CSC591/791: Spatial and Temporal Data Mining
Final Exam. Due: 05/03/15 @ 11.55pm EST.

Student Name: Abhishek Kumar Agrawal
Student ID: 200061445
Unity ID: akagrawa

Notes:

- Read each question carefully before answering it.
- Answer all questions to the extent possible, be brief and to the point.
- You can use appropriate resources (papers, books, discussions with friends), but answers should be yours.
- If you are in doubt, make better judgment; write your assumptions if any clearly, etc. Please note that only reasonable assumptions will be entertained (instructors decision is final).
- Hints: Treat it as in class exam; answer easy questions first, answer all questions.
- Submit pdf file and "R" code as separate text file.
- If you have to compare two figures, use consistent colors (symbols) that make comparison easy, and don't distort resolution.
- File names should be: your_last_name_student_id.zip (Submit single .zip file; which should include your solution file: your_last_name_student_id.pdf; and all your "R" scripts or any other ancillary information requested.)

#Q	Max Points	Your Score
1	25	
2	40	
3	20	
4	15	

Q1. General Concepts.

a. **List 4 unique characteristics of spatial data (5 points)**

Solution: Spatial data have been collected in various applications, ranging from remote sensing, to geographical information systems(GIS), computer cartography, environmental assessment and planning etc. In these application areas, spatial data contain certain distinguish and unique characteristics that has evolved the field of spatial data mining. Few of these unique characteristics are as follows:

1. **Spatial Constraint:** In spatial data mining tasks such as clustering of spatial data, there exist certain constraints in the data for which specific modeling techniques have been devised altogether separately. eg. Must link, cannot link constraints in spatial clustering.

2. **Spatial Diversity:** Spatial autocorrelation in spatial data has an implicit influence on the spatial distribution of a given attribute, and the diversity exhibited. Attributes values of a spatial entity are influenced by the neighbouring entities. When different entities are closer, spatial diversity increases, and when similar entities are closer, diversity decreases. Therefore, such spatial autocorrelation should be considered within a classification process.

3. **Spatial Co-location:** In spatial data, there exist certain features that are frequently co-located. Finding such interesting co-location pattern in spatial data has devised many spatial association rule mining algorithms.

4. **Spatial scan:** In spatial dataset, detecting spatial events have been always a challenging tasks. Those spatial events when cross certain threshold, qualifies for spatially interesting or outlier regions indicating difference in properties with respect to other spatial regions. Spatial scan statistics are one of the domain to find statistically significant interesting regions.

b. **Choose any 4 of your favorite algorithms (no more than one algorithm from each of the following categories: clustering, classification, association rules, anomalies), and show how you can model each of the unique characteristics you listed for above question (1.a). To answer this question, first list the algorithm precisely (in pseudo code), and then describe each step in the algorithm briefly. (20 points)**

Solution:

Clustering : **C-DBSCAN: Density-Based Clustering with Constraints[4]**

Pseudo Code :
Data: A set of instances, D.
A set of must-link constraints, ML, and a set of cannot-link constraints, CL.

```
Result: The set D partitioned into clusters that satisfy ML and CL.
begin
    Step 1: Partitioning the data space. kdtree := BuildKDTree(D).
    Step 2: Creating local clusters under Cannot-Link constraints.
    repeat
        for (all unlabeled points in a leaf X) do
            Select an arbitrary point $p_i$ from X.
            $X_{pi}$ ← all points in X that are within Eps radius of $p_i$
            if ( $X_{pi}$ contains less than MinPts points) then
                Label $p_i$ as NOISE.
            else if (exists a Cannot-Link constraint in CL among points in $X_{pi}$ ) then
                Create a local cluster for each point in X. Break.
            else
                Label $p_i$ as CORE. Label $X_{pi}$ as LOCAL CLUSTER.
    until (all leaves of the kdtree have been processed)

Step 3a: Merging local clusters under Must-Link constraints.
    for (each constraint m ∈ ML) do
        Join clusters involved in constraint m into cluster Y.
        Label Y as CORE LOCAL CLUSTER.
Step 3b: Merging clusters under Cannot-Link constraints.
    for (each core (local) cluster Y) do
        while (number of local clusters NLC decreases) do
            closestCluster ← closest local cluster to Y.
            if ( ∄Cannot-Link constraint in CL between points of Y and closestCluster)
            then
                Y ← Y ∪ closestCluster. Label Y as CORE CLUSTER.
                NLC-=1.
end
```

Description : C-DBScan is a clustering algorithm based on DBScan that captures spatial constraint such as cannot link and must link. The algorithm first applies a KD-Tree to divide the data space into iteratively into cubes by splitting planes that are perpendicular to the axes. Each cube becomes a node and is further partitioned, as long as it contains a minimum number of data points (the MinPts threshold of DBSCAN). In the next step, only the neighbourhoods within the same node are joined first. They are merged into the local clusters using density connected and density reachable principle of DBScan while enforcing Cannot-Link Constraint. If there is no Cannot-Link constraints, the conventional DBScan is invoked for each data point p in the leaf. After this step, the algorithm enforces the Must-Link Constraint, in which the data points involved in Must-Link relation if belongs to different local clusters are merged into a local core cluster. In the final step, once again the Cannot-Link constraints are verified so to make sure if any points having such constraints have not been merged together in the previous merge step.

Classification: Spatial Decision Tree[5]

In this algorithm, we will compute spatial entropy that capture the spatial diversity in the quantitative terms.

```
Pseudo Code : To compute Spatial Entropy
Data : Spatial entities C of a given Dataset D;
begin
        Step 1: Compute the intra-distance $d_i^{int}$ i.e. the average distance between the entities belonging to same category of the classification.
        For all $C_i \varepsilon C$, compute $d_i^{int}$

        $d_i^{int} = \frac{1}{|C_i| * (|C_i|-1)} \sum_{j \varepsilon C_i} \sum_{k \varepsilon C_i; k \neq j} dist(j,k)$ if $|C_i| > 1;\ and\ d_i^{int} = \lambda,\ otherwise$

        $C_i$ denotes the subset of C whose entities belong to the $ith$ category of the classification.

        Step 2 : Compute the extra-distance $d_i^{ext}$ i.e. the average distance between the entities of a given category and the distance of all the other categories.
        For all $C_i \varepsilon C$, compute $d_i^{ext}$
        $d_i^{ext} = \frac{1}{|C_i| * |C - C_i|} \sum_{j \varepsilon C_i} \sum_{k \varepsilon (C - C_i)} dist(j,k)$ if $C_i \neq C;\ and\ d_i^{ext} = \beta,\ otherwise$

        here $dist(j,k)$ gives the distance between the entities j and k;

        Step 3: In the final step, we use the distance calculated above, in the computation of spatial entropy.
                $Entropy_s(A) = -\sum_{i=1}^{n} \frac{d_i^{int}}{d_i^{ext}} P_i \, log_2 P_i$
        Here $P_i$ denotes the proportion of each category $i$
end
```

Description: Spatial entropy is an impurity measure that helps finding the better splits in spatial decision tree unlike conventional entropy where info gain split does not model any type of spatial diversity present in the spatial dataset. In the first step of this process, we calculate the intra distance d_i^{int} for every spatial entity C_i belonging to the same category of the classification, which is the average distance between the entities of C_i. Similarly in the next step we calculate d_i^{ext} i.e. the average distance between the entities of a given category and the distance of all the other categories. Here λ is a constant taken relatively small, and β a constant taken relatively high; these constants avoid the "noise" effect of null values in the calculation of the average distances. Now once these distance has been calculated for each pair of spatial entity, we then calculate spatial entropy which is just an extension of conventional entropy. We integrate these average distances in such a form that exhibits an increase of spatial entropy when the intra-distance d_i^{int} increases and extra-distance d_i^{ext} decreases, and vice versa. In this way, the spatial entropy surpasses the conventional entropy in the evaluation of the diversity a given spatial system exhibits.

Association Mining: Mining co-location patterns[6]

Pseudo Code:
Input:

(a) E = {Event-ID, Event-Type, Location in Space} representing a set of events;
ET = {Set of boolean spatial event types};
(b) Neighborhood relationship function; pair of spatial points;
(c) Interest measure function (e.g. prevalence, conditional probability);
(d) Threshold on prevalence measure and conditional probability;

Output:

A set of co-locations with values of interest measures (i.e. prevalence, conditional probability) satisfying threshold.

Data Structure:

k = Co-location size
C_k = set of candidate size k co-locations in iteration k = 1, 2, ..., P
T_k = set of table instances of co-locations in C_k for k = 1, 2, ..., P
P_k = set of prevalent size k co-locations for k = 1, 2, ..., P
R_k = set of co-location rules of size k for k = 1, 2, ..., P
T_ C_k = set of coarse-level table instances of size k co-location in C_k for k= 1, 2, ..., P

Steps:

```
Co-location-size k = 1
$C_1$ = ET;
$P_1$ = ET;
$T_1$ = generate_table_instance( $C_1$ , E);

Initialize data structure $C_k$ , $T_k$, $P_k$, $R_k$, T_ $C_k$ to be empty for k >1
while(not empty $P_k$ ) do{
        $C_{k+1}$ = generate_candidate_colocation ( $C_k, k$ );
        if (fmul = true) then {
                $C_{k+1}$ = multi_resolution_pruning ( $C_{k+1}$ );
        }
        $T_{k+1}$ = generate_table_instance ( q , $C_{k+1}$ , $T_k$ );
        $P_{k+1}$ = select_prevalent_colocation(q , $C_{k+1}$ , $T_{k+1}$ );
        $R_{k+1}$ = generate_colocation_rule ( $P_{k+1}$ , $T_{k+1}$ );
}
return union ( $R_2$ , ..., $R_{k+1}$ );
```

Description: In this algorithm, an event centric model is defined to model the co-location patterns in the continuous spatial data. The algorithm takes a set ET of spatial event types, a set E of event instances, user-defined functions representing spatial neighborhood relationships as well as interest measures (e.g prevalence, conditional probability) and thresholds for prevalence based pruning. The algorithm outputs a set of prevalent co-location rules with the values of the interest measures. The initialization step assigns starting values to various data-structures used in the algorithm. Then the algorithm iteratively perform four basic tasks, namely generation of candidate co-locations, generation of table instances of candidate co-locations, pruning, and generation of co-location rules. These tasks are carried out inside a loop iterating over the size of the co-locations.

Spatial Scan Statistics Analysis: There is a generalized spatial scan static algorithm that can be specified for any spatial scan statistics measures, such as Kuldroff's original spatial scan[7].

Pseudo Code:
Data: A set of spatial locations s_i

Step 1: Choose a set of spatial regions to search over, where each spatial region S consists of a set of spatial locations s_i .
Step 2: Choose a model of the data under H_0 (the null hypothesis of no clusters) and $H_1(S)$ the alternative hypothesis assuming a cluster in region S.
Step 3: Derive a score function F(S) based on $H_1(S)$ and H_0 .
Step 4: Find the "most interesting region‘ i.e. those regions S with the highest values of F(S).
Step 5: Determine whether each of these regions is "interesting" either by performing significance testing or calculating posterior probabilities.

<u>For Kuldroff's Original Spatial Scan</u>
Description: After choosing a set of spatial regions to search over, the model for null hypothesis and alternate hypotheses are defined. H_0: no events in the spatial regions, and $H_1(S)$ the alternative hypothesis assuming a cluster in region S. Now score function in spatial scan is defined as likelihood ratio of data points belonging to both null and alternate hypothesis region.

$$F(S) = \frac{Pr(Data|H_1(S))}{Pr(Data|H_0)}$$

In Kuldroff's original model, the event distribution is spatial region assume to follow poisson distribution.

$$F(S) = \left(\frac{C}{B}\right)^{C} \left(\frac{C_{tot}-C}{B_{tot}-B}\right)^{C_{tot}-C} \left(\frac{C_{tot}}{B_{tot}}\right)^{-C_{tot}}$$

Here c_i count for location s_i which follow Possion(q b_i)
b_i = baseline for location i.e. expected or estimated count
q = risk (expected ration of count to baseline)

Now in the last step, we find most interesting region where F(S) is highest

$$S^* = argmax_s\ F(S)$$

Now for the significant test, we can check for

$$\begin{aligned} H_0 \quad &: q = q_{all} \text{ everywhere} \\ H_1(S) &: q = q_{in} \text{ inside S,} \\ & \quad q = q_{out} \text{ outside S,} \\ & \quad\quad q_{in} > q_{out} \end{aligned}$$

Here the regions where $q_{in} > q_{out}$ are significant enough to reject null hypothesis.

Q2. Bayesian Classification: For the given training (ilk-tr-d.csv) and test (ilk-te-d.csv) datasets, answer the following questions:

a. Assuming features are correlated, and generated by Gaussian distribution, and equal 'a priori' probability for each class, do maximum likelihood classification (MLC), and report test accuracy (full error/contingency matrix and overall accuracy). (10 points)

Assuming samples are generated using gaussian distribution,
we perform MLC classification and compute,

$$P(\omega_i \mid x) = \frac{P(x|\omega_i) * P(\omega_i)}{P(x)}$$

Here ω_i is the prior probability, and assuming an equal a priori probability of each class and $P(x)$ as constant for all samples, we can just compute

$$P(\omega_i \mid x) = P(x| \omega_i)$$

Note: Please find the R script with filename **question2.R** and helper script **gaussian.R .** Here in the predict function classification= "mlc" parameter performs maximum likelihood classification.

Test Accuracy : 87.71429%
Contingency matrix :

	Reference				
Prediction	**1**	**2**	**3**	**4**	**5**
1	176	4	0	0	0
2	21	146	0	0	0
3	0	0	170	1	0
4	3	0	5	99	52
5	0	0	0	0	23

Overall Accuracy(Using Test and Training Data): 89.67273%

b. Assuming features are independent, do Naïve Bayes (NB) classification, and report test accuracy (full error/contingency matrix and overall accuracy). (10 points)

Assuming features are independent, performing Naive Bayes classification

Note: Please find the R script with filename question2.R

Test Accuracy : 75.428%

Contingency matrix :

		Reference				
Prediction		**1**	**2**	**3**	**4**	**5**
	1	143	0	25	0	0
	2	13	150	0	0	0
	3	31	0	139	8	0
	4	13	0	11	65	44
	5	0	0	0	27	31

Overall Accuracy(Using Test and Training Data): 77.6%

c. **Now assuming that 'a priori' probability is not same for each class, repeat question 2.a. (10 points).**

Now assuming that 'a priori' probability is not same for each class, we can bias the probability with class proportion weight i.e. class having more samples in training data will have more weights.

we perform MAP(Maximum a priori) classification and compute,

$$P(\omega_i \mid x) = \frac{P(x|\omega_i) * P(\omega_i)}{P(x)}$$

Here ω_i is the prior probability, and assuming a priori probability is **not** same of each class and $P(x)$ as constant for all samples, we can just compute

$$P(\omega_i \mid x) = P(x| \omega_i) * P(\omega_i)$$

Note: Please find the R script with filename question2.R and helper script gaussian.R Here in the predict function classification= "mla" parameter performs maximum a-priori classification.

Test Accuracy : 87.42857%

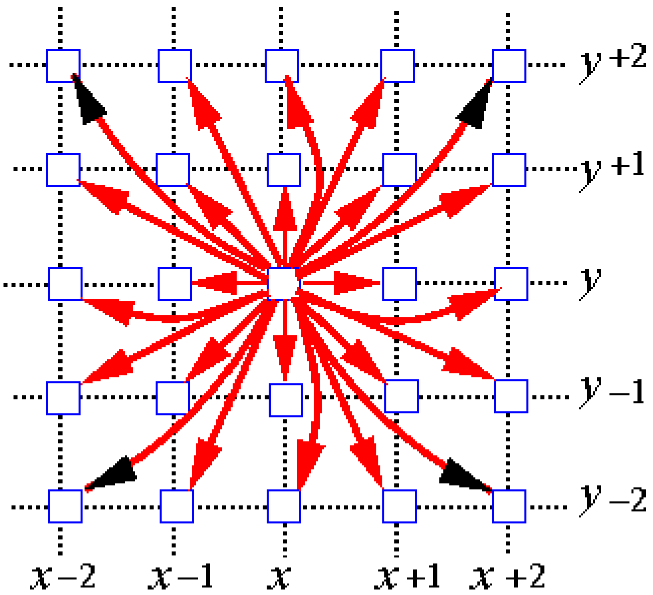
Contingency matrix :

		Reference				
Prediction		**1**	**2**	**3**	**4**	**5**
	1	177	4	0	0	0
	2	21	146	0	0	0
	3	0	0	174	2	0
	4	2	0	1	98	58
	5	0	0	0	0	17

Overall Accuracy(Using Test and Training Data): 90.4%

d. Assuming samples are spatially autocorrelated, describe a Bayesian classifier that models neighborhood (aka contextual) information in classification. (10 points).

Solution: In the questions above, so far we were building non-contextual model with respect to the neighborhood of the samples. But to capture the more accurate spatial context there exist many such algorithms that considers neighborhood properties into account to build context sensitive models.



Markov Random Field Classifier[3] is one such classifier in this domain that utilizes the Markov property to model the neighbourhood context in the classification process. The Markov property specifies that the variable depends only on the neighbours and is independent of all the variables as shown in the figure above. A middle site class label probability depends on its neighbourhood properties. A set of random variables whose interdependency relationship is represented by an undirected graph is called a Markov Random Field. Certain spatial classification problem can be modeled in this framework assuming that the class label, $f_L(s_i)$, of different locations s_i constitute an MRF.

The **Bayesian** rule can be used to predict label $f_L(s_i)$ from feature value vector X and neighborhood class label vector L_M as follows:

$$Pr\,(l(s_i)|X,\ L \setminus l(s_i)) = \frac{Pr\,(X\,(s_i)\,|l\,(s_i)\,,L \setminus l(s_i))\;Pr(l(s_i)|L \setminus l(s_i))}{Pr(X(s_i))}$$

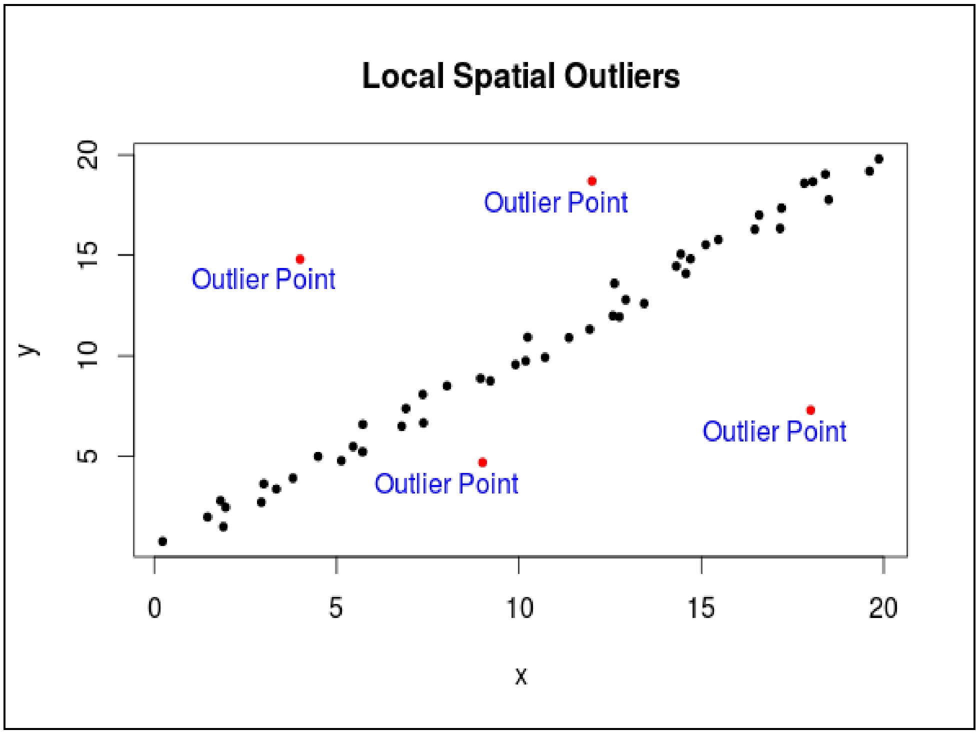
The solution procedure can estimate $Pr(l(s_i)|\,L \setminus l(s_i))$ from the training data by examining the ratios of the frequencies of class labels to the total number of locations in the spatial framework.

Q3. Spatial Autocorrelation, Anomalies/Outliers:

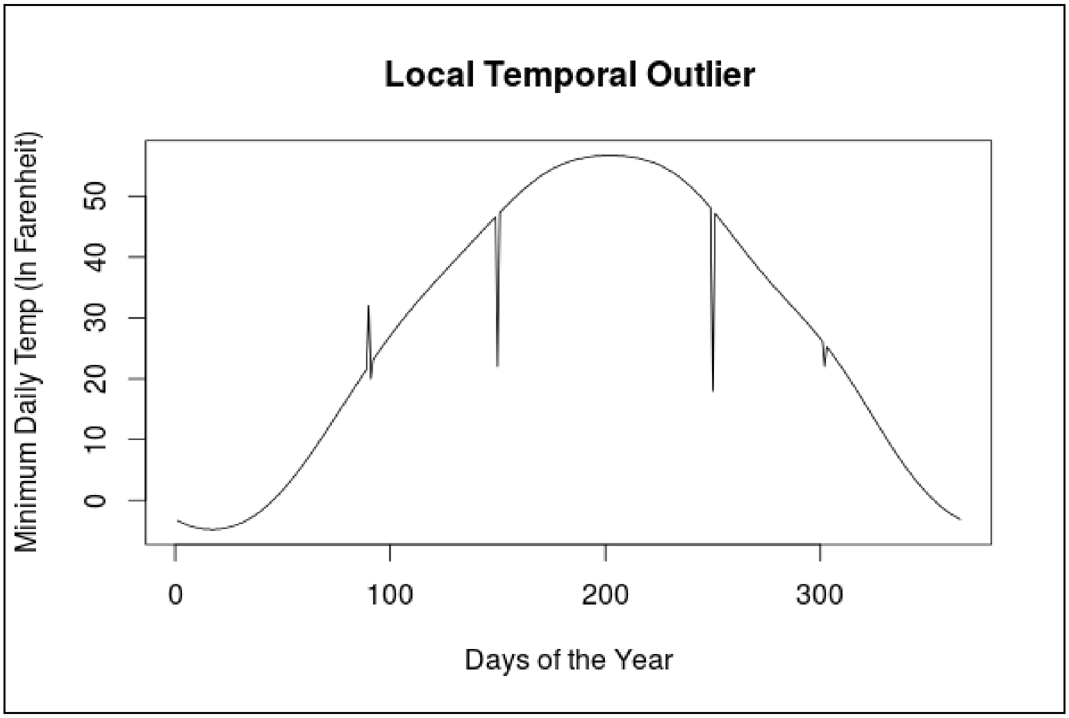
a. **Using separate plots, show why global methods like +/- 2 standard deviations do not capture local spatial and temporal outliers (2 points)**

Solution: Global methods like +/- 2 standard deviations check for outliers wrt. to the overall attribute values. In this case, certain local outlier values with respect to their neighbours are ignored resulting in non-detection of such local outliers.

Local Spatial Outlier : In the plot below, we can see that points in x,y plane are in the range of 1 to 20, and if we apply +/- 2 standard deviation method to detect the outliers, we won't find any local outliers because outlier points (points in red) are within the range of normal xy points distribution. Here we can clearly observe that some of the points(red points) are clearly out of range with respect to their neighbours. Eg. the black points can be river body and the red points have been misclassified as part of river when it is far apart from the river boundary locations.



Local Temporal Outlier : In the timeseries plot below, we can see the minimum temperature distribution of an year(365 days). We can see the temperature in the range of [-15, 55] Fahrenheit for a yearly collected data. Here we can see sudden spikes in the temperature which in general have values within +/- 2 standard deviation but with respect to its immediate neighbors they are local outliers.



b. **Analyze the given dataset to validate (accept or reject) the following null hypothesis. H0:Zero spatial autocorrelation in the variable “oz.” (Show your work). (10 points)**

Given: Ozone DataSet
H0 : Zero Autocorrelation in the variable “oz”

Here with the given dataset, we need to perform the statistical significance test on the variable “oz” in order to accept or reject the null hypothesis. We will calculate **Moran’s I** which is a measure of spatial autocorrelation i.e. how spatially related the values of a variable are based on the locations where they were measured. And based on the results we can comment on the proposed hypothesis.

To calculate Moran’s I, we will use R package **“ape”**. To calculate Moran’s I, we will need to generate a matrix of inverse distance weights. The inverse of distance measure will give more weights to the points that are closer than to the points that are far apart.

Below is the R code for computing Moran’s I. Also find the R script question3b.R to reproduce the results as shown below.

```
## Calculating Moran’s I
library(ape)

## Reading data from csv file
ozoneData <- read.csv("spatial-oz.csv")

## Computing Distance Matrix
distance <- as.matrix(dist(cbind(ozoneData$Lon, ozoneData$Lat)))

## Computing inverse of distance matrix and setting diagonal elements
to ## zero
distance_inv <- 1/distance
diag(distance_inv) <- 0

## Computing Moran’s I
Moran.I(ozoneData$oz, distance_inv)
```

Output

```
$observed
[1] 0.2265500981

$expected
[1] -0.03225806452

$sd
[1] 0.03431137693

$p.value
[1] 0.000000000000004596323322
```

The observed/computed value is significantly higher than the expected value, which indicates it has the higher positive correlation.
Here we can see that the **p value is very low**, and based on the results we can **reject** the null hypothesis that there is zero autocorrelation present in the variable **oz** at alpha = 0.05.

$p \ll (\alpha$ level) $[\alpha = 0.05]$ i.e. 95% confidence interval

Here lower p-value indicates that there is very low probability that the pattern observed is occurred by chance or at random, rather it is significant enough to be not ignored. Hence we reject the null hypothesis of zero auto-correlation for variable oz.

c. **Compare and contrast Ansilin’s local Moran’s I with Getis-Ord G statistic. (3 points).**

Ansilin’s Local Moran’s I	Getis-Ord G Statistic
1. Ansilin’s Local Moran’s I is the **local statistic** measure of spatial autocorrelation in which we identify the variation across the study area, focusing on individual features and their relationships to nearby features.	1. Getis-Ord G Statistic is the **global statistic** measure of spatial autocorrelation in which we identify and measure the pattern of the entire study area. Here specific pattern locations are not indicated separately.
2. Given a set of weighted features, it identifies where high or low values cluster spatially, & features with values that are very different from surrounding feature values.	2. It measures how concentrated the high or low values are for a given study area for a given set of features.
3. In this measure, given a set of weighted features with respect to the study area, we calculate the s a Local Moran's I value, a Z score, a p-value for each feature. The Z score & p-value represent the statistical significance of the computed index value I.	3. In this measure we calculate the High/Low General G value (observed & expected) ,and the associated p-value and z-score for a given input feature. The results of the analysis are interpreted within the context of a null hypothesis.
4. The Z scores & p-values are measures of statistical significance which tell you whether or not to reject the null hypothesis, feature by feature. They, in effect, indicate whether the apparent similarity (or dissimilarity) in values for a feature & its neighbors is greater than one would expect in a random distribution.	4. The null hypothesis for the General G statistic states "there is no spatial clustering of the values". When the absolute value of the Z score is large & the p-value is very small, the null hypothesis can be rejected.
5. 5. A positive value for I indicates that the feature is surrounded by features with similar values. Such a feature is part of a cluster. A negative value for I indicates that the feature is surrounded by features with	5. If the null hypothesis is rejected, then the sign of the Z score becomes important. If the Z score value is positive, it means that high values cluster together in the study area. If the Z Score value is negative, it

dissimilar values. Such a feature is an outlier.	means that low values cluster together

d. Write a procedure (step-by-step algorithm) to find spatial outliers using either “Moran’s I” or Getis-Ord G statistic. (5 points)

Solution: Finding spatial outlier using local statistical measure of spatial autocorrelation “Ansilin’s local Moran’s I”. In this measure the computed statistical index I, is used to detect the spatial outlier. Here for each study area i,

$$I_i = z_i \sum_j w_{ij} z_j$$

Step-by-Step Algorithm

1. In the first step, we will calculate the weight matrix, which is actually a spatial weight matrix where the weights indicates the spatial proximity. Higher value of w_{ij} indicates that the study area i and j are spatially close. To build spatial weight matrix:
    a. Contiguity or distance matrix is constructed between every pair of study regions. Here we can either take the actual average euclidean distance if the study regions are point values or define the neighbourhood binary relations i.e. assign value 1 if neighbour(sharing boundaries) or else 0 (not sharing boundaries).
    b. Now we compute the weight corresponding to the distance metric computed in the previous step. This weight can be taken as inverse of the distance if actual distance is measured or standardized the rows with respect to their total neighbors count if binary relations of distance is computed. Here each row in the matrix corresponds to a separate study region i.

2. In this step, we will compute z_i ,the z-score of the feature value for each study region i. Here z_i is nothing but the feature value x_i in the standardized form.

$$z_i = \frac{x_i - \bar{x}}{SD_x}$$

3. In this step, we perform matrix multiplication of the weight matrix and z score and compute the rowsums.

$$\sum_j w_{ij} z_j$$

4. In the final step, we multiply the rowsum vector with the z-score of the corresponding study region i to get Ansilin’s local Moran’s I value for each region i.

$$I_i = z_i \sum_j w_{ij} z_j$$

Interpretations:

- If the I value is positive, the features on which we performed our analysis are surrounded by features with similar values, either high or low. This indicates that the feature is a part of a cluster. Here statistically significant clusters can consists of high values(HH) or low values (LL).
- If the I value is negative, the features on which we performed our analysis are surrounded by features with dissimilar values. This indicates that the feature is an outlier and thus the study area emerges out as a **spatial outlier** in context with its neighbourhood study areas. Here statistically significant outliers can be a feature with a high value surrounded by features with low values (HL) or a feature with a low value surrounded by features with high values (LH).

  Here the significance test is performed to either accept or reject the null hypothesis i.e. if the spatial study region is an outlier significantly or just by chance.

Q4: Overall: List top 5 open spatial data mining challenges based on your understanding of the course materials (including papers from reading list). (15 points)

Solution:

Spatial data mining tasks possess many challenges as compare to general data mining processes in terms of data extraction and pre-processing, data modeling and finding significant patterns. General purpose data mining methods are not suitable for spatiotemporal data due to the **complexity** present in its data types, the **spatial relationships** and **spatial autocorrelation** properties it holds[3]. Some of the open challenges for spatial data mining based on my understanding of the course materials are as below :

(1) The spatial relationships among the variables: General purpose data mining techniques assumes explicit relationships i.e. arithmetic relation, ordering, instance-of, subclass-of etc. among its data variables whereas geographical data have implicit with spatial relationships among its variables such as overlap, intersect, direction etc. Such implicit relationships are hard to capture, as it requires ample domain expertise as well as context sensitive informations. For example, modeling constraints and obstacles while performing spatial clustering is a challenge where the additional task from normal clustering is to maintain the spatial context of togetherness as well as separateness.

(2) Mixed distributions: Most of the data mining techniques such as Naive Bayesian assumes data to be normally distributed or K-means clustering methods which performs well on data having one type of distribution. Whereas spatiotemporal data constitutes of various mixture models and distributions which makes general-purpose data mining methods have poor performance. For example, many image processing techniques assumes to have mixture of various gaussian distributions for modeling. Such multivariate gaussian mixture models itself possess many challenges in terms of computation, precision and level of approximation. In this direction there exist many data distribution mixture models that can provide a base for data modeling and hence projects an open challenge for researchers to explore and compare them with existing methods in the pursuit of less expensive and more accurate algorithms.

(3) Non iid : In conventional data mining techniques, observations are assumed to be independent and identically distributed. But spatiotemporal data are having features that are highly dependent among themselves locally. There exist spatial autocorrelation among the features that requires separate set of techniques and statistical measures to handle while modeling. For example, in classification process we need to consider neighborhood relations as the nearby objects influence the decision process. Many state of the art methods have been modified to capture such **spatial autocorrelation;** decision trees considers spatial entropy when modeling on spatial data, markov chain models takes the influence of the immediate neighbours in the probability distribution of the data points. Similarly there still exist many such methods in which if spatial context are considered while modeling, can prove to be an effective technique in spatial data mining.

(4) Data extraction pre-processing: Data extraction and pre-processing is always a more challenging and tedious task in any data mining problem. Here in spatial data mining, spatial data are stored in various formats such as VHR images, GIS databases, cartographic data, census data etc. Given such diverse raw format of data, applying any data mining process is a challenging process. Almost all data mining algorithms works on feature space of the data objects, here these objects can be anything ranging from point, polygon, line to spatial location or census block areas depending on the context of the problem. Extracting features from data objects requires another set of tools and techniques which adds even more complexity.

For example, segmentation of images, transactionization of spatial objects or events for co-location mining etc. are such open data extraction and pre-processing challenges that draws attention of many researchers in this domain to find more effective, less complex, and computationally inexpensive techniques. Improving these techniques has a direct impact on spatial data mining algorithms performance and accuracy.

(5) Parameterized Algorithms: Many algorithms in various spatial data mining tasks are parameterized. Given the vast and continuous nature of spatial data, finding appropriate parameters in this context is again a challenging task There are many algorithms whose computational cost and effectiveness depends on the selection of these input parameters. Sometimes these parameters are provided by domain experts and rest are determined using some machine learning techniques.

For example, many algorithm performs spatial scanning of the data, for which they divide space into MBR(Minimum bounding rectangles) or grids whose appropriate size and shapes drastically reduce the model complexity. There exists many heuristics and approximation based algorithms that help in finding the appropriate parameters for such problems, but still many methods are computationally expensive such as SatScan where finding optimization and high performance computing techniques are open challenges to the research community.

References :

[1] Spatial Autocorelation Statistics http://www.cego.lsu.edu/documents/reviews/geospatial/spatial_autocorrelation.pdf

[2] "Welcome to the Institute for Digital Research and Education." How Can I Calculate Moran's I in R Web. 02 May 2015.

[3] Berthod, Marc, et al. "Bayesian image classification using Markov random fields." Image and Vision Computing 14.4 (1996): 285-295.

[4] Carlos Ruiz., et. al. CDBSCAN: DensityBased Clustering with Constraints.

[5] Xiang Li and Christophe Claramunt. A Spatial Entropy-Based Decision Tree for Classification of Geographical Information. Transactions in GIS Volume 10, Issue 3, pages 451–467, May 2006

[6] Huang, Yan, Shashi Shekhar, and Hui Xiong. "Discovering colocation patterns from spatial data sets: a general approach." Knowledge and Data Engineering, IEEE Transactions on 16.12 (2004): 1472-1485.

[7] Vatsavai, Ranga R. "Spatial and Spatiotemporal Data Mining: Recent Advances." Next Generation of Data Mining. By Shashi Shekhar. 545-74.