Cluster Analysis (b)

Lijun Zhang
zlj@nju.edu.cn
http://cs.nju.edu.cn/zlj

Outline

- **Grid-Based and Density-Based Algorithms**
- Graph-Based Algorithms
- Non-negative Matrix Factorization
- Cluster Validation
- Summary

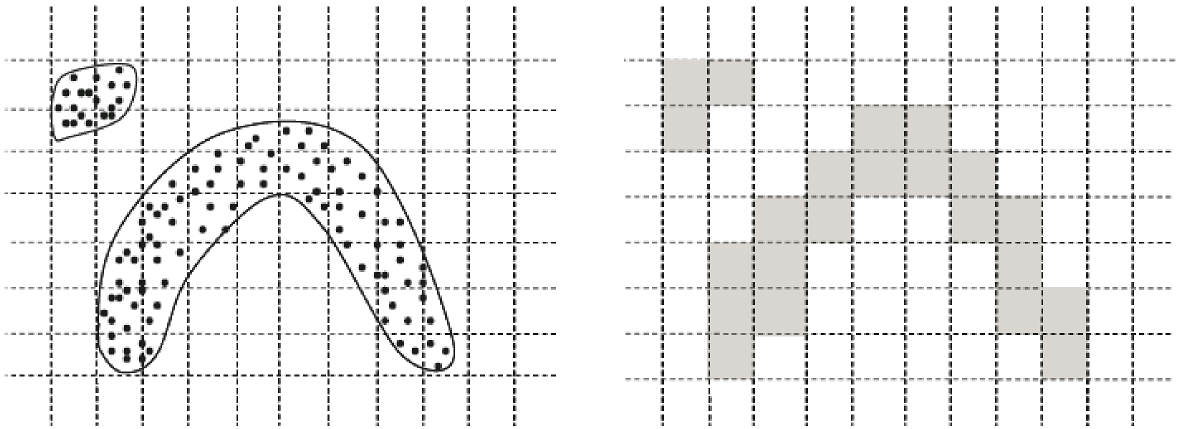
Density-Based Algorithms

- One Motivation
  - Find clusters with arbitrary shape
- The Key Idea
  - Identify fine-grained dense regions
  - Merge regions into clusters
- Representative Algorithms
  - Grid-Based Methods
  - DBSCAN
  - DENCLUE

Grid-Based Methods

The Algorithm

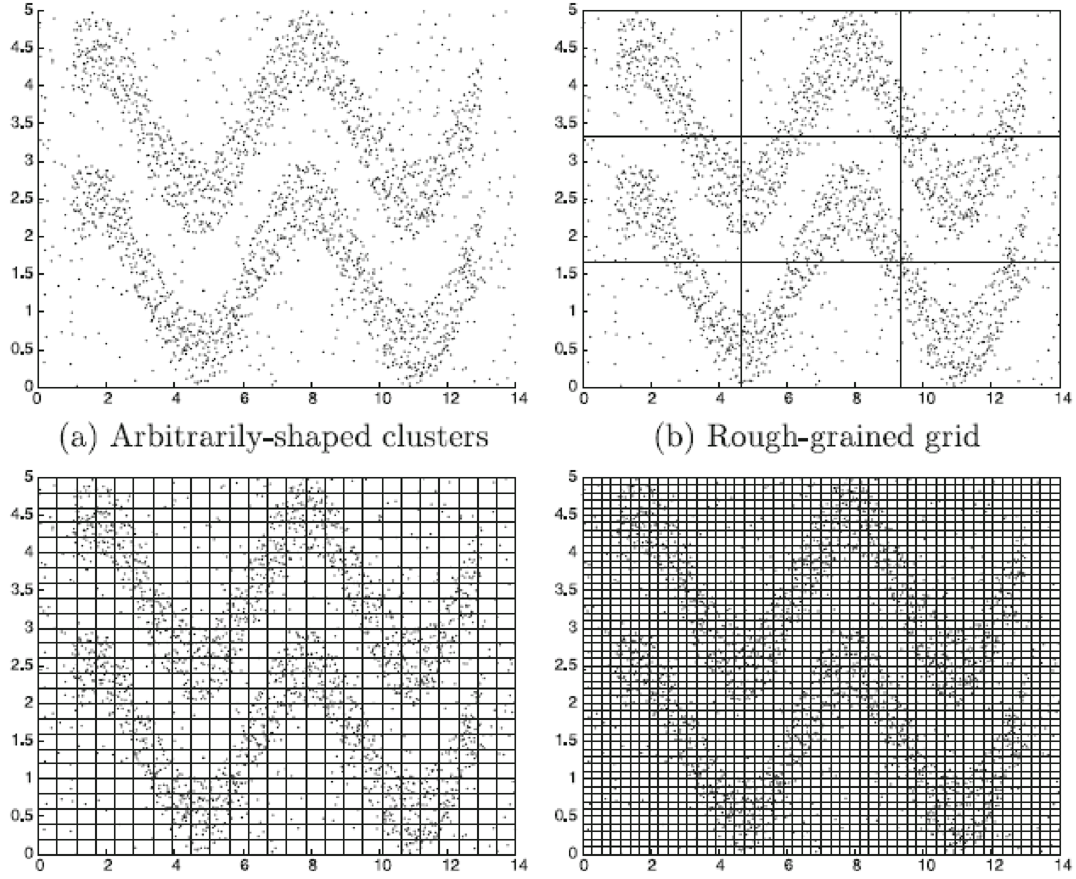
Algorithm *GenericGrid*(Data: $\mathcal{D}$, Ranges: p, Density: τ)
begin
  Discretize each dimension of data $\mathcal{D}$ into p ranges;
  Determine dense grid cells at density level τ;
  Create graph in which dense grids are connected if they are adjacent;
  Determine connected components of graph;
  return points in each connected component as a cluster;
end



(a) Data points and grid   (b) Agglomerating adjacent grids

Limitations-2 Parameters (1)

- The number of Grids

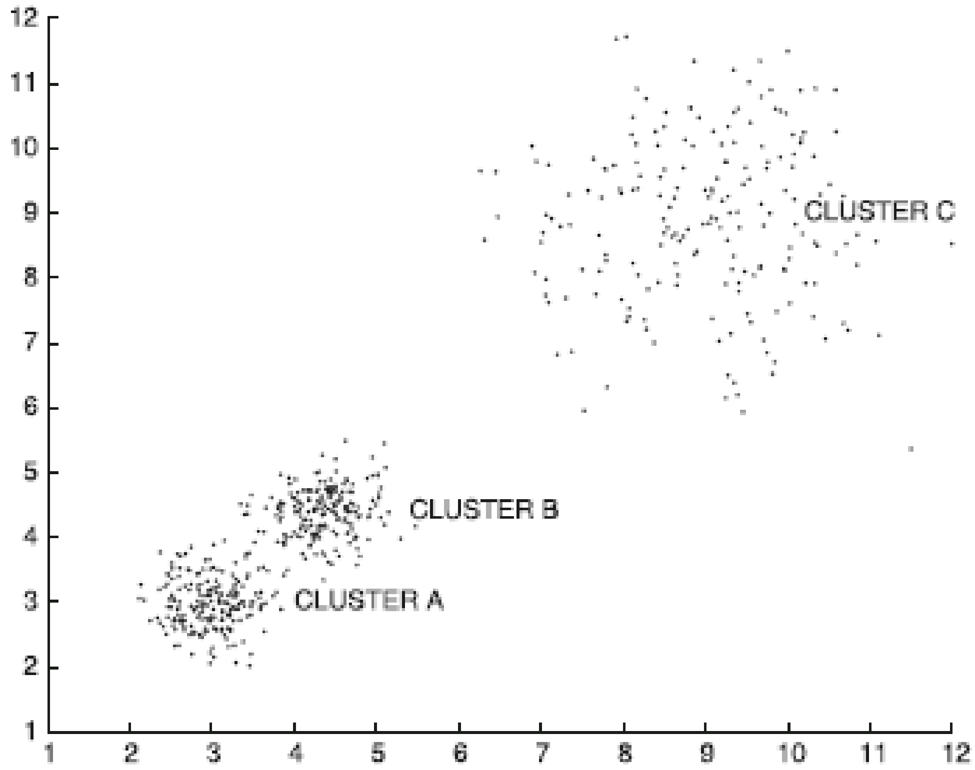


(a) Arbitrarily-shaped clusters (b) Rough-grained grid

(c) Moderate-grained grid (d) Fine-grained grid

Limitations-2 Parameters (2)

- The Level of Density

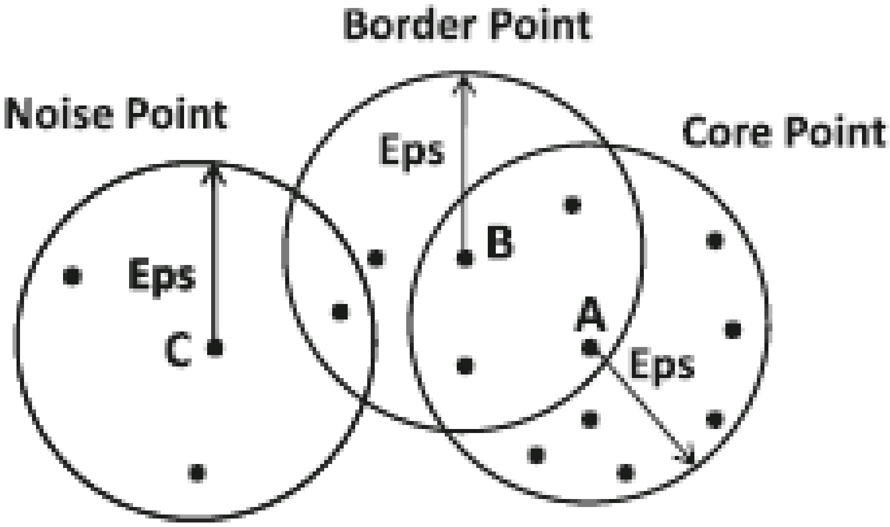


DBSCAN (1)

1. Classify data points into
   - Core point: A data point is defined as a core point, if it contains at least τ data points within a radius Eps.
   - Border point: A data point is defined as a border point, if it contains less than τ points, but it also contains at least one core point within a radius Eps.
   - Noise point: A data point that is neither a core point nor a border point is defined as a noise point.

DBSCAN (2)

1. Classify data points into Core point, Border point, and Noise points.

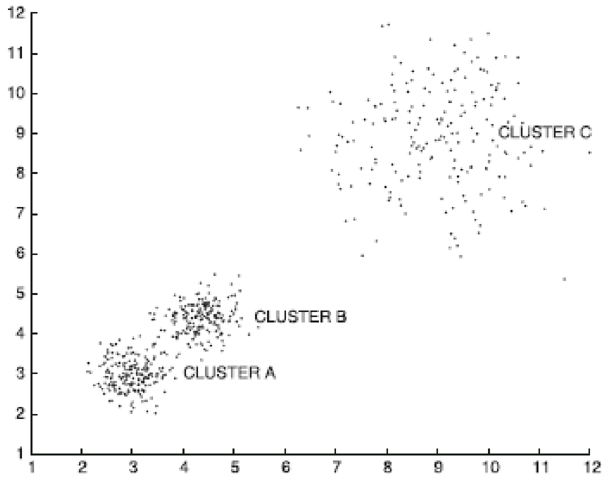


DBSCAN (3)

1. Classify data points into Core point, Border point, and Noise points.
2. A connectivity graph is constructed with respect to the core points
   - Core points are connected if they are within Eps of one another
3. Determine connected components
4. Assign each border point to connected component
   - with which it is best connected

Limitations of DBSCAN

- Two Parameters
  - Radius Eps and Level of Density τ



  - They are related to each other
- High Computational Cost
  - Identifying neighbors $O(n^2)$

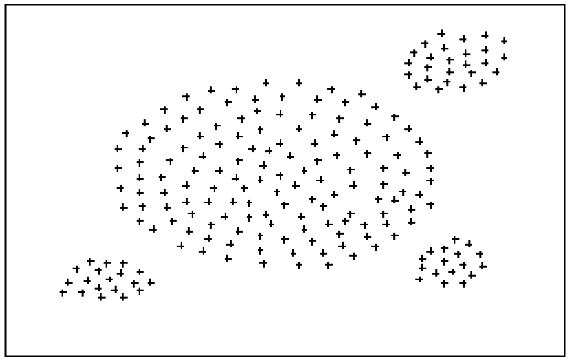
DENCLUE—Preliminary

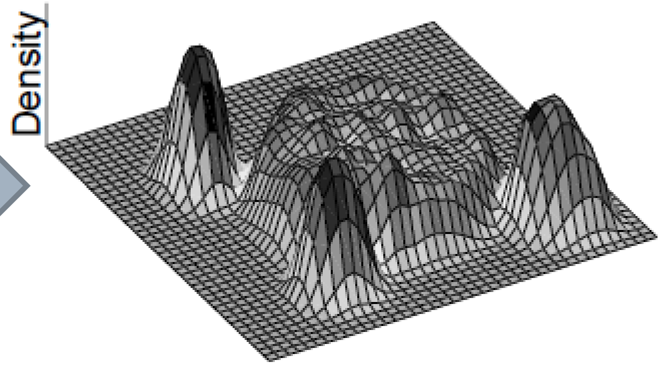
- Kernel-density Estimation
  - Given n data points $\overline{X_1}, \ldots, \overline{X_n}$

$$f(\overline{X}) = \frac{1}{n} \sum_{i=1}^{n} K(\overline{X} - \overline{X_i}).$$

  - $K(\cdot)$ is a kernel function

$$K(\overline{X} - \overline{X_i}) = \left(\frac{1}{h\sqrt{2\pi}}\right)^d e^{-\frac{||\overline{X} - \overline{X_i}||^2}{2 \cdot h^2}}.$$

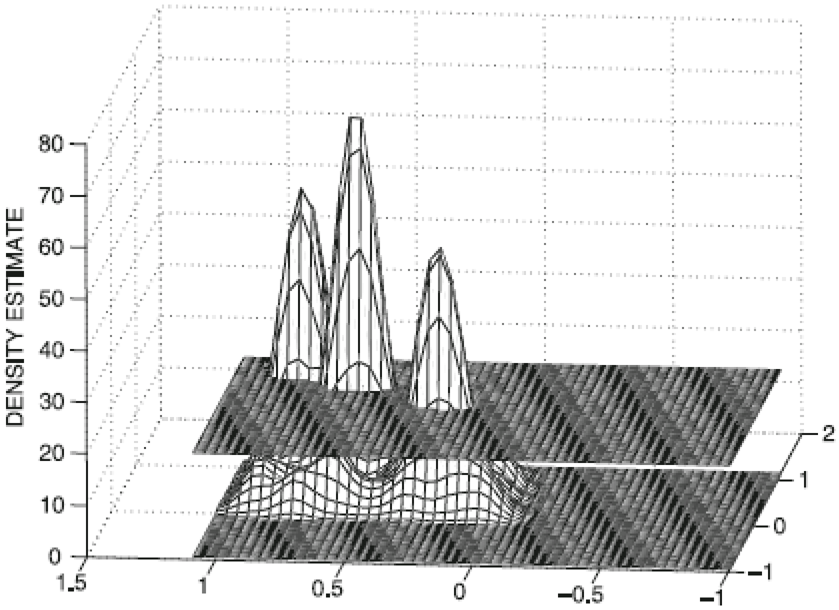




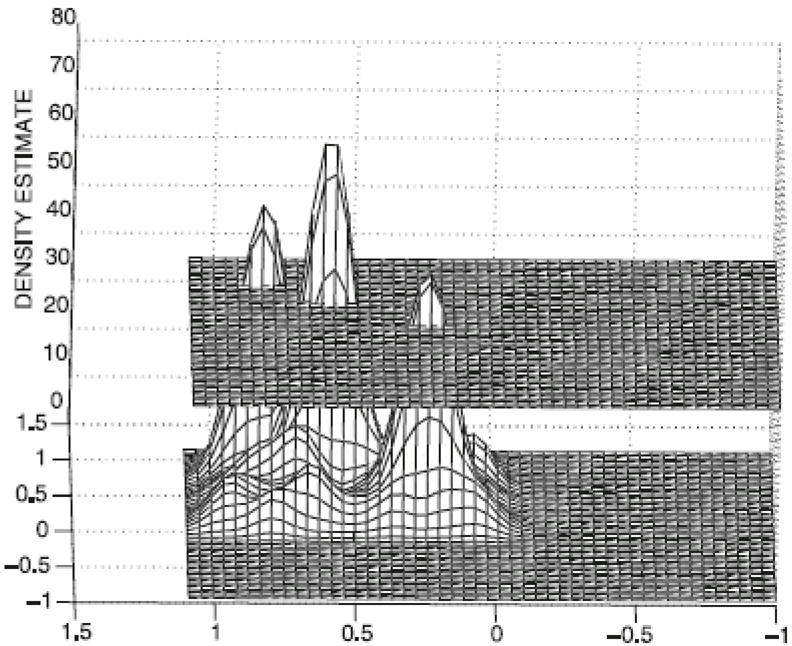
[Hinneburg and Keim, 1998]

DENCLUE—The Key Idea

- Determine clusters by using a density threshold τ



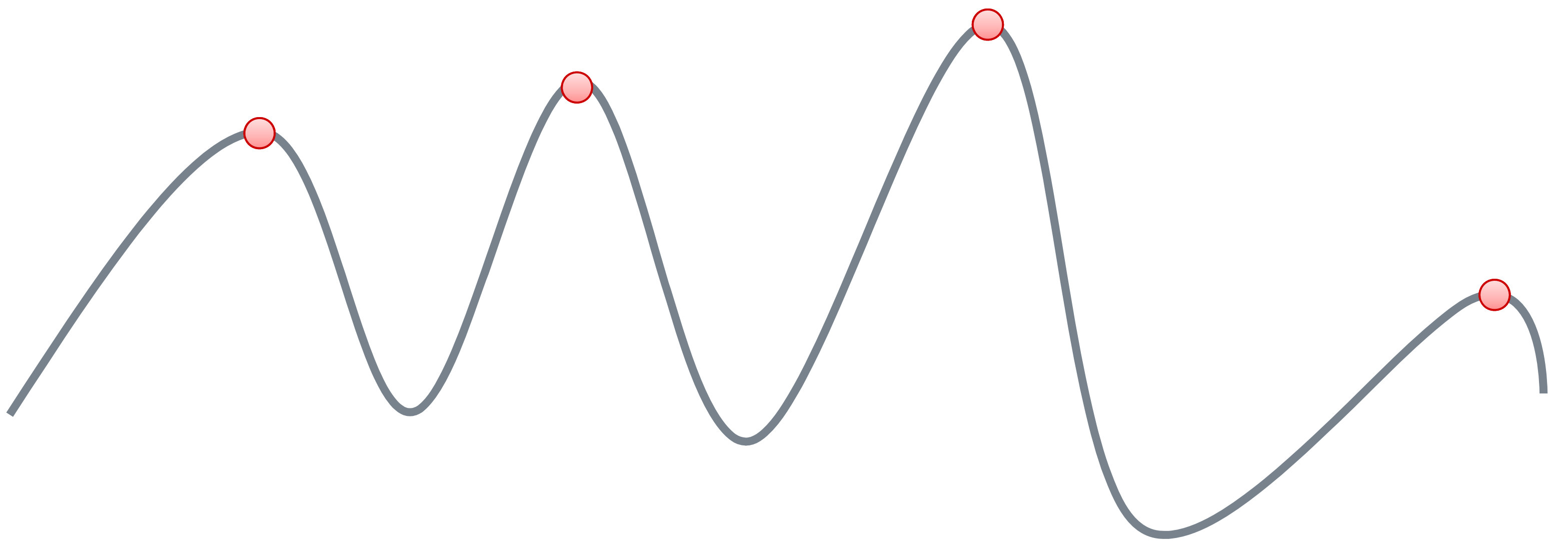
2 clusters



3 clusters

DENCLUE—Procedure

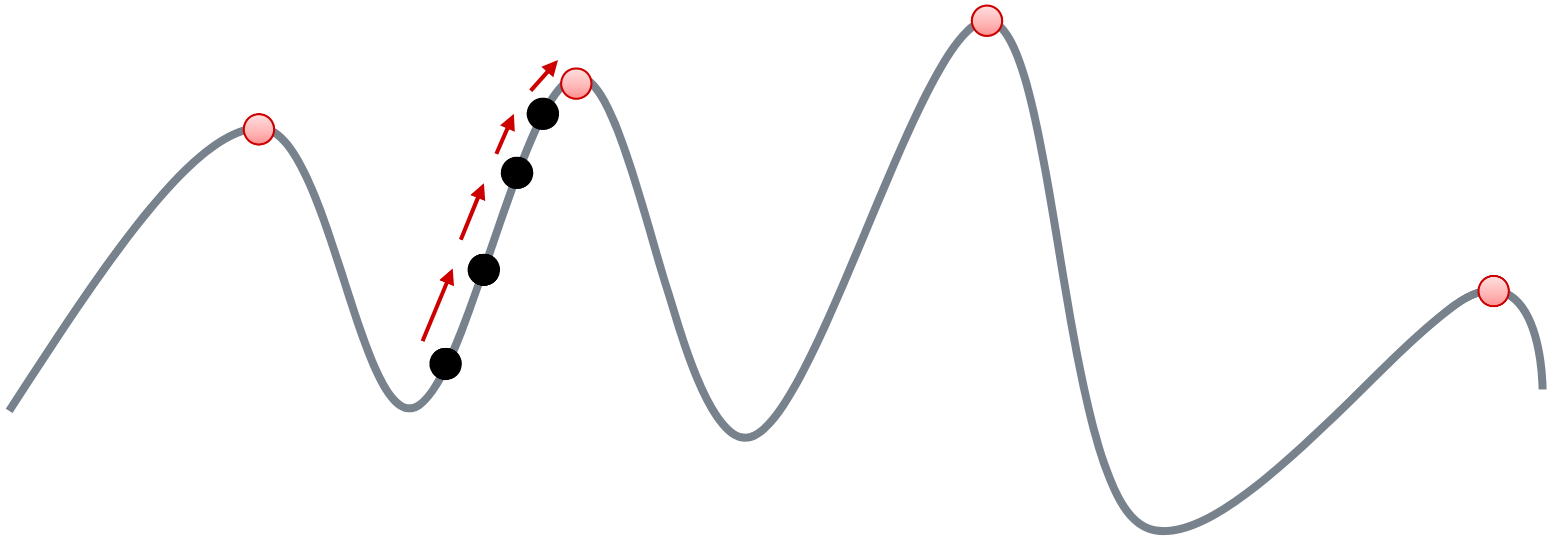
- Density Attractors
  - Local Maximum/Peak



DENCLUE—Procedure

- Density Attractors
  - Local Maximum/Peak
- Identify a Peak for Each Data Point
  - An iterative gradient ascent

$$\overline{X^{(t+1)}} = \overline{X^{(t)}} + \alpha \nabla f(\overline{X^{(t)}})$$



DENCLUE—Procedure

- Density Attractors
  - Local Maximum/Peak
- Identify a Peak for Each Data Point
  - An iterative gradient ascent

$$\overline{X^{(t+1)}} = \overline{X^{(t)}} + \alpha \nabla f(\overline{X^{(t)}})$$

- Post-Processing
  - Attractors whose density is smaller than τ are excluded
  - Density attractors are connected to each other by a path of density at least τ will be merged

DENCLUE—Implementation

- Gradient Ascent
  - Gradient

$$\nabla f(\overline{X}) = \frac{1}{n}\sum_{i=1}^{n} \nabla K(\overline{X} - \overline{X_i}).$$

  - Gaussian Kernel

$$\nabla K(\overline{X} - \overline{X_i}) \propto (\overline{X_i} - \overline{X}) K(\overline{X} - \overline{X_i})$$

- Mean-shift Method

$$\overline{X^{(t+1)}} = \frac{\sum_{i=1}^{n} \overline{X_i} K(\overline{X^{(t)}} - \overline{X_i})}{\sum_{i=1}^{n} K(\overline{X^{(t)}} - \overline{X_i})}$$

  - Converges much faster

Outline

- Grid-Based and Density-Based Algorithms
- **Graph-Based Algorithms**
- Non-negative Matrix Factorization
- Cluster Validation
- Summary

Graph Construction for a Set of n Points $\mathcal{O} = \{O_1, \dots, O_n\}$

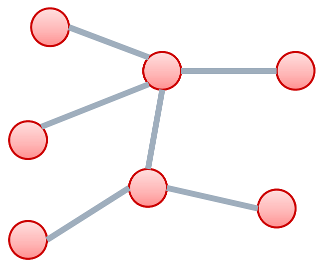
- A node is defined for each $O_i \in \mathcal{O}$
- An edge exists between O_i and O_j
  - If the distance $d(O_i, O_j) \leq \epsilon$
  - If either one is a m-nearest neighbor of the other (A better approach)
- If there is an edge, then its weight is
  - 1
  - Heat Kernel: $e^{-d(O_i,O_j)^2/t^2}$

Spectral Clustering

□ Dimensionality Reduction

- Find a low-dimensional representation for each node in the graph

$$o_1, o_2, \vdots, o_n \Rightarrow$$



$$\Rightarrow \mathbf{y}_1 \in \mathbb{R}^k, \mathbf{y}_2 \in \mathbb{R}^k, \vdots, \mathbf{y}_n \in \mathbb{R}^k$$

$$W = [w_{ij}] \in \mathbb{R}^{n \times n}$$

- Laplacian Eigenmap [Belkin and Niyogi, 2002]

□ k-means

- Apply k-means to new representations of the data

Laplacian Eigenmap (1)

- The Objective Function ($k = 1$)
  - $y_i \in \mathbb{R}$ is a 1-dimensional representation of O_i
  - w_{ij} is the similarity between O_i and O_j

$$O = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \left(y_i - y_j \right)^2$$

  - Similar points will be mapped closer
    - ✓ Similar points have larger weights

Laplacian Eigenmap (2)

The Objective Function ($k = 1$)

- Vector Form

$$O = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (y_i - y_j)^2 = 2\mathbf{y}^\top L \mathbf{y}$$

- $\mathbf{y} = [y_1, \dots, y_n]^\top \in \mathbb{R}^n$
- $L = D - W \in \mathbb{R}^{n \times n}$ is the graph Laplacian
  - ✓ Positive Semidefinite (PSD)
- $W = [w_{ij}] \in \mathbb{R}^{n \times n}$ is the similarity matrix
- $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix with $D_{ii} = \sum_{j=1}^{n} w_{ij}$

Laplacian Eigenmap (3)

- The Optimization Problem ($k = 1$)

$$\min_{\mathbf{y}\in\mathbb{R}^n} \quad \mathbf{y}^\top L\mathbf{y}$$
$$\text{s.t.} \quad \mathbf{y}^\top D\mathbf{y} = 1$$

  - Add a Constraint to Remove Scaling Factor
    - ✓ D is introduced for normalization [Luxburg, 2007]

- The Solution $$L\mathbf{y} = \lambda D\mathbf{y}$$

  - Generalized Eigenproblem [Luxburg 2007]
  - The smallest eigenvector is $\mathbf{y}^1 = \mathbf{1}$
    - ✓ Useless since $y_1^1 = y_2^1 = \cdots = y_n^1$

Laplacian Eigenmap (3)

- The Optimization Problem ($k = 1$)

$$\min_{\mathbf{y}\in\mathbb{R}^n} \quad \mathbf{y}^\top L\mathbf{y}$$
$$\text{s.t.} \quad \mathbf{y}^\top D\mathbf{y} = 1$$

  - Add a Constraint to Remove Scaling Factor
    - ✓ D is introduced for normalization [Luxburg, 2007]

- The Solution $L\mathbf{y} = \lambda D\mathbf{y}$

  - Generalized Eigenproblem [Luxburg 2007]
  - The smallest eigenvector is $\mathbf{y}^1 = \mathbf{1}$
  - Use the second smallest eigenvector $\mathbf{y}^2$
    - ✓ The new representation for o_i is y_i^2

Laplacian Eigenmap (4)

- The Objective Function ($k > 1$)
  - Vector Form

$$O = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \left\| \mathbf{y}_i - \mathbf{y}_j \right\|_2^2 = 2\mathrm{trace}(Y^T L Y)$$

  - $Y = [\mathbf{y}_1, \dots, \mathbf{y}_n]^\top \in \mathbb{R}^{n \times k}$
  - $L = D - W \in \mathbb{R}^{n \times n}$ is the graph Laplacian
  - $W = [w_{ij}] \in \mathbb{R}^{n \times n}$ is the similarity matrix
  - $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix with $D_{ii} = \sum_{j=1}^{n} w_{ij}$

Laplacian Eigenmap (4)

- The Optimization Problem ($k > 1$)

$$\min_{Y \in \mathbb{R}^{n \times k}} \quad \operatorname{trace}(Y^\top L Y)$$
$$\text{s.t.} \quad Y^\top D Y = I$$

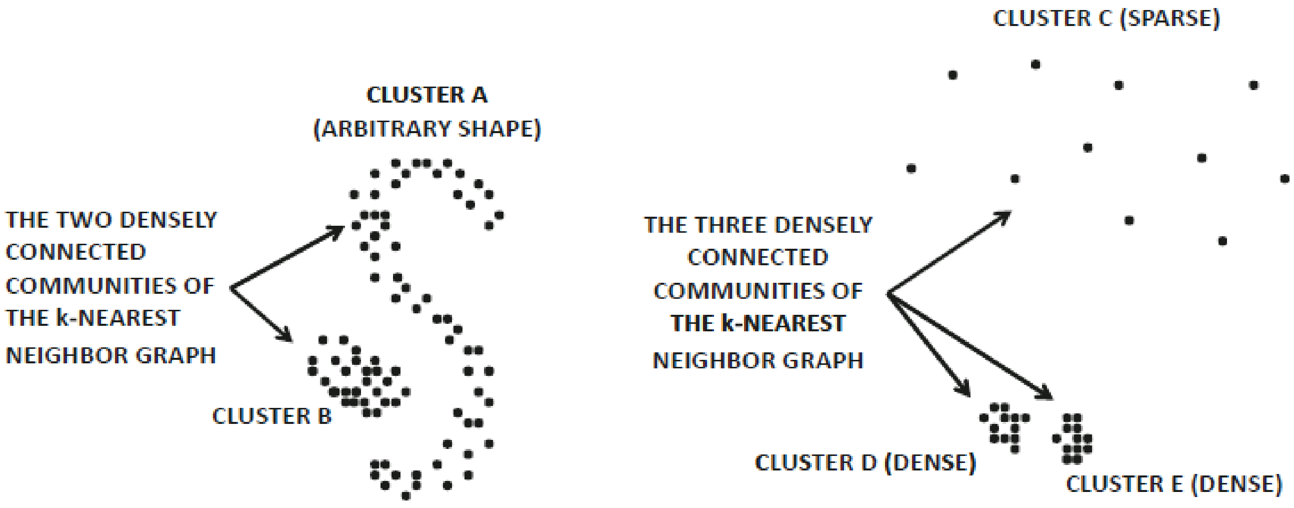
- The Solution

$$L\mathbf{y} = \lambda D\mathbf{y}$$

  - Generalized Eigenproblem [Luxburg 2007]
  - Use $Y = [\mathbf{y}^2, \dots, \mathbf{y}^{k+1}] \in \mathbb{R}^{n \times k}$ as the optimal solution
    - ✓ $\mathbf{y}^i$ is the i-th generalized eigenvector
    - ✓ The new representation $\mathbf{y}_i \in \mathbb{R}^k$ for O_i is the i-th row of Y
  - Don't forget the normalization $Y^\top D Y = I$

Properties of Spectral Clustering

- Varying Cluster Shape and Density



(a) Varying cluster shape (b) Varying cluster density

  - Due to the nearest neighbor graph

- High Computational Cost

Outline

- Grid-Based and Density-Based Algorithms
- Graph-Based Algorithms
- **Non-negative Matrix Factorization**
- Cluster Validation
- Summary

Non-negative Matrix Factorization (NMF)

- Let $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n] = \mathbb{R}^{d\times n}$ be a non-negative data matrix
- NMF aims to factor X as $U \times V^\top$
  - $U \in \mathbb{R}^{d\times k}$ and $V \in \mathbb{R}^{n\times k}$ are non-negative
- The Optimization Problem

$$\begin{aligned} \min_{U\in\mathbb{R}^{d\times k}, V\in\mathbb{R}^{n\times k}} \quad & \|X - UV^\top\|_F^2 \\ \text{s.t.} \quad & U \geq 0, V \geq 0 \end{aligned}$$

  - Non-convex

Interpretation of NMF (1)

- Matrix Appromation

$$X \approx UV^\top$$

- Element-wise
  - $X = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$, where $\mathbf{x}_i \in \mathbb{R}^d$
  - $U = [\mathbf{u}_1, \dots, \mathbf{u}_k] \in \mathbb{R}^{d\times k}$, where $\mathbf{u}_i \in \mathbb{R}^d$
  - $V^\top = [\mathbf{v}_1, \dots, \mathbf{v}_n] \in \mathbb{R}^{k\times n}$, where $\mathbf{v}_i \in \mathbb{R}^k$
    - ✓ $\mathbf{v}_i$ is the i-th column of $V^\top$
    - ✓ $\mathbf{v}_i^\top$ is the i-th row of V
  - Then,

$$\mathbf{x}_i \approx U\mathbf{v}_i = \sum_{j=1}^{k} \mathbf{u}_j v_{ij}$$

    - ✓ v_{ij} is the j-th element of vector $\mathbf{v}_i$

Interpretation of NMF (2)

Vector Approximation

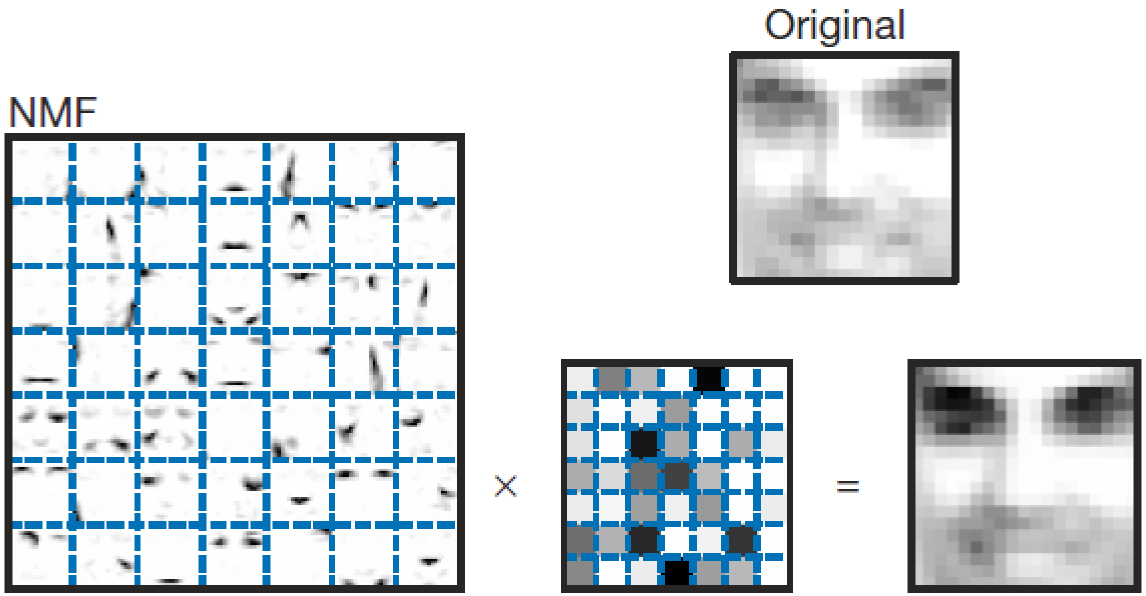
$$\mathbf{x}_i \approx U\mathbf{v}_i = \sum_{j=1}^{k} \mathbf{u}_j v_{ij}$$

- $\mathbf{u}_1, \dots, \mathbf{u}_k \in \mathbb{R}^d$ can be treated as basis vectors
  - ✓ They may be not orthonormal
  - ✓ They are non-negative

- $\mathbf{v}_i = [v_{i1}, \dots, v_{ik}]^\top \in \mathbb{R}^k$ can be treated as a new k-dimensional representation of $\mathbf{x}_i$

Parts-Based Representations

- When each $\mathbf{x}_i$ is a face image



- [Lee and Seung, 1999]

Clustering by NMF

- Vector Approximation

$$\mathbf{x}_i \approx U\mathbf{v}_i = \sum_{j=1}^{k} \mathbf{u}_j v_{ij}$$

  - $\mathbf{u}_j$ can be treated as an representative of the j-th cluster
  - v_{ij} can be treated as the association between $\mathbf{x}_i$ and $\mathbf{u}_j$

- The cluster label l_i for $\mathbf{x}_i$

$$l_i = \mathrm{argmax}_j v_{ij}$$

  - [Xu et al., 2003]

An Example

- Discover both Row and Column Clusters

$$\begin{bmatrix} 2 & 2 & 1 & 2 & 0 & 0 \\ 2 & 3 & 3 & 3 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 3 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 2 & 1 & 2 \end{bmatrix} \approx \begin{bmatrix} 2 & 0 \\ 3 & 0 \\ 1 & 0 \\ 2 & 1 \\ 0 & 1 \\ 0 & 2 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Optimization in NMF

- Alternating between U and V

$$u_{ij} \leftarrow u_{ij} \frac{(\mathbf{XV})_{ij}}{(\mathbf{UV}^T\mathbf{V})_{ij}}$$

$$v_{ij} \leftarrow v_{ij} \frac{(\mathbf{X}^T\mathbf{U})_{ij}}{(\mathbf{VU}^T\mathbf{U})_{ij}}$$

  - Local Optimal Solutions
    - ✓ Run multiple times and choose the best one

- Other Optimization Algorithms are also Possible

Outline

- Grid-Based and Density-Based Algorithms
- Graph-Based Algorithms
- Non-negative Matrix Factorization
- **Cluster Validation**
- Summary

Concepts

- Cluster validation
  - Evaluate the quality of a clustering
- Internal Validation Criteria
  - Do not need additional information
  - Biased toward one algorithm or the other
- External Validation Criteria
  - Ground-truth clusters are known
  - Ground-truth may not reflect the natural clusters in the data

Internal Validation Criteria

- Sum of square distances to centroids

$$\sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \mathcal{C}_j} \left\| \mathbf{x}_i - \mathbf{c}_j \right\|_2^2$$

- Intracluster to intercluster distance ratio

$$Intra = \sum_{(\overline{X_i}, \overline{X_j}) \in P} dist(\overline{X_i}, \overline{X_j}) / |P|$$

$$Inter = \sum_{(\overline{X_i}, \overline{X_j}) \in Q} dist(\overline{X_i}, \overline{X_j}) / |Q|.$$

- Silhouette coefficient
- Probabilistic measure

External Validation Criteria

- Class Labels
  - The Ground-truth
- Confusion Matrix
  - Each row i corresponds to the class label j
  - Each column j corresponds to the algorithm-determined cluster j

Cluster Indices	1	2	3	4
1	97	0	2	1
2	5	191	1	3
3	4	3	87	6
4	0	0	5	195

Cluster Indices	1	2	3	4
1	33	30	17	20
2	51	101	24	24
3	24	23	31	22
4	46	40	44	70

  - Ideal clustering $\Rightarrow$ a diagonal matrix after permutation

Notations

- m_{ij}: number of data points from class (*ground-truth*) cluster i that are mapped to (*algorithm-determined*) cluster j
- N_i: number of data points in *true cluster* i

$$N_i = \sum_{j=1}^{k_d} m_{ij} \qquad \forall i = 1 \ldots k_t$$

- M_j: number of data points in *algorithm-determined* cluster j

$$M_j = \sum_{i=1}^{k_t} m_{ij} \qquad \forall j = 1 \ldots k_d$$

Purity

- For a given algorithm-determined cluster j, define P_j as number of data points in its *dominant* class

$$P_j = \max_i m_{ij}.$$

- The overall purity

$$\text{Purity} = \frac{\sum_{j=1}^{k_d} P_j}{\sum_{j=1}^{k_d} M_j}.$$

  - High values of the purity are desirable

Gini index

- Limitation of Purity
  - Only accounts for the dominant label in the cluster and ignores the distribution of the remaining points
- Gini index G_j for column (algorithm-determined cluster) j

$$G_j = 1 - \sum_{i=1}^{k_t} \left(\frac{m_{ij}}{M_j} \right)^2$$

- The average Gini coefficient
  - Low values

$$G_{average} = \frac{\sum_{j=1}^{k_d} G_j \cdot M_j}{\sum_{j=1}^{k_d} M_j}$$

Outline

- Grid-Based and Density-Based Algorithms
- Graph-Based Algorithms
- Non-negative Matrix Factorization
- Cluster Validation
- **Summary**

Summary

- Grid-Based and Density-Based Algorithms
  - Grid-Based Methods
  - DBSCAN, DENCLUE
- Graph-Based Algorithms
  - Laplacian Eigenmap
- Non-negative Matrix Factorization
- Cluster Validation
  - Purity, Gini index

Reference

- [Belkin and Niyogi, 2002] Belkin, M. and Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. In NIPS 14, pages 585–591.
- [Luxburg, 2007] Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and Computing, 17(4):395–416.
- [Lee and Seung, 1999] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755):788–791.
- [Xu et al., 2003] Xu, W., Liu, X., and Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In SIGIR, pages 267–273.
- [Hinneburg and Keim, 1998] Hinneburg, A. and Keim, D. A. (1998). An efficient approach to clustering in large multimedia databases with noise. In KDD, pages 58–65.