



**HAL**  
open science

## Integrating spatial information into probabilistic relational model

Rajani Chulyadyo, Philippe Leray

► **To cite this version:**

Rajani Chulyadyo, Philippe Leray. Integrating spatial information into probabilistic relational model. 2015 IEEE International Conference on Data Science and Advanced Analytics (IEEE DSAA'2015), 2015, Paris, France. 10.1109/DSAA.2015.7344800 . hal-01201226

**HAL Id: hal-01201226**

**<https://hal.science/hal-01201226>**

Submitted on 15 Apr 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Integrating Spatial Information into Probabilistic Relational Models

Rajani Chulyadyo  
LINA UMR 6241, DUKe Research Group  
DataForPeople  
Nantes, France  
Email: rajani.chulyadyo@univ-nantes.fr

Philippe Leray  
LINA UMR 6241, DUKe Research Group  
Nantes, France  
Email: philippe.leray@univ-nantes.fr

**Abstract**—Growing trend of using spatial information in various domains has increased the need for spatial data analysis. As spatial data analysis involves the study of interaction between spatial objects, Probabilistic Relational Models (PRMs) can be a good choice for modeling probabilistic dependencies between such objects. However, standard PRMs do not support spatial objects. Here, we present a general solution for incorporating spatial information into PRMs. We also explain how our model can be learned from data and discuss on the possibility of its extension to support spatial autocorrelation.

## I. INTRODUCTION

Availability of GPS to civilian users, advances in mobile communication and wireless technology have opened the door to many spatial technologies. The development of consumer GPS tools, Volunteered Graphical Information (VGI) tools and location-aware mobile devices have revolutionized the way of collecting and using location information. The growing trend of using spatial information/databases in a wide range of application domain has increased the need for analysis of spatial data.

Spatial data analysis is certainly not a new field [1]. Several methods have been devised to extract patterns from spatial data and understand underlying phenomena [2], [3]. A Bayesian network is one of such methods that have been used with spatial information in various domains [4], [5], [6], [7], [8], [9], [10] for dependency analysis. Recently, object-oriented Bayesian networks have also been used for modeling spatial interactions [5]. However, these approaches are mostly dedicated for specific problems, and many of them do not capture spatial dependencies well. [4]’s approach to modeling spatial dependencies sounds appealing as they present some promising results with their intuitive approach. They automatically learn Bayesian networks from data, limiting spatial dependencies within the vicinity of nodes.

Most of the techniques of spatial data analysis work with flat representation of data. However, real-world applications are generally conceptualized in terms of objects and relations between them, and, hence, data need to be transformed into the required flat format before applying those methods. Besides, analysis of spatial information usually involves the study of interaction between spatial objects. In such relational domains, Probabilistic Relational Models (PRMs) [11] can be employed

to learn probabilistic models. A PRM is a relational extension of Bayesian Networks and models the uncertainty over the attributes of objects in the domain and the uncertainty over the relations between the objects. However, standard PRMs do not support spatial objects. Therefore, we aim at integrating spatial information into PRMs to enable them to handle spatial objects too. Our motivation is also driven by [12]’s perspective on spatial data mining in relational domain. In their paper [12], the authors argue that multi-relational setting is the most suitable for spatial data mining problems and also mention the possibility of using PRMs with spatial relational databases. This view has also been supported by several works [13], [14] not related to PRMs.

We propose to extend standard PRMs to support spatial objects. Our model provides a general way to incorporate spatial information into a PRM and model spatial dependencies. In this paper, we are mainly concerned with geographically referenced objects and deal with vector representation of spatial data, where a spatial object is described by its attributes and its geometry. Though the attributes of spatial objects can be modeled by descriptive attributes in standard PRMs, their geometry cannot be modeled in PRMs in a straightforward manner because the geometry can be as simple as points, which can be described by a pair of coordinates, and also as complex as lines and polygons, which are represented by a sequence of pairs of coordinates. Thus, spatial geometry attributes cannot be treated as simple continuous variables.

One approach to support spatial objects in PRMs is to partition spatial geometry attributes and learn a regular PRM from data using the partitions. However, with this approach, we may miss some useful probabilistic dependencies because the same partitioning level may not be appropriate to detect dependencies of the partitioned attribute with any attributes. Thus, we propose a different approach where the partitioning of spatial geometry attributes is interleaved with the learning process. We first adapt the relational schema for spatial attributes and then define a model over the adapted relational schema. While learning the model from data, adaptive partitioning is performed on spatial geometry attributes to find the partitions that best describe dependencies. We present two different approaches to adapt the partitioning process. We should note here that the partitioning of spatial geometry attributes is

different from discretization of continuous variables in that the former results into spatial objects again. For instance, the partitioning of a set of spatial point objects might result into a set of spatial polygon objects. Because the spatial attributes are not disregarded after partitioning, the model can be further enhanced by adding a support for spatial functions to explore more useful attributes from spatial attributes. We also point to the possibility of modeling spatial autocorrelation [15] with our model. Though spatial autocorrelation cannot be modeled directly in PRMs because of acyclicity constraint [16], we present our idea of modeling spatial autocorrelation by introducing (or deriving) aggregated attributes and adding a special constraint on the orientation of edges between the aggregated attribute and the original attribute to avoid cycles.

This paper is organized as follows. In section 2, we present a brief overview of spatial data and PRMs. In section 3, we present our model in detail and illustrate the model with an example. We discuss about the model in section 4 and finally conclude the paper with our future perspective on the model.

## II. BACKGROUND

### A. Spatial data

*Tessellations* and *vectors* are two basic data types used to represent spatial information though *network* data type has also been reported in some literature [3]. In the tessellation representation of spatial data, the space is partitioned into mutually exclusive cells that together make up the complete coverage. Each cell is associated with thematic/attribute values that represent the conditions for the area covered by that cell. In the vector data representation, spatial objects are modeled using *geometry* and their attributes. The geometry is made up of one or more interconnected vertices. A *vertex* describes a position in space using an x, y and optionally z axis. Spatial objects can be zero-dimensional (point), 1-dimensional (line) or 2-dimensional (surface).

### B. Probabilistic Relational Model (PRM)

A probabilistic relational model (PRM) [17], [11] is a formal approach to relational learning. It defines a template for probability distributions over relational domains. PRMs can be later instantiated with a particular set of objects and relations between them to obtain a Bayesian Network which defines probability distributions over the attributes of the objects. A PRM is comprised of two components: a *relational schema* of the domain and a *probabilistic model*.

a) *Relational schema*: A *relational schema* describes a set of classes and the relationships between them. Each class  $X \in \mathcal{X}$  is described by a set of descriptive attributes  $\mathcal{A}(X)$  and a set of reference slots  $\mathcal{R}(X)$ . The attributes  $X.A_i$  are random variables with states from a discrete domain  $\mathcal{V}(X.A_i)$ . A reference slot  $X.\rho$  that relates an object of class  $X$  to an object of class  $Y$  has  $\text{Domain}[\rho] = X$  and  $\text{Range}[\rho] = Y$ . The inverse of a reference slot  $\rho$  is called *inverse slot* and is denoted by  $\rho^{-1}$ . In the context of relational databases, a class refers to a single database table, descriptive attributes refer to the standard attributes of tables, and reference slots

are equivalent to foreign keys. While a reference slot gives a direct reference of an object with another, objects of one class can be related to objects of another class indirectly through other objects. Such relations are represented with the help of a *slot chain* – a sequence of slots (reference slots and inverse slots)  $\rho_1, \rho_2, \dots, \rho_n$  such that for all  $i$ ,  $\text{Range}[\rho_i] = \text{Domain}[\rho_{i+1}]$ . Slot chains may lead to one-to-many or many-to-many relations, in which case we need an *aggregator*, i.e. a function, denoted  $\gamma$ , which takes a multi-set of values and produces a single value (e.g., average, mode, cardinality etc.) as a summary of the set of related objects.

To illustrate these concepts, we use a relational schema of a system where users order foods in restaurants and rate the service of the restaurants. The schema is shown in figure 1a. We will refer to the same system throughout the paper to give examples. Here, Restaurant, User and Cuisine are also called entity classes. User\_satisfaction and Food\_order are called relationship classes because they represent the relationships Restaurant–User and User–Cuisine respectively. The attributes User.age, User.gender, User\_satisfaction.service\_rating etc. are descriptive attributes whereas User\_satisfaction.user\_id, which refers to User.user\_id, is a reference slot whose domain and range are User\_satisfaction and User respectively. Here, Restaurant and User\_satisfaction objects are directly linked through the reference slot User\_satisfaction.resto\_id. Note that Restaurant.resto\_id<sup>-1</sup> is the inverse of User\_satisfaction.resto\_id and gives all User\_satisfaction objects corresponding to Restaurant objects. Restaurant objects can also be indirectly related to User objects through the slot chain User\_satisfaction.resto\_id<sup>-1</sup>.user\_id, which gives all the users whose satisfaction level and/or service rating about the restaurants are available. As there is a many-to-many relationship between the classes Restaurant and User, this slot chain may result into more than one user for a single restaurant. In such case, we need an aggregator (such as average) to summarize (or aggregate) the resulting set. For instance, AVERAGE(User\_satisfaction.resto\_id<sup>-1</sup>.user\_id.age) gives the average age of the users who have rated the restaurants.

b) *Probabilistic model*: A PRM specifies a *probabilistic model* for classes of objects. It represents generic probabilistic dependencies between the attributes of classes in the relational schema. The dependencies can be between the attributes of the same class or between the attributes of different classes. Like in Bayesian networks, the dependency structure is associated with the conditional probability distribution of each node (attribute) given its parents.

Figure 1b depicts a PRM that corresponds to the relational schema in 1a. Here, the dashed lines indicate that the classes are linked through reference slots in the relational schema, and the edge with 'Agg' denotes that the child node depends on the aggregated value of the parent node.

Formally, a regular PRM  $\Pi$  for a relational schema  $\mathcal{R}$  is defined as follows [11]. For each class  $X \in \mathcal{X}$  and each descriptive attribute  $A \in \mathcal{A}(X)$ , we have:

- a set of parents  $\text{Pa}(X.A) = \{U_1, \dots, U_l\}$ , where each  $U_i$

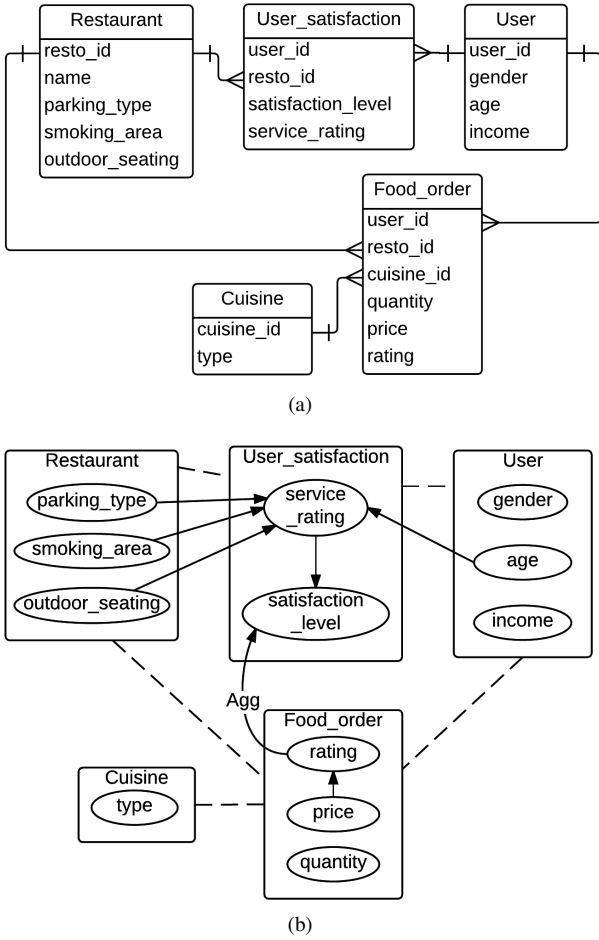


Fig. 1. (a) An example of a relational schema of a system where users orders foods in restaurants and rate the services of the restaurants. The relationships between classes are denoted by the lines connecting them, and the type of their relationships is given by the line ends, e.g. the class Restaurant has a one-to-many relationship with the class User\_satisfaction. (b) A PRM that corresponds to the relational schema in 1a. Each descriptive attribute is denoted by a node. The directed edges between the nodes represent probabilistic dependencies between the nodes. The dashed lines between the classes indicate that the classes are related to each other through reference slots in the relational schema. An aggregator is indicated by ‘Agg’ in an edge. An aggregated dependency is present in the edge from Food\_order.rating to User\_satisfaction.satisfaction\_level, which means that the users’ satisfaction level depends probabilistically on the aggregated value of their rating on foods ordered by them.

has the form  $X.B$  or  $\gamma(X.\tau.B)$ , where  $B$  is an attribute of any class,  $\tau$  is a slot chain and  $\gamma$  is an aggregator of  $X.\tau.B$ .

- a conditional probability distribution (CPD),  $P(X.A|Pa(X.A))$ .

Probabilistic inference is performed on a *Ground Bayesian Network (GBN)* obtained by instantiating a PRM for a particular *instance*  $\mathcal{I}$  of the associated relational schema (i.e. a set of objects and relations between them). It specifies probability distributions over the attributes of the objects. A GBN has a node for every attribute of every object in  $\mathcal{I}$ , and probabilistic dependencies and CPDs as in the PRM. As GBNs tend to be very big, standard inference algorithms for Bayesian networks

need to be adapted to infer the value of unobserved attributes [11].

*Learning a PRM* involves two tasks – *parameter estimation* and *structure learning*. The aim of parameter estimation is to find parameters  $\theta_s$  that define CPDs for a dependency structure  $\mathcal{S}$  given a complete instantiation  $\mathcal{I}$ . While learning the structure of a PRM, we need to make sure that the structure is not only legal (i.e. acyclic) but also the best one among the candidate structures. [11] propose to apply greedy hill-climbing algorithm over relational attributes (limiting the maximum length of slot chain) to explore structure search space and use a *Bayesian scoring function* to evaluate legal candidate structures. While exploring and evaluating structures, one needs to compare score of structures with only small local differences (e.g., comparing a structure with another structure obtained from it by adding only one edge between two nodes). In such case, a *decomposable* Bayesian scoring function can simplify the computation because it allows computing the separate contribution of each variable to the quality of the network so that it will be sufficient to compute the score of only those terms that involve the variables being modified instead of computing score of the complete structure again. A scoring function is *decomposable* if the global score assigned to a Bayesian network  $B$  with  $N$  nodes for the given data  $D$  can be represented as a sum of local score that depends only on each node  $X_i$  and its parents  $Pa(X_i)$ , i.e.,

$$(Global)Score(B, D) = \sum_{i=1}^N (local)Score(X_i|Pa(X_i)) \quad (1)$$

### III. PRM WITH SPATIAL ATTRIBUTES (PRM-SA)

We begin by defining some basic concepts related to our model. We, then, define our model and illustrate the model with examples. Then we explain how to learn our model.

#### A. Definitions

We propose to incorporate the vector representation of spatial objects, where a spatial object is described by its location in space in terms of geometry and its attributes, in PRMs. The attributes of a spatial object can form descriptive attributes in the relational schema whereas its geometry cannot be incorporated as descriptive attributes because the geometry of spatial objects is represented by a set of points (vertices or coordinates). Thus, we coin a term *spatial geometry attribute* or simply *spatial attribute* to describe such attributes to use them in a PRM.

##### Definition 1: Spatial geometry attribute

An attribute is a *spatial geometry attribute* if its value  $s$  is a sequence of pairs of coordinates in *geographic coordinate reference systems* and defines the geometry of its class. i.e.  $s = ((x, y)_n : n \in \mathbb{N})$  where  $x$  is called *longitude*,  $y$  is called *latitude* and  $n$  is the cardinality of  $s$ . Here,  $s$  represents

- a point geometry if  $n = 1$ ,
- a line geometry if  $n \geq 2$  and  $s_1 \neq s_n$ ,
- a polygon geometry if  $n \geq 2$  and  $s_1 = s_n$ . ■

*Definition 2: Spatial class*

Let  $\mathcal{SA}(X)$  be the set of spatial geometry attributes in a class  $X$ . A class  $X$  is a *spatial class* if  $\mathcal{SA}(X)$  is not empty. ■

To illustrate these concepts, we refer to the system in figure 1 and extend the relational schema in figure 1a with a spatial geometry attribute in the class Restaurant as shown in figure 2a. Here, the spatial attribute  $Restaurant.location$  represents the location of the objects of the spatial class Restaurant.

As the set of possible values of a spatial attribute is infinite, conditional probability distributions associated with spatial attributes would be very big. It demands an extensive computation for learning as well as inference and this is practically too difficult to achieve. Therefore, we propose to partition this set into a finite number of disjoint subsets with the help of a *spatial partition function*. Each partition is then represented by a class, which we call a *spatial partition class*, and a reference slot (we call it a *spatial reference slot* or *spatial ref. slot*) that refers to the objects of the partition class is added in the corresponding spatial class. Partition functions are responsible for creating the objects of partition classes and mapping the values of a spatial attribute to their corresponding partitions.

*Definition 3: Spatial partition function, Spatial partition class*

Let  $X.SA$  be a spatial geometry attribute of a spatial class  $X$ . We define a *spatial partition function*  $f_{sa} : X.SA \rightarrow Range[f_{sa}]$  where  $Range[f_{sa}]$  is a finite set of spatial partitions represented by a *spatial partition class*  $P_{XSA}$ . Thus,  $f_{sa}$  associates each  $sa \in Domain[X.SA]$  to an object of  $P_{XSA}$  determined by the function itself. ■

Partition functions essentially map a spatial object to a region such that spatial objects meeting some partitioning criteria are grouped together. Such mapping can be achieved in many ways. One way is to use regular square or hexagonal (honeycomb) grids to partition the spatial region. Partitions can also be created by using standard, publicly available knowledge such as administrative boundaries. In the absence of such knowledge, *spatial clustering algorithms* can be used. [18] have presented a survey on several spatial clustering methods. Some clustering algorithms, such as K-means, require users to provide the number of clusters/regions and some others, such as DBSCAN, can determine the number of clusters themselves. It should be noted that granularity of partitioning methods depends on the context of the problem. Also note that when using knowledge to create partitions, we may have access to some extra information. Such information can be considered as descriptive attributes of partition classes. The introduction of partition classes enables us to implement hierarchical clustering as well because partition classes can contain spatial attributes, which can be further partitioned thereby creating a hierarchy of spatial partition classes.

We refer to figure 2a for examples. As the spatial attribute  $Restaurant.location$  can take infinitely many values, we define a partition function to partition its possible values into a finite set of spatial partitions. Thus, we add a spatial par-

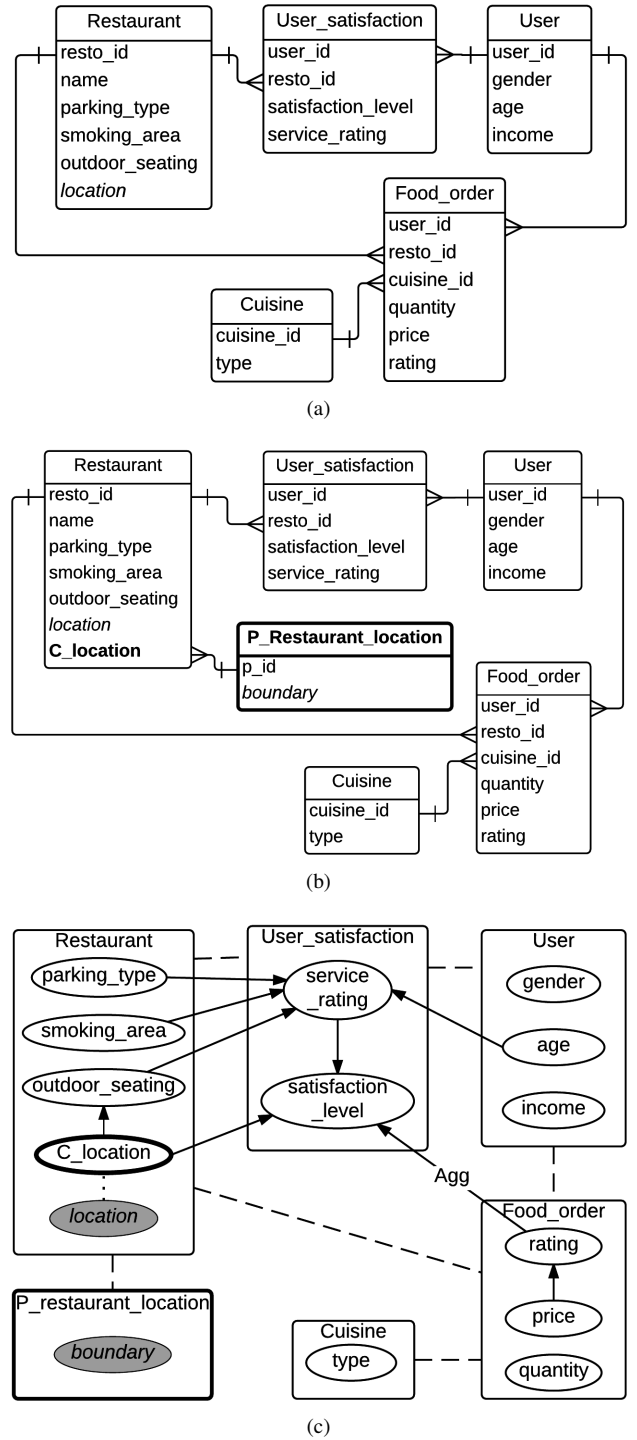


Fig. 2. (a) An example of a relational schema with a spatial attribute  $Restaurant.location$ , which cannot be handled by standard PRMs. (b) The relational schema adapted for the spatial attribute  $Restaurant.location$  as proposed in definition 4.1. Here, spatial attributes are shown in italicized font and the added spatial reference slot and the spatial partition class are shown in boldface. (c) A PRM-SA as proposed in definition 4. The gray nodes are spatial attributes and the one with thick border is the spatial reference slot associated with the spatial attribute  $Restaurant.location$ . The dotted line between  $Restaurant.location$  and  $Restaurant.C_location$  indicates that the spatial attribute and the spatial ref. slot are associated through a spatial partition function.

tion class `P_Restaurant_location` that represents the spatial partitions of `Restaurant.location`. The objects of this spatial partition class will then be referenced in the spatial class `Restaurant` by the spatial reference slot `Restaurant.C_location`. Figure 2b shows the relational schema adapted for the spatial attribute. Here, we have assumed that the spatial partition class will have an additional attribute called *boundary*, which is again a spatial attribute. In this example, we can assume that this attribute represents the convex hull formed by all the locations mapped to the particular partition object. However, the attributes present in the partition classes depend on the context. If we use the information about administrative boundaries of cities or regions, `P_restaurant_location.boundary` might represent the boundary of the specified location, and we might even have extra information about the partitions such as population, average income, demographic structure etc. of the location. If hierarchical administrative division is available, we can incorporate hierarchical partitions by further partitioning `P_restaurant_location.boundary` and adding another partition class, say `P_restaurant_location_boundary` (not shown in the figure).

We now define our model, which is based on standard PRMs and supports spatial data.

*Definition 4: PRM with spatial attributes (PRM-SA)*

Let  $\mathcal{A}(X)$  and  $\mathcal{SA}(X)$  denote the set of descriptive attributes and geometry attributes respectively in class  $X$ .

For each spatial class  $X \in \mathcal{X}$  such that  $\mathcal{SA}(X) \neq \emptyset$  and for each geometry attribute  $SA \in \mathcal{SA}(X)$ , we define the following:

- a new partition class  $P_{XSA}$ ,
- a partition function  $f_{sa} : SA \rightarrow P_{XSA}$  that creates instances of  $P_{XSA}$  associating each  $sa \in \text{Domain}[SA]$  to one of the instances of  $P_{XSA}$ , and
- a new spatial reference slot  $X.C_{SA}$  associated with  $f_{sa}$ .

Then, we adapt the relational schema for spatial attributes and define the probabilistic model in the following way.

*Definition 4.1: Adapted relational schema*

The relational schema is adapted for spatial attributes by adding  $P_{XSA}$  and  $X.C_{SA}$  associated with  $f_{sa}$  for each spatial class  $X \in \mathcal{X}$  and for each geometry attribute  $SA \in \mathcal{SA}(X)$ .

*Definition 4.2: Probabilistic model of a PRM-SA*

Let  $\mathcal{P}_{SA}$  and  $\mathcal{C}_{SA}$  be the set of partition classes and the set of added spatial reference slots respectively. Then, for each class  $X \in \{\mathcal{X} \cup \mathcal{P}_{SA}\}$  and each attribute  $A \in \{\mathcal{A}(X) \cup \mathcal{C}_{SA}(X)\}$ , we have

- a set of parents  $Pa(X.A) = \{U_1, \dots, U_l\}$ , where each  $U_i$  has the form  $X.B$  or  $\gamma(X.\mathbf{K}.B)$ , where  $B$  is an attribute of any class,  $\mathbf{K}$  is a slot chain and  $\gamma$  is an aggregate of  $X.\mathbf{K}.B$ ,
- a legal conditional probability distribution CPD,  $P(X.A|Pa(X.A))$ . ■

A PRM-SA that corresponds to the schema in figure 2a is shown in figure 2c. Here, the gray nodes are spatial attributes and the nodes/classes with thick border are the ones that are not present in the original relational schema. The probabilistic dependencies shown in the examples are hypothetical.

Probabilistic inference is performed on a ground Bayesian network (GBN) obtained by instantiating a PRM-SA for a given relational skeleton. Here, the skeleton must also include the objects of spatial classes. Given such a relational skeleton, a PRM-SA induces a GBN that specifies probability distributions over the attributes of the objects. Here, we need to ensure that the probability distributions are *coherent*, i.e. the sum of probability of all instances is 1. In Bayesian networks, this requirement is satisfied if the dependency graph is acyclic [19]. Following [11]’s approach, we consider *instance dependency graph* to check whether the dependency structure  $\mathcal{S}$  of a PRM-SA is acyclic relative to a given relational skeleton. Due to the presence of spatial attributes, standard instance dependency graphs need to be redefined with some adaptations for PRM-SA but we can still follow [11]’s proof to show that the dependency structure  $\mathcal{S}$  for a PRM-SA is guaranteed to be acyclic for the given relational skeleton if the corresponding instance dependency graph is acyclic.

*Definition 5: Instance dependency graph (IDG)*

The instance dependency graph  $G_{\sigma_r}$  for a PRM-SA  $\Pi$  with partition classes  $\mathcal{P}_{SA}$  and a relational skeleton  $\sigma_r$  is defined as follows. For each object  $x \in \sigma_r(X)$  in each class  $X \in \{\mathcal{X} \cup \mathcal{P}_{SA}\}$ , we have the following nodes: a node  $x.A$  for each descriptive attribute  $X.A$ , and a node  $x.C_{SA}$  for each spatial reference slot  $X.C_{SA}$ . The graph has the following edges:

- 1) Type I edges: For each formal parent of  $x.A$ ,  $X.B$ , we introduce an edge from  $x.B$  to  $x.A$ .
- 2) Type II edges: For each formal parent  $X.\mathbf{K}.B$ , and for each  $y \in x.\mathbf{K}$ , we define an edge from  $y.B$  to  $x.A$ .
- 3) Type III edges: For any spatial attribute  $x.SA$  in each spatial class  $X \in \mathcal{X}$ , we define an edge  $x.SA \rightarrow x.C_{SA}$ .
- 4) Type IV edges: For any attribute  $p.A$  in each spatial partition class  $P \in \mathcal{P}_{SA}$  and  $p \in \sigma_r(P)$ , we add an edge  $p.A \leftarrow x.A$  if  $P.A$  is derived from  $X.A$ . ■

Following [11] again, we can demonstrate that the probabilistic model of a PRM-SA is coherent for *any* relational skeleton if the corresponding *class dependency graph* is acyclic. Here again, because of the presence of spatial attributes, we redefine class dependency graph for PRM-SA.

*Definition 6: Class dependency graph (CDG)*

The class dependency graph  $G_{\Pi}$  for a PRM-SA  $\Pi$  is defined as follows. The dependency graph has the following nodes: a node for each descriptive attribute  $X.A$ , and a node for each spatial reference slot  $X.C_{SA}$ . The graph has the following edges:

- 1) Type I edges: For any attribute  $X.A$  and its parent  $X.B$ , we introduce an edge from  $X.B$  to  $X.A$ .
- 2) Type II edges: For any attribute  $X.A$  and its parent  $X.\mathbf{K}.B$ , we introduce an edge from  $Y.B$  to  $X.A$ , where  $Y = \text{Range}[X.\mathbf{K}]$ .
- 3) Type III edges: For any spatial attribute  $X.SA$  in each spatial class  $X \in \mathcal{X}$ , we define an edge  $X.SA \rightarrow X.C_{SA}$ .

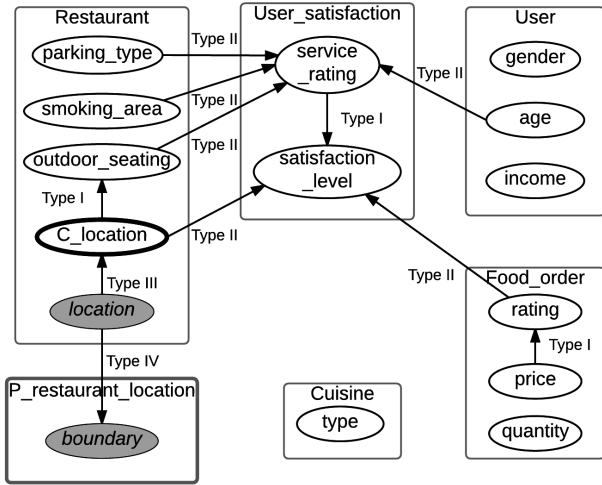


Fig. 3. The class dependency graph for the PRM-SA in figure 2c. Because there is no cycle in this graph, we can conclude that the dependency structure in figure 2c is acyclic for any relational skeleton.

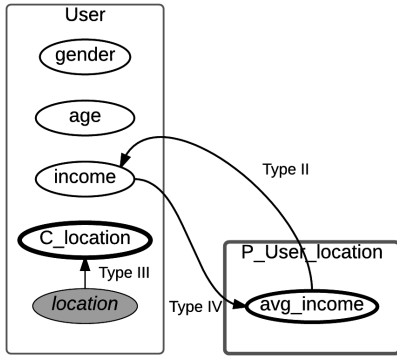


Fig. 4. An example of a class dependency graph with a cycle. Because of the presence of a cycle, we can conclude that the dependency structure cannot be guaranteed to be acyclic for any relational skeleton.

- 4) Type IV edges: For any attribute  $P.A$  in each spatial partition class  $P \in \mathcal{P}_{S,A}$ , we add an edge  $P.A \leftarrow X.A$  if  $P.A$  is derived from  $X.A$ . ■

Figure 3 shows the class dependency graph for the PRM-SA in figure 2c. Because this graph is acyclic, the dependency structure in figure 2c is guaranteed to be acyclic regardless of relational skeleton.

Let us consider another hypothetical example where there is an attribute  $User.location$  and a partition class  $P\_User\_location$  that represents the partitions of users' location. Suppose there exists a dependency that says the income of a user depends on the average income of the users in his community, i.e. users with similar income tend to live in the same community. In this case, the dependency structure of a PRM-SA will have an edge from  $P\_User\_location.avg\_income$  to  $User.income$ . Although there is no cycle in the structure, it is incoherent because the class dependency graph of this structure contains a cycle as shown in figure 4. This is due to the type IV edge that is added because  $P\_User\_location.avg\_income$  is derived from

$User.income$ .

## B. Learning PRM-SA

As with standard PRMs, learning a PRM-SA involves the tasks of parameter estimation and structure learning. Parameters of a PRM-SA can be learned in the same way as for a regular PRM. However, the instances of all partition classes must be included in the instantiation of the relational schema.

As for structure learning, following [17]'s approach, we apply relational greedy search algorithm to explore the search space of candidate structures and evaluate legal candidate structures using *score-based* methods. However, due to the introduction of partition classes along with partition functions, we need to adapt the search algorithm to deal with partitions. Here, we come across two situations – 1) when the number of partitions of the spatial attribute (i.e.  $Cardinality(Range[f_{sa}])$ , let's denote it by  $k_{sa}$ ) is known, and 2) when it is unknown. In the former case, standard relational greedy search algorithm can be applied to explore the search space. However, the situation is complicated in the latter case where the number of partitions needs to be determined by the algorithm.

We present two approaches to learn a PRM-SA when  $k_{sa}$  is unknown. A naive way is to add new operators *increase\_k* and *decrease\_k*, which increases or decreases the number of partitions respectively, and use these operators along with *add*, *delete* and *revert edge* operators of standard greedy search algorithm to find neighborhood of a structure. In our second approach, we separate the tasks of greedy search over candidate structures and finding the optimal number of partitions of spatial attributes. The basic idea is to pick the best scoring structure among candidate structures, and then find the optimal number of partitions of the spatial attributes in this structure if this structure is obtained by changing (i.e. adding, deleting or reverting) an edge that involves a *spatial reference slot*. To simplify the computation, we use a *decomposable* Bayesian scoring function. So, it is sufficient to compute the score of only those terms that involve the variables being modified.

An important point to note here is that if a spatial reference slot appears with other spatial reference slots in the local score terms of the scoring function (while finding the optimal number of partitions), we need to vary the cardinality of the set of all spatial reference slots that appear together with the target spatial reference slot. To visualize this concept, we propose to *moralize*<sup>1</sup> the structure and find *2-vertex cliques* of spatial reference slots. Those connected through the cliques form a set of variables whose cardinality should be varied altogether when optimizing the number of partitions. This concept is illustrated in figure 5. Note that spatial partition classes associated with each spatial attributes exist here but they are not shown in the diagram to save some space. Suppose the PRM-SA in figure 5a is obtained at some point during structure learning by adding

<sup>1</sup>A moral graph is the equivalent undirected graph of a directed acyclic graph and is obtained by adding an edge between the nodes that have a common child and changing directed edges to undirected ones.

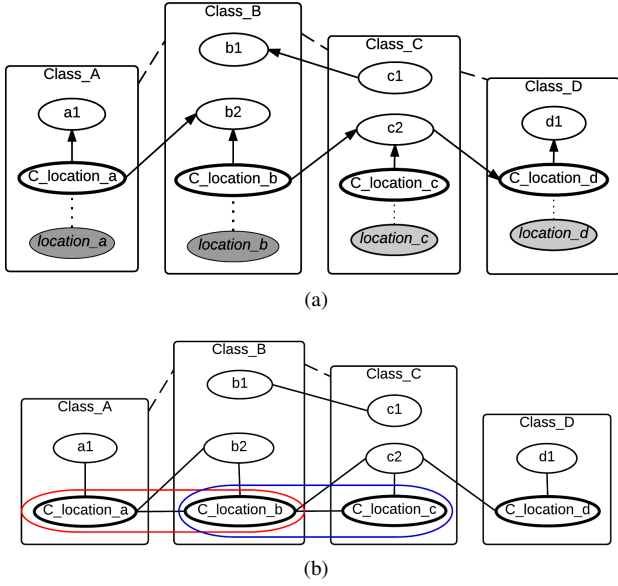


Fig. 5. (a) A PRM-SA with multiple spatial attributes. (b) The corresponding moralized graph used to identify the set of partition functions to be optimized. Here, the pairs of spatial reference slots  $\{\text{Class\_A.C\_location\_a}, \text{Class\_B.C\_location\_b}\}$  and  $\{\text{Class\_B.C\_location\_b}, \text{Class\_C.C\_location\_c}\}$  form 2-vertex cliques. Because these two cliques are connected, we need to find the optimal number of partitions for these three spatial ref. slots together.

an edge from  $\text{Class\_A.C\_location\_a}$  to  $\text{Class\_A.a1}$ . Because the added edge contains a spatial reference slot, we need to find the best number of partitions for this node. The score  $S$  of this structure is

$$\begin{aligned}
 S = & S(\text{Class\_A.a1}|\text{Class\_A.C\_location\_a}) \\
 & + S(\text{Class\_A.C\_location\_a}) + S(\text{Class\_B.b1}|\text{Class\_C.c1}) \\
 & + S(\text{Class\_B.b2}|\text{Class\_A.C\_location\_a}, \text{Class\_B.C\_location\_b}) \\
 & + S(\text{Class\_C.c2}|\text{Class\_B.C\_location\_b}, \text{Class\_C.C\_location\_c}) \\
 & + S(\text{Class\_C.c1}) + S(\text{Class\_D.d1}|\text{Class\_D.C\_location\_d}) \\
 & + S(\text{Class\_D.C\_location\_d}|\text{Class\_C.c2})
 \end{aligned}$$

From equation 2, it is clear that changing the number of partitions of  $\text{Class\_A.C\_location\_a}$  affects the score of the nodes  $\text{Class\_A.a1}$  and  $\text{Class\_B.b2}$ . However, to find the best score for the node  $\text{Class\_B.b2}$ , we need to find the optimal number of partitions for  $\text{Class\_B.C\_location\_b}$  too because it appears with the spatial reference slot  $\text{Class\_A.C\_location\_a}$  in local score terms of the scoring function.  $\text{Class\_B.C\_location\_b}$  also appears with  $\text{Class\_C.C\_location\_c}$  in the scoring function. As a result, partition functions of all of the three spatial ref. slots need to be optimized together. To identify this set, we can moralize the dependency structure of the PRM-SA as shown in figure 5b and find out all 2-vertex cliques of spatial reference slots. The three spatial reference slots  $\text{Class\_A.C\_location\_a}$ ,  $\text{Class\_B.C\_location\_b}$  and  $\text{Class\_C.C\_location\_c}$  in figure 5b are connected through cliques. Therefore, we find the best

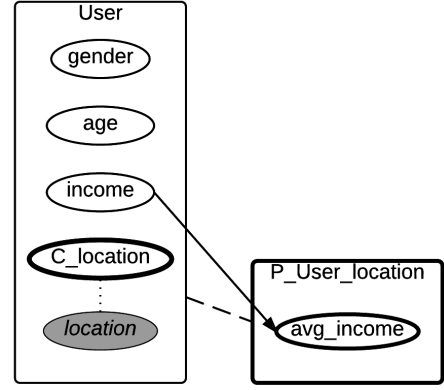


Fig. 6. An example of a dependency structure that models the dependency of an attribute with the aggregated value of the same attribute of spatial objects in the same cluster. This concept can be used to model spatial autocorrelation.

number of partitions for these three spatial reference slots altogether.

An interesting property of this approach of learning PRM-SA is that because of the separation of greedy search and partition size optimization, we can come up with different heuristics that involve different combinations of these two tasks. For example, we can perform one operation (add, delete or revert) of greedy search and then find the cardinality of spatial reference slots, or we can find the optimal cardinality of all spatial reference slots after a complete greedy search and so on. However, the best approach among these is yet to be discovered.

#### IV. DISCUSSION

Our model sounds somewhat similar to *PRM with reference uncertainty (PRM-RU)* [11] because the notion of partition and partition function has already been coined there. However, there is no reference uncertainty in our model. We partition spatial objects because the set of possible values of a geometry attribute is so big (or infinite) that specifying a probability distribution over the set of all values of a geometry attribute is too much of work. Unlike in PRM-RU, where more than one attribute (called *partition attributes*) can define a partition, there is only one partition attribute, i.e. the corresponding spatial attribute, for each partition class in our model. On the other hand, type III edges in IDG and CDG of our model correspond to type IV edges in those of PRM-RU.

Learning structure of our model when the number of partitions of spatial reference slots is unknown is inspired by standard methods of latent variable discovery in Bayesian networks [20]. We deal with the problem of finding the right dimensionality of special variables but these variables, in fact, are not latent. Thus, unlike in latent variable discovery techniques, we do not need to apply *Expectation Maximization (EM)* algorithm to score those special variables because our partition functions take care of providing complete data for those variables.



Researchers advocate the consideration of autocorrelation in spatial data analysis. However, autocorrelation cannot be modeled directly in PRMs because of acyclicity constraint [16]. An attribute of an instance depending on the same attribute of neighboring instances would create a cycle because of the fact that neighborhood is a mutual concept. Due to this reason, we do not model spatial autocorrelation directly. Nevertheless, it is still possible to extend our model to enforce the modeling of spatial autocorrelation by adding aggregated descriptive attributes in partition classes with a special constraint that the aggregated attribute must always be a child to ensure acyclicity. For example, let us consider the relational schema in figure 2a. Suppose there is an attribute *User.location* and a partition class *P\_User\_location* that represents the partitions of users' location. Modeling spatial autocorrelation of users' income (i.e. users' income depending on the income of neighboring users) would create a cycle in the dependency structure. To avoid this, we could add an attribute *avg\_income* in the *P\_User\_location*, which is, in fact, the average (aggregation) of *User.income*. *P\_User\_location.avg\_income* would, then, be enforced to be a child of *User.income* as shown in figure 6, otherwise we may come across the situation as in figure 4. This models the dependency of an attribute with the aggregated value of the same attribute of spatial objects in the same cluster. Relational Markov Networks (RMNs) could be another solution to model this spatial autocorrelation because dependencies are represented by an undirected graph in this type of probabilistic graphical models. However, learning RMNs is much more complex than learning PRMs.

## V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a generic framework to incorporate spatial information into PRMs. Our model extends standard PRMs and provides a general solution to model spatial dependencies in PRMs. We have also presented our approaches to learning the model. Our model opens a possibility to model spatial autocorrelation with the introduction of aggregated attributes on partition classes and a special constraint on the orientation of edges to avoid cycles. Because the spatial attributes are preserved in a PRM-SA, our model can be extended to support spatial functions that work directly with the spatial attributes. We are currently working to implement this model in PILGRIM, a software platform to work with probabilistic graphical models, being developed at our research lab, and to perform experiments to prove our concept. In future, we are planning to enhance this model with [2] H. J. Miller and J. Han, *Geographic data mining and knowledge discovery*. CRC Press, 2009.

spatial functions to explore more meaningful attributes in the domain. We aim at applying this model in a recommender system for a real-world application.

## REFERENCES

- [1] M. F. Goodchild, "Twenty years of progress: GIScience in 2010," *Journal of Spatial Information Science*, vol. 1, no. 1, pp. 3–20, 2010.
- [2] S. Shekhar, M. R. Evans, J. M. Kang, and P. Mohan, "Identifying patterns in spatial information: A survey of methods," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 193–214, May 2011.
- [3] R. Cano, C. Sordo, and J. Gutierrez, "Applications of Bayesian networks in meteorology," *Advances in Bayesian networks*, pp. 309–327, 2004.
- [4] L. Wilkinson, Y. E. Chee, A. E. Nicholson, and P. F. Quintana-Ascencio, "An Object-oriented Spatial and Temporal Bayesian Network for Managing Willows in an American Heritage River Catchment," in *UAI Application Workshops*, 2013, pp. 77–86.
- [5] L. Li, J. Wang, H. Leung, and C. Jiang, "Assessment of catastrophic risk using bayesian network constructed from domain knowledge and spatial data," *Risk Analysis*, vol. 30, no. 7, pp. 1157–1175, 2010.
- [6] S. Sarkar and K. Boyer, "Integration, inference, and management of spatial information using bayesian networks: perceptual organization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 3, pp. 256–274, Mar 1993.
- [7] M.-h. Park, J.-h. Hong, and S.-b. Cho, "Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices," *Ubiquitous Intelligence and Computing*, pp. 1130–1139, 2007.
- [8] J. Huang and Y. Yuan, "Construction and Application of Bayesian Network Model for Spatial Data Mining," in *IEEE International Conference on Control and Automation. ICCA 2007*, 2007, pp. 2802–2805.
- [9] A. R. Walker, B. Pham, and M. Moody, "Spatial Bayesian Learning Algorithms for Geographic Information Retrieval," pp. 105–114, 2005.
- [10] L. Getoor, "Learning Statistical Models from Relational Data," PhD Thesis, Stanford University, 2001.
- [11] D. Malerba, "A relational perspective on spatial data mining," *International Journal of Data Mining, Modelling and Management*, vol. 1, no. 1, pp. 103–118, 2008.
- [12] M. Ceci and A. Appice, "Spatial associative classification: propositional vs structural approach," *Journal of Intelligent Information Systems*, vol. 27, no. 3, pp. 191–213, 2006.
- [13] D. Malerba, A. Appice, A. Varlaro, and A. Lanza, "Spatial clustering of structured objects," in *Inductive Logic Programming*. Springer, 2005, pp. 227–245.
- [14] D. A. Griffith, "What is spatial autocorrelation? Reflections on the past 25 years of spatial statistics," *Espace géographique*, vol. 21, no. 3, pp. 265–280, 1992.
- [15] J. Neville and D. Jensen, "Collective Classification with Relational Dependency Networks," in *Workshop on Multi-Relational Data Mining. KDD-2003*, 2003, pp. 77–91.
- [16] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *IJCAI*, vol. 16, 1999, pp. 1300–1309.
- [17] J. Han, M. Kamber, and A. K. H. Tung, "Spatial Clustering Methods in Data Mining: A Survey," in *Geographic Data Mining and Knowledge Discovery, Research Monographs in GIS*, H. J. Miller and J. Han, Eds. Taylor and Francis, 2001, pp. 1–29.
- [18] L. Getoor, N. Friedman, D. Koller, and A. Pfeffer, "Learning Probabilistic Relational Models," in *Relational Data Mining*, S. Dzeroski and N. Lavrač, Eds. Springer Berlin Heidelberg, 2001, pp. 307–335.
- [19] G. Elidan and N. Friedman, "Learning the Dimensionality of Hidden Variables," in *UAI-2001*. San Francisco, CA: Morgan Kaufmann Publishers, 2001, pp. 144–151.