Chapter-6
Spatial and Web Mining

CSC603.6: Students should be able to Describe complex data types with respect to spatial and web mining.

By,
Safa Hamdare

Outline

- **Spatial Mining:**
  - Spatial Data,
  - Spatial Vs. Classical Data Mining,
  - Spatial Data Structures,
  - Mining Spatial Association and Co-location Patterns,
  - Spatial Clustering Techniques: CLARANS Extension,
- **Web Mining:**
  - Web Content Mining,
  - Web Structure Mining,
  - Web Usage mining,
  - Applications of Web Mining

Spatial Mining

...

University Question

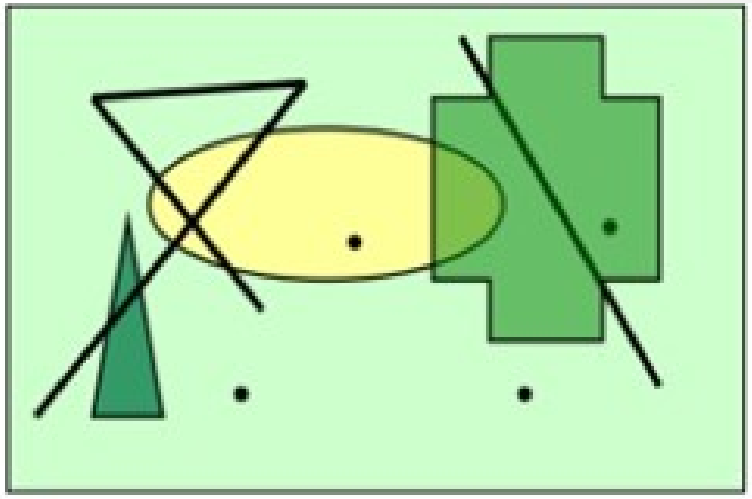
- What is spatial data? Explain CLARANS Extension. **(Dec 2019) 10 marks**
- What are spatial data structures? Outline their importance in GIS. **(May 2019) 5 marks**

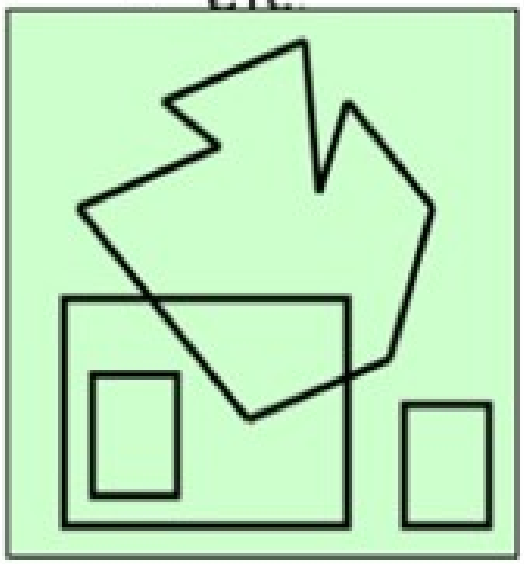
What is Spatial Data?

- broadly be defined as data which covers multidimensional points, lines, rectangles, polygons, cubes and other geometric objects. Spatial data occupies a certain amount of space called it's spatial extent, which is characterized by location and boundary.
- USES
  - Geographic Information Systems.
  - CAD/CAM It can
  - Multimedia Applications
    - Content based image retrieval
    - Fingerprint matching
    - MRI (Digitized medical images)

What is Spatial Data?

- Objects of types:
  - points
  - lines
  - polygons
  - etc.





Used in/for:

- GIS - **G**eographic **I**nformation **S**ystems
- GPS - **G**lobal **P**ositioning **S**ystem
- Environmental studies
- etc …

Features of Spatial Data

- Specific features of spatial data are rich data types, implicit spatial relationships among the variables, observations that are not independent, spatial auto correlation among the features.

- It has two distinct types of attributes i.e. spatial attributes, non spatial attributes. Spatial attributes are used to define the spatial locations and extend of spatial objects.

Spatial vs Non Spatial data

- Non-spatial Information
  - *Same as data in traditional data mining*
  - *Numerical, categorical, ordinal, boolean, etc*

    e.g., city name, city population
- *Spatial Information*
  - *Spatial attribute: geographically referenced*
  - Neighborhood and extent
  - Location, e.g., longitude, latitude, elevation
- *Spatial data representations*
  - Raster: gridded space
  - Vector: point, line, polygon
  - Graph: node, edge, path

Example of Spatial Data

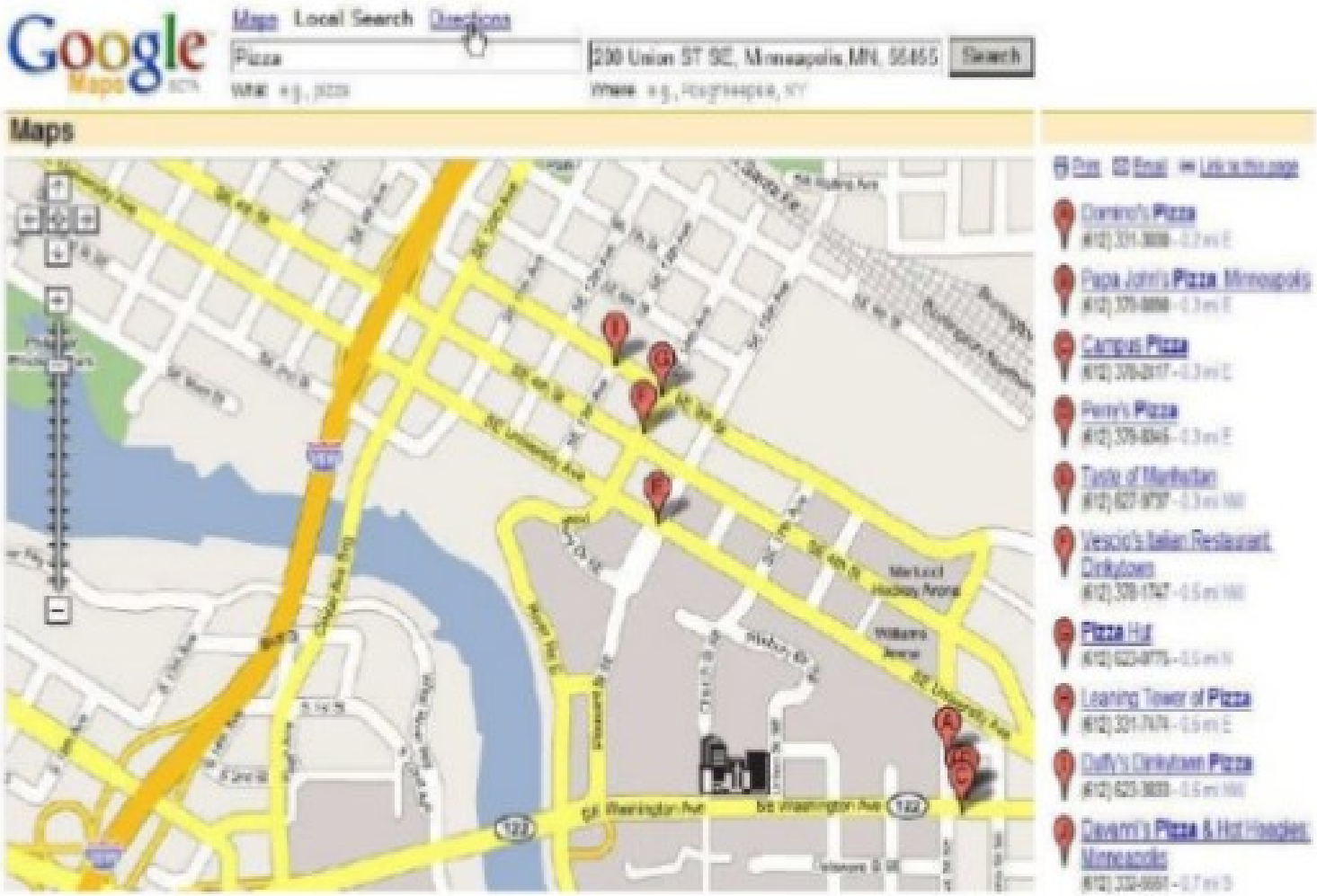


Figure 1: Google Local Search (http://maps.google.com)

Types of spatial Database

- Region Data: It has a spatial extent having a location and boundary. Region data basically is the geometric approximation to an actual database.

- Point Data: Point data consists of collection of points in a multidimensional space. It doesn't cover any area of space.

What is Spatial Data mining?

- It is defined as the non-trivial search for interesting and unexpected spatial patterns from spatial databases.
- New understanding of geographic processes for critical questions like how is the health of planet Earth? Characterize effects of human activity on environment and ecology ? needs spatial data mining.

Spatial vs Classical Data mining

Data inputs of **spatial data mining** are more complex	**Data** inputs of **Classical data mining** are more Simple
Include extended objects such as lines, polygons **and** points	Include objects like name, number.
Spatial attributes of a **spatial** object include information related to **spatial** locations, such as elevation, latitude **and** longitude, as well as shape.	Attributes in classical data mining includes information related to Numerical, categorical, ordinal, Boolean

Spatial Data in GIS

- A geographic information system is any system for capturing, storing, analyzing and managing data and associated attributes which are spatially referenced to Earth.
- There are two broad methods used to store data in a GIS i.e. Raster and Vector. In a GIS, geographical features are often expressed as vectors, by considering those features as geometrical shapes like point, chains, polygons .

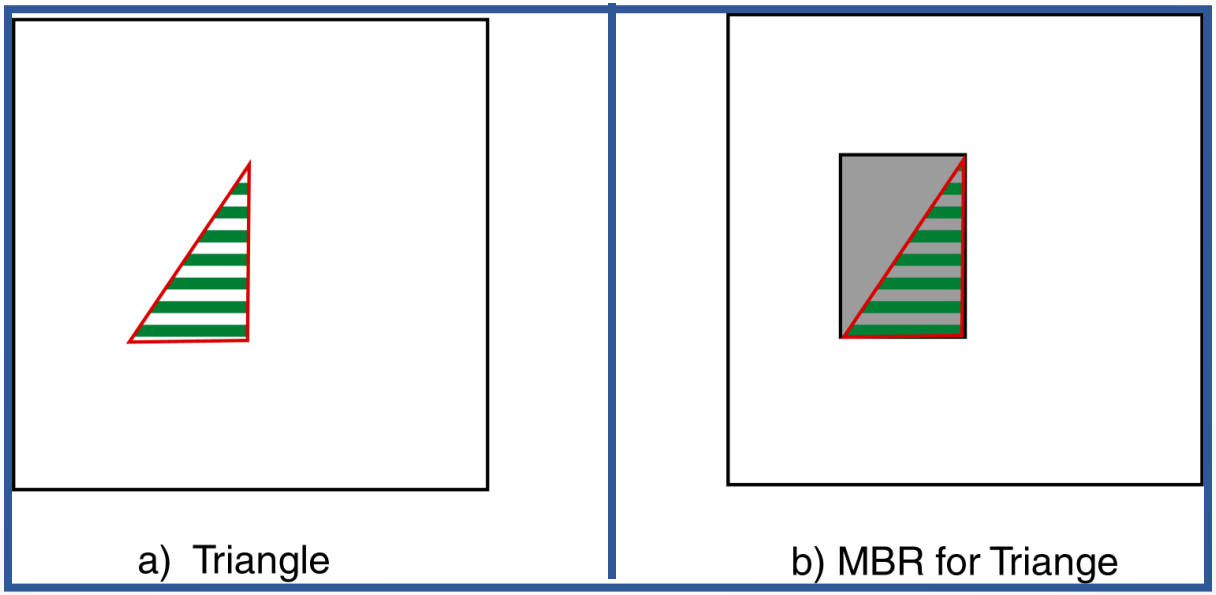
Spatial Data structures used in GIS

In order to handle spatial data efficiently, as required in computer aided design and geo-data applications, a database system needs an index mechanism that will help it retrieve data items quickly according to their spatial locations.

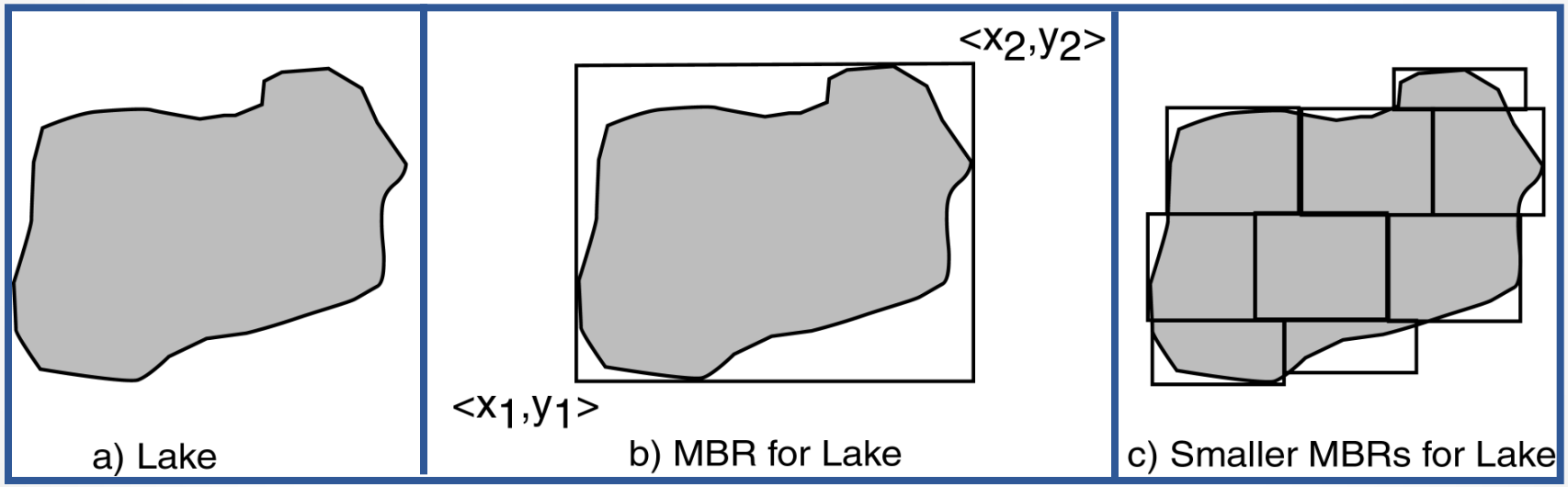
- Quad tree
- k-d tree
- R-tree
- R+-tree

MBR

- Minimum Bounding Rectangle
- Smallest rectangle that completely contains the object



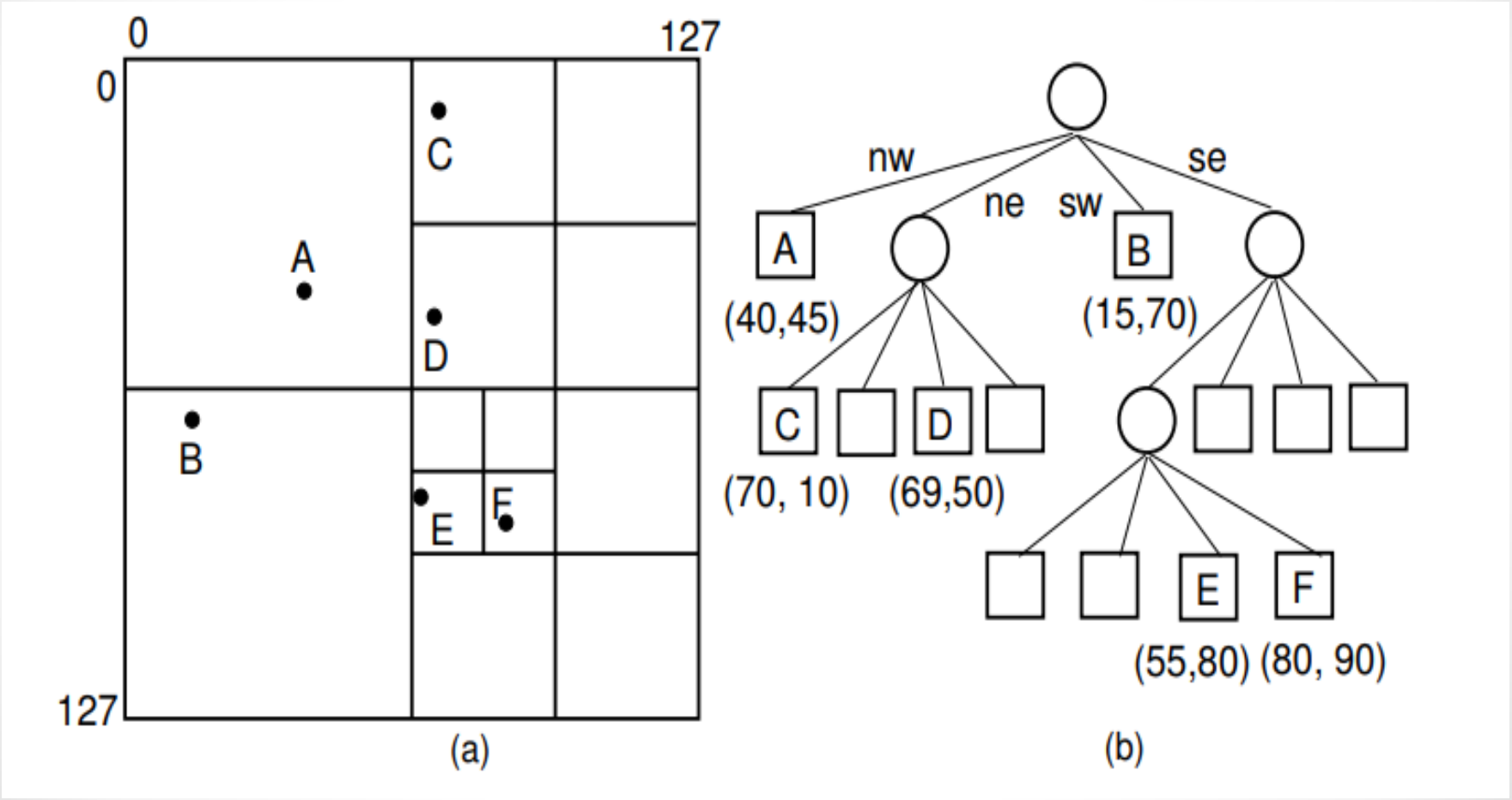
MBR Examples



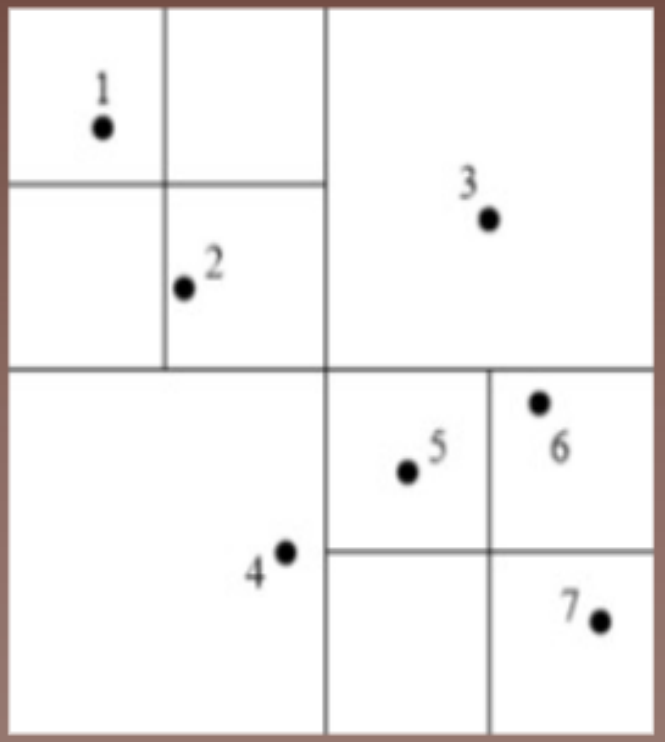
1. Quad tree

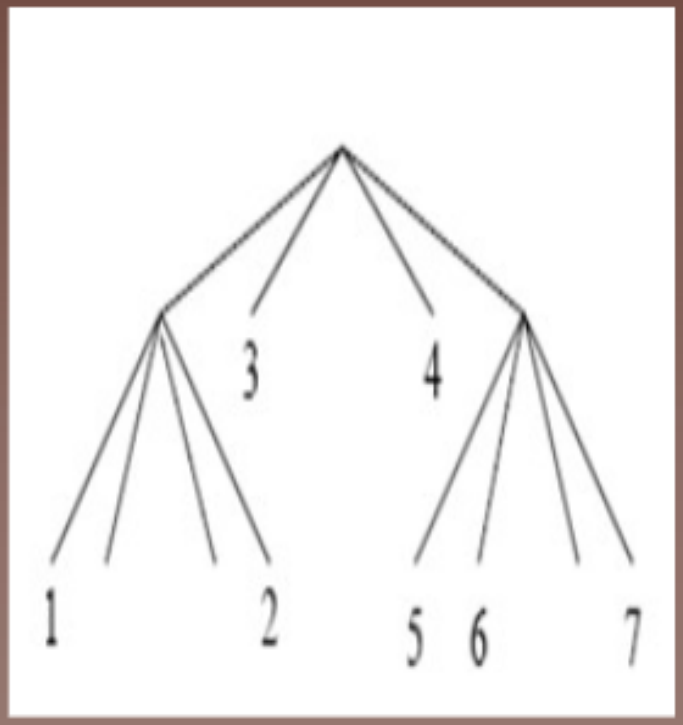
- It is used to store 2D space.
- Each node of a quad tree is associated with a rectangular region of space.
- The top node is associated with the entire target space.
- Each internal node splits the space into four disjunct sub spaces according to the axes.
- Each of these sub spaces is split recursively until there is at most one object inside each of them.

Division of space by quad tree



Division of space by quad tree

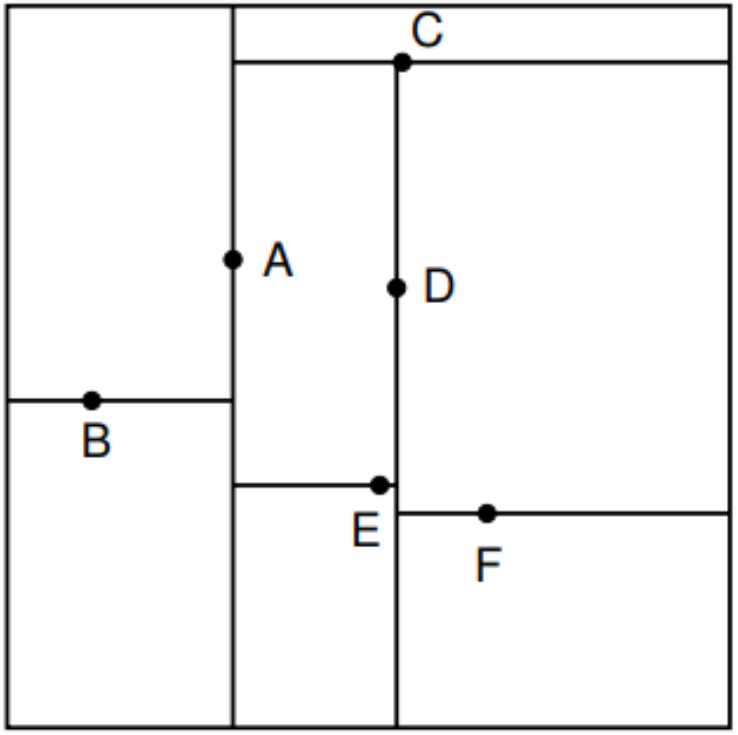




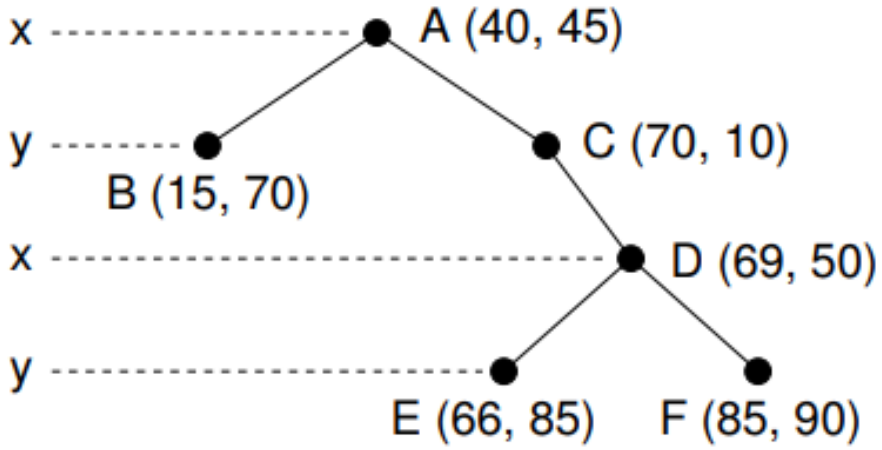
2. K-d tree

- The k-d tree is a modification to the BST that allows for efficient processing of multidimensional keys.
- The k-d tree differs from the BST in that each level of the k-d tree makes branching decisions based on a particular search key associated with that level, called the discriminator.
- We define the discriminator at level i to be i mod k for k dimensions.
  - For example, assume that we store data organized by xy-coordinates. In this case, k is 2 (there are two coordinates), with the x coordinate field arbitrarily designated key 0, and the y-coordinate field designated key 1. At each level, the discriminator alternates between x and y.

Division of space by k-d tree



(a)



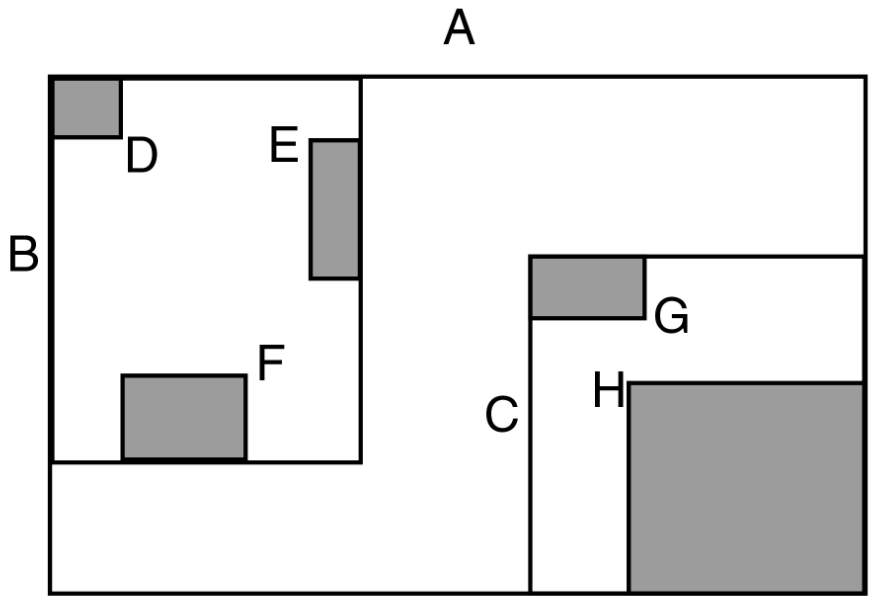
(b)

A(40,45), B(15,70), C(70,10), D(69,50), E(66,85), F(85,90)

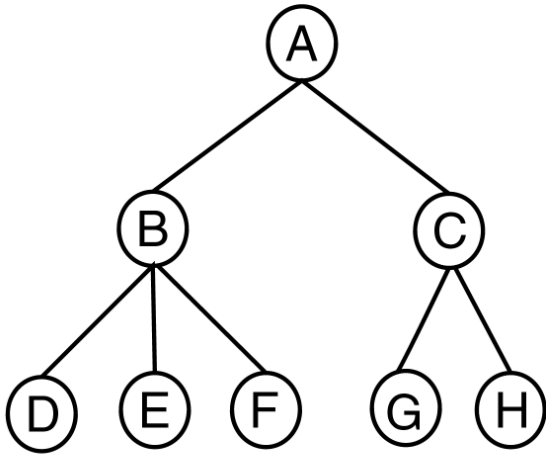
3. R-Tree

- It is a balanced tree structure with index object stored in the leaf node.
- As with Quad Tree the region is divided into successively smaller rectangles (MBRs).
- Rectangles need not be of the same size or number at each level.
- Rectangles may actually overlap.
- Lowest level cell has only one object.
- Tree maintenance algorithms similar to those for B-trees.

Division of space by R tree

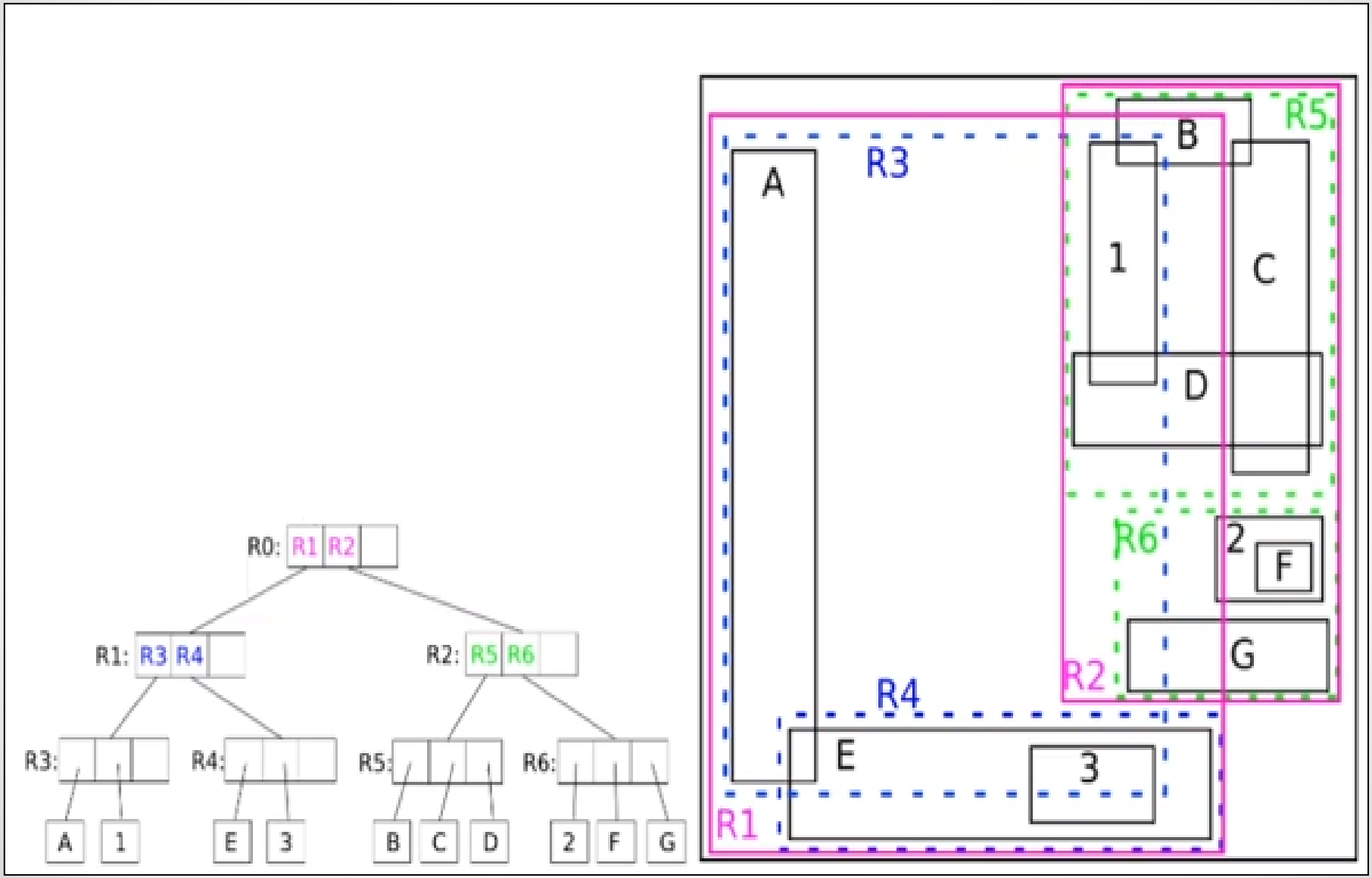


a) Partitioning with MBRs

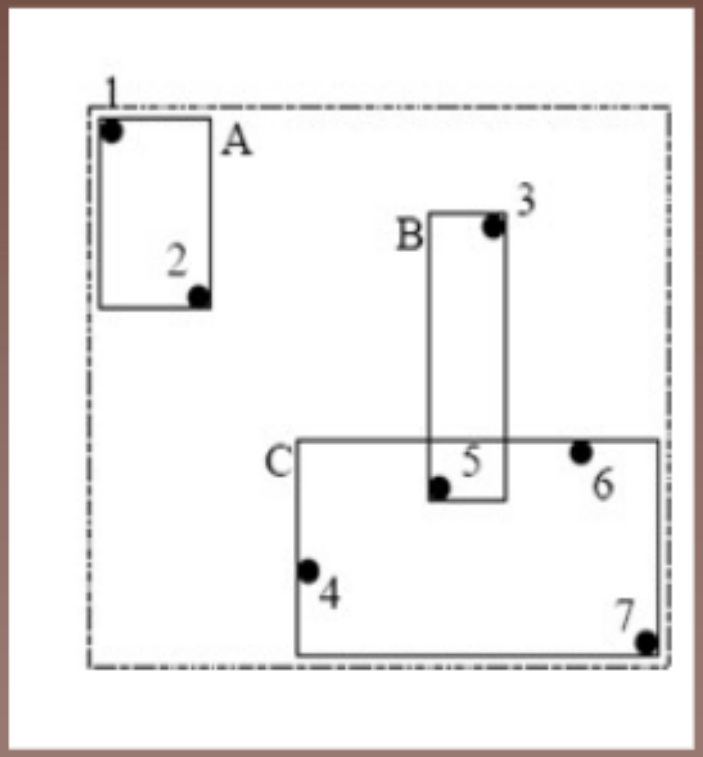


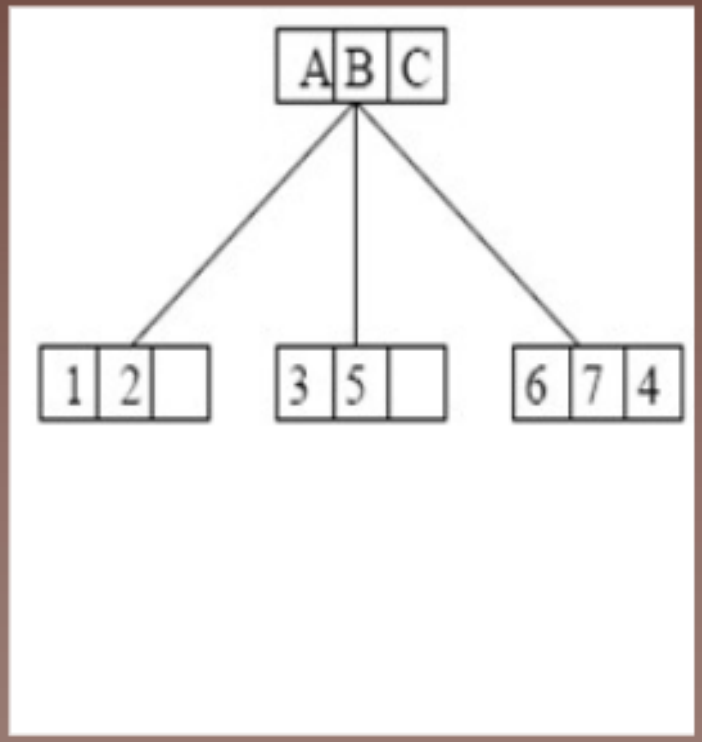
b) R-Tree

Division of space by R tree



Division of space by R tree

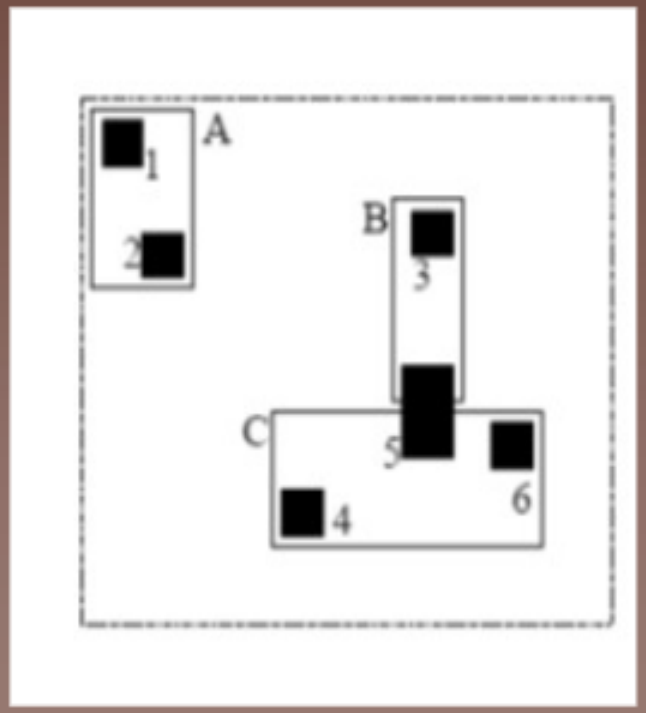


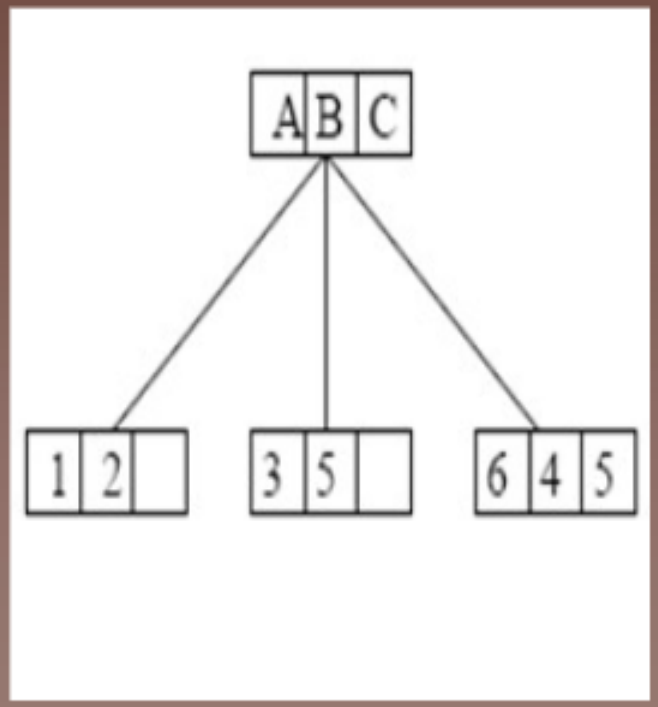


4. R +tree

- It is an extension of R-tree.
- Here bounding rectangle of nodes at one level do not overlap. This feature decreases the number of searched branches of the tree and reduces the time consumption and increases the space consumption .
- Here the data objects are allowed to split so that different parts of one object can be stored in more nodes of one tree level.
- Root has at least two children unless it is a leaf.
- All leaves are at same level.
- There is no constraint on the minimum number of entries at each node.

Division of space by R + tree





Spatial data mining tasks

Basic tasks of spatial data mining are:

1. **Classification** – *finds a set of rules which determine the class of the classified object according to its attributes*
   - **E.g.:** "IF population of city = high AND economic power of city = high THEN unemployment of city = low" or classification of a pixel into one of classes, e. g. water, field, forest.

Spatial data mining tasks

Basic tasks of spatial data mining are:

2. **Association rules** – *find (spatially related) rules from the database.*

- Association rules describe patterns, which are often in the database.
- The association rule has the following form: A → B(s%, c%), where s is the support of the rule (the probability, that A and B hold together in all the possible cases) and c is the confidence (the conditional probability that B is true under the condition of A
- **E. g.** "if the city is large, it is near the river (with probability 80%)" or "if the neighbouring pixels are classified as water, then central pixel is water (probability 80%)."

Spatial data mining tasks

Basic tasks of spatial data mining are:

3. **Characteristic rules** – describe some part of database **e. g.** "bridge is an object in the place where a road crosses a river."

4. **Discriminant rules** – describe differences between two parts of database **e. g.** find differences between cities with high and low unemployment rate.

5. **Clustering** – groups the object from database into clusters in such a way that object in one cluster are similar and objects from different clusters are dissimilar **e. g.** we can find clusters of cities with similar level of unemployment or we can cluster pixels into similarity classes based on spectral characteristics.

Spatial data mining tasks

Basic tasks of spatial data mining are:

6. **Trend detection** – finds trends in database. A trend is a temporal pattern in some time series data. A spatial trend is defined as a pattern of change of a non-spatial attribute in the neighbourhood of a spatial object **e. g.** “when moving away from Brno, the unemployment rate increases” or we can find changes of pixel classification of a given area in the last five years.

CLARA (Clustering Large Applications)

- Drawback of k-medoids (PAM):
  - More costly
  - RAM storage problem
  - Not good for large data sets
  - Complexity $O(k(n-k)^2)$ or $O(k^3 + nk)$

CLARA (Clustering Large Applications)

- CLARA known as Clustering Large Applications algorithm developed by Kaufman and Rousseeuw in 1990.
- It is an improved version of k-medoids to deal with data containing a large number of objects (more than thousand or ten thousand observations) in order to reduce computing time.
- It uses a random sample or small portion of the database instead of taking the whole data set so that it removed RAM storage problem.

CLARA (Clustering Large Applications)

- The PAM algorithm applies on each sample to compute the best medoids.
- The quality of final medoids is measured by average dissimilarity between every object in the whole data set and the medoid of each cluster, defined as the cost function.

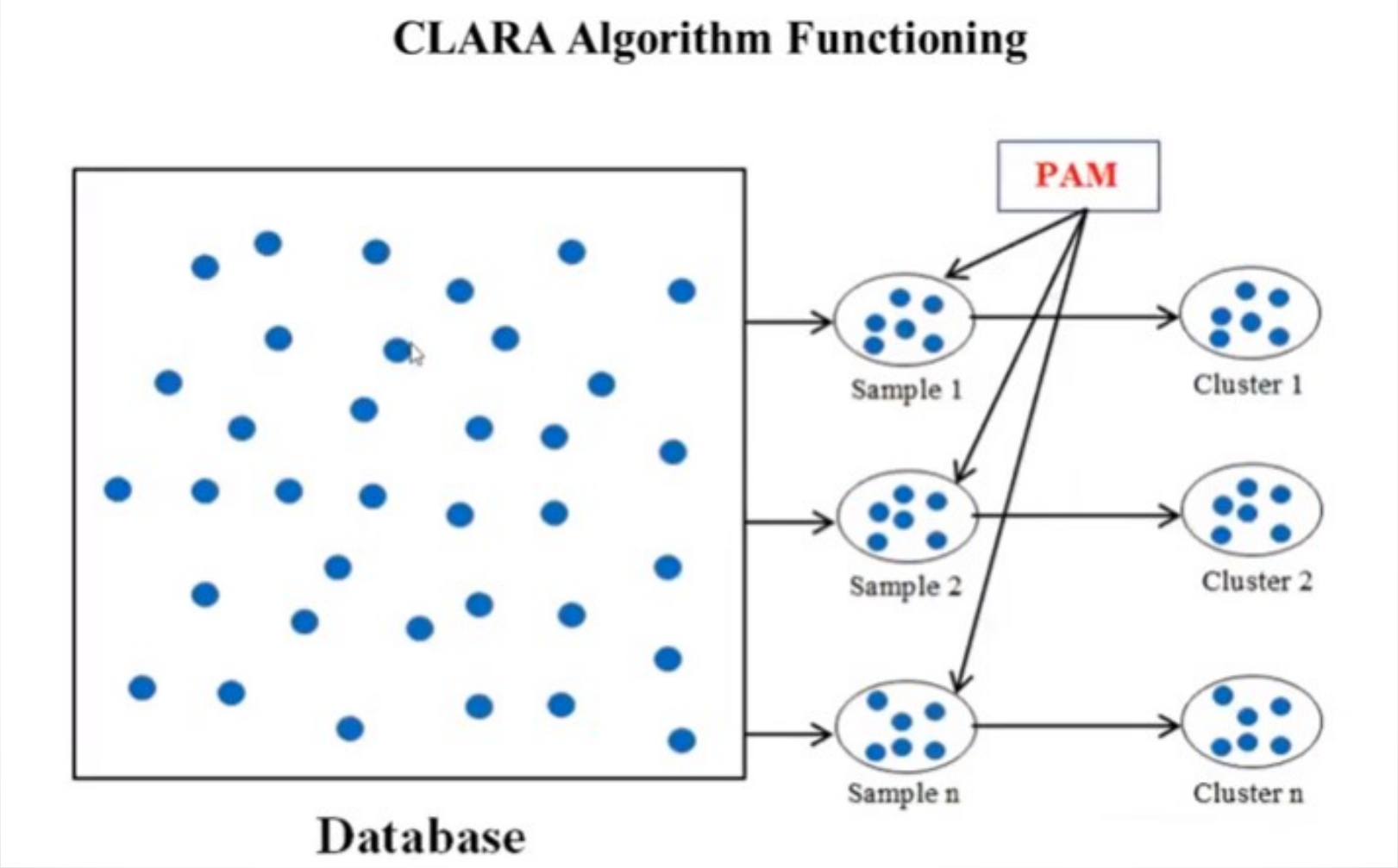
CLARA (Clustering Large Applications)

CLARA Algorithm Functioning

Records	X	Y
1	2.45	5.78
2	7.12	6.45
3	8.60	2.56
4	3.41	3.60
5	4.90	8.69
6	2.61	5.10
.	.	.
.	.	.
.	.	.
1000	6.69	9.50

Sample
S1
S2
S3
S4
S5
.
.
.
.
Sn

CLARA (Clustering Large Applications)



CLARA (Clustering Large Applications)

CLARA Algorithm Steps

Step 1: Draw randomly multiple sample from the original database with fixed size.

Step 2: Apply PAM algorithm on each sample and choose the corresponding k representative objects or medoids and assign each observation of the entire sample to the closest medoid.

Step 3: Calculate mean or sum of the dissimilarities of the observations to their closest medoid for good quality clusters.

Step 4: Recall the sample for which the mean is minimum and continue repeat above steps to the final clustering.

CLARA (Clustering Large Applications)

Complexity of CLARA Algorithm

- The complexity of computing the medoids sample is $O(ks^2+k(n-k))$.
  - s is the size of the sample,
  - k is the number of clusters,
  - n is the total number of objects.

CLARANS (Clustering Large Applications based upon Randomized Search)

CLARANS Algorithm

- k-medoids (PAM) finds the best k medoids between given database.
- CLARA finds the best k medoids among the selected data samples.

- **Limitations of CLARA**:
  - The best k medoids may not be selected during the sampling process, in this case, CLARA will never find the best clustering.
  - If the sampling is biased or partial, we cannot find good quality of clusters.
  - Trade-off efficiency.

CLARANS (Clustering Large Applications based upon Randomized Search)

- CLARANS known as Clustering Large Applications based upon RANdomized Search.
- CLARANS like PAM starts with a randomly selected set of k medoids (or few pairs) instead of examining all pairs, for swapping at the current state.
- If check at most the **"maxneighbor"** number of pairs for swapping and, if a pair with negative cost is found, it update the medoids as a local optimum and restarts with a new randomly selected medoid, set to search for another local optimum.
- CLARANS stops after the **"numlocal"** number of local optimal medoid sets are determined, and return the best among these.

CLARANS (Clustering Large Applications based upon Randomized Search)

CLARANS Algorithm Steps

Step 1: Input parameters *numlocal* and *maxneighbor*. Initialize *i* to 1, and *mincost* to a large number.

Step 2: Set *current* to an arbitrary node in $G_{n,k}$.

Step 3: Set *j* to 1.

Step 4: Consider a random neighbor S of *current* and calculate the cost differential of the two nodes.

Step 5: If S has a lower cost, set current to S, and go to Step 3.

Step 6: Otherwise, increment *j* by 1. If $j \leq maxneighbor$, go to Step 4.

Step 7: Otherwise, when $j > maxneighbor$, compare the cost of *current* with *mincost*. If the previous is less than *mincost*, set mincost to the cost of current and set *bestnode* to *current*.

Step 8: Increment *i* by 1. If $i > numlocal$, output *bestnode* and halt. Otherwise, go to Step 2.

CLARANS (Clustering Large Applications based upon Randomized Search)

Comparison between CLARA & CLARANS

- Like CLARA, CLARANS does not check every neighbors of a node.
- Unlike CLARA, CLARANS does not restrict its search to a particular graph. It search the original graph $G_{n,k}$.
- CLARA draws a sample of *nodes* at the beginning of a search, CLARANS draws a sample of *neighbors* in each step of a search.
- CLARANS gives higher quality clustering than CLARA.
- CLARANS requires a very small number of searches than CLARA.

CLARANS (Clustering Large Applications based upon Randomized Search)

Advantages of CLARANS Algorithm

- It is more efficient and scalable than both PAM and CLARA.
- It does not restrict the search to any particular subset of objects.
- It improves the time complexity based on randomized search.
- It used for large database.
- It gives higher quality clustering.
- It requires very small number of searches.

Web Mining

• • •

University Question

- What is Web Structure Mining? List the, approaches used to structure the web pages to improve on the effectiveness of search engines and crawlers. Explain Page Rank technique in detail? **(Dec 2019) 10 marks**
- With respect to web mining, is it possible to detect visual objects using meta-objects? **(May 2019) 5 marks**

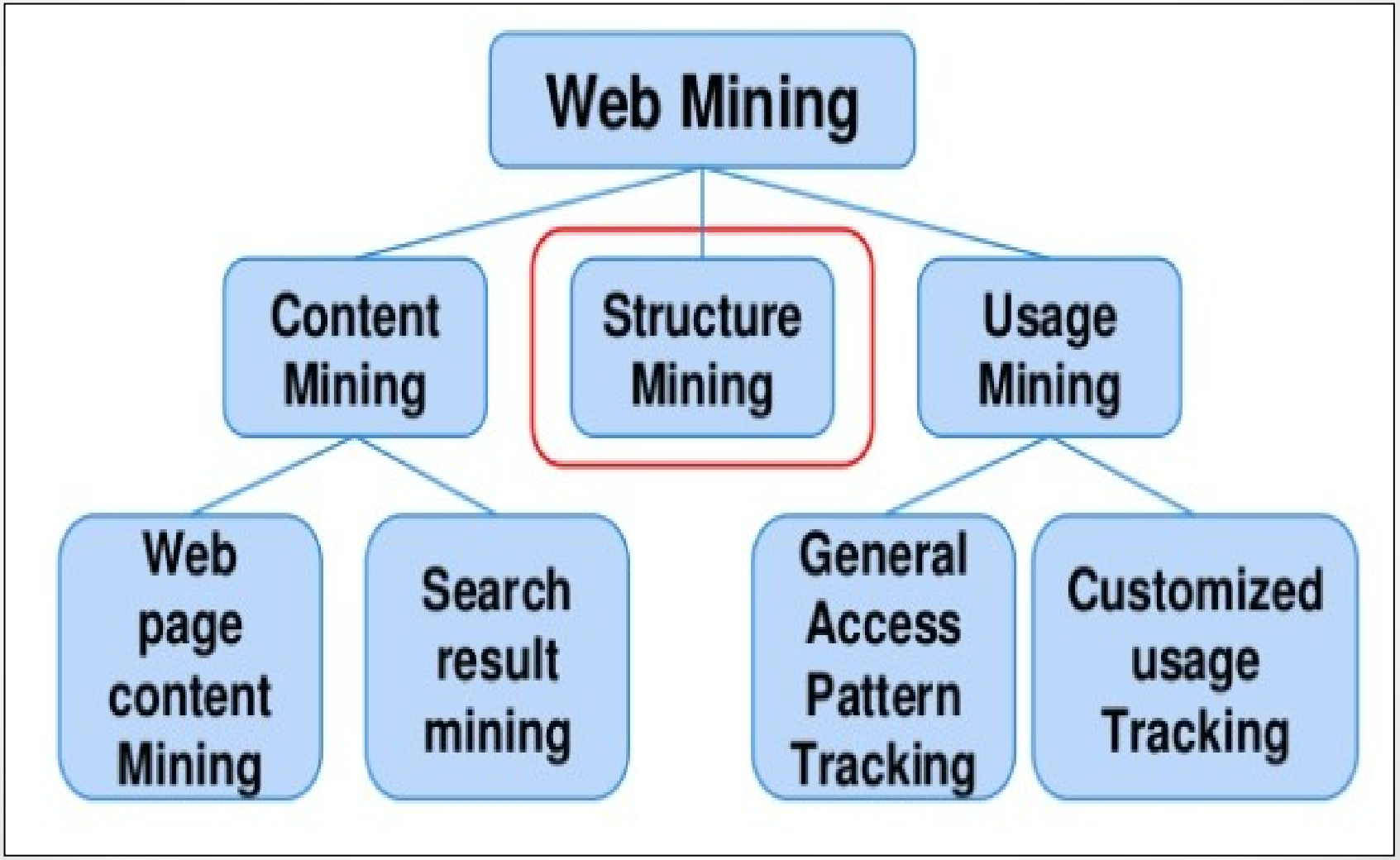
What is Web Mining

The application of **data mining** techniques to discover patterns from the Web.

Web data consist of:

- Web Content (text, images, records, etc)
- Web Structure (hyperlinks, tags, etc)
- Web Usage (http logs, app server logs, etc)

Web Mining Taxonomy



Web Mining Taxonomy

- **Web content mining**: focuses on techniques for assisting a user in finding documents that meet a certain criterion
- **Web structure mining**: aims at developing techniques to take advantage of the collective judgement of web page quality which is available in the form of hyperlinks
- **Web usage mining**: focuses on techniques to study the user behaviour when navigating the web

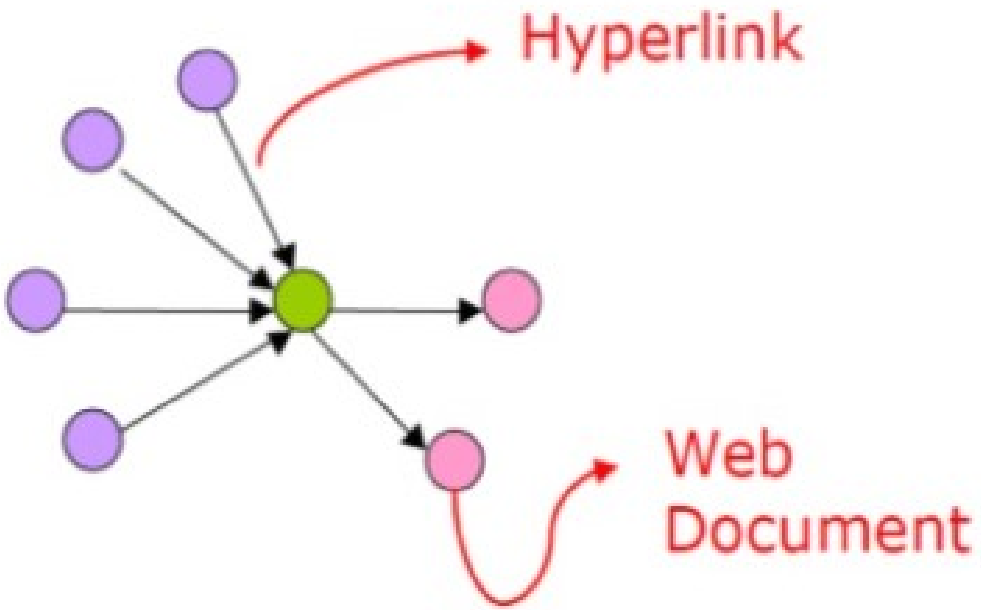
  (also known as Web log mining and clickstream analysis)

1. Web Content Mining

- Can be thought of as extending the work performed by basic search engines
- Search engines have crawlers to search the web and gather information, indexing techniques to store the information, and query processing support to provide information to the users
- Web Content Mining is: the process of extracting knowledge from web contents

2. Web Structure Mining

The structure of a typical Web graph consists of Web pages as nodes, and hyperlinks as edges connecting between two related pages



2. Web Structure Mining

Web Structure Mining can be is the process of discovering structure information from the Web

- This type of mining can be performed either at the (intra-page) document level or at the (inter-page) hyperlink level
- The research at the hyperlink level is also called **Hyperlink Analysis**

2. Algorithm for Web Structure Mining

PageRank algorithm (Google Founders)

* Looks at number of links to a website and importance of referring links
* Computed before the user enters the query.

HITS algorithm (Hyperlinked Induced Topic Search)

* User receives two lists of pages for query (authority and link pages)
* Computations are done after the user enters the query.

Web Structure Mining- Page Rank

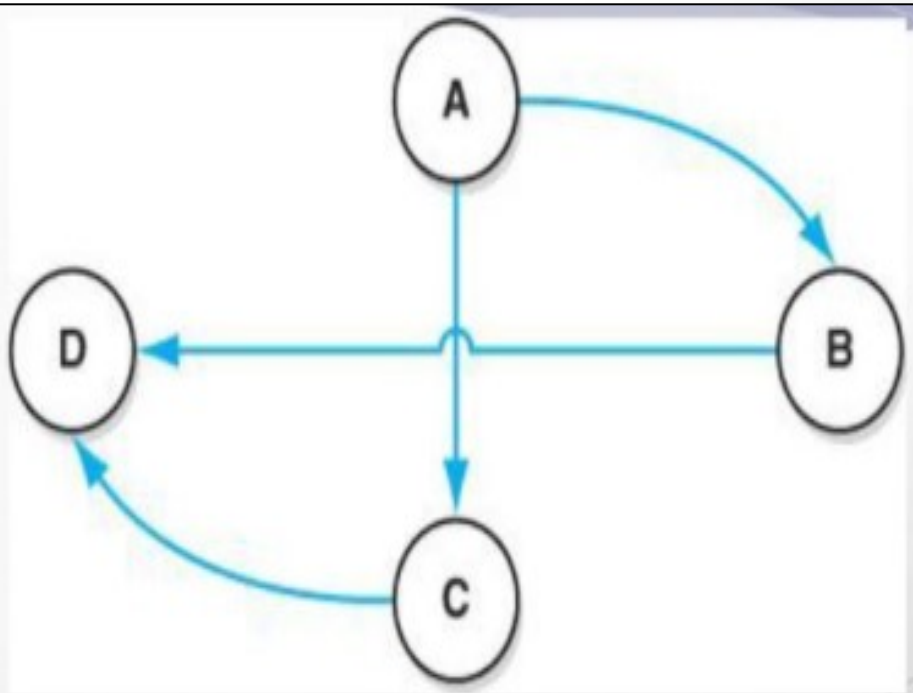
* The idea of the algorithm came from academic citation literature.
* It was developed in 1998 as part of the Google search engine prototype
* Studies **citation relationship** of documents within the web.
* Google search engine ranks documents as a function of both the **query terms** and the **hyperlink structure** of the web.

Web Structure Mining- Page Rank

* The PageRank produces ranking **independent** of a user's query.
* The importance of a web page is determined by the number of other **important** web pages that are pointing to that page and the **number** of out links from other web pages.

Web Structure Mining- Page Rank

Page **A** is a backlink of page **B** and page **C**, while page **B** and page **C** are backlinks of page **D**.



Backlink = Outlink= OutDegree

Web Structure Mining- Page Rank

The PageRank of a page *u* is computed as follows:

$$PageRank(u) = (1-d) + d \sum_{(v,u)\in E} \frac{PageRank(v)}{OutDegree(v)}$$

where, OutDegree(v) represents the number of links going out of the page *v* and parameter *d* be a damping factor, which can be a real number between 0 and 1.

The value of *d* is generally taken as 0.85.

Web Structure Mining- Page Rank

PageRank Algorithm: Assume that there are n linked pages.

Let $S_G = (V, E)$. (V = set of pages, E = set of hyperlinks between pages)

Initialize $PageRank(p) = 0$ for all the pages.

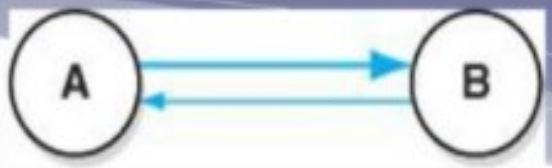
Repeat until *PageRank* vector converges (i.e. stabilize or do not change):

For all pages $u \in V$

$$PageRank(u) = (1-d) + d \sum_{(v,u) \in E} \frac{PageRank(v)}{OutDegree(v)}$$

Return *PageRank* vector

Page Rank Example



OutDegree(A) = 1 and OutDegree(B) = 1).

Here, we do not know what their PageRanks should be to begin with, so we can take a guess at 1.0 , assuming d=0.85, and perform following calculations

PageRank(A)= (1 – d) + d (PageRank(B)/1)
PageRank(B)= (1 – d) + d (PageRank(A)/1)

PageRank(A)= 0.15 + 0.85 * 1=1

PageRank(B)= 0.15 + 0.85 * 1=1

We calculated that the PageRank of A and B is 1.

Page Rank Example

Now, we plug in 0 as the guess and perform calculations again:

PageRank(A) = 0.15 + 0.85 * 0= 0.15

PageRank(B) = 0.15 + 0.85 * 0.15= 0.2775

We have now another guess for PageRank(A) so we use it to calculate PageRank(B) and continue:

PageRank(A) = 0.15 + 0.85 * 0.2775 = 0.3859

PageRank(B) = 0.15 + 0.85 * 0.3859 = 0.4780

Page Rank Example

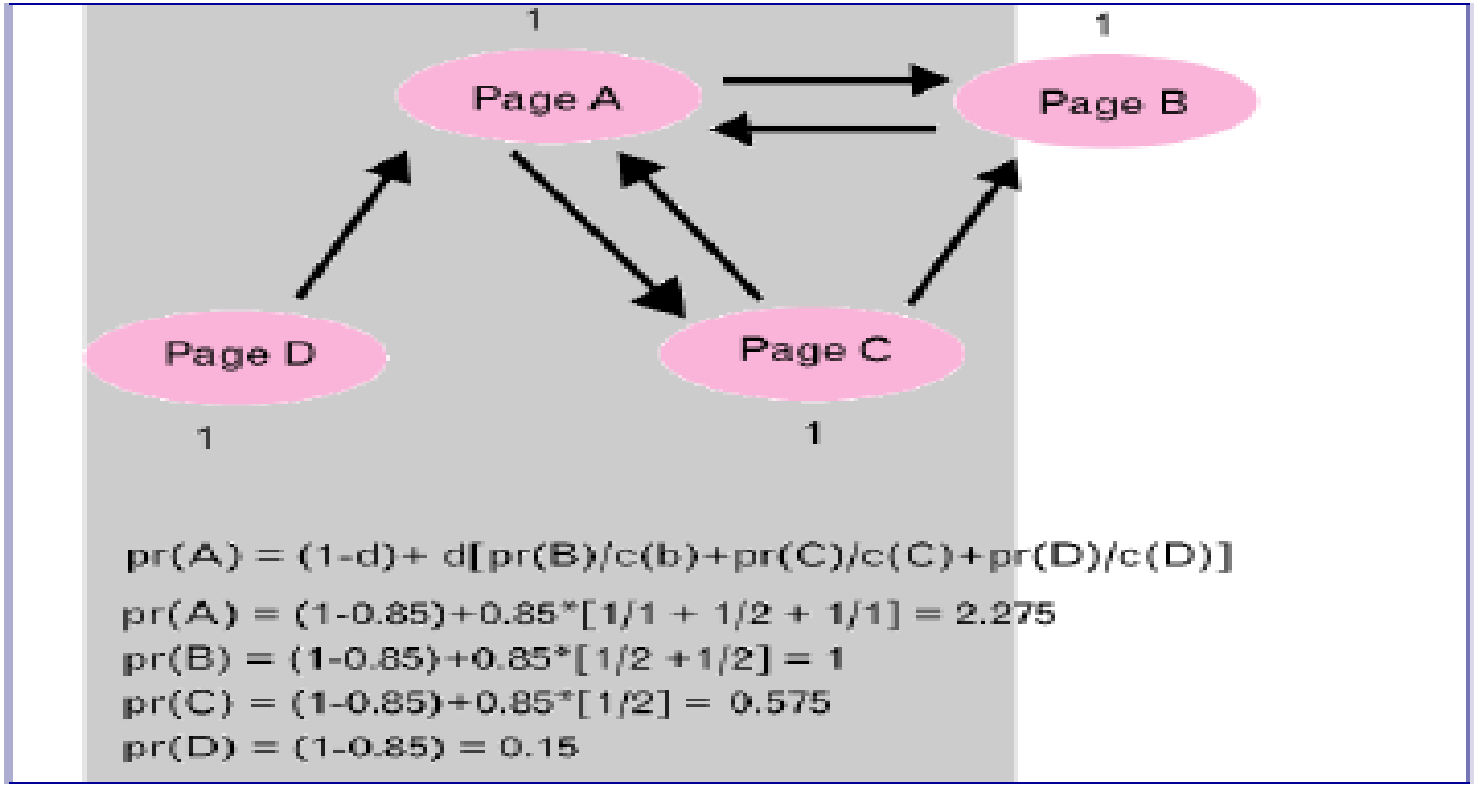
Repeating the calculations, we get:

PageRank(A) = 0.15 + 0.85 * 0.4780 = 0.5563

PageRank(B) = 0.15 + 0.85 * 0.5563 = 0.6229

If we repeat the calculations, eventually the PageRanks for both the pages converge to 1.

Page Rank Example



Repeat until pagerank vector converges...

Page Rank Example

Remarks on PageRank Algorithm:

* A page with no successors has no scope to send its importance. As well, a group of pages that have no links out of the group will eventually collect all the importance of the Web.

Page Rank Example

Sample Scores with Their Meaning

0/10:	The site or page is probably new.
3/10:	The site is perhaps new, small in size, and has very little or no worthwhile arriving links. The page gets very little traffic.
5/10:	The site has a fair amount of worthwhile arriving links and traffic volume. The site might be larger in size and gets a good amount of steady traffic with some return visitors.
8/10:	The site has many arriving links, probably from other high-PageRank pages. The site perhaps contains a lot of information and has a higher traffic flow and return visitor rate.
10/10:	The Web site is large, popular, and has an extremely high number of links pointing to it.

HITS Algorithm

HITS (Hyperlink-Induced Topic Search)

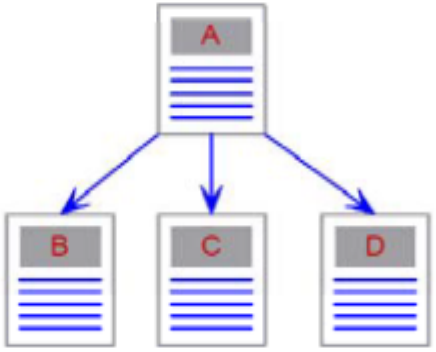
- HITS uses hyperlink structure to identify authoritative Web sources for broad-topic information discovery

  Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604-632.

- Premise: Sufficiently broad topics contain communities consisting of two types of hyperlinked pages:
  - Authorities: highly-referenced pages on a topic
  - Hubs: pages that "point" to authorities
  - A good authority is pointed to by many good hubs; a good hub points to many good authorities

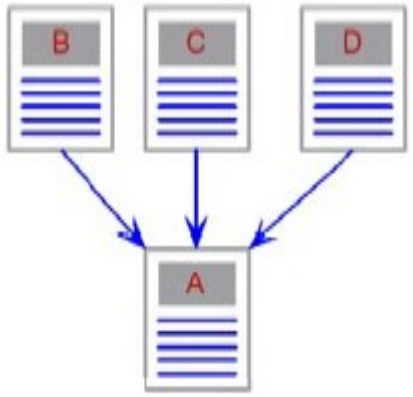
HITS Algorithm

Hubs



Pages that link to a collection of authoritative pages on a broad topic
pages point to interesting links to authorities = relevant pages

Authorities



Relevant pages of the highest quality on a broad topic

HITS Algorithm

Let ***a*** is the vector of authority scores and ***h*** be the vector of hub scores

a=[1,1,….1], h = [1,1,…..1] ;

do

a=Aᵀh; (Authority update role)

h=Aa; (Hub update role)

Normalize **a** and **h**; (divided each node to square sum of other nodes)

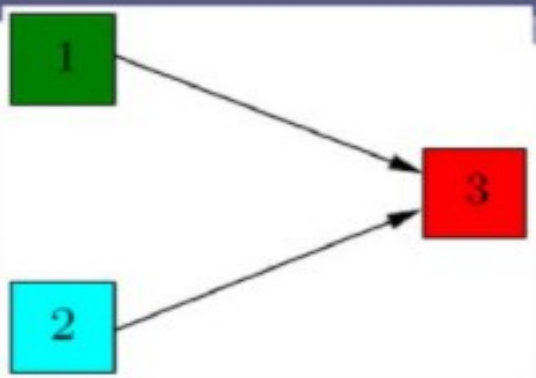
while ***a*** and ***h*** do not converge (reach a convergence threshold)

$a^* = a$;

$h^* = h$; **return a^*,h^***

The vectors a* and h* represent the authority and hub weights

HITS Algorithm



$$A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad A^t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

$$u = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Initial hub vector

$$v = A^t \cdot u = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

Authority vector

$$u = A \cdot v = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}$$

Updated hub vector

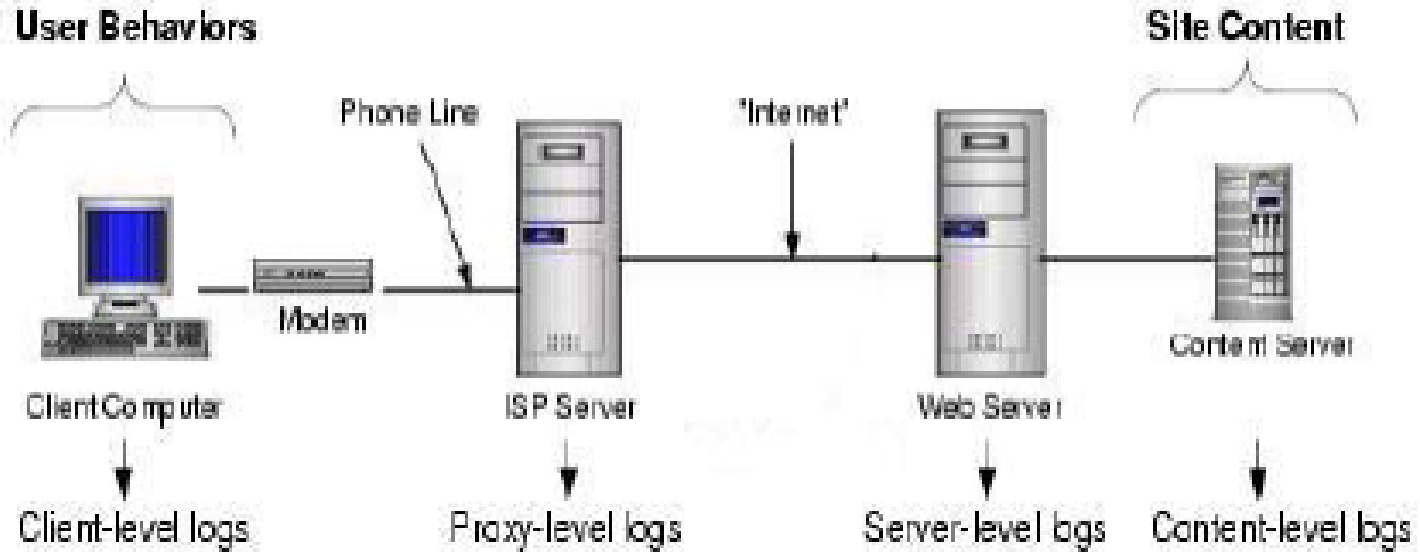
3. Web Usage Mining

- Pages contain information
- Links are "roads"
- How do people navigate over the Internet?
- ⇒ Web usage mining (Clickstream Analysis)

- Information on navigation paths is available in log files.
- Logs can be examined from either a client or a server perspective.

3. Web Usage Mining

Data Sources



Thank You