Discussion Section 4 (CS145)

2015-10-23

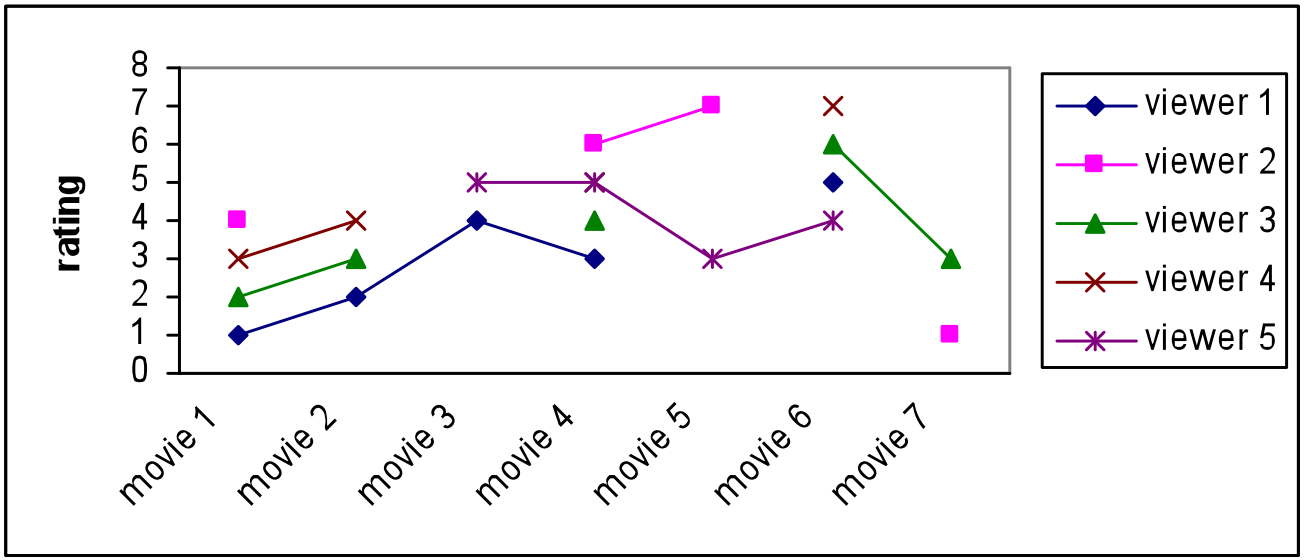
Week 04

Outline

- Midterm Nov 18
- Homework #1 solution
- Review:
  - Bi-Cluster
  - Density-based clustering:
    - OPTICS
  - Grid-Based clustering:
    - STING
  - Clustering on high-dimensional data:
    - CLIQUE (Density and Grid based)

Bi-Clustering

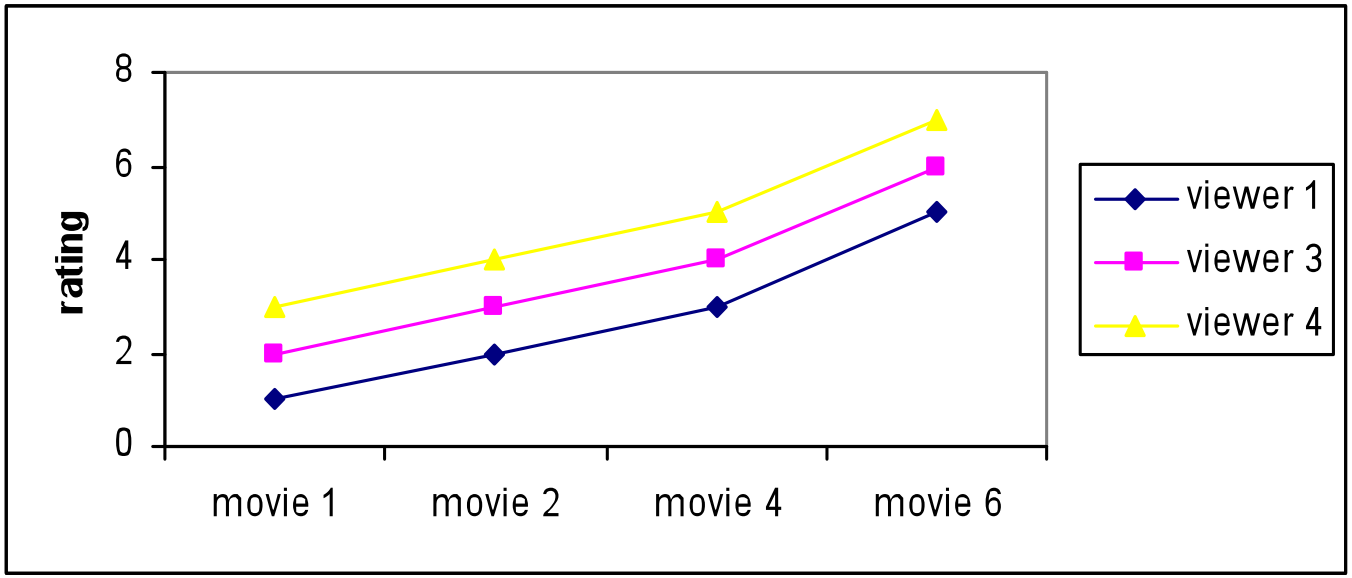
- Motivation:
  - E-commerce example: movie rating



  - It is hard to see the global pattern

Bi-Clustering

- Motivation
  - E-commerce example: movie rating



- It is easier to see the pattern from a subset of data
- We can build a recommendation system using bi-clustering

Bi-Clustering

- Goal
  - To identify a subset of objects (ie, the movies), and a subset of attributes (ie, the viewers) that bear a constant shift (a strong coherence).

Bi-Clustering

Let ***A*** be the original data matrix:

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6	Movie 7
Viewer 1	1	2	4	3		5	
Viewer 2	4			6	7		1
Viewer 3	2	3		4		6	3
Viewer 4	3	4		5		7	
Viewer 5			5	5	3	4	

Let ***A(I, J)*** be one of the bi-clusters:

	Movie 1	Movie 2	Movie 4	Movie 6
Viewer 1	1	2	3	5
Viewer 3	2	3	4	6
Viewer 4	3	4	5	7

I = a set of rows
J = a set of columns

Bi-Clustering

Let ***A(I, J)*** be one of the bi-clusters:

	Movie 1	Movie 2	Movie 4	Movie 6
Viewer 1	1	2	3	5
Viewer 3	2	3	4	6
Viewer 4	3	4	5	7

- For a bi-cluster *A(I, J)*
  - Average of row *i*: $a_{iJ} = \frac{1}{|J|}\sum_{j} a_{ij}$
  - Average of column *j*: $a_{Ij} = \frac{1}{|I|}\sum_{i} a_{ij}$
  - Overall average: $a_{IJ} = \frac{1}{|I||J|}\sum_{i}\sum_{j} a_{ij}$
  - Residual of a_{ij}

$$R_{ij} = a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}$$

  - Mean square residue

$$H(I,J) = \frac{1}{|I||J|}\sum_{i\in I, j\in J} R_{ij}^{2}$$

Note: If all the elements of the bicluster have small residues, clearly the mean squared residue will be small.

Bi-Clustering

- Cheng-Church algorithm
  - Find bi-clusters that are as large as possible, with the restriction that the H-score (mean square residues) must be less than some threshold δ
    - δ-cluster
  - It finds one bi-cluster at each iteration
  - At the end of each iteration, it masks the data in the found bi-cluster with random entries

Bi-Clustering

- Cheng-Church algorithm (pseudocode)

input: matrix A , number of clusters N , $\delta >= 0$, $\alpha > 0$

output: a set of N biclusters, each with $H(I, J) <= \delta$

for $i \leftarrow 1 \ldots N$:

initialize (I, J) to all rows and all columns

Initially, set the entire matrix as a bi-cluster

multiple node deletion

single node deletion

Delete one or multiple rows and columns so that *H(I,J) <= δ*

node addition

append (I, J) to results

Add the bi-cluster to result

mask A_{IJ} with random entries

To avoid getting the same bi-cluster in the next iteration, after finding a bi-cluster, its entries in the original matrix is replaced by random data.

return results

Bi-Clustering

- Cheng-Church algorithm (pseudocode)

Input:

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6	Movie 7
Viewer 1	1	2	4	3		5	
Viewer 2	4			6	7		1
Viewer 3	2	3		4		6	3
Viewer 4	3	4		5		7	
Viewer 5			5	5	3	4	

After the first iteration, we found the first bi-cluster:

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6	Movie 7
Viewer 1	1	2	4	3		5	
Viewer 2	4			6	7		1
Viewer 3	2	3		4		6	3
Viewer 4	3	4		5		7	
Viewer 5			5	5	3	4	

Bi-Clustering

- Cheng-Church algorithm (pseudocode)

Input:

	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6	Movie 7
Viewer 1	1	2	4	3		5	
Viewer 2	4			6	7		1
Viewer 3	2	3		4		6	3
Viewer 4	3	4		5		7	
Viewer 5			5	5	3	4	

At the end of the first iteration, we mask the bi-cluster entries in the original matrix by random data:

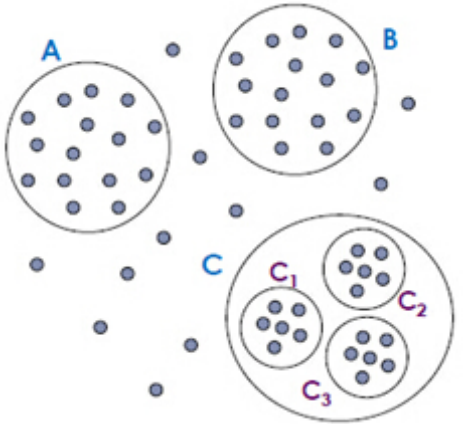
	Movie 1	Movie 2	Movie 3	Movie 4	Movie 5	Movie 6	Movie 7
Viewer 1	6	3	4	1		2	
Viewer 2	4			6	7		1
Viewer 3	1	2		2		7	3
Viewer 4	2	6		3		4	
Viewer 5			5	5	3	4	

Bi-Clustering

- Cheng-Church algorithm
  - The quality of the bi-cluster degrades (smaller volume, higher residue) due to the insertion of random data.
  - More about this algorithm
    - http://www.kemaleren.com/cheng-and-church.html

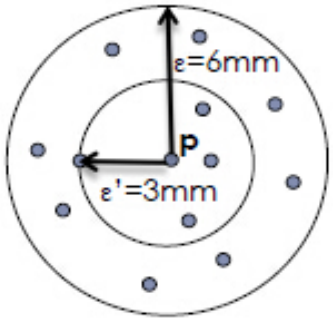
OPTICS

- Motivation:
  - → Very different local densities may be needed to reveal clusters in different regions.
  - → Clusters A, B, C_1, C_2, and C_3 cannot be detected using one global density parameter.
  - → A global density parameter can detect either A, B, C or C_1, C_2, and C_3.
- Solutions:
  - → OPTICS



OPTICS

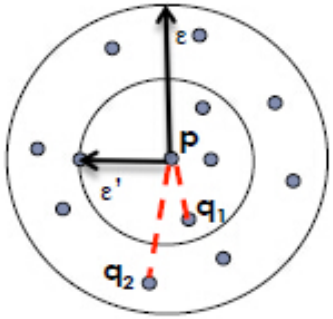
- The **core-distance** of an object is smallest the ε'that makes **{p}** a core object
  - → If **p** is not a core object, the core distance of **p** is **undefined**
  - → Example (ε, MinPts= 4)
    - ε' is the core distance of p
    - It is the distance between p and the fourth closest object



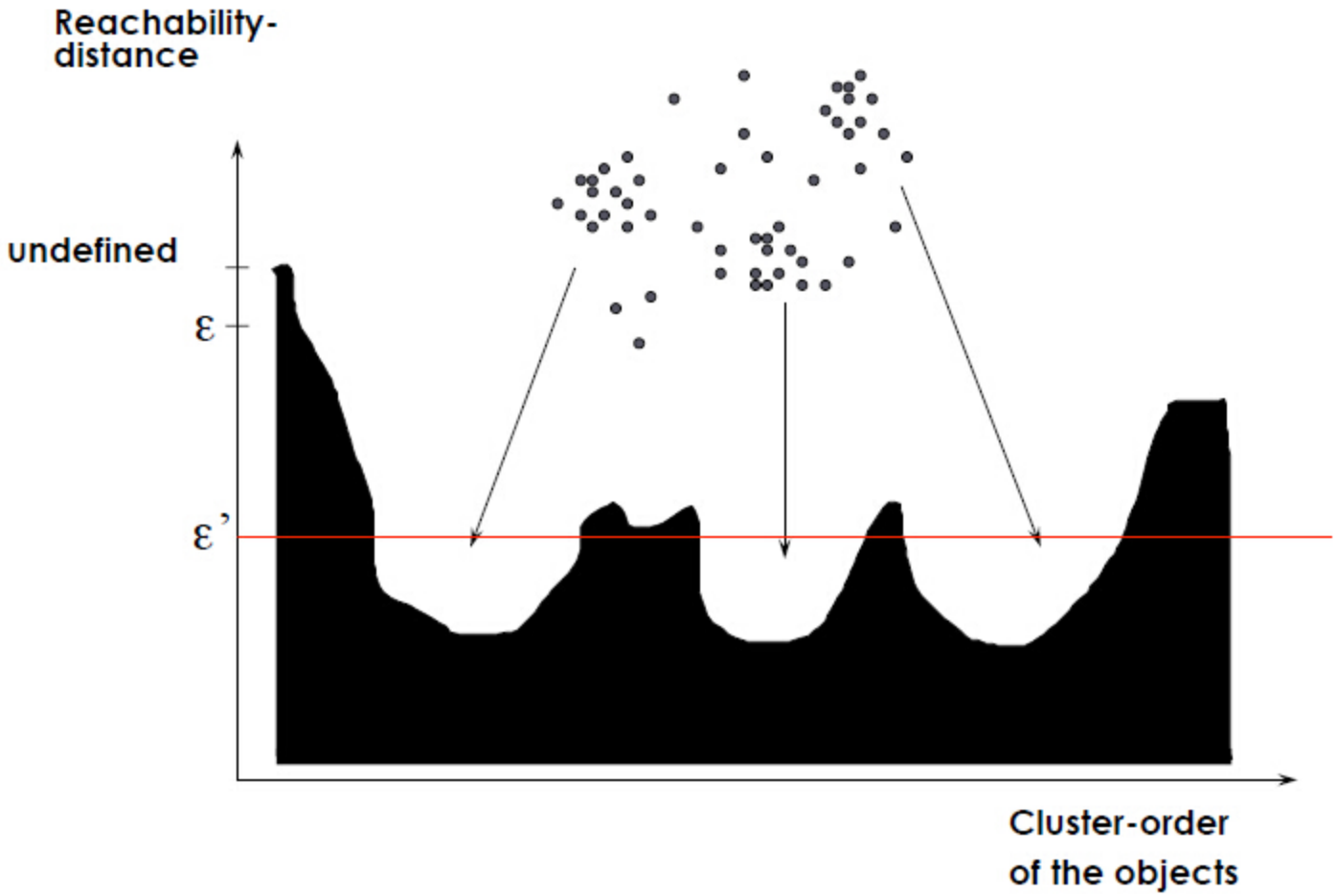
- The **reachability-distance** of an object **q** with respect to object to object **p** is:

Max(core-distance(p), Euclidian(p,q))

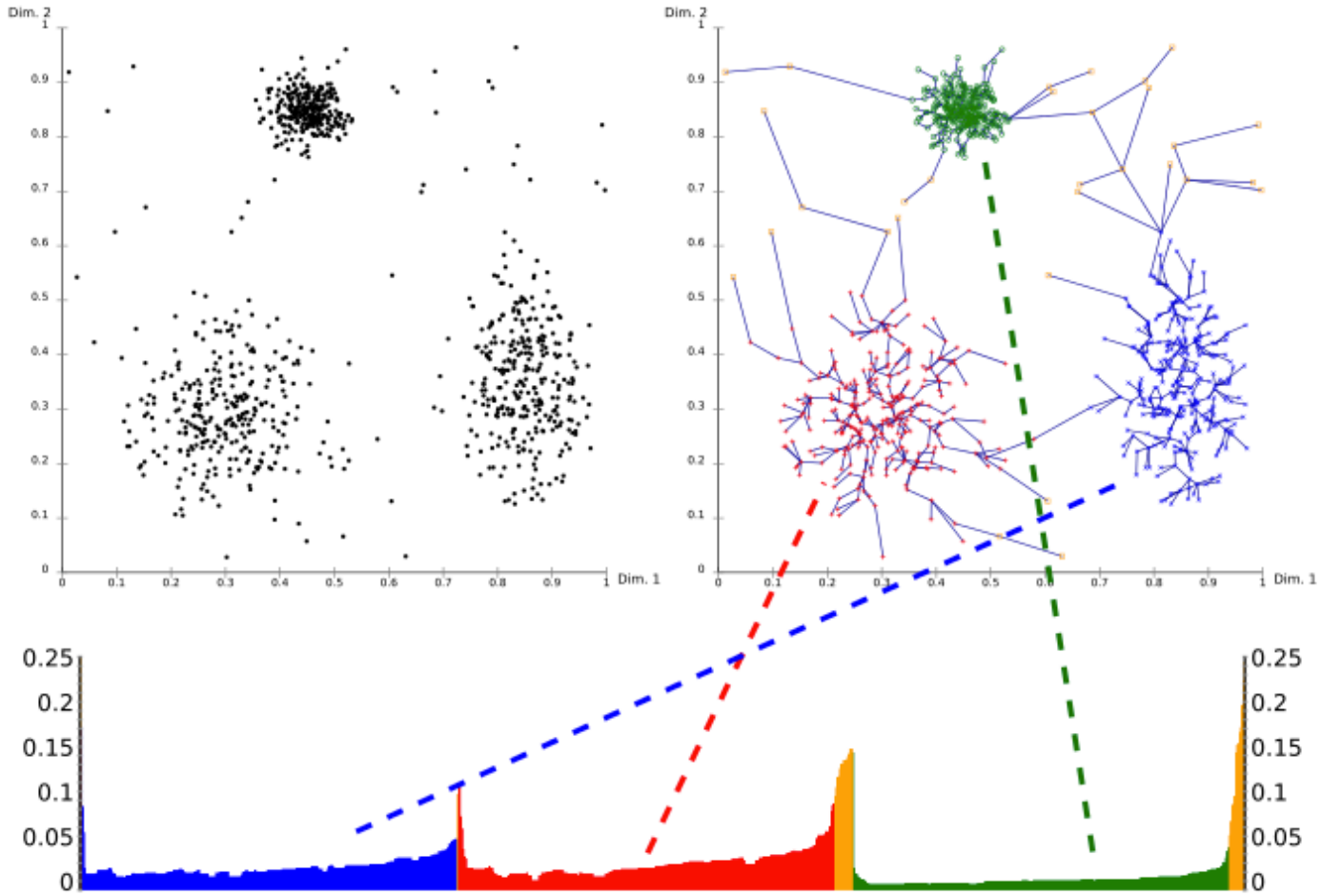
  - → **Example**
    - Reachability-distance(q_1,p)=core-distance(p)=ε'
    - Reachability-distance(q_2,p)=Euclidian(q_2,p)



OPTICS


Reachability-
distance
undefined
ε
ε'
Cluster-order
of the objects

OPTICS



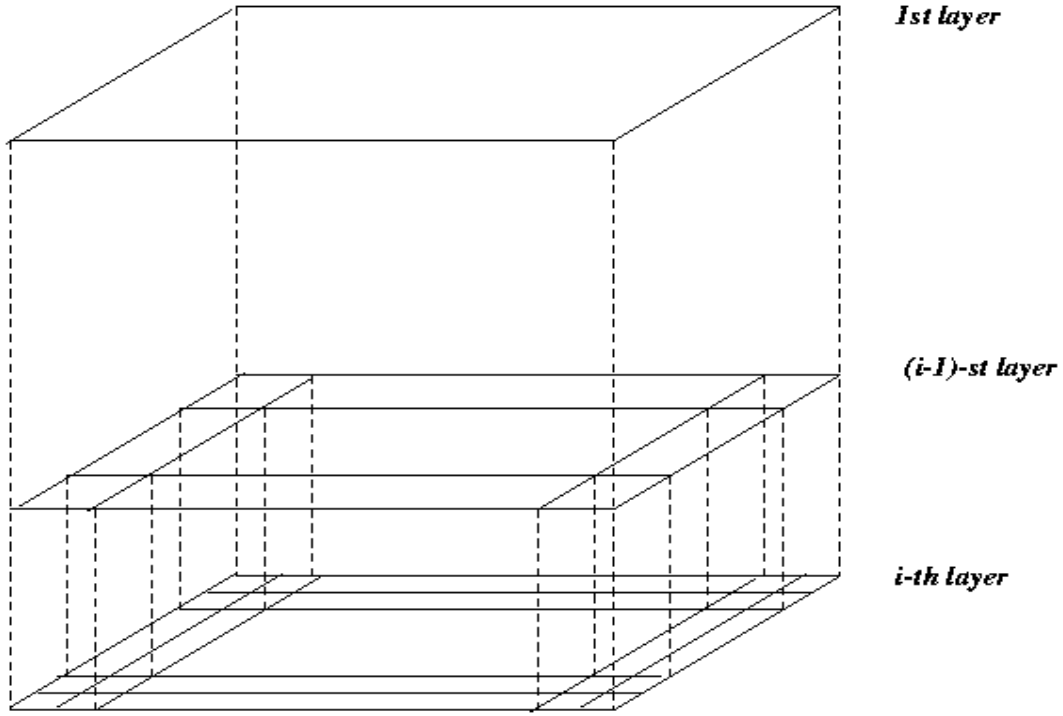
https://en.wikipedia.org/wiki/OPTICS_algorithm

Grid-based clustering

- Use multi-resolution grid data structure.
- Clustering complexity depends on the number of populated grid cells and not on the number of objects in the dataset.
- Methods:
  - STING

STING

- The spatial area is divided into rectangular cells.
- There are several levels of cells corresponding to different levels of resolution.



STING

- Each cell at a high level is partitioned into a number of smaller cells in the next lower level.
- Statistical info of each cell is calculated and stored beforehand and is used to answer queries.
- Parameters of higher level cells can be easily calculated from parameters of lower level cell.
    - count, mean, min, max
- Use a top-down approach to answer spatial data queries.

STING

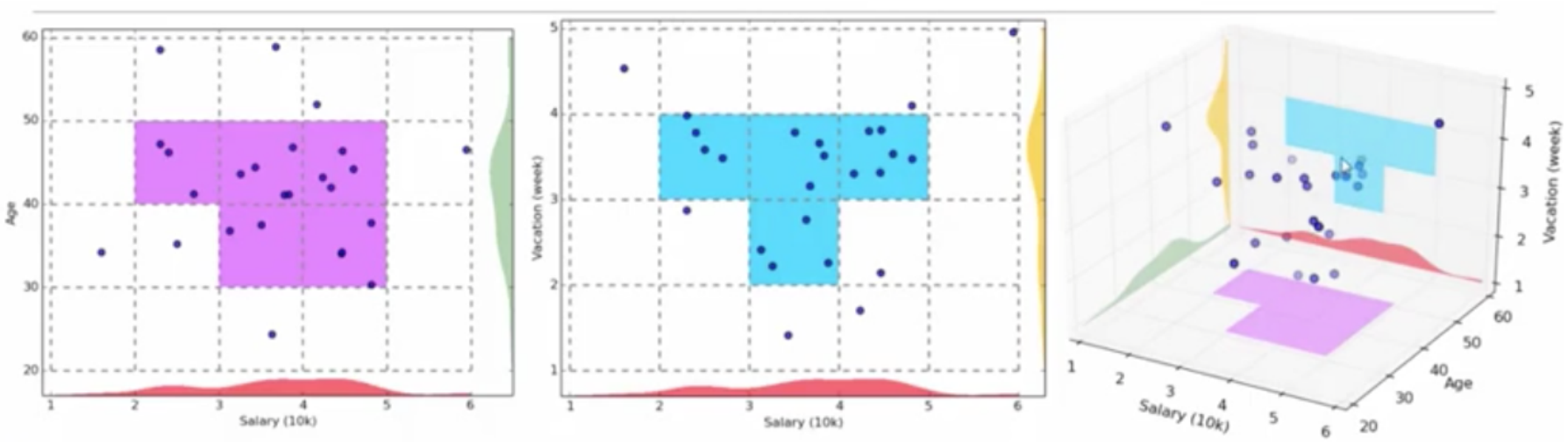
- Advantages:
  - Query-independent, easy to parallelize, incremental update.
  - O(K), where K is the number of grid cells at the lowest level.
- Disadvantages:
  - All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected.

Clique

- **CLIQUE (CLustering in QUEst)** was the first algorithm proposed for dimension **growth subspace clustering** in high-dimensional space
- Start at single-dimensional subspaces and grow upward to higher dimensional ones
- CLIQUE partitions each dimension like a grid structure and determines whether a cell is dense based on the number of points it contains
- CLIQUE is an integration of grid-based and density-based methods

Clique

- Partition the d-dimensional data space into non overlapping rectangular units (done in 1-D for each partition)
  - Identify dense units
  - A unit is dense if the fraction of total data points contained in it exceeds an input model parameter



Clique

- The property adapted by CLIQUE states:
  - → If a k-dimensional unit is dense, then so are its projections in (k-1) dimensional space
- Generate potential or candidate dense sense units in k-dimensional space from dense units found in (k-1) dimensional space
- The resulting space searched is much smaller than the original space
- The dense units are then examined to determine clusters