Chapter 3

GRAPH-BASED REPRESENTATIONS

This chapter describes our graph-based model to represent together spatial data, non-spatial data and spatial relations among the spatial objects. We propose to include three types of spatial relations: topological, distance and direction. Section 3.1 presents some issues related to the graph-based knowledge discovery. Our methodology is detailed in Section 3.2. In Section 3.3 we describe five graph-based representations based on our model. Finally, an example of the applicability of the proposal is presented in Section 3.4.

3.1 Generalities

In Chapter 2 we presented several approaches developed to search knowledge in spatial databases. The differences between these approaches are based on the data representation and the data mining algorithm used in the search task. Certainly, the data representation used by a mining tool is very important, and it has to be powerful enough to represent domains containing complex relations among their components (i.e., spatial data domain).

A graph-based representation has these characteristics [3, 5, 17, 36]. It has the benefits of being easy to understand and flexible enough to create different representations of the same domain. The domain (i.e., data and relationships) is described using graphs. These graphs become the input to a graph-based discovery tool which uses a heuristic to choose the subgraphs that are considered important (patterns).

A graph is defined as a pair $G = (V,E)$. $V = \{v_1,\ldots,v_n\}$ denotes a finite set of elements called vertices. E is a set of edges e satisfying $E \subseteq [V]^2$. Then, each edge $e \in E$ is a pair (v_i,v_j). If (v_i,v_j) is an ordered pair for any $(v_i,v_j) \in E$, then $G = (V,E)$ is said to be a directed graph. A labeled graph has labels associated with its edges and vertices.

In knowledge discovery systems using a graph-based approach, the data mining algorithm uses graphs to represent knowledge; this means that the data preparation phase includes the transformation of the data to a graph format. The *search space* of a graph-based data mining algorithm consists of all the subgraphs that can be derived from its input graph.

In the literature there exist several definitions about what a spatial data is. However, before presenting our spatial model definition, we precise the following issues:

- When we speak about a geometric object we refer to an object describing a form (i.e., point, line, and polygon).
- A spatial object refers to an object represented by a coordinate system (i.e., Cartesian coordinates).

- A geographic object may be considered a specialization of a spatial object because it is represented by a coordinate system but related to earthly coordinates (sometimes called Geodetic or Geographic coordinates).

A model is a simplification of the reality. It is not the reality, rather it represents the reality. A model is used to explain or to understand the reality. A model can be, for instance, an equation, a hypothesis or a structured idea. A spatial model is therefore an abstraction of spatial data that generates useful information to help us understand, describe, and predict how things work and/or solve problems in the real world. When we work with geographic coordinates we can talk about a geographic model.

3.2 Methodology

The idea is to create a graph-based model to represent together spatial data, non-spatial data and the spatial relations between spatial objects. We will generate datasets composed of graphs with a set of these three elements. We argue that by mining a dataset with these characteristics a data mining algorithm can search patterns involving all these elements, at the same time, improving the results of the spatial analysis process.

For example, finding out interesting patterns of objects located at some distance from a particular point; focusing on a current problem such as finding risk zones near the Popocatépetl volcano (México). In addition, it would be important to know the characteristics of the evacuation routes which would be used in situations of volcanic

activity, i.e., what are the characteristics of the evacuation routes; could they withstand the atmospheric conditions and the passage of vehicles in an emergency situation?

A significant characteristic of spatial data is that the attributes of the neighbors of an object may have an influence on the object itself. So, we propose to include in the model the three types of relationship mentioned in Chapter 2: topological, distance, and direction relations.

As we mentioned, the basis is that in a spatial database there exist spatial objects, these objects interact with other objects (spatial relations) and they may have several attributes describing them (non-spatial data). Thus, we propose to create graphs that help us to describe these interconnections between all these elements.

A simple way can be as follows: for each object we create a vertex representing the object itself and join two vertices with a directed labeled edge if they have a spatial relation in common (we add one edge for each spatial relation among vertices). Additionally, we can also create vertices representing the value of their non-spatial attributes and directed labeled edges joining each attribute to its object (one edge per attribute).

But there is a higher complexity level when working with general graphs (a graph with multiple edges between vertices) than with simple graphs (a graph with at most one edge between any given pair of vertices and with no loops). Thus, our idea is to work with simple graphs. So, we propose to improve this representation to the one shown in Figure 3.1 (using *UML* notation).

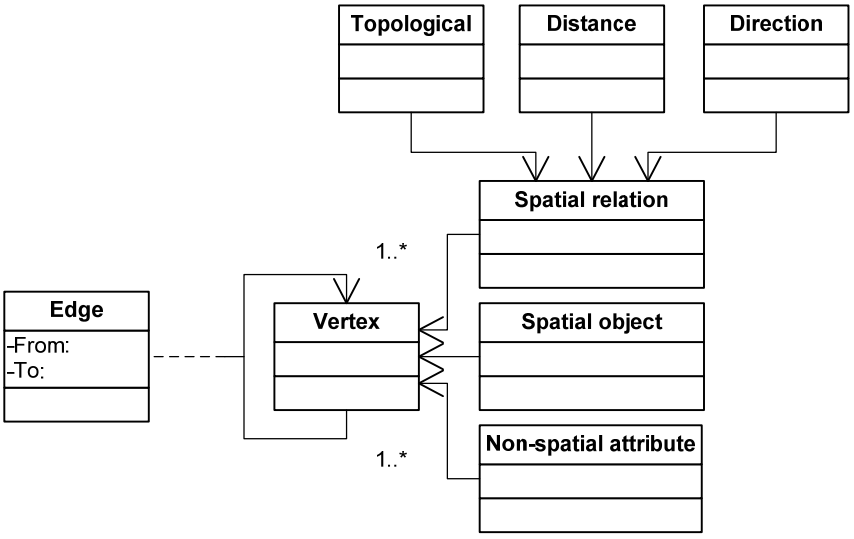

Figure 3.1. General graph-based model to represent spatial data.

In the model, the spatial data (i.e., spatial objects), non-spatial data (i.e., attributes), and spatial relations are represented as a collection of one or more directed graphs. Therefore, a directed graph contains a collection of vertices and edges representing all these elements.

Vertices represent either spatial objects, spatial relation types between two spatial objects (binary relation), or non-spatial attributes describing the spatial objects. Edges represent a link between two vertices of any type. According to the type of vertices that an edge joins, an edge can represent either an attribute name or a spatial relation name. The attribute name can refer to a spatial object or a non-spatial entity. We use directed edges to represent directional information of relations among elements (i.e., object *x* covers object *y*) and to describe attributes about objects (i.e., object *x* has attribute *z*).

This knowledge representation has the capability to describe a spatial dataset using graphs, allowing a graph-based mining tool to mine it as a whole. The capabilities of the model to represent the relation between these objects will be of great impact in the results of the data mining processes, since the world is described by objects and the relation between these objects, we can figure out the relations as the elements describing the interaction of the objects with each other.

3.3 Spatial Graph-based Data Representations

In the construction of the graph there are issues such as the graph complexity and size that have a direct impact over the data mining algorithm performance. The quality of results refers to another important aspect. Testing several representations will allow us to produce comparisons among obtained results, to evaluate them, and finally to make a decision for selecting the one(s) which offers the better results according to our criteria for success. One approach is to show that the model allows discovering known patterns. Another approach is to have a domain expert saying that the discovered patterns are interesting. We have worked with domain experts for developing these tasks.

Currently, we have developed five models to represent spatial data, non-spatial data and spatial relationships among the spatial objects as a unique dataset from the general model. Three issues define the characteristics (i.e., number of vertices and edges, simple graph, etc.) of the graphs created from these models:

- Equivalent spatial relations.

- Symmetric spatial relations.
- The way to represent objects and their relations in the model.

In the following subsections we will explain how these three characteristics affect the structure and composition of the graph. First, we will talk about the equivalent relations, next the symmetric relations and finally the five created models.

Equivalent Spatial Relations

Suppose that our dataset is composed of an object *A* disjoint of an object *B*, this implies that object *B* is disjoint of object *A*. In this case the two objects are disjoint each other (an equivalent relation). When creating the graph, this relation can be represented by two directed edges, one edge labeled as "DISJOINT" from object *A* to object *B* and vice versa. However we can use the following principle:

2 directed edges, 1 edge $e(v_i,v_j)$ and 1 edge $e(v_j,v_i)$	equal to	1 undirected edge $e = ij$

By applying this principle we can replace the two directed edges by only one undirected edge labeled as the equivalent relation without losing the representation of the spatial relation among the objects.

The equivalent relations implemented in our research are the following:

- TOUCH
- DISJOINT

- OVERLAP
- EQUAL
- CLOSE

Symmetric Spatial Relations

Suppose that our dataset is composed of an object *A* South of an object *B*, when we create the graph this relation is represented by a directed edge from object *A* to object *B* labeled as "South_of". But it implies the relation object *B* is North of object *A* (it is a symmetric relation). This last relation in some models is not represented.

According to the model the representation of a symmetric relation implies one of the following options:

- The addition of a directed edge labeled as the symmetric relation.
- The addition of a vertex labeled as the spatial relation (i.e., topological, direction) and a directed edge labeled as the symmetric relation.

The symmetric relations implemented in our research are the following:

- CONTAINS ←→ INSIDE
- COVERS ←→ COVEREDBY
- NORTH_OF ←→ SOUTH_OF
- EAST_OF ←→ WEST_OF
- NORTHEAST_OF ←→ SOUTHWEST_OF
- SOUTHEAST_OF ←→ NORTHWEST_OF

Models

As we have commented, testing several representations will allow us to produce comparisons among obtained results, to evaluate them, and finally to make a decision for selecting the one(s) which offers the better results (in term of quality). The following five models were developed to answer issues such as: in some models we obtain a reduction in the number of vertices and edges, but do they give us the same representation of our data? Are we gaining in the size reduction, but what are we loosing? There is a higher complexity working with general graphs than with simple graphs, does it affect the generated results? What about time processing?

In order to describe the characteristics of each model we will use the sample dataset shown in Figure 3.2. Our dataset is composed of two spatial objects, object *A* representing a *house* and object *B* representing a *lake*, and the following three spatial relations among them:

- Distance relation
    - Object *A* close object *B* (equivalent relation).
- Topological relation
    - Object *A* touch object *B* (equivalent relation).
- Direction relation
    - Object *A* South of object *B* (symmetric relation).

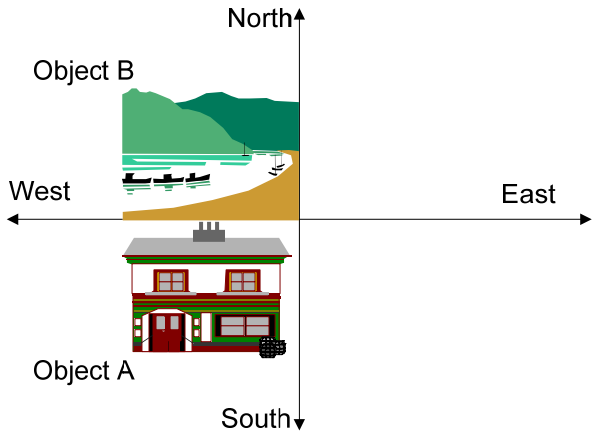

Figure 3.2. Sample dataset.

Additionally, to evaluate the characteristics of each model (the model is itself a graph) we have developed the following nine evaluation metrics:

- **Num. vertices.** Total number of vertices in the graph.
- **Num edges.** Total number of edges in the graph.
- **Size (vertices + edges).** Total number of vertices plus total number of edges in the graph.
- **% increment.** This item represents the percent of increment (in term of vertices plus edges) of this graph with respect to the graph created by using the base model.
- **Simple graph.** To indicate if the graph is a simple one (graph with at most one edge between any given pair of vertices and with no loops).
- **Directed edge.** To indicate if the graph has directed edges.
- **Undirected edge.** To indicate if the graph has undirected edges.
- **Complete information.** The item shows if the symmetric relations, created from the original spatial relations, are also represented in the graph.

- **Redundant "Relation" edge.** We introduce this edge as strategy for avoiding creating complex graphs. Remember that we consider simple graph a graph with at most one edge between any given pair of vertices and with no loops. As we will see in the description of those models, we use this special edge to join the vertices representing spatial object with the vertices representing the spatial relation types (i.e., topological, distance and direction). In models #3, #4 and #5 we add a "Relation" edge for representing explicitly the fact that there is a relation between two spatial objects. This metrics tells us if a redundant "Relation" edge is presented in the graph.

Model #1 - base model

Figure 3.3 shows the first model created for representing spatial data as proposed. The characteristics of the model according to the metrics are:

- **Num. vertices:** 2 vertices for representing each spatial object (i.e., object *A*, and object *B*).
- **Num. edges:** 4 edges, 3 edges for representing the original spatial relations (i.e., "close", "touch", and "South_of" relations) and 1 edge for representing the "North_of" relation created from the original "South_of" symmetric relation. The "North_of" relation is itself a symmetric relation.
- **Size (vertices + edges):** 6
- **% increment:** 0%, it is the *base model*.
- **Simple graph.** No, it is a complex graphs with 4 edges linking 2 vertices.

- **Directed edge.** Yes, they are used for representing the "South_of" and "North_of" symmetric relations. The direction of the edges is according to the lecture of the relations among the objects.
- **Undirected edge.** Yes, they are used for representing the "close" and "touch" equivalent relations.
- **Complete information.** Yes, in the graph is represented the "North_of" symmetric relation created from the original "South_of" symmetric relation.
- **Redundant "Relation" edge.** No, in the model we do not use "Relation" edges.

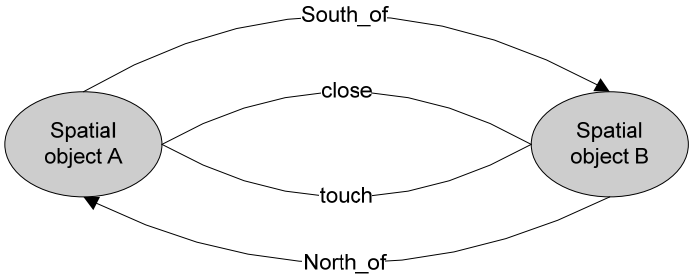

Figure 3.3. Model #1 - base model.

Model #2 - single replication of relation types, complete information

In Figure 3.4 we show our second model created for representing spatial data. The characteristics of the model according to the metrics are:

- **Num. vertices:** 5 vertices, 2 vertices for representing the spatial objects and 3 vertices for representing the "topological", "distance", and "direction" spatial relation types. For each spatial relation type among two spatial objects we add 1 vertex labeled as its name. In the example there exist 1 "topological" relation, 1 "distance" relation, and 1 "direction" relation.

- **Num. edges:** 6 edges, 3 edges for representing the original spatial relations, 2 edges for representing the equivalent relations (i.e., "close" and "touch" relations) created from the original ones, and 1 edge for representing the symmetric relation (i.e., "North_of" relation) created also from the original relations.
- **Size (vertices + edges):** 11
- **% increment:** +83.33%
- **Simple graph.** Yes, there exists at most 1 edge between any given pair of vertices.
- **Directed edge.** Yes, they are used for representing all relations. The direction of the edges is from the vertices representing the spatial objects to the vertices representing the spatial relation types.
- **Undirected edge.** No, in the model we do not use undirected edges.
- **Complete information.** Yes, we represent the symmetric relations created from the original ones.
- **Redundant "Relation" edge.** No, in the model we do not use "Relation" edges.

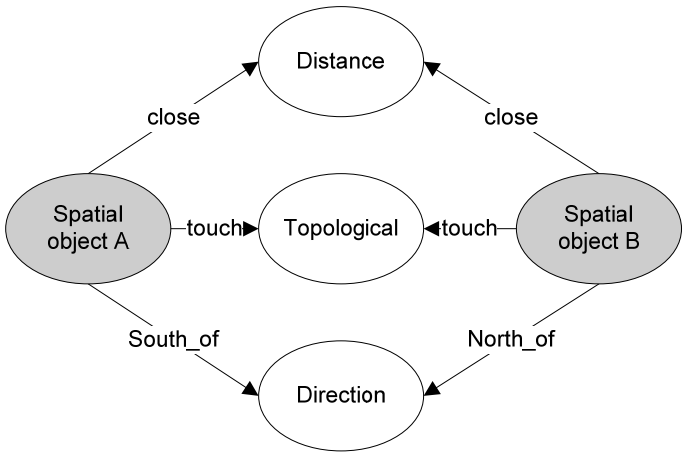

Figure 3.4. Model #2 - single replication of relation types, complete information.

Model #3 - double replication of relation types, no complete information

In Figure 3.5 we show our third model created for representing spatial data. The characteristics of the model according to the metrics are:

- **Num. vertices:** 8 vertices, 2 vertices for representing the spatial objects and 6 vertices for representing the "topological", "distance", and "direction" spatial relation types. For each spatial relation type among two spatial objects we add 2 vertices labeled as its name. For instance, in the example there exist three relations: 1 "topological" relation, 1 "distance" relation, and 1 "direction" relation, so we add 6 vertices, 2 per each spatial relation type.
- **Num. edges:** 9 edges, 6 edges (the "Relation" edges) to link the vertices representing the spatial objects to the vertices representing the spatial relation types (from each vertex representing a spatial object start 3 edges since we have 3 relations), and 3 edges for representing the original spatial relations. These last 3 edges are used to join the vertices representing the spatial relation types.
- **Size (vertices + edges):** 17
- **% increment:** +183.33%
- **Simple graph.** Yes, there exists at most 1 edge between any given pair of vertices.
- **Directed edge.** Yes, they are used for representing the symmetric relations and the "Relation" edges. The direction of the "Relation" edges is from the vertices representing the spatial objects to the vertices representing the spatial relation types. The direction of the other edges is according to the lecture of the relations among the objects.
- **Undirected edge.** Yes, they are used for representing the equivalent relations.

- **Complete information.** No, we do not represent the symmetric relations created from the original ones.
- **Redundant "Relation" edge.** Yes, we use in the model "Relation" edges for representing explicitly the existence and type of a relation among 2 spatial objects. Additionally, we use this edge for avoiding creating complex graphs.

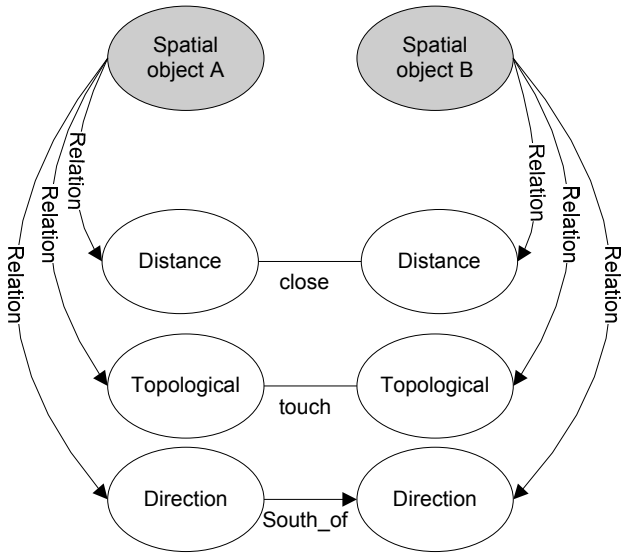

Figure 3.5. Model #3 - double replication of relation types, no complete information.

Model #4 - single replication of relation types, no complete information

In Figure 3.6 we show our fourth model created for representing spatial data. The characteristics of the model according to the metrics are:

- **Num. vertices:** 5 vertices, 2 vertices for representing the spatial objects and 3 vertices for representing the "topological", "distance", and "direction" spatial relation types. For each spatial relation type among two spatial objects we add 1 vertex labeled as its name. In the example there exist 1 "topological" relation, 1 "distance" relation, and 1 "direction" relation.

- **Num. edges:** 6 edges, 3 edges (the "Relation" edges) to link a vertex representing a spatial object (in the example we have 2 vertices since there are 2 spatial objects) to the vertices representing the spatial relation types (from this vertex representing a spatial object start 3 edges since we have 3 relation types), and 3 edges for representing the original spatial relations. These last 3 edges are used to link the vertices representing the spatial relation types to the un-used vertex representing the other spatial object.
- **Size (vertices + edges):** 11
- **% increment:** +183.33%
- **Simple graph.** Yes, there exists at most 1 edge between any given pair of vertices.
- **Directed edge.** Yes, they are used for representing the symmetric relations and "Relation" edges. The direction of the "Relation" edges is from a vertex representing a spatial object to the vertices representing the spatial relation types. The direction of the other edges is from the vertices representing the spatial relation types to the un-used vertex representing the other spatial object (it is according to the lecture of the relations among the objects).
- **Undirected edge.** Yes, they are used for representing the equivalent relations.
- **Complete information.** No, we do not represent the symmetric relations created from the original ones.
- **Redundant "Relation" edge.** Yes, we use in the model "Relation" edges for representing explicitly the existence and type of a relation among 2 spatial objects. Additionally, we use this edge for avoiding creating complex graphs.

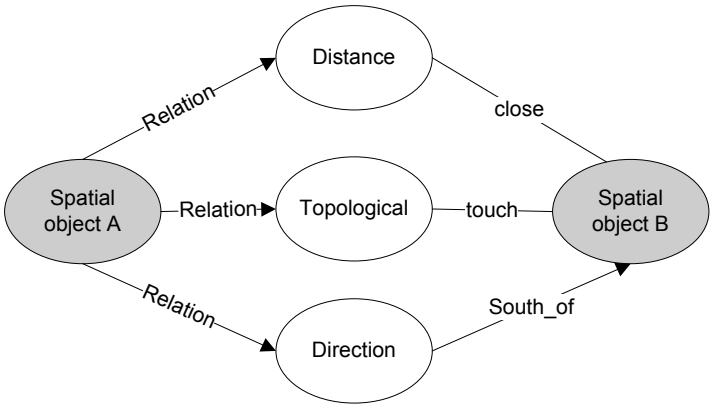

Figure 3.6. Model #4 - single replication of relation types, no complete information.

Model #5 - double replication of relation types, complete information

In Figure 3.7 we show our fifth model created for representing spatial data. The characteristics of the model according to the metrics are:

- **Num. vertices:** 8 vertices, 2 vertices for representing the spatial objects and 6 vertices for representing the "distance", "topological", and "direction" spatial relation types. For each spatial relation type we add 2 vertices labeled as its name. In the example we add 2 vertices for the "topological" relation, 2 vertices for the "distance" relation, and 2 vertices for the "direction" relation.
- **Num. edges:** 12 edges, 6 edges (the "Relation" edges) to link the vertices representing the spatial objects to the vertices representing the spatial relation types, and 6 edges for representing the original spatial relations and those ones generated from them (equivalent and symmetric relations). These last 6 edges are used to link the vertices representing the spatial relation types to the vertices representing each spatial object (3 edges for each spatial object).
- **Size (vertices + edges):** 20

- **% increment:** +233.33%
- **Simple graph.** Yes, there exists at most 1 edge between any given pair of vertices.
- **Directed edge.** Yes, they are used for representing all relations.
- **Undirected edge.** No, in the model we do not use undirected edges.
- **Complete information.** Yes, we represent the symmetric relations created from the original ones.
- **Redundant "Relation" edge.** Yes, we use in the model "Relation" edges for representing explicitly the existence and type of a relation among 2 objects. Additionally, we use this edge for avoiding creating complex graphs.

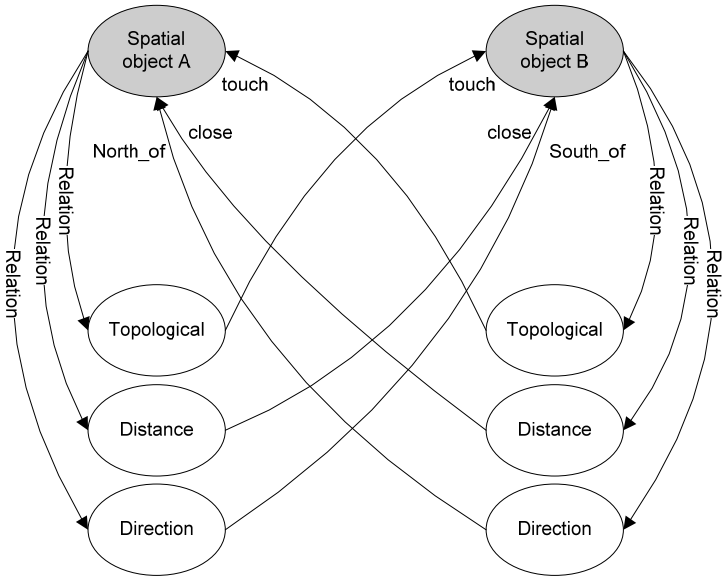

Figure 3.7. Model #5 - double replication of relation types, complete information.

Table 3.1 presents the results of the nine metrics developed to evaluate the characteristics of each model. Model #1 is named the base model.

Model	Num. Vertices (1)	Num. Edges (2)	Size (v + e) (3)	% Increment (4)	Simple Graph (5)	Directed Edge (6)	Undirected Edge (7)	Complete Information (8)	Redundant "Relation" Edge (9)
#1	2	4	6	-	No	Yes	Yes	Yes	No
#2	5	6	11	+83.33	Yes	Yes	No	Yes	No
#3	8	9	17	+183.33	Yes	Yes	Yes	No	Yes
#4	5	6	11	+83.33	Yes	Yes	Yes	No	Yes
#5	8	12	20	+233.33	Yes	Yes	No	Yes	Yes

Table 3.1. Characteristics of the graph-based representation models.

The metrics were proposed based on the causes/effects each of them has both into the created graph and the mining algorithm. We visualize four significant issues related directly to these metrics:

1. Search space

The search space of a graph-based data mining algorithm consists of all the subgraphs that can be derived from its input graph, thus, the number of vertices (1) and edges (2) of the created graph (3) are two important issues defining the search space size[1] for the discovery system. Therefore, the objective must be to minimize the number of vertices and edges used to create the graphs but at the same time to maximize the representativeness of the dataset. As we can see in Table 3.1, the model using the minimum number of vertices and

[1] The edge type (directed or undirected) and graph type (simple or complex) are issues that also define the search space of a graph-based data mining algorithm. In Chapter 4 we describe the Subdue system, our graph-based data mining tool, and we show the way Subdue deals with these issues.

edges to represent our sample dataset is model #1 (2 vertices and 4 edges) whereas model #5 is the opposite case (8 vertices and 12 edges).

2. Processing time

The search space size plays a relevant role regarding to the processing time used to discover patterns. If we have a large search space the algorithm would require more time to evaluate all the possible subgraphs. Therefore, a comparison among the "percentage of increment" (4) metric of the proposed models is presented in Table 3.1. Remember this metric compares the size of a given model with respect to model #1 (base model). For instance, model #5 has a graph size increment of 233.33% with respect to model #1. In other words, the mining algorithm will require the evaluation of 233.33% more vertices and/or edges by using model #5 instead of model #1 for the same dataset.

3. Graph complexity

Next chapter describes the Subdue system, our graph-based data mining tool. As we will see there exists a higher complexity for the mining algorithm to work with complex graphs than with simple ones (i.e., at most an edge among any given pair of vertices and with no loops). For instance, in the graph match process, in the expanding phase (Subdue uses an "expanding" approach to discover patterns), and in the graph compression stage. Therefore, the objective was to propose graph-based models that allow us to create simple graphs. As we can see in Table 3.1, only model #1 does not create simple graphs.

Thus, as a strategy to break the multiplicity of edges among two vertices (for instance, vertices representing two spatial objects meeting two or more spatial relations among them) we use the following approaches:

- To add a new vertex labeled as the relationship type (i.e., topological, distance, and direction) for each spatial relation among the spatial objects. This approach is used in model #2.
- To add a new vertex (model #4) or two new vertices (model #3 and model #5) labeled as the relationship type (i.e., topological, distance, and direction) by each spatial relation among the spatial objects, and to link this new vertex (model #4) or new vertices (model #3 and model #5) with the vertices representing the spatial objects by using edges labeled as "Relation". The approach used to link the vertices is different in each model as we have mentioned in the definition of each of them. This nomenclature is used to represent the fact that there exists a spatial relation among these spatial objects. These edges are known as redundant relation edges (9).

4. Data representativeness

The directed edges (6), undirected edges (7), and complete information (8) metrics are used to maximize the representativeness of the dataset and minimizing, as most as possible, the graph size and its complexity. Directed edges are used to represent the symmetric spatial relations (object *A* North_of object *B,* implies, *B* South_of *A*), the redundant relation edges, and the non-spatial attributes describing the spatial objects. Undirected edges are used to represent equivalent spatial relations (the relation is represented by an undirected edge

instead of two directed edges). Finally, complete information means that symmetric spatial relations among spatial objects are also represented into the model.

3.4 Use-case

To show the applicability of our model, we will use a dataset from the Popocatépetl volcano database [38, 42]. The database contains data related to several issues in the zone such as settlements, rivers, and evacuation roads in the zone, just to mention a few of them.

Figure 3.8 shows a fragment of the layers “roads” and “settlements” of the Popocatépetl volcano zone. The roads layer (shown in green color) represents the roads in the zone; it is composed of spatial objects (i.e., lines) and non-spatial data describing the characteristics of those roads (i.e., id, start point, end point, length, and type). The settlements layer (shown in pink color) represents population areas in the zone. The layer is composed of spatial objects (i.e., polygons) and non-spatial data describing the characteristics of the settlements (i.e., id, area, perimeter, and type).

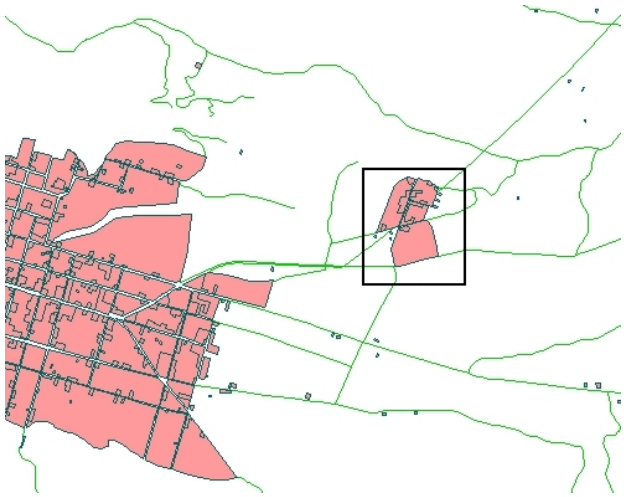

Figure 3.8. Spatial database representing some objects of the world.

Suppose we are interested in the identification of relations among the roads starting in, or crossing a settlement (i.e., the type and characteristics of roads in relation to the number of people living in a settlement that in case of a volcanic contingency are required to be evacuated). So, we need to create a dataset involving these elements for mining it and search for patterns that could help us to evaluate the characteristics of the roads and, may be, to make decisions for improving them (or build new ones) for their utilization in case of volcano activity. In this case we are working with different types of spatial objects and also we are adding non-spatial data and relationships (as touch or overlap) to our dataset.

The construction of the dataset needs to satisfy some constraints. For example, if we include all the elements existing in the data layers we might build a huge graph, and this will have a direct impact in the data mining algorithm. So, we explore methodologies to deal with topics such as complexity, the size of the graph, noise and quality of the data. A solution for the problem of creating a huge graph is to limit the set of elements to be

included in it by using *selection windows*. The user can create these windows and only the elements inside them are candidates to be included as objects in the graph. The idea is shown in Figure 3.9. Suppose the user will work with *n* data layers, thus, for each layer we will select only the spatial objects inside the area delimited by the *selection window*. We can see this functionality such as a drill operation over all the spatial layers the user works with.

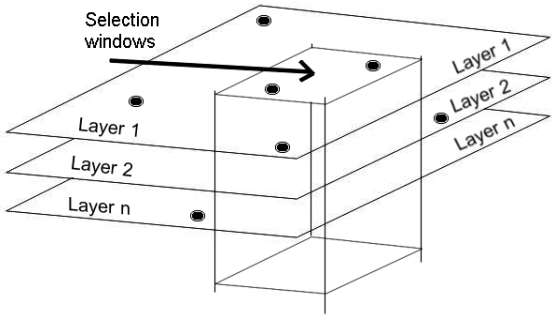

Figure 3.9. Selection window.

Once we have defined the working area, as shown in Figure 3.8, we can build the graph. The process consists of three phases. In the first phase, the user has to choose the spatial relation(s) to evaluate among the spatial objects (in the example the touch or overlap topological relations). The second phase involves the validation process, only the objects covered by the relation(s) become elements to be included in the graph.

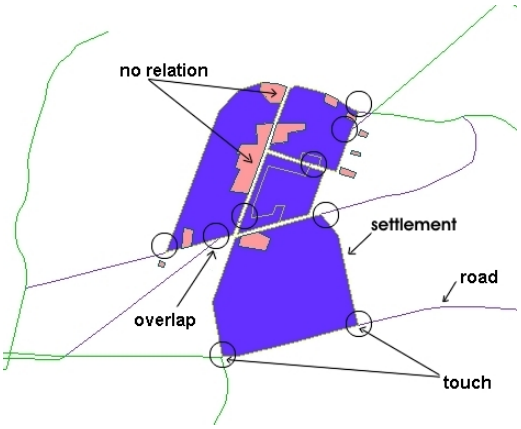

Figure 3.10. Querying a spatial database.

The last phase consists of building the graph using the results of the validation process. Figure 3.10 presents an example of this functionality. This time only the area delimited by a *selection window* is shown. In the figure, the spatial objects satisfying either the touch or overlap relations are shown in blue color, the rest of the objects are shown in their original colors. The circles show some examples where a road is starting or crossing a settlement.

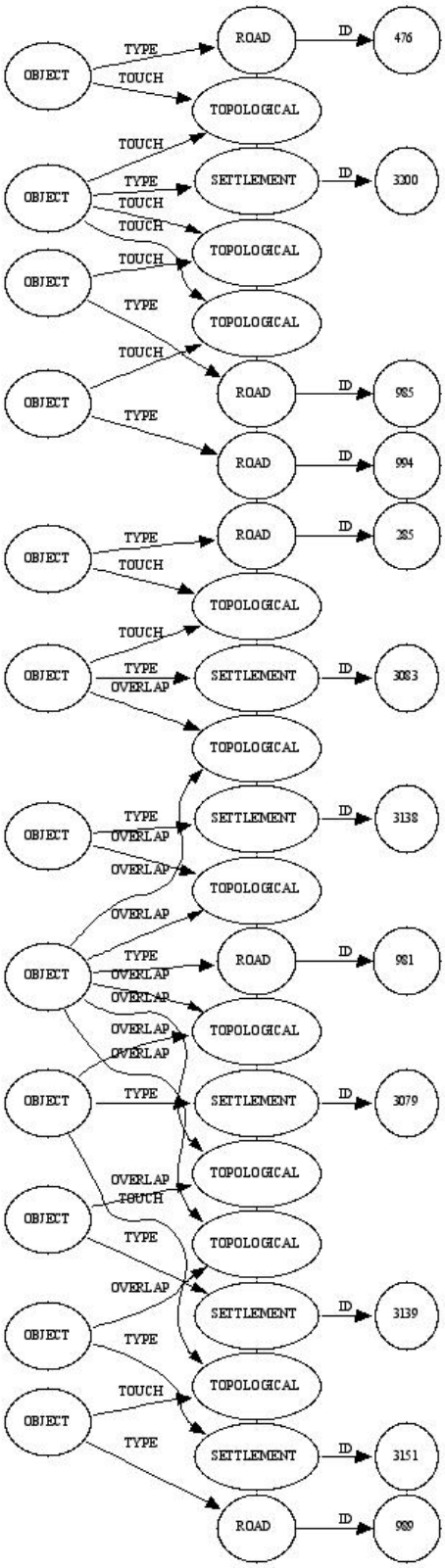

Figure 3.11. Graph-based representation for spatial data.

Figure 3.11 shows the graph created using model #2 (we used the Graphviz System [19] to draw the graph). The vertices in the graph represent either spatial objects (i.e., road, settlement), spatial relations between two objects (i.e., topological), or attributes describing the objects (i.e., ID value). In this example, we use a special vertex labeled as "object" for expressing the interconnection between the spatial objects, their spatial relations, and non-spatial attributes (i.e., object type road with ID 989 overlaps or touches with object…).

According to the type of vertices that an edge joins, the edge can be labeled as either a spatial relation name (i.e., overlap, touch), or as the name of an attribute describing the characteristics of an object (i.e., type, ID).

We may read the graph as follows: there are six roads and six settlements inside the working area meeting either a touch or overlap relation in the form road → settlement. We suppose that each polygon object in the map represents a settlement. Some roads start from a settlement and others cross a settlement. For example, the road with ID *981* crosses the settlements with ID's *3079*, *3139*, *3151*, *3083* and *3138*. This means that we need to be careful with this road since it has interaction with several settlements and in case of a contingency it will be widely used. In the graph we included only the object's ID non-spatial attribute for each object.

This is a simple example; in the real world the spatial data layers may have hundreds or thousands of objects, and each object may have dozens of attributes describing it. Joining all these elements will allow creating large graphs representing the elements found in a

spatial database improving the results of the data mining task. Once we have created the graph, it will be used as input to a graph-based mining system.

3.5 Conclusion

Our idea is to propose a graph-based model to represent together spatial data, non-spatial data and the spatial relations between spatial objects. Based on the model we generate datasets composed of graphs with a set of these three elements. Our argumentation is that by mining a dataset with these characteristics a data mining algorithm can search patterns involving all these elements at the same time improving the results of the spatial analysis process.

We have presented an example of the applicability of the proposal using data from a Popocatépetl volcano database. The created graph will be the input for a graph-based mining system. We propose to use the Subdue system, which will be described in the next chapter.