Chapter 7

Introduction to Spatial Data Mining

In this chapter we present an overview of some important concepts related to the relatively new and rapidly developing field of *data mining.* Our focus, of course, is on the *mining* of spatial data, but the set of techniques that we will discuss applies to many different types of datasets, including temporal, multimedia, and text databases.

Data mining is the process of discovering interesting and potentially useful *patterns* of information embedded in large databases. The mining metaphor is meant to convey an impression that *patterns* are *nuggets* of precious information hidden within large databases waiting to be discovered. Data mining has been quickly embraced by the commercial world as a way of harnessing information from the large amounts of data that corporations have collected and meticulously stored over the years.

If data mining is about extracting patterns from large databases, then the largest databases have a strong spatial component. For example, the Earth Observation Satellites, which are systematically *mapping* the entire surface of the earth, collect about one terabyte of data every day. Other large spatial databases include the U.S. census, and the weather and climate databases. The requirements of mining spatial databases are different from those of mining classical relational databases. In particular, the notion of *spatial autocorrelation* that similar objects tend to cluster in geographic space, is central to spatial data mining.

The complete data-mining process is a combination of many subprocesses which are worthy of study in their own right. Some important 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. For spatial data, the issue of scale the level of aggregation at which the data are being analyzed, is also very important. It is well known in spatial analysis that identical experiments at different levels of scale can sometimes lead to contradictory results. Our focus in this chapter is limited to the design of data-mining algorithms. In particular we describe how classical data-mining algorithms can be extended to model the spatial autocorrelation property. Here it is important to understand the distinction between spatial data mining and spatial data analysis. As the name implies, spatial data analysis covers a broad spectrum of techniques that deals with both the spatial and non spatial characteristics of the spatial objects. On the other hand spatial data mining techniques are often derived from spatial statistics, spatial analysis, machine learning and data bases, and are customized to analyze massive data sets. This chapter provides an introduction to the up coming field of spatial data mining often building on the well know techniques in spatial analysis and spatial statistics. More regorous treatment of spatial analysis and spatial statistics can be found in [Bailey and Gatrell, 1995], [Fotheringham and Rogerson, 1994], [Goodchild, 1986], [Fischer and Getis, 1997], [Cressie, 1993].

In Section 7.1 we introduce the data-mining *process* and enumerate some well-known techniques that are associated with data mining. In Section 7.2 we introduce the important concept of spatial autocorrelation and show how it can be calculated and integrated into *classical* data-mining techniques. In Section 7.3 we discuss classification techniques and introduce PLUMS model. Section 7.4 deals with association rule discovery techniques and Section 7.5 deals with various clustering techniques. In Sections 7.6 and 7.7, we discuss advanced techniques like Markov Random Fields and spatial outlier detection.

7.1 Pattern Discovery

Data mining is the process of discovering potentially interesting and useful *patterns* of information embedded in large databases. A pattern can be a summary statistic, like the mean, median, or standard deviation of a dataset, or a simple rule like “Beach property is, on average, 40 percent more expensive than inland property”.

A well-publicized pattern, which has now become part of data mining lore, was discovered in the transaction database of a national retailer: “People who buy diapers in the afternoon also tend to buy beer”. This was an unexpected and interesting finding which the company put to profitable use by rearranging the store. Thus data mining encompasses a set of techniques to generate hypotheses, followed by their validation and verification via standard statistical tools. For example, if the store has a modest 100 items, then finding which two items are correlated, or “go together,” will require 4,950 correlation tests. The promise of data mining is the ability to rapidly and automatically search for *local* and potentially *high-utility* patterns using computer algorithms.

7.1.1 The Data-Mining Process

The entire data-mining process in shown in Figure 7.1. In a typical scenario a domain expert (DE) consults a data-mining analyst (DMA) to solve a specific problem. For example, a manager in a city law enforcement department may want to explain the unusually high crime rate that the city is witnessing that year. The DE has access to a database which may provide clues to the specific problem that she wants the DMA to solve. An iterative process leads the DE and the DMA to agree upon a *problem statement* whose solution may provide a satisfactory answer to the original problem.

Now the DMA must decide which technique or combination of techniques is required to address the problem. For example, the DMA may decide that the problem is best addressed in the framework of *classification,* in which case the goal may be to build a model that predicts the crime rate as a function of other socioeconomic variables. Once an appropriate

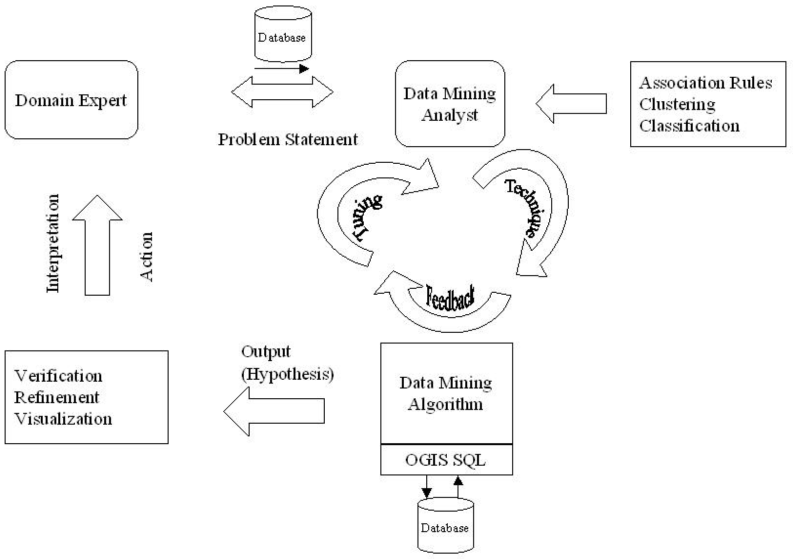


Figure 7.1: Data-mining Process. The data-mining process involves a close interaction between a domain expert and a data-mining analyst. The output of the process is a set of hypothesis (patterns), which can then be rigorously verified by statistical tools and visualized using a GIS. Finally the analyst can interpret the patterns and make and recommend appropriate action.

technique is selected, a suitable data-mining algorithm is chosen to implement the technique. For classification, the DMA may decide to use *linear regression* instead of decision trees because the class attribute is continuous-valued.

In an ideal case the data-mining algorithm should be designed to directly access the database using SQL (OGIS SQL for spatial databases) but typically a time-consuming exercise is involved in transforming the database into an algorithm-compatible format. The selection of a technique and the choice of an appropriate algorithm is also a nondeterministic, iterative process. For example, most algorithms require the adjustment of user-defined parameters, and in most cases there is no way to judge beforehand what are the right parameters to set for a specific database.

The output of a data-mining algorithm is typically a *hypothesis* which can be in the form of model parameters (as in regression), rules, or labels. Thus the output is a potential *pattern*. The next step is verification, refinement, and visualization of the pattern. For spatial data this part of the process is typically done with the help of GIS software. The final part of the data-mining process is the interpretation of the pattern, and where possible, a recommendation for appropriate action. For example, the conclusion might be that the high crime rate is directly attributable to a downturn in the city's economic condition, in which case the law enforcement manager can direct the result to appropriate authorities in the city government. Or the data mining results might indicate that the high crime rate is a result of exceptionally high crime activity in a few neighborhoods ("hot spots"). In this case the law enforcement agencies can *saturate* those neighborhoods with police patrols.

7.1.2 Statistics and Data Mining

The entire data-mining process described above looks suspiciously like statistics! So where is the difference? One way to view data mining is as a *filter* step before the application of rigorous statistical tools. The role of the filter step is to literally plow through reams of data and generate some potentially interesting hypothesis which can then be verified using statistics. This is similar to the use of R-trees to retrieve MBRs (minimum bounding rectangles) to answer range queries. The R-tree and MBRs provided a fast filter to search the space for potential candidates which satisfy a range query. The difference is that while R-trees guarantee that there will be no *false dismissals,* such a concept does not exist in data mining, at least not yet. A detailed discussion of difference between data mining and statistics is given in [Hand, 1999].

7.1.3 Data Mining as a Search Problem

Data mining is the search for *interesting* and *useful* patterns in large databases. A data-mining algorithm searches a potentially large space of patterns to come up with candidate patterns which can be characterized as interesting or useful or both. For example, consider a 4×4 image where we want to classify each pixel into one of two classes, *black* or *white* in Figure 7.2. Then there are a total of 2^{16} potential combinations. Now if we assert that each 2×2 block can only be assigned to one class, black or white, then the number of combinations reduces to 2^4. This restriction, though severe, is not completely unjustified. As it happens, most neighboring pixels of an image tend to belong to the same class, especially in a high-resolution image.

7.1.4 Unique Features of Spatial Data Mining

The difference between classical and spatial data mining parallels the difference between classical and spatial statistics. One of the fundamental assumptions that guide statistical analysis is that the data samples are independently generated, as with successive tosses of a coin, or the rolling of a die. When it comes to the analysis of spatial data, the assumption about the independence of samples is generally false. In fact spatial data tends to be highly self-correlated. For example, people with similar characteristics, occupations, and backgrounds, tend to cluster together in the same neighborhoods. The economies of a region tend to be similar. Changes in natural resources, wildlife, and temperature vary gradually over space. In fact this property of like things to cluster in space is so fundamental that geographers have elevated it to the status of the first law of geography: *Everything is related to everything else, but nearby things are more related than distant things* [Tobler, 1979]. In

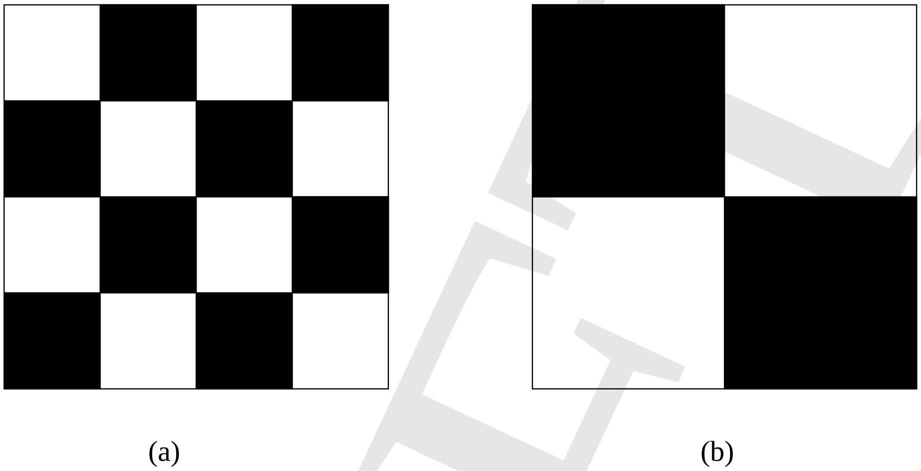


Figure 7.2: Search results of data-mining algorithm. (a) One potential pattern out of a total of 2^{16}. (b) If we constrain the patterns to be such that each 4×4 block can only be assigned one class, then the potential number of patterns is reduced to 2^4. Based on other information, a data-mining algorithm can quickly discover the "optimal" pattern.

spatial statistics, an area within statistics devoted to the analysis of spatial data, this is called `spatial autocorrelation`.

7.1.5 Famous Historical Examples of Spatial Data Exploration

Spatial data mining is a process of automating the search for potentially useful patterns. Some well-known examples of what we now call spatial data mining occurred well before the the invention of computers. [Griffith, 1999] provides some examples:

1. In 1855 when the Asiatic cholera was sweeping through London, an epidemiologist marked all locations on a map where the disease had struck and discovered that the locations formed a cluster whose centroid turned out to be a water pump. When the government authorities turned off the water pump, the cholera began to subside. Later scientists confirmed the water-borne nature of the disease.

2. The theory of Gondwanaland that the all the continents formed one land mass was postulated after R. Lenz discovered (using maps) that all the continents could be fitted together into one-piece (like one giant jigsaw puzzle). Later fossil studies provided additional evidence supporting the hypothesis.

3. In 1909 a group of dentists discovered that the residents of Colorado Springs had unusually healthy teeth, and they attributed it to high level of natural fluoride in the local drinking water supply. Researchers later confirmed the positive role of fluoride in controlling tooth decay. Now all municipalities in the United States ensure that all drinking water is fortified with fluoride.

The goal of spatial data mining is to automate the discoveries of such correlations, which can be then be examined by specialists for further validation and verification.

Attribute	**Type**	**Role**	**Description**
Vegetation Durability(VD)	Ordinal	Independent	Ordinate scale from 10 to 100
Stem Density (SD)	Numeric	Independent	In number of stems/m^2
Stem Height (SH)	Numeric	Independent	In centimeters above water
Distance to Open Water(DOP)	Numeric	Independent	In meters
Distance to Edge (DTE)	Numeric	Independent	In meters
Water Depth (WD)	Numeric	Independent	In centimeters
Red-winged Blackbird	Binary	Dependent	Record the presence/abscence of the nest in the cell

Table 7.1: Habitat variables used for predicting the locations of the nests of the red-winged blackbirds. There are six independent variables and one dependent variable. The type of the dependent variable is binary.

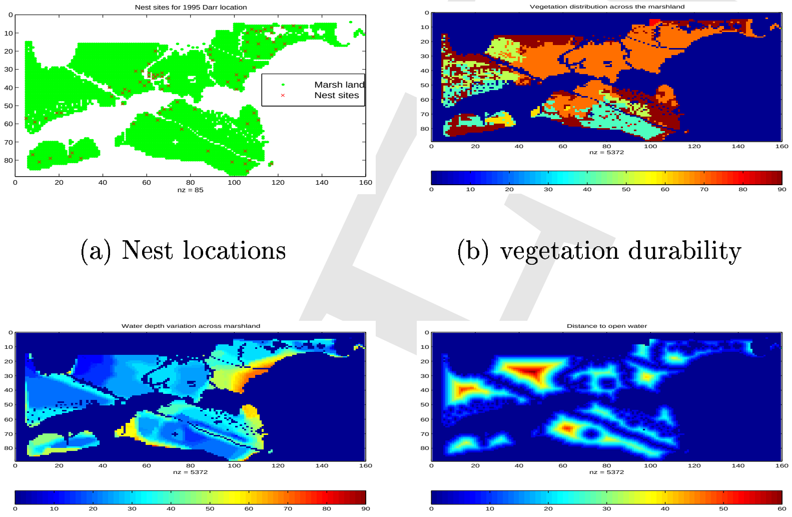
7.2 Motivating Spatial Data Mining

7.2.1 An Illustrative Application Domain

We now introduce an example which will be used throughout this chapter to illustrate the different concepts in spatial data mining. We are given data about two wetlands on the shores of Lake Erie in Ohio, USA, in order to *predict* the spatial distribution of a marsh-breeding bird, the red-winged blackbird (*Agelaius phoeniceus*). The names of the wetlands are Darr and Stubble, and the data was collected from April to June in two successive years, 1995 and 1996.

A uniform grid was imposed on the two wetlands, and different types of measurements were recorded at each cell or pixel. The size of each pixel was five meters. The values of seven attributes were recorded at each cell, and they are shown in Table 7.1. Of course domain knowledge is crucial in deciding which attributes are important and which are not. For example, Vegetation Durability was chosen over Vegetation Species because specialized knowledge about the nesting habits of the red-winged blackbird suggested that the choice of nest location is more dependent on the plant structure and its resistance to wind and wave action than on the plant species.

Our aim is to build a model for predicting the location of bird nests in the wetlands. Typically the model is built using a portion of the data, called the *Learning* or *Training* data and then tested on the remainder of the data, called the *Testing* data. For example, later on we show how to build a model using the 1995 data on the Darr wetland and then test it on either the 1996 Darr or 1995 Stubble wetland data. In the learning data all the attributes are used to build the model, and in the training data one value is *hidden*, (in our case the location of the nests). Using knowledge gained from the 1995 Darr data and the value of the independent attributes in the test data, we want to predict the location of the nests in Darr 1996 or in Stubble 1995.


Nest sites for 1995 Darr location
Marsh land
Nest sites
nz = 85
Vegetation distribution across the marshland
nz = 5372
Water depth variation across marshland
nz = 5372
Distance to open water
nz = 5372

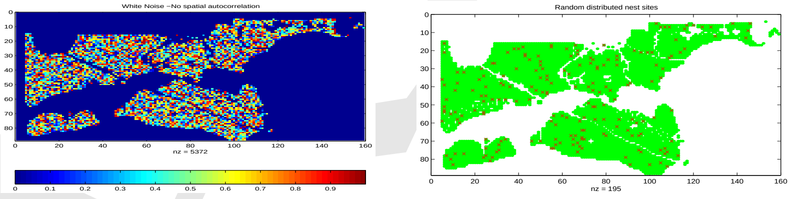
(a) Nest locations

(b) vegetation durability

(c) Water depth

(d) Distance to open water

Figure 7.3: Darr wetland, 1995. (a) Learning dataset: The geometry of the marshland and the locations of the nests; (b) spatial distribution of *vegetation durability* over the marshland; (c) spatial distribution of *water depth*; and (d) spatial distribution of *distance to open water*.


White Noise −No spatial autocorrelation
nz = 5372
Random distributed nest sites
nz = 195

(a) Pixel property with independent identical distribution

(b) Random nest locations

Figure 7.4: Spatial distribution satisfying random distribution assumptions of classical regression

In this chapter we focus on three independent attributes, namely, *vegetation durability*, *distance to open water* and *water depth*. The significance of these three variables was established using classical statistical analysis. The spatial distribution of these variables and the actual nest locations for the Darr wetland in 1995 are shown in Figure 7.3. These maps illustrate two important properties which are inherent in spatial data.

1. The values of attributes which are referenced by spatial location tend to vary gradually over space. While this may seem obvious, classical data-mining techniques, either

explicitly or implicitly, assume that the data is *independently* generated. For example, the maps in Figure 7.4 show the spatial distribution of attributes if they were independently generated. This property of "smoothness" across space is called *spatial autocorrelation.*

2. The spatial distribution of attributes sometimes shows distinct local trends which contradict the global trends. This is most vivid in Figure 7.3b, where the spatial distribution of *vegetation durability* is jagged in the western section of the wetland compared to the overall impression of uniformity across the wetland. Thus spatial data is not only not *independent*, it is also not *identically* distributed.

We now show how to quantify the notion of spatial autocorrelation and spatial heterogeneity.

7.2.2 Measures of Spatial Form

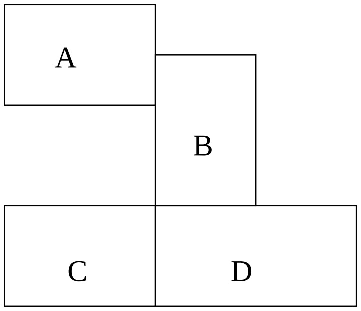
As discussed in previous chapters, space can be viewed as *continuous* or *discrete.* Spatial continuity is common in most earth science data sets. Often it is difficult to represent data in continuous form, as an infinite number of samples exist in continuous space. On the other hand only finite number of samples are enumerated in discrete space. In continuous space places are identified by coordinates, and in discrete space places are identified as objects. Spatial statistics are used for exploring geographic information. The term *geostatistics* is normally associated with continuous space and the term *spatial statistics* is associated with discrete space.

Centrality, dispersion and shape are used to characterize spatial form.

Mean center is the average location, computed as the mean of X and mean of Y coordinates. The mean center is also known as the center of gravity of a spatial distribution. Often the *weighted mean center* is appropriate measure for several spatial applications, for e.g., center of population. The *weighted mean center* is computed as the ratio between the sum of the coordinates of each point multiplied by its weight (e.g., number of people in block) and the sum of the weights. The measure *center* is used in several forms. It can be used to simplify complex objects (e.g., to avoid storage requirements and complexity of digitation of boundaries, a geographic object can be represented by its center), or for identifying the most effective location for a planned activity (e.g. a distribution center should be located a central point so that travel to it is minimized).

Dispersion is a measure of the spread of a distribution around its center. Often used measures of dispersion and variability are *range, standard deviation, variance and coefficient of variance.* Dispersion measures for geographical distributions are often calculated as the summation over the ratio of the weight of geographic objects and the proximity between them. *Shape* is multi-dimensional, and there is no single measure to capture all of the dimensions of the shape. Many of shape measures are based on comparison of the shape's perimeter with that of a circle of the same area.

Measures of *spatial dependence*: It is a very common observation in many geospatial applications that the events at a location are influenced by the events at neighboring locations.



(a) Map

$$\begin{array}{c|cccc|} & A & B & C & D \\ \hline A & 0 & 1 & 0 & 0 \\ B & 1 & 0 & 1 & 1 \\ C & 0 & 1 & 0 & 1 \\ D & 0 & 1 & 1 & 0 \end{array}$$

(b) Boolean *W*

$$\begin{array}{c|cccc|} & A & B & C & D \\ \hline A & 0 & 1 & 0 & 0 \\ B & 0.3 & 0 & 0.3 & 0.3 \\ C & 0 & 0.5 & 0 & 0.5 \\ D & 0 & 0.5 & 0.5 & 0 \end{array}$$

(c) Row-normalized *W*

Figure 7.5: (a) A spatial lattice: (b) its contiguity matrix; (c) its row-normalized contiguity matrix.

Spatial dependence can be defined as 'the propensity of a variable to exhibit similar (of different) values as a function of the distance between the spatial locations at which it is measured.' Spatial autocorrelation is used to measure spatial dependence.

Spatial autocorrelation is a property that is often exhibited by variables which are sampled over space. For example, the temperature values of two locations near to each other will be similiar. Similarly, soil fertility varies gradually over space and so do rainfall and pressure. In statistics there are measures to quantify this interdependence. One such measure is called Moran's *I*.

Moran's *I*: A Global Measure of Spatial Autocorrelation

Given a variable $x = \{x1, \ldots, x_n\}$ which is sampled over n locations, Moran's I coefficient is defined as

$$I = \frac{zWz^t}{zz^t}$$

where $z = \{x_1 - \bar{x}, \ldots, x_n - \bar{x}\}$, $\bar{x}$ is the mean of x, W is the $n \times n$ row-normalized contiguity matrix, and z^t is the transpose of z. For example, a spatial lattice, its contiguity matrix, and its row-normalized contiguity matrix are shown in Figure 7.5.

The key point to note here is that Moran's I coefficient depends not only on the different values of the variable x but also on their arrangement. For example Moran's I coefficients of the two 3×3 images shown in Figure 7.6 are different even though the sets of values of the pixels are identical. Moran's I coefficients for the four-neighbor relation and the eight-neighbor relation are shown in Table 7.2.

Local Indicators of Spatial Autocorrelation

With the wide availability of high-resolution image data and the increasing use of Global Positioning System (GPS) devices to mark the locations of samples in field work, the fact

3	g	h	i
2	d	e	f
1	a	b	c
	1	2	3

(a)

3	5	8	10
2	7	16	11
1	13	14	6
	1	2	3

(b)

3	5	7	13
2	8	16	14
1	10	11	6
	1	2	3

(c)

Figure 7.6: The Moran's I coefficient. The pixel value sets of the two images are identical, but they have different Moran's I coefficients.

Explanatory Variable	**Four-Neighbor**	**Eight-Neighbor**
Distance to edge	0.7606	0.9032
Distance to open water	0.7342	0.8022
Depth	0.6476	0.7408
Vegetation height	0.7742	0.8149
Stem density	0.6267	0.7653
Vegetation durability	0.3322	0.4851

Table 7.2: Moran's I coefficient of explanatory variables to predict nest locations for the red-winged blackbird

that spatial autocorrelation exists is often moot. As a consequence, spatial statisticians often use local measures of spatial autocorrelation to track how spatial dependence varies in different areas within the same spatial layer. A substantial variation in local autocorrelation at different locations indicates the presence of spatial heterogeneity, as is evident in the *vegetation durability* layer in Figure 7.3b. The *local Moran's I* measure defined at location i is

$$I_i = \frac{z_i}{s^2} \sum_j \frac{W_{ij}}{z_j},\ i \neq j$$

where $z_i = x_i - \bar{x}$ and s is the standard deviation of x. For example, in Table 7.3 Moran's I coefficient is

$$I_{75} = \frac{(75 - 55.82)}{675.32}(71 + 85 + 61 + 63 - 4(55.82)) = 1.6109$$

7.2.3 Spatial Statistical Models

Statistical models are often used to represent the observations in terms of random variables. These models then can be used for estimation, description, and prediction based on proba-

40	41	39	44	52	64	74	67	63	63	57	47	48	59	62	50
48	50	51	45	78	82	92	109	115	98	83	74	70	76	95	70
40	40	41	46	86	92	79	97	123	107	115	110	101	83	78	56
40	39	38	47	74	103	82	89	94	91	115	121	113	104	88	56
45	44	46	51	82	98	74	72	59	71	83	83	83	103	106	64
50	43	44	44	67	88	74	59	45	85	107	88	70	97	115	75
48	40	41	41	71	85	98	82	51	86	118	91	66	86	100	78
48	45	40	47	98	95	89	71	52	81	110	71	45	46	54	53
52	48	56	61	103	91	85	$\boxed{75}$	63	72	94	57	37	35	36	39
48	48	79	78	40	45	51	61	64	58	58	48	53	47	40	43
37	45	47	41	26	25	28	29	31	33	35	37	56	57	46	47
27	28	29	28	27	26	27	29	28	28	29	32	44	45	40	47
29	26	24	27	29	28	27	27	27	28	28	34	41	38	38	47
28	27	25	27	27	27	26	27	27	28	27	38	53	47	36	48
25	27	26	25	28	34	31	27	27	28	28	34	45	48	38	48
25	26	27	28	34	39	32	29	27	29	28	31	37	41	41	47

Table 7.3: A 16 × 16 gray-scale image.

bility theory.

Point process: A point process is a model for the spatial distribution of the points in a point pattern. Several natural processes can be modeled as spatial point patterns. The positions of trees in a forest, locations of gas stations in a city, are all examples of point patterns. A spatial point process is defined as $Z(t) = 1; \forall t \in T$ or $Z(A) = N(A), A \subset T$, where both $Z(.)$ and T are random. Here T is the index set ($T \subset \Re^d$), and $Z(.)$ is the spatial process. Spatial point processes can be broadly grouped into random or non-random processes. Real point patterns are are often compared with a random pattern (generated by a Poisson process) using the average distance between a point and its nearest neighbor. For a random pattern, this (average) distance is expected to be $\frac{1}{2*\sqrt{density}}$, where density is the average number of points per unit area. If for a real process, this computed distance falls within a certain limit, then we conclude that the pattern is generated by a random process, otherwise it is non-random process. Patterns generated by a non-random process can be either clustered (aggregated patterns) or uniformly spaced (regular patterns).

Lattices: A lattice D is denoted by $Z(s) : s \in D$, where the index set D is a countable set spatial sites at which data are observed. Here the 'lattice' referes to a countable collection of regular or irregular spatial sites. Several spatial analysis functions (e.g., spatial dependence, spatial autoregression, Markov Random Fields) can be applied on lattice models.

Geostatistics: Geostatistics deals with analysis of spatial continuity which is an inherent characteristic of spatial data sets. Geostatics provides a set of statistical tools for modeling spatial variability and interpolation (prediction) of attributes at unsampled locations. Spatial variability can be analyzed using *variograms*. The amount and form of spatial autocorrelation can be described by a variogram, which summarizes the relationship between differences in pairs of measurements and the distance between corresponding pair of points. Spatial interpolation (prediction) techniques are used to estimate the values at unsampled locations using the known values at sampled locations. Kriging is a well known estimation procedure used in geostatistics. Kriging uses known values (at sampled locations) and a semivariogram (estimated from the data) to determine unknown values. Kriging offers better estimates over conventional interpolation methods (like weighted average, nearest neighbor) for spatial data sets, because it takes into account the spatial autocorrelation.

7.2.4 The Data-Mining Trinity

Data mining is a truly multidisciplinary area, and there are many novel ways of extracting patterns from data. Still, if one were to *label* data-mining techniques, then the three most noncontroversial labels would be *classification, clustering, and association rules*. Before we describe each of these classes in detail, we present some representative examples where these techniques can be applied.

Location Prediction and Thematic Classification

The goal of *classification* is to estimate the value of an attribute of a relation based on the value of the relation's other attributes. Many problems can be expressed as classification problems. For example, determining the locations of nests in a wetland based upon the value of other attributes (*vegetation durability, water depth*) is a classification problem sometimes also called the *location prediction* problem. Similarly, predicting where to *expect* hot spots in crime activity can be cast as a location prediction problem. Retailers essentially solve a location prediction problem when they decide upon a location for a new store. The well-known expression in real-estate, "Location is everything," is a popular manifestation of this problem.

In thematic classification, the goal is to categorize the pixels of satellite images based upon the values of the "spectral signatures" recorded by receivers on board the satellite. The problem of thematic classification has deep spatial connections because in most instances pixels which are neighbors on the image belong to the same class. Thus satellite images naturally exhibit high spatial autocorrelation.

Determining the Interaction Between Attributes

Rapid pattern detection within a large volume of data that is being continuously generated and stored in databases is one of the motivations behind data mining. One of the simplest and probably most well-known data-mining techniques is the discovery of relationships within attributes of a relation. For example, in the context of supermarket data analysis, a pattern of the form $X \rightarrow Y$ means that people who buy the product X also have a high likelihood of buying product Y. In the context of spatial databases, we have rules of the form of *is_close(house, beach)* → *is_expensive(house)*; that is, houses which are close to the beach are likely to be expensive. In the context of the bird habitat examples of the rules obtained were *low vegtation durability* → *high stem density.* In Section 7.4.1, we discuss *Apriori*, arguably the most well-known algorithm for discovering association rules.

Identification of Hot Spots: Clusters and Outliers

As noted in Section 7.1.1, law enforcement agencies use hot spot analysis to determine areas within their jurisdiction which have unusually high levels of crime. They do this by recording the location of each crime and then using outlier detection and clustering techniques to determine areas of high crime density. Outlier detection and clustering can also be used to determine hot spots of nest location, disease clusters for cancer.

The goal of outlier detection is to discover a "small" subset of data points which are often viewed as noise, error, deviations or exceptions. Outlier have been informally defined as observations which appear to be inconsistent with the remainder of the data set. The identification of outliers can lead to the discovery of unexpected knowledge and has a number of practical applications in areas such as credit card fraud, the performance analysis of athletes, voting irregularities, bankruptcy, weather prediction and "hot spot" detection.

Another practical example of using spatial point clustering is to determine the location of service stations. For example, suppose a car company has information about the geographic location of all its customers and would like to open new service centers to cater exclusively to their customers. Clustering methods can be employed to determine the "optimal" location of service centers.

Clustering is an example of unsupervised learning, as no knowledge of the labels or the numbers of labels is known a priori. As a result, clustering algorithms have to work "harder" to determine the likely clusters. We will discuss two methods of clustering later. The K-medoid is a deterministic clustering algorithm where each record is placed exclusively in one cluster. Probabilistic clustering on the other hand species the probability of each record belonging to any cluster.

In their simplest form, hot spots are regions in the study space which stand out compared to the overall behavior prevalent in the space. Thus, hot spots can be identified by merely inspecting the distribution of the data on the map or by thresholding. For example all regions where the attribute value (e.g., crime rate) is at least two standard deviations away from the mean can be labeled as hot spots. From a spatial autocorrelation perspective, hot spots are locations where high local spatial autocorrelation exists. Before we describe each of the three data-mining techniques in detail, we reiterate that scale (the level of aggregation) is very important at all levels of data-mining analysis.

In the following three sections we cover three major approaches in data mining, namely, classification, association rules and clustering.

7.3 Classification Techniques

Simply stated, the classification is to find a function

$$f : D \to L.$$

Here D, the domain of f, is the space of attribute data and, L is the set of labels. For example, in our illustrative bird-habitat domain, D is the three-dimensional space consisting of *vegetation durability*, *water-depth*, and *distance to open water*. The set L consists of two labels: *nest* and *no-nest*. The goal of the classification problem is to determine the appropriate f, from a given finite subset $Train \subset D \times L$. The success of classification is determined by the accuracy of f when applied to a data set $Test$ which is disjoint from the $Train$ data. The classification problem is known as *predictive modeling* because f is used to predict the labels L when only data from the set D is given.

There are many techniques available to solve the classification problem. For example, in maximum-likelihood classification the goal is to completely specify the joint-probability distribution $P(D, L)$. This is usually accomplished by an application of the Bayes theorem and is the method of choice in remote-sensing classification. In the business community, decision-tree classifiers are the method of choice because they are simple to use. The decision-tree classifiers divide the attribute space (D in our case) into regions and assign a label to each region. Neural networks generalize the decision-tree classifiers by computing regions which have non-linear boundaries. Another common method is to use regression analysis to model the interaction between D and L using an equation. For example, the linear equation $y = mx + c$ is used for modeling (x, y) in linear regression analysis.

In this chapter our focus is on extending classical data-mining techniques to incorporate spatial autocorrelation, which is the key distinguishing property of spatial data. Using linear regression as a proptype, we will show how classification methods can be extended to model spatial autocorrelation. We have chosen linear regression analysis to expound spatial classification because this method is most widely known, and spatial regression is probably the most well-studied method for spatial classification in the spatial statistics community.

7.3.1 Linear Regression

When the class variable is real-valued, it is more appropriate to calculate the conditional expectation rather than the conditional probability.Then the goal of classification is to compute

$$E[C|A_1, \ldots, A_n]$$

Writing in a more familiar notation, with C replaced by Y and the A'_is by X'_is, and assuming that all the attributes are *identically and independently* generated standard normal random variables, the *linear* regression equation is

$$E[Y|\mathbf{X} = \mathbf{x}] = \alpha + \beta\mathbf{x}.$$

where $\mathbf{X} = (X_1, \ldots, X_n)$. This expression is equivalent to the more familiar expression

$\mathbf{Y} = \mathbf{X}\beta + \epsilon$. Once again, the training data can be used to calculate the parameter vector β, which in turn can be used to calculate the value of the *class* attribute in the test data set.

7.3.2 Spatial Regression

As we have shown before, when variables are spatially referenced, they tend to exhibit spatial autocorrelation. Thus the above assumption of identical independent distribution (i.i.d) of random variables is not appropriate in the context of spatial data. Spatial statisticians have proposed many methods to extend regression techniques that account for spatial autocorrelation. The simplest and most intuitive is to modify the regression equation with the help of the contiguity matrix W. Thus the spatial autoregressive regression (SAR) equation is

$$\mathbf{Y} = \rho W \mathbf{Y} + \mathbf{X}\beta + \epsilon$$

The solution procedure for the SAR equation is decidedly more complex than the classical regression equation because of the presence of the $\rho W Y$ term on the right side of the equation. Also notice that the W matrix is quadratic in terms of the data samples. Fortunately very few entries of W are nonzero, and sparse matrix techniques are used, which exploit this fact, to speed up the solution process.

7.3.3 Model Evaluation

We have discussed two general models to solve the classification problem, namely, linear regression and spatial autoregressive regression (SAR). We now show the standard ways to evaluate the performance of models and explain why the standard ways of evaluation are not adequate in the context of spatial data mining.

In the case of a two-class classification problem, like `nest` or `no-nest`, there are four possible outcomes that can occur. For example, a `nest` can be correctly predicted, in which case it is called a `true-positive` (TP). A model can predict a *nest* where actually there was a `no-nest`, in which case it is a false-positive (FP). Similarly a `no-nest` can be correctly classified, a `true-negative` (TN), and a `no-nest` can be predicted where there was actually a nest, which is a `false-negative` (FN). All the four combinations are shown in Figure 7.7.

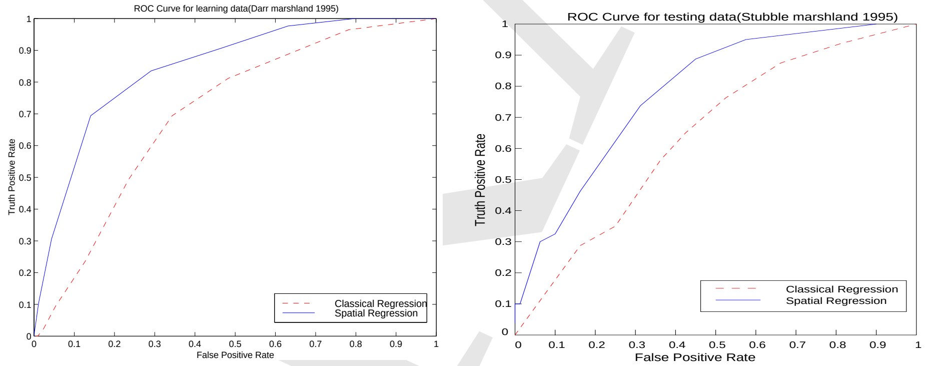
In classification the goal it to predict the conditional probability of one attribute on the basis of the values of the other attributes. Thus the outcome of a classification techniques are probabilities. The way the probabilities are converted to actual class labels is to choose a cut-off probability b, and label all records whose predicted probability is greater than b by one class label, say *nest* and label the remaining records as *no-nest*. By varying b we can get a good estimate of how two different classifiers are behaving vis-a-vis each other. Thus for a given cut-off b the True-Positive Rate (TPR(b)) and the False-Positive Rate (FPR(b)) is defined as

$$TPR(b) = \frac{TP(b)}{TP(b)+FN(b)}$$
$$FPR(b) = \frac{FP(b)}{FP(b)+TN(b)}$$

		ACTUAL CLASS	
		nest	*no-nest*
PREDICTED CLASS	*nest*	*TRUE POSITIVE (TP)*	*FALSE POSITIVE (FP)*
	no-nest	*FALSE NEGATIVE (FN)*	*TRUE NEGATIVE (TN)*

Figure 7.7: The four possible outcomes for a two-class prediction

Now if we plot TPR vs. FPR for the two classifiers under consideration, then the classifier whose curve is further above the diagonal $TPR = FPR$ is the better model for that specific data set. These curves are called receiver operating characteristics (ROC) curves. We have compared the classical regression and spatial autoregressive regression (SAR) model on the Darr 1995 training set and the Stubble 1995 test set. The results in Figure 7.8 clearly show that including the spatial autocorrelation term ρWY leads to substantial improvement in the learning and predictive power of the regression model.



(a) Training Data

(b) Test Data

Figure 7.8: ROC curves.(a) Comparison of the ROC curves for classical and spatial autoregression regression (SAR) models on the 1995 Darr wetland data. (b) Comparison of the two models on the 1995 Stubble wetland data.

The model evaluation technique described above is not particularly suited for the context of spatial data. Consider the example shown in Figure 7.9. Here the goal is to predict the locations marked A using regression analysis. The ROC curves will fail to distinguish between the model which predicts the locations shown in Figure 7.9 c and another model which predicts locations shown in Figure 7.9d, even though the predictions in Figure 7.9d are closer to the actual locations than those predicted by Figure 7.9c . We have used this

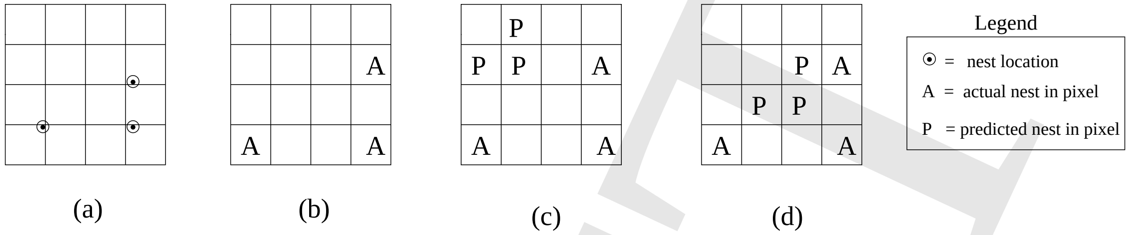


Figure 7.9: Problems of ROC curves with spatial data. (a)The actual locations of nest's; (b)pixels with actual nests; (c)location predicted by a model; (d)location predicted by another mode. Prediction (d) is spatially more accurate than (c). Classical measures of classification accuracy will not capture this distinction.

observation to design a new framework to solve the two-class spatial classification problem in the context of the location prediction problem, which we describe next.

7.3.4 Predicting Location Using Map Similarity (PLUMS)

The location prediction problem is a generalization of the nest-location prediction problem. It captures the essential properties of similar problems from other domains, including crime prevention and environmental management. The problem is formally defined as follows:

Given:

- A spatial framework S consisting of sites $\{s_1, \ldots, s_n\}$ for an underlying geographic space G.
- A collection of explanatory functions $f_{X_k} : S \rightarrow R^k, k = 1, \ldots K$, where R^k is the range of possible values for the explanatory functions.
- A dependent class variable $f_C : S \rightarrow C = c_1, \ldots c_M$
- An value for parameter α, relative importance of spatial accuracy.

Find: Classification model: $\hat{f}^C : R^1 \times \ldots R^k \rightarrow C$.

Objective: Maximize similarity $(map_{s_i \in S}(\hat{f}_C(f_{X_1}, \ldots, f_{X_k})), map(f_C))$
$= (1-\alpha)$ classification_accuracy$(\hat{f}_C, f_C) + (\alpha)$spatial_accuracy$((\hat{f}_C, f_C)$

Constraints:

1. Geographic space S is a multidimensional Euclidean space. [1]
2. The values of the explanatory functions, $f_{X_1}, \ldots, f_{X_k}$, and the dependent class variable, f_C, may not be independent with respect to the corresponding values of nearby spatial sites (i.e., spatial autocorrelation exists).
3. The domain R^k of the explanatory functions is the one-dimensional domain of real numbers.
4. The domain of dependent variable, $C = 0, 1$.

[1] The entire surface of the earth cannot be modeled as a Euclidean space, but locally the approximation holds true.

The above formulation highlights two important aspects of location prediction. It explicitly indicates that (i) the data samples may exhibit spatial autocorrelation and, (ii) an objective function (i.e., a map similarity measure), is a combination of classification accuracy and spatial accuracy. The *similarity* between the dependent variable f_C and the predicted variable $\hat{f}_C$ is a combination of the "traditional classification" accuracy and representation-dependent "spatial classification" accuracy. The regularization term α controls the degree of importance of **spatial accuracy** and is typically domain dependent. As $\alpha \to 0$, the map similarity measure approaches the traditional classification accuracy measure. Intuitively, α captures the spatial autocorrelation present in spatial data.

7.3.5 Markov Random Fields

Markov random field based Bayesian classifiers estimate classification model $\hat{f}_C$ using MRF and Bayes' rule. A set of random variables whose interdependency relationship is represented by an undirected graph (i.e., a symmetric neighborhood matrix) is called a Markov Random Field. The Markov property specifies that a variable depends only on its neighbors and is independent of all other variables. The location prediction problem can be modeled in this framework by assuming that the class label, $l_i = f_C(s_i)$, of different locations, s_i, constitute an MRF. In other words, random variable l_i is independent of l_j if $W(s_i, s_j) = 0$.

The Bayesian rule can be used to predict l_i from feature value vector X and neighborhood class label vector L_i as follows:

$$Pr(l_i|X, L_i) = \frac{Pr(X|l_i, L_i)Pr(l_i|L_i)}{Pr(X)} \tag{7.1}$$

The solution procedure can estimate $Pr(l_i|L_i)$ from the training data, where L_i denotes a set of labels in the neighborhood of s_i excluding the label at s_i, by examining the ratios of the frequencies of class labels to the total number of locations in the spatial framework. $Pr(X|l_i, L_i)$ can be estimated using kernel functions from the observed values in the training dataset. For reliable estimates, even larger training datasets are needed relative to those needed for the Bayesian classifiers without spatial context, since we are estimating a more complex distribution. An assumption on $Pr(X|l_i, L_i)$ may be useful if the training dataset available is not large enough. A common assumption is the uniformity of influence from all neighbors of a location. For computational efficiency it can be assumed that only local explanatory data $X(s_i)$ and neighborhood label L_i are relevant in predicting class label $l_i = f_C(s_i)$. It is common to assume that all interaction between neighbors is captured via the interaction in the class label variable. Many domains also use specific parametric probability distribution forms, leading to simpler solution procedures. In addition, it is frequently easier to work with a Gibbs distribution specialized by the locally defined MRF through the Hammersley-Clifford theorem.

Both SAR and MRF Bayesian classifiers model spatial context and have been used by different communities for classification problems related to spatial datasets. Now we compare these two approaches to modeling spatial context, using a probabilistic framework.

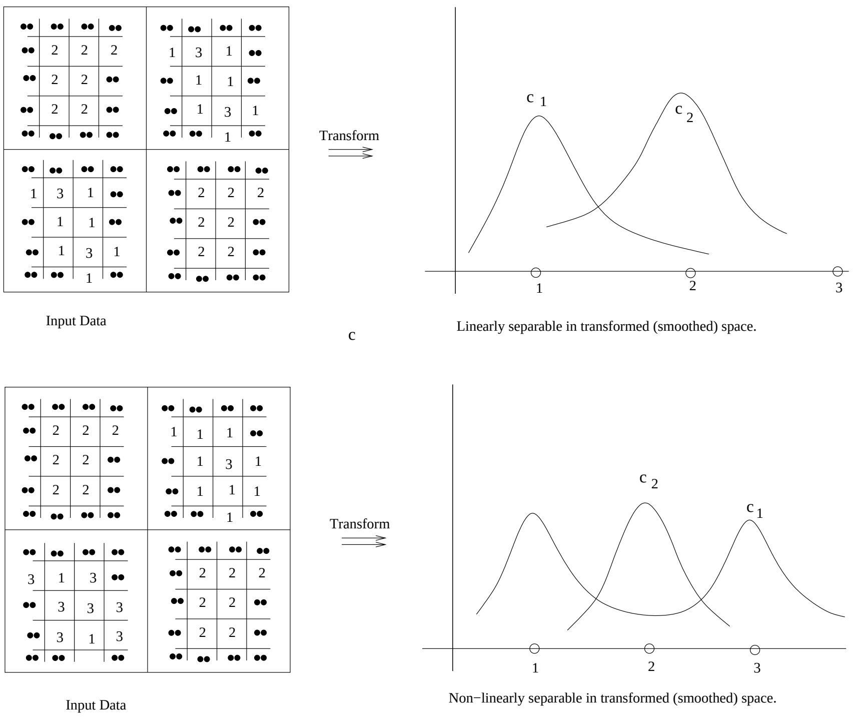

Transform
c 1
c 2
1
2
3
Input Data
c
Linearly separable in transformed (smoothed) space.
Transform
c 2
c 1
1
2
3
Input Data
Non−linearly separable in transformed (smoothed) space.

Figure 7.10: Spatial datasets with *salt and pepper* spatial patterns

Comparison of SAR and MRF Using a Probabilistic Framework

We use a simple probabilistic framework to compare SAR and MRF in this section. We will assume that classes $l_i \in (c_1, c_2, \ldots, c_M)$ are discrete and that the class label estimate $\hat{f}_C(s_i)$ for location s_i is a random variable. We also assume that feature values (X) are constant since there is no specified generative model. Model parameters for SAR are assumed to be constant, (i.e., β is a constant vector and ρ is a constant number). Finally, we assume that the spatial framework is a regular grid.

We first note that the basic SAR model can be rewritten as follows:

$$y = X\beta + \rho W y + \epsilon$$

$$(I - \rho W)y = X\beta + \epsilon$$

$$y = (I - \rho W)^{-1} X\beta + (I - \rho W)^{-1}\epsilon = (QX)\beta + Q\epsilon \tag{7.2}$$

where $Q = (I - \rho W)^{-1}$ and $\beta,\ \rho$ are constants (because we are modeling a particular problem). The effect of transforming feature vector X to QX can be viewed as a spatial smoothing operation. The SAR model is similar to the linear logistic model in terms of the transformed feature space. In other words, the SAR model assumes the linear separability of classes in transformed feature space.

Figure 7.10 shows two datasets with a *salt and pepper* spatial distribution of the feature values. There are two classes, c_1 and c_2, defined on this feature. Feature values close to 2 map to class c_2 and feature values close to 1 or 3 will map to c_1. These classes are not linearly separable in the original feature space. Local spatial smoothing can eliminate the *salt and pepper* spatial pattern in the feature values to transform the distribution of the feature values. In the top part of Figure 7.10, there are few values of 3 and smoothing revises them close to 1 since most neighbors have values of 1. SAR can perform well with this dataset since classes are linearly separable in the transformed space. However, the bottom part of Figure 7.10 shows a different spatial dataset where local smoothing does not make the classes linearly separable. Linear classifiers cannot separate these classes even in the transformed feature space assuming $Q = (I - \rho W)^{-1}$ does not make the classes linearly separable.

Although MRF and SAR classification have different formulations, they share a common goal, estimating the posterior probability distribution: $p(l_i|X)$. However, the posterior for the two models is computed differently with different assumptions. For MRF the posterior is computed using Bayes' rule. On the other hand, in logistic regression, the posterior distribution is directly fit to the data. For logistic regression, the probability of the set of labels L is given by:

$$Pr(L|X) = \prod_{i=1}^{N} p(l_i|X) \tag{7.3}$$

One important difference between logistic regression and MRF is that logistic regression assumes no dependence on neighboring classes. Given the logistic model, the probability that the binary label takes its first value c_1 at a location s_i is:

$$Pr(l_i|X) = \frac{1}{1 + \exp(-Q_i X \beta)} \tag{7.4}$$

where the dependence on the neighboring labels exerts itself through the W matrix, and subscript i (in Q_i) denotes the i^{th} row of the matrix Q. Here we have used the fact that y can be rewritten as in equation 7.2.

To find the local relationship between the MRF formulation and the logistic regression formulation (for the two class case $c_1 = 1$ and $c_2 = 0$), at point s_i

$$\begin{aligned} Pr((l_i = 1)|X, L_i) &= \frac{Pr(X|l_i = 1, L_i)Pr(l_i = 1, L_i)}{Pr(X|l_i = 1, L_i)Pr(l_i = 1, L_i) + Pr(X|l_i = 0, L_i)Pr(l_i = 0, L_i)} \\ &= \frac{1}{1 + \exp(-Q_i X \beta)} \end{aligned} \tag{7.5}$$

which implies

$$Q_i X \beta = \ln(\frac{Pr(X|l_i = 1, L_i)Pr(l_i = 1, L_i)}{Pr(X|l_i = 0, L_i)Pr(l_i = 0, L_i)}) \tag{7.6}$$

This last equation shows that the spatial dependence is introduced by the W term through Q_i. More importantly, it also shows that in fitting β we are trying to simultaneously fit the relative importance of the features and the relative frequency $(\frac{Pr(l_i=1,L_i)}{Pr(l_i=0,L_i)})$ of the labels. In contrast, in the MRF formulation, we explicitly *model* the relative frequencies in the class prior term. Finally, the relationship shows that we are making distributional assumptions about the class conditional distributions in logistic regression. Logistic regression and logistic SAR models belong to a more general exponential family. The exponential family is given by

$$Pr(u|v) = e^{A(\theta_v)+B(u,\pi)+\theta_v^T u} \tag{7.7}$$

where u, v are location and label respectively. This exponential family includes many of the common distributions such as Gaussian, Binomial, Bernoulli, and Poisson as special cases. The parameters θ_v and π control the form of the distribution. Equation 7.6 implies that the class conditional distributions are from the exponential family. Moreover the distributions $Pr(X|l_i = 1, L_i)$ and $Pr(X|l_i = 0, L_i)$ are matched in all moments higher than the mean (e.g., covariance, skew, kurtosis, etc.), such that in the difference $ln(Pr(X|l_i = 1, L_i)) - ln(Pr(X|l_i = 0, L_i))$, the higher order terms cancel out, leaving the linear term $(\theta_v^T u)$ in equation 7.7 on the left hand-side of equation 7.6.

7.4 Association Rule Discovery Techniques

Association rules are patterns of the form $X \to Y$. One of the more famous patterns in data mining, *Diapers* $\to$ *Beer*, is an example of an association rule. Association rules, as they are currently expressed, are a weaker form of correlation, since they do not discover negative associations. For example, the rule *Tofu* $\overset{neg}{\to}$ *Beef (people who buy tofu are not likely to buy beef)* may hold true but is not considered an association rule. In probabilistic terms an association rule $X \to Y$ is an expression of conditional probability, $P(Y|X)$.

An association rule is characterized by two parameters: *support* and *confidence*. Formally let $I = \{i_1, i_2, \ldots, i_k\}$ be a set of items, and $T = \{t_1, t_2, \ldots, t_n\}$ be a set of transactions, where each t_i is a *subset* of I. Let C be a subset of I. Then the *support* of C with respect to T is the number of transactions that contain C: $\sigma(C) = \{|t|t \in T, C \subset t$. Then $i_i \to i_2$ if and only if the following two conditions hold:

Support: i_1 and i_2 occur in at least s percent of the transactions: $\frac{\sigma(i_1 \wedge i_2)}{|T|}$.

Confidence: Of all the transactions in which i_1 occurs, at least c percent of them contain i_2: $\frac{\sigma(i_1 \wedge i_2)}{\sigma(i_1)}$.

Rule	Support	Confidence
$A \Rightarrow B$	0.50	1.0
$B \Rightarrow C$	0.25	0.33
$F \Rightarrow E$	0.25	1.0

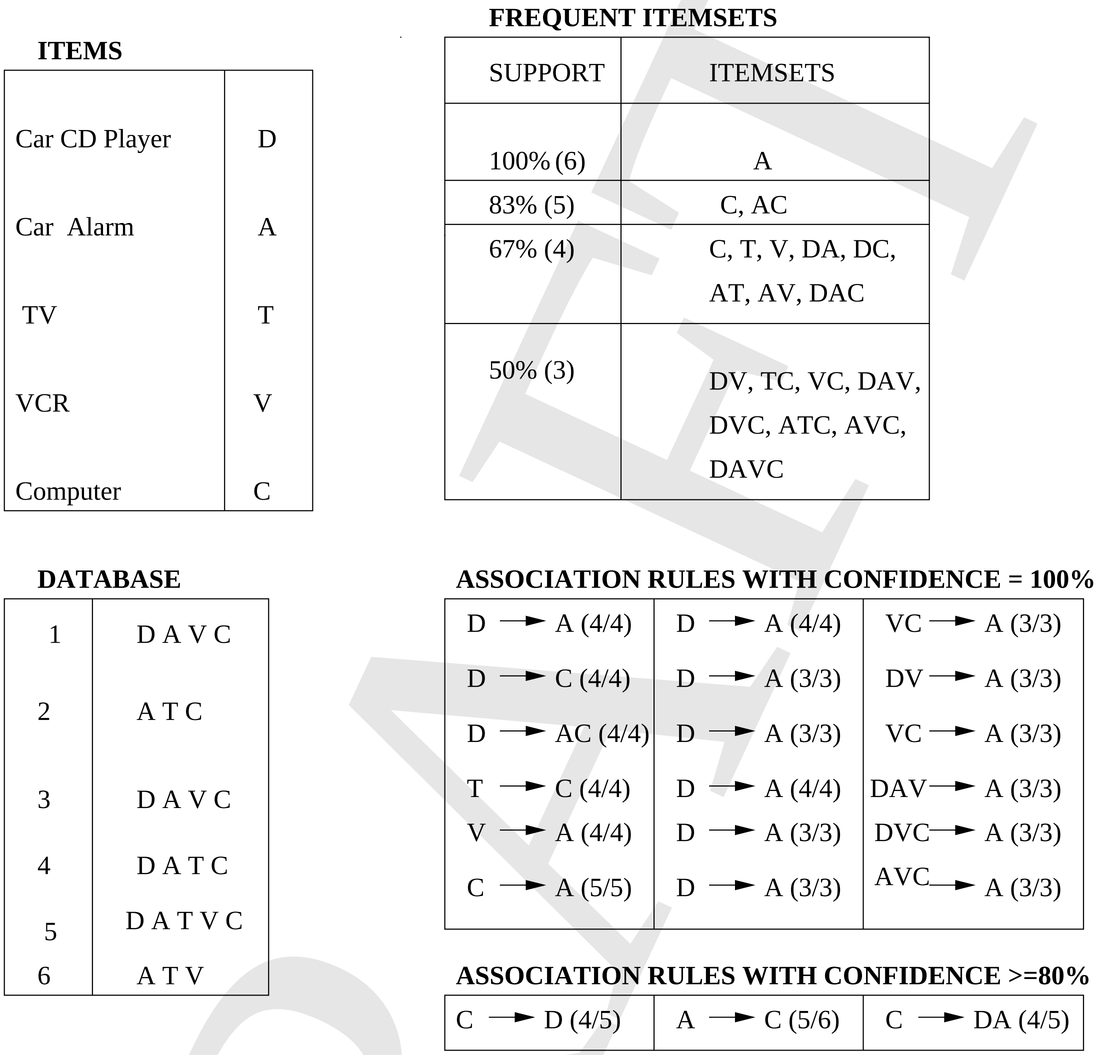
Table 7.4: Support and confidence of three rules

For example, consider a set $I = \{A, B, C, D, E, F\}$ of letters and a transaction set $T = \{ABC, ABD, BDE, CEF\}$ of words where the intra word ordering is irrelevant (i.e., $ABC = BCA = CAB$). Table 7.4 shows the support and confidence of three rules: $A \Rightarrow B$, $B \Rightarrow C$, $F \Rightarrow E$. For another example, see Figure 7.11, which shows a snapshot of sales at an electronics store. Also shown are examples of item sets which enjoy high support and association rules with high confidence. We now describe *Apriori*, an algorithm to rapidly discover association rules in large databases.

7.4.1 *Apriori*: An Algorithm for Calculating Frequent Itemsets

The *Apriori* algorithm is probably the most well-known algorithm for discovering frequent item sets. Frequent item sets are sets which satisfy the support threshold as defined above. The algorithm exploits a simple but fundamental property of the support measure: *If an itemset has high support, then so do all its subsets.* The outline of the *Apriori* algorithm is shown below.

```
FrequentItemSet := $\emptyset$ ;
$k$ := 1;
```



ITEMS

Car CD Player	D
Car Alarm	A
TV	T
VCR	V
Computer	C

FREQUENT ITEMSETS

SUPPORT	ITEMSETS
100% (6)	A
83% (5)	C, AC
67% (4)	C, T, V, DA, DC, AT, AV, DAC
50% (3)	DV, TC, VC, DAV, DVC, ATC, AVC, DAVC

DATABASE

1	D A V C
2	A T C
3	D A V C
4	D A T C
5	D A T V C
6	A T V

ASSOCIATION RULES WITH CONFIDENCE = 100%

D → A (4/4)	D → A (4/4)	VC → A (3/3)
D → C (4/4)	D → A (3/3)	DV → A (3/3)
D → AC (4/4)	D → A (3/3)	VC → A (3/3)
T → C (4/4)	D → A (4/4)	DAV → A (3/3)
V → A (4/4)	D → A (3/3)	DVC → A (3/3)
C → A (5/5)	D → A (3/3)	AVC → A (3/3)

ASSOCIATION RULES WITH CONFIDENCE >=80%

C → D (4/5)	A → C (5/6)	C → DA (4/5)

Figure 7.11: Example database, frequent itemsets, and high-confidence rules

```
**While** $CandidateSet_k \neq \emptyset$ **do**
        Create counter for each itemset in $CandidateSet_k$
        **forall** transactions in database **do**
                Increment counter of itemset in $CandidateSet_k$
                which occurs in the transaction;
        $Level_k :=$ All elements in $CandidateSet_k$ which
                        exceed the support threshold
        FrequentItemSet : = FrequentItemSet $\cup Level_k$;
        $CandidateSet_{k+1}$ : = All k+1-itemsets whose k-item subsets
                        are in $Level_k$.
        $k := k + 1$;
**end**
```

Apriori first discovers all the 1-itemsets (singletons) which are *frequent* (i.e, which exceed the support threshold). The second step is to combine all frequent itemsets to form 2-itemsets: $CandidateSet_2$. The algorithm then parses this set to search for frequent 2-itemsets. This process goes on: frequent 2-itemsets are combined to form 3-itemsets, until the set $CandidateSet_k$ is empty.

After all the frequent itemsets have been calculated, the next step is to search for rules which satisfy the minimum confidence requirement. This is done as follows. Given a frequent itemset $\{ABC, \}$ all combinations are checked to see if they satisfy the confidence parameter c. For example for each of the following rules,

$$\begin{array}{c} \{AB\} \rightarrow \{C\} \\ \{BC\} \rightarrow \{A\} \\ \{CA\} \rightarrow \{B\} \end{array}$$

the *confidence* measure is to be checked. Those that cross the threshold c are legitimate *association rules.*

There are two approaches towards generating spatial association rules. In the first approach the focus is on spatial predicates rather than items. The second approach generalizes the notion of a transaction to include neighborhoods (called co-location rules).

7.4.2 Spatial Association Rules

Spatial association rules are defined in terms of spatial predicates rather than items. A spatial association rule is a rule of the form

$$P_1 \wedge P_2 \wedge \ldots \wedge P_n \rightarrow Q_1 \wedge \ldots \wedge Q_m$$

where at least one of the $P_i's$ or $Q_j's$ is a spatial predicate. For example, the rule

$$is\_a(x, country) \wedge touches(x, Meditteranean) \overset{s\%, c\%}{\rightarrow} is_a(x, wine - exporter)$$

(i.e., a country which is adjacent to the Meditteranean Sea is a wine-exporter) is an association rule with support s and confidence c. Table 7.5 shows examples of association rules that were discovered in the Darr 1995 wetland data. Association rules were designed for categorically valued datasets, and therefore their application to datasets which are numerically valued is limited. This is because the transformation from numeric to categorical data involves a process of discretization which in most instances is quite arbitrary. For example, in the Darr wetland example, what is a `high Stem-Height`? Actually, because of spatial autocorrelation, the choice of discretization is probably less arbitrary, because if a location has `high Stem-Height,` then so do its neighboring locations.

7.4.3 Co-location Rules

Co-location rules attempt to generalize association rules to data sets which are indexed by space. There are several crucial differences between spatial and non-spatial associations including

Spatial Association Rule	**Sup.**	**Conf.**
$Stem\_height(x, high) \wedge Distance\_to\_edge(x, far) \rightarrow Vegetation\_Durability(x, moderate)$	0.1	0.94
$Vegetation\_Durability(x, moderate) \wedge Distance\_to\_water(x, close) \rightarrow Stem\_Height(x, high)$	0.05	0.95
$Distance\_to_{w}ater(x, far) \wedge Water\_Depth(x, shallow) \rightarrow Stem\_Height(x, high)$	0.05	0.94

Table 7.5: Examples of spatial association rules discovered in the 1995 Darr wetland data

1. The notion of an atomic *transaction* is absent in spatial situations. This is because spatial events are influenced by those in their neighborhood. For example, there is a high likelihood that regions with high per-capita income tend to tightly cluster near each.

2. Spatial data sets are item *sparse*, i.e., there are much fewer *items* in a spatial situation than in a non-spatial situation. For example, in a retail setting it is common to deal with distinct items which run into the tens of thousands. This is not the case for spatial data sets where the equivalent of spatial items are almost never more than one hundred. This implies that level-wise approaches, like *Apriori*, are not necessarily applicable in spatial situations.

3. In most instances, spatial *items* are discreteized version of continuous variables. For example, in the United States high per-capita income regions may be defined as regions where the mean yearly income is greater than fifty thousand dollars.

In this approach of spatial association rules discovery, the notion of a transaction is replaced by nighborhood. We explain this approach with the help of an example. The co-location pattern discovery process finds frequently co-located sub-sets of spatial event types given a map of their locations (see Figure 7.12). For example, analysis of habitats of animals and plants may identify co-location of predator-prey species, symbiotic species, and fire events with ignition sources. Readers may find it interesting to analyze the map in Figure 7.12 to find co-location patterns. There are two co-location patterns of size 2 in this map.

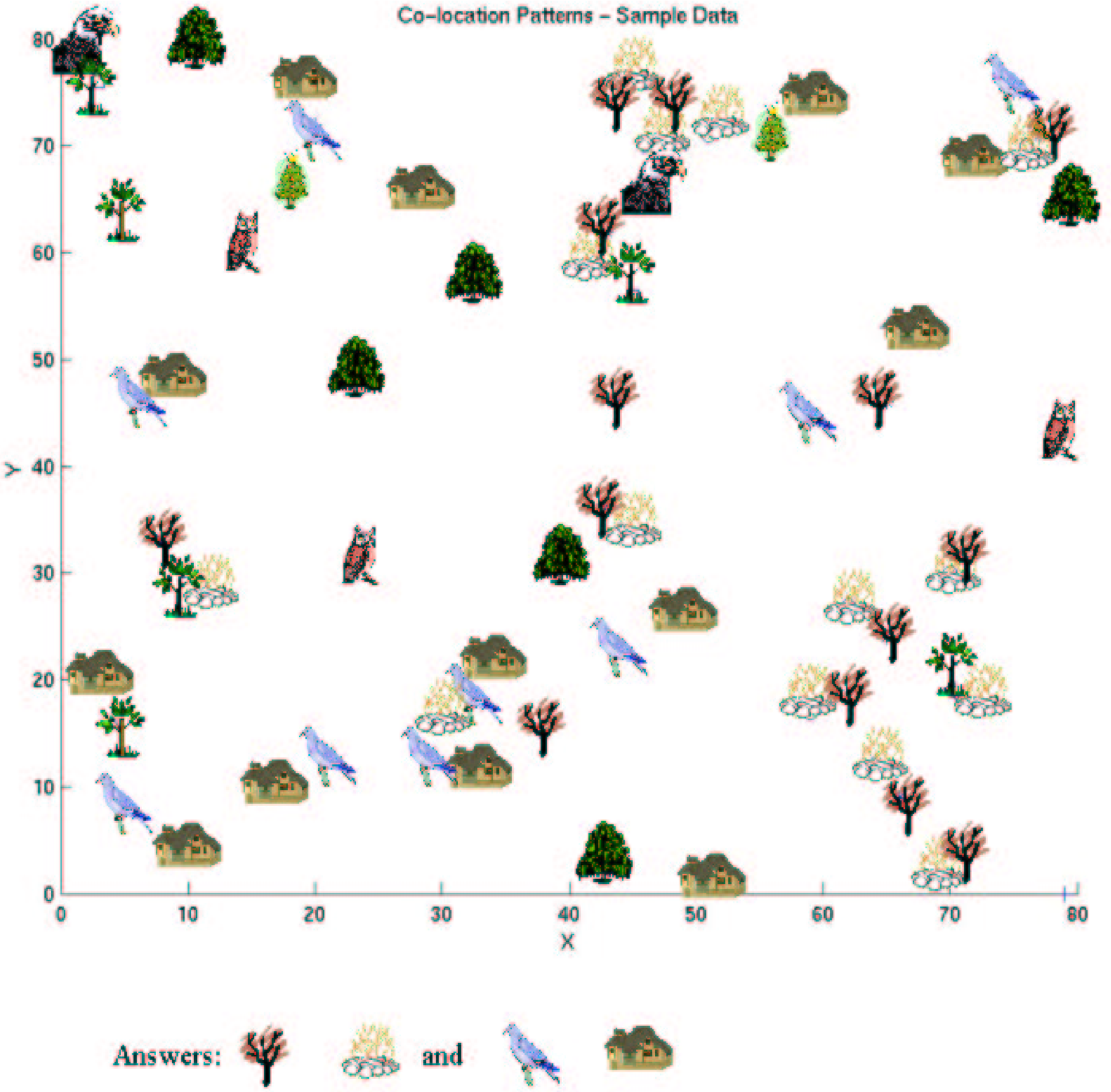


Figure 7.12: Sample co-location patterns

7.5 Clustering

Clustering is a process for discovering "groups", or clusters, in a large database. Unlike classification, clustering involves no a priori information either on the number of clusters or what the cluster labels are. Thus there is no concept of training or test data data in clustering. This is the reason that clustering is also referred as *unsupervised learning.*

The clusters are formed on the basis of a "similarity" criterion which is used to determine the relationship between each pair of tuples in the database. Tuples which are similar are usually grouped together, and then the group is labeled. For example, the pixels of satellite images are often clustered on the basis of the spectral signature. This way a remotely sensed image can be quickly segmented with minimal human intervention. Of course a domain expert does have to examine, verify, and possibly refine the clusters. A famous example of population segmentation occurred in the 1996 U.S. presidential election when political pundits identified "Soccer Moms" as the swing electorate who were then assiduously courted by major political parties. Clustering is another technique to determine the "hot spots" in crime analysis and disease tracking.

Clustering is a very well-known technique in statistics and the data-mining role is to scale a clustering algorithm to deal with the large datasets which are now becoming the norm rather than the exception. The size of the database is a function of the number of records in the table and also the number of attributes (the dimensionality) of each record. Besides the volume, the type of the data, whether it is numeric, binary, categorical, or ordinal is an important determinant in the choice of the algorithm employed.

It is convenient to frame the clustering problem in a multidimensional attribute space. Given n data objects described in terms of m variables, each object can be represented as a point in an m-dimensional space. Clustering then reduces to *determining high-density groups of points from a set of non-uniformly distributed points.* The search for potential within the multidimensional space is then driven by a suitably chosen similarity criterion.

For example, the counties in the United States can be clustered on the basis of, say, four attributes: *rate-of-unemployment, population, per-capita-income, and life-expectancy.* Counties which have similar values for these attributes will be grouped or clustered together.

When dealing with attribute data that is referenced in physical space, the clustering problem can have two interpretations. Consider the plot shown in Figure 7.13, which shows the variation of an attribute value (e.g., population density) as a function of location shown on the x-axis. Now what are the clusters, and how do we interpret them? For example, if our goal is to identify *central* cities and their zones of influence from a set of cities which dominate other cities as measured by the variance of an attribute value across the landscape, then we are looking for spatial clusters marked S1 and S2 in Figure 7.13. On the other hand, if our goal is to identify pockets in the landscape where an attribute (or attributes) are homogeneously expressed, then we are looking for clusters marked A1 and A2. While the second interpretation is essentially nonspatial, the spatial aspects exist because of the spatial autocorrelation that may exist in the attribute data. The clusters identified should be spatially homogeneous and not "speckled." These two interpretations of the clustering problem are formally defined as below:

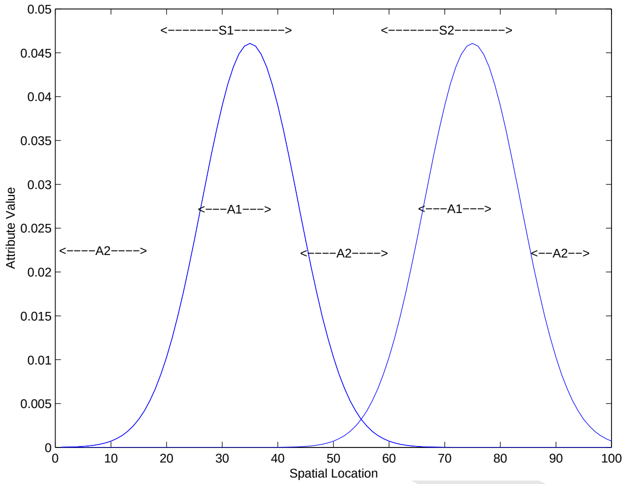


Figure 7.13: Two interpretations of spatial clustering. If the goal is to identify locations which dominate the surroundings (in terms of influence), then the clusters are S1 and S2. If the goal is to identify areas of homogeneous values, the clusters are A1 and A2.

Definition 1:

Given: A set $S = \{s_1, \ldots, s_n\}$ of spatial objects (e.g., points) and a real-valued, nonspatial attributes f evaluated on S, (i.e., $f : S \rightarrow R$).

Find: Two disjoint subsets of S, C, and $NC = S - C$, where $C = \{s_1, \ldots, s_k\}$, $NC = \{nc_1, \ldots, nc_l\}$, and $k < n$.

Objective: $\min_{C \subset S} \sum_{j=1}^{l} |f(nc_j) - \sum_{i=1}^{k} \frac{f(c_i)}{(dist(nc_j, c_i)}|^2$

Where: $dist(a, b)$ is the Euclidean or some other distance measure.

Constraints:

1. The dataset conforms to the theory of central places, which postulates that the influence of a central city decays as the square of the distance.
2. There is at most one nonspatial attribute.

Definition 2:

Given: (1) A set $S = \{s_1, \ldots, s_n\}$ of spatial objects (e.g., points) and a set of real-valued, nonspatial attributes f_i $i = 1, \ldots, I$ defined on S, (i.e., for each i, $f_k : S \rightarrow R$); and (2) neighborhood structure E on S.

Find: Find K subsets $C_k \subset S$, $k = 1, \ldots, K$ such that

Objective: $\min_{C_k \subset S} \sum_{C_k, s_i \in C_k, s_j \in C_k} dist(F(s_i), F(s_j)) + \sum_{i,j} nbddist(C_i, C_j)$

Where: (1) F is the cross-product of the $f_i's$, $i = 1, \ldots, n$; (2) $dist(a, b)$ is the Euclidean or some other distance measure and (3) $nbddist(C, D)$ is the number of points in C and D which belong to E.

Constraints: $|C_k| > 1$ for all $k = 1, \ldots, K$.

Categories of Clustering Algorithms

Cluster analysis is one of most often performed data analysis technique in many fields. This has resulted in a multitude of clustering algorithms, so it is useful to categorize them into groups. Based on the technique adopted to define clusters, the clustering algorithms can be categorized into four broad categories:

1. *Hierarchical* clustering methods starts with all patterns as a single cluster, and successively performs splitting or merging until a stopping criterion is met. This results in a tree of clusters, called *dendograms*. The dendogram can be cut at different levels to yield desired clusters. Hierarchical algorithms can further be divided into *agglomerative* and *divisive* methods. Some of the hierarchical clustering algorithms are: balanced iterative reducing and clustering using hierarchies (BIRCH), clustering using representatives (CURE), and robust clustering using links (ROCK).

2. *Partitional* clustering algorithms start with each pattern as a single cluster and iteratively reallocates data points to each cluster until a stoping criterion is met. These methods tend to find clusters of spherical shape. *K-Means* and *K-Medoids* are commonly used partitional algorithms. Squared error is the most frequently used criterion function in partitional clustering. Some of the recent algorithms in this category are: partitioning around medoids (PAM), clustering large applications (CLARA), clustering large applications based on randomized search (CLARANS), and expectation-maximization (EM).

3. *Density-based* clustering algorithms tries to find clusters based on density of data points in a region. These algorithms treat clusters as dense regions of objects in the data space. Some of the density-based clustering algorithms are: density-based spatial clustering of applications with noise (DBSCAN), and density based clustering (DENCLUE).

4. *Grid-based* clustering algorithms first quantize the clustering space into a finite number of cells and then perform the required operations on the quantized space. Cells that contain more than certain number of points are treated as dense. The dense cells are connected to form the clusters. Grid-based clustering algorithms are primarily developed for analyzing large spatial data sets. Some of the grid-based clustering algorithms are: statistical information grid-based method (STING), STING+, WaveCluster, BANG-clustering, and clustering in quest (CLIQUE).

Some times the distinction among these categories are diminishing, and some algorithms can even be classified into more than one group. For example, clustering in quest (CLIQUE) can be considered as both density-based and grid-based clustering method.

We now describe two well-known approaches to clustering, the *K-medoid* algorithm and mixture analysis using the expectation-maximization (EM) algorithm. We will also briefly discuss how the EM algorithm can be modified to account for the special nature of spatial data.

7.5.1 *K-medoid*: An Algorithm for Clustering

We will restrict our attention to points in the two-dimensional space R^2, though the technique can be readily generalized to a higher dimensional space. Given a set P of n data points, $P = \{p_1, p_2, \ldots, p_n\}$ in R^2, the goal of K-medoid clustering is to partition the data points into k clusters such that the following objective function is minimized:

$$J(M) = J(m_1, \ldots, m_k) = \sum_{i=1}^{k} \sum_{p \in C_i} d(p, m_i)$$

In $J(C)$, m_i is the representative point of a cluster C_i. If m_i is restricted to be a member of P, then it is called a *medoid*. On the other hand, if m_i is the average of the cluster points and not necessarily a member of P, then it is called the *mean*. Thus the K-mean and the K-medoid approaches are intimately related. Even though the K-mean algorithm is better well-known, we focus on the *K-medoid* approach because the medoid, like the median, is less sensitive to outliers.

The K-medoids characterize the K clusters, and each point in P belongs to its nearest medoid. Since we have restricted the ambient space to be R^2, the distance function d is the usual Euclidean distance.

$$d(p, m_i) = ((p(x) - m_i(x))^2 + (p(y) - m_i(y))^2)^{\frac{1}{2}}.$$

Thus the K-medoid approach transforms the clustering problem into a search problem. The search space X is the set of all k-subsets M of P (i.e., $|M| = k$), and the objective function is $J(M)$. X can be modeled as a graph, where the nodes of the graph are the elements of X. Two nodes M_1 and M_2 are *adjacent* if $|M_1 \cap M_2| = k - 1$ (i.e., they differ by one and only one data point).

The *K-medoid* algorithm consists of the following steps:

1. Choose an arbitrary node M_o in X.
2. Iteratively move from current node M_t to an adjacent node M_{t+1} such that $J(M_{t+1}) < J(M_t)$. The move from current node to adjacent node consists of replacing a current medoid m with a data point $p \in P$. Thus $M_{t+1} = M_t \cup \{p\} - \{m\}$.
3. Stop when $J(M_{t+1}) \geq J(M_t)$ for all adjacent nodes.

Step 2 is the heart of the algorithm. There are many options available to move from a node to its adjacent node. Table 7.6 lists some of the options. The table includes the name of each option as it is referred to in the literature, the strategy for moving, and whether the option will guarantee a local optima. All the options are examples of local search because only the adjacent nodes are explored.

Local Search	**Strategy to move from M_t to $M_{t+1} = M_t \cup \{p\} - \{m\}$**	**Guarentee local optima**
Global hill climbing (HC)	Move to the best neighbor	Yes
Randomized HC	Move to best of sampled neighbors	No
Local HC	Move to a new neighbor as soon as it is found	Yes
Distance-restricted HC	Move to best neighbor within a specified distance	No

Table 7.6: Four options for local search in clustering

7.5.2 Clustering, Mixture Analysis, and the EM Algorithm

One drawback of the K-medoid (or K-mean) approach is that it produces "hard" clusters; that is, each point is uniquely assigned to one and only one cluster. This can be a serious limitation because it is not known a priori what the actual clusters are. In the statistics literature the clustering problem is often recast in terms of *mixture models*. In a mixture model the data is assumed to be generated by a sequence of probability distributions where each distribution generates one cluster. The goal then is to identify the parameters of each probability distribution and their weights in the overall mixture distribution. In a mixture model each instance of the database belongs to all the clusters but with a different grade of membership, which is quantified by the weights of the individual distributions in the mixture model. Thus the mixture model framework is more flexible than the K-medoid approach. Typically each probability distribution is represented as a normal distribution, and the challenge is to determine the mean, variance, and weight of each distribution. The assumption of normality is not as restrictive as it might appear because a statistics theorem guarantees that any probability distribution can be expressed as a finite sum of normal distributions.

A Finite Mixture Example

Consider the gray-scale 4 × 4 image shown in Figure 7.14. Assume we want to partition the set of pixels into two clusters, A and B, where each cluster is modeled as a Gaussian distribution. The finite mixture problem is to calculate the parameters $\mu_A, \mu_B, \sigma_A, \sigma_B, p_A, p_B$.

For the moment, assume the cluster membership of each pixel is given as shown in Figure

12	15	25	20
17	10	2	18
11	5	17	17
7	9	12	13

(a)

1	2	2	2
2	1	1	1
1	1	2	2
1	1	2	1

(b)

1	1	2	2
1	1	1	2
1	1	1	1
1	1	1	1

(c)

Figure 7.14: (a) A gray-scale 4×4 image. (b) The labels of the image generated using the EM algorithm and (c) The labels generated for the same image using the Neighborhood EM algorithm. Notice the spatial smoothing attained by modifying the objective function.

7.14b. Then all the parameters can be easily calculated. For example,

$$\begin{aligned}
\mu_A &= \frac{12+10+2+18+11+5+7+9+13}{9} = 9.7 \\
\sigma_A &= \frac{(12-\mu_1)^2+(10-\mu_1)^2+\ldots+(13-\mu_1)^2}{8} = 4.7 \\
p_A &= \frac{9}{16}
\end{aligned}$$

Similarly $\mu_B = 17.6, \sigma_B = 4.1$, and $p_A = \frac{5}{16}$. Computing the probability of a given pixel value belonging to cluster is then a simple exercise using Bayes's Theorem. For example, given a pixel value x, the probability that it belongs to cluster A is

$$\begin{aligned}
P(A|x) &= \frac{P(x|A)p_A}{P(x)} \\
&= \frac{P(x|A)p_A}{P(x|A)p_A + P(x|B)p_B} \\
&= \frac{N(x,\mu_A,\sigma_A)p_A}{N(x,\mu_A,\sigma_A)p_A + N(x,\mu_B,\sigma_B)p_B}
\end{aligned}$$

where

$$N(x, \mu_A, \sigma_A) = \frac{1}{\sqrt{(2\pi)}\sigma_A} \exp^{\frac{-(x-\mu_A)^2}{2\sigma^2}}$$

Now in our case the cluster labels are not known and neither are the distribution parameters. All we know is that there are two clusters and that each cluster is modeled as a Gaussian distribution. At first this problem may appear to be unsolvable because there are too many unknowns: cluster labels for each pixel and the distribution parameters of the cluster. Problems of this type can be solved using the expectation-maximization (EM) algorithm. The EM algorithm, like the *K-medoid* algorithm, is an iterative algorithm which begins with the guess estimate of the distribution parameters. It then computes the "expected values" of the data given the initial parameters. The new, expected, data values are then used to calculate the maximum likelihood estimate for the distribution parameters(see the appendix for a brief discussion of maximum likelihood estimation). This procedure is iterated until some convergence criterion is met. The EM algorithm guarantees that the maximum likelihood estimate will improve after each iteration, though the convergence can be slow. The steps of the EM algorithm follow:

1. Guess the initial model parameters: u_A^0, Σ_A^0 and p_A^0 and u_B^0, Σ_B^0 and p_B^0.

2. At each iteration j, calcuate the probability that the data object x belongs to clusters A and B.
$$P(A|x) = \frac{p_A^j P^j(x|A)}{P^j(x)} \quad P(B|x) = \frac{p_B^j P^j(x|B)}{P^j(x)}$$

3. Update the mixture parameters on the basis of the new estimate:
$$p_A^{j+1} = \tfrac{1}{n}\textstyle\sum_x P(A|x) \qquad p_B^{j+1} = \tfrac{1}{n}\textstyle\sum_x P(B|x)$$
$$\mu_A^{j+1} = \frac{\sum_x xP(A|x)}{\sum_x P(A|x)} \qquad \mu_B^{j+1} = \frac{\sum_x xP(B|x)}{\sum_x P(B|x)}$$
$$\sigma_A^{j+1} = \frac{\sum_x P(A|x)(x-\mu_A^{j+1})^2}{\sum_x P(A|x)} \qquad \sigma_B^{j+1} = \frac{\sum_x P(B|x)(x-\mu_B^{j+1})^2}{\sum_x P(B|x)}$$

4. Compute the log estimate $E_j = \sum_x log(P^j(x))$. If for some fixed stopping criterion ϵ, $|E_j - E_{j+1}| \leq \epsilon$, then stop; else set $j = j + 1$.

The Neighborhood EM Algorithm

A careful reader may have noticed that the EM algorithm completely ignores the spatial distribution of the pixel; it only works with the pixel values. Thus if we rearrange the pixel values shown in Figure 7.14a, the EM algorithm will still come up with the same cluster labeling and the same values of the distribution parameters.[2] Such a solution, as we know, does not take into account the spatial autocorrelation property inherent in spatial data. As we have mentioned before, the search space for spatially referenced data is a combination of a conceptual attribute space and the physical (geographic) space. The spatial autocorrelation property then implies that the clusters should vary gradually in the physical space.

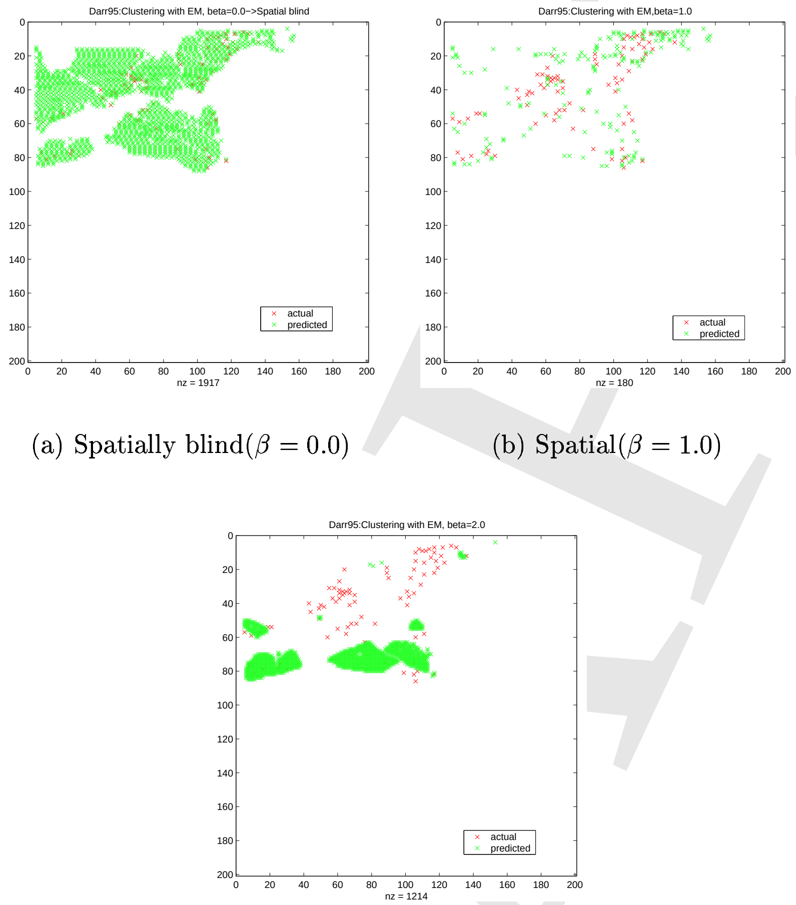
In order to make the EM algorithm spatially sensitive, we first follow the recipe proposed by Ambroise et al. (1997).

Step 1: The EM algorithm for mixture models is equivalent to the optimization of the following objective function:

$$D(c, \mu_k, \sigma_k, p_k) = \sum_{k=1}^{2}\sum_{i=1}^{n} c_{ik}\log(p_k N(x_i, \mu_k, \sigma_k)) - \sum_{k=1}^{2}\sum_{i=1}^{n} c_{ik} log(c_{ik})$$

where $\mathbf{c} = \mathbf{c_{ik}}, \mathbf{i} = \mathbf{1}, \ldots, \mathbf{n}$ and $k = 1, \ldots K$ define a fuzzy classification representing the grade of membership of data point $\mathbf{x_i}$ into cluster k. The c_{ik}'s satisfy the constraints $(0 < c_{ik} < 1,\ \sum_{k=1}^{2} c_{ik} = 1, \sum_{i=1}^{n} c_{ik} > 0)$. Again we have two clusters $k = 1, 2$, and there are n data points.

[2] Actually because of the randomness of the initial parameters, each run of the EM algorithm can potentially result in a different solution.



(a) Spatially blind($\beta = 0.0$)

(b) Spatial($\beta = 1.0$)

(c) Spatial($\beta = 2.0$)

Figure 7.15: Using the Neighborhood EM algorithm. (a) As expected clustering without any spatial information leads to poor results. (b) including spatial information ($\beta = 1.0$) leads to dramatic improvement of results. (c) overemphasizing spatial information ($\beta = 2.0$) again leads to poor results.

Step 2 In order to account for spatial autocorrelation, we introduce a new term,

$$G(c) = \frac{1}{2}\sum_{k=1}^{2}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ik}c_{jk}w_{ij}$$

where $W = (w_{ij})$ is the contiguity matrix as defined before.

The new "spatially weighted" objective function is

$$U(c, \mu, \sigma) = D(c, \mu, \sigma) + \beta G(c)$$

where $\beta \geq 0$ is a parameter to control the spatial homogeneity of the dataset.

Step 3: Except for the new parameter c, which is an $n \times 2$ matrix, all the parameters are calculated exactly as before. The formula for c'_{ik} is

$$c_{ik}^{m+1} = \frac{p_k^m N(x_i, \mu_k, \sigma_k) \exp\{\beta \sum_{j=1}^{d=n} c_{jk}^{m+1} w_{ij}\}}{\sum_{l=1}^{2} p_l^m N(x_i, \mu_l^m, \sigma_l^m) \exp\{\beta \sum_{j=1}^{n} c_{jl}^{m+1} w_{ij}}$$

At each iteration m ${}_{ik}$then c'can be solved using a fixed point iterative scheme.

We have carried out experiments using the neighborhood EM (NEM) algorithm on the bird dataset. We assume two clusters corresponding to the presence/absence of nests. When $\beta = 0$, the NEM reduces to the classical EM algorithm. We varied the β parameters, and the results are shown in Figure 7.15. The results lead us to conclude that including spatial information in the clustering algorithm leads to a dramatic improvement of results (Figure 7.15b compared with Figure 7.15a), but overemphasizing spatial information leads to "oversmoothing" and degradation in accuracy.

7.5.3 Strategies for Clustering Large Spatial Databases

We now show how the *K-medoid* algorithm can be *scaled* by taking advantage of *spatial* index structures that were introduced in chapter 4.

Assume we have a spatial database of n points which is too large for all the points to reside in the main memory at the same time. We make the following additional assumptions:

1. A spatial index structure like the R^*-tree or Z-order is available in the SDBMS.
2. c is the average number of points stored in a disk page.
3. k is the number of clusters.
4. The cost of the *K-medoid* algorithm is dominated by Step 2, the computation of $J(M_{t+1}) - J(M_t)$.

Sampling via the R^*-Tree

The leaves of the R^*-tree correspond to collections of points associated with a minimum bounding box (MBR). We can choose a representative sample of the n points by selecting one sample point from each leaf node. A natural choice of the sample point is a data point *closest* to the centroid of the MBR. Thus instead of n points the algorithm only has to cluster, on the average, n/c points.

Choose Only Relavant Clusters

One way to compute $J(M_{t+1}) - J(M_t)$, is to loop over all the nonmedoid points and calculate the distance *afresh*. The cost associated with such a strategy can be prohibitive, given the large size of the database. Fortunately only the nonmedoid points associated with the old medoid m and new medoid p in $M_{t+1} = M_t \cup \{p\} - \{m\}$ contribute to $J(M_{t+1}) - J(M_t)$. Thus only the cluster points of m and p have to be fetched into the main memory. The success of this approach is predicated upon the efficient retrieval of points corresponding to a cluster.

One way to efficiently retrieve the cluster points of a medoid is to observe that the *Voronoi polygons* associated with the medoids contain their cluster points. Thus a range query where the query region is the Voronoi polygon will retrieve all the cluster points of a medoid m. Such a query can be efficiently processed using either the R^*-tree or the Z-order index.

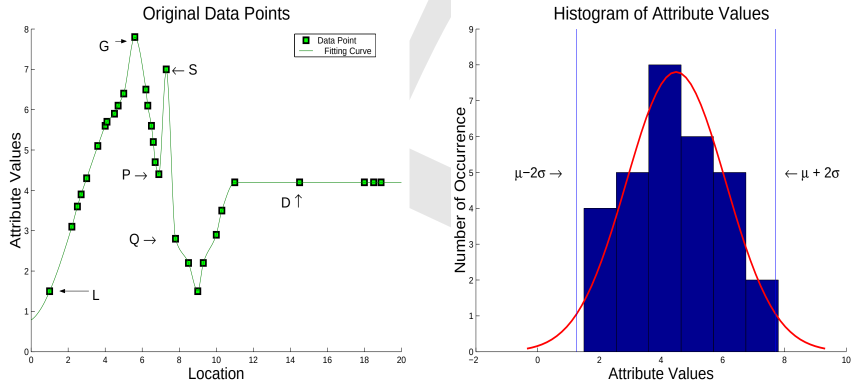
7.6 Spatial Outlier Detection

Global outliers have been informally defined as observations in a data set which appear to be inconsistent with the remainder of that set of data [Barnett and Lewis, 1994], or which deviate so much from other observations so as to arouse suspicions that they were generated by a different mechanism [Hawkins, 1980]. The identification of global outliers can lead to the discovery of unexpected knowledge and has a number of practical applications in areas such as credit card fraud, athlete performance analysis, voting irregularity, and severe weather prediction. This section focuses on spatial outliers, i.e., observations which appear to be inconsistent with their neighborhoods. Detecting spatial outliers is useful in many applications of geographic information systems and spatial databases. These application domains include transportation, ecology, public safety, public health, climatology, and location based services.

We model a spatial data set to be a collection of spatially referenced objects, such as houses, roads, and traffic sensors. Spatial objects have two distinct categories of dimensions along which attributes may be measured. Categories of dimensions of interest are spatial and non-spatial. Spatial attributes of a spatially referenced object includes location, shape, and other geometric or topological properties. Non-spatial attributes of a spatially referenced object include traffic-sensor-identifiers, manufacturer, owner, age, and measurement readings. A spatial neighborhood of a spatially referenced object is a subset of the spatial data based on a spatial dimension, e.g., location. Spatial neighborhoods may be defined based on spatial attributes, e.g., location, using spatial relationships such as distance or adjacency. Comparisons between spatially referenced objects are based on non-spatial attributes.

A spatial outlier is a spatially referenced object whose non-spatial attribute values are significantly different from those of other spatially referenced objects in its spatial neighborhood. Informally, a spatial outlier is a local instability (in values of non-spatial attributes) or a spatially referenced object whose non-spatial attributes are extreme relative to its neighbors, even though they may not be significantly different from the entire population. For example, a new house in an old neighborhood of a growing metropolitan area is a spatial outlier based on the non-spatial attribute house age.

We use an example to illustrate the differences among global and spatial outlier detection methods. In Figure 7.16(a), the X-axis is the location of data points in one dimensional space; the Y-axis is the attribute value for each data point. Global outlier detection methods ignore the spatial location of each data point, and fit the distribution model to the values of the non-spatial attribute. The outlier detected using a this approach is the data point G, which has an extremely high attribute value 7.9, exceeding the threshold of $\mu + 2\sigma = 4.49 + 2 * 1.61 = 7.71$, as shown in Figure 7.16(b). This test assumes a normal distribution for attribute values.

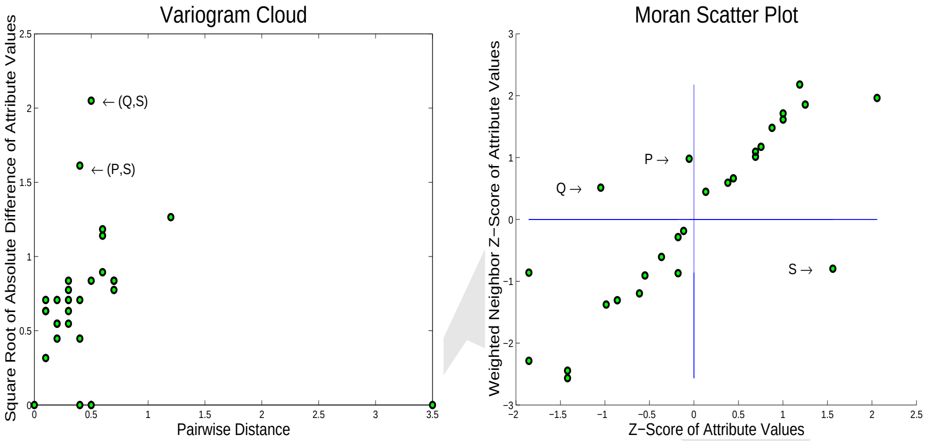


(a) An Example Data Set (b) Histogram

Figure 7.16: A Data Set for Outlier Detection

Tests to detect spatial outliers separte the spatial attributes from the non-spatial attributes. Spatial attributes are used to characterize location, neighborhood, and distance. Non-spatial attribute dimensions are used to compare a spatially referenced object to its neighbors. Spatial statistics literature provides two kinds of tests, namely graphical tests and quantitative tests. Graphical tests are based on visualization of spatial data which highlight spatial outliers. Example methods include variogram clouds and Moran scatterplots. Quantitative methods provide a precise test to distinguish spatial outliers from the remainder of data. Scatterplots [Luc, 1994] are a representative technique from the quantitative family.

A variogram-cloud displays data points related by neighborhood relationships. For each pair of locations, the square-root of the absolute difference between attribute values at the locations versus the Euclidean distance between the locations are plotted. In data sets exhibiting strong spatial dependence, the variance in the attribute differences will increase with increasing distance between locations. Locations that are near to one another, but with large attribute differences, might indicate a spatial outlier, even though the values at both locations may appear to be reasonable when examining the data set non-spatially. Figure 7.17(a) shows a variogram cloud for the example data set shown in Figure 7.16(a). This plot shows that two pairs (P, S) and (Q, S) in the left hand side lie above the main group of pairs, and are possibly related to spatial outliers. The point S may be identified as a spatial outlier since it occurs in both pairs (Q, S) and (P, S). However, graphical tests of spatial outlier detection are limited by the lack of precise criteria to distinguish spatial outliers. In addition, a variogram cloud requires non-trivial post-processing of highlighted pairs to separate spatial outliers from their neighbors, particularly when multiple outliers are present or density varies greatly.



(a) Variogram cloud

(b) Moran scatterplot

Figure 7.17: Variogram Cloud and Moran Scatterplot to Detect Spatial Outliers

A Moran scatterplot [Luc, 1995] is a plot of normalized attribute value ($Z[f(i)] = \frac{f(i)-\mu_f}{\sigma_f}$) against the neighborhood average of normalized attribute values ($W \cdot Z$), where W is the row-normalized (i.e., $\sum_j W_{ij} = 1$) neighborhood matrix, (i.e., $W_{ij} > 0$ iff neighbor(i,j)). The upper left and lower right quadrants of Figure 7.17(b) indicate a spatial association of dissimilar values: low values surrounded by high value neighbors (e.g., points P and Q), and high values surrounded by low values (e.g,. point S). Thus we can identify points(nodes) that are surrounded by unusually high or low value neighbors. These points can be treated as spatial outliers.

Definition: $Moran_{outlier}$ is a point located in upper left and lower right quadrants of Moran scatterplot. This point can be identified by $(Z[f(i)]) \times (\sum_j (W_{ij} Z[f(j)])) < 0$.

A scatterplot [Luc, 1994] shows attribute values on the X-axis and the average of the attribute values in the neighborhood on the Y-axis. A least square regression line is used to identify spatial outliers. A scatter sloping upward to the right indicates a positive spatial autocorrelation (adjacent values tend to be similar); a scatter sloping upward to the left indicates a negative spatial autocorrelation. The residual is defined as the vertical distance (Y-axis) between a point P with location (X_p, Y_p) to the regression line $Y = mX + b$, that is, residual $\epsilon = Y_p - (mX_p + b)$. Cases with standardized residuals, $\epsilon_{standard} = \frac{\epsilon - \mu_\epsilon}{\sigma_\epsilon}$, greater than 3.0 or less than -3.0 are flagged as possible spatial outliers, where μ_ϵ and σ_ϵ are the mean and standard deviation of the distribution of the error term ϵ. In Figure 7.18(a), a scatter plot shows the attribute values plotted against the average of the attribute values in neighboring areas for the data set in Figure 7.16(a). The point S turns out to be the farthest from the regression line and may be identified as a spatial outlier.

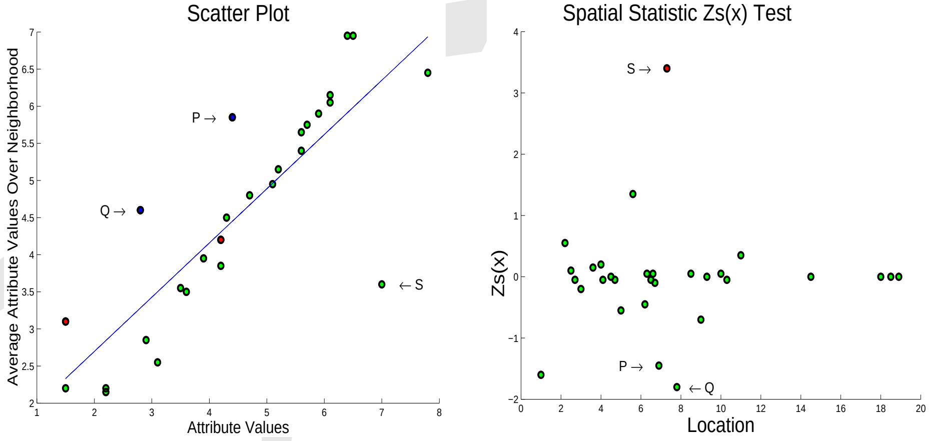
Definition: $Scatterplot_{outlier}$ is a point with significant standardized residual error from the least square regression line in a scatter plot. Assuming errors are normally distributed, then

$\epsilon_{standard} = |\frac{\epsilon-\mu_\epsilon}{\sigma_\epsilon}| > \theta$ is a common test. Nodes with standardized residuals $\epsilon_{standard} = \frac{\epsilon-\mu_\epsilon}{\sigma_\epsilon}$ from regression line $Y = mX + b$ and greater than θ or less than $-\theta$ are flagged as possible spatial outliers. The μ_ϵ and σ_ϵ are the mean and standard deviation of the distribution of the error term ϵ.

A location may also be compared to its neighborhood using the function $S(x) = [f(x) - E_{y\in N(x)}(f(y))]$, where $f(x)$ is the attribute value for a location x, $N(x)$ is the set of neighbors of x, and $E_{y\in N(x)}(f(y))$ is the average attribute value for the neighbors of x. The statistic function $S(x)$ denotes the difference of the attribute value of a sensor located at x and the average attribute value of $x's$ neighbors.

Spatial Statistic $S(x)$ is normally distributed if the attribute value $f(x)$ is normally distributed. A popular test for detecting spatial outliers for normally distributed $f(x)$ can be described as follows: Spatial statistic $Z_{s(x)} = |\frac{S(x)-\mu_s}{\sigma_s}| > \theta$. For each location x with an attribute value $f(x)$, the $S(x)$ is the difference between the attribute value at location x and the average attribute value of $x's$ neighbors, μ_s is the mean value of $S(x)$, and σ_s is the value of the standard deviation of $S(x)$ over all stations. The choice of θ depends on a specified confidence level. For example, a confidence level of 95 percent will lead to $\theta \approx 2$.

Figure 7.18(b) shows the visualization of spatial statistic $Z_{s(x)}$ method described earlier in Section 1.1 and Example 1. The X-axis is the location of data points in one dimensional space; the Y-axis is the value of spatial statistic $Z_{s(x)}$ for each data point. We can easily observe that the point S has the $Z_{s(x)}$ value exceeding 3, and will be detected as spatial outlier. Note the two neighboring points P and Q of S have $Z_{s(x)}$ values close to -2 due to the presence of spatial outlier in their neighborhoods. Example 1 has already shown that $Z_{s(x)}$ is a special case of S-outlier.



(a) Scatterplot

(b) Spatial statistic $Z_{s(x)}$

Figure 7.18: Scatterplot and Spatial Statistic $Z_{s(x)}$ to Detect Spatial Outliers

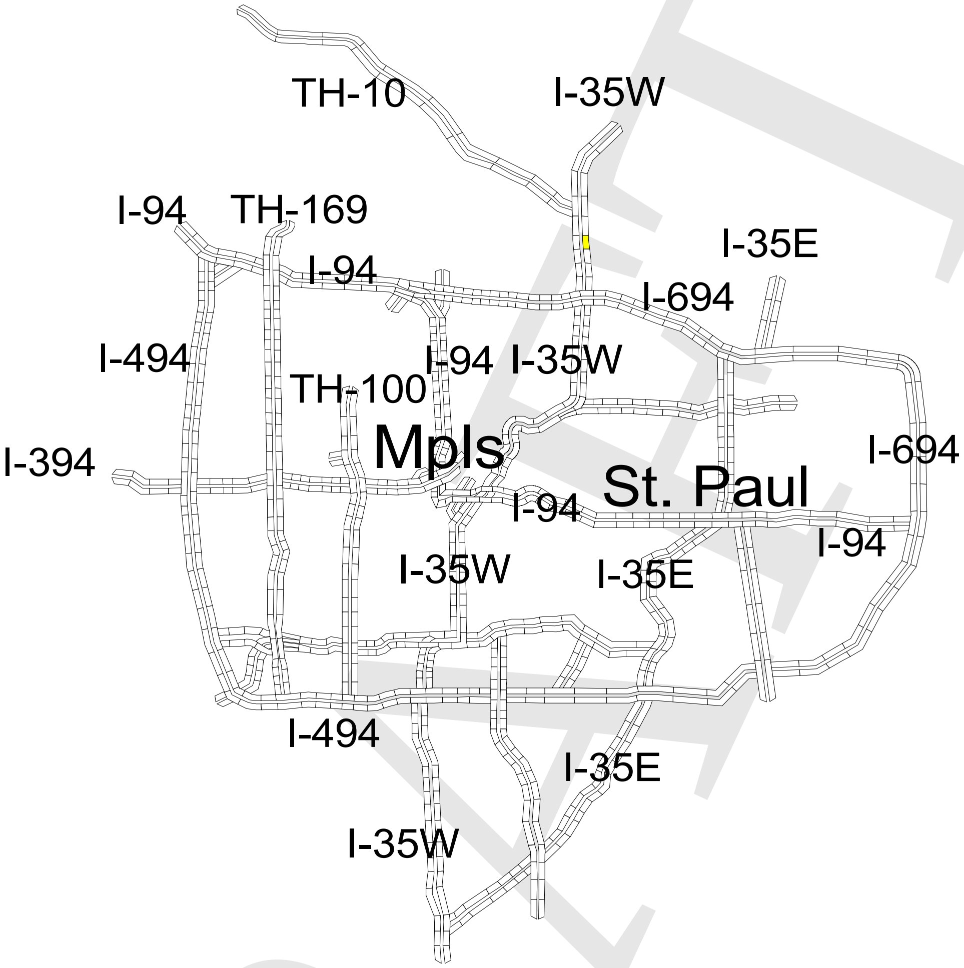


Figure 7.19: A network of traffic sensor stations

We now give an application domain case study of spatial outliers. The map shown in Figure 7.19 shows a network of sensor stations embedded in Interstate highways surrounding the Twin-Cities metropolitan area in Minnesota, USA. Each of the nine hundred stations measure the traffic volume and occupancy on a particular stretch of the highway at regular intervals. The natural notion of a neighborhood is defined in terms of graph connectivity rather than Euclidean distance. Our objective is to determine stations which are "outliers" based on the values of the volume and occupancy at each station.

The three neighborhood definitions we consider are shown in Figure 7.20. We consider spatio-temporal neighborhoods because time along with space are crucial for the discovery of spatial outliers. In Figure 7.20, $\{s_1, t_1\}$ and $\{s_3, t_3\}$ are spatial neighbors of of $\{s_2, t_2\}$ if s_1 and s_3 are connected to s_2 in a spatial graph. Two data points $\{s_2, t_3\}$ are temporal neighbors of $\{s_2, t_2\}$ if t_1, t_2 and t_3 are consecutive time slots. In addition, we define a neighborhood based on both space and time series as a spatial-temporal neighborhood. In Figure 7.20, $(s_1, t_1), (s_1, t_2), (s_1, s_3), (s_2, t_1)$ are the spatial-temporal neighbors of (s_2, t_2) if s_1 and s_3 are connected to s_2 in a spatial graph and t_1, t_2 and t_3 are consecutive time slots.

The choice of the test statistic to probe the existence of outliers is the next task to

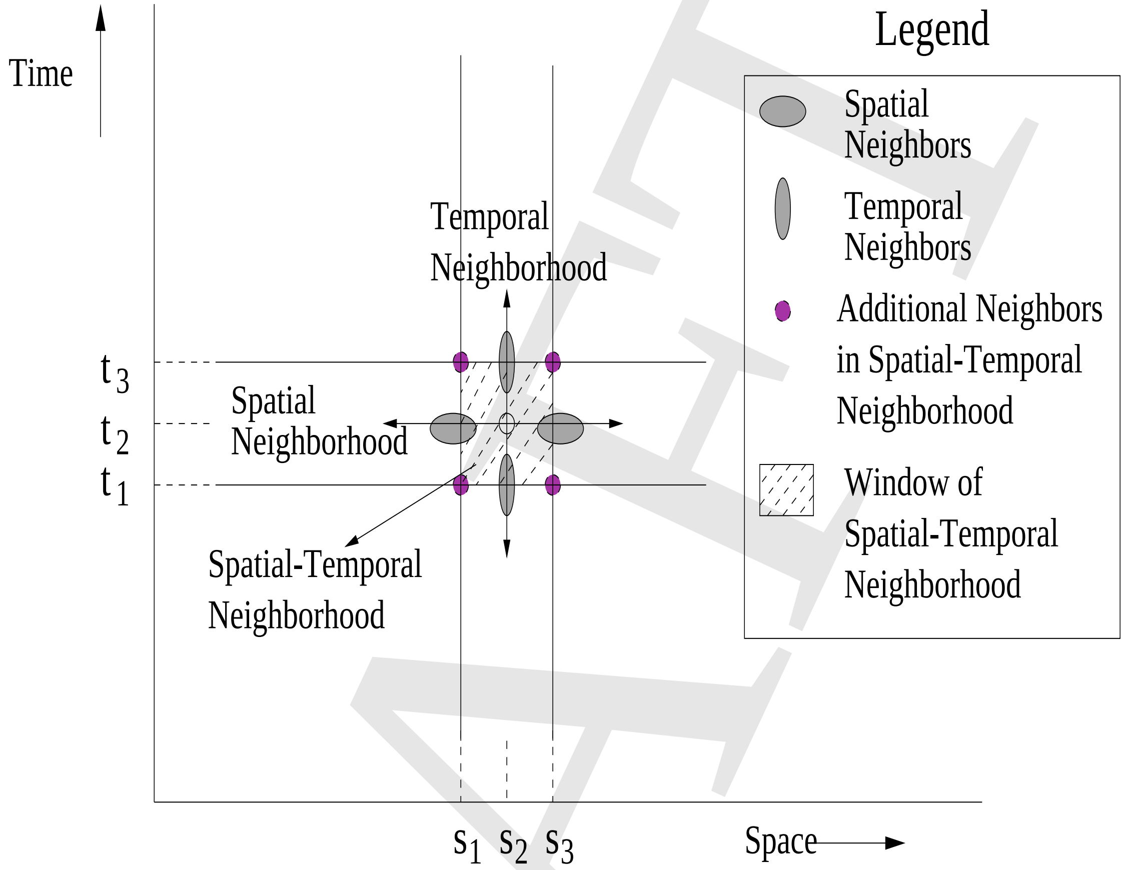


Figure 7.20: Spatial and temporal neighborhoods

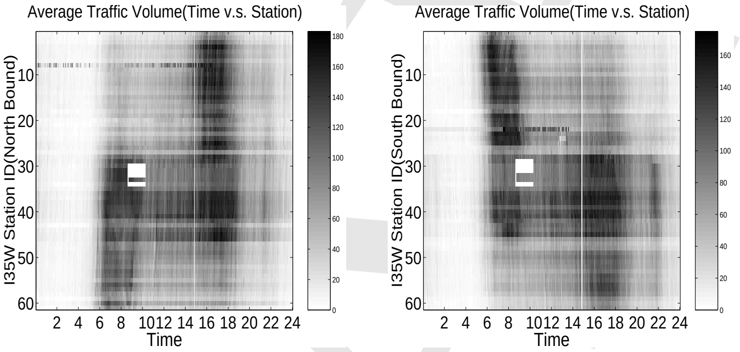
consider. In this application we used $S(x) = [f(x) - E_{y \in N(x)}(f(y))]$, where $f(x)$ is the attribute value(volume or occupancy) for neighbors of x. If $f(x)$ has a Normal distribution then it can be shown the $S(x)$ has a Normal distribution too(see Exercises). A data point x is considered an outlier if the z-score

$$\frac{S(x) - \mu_s}{\sigma_s} > \theta$$

The choice of θ depends on the specified confidence interval. For example, a confidence interval of 95 percent will lead to $\theta = 2$.

The third and final task for detecting outliers is the design and application of an "efficient" algorithm to calculate the test statistic and apply the outlier detection test. This a non-trivial task because the size of typical data sets are too large to fit in the primary memory. For example in the traffic data there are approximately one thousand sensors and they emit a reading every five minutes. Thus for a six month time frame, and assuming each reading generates 100 bytes of data, the size of the data set is approximately $100 \times 12 \times 24 \times 180 \times 1000 = 5$ Gigabytes. Thus it becomes imperative that I/O efficient algorithms be used to discover

The effectiveness of $Z_{s(x)}$ method on the Minneapolis-St. Paul Twin-Cities traffic data set is illustrated in the following example. Figure 7.21 shows one example of traffic flow outliers. Figures 7.21(a) and (b) are the traffic volume maps for I-35W north bound and south bound, respectively, on 1/21/1997. The X-axis is a 5-minute time slot for the whole day and the Y-axis is the label of the stations installed on the highway, starting from 1 on the north end to 61 on the south end. The abnormal white line at 2:45PM and the white rectangle from 8:20AM to 10:00AM on the X-axis and between stations 29 to 34 on the Y-axis can be easily observed from both (a) and (b). The white line at 2:45PM is an instance of temporal outliers, where the white rectangle is a spatial-temporal outlier. Both represent missing data. Moreover, station 9 in Figure 7.21(a) exhibits inconsistent traffic flow compared with its neighboring stations, and was detected as a spatial outlier. Station 9 may be a malfunctioning sensor.



(a) I-35W North Bound

(b) I-35W South Bound

Figure 7.21: Spatial outliers in traffic volume data

7.7 Summary

Data mining is a rapidly developing area which lies at the intersection of database management, statistics and artificial intelligent. Data mining provides semi-automatic techniques for discovering unexpected patterns in very large quantities of data.

Spatial data mining is a niche area within data mining for the rapid analysis of spatial data. Spatial data mining has can potentially influence major scientific challenges including global climate change and genomics.

The distinguishing characteristic of spatial data mining can be neatly summarized by the first law of geography: All things are related but nearby things are more related than distant things. The implication of this statement is that the standard assumption of independence and identically distributed (iid) random variables, which characterize classical data mining, is not applicable for the mining of spatial data. Spatial statisticians have coined the word *spatial-autocorrelation* to capture this property of spatial data.

The important techniques in data mining are : *association rules, clustering, classification* and *regression*. Each of these techniques have to be modified before they can be used to mine spatial data. In general there are two strategies available to modify data mining techniques to make them more sensitive for spatial data: the underlying statistical model which is based on the iid assumption can be corrected or the objective function which drives the search can be modified to include a spatial term. The Spatial Autoregressive Regression technique is an example of the first approach and the Neighborhood EM algorithm is an example of the latter.

7.8 Appendix: Bayesian Calculus

Probability theory provides a mathematical framework to deal with uncertainty. It is also a cornerstone of data mining, because in data mining we are trying to *generalize* our findings on the basis of a finite, albeit large, database.

Given a set of *events* Ω, the *probability* P is a function from Ω into $[0, 1]$ which satisfies the following two axioms:

1. $P(\Omega) = 1$.

2. If A and B are mutually exclusive events (like the rolling of two dice), then

$$P(AB) = P(A)P(B)$$

7.8.1 Conditional Probability

The notion of *conditional* probability is central to data mining. A conditional probability, $P(A|B) = \alpha$, means that, given the event B has occurred, the probability of event A is α. Thus if event B has occurred and everything else is irrelevant to A, then $P(A) = \alpha$.

The basic rule for probability calculus is the following:

$$P(AB) = P(A|B)P(B) = P(B|A)P(A).$$

In words this statement says that the joint-probability $P(AB)$ is the product of the conditional ($P(A|B)$ and the marginal $P(B)$. A simple manipulation of the above rule results in Bayes's Theorem.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

In the context of Bayes's rule, $P(A|B)$ is called the *posterior* probability, and the $P(B)$ is called the *prior*. Bayes's rule allows the inversion of probabilities, which is the cornerstone of classification. For example, this allows probabilities to be calculated on the test data based on the probabilities calculated on the training data.

7.8.2 Maximum Likelihood

Suppose we know that a random variable A is governed by a normal distribution $N(\theta)$, where $\theta = (\mu, \sigma)$ are the mean and standard deviation of the distribution. The goal of probability theory is to study the chances of A in a sample space, given that θ is fixed. In statistics, we invert the problem (using Bayes's Theorem) and study the chances of the parameter θ, given that A has happened (fixed). The normal distribution(or any other distribution) $N(A, \theta)$ as a function of θ (not x) is the likelihood function of θ. We then want to choose θ that has the *maximum likelihood* of generating the data A. This point of view connects a statistical problem with differential calculus.

Bibliographic Notes

7.1 Data mining is a rapidly developing area which lies at the intersection of database management, artificial intelligence, and statistics. Spatial data mining is an important but evolving area within data mining. [Han et al., 1997] design a system prototype of spatial data mining.

7.1 The data for the Darr and Stubble wetlands was collected by Uygar Ozesmi. Classical data-mining techniques, like logistic regression and neural networks, were applied to predict the location of bird nests.

7.2 Spatial statisticians have grappled with spatial data for a long time. They have identified and coined terms like *spatial autocorrelation* and *spatial heterogeneity* to identify the unique properties of spatial data.

7.3 For an overview of classical data-mining techniques consult [Han and Kamber, 2000]. Spatial regression has been extensively discussed in [Anselin, 1988, LeSage, 1997]. LeSage has also provided an excellent MATLAB toolbox on the web for different spatial regression models.

7.4 The *Apriori* algorithm was introduced by [Agrawal and Srikant, 1994]. [Koperski and Han, 1995] carried out the first known extensions for spatial data.

7.4.3 For a more detailed of Spatial Co-location Patterns, consult [Shekhar and Huang, 2001].

7.5 Our discussion of the K-medoid algorithm are from [Estivill-Castro and Murray, 1998]. The EM algorithm is due to [Dempster et al., 1977] and the extensions for spatial data are due to [Ambroise et al., 1997]. [Ordonez and Cereghini, 2000] present an efficient SQL implementation of the EM algorithm to perform clustering in very large databases. For spatial clustering based on wavelet transforms, consult [Sheikholeslami et al., 1998]. [Wang et al., 1997] discuss a hierarchical statistical information grid based approach for clustering and region oriented queries.

CityID	Facilities
1	a,b,e
2	b,c,d
3	c,e
4	d,c
5	d,e
6	a,c,e
7	a,b,c,e
8	a,b,c,d,e

Table 7.7: Database of facilities

Exercises

1. Consider the following database(7.7) about entertainment facilities in different cities.

   (a) Compute the support for item sets $\{a, b\}$, $\{c\}$ and $\{a, b, c\}$.

   (b) Compute the confidence for the association rules $\{a, b\} \rightarrow \{c\}$.

   (c) Compute the confidence for the association rules $\{c\} \rightarrow \{a, b\}$.

   (d) Why is the confidence not symmetric but support is?

   (e) Extract spatial association rules with a support more than 30 and a confidence more than 70 percent from the following table. X represents lakes in the database (the total number of lakes is 100). For each rule, write the support and the confidence.

spatial predicate	count
near(X,forest)	45
inside(X,state_park)	90
adjacent(X,federal_land)	50
near(X,forest) and inside(X,state_park)	30
near(X,forest) and adjacent(X,federal_land)	20
near(X,forest) and inside(X,state_park) and adjacent(X,federal_land)	10

Rule	**support**	**confidence**
lake(X) ⇒ near(forest)		
lake(X) and inside(X, state_park) ⇒ near(X, forest_land)		
lake(X) and inside(X, state_park) ⇒ adjacent(X,federal_land)		
lake(X) and inside(X, state_park) and adjacent(X,federal_land) ⇒ near(forest)		

   (f) In the computation of $J(M_{t+1}) - J(M_t)$ in the *K-medoid* algorithm, why do only the nonmedoid points of the m_o(medoid-old) and m_n(medoid-new) in $M_{t+1} = M_t \cup \{m_n\} - \{m_o\}$ have to fetched in the main memory?

Hint: All the nonmedoid points satisfy one of the four following cases:

i. $p \ni C_{m_o} \wedge \exists m \in M_t$ such that$d(p, m) < d(p, m_n) \Rightarrow p \in C(m) inM_{t+1}$.
ii. $p \ni C_{m_o} \wedge \forall m \in M_t, d(p, m) < d(p, m_n) \Rightarrow p \in C(m_n) inM_{t+1}$.
iii. $p \ni C_{m_o} \wedge \exists m_1 \in M_t$such that$d(p, m_1) < d(p, m_n) \Rightarrow p \in C(m_1) inM_{t+1}$.
iv. $p \ni C_{m_o} \wedge \exists m_1 \in M_t$such that $d(p, m_1) > d(p, m_n) \Rightarrow p \in C(m_n) inM_{t+1}$.

(g) Assume all the clusters have the same size. What is the performance gain due to the above approach?

2. Consider a dataset with N features and T transactions. How many distinct associations can be enumerated, and how may distinct association rules can be found?

3. Which data mining technique would you use for following scenarios:

(a) An astronomer wants to determine if an unknown object in the sky is a special kind of galaxy(i.e. bent-double galaxy).
(b) A meteorologist wants to predict the weather(temperature and precipitation) for the Thanksgiving weekend.
(c) A urban planner who is designing a shopping mall wants to determine which categories of stores tend to be visited together.
(d) A political analyst wants to group cities according to their voting history in last twenty years.
(e) In order to plan to police patrols the public safety department wants to identify hot-spots on a city map.
(f) Epidemiologists want to predict the spread and movement of the Blue Nile virus.
(g) Doctors want to determine if spatial location has an affect on the cancer-rate.
(h) Natural resource planners want to assess the total area of Pine forest stands using remotely sensed images.

4. Compare and contrast:

(a) association rules vs. statistical correlation.

(b) auto-correlation vs. cross-correlation.

(c) classification vs. location prediction.

(d) Hot-Spots vs. Clusters.

5. Consider the following set of nine points: $(0, 0), (0, 1), (1, 1), (1, 0), (2, 3)(5, 5), (5, 6), (6, 6), (6, 5)$.

(a) Assuming all the points belong to a single cluster. Calculate the mean and the medoid of the cluster.

(b) Compare the mean and the medoid as the most representative point of the cluster. Use average distance from the representative point to all points in the cluster as a comparison metric.

(c) Consider the scenario when the first four points are in one cluster and the last four are in the second cluster. Compute means as representative points for these clusters. Which cluster should the remaining point ($(2,3)$) be assigned to?

6. What is special about spatial data mining relative to mining relational data? Is it adequate to materialize spatial features to be used as input to classical data mining algorithms/models?

7. What is special about spatial statistics relative to statistics?

8. Which of the following spatial features show positive spatial auto correlation? Why? (Is there a physical/scientific reason?)

   Elevation slope, water content, temperature, soil type, population density, annual precipitation (rain, snow).

9. Classify the following spatial point functions into classes of positive spatial autocorrelation, no spatial autocorrelation, and negative spatial autocorrelation:

   (a) $f(x,y) = 1$.

   (b)

$$f(x,y) = \begin{cases} 1, & \text{if } |x+y| \text{ is even,} \\ 0, & \text{otherwise.} \end{cases}$$

   (c) $f(x,y) = (x-x_0)^2 + (y-y_0)^2$.

   (d) $f(x,y)$ is a random number from $[0,1]$.

10. Discuss the following assertion from an expert on marketing data analysis about mining numeric data sets: ”The only data mining techniques one needs is linear regression, if features are selected carefully.”

11. Compute Moran’s I for the gray-scale image shown in Figure 7.14a.