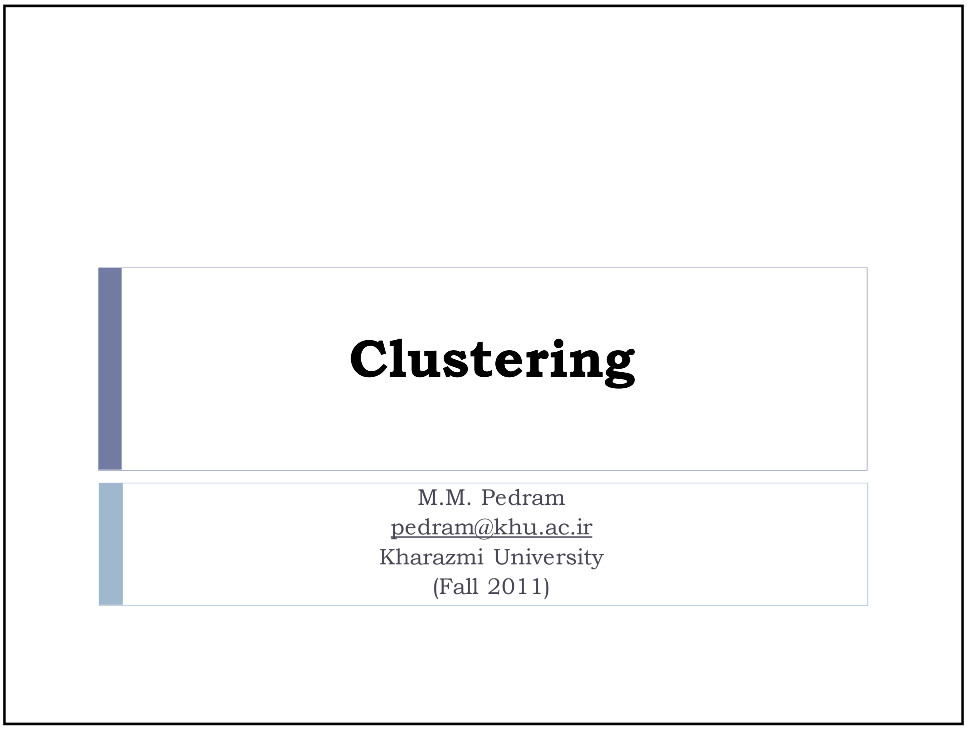
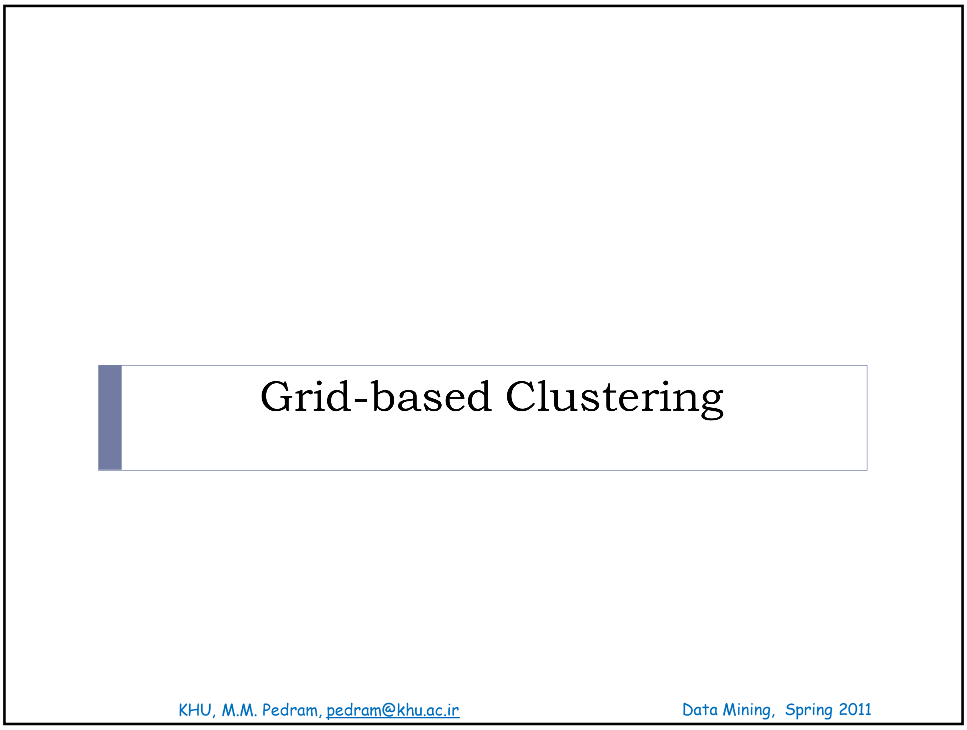


Clustering

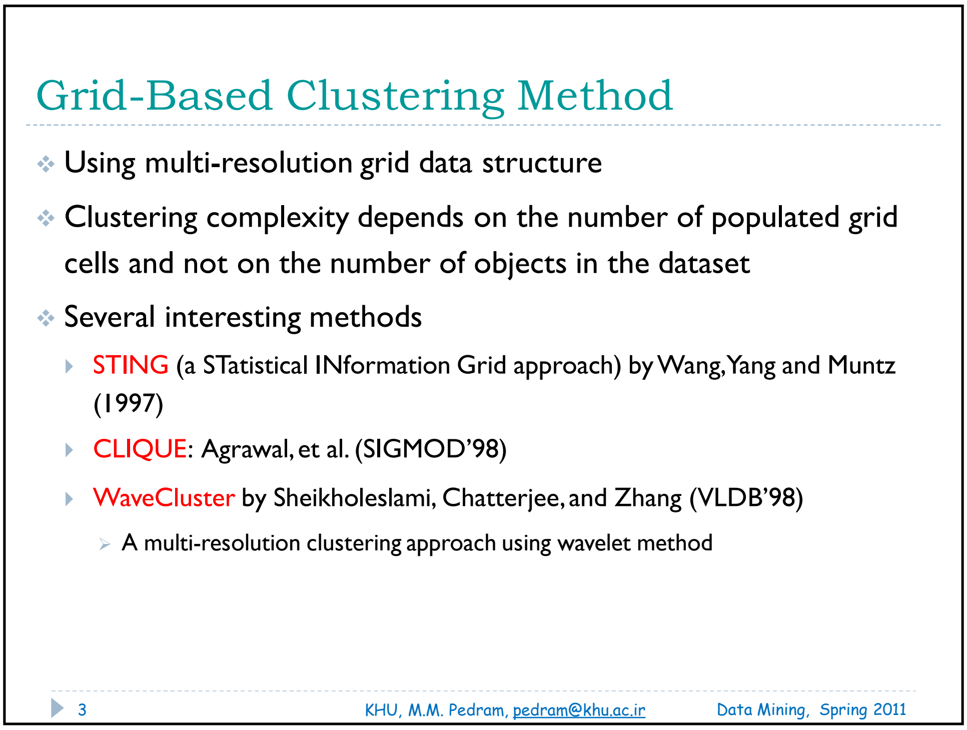
M.M. Pedram
pedram@khu.ac.ir
Kharazmi University
(Fall 2011)

1



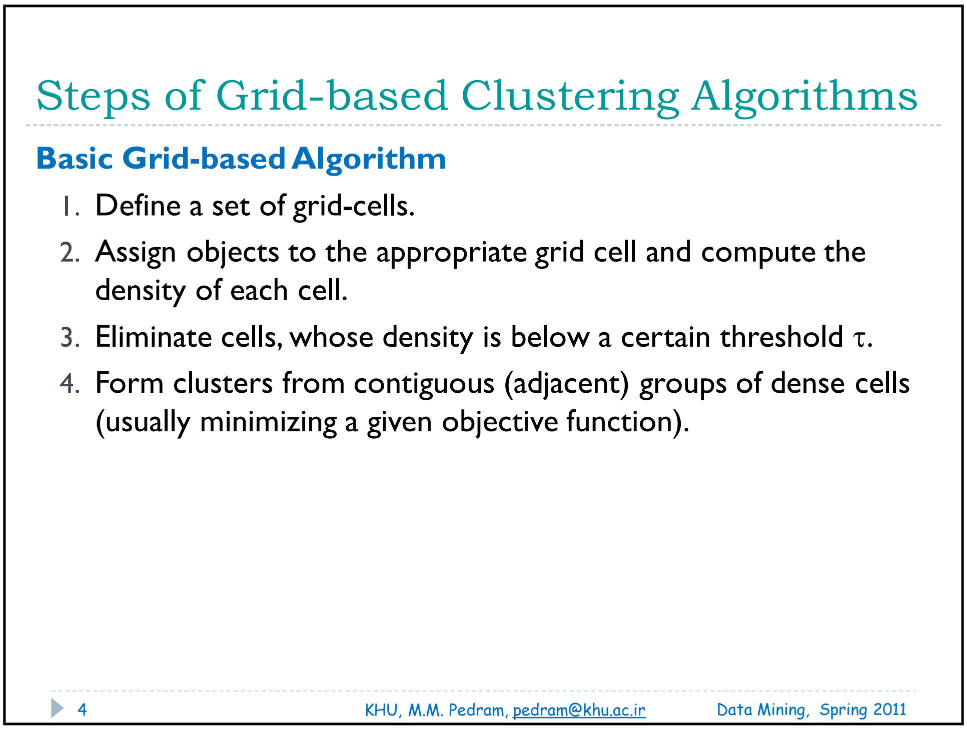
Grid-based Clustering

2



Grid-Based Clustering Method

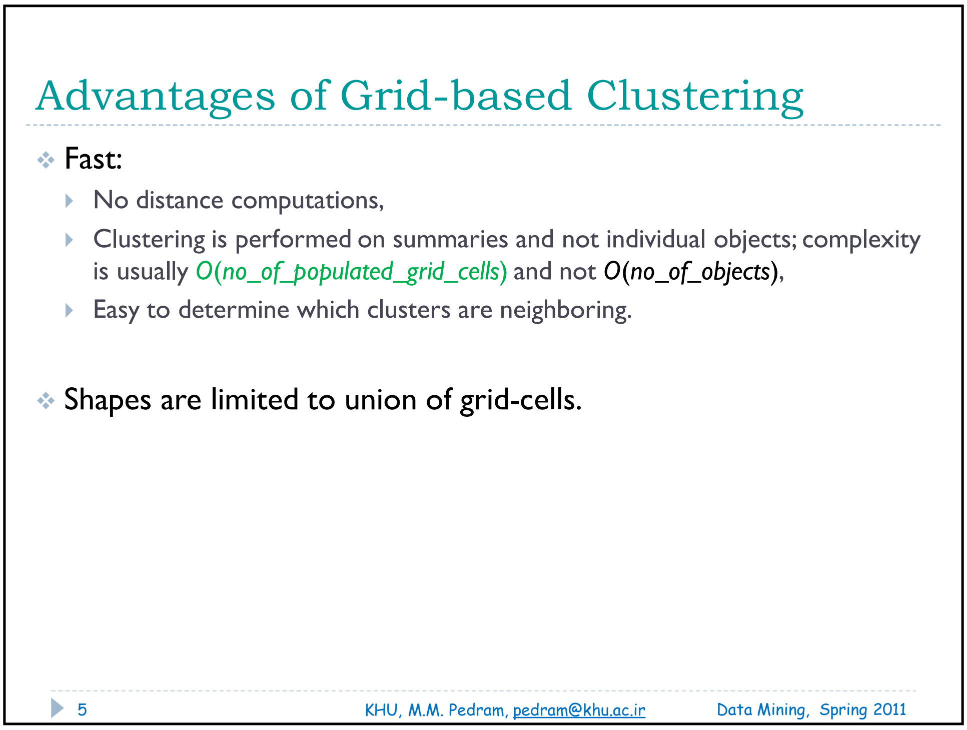
- Using multi-resolution grid data structure
- Clustering complexity depends on the number of populated grid cells and not on the number of objects in the dataset
- Several interesting methods
  - STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
  - CLIQUE: Agrawal, et al. (SIGMOD'98)
  - WaveCluster by Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
    - A multi-resolution clustering approach using wavelet method



Steps of Grid-based Clustering Algorithms

Basic Grid-based Algorithm

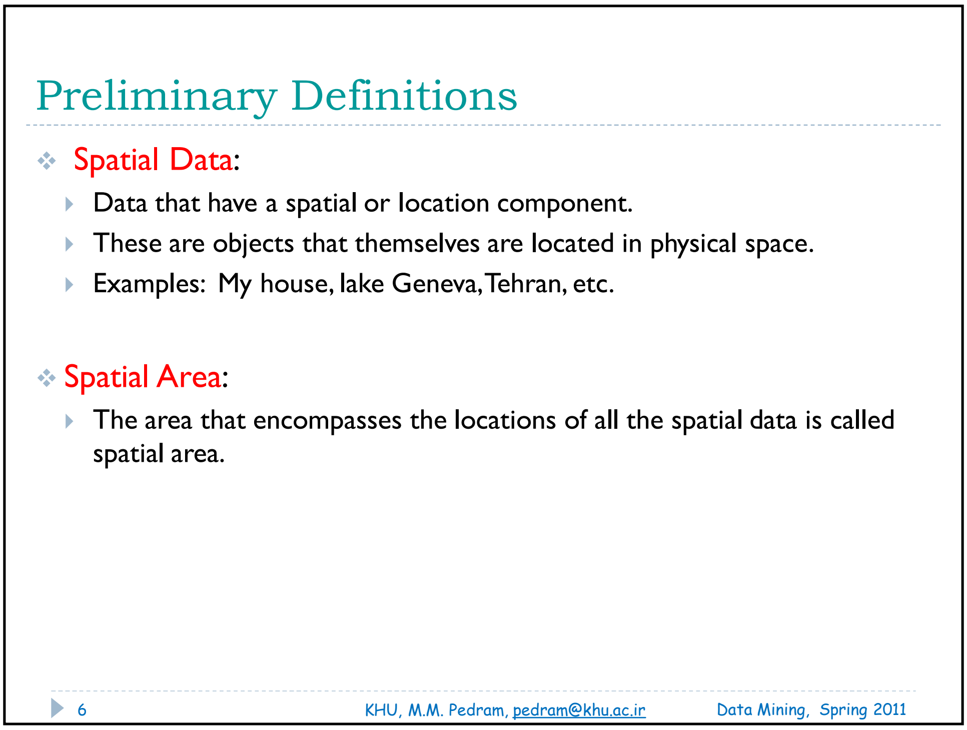
1. Define a set of grid-cells.
2. Assign objects to the appropriate grid cell and compute the density of each cell.
3. Eliminate cells, whose density is below a certain threshold τ.
4. Form clusters from contiguous (adjacent) groups of dense cells (usually minimizing a given objective function).



Advantages of Grid-based Clustering

- Fast:
  - No distance computations,
  - Clustering is performed on summaries and not individual objects; complexity is usually *O(no_of_populated_grid_cells)* and not *O(no_of_objects)*,
  - Easy to determine which clusters are neighboring.

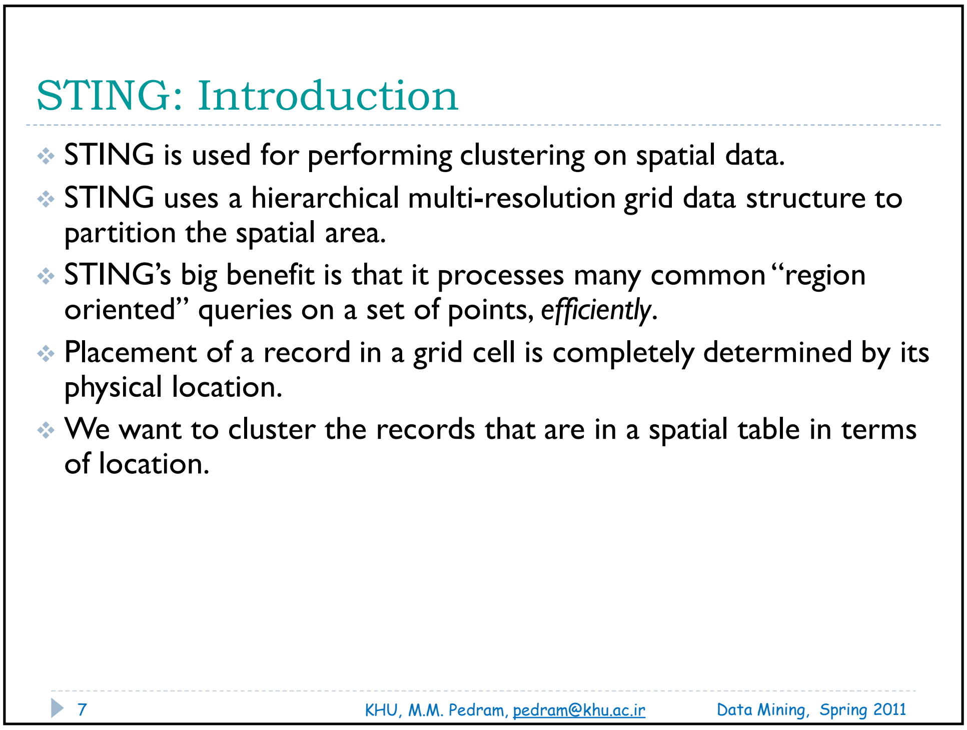
- Shapes are limited to union of grid-cells.



Preliminary Definitions

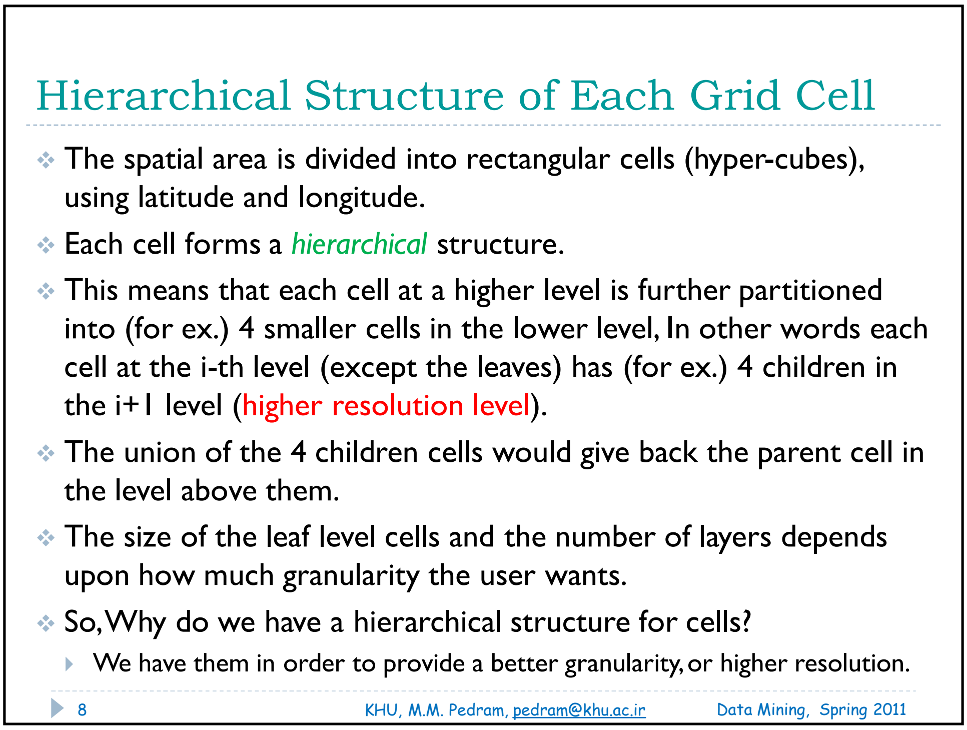
- Spatial Data:
  - Data that have a spatial or location component.
  - These are objects that themselves are located in physical space.
  - Examples: My house, lake Geneva, Tehran, etc.

- Spatial Area:
  - The area that encompasses the locations of all the spatial data is called spatial area.



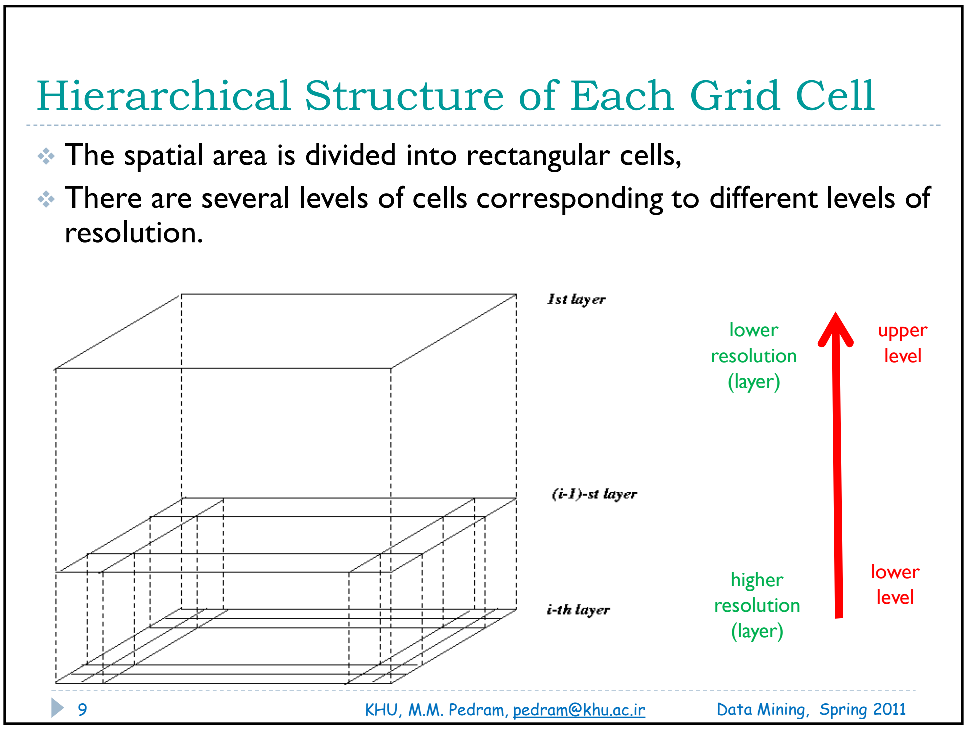
STING: Introduction

- STING is used for performing clustering on spatial data.
- STING uses a hierarchical multi-resolution grid data structure to partition the spatial area.
- STING's big benefit is that it processes many common "region oriented" queries on a set of points, *efficiently*.
- Placement of a record in a grid cell is completely determined by its physical location.
- We want to cluster the records that are in a spatial table in terms of location.



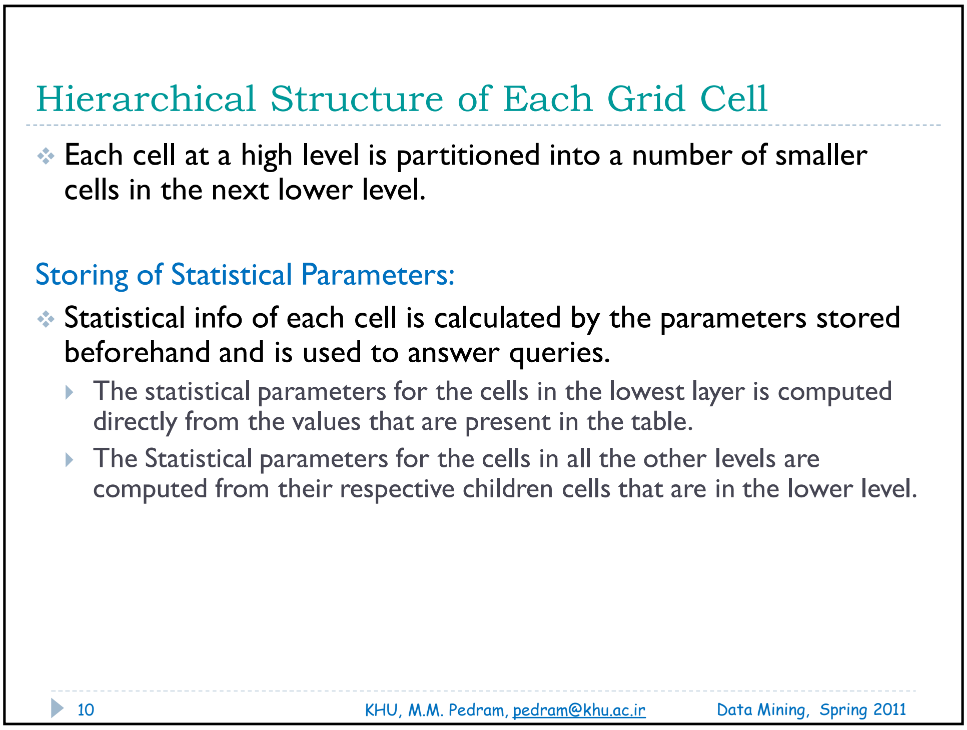
Hierarchical Structure of Each Grid Cell

- The spatial area is divided into rectangular cells (hyper-cubes), using latitude and longitude.
- Each cell forms a *hierarchical* structure.
- This means that each cell at a higher level is further partitioned into (for ex.) 4 smaller cells in the lower level, In other words each cell at the i-th level (except the leaves) has (for ex.) 4 children in the i+1 level (higher resolution level).
- The union of the 4 children cells would give back the parent cell in the level above them.
- The size of the leaf level cells and the number of layers depends upon how much granularity the user wants.
- So, Why do we have a hierarchical structure for cells?
  - We have them in order to provide a better granularity, or higher resolution.



Hierarchical Structure of Each Grid Cell

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

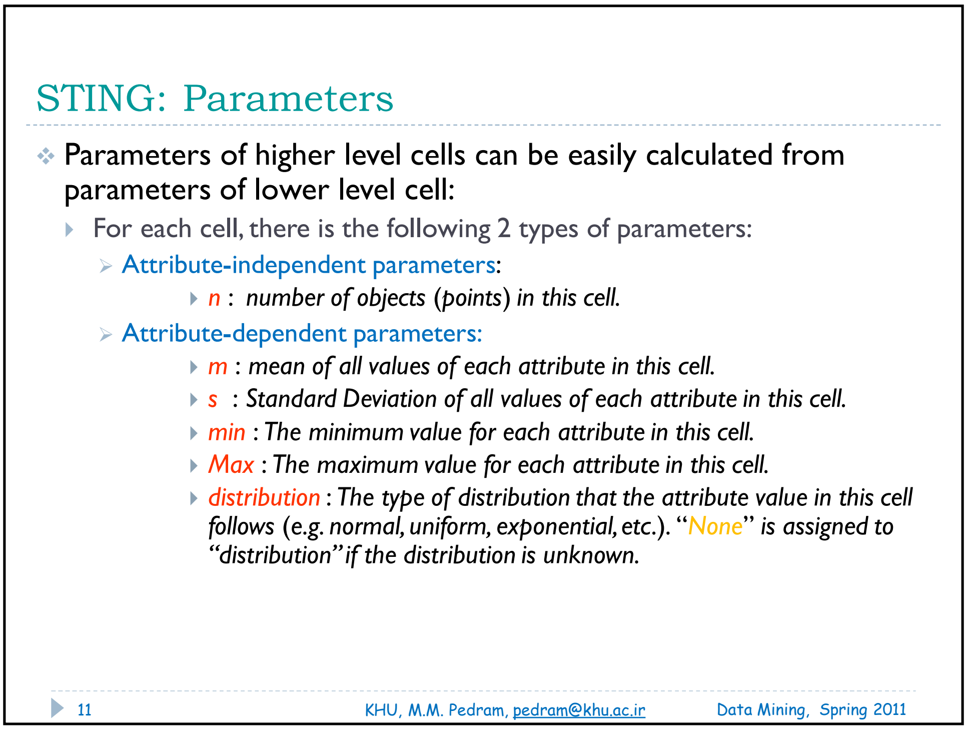


Hierarchical Structure of Each Grid Cell

- Each cell at a high level is partitioned into a number of smaller cells in the next lower level.

Storing of Statistical Parameters:

- Statistical info of each cell is calculated by the parameters stored beforehand and is used to answer queries.
  - The statistical parameters for the cells in the lowest layer is computed directly from the values that are present in the table.
  - The Statistical parameters for the cells in all the other levels are computed from their respective children cells that are in the lower level.

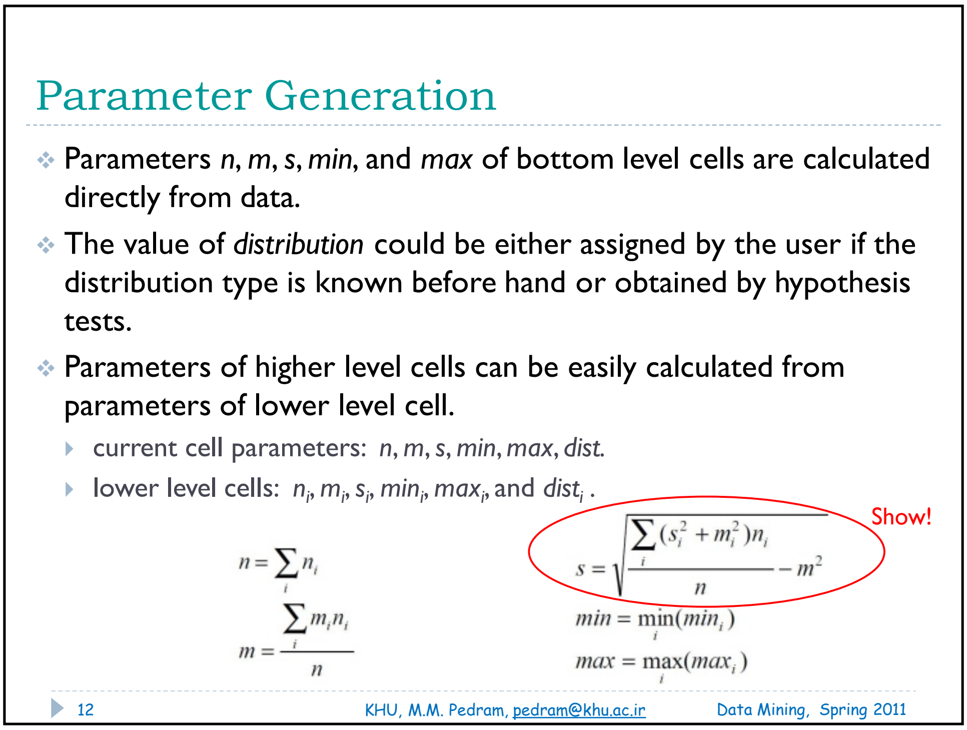


STING: Parameters

- Parameters of higher level cells can be easily calculated from parameters of lower level cell:
  - For each cell, there is the following 2 types of parameters:
    - Attribute-independent parameters:
      - *n : number of objects (points) in this cell.*
    - Attribute-dependent parameters:
      - *m : mean of all values of each attribute in this cell.*
      - *s : Standard Deviation of all values of each attribute in this cell.*
      - *min : The minimum value for each attribute in this cell.*
      - *Max : The maximum value for each attribute in this cell.*
      - *distribution : The type of distribution that the attribute value in this cell follows (e.g. normal, uniform, exponential, etc.). "None" is assigned to "distribution" if the distribution is unknown.*

11 KHU, M.M. Pedram, pedram@khu.ac.ir Data Mining, Spring 2011

11



Parameter Generation

- Parameters *n*, *m*, *s*, *min*, and *max* of bottom level cells are calculated directly from data.
- The value of *distribution* could be either assigned by the user if the distribution type is known before hand or obtained by hypothesis tests.
- Parameters of higher level cells can be easily calculated from parameters of lower level cell.
  - current cell parameters: *n, m, s, min, max, dist.*
  - lower level cells: n_i, m_i, s_i, min_i, max_i, and $dist_i$.

$$n = \sum_i n_i$$

$$m = \frac{\sum_i m_i n_i}{n}$$

$$s = \sqrt{\frac{\sum_i (s_i^2 + m_i^2) n_i}{n} - m^2}$$

Show!

$$min = \min_i(min_i)$$

$$max = \max_i(max_i)$$

12 KHU, M.M. Pedram, pedram@khu.ac.ir Data Mining, Spring 2011

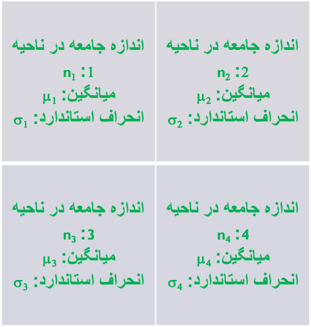
12

Parameter Generation

❖ فرض کنید در هر مربع i از چهار مربع (ناحیه) نشان داده شده در شکل زیر (سمت چپ)، اندازه جامعه با n_i، و پارامترهای میانگین و واریانس به ترتیب با μ_i و σ_i^2 نشان داده شوند.

□ به عنوان مثال، تعداد خانه‌های یک ناحیه

□ به عنوان مثال، میانگین قیمت خانه‌های آن ناحیه



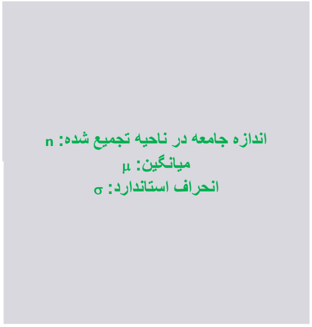
اندازه جامعه در ناحیه
n_1 :1
میانگین: μ_1
انحراف استاندارد: σ_1

اندازه جامعه در ناحیه
n_2 :2
میانگین: μ_2
انحراف استاندارد: σ_2

اندازه جامعه در ناحیه
n_3 :3
میانگین: μ_3
انحراف استاندارد: σ_3

اندازه جامعه در ناحیه
n_4 :4
میانگین: μ_4
انحراف استاندارد: σ_4

نواحی چهارگانه در سطح k+1



اندازه جامعه در ناحیه تجمیع شده: n
میانگین: μ
انحراف استاندارد: σ

ناحیه حاصل از تجمیع نواحی ۱ تا ۴ در سطح k

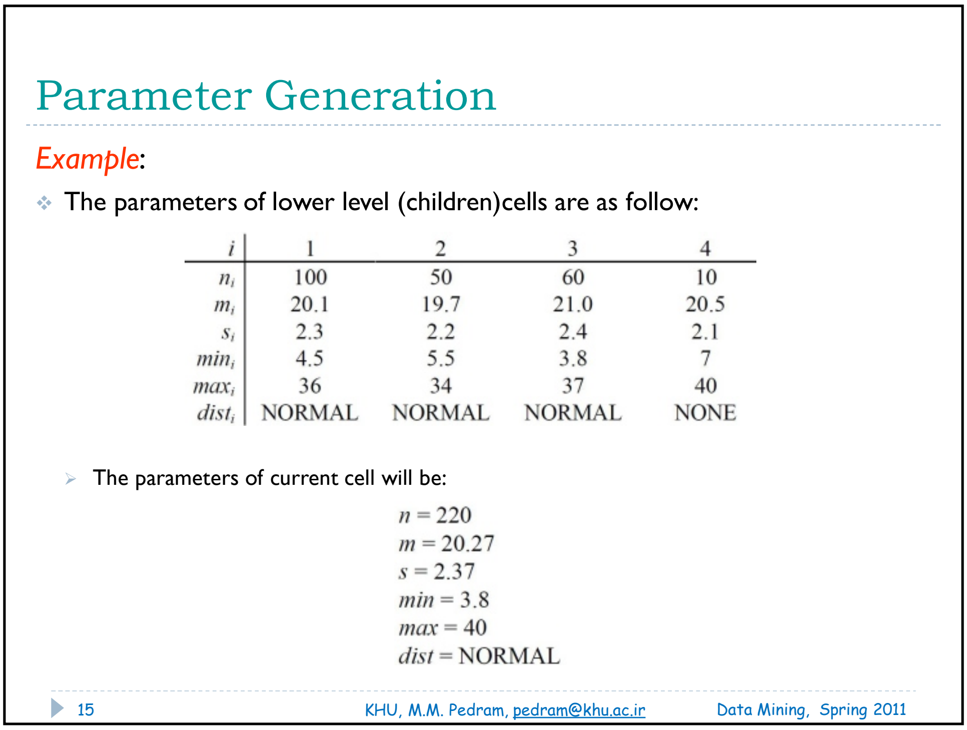
Parameter Generation

❖ The determination of *dist* for a parent cell:

- First set *dist* as the distribution type followed by most points in this cell. This can be done by examining $dist_i$ and n_i.
- Then estimate *confl* : the number of points that conflict with the distribution determined by *dist*, *m*, and *s* according to the following rule:

1. If $dist_i \neq dist$, $m_i \approx m$ and $s_i \approx s$, then *confl* is increased by an amount of n_i;
2. If $dist_i \neq dist$, but either $m_i \approx m$ or $s_i \approx s$ is not satisfied, then set *confl* to *n* (This enforces *dist* will be set to NONE later);
3. If $dist_i = dist$, $m_i \approx m$ and $s_i \approx s$, then *confl* is increased by 0;
4. If $dist_i = dist$, but either $m_i \approx m$ or $s_i \approx s$ is not satisfied, then *confl* is set to *n*.

Finally, if $confl/_n$ is greater than a threshold τ (a small constant, say 0.05, which is set before the hierarchical structure is built), then *dist* = NONE; otherwise, we keep the original type.



Parameter Generation

Example:

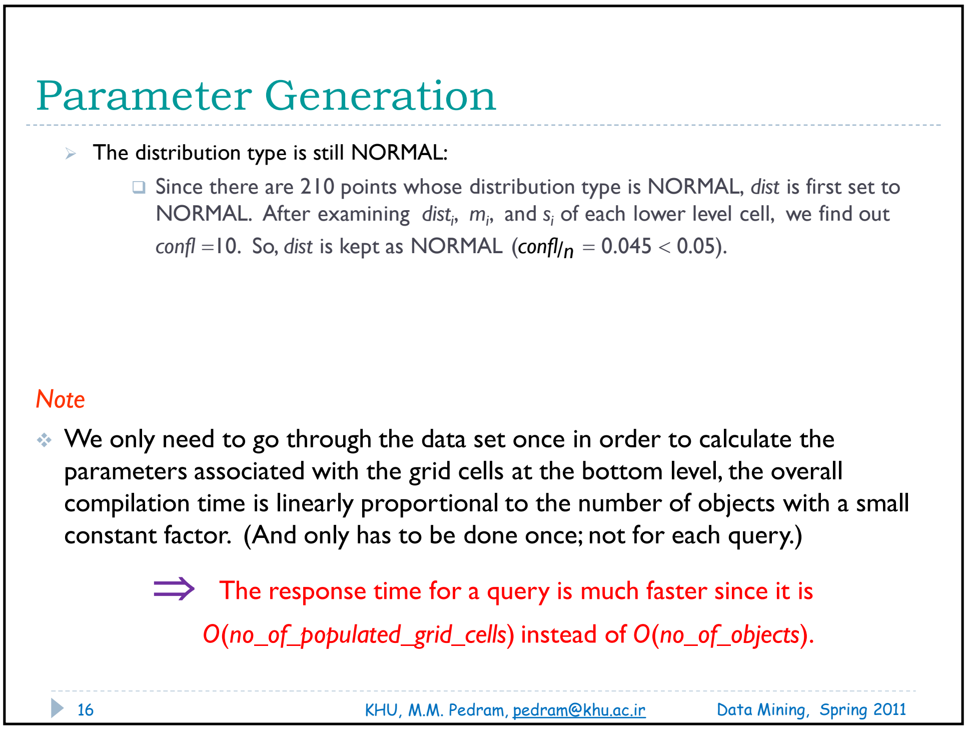
- The parameters of lower level (children)cells are as follow:

i	1	2	3	4
n_i	100	50	60	10
m_i	20.1	19.7	21.0	20.5
s_i	2.3	2.2	2.4	2.1
min_i	4.5	5.5	3.8	7
max_i	36	34	37	40
$dist_i$	NORMAL	NORMAL	NORMAL	NONE

- The parameters of current cell will be:

$$n = 220$$
$$m = 20.27$$
$$s = 2.37$$
$$min = 3.8$$
$$max = 40$$
$$dist = \text{NORMAL}$$

15



Parameter Generation

- The distribution type is still NORMAL:
  - Since there are 210 points whose distribution type is NORMAL, *dist* is first set to NORMAL. After examining $dist_i$, m_i, and s_i of each lower level cell, we find out *confl* =10. So, *dist* is kept as NORMAL ($confl/n = 0.045 < 0.05$).

Note

- We only need to go through the data set once in order to calculate the parameters associated with the grid cells at the bottom level, the overall compilation time is linearly proportional to the number of objects with a small constant factor. (And only has to be done once; not for each query.)

⇒ The response time for a query is much faster since it is *O*(*no_of_populated_grid_cells*) instead of *O*(*no_of_objects*).

16

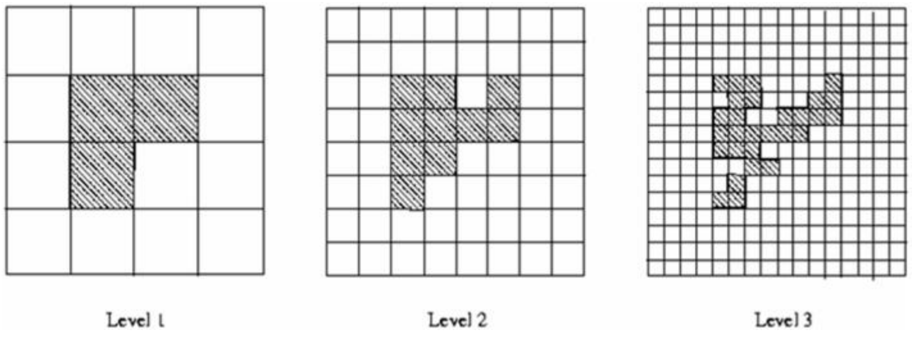
STING: Query Processing

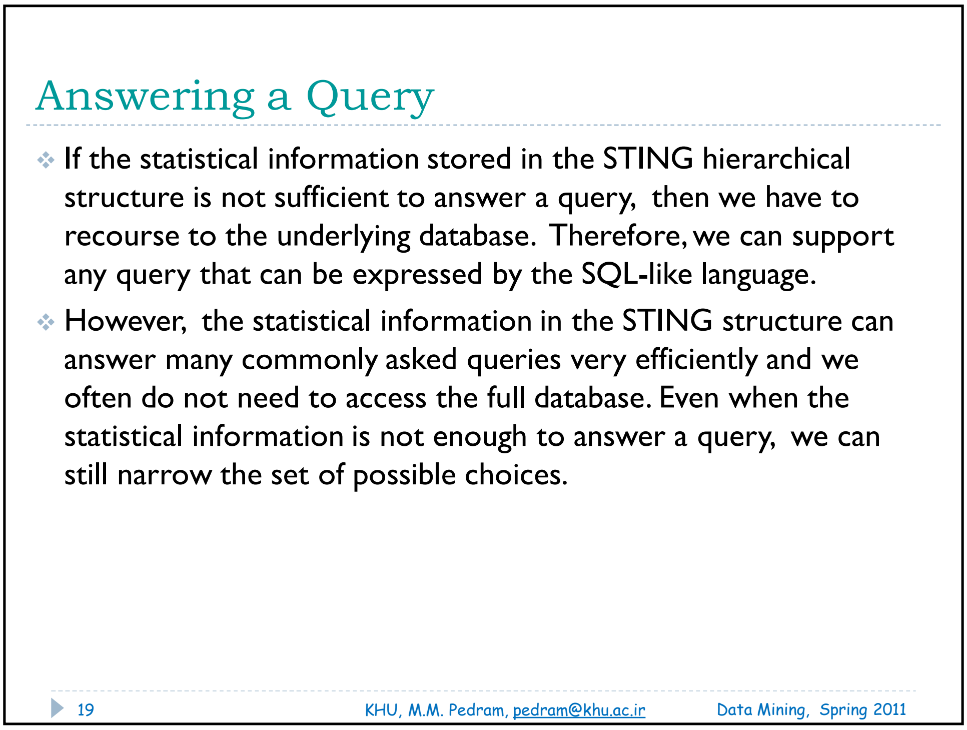
Use a top-down approach to answer spatial data queries:

1. Start from a pre-selected layer (typically with a small number of cells);
2. From the pre-selected layer until you reach the bottom layer do the following:
   - For each cell in the current level compute the confidence interval indicating a cell's relevance to a given query;
     - □ If it is relevant, include the cell in a cluster;
     - □ If it irrelevant, remove cell from further consideration;
   - Look for relevant cells at the next lower layer.
3. Combine relevant cells into relevant regions (based on grid-neighborhood) and return the so obtained clusters as your answers.

How are Queries Processed ?

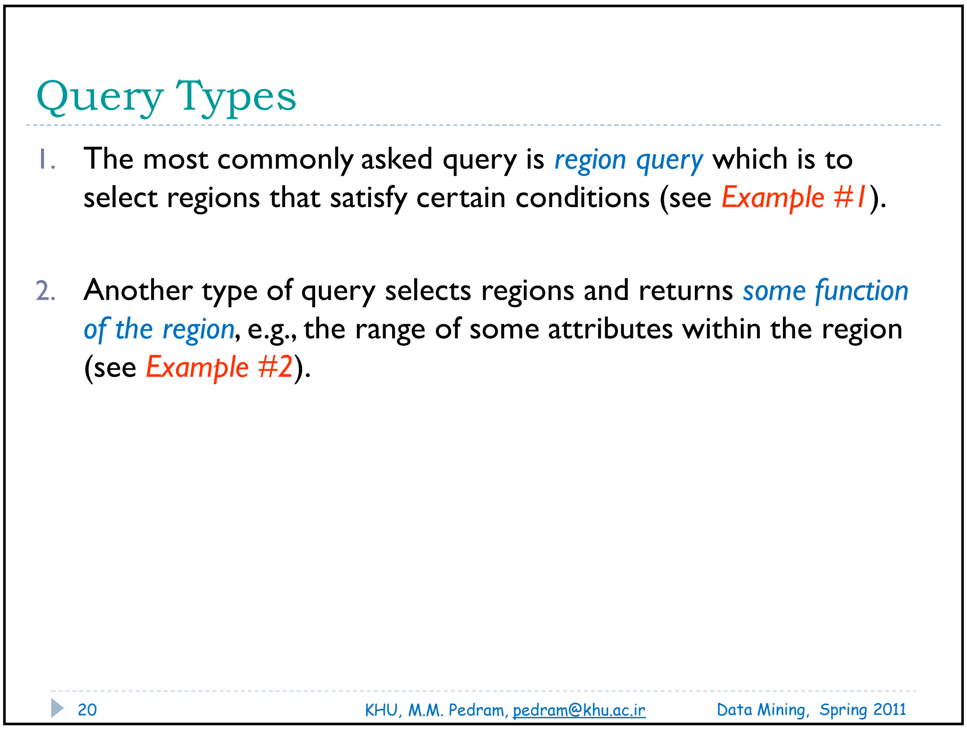
- Different Grid Levels during Query Processing.





Answering a Query

- If the statistical information stored in the STING hierarchical structure is not sufficient to answer a query, then we have to recourse to the underlying database. Therefore, we can support any query that can be expressed by the SQL-like language.
- However, the statistical information in the STING structure can answer many commonly asked queries very efficiently and we often do not need to access the full database. Even when the statistical information is not enough to answer a query, we can still narrow the set of possible choices.



Query Types

1. The most commonly asked query is *region query* which is to select regions that satisfy certain conditions (see *Example #1*).

2. Another type of query selects regions and returns *some function of the region*, e.g., the range of some attributes within the region (see *Example #2*).

Query Types

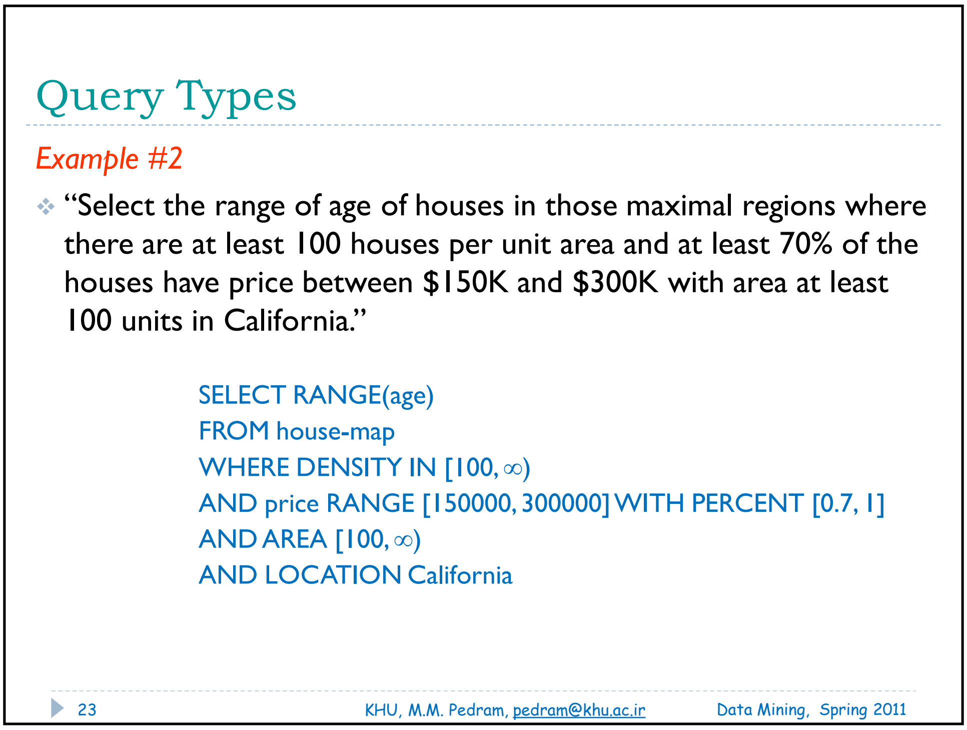
- Assume that the spatial area is the house map of the regions of our city.

Query Types

Example #1

- Select the maximal regions that have at least 100 houses per unit area and at least 70% of the house prices are above $400K and with total area at least 100 units with 90% confidence.

```
SELECT REGION
FROM house-map
WHERE DENSITY IN [100, ∞)
AND price RANGE [400000, ∞) WITH PERCENT [0.7, 1]
AND AREA [100, ∞)
AND WITH CONFIDENCE 0.9
```

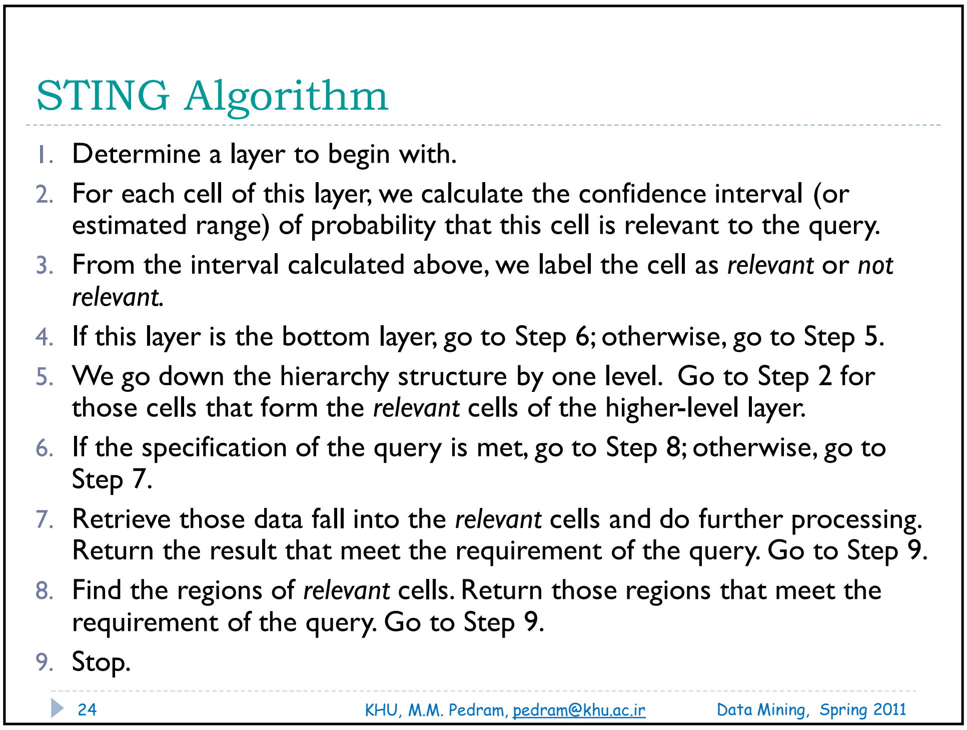


Query Types

Example #2

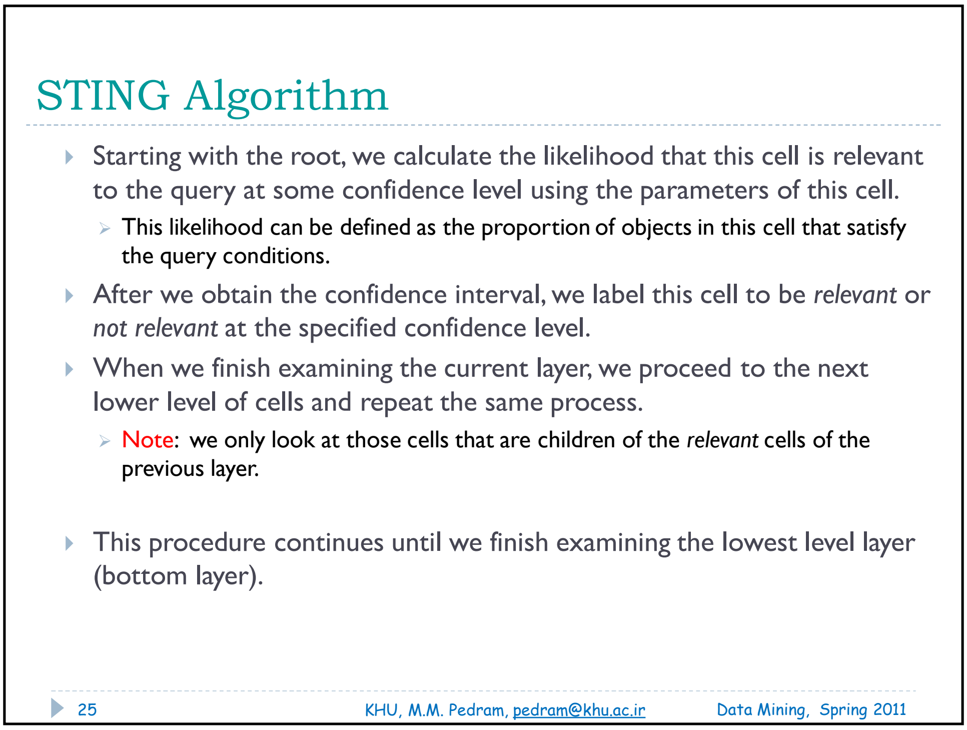
- "Select the range of age of houses in those maximal regions where there are at least 100 houses per unit area and at least 70% of the houses have price between $150K and $300K with area at least 100 units in California."

```
SELECT RANGE(age)
FROM house-map
WHERE DENSITY IN [100, ∞)
AND price RANGE [150000, 300000] WITH PERCENT [0.7, 1]
AND AREA [100, ∞)
AND LOCATION California
```



STING Algorithm

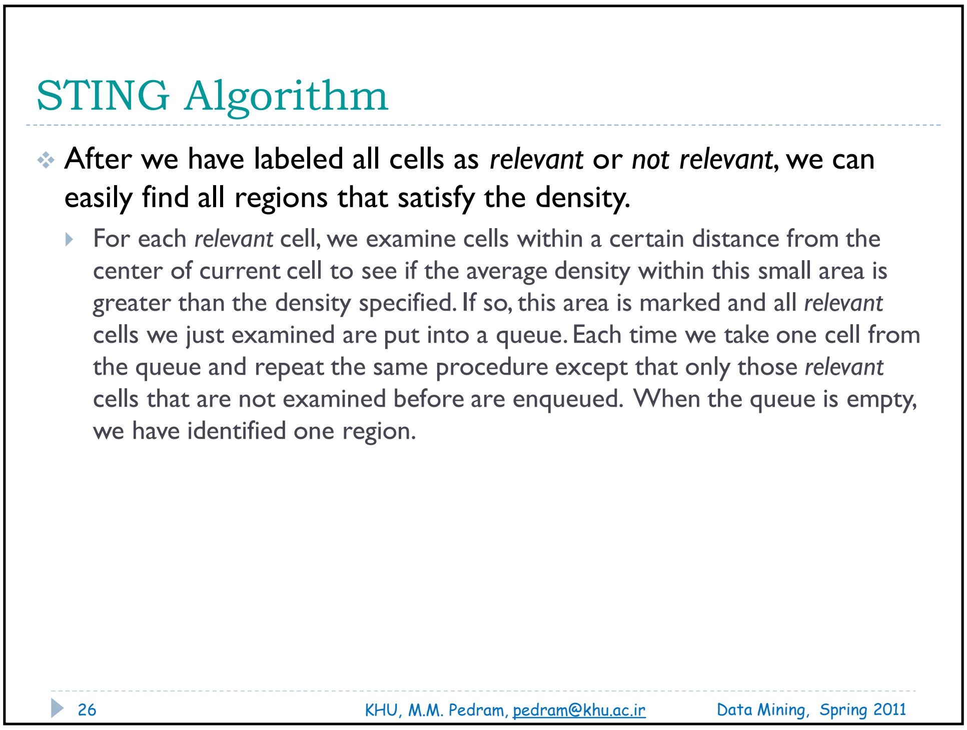
1. Determine a layer to begin with.
2. For each cell of this layer, we calculate the confidence interval (or estimated range) of probability that this cell is relevant to the query.
3. From the interval calculated above, we label the cell as *relevant* or *not relevant.*
4. If this layer is the bottom 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.
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.



STING Algorithm

- Starting with the root, we calculate the likelihood that this cell is relevant to the query at some confidence level using the parameters of this cell.
  - This likelihood can be defined as the proportion of objects in this cell that satisfy the query conditions.
- After we obtain the confidence interval, we label this cell to be *relevant* or *not relevant* at the specified confidence level.
- When we finish examining the current layer, we proceed to the next lower level of cells and repeat the same process.
  - Note: we only look at those cells that are children of the *relevant* cells of the previous layer.
- This procedure continues until we finish examining the lowest level layer (bottom layer).

25



STING Algorithm

- After we have labeled all cells as *relevant* or *not relevant*, we can easily find all regions that satisfy the density.
  - For each *relevant* cell, we examine cells within a certain distance from the center of current cell to see if the average density within this small area is greater than the density specified. If so, this area is marked and all *relevant* cells we just examined are put into a queue. Each time we take one cell from the queue and repeat the same procedure except that only those *relevant* cells that are not examined before are enqueued. When the queue is empty, we have identified one region.

26

Example

- Suppose the objects in our database are houses and price is one of the attributes.

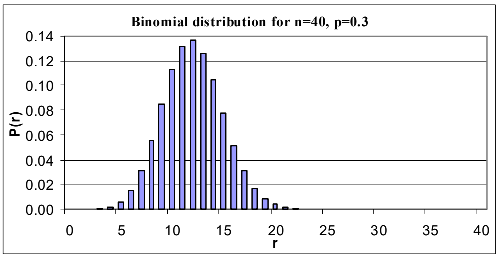
Query:

- "Find those regions with area at least *A* where the number of houses per unit area is at least *c* and at least β% of the houses have price between *a* and *b* with $(1 - \alpha)$ confidence"
  - where $a < b$. Here, *a* could be $-\infty$ and *b* could be $+\infty$.

```
SELECT REGION
FROM house-map
WHERE DENSITY IN [c, ∞)
AND price RANGE [a, b] WITH PERCENT [β%, 1]
AND AREA [A, ∞)
AND WITH CONFIDENCE (1 – α)
```

Remember #1

- Binomial Probability Distribution



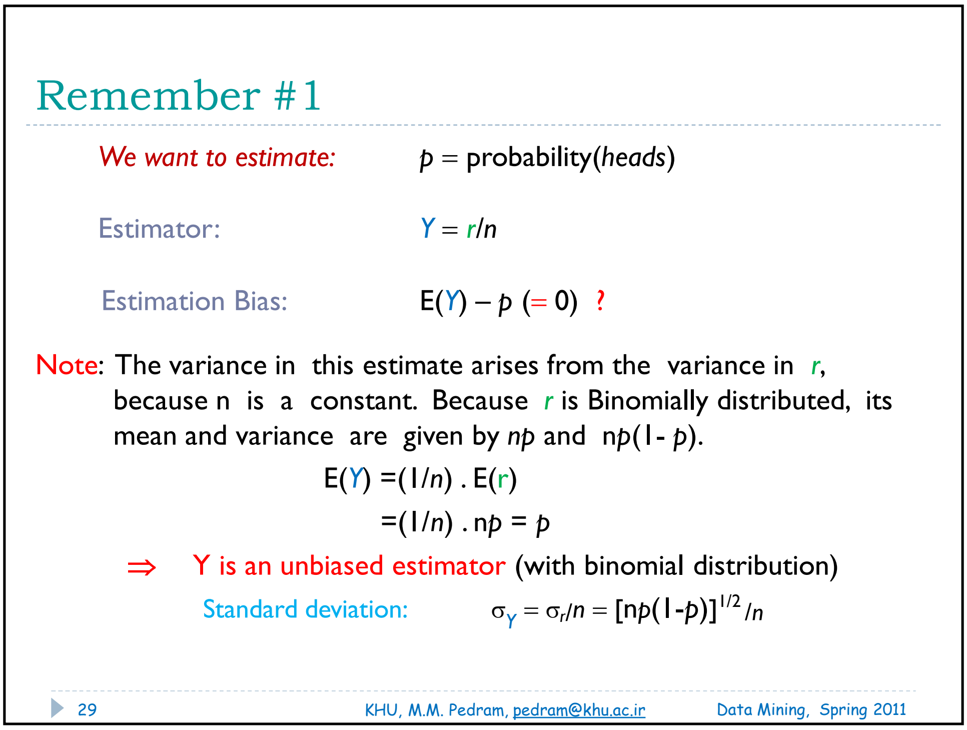
$$P(R = r) = \frac{n!}{r!(n-r)!} p^r (1-p)^{n-r}$$

Probability P(r) of r heads(errors)in n coin flips, if p = probability(*heads*)

$$E[X] \equiv \sum_{i=0}^{n} i \cdot P(i) = np$$

$$Var(X) \equiv E[(X - E[X])^2] = np(1-p)$$

$$\sigma_X \equiv \sqrt{E[(X - E[X])^2]} = \sqrt{np(1-p)}$$



Remember #1

We want to estimate: p = probability(*heads*)

Estimator: $Y = r/n$

Estimation Bias: $E(Y) - p \; (= 0)$?

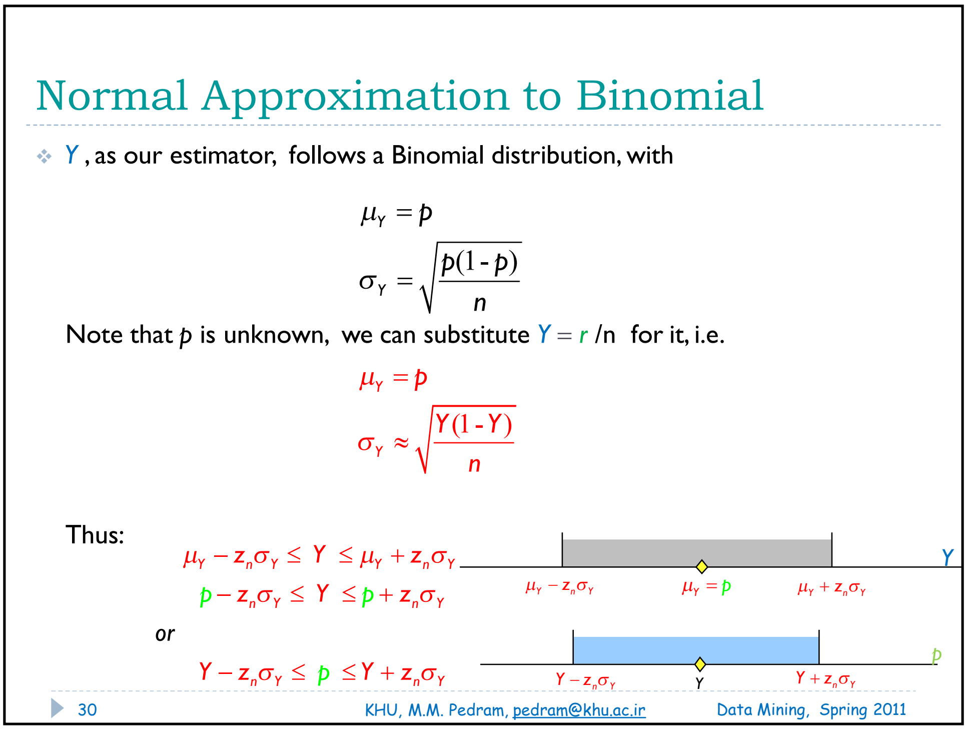
Note: The variance in this estimate arises from the variance in r, because n is a constant. Because r is Binomially distributed, its mean and variance are given by np and $np(1 - p)$.

$$E(Y) = (1/n) \,.\, E(r)$$
$$= (1/n) \,.\, np = p$$

⇒ Y is an unbiased estimator (with binomial distribution)

Standard deviation: $\sigma_Y = \sigma_r/n = [np(1-p)]^{1/2}/n$

29



Normal Approximation to Binomial

- Y, as our estimator, follows a Binomial distribution, with

$$\mu_Y = p$$
$$\sigma_Y = \sqrt{\frac{p(1-p)}{n}}$$

Note that p is unknown, we can substitute $Y = r/n$ for it, i.e.

$$\mu_Y = p$$
$$\sigma_Y \approx \sqrt{\frac{Y(1-Y)}{n}}$$

Thus:

$$\mu_Y - z_n\sigma_Y \le Y \le \mu_Y + z_n\sigma_Y$$
$$p - z_n\sigma_Y \le Y \le p + z_n\sigma_Y$$

or

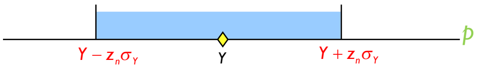
$$Y - z_n\sigma_Y \le p \le Y + z_n\sigma_Y$$

30

Remember #2

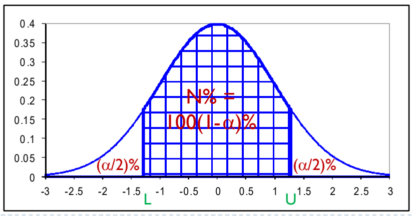
Two-Sided and One-Sided Bounds

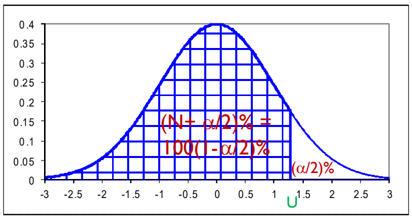
A N% = 100(1-α)% confidence interval with lower bound ***L*** and upper bound ***U***, implies a (N+ α/2)% = 100(1- α/2)% confidence interval with lower bound *L* and no upper bound, or with upper bound *U* and no lower bound.



N% of area (probability) lies in $\mu \pm z_N\sigma$

N% :	50%	68%	80%	90%	95%	98%	99%
z_N :	0.67	1.00	1.28	1.64	1.96	2.33	2.53



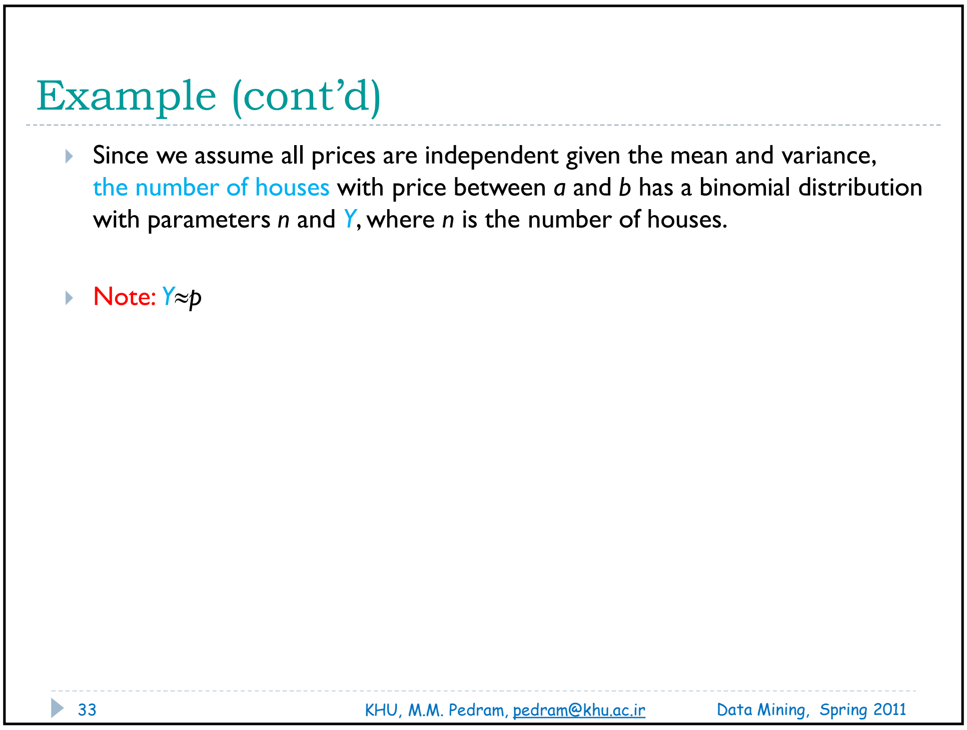


Example (cont'd)

- We begin from the top layer that has only one cell and stop at the bottom level. Assume that the price in each bottom layer cell is approximately normally distributed.
- For each cell, we want to calculate *the proportion of houses* (Y=*p*) whose price is within the range [*a*, *b*].
- If the distribution type is normal for Y=*p*, the probability that a price is between *a* and *b* is:

$$\begin{aligned} Y &= P(a \le price \le b) \\ &= P(\frac{a-m}{s} \le \frac{price-m}{s} \le \frac{b-m}{s}) \\ &= P(\frac{a-m}{s} \le Z \le \frac{b-m}{s}) \\ &= \Phi(\frac{b-m}{s}) - \Phi(\frac{a-m}{s}) \end{aligned}$$

(*m* and *s* are the mean and standard deviation of all prices in this cell, respectively.)

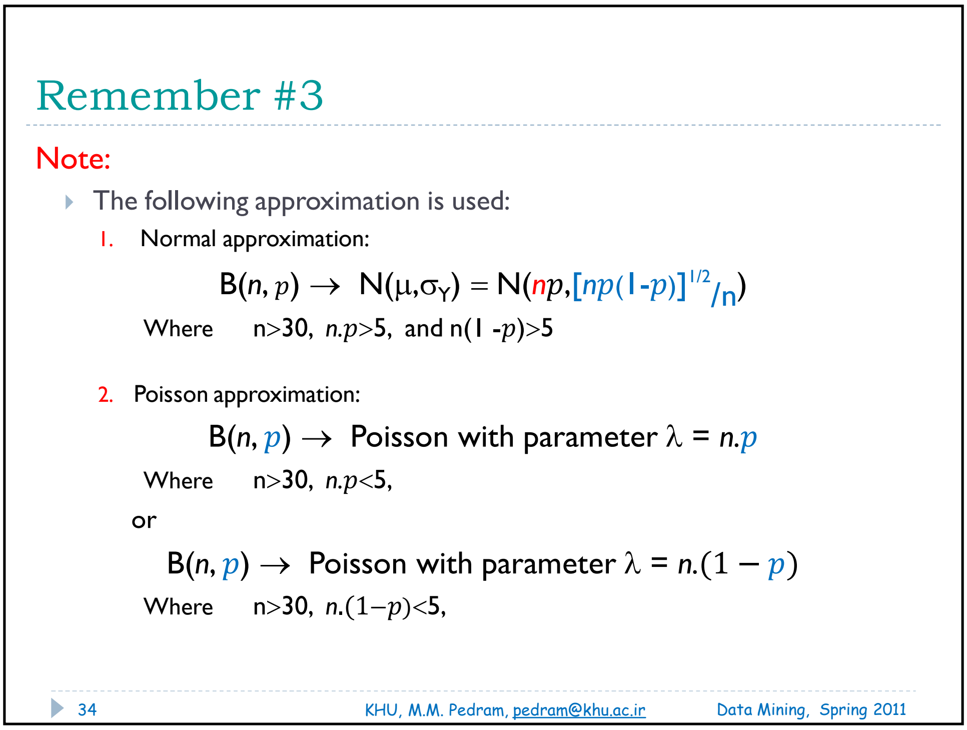


Example (cont'd)

- Since we assume all prices are independent given the mean and variance, the number of houses with price between *a* and *b* has a binomial distribution with parameters *n* and *Y*, where *n* is the number of houses.

- Note: $Y \approx p$

33



Remember #3

Note:

- The following approximation is used:
  1. Normal approximation:

$$B(n, p) \rightarrow N(\mu, \sigma_Y) = N(np, [np(1-p)]^{1/2}/n)$$

Where $n>30$, $n.p>5$, and $n(1-p)>5$

  2. Poisson approximation:

$$B(n, p) \rightarrow \text{Poisson with parameter } \lambda = n.p$$

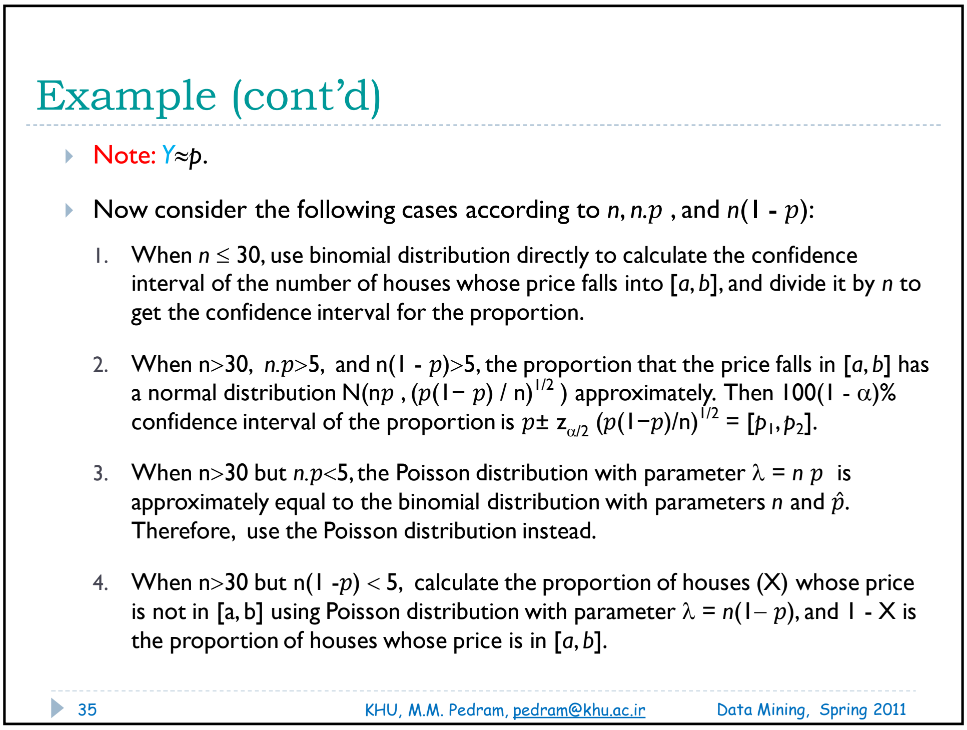
Where $n>30$, $n.p<5$,

or

$$B(n, p) \rightarrow \text{Poisson with parameter } \lambda = n.(1-p)$$

Where $n>30$, $n.(1-p)<5$,

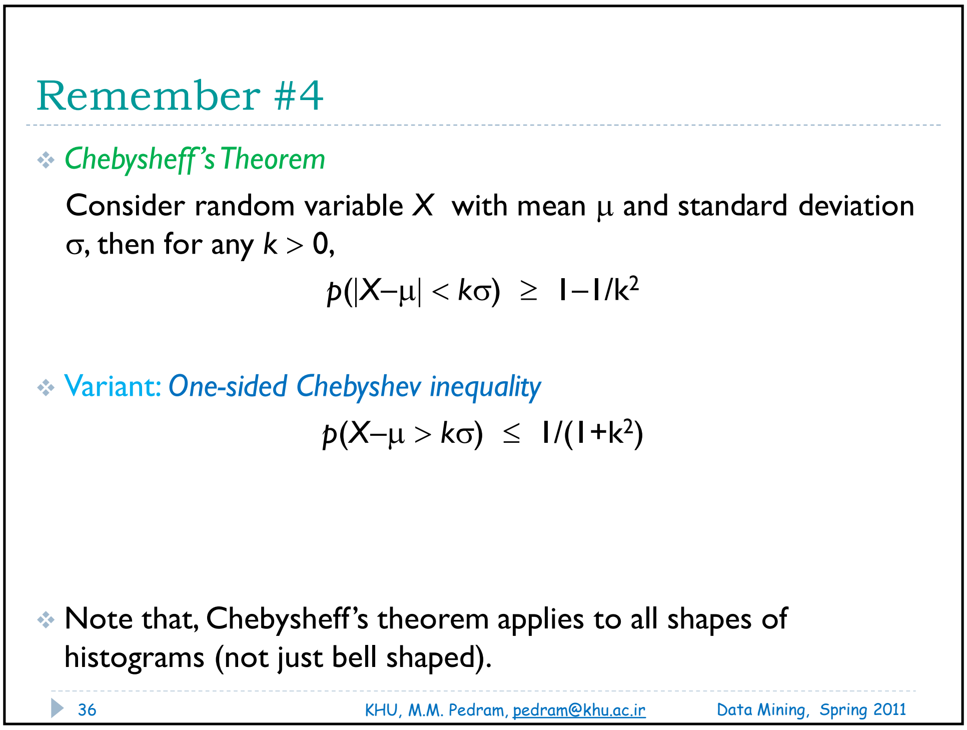
34



Example (cont'd)

- Note: $Y \approx p$.
- Now consider the following cases according to n, $n.p$, and $n(1 - p)$:

1. When $n \leq 30$, use binomial distribution directly to calculate the confidence interval of the number of houses whose price falls into $[a, b]$, and divide it by n to get the confidence interval for the proportion.
2. When n>30, $n.p>5$, and $n(1 - p)>5$, the proportion that the price falls in $[a, b]$ has a normal distribution $N(np , (p(1- p) / n)^{1/2})$ approximately. Then $100(1 - \alpha)\%$ confidence interval of the proportion is $p \pm z_{\alpha/2}\ (p(1-p)/n)^{1/2} = [p_1, p_2]$.
3. When n>30 but $n.p<5$, the Poisson distribution with parameter $\lambda = n\ p$ is approximately equal to the binomial distribution with parameters n and $\hat{p}$. Therefore, use the Poisson distribution instead.
4. When n>30 but $n(1 -p) < 5$, calculate the proportion of houses (X) whose price is not in [a, b] using Poisson distribution with parameter $\lambda = n(1- p)$, and 1 - X is the proportion of houses whose price is in $[a, b]$.



Remember #4

- *Chebysheff's Theorem*

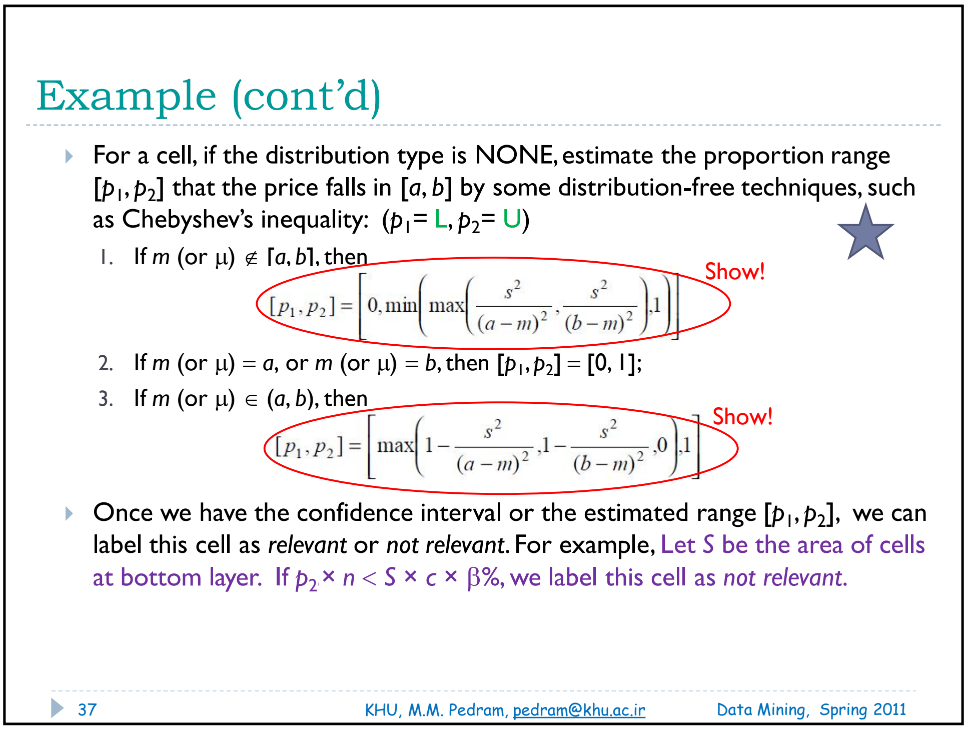
Consider random variable X with mean μ and standard deviation σ, then for any $k > 0$,

$$p(|X-\mu| < k\sigma) \;\geq\; 1-1/k^2$$

- Variant: *One-sided Chebyshev inequality*

$$p(X-\mu > k\sigma) \;\leq\; 1/(1+k^2)$$

- Note that, Chebysheff's theorem applies to all shapes of histograms (not just bell shaped).



Example (cont'd)

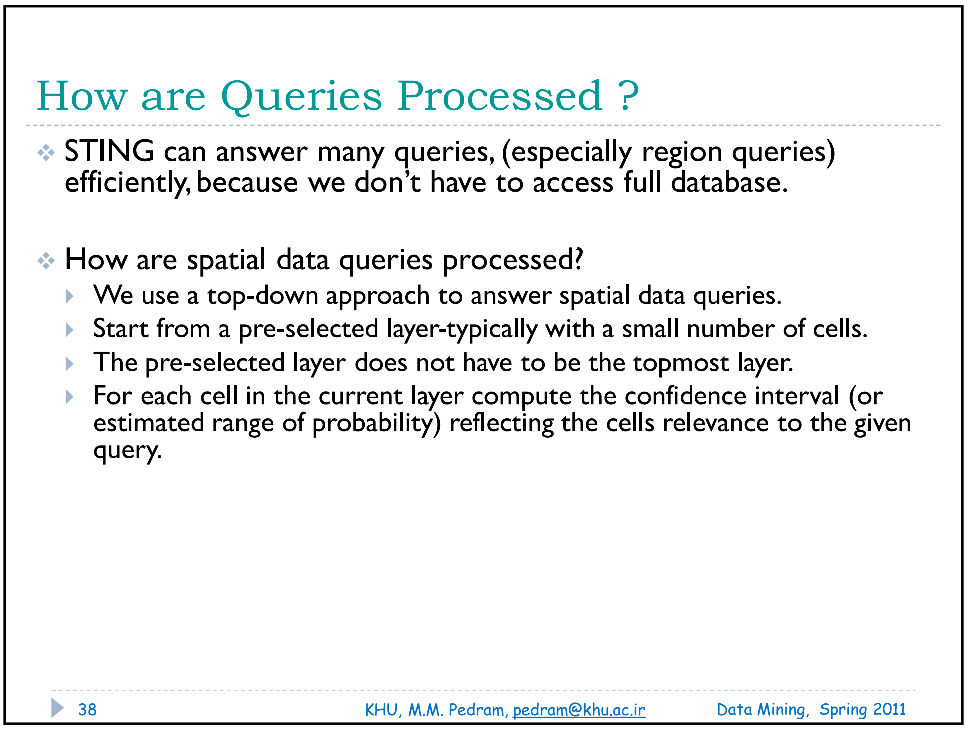
- For a cell, if the distribution type is NONE, estimate the proportion range $[p_1, p_2]$ that the price falls in $[a, b]$ by some distribution-free techniques, such as Chebyshev's inequality: (p_1= L, p_2= U)
  1. If m (or μ) $\notin [a, b]$, then

     $$[p_1, p_2] = \left[0, \min\left(\max\left(\frac{s^2}{(a-m)^2}, \frac{s^2}{(b-m)^2}\right), 1\right)\right]$$

     Show!
  2. If m (or μ) $= a$, or m (or μ) $= b$, then $[p_1, p_2] = [0, 1]$;
  3. If m (or μ) $\in (a, b)$, then

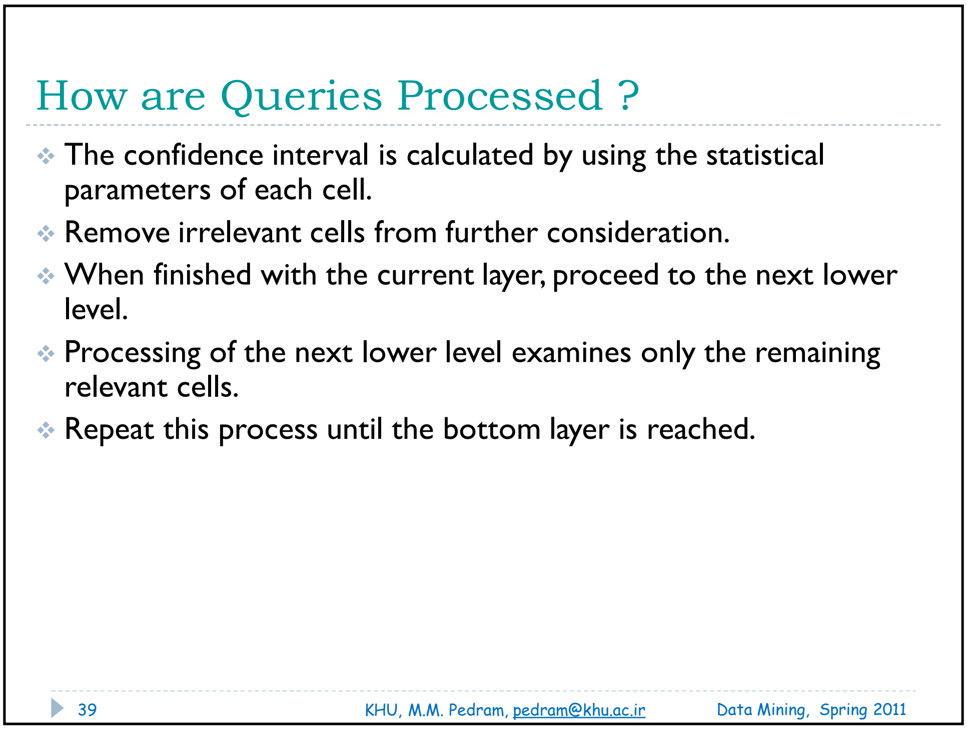
     $$[p_1, p_2] = \left[\max\left(1 - \frac{s^2}{(a-m)^2}, 1 - \frac{s^2}{(b-m)^2}, 0\right), 1\right]$$

     Show!
- Once we have the confidence interval or the estimated range $[p_1, p_2]$, we can label this cell as *relevant* or *not relevant*. For example, Let S be the area of cells at bottom layer. If $p_2 \times n < S \times c \times \beta\%$, we label this cell as *not relevant*.



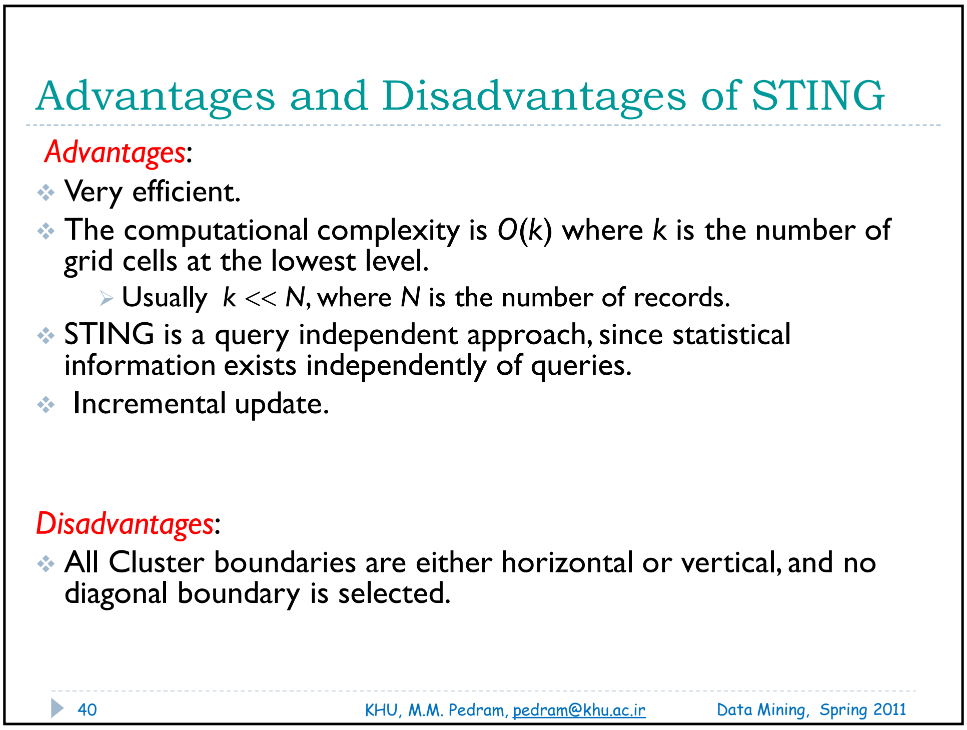
How are Queries Processed ?

- STING can answer many queries, (especially region queries) efficiently, because we don't have to access full database.
- How are spatial data queries processed?
  - We use a top-down approach to answer spatial data queries.
  - Start from a pre-selected layer-typically with a small number of cells.
  - The pre-selected layer does not have to be the topmost layer.
  - For each cell in the current layer compute the confidence interval (or estimated range of probability) reflecting the cells relevance to the given query.



How are Queries Processed ?

- The confidence interval is calculated by using the statistical parameters of each cell.
- Remove irrelevant cells from further consideration.
- When finished with the current layer, proceed to the next lower level.
- Processing of the next lower level examines only the remaining relevant cells.
- Repeat this process until the bottom layer is reached.



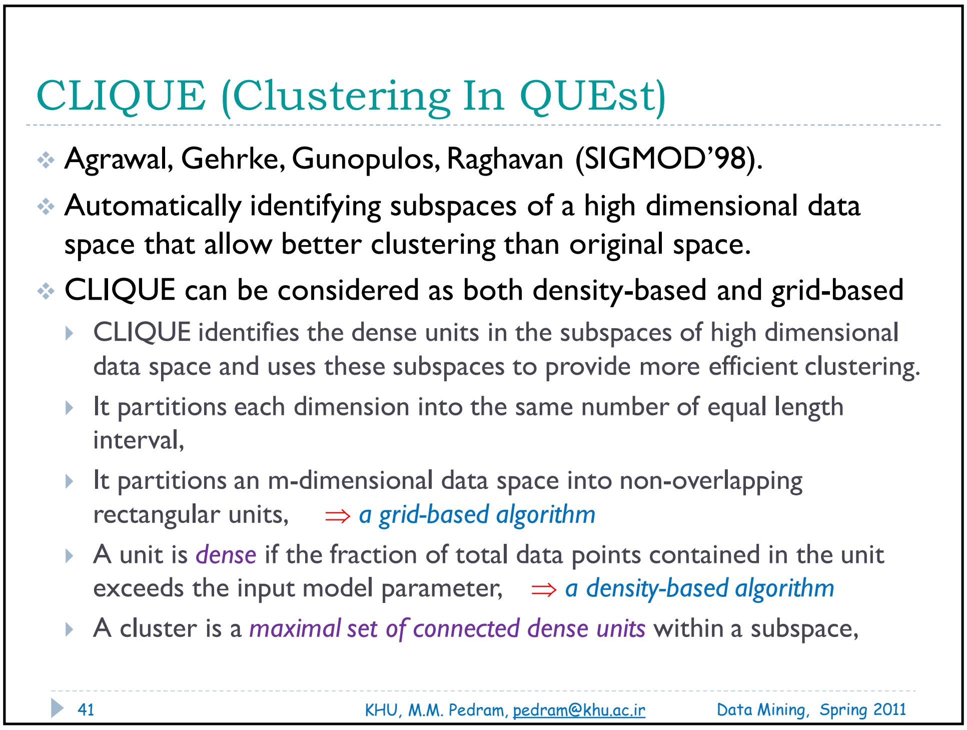
Advantages and Disadvantages of STING

Advantages:

- Very efficient.
- The computational complexity is $O(k)$ where k is the number of grid cells at the lowest level.
  - Usually $k << N$, where N is the number of records.
- STING is a query independent approach, since statistical information exists independently of queries.
- Incremental update.

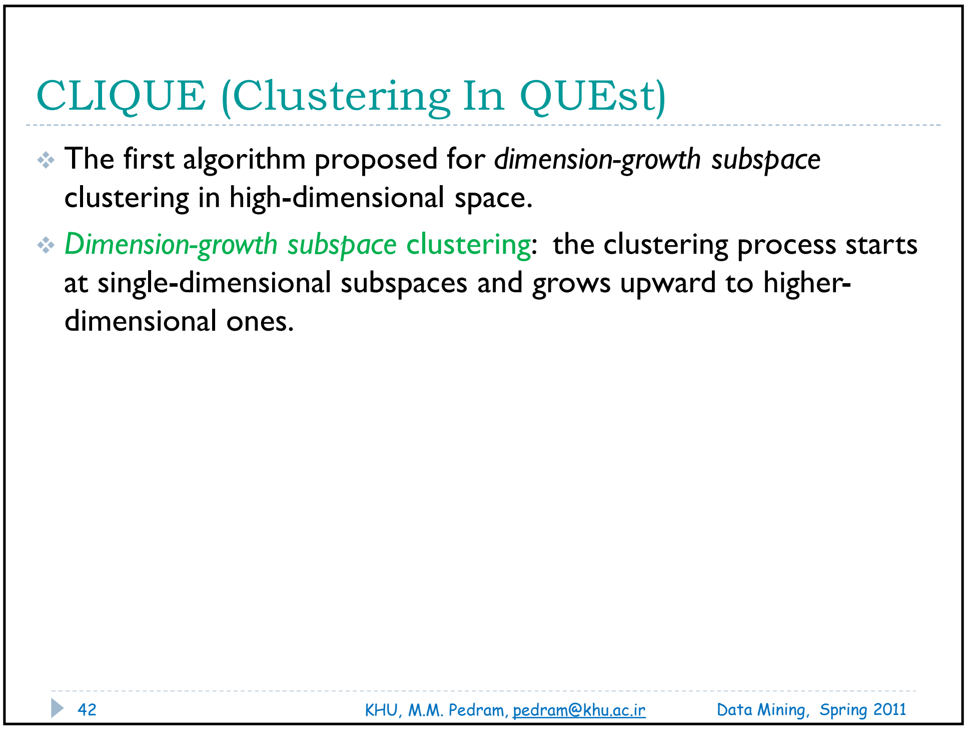
Disadvantages:

- All Cluster boundaries are either horizontal or vertical, and no diagonal boundary is selected.



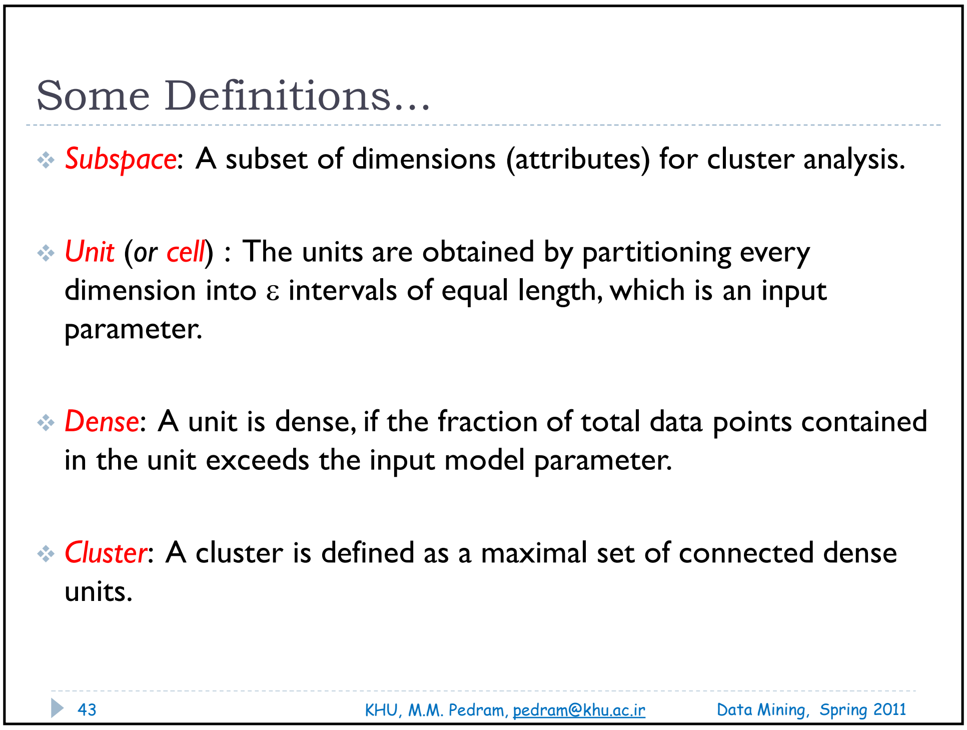
CLIQUE (Clustering In QUEst)

- Agrawal, Gehrke, Gunopulos, Raghavan (SIGMOD'98).
- Automatically identifying subspaces of a high dimensional data space that allow better clustering than original space.
- CLIQUE can be considered as both density-based and grid-based
  - CLIQUE identifies the dense units in the subspaces of high dimensional data space and uses these subspaces to provide more efficient clustering.
  - It partitions each dimension into the same number of equal length interval,
  - It partitions an m-dimensional data space into non-overlapping rectangular units, ⇒ *a grid-based algorithm*
  - A unit is *dense* if the fraction of total data points contained in the unit exceeds the input model parameter, ⇒ *a density-based algorithm*
  - A cluster is a *maximal set of connected dense units* within a subspace,



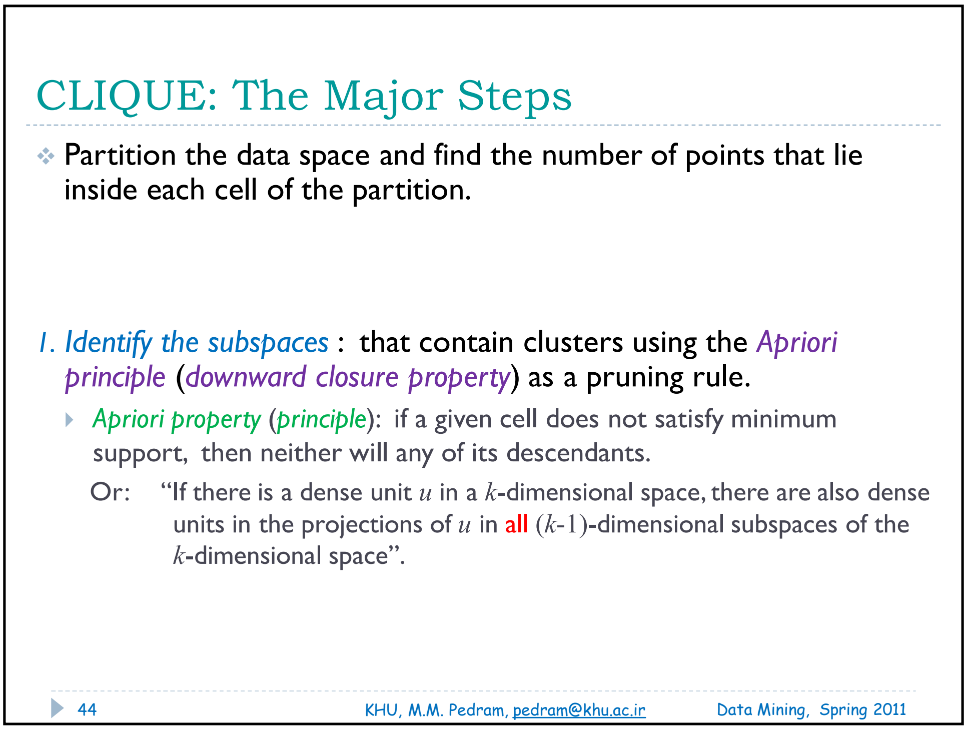
CLIQUE (Clustering In QUEst)

- The first algorithm proposed for *dimension-growth subspace* clustering in high-dimensional space.
- *Dimension-growth subspace* clustering: the clustering process starts at single-dimensional subspaces and grows upward to higher-dimensional ones.



Some Definitions…

- *Subspace*: A subset of dimensions (attributes) for cluster analysis.
- *Unit* (*or cell*) : The units are obtained by partitioning every dimension into ε intervals of equal length, which is an input parameter.
- *Dense*: A unit is dense, if the fraction of total data points contained in the unit exceeds the input model parameter.
- *Cluster*: A cluster is defined as a maximal set of connected dense units.

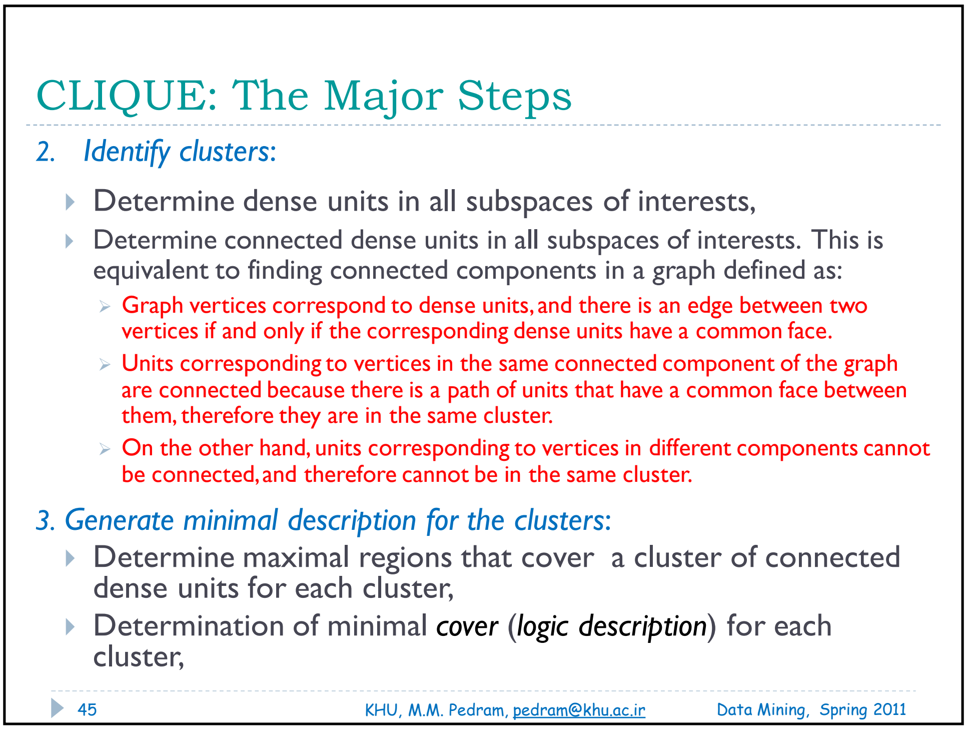


CLIQUE: The Major Steps

- Partition the data space and find the number of points that lie inside each cell of the partition.

1. *Identify the subspaces* : that contain clusters using the *Apriori principle* (*downward closure property*) as a pruning rule.
   - *Apriori property* (*principle*): if a given cell does not satisfy minimum support, then neither will any of its descendants.

     Or: "If there is a dense unit u in a k-dimensional space, there are also dense units in the projections of u in all $(k\text{-}1)$-dimensional subspaces of the k-dimensional space".

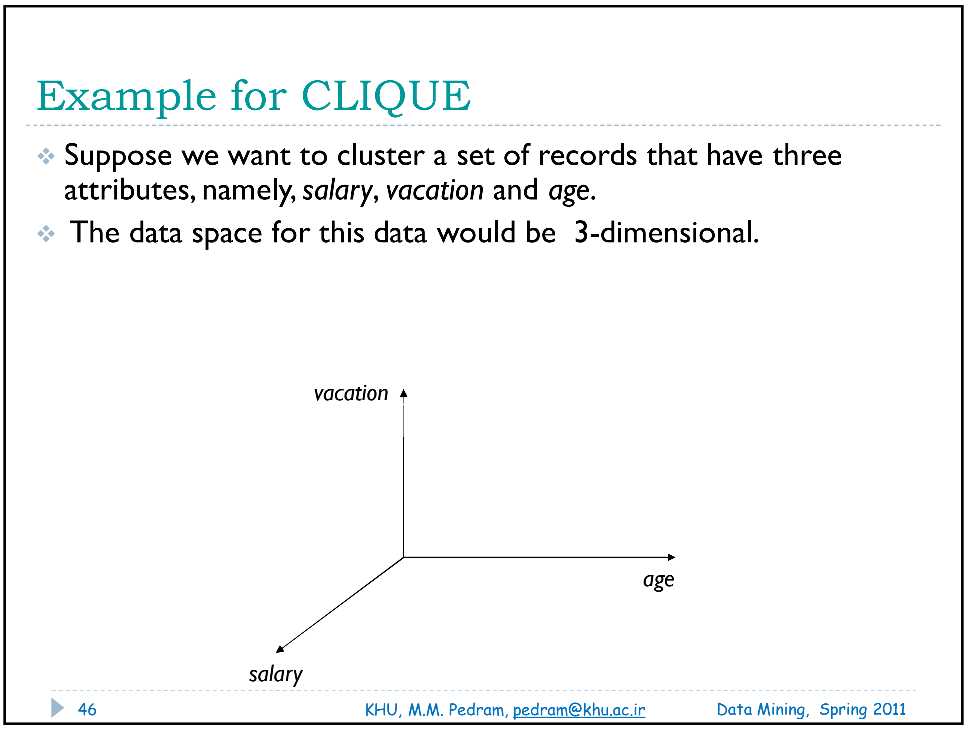


CLIQUE: The Major Steps

2. *Identify clusters:*
   - Determine dense units in all subspaces of interests,
   - Determine connected dense units in all subspaces of interests. This is equivalent to finding connected components in a graph defined as:
     - Graph vertices correspond to dense units, and there is an edge between two vertices if and only if the corresponding dense units have a common face.
     - Units corresponding to vertices in the same connected component of the graph are connected because there is a path of units that have a common face between them, therefore they are in the same cluster.
     - On the other hand, units corresponding to vertices in different components cannot be connected, and therefore cannot be in the same cluster.

3. *Generate minimal description for the clusters:*
   - Determine maximal regions that cover a cluster of connected dense units for each cluster,
   - Determination of minimal *cover* (*logic description*) for each cluster,

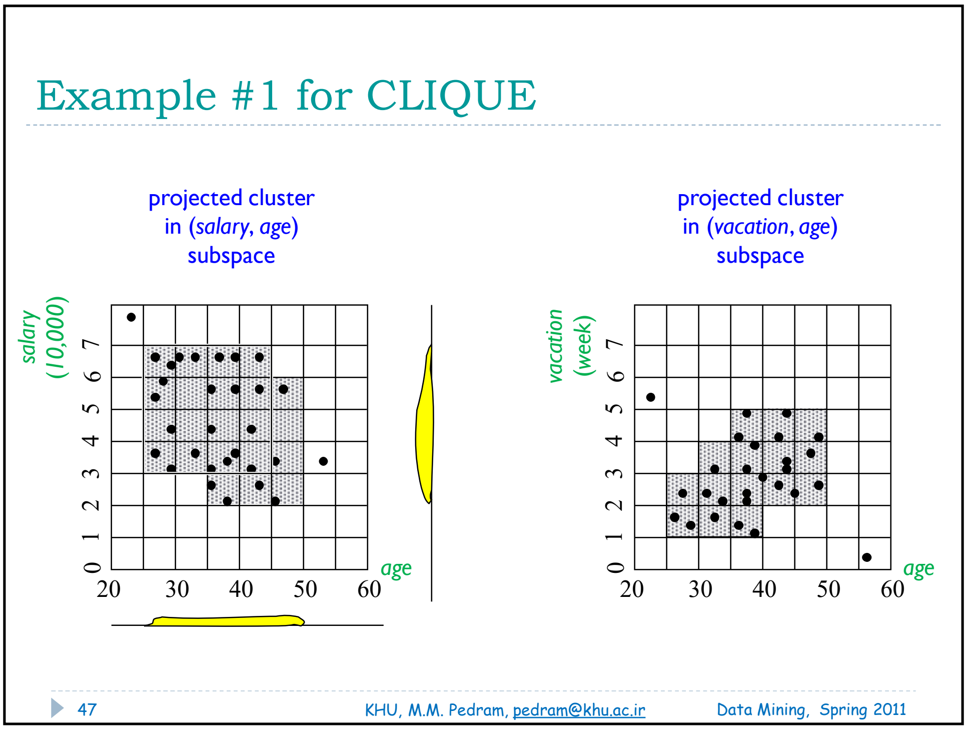
45



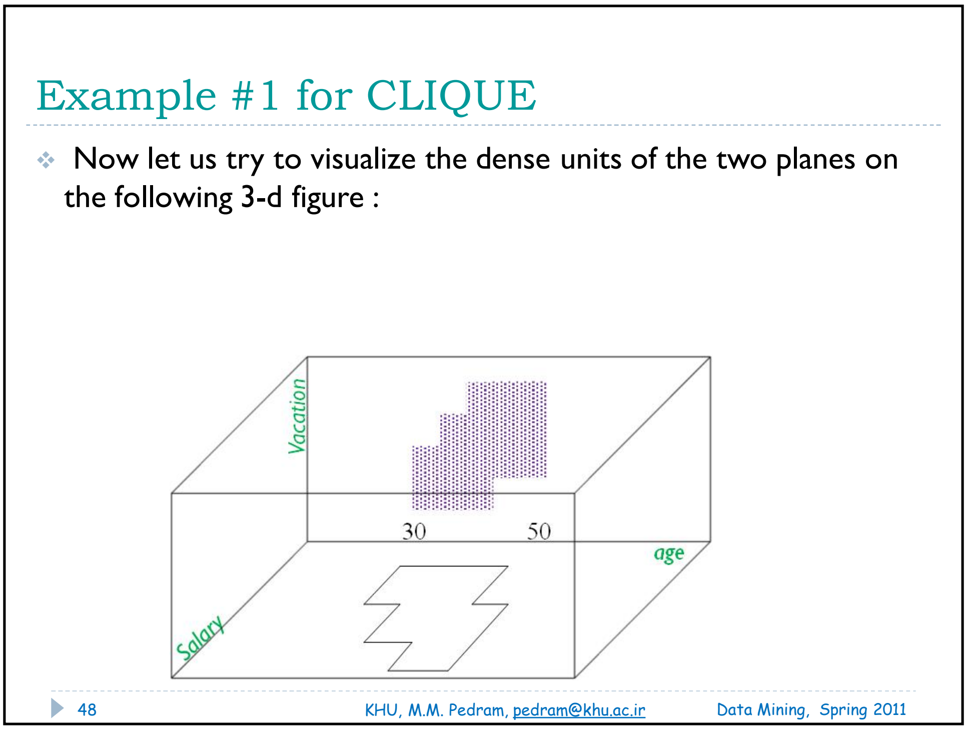
Example for CLIQUE

- Suppose we want to cluster a set of records that have three attributes, namely, *salary*, *vacation* and *age*.
- The data space for this data would be 3-dimensional.

46



Example #1 for CLIQUE

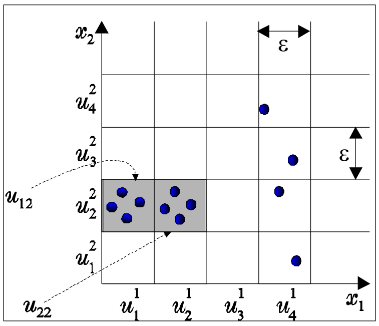


Example #1 for CLIQUE

- Now let us try to visualize the dense units of the two planes on the following 3-d figure :

Example #2 for CLIQUE

- The downward closure property of the density is shown for units u_{12} , u_{22}.
- A 2-dimensional grid of lines of edge size ε applied in the two-dimensional feature space.
  - For density $\tau = 3$:
    - $u_1^1, u_2^1, u_4^1, u_2^2$ are 1-dimensional dense units, each containing 4, 4, 4 and 9 points, respectively.
    - u_{12} and u_{22} are two-dimensional dense units each containing 4 points.
  - u_{12} and u_{22} are directly connected.



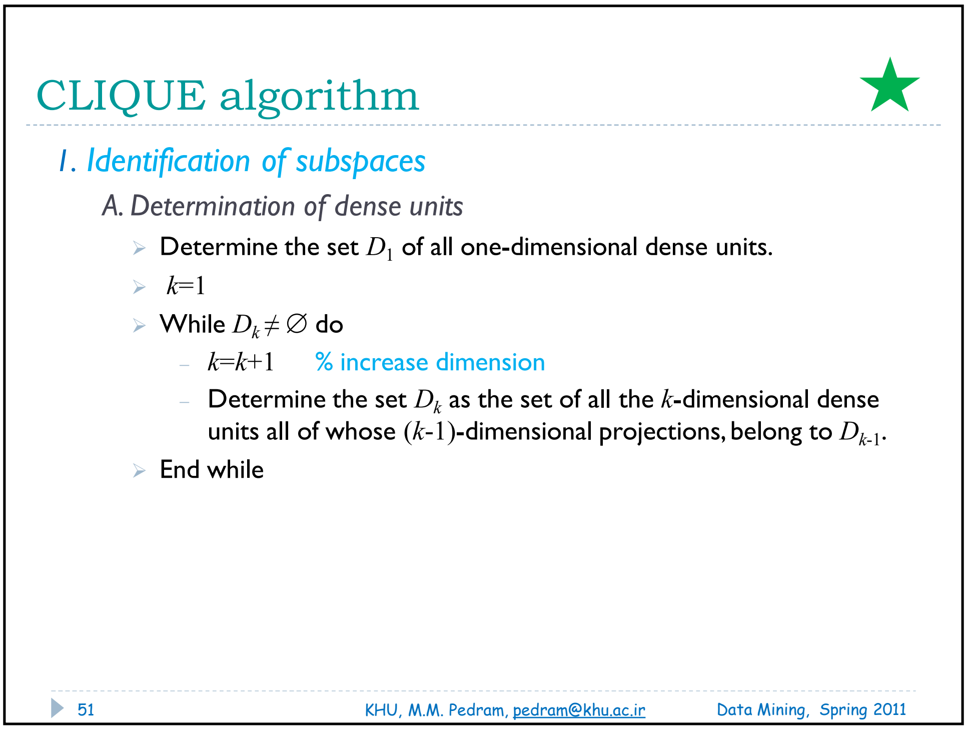
CLIQUE

Note:

- Why does CLIQUE confine its search for dense units of higher dimensionality to the intersection of dense units in the subspaces?
  - The identification of the candidate search space is based on the Apriori property.
  - The property applies prior knowledge of items in the search space so that portions of space can be pruned.
  - If a collection of points S is a cluster in a k-dimensional space, then S is also part of a cluster in any (k - l)-dimensional projections of this space

Note:

- Instead of using a density threshold, we would use entropy as a measure of the quality of subspace clusters.

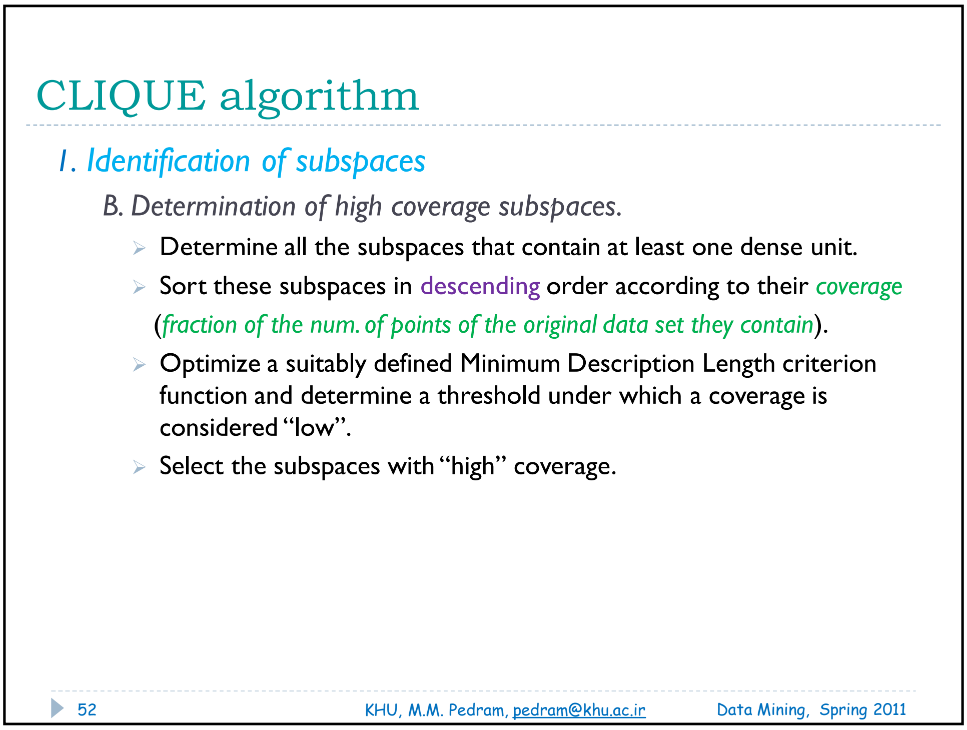


CLIQUE algorithm

I. *Identification of subspaces*

A. Determination of dense units

- Determine the set D_1 of all one-dimensional dense units.
- $k=1$
- While $D_k \neq \varnothing$ do
  - $k=k+1$ % increase dimension
  - Determine the set D_k as the set of all the k-dimensional dense units all of whose $(k\text{-}1)$-dimensional projections, belong to $D_{k\text{-}1}$.
- End while



CLIQUE algorithm

I. *Identification of subspaces*

B. Determination of high coverage subspaces.

- Determine all the subspaces that contain at least one dense unit.
- Sort these subspaces in descending order according to their *coverage* (*fraction of the num. of points of the original data set they contain*).
- Optimize a suitably defined Minimum Description Length criterion function and determine a threshold under which a coverage is considered "low".
- Select the subspaces with "high" coverage.

CLIQUE algorithm

2. *Identification of clusters*

```
• For each high coverage subspace $S$ do
    – Consider the set $E$ of all the dense units in $S$.
    – $m'=0$
    – While $E \neq \emptyset$
        ○ $m' = m' + 1$
        ○ Select a randomly chosen unit $u$ from $E$.
        ○ Assign to $C_{m'}$, $u$ and all units of $E$ that are connected to $u$.
        ○ $E=E-C_{m'}$
    – End while
• End for
```

The clusters in the data set are all clusters identified in all high coverage subspaces (they are consisted of units).

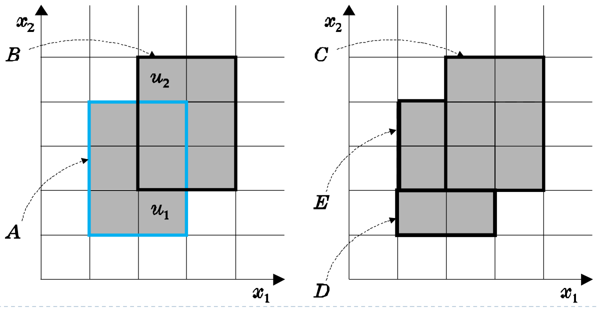
CLIQUE algorithm

3. *Minimal description of clusters*

The minimal description of a cluster C, produced by step 2, is the minimum possible union of hyper-rectangular regions.

For example

- $A \cup B$ is the minimum cluster description of the shaded region.
- $C \cup D \cup E$ is a non-minimal cluster description of the same region.



CLIQUE algorithm

```
For each cluster $C$ do
  1st stage
  • $c$=0
  • While $C \neq \varnothing$
      – $c$=$c$+1
      – Choose a dense unit in $C$
      – For $i$=1 to $l$
          o Grow the unit in both directions along the $i$-th dimension, trying to
            cover as many units in $C$ as possible (boxes that are not belong to $C$
            should not be covered).
      – End for
      – Define the set $I$ containing all the units covered by the above procedure
      – $C$=$C$-$I$
  • End while
  2nd stage
  • Remove all covers whose units are covered by at least another cover.
```

How to obtain a maximal region covering a dense unit u

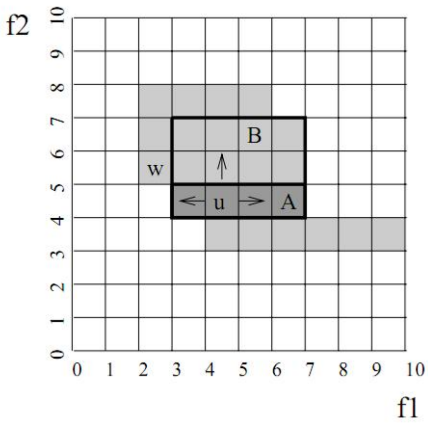
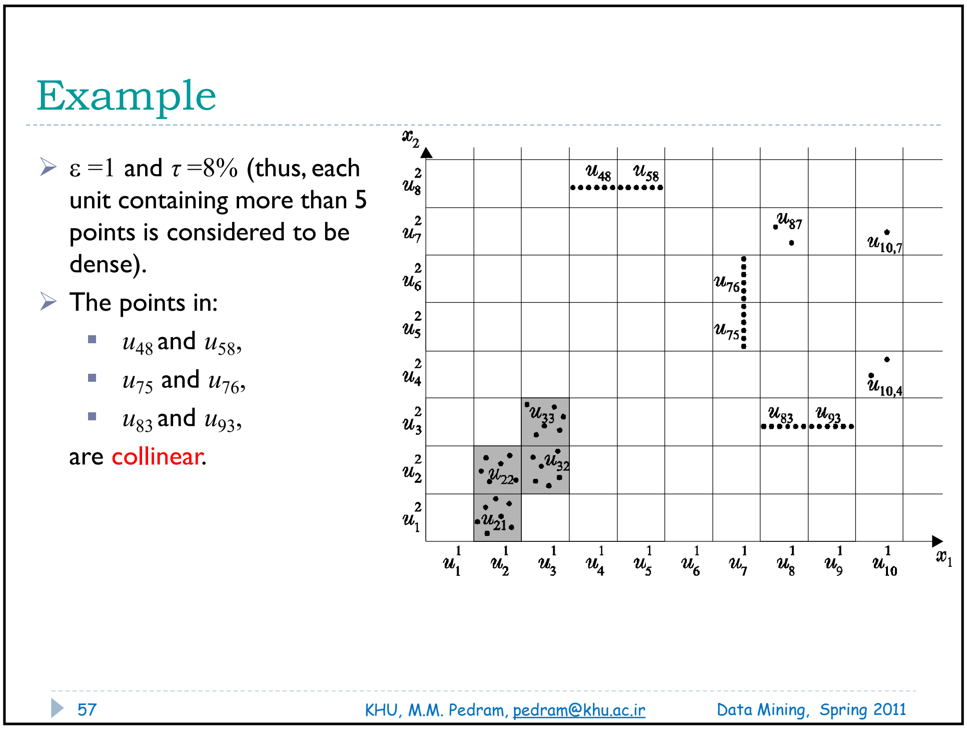


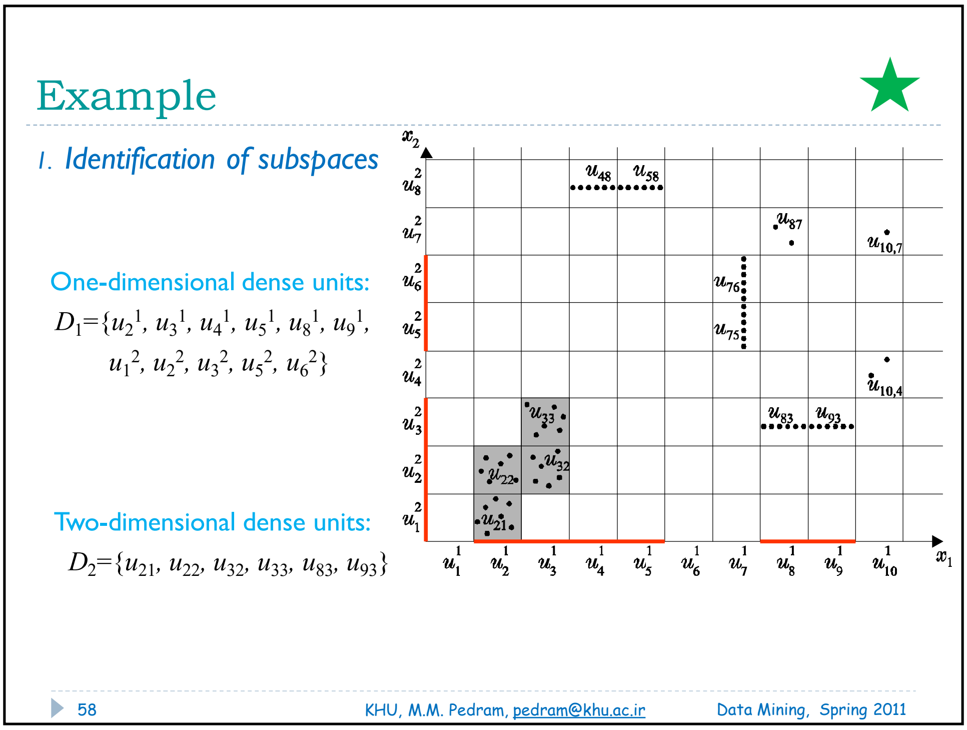
Figure 4: Illustration of the greedy growth algorithm.



Example

- ε =1 and τ =8% (thus, each unit containing more than 5 points is considered to be dense).
- The points in:
  - u_{48} and u_{58},
  - u_{75} and u_{76},
  - u_{83} and u_{93},

  are collinear.



Example

1. *Identification of subspaces*

One-dimensional dense units:

$D_1=\{u_2^1, u_3^1, u_4^1, u_5^1, u_8^1, u_9^1, u_1^2, u_2^2, u_3^2, u_5^2, u_6^2\}$

Two-dimensional dense units:

$D_2=\{u_{21}, u_{22}, u_{32}, u_{33}, u_{83}, u_{93}\}$

Example

Notes:

- Although each one of the u_{48}, u_{75}, u_{76} contains more that 5 points, they are not included in D_2.
- Although it seems unnatural for u_{83} and u_{93} to be included in D_2, they are included since u_3^2 is dense.
- All subspaces of the two-dimensional space contain clusters.

Example

2. *Identification of clusters*

One-dimensional clusters:

$C_1=\{u_2^1, u_3^1, u_4^1, u_5^1\}$,

$C_2=\{u_8^1, u_9^1\}$,

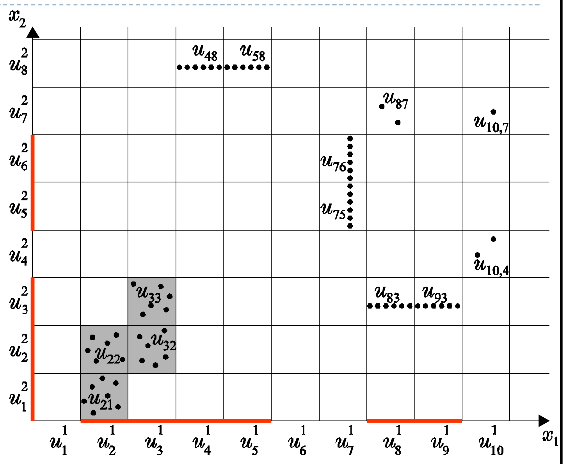
$C_3=\{u_1^2, u_2^2, u_3^2\}$,

$C_4=\{u_5^2, u_6^2\}$.

Two-dimensional clusters:

$C_5=\{u_{21}, u_{22}, u_{32}, u_{33}\}$,

$C_6=\{u_{83}, u_{93}\}$.



Example

3. *Description of clusters*

$C_1=\{(x_1): 1 \le x_1 < 5\}$

$C_2=\{(x_1): 7 \le x_1 < 9\}$

$C_3=\{(x_2): 0 \le x_2 < 3\}$

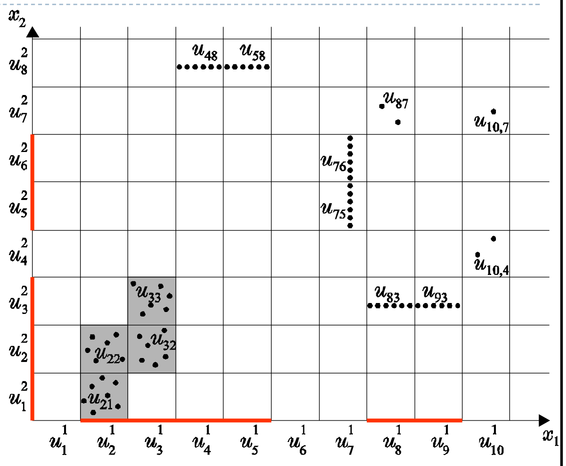
$C_4=\{(x_2): 4 \le x_2 < 6\}$

$C_5=\{(x_1, x_2): 1 \le x_1 < 2, 0 \le x_2 < 2\}$

$\cup$

$\{(x_1, x_2): 2 \le x_1 < 3, 1 \le x_2 < 3\}$

$C_6=\{(x_1, x_2): 7 \le x_1 < 9, 2 \le x_2 < 3\}$



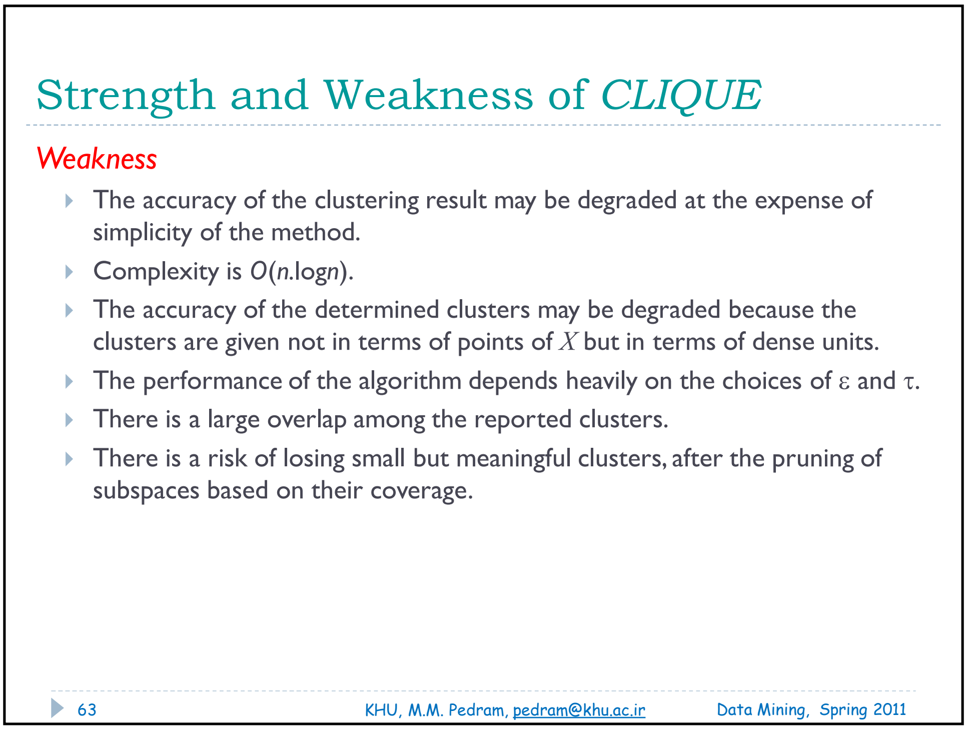
Note

- C_2 and C_6 are essentially the same cluster, which is reported twice by the algorithm.

Strength and Weakness of *CLIQUE*

Strength

- It *automatically* finds subspaces of the highest dimensionality such that high density clusters exist in those subspaces,
- It is *insensitive* to the order of records in input and does not presume some canonical data distribution,
- It scales *linearly* with the size of input, but and has *exponential* scalability as the number of dimensions in the data increases.
- It does not impose any data distribution hypothesis on the data set.



Strength and Weakness of *CLIQUE*

Weakness

- The accuracy of the clustering result may be degraded at the expense of simplicity of the method.
- Complexity is $O(n.\log n)$.
- The accuracy of the determined clusters may be degraded because the clusters are given not in terms of points of X but in terms of dense units.
- The performance of the algorithm depends heavily on the choices of ε and τ.
- There is a large overlap among the reported clusters.
- There is a risk of losing small but meaningful clusters, after the pruning of subspaces based on their coverage.