Midterm 1.

Date: 2/21/2005

Due: 2/23/2005 at 11.55am (submit via moodle)

Student Name: Abhishek Kumar Agrawal

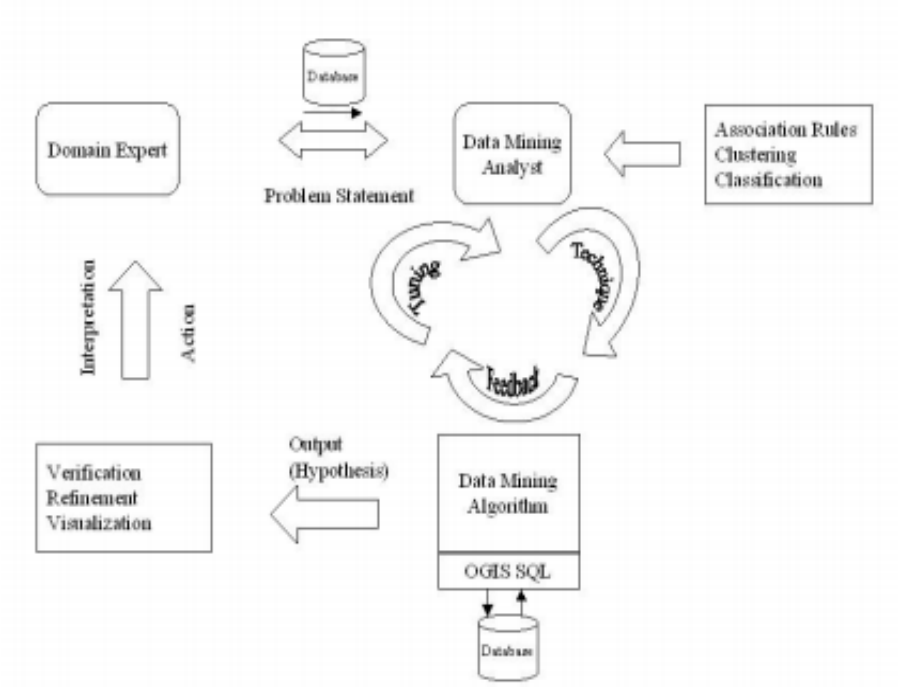
Student ID: akagrawa

Guidelines:

1. Answer all questions to the extent possible, be brief and to the point
2. You can use appropriate resources (papers, books, discussions with friends), but answers should be yours.
3. If you are in doubt, make better judgment; write your assumptions if any clearly, etc. Please note only reasonable assumptions are entertained (instructor decision is final).
4. Hints: Treat it as in class exam; answer easy questions first, answer all questions.
5. Submit pdf file and "R" code as separate text file.
6. File names should be: your_last_name_student_id.pdf (or .R)

Q 1: List and explain major steps in spatial data mining.

Spatial data mining is the process of discovering interesting and previously unknown, but potentially useful patterns from the spatial and spatiotemporal data[1]. Just like any traditional data mining techniques, the complete spatial data-mining process is a combination of many subprocesses. Some important and key subprocesses are data extraction and data cleaning, feature selection, algorithm design and tuning, and the analysis of the output when the algorithm is applied to the data[2]. Among these steps, the notion of spatial autocorrelation that similar objects tend to cluster in geographic space is central to spatial data mining which distinguish it from other data mining approaches.



a. **Data Extraction:** In this step, spatial data is extracted from spatial data sources i.e spatial databases or image files, or from some other spatial data sources (sensors etc.) and loaded into your data processing frameworks.

b. **Data Cleaning:** In this step, the data is pre-processed in order to avoid some missing or garbage value. Since the spatial data is autocorrelated we cannot replace missing values with any arbitrary values. It must be in consensus with its neighbourhood values. We can also perform various transformation based on the nature of the model we are willing to apply.

c. **Feature selection:** In this step, we can perform correlation analysis to find different uncorrelated features that captures the maximum variability in the dataset.

d. **Algorithm design and tuning:** In this step, we design and implement an algorithm that generated the model that can answer our business requirement i.e. spatial classification, clustering, association rule mining or anomaly detection. Based on our requirement with the data, we implement and tune the best discovered model that outputs a hypothesis which can in the form of model parameters, rules or labels.

e. **Analysis of output:** This is probably the last step in the process before the re-iteration or final output. This includes verification, refinement and visualization of the patterns we obtained from previous step. For spatial data this part is typically done with the help of GIS software or similar software frameworks.

Q2: Compare and contrast 3 different spatial clustering algorithms. List at least two advantages or disadvantages of each method.

COD - CLARANS	**STING Clustering**	**DBClu-c**
● Partition based clustering This clustering algorithm[3] defines the relationship between obstacles and data objects by visibility graphs to compute obstructed distances between data objects. And then combine the data objects into clusters that are not visually obstructed.	● Grid Based Clustering STING[4] (Statistical Information Grid) algorithm first quantize the clustering space into a finite number of cells and then perform the required operations on the quantized space. Cells that contains more than certain number of points(density) are treated as dense. These dense cells are connected to form cluster.	● Density Based Clustering DBClu-c[5] is an extension of DBScan clustering algorithm that takes advantage of constraint modeling to efficiently cluster data while considering all physical constraints. Here the constraints are captured as obstacles and the modeling of such physical obstacles is done using polygons as obstacles and the object visibility the clustering criterion.
<u>Advantages:</u> **(1)** It guarantees to provide a cluster solution that have minimum distance to the cluster centers despite of having obstacles. **(2)** The performance in the model can be observed by	**(1)** STING is computationally quite fast as compare to other approaches, it has a much smaller response time and computational complexity is linearly proportional to the number of leaves.	**(1)** These models can detect clusters of arbitrary shapes while considering all physical constraints and are insensitive to noise, input order, and the difficulty of constraints.

utilizing the pre-processing computations performed and the results of clustering are far better than normal clustering algorithms that do not consider obstacles. Using the concept of micro-clustering enhances the clustering speed.	**(2)** STING can support different resolution of query result as it maintains the hierarchy of the spatial regions. **(3)** Clustering is performed on summaries and not individual objects; complexity is usually O(#populated-grid-cells) and not on O(#objects)	**(2)** Owing to the effectiveness of the density-based approach, DBClu-c finds clusters of arbitrary shapes and sizes with minimum domain knowledge. **(3)** Enhance performance of processing large number of obstacles by reducing search spaces using SR-tree type tree data structures.
Disadvantages: **(1)** Require expensive preprocessing steps. **(2)** It inherits the drawbacks of CLARANS ,i.e. (i) parameterize, number of clusters(k) and the performance decreases with increase of value of k (ii) Micro clustering method, detection of only spherical shaped clusters **(3)** Micro-clustering method will be ineffective in case of having more obstacles than the data points.	**(1)** All the cluster boundaries are either horizontal or vertical, and no-diagonal boundary is detected. In other words, arbitrary shaped clusters are not detected. **(2)** The accuracy of the clustering result may be degraded at the expense of simplicity of this method. Also like density based clustering, this method also suffers from varying density in data.	**(1)** The algorithm is constrained for two dimensional dataset and thus cannot be applied for multi-dimensional spatial datasets with 3D physical constraints. **(2)** Certain must link physical constraints such as connectivity due to bridges and pedestrian-ways are not considered into the model which provides visibility scope from obstructed spatial data objects.

Q 3: Given a spatialdatabase, consisting 4 different layers:(1) roads (lines), (2) bars (points), (3) schools, (4) religious establishments. Your objective is to design spatial data mining framework to study the relationship between different places and crime. (You can use different types of predicates – see OpenGIS simple feature specification, different spatial data mining algorithms). Clearly write step-by-step description of the proposed solution (try to use graphical examples to clarify each step).

Solution : In this data mining task, we are given with a spatial database that has 4 different layers of objects i.e. roads(lines), bars(points), school and religious establishments. Now we need study the relationship between crime occurring and different locations. The spatial data mining method applicable for this study is **co-location mining**.

Co-location mining can be viewed as an association mining method, where we can define objects or events in the form of transaction. And further these transactions can be mined using association rule mining techniques to find the association with crime and different places. Now, we can elaborate the process in step by step manner in order to find our desired co-location patterns.

Step 1 : Defining feature specification : We can use some basic spatial relationships provided by OpenGIS databases such as Equals, Disjoints, Overlaps, Contains etc for our dataset. In order to maintain and retrieve data from our spatial databases, we can define the certain spatial analysis function as below:

a. Distance(anotherGeometry:Geometry): Double—Returns the shortest distance between any two points in the two geometries as calculated in the spatial reference system of this Geometry.

Eg : Distance (Bar X, Road Y) ,
Distance (School A, Religious_Place B) etc.

b. Intersection(anotherGeometry:Geometry): Geometry—Returns a geometry that represents the point set intersection of this Geometry with anotherGeometry.

Eg: Intersection(Road X, Road Y)

c. GetCrimeLocation(crimeObj, Location) : Geometry - Returns a geometry that represents the nearest spatial object in our database where the crimeObj has been found. Here the numeric values indicate Location coordinates (latitude : longitude).

Eg : GetCrimeLocation(“Gun”, “22 :12”) = Bar X
GetCrimeLocation(“Knife”, “22 : 20”) = Road Y

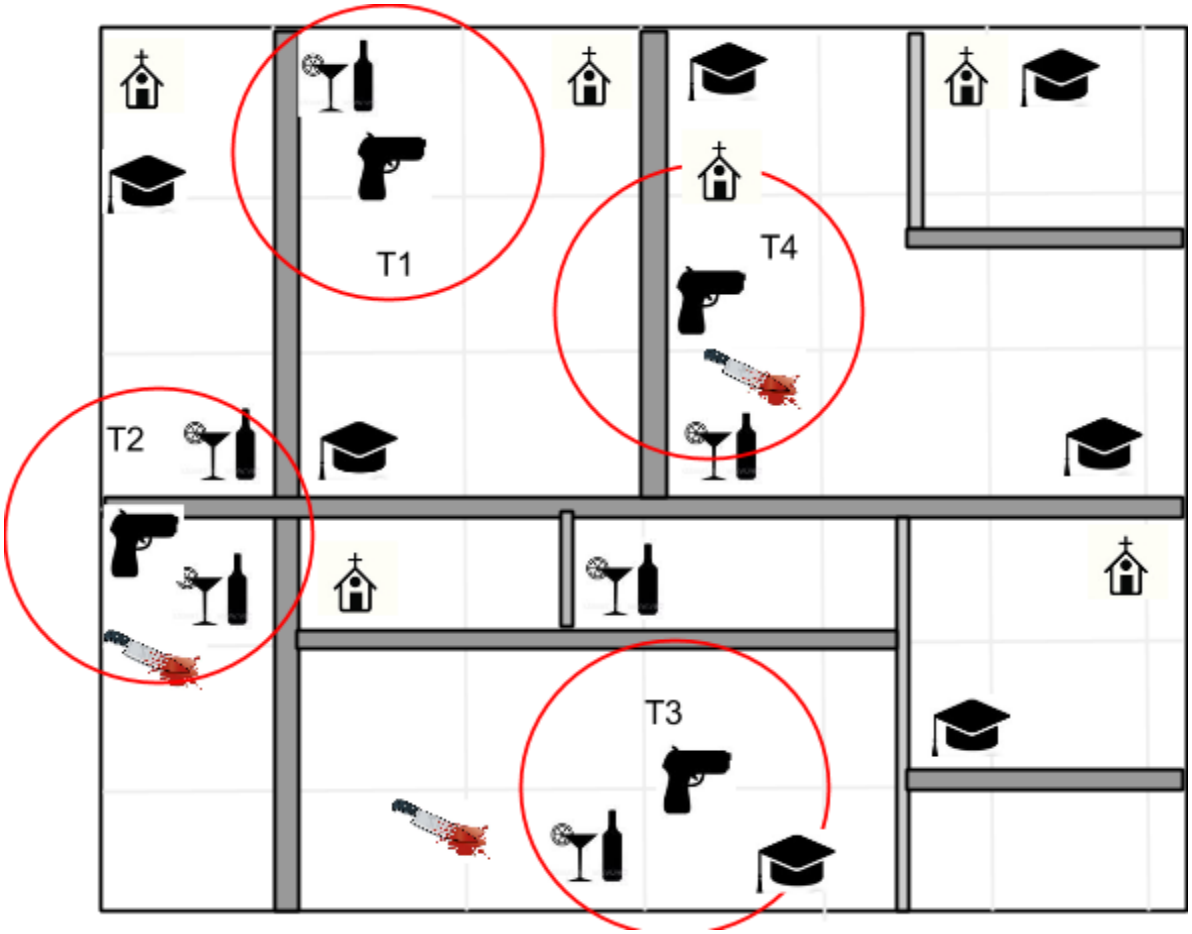
d. getAllCrimeObjLocation(crimeObj) : [Location] - Returns a list of location coordinates from our database where the crimeObj have been found.

Eg : getAllCrimeObjLocation(“Gun”) = [“22:12”, “22:20”, “42:43”]
getAllCrimeObjLocation(“Knife”) = [“12:12”, “14:20”, “42:43”]

Step 2: Co-location Mining Technique: Now we can define our co-location mining approach : Reference Feature Centric co-location mining.

In this approach, we will define our transactions based on the objects located near the crime objects in certain defined proximity range.

Step 3: Transactionization : From the below diagram and the reference object we can have the transactions as follows:



Gun = “G” **Road = “R”** **School = “S”**
Blood Knife =”K” **Bar = “B”** **Church = “C”**

Reference Item “Gun” => G

TID	**Item-set**
T1	{G, B, R}
T2	{G, B, K, R}
T3	{G, B, S}
T4	{G, B, K, C, R}

We now define minimum support = 3

Step 4: Using Apriori Principle to find Frequent ItemSet:

1-item frequent-itemset F_1 . In this step, we will run a single pass over our dataset to determine the support of each item. Upon completion of this step, the set of all frequent 1-itemset, F_1 will be known.

Candidate 1- Itemsets

Item	Count
G	4
R	3
B	4
K	2
S	1
C	1

Since min_support = 3, few items will be pruned in this step.
This means 1-item frequent-itemset F_1 **= [G, R, B, K]**

Following the same Apriori principle steps we can summarize below frequent item sets:
1- item Frequent itemset as F_1 = [G, R, B, K]
2- item Frequent itemset as F_2 = [{G, R}, {G, B}, {R, B}]
3- item Frequent itemset as F_3 = [{G, R, B}]

Step 5: Association Rules Identification: From the above frequent itemset we can define crime related association rules as follows:

Rule { R, B} = > G i.e. { Road, Bar } => Gun (Crime)
support = 4/4 => 100% confidence = 4/4 => 100%

Here we can repeat the process for other crime objects also to obtain more crime related co-location patterns. Similarly we can have event driven co-location mining where crime events can have co-location patterns with respect to spatial data.

Q 4: Describe two distinct ways to model spatial constraints in clustering. List at least two advantages and disadvantages for each.

Solution: Modeling spatial constraints have always been a key requirement in spatial clustering techniques. Hence there are many models that either have enhanced the existing clustering algorithm or devised an efficient to handle constraints from scratch. Here we can highlight two distinct clustering techniques that model physical obstacles as spatial constraints while forming clusters.

Density Based Clustering: In this approach, density based clustering algorithms like DBSCAN, DENCLUE etc. are enhanced to capture the spatial constraints in their clustering model. These improvements have shown better clustering results over spatial data than the normal approach. In these method, the obstacles are modeled using obstructed distances that does not allow points to form cluster together or similarly constructing obstacles as polygons that can prevent the visibility of the spatial points forbidding them to cluster together. We have following advantages and disadvantages using this approach:

Advantages:

a) These models can detect clusters of arbitrary shapes while considering all physical constraints and are insensitive to noise, input order, and the difficulty of constraints.

b) For space search in which they employ the fast and efficient data structures like quad-tree, kd-tree, SR-tree that guarantees fast lookups and hence scales with the data.

c) Enhance performance of processing large number of obstacles by reducing search spaces. Employing such efficient techniques leverages up near linearithmic solutions which is quite efficient and fast compare to other clustering techniques in spatial domain.

Disadvantages:

a) Spatial density based clustering algorithm inherits the disadvantages of regular density based clustering algorithms. Since its basic approach is to cluster data objects that have somewhat similar density, so it cannot perform well for varying density dataset which is common in spatial data mining.

b) Some of these algorithms are limited to 2-Dimensional data and cannot model altitude type constraints while clustering. Also bridge type connecting constraints are difficult and complex to model.

Partition Based Clustering: Partition based clustering algorithms start with each pattern as a single cluster and iteratively reallocates data points to each cluster until a stopping criterion is met. These method tend to find clusters of spherical shapes. These models when enhanced to encounter physical obstacles as constraints provides robust solutions for constraint based clustering. K-Mediods, and K means variants are the commonly used algorithms in this space. Some of the recent algorithms in this category are: Partitioning around mediods(PAM), clustering large applications(CLARA), Clustering with obstructed distance version(COD) of CLARANS i.e. COD-CLARANS.

Advantages:

(1) These are the most common clustering algorithms and modeling the physical obstacles as the visibility criterion are comparatively easy in such algorithms. Using fast index based data structures such as R*tree are used to boost up the lookup operations while checking for visibility of data points in the presence of obstacles.

(2) Checking all possible subset systems is even more computationally infeasible when checking against constrains, certain greedy heuristics such as micro-clustering

approaches are used in the form of iterative optimization which provides significant performance gain.

(3) Distance function can change the space to decrease constraint violations made by cluster assignments and hence can even model for different resolutions based constraints on the spatial datasets.

Disadvantages:

(1) Partition based algorithms are mostly parameterized and their performance varies based on the parameters chosen (number of clusters, k) and more oftenly it is difficult to find the optimal number of clusters.

(2) The model objective of such algorithms are to minimize the squared distance errors and hence tend to give clusters of spherical shapes. The performance with arbitrary shaped clusters having obstacles are comparatively poor than density based clustering algorithms.

(3) These models are susceptible to initial assignments of prototypes and sensitive towards the noise or outlier data points.

Q 5: Extend Gaussian Mixture Model (GMM) based clustering for large data using sampling. Use the data from HW2 (ilk-3b-1024.tif). As a baseline method, you can use "MClust" package in R.

Solution :

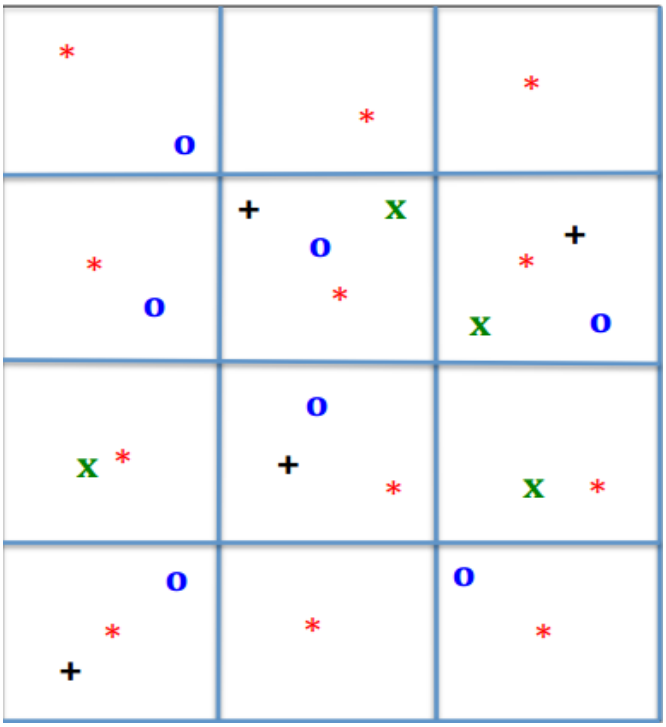
Please refer abhishek_akagrawa_midterm1.R file for the code.
2-D plot for each sample cluster is present inside img folder.

	Sample Size	
Sample 1	20972	~ 2 %
Sample 2	14980	~ 1.4 %
Sample 3	11651	~ 1.1%
Sample 4	10486	~ 1.04%
Sample 5	10488	~ 1.04%

Model Merging : The models are merged pairwise by observing the mean and variance values of independent clusters. Clusters having comparable mean and variance are considered to be part of same gaussian distribution. Hence those models are merged having their new cluster mean and variance as an average of two models. Likewise we can generalize this approach to merge rest of the generated model to obtain our final global model. This model is then used to predict the final cluster labels for the dataset.

Please refer abhishek_akagrawa_midterm1.R file for the code.
Here we can have improvement on matching clusters using pairwise KL divergence method.

Q 6. For the given dataset below:



Assume unit square grids (all grids width = height)
(a) Use the grids to transactionize the data and apply "Apriori" algorithm to find frequent itemsets with support count of 3. Show each step.

Solution: In order to apply Association rule mining over given Spatial data, the data is needed to be transactionize. And from the above data we can assume that every single unit grid represents a unique transaction.
In order to find the frequent-itemset with support count 3 using Apriori algorithm, the following steps are applied:

Step 1 : Listing grids as transactions in the transaction table

* = A	o = B	+ = C	x = D

Assuming the symbols to be replaced by alphabets as listed above, and transactions taken row-wise.

TID	Item-set
1	{A, B}

2	{A}
3	{A}
4	{A, B}
5	{A, B, C, D}
6	{A, B, C, D}
7	{A,D}
8	{A, B, C}
9	{A, D}
10	{A, B, C}
11	{A}
12	{A,B}

Now we can apply the traditional apriori algorithm to find frequent-itemset on our transactionized data.

Step 2: 1-item frequent-itemset F_1 . In this step, we will run a single pass over our dataset to determine the support of each item. Upon completion of this step, the set of all frequent 1-itemset, F_1 will be known.

Candidate 1- Itemsets

Item	**Count**
A	12
B	7
C	4
D	4

Since min_support = 3, no items will be pruned in this step.(All green)
This means 1-item frequent-itemset F_1 **= [A, B, C, D]**

Step 3: Now according to apriori principle, we will generate k frequent-itemset using k-1 frequent itemsets found in the previous step. Hence for 2-item frequent-itemset F_2 will be generated using the combination of F_1 itemsets as shown below.

Candidate 2-Itemsets

Itemset	Count
{A,B}	7
{A,C}	4
{A,D}	4
{B,C}	4
{B,D}	2
{C,D}	2

Since min_support = 3, 2 itemsets {B,D} and {C,D} are pruned in this step. Rest itemset are considered as frequent-itemset.
This means 2-item frequent-itemset F_2 **= [{A, B}, {A, C}, {A,D}, {B,C}]**

Step 3: Now for 3-item frequent-itemset F_3 will be generated using the combination of F_2 itemsets as shown below.

Candidate 3-Itemsets

Itemset	Count
{A,B,C}	4

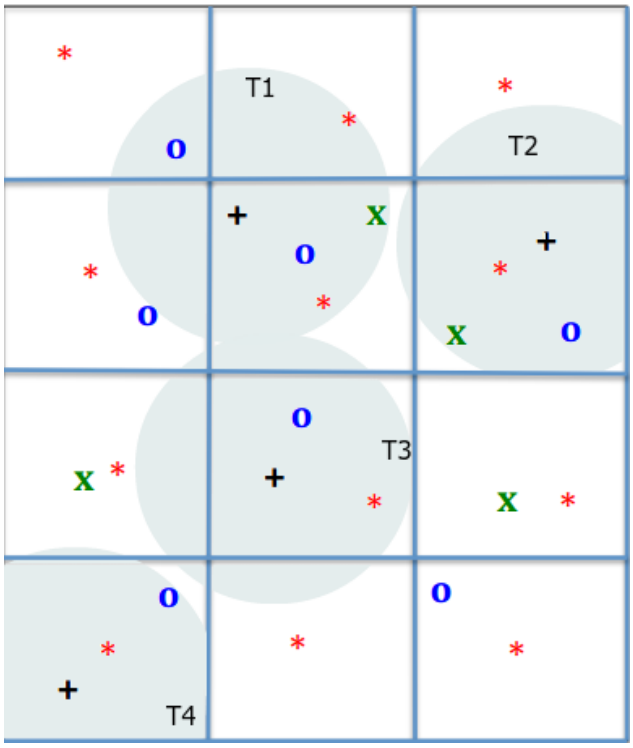
Since min_support = 3, 3 itemsets {A,B,C} is the only 3 Candidate and frequent itemset.
This means 3-item frequent-itemset F_3 **= [{A, B, C}]**

To summarize, we have
1- item Frequent itemset as F_1 = [A, B, C, D] => [* , o, +, x]
2- item Frequent itemset as F_2 = [{A, B},{A,C}, {A,D}, {B,C}] => [{*, o}, {*, +}, {*, x}, {o,+}]
3- item Frequent itemset as F_3 = [{A, B, C}] => [{ * , o, +}]

(b) Repeat question 6(a) for a reference feature "+"

Solution: With respect to the reference feature "+", we need to define transaction that are geographically close to that reference point[6].

Assumption : The proximity distance is taken to be a radial distance "g_close" = ε
Here, the ε radius is taken as 1 inch as per the given figure resolution.



Step 1: Now based on the distance closeness measure wrt reference point "+" as center as shown in the figure above, we can have our transactions defined using boolean spatial features as follows:

* = A	o = B	+ = C	x = D

Reference Item "+" => C

TID	Item-set
1	{A, B,C, D}
2	{A, B, C,D}
3	{A, B, C}
4	{A, B,C}

Now using the same Apriori principle, we can find frequent itemsets from the above transaction table.

Step 2: 1-item frequent-itemset F_1 . In this step, we will run a single pass over our dataset to determine the support of each item. Upon completion of this step, the set of all frequent 1-itemset, F_1 will be known.

Candidate 1- Itemsets

Item	Count
A	4
B	4
C	4
D	2

Since min_support = 3, item "D" will be pruned in this step. Rest items are declared as frequent. This means 1-item frequent-itemset F_1 **= [A, B,C]**

Step 3: Now for 2-item frequent-itemset F_2 will be generated using the combination of F_1 itemsets as shown below.

Candidate 2-Itemsets

Itemset	Count
{A,B}	4
{A,C}	4
{B,C}	4

Since min_support = 3, itemset {A,B}, {A,C} and B,C}{ are frequent itemset.
This means 2-item frequent-itemset F_2 **= [{A, B}, {A,C}, B,C}]**

Step 3: Now for 3-item frequent-itemset F_3 will be generated using the combination of F_2 itemsets as shown below.

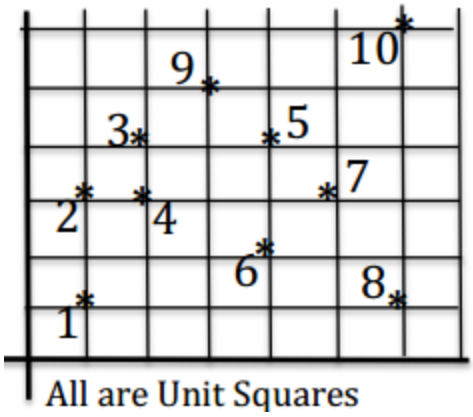
Candidate 3-Itemsets

Itemset	Count
{A,B,C}	4

Since min_support = 3, 3 itemsets {A,B,C} is the only 3 Candidate and frequent itemset.
This means 3-item frequent-itemset F_3 **= [{A, B, C}]**

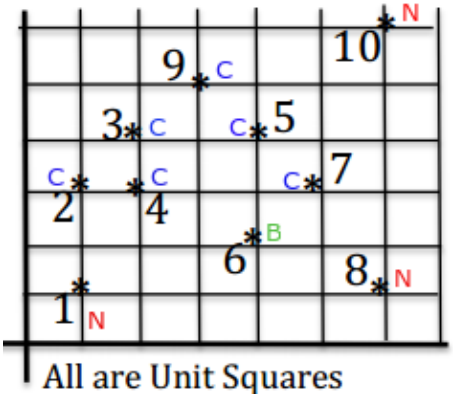
To summarize, according to the reference point "+", we have
1- item Frequent itemset as F_1 = [A, B,C] => [* , o, +]
2- item Frequent itemset as F_2 = [{A, B}] => [{*, o}, {*, +}, {o, +}]
3- item Frequent itemset as F_3 = [{A, B, C}] => [{ * , o, +}]

Q 7: DBSCAN and Constraints. Given the dataset below, answer following questions.

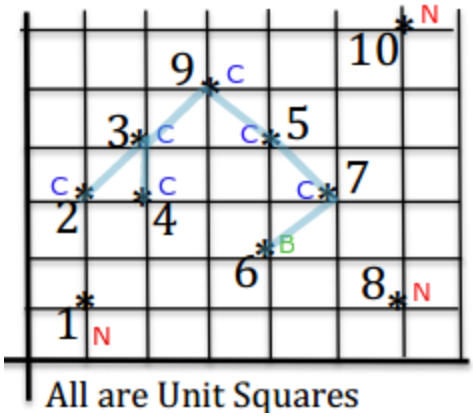


Assume all unit squares (that is, for each block, width = height), and all data points are at the intersection of grid lines. Use Euclidean distances.
(a) Mark core, border, and noise points (5 points). MinPoints = 3 (including the center point), and radius (EpsilonDist) = 1.5 units.

Core points : 2, 3, 4, 5, 7, 9
Border points : 6
Noise points : 1, 8, 10

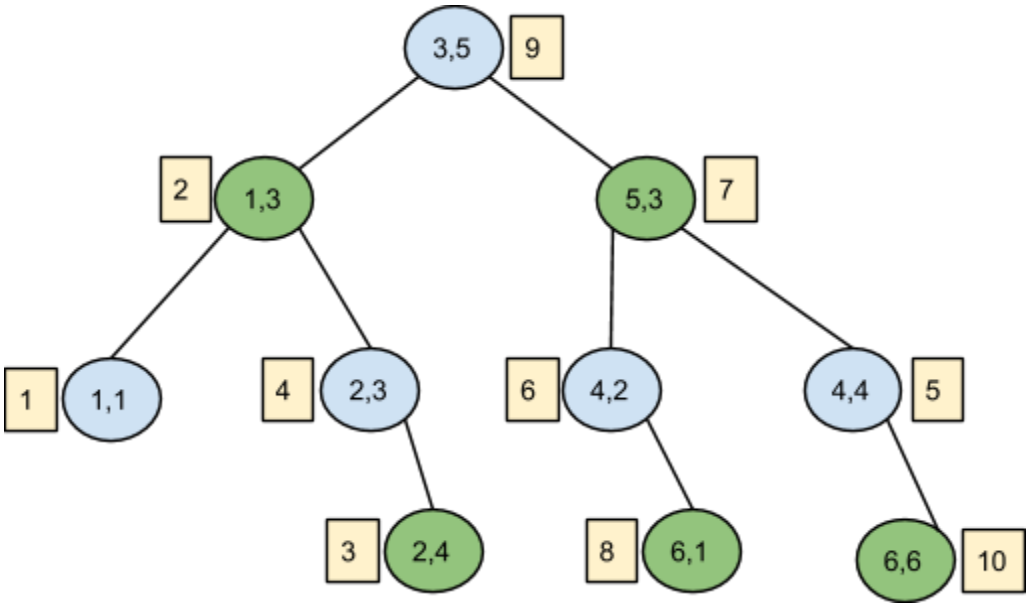


(b) Find the clusters (mark it on the figure below)
After removing noise points, and joining the core points within 1.5 units of EplisonDist, we get one cluster as shown in the figure below.



(c) Show kd-tree for this data (try to balance tree)

For the given dataset, we can construct kd-tree by dividing the nodes based on their x-y coordinates. Blue nodes signifies the x axis split(vertical cut) and green node signifies y axis split (horizontal cut). In vertical cut, values having x coordinate value equal or higher are kept on right side and lesser value nodes are kept on left side. In horizontal cut, values having y coordinate value equal or higher are kept on right side and lesser value nodes are kept on left side.The data point is written in the rectangular box beside the node.



The above tree is balanced since the min_depth = max_depth +/- 1

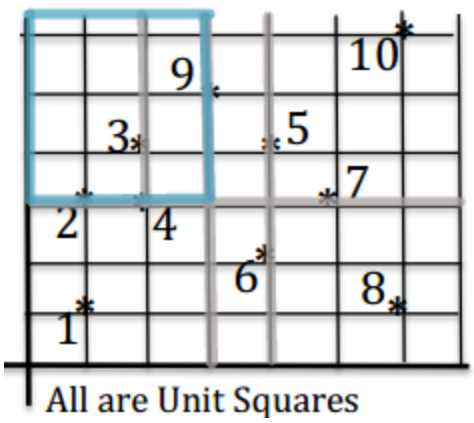
(d) Using the kd-tree, show how to find neighbor of point "4"

In order to find the neighbour point of "4", we will follow the steps as shown below:

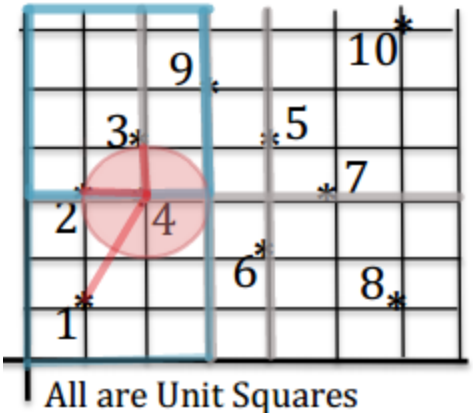
Step 1: Find the element 4, i.e. find the coordinate (2,3) and locate the local branch in lies in.

Here we observe all the points which comes in the sub-tree where the point "4" lies. And we compute the distance of each point in that subtree.

dist(3, 4) = 1
We have only one point 3 in that subtree, hence we get the radius as 1.



Step 2 : Now we draw the circle having 1 unit radii and “4” as center. Hence we discover the point 2 in the neighbourhood.



Step 3: In order to consider point 2, we expand the subtree to cover the larger portion of the neighbourhood. And similar way as step 2, we find the distance from all the points of the subtree to point “4”.

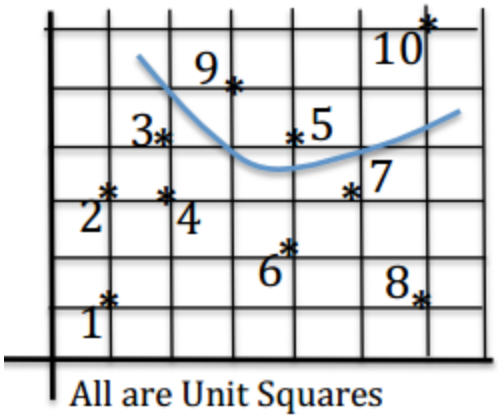
dist(2, 4) = 1
dist(1, 4) = 2.236

Among these distances, we have the smaller radii of unit 1, this means we have obtained the optimal neighbourhood radii and the points lying within the radii is the nearest neighbour.

Step 4: Here we have point “2” and “3” equi-distance from point “4”, hence both are the nearest neighbour to point “4”.

(e) Now assuming physical constraint, mark new clusters and explain your answer.

Given the constraints, we can apply constraint based DBscan method to identify the new clusters. Here we can use the cannot-link and must-link constraints of C-DBScan[7].



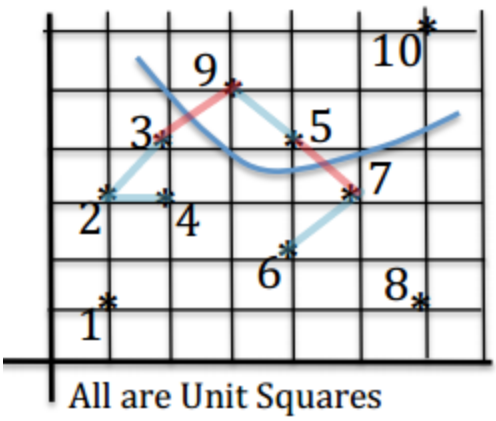
From the figure we can devise the following must and cannot link constraints.

Cannot Link Constraints:
point(9,3), point(9,3), point(9,7)
point(5,7), point(3,5), point(5,6) , point(5,3)
point(3,7), point(10,7), and likewise

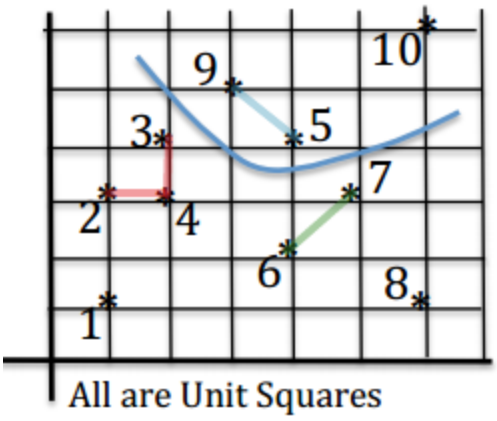
Since we do not have any bridge constraints for this data, there is no such must link constraints apart from regular DBScan linkages.

Hence we have follow the two step method:

1) first we use regular DBScan to find normal clusters. Here we have linked the cannot link constraints points that we have remove in the next step.



2) remove the linkage having physical constraints defined above. Here we can see that after applying the cannot-link constraints, we obtained 3 -different clusters.



References :

[1] Vatsavai, Ranga Raju, et al. "Spatiotemporal data mining in the era of big spatial data: algorithms and applications." Proceedings of the 1st ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data. ACM, 2012.

[2] Introduction to spatial data mining, Chapter 7 http://www.spatial.cs.umn.edu/Book/sdb-chap7.pdf

[3] Tung, A., Jean Hou, and Jiawei Han. "Spatial clustering in the presence of obstacles." Data Engineering, 2001. Proceedings. 17th International Conference on. IEEE, 2001.

[4] Wang, Wei, Jiong Yang, and Richard Muntz. "STING: A statistical information grid approach to spatial data mining." VLDB. Vol. 97. 1997.

[5]] Osmar R. Zaiane, et. al. Clustering Spatial Data in the Presence of Obstacles: a Density-Based Approach.

[6] Koperski, Krzysztof, and Jiawei Han. "Discovery of spatial association rules in geographic information databases." Advances in spatial databases. Springer Berlin Heidelberg, 1995.

[7] Carlos Ruiz., et. al. C-DBSCAN: Density-Based Clustering with Constraints.