CHAMELEON: Hierarchical Clustering

CHAMELEON: Hierarchical Clustering Using Dynamic Modeling

- CHAMELEON: A graph partitioning approach (G. Karypis, E. H. Han, and V. Kumar, 1999)

- Measures the similarity between pair of clusters based on a dynamic model

- Cluster similarity is assessed based on how well connected objects within cluster and the proximity of the cluster.

- Two clusters are merged only if the interconnectivity and closeness (proximity) between two clusters are high relative to the internal interconnectivity of the clusters and closeness of items within the clusters

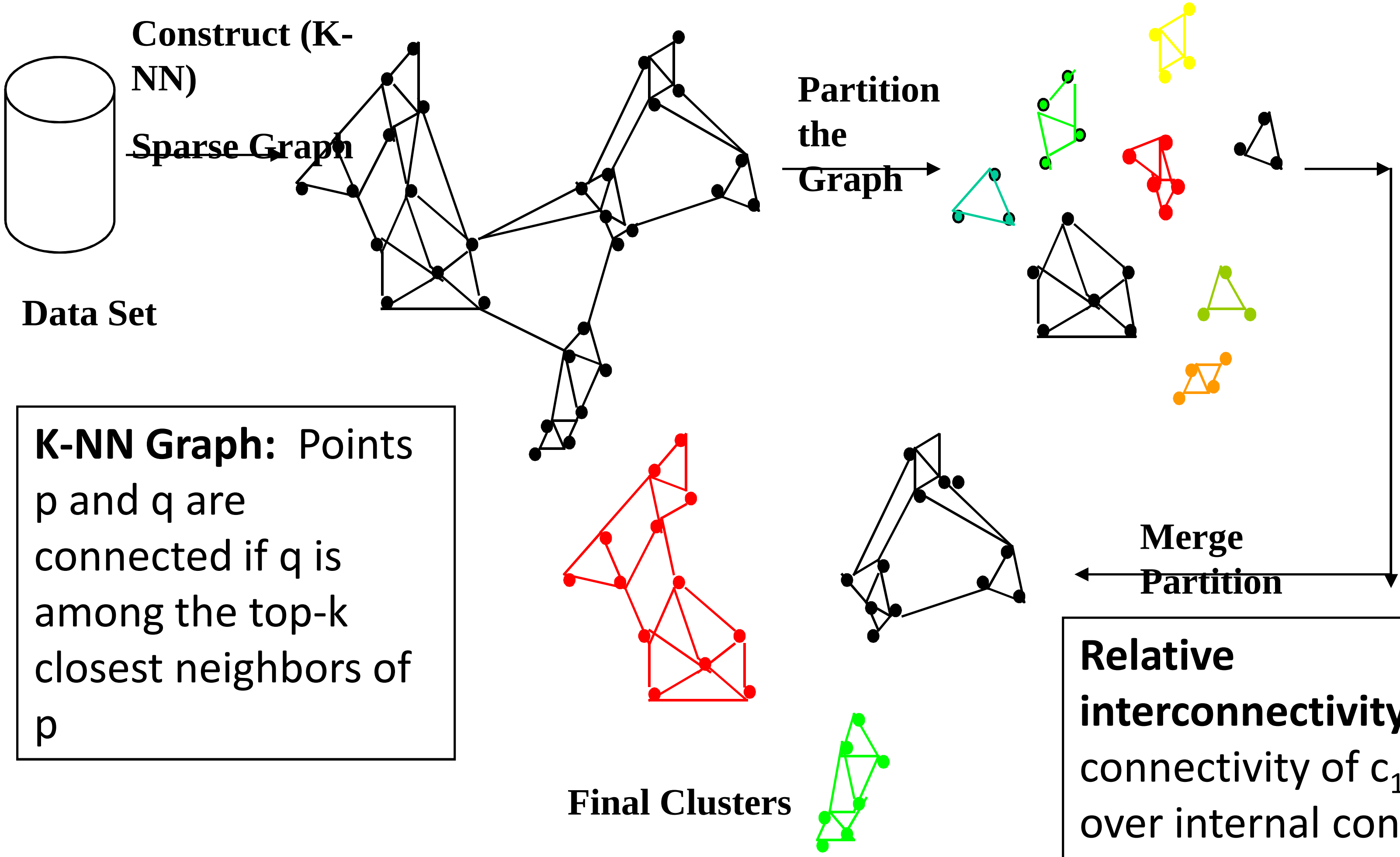
CHAMELEON: Hierarchical Clustering Using Dynamic Modeling

- *A graph-based, two-phase algorithm*

1. **Use a graph-partitioning algorithm:** Cluster objects into a large number of relatively small sub-clusters

2. **Use an agglomerative hierarchical clustering algorithm:** Find the genuine clusters by repeatedly combining these sub-clusters

Overall Framework of CHAMELEON



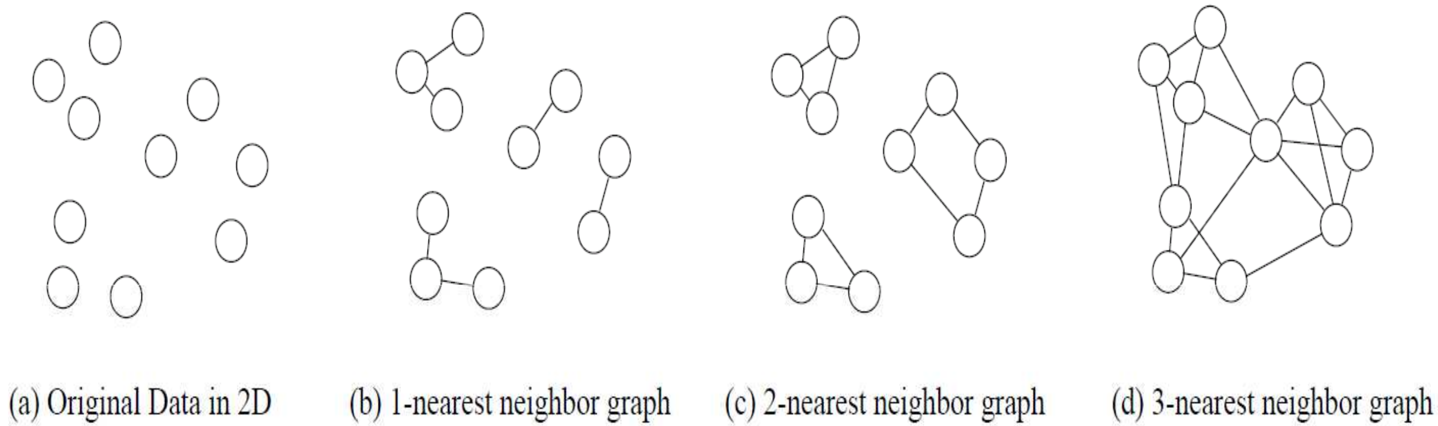
K-NN Graph: Points p and q are connected if q is among the top-k closest neighbors of p

Relative interconnectivity: connectivity of c_1 and c_2 over internal connectivity

Relative closeness: closeness of c_1 and c_2 over internal closeness

KNN Graphs and Interconnectivity

- K-nearest neighbor (KNN) graphs from an original data in 2D:



(a) Original Data in 2D (b) 1-nearest neighbor graph (c) 2-nearest neighbor graph (d) 3-nearest neighbor graph

- $EC_{\{Ci\ ,Cj\ \}}$: The absolute interconnectivity between C_i and C_j
  - The sum of the weight of the edges that connect vertices in C_i to vertices in C_j

KNN Graphs and Interconnectivity

- **Internal interconnectivity of a cluster C_i** : The size of its min-cut bisector EC_{Ci} (i.e., the weighted sum of edges that partition the graph into two roughly equal parts)

- Relative Interconnectivity (RI): $EC_{\{Ci ,Cj \}}$:The absolute interconnectivity between C_i and C_j normalized with respect to the internal interconnectivity of two clusters C_i and C_j

$$RI(C_i, C_j) = \frac{|EC_{\{C_i, C_j\}}|}{\frac{|EC_{C_i}| + |EC_{C_j}|}{2}}$$

Relative Closeness & Merge of Sub-Clusters

- **Relative closeness** between a pair of clusters C_i and C_j : The absolute closeness between C_i and C_j normalized w.r.t. the internal closeness of the two clusters C_i and C_j

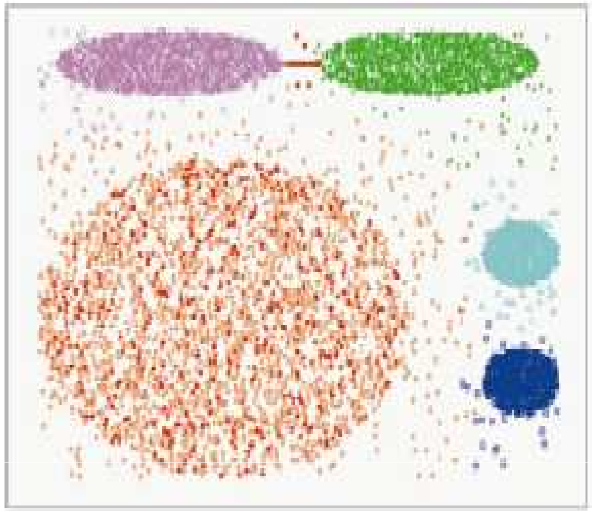
$$RC(C_i, C_j) = \frac{\overline{S}_{EC_{\{C_i, C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|}\overline{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|}\overline{S}_{EC_{C_j}}}$$

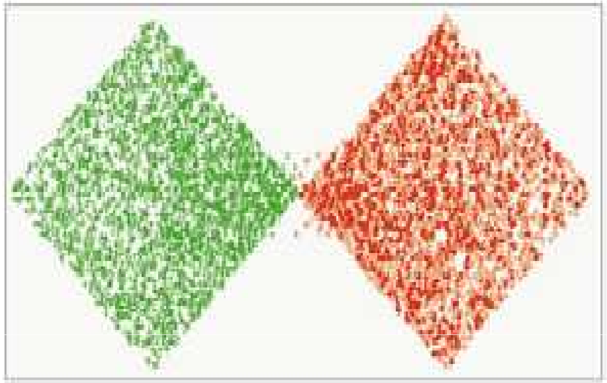
  - where $\overline{S}_{EC_{C_i}}$ and $\overline{S}_{EC_{C_j}}$ are the average weights of the edges that belong to the min-cut bisector of clusters C_i and C_j , respectively, and $\overline{S}_{EC_{\{C_i, C_j\}}}$ is the average weight of the edges that connect vertices in C_i to vertices in C_j

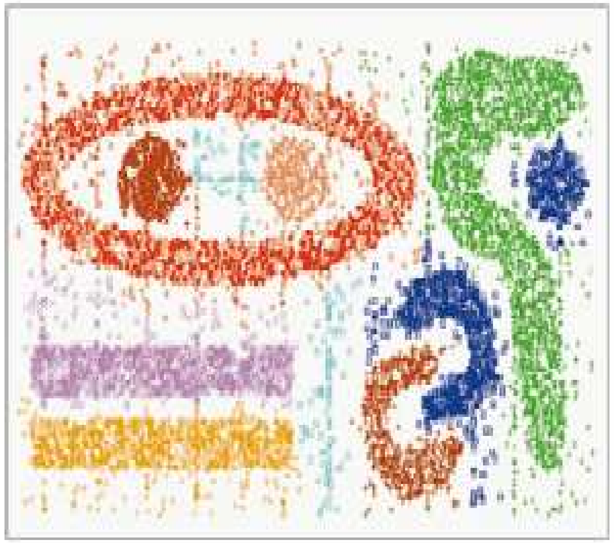
Relative Closeness & Merge of Sub-Clusters

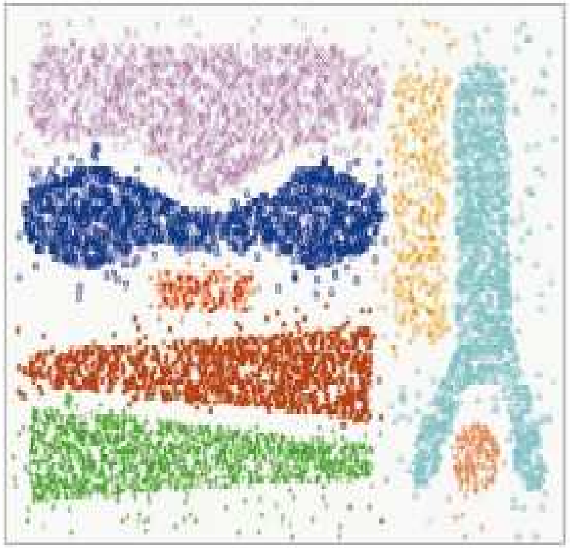
- ***Merge Sub-Clusters:***
  - Merges only those pairs of clusters whose RI and RC are both above some user-specified thresholds
  - Merge those maximizing the function that combines RI and RC

CHAMELEON: Clustering Complex Objects









Grid Based Methods

- **Data driven methods:** Partition the set of objects and adapt distribution of objects in embedding space.
- **Space driven methods:** Partitioning the embedded space into cells independent of the distribution of input objects.
- **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
  - Advantage is fast processing time, independent of no of data objects but dependent on no of cells in each dimension

Grid Based Methods

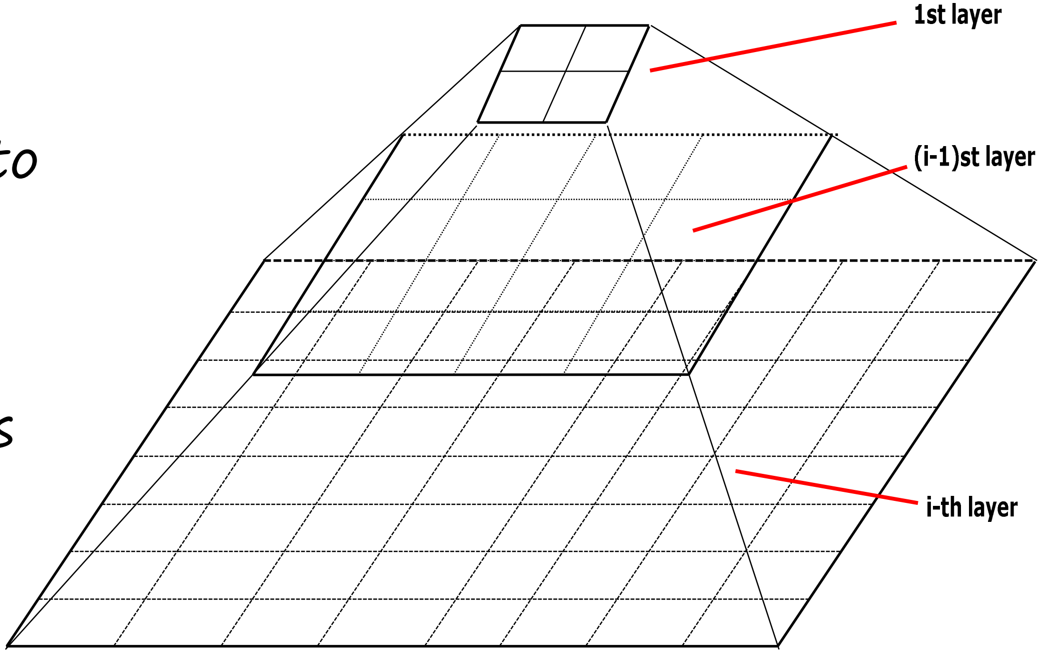
- ***Features and challenges of a typical grid-based algorithm***
  - Efficiency and scalability: # of cells << # of data points
  - Uniformity: Uniform, hard to handle highly irregular data distributions
  - Locality: Limited by predefined cell sizes, borders, and the density threshold
  - Curse of dimensionality: Hard to cluster high-dimensional data

Grid-Based Clustering Methods

- 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

STING: A Statistical Information Grid Approach

- STING (Statistical Information Grid): 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 is partitioned to form a number of cells at the next lower level.
- Top-down Approach



Query Processing in STING and Its Analysis

- Statistical information of each cell is calculated and stored beforehand and is used for query processing and data analysis
- 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.
- The type of distribution of a higher-cell is computed based on the distribution of lower level cell in conjunction with filtering process.

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
  - For each cell in current layer calculate the **confidence** interval reflecting the **cell's relevancy to the query.**
  - Only children of likely relevant cells are recursively explored
  - Repeat this process until the bottom layer is reached
  - If query specification is met, the regions of relevant cells that satisfy the query is retuned

Algorithm

1. Determine a layer to begin with.
2. For each cell of this layer, we calculate the confidence interval (or estimated range) of probability that this cell is relevant to the query.
3. From the interval calculated above, we label the cell as relevant or not relevant.
4. If this layer is the bottom layer, go to Step 6; otherwise, go to Step 5.
5. We go down the hierarchy structure by one level. Go to Step 2 for those cells that form the relevant cells of the higher level layer.

Algorithm

6. If the specification of the query is met, go to Step 8; otherwise, go to Step 7.
7. Retrieve those data fall into the relevant cells and do further processing. Return the result that meet the requirement of the query. Go to Step 9.
8. Find the regions of relevant cells. Return those regions that meet the requirement of the query. Go to Step 9.
9. Stop.

Query Processing in STING and Its Analysis

- *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)
  - Quality depends on the granularity of the lowest level of grid structure, if fine cost of processing increases else decreases.
- *Disadvantages*
  - Its probabilistic nature may imply a loss of accuracy in query processing

CLIQUE: Grid-Based Subspace Clustering

- **CLIQUE** (Clustering In QUEst) (Agrawal, Gehrke, Gunopulos, Raghavan: SIGMOD'98)
- A data objects with 10 attributes may be irrelevant,the values of attributes may vary
- Search for clusters with different subspaces of data.
- Eg: consider a medical record containing information like personal, numerous symptoms, conditions and family history.
- For nontrivial group of patients even most of the attributes strongly disagree.
- Diffcult to find the culster in entire dataspace so find cluster of similar patients at lower level using symptoms

CLIQUE: Grid-Based Subspace Clustering

- *CLIQUE is a* ***density-based*** *and* ***grid-based*** *subspace clustering algorithm*
- ***Grid-based:*** *It discretizes the* ***data space*** *through a grid of cells 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

CLIQUE: Grid-Based Subspace Clustering

- ***Subspace clustering**: A subspace cluster is a set of neighboring dense cells in an arbitrary subspace.*
- *It automatically identifies subspaces of a high dimensional data space that allow better clustering than original space using the Apriori principle.*
  - A k-dimension cell c(k>1) can have atleast l points only if every(k-1) dimensional subspace has atleast l points.(Here it is expected that if some thing is dense in higher dimensional space it cant be sparse in lower dimension state.)

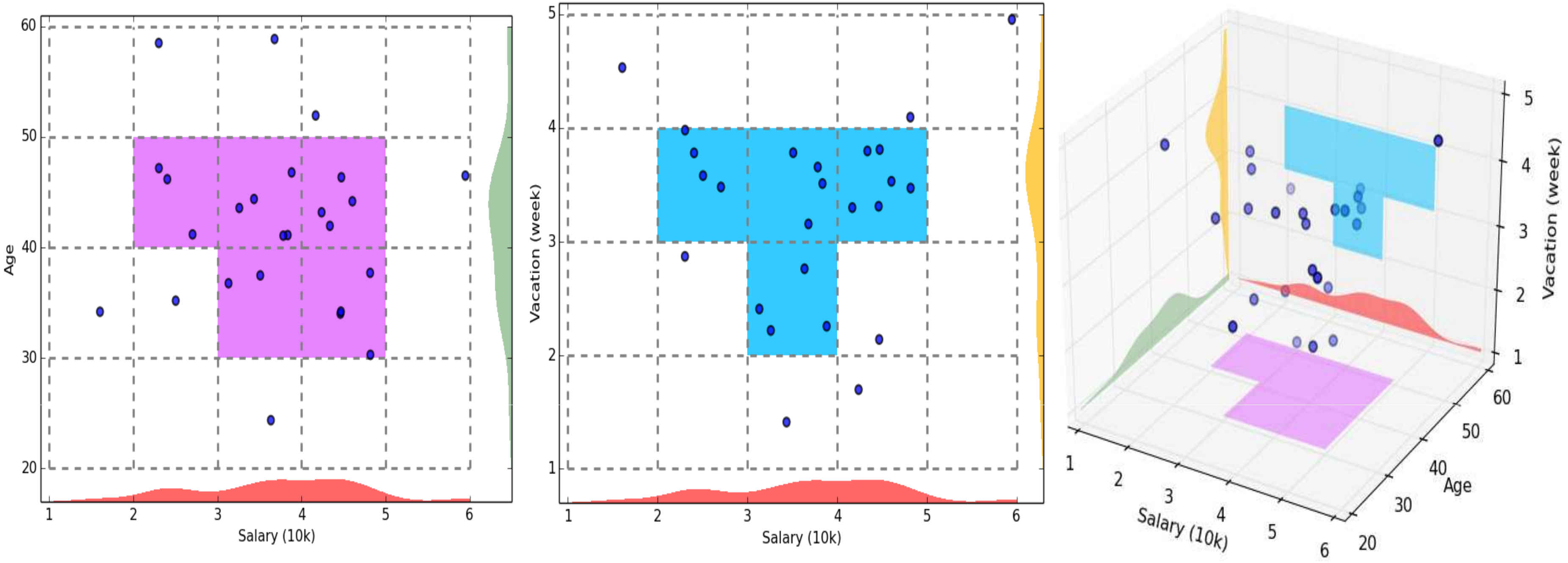
CLIQUE: SubSpace Clustering with Aprori Pruning

- 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

CLIQUE: SubSpace Clustering with Aprori Pruning



CLIQUE: SubSpace Clustering with Aprori Pruning

- CLIQUE then iteratively joins two k-dimensional dense cells, c1 and c2, in subspaces (Di1 ,....,Dik and Dj1 , ... ,Djk)

- if (Di1 ,....,Dik AND Dj1 , . . . , DJK) c1 and c2 share the same intervals in those dimensions. The join operation generates a new (k+1) dimensional candidate cell c in space (Di1 ,....,Dik-1 ,Dik ,Djk)

- CLIQUE checks whether the number of points in c passes the density threshold. The iteration terminates when no candidates can be generated or no candidate cells are dense.

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

Major Steps of the CLIQUE Algorithm

- *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