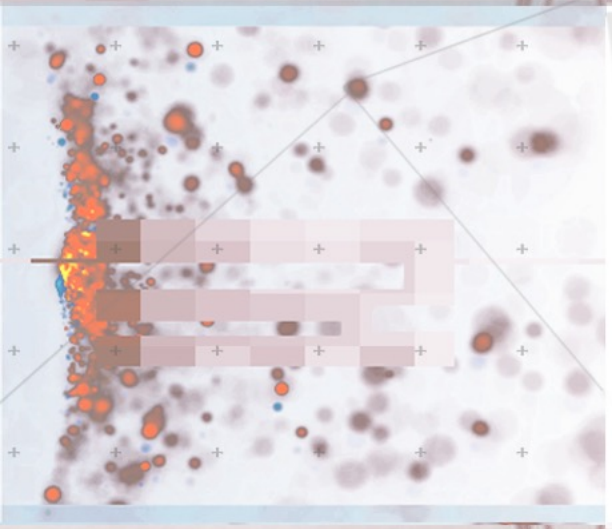
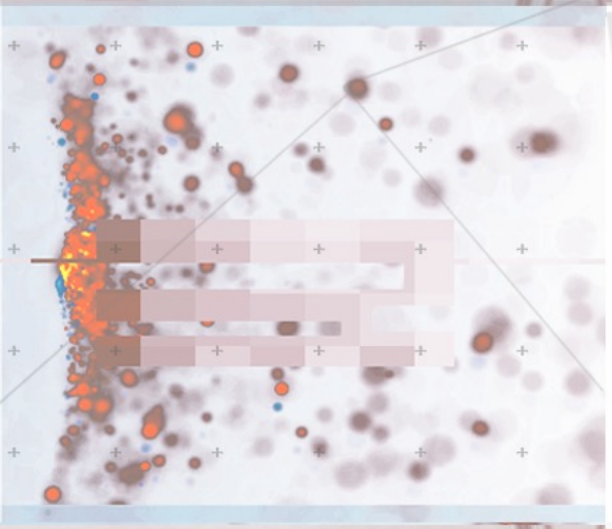


# Density-Based and Grid-Based Clustering Methods

# Density-Based and Grid-Based Clustering Methods

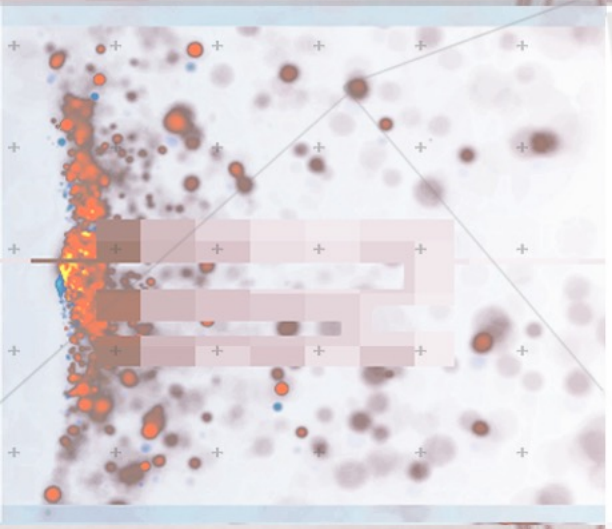
- Density-Based Clustering
  - Basic Concepts
  - DBSCAN: A Density-Based Clustering Algorithm
  - OPTICS: Ordering Points To Identify Clustering Structure
- Grid-Based Clustering Methods
  - Basic Concepts
  - STING: A Statistical Information Grid Approach
  - CLIQUE: Grid-Based Subspace Clustering



# Session 1: Basic Concepts of Density-Based Clustering

# Density-Based Clustering Methods

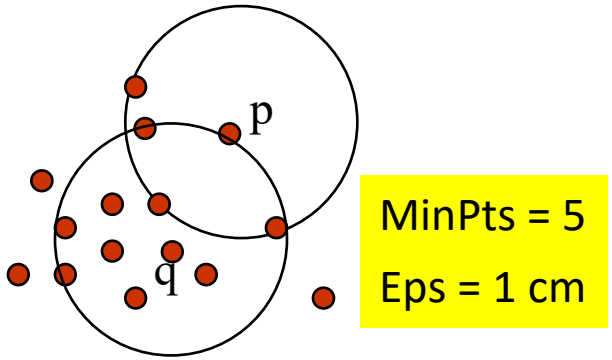
- Clustering based on density (a local cluster criterion), such as density-connected points
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan (only examine the local region to justify density)
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96) — To be covered in this lecture
  - OPTICS: Ankerst, et al (SIGMOD'99) — To be covered in this lecture
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (also, grid-based) — To be covered in this lecture

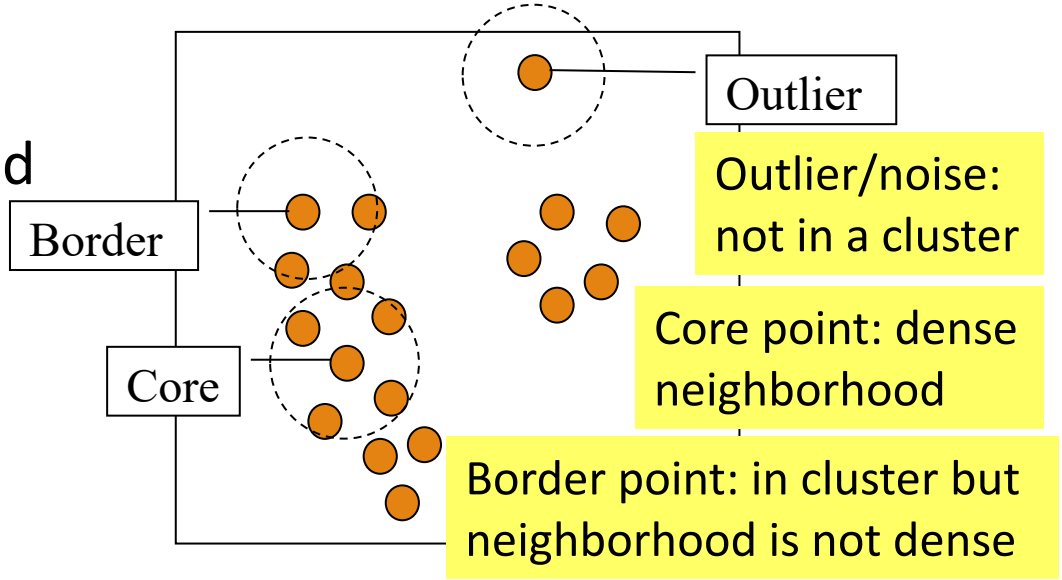


# Session 2: DBSCAN: A Density-Based Clustering Algorithm

# DBSCAN: A Density-Based Spatial Clustering Algorithm

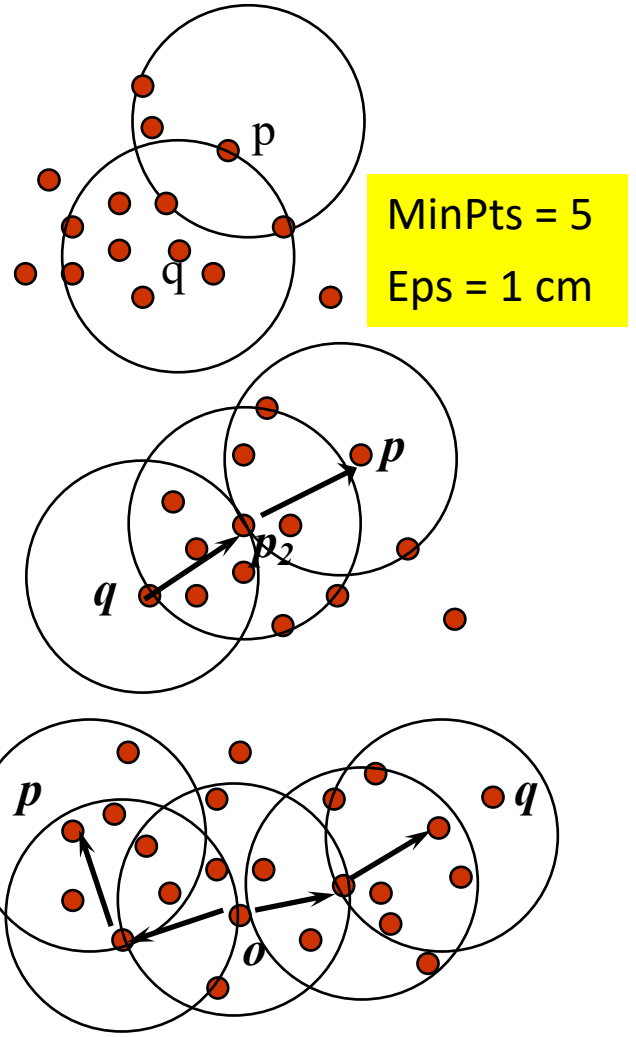
- DBSCAN (M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, KDD'96)
  - Discovers clusters of arbitrary shape: Density-Based Spatial Clustering of Applications with Noise
- A *density-based* notion of cluster
  - A *cluster* is defined as a maximal set of density-connected points
- Two parameters:
  - *Eps* (ε): Maximum radius of the neighborhood
  - *MinPts*: Minimum number of points in the Eps-neighborhood of a point
- The Eps(ε)-neighborhood of a point *q*:
  - $N_{Eps}(q)$: {p belongs to D | dist(p, q) ≤ Eps}



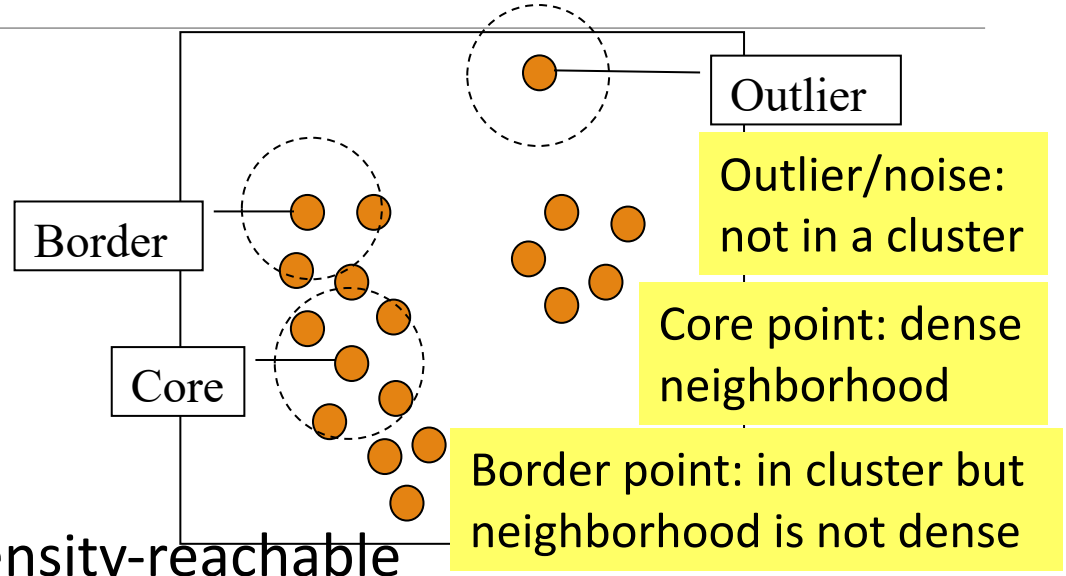
# DBSCAN: Density-Reachable and Density-Connected

- **Directly density-reachable**:
  - A point *p* is directly density-reachable from a point *q* w.r.t. *Eps* (ε), *MinPts* if
    - *p* belongs to $N_{Eps}(q)$
    - **core point** condition: $|N_{Eps}(q)| \geq MinPts$
- **Density-reachable**:
  - A point *p* is density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if there is a chain of points $p_1, \ldots, p_n, p_1 = q, p_n = p$ such that $p_{i+1}$ is directly density-reachable from $p_i$
- **Density-connected:**
  - A point *p* is density-connected to a point *q* w.r.t. *Eps*, *MinPts* if there is a point *o* such that both *p* and *q* are density-reachable from *o* w.r.t. *Eps* and *MinPts*



# DBSCAN: The Algorithm

- **Algorithm**
  - Arbitrarily select a point *p*
  - Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*
    - If *p* is a core point, a cluster is formed
    - If *p* is a border point, no points are directly density-reachable from *p,* and DBSCAN visits the next point of the database
  - Continue the process until all of the points have been processed
- **Computational complexity**
  - If a spatial index is used, the computational complexity of DBSCAN is O(nlogn), where n is the number of database objects
  - Otherwise, the complexity is $O(n^2)$



# DBSCAN Is Sensitive to the Setting of Parameters

*Figure 8. DBScan results for DS1 with MinPts at 4 and Eps at (a) 0.5 and (b) 0.4.*

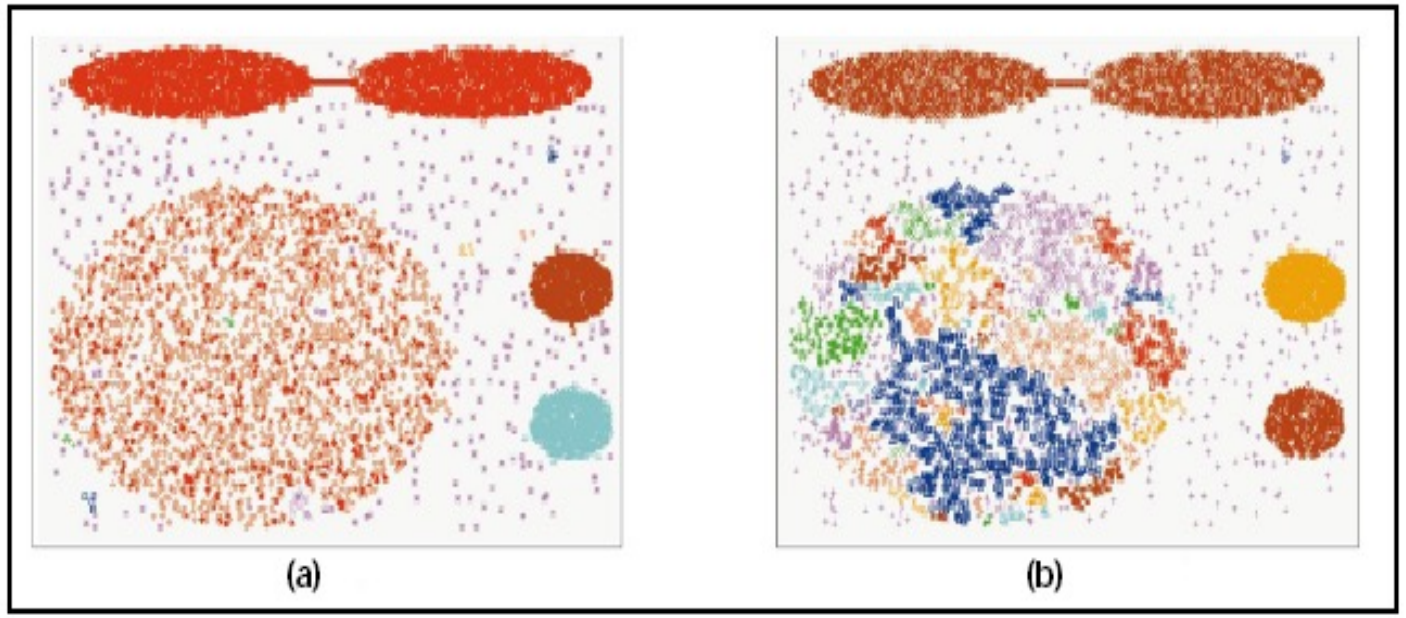
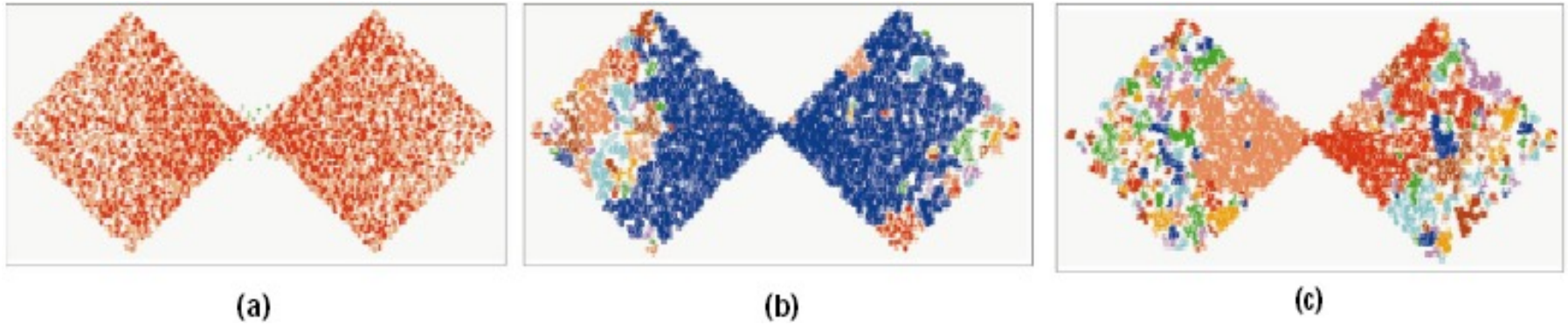
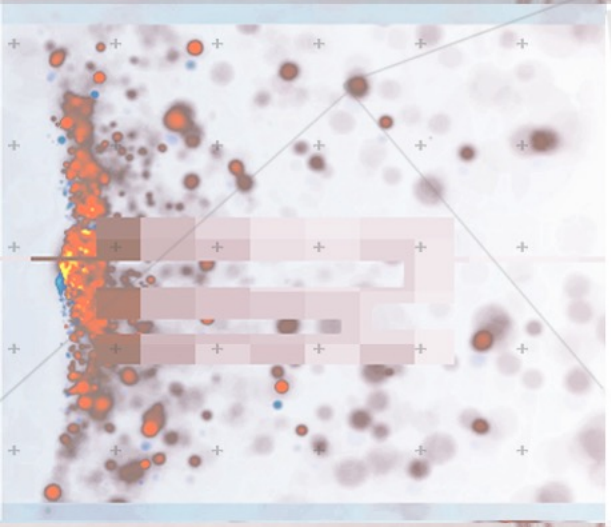



*Figure 9. DBScan results for DS2 with MinPts at 4 and Eps at (a) 5.0, (b) 3.5, and (c) 3.0.*




Ack. Figures from G. Karypis, E.-H. Han, and V. Kumar, *COMPUTER*, 32(8), 1999

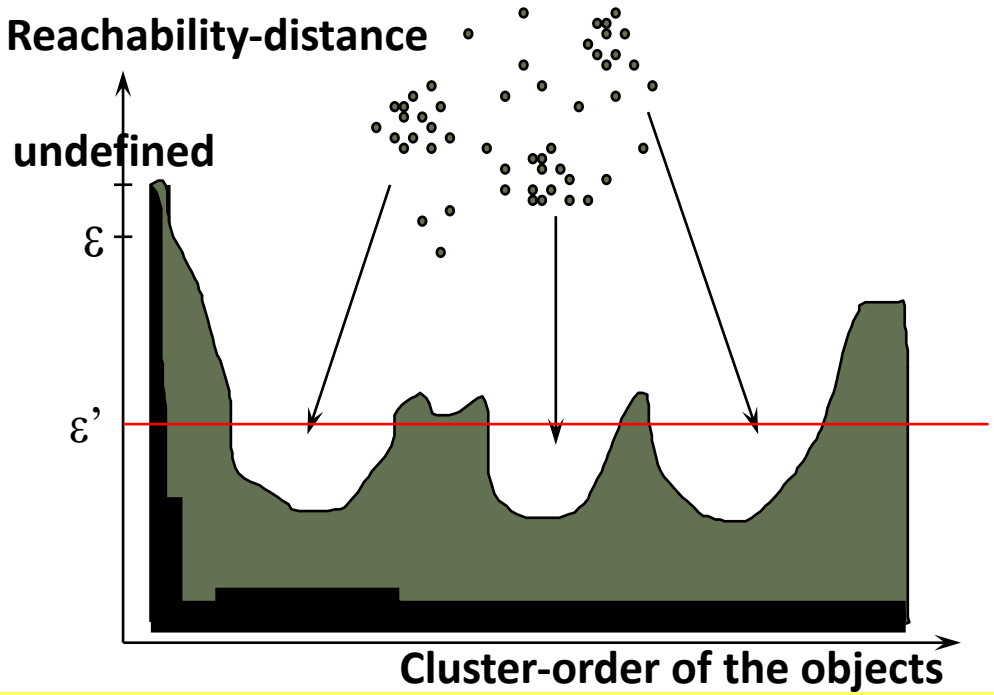


# Session 3: OPTICS: Ordering Points To Identify Clustering Structure

# OPTICS: Ordering Points To Identify Clustering Structure

- OPTICS (Ankerst, Breunig, Kriegel, and Sander, SIGMOD'99)
  - DBSCAN is sensitive to parameter setting
  - An extension: finding clustering structure
- Observation: Given a *MinPts*, density-based clusters w.r.t. a higher density are completely contained in clusters w.r.t. to a lower density
- Idea: Higher density points should be processed first—find high-density clusters first
- OPTICS stores such a clustering order using two pieces of information:
  - *Core distance* and *reachability distance*

Reachability plot for a dataset




- Since points belonging to a cluster have a low reachability distance to their nearest neighbor, valleys correspond to clusters
- The deeper the valley, the denser the cluster

# OPTICS: An Extension from DBSCAN

- **Core distance** of an object *p*: The smallest value $\varepsilon$ such that the $\varepsilon$-neighborhood of *p* has at least *MinPts* objects

  Let $N_\varepsilon(p)$: $\varepsilon$-neighborhood of *p*

  $\varepsilon$ is a distance value

  Core-distance$_{\varepsilon, MinPts}(p)$ = Undefined if card($N_\varepsilon(p)$) < *MinPts*

  *MinPts*-distance(p), otherwise

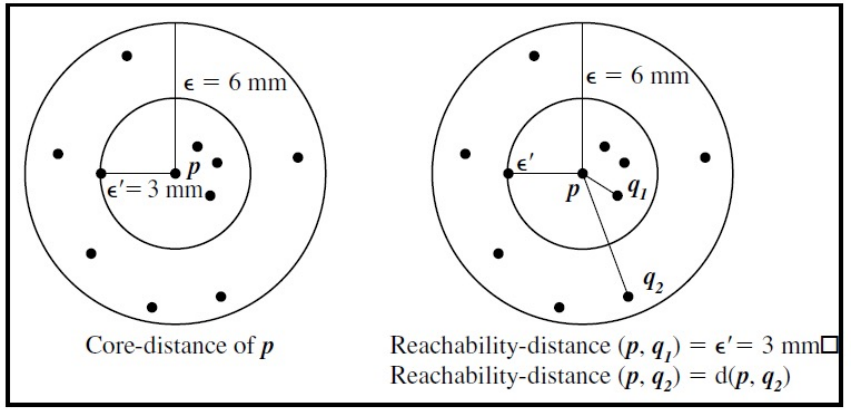


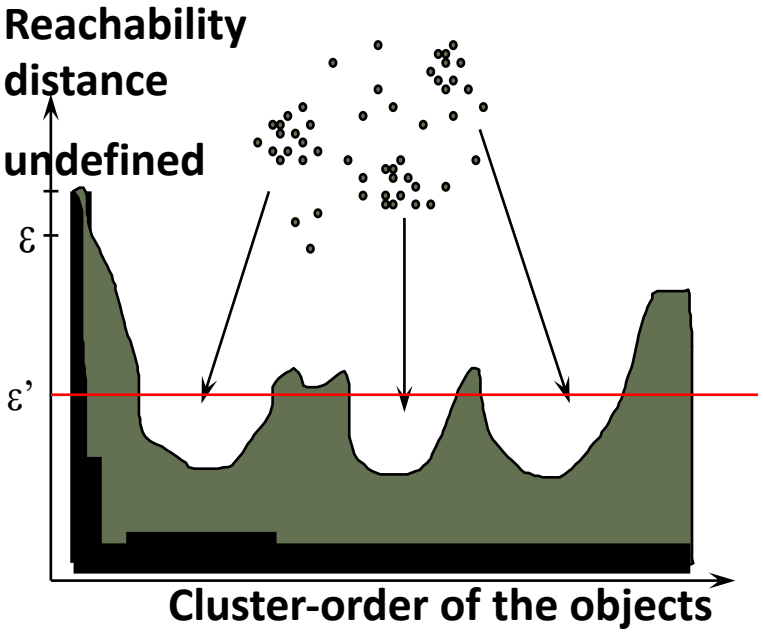

Figure 10.16: OPTICS terminology. Based on [ABKS99].

- **Reachability distance** of object *q* from core object *p* is the min. radius value that makes *q* density-reachable from *p*

  Reachability-distance$_{\varepsilon, MinPts}(p, q)$ =

  Undefined, *if p is not a core object*

  max(core-distance(*p*), distance (*p*, *q*)), *otherwise*

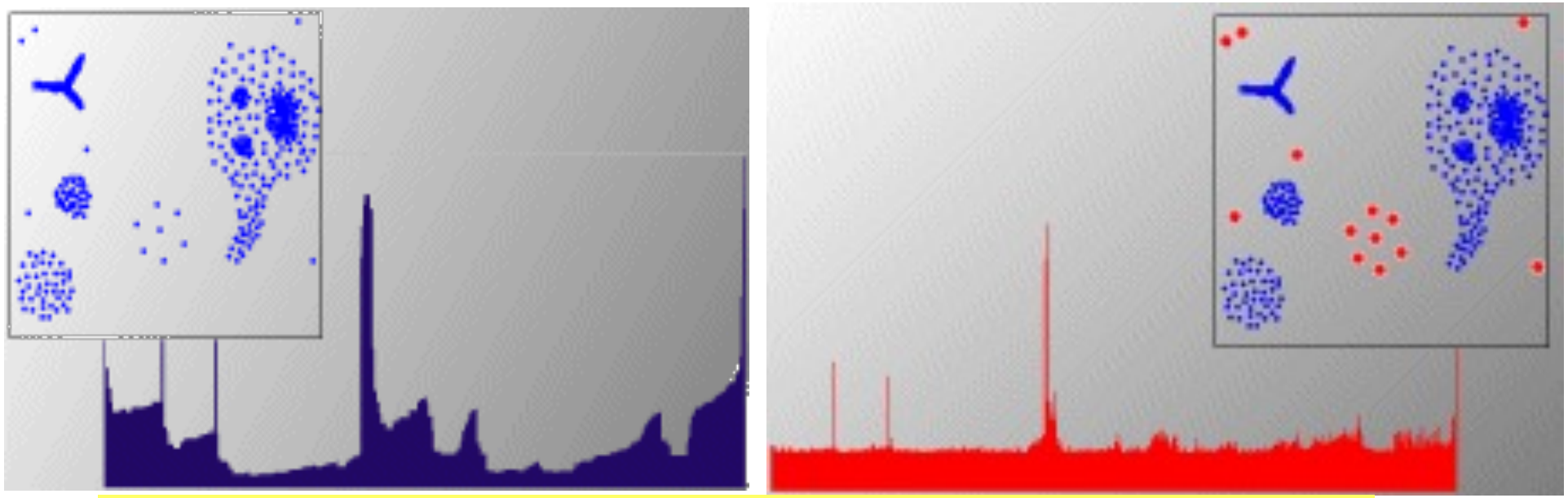



- Complexity: O(*N* log*N*) (if index-based)

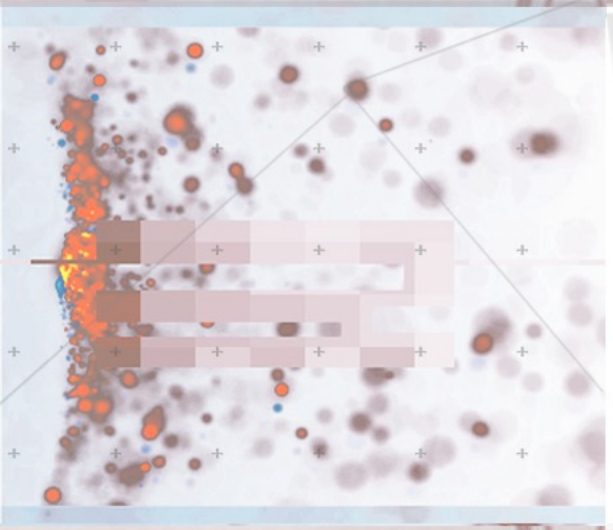
  where *N*: # of points

# OPTICS: Finding Hierarchically Nested Clustering Structures

- OPTICS produces a special cluster-ordering of the data points with respect to its density-based clustering structure
  - The cluster-ordering contains information equivalent to the density-based clusterings corresponding to a broad range of parameter settings
  - Good for both automatic and interactive cluster analysis—finding intrinsic, even hierarchically nested clustering structures



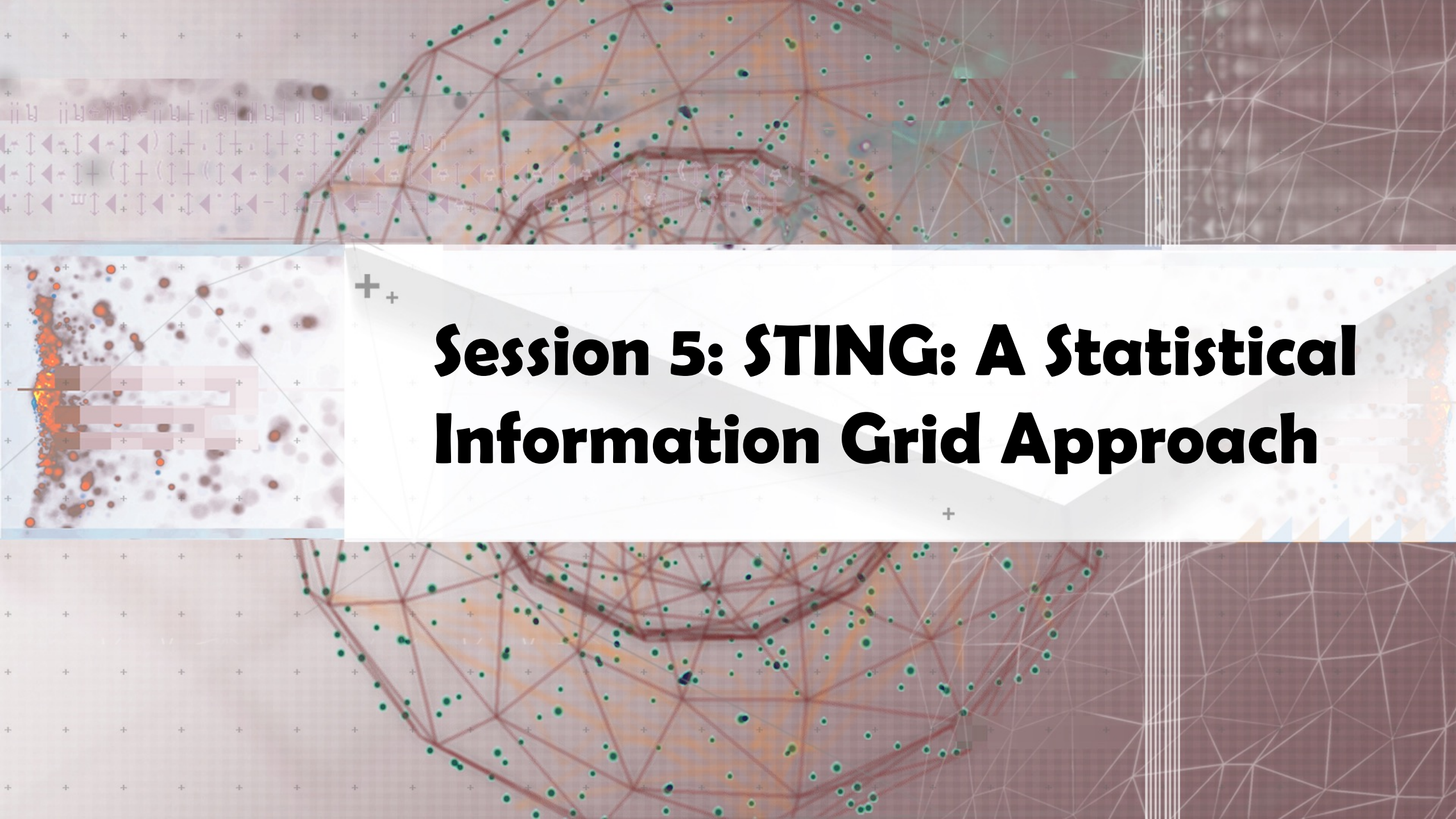
Finding nested clustering structures with different parameter settings



# Session 4: Grid-Based Clustering Methods

# Grid-Based Clustering Methods

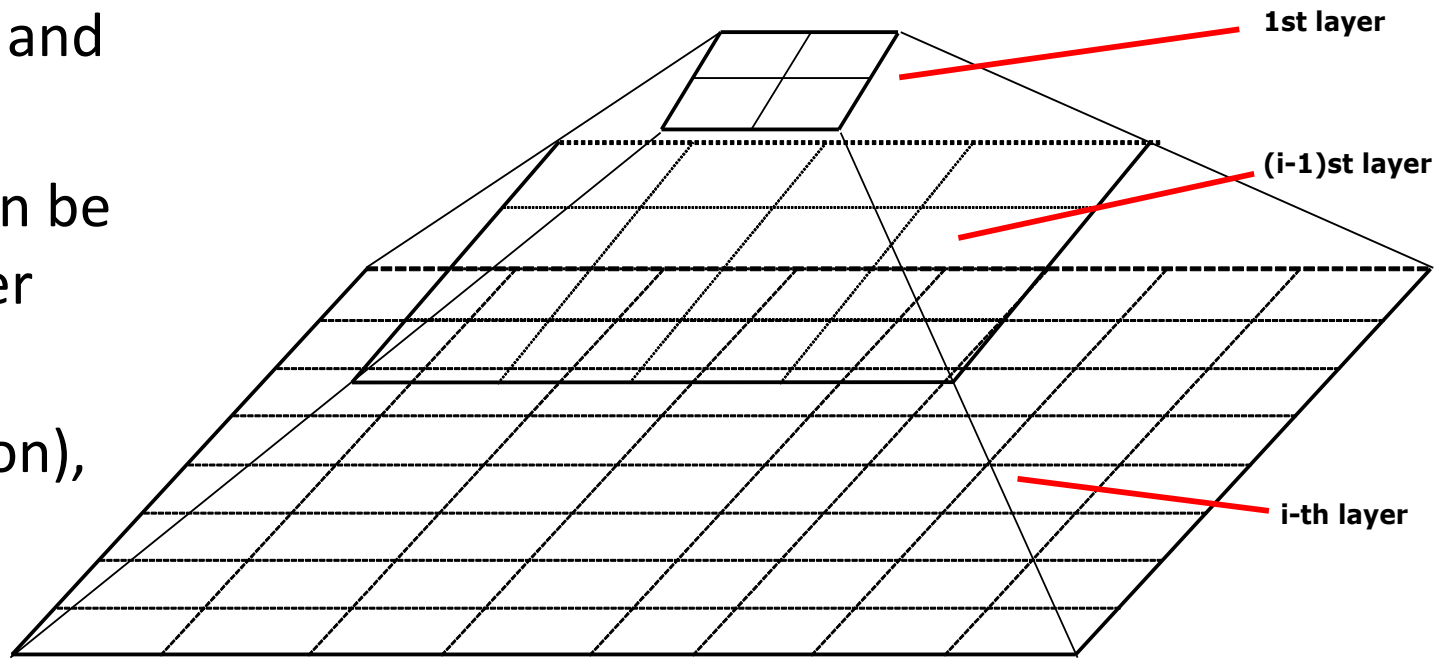
- Grid-Based Clustering: Explore multi-resolution grid data structure in clustering
  - Partition the data space into a finite number of cells to form a grid structure
  - Find clusters (dense regions) from the cells in the grid structure
- Features and challenges of a typical grid-based algorithm
  - Efficiency and scalability: # of cells << # of data points
  - Uniformity: Uniform, hard to handle highly irregular data distributions
  - Locality: Limited by predefined cell sizes, borders, and the density threshold
  - Curse of dimensionality: Hard to cluster high-dimensional data
- Methods to be introduced
  - **STING** (a STatistical INformation Grid approach) (Wang, Yang and Muntz, VLDB'97)
  - **CLIQUE** (Agrawal, Gehrke, Gunopulos, and Raghavan, SIGMOD'98)
    - Both grid-based and subspace clustering



# Session 5: STING: A Statistical Information Grid Approach

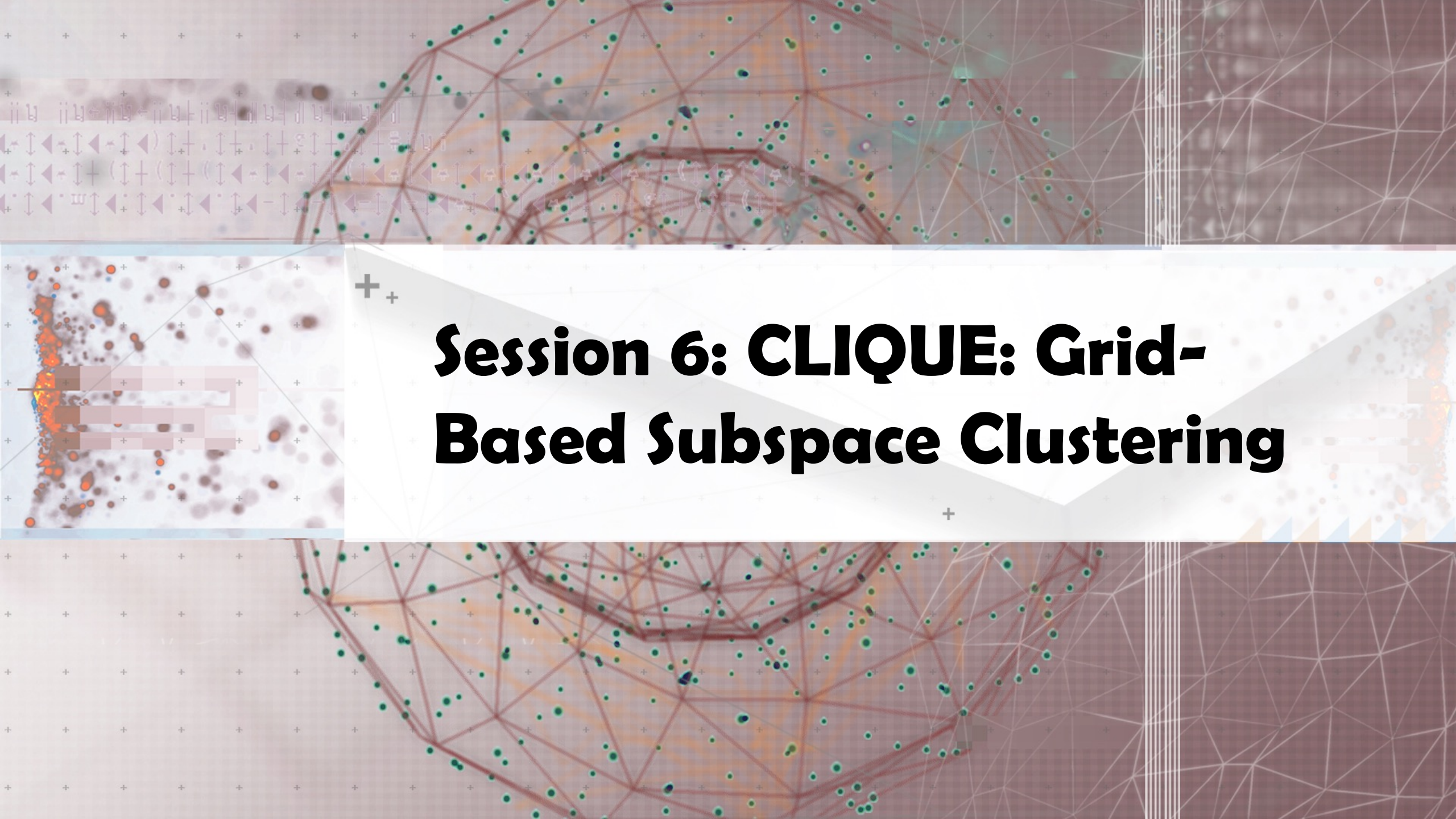
# STING: A Statistical Information Grid Approach

- STING (Statistical Information Grid) (Wang, Yang and Muntz, VLDB'97)
- The spatial area is divided into rectangular cells at different levels of resolution, and these cells form a tree structure
- A cell at a high level contains a number of smaller cells of the next lower level
- Statistical information of each cell is calculated and stored beforehand and is used to answer queries
- Parameters of higher level cells can be easily calculated from that of lower level cell, including
  - *count*, *mean*, *s*(standard deviation), *min*, *max*
  - type of distribution—*normal*, *uniform*, etc.



# Query Processing in STING and Its Analysis

- To process a region query
  - Start at the root and proceed to the next lower level, using the STING index
  - Calculate the likelihood that a cell is relevant to the query at some confidence level using the statistical information of the cell
  - Only children of likely relevant cells are recursively explored
  - Repeat this process until the bottom layer is reached
- Advantages
  - Query-independent, easy to parallelize, incremental update
  - Efficiency: Complexity is O(K)
    - K: # of grid cells at the lowest level, and K << N (i.e., # of data points)
- Disadvantages
  - Its probabilistic nature may imply a loss of accuracy in query processing

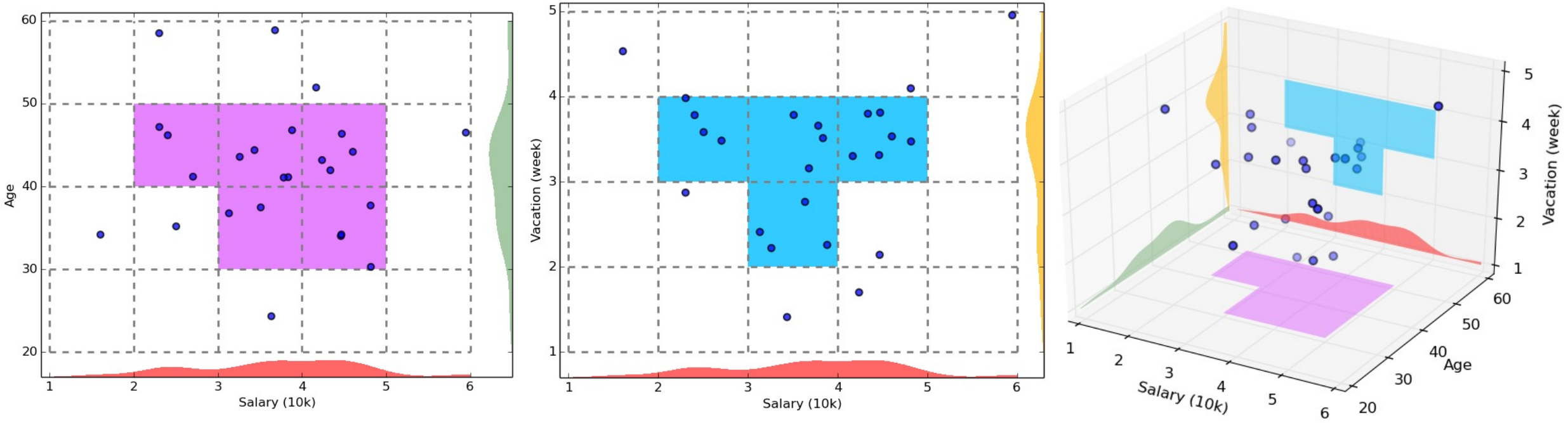


# Session 6: CLIQUE: Grid-Based Subspace Clustering

# CLIQUE: Grid-Based Subspace Clustering

- CLIQUE (Clustering In QUEst) (Agrawal, Gehrke, Gunopulos, Raghavan: SIGMOD'98)
- CLIQUE is a **density-based** and **grid-based** subspace clustering algorithm
  - **Grid-based**: It discretizes the data space through a grid and estimates the density by counting the number of points in a grid cell
  - **Density-based**: A cluster is a maximal set of connected dense units in a subspace
    - A unit is dense if the fraction of total data points contained in the unit exceeds the input model parameter
  - **Subspace clustering**: A subspace cluster is a set of neighboring dense cells in an arbitrary subspace. It also discovers some minimal descriptions of the clusters
- It automatically identifies subspaces of a high dimensional data space that allow better clustering than original space using the Apriori principle

# CLIQUE: SubSpace Clustering with Aprori Pruning



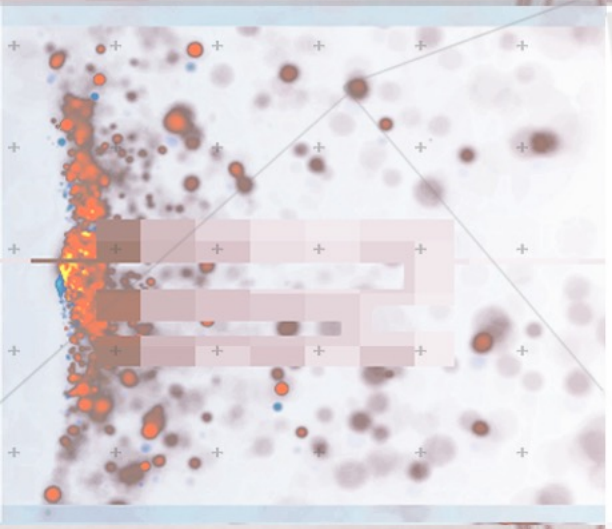

- Start at 1-D space and discretize numerical intervals in each axis into grid
- Find dense regions (clusters) in each subspace and generate their minimal descriptions
- Use the dense regions to find promising candidates in 2-D space based on the Apriori principle
- Repeat the above in level-wise manner in higher dimensional subspaces

# Major Steps of the CLIQUE Algorithm

- Identify subspaces that contain clusters
  - Partition the data space and find the number of points that lie inside each cell of the partition
  - Identify the subspaces that contain clusters using the Apriori principle
- Identify clusters
  - Determine dense units in all subspaces of interests
  - Determine connected dense units in all subspaces of interests
- Generate minimal descriptions for the clusters
  - Determine maximal regions that cover a cluster of connected dense units for each cluster
  - Determine minimal cover for each cluster

# Additional Comments on *CLIQUE*

- Strengths
  - *Automatically* finds subspaces of the highest dimensionality as long as high density clusters exist in those subspaces
  - *Insensitive* to the order of records in input and does not presume some canonical data distribution
  - Scales *linearly* with the size of input and has good scalability as the number of dimensions in the data increases
- Weaknesses
  - As in all grid-based clustering approaches, the quality of the results crucially depends on the appropriate choice of the number and width of the partitions and grid cells



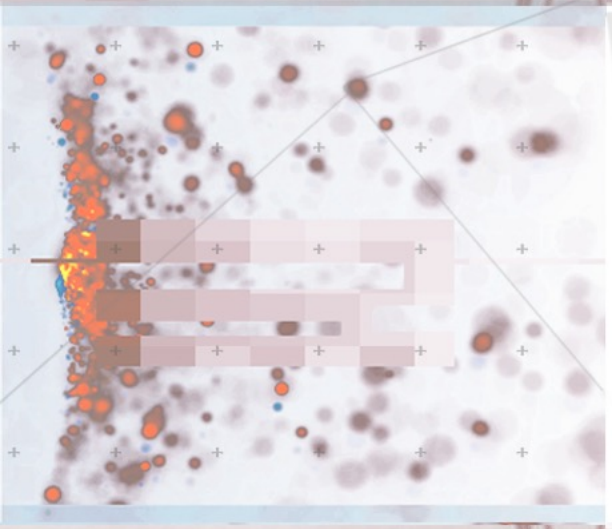
# Summary

# Summary: Density-Based and Grid-Based Clustering Methods

- Density-Based Clustering
  - Basic Concepts
  - DBSCAN: A Density-Based Clustering Algorithm
  - OPTICS: Ordering Points To Identify Clustering Structure
- Grid-Based Clustering Methods
  - Basic Concepts
  - STING: A Statistical Information Grid Approach
  - CLIQUE: Grid-Based Subspace Clustering

# Recommended Readings

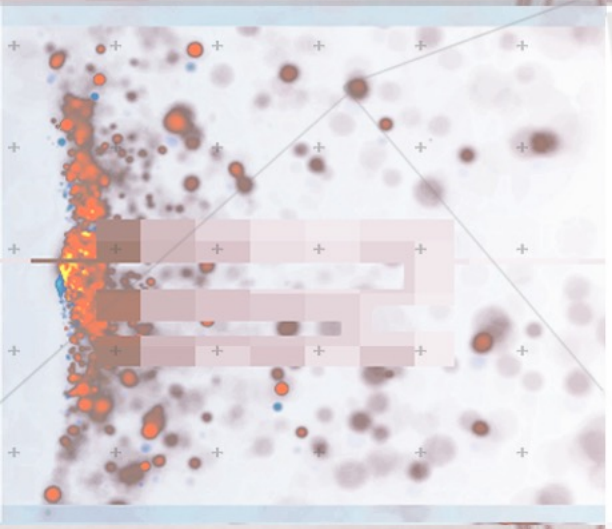
- M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases. KDD'96
- W. Wang, J. Yang, R. Muntz, STING: A Statistical Information Grid Approach to Spatial Data Mining, VLDB'97
- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. SIGMOD'98
- A. Hinneburg and D. A. Keim. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. KDD'98
- M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering Points to Identify the Clustering Structure. SIGMOD'99
- M. Ester. Density-Based Clustering. In (Chapter 5) Aggarwal and Reddy (eds.), Data Clustering: Algorithms and Applications . CRC Press. 2014
- W. Cheng, W. Wang, and S. Batista. Grid-based Clustering. In (Chapter 6) Aggarwal and Reddy (eds.), Data Clustering: Algorithms and Applications. CRC Press. 2014



# Probabilistic Model-Based Clustering Methods

# Probabilistic Model-Based Clustering Methods

- Basic Concepts of Probabilistic Model-Based Clustering
- Mixture Models for Cluster Analysis
- Gaussian Mixture Models
- The Expectation-Maximization (EM) Algorithm (Univariate)
- The Expectation-Maximization (EM) Algorithm (Multivariate)
- Analysis of the Mixture Model Methods



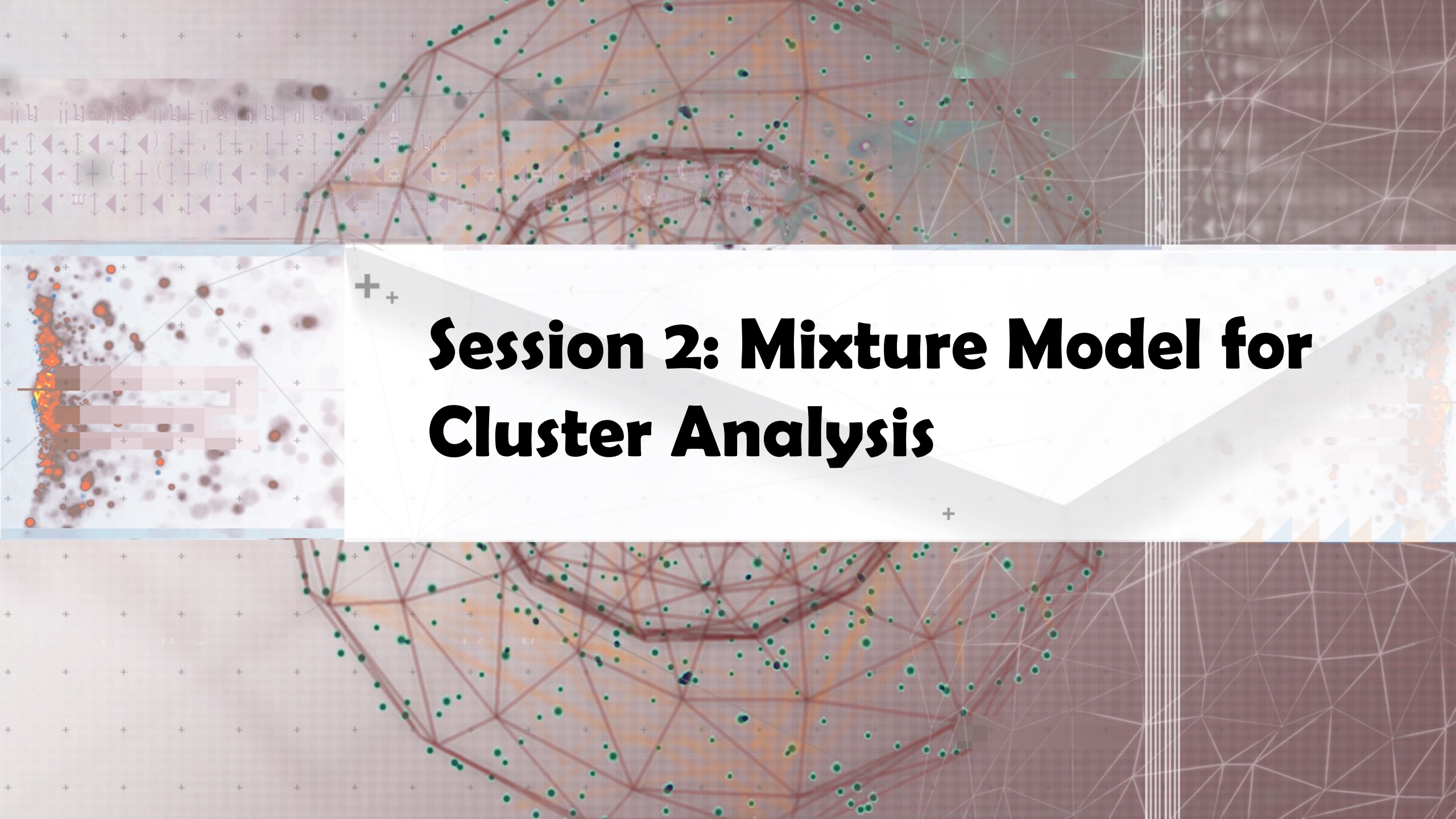
# Session 1: Basic Concepts of Probabilistic Model-Based Clustering

# Probabilistic Model-Based Clustering: Basic Concepts

- Probabilistic model
  - Model the data from a *generative process*
  - Assume the data are generated by a mixture of underlying probability distributions
  - Attempt to optimize the fit between the observed data and some mathematical model using a probabilistic approach
- Probabilistic model-based clustering
  - Each cluster can be represented mathematically by a parametric probability distribution (e.g., Gaussian or Poisson distribution)
  - Cluster: Data points (or objects) that most likely belong to the same distribution
  - Clustering: Parameter estimation so that they will have a *maximum likelihood fit* to the model by a mixture of *K* component distributions (i.e., K clusters)
- Broad applications
  - Image segmentation, document clustering, topic modeling, etc.

# Typical Probabilistic Model-Based Clustering Methods

- **Mixture models**
  - Assume observations to be clustered are drawn from one of several components
  - Infer the parameters of these components (i.e., clusters) and assign data points to specific components of the mixture
- The **Expectation-Maximization (EM)** algorithm
  - A general technique to find maximum likelihood estimations in mixture models
  - The EM algorithm for Gaussian mixture model
- **Probabilistic topic models** for text clustering and analysis (to be covered in the "Text Mining" course)
  - Probabilistic latent semantic analysis (PLSA)
  - Latent Dirichlet allocation (LDA)



# Session 2: Mixture Model for Cluster Analysis

# Model-Based Clustering

- A set $C$ of $k$ probabilistic clusters $C_1, \ldots, C_k$ with probability density functions $f_1, \ldots, f_k$, respectively, and their probabilities $\omega_1, \ldots, \omega_k$
- Probability of an object $o$ generated by cluster $C_j$ is $P(o, C_j) = P(C_j)P(o|C_j) = \omega_j f_j(o)$
- Probability of $o$ generated by the set of cluster $\boldsymbol{C}$ is $P(o \mid \mathbf{C}) = \sum_{j=1}^{k} \omega_j f_j(o)$
- Since objects are assumed to be generated independently, for a data set $D = \{o_1, \ldots, o_n\}$, we have

$$P(D|\boldsymbol{C}) = \prod_{i=1}^{n} P(o_i|\boldsymbol{C}) = \prod_{i=1}^{n} \sum_{j=1}^{k} \omega_j f_j(o_i)$$

- Task: Find a set $C$ of $k$ probabilistic clusters so that $P(D|\boldsymbol{C})$ is maximized
  - Maximizing $P(D|\boldsymbol{C})$ is often intractable since the probability density function of a cluster can take an arbitrarily complicated form
  - To make it computationally feasible (as a compromise), assume the probability density functions are some parameterized distributions

# Parametric Mixed Models

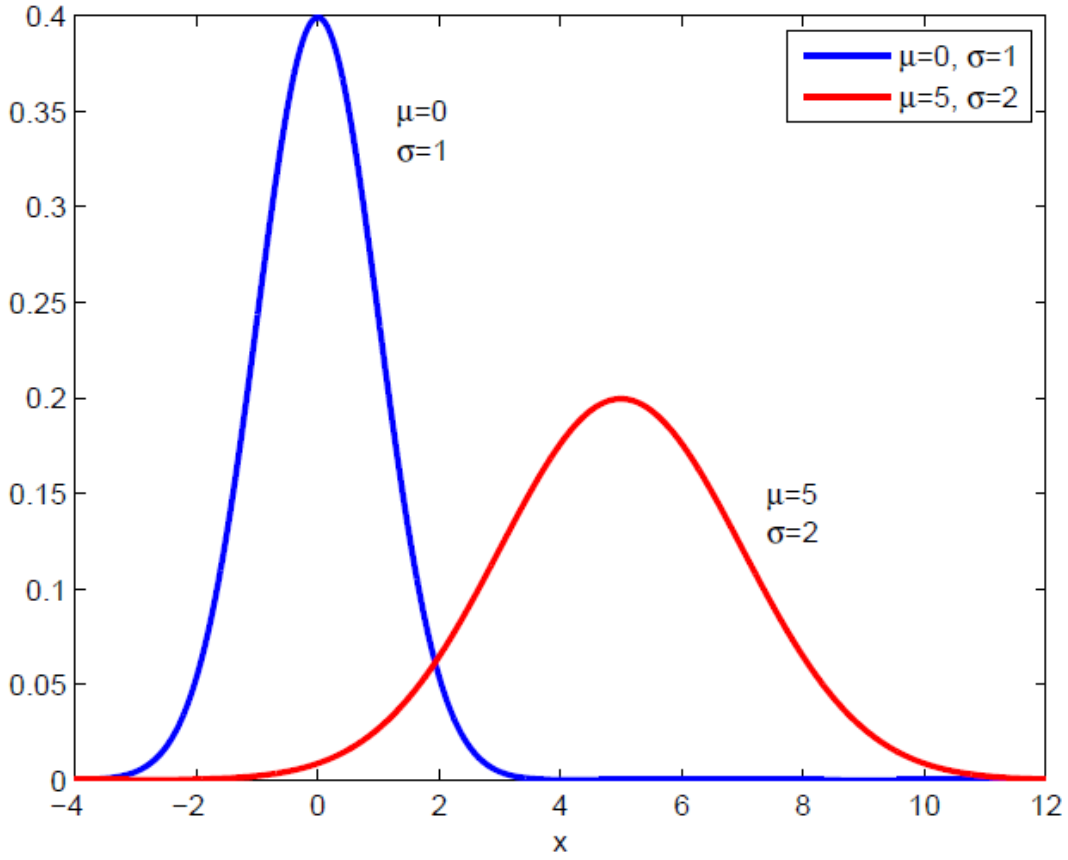
- **Our task** is to infer a set of *K* probabilistic clusters that is mostly likely to generate *D*
  - The values of the discrete latent variables can be interpreted as the assignments of data points to specific components (i.e., clusters) of the mixture
- Each cluster is mathematically represented by a **parametric distribution**
- In principle, the mixtures can be constructed with any types of components, and we could still have a perfectly good mixture model
- In practice, a lot of effort is given over to **parametric mixture models**, where all components are from the same parametric family of distributions but with different parameters
  - Ex. All Gaussians with different means and variances, all Poisson distributions with different means, or all power laws with different exponents
- Two most common mixtures: Mixture of **Gaussian** (continuous) and mixture of **Bernoulli** (discrete) distributions

# Session 3: Gaussian Mixture Models

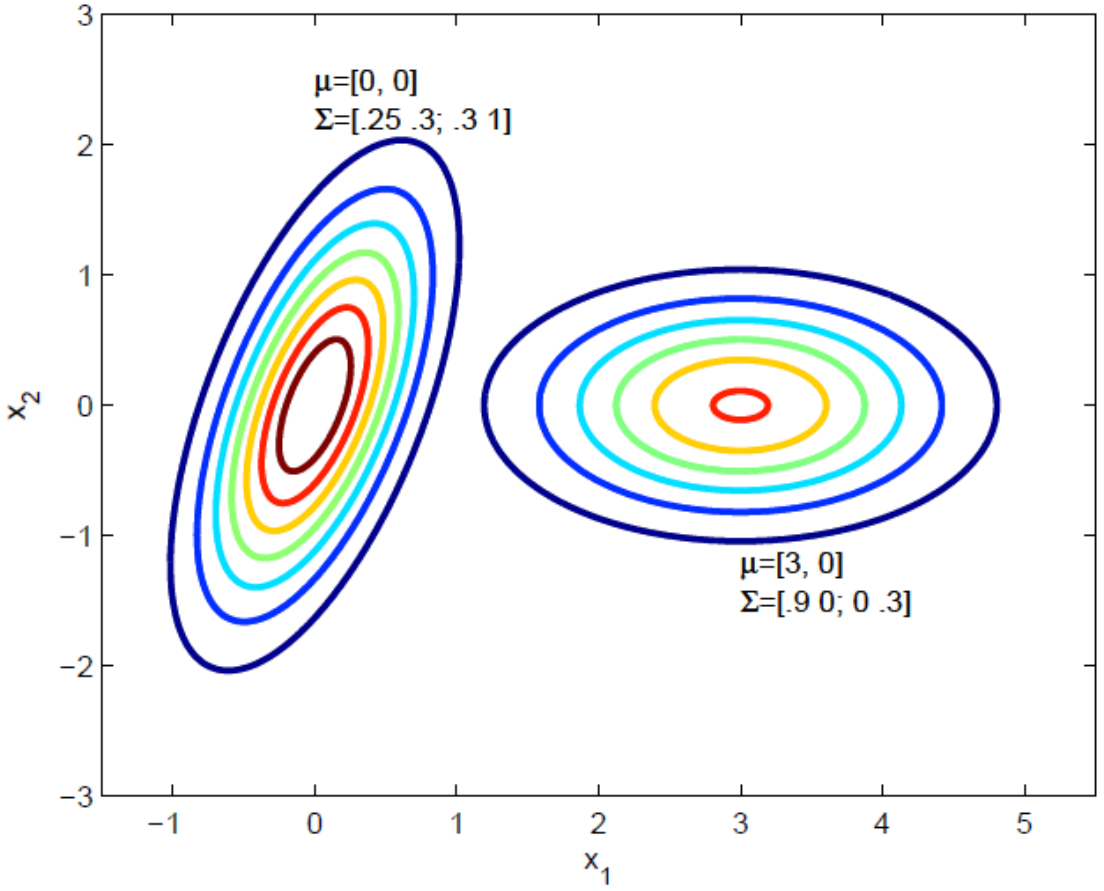
# Univariate and Multivariate Gaussian Distributions

- Plots and contours for Gaussian distributions for various parameters

Plots of the univariate Gaussian distribution for various parameters of $\mu$ and $\sigma$




Contours of the multivariate (2-D) Gaussian distribution for various parameters of $\mu$ and $\Sigma$



# Gaussian Mixture Model

- We assume each cluster $C_i$ is characterized by a multivariate normal distribution

$$f_i(\mathrm{x}) = f(\mathrm{x} \mid \mu_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^d \mid \Sigma_{\mathrm{i}} \mid}} \exp\{-\frac{(\mathrm{x}-\mu_i)^T \Sigma_i^{-1} (\mathrm{x}-\mu_i)}{2}\}$$

  where the cluster mean $\mu_i$ and covariance matrix $\sum_i$ are unknown parameters, and $f_i(x)$ is the probability density $x$ attributable to cluster $C_i$

- We assume the probability density function of ***X*** is given as a Gaussian mixture model over all the $k$ cluster normals defined as

$$f(\mathrm{x}) = \sum_{i=1}^{k} f_i(\mathrm{x}) P(C_i) = \sum_{i=1}^{k} f(\mathrm{x} \mid \mu_{\mathrm{i}}, \Sigma_{\mathrm{i}}) P(C_i)$$

  where the prior probabilities $P(C_i)$ (called mixture parameters) must satisfy

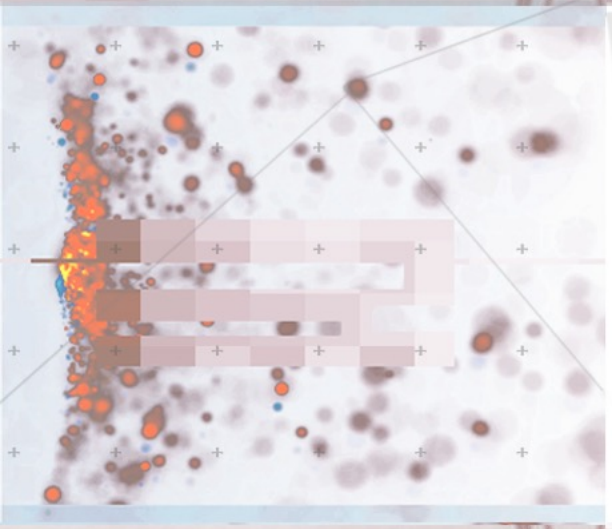
$$\sum_{i=1}^{k} P(C_i) = 1$$

# Maximum Likelihood Estimation of Gaussian Mixture Model

- Maximum Likelihood Estimation (MLE)
  - Given the dataset $D$, the likelihood of the model parameters $\vartheta$ is:

$$P(\mathbf{D}\,|\,\boldsymbol{\theta}) = \prod_{j=1}^{n} f(\mathrm{x}_j) \quad \text{or written as} \quad \ln P(\mathbf{D}\,|\,\boldsymbol{\theta}) = \sum_{j=1}^{n} \ln f(\mathrm{x}_j) = \sum_{j=1}^{n} \ln \sum_{i=1}^{k} f(\mathrm{x}_j|\mu_i,\Sigma_i) P(C_i)$$

  - MLE is to choose parameters $\vartheta$: $\theta^* = \arg\max_{\theta}\{P(D\,|\,\theta)\}$
    - or maximize the log-likelihood: $\theta^* = \arg\max_{\theta}\{\ln P(D\,|\,\theta)\}$
- Directly maximizing the log-likelihood over $\vartheta$ is hard
- We can use EM approach for finding the maximum likelihood estimation for the parameters $\vartheta$
- **Expectation step**: Given current estimates for $\vartheta$, compute the cluster posterior probability $P(C_i|x_j)$ via Bayes theorem: $P(C_i \mid x_j) = \dfrac{f_i(x_j)\cdot P(C_i)}{\sum_{a=1}^{k} f_a(x_j)\cdot P(C_a)}$
- **Maximization step:**
  - Using weight $P(C_i|x_j)$ re-estimate $\vartheta$, i.e., re-estimate $\mu_i$, $\sum_i$ and $P(C_i)$ for each cluster $C_i$



# Session 4: The Expectation-Maximization (EM) Algorithm (Univariate)

# The Expectation-Maximization Framework for K-Means and EM

- The *k*-means algorithm has two steps at each iteration
  - **Expectation Step** (E-step): Given the current cluster centers, each object is assigned to the cluster whose center is closest to the object. An object is *expected to belong to the closest cluster.*
  - **Maximization Step** (M-step): Given the cluster assignment, the algorithm *adjusts the center* for each cluster so that *the sum of distance* from the objects assigned to this cluster and the new center is minimized
- **The (EM) algorithm:** A framework to approach maximum likelihood or maximum a posteriori estimates of parameters in statistical models
  - **E-step** assigns objects to clusters according to the current parameters of probabilistic clusters
  - **M-step** finds the new clustering or parameters that minimize the sum of squared errors (SSE) or the expected likelihood

# Expectation-Maximization for One Dimension (Univariate)

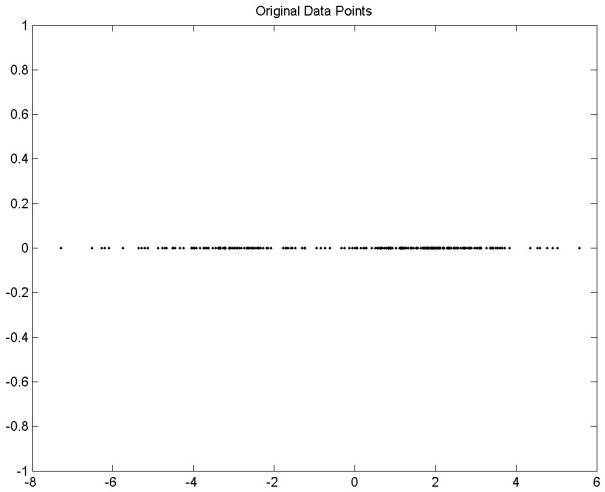
- Consider a dataset ***D*** consisting of a single attribute *X*, where each point $x_i$ (*i* = 1, …, n) is a random sample from *X*
- For the mixture model, we use univariate normals for each cluster

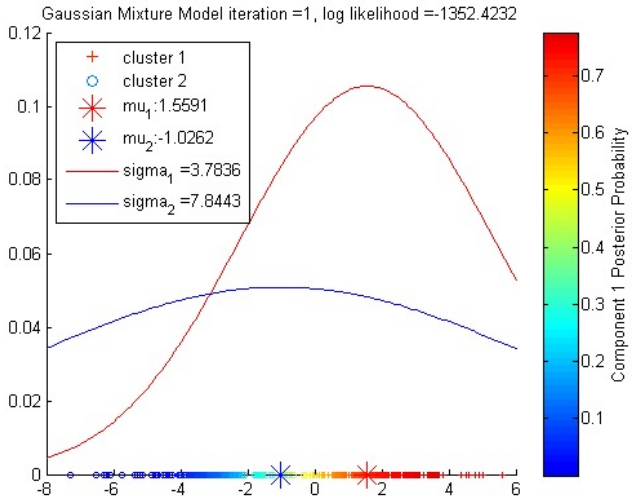
$$f_i(x) = f(x \mid \mu_i, \sigma_i^2) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\{-\frac{(x - \mu_i)^2}{2\sigma_i^2}\}$$

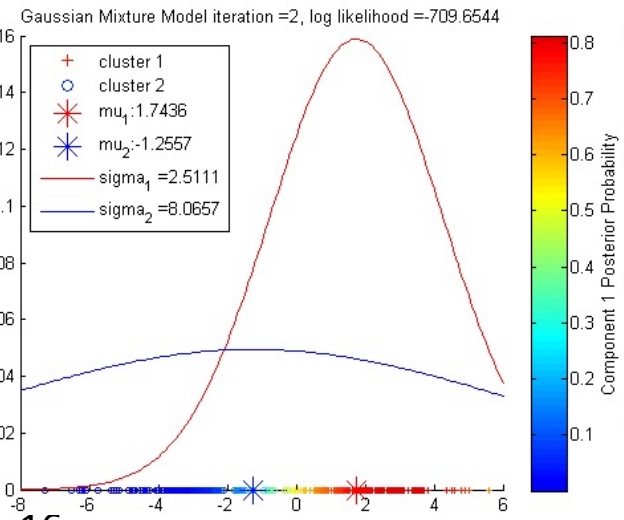
- Initialization:
  - For each cluster $C_i$, with *i* = 1, …, *k*, randomly initialize cluster parameters:
    - $\mu_i$ is selected uniformly at random; $\sigma_i^2 = 1$; $P(C_i) = 1/k$ (each cluster has equal prob.)
- Expectation step:
  - Calculate the posterior probability $P(C_i \mid x_j)$: $P(C_i \mid x_j) = \frac{f(x_j \mid \mu_i, \sigma_i^2) \cdot P(C_i)}{\sum_{a=1}^{k} f(x_j \mid \mu_a, \sigma_a^2) \cdot P(C_a)}$
- Maximization step:
  - Compute the maximum likelihood estimates of the cluster parameters by re-estimating $\mu_i$, $\sigma_i^2$ and $P(C_i)$ for each cluster $C_i$

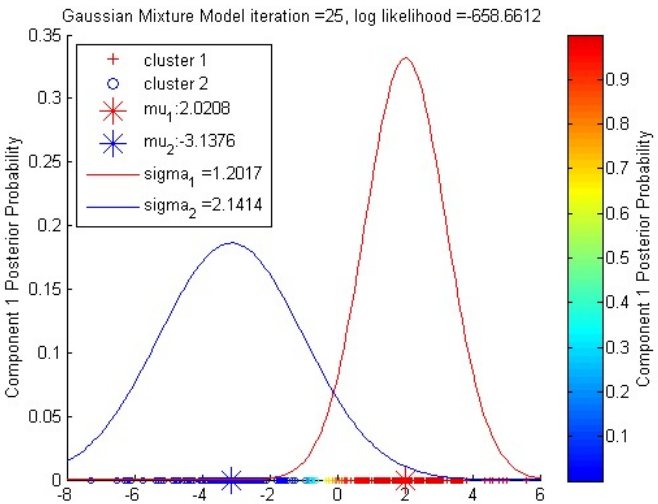
# Demonstration of the EM Execution for One Dimensional Data

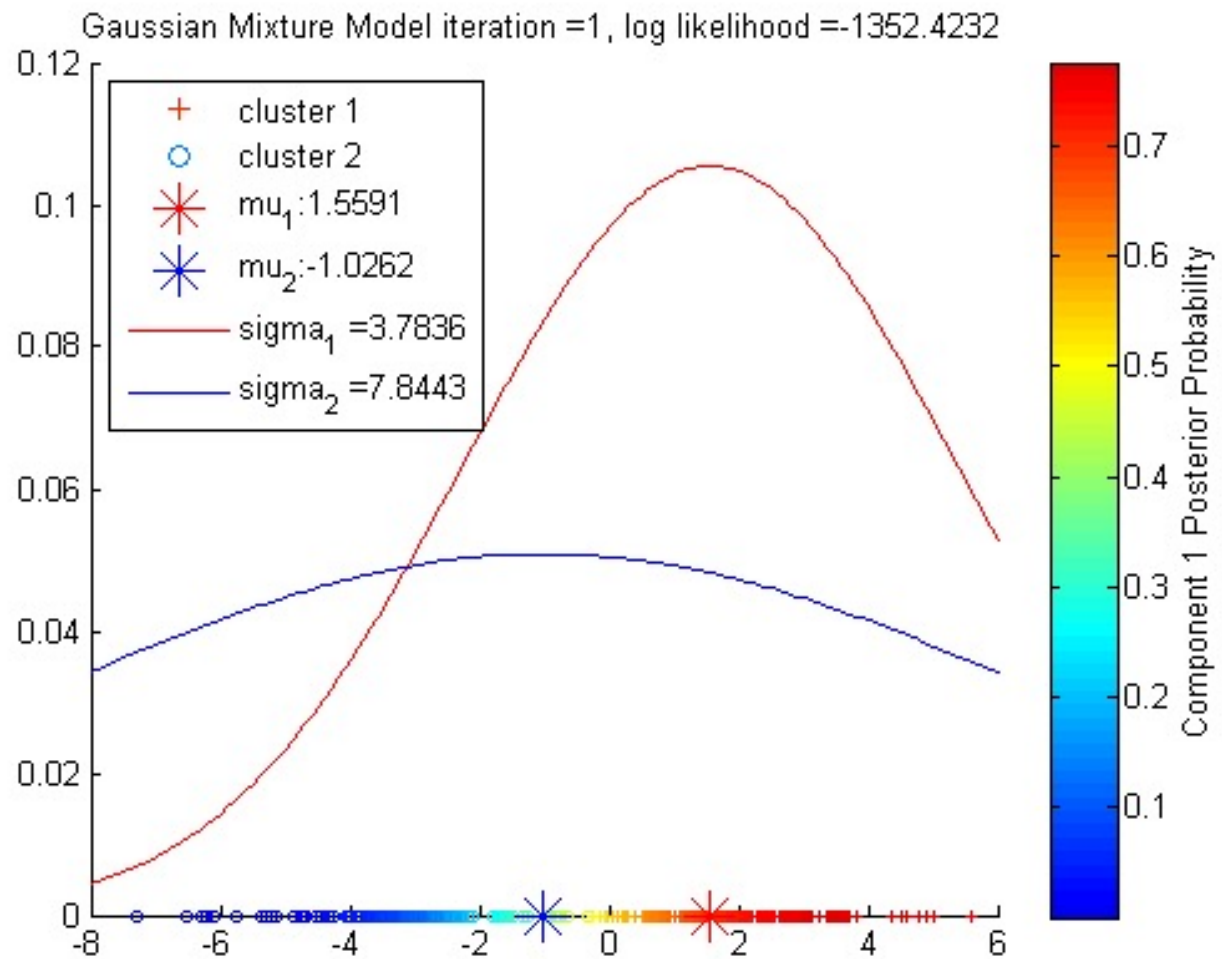
- The execution of the EM Algorithm for Univariate (Single Dimension)

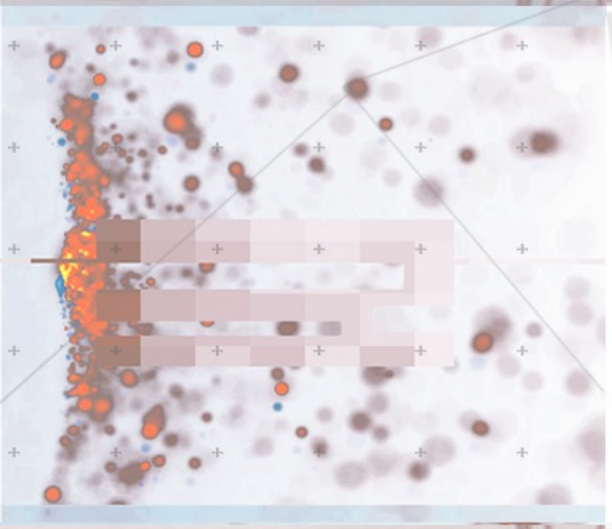




# Session 5: The Expectation-Maximization (EM) Algorithm (Multivariate)

# The Expectation Maximization Algorithm (Multivariate)

- Randomly initialize $\mu_1, \cdots, \mu_k$; $\Sigma_i \leftarrow \mathbf{I}\ \forall i = 1, \ldots, k$ ; $P(C_i) \leftarrow 1/k\ \forall i = 1, \ldots, k$ // **Initialization**
- **Repeat**

// **Expectation Step:** Assigns objects to clusters according to the current parameters of probabilistic clusters

- for $i$ = 1, . . . , $k$ and $j$ = 1, . . . , $n$ do

$$w_{ij} \leftarrow \frac{f(x_j \mid \mu_i, \Sigma_i) \cdot P(C_i)}{\sum_{a=1}^{k} f(x_j \mid \mu_a, \Sigma_a) \cdot P(C_a)}$$

// Calculate the posterior probability $P(C_i|x_j)$

// **Maximization Step:** Finds the new clustering or parameters that minimize SSE or the expected likelihood

- for $i$ = 1, . . . , $k$ do

$$\mu_i \leftarrow \frac{\sum_{j=1}^{n} w_{ij} \cdot \mathrm{x}_j}{\sum_{j=1}^{n} w_{ij}}$$

// re-estimate mean

$$\Sigma_i \leftarrow \frac{\sum_{j=1}^{n} w_{ij} (\mathrm{x}_j - \mu_i)(\mathrm{x}_j - \mu_i)^T}{\sum_{j=1}^{n} w_{ij}}$$

// re-estimate covariance matrix

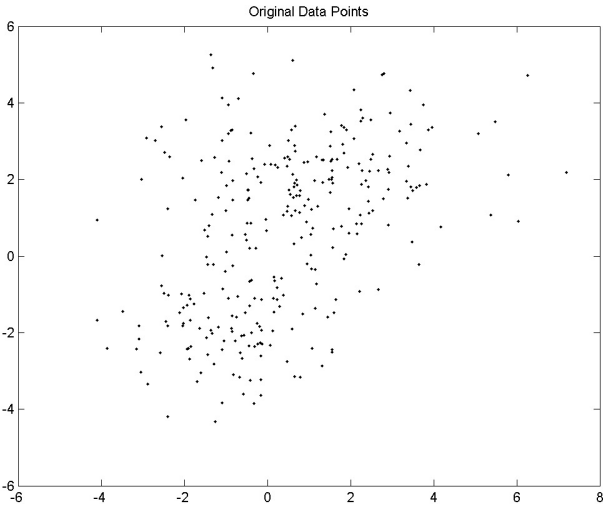
$$P(C_i) \leftarrow \frac{\sum_{j=1}^{n} w_{ij}}{n}$$

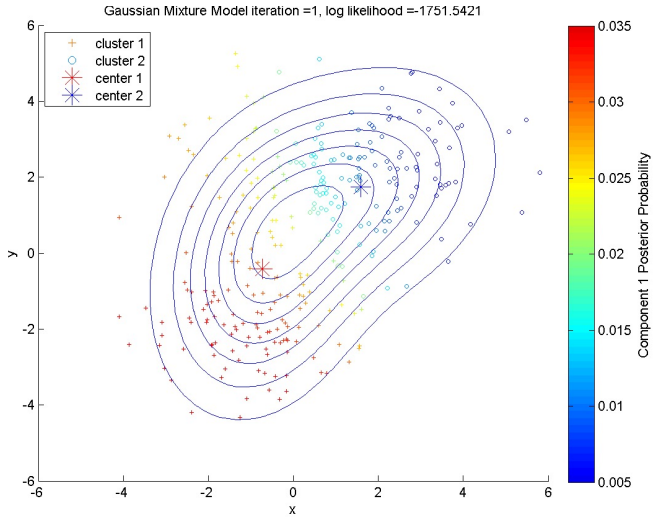
// re-estimate priors

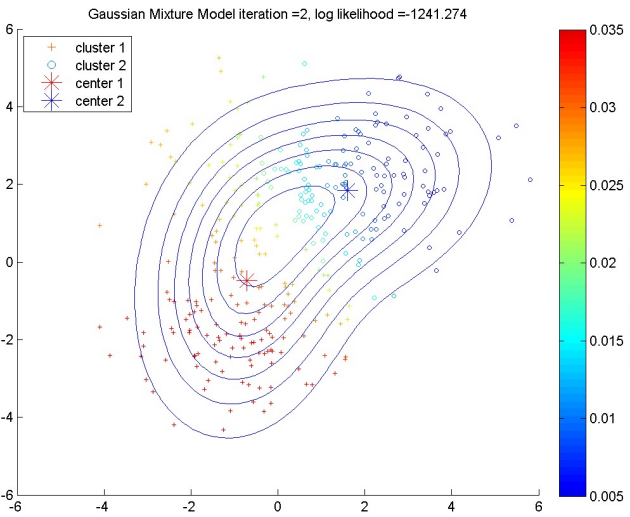
- **Until** the sum of the changes of the means across two iterations is no greater than threshold ϵ

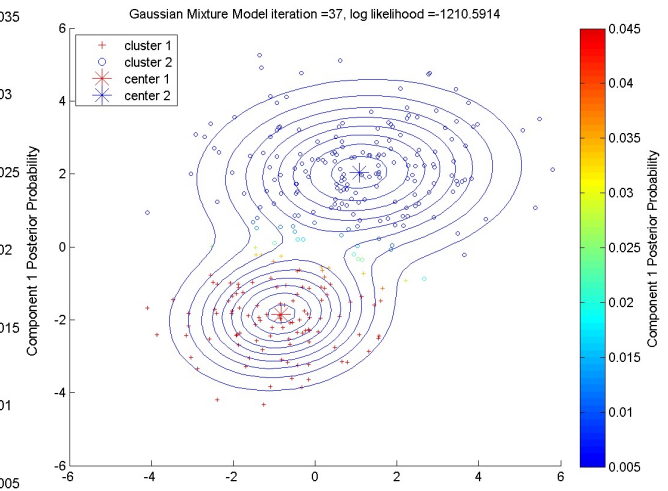
# Demonstration of the EM Execution for Two Dimensional Data

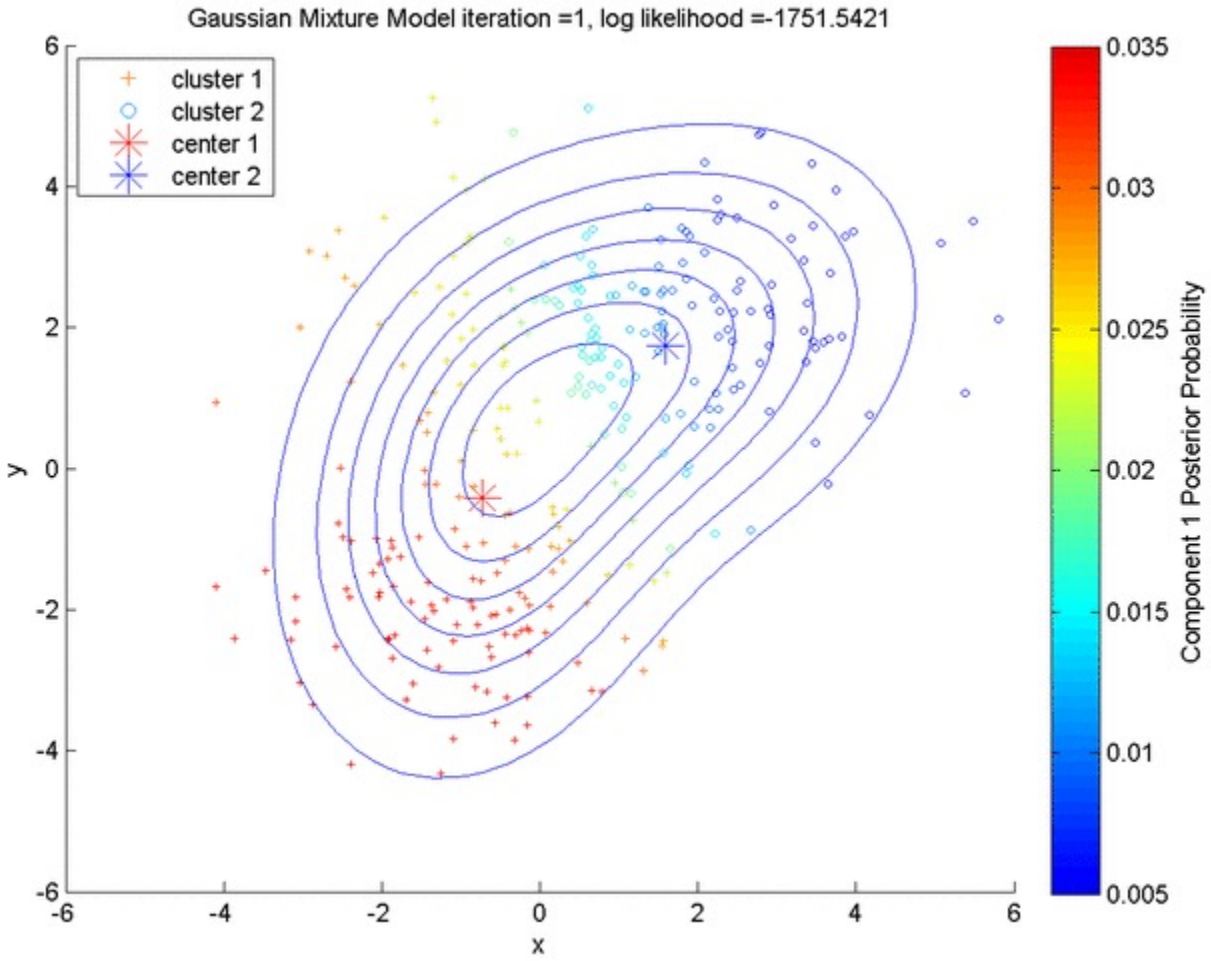
- The execution of the EM algorithm for a two-dimensional data set



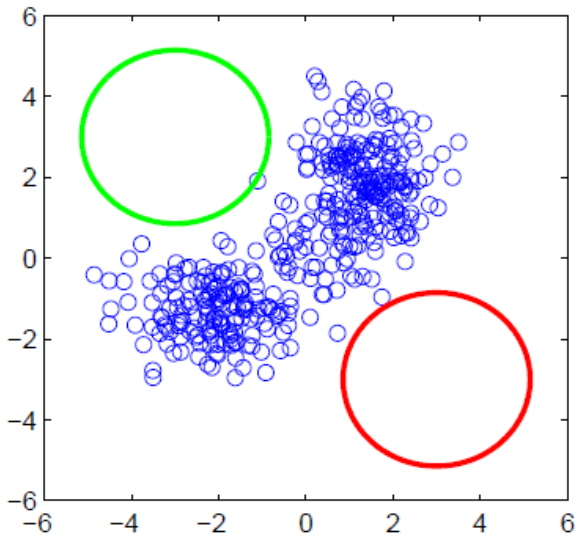



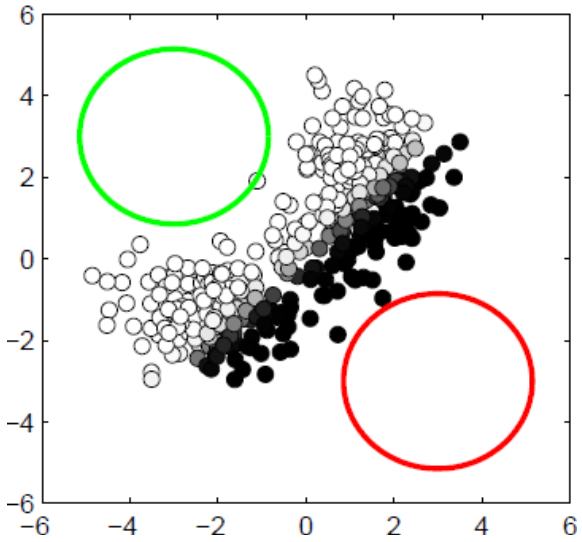



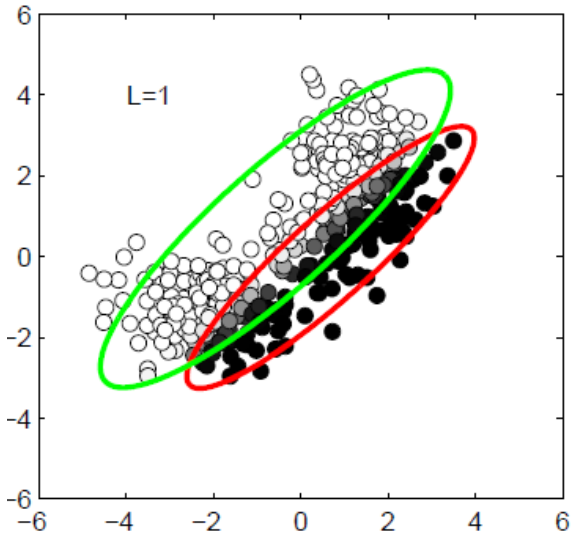
# Illustration of the EM Algorithm for Two Gaussian Components



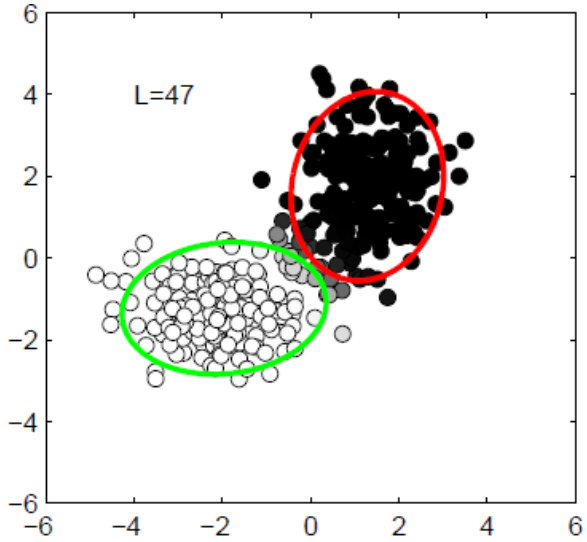
A randomly generated data set (in blue circles). A random initialization of the mixture model: The two Gaussian components are shown as green and red circles



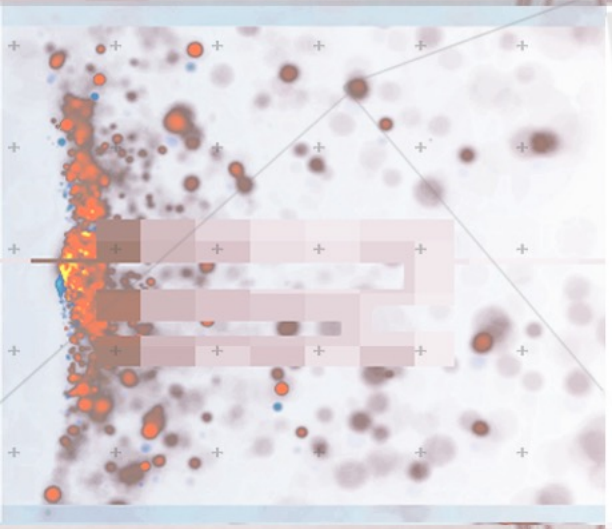
After the initial E-step: Each data point is depicted using a proportion of white ink and black ink according to the posterior probability generated by the corresponding component




After the first M-step: The means and covariances of both components have changed




The results after 47 cycles of EM: Close to convergence



# Session 6: Analysis of the Mixture Model Methods

# K-Means Can Be Considered as a Special Case of EM

- K-means can be considered as a special case of the EM algorithm, where

$$P(\mathrm{x}_j \mid C_i) = \begin{cases} 1 & \text{if } C_i = \arg\min_{C_a} \{\| x_j - \mu_a \|^2\} \\ 0 & \text{otherwise} \end{cases}$$

$$P(C_i \mid \mathrm{x}_j) = \begin{cases} 1 & \text{if } \mathrm{x}_j \in C_i, \text{i.e., } C = \arg\min_{C_a} \{\| x_j - \mu_a \|^2\} \\ 0 & \text{otherwise} \end{cases}$$

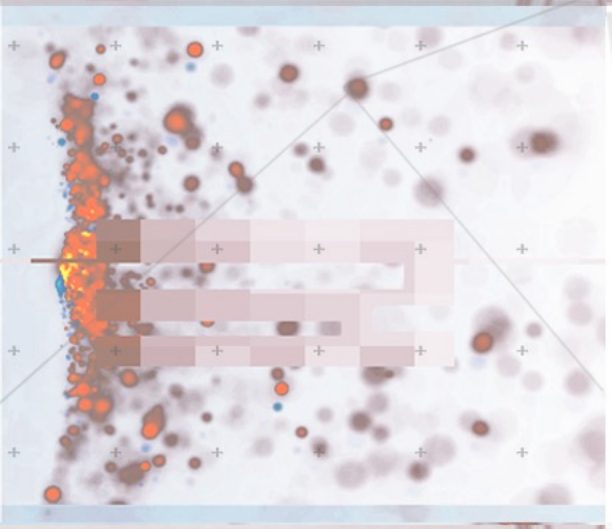
- K-means can be viewed as a hard-EM: In the E-step, we take the local minimum instead of a distribution
- The Gaussian Mixture Model (GMM) is the soft version of k-means
  - We calculate the distribution instead of the most likely one in the E-step and use the weighted sum to compute the new centers in the M-step
  - GMM introduces variance to learning, whereas clusters in k-means have the same variance

# Initialization and Speed-Up of Expectation-Maximization

- Hard vs. soft clustering assignments
  - K-Means: Hard assignment clustering—Each point can belong to only one cluster
  - Probabilistic clustering: Soft assignment of points to clusters—Each point has a probability of belonging to each cluster
- Compared with K-means algorithm, the EM algorithm for Gaussian mixture model (GMM) takes many more iterations to reach convergence
- To find a suitable initialization and speed up the convergence for a GMM:
  - First run the *K*-means algorithm, and then choose the means and covariances of the clusters and the fractions of data points assigned to the respective clusters for initializing $\mu_k$, $\Sigma_k$ and $P(C_i)$, respectively
- A Gaussian component collapses onto a particular data point (called: singularity)
  - When detecting a Gaussian component is collapsing, reset its mean and covariance, and then continue with the optimization

# Strengths and Weaknesses of Mixture Models

- Strengths
  - Mixture models are more general than partitioning and fuzzy clustering
  - Clusters can be characterized by a small number of parameters
  - The results may satisfy the statistical assumptions of the generative models
- Weaknesses
  - Converge to local optimal (overcome: run multiple times with random initialization)
  - Computationally expensive if the number of distributions is large or the data set contains very few observed data points
  - Need large data sets
  - Hard to estimate the number of clusters



# Summary

# Summary: Probabilistic Model-Based Clustering Methods

- Basic Concepts of Probabilistic Model-Based Clustering
- Mixture Models for Cluster Analysis
- Gaussian Mixture Models
- The Expectation-Maximization (EM) Algorithm (Univariate)
- The Expectation-Maximization (EM) Algorithm (Multivariate)
- Analysis of the Mixture Model Methods

# Recommended Readings


- A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society.* 1977
- G. J. McLachlan and K. E. Bkasford. *Mixture Models: Inference and Applications to Clustering*. John Wiley & Sons, 1988
- K. Burnham and D. Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer Verlag, 2002
- C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006
- M. J. Zaki and W. Meira, Jr.. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press, 2014
- H. Deng and J. Han, *Probabilistic Models for Clustering*, in (Chapter 3) C. Aggarwal and C. K. Reddy (eds.), *Data Clustering: Algorithms and Applications*. CRC Press, 2014