FUZZY SPATIAL DATA CUBE CONSTRUCTION AND ITS USE IN ASSOCIATION RULE MINING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

NARİN IŞIK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

MAY 2005

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Ayşe Kiper
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science of Computer Engineering.

Prof. Dr. Adnan Yazıcı
Supervisor

Examining Committee Members

Assoc. Prof. İsmail Hakkı Toroslu (METU, CENG) _____

Prof. Dr. Adnan Yazıcı (METU, CENG) _____

Assoc. Prof. Ali Hikmet Doğru (METU, CENG) _____

Assist. Prof. Zuhal Akyürek (METU, GGIT) _____

Assist. Prof. Halit Oğuztüzün (METU, CENG) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Surname, Name :

Signature :

ABSTRACT

FUZZY SPATIAL DATA CUBE CONSTRUCTION
AND ITS USE IN ASSOCIATION RULE MINING

Işık, Narin

M.S., Department of Computer Engineering

Supervisor : Prof. Dr. Adnan Yazıcı

May 2005, 109 pages

The popularity of spatial databases increases since the amount of the spatial data that need to be handled has increased by the use of digital maps, images from satellites, video cameras, medical equipment, sensor networks, etc. Spatial data are difficult to examine and extract interesting knowledge; hence, applications that assist decision-making about spatial data like weather forecasting, traffic supervision, mobile communication, etc. have been introduced. In this thesis, more natural and precise knowledge from spatial data is generated by construction of fuzzy spatial data cube and extraction of fuzzy association rules from it in order to improve decision-making about spatial data. This involves an extensive research about spatial knowledge discovery and how fuzzy logic can be used to develop it. It is stated that incorporating fuzzy logic to spatial data cube construction necessitates a new method for aggregation of fuzzy spatial data. We illustrate how this method also enhances the meaning of fuzzy spatial generalization rules and fuzzy association rules with a case-study about weather pattern searching. This study contributes to spatial knowledge discovery by generating more understandable and interesting knowledge from spatial data by extending spatial generalization with fuzzy memberships, extending the spatial aggregation in spatial data cube construction by utilizing weighted measures, and generating fuzzy association rules from the constructed fuzzy spatial data cube.

Keywords: Fuzzy spatial data cube, spatial data cube, fuzzy data cube, fuzzy association rules, spatial knowledge discovery.

ÖZ

BULANIK UZAYSAL VERİ KÜPLERİNİN OLUŞTURULMASI
VE İLİŞKİ KURALLARININ BULUNMASINDA KULLANILIŞI

Işık, Narin

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi : Prof. Dr. Adnan Yazıcı

Mayıs 2005, 109 sayfa

Uzaysal veri tabanlarının popülerliği, dijital haritaların, uydu görüntülerinin, video kameraların, tıbbi donanımların ve algılayıcı şebekelerin kullanımıyla artmaktadır. Uzaysal verilerin incelenmesinin ve onlardan işe yarar bilgi üretiminin zorluğu, onlar hakkında daha kolay kararlar alınmasını sağlayacak hava durumu tahmini, trafik denetimi, mobil komünikasyon gibi uygulamarı gündeme getirmiştir. Bu tezde, uzaysal veriler hakkında karar almayı kolaylaştırmak için, bulanık uzaysal veri küpü oluşturarak ve ondan bulanık ilişki kuralları üreterek, uzaysal verilerden daha doğal ve güvenilir bilgi üretilmektedir. Bunun için, uzaysal verilerden bilgi ortaya çıkarmak ve bunu bulanık mantık ile geliştirmek konularında kapsamlı bir araştırma yapılmıştır. Uzaysal veri küplerinde bulanık mantık kullanımı, verilerin bir araya toplanması için yeni bir metod gerektirmektedir. Biz bu metodun bulanık genelleme kurallarının ve bulanık ilişki kurallarının anlamını nasıl arttırdığını, hava durumu kalıplarının araştırıldığı örnek bir çalışma ile de göstermekteyiz. Bu çalışma, uzaysal verilerin genellemesini bulanık üyeliklerle geliştirerek; uzaysal veri küplerini oluştururken uzaysal verilerin bir araya toplanmasını ağırlıklı ölçümlerin kullanımıyla genişleterek ve oluşturulan bulanık uzaysal veri küpünden bulanık ilişki kurallarını çıkararak daha anlaşılır ve daha işe yarar bilgi üretmekte ve uzaysal bilgi üretimine katkıda bulunmaktadır.

Anahtar Kelimeler: Bulanık uzaysal veri küpü, uzaysal veri küpü, bulanık veri küpü, bulanık ilişki kuralları, uzaysal bilgi üretimi.

ACKNOWLEDGMENTS

The author wishes to express her deepest gratitude to her supervisor Prof. Dr. Adnan Yazıcı for his guidance, advice, criticism, encouragements and insight throughout the research and to her family for their support.

TABLE OF CONTENTS

ABSTRACT iv
ÖZ v
ACKNOWLEDGMENTS vi
TABLE OF CONTENTS vii
LIST OF TABLES ix
LIST OF FIGURES x
ABBREVIATIONS xii
CHAPTER
1. INTRODUCTION 1
2. BACKGROUND 6
2.1 Data Warehousing 6
2.2 OLAP 11
2.2.1 Conceptual data model 12
2.2.2 Database Design Methodology 14
2.2.3 OLAP Servers 15
2.2.4 OLAP Operations 17
2.2.5 Data cube 17
2.2.6 OLAP Mining 22
2.3. Fuzzy OLAP 24
2.4. Spatial OLAP 26
2.4.1 Spatial Knowledge Discovery 27
2.4.2 Spatial Data Cube 29
2.5. Knowledge Discovery and Data Mining 33
2.5.1 Association Rules 34
3. FUZZY SPATIAL DATA CUBE 38
3.1 Fuzzy Spatial Data Cube Construction 39
3.1.1 Collection of spatial data 43
3.1.2 Determination of Dimensions and Measures 45
3.1.3 Definition of Fuzzy Hierarchies 47
3.1.4 Determination of All Aggregation Types 48
3.1.5 Generalization of Dimensions using Fuzzy Logic 50
3.1.6 Fuzzy Spatial Aggregation 53
3.1.7 Generalization of Measures using Fuzzy Logic 57
3.2 Meaning of Values in the Fuzzy Spatial Data Cube 58
3.3 Fuzzy Association Rule Generation from Fuzzy Spatial Data Cube 60
3.4 Implementation 64
4. CASE-STUDY: “Weather Pattern Searching” 69
4.1. Construction of New Fuzzy Spatial Data Cube 70
4.2. View Constructed Fuzzy Spatial Data Cube 82
4.3. Finding Association Rules 86
5. DISCUSSION 88

6. CONCLUSION..92
REFERENCES 94

LIST OF TABLES

Table 1: Differences between Data Warehouses and Operational Databases.....8
Table 2: “Sales” Summary Table 18
Table 3: Result of a Roll-Up Operation.....32
Table 4: Record Containing Membership Values 36
Table 5: Spatial Generalization 41
Table 6: Fuzzy Spatial Generalization.....41
Table 7: Generalizations with respect to dimensions 58
Table 8: Fuzzy Association Rules 63
Table 9: Performance Metrics 68

LIST OF FIGURES

Figure 1: The Data Warehouse Architecture [1] 9
Figure 2: Multidimensional Data 12
Figure 3: A Concept Hierarchy for Location Dimension 13
Figure 4: The Star Schema 14
Figure 5: The Snowflake Schema 15
Figure 6: Data Cube 19
Figure 7: Cuboids of the Data Cube 20
Figure 8: The Lattice of Cuboids with Hierarchical Dimensions 21
Figure 9: Fuzzy Data Cube 25
Figure 10: Fuzzy Hierarchies 26
Figure 11: The Star Schema for Spatial Data Warehouse 30
Figure 12: Hierarchies for the Star Schema 31
Figure 13: Example Data for Hierarchies in Star Schema 31
Figure 14: Dimensions and Their Memberships 42
Figure 15: Format of data 44
Figure 16: The Geo-object Table 45
Figure 17: Membership Function Types 46
Figure 18: The Gaussian membership function 47
Figure 19: The season fuzzy hierarchy for the temperature dimension 48
Figure 20: Group-by Expressions 49
Figure 21: Pseudo code for determining aggregation types 49
Figure 22: The lattice of fuzzy spatial data cube 50
Figure 23: Choice of the most appropriate fuzzy set 50
Figure 24: Pseudo code for generalization of dimensions 52
Figure 25: Generalized geo-objects 52
Figure 26: Aggregated fuzzy values 54
Figure 27: Pseudo Code for Fuzzy Spatial Aggregation 56
Figure 28: Fuzzified measures 57
Figure 29: Pseudo Code for Generalization of Measures 58
Figure 30: Pseudo code for generating fuzzy spatial association rules 64
Figure 31: Deployment Diagram 65
Figure 32: Interfaces 65
Figure 33: The Class Diagram 66
Figure 34: Database Design 67
Figure 35: Start page of the fuzzy spatial data cube construction application 69
Figure 36: Choice of input file 70
Figure 37: Format of the input file 71
Figure 38: Summary of the data read 72
Figure 39: Candidates for dimensions and measures 73
Figure 40: Definition of dimensions and measures 74
Figure 41: Fuzzy set definition 75
Figure 42: Hierarchy definition 76

Figure 43: Addition of hierarchy value....76
Figure 44: Group by expressions....77
Figure 45: Necessary data for the cube....78
Figure 46: Crisp data in "data" table....78
Figure 47: Display of generalized data....79
Figure 48: Generalized data in "extendeddata" table....79
Figure 49: Display of aggregated data....80
Figure 50: Aggregated data in "aggregate" table....80
Figure 51: Display of generalized measures....81
Figure 52: Aggregated data in "aggregate" table with generalized measures....81
Figure 53: View options for the constructed cube....82
Figure 54: View of fuzzy spatial base data....83
Figure 55: View of fuzzy spatial aggregated data....84
Figure 56: View of dimensions info....85
Figure 57: View of measures info....85
Figure 58: Threshold values for significance and certainty factors of fuzzy association rules....86
Figure 59: Display of fuzzy association rules....87
Figure 60: Spatial Generalization....88
Figure 61: Fuzzy Spatial Generalization....88
Figure 62: Aggregation in fuzzy data cubes....90

ABBREVIATIONS

CWM	:	Common Warehouse Model
DBMS	:	Database Management System
DSS	:	Decision Support System
ER	:	Entity Relationship
GIS	:	Geographic Information System
GUI	:	Graphical User Interface
KDD	:	Knowledge Discovery in Databases
MBR	:	Minimum Bounding Rectangle
OLAP	:	On-line Analytical Processing
OLTP	:	On-line Transaction Processing
OMG	:	Object Management Group

CHAPTER 1

INTRODUCTION

Decision support systems (DSSs) are database management systems that run queries efficiently to make decisions [2]. They contain concepts of both data warehousing and On-line Analytical Processing (OLAP) [1, 2] since they need the knowledge that might be missing from operational databases. DSSs are very useful in understanding the trends and making predictions and that qualification makes them the focus of the database industry. They necessitate historical data that is consolidated from heterogeneous sources by data warehouses.

Data warehouses are decision support databases that are maintained separately from the operational databases of organizations. They keep the organization-wide snapshots (i.e., subject – oriented collection) of data, which are used to improve making decisions [2, 3, 4]. They extract useful information from data that does not exist in operational databases [10]. Knowledge workers benefit from that extracted information since they are interested in the big picture, not the specific details.

Data warehouses support information processing by providing a solid platform of consolidated, historical data for analysis. Numerical information is stored in multidimensional fashion and analyzed in data warehouses by generating high-level aggregates that summarize the numeric values. OLAP corresponds to one class of decision-making queries that are run at data warehouses. OLAP operators manipulate data along the multiple dimensions. These operators are most frequently used to increase or decrease the level of aggregation.

A data warehouse is based on a multidimensional data model that views data in the form of a data cube while relational model views data in the form of tables.

Conceptual model for OLAP is multidimensional view of data in the warehouse. A data cube allows data to be modeled and viewed in multiple dimensions.

In “Fuzzy OLAPs”, OLAP mining and fuzzy data mining are combined to get benefit of the fact that fuzzy set theory treats numerical values in more natural way, increases the understandability, and extracts more generalizable rules since numerical data are interpreted with words [14, 15, 16]. Fuzzy OLAPs are performed on fuzzy multidimensional databases in which multidimensional data model of data warehouses is extended to manage imperfect and imprecise data (i.e., bad sales) from real world and run more flexible queries (i.e., select middle sales).

Furthermore, Kuok et. al. [29], has generated fuzzy association rules by introducing significance and certainty factors and proposing methods for computing these factors.

Spatial databases are able to store information about the position of individual objects in space. On the other hand, many applications that assist decision-making about spatial data like weather forecasting, traffic supervision or mobile communications necessitate summarized data like general weather patterns according to region partitions, number of cars in an area, or phones serviced by a cell. Moreover, creation of maps from satellite images and usage of telemetry systems, remote sensing systems or medical imaging results in a huge amount of spatial data. That causes to difficulties when examining large amounts of spatial data and extracting interesting knowledge or characteristic rules from them. Obtaining this information from operational (i.e., transactional) spatial databases is quite expensive. In that point, spatial data-warehouses and OLAP becomes crucial for spatial knowledge discovery.

Regarding spatial data warehouses, Stefanovic et. al. [9] has studied methods for spatial OLAP by combining non-spatial OLAP methods and spatial databases. They

propose a model for spatial data-warehouses, which has both spatial and non-spatial dimensions and measures are regions in space. They also propose a method for spatial data cube construction called “object-based selective materialization”. Their study is similar to the study of Harinarayan et. al. [10], with the difference being the finer granularity of the selected objects. Zhou et. al. [26], has proposed an efficient polygon amalgamation method for merging spatial objects. This method was applied on the computation of aggregation for spatial measures. Multi-resolution amalgamation was proposed by Prasher and Zhou [22], in which dynamic aggregation is done for spatial data cube generation. In this method, the resolutions of regions are changed in order to keep spatial data at much higher resolutions and by amalgamation objects are re-classified into high-level objects that form a new spatial layer. Papadias et. al. [25] has proposed a method that combines spatial indexing with materialization technique. In this method, aggregated results are stored in the spatial index.

On behalf of spatial knowledge discovery, Lu and Han [6] have proposed two generalization algorithms for spatial data which are non-spatial data and spatial data dominated generalizations. In these algorithms, generalization is done according to high level concepts, boundaries of which are sharply defined. In particular, regions that have temperature values in interval [10-20] are first generalized to “[10-20] temperature regions” and then further to “mild regions”.

In this thesis, spatial data cubes and fuzzy data cubes are tried to be harmonized in order to get benefit from the strengths of both of these concepts. Spatial and non-spatial dimensions and measures considered for spatial OLAP in [9], spatial generalization described in [6] and fuzzy association rules asserted in [29] are combined in one study and further improved. Better analysis and understanding of huge spatial data by using fuzzy set theory for spatial data cubes are aimed. More specifically, spatial generalization algorithms, that were previously mentioned [6], are enhanced by introducing fuzzy logic in determining high level concepts (i.e.,

membership values of high level concepts are computed). Moreover, a new aggregation method is introduced for fuzzy spatial data cube in which membership values of the more significant regions have greater weight for the aggregated region. Furthermore, we illustrate how this method is used for fuzzy spatial generalization rules and fuzzy association rules. Fuzzy association rule mining is applied over the generated fuzzy spatial data cube and computation of significance and certainty factors are adapted according to it instead of applying spatial association rules [4] which have more complex computation.

This thesis differentiates from the previous studies about spatial knowledge discovery in the following ways:

1. Generalization algorithms in [6] are extended by fuzzy high-level concepts and their memberships. Hence, more understandable and meaningful generalization rules are extracted due to fuzzy set theory (i.e., generalizations like "hot region with %89 reliability" instead of "[20-25] ºC temperature region").
2. Spatial aggregation in spatial data cube construction is extended by calculating fuzzy memberships of dimensions and measures for the aggregated cells considering more significant regions with greater weight. More accurate generalization rules are extracted at higher levels of abstraction.
3. Deviations in precision values for reliability of characteristics of spatial data along time can be tracked.
4. Each cell in a fuzzy spatial data cube has its own membership for the values of fuzzy dimensions and fuzzy measures it satisfies. On the contrary, in fuzzy data cube, cells in a slice (i.e., cells that have a common value for one dimension) of a fuzzy data cube has the same membership.
5. Fuzzy hierarchies are handled during fuzzy spatial data cube in order to increase the level of abstraction (i.e., level of precise knowledge extracted from spatial data).

6. In previous studies, spatial association rules were mined from spatial data which has a high computational complexity. In the constructed fuzzy spatial data cube, more flexible association rules can be discovered by mining fuzzy association rules since data is both fuzzy and spatial in order to avoid the complexity of spatial association rules.

In the rest of the thesis, first, background is given in Chapter 2 that includes data-warehousing; OLAP concepts like conceptual data model, database design methodology, OLAP servers, data cube, OLAP mining; fuzzy OLAP; spatial OLAP; knowledge discovery in spatial data and data mining. The methodology followed in fuzzy spatial data cube construction and its application in fuzzy association rule mining is explained in Chapter 3. The case-study "Weather Pattern Searching" is introduced in Chapter 4. Finally, Chapter 5 includes the final comments and Chapter 6 draws a number of conclusions that summarizes the study.

CHAPTER 2

BACKGROUND

The increase in size of available spatial data has brought the problem of encountering difficulties in knowledge discovery. Construction of spatial data warehouses has improved the quality of decision making about such kind of voluminous data. They have enabled the summarization and characterization of large sets of spatial objects in different dimensions and at different levels of abstraction. The summarized data have helped in extracting information and giving business decisions.

On the other hand, fuzzy data cubes that correspond to the multidimensional data model of fuzzy data warehouses, have enabled the extraction of relevant knowledge in a more human understanding and given results to queries with a certain precision about the reliability of that knowledge.

In the following sub-chapters, the concepts of data warehousing are explained in more details. It can be easily noticed that these two separate concepts – spatial data cubes and fuzzy data cubes – need to be harmonized and handled together in order to benefit from the strengths of both of them.

2.1 Data Warehousing

Data warehousing is a collection of decision support technologies like OLAP, data mining and powerful statistical capabilities that are used by knowledge workers (i.e., executive, manager, analyst) to make better and faster decisions. Inmon (1992) has given a formal definition that is also the oldest definition of a data warehouse as "subject-oriented, integrated, time-varying, non-volatile collection of data in support of management's decision making process." (cited in [1, 4, 17, 27]). It is subject-oriented, since it is organized around major subjects and excludes data that are not

useful; it is integrated, since it is cooperated from multiple heterogeneous data sources like relational databases and flat files; it is time-variant, since it provides infrormation from historical perspective; finally it is non-volatile, since it is physically seperated from data transformed from operational environment [4]. The terms “decision support system” and “data warehouse” are sometimes used interchangeably. Actually, data-warehouses correspond to the database of decision support systems.

Data warehousing is applied in manufacturing for order shipment; retail for user profiling and inventory management; financial services for claims analysis, risk analysis, credit card analysis, and fraud detection; transportation for fleet management; telecommunications for call analysis and fraud detection; utilities for power usage analysis, and healthcare for outcomes analysis [1]. Data warehouses are useful for information processing (querying, basic statistical analysis, reporting), analytical processing (OLAP), and data mining (knowledge discovery from hidden patterns, associations, classification, prediction) since they provide a solid platform of consolidated and historical data for analysis [4].

Data warehouses and operational (i.e., transactional) databases of an organization are maintained separately since they have different functional and performance metrics. Warehouses are very large in size and include historical and summarized data since they are used to understand the trends or make predictions. Data are consolidated from heterogeneous sources and reconciled about the quality, representation, code, and format. They access huge amount of data and run complex queries that include many scans, joins and aggregations. These queries may be OLAP queries or general decision support system queries and are run in parallel. Execution of OLAP queries in operational databases would result in an unacceptable performance [1, 2, 4, 20].

Likewise, multidimensional model of data in a data warehouse facilitates analysis and visualization. In order to support the multidimensional data model and OLAP

operations, data warehouses have special data organization, access and implementation methods different from operational databases. On the whole, these are the reasons for separating warehouses and operational databases. They have substantial differences [1, 2, 3, 4, 27] and these differences are briefly summarized in Table 1:

Table 1: Differences between Data Warehouses and Operational Databases

	Data Warehouses	**Operational Databases**
Processing	OLAP	OLTP
Operations	Rollup, drill-down, etc.	Read, update
Task Types	Complex, ad-hoc	Structured, repetitive
Transactions	-	Short, atomic, isolated
Data	Historical, summarized and consolidated, read-only	Detailed, up-to-date
Data Structure	De-normalized, redundant data structure	Normalized
Data Access	Millions of records	A few (tens of) records
Size	Hundreds of GBs to TBs	Hundreds of MBs to GBs
Critical issues	Data integration	Consistency, recoverability
Performance metric	Query throughput and response time	Transaction throughput
Target	Market	Customer
Database design	Star/Snowflake schema	ER, UML
Users	Knowledge worker	IT professional
Function	Decision Support	Day-to-day operations

In order to build and maintain a data warehouse, some steps should be followed [1, 2, 3, 4, 27]. Formerly, an OLAP server should be selected according to the database(s) that will be used in the data warehouses. Subsequently, an integrated schema and

some complex queries should be defined at design time depending on the data sources. Next, the architecture of the data warehouse should be defined; integrated enterprise warehouse vs. data-marts, centralized vs. decentralized warehouse. An integrated enterprise warehouse collects information about all subjects in the organization. On the other hand, data-marts are smaller functional warehouses or decision support systems that focus on specific subjects and contain homogeneous data. The former is complex, time consuming and requires extensive business modeling while the other runs faster but may have complex integration. Moreover, data warehouses can be built centralized (i.e., central data warehouse with data-marts) or decentralized (i.e., distributed data warehouses).

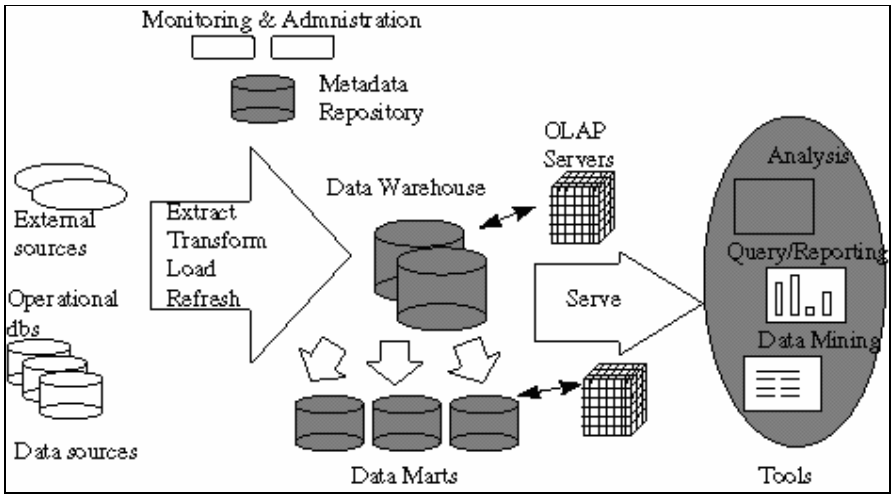


Figure 1: The Data Warehouse Architecture [1]

Typical data-warehouse architecture is as displayed in Figure 1. It consists of back-end tools, metadata repository, data warehouse and optional data marts, and front end tools.

Back end tools are used in order to extract data from multiple operational databases and external sources, clean and transform the data, integrate data according to the

defined schema, load data to the data warehouse and refresh data to reflect the updates at the sources.

Before running queries on data warehouses, consolidated data is cleaned [1, 2, 3, 4], in other words corrected. Since large volumes of data from multiple sources are involved, there is a high probability of errors and anomalies. Hence, errors and anomalies in data should be detected and corrected. Semantic conflicts and inconsistencies like missing entries and invalid values between distinct sources should be handled. This involves data migration, data scrubbing and data auditing. Data migration is the use of some transformation rules, i.e., replace “last name” with “surname”, in order to convert data from legacy or host format to warehouse format. Data scrubbing is the usage of domain-specific knowledge to modify data by parsing and fuzzy matching, i.e., zip codes, postal addresses. Data auditing is discovering rules and relationships in a similar way of data mining.

After data cleaning, data are loaded to the data warehouse [1, 2, 3, 5]. This process involves sorting, indexing, summarization, aggregation, integrity constraints checking, building derived tables and indices, materializing views, etc. Parallelism is used in order to decrease the time consumed during loading. A full load builds a new database and is a long batch transaction while an incremental load (refresh) includes only updates to the source data, breaks the load into smaller transactions and commits periodically.

The front end tools consist of some query tools, report writers, analysis tools, and data mining tools [1]. OLAP operations are executed by these front end tools to enable the end user to query in terms of domain specific business data.

As well as back and front end tools, a data warehouse manages some metadata that describes the integrated data and the integration process [1, 3, 4, 27]. Administrative metadata are required to setup and use the warehouse and include information about

target tables and their source definitions, front and back end tools, schema definitions, derived data, dimensions, hierarchies, queries, reports, data mart location and contents, data partitions, data extraction, cleaning, transformation rules, refresh, access control, etc. Operational metadata are obtained during operation of the warehouse and contain history of migrated data, transformation paths, currency of data, usage statistics, error reports, etc.

2.2 OLAP

OLAP is a computing technique for summarizing, consolidating, viewing, applying formulae to and synthesizing data according to multiple dimensions in data warehouses [14, 16, 18, 20]. OLAP is needed for executing complex queries on huge amount of data that are processed on aggregated level, not on individual level of records as for OLTP. The main target of introducing OLAP to the literature was fast, flexible data summarization and analysis which also involves hierarchical structures and simple computation of aggregations. That is why, OLAP applications were introduced for users who frequently want a higher-level aggregated view of the data to understand the trends and make business decisions like analysts and managers.

That terminology was first introduced by Codd in 1993 (cited in [14]) for the kind of technologies that provide means of collecting, storing, and dealing with multidimensional data with a view to deploying analysis process. It covers a class of technology for access and analysis of multidimensional view of business data [20].

OLAP necessitates a multidimensional data model to facilitate analysis and visualization. That model is designed differently from relational data models which were designed by entity relationship diagrams. Moreover, OLAP necessitates different OLAP servers according to the type of the database on which data-warehouse is implemented since that will affect the OLAP operations. Furthermore, in relational data model, data are stored in tables; but in multidimensional data

model, data are stored in n-dimensional spreadsheet (i.e., data cubes). All these concepts (i.e., conceptual data model, database design methodology, OLAP servers, OLAP operations, data cubes, OLAP mining) are explained in more details in the following sub-chapters.

2.2.1 Conceptual data model

The conceptual model in warehouses is represented by the multidimensional view of data [1, 2, 3, 4, 5, 14, 15, 20, 25]. That model constitutes the technical basis for the calculations and analysis needed for business intelligence. It contains a set of dimensions and numeric measures that depends on the dimensions. Dimensions are the business perspective of the database and measures are data that are interested. Dimensions in a multidimensional view correspond to the fields in a relational database, while the measures (i.e., cells) correspond to the records. Numeric measures such as sales, budget, revenue, etc. constitute the subject of analysis since they numeric values that can be summarized and used to monitor the business.

The multidimensional model can be thought as a multidimensional space in which dimensions correspond to the coordinates and measures correspond to the values (i.e., cells) uniquely determined by their dimensions or a multidimensional array of numeric measures as displayed in Figure 2 below:

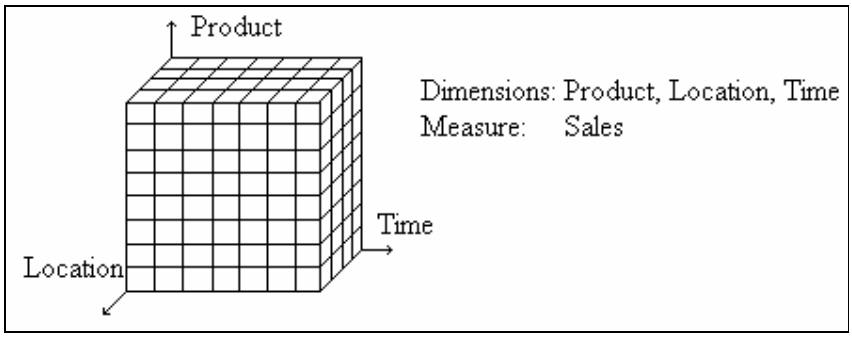


Figure 2: Multidimensional Data

Each dimension can have a set of attributes represented in a dimension table. One of these attributes is the key attribute that is used to determine the dimension. Other attributes can be used to define hierarchies on the dimension as in Figure 3.a. Hierarchies contain levels and enable summarizing specific data in a more general sense. For example, the "location" dimension can be organized as "city-country-region-all" hierarchy as in Figure 3.b. Aggregations are done over the measures by dimensions or their hierarchies as depicted in Figure 3.c. Actually, all dimensions have at least one higher hierarchical level that is the "all" level. In a dimension table, columns that do not participate in a hierarchy are member properties. They can be helpful in additional calculations with dimensions if requested in queries. To illustrate, derived variables can be calculated by these member properties, i.e., Gross Margin = Revenues – Expenses. The derived variables take up no space in the database but are computed on the fly; hence, they are useful to reduce the size of the database and reduce the consolidation time in spite of the little overhead at run time.

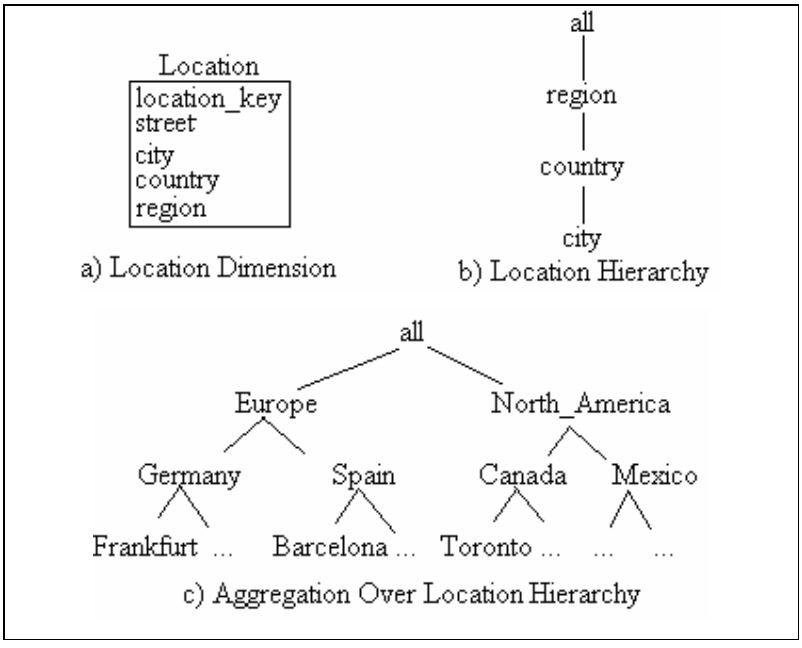


Figure 3: A Concept Hierarchy for Location Dimension

Numerical measures are classified in three types [4, 9, 12, 25]:

- **Distributive:** Measures computed by dividing data in disjoint sets, aggregating separately and combining later, i.e., max, min, count, and sum.
- **Algebraic:** Measures that can be expressed in terms of other distributive measures, i.e., average, standard deviation.
- **Holistic:** Measures that cannot be computed in parts and combined, i.e., median, most frequent, and rank.

2.2.2 Database Design Methodology

Relational databases are easily designed by entity relationship diagrams and normalization techniques which are inappropriate for multidimensional databases since efficiency in querying and loading data is very crucial. On the contrary, relational databases can be designed in such a way that they can reflect the multidimensional view of data.

The most common schema type used in relational databases for OLAP is the star schema. Star schema has one table per dimension and a fact table that contains the key attributes of dimensions and measures as depicted in Figure 4 [1, 3, 4, 5]. Hierarchies are not explicitly supported but dimensions are easily browsed by star schema. OLAP operations run faster in data-warehouses designed by star-schema than they do in data-warehouses designed by the snowflake schema.

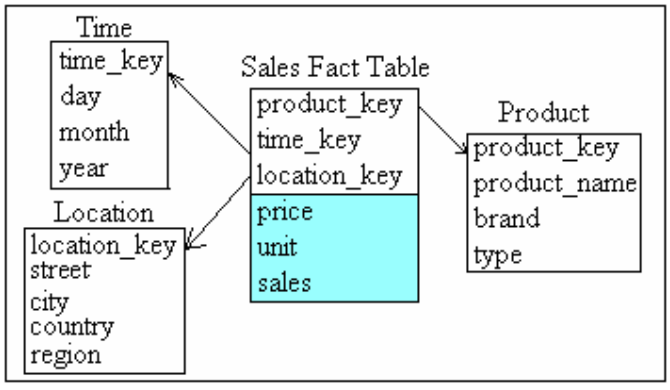


Figure 4: The Star Schema

Snowflake schemas are very similar to star schemas. But they explicitly represent their hierarchies by the normalization of dimensions. Dimension tables are more easily maintained according to star schema. On the other hand, they are inefficient with respect to star schemas since they need more join operations while running OLAP operations [1, 5]. Figure 5 below, depicts how a data-warehouse designed by the star-schema in Figure 4 would be reflected in snowflake schema.

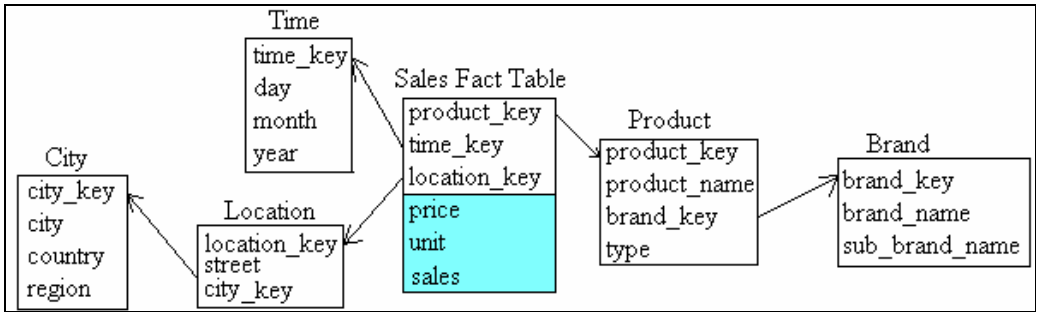


Figure 5: The Snowflake Schema

2.2.3 OLAP Servers

Data that are either in the warehouse or data marts are stored by OLAP servers [1, 20]. OLAP servers present the multidimensional view of data and are optimized for analysis. There are different OLAP servers depending on the kind of DBMS the data warehouse is implemented on, access methods and query processing techniques like ROLAP, MOLAP, HOLAP, DOLAP and JOLAP.

ROLAP (Relational OLAP) stores data in relational databases [1, 2, 3, 4, 5, 15]. They are extended relational DBMSs or intermediate servers in front of the relational DBMSs. Aggregations are stored in relational database and that requires access to the database at each query. ROLAP servers have extensions to SQL (i.e., aggregate functions like rank, percentile; reporting features like moving average; multiple group-by like roll-up and cube [11], comparisons) and special access and

implementation methods to support the multidimensional data model and OLAP operations. Mapping of OLAP queries to SQL reduces the efficiency. Standard relational query processing is supported with some indexes and pre-computation. ER diagrams and normalization techniques are inefficient, so de-normalized schemas like star schema or snowflake schema are required. A standard data model is adopted.

MOLAP (Multidimensional OLAP) stores the multidimensional data in special structures over which the OLAP operations are directly implemented [1, 2, 3, 4]. Special index structures, which are better than ROLAP, are used. Contrarily, the data cube has sparse data and that causes poor storage utilization. In order to compensate for this fact, two-level storage representation in which, sparse dimensions are indexed over dense dimensions is used. Pre-computation is common. OLAP operations run more efficiently for small databases. A standard multidimensional data model for MOLAP servers is not developed.

HOLAP (Hybrid OLAP) combines both ROLAP and MOLAP [2, 4, 5]. At the low-level, relational tables are used while at the high-level array-based multidimensional storage is preferred. HOLAP servers are useful to manage large and permanent warehouses.

DOLAP (Directory OLAP) extracts data for manipulation from a relational or multidimensional database and access them via an OLAP engine. Small amounts of data are stored in files on a user's desktop computer. To date, this is a very popular OLAP server, but the growth in the use of Web-based thin clients leads companies for solutions that move client OLAP processing to Web-based servers [19].

JOLAP (J2EE object-oriented interface to OLAP) is being developed by Java community. A standard set of object classes and methods for business intelligence are provided in that interface. JOLAP makes extensive use of the OMG Common Warehouse Model (CWM) [19].

2.2.4 OLAP Operations

The most common OLAP operations that manipulate data along dimensions are as the following [1, 2, 3, 4, 5, 15, 17]:

- **Roll-up:** Increase the level of aggregation or simply apply group-by operation on dimensions. The location dimension can be rolled up into city-country-region-all.
- **Drill-down:** Decrease the level of aggregation, increase the level of details. It is the converse operation of roll-up.
- **Slice:** Selection in dimension values. Summarized data is extracted for a given dimension value. Single item is extracted when selection on all dimensions is done.
- **Dice:** Projection in dimensions. A subcube is extracted. Several slices are intersected. Items are compared in a cross-tabulated table.
- **Pivot:** Re-orient the multidimensional view, i.e., rotate the axes. Data is examined from different angles.
- **Ranking:** Sorting.
- **Filtering:** Perform selection using some constants.
- **Defining computed attributes**.

2.2.5 Data cube

A data cube is a set of data organized as a multidimensional array of values that represents measures over several dimensions. Dimensions can have hierarchies defined over them to organize data on more than one level of aggregation.

Multidimensional data model of warehouses views data in the form of a data cube (i.e., n-dimensional spreadsheet) [4, 11, 15, 18], while relational data model views data in the form of relations (i.e., tables). Multidimensional and logical view of data is presented in data cubes in the former model. The term “multidimensional

databases" is sometimes used to refer to "data cubes" [17, 18]. By the use of data cubes, data can be modelled and viewed in multiple dimensions. Data cubes are suitable for handling hierarchies and aggregations required for data mining. Data analysing needs aggregated data extracted to a file or table to make visualization of results in graphical way. Data cubes can be successfully used to present data mining results to the business analysts. That makes data cubes essential for data analysis and data mining.

A data cube should have at least one dimension, at least one measure, and only one fact table [5]. Besides the fact table and dimension tables, there are also summary tables that contain the pre-aggregated data. A data cube consists of the base data and subaggregates [1, 2]. Summary tables are the most focused part of a data cube and can be represented as seperate fact tables that use seperate shrunken dimension tables or the same fact table can be used by adding a field for the aggregation level and using "NULL" or the dummy value "ALL" as the value for attributes other than the aggregated ones [1, 11]. In order to reduce response time, most system designers consolidate totals and keep them in a relational database as depicted in Table 2 [20].

Table 2: "Sales" Summary Table

Time	Product	Location	Price	Unit	Sales
2004	oven	Ankara	33	12	396
2004	oven	Bursa	31	13	403
2004	oven	NULL	-	25	799
2004	refrigerator	İstanbul	54	8	432
2004	refrigerator	İzmir	50	6	300
2004	refrigerator	NULL	-	14	732
2004	NULL	NULL	-	39	1135
...	...	...	...	...	...

A cube also contains members from all hierarchy levels of each dimension [5]. Conceptually, the cube contains values for each measure - that are summarized at each possible hierarchy level for each dimension - and that can be computed dynamically or pre-calculated [5].

In order to generate the aggregated data displayed in the Table 2 above, the union of all possible SQL statements with group by clauses should be computed. Gray et. al. [11] has introduced the cube operator that builds a table with all aggregate values.

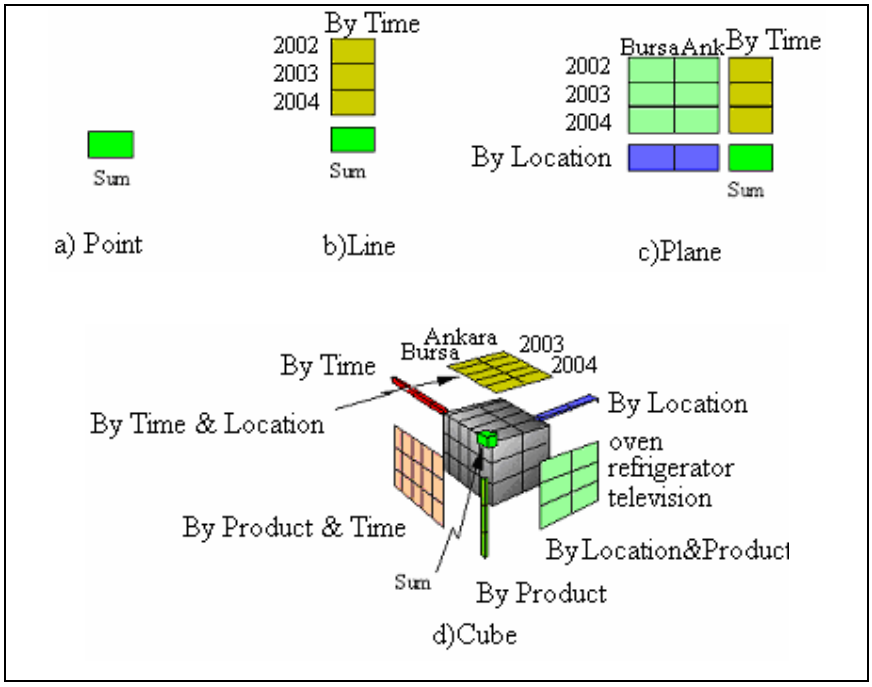


Figure 6: Data Cube

A data cube with n dimensions can have 2^n cuboids [4, 9, 10, 11] based on all possible summarizations of dimensions. These cuboids correspond to points, lines, planes, cubes or hyper-cubes beside the core data cube as depicted in Figure 6. n-D cuboid is the base cuboid since it contains all dimensions at primitive abstraction level since it is the result of applying the group-by operation to all dimensions. That

cuboid should be certainly pre-computed since it cannot be constructed from other views. 0-D cuboid is the apex cuboid and holds the highest level of summarization (i.e., no group-by's in aggregation). The lattice of cuboids corresponds to the summarized data in a data cube (see Figure 7 below). Cuboids except the "n-D cuboid" can be computed from each other. The lattice expresses the dependencies between the views. A cuboid is determined by which combination of dimensions the group-by operator is applied.

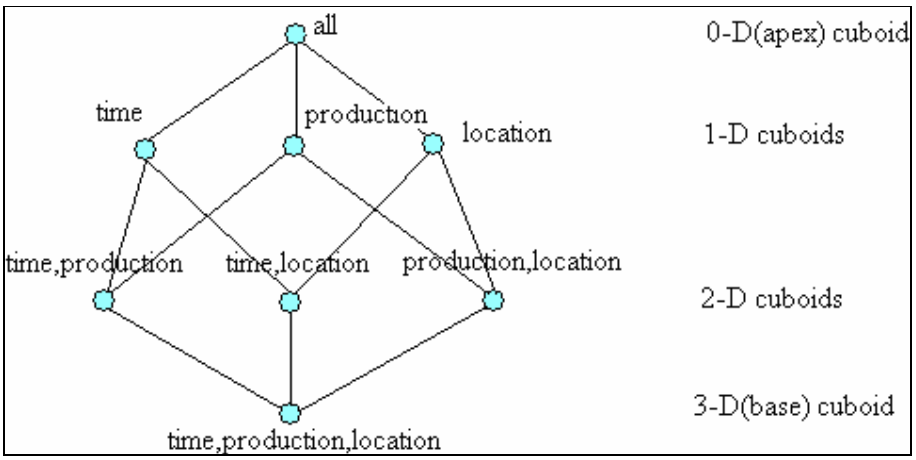


Figure 7: Cuboids of the Data Cube

When hierarchies are defined over dimensions, the lattice of cuboids becomes more complex since each hierarchy level of the dimensions is individually handled by executing the group-by operator for that level [10]. Hierarchies are also assumed to be lattices as depicted in Figure 8.

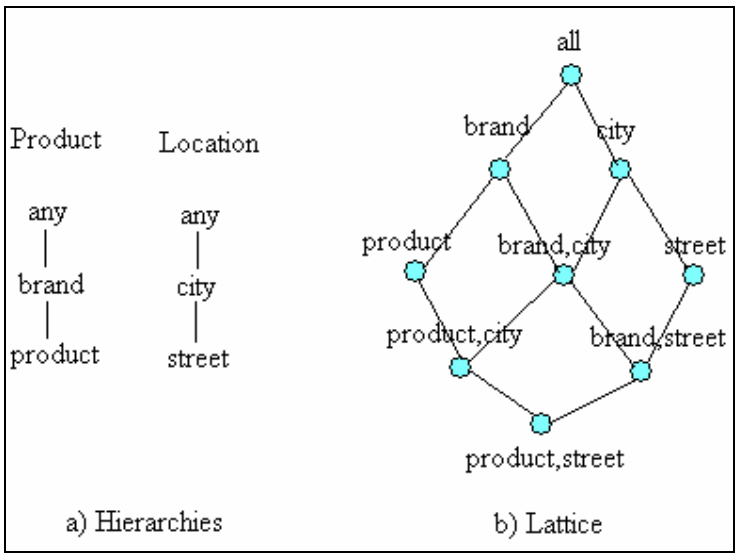


Figure 8: The Lattice of Cuboids with Hierarchical Dimensions

Materialization of summary data accelerates queries [1, 2, 4, 5]. In order to materialize views (i.e., compute cuboids), first the necessary dimensions are selected and then they are joined by the fact table. Deciding about which view to materialize depends on the workload characteristics, cost of incremental update, storage boudaries and cost of aggregation. Sometimes, the data cube may contain much more data than the fact table and this can cause to data explosion which can be avoided by not materializing the views but calculating on demand. Most of the research that is done about warehouses focuses on materialized views. Many studies are done about selection of the views to be materialized. Full, none or partial materialization can be done. Materializing all views shortens the query-reponse time but may expose huge amount of storage, while not materializing any view requires dynamic computation and no extra space. Cuboid-based selective materialization (HRU Greedy algorithm) [10] tries to decide which group-bys (i.e., cuboids) will be pre-computed and indexed. On the other hand, object-based selective materialization [9] determines which cells of cuboids should be materialized. These both methods are important studies for partial materialization which try to balance between query-response time and storage utilization.

There are some studies about efficient methods for cube computation which optimize the cube operator that is n-dimensional generalization of the group by operator [11]. Agarwal et. al. [12] has proposed some ROLAP-based cubing algorithms in which sorting, hashing and grouping operations reorder and cluster related records and grouping is done on some subaggregates as a partial grouping step. Previously computed aggregates (i.e., intermediate group-bys) are used to compute further aggregates. Array-based cubing algorithm, in which arrays are partitioned into chunks to make subcubes fit in memory, was proposed by Zhao [13]. This algorithm computes aggregates by visiting cells in the order that minimizes access count and reduces memory usage [13]. But that multi-way array aggregation works well only when number of dimensions is small. Bottom-up computation method was proposed by Bayer and Ramarkrishnan (1999) to handle large number of dimensions in multi-way aggregation [cited in 4].

2.2.6 OLAP Mining

OLAP mining is the integration of OLAP and data mining [17, 18]. Garbage data are gathered in many application areas and OLAP helps to handle this huge amount of data and to convert them to useful data. But that is not enough, right set of tools are also needed to manage the system and extract the interesting knowledge and use it in business environment. That can be achieved by not using OLAP and data mining one over the other, but by bridging them together. OLAP mining provides flexible mining of interested knowledge since data mining can be performed at multidimensional and multi–level abstraction space in a data cube. That can help users find the desired knowledge by enabling use of cubing and mining in interleaved or integrated way.

Business problems like market analysis and financial forecasting, require query-centric database schemas that are array oriented and multidimensional since they need to retrieve large number of records from very large data sets and summarize

them quickly. Data are examined from multiple points of view and that increases the need for OLAP in business environment since business problems are expressed in multiple dimensions. On the other hand, data analysts need data mining algorithms as association, classification, etc. individually or in combination in order to gain insight into business and discover interesting patterns and relationships and finally conclude in strategic and tactical decisions.

OLAP is retrospective (i.e., historical data is handled) and deductive by nature and it is driven by experts, but data mining is proactive (i.e., up-to-date rules are extracted) and inductive in nature and is driven by the data itself. These two technologies complement each other in business analytics and facilitates the interactive analyze of data. Data warehouses can provide data to both OLAP and data mining since both technologies require cleansed, consistent and integrated data [18]. Moreover, data mining algorithms can be useful to collect more meaningful meta-data model for building an OLAP cube. Furthermore, they can help in determining which cuboids are to be materialized and optimize the cost and business value of maintaining large cubes for different functional areas of the business [18].

OLAP mining areas are grouped as pre-processing areas (cube-building stage) and post-processing areas (cube- analysis stage). Pre-processing areas involve principal component analysis, clustering or association among data mining techniques since they can assist in dimension reduction, which is considered when the data cube is built. Post-processing of cubes involves modeling or prediction using neural networks or fuzzy logic, analysis of interested data using sophisticated statistical methods. Mainly, data mining and OLAP are used together when a sub-cube is selected by OLAP operations and several data mining algorithms are used to clarify various business questions since a data cube presents logically grouped views and aggregations at different levels appropriate for these questions [18].

Data mining tools need integrated, consistent and cleaned data as in data warehouses. Data analysts may select the desired portion of data by OLAP operations like slice

and dice and analyze the data at the desired level of abstraction and extract interested knowledge. Moreover, using OLAP and data mining concepts together helps users to predict what kind of knowledge can be mined by using different data mining tasks dynamically. These are the reasons why OLAP and data mining needs to be used together.

OLAP mining functions can be described as below [17]:

- **Cubing then mining:** By the use of OLAP operations the interested portion of the data is selected and the required abstraction level is determined. Then the desired data mining process is executed.
- **Mining then cubing:** After data mining is performed on the data cube, mining results can be analyzed by OLAP operations.
- **Cubing while mining:** Initiate cubing operations during mining when similar mining operations are performed at multiple granularities like mining association rules and drilling down along a dimension.
- **Backtracking:** The mining process backtracks a few steps or to a marker and explores alternative mining paths.
- **Comparative mining:** Alternative data mining processes are compared and quality of different algorithms are seen.

An OLAP based mining system, DBMiner, is already developped for characterization, association, classification, prediction, clustering and sequencing [24]. Furthermore, OLAP vendors are tending to develop an interface for Web services messaging between OLAP and data mining applications and include that feature in a service provider such as an OLAP engine [19].

2.3. Fuzzy OLAP

The need to handle imperfect data that are either uncertain or imprecise and run flexible queries on warehouses has motivated studies on fuzzy OLAP. Fuzzy OLAP

enables the extraction of relevant knowledge in a more natural language and give results to the queries with a certain precision about the reliability of the knowledge.

Fuzzy decision trees can be constructed with multidimensional DBMS to deal with real world data and perform OLAP based mining [14, 16]. In other words, OLAP mining and fuzzy data mining can be combined to increase the understandability of the discovered knowledge since fuzzy set theory treats numerical values in more natural way. Moreover, it helps to extract more generalizable rules since numerical data is manipulated with words.

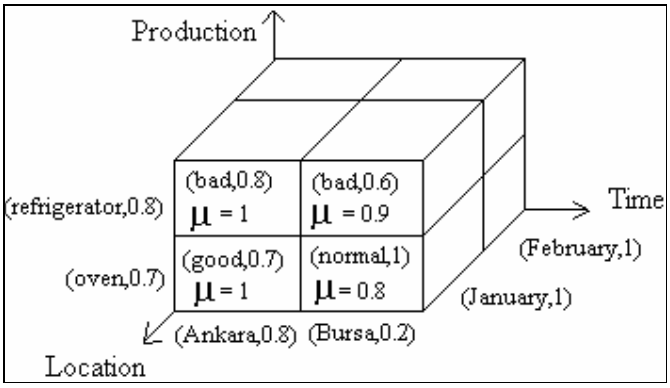


Figure 9: Fuzzy Data Cube

A new model is proposed for fuzzy multidimensional databases from which fuzzy summaries are generated like “Most sales are medium: 0.16” in [15, 16]. Here, 0.16 corresponds to the truth value of the summary. Methods used to generate fuzzy summaries are based on algorithms proposed for association rules. Fuzzy set theory is also used to represent fuzzy quantifiers like few, most, etc. In this model, domains of dimensions are a finite set of elements which consists of fuzzy set label/precise value and the degree of confidence/estimated correctness. For example, the domain of the dimension “product” can be defined as $Domain_{Product}$ = {(oven,0.7), (television,1), (refrigerator,0.8)}. That means slices of the cube belong to the cube at some extent as the slice corresponding to “oven” along production dimension belongs to the cube with degree 0.7. Additionally, each cell in the fuzzy cube,

belongs to the cube with a degree, i.e., confidence value as displayed in Figure 9 above. She represents the cube as Product x Location x Time = Sales x [0,1]. The measure values are also fuzzified with fuzzy labels and their membership values for each cell.

Fuzzy hierarchies are represented to indicate that some values may gradually belong to more than one of the defined higher levels like in Figure 10 below [15, 16].

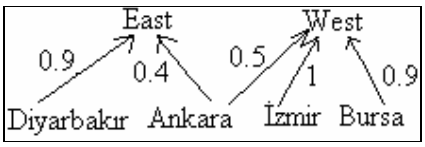


Figure 10: Fuzzy Hierarchies

During aggregation, all membership values (or membership values greater than a threshold value) of the cells whose corresponding slices belong to the cube with a degree greater than a predefined value are summed in order to compute the degree to which the aggregated cell belongs to the cube [15, 16].

While generating summaries like "Q y_i are S : τ" where Q is a quantifier, y_i is an object (i=1,...,n), S is the summarizer and τ is the degree of truth of the summary, arithmetic cummilation of membership values of the summarizer (membership of measure values) is computed as $\tau=\mu((1/n)*(\sum \mu_S(y_i)))$ [16].

2.4. Spatial OLAP

The increase in spatial data and human limitation in analyzing spatial data in detail and needs in development of geographical information systems, medical imaging and robotics systems make knowledge discovery in spatial databases very crucial [8]. This involves introduction of spatial components in relational and object-relational databases which necessitates the extension of data warehouses and OLAP for spatial

data. Because spatial OLAP should provide efficient spatial OLAP operations for the summarization and characterization of large sets of spatial objects in different dimensions and at different levels of abstraction. Spatial objects should have fast and flexible representation for their collective, aggregated and general properties.

In the following sections, knowledge discovery from spatial data and spatial data cube construction for spatial data warehouses are explained in more details.

2.4.1 Spatial Knowledge Discovery

Knowledge discovery in spatial databases corresponds to the extraction of interesting spatial patterns and features, general relationships between spatial and non-spatial data, and other implicit general data characteristics [6, 8].

Spatial data like maps, images from satellites, video cameras, medical equipment, etc. have both non-spatial and spatial components. Non-spatial components correspond to the usual data that can be stored in relational databases. Spatial components correspond to multi-dimensional data that are stored in spatial data structures [8].

Spatial objects [6, 7, 8, 9, 21] have non-spatial attributes linked to them. Spatial data is usually stored as thematic maps. They have two structure-specific representations. In raster representation, an attribute value is associated with each pixel as <<x, y>, attribute>; while in a vector representation, an object is specified by it geometry and associated thematic attributes as <geometry, attribute>. The geometry of the object is represented as a sequence of points. The vector format has advantages over the raster format like high accuracy, compactness, and easiness of object identification and manipulation. There are also some vendor-specific representations.

A generalization-based knowledge discovery mechanism is developed to integrate attribute-oriented induction on non-spatial data and spatial merge and generalization of spatial data in [6]. Generalization rules that are general data characteristics and/or relationships are extracted. Induction is done via ascending the thematic concept hierarchies and spatial hierarchies. General data characteristics and relationships between spatial and non-spatial attributes are summarized at high levels. For example, suppose that a hierarchy is defined as "corn", "wheat" and "rice" being the leaves and "grain" being their parent, regions that grow corn, wheat and rice can be generalized as "grain-production-area". Moreover, regions with precipitation measurements between 2.0 and 5.0 can be generalized as "wet-area".

The corresponding generalization rules are extracted by two generalization algorithms depending on which component is first generalized, spatial or non-spatial. To extract general knowledge from spatial databases, generalization is performed both on spatial and non-spatial data. When one component is generalized, the other component is adjusted accordingly as described below [6, 8, 23].

- **Non-Spatial Data Dominated Generalization:**

  Input: A spatial database consisting of both non-spatial and spatial data, a learning request and a set of concept hierarchies.

  Output: A rule characterizing general properties of spatial object

  Method: Related non-spatial data is collected by a SQL query. Attribute oriented induction is applied repeatedly by ascending the concept hierarchy, merging identical non-spatial tuples and collecting spatial object pointers. Generalize the spatial data by retrieving spatial objects for each generalized non-spatial tuple and performing spatial merge possibly for neighbor objects that belong to the same generalized tuples. Induction is done until a value for every attribute is generalized to the desired level, which is specified by the generalization threshold for that attribute.

- **Spatial Data Dominated Generalization:**

  Input: A spatial database consisting both of non-spatial and spatial data, a learning-request, a set of concept hierarchies, and a spatial hierarchy.

  Output: A rule characterizing general properties of spatial object

  Method: Related spatial data is collected by a SQL query. Spatial oriented induction is applied by ascending the spatial hierarchy, merging identical non-spatial tuples. Generalization of spatial objects is done until the spatial generalization threshold, which is the maximum number of regions in a generalized relation, is reached. Generalized relation is a table that contains the generalized values (i.e., labels) for attributes. The non-spatial data for each generalized spatial object is retrieved and analyzed by attribute-oriented induction. Concept hierarchy is climbed and attribute values are changed to generalized values and identical tuples are merged. Generalization threshold determines when the generalization process will stop.

These two algorithms were used in GeoMiner system [23, 24], which is a spatial data mining system. In that system, a set of characteristic rules are found at multiple levels of abstraction from a relevant set of data in a spatial database.

2.4.2 Spatial Data Cube

Spatial data-warehouses enable spatial data analysis and spatial data mining. Spatial data-warehouses couples both spatial and non-spatial technologies. Stefanovic et. al. [9] have studied methods for spatial OLAP by combining non-spatial OLAP methods and spatial databases. They propose a model for spatial data warehouses that has both spatial and non-spatial dimensions and measures and a method for spatial data cube construction called object-based selective materialization. They extend the data-warehouse definition of Inmon (1992), cited in [1, 4, 17] to spatial data-warehouse as "subject-oriented, integrated, time-variant, and non-volatile collection of both spatial and non-spatial data in support of management's decision-making process".

A spatial data-warehouse may be used to view the weather patterns on a map by region, month, and different combinations of temperature and precipitation like “hot and wet regions in the Marmara Region”. Spatial objects are summarized and characterized in different dimensions and at different level of abstraction by spatial OLAP operations in spatial data cubes.

In a spatial data cube both dimensions and measures may contain spatial components [4, 9, 27]. Dimensions can be non-spatial, spatial-to-non-spatial, and spatial-to-spatial. A non-spatial dimension contains non-spatial data (i.e., temperature, precipitation), generalization of which is non-spatial (i.e., hot, wet). Spatial-to-non-spatial dimension has spatial data (i.e., state) and is generalized to non-spatial data (i.e., big_state). Spatial-to-spatial dimension has spatial data (i.e., equi-temperature-region) and all generalized data are spatial (i.e., 0-5_degree_regions). Measures are classified as numerical if they have numeric data or spatial if they contain a collection of pointers to spatial objects. Spatial measures distinguish a spatial data cube from a non–spatial data cube since OLAP operations for spatial dimensions would be handled in similar way in the non-spatial data cubes.

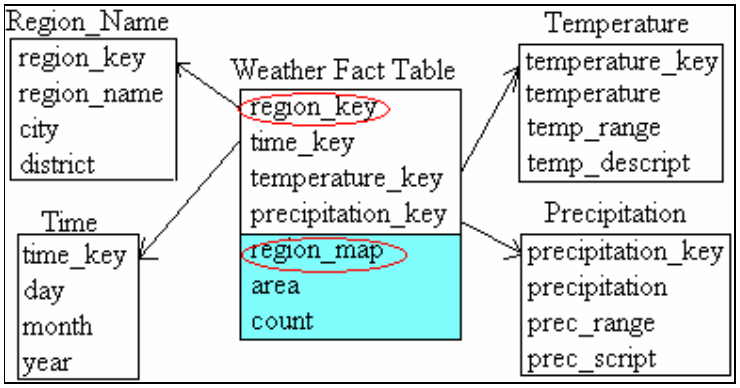


Figure 11: The Star Schema for Spatial Data Warehouse

Star/snowflake schemas can be used to design spatial data cubes. In the star schema shown in Figure 11 above, Region_Name, Time, Temperature and Precipitation are the dimensions and region_map, area and count are the measures. Hierarchies for

each dimension are shown in Figure 12 and an example for them is depicted in Figure 13. Region_map is a spatial measure that contains spatial pointers for the regions, area is a numerical measure and contains the sum of the areas of the regions, and count is also numerical measure that keeps the count of regions aggregated.

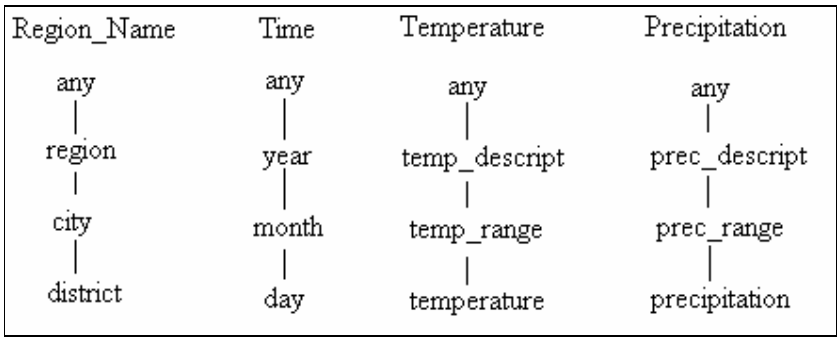


Figure 12: Hierarchies for the Star Schema

Numeric values of temperature or precipitation are first generalized to ranges and then to more descriptive names as in the example illustrated in Figure 13.

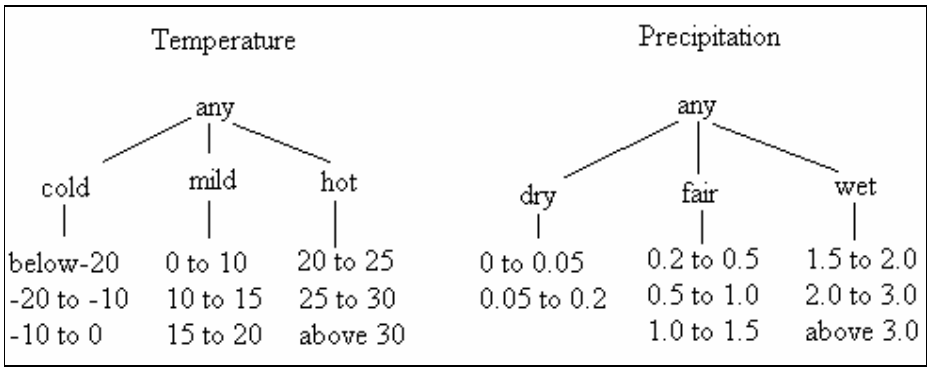


Figure 13: Example Data for Hierarchies in Star Schema

The result obtained when a roll-up operation is applied by the use of the hierarchies defined as in the Figure 12 above is illustrated in Table 3 below:

Table 3: Result of a Roll-Up Operation

Region_Name	Time	Temperature	Precipitation	Region_map	Area	Count
AA00	January	hot	dry	{AK04,AK07, ... VS67}	200	21
AA01	February	cold	fair	{AG10,AG05, ... TP90}	230	20
...	...	...	...	...	...	...

Different methods for computing spatial data cubes are listed below [4, 9, 27]:

- **On-line aggregation:** Pointers to spatial objects are collected and stored in a spatial data cube. None of the cuboids are pre-computed. It is expensive and slow when data is large. Hence, it needs efficient aggregation techniques. Moreover, redundant computation is also possible.
- **Materialization of all cuboids:** All possible computations are pre-computed and stored. It brings a huge space overhead. That is the method preferred in this thesis work in order to simplify the demonstration of fuzzy spatial data cube and since the test data is not very large. In this thesis, spatial merge operation is not applied during aggregation; only the pointers to the spatial objects are collected together.
- **Materialization of cuboids roughly:** Rough approximations of spatial objects are pre-computed and stored in a spatial data cube. That brings an accuracy trade-off.
- **Selective computation:**
  - Materialize a part of the cube (some of the cuboids) [10]: Cuboids, which are to be pre-computed and stored, are determined by HRU Greedy algorithm.
  - Materialize parts of the cuboids [9]: Materialize the frequently used and shared spatial objects. Granularity for materialization is not at the cuboids level, but at the cell level. Individual cells are examined to see whether a group of spatial objects within a cell should be pre-aggregated.
  - Combine spatial indexing with pre-aggregated results [25]: In this method, a spatial index is built on the objects of finer granularity in the spatial dimension and groupings of the index are used to define a

hierarchy. That hierarchy is incorporated to the lattice model to select appropriate aggregations for materialization by HRU Greedy algorithm. Spatial aggregation is done by traversing the index in a breadth-first manner in order to compute efficient group-by queries.

- **Multi-resolution amalgamation [21, 22, 26]:** The spatial data cube is generated dynamically since aggregations are done dynamically. The resolution of a region is changed, in other words precision of spatial data can be changed to some extent since most of spatial data are stored at much higher resolutions than are necessary for some applications. Multi-resolution spatial databases try to support deriving proper level of details on-the-fly. Data is stored with the finest available level of detail in the database and the multi-resolution database is capable to reduce that level of details according to applications. Amalgamation re-classifies objects into higher level objects that form a new spatial layer. New geometries generated by amalgamation can be used in further spatial analysis. Generalization of spatial data involves object selection, object simplification, and operations like tokenization and amalgamation. The tokenization operation corresponds to the replacement of geometrically small but semantically important objects, like telephone boxes and police stations by a token. The amalgamation operation merges the similar objects like several neighbor buildings into one large object. Applying polygon amalgamation operation finds the boundary of the union of a set of polygons. This method is useful for merging spatial objects during aggregation.

2.5. Knowledge Discovery and Data Mining

A knowledge discovery process constitutes the following steps: collection of data together, cleansing the data and fitting it together, selecting the necessary data, crunching and squeezing the data to extract the essence of it and evaluation of the output and usage of it [8].

Data mining corresponds to one step of knowledge discovery, which is the extraction of implicit information from a large dataset [4, 8]. It is the extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) information or patterns from data in large databases. But in many sources the terms “data mining” and “knowledge discovery in databases” are used interchangeably. Data mining can be sub-divided into web mining, text mining, geo-spatial data mining, multimedia (sound, video, images) mining.

A data mining system can generate thousands of patterns but only few of them are interesting. What makes a pattern interesting is its understandability, validity on new or test data with a degree of certainty, usefulness, and ability to validate a hypothesis to be confirmed. That is why users should direct what is to be mined.

Data mining has functionalities like characterization (generalization, summarization, contrasting data characteristics like dry vs. wet regions), association (correlation and causality), classification, prediction, cluster analysis, outlier analysis, and trend and evolution analysis.

2.5.1 Association Rules

Association rule mining [4, 8, 18, 29] corresponds to finding frequent patterns, associations, correlations, and causal structures among sets of items or objects in information repositories. It has application areas in basket data analysis, cross marketing, catalog design, loss-leader analysis, clustering, classification, etc. These rules may be very helpful in customizing marketing program, advertisement, and sales promotion since they are able to tell rules like “People who buy butter and milk will also buy bread”.

Association rule mining was first aimed at discovering associations between items in transactional databases. Given $D = \{T_1 \ldots T_n\}$ a set of transactions and $I = \{i_1 \ldots i_n\}$ a

set of items such that any T_i in D is a set of items in I. An association rule is an implication A->B where A and B are subsets of T_i given some support and confidence thresholds. In other words, an association rule is a rule that correlates the presence of one set of items with that of another set of items. The rule has the form "Body->Head [support, confidence]" (or "If X is A then Y is B") where support is the probability that a transaction contains the Body and confidence is the conditional probability that a transaction having the Body also has the Head.

$$\text{Support of "Body->Head"} = \frac{\text{(\# of transactions containing Body}}{\text{(total \# of transactions)}}$$

$$\text{Confidence of "Body->Head"} = \frac{\text{(\# of transactions containing both Body and Head)}}{\text{(\# of transactions containing Body)}}$$

Association on a data-warehouse requires aggregation to be performed at different levels, which can be slow. Since an OLAP cube has pre-computed results (i.e., count of tuples), performing association on an OLAP cube is much faster.

In order to find associations, first, all frequent items are found. Frequent items correspond to items that are more frequent, i.e., have supports greater than the initially determined minimum support. Then, frequent items are combined into item sets. After all item sets are found, they are used to produce association rules according to the initially defined minimum confidence. While generating association rules, the most important thing is to find the frequent item sets. The most famous algorithm is the Apriori algorithm that has many variations and improvements on it. One problem with the Apriori algorithm is that it misses all item sets with recurrent items.

Spatial association rules also have the form "Body->Head [support, confidence]". Here, "Body" and "Head" can be sets of spatial or non-spatial predicates such as [4] topological relations (intersects, overlaps, disjoint, etc.), spatial orientations (left_of,

west_of, under, etc.), and distance information (close_to, within_distance, etc.). Is_a(x, large_town) ^ intersect(x, highway) -> adjacent_to(x, water) [7%, 85%] is an example for spatial association rules.

Spatial associations are mined in two steps. In the first step, rough spatial computation is done to filter out the irrelevant spatial objects. MBR or R-tree is used for rough estimation. In the second step, detailed spatial algorithm is applied to refine the mined rules. Algorithm is applied only to the objects that have passed the rough spatial association test with a value greater than the minimum support [4].

Similarly, fuzzy association rules have the form "If X is A then Y is B" where X and Y are disjoint sets of attributes and A and B are fuzzy sets that describe X and Y respectively [29]. Fuzzy association rules are more understandable to human because fuzzy sets handle numerical data better since they soften the sharp boundaries of data. The semantics of the rule is when "X is A" is satisfied it can be implied that the consequent part "Y is B" is also satisfied. Interesting rules have enough significance and high certainty factors. Significance is for the satisfiability of the item-sets and certainty is for the satisfiability of the rules.

Table 4: Record Containing Membership Values

	temperature		precipitation		area	
	label	membership	label	membership	label	membership
t_1	Hot	0.9	dry	0.2	large	0.5
t_2	Hot	0.7	dry	0.4	large	0.8
t_3	Hot	0.8	dry	0.3	large	0.2
…	…	…	…	…	...	...

While generating fuzzy association rules, first large item-sets, those with significance higher than a user specified threshold, are found. The significance is calculated by summing the votes of all records for the specified item-set and by dividing that sum

to the count of the records. A vote of the record corresponds to the production of the membership values for the fuzzy sets in A that are described for X. For example, if X = {temperature, precipitation} and A = {hot, dry}. Suppose that we have the following records T = {t_1, t_2, $t_{3\ \ldots}$} depicted in Table 4. Significance of the rule is $S_{(X,A)}$=(0.9*0.2 + 0.7*0.4 + 0.8*0.3)/3=0.23.

After discovering large item-sets, interesting rules are generated according to the certainty factor that is computed as $C_{((X,A),(Y,B))}=S_{(Z,C)}/S_{(X,A)}$ where $X \subset Z$, $Y = Z - X$ and $A \subset C$, $B = C - A$. For the example above, if Y={area} and B={large}, $C_{((X,A),(Y,B))}$ = ((0.9*0.2*0.5 + 0.7*0.4*0.8 + 0.8*0.3*0.2)/3)/0.23=0.517.

In short, if the rule has enough significance it is determined as one of the large item-sets. Then, if it also has enough certainty it is specified as one of the interesting association rules in the database.

CHAPTER 3

FUZZY SPATIAL DATA CUBE

There is huge amount of spatial data available which are not useful unless knowledge is obtained. Especially, GISs need to store, manipulate and analyze voluminous amounts of spatial data. For such systems, it is also very probable to couple the available data with remotely sensed data so as to facilitate and analyze data timely and conclude in site specific decisions.

Spatial data warehouses are very suitable for systems that need to store large amounts of spatial data and analyze them like GIS since spatial data warehouses integrate different sources together, enable characterization of spatial data, summarize data in different dimensions at different levels of abstraction, and facilitate discovery of knowledge and decision making.

Equally important, fuzzy data warehouses incorporate fuzzy logic into their multidimensional data model and construct fuzzy data cubes in order to increase the understandability and nature of the extracted knowledge. They also give results to queries with a certain precision about the reliability of that knowledge.

In this thesis, more understandable and precise knowledge generated from spatial data by construction of fuzzy spatial data cube and extraction of fuzzy association rules from the corresponding cube. In the following sub-chapters, it is illustrated how spatial data cubes and fuzzy data cubes can be harmonized together and how this benefits to spatial knowledge discovery.

3.1 Fuzzy Spatial Data Cube Construction

Data mining discovers nontrivial and interesting knowledge or patterns from data. It has functionalities like characterization, comparison, classification, association, prediction, cluster analysis and time-series analysis. In this thesis, characterization and association aspects are considered over fuzzy spatial data cube to discover precise multilevel knowledge from spatial data.

Characterization (i.e., generalization) can be used to generalize task relevant data into generalized data cube. Characteristic rules, which are extracted from a generalized data cube, summarize general characteristic of user-specified data. Similarly, characteristic rules, which are extracted from a fuzzy spatial data cube, can summarize the climate data for a region with the extension that they can also present the general characteristics of the region's climate with some precision. The raw data for one region can be generalized into concepts like cold (0.9), mild (0.7) and hot (0.5) for temperature, and dry (0.28), wet (0.72) for precipitation with the precision values that indicates the degree of reliability of the generalization. Sub-regions that are described by the same high-level concepts can be aggregated together with a recomputed precision which is the subject of this thesis.

Instead of generalizing spatial data to "[20-25] temperature regions", "[25-30] temperature regions" and "[above 30] temperature regions" and then aggregating them to "hot regions"; generalizing each spatial datum to "hot region" with the precision value for the reliability to that generalization μ_{hot} and then aggregating them to "hot regions" with a μ_{hot} for the aggregated regions is more meaningful and natural. Different temperature values will cause to generalizations with different precision values. Introducing fuzzy logic to spatial generalizations helps to have more smooth generalizations.

Association discovers a set of association rules in the form of X_1 ^ … ^ X_n -> Y_1 ^… Y_m, at multiple levels of abstraction from the relevant set(s) of data in a database. Association rule discovery necessitates the computation of support to find the frequent item sets and the computation of confidence to find the interesting rule. Computation of these factors requires the count of occurrences of the corresponding item set and the count of all item sets. The data cubes facilitate efficient mining of association rules since a count cell stores the number of occurrences of the corresponding multi-dimensional data values and a dimension count cell stores the sum of counts of the whole dimension. These count cells simplifies the calculation of support and confidence measures of the association rules. But, in fuzzy data cubes, the interesting association rules can be determined according to their significance and certainty factors, these reflects the reliability to generalization, instead of support and confidence factors which reflect the frequency of the data.

Fuzzy spatial data cube considered in this thesis combines both some features of spatial data cube and fuzzy data cube that take place in the literature and also differentiates from them at some points. These similarities and differences are explained in more details below:

- In fuzzy spatial generalization rules, besides the spatial generalization (i.e., hot), the membership value of the generalization is also computed (i.e., hot (0.96)) according to the defined fuzzy labels and membership functions. Membership functions can have values in the range [0,1] according to the fuzzy set theory, and take as input numeric values such as 20 for 20°C temperature.
- In contrast to spatial data cube construction proposed by Stefanovic [9] in which numeric values are first generalized to ranges and then to more descriptive names, in fuzzy spatial data cube they are generalized directly to their descriptive names (fuzzy labels) with the computation of their contribution to the corresponding descriptive names (membership values).

- Fuzzy spatial data cubes are very similar to the spatial data cube considering their dimensions and measures with the exception that fuzzy generalization rules are discovered at multiple levels of abstraction from spatial data by the help of defined fuzzy hierarchies.
- In fuzzy spatial data cubes, measures are also fuzzified (i.e., generalized with their precision values) as in fuzzy data cubes.
- Generalization rules for spatial data are extended for fuzzy spatial data cube since generalization is not done only according to the determined characteristic but it is done considering both the characteristic and membership values for that characteristic of the generalized data. A new membership is needed to be calculated for the generalized value in the fuzzy spatial data cube for the aggregated regions as illustrated in Table 5 and Table 6. The membership value of the generalized aggregated region is computed by considering the weight of one of the measures according to the context of the application.

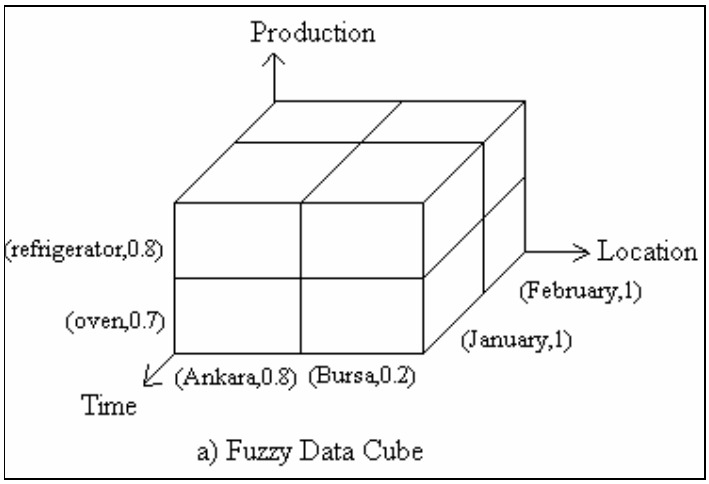
Table 5: Spatial Generalization

Region	Temperature
R1	Hot
R2	Hot
R3	Hot
R1, R2, R3	Hot

Table 6: Fuzzy Spatial Generalization

Region	Temperature	Temperature Membership
R1	Hot	0.98
R2	Hot	0.67
R3	Hot	0.56
R1, R2, R3	Hot	?

- In Laurent's study [14, 15, 16] for fuzzy cubes, each slice corresponds to the cube with a membership value, i.e., a value of one dimension has the same membership value for all the cells in the slice. But in our fuzzy spatial data cube each cell has its individual membership value for the corresponding dimension value since spatial objects might have common properties but each spatial object might have that property with a different degree than other spatial objects as displayed in Figure 14 below.



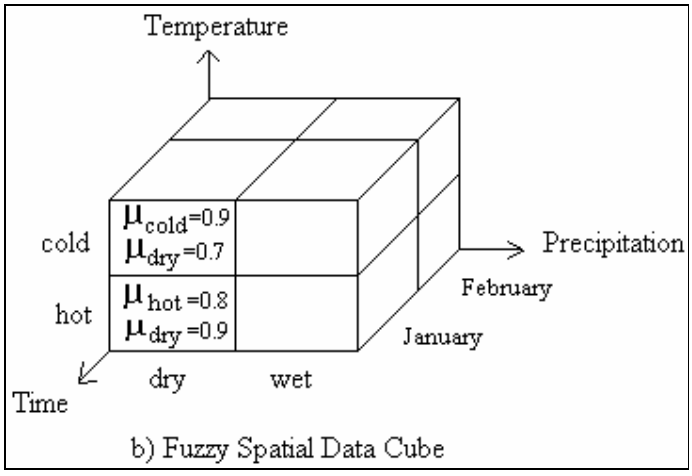


Figure 14: Dimensions and Their Memberships

- In Laurent's study [14, 15, 16]; aggregation is done by computing the degree to which the aggregated cell belongs to the cube, as explained in section "2.3 Fuzzy OLAP". That is done by computing the arithmetic cummilation of the membership values of cells that are being aggregated. On the contrary in our study, aggregation is done by multiplying membership values of each cell with the weighted value of the tuple, summing these multiplications and dividing to the sum of the weighted values. Aggregation of spatial objects in fuzzy spatial data cube will be explained in more details in the following sections.

With the use of fuzzy spatial data cubes, generalization rules such as "Ankara was %80 hot and %78 dry in June, 2003." can be easily extracted. Moreover, an additional generalization rule like "Ankara was %85 hot and %96 dry in June, 2004" could help to conclude that the increase in hotness for the regon Ankara has also increased dryness of weather, in other words, increase in temperature has decreased the precipitation in one year period. Incorporating fuzzy logic in spatial data cubes increases the reliability to the generalizations due to the computed precisions and helps to identify the deviations in properties of spatial regions through time.

The construction of fuzzy spatial data cube constitutes of the following steps:

3.1.1 Collection of spatial data

In an enterprise spatial data mining system, spatial data, non-spatial data and concept hierarchies would be stored separately as in the GeoMiner system [23, 24]. Spatial data and spatial concept hierarchies would be stored in spatial databases, and non-spatial data and relational concept hierarchies would be stored in relational databases. But in this study, the spatial and non-spatial input data are read from a file in XML format since the size of the test data has been kept feasible, and the concept hierarchies are determined according to the user requests. Data have vector

representation as depicted in the Figure 15 below. Geometry of objects is specified between "<topology>" tags and included between "<geoobject>" tags with the associated object attributes.

```
<geoobjects>
      <geoobject>
            <time>…</time>
            <temperature>…</temperature>
            <precipitation>…</precipitation>
            <area>…</area>
            <topology>
                  <point>
                        <x>…</x>
                        <y>…</y>
                  </point>
                  …
            </topology>
      </geoobject>
      …
</geoobjects>
```

Figure 15: Format of data

Data are read from the file into relational tables. In this work, it is assumed that multi-dimensional data are mapped on a relational database with a start schema that will be further explained in the following steps of fuzzy spatial data cube construction. For example, non-spatial attributes are read into a table as in the Figure 16 below.

id	time	temperature	precipitation	area
r1	2004-03-22	23	1.5	30
r2	2004-04-22	15	2.5	50
r3	2004-05-18	36	1.0	65
r4	2004-06-02	31	5.1	38
r5	2004-07-16	-10	0.1	70
r6	2004-08-22	13	3.0	43

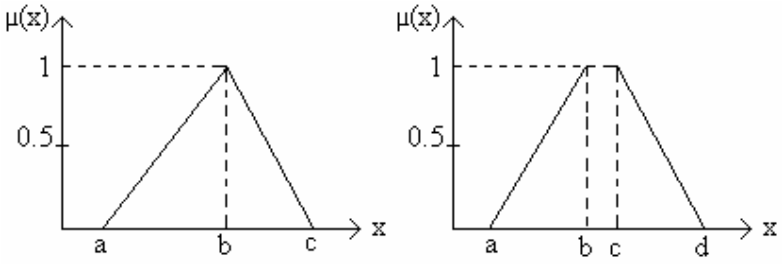
Figure 16: The Geo-object Table

3.1.2 Determination of Dimensions and Measures

After data is read from the source file, the dimensions and measures that will be included in the cube are selected. This is needed because many dimensions may be available but only the interested and significant dimensions should be included in the cube due to size and performance constraints of the data cube.

Fuzzy membership functions are defined for dimensions and for measures that are determined to be fuzzy. In the Geo-object table depicted in Figure 16 above, “id” column corresponds for the ids of the geographic objects. The dimensions and measures can be chosen among the “time”, “temperature”, “precipitation” and “area” columns. In that case, “time”, “temperature” and “precipitation” columns are more appropriate for being dimensions while the “area” column is more appropriate for being the measure. By such dimension and measure selection, weather patterns can be extracted on the map by month (or year), by temperature and by precipitation, and it can be easily seen on which regions and areas these patterns are valid.

Fuzzy membership functions are defined by using different types of membership functions [28] as displayed in Figure 17 below.



a) Triangular b) Trapezoidal

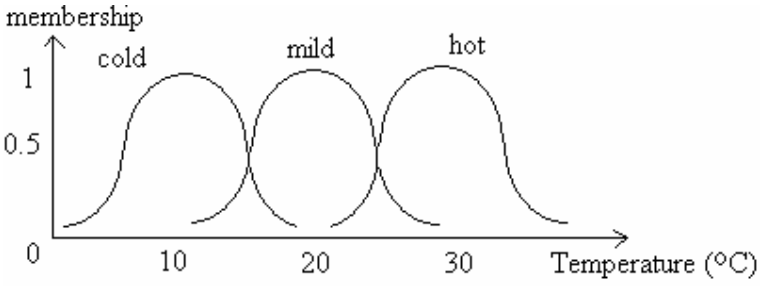
c) Gaussian d) Bell Shaped

e) Sigmoidal f) S

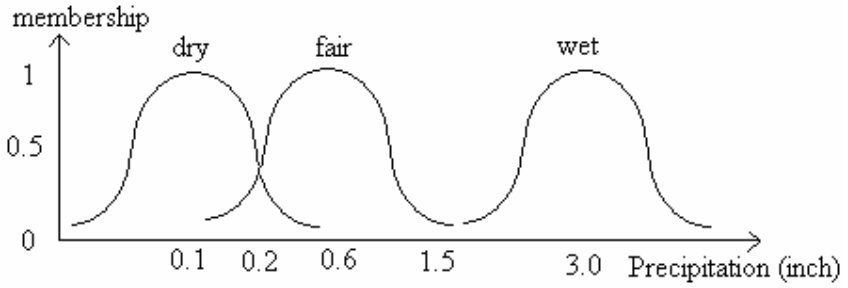
g) Π Shaped 1 h) Π Shaped 2

Figure 17: Membership Function Types

In Figure 18.a and Figure 18.b, Gaussian membership functions are depicted for temperature and precipitation dimensions. A membership function can also be defined for the measure column "area".



a) Membership for temperature dimension



b) Membership for precipitation dimension

Figure 18: The Gaussian membership function

3.1.3 Definition of Fuzzy Hierarchies

After defining the dimensions and measures and identifying their membership functions, fuzzy hierarchies can also be defined for dimensions. For example, for the temperature dimension, the fuzzy labels "hot", "cold", "cool" and "cold" and their membership functions can be defined as in Figure 18-a above. Moreover, hot and cold regions can be classified to the regions that have the season "summer", while cool and cold regions can be classified to the regions that have the season "winter". A hot region can be classified to a region that have the season summer with the

precision 1, while a mild region can be classified to a region that have the season summer with the precision 0.85 as the season fuzzy hierarchy depicted in Figure 19 below. The membership values of fuzzy labels to the upper fuzzy levels on the hierarchy are defined as constants. In that case, a mild region with the membership value 0.6 that is computed by the Gaussian membership function defined for the fuzzy label "mild", can be classified to a region that has the season winter with the membership value 0.6*0.85=0.51.

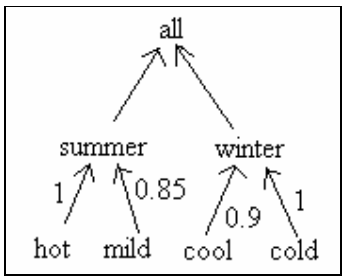


Figure 19: The season fuzzy hierarchy for the temperature dimension

3.1.4 Determination of All Aggregation Types

Before constructing the cube, all combinations of dimensions and their hierarchies (if exist) are found to be applied the group-by operator. Each of the cuboids is materialized according to each possible group-by expression. Storage penalties are not considered in this study; hence all cuboids are calculated and stored. The power set, i.e., set of all subsets, of dimensions and their hierarchies are computed as the cube operator does in [11]. For the dimensions "temperature" and "precipitation" and the hierarchy "season" defined for the temperature dimension, the group-by expressions would be found in the steps illustrated in the Figure 20 according to the pseudo code given in Figure 21 below. The complexity for that piece of code would be O(d*2d*h), d being the size of the dimensions, 2d being the count of the group-by expressions and h being the average number of hierarchical levels for dimensions.

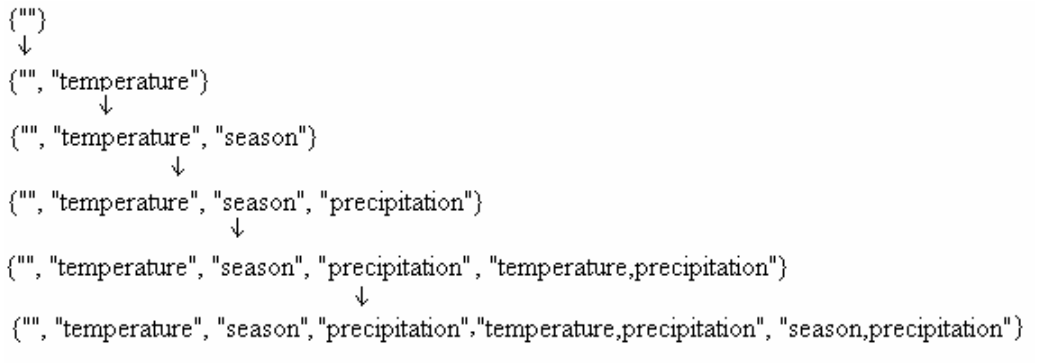


Figure 20: Group-by Expressions

```
Input: dimensions
Output: groupByExpressions

1. set groupByExpressions to an empty list
2. add an empty value to groupByExpressions
3. for each dimension
4. get dimensionName of the dimension
5.   set newExpressions to an empty list
6.   for each expression in groupByExpressions
7.      set newExpression to appendage of dimensionName to
        expression
8.      add newExpression to newExpressions
9.      get hierarchical values of the dimension
10.     for each hierarchical value of the dimension
11.        get hiearchyName of the hierarchical value
12.        set newExpression to appendage of hiearchyName to
           expression
13.        add newExpression to newExpressions
14.    end for
15.  end for
16.  add newExpressions to groupByExpressions
17.end for
```

Figure 21: Pseudo code for determining aggregation types

The spatial data cube will constitute a lattice of cuboids determined by the group-by expressions. That lattice will expresses the dependencies between the views, i.e., cuboids. Each of the cuboids will be the result of the process of one of these expressions. The lattice for the spatial data cube with the dimensions “temperature” and “precipitation” and the hierarchy “season” will be as in the Figure 22 below.

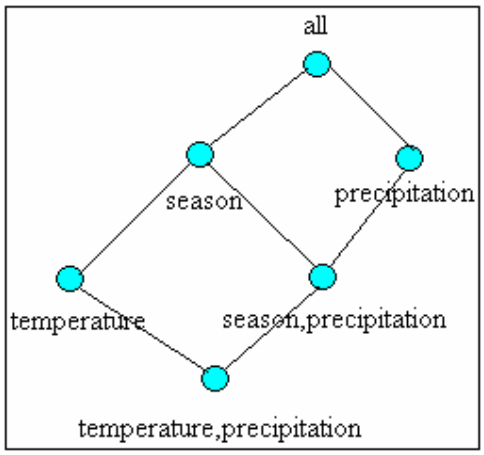


Figure 22: The lattice of fuzzy spatial data cube

3.1.5 Generalization of Dimensions using Fuzzy Logic

In the next step, dimensions are generalized to their defined fuzzy labels and their membership values are computed. For each region, all fuzzy labels defined for each dimension are considered and the label with the highest membership value dominates the other labels. For example, a region can be “hot” with precision 0.4, “cold” with precision 0.1 and “mild” with precision 0.9 considering the temperature dimension as in Figure 23 below.

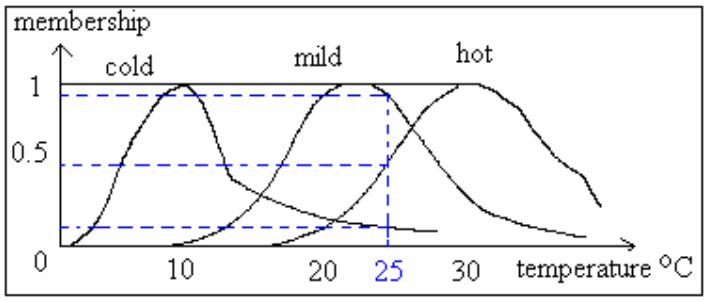


Figure 23: Choice of the most appropriate fuzzy set

In that case, that region will be considered as "mild" with the precision 0.9 since μ_{mild} is the highest membership value for the "temperature" fuzzy dimension. Furthermore, if there are hierarchies defined over the dimensions, the precision of the hierarchical level that the region belongs will be computed by the multiplication of the constant defined for the hierarchical value and the precision found for the fuzzy label. That induction will continue till the highest level of the hierarchy. For example, the "mild" region with precision 0.9 will be classified to a region that has the season "summer" with the precision $\mu_{mild}*\mu_{summer}=0.9*0.85=0.765$ since mild was classified to the "summer" hierarchical level with the contribution 0.85 in Figure 19. The pseudo code for generalization of dimensions is given below in Figure 24. The complexity of this piece of code would be O(n*d*h), n being the count of the tuples in the database, d being the size of the dimensions and h being the average number of hierarchical levels for dimensions.

```
Input: dimensions, crisp data
Output: tuples with fuzzified dimensions

1. if at least one of dimensions is fuzzy then
2.   get the crisp data schema
3.   for each dimension
4.     if dimension is fuzzy
5.       add a new column to the schema for the membership value
6.       get hierarchical values of the dimension
7.       for each hierarchical value of the dimension
8.         add a new column to the schema for the membership
           value
9.       end for
10.    end if
11.  end for
12.  create a new table according to the new schema
13.  get crisp data
14.  for each tuple in the crisp data
15.    for each dimension
16.      if dimension is fuzzy then
17.        get fuzzySets of the dimension
18.        for each fuzzySet
19.           compute the membership value of the tuple to the
              fuzzySet
20.        end for
21.        set bestFuzzySet to the fuzzySet with max membership
           value
22.        set bestFuzzyLabel to the fuzzyLabel of bestFuzzySet
```

```
23.         set bestMemVal to membership value of bestFuzzySet
24.         copy bestFuzzyLabel to the new table
25.         copy bestMemVal to the new table
26.         get hierarchical levels of the dimension
27.         for each hierarchical level of the dimension
28.            get the hierarchical value of bestFuzzyLabel
29.            get the hierarchical membership of bestFuzzyLabel
30.            set hierMemOfTuple to bestMemVal*( hierarchical
               membership of bestFuzzyLabel)
31.            copy hierarchical value of bestFuzzyLabel to the
               new table
32.            copy hierMemOfTuple to the new table
33.         end for
34.       else
35.         copy dimension value of the tuple to the new table
36.       end if
37.     end for
38.     copy measure values of the tuple to the new table
39.   end for
40.end if
```

Figure 24: Pseudo code for generalization of dimensions

The crisp data taken as input in Figure 16 is now generalized as in Figure 25 below. At the beginning the only available data for region r1 was that it had 23 ºC temperature, 1.5 gr/cm^3 precipitation and 30 km^2 area. After the generalization, we know that the region is 99% "mild" and it is 84% among regions that has the season "summer" and it has 93% wet precipitation.

temp	tempMem	season	seasonMem	prec	precMem	area	ids
mild	0.9943907906910808	summer	0.8452321720874187	wet	0.9394130628134758	30	r1
mild	0.9844964370054085	summer	0.8368219714545971	wet	0.993079612490316	50	r2
hot	0.990049833749168	summer	0.990049833749168	wet	0.8948393168143698	65	r3
hot	0.9997222607988973	summer	0.9997222607988973	wet	0.8847059049434836	38	r4
hot	0.6411803884299546	summer	0.6411803884299546	dry	1.0	70	r5
cool	0.9777512371933365	winter	0.8799761134740027	wet	1.0	43	r6

Figure 25: Generalized geo-objects

3.1.6 Fuzzy Spatial Aggregation

After determining the fuzzy labels and computing their memberships for dimensions and their hierarchies, aggregation can be performed for the regions. Aggregation is done in a separate table according to the group-by expressions previously computed. Regarding the spatial measures, the aggregation is done by the collection of the spatial pointers of the aggregated regions. Spatial aggregation, such as region merge or map overlay, may be performed for them, but this is not included in this thesis, since that requires additional computation that is out of the research of this thesis. In this study, all cuboids are materialized because mining at multiple level of abstraction is greatly enhanced when cuboids contain materialized spatial measures.

During determination of dimensions and measures, one of the measures should be defined to be the weighted measure since it will be needed during computation of aggregations. For example, while aggregating regions into one aggregated region, the area can be defined as the weighted measure and if there are some fuzzy dimensions, their membership values can be computed by summing the product of each of the aggregated membership value with that weighted measure and dividing the sum to the summation of the weighted measure (i.e., areas) of all regions that are being aggregated. The total area of the accumulated regions will be the total sum of the areas of the aggregated regions. The weighted measure is also used in the same way while aggregating according to the fuzzy hierarchies. The aggregation done according to the group-by expressions is illustrated in the Figure 26 below.

temp	tempMem	season	seasonMem	prec	precMem	area	count	ids
cool	0.9777512661246366	winter	0.8799760507982831	null	null	43.0	1	r 6
hot	0.8510133820462089	summer	0.8510133820462089	null	null	173.0	3	r3,r4,r5
mild	0.9882067918777466	summer	0.8399758100509643	null	null	80.0	2	r1,r2

a) group by temperature

temp	tempMem	season	seasonMem	prec	precMem	area	count	ids
null	null	summer	0.8475232407038391	null	null	253.0	5	r1,r2,r3,r4,r5
null	null	winter	0.8799760507982831	null	null	43.0	1	r 6

b) group by season

temp	tempMem	season	seasonMem	prec	precMem	area	count	ids
hot	0.641180419921875	summer	0.641180419921875	dry	1.0	70.0	1	r 5
hot	0.9900498610276444	summer	0.9900498610276444	fair	0.894839360163762	65.0	1	r 3
cool	0.9777512661246366	winter	0.8799760507982831	wet	1.0	43.0	1	r 6
hot	0.9997222298070004	summer	0.9997222298070004	wet	0.8847058948717619	38.0	1	4
mild	0.9882067918777466	summer	0.8399758100509643	wet	0.9729546546936036	80.0	2	r1,r2

c) group by temperature, precipitation

temp	tempMem	season	seasonMem	prec	precMem	area	count	ids
null	null	summer	0.641180419921875	dry	1.0	70.0	1	r 5
null	null	summer	0.9900498610276444	fair	0.894839360163762	65.0	1	r 3
null	null	summer	0.891419572345281	wet	0.9445355625475867	118.0	3	r1,r2,r4
null	null	winter	0.8799760507982831	wet	1.0	43.0	1	r 6

d) group by season, precipitation

temp	tempMem	season	seasonMem	prec	precMem	area	count	ids
null	null	null	null	dry	1.0	70.0	1	r 5
null	null	null	null	fair	0.894839360163762	65.0	1	r 3
null	null	null	null	wet	0.9593490458423306	161.0	4	r1,r2,r4,r6

e) group by precipitation

Figure 26: Aggregated fuzzy values

For example, there are two regions with "mild" temperature in Figure 25. When aggregation is done and "group by temperature" expression is processed these two regions will be aggregated into a larger region with the generalization "mild" temperature. The membership of that region for the fuzzy label "mild" defined for the temperature will be computed as μ_{mild} = (0.994*30 + 0.985*50) / (30+50) = 0.988. The membership of that region for the fuzzy hierarchy "season" will be computed in the similar way as μ_{mild} = (0.845*30 + 0.837*50) / (30+50) = 0.84.

The areas of these two regions will be summed and the pointers referring to them will be accumulated for the aggregated larger region as depicted in Figure 26-a above.

Here, what makes the use of the area value as the weighted column reasonable is the fact that it was defined as the interested measure before constructing the cube with the temperature and precipitation dimensions. In other words, in the construction of that cube, weather patterns in respect of temperature and precipitation were required and these patterns were for being analyzed according to regions and areas they are dispersed like "wet and hot regions in July, 2004" or "dry regions in 2004".

If we were interested in socio-economic patterns we would choose time, average income per person, educational attainment as dimensions and population as measure. In addition to population, land cover can also be a measure (there may be multiple measures to be examined), but the weighted measure should be the population. The important point while determining the weighted measure is to choose the measure on which dimensions are dependent. If the region R1 has a "high" income per person with the precision 0.9 and population 100, while the region R2 has a "high" income per person with the precision 0.7 and population 300, and these regions are generalized to a region with high income, it sounds reasonable to compute the precision of having high income of the generalized region as μ_{high} = (0.9*100 + 0.7*300) / (100+300) = 0.75. The mesaures population and land cover for the aggregated region would correspond to the summation of the population and the summation of the areas of the regions that are aggregated (i.e., R1 and R2).

The pseudo code for fuzzy spatial aggregation is given below in Figure 27. The complexity of that algorithm is $O(n^2*d*h)$, n being the count of the tuples in the database, d being the size of the dimensions and h being the average number of hierarchical levels for dimensions.

```
Input:  groupByExpressions,  measures,  tuples  with  fuzzified
       dimensions
Output: aggregated tuples with fuzzified dimensions

1. get measures
2. for each measure
3.    if measure is weighted
4.       set weightedColumn the measure
5.    end if
6. end for
7. create  new  table  for  aggregation  (with  the  schema  for
   generalized dimension)
8. for  each  expression  in  groupByExpressions  (“group  by
   temperature, precipitation”)
9.    group tuples by that expression and for each sub-group get
      9.1.  the  summation  of  production  of  memberships  of
      dimensions  in  the  expression  and  the  value  in
      weightedColumn
      9.2.  the  summation  of  production  of  memberships  of
      hierarchies of dimensions in the expression and the value
      in weightedColumn
      9.3. count of tuples
      9.4. summation of measures
10.   for each grouped tuple according to the expression (“hot,
      wet”)
11.      get pointers to regions that this tuple is grouped by
12.      for each dimension in the expression
13.         find  the  aggregated  membership  by  dividing  the
            (summation of production of memberships of dimension
            in the expression and the value in weightedColumn)
            to (summation of measures)
14.         get hierarchical levels of the dimension
15.         for each hierarchical level of the dimension
16.            find the aggregated membership by dividing the
               (summation  of  production  of  memberships  of
               hierarchical  level  and  the  value  in
               weightedColumn) to (summation of measures)
17.         end for
18.       end for
19.       insert the aggregated tuple in the new table with
          pointers, count of regions, fuzzy labels, aggregated
          memberships of dimensions and hierarchies and summed
          measures.
20.    end for
21. end for
```

Figure 27: Pseudo Code for Fuzzy Spatial Aggregation

3.1.7 Generalization of Measures using Fuzzy Logic

After the aggregation is done as in Figure 26, the measures still have numerical values. As a final step, measures can be fuzzified in a similar way the dimensions were fuzzified (if there is any measure that have been defined to be fuzzy) as depicted in the Figure 28 below. For each tuple in the aggregated table or cell in the cuboids of the cube, all fuzzy labels ("small", "large", "mid") defined for each fuzzy measure (i.e., area) are considered and the label with the highest membership value dominates the other labels. For example, for the "mild" regions (aggregated regions r1 and r2) the precisions for "small", "large", "mid" are all computed. Since the membership for the fuzzy label "mid" is found to be the highest membership for the measure area, that measure is generalized to be "mid" for the mild regions.

temp	tempMem	season	seasonMem	prec	precMem	area	count	ids	areaMem
cool	0.9777512661246366	winter	0.8799760507982831	null	null	small	1	r6	0.9951119854158298
hot	0.8510133820462089	summer	0.8510133820462089	null	null	mid	3	r3,r4,r5	0.9941394625631884
mild	0.9882067918777466	summer	0.8399758100509643	null	null	mid	2	r1,r2	0.9470111191252555
null	null	summer	0.8475232407038391	null	null	large	5	r1,r2,r3,r4,r5	0.9975031223974601
null	null	winter	0.8799760507982831	null	null	small	1	r6	0.9951119854158298
hot	0.641180419921875	summer	0.641180419921875	dry	1.0	small	1	r5	0.9607894391523232
hot	0.9900498610276444	summer	0.9900498610276444	fair	0.894839360163762	small	1	r3	0.9777512371933363
cool	0.9777512661246366	winter	0.8799760507982831	wet	1.0	small	1	r6	0.9951119854158298
hot	0.9997222298070004	summer	0.9997222298070004	wet	0.8847058948717619	small	1	r4	0.985703184122443
mild	0.9882067918777466	summer	0.8399758100509643	wet	0.9729546546936036	mid	2	r1,r2	0.9470111191252555
null	null	summer	0.641180419921875	dry	1.0	small	1	r5	0.9607894391523232
null	null	summer	0.9900498610276444	fair	0.894839360163762	small	1	r3	0.9777512371933363
null	null	summer	0.891419572345281	wet	0.9445355625475867	mid	3	r1,r2,r4	0.9886867043496658
null	null	winter	0.8799760507982831	wet	1.0	small	1	r6	0.9951119854158298
null	null	null	null	dry	1.0	mid	1	r5	0.9607894391523232
null	null	null	null	fair	0.894839360163762	small	1	r3	0.9777512371933363
null	null	null	null	wet	0.9593490458423306	large	4	r1,r2,r4,r6	0.998656458916103

Figure 28: Fuzzified measures

The pseudo code for generalization of measures is given below in Figure 29. The complexity of that piece of code is O(n*m), n being the count of the tuples in the aggregated table and m being the count of the measures.

```
Input: measures, aggregated tuples with fuzzified dimensions
Output: aggregated tuples with fuzzified dimensions and measures

1. if there are any fuzzy measures then
2.    get measures
3.    for each measure
4.       if measure is fuzzy then
5.          insert a new column to the aggregate table for the
            membership of the measure
6.       end if
7.    end for
8.    for each tuple in aggregate table
9.       for each measure
10.         if measure is fuzzy then
11.            get fuzzySets of the measure
12.               for each fuzzySet
13.                  compute the membership value of the tuple
                     to the fuzzySet
14.               end for
15.            set bestFuzzySet to the fuzzySet with max
               membership value

16.            copy fuzzyLabel of bestFuzzySet to aggregate
               table
17.            copy membership of bestFuzzySet to aggregate
               table
18.         end if
19.      end for
20.   end for
21. end if
```

Figure 29: Pseudo Code for Generalization of Measures

3.2 Meaning of Values in the Fuzzy Spatial Data Cube

The crisp data that were available at the beginning (see Figure 16) are generalized and fuzzified. The meaning of the obtained knowledge can be commented as in Table 7 below for each tuple in the table or cell in the cube in Figure 28:

Table 7: Generalizations with respect to dimensions

Generalizations with respect to temperature (and also season)
Cool (%98) regions (r6) that have the season winter (%88) cover small (%99) lands.

Hot (%85) regions (r3, r4, r5) that have the season summer (%85) cover medium sized (%99) lands.
Mild (%99) regions (r1, r2) that have the season summer (%83) cover medium sized (%94) lands.
Generalizations with respect to season
Regions (r1, r2, r3, r4, r5) that have the season summer (%85) cover large (%99) lands.
Regions (r6) that have the season winter (%88) cover small (%99) lands.
Generalizations with respect to temperature (and also season) and precipitation
Hot (%64) regions (r5) that have the season summer (%64) and dry (%100) precipitation cover small (%96) lands.
Hot (%99) regions (r3) that have the season summer (%99) and fair (%89) precipitation cover small (%97) lands.
Cool (%98) regions (r6) that have the season winter (%88) and wet (%100) precipitation cover small (%99) lands.
Hot (%99) regions (r4) that have the season summer (%99) and wet (%88) precipitation cover small (%98) lands.
Mild (%98) regions (r1, r2) that have the season summer (%84) wet (%97) precipitation cover medium sized (%94) lands.
Generalizations with respect to season and precipitation
Regions (r5) that have the season summer (%64) and have dry (%100) precipitation cover small (%96) lands.
Regions (r3) that have the season summer (%99) and have fair (%89) precipitation cover small (%97) lands.
Regions (r1, r2, r4) that have the season summer (%89) and have wet (%95) precipitation cover medium sized (%98) lands.
Regions (r6) that have the season winter (%88) and have wet (%100) precipitation cover small (%99) lands.
Generalizations with respect to precipitation

Dry (%100) regions (r5) cover small (%96) lands.
Fair (%89) regions (r3) cover small (%97) lands.
Wet (%96) regions (r1, r2, r4, r6) cover medium sized (%99) lands.

In the generalizations above, the percentage value (on the right of the fuzzy labels) tells to which extent it is possible to rely on the generalized tag. This helps to differentiate the precisions of hotness for different regions. Moreover, in the generalizations above it is clarified to which regions the word “regions” refers.

These generalizations can be used for dynamic drill-down and roll-up along any dimension to explore the desired patterns. The roll-up operation would correspond to the spatial generalization with precision value (fuzzy membership value) and the drill-down operation would correspond to the spatial specialization with precision values in the fuzzy spatial data cube. In other words, roll-up operation corresponds to progressive generalization and drill-down operation corresponds to progressive deepening. These operations can be performed easily on the generated generalized fuzzy spatial data, to see the non-spatial attributes and their precisions of the generalized spatial regions and details of the sub-regions.

3.3 Fuzzy Association Rule Generation from Fuzzy Spatial Data Cube

All of the generalization rules obtained in Figure 28 may not always be interesting. In order to obtain the interesting knowledge from the fuzzy spatial data cube it would be wise to generate fuzzy association rules from them.

In order to remind how fuzzy association rules were generated in [29], the process can be briefly summarized as the following. Fuzzy association rules were generated by first computing significance of item-sets and selecting those with a higher significance than a user specified threshold. Then rules of these selected item-sets which have a higher certainty than a user defined certainty are identified as the

interesting association rules. The significance of a fuzzy association rule that is going to be generated from a relational database was defined as the division of summation of all the votes of all tuples the specified item-set occurs to the count of such tuples.

On the other hand, spatial associations were mined in two steps. In the first step, spatial association test was done to filter relevant spatial objects with a support greater than the minimum support. Next a detailed spatial algorithm was applied to refine the mined rules.

Generating association rules from the fuzzy spatial data cube would be very useful, since the spatial data has computed precision values for its fuzzy dimensions and measures and fuzzy association rule mining is more easily computed than the spatial association rule mining. Here, it is assumed that the reliability to generalizations is more important than the frequency of the data. Hence, this will help not to miss the rules which are not frequent but significant.

In the fuzzy spatial data cube constructed in this study, tuples in which the interested item-set occurs are already aggregated to a high-level tuple. Membership values of the dimensions of that tuple are more realistic since they are computed by taking the weighed measure value in account. Here, the term “item-set” corresponds to values of the “group by expression”, i.e., “hot, dry” for “group by temperature, precipitation”. In the fuzzy spatial data cube, knowledge that will be obtained by these group-by expressions is interested and valuable.

Considering the fuzzy association rule tried to be generated from the data in Table 4 where X = {temperature, precipitation}, A={hot, dry}, Y={area}, B={large}, the significance was computed in [29] as $S_{(X,A)} = (0.9*0.2 + 0.7*0.4 + 0.8*0.3) / 3=0.23$.

In the fuzzy spatial data cube, there would be an aggregated record for “hot, dry” item-set which was going to be obtained by the “group by temperature, precipitation”

expression. The membership values of “hot” and “dry” fuzzy labels would be computed by taking into account the weighted measure value in the area column as described in the “3.1.6 Fuzzy Spatial Aggregation” section above. Hence, the vote of only that aggregated record would correspond to the significance of the item-set/group-by expression which is the multiplication of the membership values of the dimensions that take part in the group by clause.

Moreover, the certainty factor of the rule was defined in [29] as $C_{((X,A),(Y,B))} = S_{(Z,C)}/S_{(X,A))} = [(0.9*0.2*0.5 + 0.7*0.4*0.8 + 0.8*0.3*0.2) / 3] / 0.23 = 0.517$.

Here, the significance of the antecedent and consequent is divided to the significance of the antecedent. In fuzzy spatial data cube, dimensions and measures are disjoint sets as the item-sets X and Y are defined in [29]. The significance of the antecedent and consequent for an association rule in the fuzzy spatial data cube would correspond to the multiplication of the membership values of dimensions and measures that take part in the rule respectively. Since the significance of rule is the multiplication of the membership values of the dimensions, the certainty of the rule would correspond to the multiplication of the membership values of measures.

The significance and certainty factors for a fuzzy association rule that would be generated from the aggregated data in the fuzzy spatial data cube are:

$$\text{Significance} = \Pi(\mu(a_i)) \text{ where } a_i \subset A,\ 0<i<=|A|,\ X \text{ is a dimension,}$$

$$\text{Certainty} = \Pi(\mu(b_i)) \text{ where } b_i \subset B,\ 0<i<=|B|,\ Y \text{ is a measure.}$$

The fuzzy association rules are generated from the fuzzy spatial data cube, results of which were depicted in Figure 26, by the threshold values for the significance and certainty as 0.85 and 0.9 respectively as in Table 8 below. Increasing the threshold values for significance and certainty factors would decrease the number of the generated fuzzy association rules.

Table 8: Fuzzy Association Rules

Cool (0.97) => small (0.99) [0.97,0.99]
Hot (0.85) => mid (0.99) [0.85,0.99]
Mild (0.98) => mid (0.94) [0.98,0.94]
Winter (0.87) => small (0.99) [0.87,0.99]
Hot (0.99), fair (0.89) => small (0.97) [0.88,0.97]
Cool (0.97), wet (1.0) => small (0.99) [0.97,0.99]
Hot (0.99), wet (0.88) => small (0.98) [0.88,0.98]
Mild (0.98), wet (0.97) => mid (0.94) [0.96,0.94]
Summer (0.99), fair (0.89) => small (0.97) [0.88,0.97]
Winter (0.87), wet (1.0) => small (0.99) [0.87,0.99]
Dry (1.0) => small (0.96) [1.0,0.96]
Fair (0.89) => small (0.97) [0.89,0.97]
Wet (0.95) => mid (0.99) [0.95,0.99]

The association rule "hot (0.99), wet (0.88) => small (0.98) [0.88,0.98]" can be commented as "Given that an area is hot (0.99) and has wet (0.88) precipitation, it can be concluded that it covers small piece of land with the certainty %98 while the significance of that area being both hot and wet is %88."

Furthermore, comparing fuzzy association rules obtained at different times would be helpful in determining the deviations seen along time. For example, considering the fuzzy association rules "In 2003, hot (0.99), wet (0.84) => small (0.98) [0.83, 0.98]" and "In 2004, hot (0.83), wet (0.92) => small (0.7) [0.76, 0.7]"; it can be concluded that the degree of hotness of the temperature has decreased in 2004, while the degree of wetness of the precipitation has increased and hot and wet areas have enlarged. It can also be concluded that the probability of regions being both hot and wet has decreased and given that a region is hot and wet it is less probable that it covers small area.

The algoritmic description of the fuuzy spatial association rule is given below in Figure 30. The complexity of that piece of code is $O(n*2^d)$, n being the count of the tuples in the aggregated table, d being the count of the dimensions and 2^d being the count of group-by-expressions.

```
Input: threshold significance, threshold certainity, aggregated
       tuples with fuzzified dimensions and measures,
       groupByExpressions
Output: fuzzy spatial association rules

1. for each tupple in aggregate table
2.    get groupByExpressions
3.    for each expression in groupByExpressions
4.      determine the tupple is aggregated by that expression.
        (membership values for dimensions in the group by
        expression should not be null and the tuple should not
        have dimensions that do not take part in the group by
        expression)
5.    end for
6.    compute the significance
7.    if significance is bigger than the thershold significance
      then
8.      compute the certainity
9.      if certainity is bigger than the thershold certainity
        then
10.        display the rule (according the expression it is
           aggregated)
11.     end if
12.   end if
13. end for
```

Figure 30: Pseudo code for generating fuzzy spatial association rules

3.4 Implementation

In this thesis, the fuzzy spatial data cube construction was implemented in Java as a Java Applet, in Eclipse 2.1 environment with JRE1.4.2_04 and on Microsoft SQL Server 2000 (Enterprise Manager) database. These two environments were combined by Microsoft SQL Server Driver for JDBC access.

The architecture of the application developed for fuzzy spatial data cube construction consists of three components as GUI, business logic, and the database component as shown in Figure 31 below.

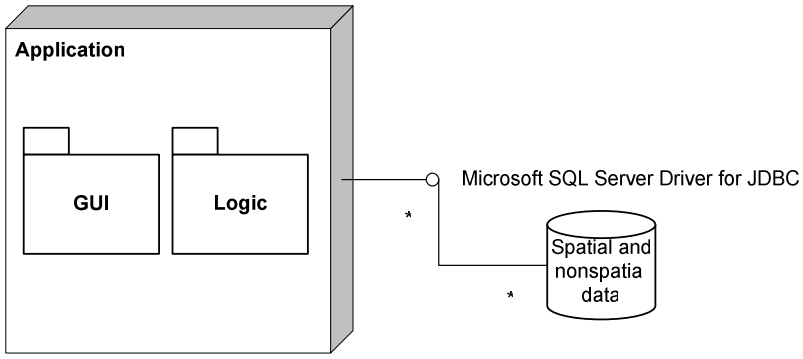


Figure 31: Deployment Diagram

The GUI component consists of Java Swing Panels that are organized as in the Figure 32 below.

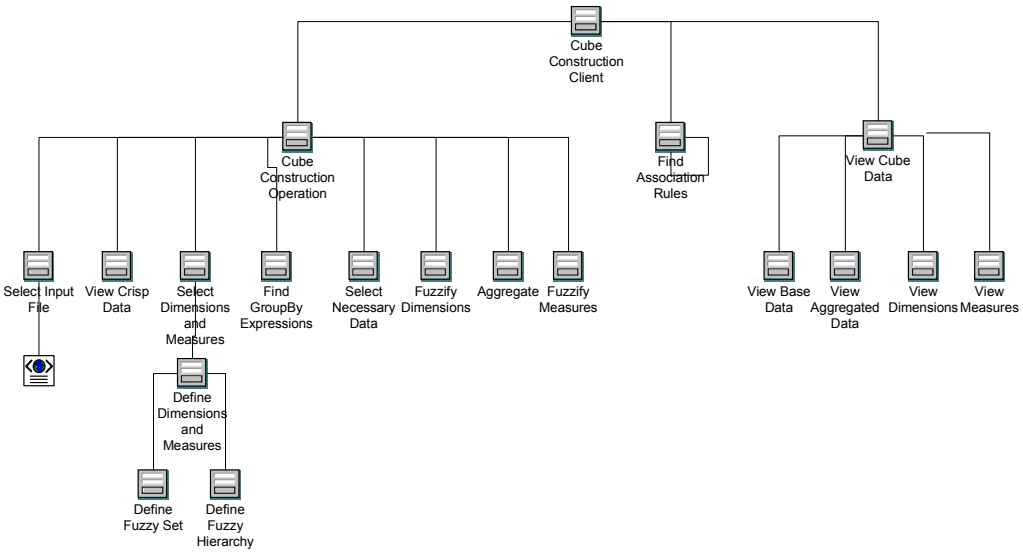


Figure 32: Interfaces

The logic component consists of Java files and is displayed in Figure 33 below.

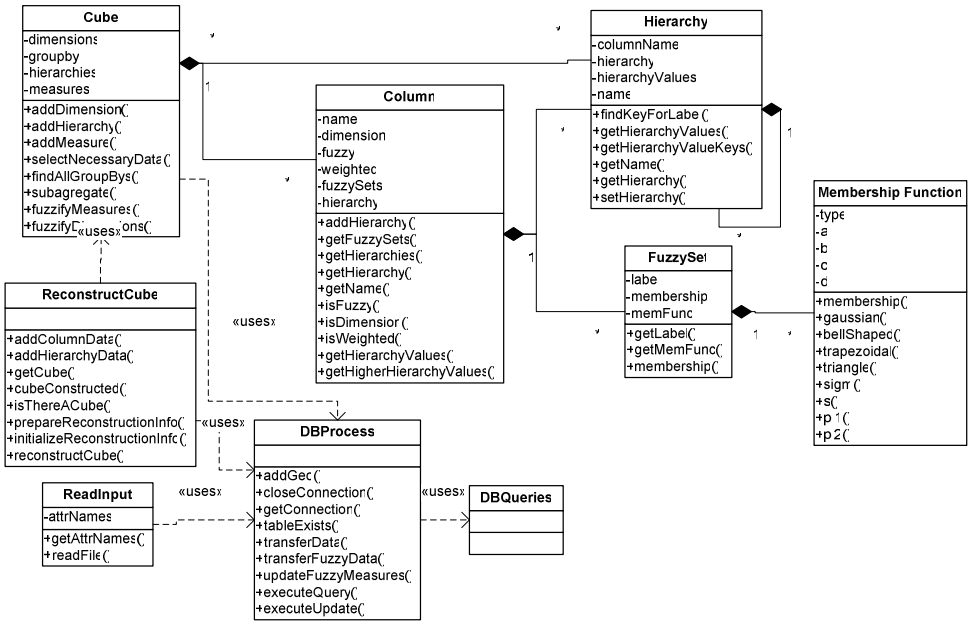
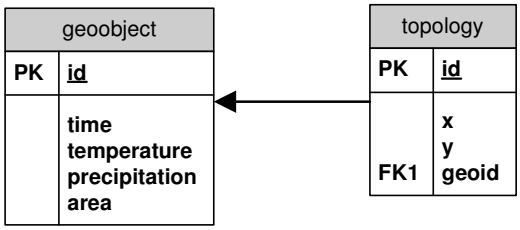
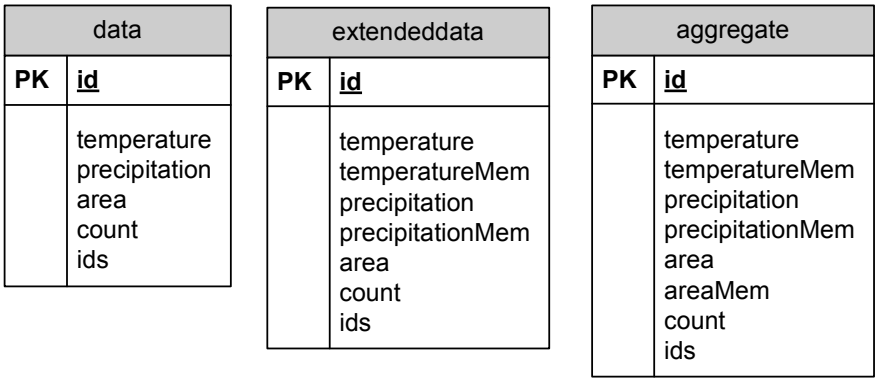


Figure 33: The Class Diagram

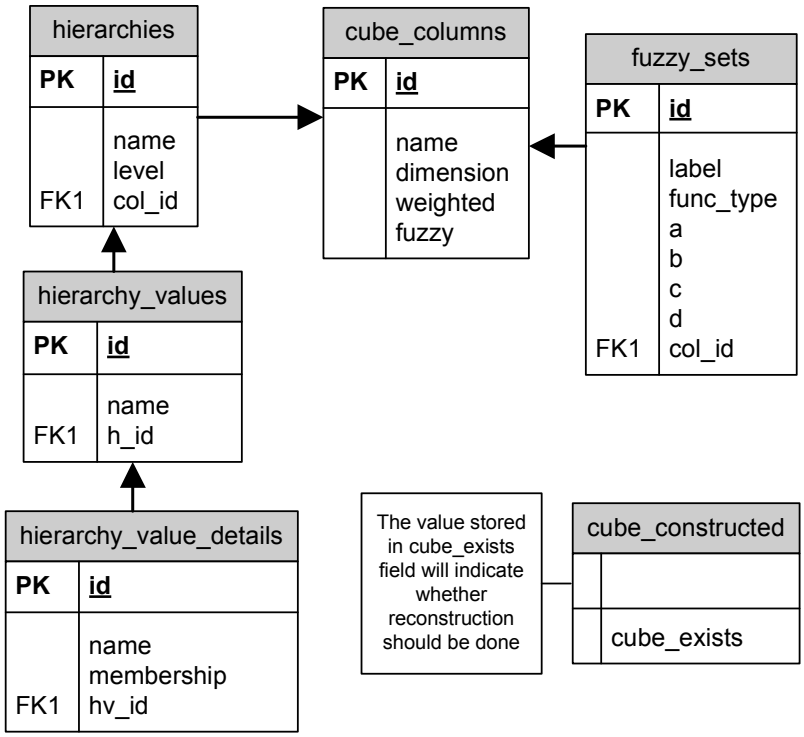
The database component corresponds to the Microsoft SQL Server 2000 database on which geographic objects are stored in the form vector data. Geographic objects, crisp data, generalized data, aggregated data and cube reconstruction information are stored as depicted in the Figure 34 below:



a) Geographic objects



b) Crisp, generalized and aggregated data



c) Reconstruction info

Figure 34: Database Design

Some performance metrics for construction of the fuzzy spatial data cube and generating fuzzy association rules from it are listed in Table 9 for different sizes of

spatial data. Threshold values for significance and certainty are 0.7 and 0.9 respectively.

Table 9: Performance Metrics

Metrics	**Values**					
Count of objects	**6**	**50**	**100**	**250**	**500**	**1000**
Time for fuzzifying dimensions (sec)	0.24	0.49	0.38	0.68	1.45	2.71
Time for aggregation (sec)	0.57	0.56	0.86	0.94	0.77	1.17
Time for fuzzifying measures (sec)	0.24	0.21	0.25	0.24	0.90	0.60
Time for association rules (sec)	0.82	0.30	0.30	0.30	0.50	0.40
Count of aggregations	15	20	27	27	27	27
Count of association rules	1	2	5	6	4	3

CHAPTER 4

CASE-STUDY: "Weather Pattern Searching"

A case-study has been implemented in order to illustrate how a fuzzy spatial data cube can be constructed and how fuzzy association rules can be generated from it according to the implementation details explained in the previous chapter. This case-study application is illustrated in the following figures. From the start page it is possible to construct a new cube, or view the constructed cube and generate association rules from it provided that there is a previously constructed cube as in Figure 35 below.

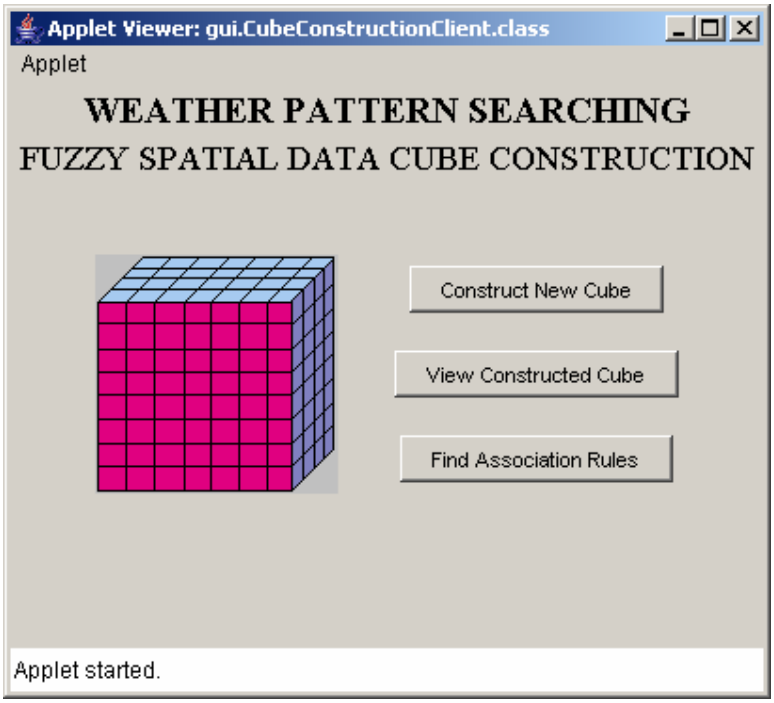


Figure 35: Start page of the fuzzy spatial data cube construction application

These three scenarios are illustrated in the following sections.

4.1. Construction of New Fuzzy Spatial Data Cube

Click “Construct New Cube” button in the start page. In the first step, an input file in XML format is selected as in Figure 36 below. In that file geographical objects are kept in vector format as in Figure 37.

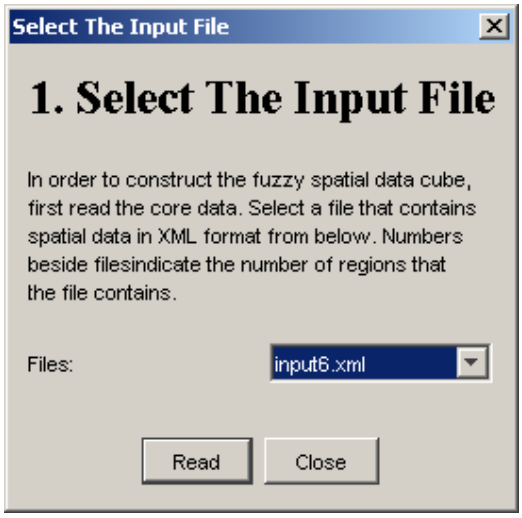


Figure 36: Choice of input file

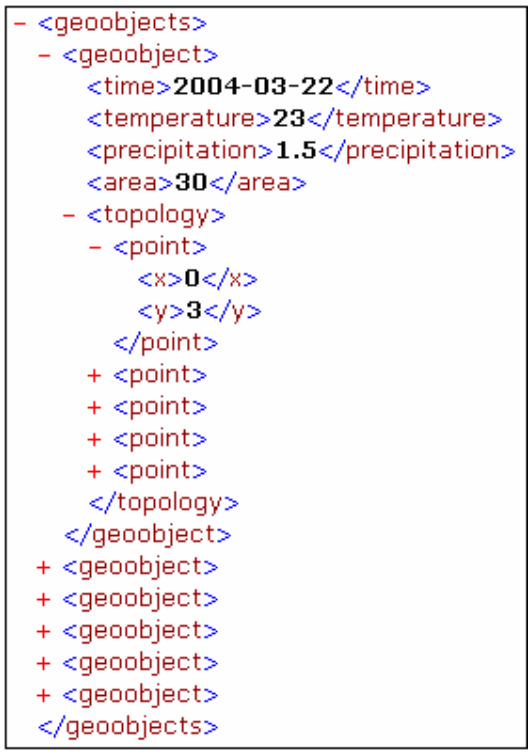


Figure 37: Format of the input file

Next, a summary of the data read from the file is given as in Figure 38 below.

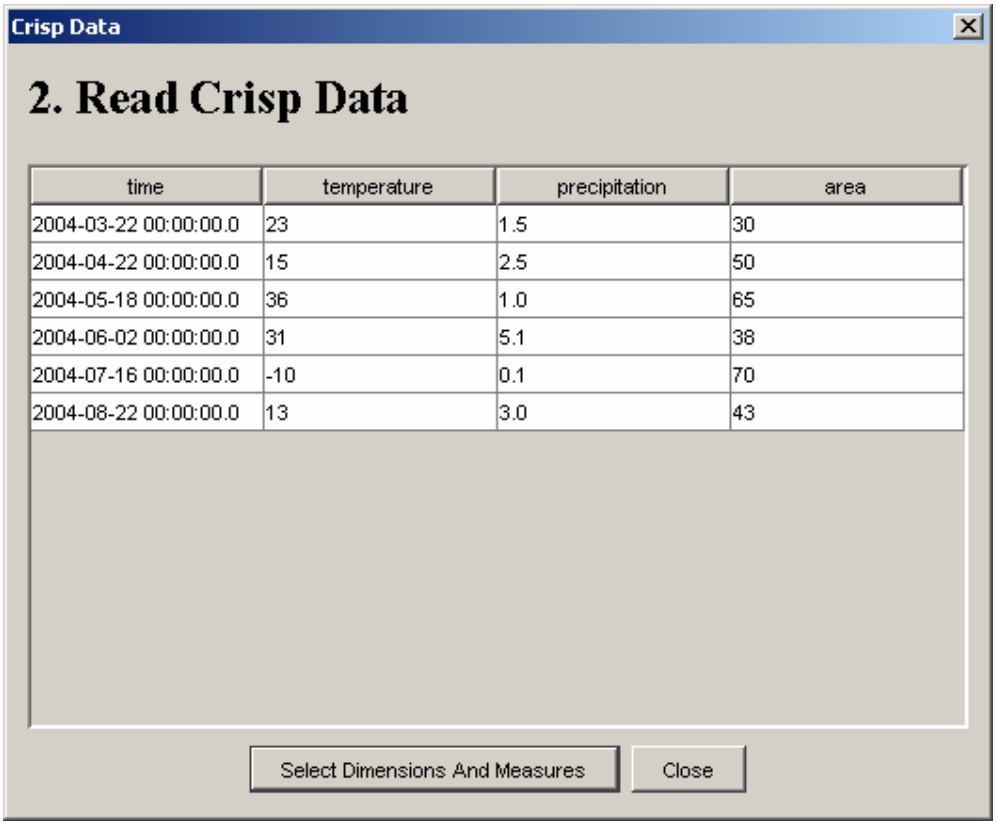


Figure 38: Summary of the data read

After data is read, the dimensions and measures are determined according to the properties of the geographic data as in Figure 39 below.

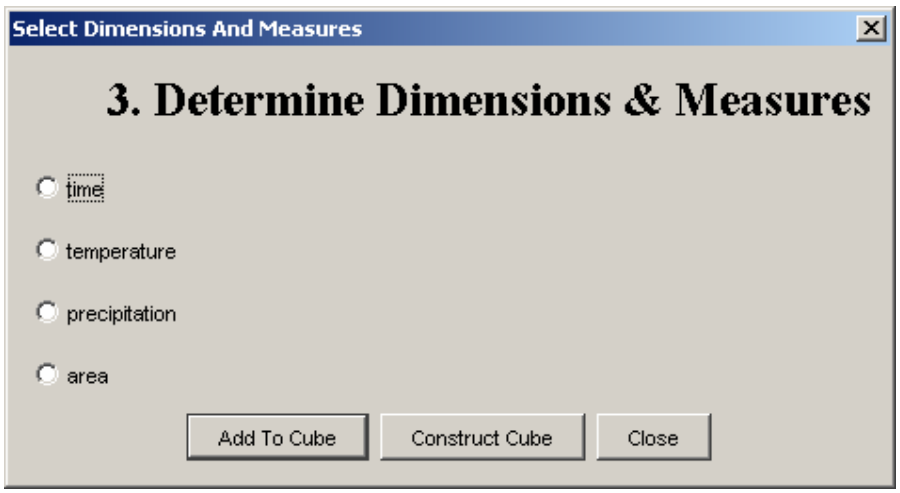


Figure 39: Candidates for dimensions and measures

While selecting a dimension or measure for the cube, one of the geographic properties listed in Figure 39 above is chosen and "Add to Cube" button is pressed. A new window as in Figure 40 opens for that property. In that window it is determined whether that property is a dimension or measure; whether it is fuzzy or not; if it is fuzzy what are the fuzzy labels, membership function types and membership function parameters, if it is a measure whether it is weighted or not. A fuzzy spatial data cube should have at least one dimension and at least one measure. One of the measures must be a weighted measure.

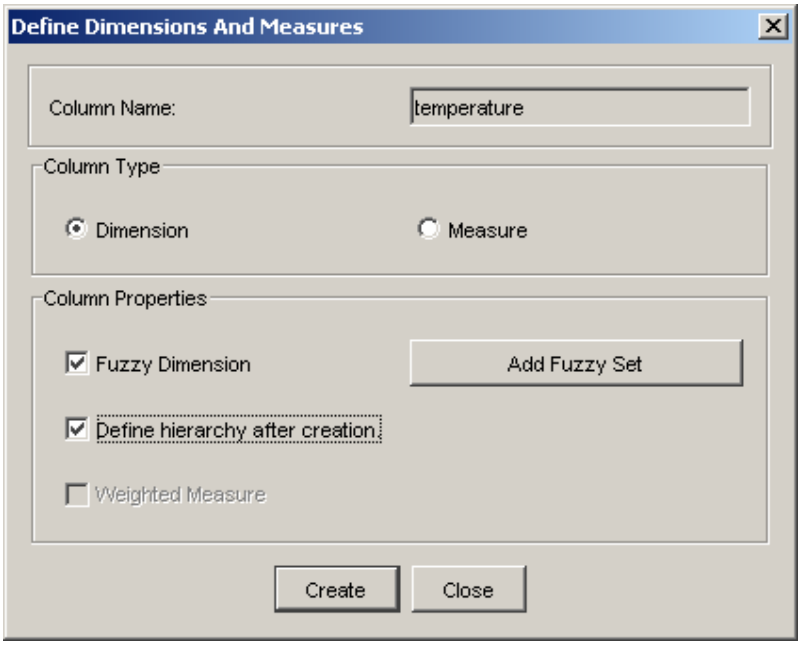


Figure 40: Definition of dimensions and measures

If the column that is going to be added to the cube is a fuzzy dimension or fuzzy measure, fuzzy sets should be defined for that column. By clicking "Add Fuzzy Set" button a new panel will be open for fuzzy set definition as in Figure 41 below. In that panel fuzzy label, membership function type, and membership function parameters are defined. Multiple fuzzy sets can be defined for a fuzzy column (dimension/measure). After all fuzzy sets are defined "Create" button should be clicked.

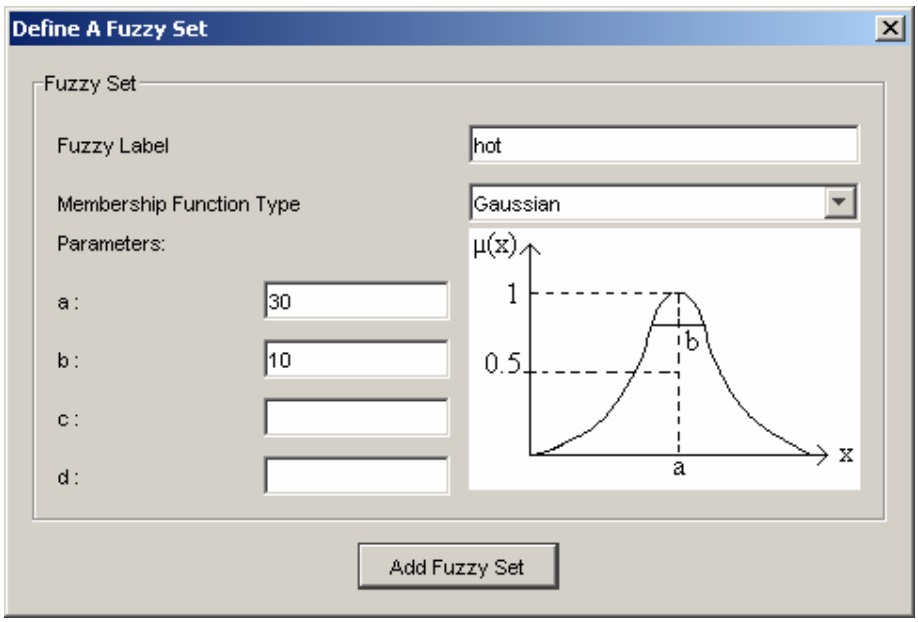


Figure 41: Fuzzy set definition

If a hierarchy is going to be defined for a fuzzy dimension the check box "Define hierarchy after creation" should be checked before clicking the "Create" button. If that check box is checked then a panel for hierarchy definition will open as in Figure 42. In that panel hierarchy level name is defined. That hierarchy level can have multiple hierarchy values each of which should be added by that panel. A hierarchical value is defined over the fuzzy sets of the column. Choose the fuzzy labels the corresponding hierarchical value is defined over and indicate the membership of these fuzzy labels to the hierarchical value and click the "Add Value" button. Next, define further hierarchical values over the remaining fuzzy labels as in Figure 43. When all hierarchical values are added click the "Create Hierarchy" button. If further hierarchies are going to be created over these hierarchical values check the "Define further hierarchical values" check box before clicking the "Create Hierarchy" button.

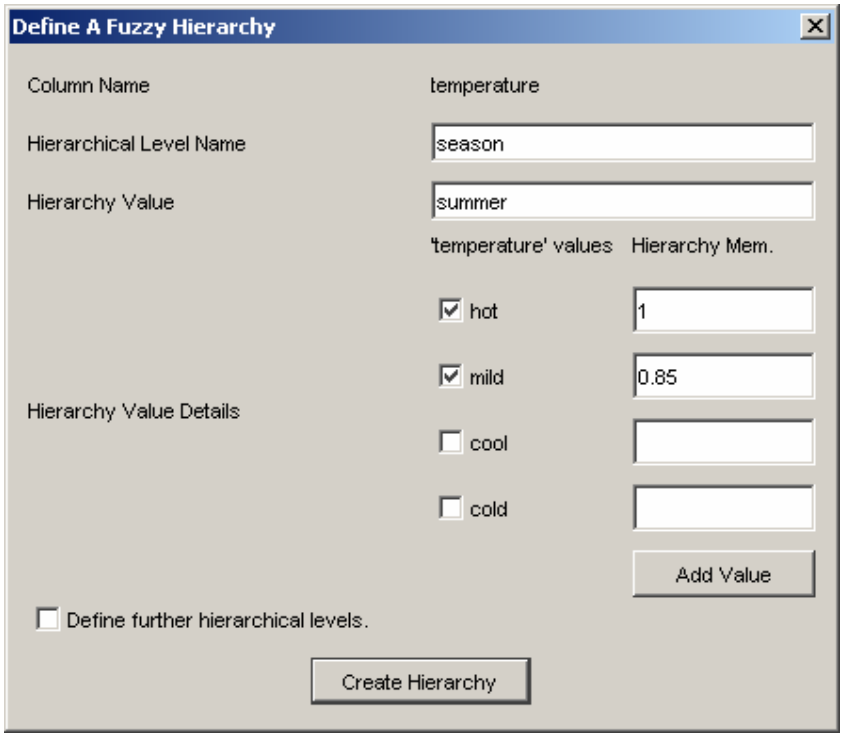


Figure 42: Hierarchy definition

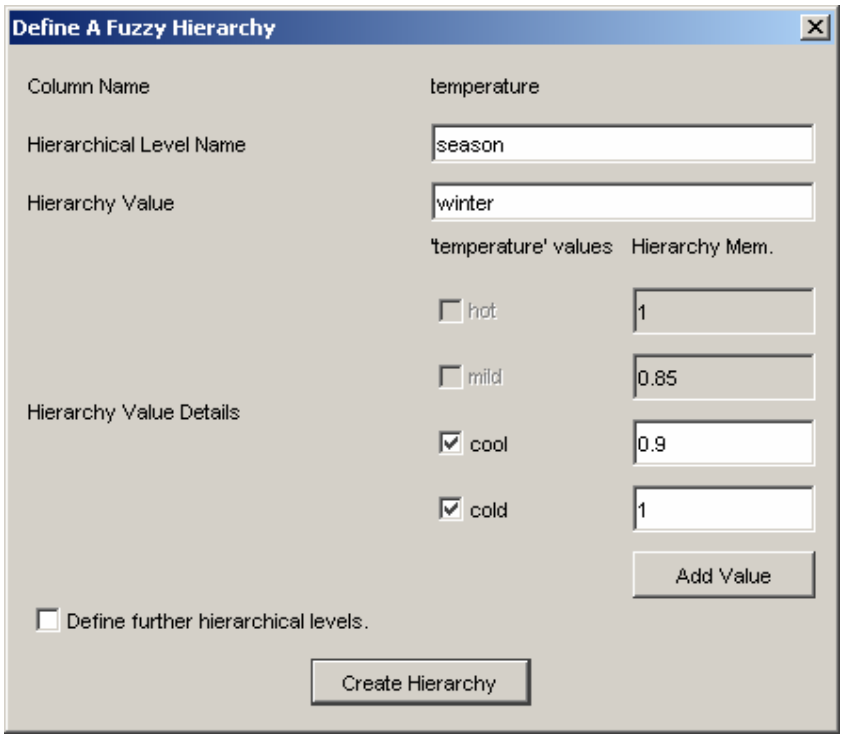


Figure 43: Addition of hierarchy value

After dimensions are determined and hierarchies are defined for them and measures are identified, click the “Construct Cube” button in the panel displayed in Figure 39 above. Next, the group-by expressions that will be executed for the fuzzy spatial data cube are identified and listed as in Figure 44.

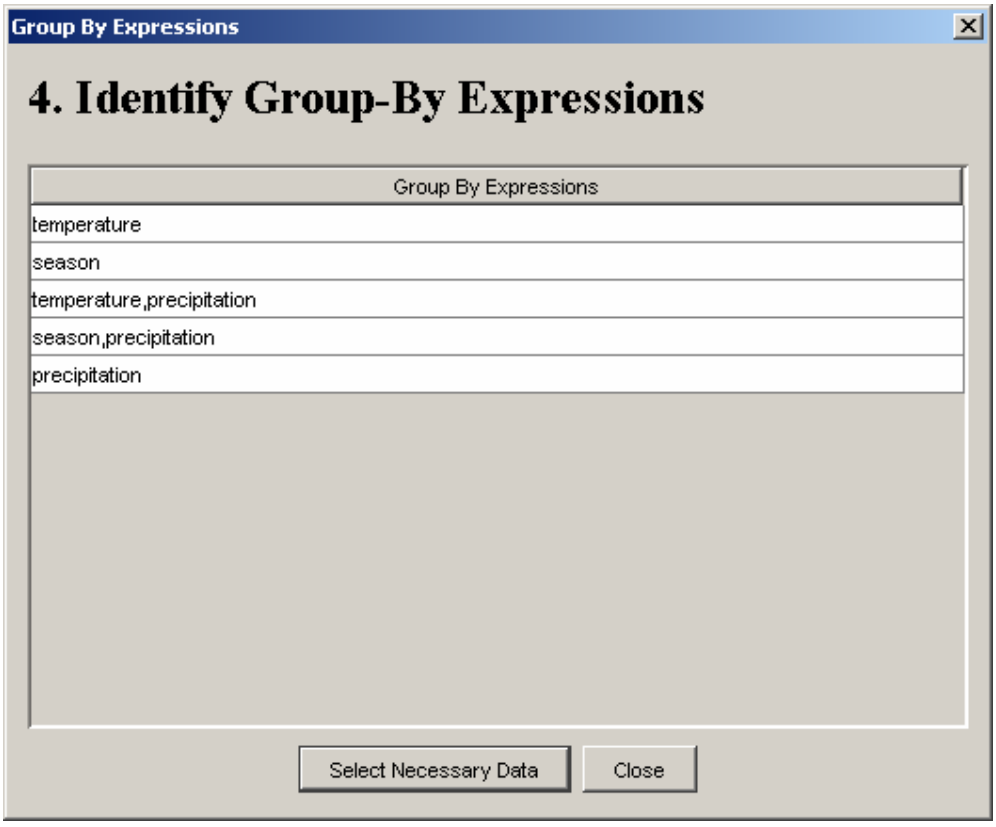


Figure 44: Group by expressions

Next, the necessary data, involving only the selected dimensions and measures, is listed as in Figure 45 and data is stored in database as in Figure 46.

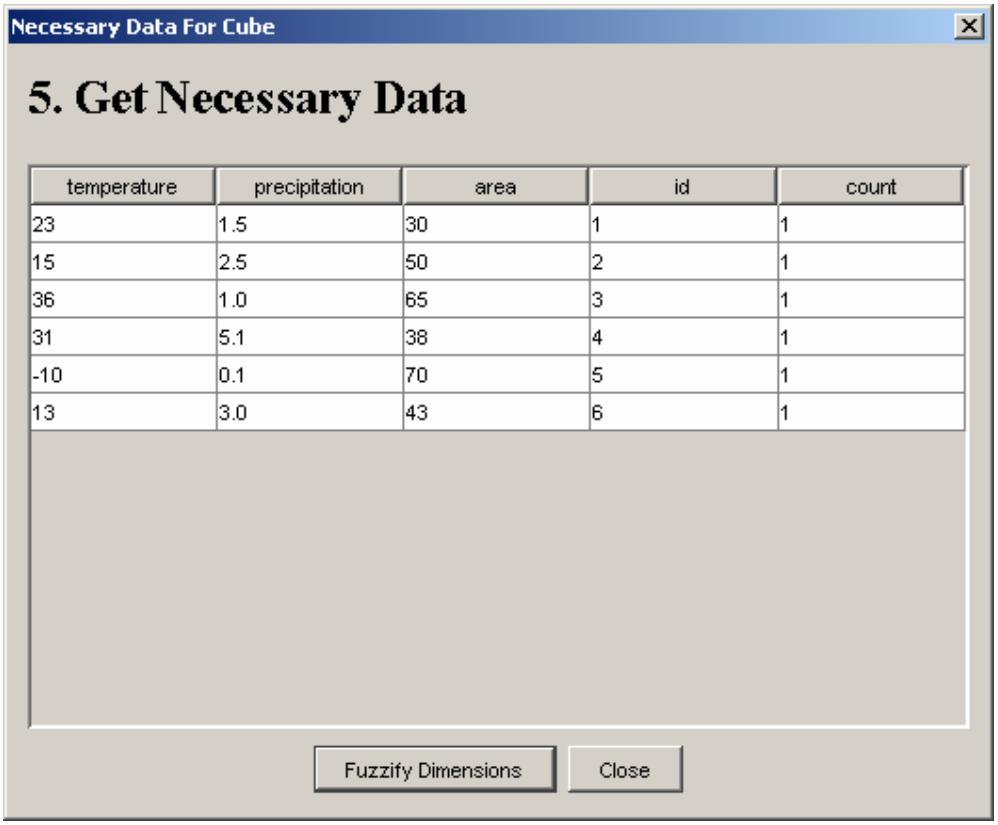


Figure 45: Necessary data for the cube

	id	temperature	precipitation	area	count	ids
▶	1	23	1.5	30	1	1
	2	15	2.5	50	1	2
	3	36	1.0	65	1	3
	4	31	5.1	38	1	4
	5	-10	0.1	70	1	5
	6	13	3.0	43	1	6
*						

Figure 46: Crisp data in "data" table

Next, columns that are fuzzy dimensions are fuzzified. It is displayed as in Figure 47 and stored in the database as in Figure 48.

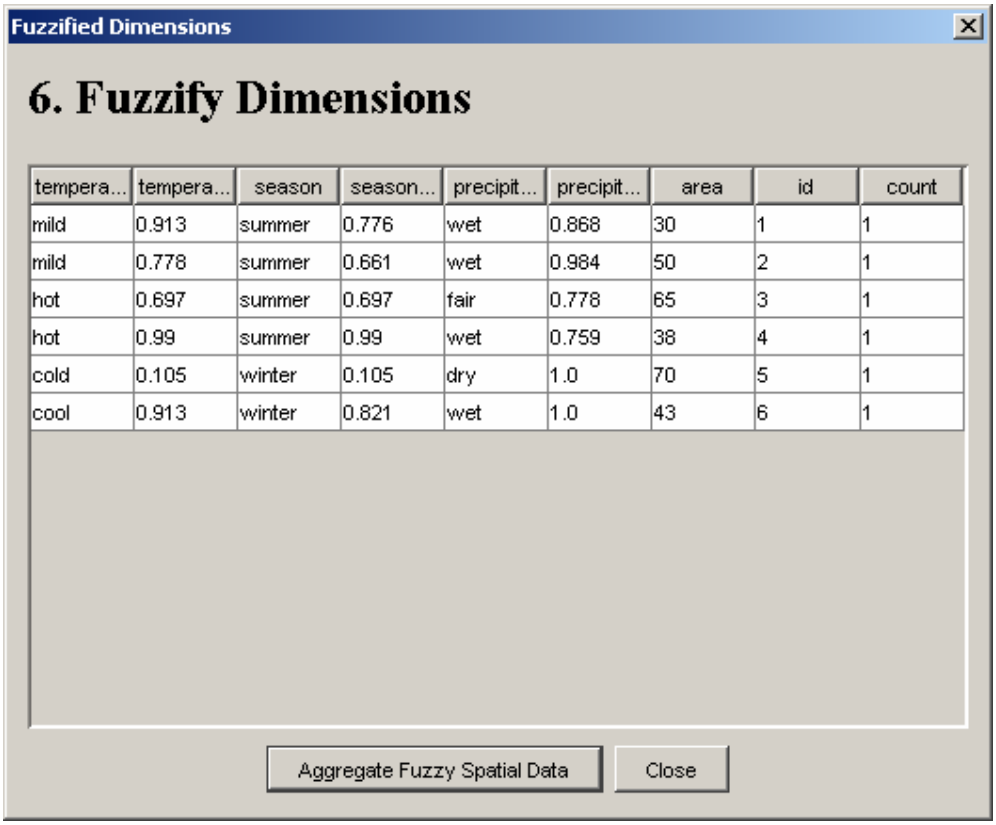
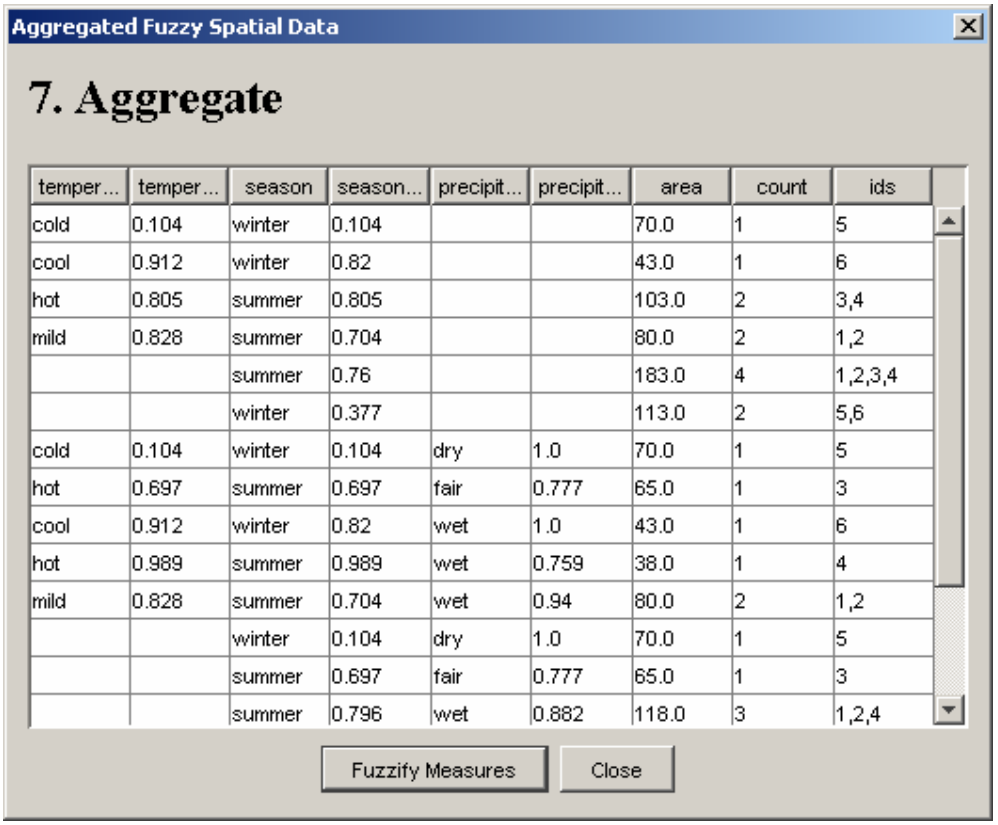


Figure 47: Display of generalized data

	id	temperature	temperatureMem	season	seasonMem	precipitation	precipitationMem	area	count	ids
▶	1	mild	0.913	summer	0.776	wet	0.868	30	1	1
	2	mild	0.778	summer	0.661	wet	0.984	50	1	2
	3	hot	0.697	summer	0.697	fair	0.778	65	1	3
	4	hot	0.99	summer	0.99	wet	0.759	38	1	4
	5	cold	0.105	winter	0.105	dry	1	70	1	5
	6	cool	0.913	winter	0.821	wet	1	43	1	6
*										

Figure 48: Generalized data in "extendeddata" table

Next, generalized data is aggregated. It is displayed as in Figure 49 and stored in the database as in Figure 50.



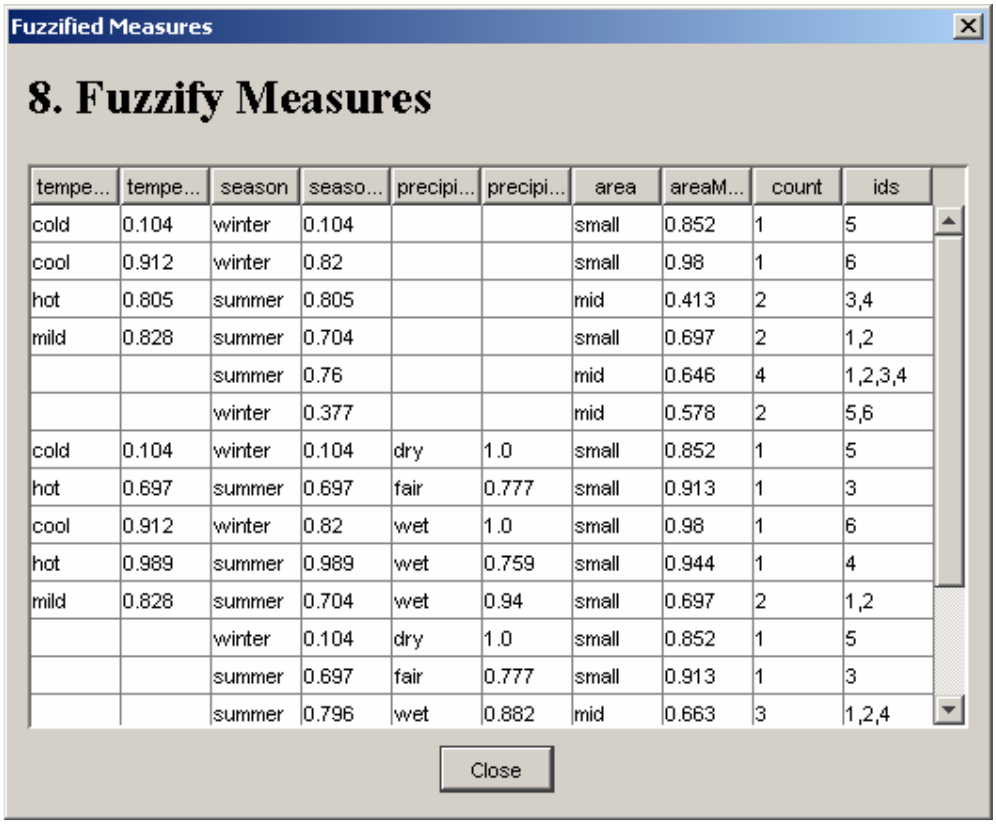
temper...	temper...	season	season...	precipit...	precipit...	area	count	ids
cold	0.104	winter	0.104			70.0	1	5
cool	0.912	winter	0.82			43.0	1	6
hot	0.805	summer	0.805			103.0	2	3,4
mild	0.828	summer	0.704			80.0	2	1,2
		summer	0.76			183.0	4	1,2,3,4
		winter	0.377			113.0	2	5,6
cold	0.104	winter	0.104	dry	1.0	70.0	1	5
hot	0.697	summer	0.697	fair	0.777	65.0	1	3
cool	0.912	winter	0.82	wet	1.0	43.0	1	6
hot	0.989	summer	0.989	wet	0.759	38.0	1	4
mild	0.828	summer	0.704	wet	0.94	80.0	2	1,2
		winter	0.104	dry	1.0	70.0	1	5
		summer	0.697	fair	0.777	65.0	1	3
		summer	0.796	wet	0.882	118.0	3	1,2,4

Figure 49: Display of aggregated data

	id	temperatur	temperatureMem	season	seasonMem	precipitation	precipitationMem	area	count	ids
▶	1	cold	0.104	winter	0.104	<NULL>	<NULL>	70.0	1	5
	2	cool	0.912	winter	0.82	<NULL>	<NULL>	43.0	1	6
	3	hot	0.805	summer	0.805	<NULL>	<NULL>	103.0	2	3,4
	4	mild	0.828	summer	0.704	<NULL>	<NULL>	80.0	2	1,2
	5	<NULL>	<NULL>	summer	0.76	<NULL>	<NULL>	183.0	4	1,2,3,4
	6	<NULL>	<NULL>	winter	0.377	<NULL>	<NULL>	113.0	2	5,6
	7	cold	0.104	winter	0.104	dry	1	70.0	1	5
	8	hot	0.697	summer	0.697	fair	0.777	65.0	1	3
	9	cool	0.912	winter	0.82	wet	1	43.0	1	6
	10	hot	0.989	summer	0.989	wet	0.759	38.0	1	4
	11	mild	0.828	summer	0.704	wet	0.94	80.0	2	1,2
	12	<NULL>	<NULL>	winter	0.104	dry	1	70.0	1	5
	13	<NULL>	<NULL>	summer	0.697	fair	0.777	65.0	1	3
	14	<NULL>	<NULL>	summer	0.796	wet	0.882	118.0	3	1,2,4
	15	<NULL>	<NULL>	winter	0.82	wet	1	43.0	1	6
	16	<NULL>	<NULL>	<NULL>	<NULL>	dry	1	70.0	1	5
	17	<NULL>	<NULL>	<NULL>	<NULL>	fair	0.777	65.0	1	3
	18	<NULL>	<NULL>	<NULL>	<NULL>	wet	0.913	161.0	4	1,2,4,6
*										

Figure 50: Aggregated data in "aggregate" table

After data is aggregated, measures that were defined as fuzzy are also generalized and displayed as in Figure 51. The aggregated data is modified in the database as in Figure 52.



tempe...	tempe...	season	seaso...	precipi...	precipi...	area	areaM...	count	ids
cold	0.104	winter	0.104			small	0.852	1	5
cool	0.912	winter	0.82			small	0.98	1	6
hot	0.805	summer	0.805			mid	0.413	2	3,4
mild	0.828	summer	0.704			small	0.697	2	1,2
		summer	0.76			mid	0.646	4	1,2,3,4
		winter	0.377			mid	0.578	2	5,6
cold	0.104	winter	0.104	dry	1.0	small	0.852	1	5
hot	0.697	summer	0.697	fair	0.777	small	0.913	1	3
cool	0.912	winter	0.82	wet	1.0	small	0.98	1	6
hot	0.989	summer	0.989	wet	0.759	small	0.944	1	4
mild	0.828	summer	0.704	wet	0.94	small	0.697	2	1,2
		winter	0.104	dry	1.0	small	0.852	1	5
		summer	0.697	fair	0.777	small	0.913	1	3
		summer	0.796	wet	0.882	mid	0.663	3	1,2,4

Figure 51: Display of generalized measures

id	temperatur	temperatureMem	season	seasonMem	precipitation	precipitationMem	area	count	ids	areaMem
1	cold	0.104	winter	0.104	<NULL>	<NULL>	small	1	5	0.852
2	cool	0.912	winter	0.82	<NULL>	<NULL>	small	1	6	0.98
3	hot	0.805	summer	0.805	<NULL>	<NULL>	mid	2	3,4	0.413
4	mild	0.828	summer	0.704	<NULL>	<NULL>	small	2	1,2	0.697
5	<NULL>	<NULL>	summer	0.76	<NULL>	<NULL>	mid	4	1,2,3,4	0.646
6	<NULL>	<NULL>	winter	0.377	<NULL>	<NULL>	mid	2	5,6	0.578
7	cold	0.104	winter	0.104	dry	1	small	1	5	0.852
8	hot	0.697	summer	0.697	fair	0.777	small	1	3	0.913
9	cool	0.912	winter	0.82	wet	1	small	1	6	0.98
10	hot	0.989	summer	0.989	wet	0.759	small	1	4	0.944
11	mild	0.828	summer	0.704	wet	0.94	small	2	1,2	0.697
12	<NULL>	<NULL>	winter	0.104	dry	1	small	1	5	0.852
13	<NULL>	<NULL>	summer	0.697	fair	0.777	small	1	3	0.913
14	<NULL>	<NULL>	summer	0.796	wet	0.882	mid	3	1,2,4	0.663
15	<NULL>	<NULL>	winter	0.82	wet	1	small	1	6	0.98
16	<NULL>	<NULL>	<NULL>	<NULL>	dry	1	small	1	5	0.852
17	<NULL>	<NULL>	<NULL>	<NULL>	fair	0.777	small	1	3	0.913
18	<NULL>	<NULL>	<NULL>	<NULL>	wet	0.913	mid	4	1,2,4,6	0.952

Figure 52: Aggregated data in "aggregate" table with generalized measures

4.2. View Constructed Fuzzy Spatial Data Cube

Click “View Constructed Cube” button in the start page. Next a panel with four options will be displayed as in Figure 53 below. It is possible to view the base data, aggregated data, dimensions and measures information of the previously constructed fuzzy spatial data cube as in Figures 54-57 below.

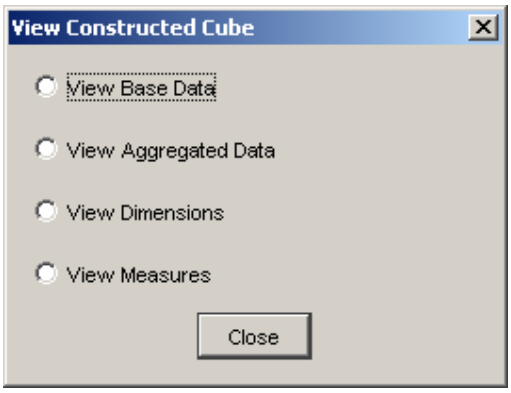


Figure 53: View options for the constructed cube

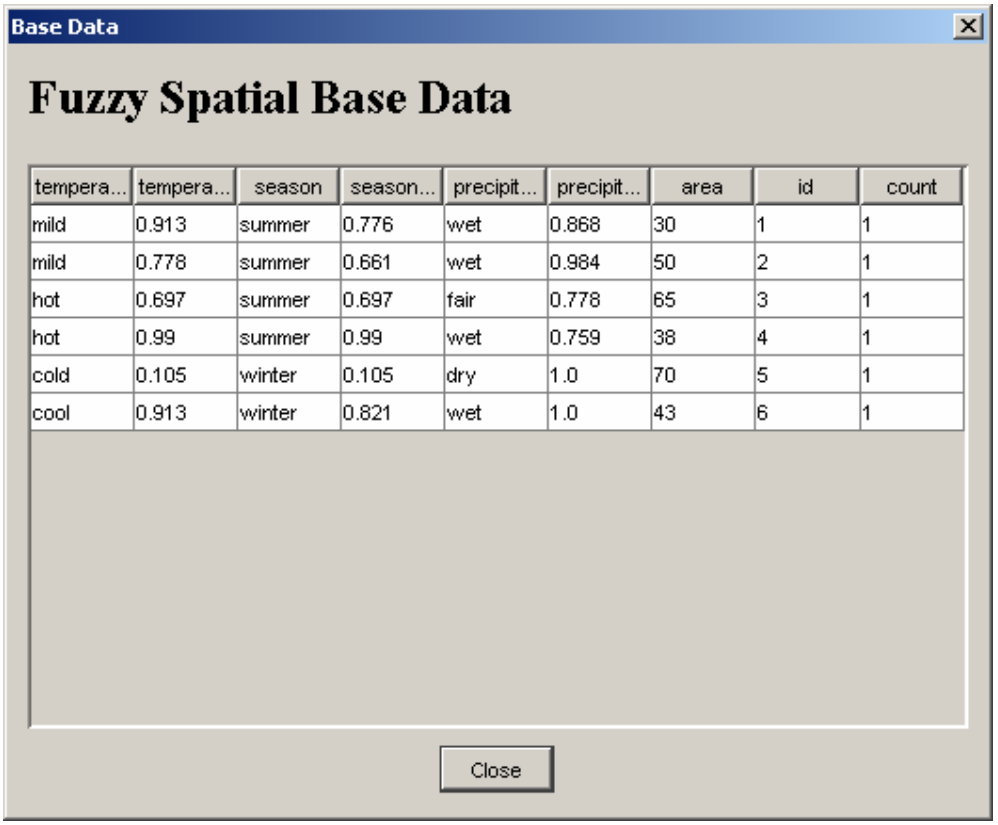


Figure 54: View of fuzzy spatial base data

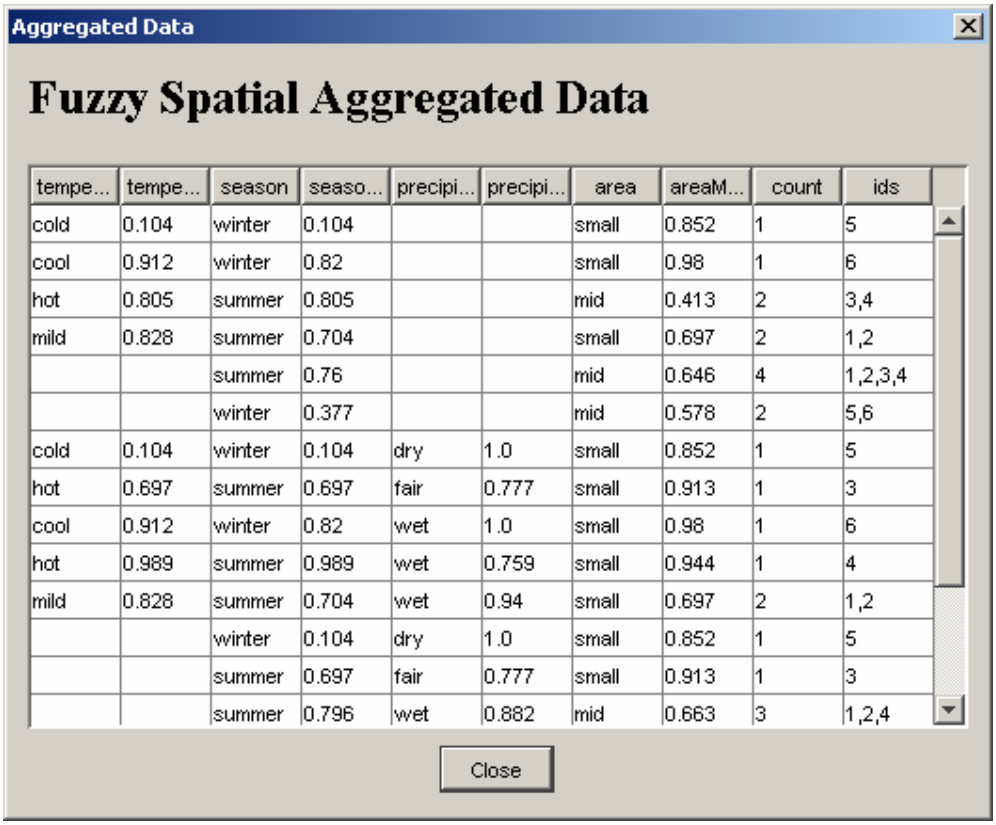


Figure 55: View of fuzzy spatial aggregated data

Figure 56: View of dimensions info

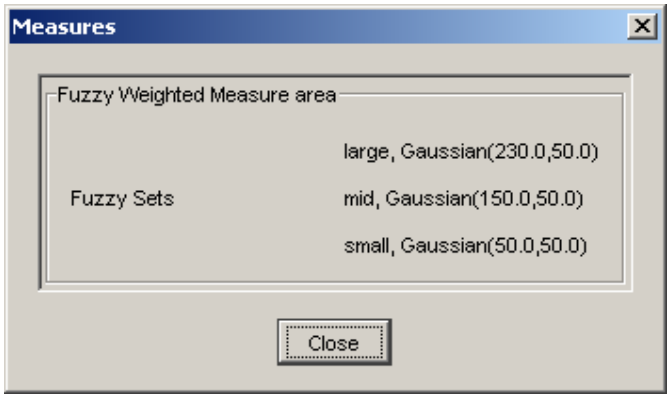


Figure 57: View of measures info

4.3. Finding Association Rules

Click “Find Association Rules” button in the start page. Next, a panel in which threshold values for significance and certainty of fuzzy association rules can be defined as displayed in Figure 58.

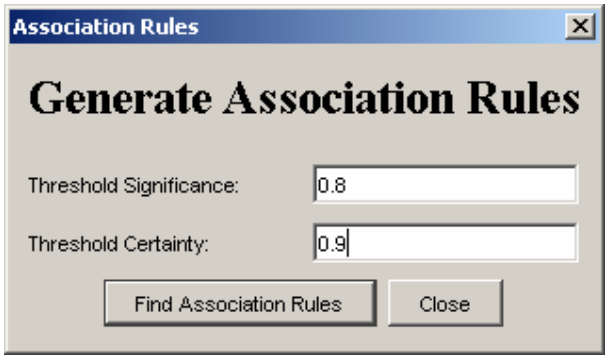


Figure 58: Threshold values for significance and certainty factors of fuzzy association rules

Finally, generated fuzzy association rules are displayed as in Figure 59 below. It is possible to further enter different values for threshold values of significance and certainty in order to find association rules with different significance and certainty.

For example, one of the generated fuzzy association rule is “temperature.cool (0.912), precipitation.wet (1.0) => area.small (0.98) [0.912, 0.98]”. That rule tells us that if a region has cool temperature and wet precipitation, we can conclude that “this region covers a small area with the certainty %98 for that conclusion and significance %91 for that region having both cool temperature and wet precipitation”. Individual reliabilities for temperature being cool are %91, for precipitation being wet is %100, and for area being small is %98.

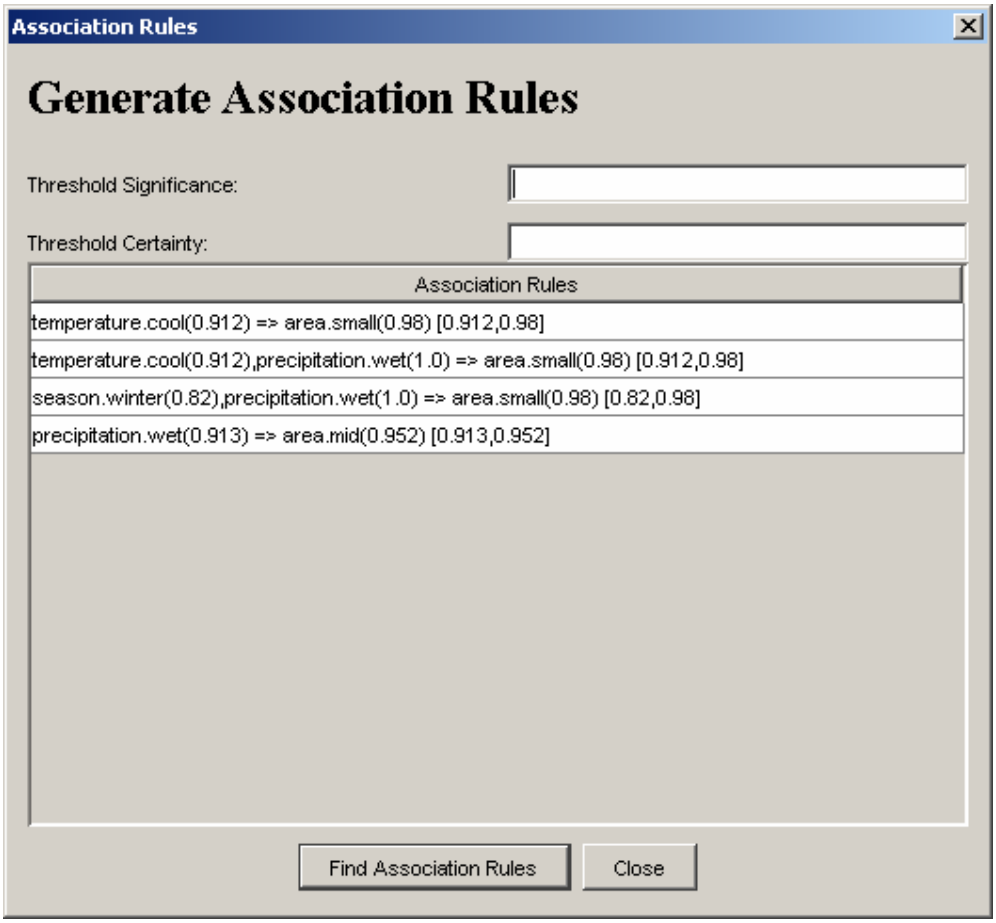


Figure 59: Display of fuzzy association rules

CHAPTER 5

DISCUSSION

Han has done some studies for spatial generalizations included in [6] and Stefanovic et. al. have extended these studies by constructing spatial data cubes included in [9]. But the generalization in their studies is imprecise as shown in Figure 60 below. For example, they have generalizations like "cold and dry regions (A04, T90 …) are 200,000m^2 wide". Here, the preciseness of hotness and dryness are missing both for individual regions on the left in Figure 60 and for the aggregated region on the right in Figure 60.

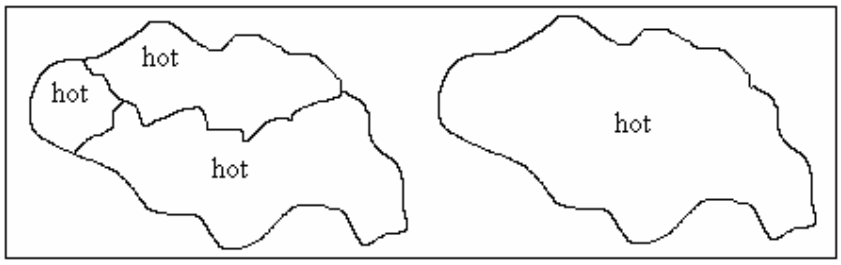


Figure 60: Spatial Generalization

Moreover, in Han's and Stefanovic's study, the measures are not generalized nor their preciseness is computed and indicated. In this thesis, that is also considered.

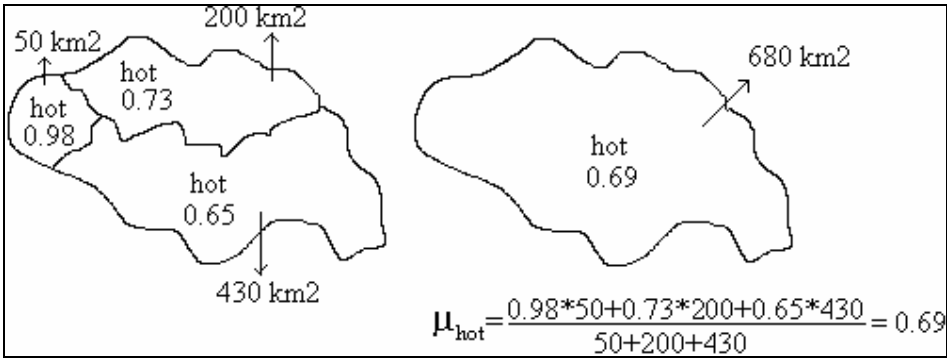


Figure 61: Fuzzy Spatial Generalization

This thesis mainly focuses on how preciseness of the generalized values can be considered as illustrated in Figure 61 above. Membership values for fuzzy dimensions and fuzzy measures of all individual regions and aggregated regions are calculated in the fuzzy spatial data cube. Before aggregation is done, spatial data are generalized and precision for generalization is calculated. Then, individual regions with common generalized values (i.e., hot) are aggregated together. Additional calculation is done for the precision of generalized aggregated spatial data.

Laurent et. al. [14, 15, 16] have done some studies for fuzzy OLAP and proposed a model for fuzzy multidimensional databases that are used for fuzzy summaries generation. In this thesis, fuzzy spatial data cube is constructed; hence the usage of spatial data differentiates the construction of such cubes from Laurent's study at some points. In their study, a value can belong gradually to more than one concept, i.e., Ankara can be generalized to take part in the east of Turkey with the precision 0.4 and it can also be generalized to take part in the west of Turkey with the precision 0.5. On the other hand, in the fuzzy spatial data cube, a value can be generalized to only one fuzzy label, to the label to which it belongs with the highest membership. Membership values for all defined fuzzy sets are computed and the maximum membership value and its corresponding fuzzy set are considered. That is important from the point of hierarchies. It is assumed that a value can be generalized to only one upper level since this makes fuzzy spatial generalization rules more sensible. For example, all regions could be generalized to hot regions, cold regions and mild regions. Some regions would contribute with low precision and some would contribute with high precision. Since that is not the desired result for spatial data, attributes of regions are generalized to the most appropriate label, i.e., to the label to which they fit with the highest membership value. For example, in Figure 61, it is supposed that the three sub-regions on the left fit best to the "hot" fuzzy label and they are aggregated to the larger region on the right.

In Laurent's study, each slice corresponds to the cube with a membership value, i.e., a value of one dimension has the same membership value for all the cells in the slice. In other words, in the fuzzy data cube, aggregated cells along one slice and individual cells along that slice have the same membership value for one dimension value. For example, the slice corresponding to sales done for oven along the production dimension can belong to the fuzzy data cube with degree 0.7. But in fuzzy spatial data cube each cell has its individual membership value for the corresponding dimension value since spatial objects might have common properties but each spatial object might have that property with a different degree than other spatial objects. Membership values of all cells along a dimension value are considered during aggregation in fuzzy spatial data cube. Fuzzy spatial data cube is semantically different than fuzzy data cube.

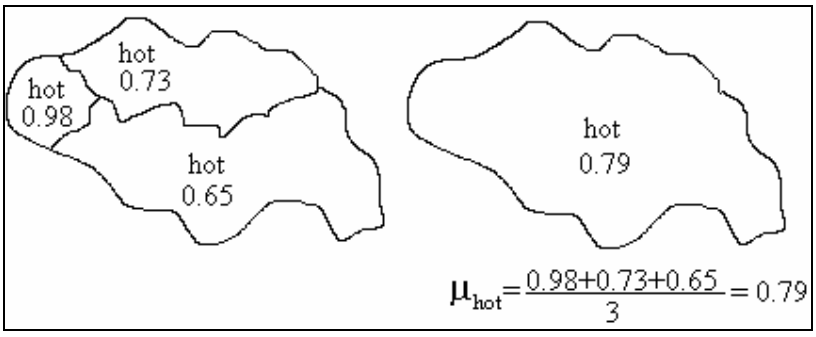


Figure 62: Aggregation in fuzzy data cubes

Furthermore, in Laurent's study, arithmetic average of membership values of cells is computed when fuzzy data are summerized. But regarding spatial data, it is not reliable to only take the arithmetic average since attributes of the spatial objects depend on space (or other issues as population). When aggregation on fuzzy spatial data is done it is necessary to take into account the weighted measure (the measure to which the other dimensions are dependent) in order to compute the precion of the generalized (i., summerized) data. In this thesis, during aggregation, the membership of generalization for the aggregated region (right side of Figure 61 above) is computed by weighting the membership values of individual regions by the value of

the measure indicated to be the weighted-measure. If aggregation in fuzzy spatial data cube had been done as it is done in fuzzy data cube, the membership value of the aggregated region would be computed as illustrated in Figure 62 below. The way of computing the membership value of the aggregated region in fuzzy spatial data cube is more realistic, since it will have a value closer to the value of the largest region (or most crowded). Taking arithmetic average may cause to some wrong deviation as it can be easily concluded by comparing Figure 61 and Figure 62.

Fuzzy spatial data cubes, make it possible to track deviations in precisions of characteristic properties of spatial data which was not an issue neither in spatial data cubes nor in fuzzy data cubes. For example, comparing generalization rules such as "Ankara was %80 hot and %78 dry in June, 2003." and "Ankara was %85 hot and %96 dry in June, 2004" enables seeing deviations in precisions of hotness (%5 increase) and dryness (%18 increase) values for Ankara between June, 2003 and June, 2004. By the help of fuzzy memberships computed for the spatial generalizations, it can be concluded that increases in temperature causes to greater increases in dryness in Ankara. Changes in spatial characteristics are easily identified in fuzzy spatial data cubes and this helps in decision-making about spatial data.

Another important issue is that, fuzzy association rules are generated from the fuzzy spatial data cube since the spatial data have precision values for the fuzzy dimensions and measures. Computations of significance and certainty factors are modified according to the aggregated data in the fuzzy spatial data cube. In previous studies, a fuzzy association rule is mined over many tuples in a relational database, but in fuzzy spatial data cube it can be mined over one aggregated tuple. Hence, fuzzy association rule mining is more feasible in fuzzy spatial data cube than it is in relational databases. Additionally, when the reliability to generalizations is more important than the frequency of the data in order not to miss the infrequent but significant rules, fuzzy association rule mining is more easily computed over spatial data than the spatial association rule mining, which has higher computational complexity.

CHAPTER 6

CONCLUSION

Transaction-oriented and analysis-oriented operations have very different response times when spatial data is very large. In order to reduce the high response time for analysis-oriented operations on spatial data, spatial data warehouses are required. Beside the fast response time, spatial data-warehouses are also needed since they keep summarized and aggregated spatial data flexibly and provide it to decision makers. They are also useful for extracting useful knowledge from detailed spatial data.

Spatial data warehouses are useful in answering analytical queries on spatial data. A spatial data warehouse can be represented by a star-schema that includes a set of dimensions of interest and a fact table that includes the aggregated results for the dimensions at the finest granularity.

Instead of keeping numerical data in rows and columns, generalizing that data to fuzzy labels and computing the memberships of these labels makes fuzzy OLAPs more meaningful and understandable due to fuzzy labels and preciseness due to the available membership values.

In this study, the concepts of fuzzy data cubes and spatial data cubes are combined to get benefit from both of them. A new method is proposed for the aggregation of the fuzzy dimensions and measures and their memberships regarding spatial data. While aggregating spatial objects, one of the attributes of the spatial objects is determined as the weighted measure, and aggregation of the fuzzy dimensions is done by computing the membership value of the aggregated spatial object by summing the product of each of the aggregated membership value with that weighted measure and dividing that sum to the summation of the weighted measures (i.e., areas) of all

aggregated spatial objects. Moreover, in this study, aggregation on fuzzy hierarchies is handled, which helps in obtaining higher levels of generalizations.

Data cubes are built for special purposes. That makes the choice of dimensions and measures very important. That choice is also very important for fuzzy spatial data cubes. Especially the choice of the weighted measure is very crucial since it will be considered in the computation of the membership values of the aggregated dimensions.

The constructed fuzzy spatial data cube can be used for generating fuzzy association rules. The way the aggregation is handled for fuzzy spatial cube in this study enhances the generation of fuzzy association rules from transactional databases. Computation of the significance and certainty factors is done more easily and more realisticly since the memberships of the dimensions for the aggregated spatial objects are computed regarding the weighted measure.

To sum up, this study contributes to the computational area by introducing fuzzy logic to spatial data cubes and proposing a new method for the aggregation of the fuzzy spatial attributes. The obtained generalization rules help in commenting the core spatial data and concluding decisions about it. Moreover, it is shown that fuzzy spatial data cubes can enhance the generation of fuzzy association rules from spatial data which increases the quantity and quality of the knowledge about the spatial data.

Future studies can be done about how fuzzy logic can be used for enhancing prediction of unknown and missing spatial data in spatial data cubes, or for comparing different classes of data and perform relevance analysis to find attributes that best distinguishes different classes, or classify spatial data and construct a model and use it to classify new data, or for clustering spatial data to find distribution patterns.

REFERENCES

[1] S. Chaudhuri, U. Dayal, "An Overview of Data Warehousing and OLAP Technology", ACM SIGMOD Record, vol 26, pp65-74, 1997

[2] *Readings in Database Systems*, Stonebraker M., Hellerstein J. M, Morgan Kaufmann Publishers, 1998.

[3] *Principles of Distributed Database Systems*, M. T. Özsu, P. Valduriez, Prentice Hall, 1999.

[4] *Data Mining: Concepts and Techniques*, J. Han, M. Kamber, Academic Press, 2001.

[5] *Microsoft SQL Server 2000 Analysis Services Step by Step*, R. Jacobson, Microsoft Press, 2000.

[6] W. Lui, J. Han, "Discovery of General Knowledge in Large Spatial Databases", Proc. of 1993, Far East Workshop on GIS, pp275-289, 1993.

[7] *Spatial Databases with Application to GIS*, P. Rigaux, M. O. Scholl, A. Voisard, Morgan Kaufmann Publishers Inc., 2002.

[8] O. R. Zaiane, Multimedia and Spatial Data Mining, Principles of Knowledge Discovery in Data, University of Alberta, 2002.

[9] N. Stefanovic, J. Han, K. Koperski, "Object-Based Selective Materialization for Efficient Implementation of Spatial Data Cubes", IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No. 6, 2000.

[10] V. Harinarayan, A. Rajaraman, J. D. Ullman, “Implementing Data Cubes Efficiently”, Proc. 1996, ACM-SIGMOD, Int’l Conf. Management of Data, pp205-216, 1996.

[11] J. Gray, A. Bosworth, A. Layman, H. Pirahesh, “Data Cube: A Relational Operator Generalizing Group-by, Cross-tab, and Roll-up.”, Proc. of the 12th Int. Conf. on Data Engineering, pp152-159, 1996.

[12] S. Agarwal, R. Agrawal, P. M. Deshpande, et. al., “On the Computation of Multidimensional Aggregates”, Proc. of the 22nd VLDB Conference, 1996.

[13] Y. Zhao, P. Deshpande, J. F. Naughton, “An Array-Based Algorithm for Simultaneous Multidimensional Aggregates”, Proceedings ACM SIGMOD International Conference on Management of Data, pp.159-170, 1997.

[14] A. Laurent, B. Bouchon-Meunier, A. Doucet, et. al., “Fuzzy Data Mining from Multidimensional Databases”, Int. Symp. On Computational Intelligence, Studies in Fuzziness and Soft Computing, 54:278-283, 2000.

[15] A. Laurent, “Generating Fuzzy Summaries from Multidimensional Databases”, Fourth International Symposium On Intelligent Data Analysis, pp24-33, 2001.

[16] A. Laurent, B. Bouchon-Meunier, A. Doucet, “Towards Fuzzy-OLAP Mining”, in Proc. Work. PKDD "Database Support for KDD", pp. 51-62, 2001.

[17] Han J., “OLAP Mining: An Integration of OLAP with Data Mining”, Proc. of the 7th IFIP2.6 Working Conference on Database Semantics (DS-7), pp1-9, 1997.

[18] Kelkar B. “Exploiting Symbiosis between Data Mining and OLAP for Business Insights”, DM Direct Newsletter, December 2001.

[19] C. White, “Intelligent Business Strategies: OLAP in the Database”, DM Review Magazine, June 2003.

[20] Pilot Software Acquisition Corp. (2002), "An Introduction to OLAP Multidimensional Terminology and Technology", http://www.pilotsoftware.com/pdf/olapwp.pdf, May 21, 2005.

[21] X. Zhou, S. Prasher, M. Kitsuregawa, “Database Support for Spatial Generalization for WWW and Mobile Applications”, Thirdth International Conference on Web Information Systems Engineering, Singapore, 2002.

[22] S. Prasher, X. Zhou, “Multiresolution Amalgamation: Dynamic Spatial Data Cube Generation”, In Proceedings of Fifteenth Australian Database Conference (ADC 2004), pp103-111, 2004.

[23] N. Stefanovic, “Design and Implementation of On-Line Analytical Processing (OLAP) of Spatial Data”, master's thesis, Simon Fraser University, Canada, Sept. 1997.

[24] J. Han, K. Koperski, N. Stefanovic, “GeoMiner: A System Prototype for Spatial Data Mining”, Proceedings of 1997 ACM-SIGMOD, International Conference on Management of Data, pp.553-556, 1997.

[25] D. Papadias, P. Kalnis, J. Zhang, Y. Tao, “Efficient OLAP Operations in Spatial Data Warehouses”, Seventh International Symposium on Spatial and Temporal Databases, 2001.

[26] X. Zhou, D. Truffet, J. Han, “Efficient Polygon Amalgamation Methods for Spatial OLAP and Spatial Data Mining”, Proceedings of the Sixth International Symposium on Large Spatial Databases, 1999.

[27] Y. Bedard, T. Merret, J. Han, “Fundamentals of spatial data warehousing for geographic knowledge discovery”, Geographic Data Mining and Knowledge Discovery, CRC Press, 2001.

[28] F. Petry, “Introduction to Fuzzy Databases", Kluwer Publisher, 1996.

[29] C. M. Kuok, A. Fu, M. H. Wong, “Mining Fuzzy Association Rules in Databases”, ACM Sigmod Record, Volume 27, pp.41-46, 1998.