22

Mining Geospatial Data

Shashi Shekhar and Ranga Raju Vatsavai
University of Minnesota

This work was supported in part by the Army High-Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement DAAD19-01-2-0014, the content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred.

INTRODUCTION

Explosive growth in geospatial data and the emergence of new spatial technologies emphasize the need for the automated discovery of spatial knowledge. Spatial data mining is the process of discovering interesting and previously unknown but potentially useful patterns from spatial databases. The complexity of spatial data and intrinsic spatial relationships limits the usefulness of conventional data mining techniques for extracting spatial patterns. In this chapter we explore the emerging field of spatial data mining and applications, especially the areas of outlier detection, colocation rules, classification/prediction, and clustering techniques that explicitly model spatial neighborhood relationships. We present several case studies utilizing a wide variety of spatial data sets.

Spatial data mining is the process of exploring large spatial data sets for extracting interesting, potentially useful, and recurrent patterns and relationships. Widespread use of spatial databases (Guting, 1994; Shekhar & Chawla, 2002; Shekhar et al., 1999; Worboys, 1995) is leading to an increasing interest in mining interesting and useful, but implicit, spatial patterns (Greenman, 2000; Koperski, Adhikary, & Han, 1996; Mark, 1999; Roddick & Spiliopoulou, 1999; Stolorz et al., 1995). Spatial data sets and patterns are abundant in many application domains related to the National Aeronautics and Space Administration, the National Imagery and Mapping Agency, the National Cancer Institute, and the United States Department of Transportation. Efficient tools for extracting information from geospatial data are crucial to organizations that make decisions based on large spatial data sets. These applications are spread across many domains including ecology and environmental management, public safety, transportation, public health, business, and travel and tourism (Albert & McShane, 1995; Haining, 1989; Hohn, Gribki, & Liebhold, 1993; Issaks, Edward, & Srivastava, 1989; Krugman, 1995; Shekhar, Yang, & Hancock, 1993; Stolorz et al., 1995; Yasui & Lele, 1997).

General purpose data mining tools such as Clementine (SPSS), See5/C5.0 (Research), and Enterprise Miner (SAS) are designed for the purpose of analyzing large commercial databases. Although these tools were primarily designed to identify customer-buying patterns in market-basket data, they also have been used in analyzing scientific and engineering, astronomical, multimedia, genomic, and Web data. Extracting interesting and useful patterns from spatial data sets is more difficult than extracting corresponding patterns from traditional numeric and categorical data due to the complexity of spatial data types, spatial relationships, and spatial autocorrelation. Specific features of geographical data that preclude the use of general purpose data mining algorithms are (a) spatial relationships among the variables, (b) spatial structure of errors, (c) mixed distributions as opposed to commonly assumed normal distributions, (d) observations that are not independent, (e) spatial autocorrelation among the features, and (f) nonlinear interaction in feature space. Of course, one can apply conventional data mining algorithms, but it is often observed that these algorithms perform more poorly on spatial data. Many supportive examples can be found in the literature; for instance, parametric classifiers such as maximum likelihood perform more poorly than nonparametric classifiers when the assumptions about the parameters (e.g., normal distribution) are violated, and Markov random fields (MRFs) perform better than per-pixel based classifiers when the features are autocorrelated.

Now the question arises whether we really need to invent new algorithms or extend the existing approaches to explicitly model spatial properties and relationships. Although it is difficult to tell the direction of future research, for now it seems both approaches are gaining momentum. In this chapter we present techniques that are specifically designed for analyzing large volumes of spatial data as well as extensions of conventional data mining techniques that explicitly model spatial concepts.

This chapter is organized as follows. In the next section we introduce spatial outlier detection techniques and their use in finding spatiotemporal outliers in traffic data. In the section after the heading "Spatial Colocation Rules" we present a new approach, called colocation mining, which finds the subsets of features frequently located together in spatial databases. The section after the heading "Location Prediction" presents extensions of classification and prediction techniques that model spatial context. Here we present a case study on a bird nest data set using MRFs and spatial autoregressive regression (SAR) techniques. The section after the heading "Clustering" explores expectation maximization (EM) and its spatial extension, neighborhood expectation maximization (NEM), and provides bird nest prediction results. Finally, this chapter concludes with a summary of techniques and results.

SPATIAL OUTLIER DETECTION TECHNIQUES

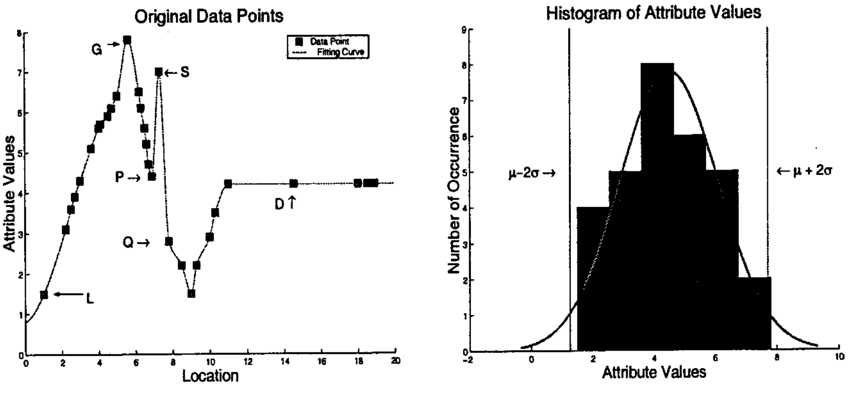
Global outliers have been defined informally as observations in a data set that appear to be inconsistent with the remainder of that set of data (Barnett & Lewis, 1994), or that deviate so much from other observations so as to arouse suspicions that they were generated by a different mechanism (Hawkins, 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, that is, observations that 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 nonspatial. Spatial attributes of a spatially referenced object include location, shape, and other geometric or topological properties. Nonspatial 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, for example, location. Spatial neighborhoods may be defined based on spatial attributes such as location, using spatial relationships such as distance or adjacency. Comparisons between spatially referenced objects are based on nonspatial attributes.

A spatial outlier is a spatially referenced object the nonspatial attribute values of which differ significantly from those of other spatially referenced objects in its spatial neighborhood. Informally, a spatial outlier is a local instability (in values of nonspatial attributes) or a spatially referenced object the nonspatial attributes of which are extreme relative to its neighbors, even though the attributes 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 nonspatial attribute house age.

Illustrative Examples and Application Domains

We use an example to illustrate the differences among global and spatial outlier detection methods. In Fig. 22.1(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



(a) An Example Data Set

(b) Histogram

FIG. 22.1. A data set for outlier detection.

spatial location of each data point and fit the distribution model to the values of the nonspatial attribute. The outlier detected using this approach is the data point G, which has an extremely high attribute value of 7.9, exceeding the threshold of $\mu + 2\sigma = 4.49 + 2 * 1.61 = 7.71$, as shown in Fig. 22.1(b). This test assumes a normal distribution for attribute values. On the other hand, S is a spatial outlier with an observed value significantly different from its neighbors P and Q.

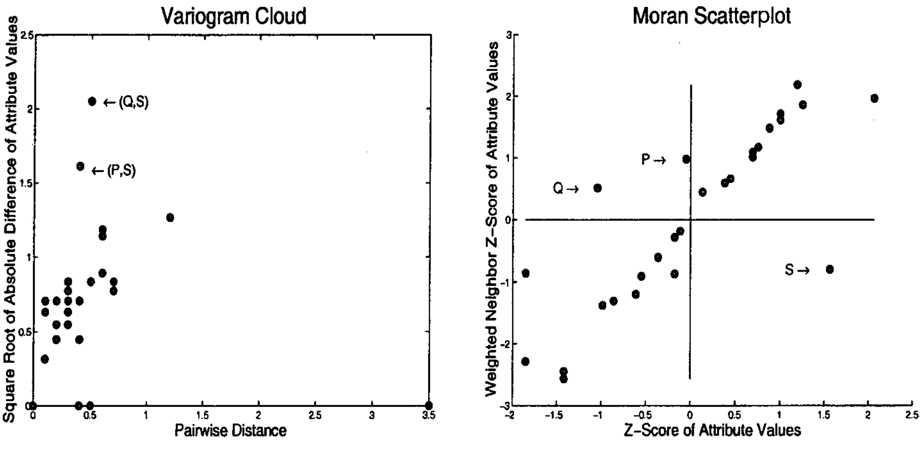
As another example we use a spatial database consisting of measurements from the Minneapolis–St. Paul freeway traffic sensor network. The sensor network includes about 900 stations, each of which contains one to four loop detectors, depending on the number of lanes. Sensors embedded in the freeways and interstate monitor the occupancy and volume of traffic on the road. At regular intervals this information is sent to the Traffic Management Center for operational purposes, for instance, ramp meter control, as well as for experiments and research on traffic modeling.

In this application we are interested in discovering the location of stations with measurements that are inconsistent with those of their spatial neighbors and the time periods when those abnormalities arise. The outlier detection tasks are to (a) build a statistical model for a spatial data set; (b) check whether a specific station is an outlier; and (c) check whether stations on a route are outliers.

Tests for Detecting Spatial Outliers

Tests to detect spatial outliers separate spatial attributes from nonspatial attributes. Spatial attributes are used to characterize location, neighborhood, and distance. Nonspatial attribute dimensions are used to compare a spatially referenced object to its neighbors. Spatial statistics literature provides two kinds of bipartite multidimensional tests, namely graphical tests and quantitative tests. Graphical tests, which are based on the visualization of spatial data, 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



(a) Variogram cloud

(b) Moran scatterplot

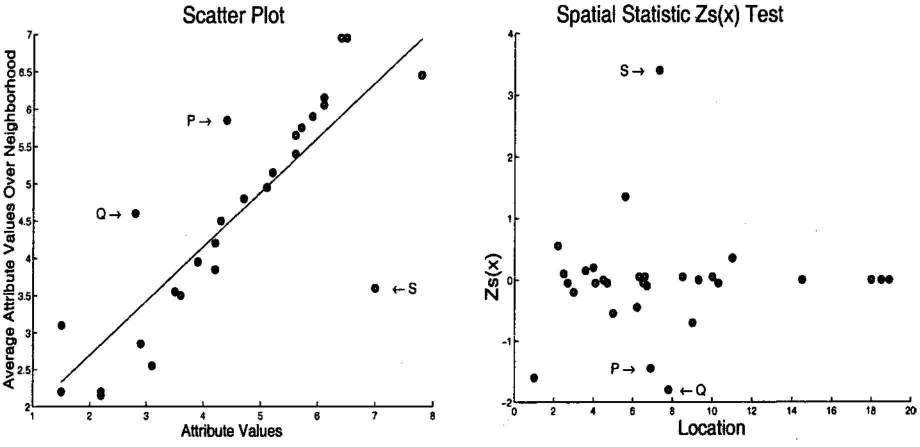
FIG. 22.2. Variogram cloud and Moran scatterplot to detect spatial outliers.

versus the Euclidean distance between the locations is plotted. In data sets exhibiting strong spatial dependence the variance in the attribute differences 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 nonspatially. Figure 22.2(a) shows a variogram cloud for the example data set shown in Fig. 22.1(a). This plot shows that two pairs (P, S) and (Q, S) on 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 because 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 nontrivial postprocessing of highlighted pairs to separate spatial outliers from their neighbors, particularly when multiple outliers are present or density varies greatly.

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 Fig. 22.2(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. A $Moran_{outlier}$ is a point located in the upper left and lower right quadrants of a 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



(a) Scatterplot

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

FIG. 22.3. Scatterplot and spatial statistic $Z_{S(x)}$ to detect spatial outliers.

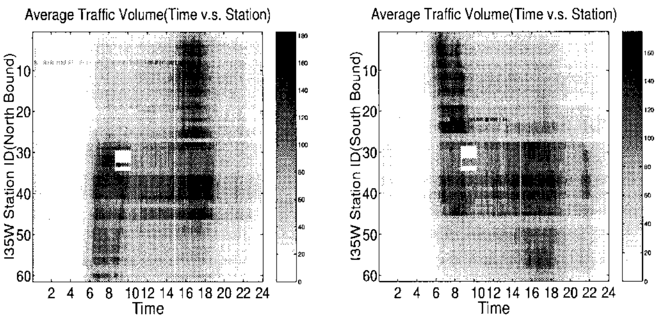
less than -3.0 are flagged as possible spatial outliers, where μ_ϵ and σ_ϵ are the mean and standard deviation of the distribution of the error term ϵ. In Fig. 22.3(a), a scatterplot shows the attribute values plotted against the average of the attribute values in neighboring areas for the data set in Fig. 22.1(a). The point S turns out to be the farthest from the regression line and may be identified as a spatial outlier.

Definition. A $Scatterplot_{outlier}$ is a point with significant standardized residual error from the least square regression line in a scatterplot. 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 (sensor) is compared with 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 statistical 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% will lead to $\theta \approx 2$.

Figure 22.3(b) shows the visualization of the spatial statistic $Z_{s(x)}$ method described earlier and in 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 point S has a $Z_{s(x)}$ value exceeding 3 and will be detected as a spatial outlier. Note that the two neighboring points P and Q of S have $Z_{s(x)}$ values close to -2 due to the presence of spatial outliers in their neighborhoods. Example 1 has already shown that $Z_{s(x)}$ is a special case of an S-outlier.



(a) I-35W North Bound (b) I-35W South Bound

FIG. 22.4. Spatial outliers in traffic volume data.

Solution Procedures

Given the components of the S-outlier definition, the objective is to design a computationally efficient algorithm to detect S-outliers. We presented scalable algorithms for spatial outlier detetection in [Shekhar, Lu, & Zhang (2001)], where we showed that almost all statistical tests are "algebraic" aggregate functions over a neighborhood join. The spatial outlier detection algorithm has two distinct tasks: The first deals with model building and the second involves a comparison (test statistic) with spatial neighbors. During model building, algebraic aggregate functions (e.g., mean and standard deviation) are computed in a single scan of a spatial-join using a neighbor relationship. In the second step a neighborhood aggregate function is computed by retrieving the neighboring nodes, and then a difference function is applied over the neighborhood aggregates and algebraic aggregates. This study showed that the computational cost of outlier detection algorithms are dominated by the disk page access time (i.e., the time spent on accessing neighbors of each point). In this study we utilized three different data page clustering schemes: the connectivity-clustered access method (CCAM; Shekhar & Liu, 1997), Z-ordering (Orenstein & Merrett, 1984), and cell-tree (Gunther, 1989) and found that CCAM produced the lowest number of data page accesses for outlier detection.

The effectiveness of the $Z_{s(x)}$ method on a Minneapolis–St. Paul traffic data set is illustrated in the following example. Figure 22.4 shows one example of traffic flow outliers. Figures 22.4(a) and (b) are the traffic volume maps for I-35W north bound and south bound, respectively, on January 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 2:45 p.m. and the white rectangle from 8:20 a.m. to 10:00 a.m. 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 2:45 p.m. is an instance of temporal outliers, whereas the white rectangle is a spatial–temporal outlier. Both represent missing data. Moreover, station 9 in Fig. 22.4(a) exhibits inconsistent traffic flow compared with its neighboring stations and was detected as a spatial outlier. Station 9 may be a malfunctioning sensor.

SPATIAL COLOCATION RULES

Association rule finding (Hipp, Guntzer, & Nakaeizadeh, 2000) is an important data mining technique that has helped retailers interested in finding items frequently bought together to make store arrangements, plan catalogs, and promote products together. In market-basket data a transaction consists of a collection of item types purchased together by a customer. Association rule mining algorithms (Agrawal & Srikant, 1994) assume that a finite set of disjoint transactions are given as input to the algorithms. Algorithms such as the a priori (Agrawal & Srikant, 1994) can efficiently find the frequent item sets from all the transactions, and association rules can be found from these frequent item sets. Many spatial data sets consist of instances of a collection of Boolean spatial features (e.g., drought, needle leaf vegetation). Although Boolean spatial features can be thought of as item types, there may not be an explicit finite set of transactions due to the continuity of underlying spaces. In this section we define colocation rules, a generalization of association rules to spatial data sets.

Illustrative Application Domains

Many ecological data sets (Li, Cihlar, Moreau, Huang, & Lee, 1997; Nepstad et al., 1999) consist of raster maps of the Earth at different times. Measurement values for a number of variables (e.g., temperature, pressure, and precipitation) are collected for different locations on Earth. Maps of these variables are available for different time periods ranging from 20 to 100 years. Some variables are measured using sensors, whereas others are computed using model predictions.

A set of events, that is, Boolean spatial features, are defined on these spatial variables. Example events include drought, flood, fire, and smoke. Ecologists are interested in a variety of spatiotemporal patterns including colocation rules. Colocation patterns represent frequent cooccurrences of a subset of Boolean spatial features. Examples of interesting colocation patterns in ecology are shown in Table 22.1. Net primary production (NPP) is a key variable for understanding the global carbon cycle and the ecological dynamics of the Earth.

The spatial patterns of ecosystem data sets include the following:

1. Local colocation patterns that represent relationships among events in the same grid cell, ignoring the temporal aspects of the data. Examples from the ecosystem domain include patterns P1 and P2 of Table 22.1. These patterns can be discovered using algorithms (Agrawal & Srikant, 1994) for mining classical association rules.
2. Spatial colocation patterns that represent relationships among events happening in different and possibly nearby grid cells. Examples from the ecosystem domain include patterns P3 and P4 of Table 22.1.

Additional varieties of colocation patterns may exist. Furthermore, the temporal nature of ecosystem data gives rise to many other time related patterns. We focus on these colocation patterns in the following sections.

TABLE 22.1
Examples of Interesting Spatiotemporal Ecological Patterns

Pattern No.	*Variable A*	*Variable B*	*Examples of Interesting Patterns*
P1	Cropland area	Vegetation	Higher cropland area alters NPP
P2	Precipitation drought index	Vegetation	Low rainfall events lead to lower NPP
P3	Smoke aerosol index	Precipitation	Smoke aerosols alter the likelihood of rainfall in a nearby region
P4	Sea surface temperature	Land surface climate and NPP	Surface ocean heating affects regional terrestrial climate and NPP

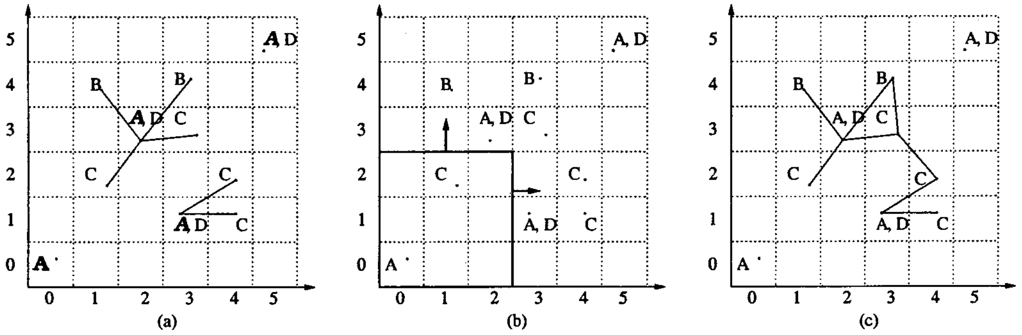


FIG. 22.5. Spatial data set to illustrate different colocation models.

Colocation Rule Approaches

Given the difficulty in creating explicit disjoint transactions from continuous spatial data, this section defines several approaches to model colocation rules. We use Fig. 22.5 as an example spatial data set to illustrate the different models. In this figure a uniform grid is imposed on the underlying spatial framework. For each grid l, its neighbors are defined to be the nine adjacent grids (including l). Spatial feature types are labeled beside their instances. Figure 22.5(a) shows a reference feature-centric model. The instances of A are connected with their neighboring instances of B and C by edges. Figure 22.5(b) shows a window-centric model. Each 3×3 window corresponds to a transaction. Figure 22.5(c) shows an event-centric model. Neighboring instances are joined by edges. We define the following basic concepts to facilitate the description of different models.

Definition 1. *A colocation is a subset of boolean spatial features or spatial events.*

Definition 2. *A colocation rule is of the form $C_1 \rightarrow C_2$ (p, cp) where C_1 and C_2 are colocations, p is a number representing the prevalence measure, and cp is a number measuring conditional probability.*

The prevalence measure and the conditional probability measure are called interest measures and are defined differently in different models.

The reference feature-centric model is relevant to application domains focusing on a specific Boolean spatial feature, for instance, cancer. Domain scientists are interested in finding the colocations of other task-relevant features (e.g., asbestos, other substances) to the reference feature. This model enumerates neighborhoods to "materialize" a set of transactions around instances of the reference spatial feature. A specific example is provided by the spatial association rule (Koperski & Han, 1995).

For example, in Fig. 22.5(a), let the reference feature be A, the set of task-relevant features be B and C, and the set of spatial predicates include one predicate named "*close_to*." Let us define *close_to* (a, b) to be true if and only if b is a's neighbor. Then for each instance of spatial feature A, a transaction that is a subset of relevant features $\{B, C\}$ is defined. For example, for the instance of A at (2,3), transaction $\{B, C\}$ is defined because the instance of B at (1,4) (and at [3,4]) and instance of C at (1,2) (and at [3,3]) are *close_to* (2,3). The transactions defined around instances of feature A are summarized in Table 22.2.

With "materialized" transactions the support and confidence of the traditional association rule problem (Agrawal & Srikant, 1994) may be used as prevalence and conditional probability measures as summarized in Table 22.3. Because one out of two nonempty transactions contains instances of both B and C and one out of two nonempty transactions contain C in

TABLE 22.2
Reference Feature-Centric View: Transactions Are Defined Around Instances of Feature A Relevant to B and C in Figure 22.5(a)

Instance of A	*Transaction*
(0,0)	ø
(2,3)	$\{B, C\}$
(3,1)	$\{C\}$
(5,5)	ø

Table 22.2, an association rule example is: $is\_type\,(i, A) \wedge \exists j\ is\_type\,(j, B) \wedge close\_to\,(j, i) \rightarrow \exists k\ is\_type\,(k, C) \wedge close\_to\,(k, i)$ with $\frac{1}{1} * 100\% = 100\%$ probability.

The window-centric model is relevant to applications like mining, surveying, and geology, which focus on land parcels. A goal is to predict sets of spatial features likely to be discovered in a land parcel given that some other features have been found there. The window-centric model enumerates all possible windows as transactions. In a space discretized by a uniform grid, windows of size $k \times k$ can be enumerated and materialized, ignoring the boundary effect. Each transaction contains a subset of spatial features of which at least one instance occurs in the corresponding window. The support and confidence of the traditional association rule problem may again be used as prevalence and conditional probability measures as summarized in Table 22.3. There are 16 3×3 windows corresponding to 16 transactions in Figure 22.5(b). All of them contain A, and 15 of them contain both A and B. An example of an association rule of this model is: *an instance of type A in a window → an instance of type B in this window* with $\frac{15}{16} = 93.75\%$ probability. A special case of the window-centric model relates to the case when windows are spatially disjoint and form a partition of space. This case is relevant when analyzing spatial data sets related to the units of political or administrative boundaries (e.g., country, state, zip code). In some sense this is a local model because we treat each arbitrary partition as a transaction to derive colocation patterns without considering any patterns cross partition boundaries. The window-centric model "materializes" transactions in a different way from the reference feature-centric model.

The event-centric model is relevant to applications such as ecology in which there are many types of Boolean spatial features. Ecologists are interested in finding subsets of spatial

TABLE 22.3
Interest Measures for Different Models

			Interest Measures for $C_1 \rightarrow C_2$	
Model	*Items*	*Transactions Defined By*	*Prevalence*	*Conditional Probability*
Local	Boolean feature types	Partitions of space	Fraction of partitions with $C_1 \cup C_2$	$Pr(C_2$ in a partition given C_1 in the partition)
Reference feature-centric	Predicates on reference and relevant features	Instances of reference feature C_1 and C_2 involved with	Fraction of instance of reference feature with $C_1 \cup C_2$	$Pr(C_2$ is true for an instance of reference features given C_1 is true for that instance of reference feature)
Window-centric	Boolean feature types	Possibly infinite set of distinct overlapping windows	Fraction of windows with $C_1 \cup C_2$	$Pr(C_2$ in a window given C_1 in that window)
Event centric	Boolean feature types	Neighborhoods of instances of feature types	Participation index of $C_1 \cup C_2$	$Pr(C_2$ in a neighborhood of $C_1)$

features likely to occur in a neighborhood around instances of given subsets of event types. For example, let us determine the probability of finding at least one instance of feature type B in the neighborhood of an instance of feature type A in Figure 22.5(c). There are four instances of type A and only one of them has some instance(s) of type B in its nine-neighbor adjacent neighborhoods. The conditional probability for the colocation rule is: *spatial feature A at location l → spatial feature type B in 9-neighbor neighborhood is 25%.*

Neighborhood is an important concept in the event-centric model. Given a reflexive and symmetric neighbor relation R, we can define neighborhoods of a location l that satisfy the definition of neighborhood in topology [Worboys, 1995] as seen in the following definitions:

Definition 3. *A neighborhood of l is a set of locations $L = \{l_1, \ldots, l_k\}$ such that l_i is a neighbor of l, that is, $(l, l_i) \in R (\forall i \in 1, \ldots, k)$.*

We generalize the neighborhood definition to a collection of locations.

Definition 4. *For a subset of locations L' if L' is a neighborhood of every location in $L = \{l_1, \ldots, l_k\}$ then L' is a neighborhood of L.*

In other words, if every l_1 in L' is a neighbor of every l_2 in L, then L' is a neighborhood of L.

The definition of neighbor relation R is an input and is based on the semantics of application domains. The neighbor relation R may be defined using topological relationships (e.g., connected, adjacent), metric relationships (e.g., Euclidean distance) or a combination (e.g., shortest path distance in a graph such as a road map). In general, there are infinite neighborhoods over continuous space, and it may not be possible to materialize all of them. But we are interested in only the locations where instances of spatial feature types (events) occur. Even confined to these locations, enumerating all the neighborhoods incurs substantial computational cost because support based pruning cannot be carried out before the enumeration of all the neighborhoods is completed and the total number of neighborhoods is obtained. Furthermore, this support based prevalence measure definition may not be meaningful because the value of the prevalence may be extremely small due to the fact that many neighborhoods are contained in bigger neighborhoods and counted multiple times. Thus, the participation index is proposed to be a prevalence measure as defined below.

Definition 5. *For a colocation $C = \{f_1, \ldots, f_k\}$ and a set of locations $I = \{i_1, \ldots, i_k\}$ where i_j is an instance of feature $f_j (\forall j \in 1, \ldots, k)$ if I is a neighborhood of I itself then I is an instance of C.*

In other words, if elements of I are neighbors to each other, then I is an instance of C. For example, {(3,1), (4,1)} is an instance of colocation $\{A, C\}$ in Fig. 22.5(c) using a nine-neighbor adjacent neighbor definition.

Definition 6. *The participation ratio $pr(C, f_i)$ for feature type f_i of a colocation $C = \{f_1, f_2, \ldots, f_k\}$ is the fraction of instances of f_i that participate in the colocation C. It can be formally defined as $\frac{|distinct(\pi_{f_i}(all\ instances\ of\ colocation\ C))|}{|instances\ of\ \{f_i\}|}$ where π is a projection operation.*

For example, in Fig. 22.5(c), instances of colocation $\{A, B\}$ are {(2,3), (1,4)} and {(2,3), (3,4)}. Only one instance (2,3) of spatial feature A out of four participates in colocation $\{A, B\}$. So $pr(\{A, B\}, A) = \frac{1}{4} = .25$.

Definition 7. *The participation index of a colocation $C = \{f_1, f_2, \ldots, f_k\}$ is $\prod_{i=1}^{k} pr(C, f_i)$.*

In Fig. 22.5(c), participation ratio pr$(\{A, B\}, A)$ of feature A in colocation $\{A, B\}$ is .25 as calculated above. Similarly $pr(\{A, B\}, B)$ is 1.0. The participation index for colocation $\{A, B\}$ is $.25 \times 1.0 = .25$.

The conditional probability of a colocation rule $C_1 \rightarrow C_2$ in the event-centric model is the probability of finding C_2 in a neighborhood of C_1 or it can be formally defined as follows.

Definition 8. *The conditional probability of a colocation rule $C_1 \rightarrow C_2$ is $\frac{|distinct(\pi_{C_1}(\text{all instances of colocation } C_1 \cup C_2))|}{|\text{instances of } C_1|}$ where π is a projection operation.*

For details of algorithms that mine colocation rules in the event centric model, refer to Shekhar and Huang (2001).

Solution Procedures

Colocation mining is a complex task. It consists of two tasks, schema level pruning and instance level pruning. At schema level pruning the apriori algorithm (Agrawal & Srikant, 1994) can be used. However, instance level pruning involves neighborhood (i.e., colocation row instance) enumeration, which is a computationally intense task. Shekhar and Huang (2001) developed pure geometric, pure combinatorial, hybrid, and multiresolution algorithms for instance level pruning. Experimental analysis shows that the pure geometric algorithm performs much better than the pure combinatorial approach. The hybrid algorithm, which is a combination of geometric and combinatorial methods, performed better than both of these approaches. On the other hand, the multiresolution algorithm outperforms all these methods when the data is "clumped." The results also show that the colocation miner algorithm is complete and correct.

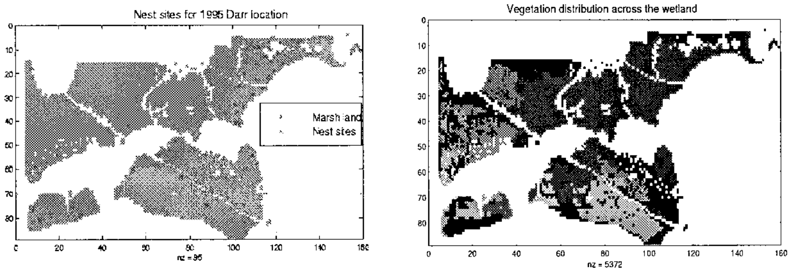
LOCATION PREDICTION

The prediction of events occurring at particular geographic locations is very important in several application domains. Crime analysis, cellular networks, and natural disasters such as fires, floods, droughts, vegetation diseases, and earthquakes are all examples of problems that require location prediction. In this section we provide two spatial data mining techniques, namely the SAR and MRF and analyze their performance in an example case, the prediction of the location of bird nests in the Darr and Stubble wetlands in Ohio.

An Illustrative Application Domain

We now introduce an example to illustrate the different concepts in spatial data mining. We are given data about two wetlands, named Darr and Stubble, on the shores of Lake Erie in Ohio to *predict* the spatial distribution of a marsh-breeding bird, the red-winged blackbird (*Agelaius phoeniceus*). 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. In total, values of seven attributes were recorded at each cell. 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 bird-nesting habits of the red-winged blackbird suggested that the choice of nest location is more dependent on plant structure, plant resistance to wind, and wave action than on the plant species.


Nest sites for 1995 Darr location
Marsh land
Nest sites
nz = 95
Vegetation distribution across the wetland
nz = 5372
0
10
20
30
40
50
60
70
80
0
20
40
60
80
100
120
140
160

(a) Nest Locations

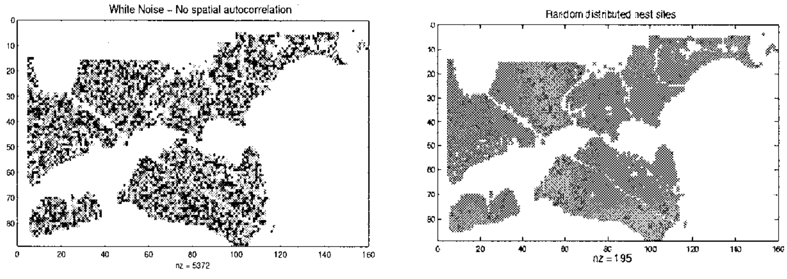
(b) Vegetation Durability

FIG. 22.6. (a) Learning data set: the geometry of the wetland and the locations of the nests; (b) the spatial distribution of vegetation durability over the marshland.

Our goal 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. In the learning data all the attributes are used to build the model, and in the testing data one value is *hidden,* in our case the location of the nests.

We focus on three independent attributes, namely vegetation durability, distance to open water, and water depth. The spatial distribution of vegetation durability and the actual nest locations for the Darr wetland in 1995 are shown in Fig. 22.6. These maps illustrate the following two important properties inherent in spatial data.

1. The values of vegetation durability and the actual nest locations that are referenced by spatial location tend to vary gradually over space (values of distance to open water and water depth are similar). Although this may seem obvious, classical data mining techniques either explicitly or implicitly assume that the data is *independently* generated. For example, the maps in Fig. 22.7 show the spatial distribution of attributes if they were independently generated. Classical data mining techniques such as logistic regression (Ozesmi & Mitch, 1997) and neural networks (Ozesmi & Ozesmi, 1999) were applied to build spatial habitat models. These models predict the bird nesting locations by considering the spatial interactions


White Noise – No spatial autocorrelation
nz = 5372
Random distributed nest sites
nz = 195
0
10
20
30
40
50
60
70
80
0
20
40
60
80
100
120
140
160

(a) pixel property with independent identical distribution

(b) Random nest locations

FIG. 22.7. Spatial distribution satisfying random distribution assumptions of classical regression.

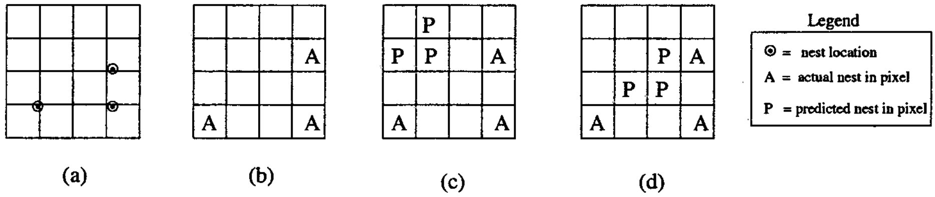


FIG. 22.8. (a) The actual locations of nests; (b) pixels with actual nests; (c) location predicted by a model; (d) location predicted by another model. Prediction (d) is spatially more accurate than (c).

among the dependent variables. Logistic regression was used because the dependent variable is binary (nest/no-nest) and the logistic function "squashes" the real line onto the unit interval. The values in the unit interval can then be interpreted as probabilities. The study concluded that with the use of logistic regression, the nests could be classified at a rate 24% better than random (Ozesmi & Ozesmi, 1999).

2. The spatial distributions of attributes sometimes have distinct local trends that contradict the global trends. This is seen most vividly in Fig. 22.6(b), in which the spatial distribution of vegetation durability is jagged in the western section of the wetland as compared with the overall impression of uniformity across the wetland. This property is called spatial heterogeneity.

The fact that classical data mining techniques ignore spatial autocorrelation and spatial heterogeneity in the model-building process is one reason these techniques do a poor job. A second more subtle but equally important reason is related to the choice of the objective function to measure classification accuracy. For a two-class problem the standard way to measure classification accuracy is to calculate the percentage of correctly classified objects. However, this measure may not be the most suitable in a spatial context. *Spatial accuracy*—how far the predictions are from the actuals—is just as important in this application domain due to the effects of the discretizations of a continuous wetland into discrete pixels, as shown in Fig. 22.8. Figure 22.8(a) shows the actual locations of nests and 22.8(b) shows the pixels with actual nests. Note the loss of information during the discretization of continuous space into pixels. Many nest locations barely fall within the pixels labeled "A" and are quite close to other blank pixels, which represent "no-nest." Now consider two predictions shown in Fig. 22.8(c) and 22.8(d). Domain scientists prefer prediction 22.8(d) over 22.8(c), because the predicted nest locations are closer on average to some actual nest locations. The classification accuracy measure cannot distinguish between 22.8(c) and 22.8(d), and a measure of spatial accuracy is needed to capture this preference.

Problem Formulation

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 $\{s_1, \dots, s_n\}$ for an underlying geographic space G.
- A collection X of explanatory functions $f_{X_k} : S \to R^k, k = 1, \dots, K$. R^k is the range of possible values for the explanatory functions. Let $X = [1, X]$, which also includes a constant vector along with explanatory functions.

- A dependent class variable $f_C : S \to C = \{c_1, \ldots, c_M\}$
- A value for parameter α, relative importance of spatial accuracy

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

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, $C = \{0, 1\}$.

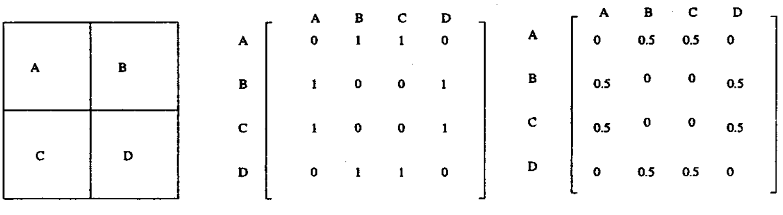
The above formulation highlights two important aspects of location prediction. It explicitly indicates that (a) the data samples may exhibit spatial autocorrelation and, (b) an objective function (i.e., a map similarity measure) is a combination of classification accuracy and spatial accuracy. The *similarity* 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.

Modeling Spatial Dependencies Using the SAR and MRF Models

Several previous studies (Jhung & Swain, 1996; Solberg, Taxt, & Jain, 1996) have shown that the modeling of spatial dependency (often called context) during the classification process improves overall classification accuracy. Spatial context can be defined by the relationships between spatially adjacent pixels in a small neighborhood. The spatial relationship among locations in a spatial framework often is modeled via a contiguity matrix. A simple contiguity matrix may represent a neighborhood relationship defined using adjacency, Euclidean distance, and so forth. Example definitions of neighborhood using adjacency include a four-neighborhood and an eight-neighborhood. Given a gridded spatial framework, a four-neighborhood assumes that a pair of locations influence each other if they share an edge. An eight-neighborhood assumes that a pair of locations influence each other if they share either an edge or a vertex.

Figure 22.9(a) shows a gridded spatial framework with four locations, A, B, C, and D. A binary matrix representation of a four-neighborhood relationship is shown in Fig. 22.9(b). The row-normalized representation of this matrix is called a contiguity matrix, as shown in Fig. 22.9(c). Other contiguity matrices can be designed to model neighborhood relationship based on distance. The essential idea is to specify the pairs of locations that influence each other along with the relative intensity of interaction. More general models of spatial relationships using cliques and hypergraphs are available in the literature (Warrender & Augusteijn, 1999).

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



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

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

(a) Spatial Framework (b) Neighbor relationship (c) Contiguity Matrix

FIG. 22.9. A spatial framework and its four-neighborhood contiguity matrix.

Logistic SAR

Logistic SAR decomposes a classifier $\hat{f}_C$ into two parts, namely spatial autoregression and logistic transformation. We first show how spatial dependencies are modeled using the framework of logistic regression analysis. In the spatial autoregression model the spatial dependencies of the error term, or the dependent variable, are directly modeled in the regression equation (Anselin, 1988). If the dependent values y_i are related to each other, then the regression equation can be modified as

$$y = \rho Wy + X\beta + \epsilon. \tag{1}$$

Here W is the neighborhood relationship contiguity matrix and ρ is a parameter that reflects the strength of the spatial dependencies between the elements of the dependent variable. After the correction term ρWy is introduced, the components of the residual error vector ϵ are then assumed to be generated from independent and identical standard normal distributions. As in the case of classical regression, the SAR equation has to be transformed via the logistic function for binary dependent variables.

We refer to this equation as the SAR model. Notice that when $\rho = 0$, this equation collapses to the classical regression model. The benefits of modeling spatial autocorrelation are many: The residual error will have much lower spatial autocorrelation (i.e., systematic variation). With the proper choice of W, the residual error should, at least theoretically, have no systematic variation. If the spatial autocorrelation coefficient is statistically significant, then SAR will quantify the presence of spatial autocorrelation. It will indicate the extent to which variations in the dependent variable (y) are explained by the average of neighboring observation values. Finally, the model will have a better fit (i.e., a higher R-squared statistic). We compare SAR with linear regression for predicting nest location in a later section.

A mixed model extends the general linear model by allowing a more flexible specification of the covariance matrix of ϵ. The SAR model can be extended to a mixed model that allows for explanatory variables from neighboring observations (LeSage & Pace, 2001). The new model (MSAR) is given by

$$y = \rho Wy + X\beta + WX\gamma + \epsilon. \tag{2}$$

The marginal impact of the explanatory variables from the neighboring observations on the dependent variable y can be encoded as a $k * 1$ parameter vector γ.

Solution Procedures

The estimates of ρ and β can be derived using maximum likelihood theory or Bayesian statistics. We have carried out preliminary experiments using the spatial econometrics Matlab package,[2] which implements a Bayesian approach using sampling-based Markov chain Monte Carlo methods (LeSage, 1997). Without any optimization, likelihood-based estimation would require $O(n^3)$ operations. Recently, Pace and Barry (1997a, 1997b) and LeSage and Pace (2001) proposed several efficient techniques to solve SAR. The techniques studied include divide and conquer, and sparse matrix algorithms. Improved performance is obtained by using LU decompositions to compute the log-determinant over a grid of values for the parameter ρ by restricting it to [0, 1].

MRF Based Bayesian Classifiers

MRF based Bayesian classifiers estimate classification model $\hat{f}_C$ using MRF and Bayes' rule. A set of random variables the interdependency relationship of which is represented by an undirected graph (i.e., a symmetric neighborhood matrix) is called an MRF (Li, 1995). 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, constitutes 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:

$$Pr(l_i \mid X, L_i) = \frac{Pr(X \mid l_i, L_i) Pr(l_i \mid L_i)}{Pr(X)} \tag{3}$$

The solution procedure can estimate $Pr(l_i \mid L_i)$ 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. $Pr(X \mid l_i, L_i)$ can be estimated using kernel functions from the observed values in the training data set. For reliable estimates even larger training data sets are needed relative to those needed for the Bayesian classifiers without spatial context, because we are estimating a more complex distribution. An assumption on $Pr(X \mid l_i, L_i)$ may be useful if the training data set 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 (Besag, 1974).

Solution Procedures

Solution procedures for the MRF Bayesian classifier include stochastic relaxation (Geman & Geman, 1984), iterated conditional modes (Besag, 1986), dynamic programming (Derin & Elliott, 1987), highest confidence first (Chou, Cooper, Swain, Brown, & Wixson, 1993) and graph cut (Boykov, Veksler, & Zabih, 1999). We follow the approach suggested in Boykov et al. (1999), where it is shown that the maximum a posteriori estimate of a particular configuration of an MRF can be obtained by solving a suitable min-cut multiway graph partitioning problem. We illustrate the underlying concept with some examples.

[2] We would like to thank James Lesage (http://www.spatial-econometrics.com/) for making the Matlab toolbox available on the Web.

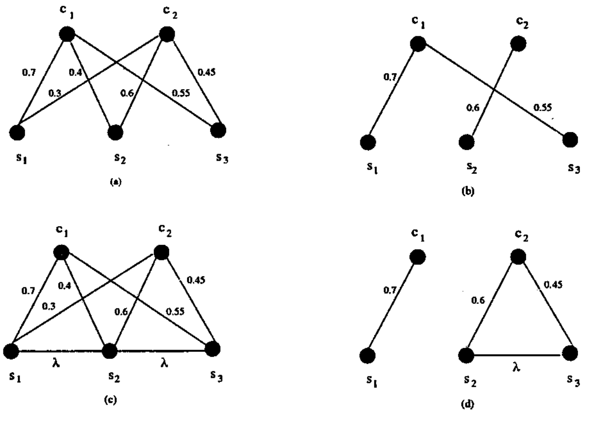


FIG. 22.10. MRF solution with graph-cut method.

Example 1. A classification problem with no spatial constraints. Even though MRFs are inherently multidimensional, we will use a simple one-dimensional example to illustrate the main points. Consider the graph $G = (V, E)$ shown in Fig. 22.10(a). The node set V itself consists of two disjoint sets, S and C. The members of S are $\{s_1, s_2, s_3\}$, and the members of C are $\{c_1, c_2\}$. Typically, the $X(s_i)$'s are the feature values at site s_i and the $c_i's$ are the labels such as nest or no-nest. There is an edge between each pair of members of the set S and each pair of members of set C. Here we interpret the edge weights as probabilities. For example, $p_1 = Pr(X(s_1) = c_1) = 0.7$ and $p_2 = Pr(X(s_1) = c_2) = 0.3$; $p_1 + p_2 = 1$.

Our goal is to provide a label for each location s_i in S using explanatory feature $X(s_i)$. This is done by partitioning the graph into two disjoint sets (not S and C) by removing certain edges such that:

1. There is a many-to-one mapping from the set S to C. Every element of S must be mapped to one and only one element of C.
2. Multiple elements of C cannot belong to a single partition. Thus, there are no edges between elements of C, and therefore, the number of partitions is equal to the cardinality of C.
3. The sum of the weights of the edges removed (the cut-set) is the minimum of all possible cut-sets.

In this example the cut-set is easily determined. For example, of the two edges connecting each element of S and an element of C, remove the edge with the smaller weight. Figure 22.10(b) shows the graph with the cut-set removed. Thus, we have just shown that when the weights of the edges are interpreted as probabilities, the min-cut graph partition induces a maximum a posterior estimate for the pixel labels. We prefer to say that the min-cut induces a Bayesian classification on the underlying pixel set. This is because we will use Bayes' theorem to calculate the edge weights of the graphs.

Example 2. Adding spatial constraints. In the previous example we did not use any information about the spatial proximity of the pixels relative to each other. We do that now by introducing additional edges in the graph structure.

Consider the graph shown in Fig. 22.10(c) in which we have added two extra edges (s_1, s_2) and (s_2, s_3) with a weight λ. In this example we have chosen $\lambda = 0.2$.

Now if we want to retain the same partitions of the graph as in example 1, then the cut-set has two extra edges, namely (s_1, s_2) and (s_2, s_3). Thus, the sum of the weights of the edges in the cut-set, W_{C1}, is

$$W_{C1} = 0.3 + 0.4 + 0.45 + 2\lambda$$

But now, depending on λ, the cut-set weight may not be minimal. For example, if $\lambda = 0.2$ then the weight of the cut-set, W_{C2}, consisting of the edges $\{(s_1, c_2), (s_2, c_1), (s_3, c_1), (s_1, s_2)\}$ is

$$W_{C2} = 0.3 + 0.4 + 0.55 + 0.2$$

Thus, $W_{C2} < W_{C1}$. In other words, if two neighboring pixels are assigned to different labels, then the edge between the two neighbors is added to the cut-set. This shows that there is a penalty associated with two neighboring nodes being assigned to different labels every time. Thus, we can model spatial autocorrelation by adding edges between the pixel nodes of the graph. We can also model spatial heterogeneity by assigning different *weights,* the λ's, to the pixel edges. Figure 22.10(d) shows that a min-cut graph partitioning does not necessarily induce a labeling where the labeling with maximum probabilities is retained. If two neighboring pixels are assigned different labels, then the edge connecting the pixels is added to the cut-set.

Here we briefly provide theoretical and experimental comparisons; more details can be found in Shekhar, Schrater, Vatsavai, Wu, and Chawla (2002). Although MRF and SAR classification have different formulations, they share a common goal, estimating the posterior probability distribution: $p(l_i \mid X)$. 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. One important difference between logistic regression and MRF is that logistic regression assumes no dependence on neighboring classes. Logistic regression and logistic SAR models belong to a more general exponential family. The exponential family is given by $Pr(u \mid v) = e^{A(\theta_v)+B(u,\pi)+\theta_v^T u}$, 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. Experiments were carried out on the Darr and Stubble wetlands to compare the classical regression, SAR, and the MRF-based Bayesian classifiers. The results showed that the MRF models yield better spatial and classification accuracies over SAR in the prediction of the locations of bird nets. We also observed that SAR predications are extremely localized, missing actual nests over a large part of the marsh lands.

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 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 that 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 also used 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 data sets that 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 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 nonuniformly distributed points. The search for potential clusters 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 four attributes: rate-of-unemployment, population, per-capita-income, and life-expectancy. Counties with 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 Fig. 22.11, 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 that 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 Fig. 22.11. On the other hand, if our goal is to identify

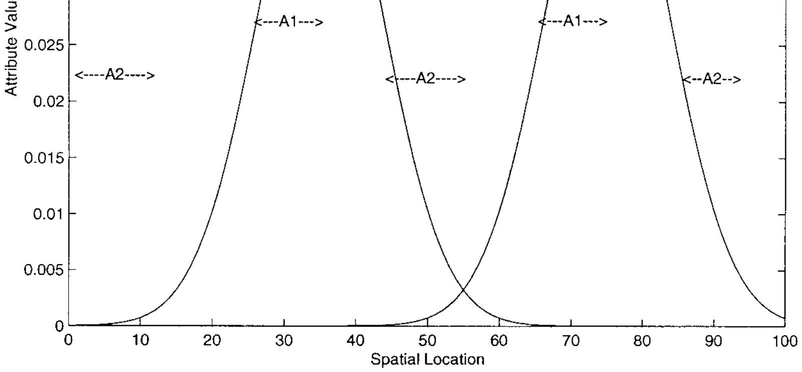


FIG. 22.11. Two interpretations of spatial clustering. If the goal is to identify locations that dominate the surroundings (in terms of influence), then the clusters are S1 and S2. If the goal is to identify areas of homogeneous values, the clusters are A1 and A2.

pockets in the landscape where an attribute (or attributes) is homogeneously expressed, then we are looking for clusters marked A1 and A2. Although 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 follows:

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.

The constraints are as follows:

1. The data set 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 K subsets $C_k \subset S$, $k = 1, \ldots, K$ such that $\min_{C_k \subset S} \sum_{C_k, s_i \in C_k, s_j \in C_k} dist(F(s_i), F(s_j)) + \sum_{i,j} nbddist(C_i, C_j)$ where (a) F is the cross-product of the $f_i's$, $i = 1, \ldots, n$; (b) $dist(a, b)$ is the Euclidean or some other distance measure, and (c) $nbddist(C, D)$ is the number of points in C and D that 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 techniques in many fields. Because this has resulted in a multitude of clustering algorithms, it is useful to categorize them into groups. Based on the technique adopted to define clusters, the clustering algorithms can be divided into four broad categories:

1. *Hierarchical* clustering methods, which start with all patterns as a single cluster and successively perform 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. The hierarchical clustering algorithms include balanced iterative reducing and clustering using hierarchies (BIRCH), clustering using representatives (CURE), and robust clustering using links (ROCK).

2. *Partitional* clustering algorithms, which start with each pattern as a single cluster and iteratively reallocate data points to each cluster until a stopping 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. The recent algorithms in this category include partitioning around medoids, clustering large applications, clustering large applications based on randomized search, and EM.

3. *Density based* clustering algorithms, which try to find clusters based on the density of data points in a region. These algorithms treat clusters as dense regions of objects in the data space. The density based clustering algorithms include density based spatial clustering of applications with noise and density based clustering.

4. *Grid based* clustering algorithms, which 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 a 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. The grid based clustering algorithms include the statistical information grid based method (STING), STING+, WaveCluster, BANG-clustering, and clustering-in-quest (CLIQUE).

Sometimes the distinction among these categories diminishes, and some algorithms can even be classified into more than one group. For example, CLIQUE can be considered as both a density based and grid based clustering method. More details on various clustering methods can be found in a recent survey paper (Halkidi, Batistakis, & Vazirgiannis, 2001).

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.

K-Medoid: An Algorithm for Clustering

We restrict our attention to points in the two-dimensional space R^2, although the technique can be readily generalized to a higher dimensional space. Given a set P of n data points, $P = \{p_1, p_2, \dots, p_n\}$ 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:

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

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 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. Because 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.

TABLE 22.4
Four Options for Local Search in Clustering

Local Search	*Strategy to Move from M_t to $M_{t+1} = M_t \cup \{p\} - \{m\}$*	*Guarantee 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

Step 2 is the heart of the algorithm. There are many options available to move from a node to its adjacent node. Table 22.4 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.

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 in which each distribution generates one cluster. The goal then is to identify the parameters of the probability distributions 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

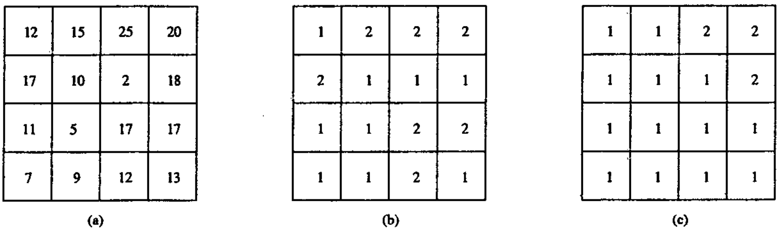
Consider the gray-scale 4 × 4 image shown in Fig. 22.12. Assume we want to partition the set of pixels into two clusters, A and B, where each cluster is modeled as a Gaussian distribution. The finite mixture problem is to calculate the parameters μ_A, μ_B, σ_A, σ_B, p_A, p_B.

For the moment, assume the cluster membership of each pixel is given as shown in Fig. 22.12b. Then all the parameters can be easily calculated. For example,

$$\mu_A = \frac{12+10+2+18+11+5+7+9+13}{9} = 9.7$$

$$\sigma_A = \frac{(12-\mu_1)^2+(10-\mu_1)^2+\cdots+(13-\mu_1)^2}{8} = 4.7$$

$$p_A = \frac{9}{16}$$



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)

FIG. 22.12. (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 NEM algorithm. Notice the spatial smoothing attained by modifying the objective function.

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 a cluster is then a simple exercise using Bayes' theorem. For example, given a pixel value x, the probability that it belongs to cluster A is

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

where

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

In our case neither the cluster labels nor the the distribution parameters are known. 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 EM algorithm. The EM algorithm, like the K-medoid algorithm, is an iterative algorithm that 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. This procedure is iterated until some convergence criterion is met. The EM algorithm guarantees that the maximum likelihood estimate will improve after each iteration, although 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, calculate the probability that the data object x belongs to clusters A and B:

$$P(A \mid x) = \frac{p_A^j P^j(x \mid A)}{P^j(x)} \quad P(B \mid x) = \frac{p_B^j P^j(x \mid B)}{P^j(x)}.$$

3. Update the mixture parameters on the basis of the new estimate:

$$p_A^{j+1} = \tfrac{1}{n}\textstyle\sum_x P(A \mid x) \qquad p_B^{j+1} = \tfrac{1}{n}\textstyle\sum_x P(B \mid x)$$

$$\mu_A^{j+1} = \frac{\sum_x xP(A \mid x)}{\sum_x P(A \mid x)} \qquad \mu_B^{j+1} = \frac{\sum_x xP(B \mid x)}{\sum_x P(B \mid x)}$$

$$\sigma_A^{j+1} = \frac{\sum_x P(A \mid x)(x - \mu_A^{j+1})^2}{\sum_x P(A \mid x)} \qquad \sigma_B^{j+1} = \frac{\sum_x P(B \mid x)(x - \mu_B^{j+1})^2}{\sum_x P(B \mid x)}$$

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 NEM 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 Fig. 22.12(b), the EM algorithm will still come up with the same cluster labeling and the same values of the distribution parameters.[3] 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.

To make the EM algorithm spatially sensitive, we first follow the recipe proposed by Ambroise, Dang, and Govaert (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 $c = c_{ik}, i = 1, \ldots, n$ and $k = 1, \ldots, K$ define a fuzzy classification representing the grade of membership of data point 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.

Step 2. To account for spatial autocorrelation, we introduce a new term,

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

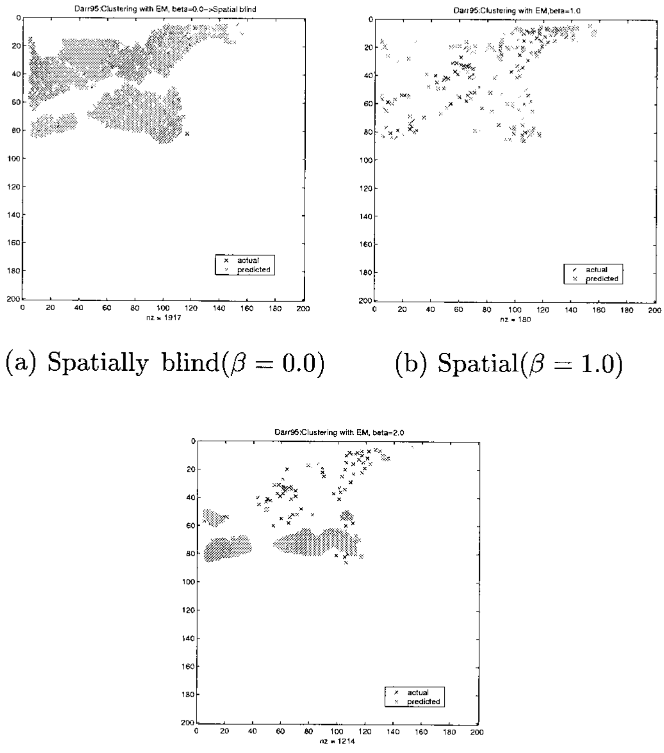
where $W = (w_{ij})$ is the contiguity matrix as defined before.

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 data set.

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



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

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

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

FIG. 22.13. Using the NEM 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 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 $c'_{ik}s$ is

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

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

We carried out experiments using the NEM algorithm on the bird data set. We assumed 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 Fig. 22.13. The results show that including spatial information in the clustering algorithm leads to a dramatic improvement in accuracy (Fig. 22.13(b) compared with Fig. 22.13(a), but overemphasizing spatial information leads to "oversmoothing" and degradation in accuracy.

SUMMARY

In this chapter we presented techniques that are specifically designed to analyze large volumes of spatial data to find spatial outliers, colocation association rules, location prediction, and spatial clustering. We applied these techniques on traffic network sensors, wetlands, and synthetic data sets. We compared the SAR and MRF models using a common probabilistic framework. Our study shows that the SAR model makes more restrictive assumptions about the distribution of features and class shapes (or decision boundaries) than MRF. We also observed an interesting relationship between classical models that do not consider spatial dependence and modern approaches that explicitly model spatial context. The relationship between SAR and MRF is analogous to the relationship between logistic regression and Bayesian classifiers. The analysis of spatial outlier detection algorithms showed the need for good clustering of data pages. The CCAM method yielded the best overall performance. We showed that the colocation miner algorithm is complete and correct and performs better than the well know a priori algorithm.

ACKNOWLEDGMENTS

We would like to thank to our collaborators Prof. Paul Schrater, Dr. Sanjay Chawla, and Prof. Uygar Ozesmi for their various contributions, and group members Yan Huang, Chang-Tien Lu, Weili Wu, and Pusheng Zang for their contributions in developing these techniques and software. The comments of Kim Koffolt have greatly improved the readability of this chapter.

REFERENCES

Agrawal, R., & Srikant, R. (1994). Fast algorithms for Mining Association Rules. In *Proceedings of 20th International Conference on Very Large Databases* (pp. 487–499). San Francisco: Morgan Kaufmann.

Albert, P., & McShane, L. (1995). A generalized estimating equations approach for spatially correlated binary data: Applications to the analysis of neuroimaging data. *Biometrics, 1,* 627–638.

Ambroise, C., Dang, V., & Govaert, G. (1997). Clustering of spatial data by the EM algorithm. *Quantitative Geology and Geostatistics, 9,* 493–504.

Anselin, L. (1988). *Spatial econometrics: methods and models.* Dordrecht, Netherlands: Kluwer.

Barnett, V., & Lewis, T. (1994). *Outliers in statistical data* (3rd ed.). New York: Wiley.

Besag, J. (1974). Spatial interaction and statistical analysis of lattice systems. *Journal of the Royal Statistical Society* (Series B) *36,* 192–236.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, 48,* 259–302.

Boykov, Y., Veksler, O., & Zabih, R. (1999). Fast approximate energy minimization via graph cuts. In *Proceedings of the International Conference on Computer Vision* (pp. 377–384). Los Alamitos, CA: IEEE Computer Society Press.

Chou, P., Cooper, P., Swain, M. J., Brown, C., & Wixson, L. (1993). Probabilistic network inference for cooperative high and low level vision. In R. Chellappa & A. K. Jain (Eds.) *Markov Random Field, Theory and Applications.* (pp. 211–243). New York: Academic Press.

Derin, H., & Elliott, H. (1987). Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 9,* 39–55.

Geman, S., & Geman, D. (1984). Stochastic relaxation, gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 6,* 721–741.

Greenman, G. (2000). Turning a map into a cake layer of information. *New York Times.* Retrieved [2000, January] from http://www.nytimes.com/library/tech/00/01/circuits/articles/20giss.html

Gunther, O. (1989). The design of the cell tree: An object-oriented index structure for geometric databases. In *Proceedings of the Fifth International Conference on Data Engineering* (pp. 598–605). Los Alamitos, CA: IEEE Computer Society Press.

Guting, R. (1994). An introduction to spatial database systems. In *Very Large Data Bases Journal, 3,* 357–399.

Haining, R. (1989). Spatial data analysis in the social and environmental sciences. Cambridge, UK: Cambridge University Press.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). Clustering algorithms and validity measures. *Proceedings of the 13th International Conference on Scientific and Statistical Database Management* (pp. 3–22). Los Alamitos, CA: IEEE Computer Society Press.

Hawkins, D. (1980). *Identification of outliers*. London: Chapman & Hall.

Hipp, J., Guntzer, U., & Nakaeizadeh, G. (2000). Algorithms for association rule mining—a general survey and comparison. In *SIGKDD Explorations, 2,* 58–64.

Hohn, M., Gribki, L., & Liebhold, A. (1993). A geostatistical model for forecasting the spatial dynamics of defoliation caused by the Gypsy moth *Lymantria dispar* (Lepidoptera: Lymantriidae). *Environmental Entomology 22,* 1066–1075.

Issaks, E. H., & Srivastava, R. M. (1989). Applied geostatistics. Oxford: Oxford University Press.

Jhung, Y., & Swain, P. H. (1996). Bayesian contextual classification based on modified M-estimates and Markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34,* 67–75.

Koperski, K., Adhikary, J., & Han, J. (1996). Knowledge discovery in spatial databases: Progress and challenges. In *Proceedings of the ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery* (pp. 55–70). Montreal, Canada.

Koperski, K., & Han, J. (1995). Discovery of spatial association rules in geographic information databases. In *Proceedings of the Fourth International Symposium on Large Spatial Databases* (Vol. 951 of Lecture Notes in Computer Science, pp. 47–66). Berlin: Springer-Verlag.

Krugman, P. (1995). *Development, geography, and economic theory.* Cambridge, MA: MIT Press.

LeSage, J. (1997). Bayesian estimation of spatial autoregressive models. *International Regional Science Review, 20,* 113–129.

LeSage, J. P., & Pace, R. (2001). Modeling spatial dependencies for mining geospational data: An introduction. In *Geographic Data Mining and Knowledge Discovery.* Taylor & Francis.

Li, S. (1995). Markov random field modeling in computer vision. New York: Springer-Verlag.

Li, Z., Cihlar, J., Moreau, L., Huang, F., & Lee, B. (1997). Monitoring fire activities in the Boreal Ecosystem. *Journal of Geophysical Research, 102,* 611–629.

Luc, A. (1994). Exploratory spatial data analysis and geographic information systems. In M. Painho (Ed.), *New tools for spatial analysis* (pp. 45–54).

Luc, A. (1995). Local indicators of spatial association: LISA. *Geographical Analysis, 27,* 93–115.

Mark, D. (1999). Geographical information science: Critical issues in an emerging cross-disciplinary research domain. *URISA Journal, 12,* 45–54.

Nepstad, D., Verissimo, A., Alencar, A., Nobre, C., Lima, E., Lefebvre, P., Schlesinger, P., Potter, C., Moutinho, P., Mendoza, E., Cochrane, M., & Brooks, V. (1999). Large-scale impoverishment of Amazonian forests by logging and fire. *Nature, 398,* 505–508.

Orenstein, A., & Merrett, T. (1984). A class of data structures for associative searching. In *Proceedings of the Symposium on Principles of Database Systems* (pp. 181–190). New York: ACM Press.

Ozesmi, U., & Mitsch, W. (1997). A spatial habitat model for the marsh-breeding red-winged black-bird (*Agelaius phoeniceus* l) in coastal Lake Erie wetlands. *Ecological Modelling, 101,* 139–152.

Ozesmi, S., & Ozesmi, U. (1999). An artificial neural network approach to spatial habitat modeling with interspecific interaction. *Ecological Modelling, 116,* 15–31.

Pace, R., & Barry, R. (1997a). Quick computation of regressions with a spatially autoregressive dependent variable. *Geographic Analysis, 29,* 232–247.

Pace, R., & Barry, R. (1997b). Sparse spatial autoregressions. *Statistics and Probability Letters, 33,* 291–297.

RuleQuest Research, Rulequest data mining tools. Retrieved [2002] from http://www.rulequest.com/.

Roddick, J.-F., & Spiliopoulou, M. (1999). A bibliography of temporal, spatial and spatio-temporal data mining research. *SIGKDD Explorations 1,* 34–38.

SAS. Enterprise miner. Retrieved [2002] from http://www.sas.com/products/miner/index.html

Shekhar, S., & Chawla, S. (2002). *A tour of spatial databases.* Prentice Hall.

Shekhar, S., Chawla, S., Ravada, S., Fetterer, A., Liu, X., & Lu, C.-T. (1999). Spatial databases—accomplishments and research needs. *Transactions on Knowledge and Data Engineering 11,* 45–55.

Shekhar, S., & Huang, Y. (2001). Colocation rules mining: A summary of results. In *Proceedings of the Spatio-Temporal Symposium on Databases* (Vol. 2121 *Lecture Notes of Computer Science*). Berlin: Springer-Verlag.

Shekhar, S., & Liu, D.-R. (1997). CCAM: A connectivity-clustered access method for aggregate queries on transportation networks. *IEEE Transactions on Knowledge and Data Engineering, 9,* 102–119.

Shekhar, S., Lu, C., & Zhang, P. (2001). A unified approach to spatial outliers detection. *Technical Report TR01-045 of Department of Computer Science, University of Minnesota.* Retrieved [2001] from http://www.cs.umn.edu/research/shashi-group/paper_list.html

Shekhar, S., Schrater, P. R., Vatsavai, R. R., Wu, W., & Chawla, S. (2002). Spatial contextual classification and prediction models for mining geospatial data. *IEEE Transactions on Multimedia, 4,* 174–188.

Shekhar, S., Yang, T., & Hancock, P. (1993). An intelligent vehicle highway information management system. *International Journal on Microcomputers in Civil Engineering, 8.*

Solberg, A. H., Taxt, T., & Jain, A. K. (1996). A Markov random field model for classification of multisource satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing, 34,* 100–113.

SPSS. Clementine home. Retrieved [2002] from http://www.spss.com/clementine/

Stolorz, P., Nakamura, H., Mesrobian, E., Muntz, R., Shek, E., Santos, J., Yi, J., Ng, K., Chien, S., Mechoso, R., & Farrara, J. (1995). Fast spatio-temporal data mining of large geophysical datasets. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (pp. 300–305). Menlo Park, CA: AAAI Press.

Warrender, C. E., & Augusteijn, M. F. (1999). Fusion of image classifications using Bayesian techniques with Markov random fields. *International Journal of Remote Sensing, 20,* 1987–2002.

Worboys, M. (1995). *GIS: A Computing Perspective.* Bristol, PA: Taylor and Francis.

Yasui, Y., & Lele, S. (1997). A regression method for spatial disease rates: An estimating function approach. *Journal of the American Statistical Association, 94,* 21–32.