CHAPTER-5
SELF-ADAPTIVE MODELING FOR SPATIAL DATA MINING
CHAPTER 5

SELF-ADAPTIVE MODELING FOR SPATIAL DATA MINING

5.1 Introduction

Geography is an integrative discipline and the geographic data under investigation often span transversely in multiple domains. The complexity of geographic problems and spatial data together with inherent spatial relationships, comprise an enormous confront to conservative data mining schemes and call for both development of new techniques and theoretical research to assist in obtains information from huge and heterogeneous spatial datasets [9][10]. Due to outsized heterogeneity of spatial data, suppliers of geographic data specify diverse models for the same spatial objects.

The Context Specific Semantics (CSS) is one of the best approaches recommended that deals with stipulation of feature space cradles. An Ontological analysis needs to be done on the fundamentals of the domain space. Spatial data mining becomes further interesting and significant as huge spatial data have been mounted up in spatial databases [3][4]. Mining spatial collocation patterns is an important spatial data-mining task with broad applications.

The software architecture is proposed in this chapter for the application. The self-adaptive software concept added to the architecture which is already discussed in the previous chapter. Self-adaptability concept is very important here because as number of disaster cases (epidemics, cyclones, earthquakes etc) is more it can afford to build separate systems to every case, the problem can be solved by using the self adaptability concept.

Software Engineering aims for the systematic, principled design and deployment of applications that fulfil software’s original promise—applications that retain full plasticity throughout their lifecycle and that are as easy to modify in the field as they are on the drawing board. Software engineers have pursued many techniques for achieving this goal: specification languages, high level programming languages, and object-oriented analysis and design, to name just a few. However, while each contributes to the goal, the sum total still falls short.

Self-adaptive software is aimed at addressing the challenges of constructing systems that autonomously respond to a variety of situations. Our goal is to design an
architecture–centric and knowledge–based approach in which adaptive behaviour is achieved through observations called collocation rules.

“Collocation is a derivation of several typematic features of spatial objects”. A general mathematical definition can be given to the collocation as just a variable therefore possessing more than one quality that belongs to a spatial object. However, the spatial object is a finite set of features, except that the features change when there is a major/great change in the spatial domain that is engulfed in the spatial object. The change of features does not affect much the collocation, only except there is a great phenomenal change. The collocation will provide all the necessary feature set required for the expression of the spatial rule [5][6][31][32]. The spatial rule defines the core of knowledge patterns in the spatial domain. The knowledge patterns can be applied to mine the quality of a spatial domain. The spatial rule carves up into spatial association rule and collocation rule. A seamless definition of a collocation rule is a set of features that belongs to a set of spatial objects prevailing in the spatial domain that describe the knowledge pattern.

5.2 Methodology

Detection of the Epidemic

The collocation rules are very useful in detecting the affected areas by finding the symptoms of a disease. The collocation rule C, i.e.

\[ C: \text{cause of epidemic} \rightarrow \text{causative agent, infection sources; in the nearby region with high probability.} \]

This typical confident collocation rule involves with both frequent and rare events because although infection is quite common and epidemic is rare the later factor implies the former one strongly.

As discussed in Law of Total Probability and Problem, by using sample identifiers, the collocation can be explained as follows:

Assuming firstly, the ‘b’ as the consequence of feature ‘a’ is developed, forms a first level of collocation, which is identified by \( a \rightarrow b \), secondly, if the consequence ‘c’ from the feature ‘b’ is developed, forms a collocation, which is identified by \( b \rightarrow c \). As ‘b’ already have an antecedent ‘a’ the consolidated version of collocation \( \{a, b\} \rightarrow c \) can be formed. If ‘c’ becomes another feature that can lead to the consequence of ‘d’, then the notation wholly represents the cause of ‘d’ as \( \{a, b, c\} \rightarrow d \). Also implies to \( \{a \cup b \cup c\} \rightarrow d \) representation.
Similarly, considering the collocation pattern for the problem:

\( C: \{\text{cause of epidemic}\} \rightarrow \{\text{causative agent, infection sources}\}; \text{in the nearby region with high probability.} \)

The collocation pattern is considered with practically proved parameters for dengue as follows …

\( C1: \{ \text{Fever, headache, Myalgia or bone pains, skin rash, nausea, Vomiting} \} \rightarrow \{ \text{Bleeding, low-levels of blood platelets, Low BP (dengue shock)} \} \)

\( C2: \{ \text{c1, Bleeding, Low-levels of blood platelets, Low BP} \} \rightarrow \{ \text{Fluid accumulation in chest & abdominal cavity leakage} \} \)

\( C3: \{ \text{C1, C2, fluid accumulation in chest & abdominal cavity leakage} \} \rightarrow \{ \text{Depletion of fluid from circulation & decreased blood supply to vital organs} \} \)

\( C4: \{ \text{C1, C2, C3, decreased blood supply to vital} \} \rightarrow \{ \text{Dengue Shock Syndrome} \} \)

\( C5: \{ \text{X, infected mosquitoes or monkeys, DENV virus} \} \rightarrow \{ \text{Dengue Shock Syndrome} \} \)

As the disease reaches “Collapse” stage, the circulation is almost completely arrested, accelerated respiration, weak pulse, decreased systolic blood pressure, diminished or no urine output.
Assuming $X$ as defined representation of collocated sequence of patterns i.e., $C_1, C_2, C_3, C_4$ the resultant collocation $C_5$ is determined [33].

That participation ratio describes the intensities of the symptoms that play important role to form the collocation rule and builds the reference feature. The general syntax for assessing the reference feature is $Pr(C, f)$.

If the probabilities of some features with respect to $C_1$ are understood is having the maximum and minimum. If the lead feature of the collocation contains least probability then collocation is considered as feebly important. If the lead feature of the collocation contains higher probability then collocation is considered as highly important. The probabilities mentioned in the problem are $\langle$excreted along with innumerable virions$\rangle$, $\langle$fever, headache, muscle and joint pains, skin rash, bleeding, low levels of blood platelets$\rangle$. If one of them or some of them exhibit high probability, then there is a high significance of occurring the disease severely, for low exhibition of probability, the existing of the disease will be indicative. However, the features and the probabilities considered will prove the collocation to be appropriate for the causation of severity of dengue spectrum (simple dengue to dengue death).

*Even though the participation index of the whole pattern could be low, there must be some spatial feature(s) with high participation ratio(s).*

5.3 An Abstract Framework

The built of framework explains the elements of the spatial knowledge support system in a work flow strategy and component architecture strategy [34][35][36][37]. The framework in work flow strategy is explained using conceptual process planning. The conceptual process planning, based on the need of the mining of collocation patterns and help provided to the health campaigners, is briefed into conceptual design and detail process design.

The following figure 5.1 describes the conceptual design of the work flow strategy.

![Figure 5.1 Conceptual Design of the Work Flow Strategy](image-url)
The conceptual design contains two important components, the spatial data-mining infrastructure and the health campaign framework. The former defines all the necessary tools for gathering the spatial data from the spatial data warehouses. The later is defined into two classes, the software tools (developed), that defines all the necessary data base management tools that store and manage the spatial data and that which are required to transform the original geo-spatial data into the schema objects of the databases like tables, etc.

The elements of detail process design are related to the components of conceptual design. Acquiring the spatial map in a database representation with demographic data, a learned database of various sicknesses caused due to epidemics, re-learning mechanism to derive mappings of new patterns with learned patterns already in the database, an antecedent-consequent based analysis, spatial rule generation are associated with the first component of conceptual design framework.

The rule application and the health-campaign framework are associated with the second component of the conceptual design framework. The following figure 5.2 shows the detail process design framework of work flow strategy.

The collocation pattern formed by this sample region acts as a cautious measure or the forecast for the bio-medical researchers, analysts and other health-care-takers of the spatial zone which will be useful for them to take suitable remedial campaigns. The boundaries of the framework are limited to design the semantic elements of the spatial knowledge support system. The detail process design in figure 5.3 explains the detailed functional decomposition of the components.

![Figure 5.2 Process Design framework](image-url)
Further by analyzing the degree of relevance of the parameters given in the derivation of the Participation Index gives the basic idea of the virulent-bacterial-causative roots that develop the epidemic. Based on the parametric values, the quantitative analysis on participation index can be used to prove various alternatives of the input parameters and the seriousness of the virulent-bacterial-causative sources.

Socio-statistical methods related to health-science can be implemented to regulate the input variables that play a parametric role of collocation rule formation, in order to prevent the epidemic in the spatial zone, if not permanently, at least suitable preventive measures can be undertaken for the affect of such candidate epidemic in the interested spatial zone.

5.4 Spatial Data Mining Collocation Patterns

Mining collocation patterns provides the standard of watching the generic features of a given spatial region with more related Boolean characters with their \( c \) (confidence) and \( s \) (support) [17][18]. The job of mining collocation moulds into spatial statistics and combinatorial approaches. The spatial collocation pattern-mining framework presented in the previous mechanisms has bias on admired events. It may overlook some extremely confident but “infrequent” collocation rules with only “support”-based trimming. Let \( F = \{f_1, ..., f_k\} \) be a set of Boolean spatial features in a spatial database \( S \), and Let \( I = \{i_1, ..., i_n\} \) be a set of \( n \) instances in the spatial database \( S \), here each instance is a vector containing of [instance-id, location, spatial features].

---

**Figure 5.3 Component Diagram**

The diagram illustrates the spatial data mining processes, including feature collection, learning & (re)learning, data store, data set, collocation rules, collocation rule report, and spatial data warehouse and control. The Health Campaign Framework is also depicted, showing the integration of health data in the mining process.
Neighbourhood relation $R$ over pair wise locations in $S$ exists ~ is believed. To find rules in the form of $A \cup B$, where $A$, $B$ are subsets of spatial features is the object of this collocation rule mining. $A$ resolves the set of spatial features which form the precursor part of the rule and $B$ describes the action and its substantial parts the support and the confidence. The rule signifies the concurrence of the spatial collocation rule soaks up the exploit of the rule in the “nearby” zones of the spatial objects that obey with the collocation rule. To capture the concept of the predicate “nearby”, the concept of neighbour-set $L$ is a set of instances such that all pair wise locations in $L$ are neighbours. Neighbourhood relation $R$ may be defined based on Euclidean distance and neighbouring instances are linked by edges. The set of spatial features are in a collocation pattern $C$, i.e., $C \subseteq F$. A neighbour-set $N$ is said to be a row occurrence of collocation pattern $C$ if each feature in $C$ emerges in an instance of $N$, and there is no proper subset of $N$. Then all row instances of a $C$ (collocation pattern) are denoted as $row \ set(C)$. In other words, $row \ set(C)$ is the set of neighbour-sets where spatial features in collocation $C$. The conditional possibility is the probability that the neighbour-set in $row \ set(A)$ is a part of a neighbour-set in $row \ set(A \cup B)$. Intuitively, the conditional probability $p$ indicates that, the incidences of the spatial features in $A$, the possibility to identify the incidence of $B$ in a nearby region is $p [1][2]$. 

Example: In the invented, figure 5.4, the landscape describes two significant spatial marks, lake and sea. The mosses and lichens at the western zone of the lake distress the water, as most area of the water is sluggish and enclosed by marsh. The people at this zone utilizing the water resources may be affected by so various kinds of fecal blemish in food and water. The sea water is spoiled with the heavy salts and base products and the crude oil, so the people in this area and the crude oil, so the people in
this area cannot utilize this water for domestic purposes, the climatic changes are affected by the water contents in the shores of sea. People breading their lives at the shores will have the indirect contamination of fecal material in the water and as well as in the form of moisture in the air. As the lake water is supplied into the agriculture lands surrounding in the adjacent zones, there people may be affected indirectly by the virulent characteristics.

Form of Collocation Rule for Epidemics

The term Collocation stands for features located together in different instances. The collocation rules are extremely useful in noticing the affected areas by verdict the indications of a disease. The collocation rule C, i.e.,

\[
C : \{ \text{epidemic cause} \} \rightarrow \{ \text{infection sources, causative agent} \} \text{ in the"nearby" zone with high possibility.}
\]

This typical confident co-location rule involves with both frequent and rare events because although infection is quite common and epidemic is rare the later factor implies the former one strongly. assume initially, the ‘b’ as the corollary of feature ‘a’ is generated, which makes the first level of collocation, this is identified by \(a \rightarrow b\), next, if the corollary ‘c’ from the feature ‘b’ is generated, makes a collocation, which is recognized by \(b \rightarrow c\). As ‘b’ previously have an antecedent ‘a’, the combined version of collocation, \(\{a, b\} \rightarrow c\) can be made. If ‘c’ becomes other feature that can guide to the corollary of ‘d’, then the details wholly represent the source of ‘d’ as \(\{a, b, c\} \rightarrow d\), also entails to \(\{a \cup b \cup c\} \rightarrow d\) representation[38].

5.5 Implementation Strategy

Software Architecture is an emerging important discipline [39][40][41][42]. Dynamically modifiable at any point in system lifecycle can be achieved by knowledge-based approach in which adaptive behaviour is achieved through observations [43][44]. An architecture-based self-adaptable system for various types of disaster is proposed. Self-adaptive systems present a unique set of challenges with respect to safety, reliability and correctness. Components are responsible for implementing application behaviour and maintaining state information.

In case of epidemic disaster, the losses of related object in spatial are men, money, and animal. A considerable amount of populated area becomes an empty (no men’s land) due to the migration of people from epidemic region to other locations.
This system is also useful for damage surveys as earthquake, hurricane etc., provided some one could adapt the related software component quickly enough [45][46][47]. The collocation rules that are proposed here is to predict the future disaster with sufficient assurance that the system would perform as intended. The collocation rules for various types of disasters are being supported as self-adaptable or commodity components. The system supports an open adaptive behaviour and adaptation plans that can be introduced during runtime. Adaptation mechanism depends on change, Change management is a controlling component that identifies change, reasoning content for change, specifying and implementing change, preserves system integrity and risks create by runtime modification. Changes can include the addition, removal or replacement of components and connectors [48][49]. The next section discusses about the epidemic disaster management system (EDMS) for particular epidemic disaster, and then discusses a common architecture called Disaster Management System (DMS) for various disasters.

5.6 Epidemic Disaster Management System (EDMS)

The Epidemic Disaster Management System (EDMS) that proposes here is to manage the epidemic disaster and it is based on the spatial collocation pattern-mining model. This system contains four-components. First component is for collecting information related to epidemics from spatial knowledge using the collocation rule based algorithm. Second component is a conventional database system, which is responsible for populating the tables with the data. The third component is to mine collocation (Data Mining). The fourth component is Database of Spatial knowledge, fifth component is for knowledge processing, and the sixth component has a report generation component and guidelines for the field staff. The field staff implements the results produced by EDMS as shown in Figures 5.5 & 5.6 [50][51][52][53].
Component based software development method is based on the design to develop software structure by selecting suitable off-the-shelf components and afterwards to accumulate them with well definite software architecture. So the latest software development hypothesis is much dissimilar from the conventional approach.

One of the most promising solutions today is the component based software development approach. These components can be developed by different developers. Component based software development (CBSD) can significantly reduce development cost and time to market, and improve maintainability, reliability and overall quality of the software systems.

![Figure 5.6 Simple Pipe and Filter Style Architecture for implementing the Disaster Management System.](image)

cooperating planners, multiple and when key changes are demand by human endorsement or guidance may be cooperate with mission analysts.

### 5.7 Disaster Management System (DMS)

Personal suffering, property loss, and economic hardships caused by disasters are reduced significantly when appropriate actions are taken in a timely manner. Good, reliable information forms the basis for those actions. The DMS monitors the situation round the clock, identifies the type of disaster, shows the status and generates reports. New and emerging information technologies, observation systems, and communications may now be integrated with state-of-the-art support tools for situational awareness, as well as risk assessment and management [47]. The DMS autonomously re-plans its strategy, adapts it, and proceed to accomplish their objectives. The latest software components are enthusiastically inserted into fields, heterogeneous structures without needed system restart, or certainly, any down time. During adaptation process, the system adapts new software collocation rule based component dynamically insert into DMS, without requiring system restart that detect the type of disaster automatically. DMS re-planning relies on analysis that includes
feedback from current performance using spatial collocation pattern mining shown in figure 5.7. This re-planning should take place separately, can involve distributed,

Figure 5.7 Proposed Software Architecture for Self-adaptable Disaster Management system for various disasters.
Throughout, system integrity needs the assurance of correctness coordination of change, and consistency. Changes can include the addition, removal or replacement of components and connectors, modifications to the configuration or parameters of components and connectors, and alterations in the components/connector networks topology. DMS can adapt itself to environmental disasters like Cyclone, Wars, Volcanoes, Floods, Earth Quake, and damage surveys in the above cases. These tools when combined provide disaster management and humanitarian assistance decision makers with powerful new and enhanced capabilities. Additionally, access by multiple users to a wide range of information, at varying levels of detail and many points in time can be easily accommodated.

Figure 5.8 Overview of Web-based Architecture for disaster Management
Figure 5.8 explains how the system is implemented using web based architecture. Using above architecture if a Collocation Rule is found for a disaster that occurs at any region can easily transferable to other locations through web. Now remaining locations which are meant for disaster identification are ready with new collocation rule, and consequently they can identify if the same disaster occurs in that region. This existing system by using self-adaptive architecture can adopt the new component (collocation rule).

If a DMS is constructed for one type of disaster i.e., epidemics, then going for a fresh specific software for each new DMS application is simply a redundant, waste of effort, money and time. Example, An airborne system sensor platform designed for environmental and land use monitoring could prove useful for damage survey following an earthquake or hurricane, provided some one could change the software quickly and with sufficient assurance that the newly adapted system would perform as intended [48][49][54].

5.8 Role of Software Architecture for Self-Adaptive Systems

This chapter examines the fundamental responsibility of software design in self-adaptive systems. A disaster management system is taken as an example and could try to implement the architecture-based self-adaptation concept. The concept of

Figure 5.9 the interaction of components using Client-Server Software Architecture
reuse and reusability is helping the users to the maximum extent for their changing needs or requirements. The primitive use of reuse is to select an adaptable construct like if statement programming level. The next reuse level is adapting or calling a subroutine from the library. Then after, Object Oriented supports the reusable concepts at subroutine as well as data structures. Now a group of such sub-routines and Data structures is available as commodity components. This requirement is because of changing requirement of user or faulty program uses exception of error handling when process is in progress (Run-time). And also consider software self-adaptability as a sub-set of total software reuse [48][49][55][56].

Adaptability is nothing but reusing a component appropriately to the context implicitly or explicitly. Software-based systems are expected to dynamically self-adapt to accommodate changing resource variability, changing user needs, and system faults. Self-adaptation currently exists in the form of programming languages feature such as exceptions and in algorithms such as fault tolerant protocols.

External mechanism uses external models and mechanisms in a closed-loop control fashion to achieve various goals by monitoring and adapting system behaviour at runtime as in figures 5.10 [55][56][57].

Control of system adaptation becomes the responsibility of components outside the system the system that is being adapted. Several researchers have proposed using architectural models that represent the system as a gross composition of components, their inter-connections, and their properties of interest. Such an architecture-based self-adaptation approach offers many benefits. Most significantly,
an abstract architectural model can provide a global perspective of the system and expose important system-level properties and integrity constraints. By adopting an architecture-based method, it offered reusable infrastructure jointly with mechanisms for deliberate that infrastructure to the needs of precise systems. These speciality mechanisms let the developer of self-adaptation capacities choose what features of the system to sculpt and scrutinize, the conditions must trigger revision, and how to acclimatize the system [48][49].

The figure 5.11 is the framework uses an abstract model to monitor executing systems runtime properties, and evaluation of adaptation components on the running system.

In particular, developers of self-adaptation capabilities use a system’s software architectural model to monitor and reason about the system. Using a system’s architecture as a control model for self-adaptation holds promise in several areas. As an abstract model, architecture can provide a global perspective of the system and expose important system-level behaviours and properties. As a locus of high-level system design decisions, an architectural model can make a system’s topological and

---

**Figure-5.11 The framework uses an abstract model to monitor**
behavioural constraints explicit, establishing an envelope of allowed changes and helping to ensure the validity of a change. Figures 3.5 & 3.6 show example of an architecture in which the components represent web clients and server clusters. Each server cluster has sub architecture. As shown in figure 3.4, system starts informing the status and generates reports for (alternative) various disasters, after identification of the type of disaster. Central to our view is the dominant role of Software architecture in planning, coordinating, monitoring, evaluation and implementation of self-adaptation. For example, Cholera spatial mining is going on, sudden Cyclone causes system to change its function accordingly. New application behaviour can be communicated to other main servers. Our system requires open adaptive behaviour because DMS will interacts with other required components which are located in central server. The figure 3.8 depicts overall self-adaptation process for epidemic disaster.

5.9 General Abstraction

In general, the representation of rules that are in the subject of data mining contains various syntactic and semantic significances. Modelling a rule definition is another important task for the data miners to bring back the emerging needs of the trends that are detected from the large spatial databases. There is lot of vagueness in defining the structures for the spatial data sets and the spatial related data sets. Reasons can be anxious from different directions of unavailability of super infrastructures, defining exact structure means making the spatial data more popular, and other defense restrictions. These reasons may look out as limitations or other way of preserving security for the data, instead of established seamless communication between one GIS to another. A geographic information system is an information system for data representing aspects of the surface of the earth together with relevant facilities such as lakes, houses, and roads. A geographic database provides spatial and non-spatial information. In the developing era of GIS and mobile communities, it is very essential to integrate several GIS societies. For globalization of several spatial domains and to restructure them all as a global spatial domain, it is very essential to heighten the abstraction of some good structure that can be easily followed into programming languages as a type of data. A type is definitive reasoning system generally, the program deviation for a mining problem can be assumed as polytypic, if some of the parameters are data types. Polytypic program deviations necessitate a general non-inductive definition of ‘data type’. A definition: a data type is a relator
that has membership interferences that can be suggested. The definition throws much light on how the various other properties that are shared by all data types influence the definition of the collocation type. In particular, case of databases, all the data types must have the unique strength and all natural transformation between data types is strong [58].

Many aspects of data types design deal with the problem of non-determinism, but ultimately they lay at the concepts of functors called relators. A functor is a constructor for a data type. A relator is which that can establish monotonic link between different facets of data types. The notion of membership has been examined in several experiments of the spatial data and the representation of the collocation erstwhile. During the course of inspection of data structures, it is not only the way to describe the ability of designing data type rather understanding the elementary aspects. This leads up to investigation of fans for the data types and it turns out that any relator with membership also has a unique fan. The definition of strong functors lead to the notion of data types what really the category theorists are working for the investigation. Contextually the technique of collection of several fan elements are grouped into a single common concept, which can be viewed as a single entity and nonessential information ignored – the abstraction [58].

5.10 The Actual Difference

The important and ragy (tr. rage) need to define the data type or data structure for storing the collocation rule is the concepts of variations (but not non-stability). The variation in the collocation is nothing but the change of intensity of the features that form as its core pattern. With regard to several changes occurred in a spatial domain the features form collocation, or could be resolved to form other new pattern or patterns. The direction of the inference changes as the changes occur to the features. Thus, this paper also strongly proposes a rule: change of gravity in the collocation.

A general definition for a collocation is proposed in [38][59][60] as a collocation is a subset of Boolean spatial features.

A collocation rule is of the form:

\[ c_1 \Rightarrow c_2(p, cp), \]

where \( c_1 \) and \( c_2 \) are collocations,

\[ c_1 \cap c_2 = \emptyset \]
$p$ is a number representing the prevalence measure, and $cp$ is a number measuring conditional probability.

According to normal spatial association rule, the identification of change of gravity may be considered as the confidence of that particular spatial item in the spatial item set that expresses equivalent to the features of a collocation [5][6][38]. A spatial association rule is a directional sign given to infer the basic characteristic, but a collocation rule could be a pattern that maps to a selected layer or layers of a spatial zone. The spatial association rule depicts the idea of relationship between reference objects of the spatial zone [5][6]. The collocation rule in its form gives a multi-dimensional or un-directional gravity identified structure. Representing a collocation rule requires a structure that can adapt to the changing position of gravity among the selected set of features in it [58].

**Change of Gravity**

Change of collocation, a pattern style representation of collocation rule contains a set of features that have various weights. The features exist in the pattern, add up and become new features or vanish as they become obsolete for knowledge representation. However, the existing and emerging features should compete with appropriate weights to survive in the pattern that which will contribute the quality for the knowledge representation. The gravity for the pattern is observed to be as the feature that has obtained more weight in the pattern. In the natural spatial domain, the features obtaining the weights and losing the weights or features losing the weights less than the threshold which leads to their disappearance is a common phenomenon.

**Syntax Modelling**

Syntax of a type can be developed and represented by canonical symbols that exhibit the storage and functional significance of a type. Let for $F$ denotes a functor and $S$ denotes a structure, then $F$ and $S$ should be relative in all aspects. $F$ and $S$ determine the functional aspects and the structural aspects of a type respectively. The proposed concept is supported by a data structure, where it can be abstracted and built as Object Type of Object-Oriented Paradigm. For given problem of spatial domain $S$, let $W$ be the type that is used to identify the collocation pattern. $S$ contains several forms of collocation patterns ($W$), which can represent different expressions of spatial knowledge. The sum total or the total glimpse of knowledge ($G$) in the problem spatial domain can be represented as:
\[ G = \sum_{i=0}^{n} W_i \]

W is defined to be the collocation pattern which is defined by a set of features. The features of a collocation are said to be organized is this experiment in two subjective components i.e., feature type and feature set. The feature type describes the structural details of the feature. The feature set describes the value details of the feature. For example if a collocation pattern that explains about some subject of spatial object is represented by \( W_i \), containing \( k \) features, then the features among them will be \( f_0 \) to \( f_k \), where \( k \) weighted features can make one collocation stand as \( W_i \). Each \( f_m \) is carefully expressed by feature type and feature set. Feature type of feature \( f_m \) may be represented by \( f_m(T) \) and feature set of feature \( f_m \) is represented by \( f_m(S) \). According to the basic programming conventions the \( f_m(T) \) can be defined as any abstract collection type that can be directly or indirectly fit into the programming definitions like array, list, stack, queue, set, etc,. The \( f_m(S) \) can be defined as any set of data that is consistent enough to express the idea and the attribute domain of the feature type. The need of representing feature as a member in the collocation like this can represent the knowledge. As a general procedure of data mining, converting the conventional data items into data sets or item sets is a general practice considered in designing the structure as a strategic weaving style. The most conventional style for representing the feature of the collocation is like template in C++. The selection of the \( f_m(S) \) and the \( f_m(T) \) for a feature are due to the natural phenomenal change in the spatial domain. As they change naturally, the importance of features varies temporally and thus the weights for each feature vary, which are naturally selected have indispensability and interdependency in the collocation. The need of integrated structure which encompasses the basics of collocation and their qualities is very essential for GIS.

A weight graph would be a better structure to represent the collocation, with an exception of weight represented in the vertices than the edges. The structure considered in this paper can be called as principal-ranking structure, a simple derivative of weight-vertex graph. Generally the weight graph has more concern about the weights of the edges, which is much bothered about the shape of graph. But in this context the shape is secondary; the importance is stressed much upon the existence of the vertex/node in the graph and its connectivity. The vertex-weight graph does not
contain any directional representations. The number of nodes that are connected in the graph is the matter of apprehension which indicates the number of features that subsist in a collocation, rather how far or near they are arranged that proposes the shape.[5][6][61][62]

For assumed graph \( G \), the notation is generally expressed a \( G = (V, E) \). Let \( G \) be a weight graph. The length of a path \( P \) is the sum of the weights of the edges of \( P \). That is, if \( P \) consists of edges \( e_0, e_1, ... e_{k-1} \) then the length of \( P \), is denoted \( w(P) \), is defined as

\[
w(P) = \sum_{i=0}^{k-1} w(e_i)
\]

But in the vertex-weighted graph the length is not the constraint, each vertex is assigned with a weight. The quantity of weight of a particular vertex is measured based on the importance of the feature in the pattern that is mostly deduced from the natural and temporal factors.

5.11 Complexity and Evaluation

The space complexity of graph is typically observed by expressing the complexity at each vertex bearing weights that represents candidate feature types, considering the direction of the connection of the vertices with least complexity. In the data-type, principal-ranking structure that is illustrated in this paper, the feature types become as collection of data, a collection type is classically experimented. Each of the collections is responsible for the connectivity of the dependent feature type collections. The overall design systematically can be observed as a fanned out dependent feature types with the more dominant feature types.

The type suggested experimentally can contain one or a set of dominant feature type collection in the core, and the other dependent feature types surrounding to the core, like a star. The star is mandatory to be observed that will identify the permanent feature type of the spatial object, which will be an important feature of the collocation. If a feature is installed in the centre of the type, in case of single existence of the feature it will act as an intelligent key feature for the collocation, which may also be brought from the original data source. In case of multiple existence of the feature, the set of the features will become as the facts for the collocation, which contain all naturally formed feature types, the intelligent keys again.
The principal-ranking structure (star-hierarchy structure) is a combination of basic B-Tree-like and Graph-List. The nodes of the star-hierarchy structure that represent the weighted-vertex-covers contain list of features sorted. The further level of nodes in the star-hierarchy structure are set-covers, contain vertex elements representing features, may be sorted according to the category of the set-covers’ [61][62][63][64][65]. But the overall structure of star-hierarchy is not necessarily constrained as sorted structure like B-Tree. The algorithmic complexity of the star-hierarchy structure is evaluated for each node and section of the structure discretely. At first-level, the weighted-vertex-covers, contains the features represented as base vertices, bear complexity as of a normal list. Similarly the vertices in the set-covers also bear complexity of a list. The classical and composite structure of the type described in this paper contains a list with several B-Tree nodes. The time complexity of the structure is evaluated by understanding the traversal of a list and a B-Tree. The general time complexity of a list containing \( k \) nodes will be \( O(k) \). The general time complexity of the B-Tree is said to be as the inequality \( h \leq \log_{(m/2)}((n+1)/2) \). That is for \( m \) nodes, \( h \) height and \( n \) elements in each node. Where the complexity of the total structure becomes as \( O(k)+k(\log_{(m/2)}((n+1)/2)) \). However the complexity of the list is within the limits of complexity of total set of trees available in the structure it is negligible and hence the complexity can form as \( k(\log_{(m/2)}((n+1)/2)) \).

According to [63] and other related [58][64][65][66][67] the concept of vertex cover applies appropriately. The vertex cover forms as the centrally installed set of vertices which represents the collection of the intelligent keys which are key feature types. Considering any heuristic graph, a collection of vertices representing the spatial objects and their features, a vertex cover of a graph is observed, a set of vertices that contains at least one end-point of each edge. A completely undirected graph is phenomenal representation of the geographical space. A vertex cover from the undirected graph is a subset \( C \subseteq V \) such that for all \( e = \{u,v\} \in E, e \cap C \neq \emptyset \). The vertex-cover, problem resolves in finding the value that is the size of the cover \( |C| \).

The data mining aspect of the graph theory persevere in designing the structure of the knowledge; approximating the covers from the graph is measured akin to the mining of rule and forming a structure. The approximation is most naturally (a heuristic) a greedy algorithm which repeatedly selects an edge that has not yet been
covered, and places one of its end points in the current covering set. Typical approximating routines are as follows. [61][62][63][64][65][66][67].

Approximating the Vertex-Cover
Input : Unweighted graph G(V,E).
Output : Vertex Cover C.
1. \( C \leftarrow \phi \)
2. while \( E \neq \phi \) do begin
   select any \( v \in V \);
   \( C \leftarrow C + v; \)
   \( E \leftarrow E \setminus \{e \in E | v \in e\}; \)
end;
3. return C.

Approximating the Weighted-Vertex-Cover
Input : G (V, E) and weight estimation function \( w \) on V.
Output : Vertex Cover C.
1. \( C \leftarrow \phi \)
2. while \( E \neq \phi \) do begin
   select any \( v \in w(V) \);
   \( c \leftarrow c + v; \)
   \( E \leftarrow E \setminus \{e \mid v \in e\}; \)
end;
3. return C.

The Weighted-Vertex-Cover, determines the ultimate set of vertices that become pivotal part of the collocation type. The selection of the other dependents to the WVC, are based on the nearness and the matching of the relationship with the vertices of the WVC.

The Maximum match algorithm is considered to find the set of vertices that depend on the WVC.

Input : Unweighted graph G(V,E).
Output : Vertex Cover C.
1. select any maximal matching \( M \subseteq E \) in G.
2. \( C \leftarrow \{v \mid v \text{ is matched in } M \}; \)
3. return C.
The user specified threshold fixes the min-dominance and max-dominance. The more weighted features among the features are the features that contain count and weight more than the user specified threshold. The resultant collocation pattern is represented appropriately as principal-ranking structure weighted vertex graph, which reflects all the dimensions of the spatial knowledge. Following the greedy heuristic of sampling the vertices of a Weighted-Vertex Graph (WVG):

The Set-Cover (SC) and the Weighted-Vertex-Cover (WVC) are derived from the WVG. The candidate Weighted-Vertex Cover (i.e., \( wvc_1 \)) that is principal for the structure, represents the dominant features of the spatial domain, and the Set-Cover

![Diagram of principal-ranking structure representation of collocation.](image1)

Figure 5.12 principal-ranking structure representation of collocation.

![Diagram of Principal-Ranking Structures for Spatial Knowledge Representation.](image2)

Figure 5.13 Principal-Ranking Structures for Spatial Knowledge Representation.
(i.e., sc₁) represents a set of vertices that are ranked based on the dependencies on the principal.

The programming approach to represent the above graph in a data structure is an object-oriented paradigm. Where all the WVCs (component of principal in the principal-ranking structure) are represented as the parent classes and conceptually identified as principal in this work, all the SCs are represented as the children surrounding the parent, a categorical sub tree under the principal tree.

A principal-ranking structure given in the Figure 5.13 is designed for representing a collocation pattern, which is a conceptually suitable structure to represent the collection of collocations. Features of high-dominance are stored in the principal and the other features are categorized into various sub trees under the principal. Thus the principal-ranking structure is useful for representing the various aspects of the spatial knowledge according to their dominance factors.

5.12 Experimental Work

The experimental work is carried out with the principles of mathematical approximation of data. The approach has made an attempt to study the concept of collocation and its salient parts distinctly and describe the relevant structure to store spatial knowledge. Schematic and mathematical assumptions are made throughout the experiment to bring forward the better structure that can help GIS, store and retrieve the knowledge. Consider S₁ to Sₙ are the spatial objects shown in the geographical map, where each spatial object inherently possesses some features. Let us interpret approximations for the experiment as follows:

Each spatial object has features, which are defined by several fuzzy sets [68]. The holder of each fuzzy set that defines the feature is represented by feature type. That means, a feature holds a feature type and a fuzzy set called as feature set which denotes several values for expressing the intensity of the feature in the spatial object. Where a = b = c = d = e may also apply for the approximations. According to [59][60][69][70] and others, the spatial features are designated as Boolean spatial features, which practically can represent only the presence or the absence of the features.

Values of the feature pertaining to a spatial object are more important and care should be taken that the object and the features could not be ignored at any instance of mining activities. The reason why the values which belong to the feature are set of values of a fuzzy set is to describe the intensity of the features with several gradients.
Let $Z$ indicate a candidate fuzzy set. $f_1$ of $S_1$ is defined by $\{ v \mid v \in Z(f_1) \}$, similarly $f_2$ of $S_1$ is defined by $\{ v \mid v \in Z(f_2) \}$, where the size (elements of fuzzy set) of feature arbitrarily differs.

The entire process of experiment is vested completely on non-spatial feature of spatial objects with fuzzy set values:

Hospital contains; crowd, working-hours and medication.

crowd is a feature which can be expressed in values as $\{dense, moderate, scant, mean, low, nil\}$.

working-hours is feature which can be expressed in values as $\{overtime, fulltime, part-time, visiting-time, lean-time, nil\}$.

medication is feature which can be expressed in values as $\{available, sufficiently-available, scantily-available, not-available/nil\}$.

Fuel-station also contains; crowd, working-hours along with storage-capacity:

storage-capacity defines the amount of service that the petrol-bunk can offer to the customers, the values that describe this feature are $\{available, sufficiently-available, scantily-available, not-available/nil\}$.

The working-hours depends on the storage-capacity of the petrol bunk.

School also contains working-hours, scholastic-offers and medication.

\[ S_1 = [ f_1 \ldots f_a ] \]
\[ S_2 = [ f_1 \ldots f_b ] \]
\[ S_3 = [ f_1 \ldots f_c ] \]
\[ S_4 = [ f_1 \ldots f_d ] \]
\[ S_5 = [ f_1 \ldots f_e ] \]
\[ S_6 = [ f_1 \ldots f_g ] \]

**Figure 5.14: Sample Spatial Zone.**

**Table 5.1 Fuzzy Set**

| $S_1$ = [ $f_1 \ldots f_a$ ] | $S_4$ = [ $f_1 \ldots f_d$ ] |
| $S_2$ = [ $f_1 \ldots f_b$ ] | $S_5$ = [ $f_1 \ldots f_e$ ] |
| $S_3$ = [ $f_1 \ldots f_c$ ] | $S_6$ = [ $f_1 \ldots f_g$ ] |
scholastic-offers describe the features pertaining to the education of the student in the school. medication, however a weak feature that seem to be irrelevant, but functionally the school may organize a medicating facilities which provides medication to the students. This feature may carry the same values that are used by the object which treats this feature as mandatory.

Support Count and its Correction:

Support Count is a statistic measure to find the dominance of the feature. Since storage-capacity – uses same fuzzy set as of the medication feature, adjustment can be made either to consider storage-capacity (3 times) or medication (3 times). In this example since storage-capacity behaviourally confines only to one object, which cannot be generalized, so only medication-3 is considered.

Spatial Object – Features

Support Count Finalized:

The overall count of the features used in the spatial domain for the spatial objects lists as Crowd-2, Working-hours-3, Medication-3 which form the structural parts of the collocation pattern.

### Table 5.2 Spatial Object Features List

<table>
<thead>
<tr>
<th>Feature</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crowd</td>
<td>{dense, moderate, scant, mean, low, nil}.</td>
</tr>
<tr>
<td>working-hours</td>
<td>{overtime, fulltime, part-time, visiting-time, lean-time, nil}.</td>
</tr>
<tr>
<td>medication</td>
<td>{available, sufficiently-available, scantily-available, not-available/nil}.</td>
</tr>
<tr>
<td>storage-capacity</td>
<td>{available, sufficiently-available, scantily-available, not-available/nil}.</td>
</tr>
<tr>
<td>Scholastic-offers</td>
<td>{offers to participants/students of the educational institution}.</td>
</tr>
</tbody>
</table>

Figure 5.15 Principal-ranking structure representation of collocation in the Experiment.
Spatial Features and Spatial Objects:

Spatial data sets are the key area for all the spatial data mining algorithms. Selection of the spatial data consisting of features for the spatial data mining is crucial when it comes for implementation. The spatial data is the representation of the spatial objects and their features that belong to a spatial layer [71]. The graph of spatial domain contains fundamental x and y co-ordinates and the description about the spatial objects and their features. This provides the ease of generating the collocation and contigs as similar to instance lookup scheme [72][73][74]. For imposing the fuzziness in the value of the features the randomization is implemented. A Java Based Random Semantic Map Data Generator, a tool developed to generate the spatial objects and their fuzzy representation of features. Randomization gives the clear scope of generating the features with more distinct alternatives for the spatial objects. sx and sy are the co-ordinates for the spatial objects with their name and type described as shown in Figure 5.5. Consolidated sets of the features are the other attributes of each object, where the attributes possess the gradient values, which are fuzzy in nature [75]. Table 5.1 shows the sample data for the experiment.

Attributes

<table>
<thead>
<tr>
<th>M- maintenance;</th>
<th>C – capacity;</th>
<th>P – power;</th>
<th>D- demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cr – crowd;</td>
<td>T – timing;</td>
<td>Md – medication</td>
<td>R – results;</td>
</tr>
<tr>
<td>O – offers;</td>
<td>A – academic-contributions</td>
<td>Tr – trade-style</td>
<td></td>
</tr>
</tbody>
</table>

The spatial objects used in this experiment are Water-resource(x), Hospital (+), School (-), Fuel-station (*), Market (o).

Figure 5.16 A Screenshot of Random Semantic Map Data Generator Tool.
Table 5.3 Output generated of Random Semantic Map Data Generator Tool.

<table>
<thead>
<tr>
<th>sno</th>
<th>sid</th>
<th>sx</th>
<th>sy</th>
<th>gdesc</th>
<th>scd</th>
<th>M</th>
<th>C</th>
<th>P</th>
<th>D</th>
<th>Cr</th>
<th>T</th>
<th>Mo</th>
<th>R</th>
<th>O</th>
<th>A</th>
<th>Tr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>60</td>
<td>200</td>
<td>Water-Resource</td>
<td>WA</td>
<td>0</td>
<td>40</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>21</td>
<td>60</td>
<td>203</td>
<td>Water-Resource</td>
<td>WA</td>
<td>90</td>
<td>0</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>62</td>
<td>193</td>
<td>Water-Resource</td>
<td>WA</td>
<td>40</td>
<td>20</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>66</td>
<td>428</td>
<td>Hospital</td>
<td>HO</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>18</td>
<td>67</td>
<td>189</td>
<td>Water-Resource</td>
<td>WA</td>
<td>0</td>
<td>40</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>17</td>
<td>71</td>
<td>119</td>
<td>Water-Resource</td>
<td>WA</td>
<td>0</td>
<td>50</td>
<td>70</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>86</td>
<td>338</td>
<td>Hospital</td>
<td>HO</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>115</td>
<td>73</td>
<td>Fuel-Station</td>
<td>FS</td>
<td>50</td>
<td>50</td>
<td>70</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>130</td>
<td>139</td>
<td>Hospital</td>
<td>HO</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>153</td>
<td>313</td>
<td>Fuel-Station</td>
<td>FS</td>
<td>90</td>
<td>0</td>
<td>70</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>157</td>
<td>212</td>
<td>Fuel-Station</td>
<td>FS</td>
<td>40</td>
<td>10</td>
<td>20</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>36</td>
<td>167</td>
<td>330</td>
<td>Water-Resource</td>
<td>WA</td>
<td>40</td>
<td>80</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>37</td>
<td>167</td>
<td>330</td>
<td>Water-Resource</td>
<td>WA</td>
<td>40</td>
<td>80</td>
<td>80</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>13</td>
<td>170</td>
<td>204</td>
<td>Hospital</td>
<td>HO</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>172</td>
<td>283</td>
<td>Fuel-Station</td>
<td>FS</td>
<td>90</td>
<td>0</td>
<td>70</td>
<td>60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>39</td>
<td>173</td>
<td>172</td>
<td>School</td>
<td>SC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>8</td>
<td>177</td>
<td>249</td>
<td>Fuel-Station</td>
<td>FS</td>
<td>40</td>
<td>90</td>
<td>70</td>
<td>30</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>46</td>
<td>165</td>
<td>366</td>
<td>School</td>
<td>SC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>194</td>
<td>83</td>
<td>Fuel-Station</td>
<td>FS</td>
<td>10</td>
<td>40</td>
<td>80</td>
<td>40</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>45</td>
<td>169</td>
<td>341</td>
<td>School</td>
<td>SC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.3 Describes the spatial objects and their values of the fuzzy features in rows

![Figure 5.17(a)](image1.png)
![Figure 5.17(b)](image2.png)

**Figure 5.17:** A principal ranking structure representing the spatial knowledge derived from the experiment for Example 1 & 2.

**Spatial Features Distinguished With Examples**

**Example 1**

Consider the rows 2 and 3, which can satisfy the predicate close_to even in the plane sweep on the coordinated spatial domain and forms a singleton-collocation.
**Example 2**

Consider the rows 3, 4, 5, in which 3 and 5 only can satisfy the predicate *close to* even in the plane sweep on the coordinated spatial domain and form a singleton-collocation, but 4 is very distinct and far which cannot form even the singleton collocation. Only 3 and 5 can form a collocation.

**Example 3**

Consider the rows 11, 12, 13, 14, though 12 and 13 are common in nature, the three objects 11, (12, 13), 14 collocate together and forms a fundamental clique, but no features match with the spatial objects, which does not form a collocation.

The general collocation tree is formed by using Boolean spatial features. This can express only the presence or the absence of the feature. The existing features in the figure 5.7 are only the features that are present. The same data considered in Table 5.1. Comparatively, with fig. 5.17(a), 5.17(b), this figure illustrates only one instance of collocation, which is inferred that much of the data is ignored for the collocation mining.

According to[71], The Boolean spatial features, just only represent geographic object types which are either present or absent at different locations in a two dimensional or three dimensional metric spaces. Collocation patterns represented in the [71] and other belonging works, the subsets of the Boolean spatial features whose instances are often located in close geographic proximity. But the co-location pattern can be proposed as a set of spatial features with values that belong to fuzzy set.

In [59][60][69][70] the representation of the collocation is made as a table instance. Even after mining the collocations instances and generating the collocation rules, the rules are just represented as a table instance only. The categorical dependencies and relationships between the features cannot be derived simply, to

![Figure 5.18 Collocations with Boolean Spatial Features](image)
mend the meaning of the collocation, rather performing many sequential passes of iterations on the rows of the table instances. The structure proposed in this work, is principal-ranking structure, and expresses the features of the collocation to store in the tree like structure. The features are basically categorized with dominance than prevalence. The minimum dominance and the maximum dominance features are categorically stored in the various sub-groups of the principal-ranking structure. The basic elements of the principal-ranking structure consists of a principal, which is a tree structure that is used to represent the maximum dominance features, and the categorical set-covers that each storing the minimum dominance features categorized.

**Semantic Influence:**

Identification of the spatial feature in the spatial object is not the fundamental requirement of spatial data mining, identifying the spatial feature and its intensity, how strongly that is influencing the spatial object, and support of participation in the formation of collocation. The gradient style of value representation to the feature can also be used to describe the various states of the feature when it prevails in a particular spatial object.

The advantage of describing the feature with a fuzzy set of values is to increase the amount of input data to the data mining algorithm, so that most of the spatial data can be considered to mine the knowledge. The aggregate of knowledge will be absolute when most of the data is used. In case of the Boolean spatial features, just a false will make the spatial feature disappear from the spatial domain, which will intensely causes the absence of the data in the input, which leads to shrinking of the knowledge quality.

**5.13 Conceptual Model**

A conceptual model figure 5.19 for storage of knowledge is as follows: This conceptual model is twofold; one is to gather the rules from the environmental and spatial data, and selecting the suitable data mining algorithms. Other is to represent rules mined by the mining algorithms to represent the structural storage in the device. The auxiliary fold in this is to form the link between the system and the client-base.

**Databases and Generalization:**

Object-Types and the Object-Instances (like column-objects and row-objects) are available in Oracle™ Databases, which directly gives the opportunity to manage the star-hierarchy of the collocation pattern.
This chapter examines the fundamental function of software structural design in self-adaptive schemes and outlines methods are considered for supporting the methodology. A collocation rule with weaker parameters may also exist in the spatial zone, but cannot play important role for the detection of the epidemic. Framework was described for the application of collocation rules i.e., spatial knowledge by the health campaign. Spatial knowledge is extracted by applying one of the data-mining concepts called Collocation rule. This Collocation rule gives us the Spatial Knowledge. Such Spatial Knowledge is mined for epidemic as an example. EDMS Architecture was proposed to help the disaster management team. A novel Self Adaptive disaster management system Architecture was also proposed when system has to adapt different disaster components.

The data structure described in this chapter is that fits all the requisites to store the spatial knowledge or collocation rule. Towards the application and integration of GIS, the feasible implementation of collocation rule that forms as a basic element of knowledge base of GIS is a more important concern; the structure has exceptional importance for the application relevant tasks. The representation of the structure is complex, and its various subscribed parts which have achieved not yet been applied by theorists is given with a proof of complexity and its performance evaluation. A join less approach of finding the collocations is worked out as a background of the experiment. The type discussed in this work ideally suits to store the nuggets of spatial.