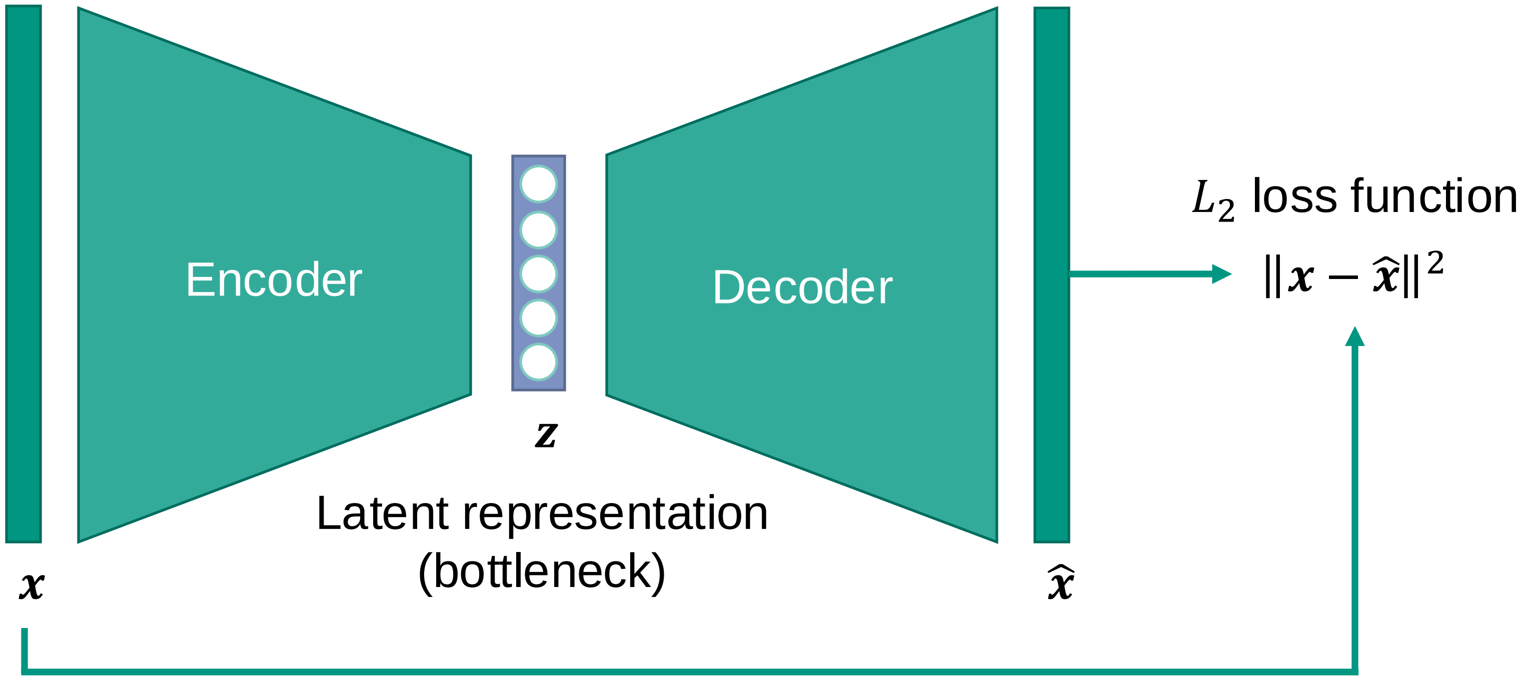
- Architecture containing Encoder and Decoder
- Tunneling of information through a bottleneck (latent representation)



- Encoder and Decoder usually deep neural networks (CNNs)

Autoencoder - Training

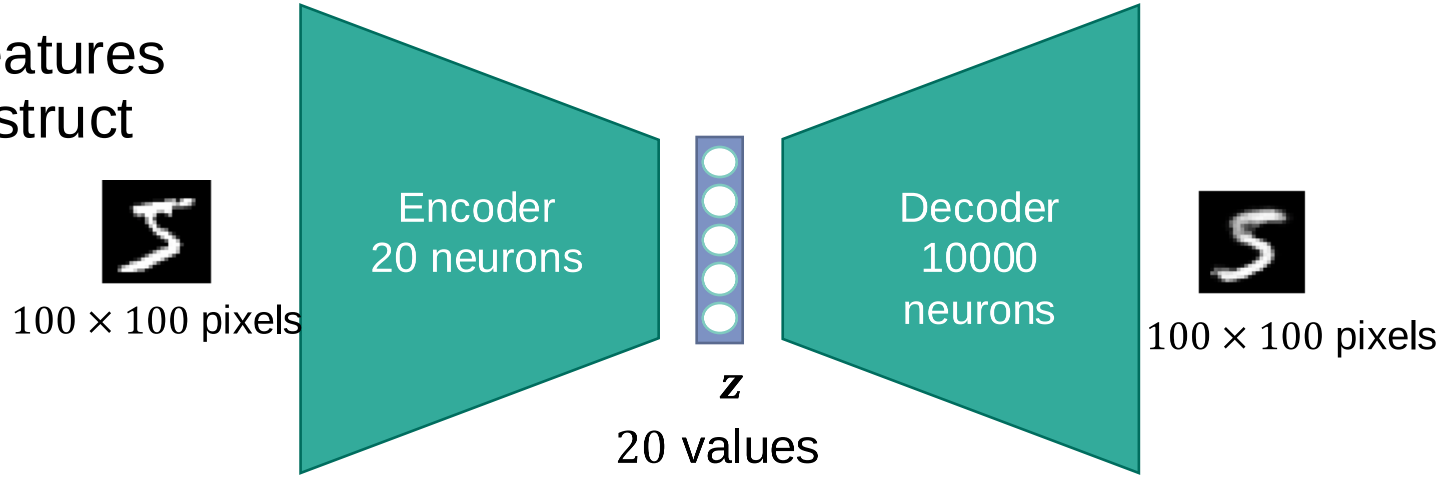
- No extra label required. Label is identical to input and is used in reconstruction loss $||x - \hat{x}||^2$



- Network must compress all relevant information in small latent vector to to reproduce the input

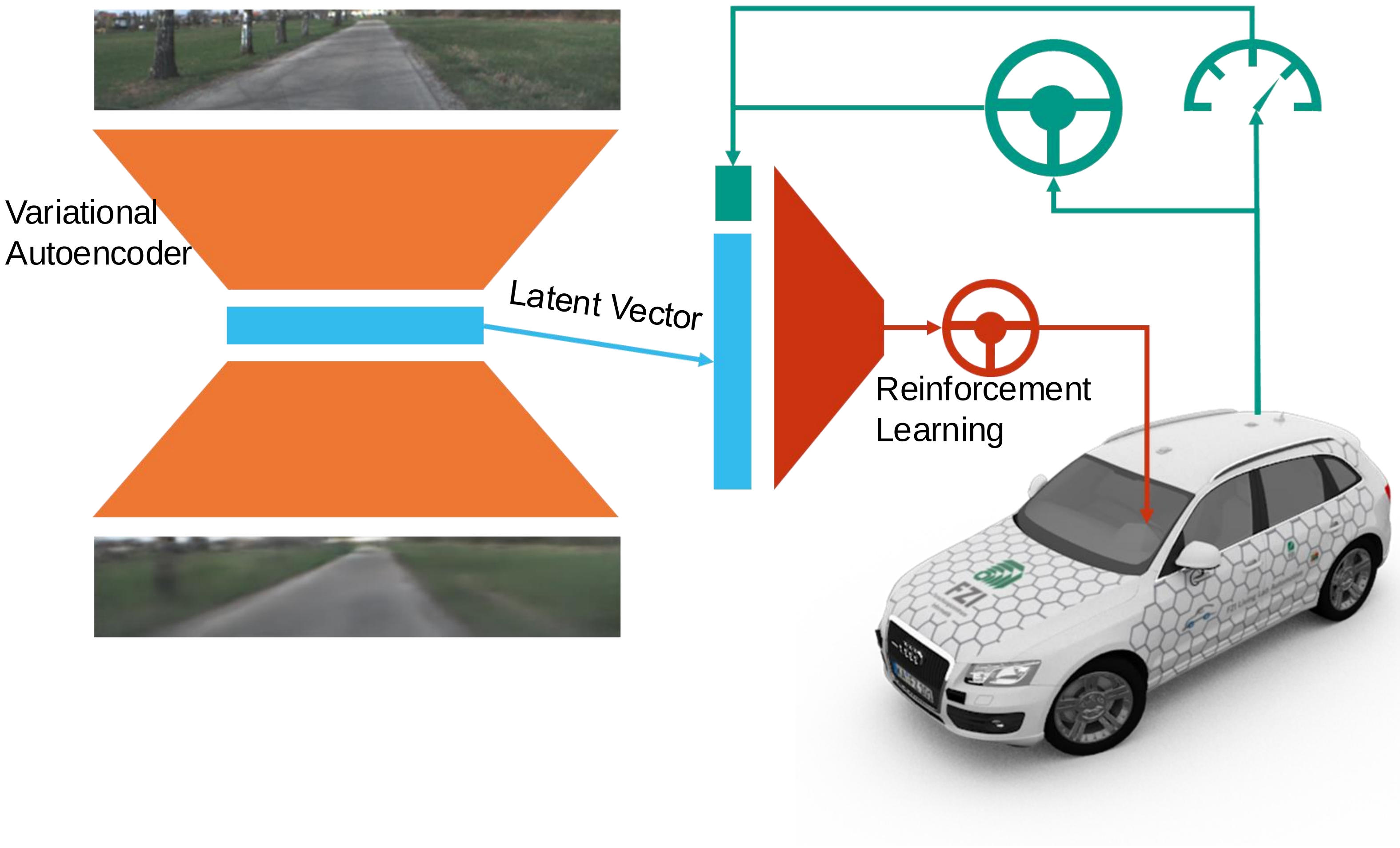
Autoencoder

- **Example:**
  - 20-dimensional latent space
  - 10,000 pixels must to be represented in 20 values
    - (lossy compression)
  - Network stores important features in the latent space to reconstruct the input



Application: Learn latent representation for Reinforcement Learning

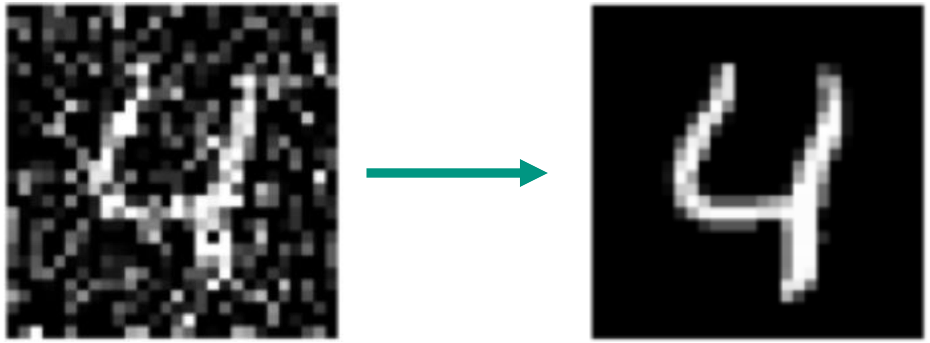
- Control with Reinforcement Learning on latent space
- Latent space by Variational Autoencoder (VAE) see ML2
- VAE trained offline with real video data
- Latent vector extended with data from CAN- Bus
- Agent converts within less than 1000 steps in real rollouts




Total Training Time: 04:36.2
Total Training Frames: 5924
Total Training Episodes: 54

Discussion Autoencoder

- Extension: De-noising Autoencoder
  - Add additional noise to input of encoder
  - Reduces overfitting.



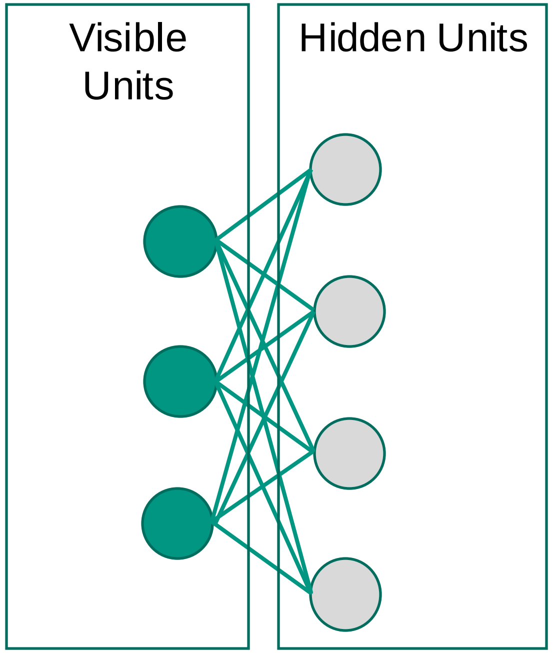
- Extension: Generative Models
  - Use Decoder to create new unseen images by sampling in latent space (and converting the latent representation into a distribution, Variational Autoencoder, see lecture ML2)

- Distinction between unsupervised and supervised learning blurry for neural networks
  - Unsupervised: does not require extra label annotation
  - Supervised: uses input as label
  - Nowadays also called: **Self-Supervised Learning**
    - (Fundamental learning method used to create today's largest modern neural networks such as ChatGPT/GPT-4, see lecture in ML2)

Restricted Boltzmann Machine (RBM) [Smolensky, 1986]

- Neurons form bipartite graph 二分图
- Two layers: Visible and hidden units
- Neurons take binary values (activated / not activated)
- Network propagation is bidirectional and establishes equilibrium
  - propagated value generally defines the probability of activation 双向传播，传播的值通常定义了激活的概率。
- Weights on edges:
  - Propagate between layers forward and backward until an equilibrium between hidden and visible layers is reached for learning data 平衡
- Example: Collaborative Filtering (Netflix) [6]

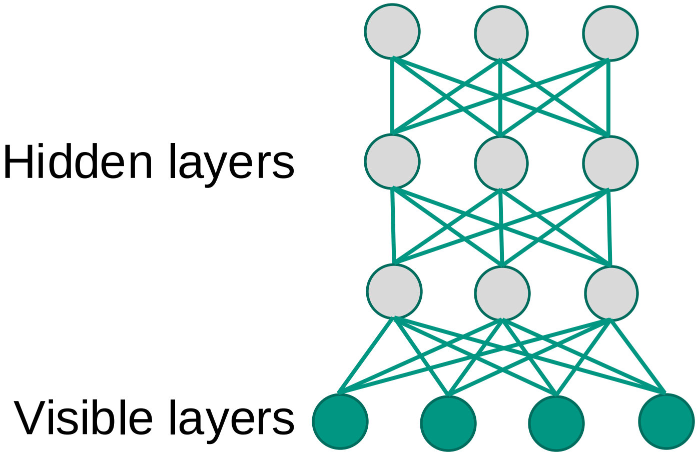
应用：商城推荐算法

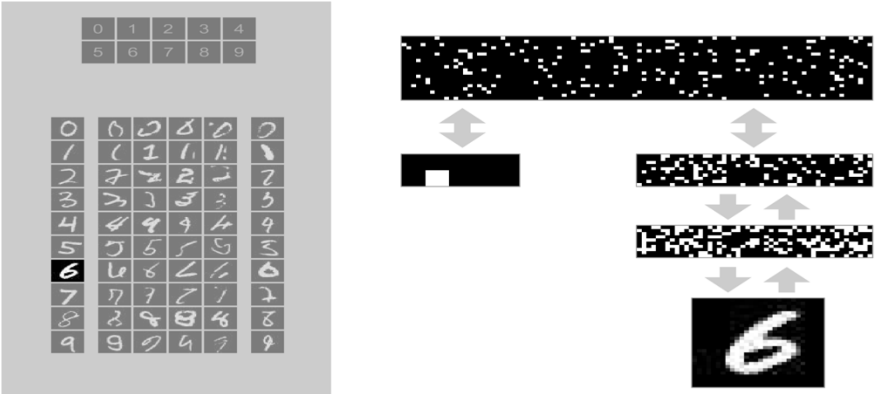


Deep Belief Networks

- Generative graphical model 带有隐藏单元的双向层聚合
- Aggregation of bidirectional layers with hidden units
- Approximation with stacked RBM 堆叠的限制玻尔兹曼机（RBM
- Layers can be trained sequentially
  → one of the first effective approaches of deep learning with deep networks
- Applications
  - Feature extraction
  - Clustering
  - Classification
  - Also, as stacked Autoencoder

DBN的层可以逐层进行无监督的预训练，然后通过有监督学习进行微调。





[Geoffry Hinton]

Outline

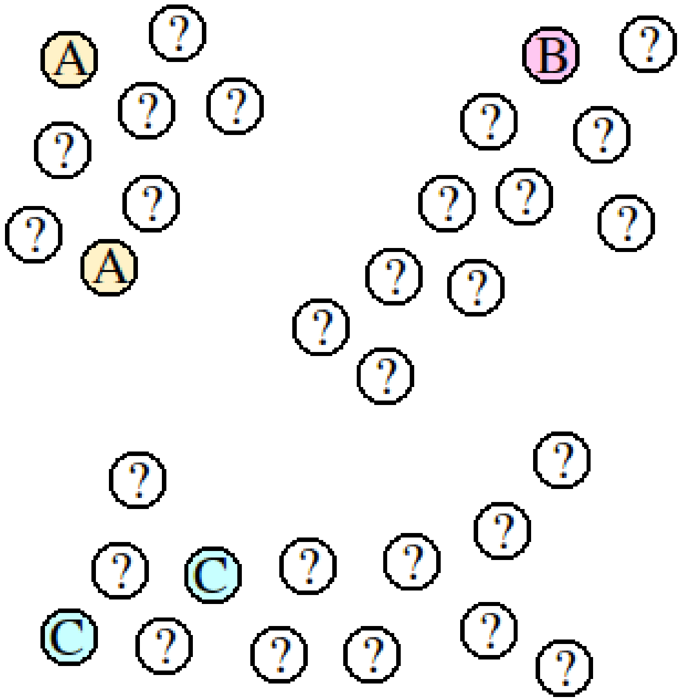
- Motivation
- Clustering
- Dimensionality Reduction / Feature Extraction
- Outlook: Advanced Methods

Outlook: Advanced Methods

- **Constrained k-mean Clustering** [3]
  - additional *must-link-* and *cannot-link*-constraints

- **Semi supervised learning** [4,5]
  - Learning method when the dataset contains little labelled data and a lot of unlabeled data
  - Example:
    - Unsupervised clustering
    - Use labeled data instances to label clusters

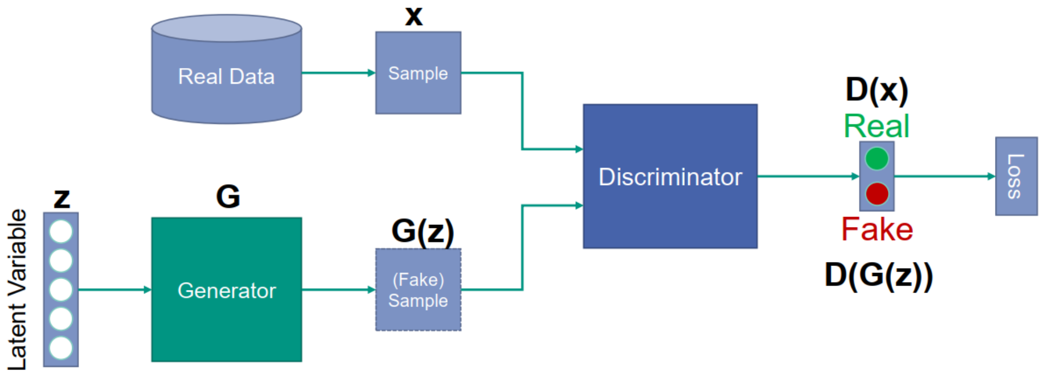
- **Self-Supervised Learning (ML2)**



Outlook: Advanced Methods (in ML2)

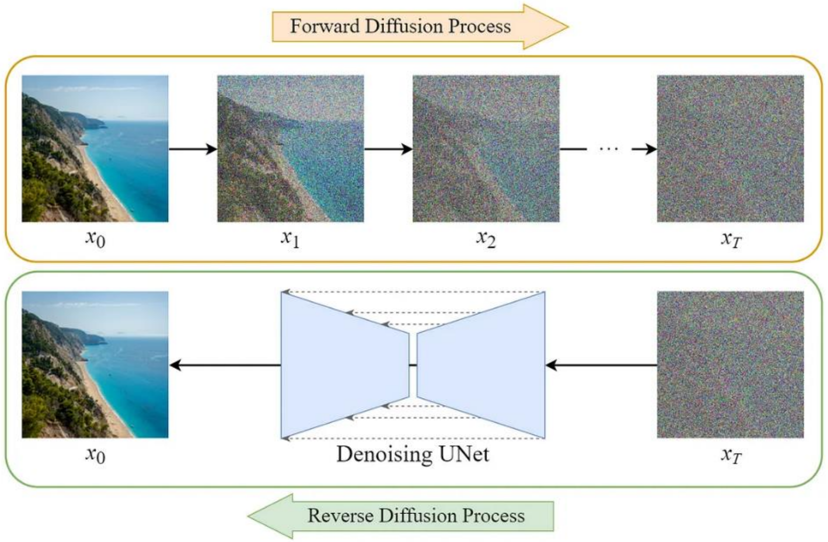
Generative Adversarial Networks

- Simultaneously train generative and discriminative model that try to “fool” each other



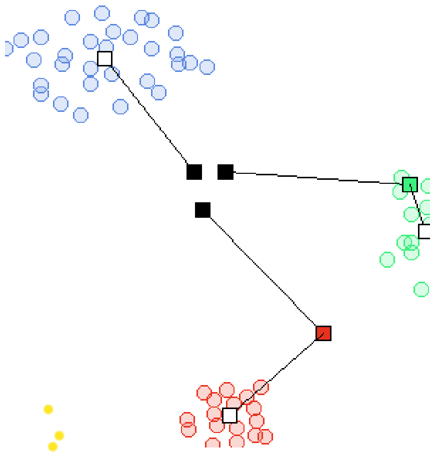
Diffusion Models

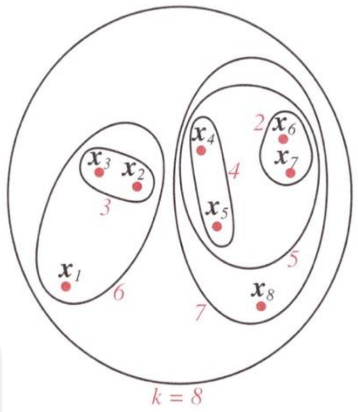
- Apply noise on image and train denoising network
  - Similar to denoising autoencoder using multiple steps)
  - Used in Dalle-2 / Stable Diffusion

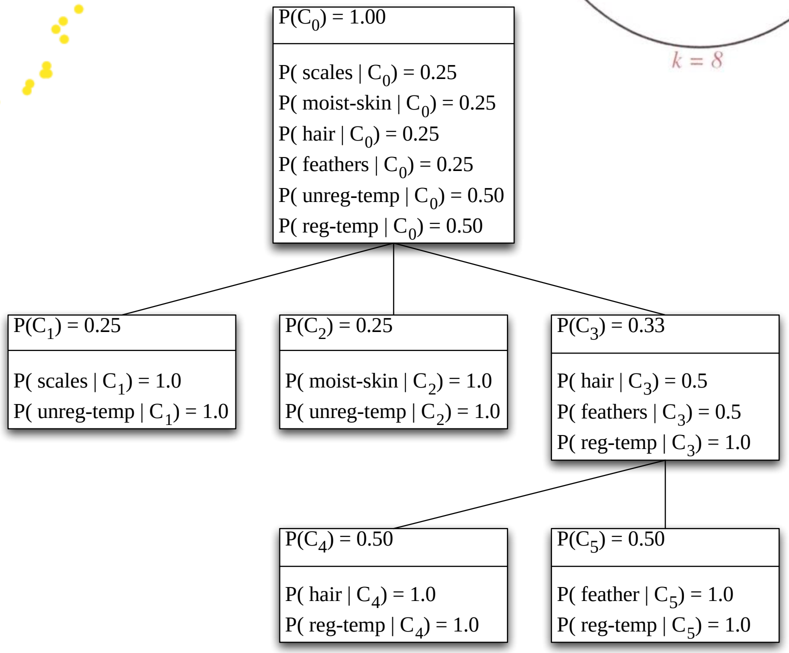


Conclusion

- Different approaches / optimizations
  - k-means clustering
  - DBSCAN / OPTICS
  - Autoencoder
  - Generative models
  - ....
- Different objectives
- Commonly used in combination with other ML algorithms







Literature

- [1] *Duda, Hart, Stork*: **Pattern Classification**. John Wiley & Sons, 2001, Kapitel 10.
- [2] *Gennari, Langley, Fisher*: **Models of incremental concept formation**. Artificial Intelligence, vol. 40, pp.11-61, 1989.
- [3] *Wagstaff, Cardie, Rogers, Schroedl:* **Constrained K-means Clustering with Background Knowledge**. Proceeding of the 8th Int. Conference on Machine Learning, pp. 577-584, 2001.
- [4] *Vapnik:* **Statistical learning theory**. Wiley, pp. 339-371, 1998.
- [5] ML2 → Sommersemester
- [6] Salakhutdinov, Ruslan & Mnih et. al. **Restricted Boltzmann machines for collaborative filtering**. ACM International Conference Proceeding, 2007
- [7] M. Ester, H-P. Kriegel, J. Sander, X. Xu: **A density-based algorithm for discovering clusters in large spatial databases with noise**. Proceedings Int. Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press, 1996
- [8] M. Ankerst, M. M. Breunig, H-P. Kriegel, J. Sander: **OPTICS: Ordering Points To Identify the Clustering Structure**. In: ACM SIGMOD Int. Conference on Management of data. ACM Press, 1999