UNIVERSITY OF PIRAEUS

DEPARTMENT OF INFORMATICS

Pattern representation and management techniques – The PBMS concept

PhD Thesis

EVANGELOS E. KOTSIFAKOS

MSc, Information Systems, AUEB (2003)

BSc, Informatics, AUEB (2001)

Athens, December 2009

University of Piraeus

Advisory Committee:

Supervisor:

Yannis Theodoridis
Assoc.Professor, U.Piraeus

Members:

M. Vazirgiannis
Assoc. Professor, Athens University of Economics and Business

M.Virvou
Professor, U. Piraeus

Thesis
Submitted for the degree of
Doctor of Philosophy
at the Department of Informatics,
University of Piraeus

EVANGELOS E. KOTSIFAKOS

"Pattern representation and management techniques – The PBMS concept "

Examination Committee:

Dimitrios Despotis
Professor, U. Piraeus

Dimitrios Apostolou
Lecturer, U. Piraeus

Charalampos Konstantopoulos
Lecturer, U. Piraeus

Aggelos Pikrakis
Lecturer, U. Piraeus

.....

EVANGELOS E. KOTSIFAKOS

Approved by the Examination Committee, ...

.....	.....	.....
Y. Theodoridis	M. Vazirgiannis	M. Virvou
Assoc. Proferssor, U. Piraeus	Assoc. Proferssor, AUEB	Professor, U. Piraeus
Supervisor	Member of Advisory Committee	Member of Advisory Committee

.....	.....	.....
D. Despotis	D. Apostolou	C. Konstantopoulos
Professor, U. PIraeus	Lecturer, U. PIraeus	Lecturer, U. PIraeus
Member of Examination Committee	Member of Examination Committee	Member of Examination Committee

.....

A. Pikrakis

Lecturer, U. PIraeus

Member of Examination Committee

To Kostis

Preface

Due to the large amount of patterns that are extracted from databases with data mining techniques, their complexity and heterogeneity, the need for Pattern Management in a unified way is emerging. A Pattern Base Management System (PBMS) supporting operations over patterns, like storage, retrieval and comparison has a wide range of applications in every scientific domain.

In this thesis we deal with the definition of a pattern representation model for a PBMS and with the very important problem of comparing crisp and fuzzy clustering patterns. We propose new clustering similarity measures and we define a novel algorithm for intuitionistic fuzzy clustering. The new measures are integrated into the PANDA comparison framework and we present real-world applications and experiments. We present a prototype PBMS, PatternMiner, an integrated and expandable environment for pattern management. Furthermore, we deal with the problem of pattern evaluation and we propose the use of ontologies to provide experts valuable semantics about the extracted patterns on a specific knowledge domain.

Evangelos E. Kotsifakos

Acknowledgment

I would like to address my special thanks to my advisor, Assnt. Prof. Yannis Theodoridis, for his guidance, support and inspiration during all these years of research and personal development. His advices enabled me to surpass all the research problems I dealt with and his knowledge and experience always provided me new ideas. Furthermore, he inspired me with his effective teaching methods that always supported the good values in personal and academic level.

Special thanks to my lab mates and research collaborators, Eirini, Nikos, Gerasimos, Nikos, Despoina, Yannis and Dimitris for their contribution in a variety of research fields. Their knowledge and scientific maturity gave me the opportunity to develop a better scientific thinking and writing.

I would like to thank Assnt. Prof. Michalis Vazirgiannis and Prof. M. Virvou for their productive comments on my work. I would also like to thank Profs. Evangelo and Mary Kontizas for their collaboration as well as Antonis, Vivi, Yannis, Dimitra and Konstantinos for their support in various technical issues.

Special thanks to my brother Alex for his valuable help, to my parents who provided me with every necessity, enabling me to reach to that level of education, to Kelly who supported and inspired me all these years, and to Kostis, who inspired my research through our endless conversations in the early years of my MSc and PhD studies and whose memory encouraged me to complete this thesis.

This research work has been supported by the project MetaOn, founded by the Operational Programme “Information Society” of the Greek Ministry of Development, General Secretariat for Research and Technology, co-funded by the European Union. Research also supported by the General Secretariat for Research and Technology of the Greek Ministry of Development under a PENED’2003 grant.

The collection of images used in the experiments of chapter 3.4 is courtesy of Dr. T.M. Lehmann, Image Retrieval in Medical Application (IRMA) group, Dept. of Medical Informatics, RWTH Aachen, Germany, http://irma-project.org.

The collection of images used in the experiments of chapter 3.5 has been provided by Foundation of the Hellenic Worlds (FHW), http://www.fhw.gr.

The data used in chapter 6.3 have been collected from the Greek Institute of Geodynamics (http://www.gein.noa.gr).

The data used in the study described in chapter 5.2 have been provided from the Sloan Digital Sky Survey (http://www.sdss.org/).

Table of Contents

List of Figures

List of Tables

1 Introduction

Data mining comprises a step of the knowledge discovery process and it mainly deals with methodologies for extracting knowledge artifacts, i.e. patterns, from large data repositories (Figure 1-1). Association rules, clusters, decision trees, are some well known patterns coming from the data mining area. Patterns can also be found in other areas, such as Mathematics (e.g. patterns in sequences, in numbers, in graphs, in shapes etc.), Geometry, Signal Processing etc. (Vazirgiannis et al., 2003). An important issue raise here: the *manipulation and management of the patterns in a unified way*, either they have been evaluated or not.

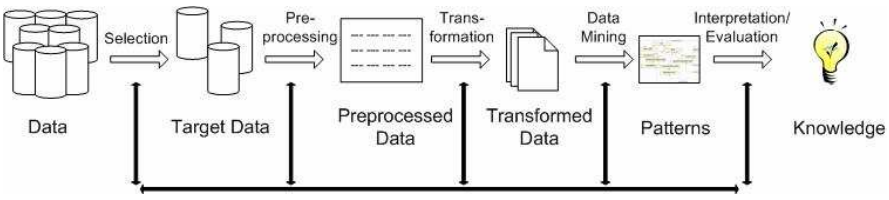


Figure 1-1 The KDD process

Currently, the majority of the available data mining tools support the visualization of patterns, and in the best case storage in relational tables. Combined with the characterization of patterns as complex, compact and rich in semantics representation of data (Rizzi et al., 2003), this issue raises the challenge for efficient pattern management. In analogy to a Database Management System (DBMS), a Pattern Base Management System (PBMS) manipulates patterns as a DBMS manipulates data. A PBMS can be used for representing, storing, querying, indexing and updating patterns. Moreover, advanced operations over patterns can be defined, such are pattern comparison and change monitor over time. Patterns are extracted from raw data and thus there exists a connection between the patterns and the data

that they have been extracted from and so, a change at raw data might suggest a change to the related patterns.

In general, basic operations over patterns (as the output of data mining algorithms) are:

- Storing the patterns extracted from the same dataset, using different parameters of a data mining task.
- Querying for previously extracted patterns using a lot of different search criteria, such as the initial dataset used, the date/time of the extraction process, the parameters used or even some properties/ values of the output.
- Comparing patterns extracted from the same dataset, extracted either with different parameters, either at different date/time.
- Monitoring pattern changes over time.

Pattern comparison is an advanced and important operation over patterns in a variety of real world applications as it provides a high level data comparison process. Comparing raw data is very time-consuming and requires a lot of processing power and I/O operations. Using patterns as compact representation of raw data, a pattern comparison process reflects the comparison of the underlying data but requires much less time and resources.

The definition of similarity operators/ measures over patterns results in a variety of interesting applications, that are, as discussed in (Ntoutsi, 2008):

1. Similarity queries
2. Monitoring and change detection
3. Dataset comparison
4. Evaluating data mining algorithms
5. Privacy aware data mining
6. Mining from distributed data sources
7. Discovering outlier or unexpected patterns

To efficient manage patterns and support the operations described above, a unified pattern representation model has to be defined, supporting advanced query and comparison operations.

In this thesis we adopt the PANDA (PAtterns for Next-generation DAtabase systems) (PANDA, 2005) project approach in data representation and management. The PANDA project deals with the problem of the unified pattern management and comparison. We define new functions for crisp and fuzzy cluster comparison and we present experiments on real-world applications such as content-based image retrieval and classification. We study the representation and management of clustering patterns focusing on the comparison of crisp and fuzzy clustering patterns as applications of frequent itemset and decision tree pattern management have been mostly discussed and presented in (Ntoutsi, 2008).

We present a PBMS prototype, PatternMiner, an XML-based integrated environment for pattern extraction, storage, retrieval and comparison. We also use the PBMS concept to facilitate the classification of a large amount of astronomical data, galaxy spectrums in particular.

Furthermore, we deal with the evaluation of the patterns extracted from data mining process in order to extend the PBMS concept to include one more step of the data mining process, that of the pattern evaluation (Figure 1-1). In this context, we study the use of ontologies that describe the domain knowledge, to evaluate patterns extracted from a large database.

While in this thesis we deal with the comparison of clusters, we use association rules patterns in our study for pattern evaluation with ontologies and decision trees in the classification of the astronomical data to point out the wide area of use of a PBMS.

1.1 Thesis organization

This thesis is organized as follows:

In chapter 2 we deal with the pattern representation and querying issues. We define a model for pattern representation, based on the pattern concepts of PANDA project and we provide definitions and examples for all the concepts that we will deal with in this thesis. Our mainly discussion on chapter 2 is to point out, through a qualitative evaluation, which is the best

representation model for a pattern-base, the relational, the object relational or the semi-structured (XML) model. In this study we use a custom XML schema to describe a pattern, based on the PANDA project approach while the scope is not to build a pattern-base to be used in our prototype, but to conclude whether an XML model is performing better for a pattern-base.

Chapters 3 and 4 deals with cluster comparison. In chapter 3 the case of crisp clustering is studied, a comparison measure is proposed based on the Expectation-Maximization clustering algorithm (Dempster et al., 1977) and two different content-based image retrieval applications, that use the PBMS concept and the proposed comparison measures, are presented. Chapter 4 deals with intuitionistic fuzzy clustering. The theory of intuitionistic fuzzy sets is presented and a variation of the Fuzzy C-Means algorithm (Bezdek, et al., 1984) is proposed, that uses a novel similarity measure for intuitionistic fuzzy data. The proposed scheme is evaluated through an image classification application.

In Chapter 5 we present an application of the PBMS concept for classification of astronomical data, a real-world case scenario of the GAIA project of the European Space Agency.

In Chapter 6 we present the PBMS prototype, PatternMiner and the study of the pattern evaluation process with the use of ontologies.

PatternMiner uses PMML (PMML, 2009) XML documents, enhanced to include all the necessary information to support all the PBMS operations such as pattern comparison and monitoring.

The pattern evaluation study is based on seismological data and the SUMO (2009) ontology of geology.

Chapter 7 summarizes the contributions of this work and discusses the open issues for future work.

2 Pattern Representation and Querying

This chapter highlights the basic notions of *patterns*, as an output of data mining process, as well as semantically rich representation of raw data in general. We describe the PANDA pattern definition model and other related work. By pointing out the need of a Pattern Base Management System (PBMS), we study three different pattern physical representation approaches in a pattern-base, the relational, the object-relational and the semi-structured (using XML). The scope of this study is to point out the best representation (and structure) model for a PBMS. Through a qualitative evaluation we conclude with the most proper representation approach for patterns, the semistructured, XML model. While we conduct our experiments using Oracle's DBMS, we will use the best representation model (semistructured) to build a PBMS from scratch (chapter 5).

2.1 Introduction

Nowadays, databases are huge, dynamic, with data from different application domains and a lot of different and complex patterns can be extracted from them. In order for someone to be able to exploit the information these patterns represent, an efficient and general-purpose Pattern Base Management System (PBMS) for handling (storing / processing / retrieving) patterns is becoming necessary for a lot scientific areas apart from data mining (Rizzi et al., 2003). Scientists of every field have their special needs for pattern creation and management and a PBMS approach would be the solution to the custom-per-problem application that they have to build.

To outline the problem of pattern management and the need for a PBMS, consider there is a large dataset and the K-Means clustering algorithm has to be applied. Using a set of user-defined parameters, a data mining tool

results in a number of *k* clusters. The output is presented in some form of text describing in general the center of every cluster and the distribution of data in every cluster. Depending the data mining tool in use, the format of the output is different. Even if there is the option to save the output, user cannot search for previous clusterings using as search criteria the parameters or the dataset that have been used. Moreover, user cannot combine or compare in any case different clusterings of the same dataset.

```
kMeans

======

Number of iterations: 4

Within cluster sum of squared errors: 80.4216834631833

Cluster centroids:

Cluster 0

       Mean/Mode:      68.0345    10.5172      6.5172      6.8621    10.4483
18.069      19.0345    27.8621   101          296.5862   254.2069   249.2759
389.0345   388.3793   337.3103   302.1034   282.8621   246.9655   223.9655
223.3448  194.4483  161.8621   97.8276   71.7931   57.4138   35.8276   12
2         0.3103    0.1379    0         0

       Std  Devs:      260.2523    28.8551    16.0326    15.3034    22.5556
38.0863     33.7104     49.6529   185.6823   459.7586   288.1144   206.0688
381.6582   346.0339   194.6038   168.1244   171.1642   157.144    169.7554
210.7751   204.1347   184.5747   127.573    104.3912    87.4894    57.857
25.9986    6.2393    1.0387    0.5158    0         0

Cluster 1

       Mean/Mode:   1958.4       161.15       86.1         77.7         91.8
104.95       88.15       98.85       84.8        78.35       79.7        99.25
125.85     114.4        87          68.4        65.55       67.95       79.2
84.95     87.85     81.05     55.85     42.1       34         35.7       29.7
21.3       5.75      0.2       0         0

       Std  Devs:    1507.5583    99.3109    54.0369    47.7715    59.9733
83.4856     75.985     109.9293   101.0714    94.0863   105.6305   128.4617
185.1507   169.1525   100.3615    74.6151    71.832     71.6075    89.7661
118.888    125.2615   122.393     94.5985    76.8313    63.8699    67.1488
56.7396   40.8232   11.7109    0.6156    0         0

=== Clustering stats for training data ===

Clustered Instances

0      29 ( 59%)

1      20 ( 41%)
```

Figure 2-1 The output of the K-means algorithm in WEKA data mining tool

Figure 2-1 and Figure 2-2 shows the output of the K-means and the J48 classification algorithm in WEKA data mining tool, respectively. Note the very specific format that the output is presented.

```
=== Classifier model (full training set) ===

J48 pruned tree

------------------
Bcolumn 39 = '(-inf-23.995]'
|  Rcolumn 10 = '(-inf-9.675]'
|   |   Rcolumn 1 = '(-inf-0.185]': 4 (12583.0)
|   |   Rcolumn 1 = '(0.185-inf)'
|   |   |   Bcolumn 40 = '(-inf-22.275]'
|   |   |   |   Bcolumn 38 = '(-inf-22.575]'
|   |   |   |   |   Bcolumn 44 = '(-inf-0.725]': 2 (2.0)
|   |   |   |   |   Bcolumn 44 = '(0.725-inf)': 4 (6.0)
|   |   |   |   Bcolumn 38 = '(22.575-inf)': 4 (439.0/1.0)


[...]


=== Summary ===

Correctly Classified Instances  26625    92.1759 %
Incorrectly Classified Instances  2260    7.8241 %
Kappa statistic       0.8712
Mean absolute error      0.0586
Root mean squared error     0.172
Relative absolute error     18.9389 %
Root relative squared error    43.7147 %
Total Number of Instances   28885

=== Confusion Matrix ===

    a          b         c          d     <-- classified as
 1231   1585      0          0   |   a = 1
  125   10377   51        16   |   b = 2
    0      194     1304        2   |   c = 3
    0       65       222   13713 |   d = 4
```

Figure 2-2 The output of the J48 classification algorithm in WEKA data mining tool

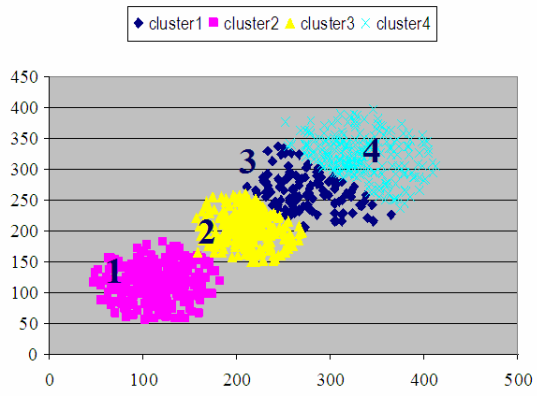
Moreover, data mining tasks that can be applied to a dataset, includes not only clustering but, in most cases, classification and association rule mining. Examples of the output of these tasks are presented in Figure 2-3. More specifically, examples of the three more common data mining pattern types are presented, association rules, clusters and decision trees. In the case of association rules, the structure of the rule is obvious (head and body), while the measures of the rule (confidence and support) are also clearly presented. In the case of clustering, four clusters (groups of data based on density/ proximity) are shown and, in the case of the decision tree,

a sample tree that classifies astronomical data based on the proper attribute is presented.

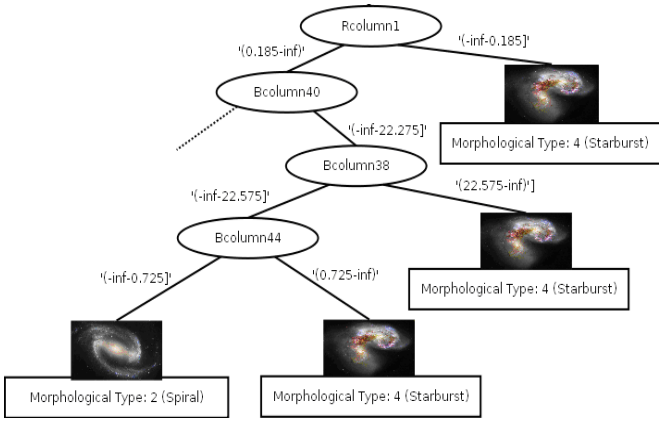
Each one of these has different output and representation, making more complex the problem of pattern management.

- `buys(x, “chips”) → buys(x, “beers”) [confidence: 0.5%, support: 60%]`
- `major(x, “CS”) ^ takes(x, “DB”) → grade(x, “A”) [confidence: 1%, support: 75%]`

Association rules



clustering



decision tree (classification)

Figure 2-3 Examples of the output of three most common data mining taks

2.2 Related Approaches

Current database systems do not support storage and management of the patterns extracted from data with data mining tools. The area of pattern representation and management is recent, and there are only few efforts. PMML (PMML, 2009), SQL/MM (SQL/MM, 2001), CWM (CWM, 2007), JDMAPI (JDMAPI, 2007), PQL (PQL, 2007) are standards and systems developed for storing data mining and statistical patterns. PMML (stands for Predictive Model Markup Language) proposed by the data mining Group (DMG) is the most popular approach. Using XML documents it provides a quick and easy way for applications to define predictive models and share these models between PMML compliant applications. PMML defines a variety of specific mining patterns (such as decision trees, association rules, neural networks etc.) but does not support custom pattern types. PMML version 3.2 provides more patterns and some functions for data preprocessing (PMML, 2009). A review of these approaches in relation to pattern management can

be found in (Catania & Maddalena, 2006). These approaches deal with common data mining patterns and do not provide pattern management functionalities.

The above approaches concentrate mostly on the definition of data mining and statistical models-patterns and the exchange of a set of patterns with specific characteristics between applications rather than on the creation of a general system for the representation and management of different pattern types. Pattern storage and querying techniques as well as pattern-to-data mapping are not among their capabilities.

During the last years two research projects, CINQ (CINQ, 2005) and PANDA (PANDA, 2005), defined the problem of pattern storage and management and proposed some solutions. CINQ aimed at studying and developing query techniques for inductive databases, i.e. databases that store the raw data along with the patterns produced by these data collections. On the other hand, PANDA aimed to the definition and design of a PBMS for the efficient representation and management of various types of patterns that arise from different application domains (not only from data mining). Patterns will reside and be managed (indexing, querying, retrieving) in the PBMS just like primitive data reside and are managed in the DBMS. Different types of patterns will be efficiently managed (generality) and new pattern types will be easily incorporated (extensibility) in the PBMS. A critical decision regarding to the PBMS is whether it should be build from scratch or as an additional layer on top of a DBMS. Building the PBMS on top of a DBMS restricts its capabilities, as the architecture of the DBMS has to be followed.

The area of pattern representation and management is recent, and there are only few efforts. PMML (PMML, 2009), SQL/MM (SQL/MM, 2001), CWM (CWM, 2007), JDMAPI (JDMAPI, 2007), PQL (PQL, 2007) are standards and systems developed for storing data mining and statistical patterns.

Recently a prototype PBMS that was based on the PANDA pattern model, called PSYCHO, has been presented (Catania, Maddalena & Mazza, 2005). PSYCHO manages different types of patterns in a unified way and it is developed with specific tools over the object-relational Oracle DBMS.

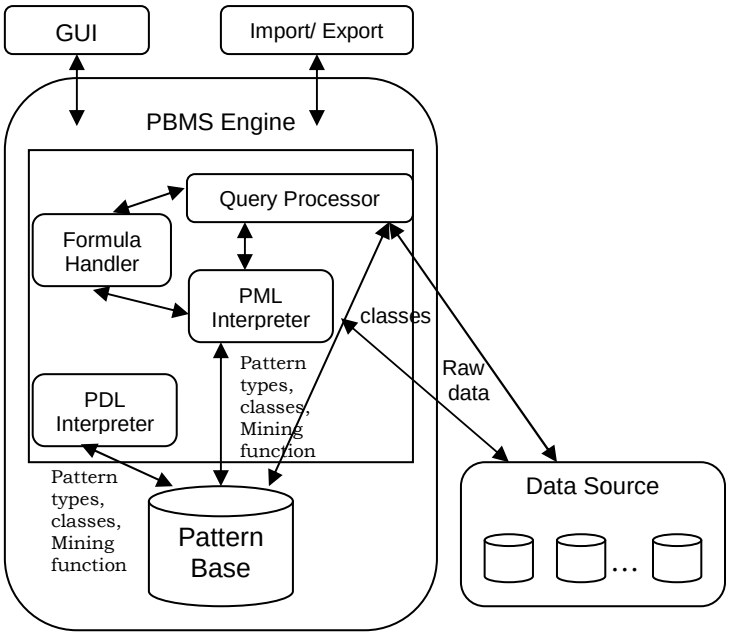


Figure 2-4 PSYCHO architecture

PSYCHO architecture is shown in Figure 2-4. The system is composed of three distinct layers. The physical layer contains both the *Pattern Base* that stores patterns and the *Data Source* that stores raw data from which patterns have been extracted. The middle layer, called *PBMS Engine*, supports functionalities for pattern manipulation and retrieval (pattern storage and querying). The external layer corresponds to a set of user interfaces (a shell and a GUI) from which the user can send requests to the engine and import/export data in other formats.

In this chapter, however, we follow the latter approach (i.e. working on top of a DBMS) in order to study which is the best representation model for a PBMS. Towards this aim, we examine three well known DBMS approaches: the relational, the object-relational and the semistructured (XML) model.

2.3 Representing Patterns in a Pattern Base Management System

From the aforementioned approaches, we adopt the PANDA project approach as it tries to incorporate all kinds of patterns. The pattern concept is the cornerstone of the PBMS. A *pattern* is a compact and rich in semantics representation of raw data. A *pattern-base* is a collection of persistently stored patterns. A *PBMS* is a system for handling patterns, defined over raw

data and organized in pattern-bases, in order to efficiently support pattern matching and to exploit pattern-related operations generating nontrivial information (Theodoridis et al, 2003). A PBMS treats patterns just like a DBMS treats raw data.

In order to efficiently manage patterns, a PBMS should fulfill some requirements (Theodoridis et al, 2003):

- *Implementation Complexity*: The Pattern-base should be easy to implement, without the use of very complex data types and management operations (insert, update, query etc).
- *Constraint implementation:* The PBMS should implement the constraints defined in the logical pattern model as well as validate patterns in line with these constraints.
- *Exploitation of patterns special characteristics:* The PBMS should take into account the special features of patterns so as to improve several operations, like indexing and query processing.
- *Query Effectiveness*: The PBMS should allow simple, yet efficient query construction. Users should be able to query for every pattern element using short and simple queries
- *Pattern Validation*: The PBMS should be able to validate patterns according to their pattern-type definition and reject patterns without the proper structure.
- *Extensibility*: The PBMS must be extensible to accommodate new kinds of patterns introduced by novel and challenging applications.
- *Generality*: The PBMS must be able to manage different types of patterns coming from different application domains.
- *Reusability:* PBMS must include constructs encouraging the reuse of what has already been defined.

The PANDA consortium has defined a logical model for the PBMS (Rizzi et al. 2003), which consists of three basic entities: pattern type, pattern and class defined as follows:

Definition 2-1. (Pattern Type): A pattern type is a quintuple $pt = (n, ss, ds, ms, f)$, where n is the pattern type name, ss is the structure schema that

describes the structure of the pattern type (in an association rule for example the structure consists of head and body), *ds* is the source schema that describes the dataset from which patterns of this pattern type are constructed, *ms* is the measure schema that defines the quality of the source data representation achieved by patterns of this pattern type and *f* is the formula that describes the relationship between the source space and the pattern space.

■

An example of the association rule pattern type is presented below:

```
n: AssociationRule
ss: TUPLE(head: SET(STRING), body: SET(STRING))
ds: BAG(transaction: SET(STRING))
ms: TUPLE(confidence: REAL, support: REAL)
f: head U body ⊆ transaction
```

Definition 2-2. (Pattern): A pattern *p*, is an instance of a pattern type *pt*, and has the corresponding values for each component.

An example of an association rule pattern, instance of the AssociationRule pattern type defined above, is the following:

```
pid: 413
s: (head={'Boots'}, body={'Socks', 'Hat'})
d: 'SELECT SETOF(article) AS transaction FROM sales GROUP BY transactionId'
m: (confidence=0.75, support=0.55)
e: {transaction: {'Boots', 'Socks', 'Hat'} ⊆ transaction}
```

Definition 2-3. (Class): A class *c*, over a pattern type *pt*, is defined as a triple *c* = (*cid*, *pt*, *pc*) where *cid* is the unique identifier of the class, *pt* is the pattern type and *pc* is a collection of patterns of type *pt*.

■

A class is defined for a given pattern type and contains only patterns of that type. Each pattern must belong to at least one class. The relationships between the three basic entities of a PBMS, i.e. pattern types, patterns and classes, are shown in the Figure 2-5 below:

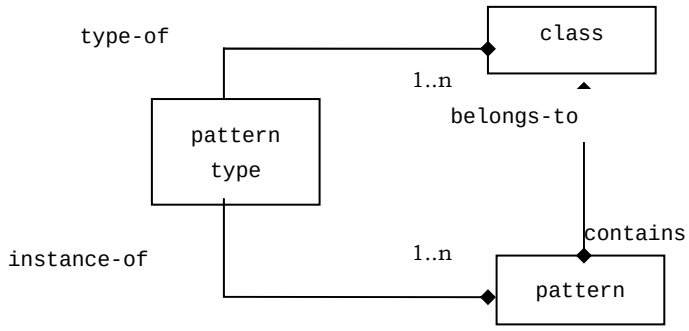


Figure 2-5 Relationships between pattern types, patterns and classes

Patterns in the PANDA framework can be either *simple* or *complex*. Simple patterns are extracted from raw data using the data mining process (clusters of raw data), while complex patterns are composed from simple ones (eg. a clustering on a set of clusters – clusters of clusters). In a complex pattern the structure component describes simple patterns and the measure component is either null either an aggregated measure depending the measures of the simple patterns. In the following section we present examples of simple and complex patterns.

Having defined the basic notions of a PBMS and the pattern structure and properties and after the examples of the pattern types, we will discuss the different options for the physical representation of patterns in the pattern-base.

2.3.1 Examples of common pattern types

In this section we present the three basic pattern types that we will deal with in this thesis, their PANDA model based representation and examples of their instances.

Frequent Itemsets – Association Rules

Association Rules represent associations/ relations between data items and they are based in Frequent Itemset Mining (FIM) (Agrawal et al, 1993). FIM find great application in retail stores where association rules based on frequent itemsets are extracted to assist store management, advertising and so on.

Itemsets can be expressed as patterns using the PANDA representation model in the following way:

Itemset =
(*SS*: {String},
MS: sup: (Real))

While a set of itemsets can be expressed as a complex pattern as shown below:

SetOfItemsets =
(*SS*: {*Itemset*},
MS: ⊥)

Association Rule patterns have two parts, head and body of the rule, which are sets of items, i.e. itemsets, while they are characterized by both measures support and confidence. Thus their PANDA model representation would be the following:

AssociationRule =
(*SS*: (head: Itemset, body: Itemset)
MS: (sup: (Real), conf: (Real)))

Note that we only define the structure and measure component of the patterns as the other three components are trivial and they are application dependant (the name, the data schema and the function component).

Clusters

Clusters are a very common pattern type, as clustering algorithms are performed very often in a large variety of applications. Commonly, clusters are either spherical or density based, depending the clustering algorithm.

A spherical (a Euclidean e.g.) cluster, such as the ones obtained from the k-means algorithm, can be modeled through a center and a radius, which form the structure schema of the cluster. For the measure schema, one could consider the cluster support, i.e., the fraction of objects that fall into the cluster, and the average intra-cluster distance:

EuclideanCluster =
(*SS* : (center: (Real), radius: (Real)),
MS: sup: (Real))

An example of such a pattern would be:

Cluster1 =
(*SS* : (center = 0.1, radius = 0.77),
MS: sup = 0.15)

Density-based clusters are produced by algorithms, like the Expectation-Maximization algorithm (Dempster et al., 1977), that uses distributions to group data points. A density-based pattern could be modeled using the mean and the standard deviation of the distribution for the pattern structure component and the support (the fraction of data points that fall into the cluster), as the measure component:

DensityBasedCluster =
(*SS* : (mean: (Real), stdDev: (Real)),
MS: sup: (Real))

An example-instance of such a pattern would be:

DensCluster =
(*SS* : (mean = 15.5, stdDev = 3.6),
MS: sup = 0.33)

Note that in most cases we have to deal with multi-dimensional datapoints and thus all the components are expressed as vectors.

Decision Trees

Decision trees are very popular data classification method and they provide an easy to understand classification model.

The leaf nodes of the tree are the classes that data points are classified, while the paths of the tree are constraints that "push" the data to the leaf nodes. Decision trees can be described using the PANDA representation by describing these constraints for all the attributes

Path =
(*SS* : $[(\text{ValueFrom: Real, ValueTo: Real})]_1^N$,
MS: sup: Real)

DecisionTree =
(*SS* : {Path},
MS: $\perp$)

An example of the decision tree pattern (from a three attribute dataset) would be the following:

aPath =
(*SS* : [(0, 8),(4, 6),(1, 2)],
MS: sup: 0.17)

aDecisionTree =
(*SS* : {Path},
MS: ⊥)

In this thesis we will deal with association rule patterns, density-based clusters as well as decision tree patterns in different applications.

2.4 Physical Representation in a Pattern-Base

For the representation and storage of the contents of a pattern-base, we examine three traditional DBMS approaches: the relational, the object-relational and the semistructured (XML) model using the entities presented in the previous section.

Next, we present each approach and give some representative queries that point out the advantages and disadvantages of each one. This comparison aims to examine the applicability of the logical model in current DBMS technology and is based on qualitative rather than quantitative criteria. The primary goal is to examine whether a PBMS can be built based on each of the three models presented, and which one is the more efficient on supporting the patterns special characteristics.

2.4.1 Relational Approach

Our main concern during the design and implementation of the pattern-base was to satisfy the three basic requirements of the logical model: generality, extensibility and pattern characteristics exploitation (Theodoridis et al, 2003). The relational schema is depicted in Figure 2-6:

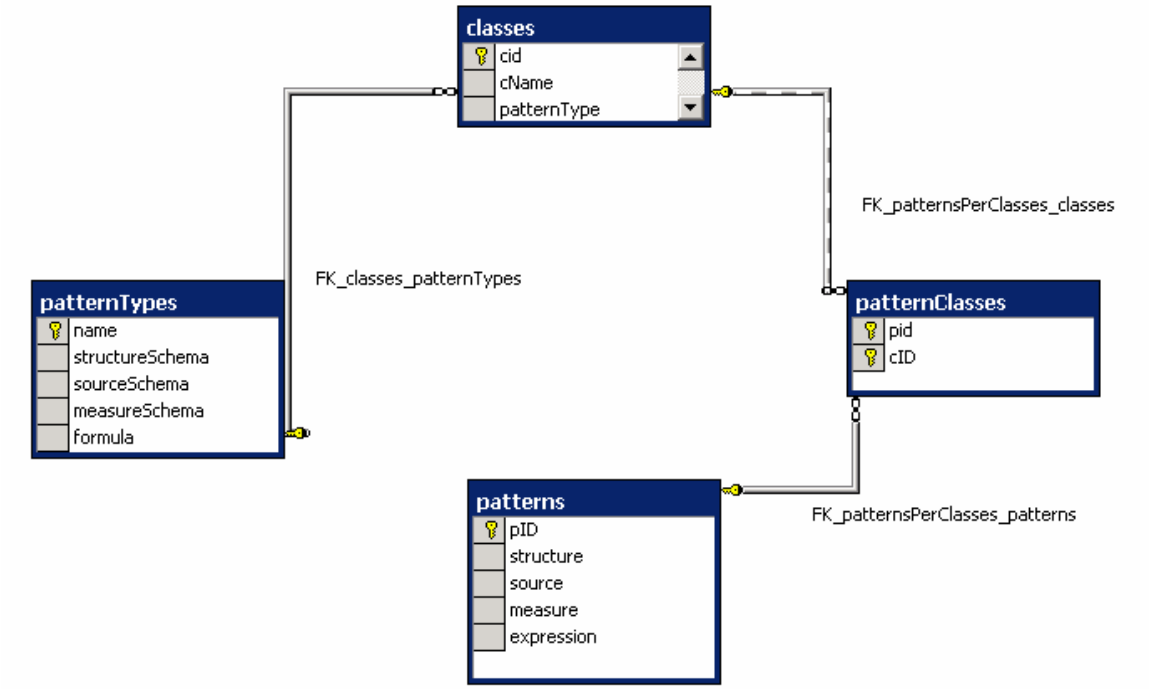


Figure 2-6 The relational schema of the pattern-base

Various pattern types are stored in the table *patternTypes*, patterns are stored in the table *patterns* and pattern classes are stored in table *classes*. The table *patternClasses* relates patterns with classes (a class contains one or more patterns of the same type and every pattern belongs to at least one class).

Below we present some representative queries. The queries will be first described in natural language and then in SQL-like syntax:

RQ1) *Find the structure of the association rules belonging to class Association_Rule_1.*

```
select patterns.structure from classes
inner join patternclasses on classes.cid = patternclasses.cid
inner join patterns on patternclasses.pid = patterns.pid
where (classes.cname='Association_Rule_1');
```

RQ2) *Return the 'head' and 'body' parts of the structure of patterns that they belong to class Association_Rule_1.*

```
Select Substr(structure,1,instr(structure,'body')-2) as head,
Substr(structure,instr(structure,'body')) as body from classes
inner join patternclasses on classes.cid = patternclasses.cid
```

```
inner join patterns on patternclasses.pid = patterns.pid
where (classesr.cname='Association_Rule_1');
```

RQ3) *Return the confidence measure from all the association rules.*

```
Select Substr(measure,1,instr(measure,'confidence')-2) as confidence, from
patterns;
```

The relational approach is characterized by simplicity and ease of implementation. However, it has a lot of disadvantages that arise from the fact that this approach does not take into account the underlying structure of pattern components (structure, measure, etc.) and treats them as simple texts/ strings. This fact makes querying a complex, time consuming and mostly ineffective process.

2.4.2 Object-Relational Approach

The object-relational model (Stonebraker, 1997; Stonebraker et al., 1999) manages to deal with the basic drawback of the relational model, by defining different objects and attributes for each pattern component and exploiting inheritance. In that way it is less complex and more efficient since querying is simpler.

The basic idea of the object-relational model (a part of it) is depicted in Figure 2-7. At the root of the object relational model stands the *Pattern* entity, which contains generic information about the pattern, such as the pattern identifier, the pattern formula and the pattern source. At the next level of the tree, the *Pattern* is specialized, according to the pattern type it belongs to, for example to association rules patterns, to clusters patterns etc. These entities differ according to their structure and measure components but they also have some attributes in common, those inherited by the *Pattern* entity. For example, object *Association Rule Pattern* contains every attribute from object *Pattern* and it also has the attribute *Structure* that consists of a *head* and a *body*. This object can be further specialized based on the measure component. As it appears in Figure 2-7, in the object *Association Rule Pattern 1* the *Measure* component consists of *confidence* and *support*, whereas in the object *Association Rule Pattern 2* the *Measure* component consists of *coverage*, *strength*, *lift* and *leverage*.

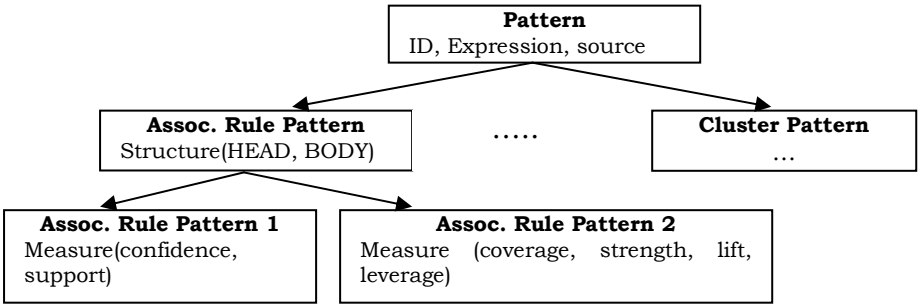


Figure 2-7 The basic idea of the object-relational approach

Below we present some representative queries for the object-relational model:

OQ1) *Find the structure of patterns association rule patterns.*

```
select p.id, treat(value(p) as hr.assrule_pattern).structureschema from
hr.tbl_patterns p;
```

OQ2) *Find the body of the structure of association rule patterns*

```
select p.id, value(e), value(f)  from hr.tbl_patterns p,
table(treat(value(p) as hr.assrule_pattern).structureschema.head) e,
table(treat(value(p) as hr.assrule_pattern).structureschema.body) f;
```

OQ3) *Find the confidence of the measure of association rule patterns.*

```
select p.id, treat(value(p) as
hr.assrule_pattern_1).measureschema.confidence as
confidence from hr.tbl_patterns p;
```

The object-relational approach overcomes some of the relational approach limitations due to the capability of modeling complex entities as objects. It also exploits the similarities among objects through inheritance. The object-relational model is more flexible and efficient from the relational model but, on the other hand, it requires exact definition of any new object and of its components.

2.4.3 Semi-structured (XML) approach

Unlike traditional databases, in an XML base the format of the data is not so rigid. This property is valuable in our case since patterns come from different application fields having thus different characteristics. For the XML

implementation, we have to create an XML schema for each pattern type. Patterns of a specific pattern type will be the XML documents (instances) of the XML schema of this type.

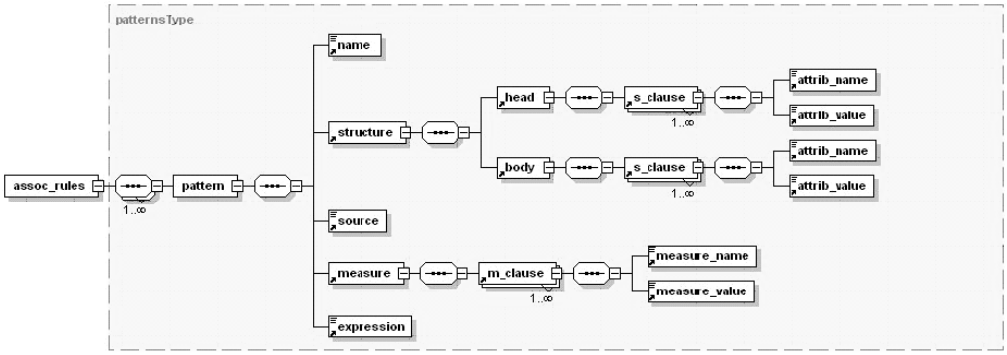


Figure 2-8 The association_rule.xsd

```
<assoc_rules ptype="association_rule">
        <pattern id="1"> <name>rule 1</name>
                <structure>
                        <head>
                                <s_clause>
                                        <attrib_name>buys</attrib_name>
                                        <attrib_value>scarf</attrib_value>
                                </s_clause>
                        </head>
                        <body>
                                <s_clause>
                                        <attrib_name>buys</attrib_name>
                                        <attrib_value>gloves</attrib_value>
                                </s_clause>
                        </body>
                </structure>
                <source>SELECT * FROM orders</source>
                <measure>
                        <m_clause>
                                <measure_name>support</measure_name>
                                <measure_value>0.35</measure_value>
                        </m_clause>
                        <m_clause>
                                <measure_name>confidence</measure_name>
                                <measure_value>0.75</measure_value>
                        </m_clause>
                </measure>
<expression>
{buys="hat",buys="cap",buys="gloves"}
</expression>
        </pattern>
```

Figure 2-9 association_rule.xml

For example, the association rules pattern type is described through the schema “association_rule.xsd” (Figure 2-8), whereas the XML document “pattern-association_rules.xml” (Figure 2-9) contains patterns of the association rule pattern type schema.

Below we present some representative queries for the XML model in ORACLE XML-SQL syntax:

XQ1) *Find the structure of the association rule patterns belonging to class “class1”.*

```
Select
extract(value(y),'//pattern[@id="'||extract(value(e),
'pid/text()')||'"]/structure') as structures from assoc_rules y, classes x,
TABLE(XMLsequence(extract(value(x),
'class[@name="class1"]//pids/pid'))) e where
existsNode(value(y),'//pattern[@id="'||extract(value(e),'pid/text()')||'"]/s
tructure') = 1
```

XQ2) *Return the ‘head’ and ‘body’ parts of the structure of patterns that they belong to class Association_Rule_1.*

```
select
extract(value(y),'//pattern[@id="'||extract(value(e),'pid/text()')||'"]//s_c
lause') as pattern_name from assoc_rules y, classes x,
TABLE(XMLsequence(extract(value(x),'class[@name="class1"]//pids/pid'))) e
where
existsNode(value(y),'//pattern[@id="'||extract(value(e),'pid/text()')||'"]//
s_clause') = 1;
```

XQ3) *Find the confidence of the measure of association rule patterns.*

```
select (extractvalue(value(val),'//text()')) as confidence from assoc_rules
a,
TABLE(xmlsequence(extract(value(a),'//m_clause[measure_name="confidence"]/me
asure_value'))) val
```

XQ4) *Find all the different measures(inside the measure component) of the association rules.*

```
select distinct extractValue(value(r), '//m_clause/measure_name/text()') as
measures from assoc_rules y, classes
x,TABLE(XMLsequence(extract(value(x),'//pids/pid'))) e,
TABLE(XMLsequence(extract(value(y),'//pattern[@id="'||
extract(value(e),'pid/text()')||'"]//m_clause'))) r;
```

With XML pattern-base, the definition of a new pattern type is easy (extensibility). Furthermore, it is possible to create a proper XML schema for a pattern type, general enough to include every variation of patterns of this type (generality). The XML schema affects also the effectiveness of querying. Queries like *XQ4* "find all the different measures of the association rules", can be easily implemented, unlike the relational and object-relational approaches.

2.4.4 A Qualitative Comparison

In this section we present the criteria for the comparison of the three alternative representations and the conclusions we reached.

 # 1. Pattern-base Implementation Complexity

All the three models we presented can be easily implemented. The simplest model is the relational, where both the pattern-base construction and insert operations can be performed in an easy and fast way. The object relational model is slightly more difficult since it requires the definition of different objects for each pattern type (and each of its variations). Insert operations are also more difficult as it should be different for each pattern type and its variations. Finally, the difficulty of the XML model is the fact that its success depends straightforward on the quality (generality) of the XML schema for each pattern type. However, after creating the proper schema insert operations can be easily performed. Furthermore, if this schema is general enough, variations of patterns belonging to a specific pattern type can be easily supported through this pattern type schema.

 # 2. Constraint Implementation

The basic constraints imposed by the logical model (Rizzi et al., 2003) are the following: (a) every pattern is an instance of one pattern type, (b) every pattern belongs to at least one class, (c) a pattern class should contain only patterns of the same pattern type.

These constraints can be easily implemented in the relational model through the foreign key constraints. In the object relational model these constraints are supported directly by the definition of the pattern type, for example it is impossible to assign a cluster into the association rule pattern type. In the

XML model, finally, the implementation of constraints are supported by the DBMS with mechanisms that associate XML documents.

 # 3. Pattern Characteristics Exploitation

According to the logical model (Rizzi et al., 2003), every pattern consists of five basic components: name, structure, source, measure and formula. However, different pattern types differentiate on some of these components, e.g. in structure or measure. If we exploit the special characteristics of each pattern type we can improve operations like indexing and querying. The relational model does not exploit the underlying structure of patterns as it considers every pattern component as a string, whereas, both object-relational and XML models take into account the special characteristics of pattern component according to the pattern type.

 # 4. Query Effectiveness

The pattern-base does not aim only at the storage of patterns but mainly at their easy management, so the effectiveness of querying is an important criterion. From the representative queries we gave above for each implementation, it is obvious that in the relational model query construction is a complex and time consuming process (it is all about string manipulation formulas). The rest two models exploit the underlying pattern structure, thus queries are expressed more easily.

 # 5. Extensibility

Extensibility is the ability to incorporate a new pattern type in the pattern-base; the easier this process is the more extensible the system is. The relational model is very extensible; a new pattern type is simply a new record in the table pattern types. The object-relational model requires the creation of new objects for every new pattern type and its components (the same stands also for the variations of a pattern type). That means that more than one association rule schema maybe required to incorporate the differences in the structure of each association rule. In the XML model a new schema is required for each new pattern type, but on the other hand, since this schema exists and is general enough, variations of patterns of this type can be easily incorporated without any modification.

 # 6. Pattern validation

The validity check during insert/ update operations in the pattern-base is critical. With the term validity we mean that each pattern in the pattern-base should follow its pattern type definition. The above criteria is violated in the relational model, whereas it stands for both XML and object relational models because of the XML schemas and the objects' definition respectively.

 # 7. Reusability

The reusability criterion is satisfied by object relational and the XML pattern-bases, since the relational approach does not support inheritance or the definition of semi-structured documents as the other two approaches do.

 # 8. Generality

All three approaches do satisfy this criterion, as in every one it is possible to define every kind of pattern-type, although it is more complex in the relational approach.

The conclusions of the evaluation are summarized in Table 2-1below:

Table 2-1 Comparison table of the three possible representation models for a pattern-base

	Relational pattern-base	**Object-relational pattern-base**	**XML pattern-base**
Implementation Complexity	High	Medium	High
Constraint Implementation	Yes	Yes	Yes
Pattern characteristics exploitation	No	Yes	Yes
Query effectiveness	Low	Medium	Medium
Pattern validation	No	Yes	Yes
Extensibility	High	Medium	High
Reusability	No	Yes	Yes
Generality	Yes	Yes	Yes

From the above table it is clear that the XML pattern-base implementation is the best among the three choices.

At this point, it should be clarified that the pattern validation issue mentioned here, refers to the XML document validation, in order to ensure that the pattern, represented in the XML document, has the proper structure, as it is defined by the PANDA framework.

2.5 Synopsis

In this chapter we introduced concepts related to "patterns" and their representation models for a pattern-base in order to incorporate them into a Pattern Base Management System (PBMS). Since patterns are compact and rich in semantics representations of raw data (Theodoridis et al., 2003), they share some common characteristics, but they are also differentiated according to the type they belong to. Moreover, there are also variations between patterns of the same type. As patterns are of great importance in many applications, the need of a Pattern Base Management System (PBMS) is emerging.

The logical model of a PBMS defined in (Rizzi et al. 2003), includes three basic concepts. The Pattern-Type, the Pattern and the Class. A pattern-base should efficiently support these concepts, thus the appropriate representation model had to be defined. As it has been shown, patterns nature requires a data-oriented approach whereas traditional databases follow a structure-oriented direction. For the pattern representation problem a semi-structured model is more appropriate than a relational or an object-relational schema. Using XML for the implementation of the pattern-base, we could achieve to build a more complete and general PBMS.

Other approaches have been proposed but their target is not an integrated and general pattern-base management system. Among the possible representation standards, the PMML is the most promising. But, although PMML is an XML-based language and tends to support more and more pattern types, a more general aspect should be adopted in a PBMS. Patterns should be defined per application or scientific area, so the system will be open to user extensions. Pattern querying and data-to-pattern mapping are issues that PMML is not currently taking into account, though important in order to create a more complete PBMS.

PMML can be used to represent data mining patterns and can be enhanced with metadata to support features that would be essential for a PBMS. In chapter 5, a more detailed description of the PMML schema and the required metadata is provided.

Having defined the appropriate representation model (XML schemata and documents), we can deal with more complex pattern operations and functions, like pattern comparison and validation. Those advanced operations are of great importance in many real-world applications. In the following chapters we deal with the comparison of crisp and fuzzy clusters, to extend the PANDA comparison framework with clustering comparison algorithms. Until now, no comparison function has been defined for the EM or the Fuzzy-C-means clustering patterns.

3 Pattern Comparison – the case of Crisp Clustering

In this chapter we present the methods and algorithms that patterns can be compared to facilitate high level comparison of raw data. We focus on clustering patterns extracted with the EM clustering algorithm (Dempster et al., 1977). These clusters are represented by distributions and thus a proper function of comparing distributions is also presented. As a case study, we provide two real-case scenarios of image comparison through clustering patterns comparison to present the potential use of the pattern comparison framework.

3.1 Introduction

We focus on clustering patterns comparison and more specifically on clusters extracted with the EM clustering algorithm (Dempster et al., 1977) as there is no function defined for the comparison of clusters represented by distributions in the PANDA framework (that will be presented in section 3.2).

EM (Expectation-Maximization) is a well known and widely used clustering algorithm that can find number of distributions of generating data and build "mixture models". It identifies groups that are either overlapping or varying sizes and shapes. EM algorithm performs maximum likelihood estimation for samples in mixture model. EM uses probability of cluster membership instead of a distance metric, and samples are not assigned to 1 cluster, but partially to different clusters (proportionally to distribution). EM is much more general than just "clustering", it finds number of distributions generating data and builds "mixture models".

In order to compare clusters described by distributions, we have to compare the distributions themselves. Towards this end, we use the *Cohen's d* (Cohen, 1988) distance function.

The general comparison process is supported by the functions of the PANDA comparison framework (Ntoutsi et al., 2007). In this chapter we define new comparison functions over the PANDA framework, extending it to support EM clustering patterns and we present real-life applications, while Ntoutsi (2008) presents applications dealing mostly with the comparison of frequent itemset and decision trees patterns.

3.2 Pattern Similarity Definition

PANDA framework (Ntoutsi et al., 2007) provides the functions to compare simple and complex patterns. Simple patterns are extracted from raw data using the data mining process (clusters of raw data), while complex patterns are composed from simple ones (eg. a clustering on a set of clusters – clusters of clusters). Using the algorithms of the PANDA framework, we can compare patterns of the same pattern type (i.e. clusters with clusters, or association rules with association rules etc.).

Due to the compact and rich in semantics representation of patterns, PANDA can be used to compare patterns with large degree of complexity. PANDA framework uses the 2-component property to compare two patterns. The basic notion of this property is that the majority of patterns can be sufficiently described by the structure and the measure component.

The similarity is expressed as a distance *dis*, where the minimum distance indicates the maximum best matching, between two patterns p_1, p_2 of the same type can be computed by combining, by means of an aggregation function f_{aggr}, the distance between both the structure *s* and the measure *m* components (Ntoutsi et al., 2007):

$$dis(p_1, p_2) = f_{aggr}(dis_{struct}(p_1.s, p_2.s), dis_{meas}(p_1.m, p_2.m)) \quad (3\text{-}1)$$

where $p_i.s$ and $p_i.m$ denote the structure and the measure respectively of the pattern p_i.

The dot in this notation denotes that the variable on the right is a member of the pattern instance on the left, according to the notation used in object-oriented modeling.

If both the patterns to be compared have the same structure component, the dissimilarity function only takes into account the distance between the measure components.

Efficient definition of the structure and measure of patterns extracted from the raw data, as well as appropriate selection of aggregation logic and distance functions to assess the respective distances, are of great importance for every single and different application.

Next, we will describe the methodology of comparing clustering patterns based on the concepts presented above.

In section 2.3 we have defined the pattern type concept as a quintuple pt = (n, ss, ds, ms, f). In order to define comparison functions for clusters and other patterns, the *ss* and *ms* parts are needed. The three other parts are not used for the comparison function and thus in the following we are not dealing with them. To simplify the notation and the description of the comparison functions, we redefine the pattern type concept as a pair *PT* = <*SS*, *MS*>, where *SS* defines the pattern space by describing the structure schema of the pattern type, while the measure schema *MS* quantifies the quality of the source data representation achieved by patterns of this pattern type. Note, that from this point forward we will refer to these two parts of the pattern type except if a more detailed pattern-type description is needed.

As an example, consider a pattern type representing Euclidean-distance, spherical-like clusters in a *D*-dimensional space. The structure of such a pattern type may be modeled by specifying the cluster center (a *D*-dimensional vector) and a radius (a real value). The measure for a cluster might be, for instance, its support, that is, the fraction of the data points represented by the cluster. As such:

$$EuclideanCluster = \begin{pmatrix} SS : \left(center : [\text{Real}]_1^{\text{D}}, radius : \text{Real}\right) \\ MS : \left(\sup p : \text{Real}\right) \end{pmatrix}$$

As already mentioned in previous chapter, a pattern type *PT* is called *complex* if its structure schema *SS* includes another pattern type, otherwise *PT* is called *simple*. Thus, a *EuclideanCluster* is a simple pattern type,

whereas a clustering extracted e.g. by a partitioning clustering algorithm is considered a complex pattern type since it can be modeled as a set of clusters with no measure component:

$$PartitioningClustering = \begin{pmatrix} SS : \{EuclideanCluster\} \\ MS : \perp \end{pmatrix}$$

In this notation, if *PT* is a pattern type then $p = < s, m >$ is an instance of *PT*, where *s*, *m* are the corresponding structure and measure values of the pattern. With respect to the previous example, a possible instance of a 3-D *EuclideanCluster* could be:

$$Cluster1 = \begin{pmatrix} s : (center : [0.1, 0.3, 0.45], radius : 0.77) \\ m : (\sup\ p : 0.15) \end{pmatrix}$$

According to PANDA framework, as mentioned above, the distance *dis* between two simple patterns p_1, p_2 is computed by the function (3-1).

On the other hand, the distance between two complex patterns is defined as the aggregate distance between their constituent patterns, according to a coupling that associates constituent patterns (this is a recursive definition since a complex pattern could be composed of other complex patterns, and so on).

PANDA framework provides a number of distance, matching and aggregation functions. More on the PANDA framework and the functions it supports can be found in (Ntoutsi et al., 2007; Ntoutsi, 2008). Following the comparison method that PANDA framework describes, we define our methodology, the distance measures and the aggregation functions that will be used to compare EM clustering patterns.

3.3 Comparison of Clustering Patterns

In order to compare clustering patterns we should describe the whole process, from cluster creation to cluster matching and comparison. The similarity of the patterns depends on the structure and measure components and thus, the details of the clustering patterns and their representation should be analyzed.

The methodology of extracting and comparing clustering patterns involves the following steps:

1. Feature extraction from raw data
2. Application of the appropriate Data mining – Clustering algorithm
3. Pattern Instantiation
4. Computation of Pattern Similarities

In detail,

1. Feature extraction from raw data

The first step is to extract the features from the raw data, that will be clustered. Raw data could be a text or an image for example.

In the latter case, the image is raster scanned with a sliding window of user-defined size, sampling image blocks at a given sampling step. The sampling step may allow consecutive blocks to overlap. For each block, a set of N features f_i, i = 1, …, N, is calculated to form a single feature vector F. The number of feature vectors produced for each image depends on the size, the dimensions of the sliding window and the sampling step. For image data, color, texture and shape are the three major classes of image features commonly used. The output of the feature extraction step is a set of vectors with N features.

2. Application of the appropriate Data mining – Clustering algorithm

In the next step, the low-level feature vectors are clustered using mixture models that model the data by a number of Gaussian distributions. A cluster corresponds to a set of distributions, one for each dimension of the dataset. Each distribution is described in terms of mean and standard deviation. A probabilistic approach to assigning feature vectors to clusters is used.

For 1-dimensional datasets, a mixture is a set of *c* Gaussian probability distributions, representing *c* clusters. The parameters of a mixture model is determined by the *Expectation Maximization* (EM) algorithm (Dempster et al., 1977). With *c* Gaussians, the probability density function of a variable *X* is

$$f(X|\theta)=\sum_{i=1}^{c} pp_i \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} e^{-\frac{1}{2}(X-\mu_i)^T \sum_i^{-1} (X-\mu_i)} \quad (3\text{-}2)$$

where $pp_i > 0$, $\sum_{i=1}^{c} pp_i = 1$, and d is the dimension of the feature vector. The set of model parameters $\theta\{pp_i, \mu_i, \Sigma_i\}$, $i = 1, \ldots, c$, consists of the prior probabilities pp_i of the Gaussian i, the mean vector μ_i and the covariance matrix Σ_i for the Gaussian i, respectively. The EM algorithm is used to estimate the maximum likelihood L of θ given a set of features $\{x_1,\ldots,x_N\}$:

$$L(\theta|X) = \log \prod_{j=1}^{N} f(x_j|\theta) \qquad (3\text{-}3)$$

The model parameters are initialized with random values. The algorithm starts by calculating the probabilities that a vector should belong to each distribution. These probabilities are used to compute a new estimate for the parameters. The whole process is repeated until the parameters converge to a constant or almost constant estimate. The algorithm results in a set of distributions, a vector of pairs of means μ and standard deviations σ, each of which corresponds to a feature, and outputs the size of the cluster (the number of vectors that belong to the cluster). The vector of means μ of the distributions for every feature represents the centroid of the cluster.

The EM algorithm exhibits many advantages over other clustering algorithms. Combining EM with the υ-fold cross-validation algorithm (Stone, 1974) the number of clusters in the output of the algorithm can automatically be determined. The υ-fold cross-validation technique works by partitioning the data into υ equally-sized segments. Starting with one cluster, EM is performed υ times holding out one segment at a time for test purposes and the likelihood is averaged over all the results. Next, EM is performed over two clusters and if the likelihood increases, the number of clusters is set to two and the process is repeated until the estimated likelihood begins to decrease (Witten and Frank, 2005).

Furthermore, the EM algorithm is more general than e.g. K-means (Hartigan, 1975), as it can find clusters of different sizes and ellipsoidal shapes. Most importantly, the distributions representing the clusters at the output of the EM algorithm can be easily utilized for pattern instantiation by the PANDA framework.

3. Pattern Instantiation

The clusters resulting from the EM algorithm are represented and handled according to the PANDA formalization presented in Section 3.2. Hence, given a clustered Object (of data) comprising of M simple patterns P_i, i = 1,…,M, and with respect to the output of the EM algorithm, a pattern P_i represents a pattern of the data:

$$P_i = \begin{pmatrix} SS : (D : [[\mu : [\text{Real}], \sigma : [\text{Real}]]_1^N), \\ MS : (pp : \text{Real}), (\text{SV} : \text{Real}) \end{pmatrix}$$

More specifically, the structure schema *SS* of a Pattern is represented by the pair (μ, o) of the distribution D_j for each of the N features (j=1, …, N) in pattern P_i, respectively. Correspondingly, the measure schema *MS* of a Pattern is represented by two values, the *prior probability* (*pp*) and the *Scatter Value* (*SV*) of P_i. Formally, the prior probability *pp* is defined as the fraction of the feature vectors of the *Object* that belong to pattern P_i. Intuitively, *pp* is equivalent with the *support* measure widely used in data mining models. In this case, it provides an indication of the size of the Object. On the other hand, *SV* is a measure of the cohesiveness of the data items in a cluster with respect to the centroid of the cluster, and it is a commonly used intrinsic measure of the quality of a cluster (Littau, 2003). Formally, the scatter value *SV* of an object is defined as:

$$SV = \sum_{k \in P_i} \left(x_k - c_{P_i} \right)^2 \qquad (3\text{-}4)$$

where x_k are the feature vectors that belong to pattern P_i and c_{P_i} is the corresponding centroid, which is also a vector having the same dimensionality as x_k, and its value in each dimension is computed as the average from the corresponding features values belonging to pattern P_i. A low scatter value indicates good scatter quality, but it should be noted that this is a relative measure of quality, since it depends on the number of items in the cluster.

In this context, an Object is considered as a complex pattern:

$$Object = \begin{pmatrix} SS : \{ P \}, \\ MS : \perp \end{pmatrix}$$

consisting of a set of simple patterns, which follow the definition in Eq. 4.

4. Computation of Pattern Similarities

Aiming at the estimation of the similarity between two Objects (defined as complex patterns), we first have to define the distance over the structures and the measures of two simple patterns P_1 and P_2. Since complex patterns are decomposed into a number of simple patterns, in comparing two objects, O_1 and O_2, we need a way to associate component patterns of O_1 to component patterns of O_2. To this end, the *coupling type* constrains the way component patterns can be associated (i.e., matched). Below, we first propose an effective way to measure the distance between two simple patterns, and then we present (see Eq. (3-12)) our choice for coupling them.

The distance between the measures of two patterns is proposed to be defined as the absolute difference of the scatter values each one weighted by the corresponding prior probability of the patterns, normalized by the sum of the two scatter values. Formally:

$$dis_{meas}(P_1, P_2) = \frac{\left| P_1.pp \cdot P_1.\mathrm{SV} - P_2.pp \cdot P_2.\mathrm{SV} \right|}{P_1.\mathrm{SV} + P_2.\mathrm{SV}} \qquad (3\text{-}5)$$

Intuitively, equation (3-5) quantifies the inter-pattern divergence between the cohesiveness of two clusters. It should be noted that this definition overrides the inefficiency of the relativeness of the scatter value with respect to the number of items in the cluster, as each scatter value is weighted by the fraction of the feature vectors of the image that belong to pattern P_i.

Regarding the structural similarity between P_1 and P_2, we search for a measure that evaluates the closeness of two sets of distributions, as P_1 and P_2 are. Further decomposing the problem, we should first define a method of computing the similarity between two distributions D_1 and D_2. To achieve this, we use the standardized difference d between two distributions, as defined by Cohen (1988). *Cohen's d* is defined as the absolute difference between the means of the distributions, divided by the root mean square of the two standard deviations.

$$d(D_1, D_2) = \begin{cases} \dfrac{|D_1.\mu - D_2.\mu|}{\sqrt{\dfrac{D_1.\sigma^2 + D_2.\sigma^2}{2}}}, & \text{if } D_1.\sigma \neq 0 \text{ or } D_2.\sigma \neq 0 \\ |D_1.\mu - D_2.\mu|, & \text{otherwise} \end{cases} \quad (3\text{-}6)$$

Cohen's distance d is a non-negative real number interpreting the overlap between two distributions. If d is zero, the distributions are identical. Low d indicates quite similar distributions whereas high d indicates quite dissimilar distributions. If both standard deviations are zero, the absolute difference between the means is used as the distance between the distributions.

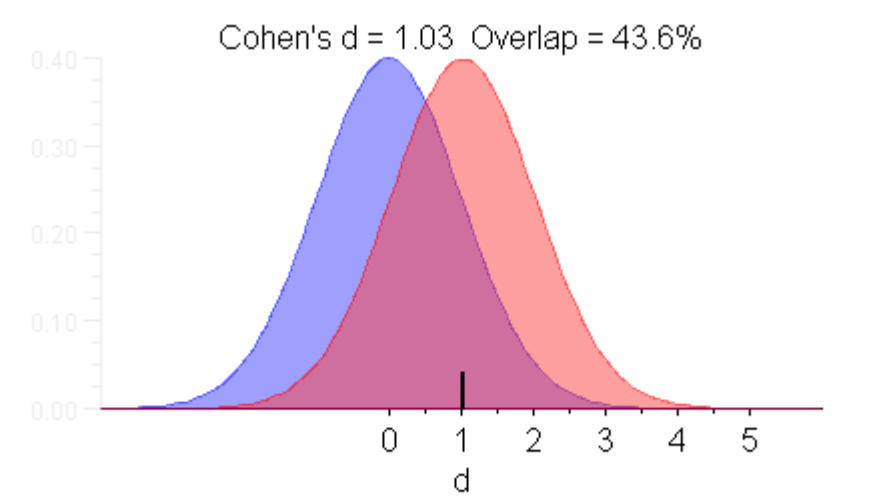


Figure 3-1 Graphical representation of the similarity between two distributions using the Cohen's d measure

Cohen's distance is the vehicle to automate and materialize the intuitive overlap between two distributions. Having this, we define that the structural distance between two sets of distributions (i.e. two patterns P_1 and P_2) should be the result of an aggregate function g_{aggr} (Eq. (3-7)), which interrelates the different distance scores achieved by each pair of distributions:

$$dis_{struct}(P_1, P_2) = g_{aggr}\left(\frac{d(D_j^1, D_j^2)}{\delta}\right), \ \forall\ j = 1, 2, ..., N \quad (3\text{-}7)$$

where d is Cohen's distance and δ is a normalization factor of the domain of values of the g_{aggr} function (g_{aggr}: $[0,1] \rightarrow [0,1]$), which intuitively corresponds

to the *Cohen's d* score over which two distributions are considered totally dissimilar (i.e. they do not overlap). In this connection, g_{aggr} function can be any mapping that initially performs a *feature selection* process and subsequently applies the aggregation function upon the selected features. Examples of such functions include: (a) the *minimum* function g_{min} (i.e. selection of the most similar distributions) (b) the *average* function g_{avg} (i.e. selection of the average among the distances computed for each pair of the N features) and (c) the *average of the k Nearest Distributions* function g_{avg_kND} (i.e. selection of $k \leq N$ most similar pairs of distributions). In the last case, the k parameter may not be given explicitly, yet it can be defined implicitly by relaxing the δ parameter. Formally:

$$g_{\min} = \min_{j=1}^{N} \{d(D_j^1, D_j^2)\} \tag{3-8}$$

$$g_{avg} = \frac{1}{N} \sum_{j=1}^{N} d(D_j^1, D_j^2) \tag{3-9}$$

$$g_{avg\_kND} = \frac{1}{k} \sum_{j=1}^{k} kND(d(D_j^1, D_j^2)) \tag{3-10}$$

where function *kND* returns the k most similar distributions.

To this point, we have defined dis_{meas} and dis_{struct} (Eq. (3-5) and (3-7), respectively) between two patterns. In the sequel, we aggregate these distances by using a *wise weighted sum function*. Formally, the distance $dis(P_1, P_2)$ between two patterns P_1 and P_2 is defined as:

$$dis(P_1, P_2) = dis_{struct}(P_1, P_2) + (1 - dis_{struct}(P_1, P_2)) \cdot dis_{meas}(P_1, P_2)^2 \tag{3-11}$$

The intuition behind our choice is that the more similar are the structures, the more the measure distance should contribute to the total distance score. This implies that if structures are totally different, the distance should be 1, irrespective of the measure. This choice further implies that we give

emphasis on the structural similarity. This is additionally strengthened by multiplying the factor 1– dis_{struct} (i.e. the similarity between the structures of the patterns) with a smaller value than the actual measure distance dis_{meas}. Recall that dis_{meas} takes values in the domain [0,1], so by taking its square we denote the relaxing of the dis_{meas} contribution.

Having defined the distance between simple patterns, to compare two Objects O_1 and O_2 (i.e. two complex patterns) we adopt the coupling methodology between the different patterns of each object as follows:

$$dis(O_1, O_2) = \frac{1}{M \cdot K} \cdot \sum_{i=1}^{M} \sum_{j=1}^{K} dis\left(P_i^{O_1}, P_j^{O_2}\right) \tag{3-12}$$

where M and K is the respective number of constituent simple patterns of each object with respect to the output of the EM algorithm. Various coupling types can be applied in the context of the PANDA framework (Bartolini et al., 2004), but the all-by-all matching expressed by Eq. (3-12) avoids bias towards specific patterns. The final outcome is the average of all possible matchings.

The best coupling type to compare two clusterings, is subject to discussion, and it is depending on the application. The possibilities are explained below.

In Figure 3-2, two clusterings are represented. Clustering A has 3 clusters while clustering B has 4 clusters.

In order to compare the two clusterings, the following couplings can be made:

Case 1. Compare each cluster of the Clustering A with each cluster of Clustering B.

In this coupling each and every pair of matchings are made and the best matching for every Cluster is kept. All the matching values are aggregated at the end of the process (using probably the mean).

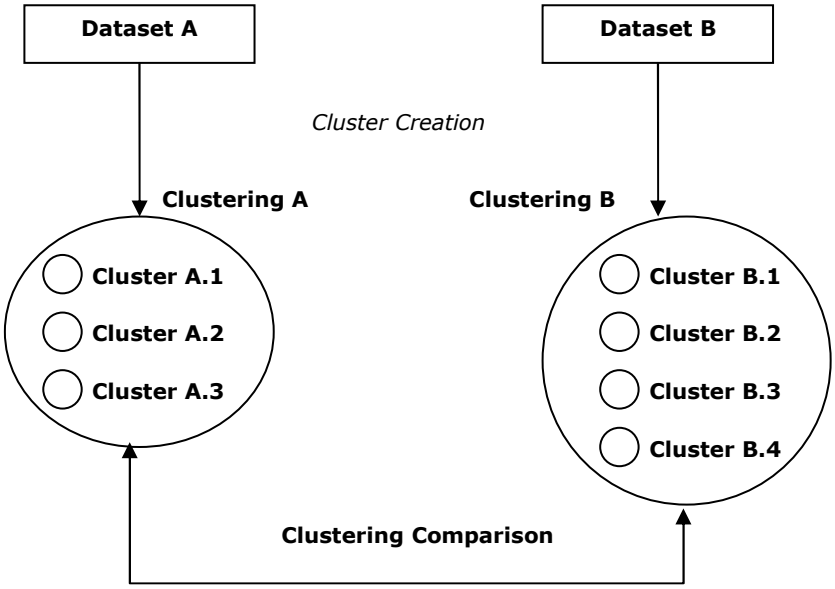


Figure 3-2 Comparing two clusterings *Clustering A* and *Clustering B*

Case 2. Compare each cluster of the Clustering A with each cluster of Clustering B but do not allow duplicate matches.

In that case if cluster A.1 for example has the best (higher value) match with say B.1, those clusters are not checked again. In the following, cluster A.2 will be checked with B.2, B.3 and B.4 and the best match will be kept.

Case 2a. This procedure may be followed from the opposite direction, that is, to check each cluster of Clustering B for its best match with clusters from Clustering A. A matrix can be constructed and the best matchings may be kept.

Case 2c. It is obvious that in case there are two clusterings with different number of clusters, like our example, not every cluster will match with another cluster.

A possible solution to this case is to sort the two clusterings from the largest to the smallest cluster, provided that larger clusters should match better with each other and thus leaving the smallest clusters unmatched.

Case 2c. Another way to overcome this issue, is to allow duplicate matches, while not compute every possible combination – that would be the first case presented.

Which is the best coupling type depends on the application and the expert user.

The concepts and the methodology described in this section can be used in many real life applications to categorize and to compare patterns extracted from a variety of raw data. In the two sections that follow, two studies for comparing clusterings extracted from images, are presented. Both studies use an image database and clustering techniques to categorize features extracted from these images. The aim of these studies is to create a content-based image retrieval methodology.

The methodology involves four steps: a) feature extraction from each of the stored and the query images, b) clustering of the extracted feature vectors per image, c) pattern instantiation of the clusters, and d) computation of pattern similarities. The registration of a new image in the database involves the first three of the four steps described for image retrieval (a, b, and c).

The first study (Iakovidis et al., 2007) uses cultural heritage images originating from the database of the Foundation of Hellenic World (FHW, 2009) while the second study (Iakovidis et al., 2006) uses radiographic images from the IRMA (Image Retrieval in Medical Applications) dataset (Lehmann, 2003).

3.4 Application I: Comparing Clusters of medical images

One of the primary tools used by physicians is the comparison of previous and current medical images associated with pathologic conditions. As the amount of pictorial information stored in both local and public medical databases is growing, efficient image indexing and retrieval becomes a necessity.

In the last decade the advances in information technology allowed the development of Content-Based Image Retrieval (CBIR) systems, capable of retrieving images based on the similarity their features have with the features of one or more query images. Some of these systems are QBIC (Faloutsos et al., 1994), VisualSEEK (Smith & Chang, 1996), Virage (Hampapur et al., 1997), Netra (Ma & Manjunath, 1999), PicSOM (Laaksonen et al., 2000), SIMPLicity Wang et al., 2001), CIRES (Iqbal &

Aggarwal, 2002), and FIRE (Deselaers et al., 2004) . More than fifty CBIR systems are surveyed in (Veltcamp & Tanase, 2000).

The benefits emanating from the application of content-based approaches to medical image retrieval range from clinical decision support to medical education and research (Müller et al., 2004). These benefits have motivated researchers either to apply general-purpose CBIR systems to medical images (Deselaers et al., 2004) or to develop dedicated ones explicitly oriented to specific medical domains. Specialized CBIR systems have been developed to support the retrieval of various kinds of medical images, including High Resolution Computed Tomographic (HRCT) images (Shyu et al., 1999), breast cancer biopsy slides (Schnorrenberg et al., 2000), Positron Emission Tomographic (PET) functional images (Cai et al., 2000), ultrasound images (Kwak, 2002), pathology images (Zheng et al., 2003) and radiographic images (El-Naqa et al., 2004).

Common ground for most of the systems cited above is that image retrieval is based on similarity measures estimated directly from low-level image features. This approach is likely to result in the retrieval of images with significant perceived differences from the query image, since low-level features usually lack semantic interpretation. This has motivated researchers to focus on the utilization of higher-level semantic representations of image contents for content-based medical image retrieval. Recent approaches include semantic mapping via hybrid Bayesian networks (Lin et al., 2006), Semantic Error-Correcting output Codes (SECC) based on individual classifiers combination (Yaoa et al., 2006), and a framework that uses machine learning and statistical similarity matching techniques with relevance feedback (Rahman et al., 2007). However, these approaches involve supervised methodologies that require prior knowledge about the dataset and introduce constraints to the semantics required for the image retrieval task.

A state-of-the-art CBIR approach has been presented in (Greenspan & Pinhas, 2007). It utilizes a continuous and probabilistic image representation scheme that involves Gaussian mixture modelling (GMM) along with information-theoretic image matching via the Kullback–Leibler (KL) measure. The results reported in (Greenspan & Pinhas., 2007) show

that this approach is very effective for radiographic image retrieval; however, its efficiency for large image retrieval tasks still remains a challenge.

In this study, we propose an unsupervised approach for efficient content-based medical image retrieval that utilizes similarity measures, defined over higher-level *patterns* that are associated with clusters of low-level image feature spaces. The proposed approach combines the advantages of the clustering-based CBIR methodologies (Stehling et al., 2001; Carson et al., 2002; Yixin Chen et al., 2005) with a semantically rich representation of medical images. Moreover, unlike related CBIR approaches that exploit multi-dimensional indexing techniques, such as *R*-trees (Faloutsos et al., 1994), (Petrakis and Faloutsos, 1997), iconic index trees (Wu & Narasimhalu, 1994), and meshes of trees (Jeng & Hsiao, 2005), the efficiency of the proposed approach is hardly affected by increasing the dimensionality of the low-level feature representation.

The major contributions of this study are the following:

- We define a novel representation of medical images treated as rich-in-semantics *complex patterns*. Each complex pattern comprises a set of simple patterns representing clusters of image regions associated with anatomic specimens in an unsupervised way. The pattern representation of clusters involves both structural descriptors and quality measures.
- We propose a novel scheme for the assessment of the similarity between complex patterns (i.e. medical images) for CBIR purposes.
- We conduct a comprehensive set of experiments over a publicly available set of radiographic images, in order to thoroughly evaluate our approach and demonstrate its effectiveness and efficiency in comparison to state-of-the-art techniques.

3.4.1 The proposed methodology

The proposed content-based medical image retrieval scheme is outlined in Figure 3-3. It involves four steps: a) low-level feature extraction from each of the registered and the query images, b) clustering of the extracted feature vectors per image, c) pattern instantiation of the resulted clusters, and d) computation of pattern similarities. The registration of a new image into the

database involves steps a, b, and c, whereas step d is processed during the retrieval task.

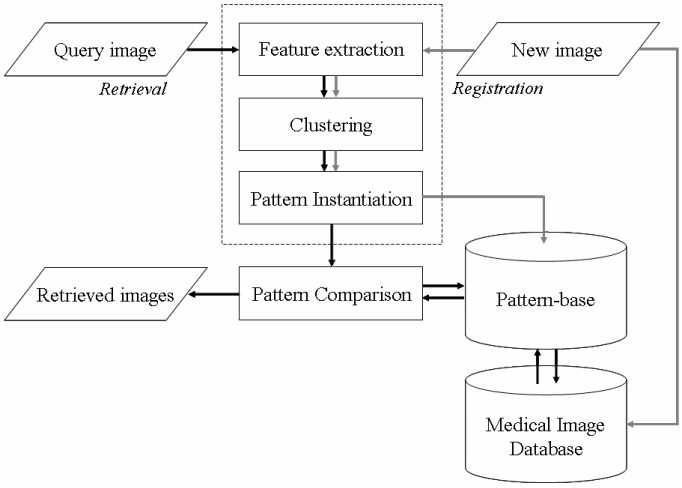


Figure 3-3 Outline of the proposed content-based image retrieval methodology. The black arrows indicate the data flow for image retrieval, whereas the grey arrows indicate the data flow for the registration of a new image.

In the case of radiographic medical image retrieval, local grey level intensity and texture features have proved to discriminate best the depicted specimens. Such features include raw pixel values used along with an image distortion similarity model, local feature histograms, and local relational features (Deselaers et al., 2004b), (Setia et al., 2006). Recently, in (Greenspan and Pinhas, 2007) it was shown that highest retrieval precision can be achieved by combining intensity and texture contrast along with the corresponding spatial coordinates. However, the introduction of spatial information into the feature vectors makes them dependent on the patients' position. Although patients are usually positioned in a standard way during the acquisition of a radiograph, there are still many cases in which this is not practically feasible. For example, this is the case with the acquisition of radiographs of critically ill patients using portable radiographic devices (Bongard and Sue, 2002) and with the acquisition of radiographs of upper or lower extremities (Karkanis et al., 2003).

In this study, we adopt a standard, multiscale statistical approach for the representation of the radiographic image regions that preserves local features, and does not depend on spatial coordinates. It is based on the 2-dimensional Discrete Wavelet Transform (2D-DWT), an efficient, yet effective transformation that has proved useful in a variety of medical image processing and analysis applications, including CBIR (Müller et al., 2004),

(Mallat, 1999, Wang et al., 1998, Karkanis et al., 2003, Wang, 2001). It enables coding of image texture into detail (higher frequency) coefficients, whereas image intensity information can be extracted from its approximation (lower frequency) coefficients (Mallat, 1999). A compact representation of the distributions of the approximation and the detail coefficients can be obtained by first–order statistical approximation.

However, it should be noted that this study focuses on the utility of the proposed pattern similarity scheme rather than on the selection of an optimal feature set for a particular image retrieval task.

According to the pattern instantiation scheme described in section 3.3, in the current application, a $Specimen_i$ is instantiated for each pattern P_i representing a physical anatomic specimen in a medical image:

$$Specimen_i = \begin{pmatrix} SS : \left(D : [[\mu : [\text{Real}], \sigma : [\text{Real}]]_1^N\right) \\ MS : \left(pp : [\text{Real}],\ SV : [\text{Real}]\right) \end{pmatrix}$$

In this context, a medical image MI is considered as a complex pattern:

$$MI = \begin{pmatrix} SS : \{Specimen\}, \\ MS : \perp \end{pmatrix}$$

consisting of a set of simple patterns (i.e. *specimens*).

Clustering and pattern similarity computation schemes also follow the schemes described in section 3.3.

In the next section the results of the experimental studies are presented.

3.4.2 Experimental Results

A number of experiments was performed with radiographic images from the IRMA (Image Retrieval in Medical Applications) dataset (Lehmann, 2004), which is often used as a reference for medical image retrieval tasks. It currently contains 10,000 arbitrarily selected anonymous radiographic images taken randomly from patients of different ages, genders and pathologies during medical routine. The images are categorized into 116 classes according to the IRMA code (Lehmann et al., 2003). This code comprises of four fields: a) the imaging modality; b) direction of the imaging device and the patient; c) the anatomic body part that is examined; and d) the system under investigation. The particular dataset comprises only of

plain x-ray images of various directions (such as anteroposterior and mediolateral), anatomic body parts (such as cranium, spine, arm, elbow and chest) and systems under investigation (such as musculoskeletal, gastrointestinal and uropoietic). The IRMA code information of each image is provided as ground truth along with the dataset. Other patient data and pathology information are unavailable. All radiographic images are in 8-bit greyscale format and have been downscaled to fit into a 256×256-pixel bounding box maintaining the original aspect ratio.

From the available dataset a subset of 90% of the images was registered in the database, whereas a non-overlapping subset of 10% of the images was used for querying the pattern-base. Each image was sampled in blocks using overlapping sliding windows. The details of the feature extraction method used include a 3-level biorthogonal spline wavelet decomposition of each sampled block and the estimation of the first two wavelet moments from each band. This process results in a 20-dimentional feature vector per block.

The determination of the sampling parameters was based on preliminary experiments seeking for the maximum average distance (Eq. 11) between complex patterns *MI* of the different categories comprising the registered dataset. The sampling parameters tested before each CBIR experiment include sliding windows of 32×32, 64×64 and 128×128 pixels. In all cases, the maximum average distance was obtained with windows of 64×64-pixels. Variation of the overlap (0%, 25%, 50% and 75%) between the sampled blocks did not affect this result. Increasing the overlap provides better localization of the patterns but produces many more sampled blocks, affecting the efficiency of both the feature extraction and the pattern instantiation tasks. Thus, a 50% overlap, i.e. a 32-pixel step, was used as a compromise between localization and efficiency.

In the following, we present qualitative results of the pattern instantiation realized via clustering, and measure the performance of the proposed scheme, in terms of effectiveness and in terms of efficiency.

Pattern Instantiation via Clustering

The feature vectors extracted from each image were clustered using an implementation of the EM algorithm available in the WEKA data mining tool (Witten and Frank, 2005) using the 10-fold cross-validation algorithm to

determine the number of clusters. Each cluster was represented by a pattern $Specimen_i$, $i = 1, \ldots, M$ (see Eq. 4), and each image was represented by a complex pattern *MI* (see Eq. 6). Figure 3-4a illustrates three radiographic images from breast, abdomen and hand categories (from left to right). The respective clusterings obtained are illustrated in Figure 3-4b. The different grey levels in Figure 3-4b indicate the different *specimen* patterns found in the images. Figure Figure 3-4c illustrates projections of the 20-dimensional feature vectors to a 3-dimensional space constructed according to the centroid-preserving projection technique (Kopanakis and Theodoulidis, 2003). It can be observed that the clustering produced is quite meaningful in terms of semantics, i.e. the breast and the perceived differences in its structure are clearly depicted, the region of the abdomen is well defined and separated from the upper part of the body, and the palm is differentiated from the fingers. However, for the fingers the algorithm assigned two specimens instead of one, but this can be attributed to the large size of the sampled blocks as compared with the gap between the fingers.

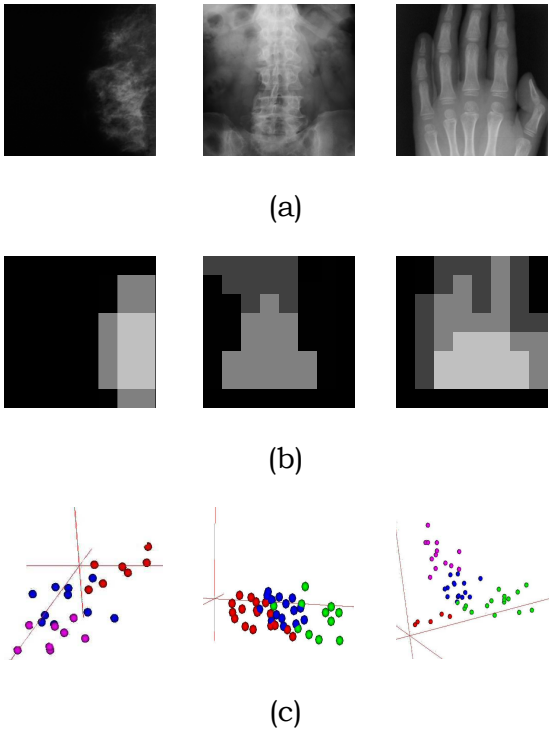


Figure 3-4 (a) Original radiographic images, (b) clustering output, and (c) three dimensional visual representation of the feature spaces.

Effectiveness

The patterns from the registered radiographic images were used to build a pattern-base (see Figure 3-3). In order to quantitatively assess the effectiveness of the proposed pattern similarity scheme, we evaluate its capability to retrieve images by adopting the popular *recall* and *precision* measures, where recall is defined as the ratio of the relevant images retrieved over the total relevant images in the database, and precision is defined as the ratio of the relevant images retrieved over the total number of images retrieved, relevant or not. To enable comparisons with other medical image retrieval methodologies using a standard single-figure measure, the Area Under the interpolated precision-recall Curve (AUC) is estimated (Davis and Goadrich, 2006).

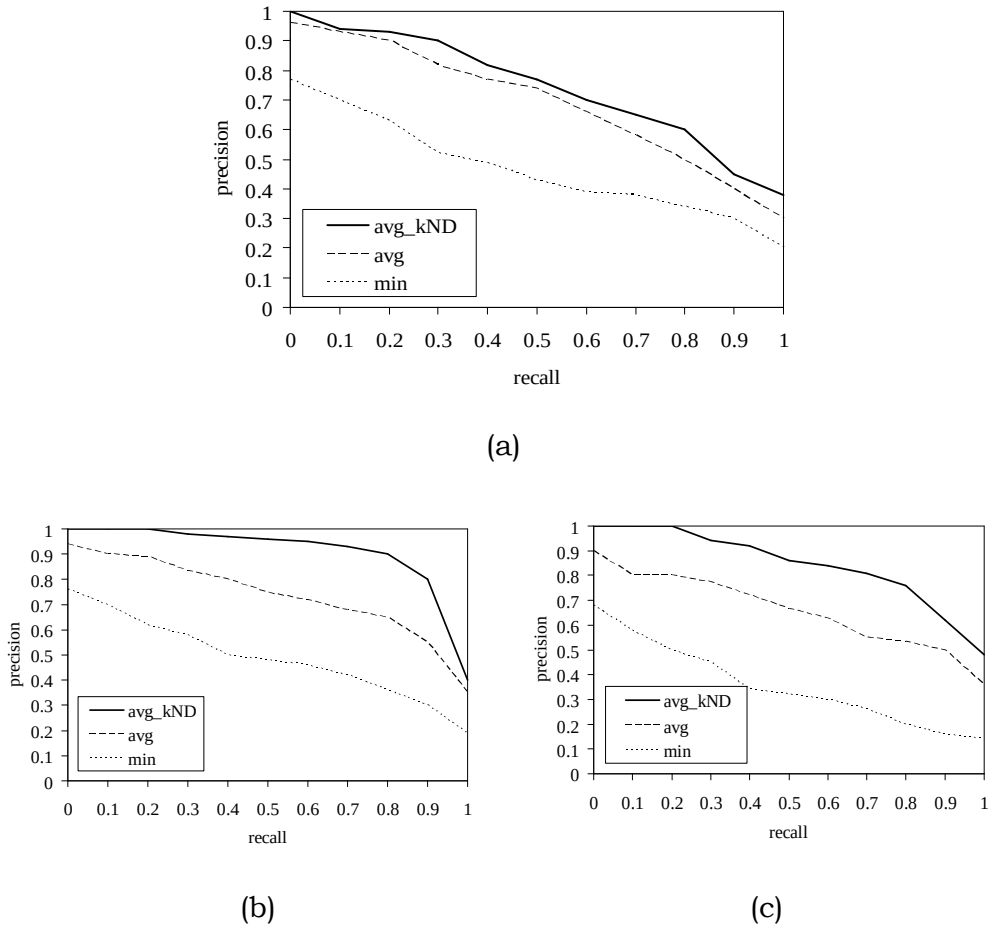


Figure 3-5 Average precision vs. recall using g_{avg_kND}, g_{avg} and g_{min} aggregation functions for (a) all, (b) chest, and (c) cranium, categories.

The proposed scheme was tested using the three alternative aggregation functions g_{aggr}. The results, in terms of average precision vs. recall estimated

for all 116 categories, are illustrated in Figure 3-5a. Indicatively, in Figure 3-5b and Figure 3-5c we present the precision vs. recall charts for two independent categories of chest and cranium radiographs. It is evident that best retrievals are achieved by using the *average of the k Nearest Distributions* function σ_{avg_kND}.

Figure 3-5a shows that for a recall of 90% the average precision achieved using σ_{avg_kND} is almost 45%, and the corresponding AUC estimated is 74%. It is worth noting that these results could only marginally improve upon a denser sampling scheme. Compared with a simple method that uses global grey level histograms as features and histogram intersection as an appropriate dissimilarity measure (Swain, M.J. and Ballard, 1991), the average precision for 90% recall is approximately 10%, and the corresponding AUC reaches only 17%. The AUC obtained with the proposed scheme using local grey level histogram information reaches 34%. The corresponding precision vs. recall curves are illustrated in Figure 3-6.

The precision reported in (Greenspan and Pinhas, 2007) for 90% recall seems to be comparable with the one achieved with the proposed approach; however, the dataset from which that precision is estimated is significantly smaller comprising only 1,500 radiographs from 17 categories. In order to derive comparable estimates between the two CBIR approaches a retrieval experiment was run with the proposed scheme on a subset of the available data generated according to the guidelines provided in (Greenspan and Pinhas, 2007). The AUC estimated for the proposed approach on this subset reached 78%, whereas the AUC estimated from (Greenspan and Pinhas, 2007) is approximately 66%.

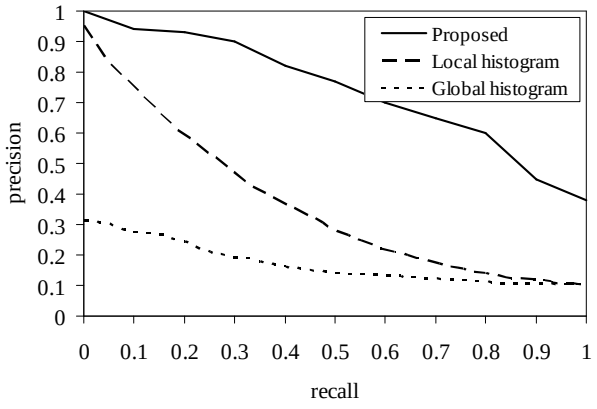
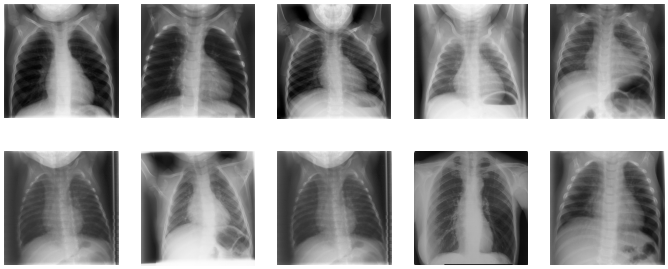
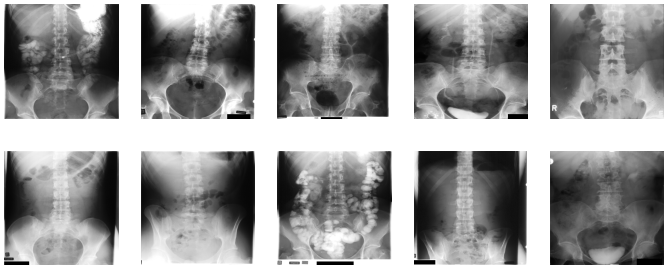


Figure 3-6 Comparative precision vs. recall chart.

Two example retrievals using g_{avg_kND} are illustrated in Figure 3-7. The first image of each sequence is the query image, and the rest are the nine retrieved images requested. Figure Figure 3-7a shows that all the retrieved images belong to the same category. Figure Figure 3-7b shows that two of the retrieved images belong to a different class than that of the query image. However, the main difference between the two categories is hardly perceivable and located in the region of pelvis (lower part of the image at the centre). Similar observations are valid for queries performed using radiographic images from other categories.



(a)



(b)

Figure 3-7 (a) A query requesting nine chest images similar to the upper-left image (1,1): All retrieved images belong to the same category; (b) A query requesting nine abdomen-gastrointestinal system images similar to the upper-left image (1,1): all retrieved images belong to the same category, except (1,4) and (2,5), which belong to abdomen- uropoietic system. (Notation (*i, j*) indicates the positioning of an image at the *i*-th row, *j*-th column in the figure.)

Efficiency

In this subsection, we measure the efficiency of the proposed medical image similarity scheme that involves pattern comparisons, in comparison with the performance of the conventional scheme that involves exhaustive comparisons of the feature vectors. A vector comparison in the conventional approach is considered equivalent to a pattern comparison in the proposed

scheme. The experiments were performed on a workstation with Intel Pentium M1.6 processor, 1 GB RAM and 60 GB hard disk.

We have chosen the sequential, exhaustive scan as the yardstick for our method, as other common methods such as R-trees are sensitive to the high dimensionality of the feature vectors, which is usual in CBIR applications (e.g. a dimensionality of 64 in (Weber, 1998) and at least $2\times N=40$ in our case, where N is the number of features in a pattern). The performance of these approaches degrades rapidly as dimensionality increases. For instance, it has been shown that even for a dimensionality of as low as 5, the R*-tree behaviour in similarity search is problematic (Weber, 1998). The main reason is that with the growth of the dimensionality the overlap in the internal nodes of the tree increases, and, as such, its discrimination ability decreases.

The speedup factor between the conventional and the proposed approach as a function of the number of blocks per image is illustrated in Figure 3-8. It can be observed that the advantage of the proposed approach increases with the number of blocks per image (e.g. by increasing the sampling step), and for a few hundreds of blocks per image it requires almost three orders of magnitude fewer comparisons than the conventional approach.

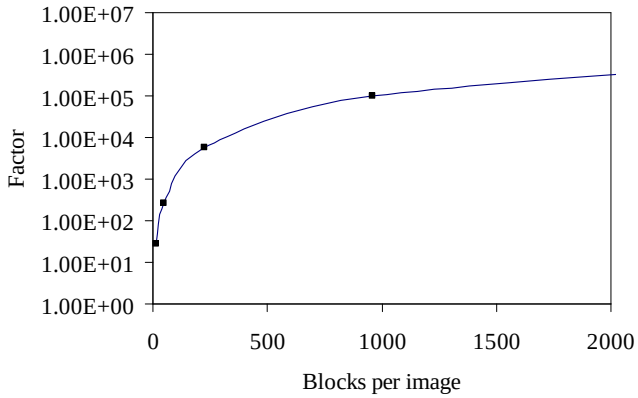


Figure 3-8 The speedup factor between the conventional and the proposed approach as a function of the number of blocks per image.

On the other hand, in (Greenspan and Pinhas, 2007) a speedup of two orders of magnitude compared to the conventional approach is reported. Moreover, in the same study it is noted that the GMM-KL framework is not yet capable of coping with large image retrieval tasks that extend more than 6,000 images due to the computational load involved with the KL measure.

We further estimated the average processing time (CPU plus I/O time) for the comparison of a pair of images. For the above experimental setting the proposed pattern similarity scheme requires always less than 0.1 msec. The average time required for the mixture model parameters converge to a constant or almost constant estimate is 0.22±0.04sec.

3.5 Application II: Comparing clusters of cultural images

Content-Based Image Retrieval (CBIR) of cultural heritage images is an emerging field of research bridging society, culture and information technology (Chen et al., 2004). Querying by example databases of paintings, sculptures, photographs, and documents of historical value from different civilizations, would facilitate both educational and research and enable the exploration of unknown inter and intra cultural associations.

Recently, studies targeting especially to the retrieval of cultural heritage images have appeared. Most of these studies propose methods based on color image features (Ardizzone et al., 2004), (Valle et al., 2006). More sophisticated approaches include the utilization of wavelet domain feature descriptors in conjunction with mixtures of stochastic models for the retrieval of Chinese paintings (Jia & Wang, 2004).

Common ground for most of the systems cited above is that image retrieval is based on similarity measures estimated directly from low-level image features, whereas it involves multidimensional, usually exhaustive, nearest neighbor searching over the whole set of the available feature vectors. However, such an approach can be time consuming with large image databases.

Research on improving the efficiency of the image retrieval process has mainly focused on image indexing techniques by utilizing data structures, such as R-trees (Faloutsos et al., 1994), (Petrakis & Faloutsos, 1997), feature index trees (Grosky & Mehrotra, 1990), iconic index trees (Wu & Narasimhalu, 1994), and meshes of trees (Jeng & Hsiao, 2005). Other approaches to improving efficiency, include clustering of the image feature spaces (Stehling et al. 2001), (Zhang R. & Zhang Z, 2002), and utilization of alternative similarity measures, usually dependent on feature sets (Berman & Shapiro, 1997), (Stehling et al., 2002).

3.5.1 The Proposed Methodology

The proposed CBIR approach is outlined in Figure 3-9. It involves four steps: a) feature extraction from each of the stored and the query images, b) clustering of the extracted feature vectors per image, c) pattern instantiation of the clusters, and d) computation of pattern similarities. The registration of a new image in the database involves the first three of the four steps described for image retrieval (a, b, and c).

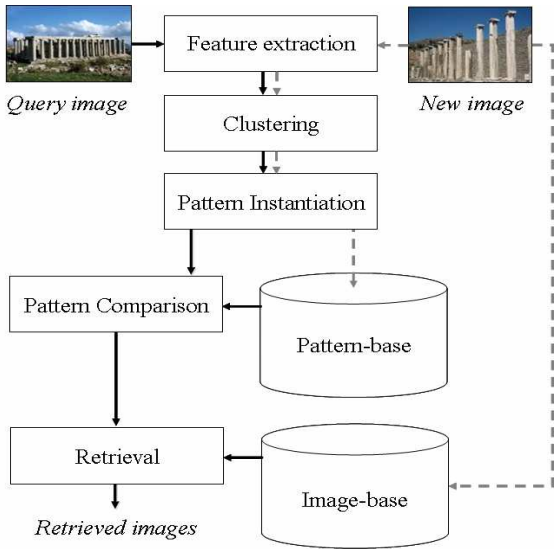


Figure 3-9 *Outline of the proposed pattern-based CBIR approach. The solid arrows indicate the data flow for image retrieval, whereas the dashed arrows indicate the data flow for the registration of a new image.*

Each of the images stored in the database, as well as the query image is raster scanned with a sliding window of user-defined size and sliding step. The sliding step may allow windows to overlap between each other. For each window N features f_i, i=1,2, N are calculated to form a single feature vector F. The number of feature vectors produced for each image depends on the size, the dimensions and the step of the sliding window.

Aiming to illumination invariant representation of the images we have considered Local Binary Pattern (LBP) distributions as features. LBP features are calculated from the weighted mean of pixel values over a small neighborhood, in which each pixel is considered separately. The LBP features were supplemented by an orthogonal measure of local contrast according to which the average of the gray levels below the center pixel is subtracted from that of the gray levels above (or equal to) the center pixel (Ojala et al., 1996). Comparative studies have demonstrated that the use of

LBP along with contrast distributions may result in higher classification accuracy than the Gabor and wavelet features, with a smaller computational overhead (Maenpaa & Pietikäinen, 2004), (Iakovidis et al., 2005).

Feature extraction is followed by clustering using the Expectation Minimization (EM) algorithm (Dempster et al., 1997). The EM algorithm is a widely-used statistical clustering method. It performs clustering by estimating the mean and standard deviation of each cluster, so as to maximize the likelihood of the observed data.

According to the scheme described in section 3.3, given a clustering of an image comprising M clusters C_i, i=1,2,..,M, a pattern $object_i$ is instantiated for each cluster C_i representing an object depicted in a cultural heritage image:

$$object_i = \begin{pmatrix} SS : (D : [[mean : [\text{Real}], \text{stdDev} : [\text{Real}]]_1^N), \\ MS : (pp : \text{Real}) \end{pmatrix}$$

where *mean* and *stdDev* are the mean and the standard deviation of the distribution D_j for every one of the N features (j=1,2,..,N) in cluster C_i, respectively, and *pp* is the prior probability of C_i. Here prior probability is defined as the fraction of the feature vectors of the image that belong to cluster C_i. Intuitively, prior probability *pp* is equivalent with the *support* measure widely used in data mining models. In our case, in addition to the qualitative aspect of the prior probability, it also provides an indication of the size of the object.

In this connection, an image is considered as a complex pattern defined by (3-13), consisting of a set of simple clusters each one of them represented by the mean and standard deviation values of a distribution.

$$image = \begin{pmatrix} SS : \{object\}, \\ MS : \perp \end{pmatrix} \quad (3\text{-}13)$$

Aiming at the definition of the similarity of two images (i.e., complex patterns), we have first to define the similarity between the measures and the structures among two clusters C_1 and C_2 (i.e., simple patterns). This similarity is expressed as the distance between two images and the components of the distance computation are analyzed below.

In the current application, a slight different approach from the one described in section 3.3 has been followed. The differences lie in the measure distance, the structure distance and the final aggregation function.

The distance between the measures of two clusters is computed using the Euclidean distance as in Eq.(3-14).

$$dis_{meas}(C_1, C_2) = \left| C_1.pp - C_2.pp \right| \tag{3-14}$$

Rephrasing the problem of defining the structural distance between C_1 and C_2 we need to find a measure for evaluating the closeness of two sets of distributions, as C_1 and C_2 are. Further decomposing the problem, we should first define a method of computing the distance between just two distributions D_1 and D_2. To achieve this, we use the standardized difference *Cohen's d* between two distributions as it has been defined by Cohen (Cohen, 1988) and is shown in Eq. (3-6).

This is a means to automate and materialize the intuitive overlap between two distributions. Having this, we let the structural distance between two sets of distributions (i.e. two clusters C_1 and C_2) be the average among the distances computed for each pair of the N features:

$$dis_{struct}(C_1, C_2) = \sum_{j=1}^{N} dis(D_j^1, D_j^2) \Big/ N \tag{3-15}$$

We aggregate the distances between the qualitative dis_{meas} and the structural dis_{struct} distances between the clusters by using the following aggregation function f_{aggr}, which gives the same weight to either of the above distances, while further weights the overall distance by the mean of prior probabilities of the clusters, as a bias towards similar and concurrently big clusters.

$$dis(C_1, C_2) = \frac{dis_{struct}(C_1, C_2) + dis_{meas}(C_1, C_2)}{2} \cdot (C_1.pp + C_2.pp) / 2 \tag{3-16}$$

Having defined the similarity between clusters (i.e., simple patterns), to compare two images I_1 and I_2 (i.e., complex patterns) we need to determine the coupling methodology between the different clusters of each image. Though various coupling types can be applied in the context of the PANDA framework (Bartolini et al., 2004), we adopt the matching of Eq.(3-17),

allowing each cluster of the first image to match more than one cluster of the second, and vice versa.

$$dis(I_1, I_2) = \frac{1}{M^2}(\sum_{i=1}^{M}\sum_{j=1}^{M} dis(C_i^{I_1}, C_j^{I_2})) \quad (3\text{-}17)$$

3.5.2 Experimental Results

The experiments aim to demonstrate the efficiency of the proposed pattern-based approach to CBIR over the approach used by conventional CBIR systems. Cultural heritage images originating from the database of the Foundation of Hellenic World (FMW, 2009) comprise the dataset used in the experiments. The images span five classes, namely ancient monuments, coins, photo portraits, drawings, and marbles. These include both color and grey level images of different sizes, inconsistently acquired from different sources. All images have been converted to 8-bit grey level format and have been downscaled to fit into a 256×256 bounding box.

Figure 3-10. *Sample images from the cultural image database used in the experiments.*

A total of 5,000 regions were sampled from the available images using 128×128-pixel windows with a 96-pixel overlap. The feature vectors extracted from each image were clustered by means of the EM algorithm implemented in the WEKA data mining tool (Witten et al., 2005). A binary clustering approach was followed, considering that the images contain one or more objects of interest of the same kind (e.g. one or two coins), and background information.

For each cluster a pattern $object_i$, i=1,2 was assigned, and each image was represented by a complex pattern *image*. The collection of patterns originating from the images registered in the database was used to build the pattern-base. Subsequent queries were executed to evaluate the performance of the proposed approach.

The performance of the proposed pattern-based approach to CBIR in comparison with the conventional, exhaustive approach is illustrated in Figure 3-11, in terms of the number of comparisons between the query and the registered data. It can be observed that the proposed approach achieves approx. 156 times less comparisons.

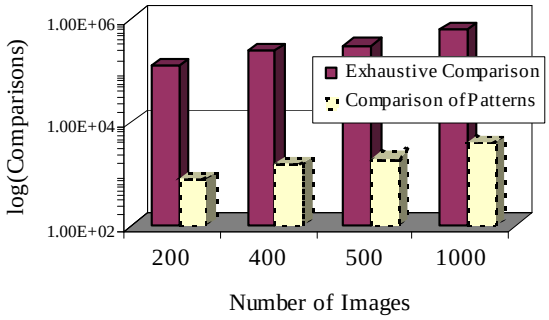


Figure 3-11 *Number of comparisons between the query and the registered data for the conventional and the proposed approaches.*

The retrieval performance of the proposed CBIR approach with the LBP and contrast distributions was estimated 80.4%. The respective performance obtained using standard 3-level Discrete Wavelet Transform (DWT) energy features was 62.2%.

3.6 Synopsis

In this chapter we described the very important process of pattern comparison focusing on clustering patterns. Pattern comparison is an advanced operation, based on the representation model of patterns and especially on the structure and measure components of patterns. The efficiency of the pattern comparison result depends not only on the effectiveness of the pattern schema (structure and measure component definition) but also on the distance, similarity and the aggregation functions that are used.

Using the PANDA framework for pattern comparison (Ntoutsi, 2008), we defined all the functions needed to compute the similarity of clustering patterns extracted from the EM clustering algorithm. Since clusters extracted with the EM algorithm are described by distributions, the *Cohen's d* (Cohen, 1988) distance function has been implemented.

We defined a methodology for the comparison of various types of data/objects (e.g. images), that includes four steps; (a) feature extraction

from raw data, (b) clustering of the extracted features, (c) Pattern Instantiation and (d) Computation of Pattern Similarities. In order to compare two objects, we have to compare the clusters (patterns) that are found in it. A high similarity value between the clusters would suggest a higher similarity for the initial objects. The similarity between two patterns of the same type is defined as the combined similarity of their structure and measure components. A *clustering* consists of a number of clusters/patterns and the similarity between two clusterings is defined by aggregating the similarities of their clusters. By finding the similarity between two clusterings, we find the similarity of the objects that those clusterings represent.

We also presented two studies of real-world image comparison cases, through pattern comparison. Both cases follow the same methodology but differ in the similarity and aggregation functions. Experimental results have shown that our methodology and the defined functions are performing well in these applications.

4 Pattern Comparison – the case of Fuzzy Clustering

In the previous chapter we dealt with the comparison of crisp clustering patterns. In this chapter we deal with the clustering of intuitionistic fuzzy data and the comparison of the extracted clusters. Fuzzy data have a membership degree for their features, while intuitionistic fuzzy data have a membership degree as well as a non-membership degree for their features. A third value, the hesitancy, is introduced to define the degree of uncertainty. In a lot of real-world applications the concept of *uncertainty* appears in various ways; data imprecision due to sampling and/or measurement errors, uncertainty in querying and answering, fuzziness by purpose during pre-processing for preserving anonymity, and so on. Intuitionistic fuzzy clustering provides an advanced technique for clustering and classifying that kind of data.

In this chapter we present the theory of intuitionistic fuzzy sets, we define a distance measure for intuitionistic fuzzy data and we present a modification of the Fuzzy C-Means (FCM) algorithm (Bezdek, et al., 1984) that incorporates this measure. We provide an experimental study of clustering images represented as intuitionistic fuzzy data (using fuzzy histograms). The clustering is used for the classification of images in predefined classes.

The PBMS concept is used here to represent the fuzzy data and the output of the clustering algorithm in the pattern-base. The output of the intuitionistic Fuzzy Clustering algorithm, stored in the pattern-base can be used for future reference, to classify new images to the classes already stored in the pattern-base. Using the proposed similarity measure for intuitionistic fuzzy data, new objects can be easily classified in the predefined classes.

4.1 Introduction

Clustering approaches based on fuzzy logic (Zadeh, 1965), such as Fuzzy C-Means (FCM) (Bezdek, et al., 1984) and its variants (Yong, 2004; Thitimajshima, 2000; Chumsamrong, et al., 2000) have proved to be competitive to conventional clustering algorithms, especially for real-world applications. The comparative advantage of these approaches is that they do not consider sharp boundaries between the clusters, thus allowing each feature vector to belong to different clusters by a certain degree (the so-called soft clustering in contrast to hard clustering produced by conventional methods). The degree of membership of a feature vector to a cluster is usually considered as a function of its distance from the cluster centroids or from other representative vectors of the cluster.

A major challenge posed by real-world clustering applications is dealing with uncertainty in the localization of the feature vectors. Considering that feature values may be subject to uncertainty due to imprecise measurements and noise, the distances that determine the membership of a feature vector to a cluster will also be subject to uncertainty. Therefore the possibility of erroneous membership assignments in the clustering process is evident. Current fuzzy clustering approaches do not utilize any information about uncertainty at the constitutional feature level.

In this chapter we introduce a modification to the FCM. The novel variant of the FCM algorithm assumes that the features are represented by intuitionistic fuzzy values, i.e. elements of an intuitionistic fuzzy set. Intuitionistic fuzzy sets, (Atanassov, 1986, 1989, 1994a, 1994b, 1999) are generalized fuzzy sets (Zadeh, 1965) that can be useful in coping with the hesitancy originating from imperfect or imprecise information (Vlachos and Sergiadis, 2006). The elements of an intuitionistic fuzzy set are characterized by two values representing their belongingness and non-belongingness to this set, respectively. In order to exploit this information for clustering we define a novel distance metric especially designed to operate on intuitionistic fuzzy vectors.

For example, in the set $A=\{x, 0.4, 0.2\}$, x is the element of the set, value 0.4 represents the membership of x to the set and the value 0.2 represents the non-membership of x to set *A*. It can be noticed that the sum of these values

(0.4+0.2) is less than one, indicating that there is a hesitancy of value 0.4 (=1-0.4-0.2) for which we do not know if the element x belongs or not to set A. In the following we will present the theory and the comparison of such sets.

The plethora and importance of the potential applications of intuitionistic fuzzy sets have drawn the attention of many researchers that have proposed various kinds of similarity measures between intuitionistic fuzzy sets. Example applications include identification of functional dependency relationships between concepts in data mining systems, approximate reasoning, pattern recognition and others. Similarity measures between intuitionistic fuzzy sets have been proposed by Chen (1995, 1997) with S_C measure, by Hong & Kim (1999) with S_H, by Fan & Zhangyan (2001) with S_L, and Li et al. (2002) who proposed the S_O measure. Dengfeng & Chuntian (2002) proposed the S_{DC} measure, Mitchell (2003) proposed a modification of the S_{DC} measure, the S_{HB} measure, Zhizhen & Pengfei (2003) proposed three measures S_e^p, S_s^p and S_h^p and three more measures have been proposed by Hung & Yang (2004), the S_{HY}^1, S_{HY}^2, and S_{HY}^3. Li et al. (2007) provide a detailed comparison of these measures, pointing out the weaknesses of each one.

Some measures, such as S_C, S_H, S_L, S_{HB} and S_{HY}^1, S_{HY}^2, and S_{HY}^3 focus on the aggregation of the differences between membership values and differences between the non-membership values while others apply distances such as Minkowski, for S_{DC}, or Hausdorff, for S_{HY}^1, S_{HY}^2, and S_{HY}^3 in order to calculate the degree of similarity of the fuzzy sets. S_{DC}, S_s^p and S_h^p focus also on the difference between membership values and non-membership values.

As regards the effectiveness of these measures, some of them, such as S_C and S_{HY}^1, S_{HY}^2, and S_{HY}^3 do not satisfy the properties of a similarity metric defined between intuitionistic fuzzy sets, whereas all of the above mentioned measures fail in specific cases that Li et al. (2007) mention with counter-intuitive examples.

4.2 Intuitionistic Fuzzy Data Clustering

In order to represent intuitionistic fuzzy data and to be able to define a proper comparison measure to be used in the PBMS, we have to provide the

theory of fuzzy and intuitionistic fuzzy sets. In the following sections an overview of the intuitionistic fuzzy set theory is presented. The similarity measures are defined and we propose an appropriate clustering scheme. Then we define the representation of the fuzzy data in the pattern base and the application that PBMS can support.

4.2.1 Intuitionistic Fuzzy Sets

The theoretical foundations of fuzzy and intuitionistic fuzzy sets are described in (Zadeh, 1965; Atanassov, 1986). This section briefly outlines the related notions used in this study.

Definition 4-1 (Zadeh, 1965). Let a set E be fixed. A fuzzy set on E is an object $\tilde{A}$ of the form

$$\tilde{A} = \left\{ \left\langle x, \mu_{\tilde{A}}(x) \right\rangle \middle| x \in E \right\}$$

where $\mu_{\tilde{A}} : E \to [0,1]$ defines the degree of membership of the element $x \in E$ to the set $\tilde{A} \subset E$. For every element $x \in E$, $0 \le \mu_{\tilde{A}}(x) \le 1$. ■

Definition 4-2 (Atanassov, 1986; Atanassov, 1994). An intuitionistic fuzzy set A is an object of the form

$$A = \left\{ \left\langle x, \mu_A(x), \gamma_A(x) \right\rangle \middle| x \in E \right\}$$

where $\mu_A : E \to [0,1]$ and $\gamma_A : E \to [0,1]$ define the degree of membership and non-membership, respectively, of the element $x \in E$ to the set $A \subset E$. For every element $x \in E$, it holds that $0 \le \mu_A(x) \le 1$, $0 \le \gamma_A(x) \le 1$ and

$$0 \le \mu_A(x) + \gamma_A(x) \le 1 \qquad (4\text{-}1)$$

For every $x \in E$, if $\gamma_A(x) = 1 - \mu_A(x)$, A represents a fuzzy set. The function

$$\pi_A(x) = 1 - \mu_A(x) - \gamma_A(x)$$

represents the degree of *hesitancy* of the element $x \in E$ to the set $A \subset E$.

■

For every two intuitionistic fuzzy sets A and B the following operations and relations are valid (Atanassov, 1986; Atanassov 1994)

$A \subset B$ iff $\forall x \in E, \mu_A(x) \le \mu_B(x)$ and $\gamma_A(x) \ge \gamma_B(x)$

$A = B$ iff $A \subset B$ and $B \subset A$

$$A^C = \left\{ \left\langle x, \gamma_A(x), \mu_A(x) \right\rangle \middle| x \in E \right\}$$

$$A \cap B = \left\{ \left\langle x, \min\left(\mu_A(x), \mu_B(x)\right), \max\left(\gamma_A(x), \gamma_B(x)\right) \right\rangle \middle| x \in E \right\}$$

$$A \cup B = \left\{ \left\langle x, \max\left(\mu_A(x), \mu_B(x)\right), \min\left(\gamma_A(x), \gamma_B(x)\right) \right\rangle \middle| x \in E \right\}$$

$$A @ B = \left\{ \left\langle x, \frac{1}{2}\left(\mu_A(x) + \mu_B(x)\right), \frac{1}{2}\left(\gamma_A(x), \gamma_B(x)\right) \right\rangle \middle| x \in E \right\}$$

$$\overset{n}{@}_{i=1} A_i = \left\{ \left\langle x, \frac{1}{n}\left(\sum_{i=1}^{n} \mu_{A_i}(x)\right), \frac{1}{n}\left(\sum_{i=1}^{n} \gamma_{A_i}(x)\right) \right\rangle \middle| x \in E \right\}$$

Definition 4-3 (Dengfeng and Chuntian, 2002). Let S be a mapping $\text{IFSs}(E) \times \text{IFSs}(E) \to [0,1]$, where IFSs($E$) denotes the set of all intutionistic fuzzy sets in E. $S(A, B)$ is said to be the degree of similarity between $A \in \text{IFSs}(E)$ and $B \in \text{IFSs}(E)$, if $S(A, B)$ satisfies the following conditions:

P1. $S(A,B) \in [0,1]$

P2. $S(A,B) = 1 \Leftrightarrow A = B$

P3. $S(A,B) = S(B,A)$

P4. $S(A,C) \le S(A,B)$ and $S(A,C) \le S(B,C)$ if $A \subseteq B \subseteq C$, $C \in \text{IFSs}(E)$ ■

Representing the data of a real-world clustering problem by means of intuitionistic fuzzy sets, is a challenging issue providing the opportunity to investigate the effectiveness of the intuitionistic fuzzy theory in practice.

4.2.2 Intuitionistic Fuzzy Data Comparison Measures

In this section we propose a novel similarity measure between intuitionistic fuzzy sets, based on the membership and non-membership values of their elements. Given an intuitionistic fuzzy set A we define two fuzzy sets, namely $M_A, \Gamma_A \in \mathcal{F}(E)$ where $\mathcal{F}(E)$ is the set of all fuzzy subsets of an element $x \in E$. The membership and non-membership of these sets is defined as $M_A = \{\mu_A(x)\}$, $\Gamma_A = \{\gamma_A(x)\}$ $\forall x \in E$. In this connection, A can be described by the pair (M_A, Γ_A).

Definition 4-5. Considering two intuitionistic fuzzy sets A=(M_A, Γ_A), B=(M_B, Γ_B), where $M_A, M_B, \Gamma_A, \Gamma_B \in \mathcal{F}(E)$, and considering E as a finite universe $E = \{x_1, x_2, ..., x_n\}$ we define the similarity measure Z_1 between the intuitionistic fuzzy sets A and B by the following equation:

$$Z_1(A, B) = \frac{z_1(M_A, M_B) + z_1(\Gamma_A, \Gamma_B)}{2} \tag{4-2}$$

Where

$$z_1(A', B') = \begin{cases} \dfrac{\sum_{i=1}^{n} \min(A'(x_i), B'(x_i))}{\sum_{i=1}^{n} \max(A'(x_i), B'(x_i))}, & A' \cup B' \neq \emptyset \\ 1, & A' \cup B' = \emptyset \end{cases} \tag{4-3}$$

with $A', B' \in \mathcal{F}(E)$.

In order to accept Z_1 as a similarity metric we need to prove that z_1 satisfies the properties defined in Definition 4-3. It is straightforward to prove that properties P1, P2 and P3 are satisfied by z_1. We supply the proof for property P4.

Lemma. For all $A', B', C' \in \mathcal{F}(E)$, where $\mathcal{F}(E)$ is the set of all fuzzy subsets of an element $x \in E$ and considering E as a finite universe $E = \{x_1, x_2, ..., x_n\}$, if $A' \subseteq B' \subseteq C'$ then $z_1(A', C') \leq z_1(A', B')$ and $z_1(A', C') \leq z_1(B', C')$.

Proof: By $A' \subseteq B' \subseteq C'$ it implies that $A'(x_i) \leq B'(x_i) \leq C'(x_i) \forall x_i \in E$ and

$$z_1(A', C') = \frac{\sum_{i=1}^{n} \min(A'(x_i), C'(x_i))}{\sum_{i=1}^{n} \max(A'(x_i), C'(x_i))} = \frac{\sum_{i=1}^{n} A'(x_i)}{\sum_{i=1}^{n} C'(x_i)}, \quad z_1(A', B') = \frac{\sum_{i=1}^{n} \min(A'(x_i), B'(x_i))}{\sum_{i=1}^{n} \max(A'(x_i), B'(x_i))} = \frac{\sum_{i=1}^{n} A'(x_i)}{\sum_{i=1}^{n} B'(x_i)},$$

$$z_1(B',C') = \frac{\sum_{i=1}^{n} \min(B'(x_i), C'(x_i))}{\sum_{i=1}^{n} \max(B'(x_i), C'(x_i))} = \frac{\sum_{i=1}^{n} B'(x_i)}{\sum_{i=1}^{n} C'(x_i)}$$. Thus, $\frac{\sum_{i=1}^{n} A'(x_i)}{\sum_{i=1}^{n} C'(x_i)} \le \frac{\sum_{i=1}^{n} A'(x_i)}{\sum_{i=1}^{n} B'(x_i)}$, $\frac{\sum_{i=1}^{n} A'(x_i)}{\sum_{i=1}^{n} C'(x_i)} \le \frac{\sum_{i=1}^{n} B'(x_i)}{\sum_{i=1}^{n} C'(x_i)}$

hence, $z_1(A',C') \le z_1(A',B')$ and $z_1(A',C') \le z_1(B',C')$

Since $A,B,C \in IFSs(E)$ and $A \subseteq B \subseteq C$ we have

$\mu_A(x) \le \mu_B(x) \le \mu_C(x)$ and $\gamma_A(x) \ge \gamma_B(x) \ge \gamma_C(x)$ $\forall x_i \in E, i = 1,2,...,n$,

therefore, $z_1(M_A, M_B)$ and $z_1(\Gamma_A, \Gamma_B)$ satisfy all properties P1-P4 and so Z_1 also satisfies these properties. Thus, Z_1 is a similarity metric. ■

To demonstrate the proposed measure a simple numeric example is given below.

Example. Assuming three sets $A,B,C \in IFSs(E)$ with A={x, 0.4, 0.2}, B={x, 0.5, 0.3}, C={x, 0.5, 0.2} we want to find whether B or C is more similar to A. Using the equations (4-2) and (4-3) we compute the similarity of B and C to set A.

$$Z_1(A,B) = \frac{\frac{0.4}{0.5} + \frac{0.2}{0.3}}{2} = 0.733 \quad , \quad Z_1(A,C) = \frac{\frac{0.4}{0.5} + \frac{0.2}{0.2}}{2} = 0.9$$

So, we conclude that C is more similar to A than B.

The proposed intuitionistic similarity measure uses the aggregation of the minimum and maximum membership values in combination with those of the non-membership values. Although it is very simple to calculate, it is sensitive to small value changes and it deals well with all the counter-intuitive cases in which other measures fail. Most of the similarity measures reviewed in Section 3.1, fail to evaluate to a valid intuitionistic value for specific cases. Some of them evaluate to 0 or 1 suggesting that the compared sets are either totally irrelevant or identical, while it is obvious that this is not true, and some other measures result in a high similarity value for obviously different sets. More specifically, in Table 4-1 we present all the counter-intuitive cases that Li, Olson and Qin (2007) have defined and the other measures fail, along with the calculation of the proposed measure for those cases.

Table 4-1 Proposed and other similarity measures with counter-intuitive cases

No	Measure	Counter-intuitive cases	Measure Values	Proposed measure value
I.	S_C , S_{DC}	$A=\{(x,0,0)\}$, $B=\{(x,0.5,0.5)\}$	$S_C(A,B)=S_{DC}(A,B)=1$	$Z_1=0$
II.	S_H, S_{HB}, S_e^p	$A=\{(x,0.3,0.3)\}$, $B=\{(x,0.4,0.4)\}$, $C=\{(x,0.3,0.4)\}$, $D=\{(x,0.4,0.3)\}$	$S_H(A,B)=S_{HB}(A,B)=S_e^p(A,B)=0.9$ $S_H(C,D)=S_{HB}(C,D)=S_e^p(C,D)=0.9$	$Z_1(A,B)=$ $Z_1(C,D)=0.75$
III.	S_H, S_{HB}, S_e^p	$A=\{(x,1,0)\}$, $B=\{(x,0,0)\}$, $C=\{(x,0.5,0.5)\}$	$S_H(A,B)=S_{HB}(A,B)=S_e^p(A,B)=0.5$ $S_H(B,C)=S_{HB}(B,C)=S_e^p(B,C)=0.5$	$Z_1(A,B)=0.5$, $Z_1(B,C)=0$
IV.	S_L and S_S^p	$A=\{(x,0.4,0.2)\}$, $B=\{(x,0.5,0.3)\}$, $C=\{(x,0.5,0.2)\}$	$S_L(A,B)=S_S^p(A,B)=0.95$ $S_L(A,C)=S_S^p(C,D)=0.95$	$Z_1(A,B)=0.73$ $Z_1(A,C)=0.9$
V.	$S_{HY}^1, S_{HY}^2, S_{HY}^3$	$A=\{(x,1,0)\}$, $B=\{(x,0,0)\}$	$S_{HY}^1(A,B)=S_{HY}^2(A,B)=S_{HY}^3(A,B)=0$	$Z_1(A,B)=0.5$
VI.	$S_{HY}^1, S_{HY}^2, S_{HY}^3$	$A=\{(x,0.3,0.3)\}$, $B=\{(x,0.4,0.4)\}$, $C=\{(x,0.3,0.4)\}$, $D=\{(x,0.4,0.3)\}$	$S_{HY}^1(A,B)=S_{HY}^1(C,D)=0.9$ $S_{HY}^2(A,B)=S_{HY}^2(C,D)=0.85$ $S_{HY}^3(A,B)=S_{HY}^3(C,D)=0.82$	$Z_1(A,B)=$ $Z_1(C,D)=0.75$
VII.	$S_{HY}^1, S_{HY}^2, S_{HY}^3$	$A=\{(x,0.4,0.2)\}$, $B=\{(x,0.5,0.3)\}$, $C=\{(x,0.5,0.2)\}$	$S_{HY}^1(A,B)=S_{HY}^1(A,C)=0.9$ $S_{HY}^2(A,B)=S_{HY}^2(A,C)=0.85$ $S_{HY}^3(A,B)=S_{HY}^3(A,C)=0.82$	$Z_1(A,B)=0.73$, $Z_1(A,C)=0.9$

In case (I) of Table 4-1 measure values $S_C(A,B)$ and $S_{DC}(A,B)$ imply that A and B are totally similar. In cases (II) and (IV) other measures result in a rather big similarity value· our measure is not that optimistic. Moreover, in case (IV) it is obvious that sets A is more similar to C than to B (A and C have the same non-membership value), something that other measures do not take into account. In (III), while B and C are totally different, measures S_H, S_{HB}, S_e^p give a similarity value of 0.5. On the contrary in (V) measures $S_{HY}^1, S_{HY}^2, S_{HY}^3$ give a similarity value of 0 even if the non-membership value of both A and B is the same, suggesting a level of similarity between the two sets. In (VI) and

(VII) measures $S_{HY}^1, S_{HY}^2, S_{HY}^3$ result in a rather high similarity value and in (VII) they do not recognize that *A* is more similar to *C* than to *B*, due to the same non-membership value of *A* and *C*.

The above indicate the intuitiveness of the proposed measure, which satisfies all the properties of a similarity metric and does not fail in cases that other measures fail. Furthermore, the proposed measure is easy to calculate and does not use exponents or other functions that significantly slow down the calculations.

4.2.3 Clustering Intuitionistic Fuzzy Data

Most clustering methods assume that each data vector belongs only to one cluster. This is rational if the feature vectors reside in compact and well-separated clusters. However, in real-world applications clusters overlap, meaning that a data vector may belong partially to more than one clusters. In such a case and in terms of fuzzy set theory (Zadeh, 1965), the degree of membership of a vector x_k to the *i*-th cluster u_{ik} is a value in the interval [0,1]. Ruspini (1969) introduced this idea which was later applied by Dunn (1973) to propose a clustering methodology based on the minimization of an objective function. In (Bezdek, et al., 1984) Bezdek introduced the Fuzzy C-Means (FCM) algorithm which uses a weighted exponent on the fuzzy memberships.

FCM is an iterative algorithm and its aim is to find cluster centroids that minimize a criterion function, which measures the quality of a fuzzy partition. A fuzzy partition is denoted by a (*c x N*)-dimensional matrix *U* of reals $u_{ik} \in [0,1], \forall 1 \le i \le c \text{ and } 1 \le k \le N$, where *c* and *N* is the number of clusters and the cardinality of the feature vectors, correspondingly. The following constraint is imposed upon u_{ik}:

$$\sum_{i=1}^{c} u_{ik} = 1, 0 < \sum_{k=1}^{N} u_{ik} < N \qquad (4\text{-}4)$$

Given this, the FCM objective function has the form:

$$J_m(U,V) = \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ik})^m d_{ik}^2$$

where V is a (p x c)-dimensional matrix storing the c centroids, p is the dimensionality of the data, d_{ik} is an A-norm measuring the distance between data vector x_k and cluster centroid v_i, and $m \in [1, \infty)$ is a weighting exponent. The parameter m controls the fuzziness of the clusters. When m approximates 1, FCM performs a hard partitioning as the k-means algorithm does, while as m converges to infinity the partitioning is as fuzzy as possible. There is no analytical methodology for the optimal choice of m.

Bezdek, Ehrlich and Full (1984) proved that if m and c are fixed parameters and $I_k, \tilde{I}_k$ are sets defined as:

$$\forall 1 \le k \le N, \begin{cases} I_k = \{ i \mid 1 \le i \le c; d_{ik} = 0 \}, \\ \tilde{I}_k = \{1,2,...,c\} \setminus I_k, \end{cases}$$

then $J_m(U,V)$ may be minimized only if:

$$\underset{\substack{1 \le i \le c \\ i \le k \le N}}{\forall} \quad u_{ik} = \begin{cases} \dfrac{(d_{ik})^{\frac{2}{1-m}}}{\sum_{j=1}^{c} (d_{jk})^{\frac{2}{1-m}}}, & I_k = \varnothing, \\ \begin{cases} 0, & i \notin I_k \\ \sum_{i \in I_k} u_{ik} = 1, & i \in I_k, \end{cases} & I_k \neq \varnothing, \end{cases} \tag{4-5}$$

and

$$\underset{1 \le i \le c}{\forall} \quad v_i = \frac{\sum_{k=1}^{N} (u_{ik})^m x_k}{\sum_{k=1}^{N} (u_{ik})^m}. \tag{4-6}$$

By iteratively updating the cluster centroids and the membership degrees for each feature vectors, FCM iteratively moves the cluster centroids to the "right" location within the data set. In detail, the algorithm that results in the optimal partition is the Picard algorithm which is described below:

Algorithm 4-1. FCM algorithm

Step 1: Determine c ($1 < c < N$), $m \in [1, \infty)$ and initialize $V^{(0)}$, $j \leftarrow 1$,

Step 2: Calculate the membership matrix $U^{(j)}$, using equation (4-5),

Step 3: Update the centroids' matrix $V^{(j)}$, using equation (4-6) and $U^{(j)}$,

Step 4: If $\left\|U^{j+1}-U^{j}\right\|_F > \varepsilon$ then $j \leftarrow j+1$ and go to Step 2.

The parameter ε makes the algorithm to converge when the improvement of the fuzzy partition over the previous iteration is below a threshold, while $\|\cdot\|_F$ denotes the Frobenious norm.

The FCM algorithm minimizes intra-cluster variance, but shares the same problems with k-means (MacQueen, 1967). It does not ensure that it converges to an optimal solution, while the identified minimum is local and the results depend on the initial choice of the centroids.

FCM tries to partition the dataset by just looking at the feature vectors and as such it ignores the fact that these vectors may be accompanied by qualitative information which may be given per feature. For example, following the idea of intuitionistic fuzzy set theory, a data point x_k is not just a p-dimensional vector $(x_{k_1},...,x_{k_p})$ of quantitative information, but instead it is a p-dimensional vector of triplets $[(x_{k_1},\mu_{k_1},\gamma_{k_1}),...,(x_{k_p},\mu_{k_p},\gamma_{k_p})]$, where for each x_{k_l} measurement there exists qualitative information which is provided via the intuitionistic membership μ_{k_l} and non-membership γ_{k_l} of the current data point to the feature l. It is evident that the FCM algorithm does not utilize intrinsically such qualitative information. In the application scenario of clustering images, a feature l may correspond to color information. Obviously, it would be of advantage if the clustering methodology could take into account the degree of membership and the degree of non-membership, regarding (for instance) how much red the image is, and how sure we are about our belief.

The main reason that FCM is unable to effectively utilize such intuitionistic vectors is that its distance function operates only on the feature vectors and not on the qualitative information which may be given per feature. In this study, we propose a different perspective by substituting the distance function with the intuitionistic fuzzy set distance metric introduced in Section 4.2.2. Using the proposed distance function the fuzzy c-means objective function takes the form:

$$J_m^{IFS}(U,V)=\sum_{i=1}^{c}\sum_{k=1}^{N}(u_{ik})^m\left|x_k-v_i\right|_{IFS} \tag{4-7}$$

The minimization of (4-7) can be achieved term by term:

$$J_m^{IFS}(U,V)=\sum_{k=1}^{N}\varphi_k(U) \tag{4-8}$$

Where

$$\underset{1\le k\le N}{\forall}\quad \varphi_k(U)=\sum_{i=1}^{c}(u_{ik})^m\left|x_k-v_i\right|_{IFS} \tag{4-9}$$

The Lagrangian of (4-9) with constraints from (4-4) is:

$$\underset{1\le k\le N}{\forall}\quad \Phi_k(U,\lambda)=\sum_{i=1}^{c}(u_{ik})^m\left|x_k-v_i\right|_{IFS}-\lambda\left(\sum_{i=1}^{c}u_{ik}-1\right) \tag{4-10}$$

where λ is the Lagrange multiplier. Setting the partial derivatives of $\Phi_k(U,\lambda)$ to zero we obtain:

$$\underset{1\le k\le N}{\forall}\quad \frac{\partial\Phi_k(U,\lambda)}{\partial\lambda}=\sum_{i=1}^{c}u_{ik}-1=0 \tag{4-11}$$

And

$$\underset{\substack{1\le z\le c\\ 1\le k\le N}}{\forall}\quad \frac{\partial\Phi_k(U,\lambda)}{\partial u_{zk}}=m(u_{zk})^{m-1}\left|x_k-v_i\right|_{IFS}-\lambda=0 \tag{4-12}$$

Solving (4-12) for u_{zk} we get:

$$u_{zk}=\left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}}\left(\left|x_k-v_z\right|_{IFS}\right)^{\frac{1}{1-m}} \tag{4-13}$$

From (4-11) and (4-13) we obtain:

$$\left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}} = \frac{1}{\sum_{j=1}^{c}\left(\left|x_k - v_j\right|_{IFS}\right)^{\frac{1}{1-m}}} \tag{4-14}$$

The combination of (4-13) and (4-14) yields:

$$\underset{\substack{1\le z\le c\\ 1\le k\le N}}{\forall} \quad u_{zk} = \frac{\left(\left|x_k - v_z\right|_{IFS}\right)^{\frac{1}{1-m}}}{\sum_{j=1}^{c}\left(\left|x_k - v_j\right|_{IFS}\right)^{\frac{1}{1-m}}} \tag{4-15}$$

Similarly with $J_m(U,V)$, $J_m^{IFS}(U,V)$ may be minimized only if:

$$\underset{\substack{1\le i\le c\\ i\le k\le N}}{\forall} \quad u_{ik} = \begin{cases} \dfrac{\left(\left|x_k - v_i\right|_{IFS}\right)^{\frac{1}{1-m}}}{\sum_{j=1}^{c}\left(\left|x_k - v_j\right|_{IFS}\right)^{\frac{1}{1-m}}}, & I_k = \varnothing, \\ \begin{cases} 0, & i \notin I_k \\ \sum_{i\in I_k} u_{ik} = 1, & i \in I_k, \end{cases} & I_k \neq \varnothing, \end{cases} \tag{4-16}$$

while the centroids are computed by (4-6).

It should be clarified that u_{ik} corresponds to the membership of the *k*-th intuitionistic fuzzy vector to the *i*-th cluster and has nothing to do with the internal intuitionistic fuzzy memberships of the vector. Furthermore, as our distance function between two vectors is computed solely upon the intuitionistic fuzzy memberships and non-memberships of the vectors, after the computation of the centroids by equation (4-6) and before the next iteration, where the u_{ik} memberships to the new clusters are updated, there is a need for an additional step which estimates the intuitionistic fuzzy memberships and non-memberships of the new (virtual) centroids. In other words, it is necessary to deduce the membership μ_{i_l} and non-membership γ_{i_l} values of each feature *l* that corresponds to the *l*-th dimension of the *i*-th centroid. At each iteration and for every centroid we extract the membership degree μ_{i_l} of centroid v_i as the average of the membership degrees of all the intuitionistic fuzzy vectors that belong to cluster *i*. Similarly, we extract the non-membership degrees γ_{i_l}. More formally, if P_i is a set defined as:

$$\underset{1\le i\le c}{\forall} \quad P_i = \left\{ k \,\middle|\, 1 \le k \le N;\, d_{ik} < d_{rk}, \forall 1 \le r \le N \wedge r \ne i \right\} \tag{4-17}$$

then the intuitionistic fuzzy set IFS_{v_i} for centroid v_i is defined as:

$$\underset{1\le i\le c}{\forall} \quad IFS_{v_i} = \underset{\forall k \in P_i}{@} IFS_k \tag{4-18}$$

From (Atanassov, 1994) we obtain:

$$\underset{1\le l\le p}{\forall} \quad \mu_{i_l} = \frac{\sum_{\forall k \in P_i} \mu_{k_l}}{|P_i|}, \quad \nu_{i_l} = \frac{\sum_{\forall k \in P_i} \gamma_{k_l}}{|P_i|} \tag{4-19}$$

Given the above discussion, the modified FCM algorithm that clusters intuitionistic fuzzy data is subsequently described:

Algorithm 4-2. Intuitionistic Fuzzy C-Means (IFCM) algorithm for clustering intuitionistic fuzzy data

Step 1 Determine c ($1 < c < N$), $m \in [1, \infty)$ and initialize $V^{(0)}$ by selecting c random intuitionistic fuzzy vectors, $j \leftarrow 1$,

Step 2: Calculate the membership matrix $U^{(j)}$, using (4-16),

Step 3: Update the centroids' matrix $V^{(j)}$, using (4-6) and $U^{(j)}$, and compute membership and non-membership degrees of $V^{(j)}$ using (4-19)

Step 4: If $\left\| U^{j+1} - U^{j} \right\|_F > \varepsilon$ then $j \leftarrow j+1$ and go to Step 2.

In comparison to the literal FCM algorithm the clustering scheme presented in Algorithm 4-1, introduces (a) a different initialization tactic of the V matrix as in our case centroid vectors are intuitionistic fuzzy vectors (step 1), (b) a new way of the calculation of the membership degrees of a vector to a cluster, taking into account both membership and non-membership values of the intuitionistic fuzzy vectors (step 2) and (c) a method to update the V matrix at each iteration based solely on the theory of the intuitionistic fuzzy sets (step 3).

Time complexity of FCM is $O(n\ f\ c^2\ i)$ where n is the number of data points, f is the number of dimensions, c the number of clusters and i the number of iterations (Hore, Hall, and Goldgof, 2007). The proposed algorithm complexity does not differ from that of the FCM algorithm as there are no new steps in the procedure and the existing steps do not add any complexity.

4.2.4 Representing Fuzzy Clusters in the Pattern-base

The intuitionistic fuzzy data can be represented in the pattern-base as simple patterns, while the output of the iFCM algorithm can be stored as complex pattern as it has been described in chapter 3. An image object can be represented as a complex pattern:

$$image = \begin{pmatrix} SS : \{object\}, \\ MS : \perp \end{pmatrix}$$

Regarding the representation of the cluster centroids, we have to define the structure and measure components, in order to be able to perform the comparison between the different objects. The fuzzified (image in this case) data, are represented by four components. The data value and the intuitionistic fuzzy values, membership, non-membership and hesitancy. The last component, hesitancy, does not need to be specified as it can be calculated using the membership and non-membership values, according to the intuitionistic fuzzy sets theory. The structure component for every object will be only represented by the data value vector, X, while the measure component will include the intuitionistic fuzzy value vectors M for membership values and Γ for non-membership values.

Thus,

$$object_i = \begin{pmatrix} SS : (X : x : [\text{Real}]), \\ MS : [M : [\text{Real}], [\Gamma : [\text{Real}]] \end{pmatrix}$$

and

$$Cluster_i = \begin{pmatrix} SS : (X : x : [\text{Real}]), \\ MS : [M : [\text{Real}], [\Gamma : [\text{Real}]] \end{pmatrix}$$

Supporting the intuitionistic fuzzy clustering with the PBMS concept a lot of useful applications are enabled. In Figure 4-1 we present the methodology of the application that classify images into classes.

The extracted features from the available images are fuzzified and the iFCM clustering algorithm is used to classify them into classes. The clustering results represent the cluster centroids and they are stored in the pattern base, using the PBMS representation as it is shown below. The proposed scheme can be also used to classify images that do not already exist in the image-base. In this case the features from the new image are extracted and the same fuzzification method is applied. The output will be then compared to the cluster centroids that are stored in the pattern-base, using the similarity measure that has been defined in section 4.2.2. The new image will be classified to the cluster with the centroid that is more similar to the image using the similarity measure Z_1 as defined in 4.2.2.

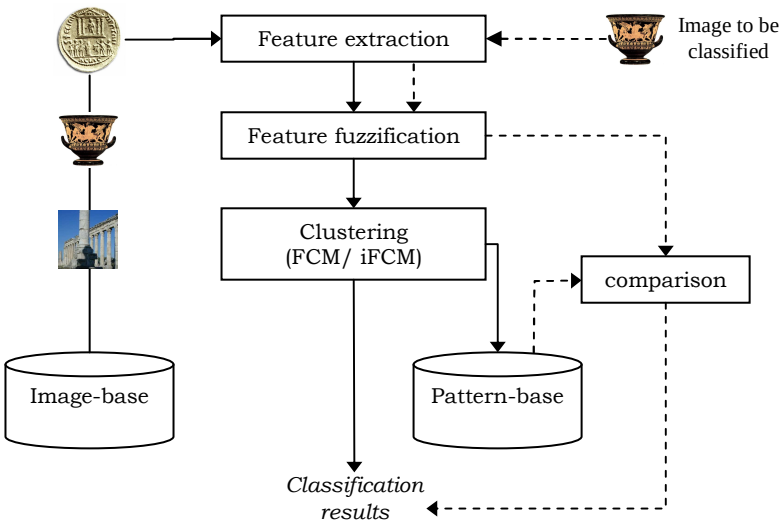


Figure 4-1 Classification of images using intuitionistic fuzzy clustering and the Pattern-base

The similarity measure Z_1 between the intuitionistic fuzzy sets A and B is defined by the following equation:

$$Z_1(A,B)=\frac{z_1(M_A,M_B)+z_1(\Gamma_A,\Gamma_B)}{2}$$

where

$$z_1(A',B')=\begin{cases}\dfrac{\sum_{i=1}^{n}\min(A'(x_i),B'(x_i))}{\sum_{i=1}^{n}\max(A'(x_i),B'(x_i))}, & A'\cup B'\neq \emptyset \\ 1, & A'\cup B'=\emptyset\end{cases}$$

with $A', B' \in \mathcal{F}(E)$.

The similarity measure is incorporated in the PANDA framework and in that case it is applied between the membership and non-membership vectors of every object.

4.3 Application: Image Classification Using Intuitionistic Fuzzy Clustering

In this section we present an experimental study of clustering intuitionistic fuzzy data. We define a proper intuitionistic fuzzy representation of images and use the proposed similarity measure to cluster images. In the available dataset there is a specific number of classes and thus, we use the clustering as a classification method to evaluate the results.

This study aims at evaluating the proposed similarity measure as long as the iFCM algorithm in general. This application does not include the classification of new images using the classification results, as this methodology has been already presented in sections 3.4 and 3.5.

4.3.1 Intuitionistic fuzzy representation of data

The proposed intuitionistic fuzzy clustering requires that each data element x of a universe E, belongs to an intuitionistic fuzzy set $A \subset E$ by a degree $\mu_A(x)$ and does not belong to A by a degree $\gamma_A(x)$. The data elements can be of any kind. For the purposes of this study, which focuses to the clustering of image data we extend the definition of the intuitionistic fuzzy representation of a grayscale digital image (Vlachos and Sergiadis, 2005), for the representation of a color digital image.

Definition 4-6. A color digital image P of $a \times b$ pixels size, composed of ξ channels P_k, k=1,2,..., ξ, digitized in q quantization levels per channel, is represented as the intuitionistic fuzzy set

$$\Phi = \left\{ \left\langle \theta_{ij}^k, \mu_\Phi(\theta_{ij}^k), \gamma_\Phi(\theta_{ij}^k) \right\rangle_k \middle| \theta_{ij}^k \in P_k,\ i = 1,2,...,a,\ j = 1,2,...,b,\ k = 1,2,...,\xi \right\} \quad (4\text{-}20)$$

where θ_{ij}^k is the value of P_k at the position (i, j), and $\mu_\Phi(\theta_{ij}^k)$ and $\gamma_\Phi(\theta_{ij}^k)$ define the membership and the non-membership of θ_{ij}^k to P_k, respectively.

As a membership function $\mu_\Phi(\theta)$, we consider the probability of occurrence of $\theta \in [0, q-1]$ in an image channel

$$\mu_\Phi(\theta) = \frac{h(\theta)}{a \cdot b}, \quad \forall \theta \in [0, q-1] \tag{4-21}$$

where

$$h(\theta) = \left\| \left\{ (i, j) \in P_k \,|\, \theta_{ij}^k = \theta;\ i = 1, \ldots, a;\ j = 1, \ldots, b,\ k = 1, 2, \ldots, \xi \right\} \right\|, \quad \blacksquare$$

is the crisp histogram of the pixel values in the channel, and $\|\cdot\|$ represents the cardinality of the enclosed set. The probability distribution described by Eq. (4-21) comprises a first-order statistical representation of the image channel that is easy to compute, and it is invariant to the rotation and translation.

Considering that real-world digital images usually contain noise of various origins, and imprecision in the channel values, the degree of belongingness of an intensity value θ in an image channel as expressed by $\mu_\Phi(\theta)$ is subject to uncertainty. In order to model this situation, we introduce a penalty factor $p(\theta)$ to $\mu_\Phi(\theta)$ so that θ belongs less to the image channel if $h(\theta)$ diverges more from the fuzzy histogram $\tilde{h}(\theta)$. The fuzzy histogram, originally proposed by Jawahar and Ray (1996), is defined as

$$\tilde{h}(\theta) = \left\| \left\{ (i, j) \in P_k \,|\, \mu_{\tilde{\theta}}(\theta_{ij}^k);\ i = 1, \ldots, a,\ j = 1, \ldots, b,\ k = 1, 2, .., \xi \right\} \right\| \tag{4-22}$$

With

$$\mu_{\tilde{\theta}}(x) = \max\left(0, 1 - \frac{|x - \theta|}{\psi}\right) \tag{4-23}$$

where parameter ψ controls the span of the fuzzy number $\tilde{\theta}: R \to [0,1]$ representing a fuzzy intensity level θ. This means that a pixel of a given channel value will contribute not only to its specific bin, but also to the bin count of the neighbouring bins in the histogram. Thus, the fuzzy histogram becomes smoother and more insensitive to noise than the corresponding crisp histogram as ψ increases.

According to the proposed formulation the non-membership of θ to an image channel can be expressed as

$$\gamma_{\Phi}(\theta) = 1 - \mu_{\Phi}(\theta) \cdot p(\theta) \tag{4-24}$$

The penalty factor *p(θ)* is chosen to be proportional to the distance between the crisp *h(θ)* and the fuzzy histogram $\tilde{h}(\theta)$, so that Eq. (4-1) is satisfied

$$p(\theta) = \lambda \cdot \frac{\left| h(\theta) - \tilde{h}(\theta) \right|}{\max_{\theta} \left(\left| h(\theta) - \tilde{h}(\theta) \right| \right)} \tag{4-25}$$

where $\lambda \in [0,1]$ is constant and the denominator facilitates normalization purposes. The physical meaning of this non-membership function is that the non-belongingness of an intensity value θ to an image channel increases by a factor that is proportional to the coarseness of the crisp histogram. So, as the noise levels in the image channel increase, the crisp histogram becomes coarser and the hesitancy in the determination of the intensity value θ increases.

The membership and the non-membership defined by equations (4-21) and (4-24) over the values of the image channels, will be considered to form feature vectors.

As shown in section 4.2.4, the cluster representation is (cluster centroid):

$$Cluster_i = \begin{pmatrix} SS : (X : x : [\text{Real}]), \\ MS : [M : [\text{Real}], [\Gamma : [\text{Real}]] \end{pmatrix}$$

while every image is a complex pattern of objects – multi-dimensional vectors of the image fuzzified characteristics represented as:

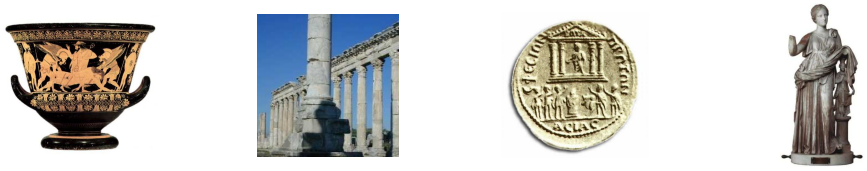
$$object_i = \begin{pmatrix} SS : (X : x : [\text{Real}])^{D}, \\ MS : [M : [\text{Real}], [\Gamma : [\text{Real}]]^{D} \end{pmatrix}$$

Every fuzzified image and the clusters centroids are stored in the pattern-base to facilitate future classification.

4.3.2 Experimental Results

Comprehensive experiments have been conducted for the evaluation of the performance of the proposed clustering algorithm, in comparison with the well established FCM. The application scenario for the experimental evaluation involves clustering of a 400 image collection spanning four

equally distributed classes of different color themes including amphorae, ancient monuments, coins, and statues (Figure 4-2).



Class A **Class B** **Class C** **Class D**

(a) (b) (c) (d)

Figure 4-2 Example images from the four classes used in the experiments, (a) amphorae, (b) ancient monuments, (c) coins, and (d) statues.

The images have been provided by the Foundation of Hellenic World, which maintains a publicly available repository of texts, images and multimedia data collections of Greek historical items and art (FHW). They are of different sizes and have been inconsistently acquired from different sources, and they have been digitized in 256 quantization levels per *RGB* channel and have been downscaled to fit into a 256×256 bounding box.

The methodology followed for the experiments is depicted in Figure 4-1.

Based on the observation that color is a discriminative feature for most of the available image classes, each image was represented by an intuitionistic fuzzy set according to (4-20), using only chromatic information so as to be approximately independent from intensity variations. In order to decorrelate the intensity from the chromatic image components, the images have been transformed to the $I_1I_2I_3$ color space according to the following equation (Ohta et al., 1980)

$$\begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix} = \begin{pmatrix} 0.333 & 0.333 & 0.333 \\ 0.500 & 0.000 & -0.500 \\ -0.500 & 1.000 & -0.500 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} \tag{4.1}$$

In this color space, the I_1 component explains the highest proportion of the total variance and represents intensity, whereas I_2 and I_3 correspond to the second and the third highest proportion respectively and carry chromatic information. A very useful property of this space is that image regions of different colors can be easily discriminated by simple thresholding operations. In other words, the histograms produced by the values of its

color components exhibit peaks corresponding to regions of different colors in the image.

Among the chromatic components of $I_1I_2I_3$ we selected I_2 as the most discriminating for the color regions comprising the available images. This is in agreement with (Ohta et al., 1980) which suggests that the discrimination power of I_2 could only marginally increase with the contribution of I_3. Moreover, we observed that the image channel corresponding to the I_3 component exhibit a low dynamic range of values, having a single-peak histogram that varies slightly between images belonging to different classes.

Examples of membership and non-membership functions used for the intuitionistic fuzzy representation of color images are illustrated in Figure 4-3. The values of the parameters used in Equations (4-23) - (4-25) for the estimation of the membership and of the non-membership functions are λ = 1 and ψ = 5.

The horizontal axes represent the values of I_2 normalized within the range [0, 255], whereas the vertical axes have been rescaled in order to improve the visibility of the graphs. The graphs focus on the regions of the membership and non-memberhip functions for which the variance is higher. The lines that intersect the frame of the graphs extending beyond the visible area join to peak membership and non-membership values.

In Figures 4-4a, 4-4c and 4-4d, the highest of the two peaks correspond to the white background regions of the images, whereas the lower peaks correspond to the depicted objects. Similarly, in Figure 4-3b the highest peak corresponds to the marble of the ancient monument and the lower peaks correspond to the sky region. As regards the non-membership functions, an intuitive interpretation could be given by considering their correlation with the corresponding membership functions. The correlation is usually less around the peaks that correspond to less homogenous image regions. For example in Figure 4-3b, the absolute correlation between the membership and the non-membership function estimated for the region of the ancient monument is 70%, whereas for the region of sky is 82%. Similarly, the absolute correlation between the membership and the non-membership functions in Figures 4-4a, 4-4b and 4-4c, for the homogenous white background regions reaches 96.5%.

Clustering experiments were conducted with all possible class combinations, using a) the proposed clustering algorithm with the intuitionistic fuzzy data, b) FCM with crisp I_2-histogram data, and c) FCM with fuzzy I_2-histogram data. In all the experiments, the same parameters (ε=0.00001, m=2.0) and initialization conditions were used. The clustering performance was evaluated in terms of classification accuracy, algorithm iterations and absolute execution time. Classification accuracy was computed as in the original FCM algorithm, by assigning an image to the cluster with the higher degree of membership value.

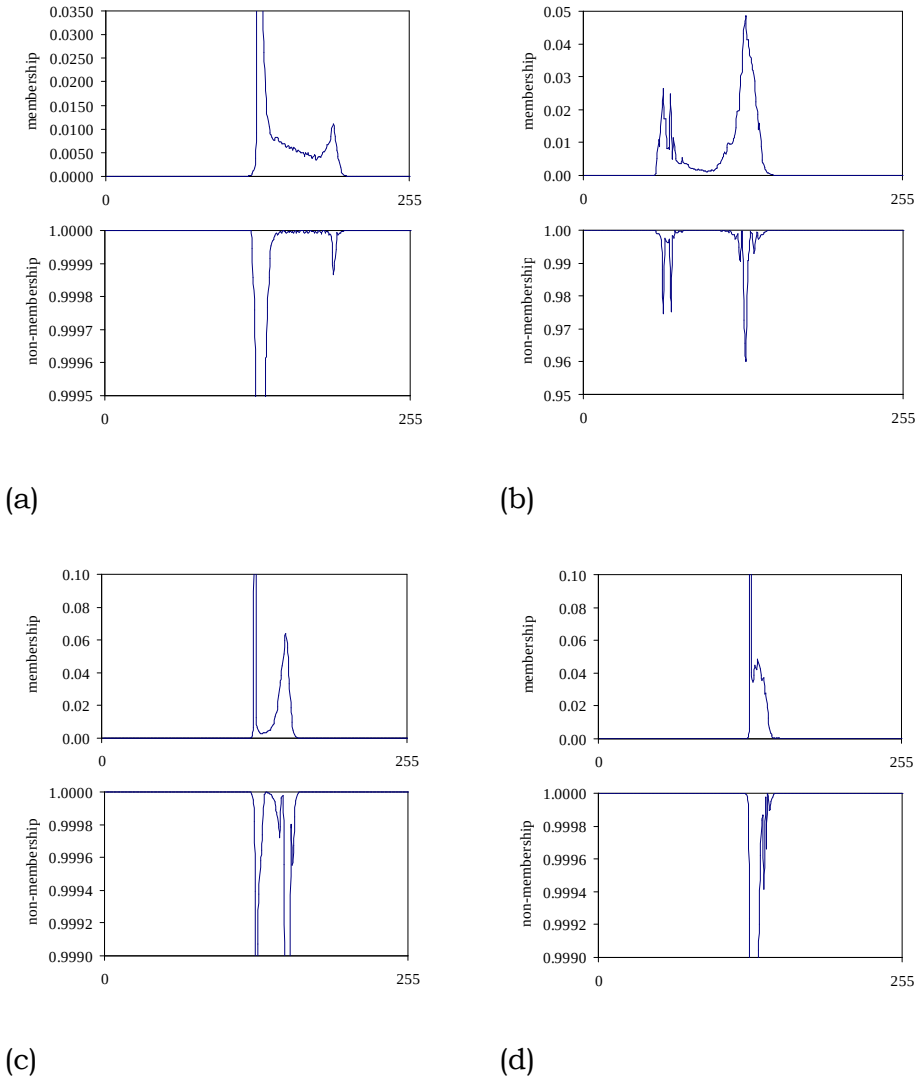
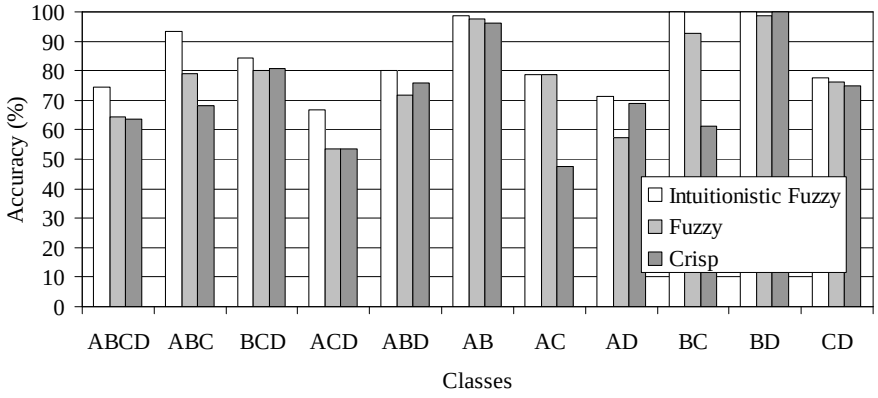
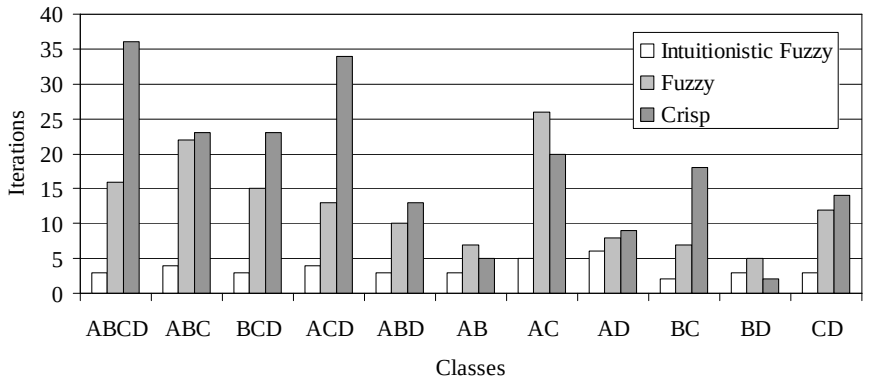


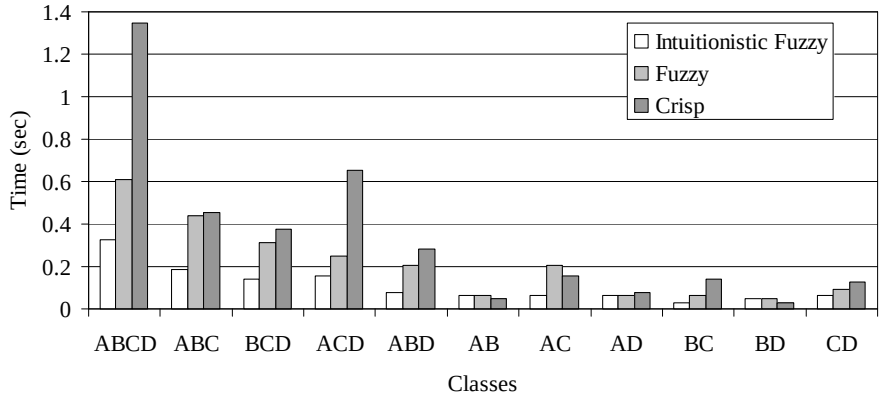
Figure 4-3 Membership and non-membership functions corresponding to the images of Figure 4-2.



(a)



(b)



(c)

Figure 4-4 Comparative results of using the proposed clustering algorithm with the intuitionistic fuzzy data, and of using the FCM with the crisp and with the fuzzy data as input: (a) classification accuracy, (b) number of iterations required for the clustering algorithms to converge, and (c) execution time required in seconds.

The experiments were executed on a PC with Intel Pentium M at 1.86 GHz, 512 MB RAM and 60 GB hard disk. The results are summarized in Figure 4-4.

Figure 4-4 shows that in all the experiments the accuracy achieved by the proposed algorithm was higher than the accuracy obtained by FCM for four or three classes. The maximum accuracies achieved with the proposed algorithm are 74.4% and 93.3% for four and for three classes respectively.

These percentages reduce to 64.4% and 79.2%, in the case of FCM clustering with fuzzy data. The results of the clustering experiments performed with data from two classes show that the accuracy of the proposed algorithm can be considered comparable with or higher than, the accuracy obtained by FCM. However, this could be attributed to a smaller contribution of the non-membership values to the representation of the images of the particular classes. The maximum accuracy obtained by both algorithms reached 100% in two cases (BC and BD).

Comparing the two algorithms in terms of efficiency, Figure 4-4b and Figure 4-4c show that the proposed algorithm has a considerable advantage over FCM, as it requires less algorithm iterations and in most cases less time to reach convergence. The average improvement in absolute execution time is 63±27%.

4.4 Synopsis

In this chapter, in contrast to chapter 3, we focused on an Intuitionistic Fuzzy Clustering scheme, that can be also supported by the PBMS concept using the proper representation. An application similar to the one presented in chapter 3 is described to classify images in predefined classes. The definition of a novel similarity measure and of a new Intuitionistic Fuzzy Clustering algorithm is presented in detail.

Clustering approaches organize a set of objects into groups whose members are proximate according to some similarity function defined on low-level features, assuming that their values are not subject to any kind of uncertainty. Furthermore, these methods assume that similarity is measured by accounting only the degree in which two entities are related, ignoring the hesitancy introduced by the degree in which they are unrelated.

Challenged by real-world clustering problems we proposed a novel fuzzy clustering scheme of datasets produced in the context of intuitionistic fuzzy set theory. More specifically, we introduced a novel variant of the Fuzzy C-Means (FCM) clustering algorithm that copes with uncertainty in the localization of feature vectors due to imprecise measurements and noise and a novel similarity measure between intuitionistic fuzzy sets, which is appropriately integrated in the clustering algorithm. We also introduced an intuitionistic fuzzy representation of color digital images as a paradigm of intuitionistic fuzzification of data.

To evaluate our approach, we described an intuitionistic fuzzification of color digital images upon which we applied the proposed scheme. The experimental evaluation of the proposed scheme shows that it can be more efficient and more effective than the well established FCM algorithm, especially as the number of clusters increases, opening perspectives for various applications.

The whole intuitionistic fuzzy clustering process is supported by the PBMS concept, enabling thus advanced classification applications.

5 Other Applications of the Pattern Base Management System

In this chapter we deal with real world problems, such as the content-based image retrieval (CBIR) problem, or the classification of astronomy data, galaxy spectrum in particular.

We have already presented applications of the PBMS concept for CBIR, thus we only summarize the approach. In the case of classifying galaxy spectrum data, we present a real world problem that we dealt in the content of our collaboration with the department of Astrophysics of the University of Athens.

5.1 Introduction

As it has been shown in sections 3.4 and 3.5, the PBMS can be used as a very functional component of a CBIR, which utilizes clustering techniques to find similar images. Figure 5-1 shows the approach proposed in section 3.5. The red dashed rectangle bounds the part of the system that can be replaced by the PBMS.

As it is shown in Figure 5-1 the PBMS replaces the core of the CBIR. Every image is represented by patterns, and thus the comparison of the patterns, reflects the comparison of images. Patterns are stored in XML documents and their comparison is easier and faster.

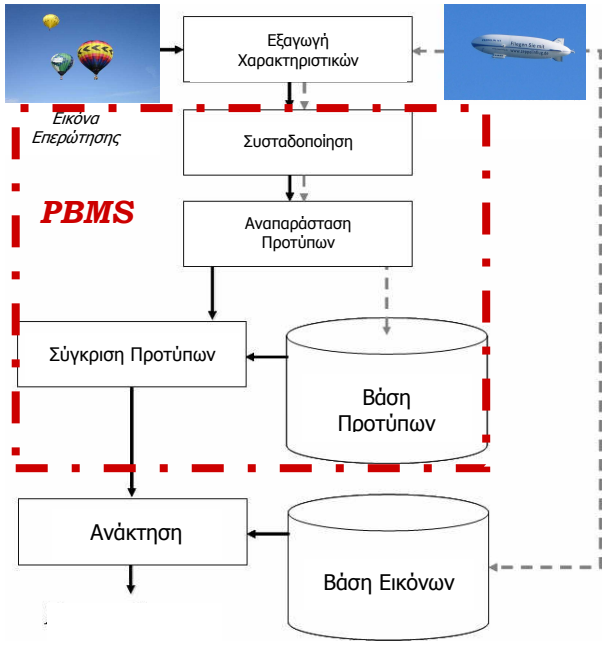


Figure 5-1 *Outline of the pattern-based CBIR approach and the part that is replaced by the PBMS.*

5.2 An application of PBMS for Categorizing Astronomical data

In the content of our collaboration with the department of Astrophysics of the University of Athens we came across the problem of finding the best model to categorize astronomical data. This task is part of the work package «Unresolved galaxy classifier» of the GAIA (2009) project of the ESA. The aim of this package is to "study, develop and test algorithms which provide optimal estimates for unresolved galaxies (classification of the galaxy spectrum), based on the assumption that the object is restricted to this class" (GAIA, 2009).

Using synthetic as well as real data (galaxy spectrums), different classification models and algorithms has to be tested. The best model will finally be used in the a system that will collect (galaxy) data from the GAIA space-telescope and will automatically classify every observed galaxy in pre-defined classes (the galaxy morphological type), saving a lot of time from the experts.

In order to find the best classification algorithm i.e. the algorithm and the parameters that give the more accurate classification, given a real and a synthetic dataset for training and testing, domain experts have to run a large

number of experiments with various algorithms. The output of these experiments has to be evaluated and the best classification model will be defined.

Three different algorithms have been used; the J.48 (a variation of the C4.5 for WEKA) and the Naïve Bayes classification algorithm from the WEKA data mining tool and the Support Vector Machine model from the R tool (R-project, 2009). J4.8 has been chosen as a variation of the very popular and successful decision tree C4.5 algorithm, while Naïve bayes is an also popular and successful classification algorithm that assumes attribute independence. SVM models (Cristianini, 2000) has been used by the astronomers to conduct the classification experiments.

Without using the PBMS concept, the experts had to manually extract and store the output of the algorithms to the file system. The comparison of the output results is in this case a manual task, with the expert browsing through the file system to find the files containing the output results, in order to compare them. The whole task requires a lot of organization effort from the users/experts and is very time consuming and confusing. They have to run the experiments, store the output into the file system, marking in a separate file the parameters used for every run, and other metadata such as the dataset used or the date and time of the algorithm execution. After the classification, experts have to manually evaluate each output and compare it with every other, using the files stored in the file system, a very time consuming and laborious task.

Using the PBMS this task is simplified. The user can run the classification algorithm of the data mining tool, and the output along with the required metadata will be stored into the pattern-base with a user-defined, easy to remember name. At any time the user can retrieve a specific classification output by posing the right query statement to see the accuracy of the pattern/model or its metadata.

Some more complicated but very useful queries can be used, such as:

- Retrieve the algorithm and the run parameters that gave the best classification accuracy using the dataset “A”.
- Retrieve the dataset that has the worst accuracy when the naïve bayes algorithm is used for the classification.

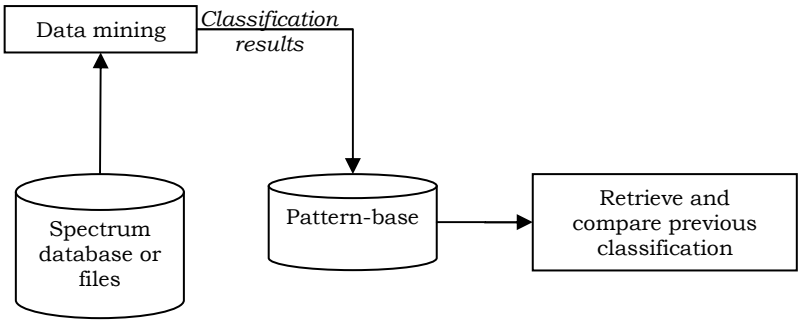


Figure 5-2 Use of the PBMS concept to run multiple classification experiments

Figure 5-2 depicts the way that multiple classification experiments can be conducted using a pattern-base to store each time the classification output and retrieving it in order to compare it with newer classification experiments. All the steps of the above process can be performed using the PBMS.

In order to use the PBMS for these experiments, an XML model has to be defined that describes the output of the classification algorithm. This is required if the PBMS does not already supports the specific algorithms and of course will be available for other future applications.

The decision trees produced as the output of the classification algorithm are represented using the model presented in section 2.3:

aPath =
(SS : [(ValueFrom: Real, ValueTo: Real)]$_1^N$,
MS: sup: Real)

aDecisionTree =
(SS : {Path},
MS: $\perp$)

The results of the three different classification algorithms – stored in the pattern based can be easily compared through the PBMS. The experts can then decide which model they will use. In Figure 5-3 a part of the classification tree is shown. The classification attribute is the galaxy morphological type.

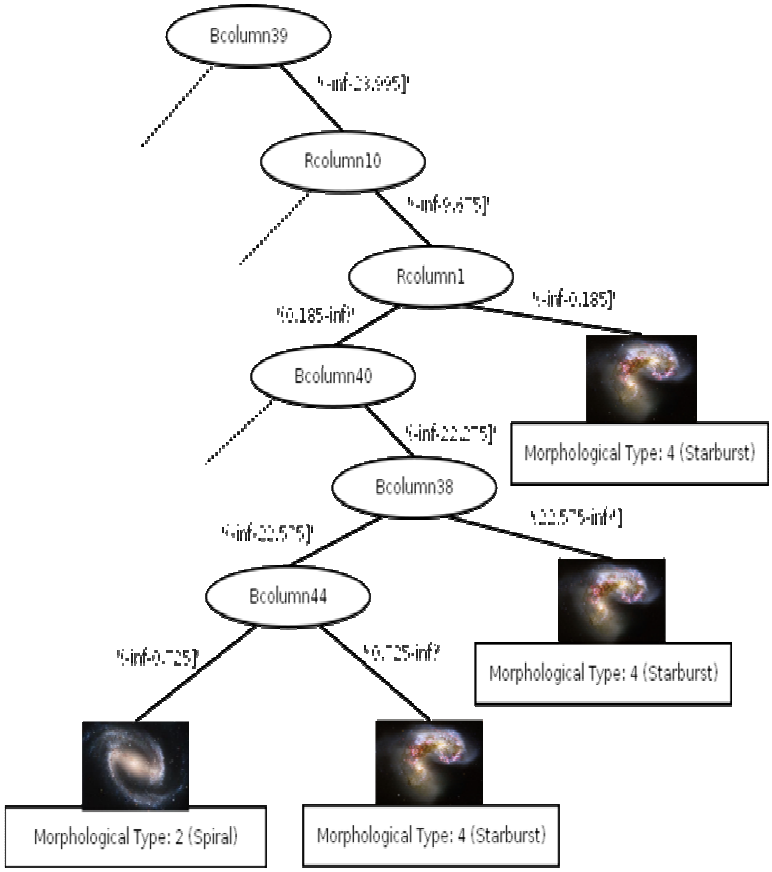


Figure 5-3 A part of the classification tree built from the J4.8 algorithm, showing the B (Blue spectrum area) and R (Red spectrum area) columns and the different classes depending on the values of the spectrum.

The four morphological types that galaxies can be categorized in the current study are: Early, Spiral, Irregular and Starburst. Sample images of these types of galaxies are shown below.

Figure 5-4 Early type galaxy

Figure 5-5 Spiral galaxy

Figure 5-6 Irregular type galaxy

Figure 5-7 Starburst type galaxy

We ran experiments using the J4.8 and the Naïve Bayes algorithm of WEKA and compared the results with those of the best SVM model from the R tool using the PBMS concept of storing every output in a common pattern-base.

Before proceeding to the experiments and the evaluation of the classification algorithms, the spectrum data had to be discretized in bins of equal width and equal frequency. The number of bins is subject for experiment and had the value of 2 to 10.

Table 5-1 The various Classification Experimentation cases

Number of bins	
2 to 10	
Discretization method	
equal width	equal frequency
Classification algorithm	
J48	Naive Bayes

All the different experimentation cases are described in Table 5-1. In every case the following parameters are taken into account:

1. The number of the data discretization bins.
2. The discretization method (equal width or equal frequency).
3. The classification algorithm (Naive Bayes, J48).

The figures below present the classification accuracy results for the two algorithms for 2 to 10 bins. More specifically, Figure 5-8 presents the classification for data discretized using equal frequency bins, whie Figure 5-9 presents the classification for data discretized using equal width bins.

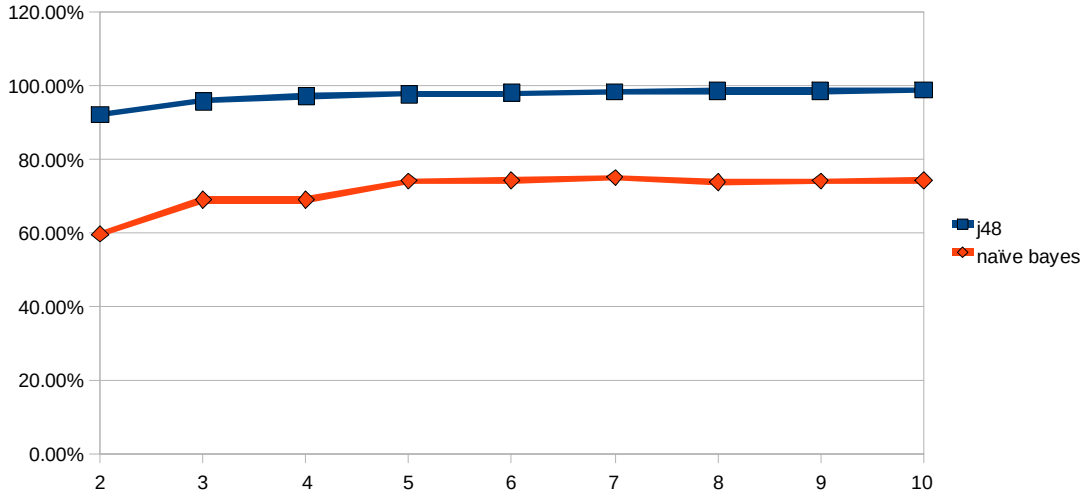


Figure 5-8 Classification results for J4.8 and Naive Bayes algorithms using equal frequency discretization bins

J4.8 performs always better with a classification accuracy of at least 95% and an average 97.25%, while naïve bayes only reaches at a maximum of 75.01%. In both cases though, the maximum accuracy is succeeded in the at the 10 bins discretization.

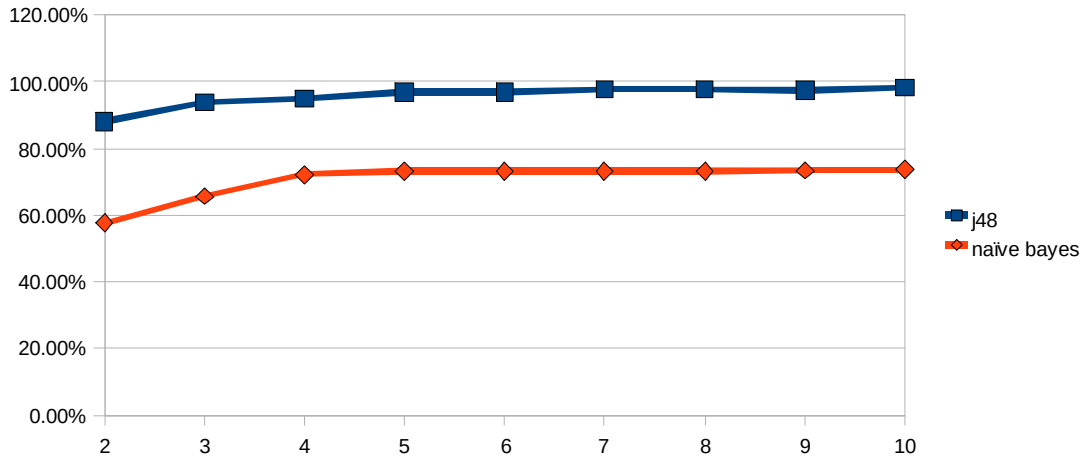


Figure 5-9 Classification results for J4.8 and Naive Bayes algorithms using equal width discretization bins

In the equal width discretization method experiment, the J4.8 also performs better with an average of 95.78% in contrast with the 70.68% of the naïve bayes. Maximum classification accuracy for J4.8 is in that case 97.79% and for the Naïve Bayes is 73.66%.

Comparing the classification accuracy for each algorithm separately and for both equal frequency and width discretization methods, we conclude that the equal frequency discretization method gives better results for both algorithms as shown in the figures below.

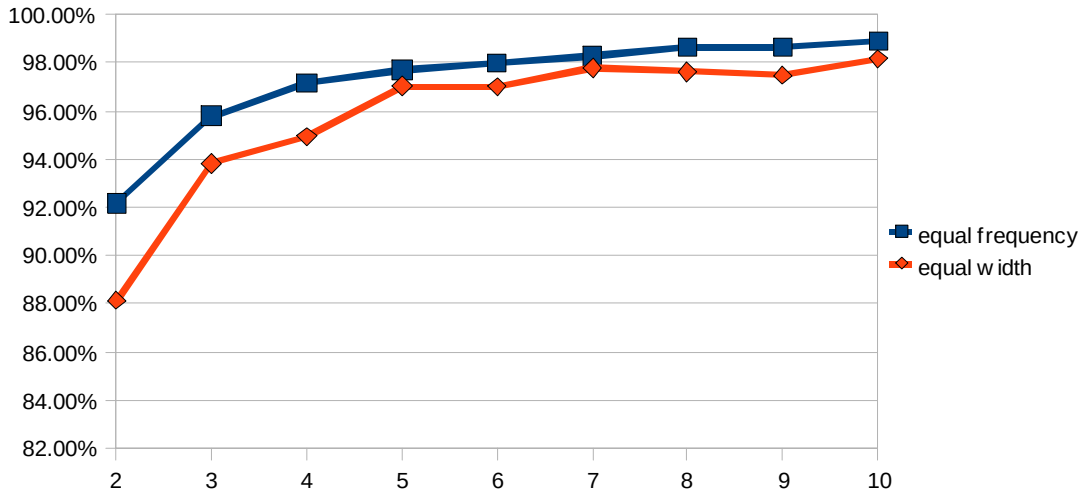


Figure 5-10 Classification results for J4.8 comparing equal frequency and width discretization methods

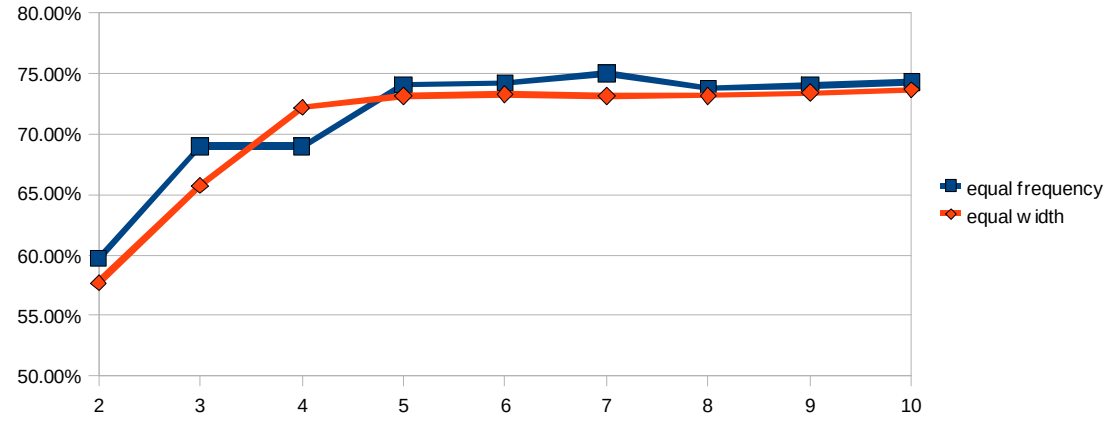


Figure 5-11 Classification results for Naïve Bayes comparing equal frequency and width discretization methods

Furthermore, we conducted experiments for the recall of both algorithms and for all four galaxy morphological types. Recall is the ratio that expresses the number of correctly classified galaxies of one class (type) to the total number of galaxies belonging to this class. In Figure 5-12 the recall for all the morphological types is presented, in the case of the J4.8 algorithm and the equal frequency discretization method. For the Spiral and Starburst galaxies the recall is very high for all number of bins. For the Irregular

galaxy type, an average of 90% recall is succeeded while in the early galaxy type, the recall percentage is low for two and three bins but it raises in the cases of six or more bins reaching 92%.

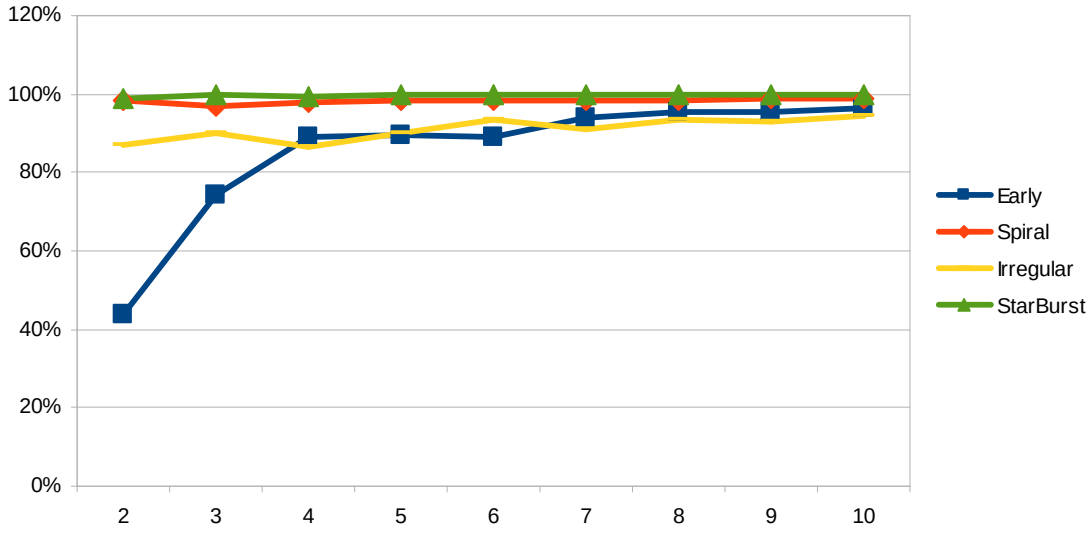


Figure 5-12 Recall ratio of all morphological types for the J4.8 algorithm and equal frequency discretization method.

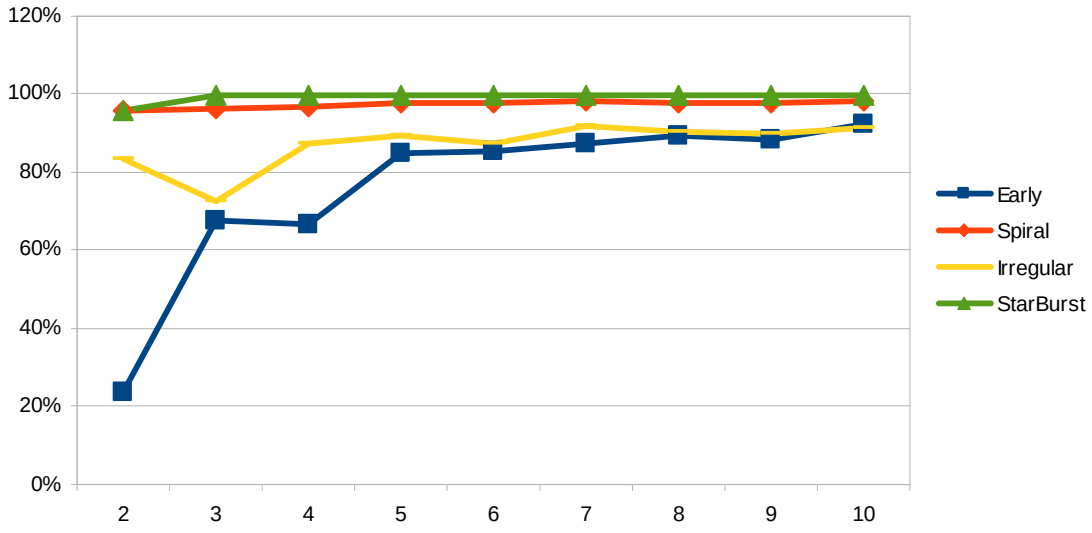


Figure 5-13 Recall ratio of all morphological types for the J4.8 algorithm and equal width discretization method.

Figure 5-13 presents the recall percentages for all the morphological types is presented, in the case of the J4.8 algorithm and the equal width discretization method. Like in the equal frequency case, J4.8 managed to classify with great success Spiral and Starbursts galaxies. The recall for the

Irregular type varies from 72.6%, for three bins to 91.7 % for seven bins. Regarding the early type recall is even lower from the previous case (of equal frequency) but it raises for five or more bins.

Respectively, Figure 5-14 and Figure 5-15 present recall percentages for all four types of galaxies when classification is performed with the Naïve Bayes and for equal frequency and equal width discretization method. Recall for all types is low enough, showing the weakness of Naïve Bayes algorithm in correctly classify the galaxies in their types.

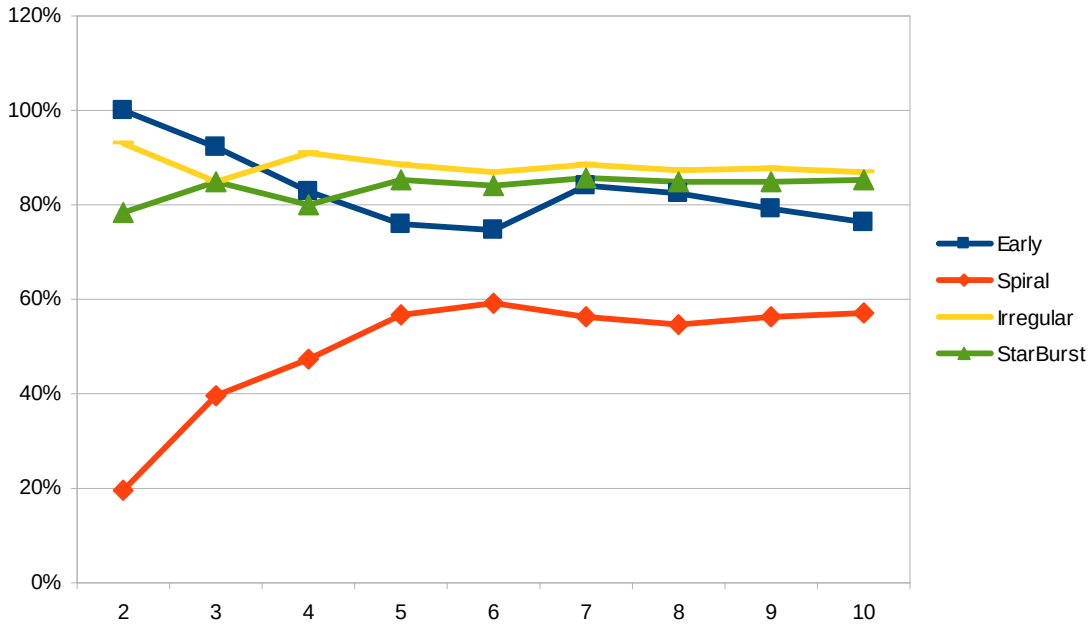


Figure 5-14 Recall ratio of all morphological types for the Naïve Bayes algorithm and equal frequency discretization method.

All the experiments have been conducted in a Pentium M 2 GHZ PC with 1 Gbyte of RAM. In Figure 5-16 the execution time (in seconds) for both algorithms and for both equal frequency and width discretization methods are shown for two to ten bins.

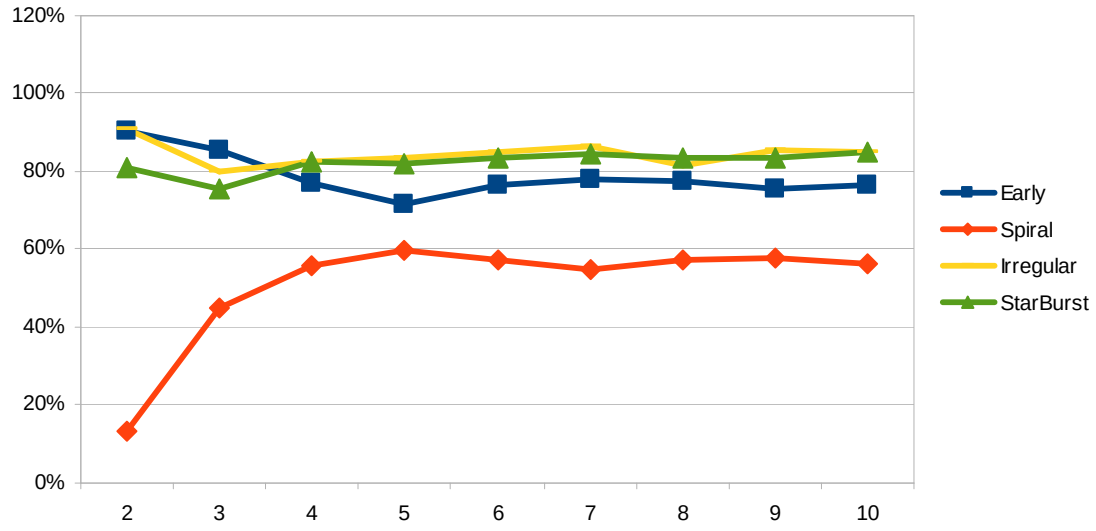


Figure 5-15 Recall ratio of all morphological types for the Naïve Bayes algorithm and equal width discretization method.

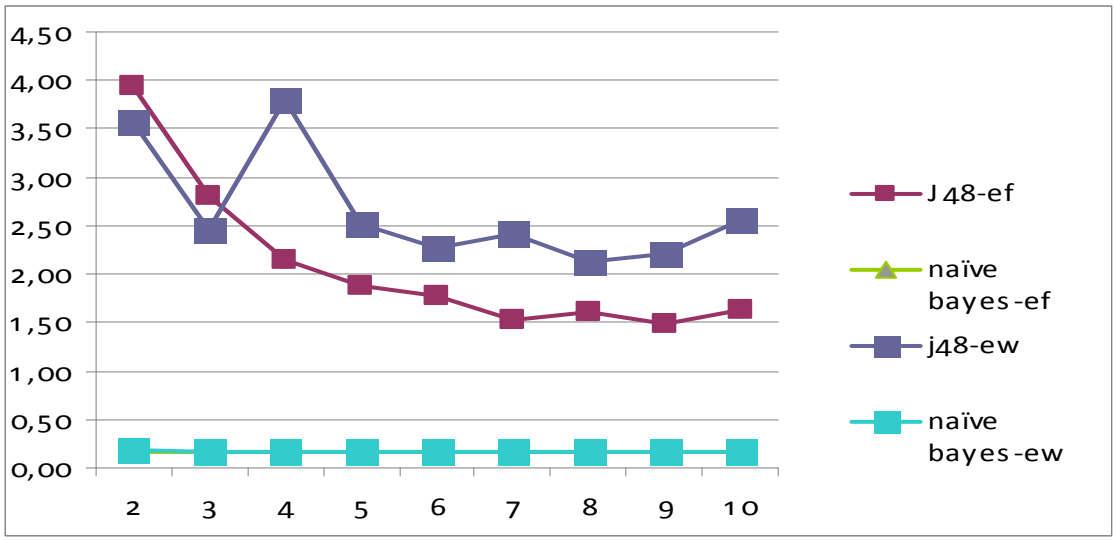


Figure 5-16 Execution time for J4.8 and Naive Bayes algororithms for equal frequency and equal width discretization methods

Naïve Bayes is a lot faster from J4.8 algorithm, while J4.8 is faster when it performs on the equal frequency discretized data, except in the case of two and three bins.

The output and the accuracy of the algorithms presented have been compared with that of the best SVM classification method performed with the R tool from the astronomers. SVM achieved a 92.2% classification accuracy. In Table 5-2 the classification accuracy for all algorithms and variations are presented.

Table 5-2 Classification accuracy of all three algorithms and for every variation of the experiments

Algorithm – discretization method	**Total classification accuracy**	**Accuracy for Early type**	**Accuracy for Spiral type**	**Accuracy for Irregular type**	**Accuracy for Starburst type**
Naive Bayes - equal frequency	71,43%	83,05%	49,63%	88,35%	83,75%
Naive Bayes- equal width	70,60%	78,75%	50,85%	84,46%	82,38%
J48 - equal frequency	97,25%	85,16%	98,25%	91,01%	99,60%
J48 - equal width	95,78%	76,18%	97,48%	87,08%	99,38%
SVM	92,20%	-	-	-	-

It is obvious that the J4.8 algorithm performs better with the equal frequency discretization method. The fact that it is much slower than Naïve Bayes is not important as the classification in the current project will be performed off-line. Figure 5-17 presents the classification accuracy for all the algorithms, showing the superiority of the J.48 algorithm, while SVM classification has almost the same good results.

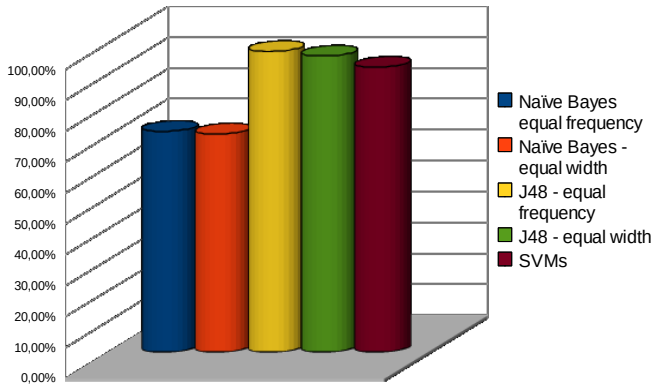


Figure 5-17 Classification accuracy for all algorithms

All the experiments and the output of the classification algorithms, as well as the classification decision tree produced, will be used in the GAIA project.

The use of the PBMS concept in applications where a lot of different experiments with various data mining algorithms and parameters are necessary is a powerful tool. All the experiments can be stored, recalled and

compared through the PBMS in an transparent and integrated way to the user.

5.3 Synopsis

In this chapter we presented a PBMS application scenario to facilitate the classification of astronomical data, showing the added-value for the domain experts of the application areas. The results of this application will be used to the GAIA ESA project, for the automatic classification of galaxies from a new space telescope.

6 PatternMiner – A Pattern Base Management System prototype

6.1 Introduction

In this chapter we present PatternMiner, a Pattern Base Management System prototype, while we further study the pattern evaluation issue, for patterns extracted from the data mining process, in order to extend the PBMS to include the pattern evaluation step. PatternMiner, is an integrated environment for pattern management and mining that deals with the whole lifecycle of patterns.

Moving one step forward from the mining, storage and comparison of the patterns, we study the problem of pattern evaluation using ontologies, to facilitate the difficult and time consuming task of the experts of pattern evaluation. We describe the problem and present a preliminery study in which we use domain ontology to filter association rules extracted from seismological data.

6.2 PatternMiner PBMS

In this section we present PatternMiner, a Pattern Base Management System prototype that is based on the theory described in the previous chapters. It is based on an XML pattern-base and uses XML documents to represent patterns, Xquery to retrieve documents. Pattern representation is based on the logical model defined on the PANDA framework, which is also used for pattern comparison tasks. Enhanced PMML schemata are used to represent the various patterns. PatternMiner is open source system and uses the WEKA data mining engine for pattern extraction.

The system architecture is presented and a demo is described followed by other potential applications.

PatternMiner, is an integrated environment for pattern management and mining that deals with the whole lifecycle of patterns from their generation (using data mining techniques) to their storage and querying, putting also emphasis on the comparison between patterns and meta-mining operations over the extracted patterns. This is in contrast to existing tools that deal with specific aspects of the pattern management problem, mostly representation and storage. Pattern comparison (comparing results of the data mining process) and meta-mining are high level pattern operations that can be applied in a variety of applications, from database change management to image comparison and retrieval. PatternMiner can also detect changes of clusterings extracted from dynamic data and thus, to provide insight on the dataset and to support strategic decisions without facing interoperability or incompatibility issues as if using different applications for each task. PatternMiner follows a modular architecture and integrates the different Data Mining components offering transparency to the end user.

PatternMiner adopts the PANDA framework concept for defining patterns and pattern-types and uses enhanced PMML schemata to implement this concept, offering interoperability with various systems supporting PMML.

No other PBMS or even another system with similar functions has been proposed, a review of all related approach can be found though in (Catania & Maddalena, 2006).

6.2.1 Implementation technologies and requirements

PatternMiner, as an integrated environment, should be transparent to the user and hide all the different components that are interconnected, providing all the functionality through its interface.

PatternMiner uses open source software, and is very easy to be upgraded or expanded.

In this section we describe the choices made for the implementation technologies and the special requirements that have been addressed.

Programming language

We choose the JAVA programming language to implement the PatternMiner tool for the following reasons.

WEKA data mining tool is an open source program, developed in JAVA and thus it is possible to use specific packages to load the data files, or to run the required algorithms and filters.

Furthermore, PatternMiner should offer a user-friendly interface. JAVA 2 GUI provides a flexible tool to implement complex interfaces. Java object-oriented properties provides also the possibility to put data and methods together, thus follow a modular approach.

Data mining engine

The data mining engine is responsible for the extraction of patterns according to user defined criteria, like dataset selection, pre-processing, mining algorithms and their parameters. We employ for this task *WEKA*, since it is an open source tool and offers a variety of algorithms for different mining tasks (including classification, clustering, and association rule extraction) as well as preprocessing capabilities over the data. Apart from the GUI version of the program, the Command line version allows to load files and execute algorithms from every other program using the provided API. The WEKA data mining tool has been tested from a lot of users and is very reliable. Every other data mining tool can be used, as long as its output is in PMML format.

Pattern representation Schema/ Model

For the pattern representation issue in the database, XML documents are used as they perform better from other approaches (Kotsifakos et al., 2005) and for the pattern comparison functions the PANDA framework (Ntoutsi, 2008) has been chosen.

In section 2.4.3 we outlined an XML schema that supports the PANDA representation model for patterns. For compatibility reasons with other database systems we adapted the PMML model to fit PANDA representation.

The pattern model according to PANDA framework is based on the quintituple pt = (n, ss, ds, ms, f). The PMML model for every pattern-type can be enhanced with metadata tags to include all the five parts of the pattern model as long as other useful information, such as the algorithm

and the parameters used for the pattern extraction, the time of the extraction etc.

In this section a more detailed description on the use of PMML in the patternMiner system will be made. The appropriate extensions to the PMML schema to implement the PANDA representation model are also will be presented.

PMML supports only pre-defined pattern-types (or models). Models that are supported in version 3.2 (PMML, 2009) are: Association Rules, Clusters, Trees, Neural networks, Series and more complex types such as Text and Support Vector Machines. With PMML, some quality measures related to the patterns can be represented. Furthermore, the relation between the patterns and the subset of input data (that the pattern represents) is also stored. The pattern extraction time is stored, too.

PMML structure includes:

1 *Header*. Includes general information about the pattern, such as the application created it, date and time of creation and a short description.

2 *Data Dictionary*. Defines the input data attributes for the patterns, their type and their value range.

3 *Transformation Dictionary*. PMML defines various kinds of simple data transformations:

   Normalization: map values to numbers, the input can be continuous or discrete.

   Discretization: map continuous values to discrete values.

   Value mapping: map discrete values to discrete values.

   Functions: derive a value by applying a function to one or more parameters

   Aggregation: summarize or collect groups of values, e.g., compute average.

   Transformation Dictionary is an optional element.

4 **Model*. Defines the specific information for each pattern type such as the data mining technique and the algorithm used for the pattern

extraction, the input data attributes used and other pattern-type specific information, like the frequent itemsets for an association rule model, or the clusters and their characteristics for a clustering model. Where * is the data mining technique name.

4.1 *Mining Schema*. Each model contains one mining schema, which lists the fields used in the model. These fields are a subset of the fields in the Data Dictionary. The mining schema contains information that is specific to a certain model, while the data dictionary contains data definitions that do not vary with the model. For example, the Mining Schema specifies the usage type of an attribute, which may be active (an input of the model), predicted (an output of the model), or supplementary (holding descriptive information and ignored by the model).

4.2 *Model Statistics*. The Model Statistics component contains basic univariate statistics about the model, such as the minimum, maximum, mean, standard deviation, median, etc., of numerical attributes.

PMML supports Model Composition. Simple models can be used as transformations. PMML offers the possibility to combine multiple conventional models into a single new one, using individual models as building blocks. This can result in models being used in sequence, where the result of each model is the input for the next one. This approach, called model sequencing, is not only useful for building more complex models, but can also be put to good use for data preparation. Another form of model composition is also supported: the result of a model can be used to select which model should be applied next. For example, a decision tree can now have an embedded regression model in each leaf node.

Both model sequencing and model selection can be combined to develop quite complex models.

PMML supports functions that can be used to perform preprocessing steps on the input data. A number of predefined built-in functions for simple arithmetic operations like sum, difference, product, division, square root, logarithm, etc., for numeric input fields, as well as functions for string handling, such as functions for trimming blanks or choosing substrings.

In PMML there is also a Model verification mechanism for model verification that increases the compatibility of models between different vendors' applications consuming PMML. A verification model provides a mechanism for attaching a sample data set with sample results so that a PMML consumer can verify that a model has been implemented correctly. This will make model exchange a lot more transparent for users and inform them in advance in case compatibility problems might arise.

Except the default information that PMML stores for every model, during the model (pattern) insertion to the XML pattern-base, for each pattern three metadata elements are added:

- dateCreated: the date and time that the pattern has been inserted to the pattern-base.
- dataFileName: the file name (including the path) of the file containing the source data.
- modelName: pattern name as user defines it.

In every PMML document a tag “extension” has been created containing necessary information for the clustering pattern. More specifically:

The Prior Probability and the scatter value of the cluster, with extension name «Prior probability» and «Scatter value» respectively.

The percentage of the instances that belong to each cluster of the complex pattern, with extension name «Clustered Instances».

By the time this dissertation was written, a new version of the PMML standard had been announced.

Version 4.0 of PMML adds the following new features:

- support for time series models;
- support for multiple models, which includes support for both segmented models and ensembles of models;
- improved support for preprocessing data, which will help simplify deployment of models;
- new models, such as survival models;

- support for additional information about models called model explanation, which includes information for visualization, model quality, gains and lift charts, confusion matrix, and related information.

This new version of PMML is a major update of PMML Version 3.2, which was released in May, 2007.

Pattern Storage and retrieval system

As it has been shown in section 2.4 the XML model is more proper to represent and manage patterns. A native XML database, thus, it would be the best choice for the storage of the patterns, having the following advantages:

- XML data are inserted into the database without the need of extra preprocess. Patterns are stored directly as XML documents.
- Every character (including the space and other special characters) of the XML document remains unchanged after the insertion into the database.
- Queries in the XML database return the whole document or part of them, preserving the hierarchical structure of the documents.

Furthermore, data exchange is far more easy and requires no transformation of the documents in different structures.

For a native XML database we choose ORACLE Berkeley DB XML (2009). Berkeley DB XML stores XML documents in logical groups "*Containers*", that are identical to "*Collections*" in other XML databases. Users can define various properties for each container, including the option for document validation, storing whole documents or specific parts and index creation.

In the current application patterns are grouped based on the data mining technique used to extract them. So, there are three basic "containers": AssociationRules.dbxml, Clustering.dbxml και Trees.dbxml.

Berkeley DB XML can also store non-XML documents as well as XML document meta-data. Metadata are user-defined couples "property-value" and they can be retrieved as child elements of the root element, while they do not really appear in the stored XML documents.

Berkeley DB XML supports the XQuery language to address queries to the database and it supports indices on the XML data to speed-up the query.

Berkeley DB XML supports command line and provides APIs for C++, Java, Tcl, Perl, Python and PHP. There are also available third-party APIs for other languages. It supports UNIX, Linux, Windows and Mac OS X.

6.2.2 System architecture

The PatternMiner architecture is depicted in Figure 6-1. In the core of the system lies the *PatternMiner engine* which arranges the communication between the different peripheral components (Data Mining engine, Pattern Base, Pattern Comparison module, Meta-mining module) and also provides the end user interface.

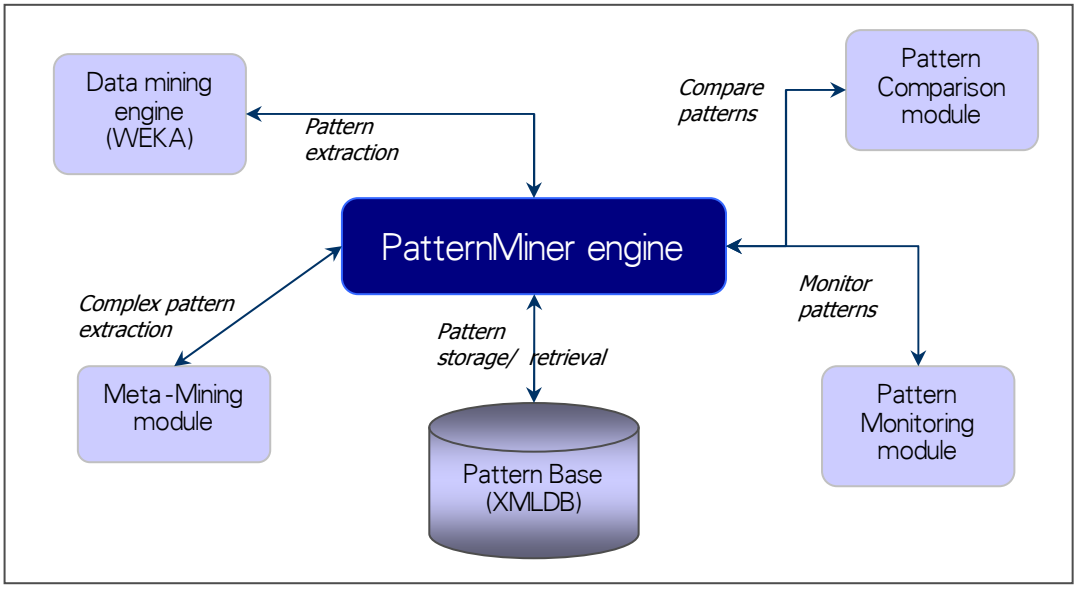


Figure 6-1 PatternMiner architecture

Pattern extraction: The Pattern Extraction component is responsible for the extraction of patterns according to user defined criteria, like dataset selection, pre-processing, mining algorithms and their parameters. We employ for this task *WEKA* (Witten and Frank, 2005), since it is an open source tool and offers a variety of algorithms for different mining tasks (including classification, clustering, and association rule extraction) as well as preprocessing capabilities over the data.

Pattern representation: Pattern representation is not a simple task mainly because one can find a great variety of pattern types (decision trees,

clusters, etc.) of varying complexity. The need for pattern representation in KDD has been recognized by both research and industrial communities and several representation approaches have been proposed. The most popular choice is *PMML* (2009), an XML-based language that provides a quick and easy way to define data mining and statistical models using a vendor-independent method and share these models between PMML compliant applications. The structure of the models in PMML is described by an XML Schema; different models have their own schemes. The term "model" in PMML is equivalent to the term "pattern type" in our approach. In PatternMiner, as we described in the previous section, we use the PMML standard for the representation of patterns, enhanced to fully match the PANDA framework concepts, and thus, we convert the output of the *Data Mining engine* component into PMML format.

Pattern storage: Since patterns are represented as XML documents (through PMML), a native XML database system is used for their storage in the *Pattern Base*. In particular, we employ the open source *Berkeley DBXML* (Oracle Corp. Berkeley DB XML, 2009), which comprises an extension of the Berkeley DB with the addition of an XML parser, XML indexes and the XQuery data query language. Berkeley DBXML stores XML documents into logical groups, called Containers (the Collections in other native XML database systems). Users can define various properties for each container (whether to store the whole document or parts of it, which indexes to create, etc.). Apart from XML documents, non-XML documents as well as metadata for the XML documents can be stored. Metadata are user-defined in the form "property-value" and easily retrieved.

Pattern querying: PatternMiner provides a basic environment for querying the pattern base. The user defines the pattern set to be queried, and imposes his/her query in the XQuery language (Xquery 1.0, 2003). Regarding the supported query types, the user can retrieve either the whole pattern or any component of the pattern (either the structure or the measure component) and of course, to impose constraints over these components. PatternMiner creates the proper connection to the pattern base and captures the result in order to return it to the user. The result is shown in the screen while it is also saved in the file system.

Pattern comparison: One of the most important operations on patterns is that of *pattern comparison*. Defining dissimilarity operators for patterns could be used to express similarity queries, including *k-nearest neighbor queries* (i.e. find the k-most similar pattern(s) to a query pattern) and *range queries* (i.e. find the most similar pattern(s) to a given pattern within a given range). Dissimilarity could be also employed in order to *monitor* and *detect changes* upon patterns extracted from a dynamic environment (Spiliopoulou et al., 2006). Recognizing the importance of dissimilarity assessment in pattern management, we distinguish the comparison process from the querying process and we implement it separately through the *Pattern comparison module*. The comparison is carried out on the basis of *PANDA* (Bartolini et al., 2004; Ntoutsi et al., 2007), a generic and flexible framework for the comparison of patterns defined over raw data and over other patterns as well. Comparison utilizes both structure and measure components of patterns. The user defines the patterns as well as the way that they should be compared, i.e. how the different components of PANDA are instantiated. The output is a dissimilarity score accompanied with a justification, a report actually of how the component patterns have been matched. In our experiments and for the needs of some real case studies (Iakovidis et al., 2006) we enhanced the PANDA framework by adding a couple of new cluster comparison algorithms.

Meta-mining: Due to the large amount of extracted patterns, several approaches have lately emerged that apply Data Mining techniques over patterns instead of raw data, in order to extract more compact information. The *Meta-mining module* takes as input a set of different clustering results extracted from the same dataset (through different clustering algorithms or different parameters) or from different datasets (through from the same generative distribution) and applies Data Mining techniques over them, in order to extract *meta-patterns*. So far, the meta-mining component focuses on meta-clustering (Caruana et al., 2006), i.e. grouping of clustering results into groups of similar clusterings. The user has full control of the clustering process by choosing the similarity function and the clustering algorithm.

Pattern Monitoring: While PatternMiner is a tool for managing all types of patterns, at the current moment we have implemented a Cluster Monitoring technique that is based on the theory and algorithm described in

(Spiliopoulou et al., 2006). In this approach, the transitions of clusters extracted upon an accumulating dataset are traced and modeled. Clustering occurs at specific timepoints and a "data ageing" function can be used to assigns lower weights to all or some of the past records. The set of features used for clustering may also change during the period of observation, thus allowing for the inclusion of new features and the removal of obsolete ones. PatternMiner assumes re-clustering rather than cluster adaptation at each timepoint, so that both changes in existing clusters and new clusters can be monitored. Transitions can be detected even when the underlying feature space changes, i.e. when cluster adaptation is not possible. Terms like cluster match, cluster overlap, cluster transition and lifetime of a cluster are core notions of cluster monitoring. This module exploits the clusterings that are stored in the pattern-base and employs the query and comparison capabilities of the system.

6.2.3 A PatternMiner Demo

In this section we present a demo of the PatternMiner system, that has been also presented in (Kotsifakos et al., 2008a, 2008b)

6.2.3.1 Demo presentation

To make clear the potential use and the value of PatternMiner, we consider a supermarket as a simple case study and its manager as the end user. Among other pattern types, the manager is interested in discovering the products that customers tend to buy together, i.e. association rules. Except for knowing the product associations at each month, the manager also wants to know how these associations change from month to month: are there any new associations, did some old association disappeared, did some association became stronger (higher confidence) or weaker. Also, he/she wants to discover groups of months with similar associations, so as to decide some strategy for each group instead of each month. This process involves storage of the patterns discovered at each month, querying, comparison and meta-mining operations over them. Existing Data Mining tools do not address all these issues. On the contrary, PatternMiner provides the manager with all this information in an easy and transparent way. We describe below how each component works for this supermarket scenario.

Pattern extraction and storage: The user defines the data source, the Data Mining algorithm and its parameters, e.g. in our case the supermarket database, the association rule algorithm and the minimum support and confidence parameters. The extraction takes place in the Data Mining engine and the results are converted into PMML format before being stored in a user-specified container in the XML pattern base (as well as in a file on the hard disk). In Figure 6-2 the pattern extraction and storage screen is depicted for the case of association rule patterns. Using PMML, the exchange of patterns between different applications is possible without the need for special import-export tools.

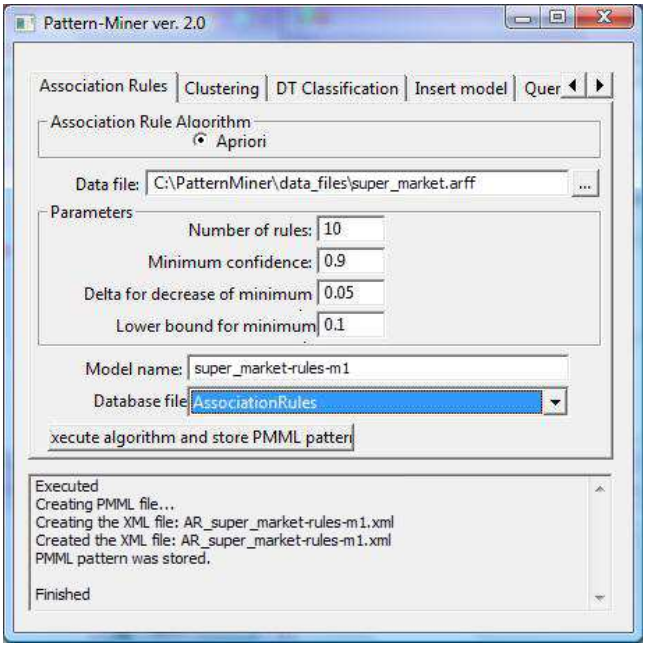


Figure 6-2 The association-rule extraction screen

Pattern querying: The user defines the pattern set to be queried and the query itself, in Xquery language. PatternMiner *engine* creates the connection to the pattern base, executes the query and returns the results to the user (and also saves them to a file). A sample query is shown in Figure 6-3, described in both natural language and Xquery.

Query (natural language):

Retrieve the association rules from the super_market dataset that have a support value greater than 0.2.

Query (XQuery):

```
declare namespace a = "http://www.dmg.org/PMML-3_1";
collection ("AssociationRules.dbxml")
[dbxml:metadata ("dbxml:dataFileName")=
"C:\PatternMiner\data_files\ supper_market.arff"]
/a:PMML /a:AssociationModel /a:AssociationRule
[@support>0.2]
```

Figure 6-3 A sample query in natural language and in XQuery

Pattern comparison: The user defines the patterns to be compared as well as the comparison parameters. In our example, the manager asks for the comparison of association rule patters extracted from the supermarket data of the two previous months, in order to inspect whether and how the buying behavior has been changed. The patterns are retrieved from the Pattern-base. Then, the manager configures PANDA by choosing the appropriate comparison function from the candidate functions implemented for each pattern type. It should be noticed that in the PANDA framework there are several comparison functions implemented, and the user, depending on the application can decide or test what function better fits his/her application. The results are returned to the manager, who can detect any changes in the sales-patterns and decide whether these changes were expected (based on company's strategy) or not (indicating some suspicious or non-predictable behavior). Based on the results, the manager can decide future strategies regarding offerings, supply etc.

The manager can also extract clusters of customers based on their buying habits or their demographics. Comparing such clusters of customers can reveal buying patterns over the year, and thus the manager can decide about the supplies. In Figure 6-4 the clustering comparison tab is shown.

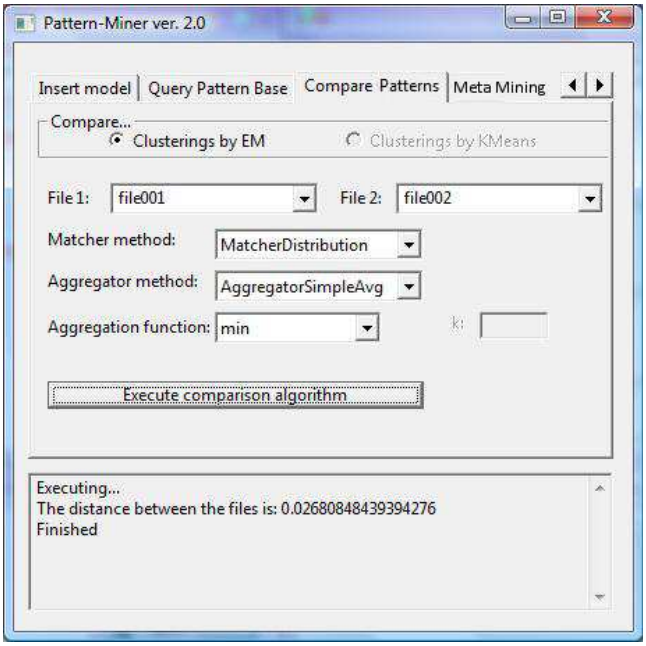


Figure 6-4 Pattern Comparison Tab in PatternMiner

Meta-mining: The user defines the pattern sets to be used as input to the Meta-mining module (e.g. sets of rules extracted at each month of 2007), selects the clustering algorithm/ parameters, as well as the similarity measure between sets of rules. The input sets are clustered into groups of similar sets of rules (e.g. March and April could be placed to the same group, since they depict similar buying behavior), which can be also stored in the pattern base for future use. The manager can exploit these results in order to decide similar strategies for months belonging to the same cluster.

Cluster Monitoring: User defines the dataset from which the clusters have been extracted. A list of all the clusterings that have been carried out over the specific dataset is available to the user, sorted by the extraction time. The supermarket manager wants to observe the customer profiles over time. Choosing the appropriate dataset (supermarket.arff), PatternMiner returns all the different clusterings that have been created from that dataset, along with the clustering algorithm and the extraction time. The manager chooses two or more clusterings and runs the cluster monitoring process. This process results in a matrix showing the clusters of the first clustering and their changes over time (new clusters, clusters that no longer exists, shrinked or expanded clusters etc). The output represents the graph depicted in Figure 6-5 (Spiliopoulou et al., 2006).

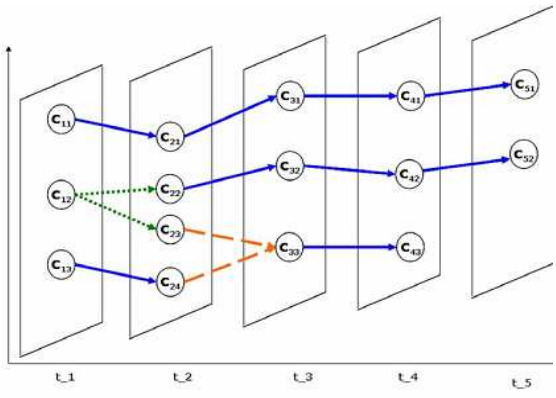


Figure 6-5 Graphical representation of cluster monitoring output

6.2.3.2 Discussion

PatternMiner is an integrated environment for pattern management that supports the whole lifecycle of patterns from their generation to their retrieval, and also offers sophisticated operations over patterns, like comparison and meta-mining. PatternMiner follows a modular architecture that employs state-of-the-art approaches at each component. The different building blocks are implemented in JAVA.

Several improvements can be carried out: First, the existing components can be enhanced. For example, the querying component could support more query types, like k-nearest neighbor queries, range queries and also the query processing could be more efficient by employing appropriate index structures. Also, the *Meta-mining module* and *cluster monitoring* can be extended so as to support more pattern types, like decision trees, association rules, sequences.

Except for the scenario we described, other potential applications include pattern validation, monitoring/ change detection, comparison of patterns extracted from different sites in a distributed environment setting, etc.

6.3 Extending PBMS to support pattern evaluation using ontologies

In the *Knowledge Discovery from Data* (KDD) process, Data Mining techniques are used to find patterns from a large collection of data (Data Mining step in Figure 1-1). The role of the domain experts in this process is crucial. Their knowledge is used in early stages to prepare data (i.e. to decide for the data cleaning and preparation) and to choose the appropriate

parameters for the data mining algorithms. Their contribution is also necessary for the evaluation and interpretation of the extracted patterns that lead to the generation of knowledge (Fayyad, Piatetsky-Shapiro, & Smyth, 1996).

In essence, extracted patterns are used from domain experts to explore new relations on data, evaluate theories on the field of interest, and discover unknown and hidden knowledge that will lead to new experiments and theories. However, some of the extracted patterns are considered trivial and some others insignificant, according to the domain knowledge. To evaluate extracted patterns experts have defined a lot of different, either objective or subjective interestingness measures based mostly on statistical properties of the patterns. Nevertheless, analyzing and assessing the usefulness of discovered patterns is a laborious task and is considered a hard problem (Piatetsky-Shapiro, 2000).

The issue raised here is related to the *incorporation of the existing domain knowledge in the Data Mining process*, and especially in the pattern evaluation phase. Several statistical and interestingness measures have been proposed for the evaluation of patterns (Piatetsky-Shapiro, 1991; Freitas, 1999; Silberschatz & Tuzhilin, 1996; Piatetsky-Shapiro, & Matheus, 1994). These measures are applied either before or during the data mining process. In the first case, they are used to reduce the number of patterns that will be extracted and to speed up the data mining process, while in the evaluation phase, they are used to clean up the patterns considered insignificant.

Nevertheless, no such measure for pattern evaluation is efficient enough as the domain expertise itself. Domain experts can better evaluate the patterns and decide whether they are trivial or not. It is the user who will distinguish interesting rare occurrences of patterns from statistical noise using his/her background knowledge (Pohle, 2003). In order to automate the pattern evaluation process, we need to incorporate the domain knowledge in it. It is generally acceptable that domain knowledge can be represented efficiently using ontologies (Pohle, 2003). An *ontology* is a specification of a conceptualization, a description of the concepts and relationships that can exist for an agent or a community of agents (Gruber, 1993).

We argue that domain knowledge expressed with ontologies could function as a filter in the evaluation phase of the KDD process. Patterns extracted from data mining algorithms would be first evaluated with respect to the ontology. Patterns that contradict to knowledge widely accepted according to the ontology provided (hereafter, called "*noisy*") will be marked as possibly invalid. Whereas, acceptable patterns, will be further evaluated by the domain expert and, if recognized as useful knowledge, the ontology could be updated to incorporate these new patterns (of course, domain experts might reconsider the ontology by adding/removing relations, associations etc). In this case, priority is given to patterns considered interesting, at the same time not conflicting with well established beliefs. This approach could reduce the cost in terms of running time of the data mining algorithm and the effort of the domain expert to evaluate the discovered patterns. Note that "noisy" patterns are marked as invalid and are not being discarded unless user wishes so. Thus the danger to drop really useful knowledge is quite limited.

Towards the purpose of incorporating the domain knowledge in the evaluation phase of the Knowledge Discovery process, we propose the use of ontologies that describe the field of interest to evaluate data mining results. In the following sections we will discuss the various challenges and problems that have to be faced considering a real case study from the seismology domain.

6.3.1 Data Mining Using Domain Knowledge

Until recently, although the importance of knowledge management was widely known, limited research has been devoted to intelligent pattern analysis and the accumulation of discovered knowledge with prior knowledge (Pohle, 2003). Regarding the use of domain knowledge in the data mining process only a few related approaches can be found. Domain knowledge can be applied in the data mining process in three different ways. In the preprocessing step (to prepare the data to be mined), during the data mining process (data mining algorithm is using the domain knowledge to decide about the next step), or after the data mining process (to evaluate the extracted patterns).

Considering the first way, X. Chen et al. (2003) propose using an ontology as a concept hierarchy to prepare demographic data for association rule

mining. In some tuples of the demographic database there are values from a lower level of the hierarchy while in other tuples, in the same column, there are values from higher levels of the hierarchy (for example the value "basketball" and the value "recreation sports" that are found at different levels in an interests hierarchy). By replacing the values of lower level with values at a higher level (raising), the authors show that the rule support is increasing and thus, more rules can be found.

Several papers can be found about how some interestingness measures (either objective or subjective) are used to evaluate extracted patterns. Objective interestingness measures are based in statistical functions. In (Piatetsky-Shapiro, 2000) basic principles of objective rule interestingness measures are defined, while in (Freitas, 1999) a comparison of objective interestingness criteria can be found. In contrast with objective interestingness measures, subjective measures try to take into account individual conditions of the human analyst. A general discussion can be found in (Silberschatz & Tuzhilin, 1996), while (Piatetsky-Shapiro, & Matheus, 1994) and (Padmanabhan, & Tuzhilin, 1998) attempt to address this problem. All these approaches provide a way to evaluate patterns but do not make use of the domain knowledge.

There are also few attempts using domain knowledge to improve evaluation of extracted patterns. Domain knowledge in the form of concept hierarchies can be used to improve Web mining results (Pohle & Spiliopoulou, 2002), while an interestingness analysis system that requires the user to express various types of existing knowledge in terms of a proprietary specification language is presented in (Liu, Hsu, S. Chen, & Ma, 2000). These approaches do use domain knowledge, but their disadvantage is that they require the user to previously provide his/her knowledge in a specified and narrow form, according to the application each time.

In order to incorporate domain knowledge in data mining and to allow conceptual model sharing in domains, the use of ontologies is necessary (Maedche, Motik, Stojanovic, Studer, & Volz, 2003). An application of using ontologies before, during and after the data mining process is the one presented by Hotho, Maedche, Staab and Zacharias (2002), in which authors use ontologies and Information Extraction technologies to improve text mining algorithms and pattern interpretation.

Our methodology uses ontologies to improve the pattern evaluation step and querying the pattern base. During the evaluation step, the system based on the provided ontology and parameters that have been defined from the domain expert, filters the patterns and marks as "noisy" patterns that contradict to domain knowledge. Domain expert can then discard or further evaluate them. The system will also use the filtering mechanism to prevent a naïve user to query the pattern base for "noisy" patterns.

6.3.2 Problem Description

Various examples indicating the need for integration of domain knowledge and data mining can be found, however, dealing with scientific data is more efficient mainly because domain experts in these areas know their data in intimate detail (Fayyad, Haussler, & Stolorz, 1996). In this section, we present a real case study of mining seismological data to illustrate the use of a PBMS and ontologies in an integrated environment for pattern management and evaluation.

6.3.2.1 A Case Scenario From The Seismological Domain

Let us consider a seismological database containing historical data about earthquake events (Theodoridis, Marketos, & Kalogeras, 2004). Such a database would include information about the event (magnitude, latitude / longitude coordinates, timestamp and depth), the geographical position of both the earthquake epicenter and the affected sites that partitions world in disjoint polygons), as well as details about the fault(s) related with the event. Additionally, our database includes demographical and other information about the administrative partitions of countries, details about the geological morphology of the areas of various countries and macroseismic information (intensity, etc) (Theodoridis, Marketos, & Kalogeras, 2004).

Seismologists use the database to store the data, a data warehouse to aggregate and analyze them, a knowledge base to store documents collected by various sources, and a tool to define ontologies to represent the domain area. Furthermore, they are interested in discovering hidden knowledge. Patterns produced by the KDD process are evaluated and stored in a PBMS. Obviously, if the above "islands of information" are not integrated under a single tool then the maximum value of the stored information could not be

utilized. The researcher is interested in posing a number of questions, perhaps having no idea about which tool to use to get the answers. Some query examples are:

- *Query 1*: Find the average magnitude and the max depth for the earthquakes happened in the North Adriatic Sea (or in a particular geographical area) for the decade 1994-2004.

- *Query 2*: Is there any information about the earthquake maximum recorded intensity when I know that the depth of the epicenter is over 60 km and the geology of the site is characterized as rocky?

- *Query 3*: Find similarities in shock sequences (a main shock that follows pre-shocks and is followed by intensive aftershocks) happened in Greece during 2004.

Query 1 can be easily answered by a data-warehouse using the average and the max function on the appropriate earthquake data. Query 2 can also be easily answered using a decision tree. In case such a decision tree model (pattern) has not been already stored in the PBMS then an appropriate classification algorithm can be applied on the data. Query 3 is more challenging since it requires the incorporation of more advanced domain knowledge: a) the specification of the similarity measure and b) the definition of the shock sequence by the domain expert.

It is clear that Query 3 requires a lot of pre-processing work to be done by the seismologist in collaboration with a database analyst. Hierarchies and rules about seismological concepts and data have to be defined before a data mining algorithm is applied. Furthermore, even when patterns are produced and stored in the PBMS some more post-processing work (similar to the pre-processing step) has to be done in order to extract the appropriate information. The seismologist may have already represented the required knowledge using ontologies, their integration into the PBMS could resolve the above problems.

On the other hand, other queries , such as:

- Query 4: Find any relation between earthquake magnitude and average temperature of the area around the epicenter during a related time period.

- Query 5: Find any relation between earthquake magnitude and season of the year.

can also be posed by a naïve (i.e. non-expert) user and answered applying data mining tasks while semantically unacceptable (for example, seismologists do not recognize any relation between either earthquake magnitude and surface temperature or earthquake magnitude and season of the year). Although, the data mining engine could return results regarding these relations, a domain expert would definitely discard them.

Nevertheless, such a filtering is nowadays done manually at a post-processing step. Exactly this is the contribution of the integrated ontology-enabled PBMS we propose: to filter out "noisy" patterns efficiently (i.e. online without the need of post-processing) and effectively (i.e. with a quality guaranteed by the ontology-filter).

6.3.2.2 Domain Knowledge Using Ontologies

One of the challenges in incorporating prior knowledge in the Knowledge Discovery process is the representation of the domain knowledge. Ontologies are useful in providing the formalization of the description of a domain. They are considered as the explicit specification of a conceptualization (Guarino & Giaretta, 1995). Using ontologies, hierarchies of concepts, constraints and axioms can be defined. In other words, ontologies provide a domain vocabulary capturing a shared understanding of terms.

To represent the seismological domain, we choose the Suggested Upper Merged Ontology (IEEE Standard Upper Ontology) (Niles & Pease, 2001), the Mid-Level Ontology (Niles & Terry, 2004) and, finally, an ontology for representing geographical information all available at (SUMO, 2009). An upper ontology is limited to concepts that are meta, generic, abstract or philosophical, and hence are general enough to address (at a high level) a broad range of domain areas. Concepts specific to particular domains are not included in an upper ontology, but such an ontology does provide a structure upon which ontologies for specific domains (e.g. medicine, finance, engineering, etc.) can be constructed. A mid-level ontology is intended to act as a bridge between the high-level abstractions of the SUMO and the low-level detail of the domain ontologies which in our case is the geography ontology. The following schema is based on the above ontologies (Figure 6-6).

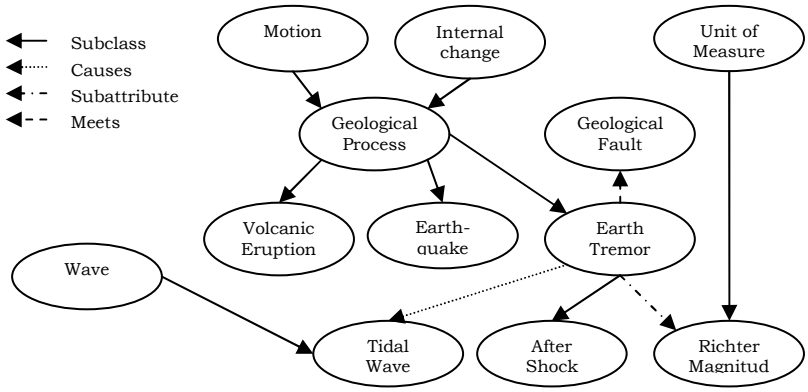


Figure 6-6 A subset of the SUMO for seismology

Obviously, the above figure does not represent the "Universe of Discourse", but is a part of the geography ontology related to seismology. It is clear that using ontologies, horizontal relationships between concepts can be defined (Pohle, 2003). For instance, in the domain of seismology there is such a relationship between seismology and geology (faults). This is important as the patterns that are stored for each domain in the PBMS, can be combined offering more complete querying and visualization capabilities to the user.

Regarding association rule mining, a general rule that can be used to evaluate patterns with ontologies, is that patterns should associate attributes that belong to the same class or to subclasses of the same class. Reasonably, association of attributes belonging in different classes (in the ontology-hierarchy graph) or in classes that are several nodes away in the ontology diagram might result in false associations of irrelevant (according to domain knowledge) attributes. Edge-distance and other approaches have been already proposed for searching semantic similarity between objects in an ontology. Such measures can be used to assess the relation between two attributes. This implies that the user can select the level of relevance between the attributes, defining the maximum distance that a class can have from another in the ontology graph.

The task of defining the rules that will be used to filter the patterns to be extracted involves the study of the ontology as well as the study of the pattern type and the results that users anticipate. Ontology components are classes, attributes and relations between them. Classes have subclasses and

each class may have a number of attributes. Usually classes for related concepts, belong to the same parent class while not related concepts are under different classes. The whole class and subclass diagram, define a kind of hierarchy with various levels of detail. For example, classes "VolcanicEruption" and "Earhtremor" (Figure 6-6) lie at the same level, while their subclasses "volcanicGasRelease" and "AfterShock" lie at lower level.

As each pattern has different structure, filters for every pattern type have to be defined. Specifically for the Association Rule pattern type, we define the Association Rule Filter. Each part of the rule contains attributes (depth, magnitude etc) that are related in the relational model, but also related in some way in the ontology. Thus, we can define for each rule a distance metric between the main earthquake class (*earth tremor*) and the nodes of the attributes contained in the rule. The shorter this distance is, the more the attributes are semantically related. In fact, we can define two approaches to measure this distance: in the so-called "*Risky*" approach, we consider the maximum distance between the nodes of the attributes and the main earthquake class, whereas in the "*Not Risky*", we consider the minimum distance between them. Obviously, the attributes of the earthquake class have distance=0 and thus there are not included in this calculation.

A user selects the level of semantic relevance by specifying the maximum distance of the nodes from the main earthquake class. For instance, one may be interested in finding relationships not just between the attributes on an earthquake but also between them and geological faults. Thus, the level of semantic relevance has to be increased so as to include the appropriate node.

With the above described process, a subgraph of the ontology that contains the attributes under consideration is constructed. Attributes of the produced rules are validated against this ontology subgraph. If all are included in the subgraph then the association rule that contains them is considered as semantically valid. Otherwise, if some of the attributes are not in the subgraph, the rules containing them are marked as "noisy". Note that the system does not reject "noisy" rules (although there is such an option) as they might contain previously unknown knowledge about the relations of some attributes, and thus domain expert's attention is required. Some rules

can lead to new interesting relations and domain experts may reconsider the ontology.

6.3.3 Preliminary Validation Study

In this section, we use the example from seismology domain and the ontology defined in section 6.3.2.2 to describe system functionality. The system performs a validation test before the data mining process, checking if the user defined parameters make sense. For example, a user could ask the system to perform the Apriori algorithm to find associations between the "magnitude" and the "date" of an earthquake. As mentioned in section 6.3.2.1 this association is not acceptable by the seismology domain and thus the system will suggest the user to change the parameters. If the user does not specify the attributes that he/she wants to search for associations, the system will perform the data mining algorithm using all attributes but, when generating the frequent itemsets, it will discard itemsets that contain values from attributes not related in the ontology. In this way, the time consuming phase of frequent itemset generation will be improved and no irrelative association rules will be generated. Of course, this is not always desirable as some interesting rules might not be generated. In this case the user should decide for these rules. So, it is given as option to the user either to enable the system to automatically discard them or just to mark the "noisy"ones for further evaluation. In the latter case, the user decides which rules are interesting and should be stored to the pattern base.

Another case is when a user is posing a query to the pattern base to retrieve patterns for example "fetch association rule patterns that contain both "season" and "depth" attributes and the support of the rule is greater than 0.3". Such rules are not valid according to the domain knowledge and thus the system notifies the user that it is rather impossible to find rules like those in the pattern base.

In our first experiments, we ran the Apriori data mining algorithm implemented in WEKA (Witten & Frank, 2005) to extract some association rules using real macroseismic data collected by the Greek Institute of Geodynamics (Seismo-Surfer). Attributes such as earthquake depth, intensity, site, date and season of the year are some of the attributes of the table that contains 10336 tuples for the earthquake events during the 20th

century. Table 6-1 lists 25 out of 70 rules extracted by Apriori confidence threshold = 30% and support threshold = 10%.

Table 6-1 Association rules extracted from seismological data

id	Association Rule	Conf.	Supp.
1	intensity≥5 → distance≤80	74%	19%
2	weekDay=Tuesday, 11≤depth≤20 → season=Summer	71%	10%
3	weekDay=Tuesday → season=Summer	71%	17%
4	weekDay=Monday → season=Spring	68%	10%
5	season=Summer → 11≤depth≤20	65%	21%
6	weekDay=Saturday → 21≤depth≤50	62%	12%
7	depth≥50 → season=Spring	60%	11%
8	distance≥150 → intensity≤3	59%	15%
9	weekDay=Tuesday, season=Summer → 11≤depth≤20	57%	10%
10	weekDay=Tuesday → 11≤depth≤20	57%	14%
11	11≤depth≤20 → season=Summer	57%	21%
12	season=Autumn → 11≤depth≤20	55%	14%
13	season=Summer → weekDay=Tuesday	54%	17%
14	intensity≤3 → distance≥150	54%	15%
15	distance≤80 → intensity≥5	52%	19%
16	distance≥150 → 1000<population≤4000	48%	13%
17	3<intensity≤4 → 80<distance<150	48%	15%
18	season=Summer, 11≤depth≤20 → weekDay=Tuesday	48%	10%
19	weekDay=Tuesday → 1000<population≤4000	46%	11%
20	season=Spring → 21≤depth≤50	46%	14%
21	intensity≤3 → 1000<population≤4000	46%	13%
22	21≤depth≤50 → season=Spring	45%	14%
23	500<population≤1000 → distance≤80	43%	11%
24	season=Spring → 1000<population≤4000	43%	13%
25	80<distance<150 → 1000<population≤4000	43%	15%

Out of these 25 rules, the domain expert marked only five rules (ids 1, 8, 14, 15, 17) as interesting and all others as "noisy" because they describe a correlation between attributes/classes that is meaningless in the domain of seismology. The system needs a threshold parameter to be defined in order to mark some rules as "noisy". This threshold is the maximum path distance from the main "earth tremor" node/class. When this threshold is defined, the system retrieves the subgraph of the ontology defined by the "earth trermor" node and all the nodes with path distance less or equal to the threshold. Every rule that has attributes belonging to that subgraph, will be considered interesting while all others will be marked as "noisy".

Trying to detect a reasonable threshold in order for the system to retrieve the rules that will match the expert's evaluation, we varied threshold value from 1 to 5 and computed the rules marked as "noisy" by the system. This is illustrated in Figure 6-7.

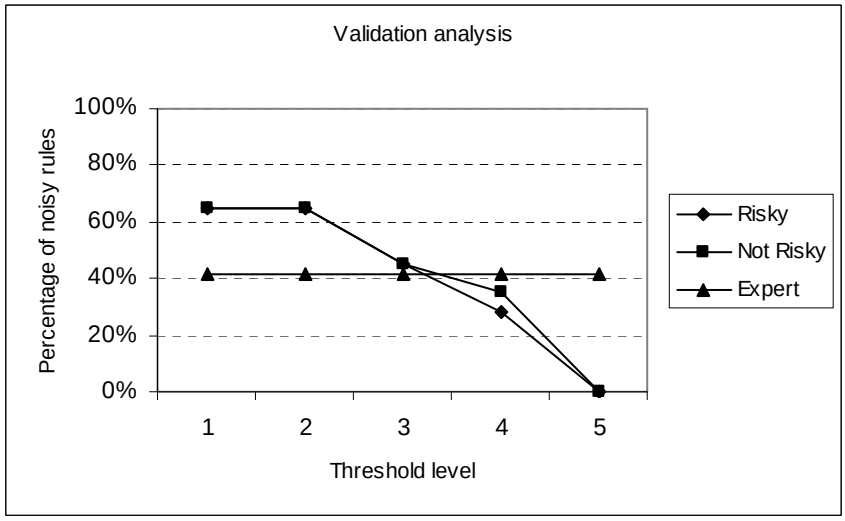


Figure 6-7 Threshold and rules rejected by the system and the seismologist

According to this experiment we conclude that with threshold 3 the system matches expert choices. As such, this threshold can be used by the user for the next running of Apriori or can even be stored as meta-data for the specific dataset and KDD process for future data mining.

With the procedure described above we can measure the percentage of the rules that will be marked as "noisy" by the system and by the expert, but we do not know if these are the same rules i.e. if the rules marked by the system are the same with the rules marked by the domain expert (precision).

While in our particular experiment we had a perfect match, it is not sure that we will have a perfect match each time.

6.3.4 Discussion

Using ontologies in the data mining process is an area of recent research and its applications could be many. Apart from geosciences, every field that has a well defined ontology can use the integrated framework to improve the KDD process. For example in the domain of B2B marketplaces, finding associations between products is more efficient when using the hierarchies defined in the product ontology. Although there is not currently universally accepted product ontology, efforts are made to integrate different product ontologies (Omelayenko, 2000) towards this end.

In order to be able to use ontologies in KDD process and to have the results available to domain experts, ontologies have to be defined in a common way. There are a lot of efforts for ontology matching (Doan, Madhavan, Domingos, & Halevy, 2003) and ontology integration (Cui, Jones, & O'Brien, 2002), (Pinto & Martins, 2001) and this illustrates the need for an ontology creation standard. In this way, exchange and comparison of ontologies describing different domains could be possible. Until now, only several domain specific ontologies and tools have been developed.

6.3.5 Extending PatternMiner prototype to support pattern evaluation

The extended system we propose provides both naïve users and domain experts functionalities for efficient pattern management and pattern evaluation using an ontology discarding the non-useful patterns and thus improving the performance of the data mining tasks and the query answering over the pattern base. The system is able to evaluate patterns before, during and after the data mining process, as well as every time user poses a query to the pattern base. The system architecture is depicted in Figure 6-8. The ontology validation extension is not integrated into the PatternMiner system, and thus it is shown separately.

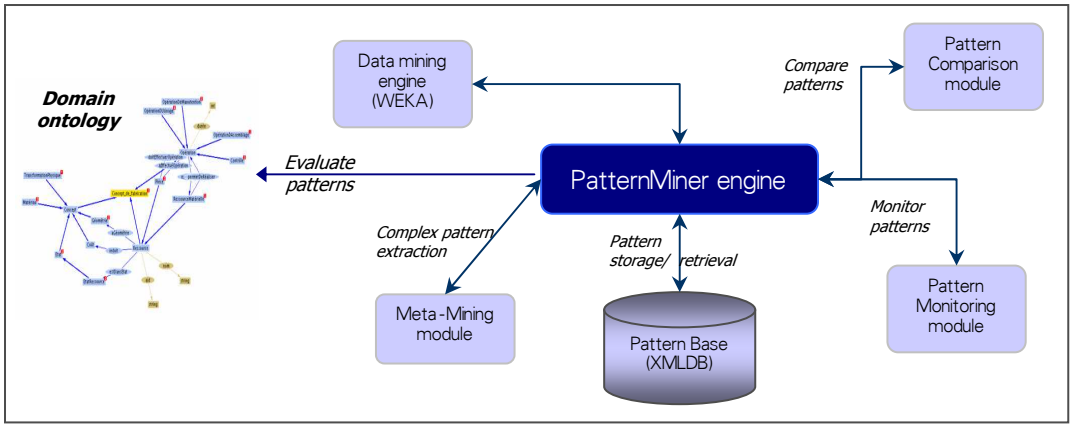


Figure 6-8 The proposed ontology-enhanced PBMS architecture

Independent from data mining engine, the PBMS stores the extracted patterns in an XML pattern base. The pattern model used is enhanced to support pattern temporal validation and semantically related pattern classes. Our extended model defines four logical concepts. *Pattern type*, *pattern*, *class* and *superclass.*

More specifically, each *pattern type* contains metadata information about:

- the data mining algorithm applied to extract the patterns it represents and its parameters,
- the date and time of the data mining process,
- the validity period,
- the data source,
- the mapping function, and finally,
- information about the structure and the measures of the patterns it represents.

Patterns are instances of pattern types. In our XML architecture, pattern types are the XML Schema for a pattern (XML document). At this point notice that in the current application, a more PANDA-specific XML schema is used instead of an enhanced PMML model to represent patterns. This shows that our system does not restrict users to use PMML documents, but every well-formed pattern schema that has the basic requirements of the PANDA model, can be used.

The pattern document contains metadata about the data mining process as well as the patterns extracted by that process. For example, an association

rule pattern instance and its pattern type are shown in Figure 6-9 and Figure 6-10, respectively.

```
<pt_assocRule xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" name="assocRule"
pt_descr="association rules" pt_id="1" xsi:noNamespaceSchemaLocation="pt_assocRule.xsd">
        <pt_metadata>
                <algorithm>apriori</algorithm>
        <parameters>min_support=0.1,min_conf=0.4,rules=10</parameters>
                <source>select * from earthquakes</source>
                <date>2006/04/12 13:03:34</date>
                <validity>2006/06/12 13:03:34</validity>
                <mapping_function>{{'depth', 'magnitude', 'season'} ⊆ transaction}
</mapping_function>     </pt_metadata>
        <patterns>
                <pattern p_id="1">
                        <structure>
                                <body>
                                        <attrib>depth</attrib>
                                        <attrib_value>0-1</attrib_value>
                                </body>
                                <head>
                                        <attrib>magnitude</attrib>
                                        <attrib_value>(3,4]</attrib_value>
                                </head>
                        </structure>
                        <measures>
                                <measure_name>support</measure_name>
                                <measure_value>0.18</measure_value>
                                <measure_name>confidence</measure_name>
                                <measure_value>0.67</measure_value>
                        </measures>
                </pattern>
                <pattern p_id="2">
                        <structure>
                                <body>
                                        <attrib>season</attrib>
                                        <attrib_value>Autumn</attrib_value>
                                </body>
                                <head>
                                        <attrib>magnitude</attrib>
                                        <attrib_value>(3-4]</attrib_value>
                                </head>
                        </structure>
                        <measures>
                                <measure_name>support</measure_name>
                                <measure_value>0.18</measure_value>
                                <measure_name>confidence</measure_name>
                                <measure_value>0.58</measure_value>
                        </measures>
                </pattern>      </patterns> </pt_assocRule>
```

Figure 6-9 Association rule patterns, XML example

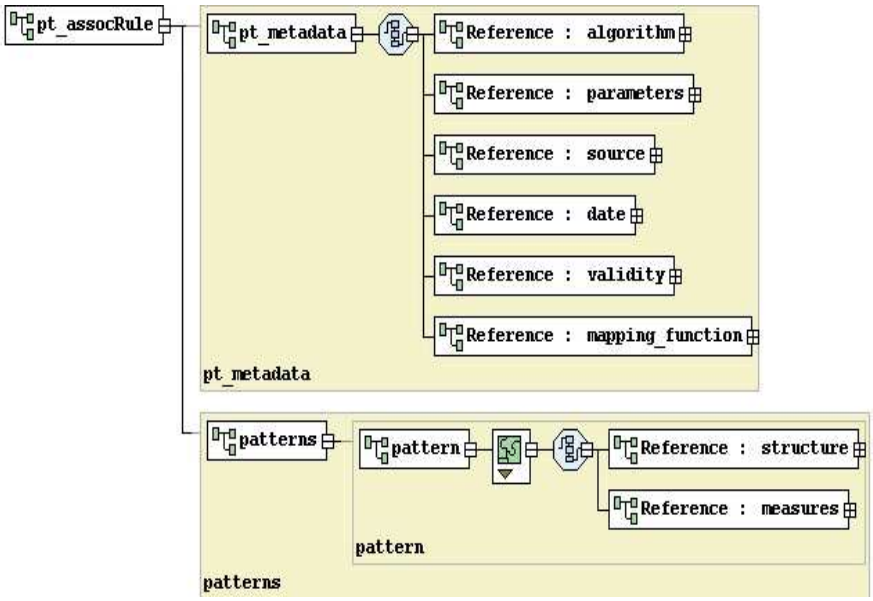


Figure 6-10 Pattern Type Association Rule XSD diagram

Apart from the *pattern type* and *pattern* concepts, *class* is defined as a set of semantically related patterns of the same pattern type. A class is defined by the user to group patterns that have a common meaning and belong to a specific pattern type. Each pattern may belong to more than one different class. For example a user could define a class containing association rules related to seismic activity in the summer of 2003. This class would contain a lot of patterns that may belong to different association rule mining result sets but it will have the same meaning for the user. Figure 6-11 illustrates the pattern base logical model.

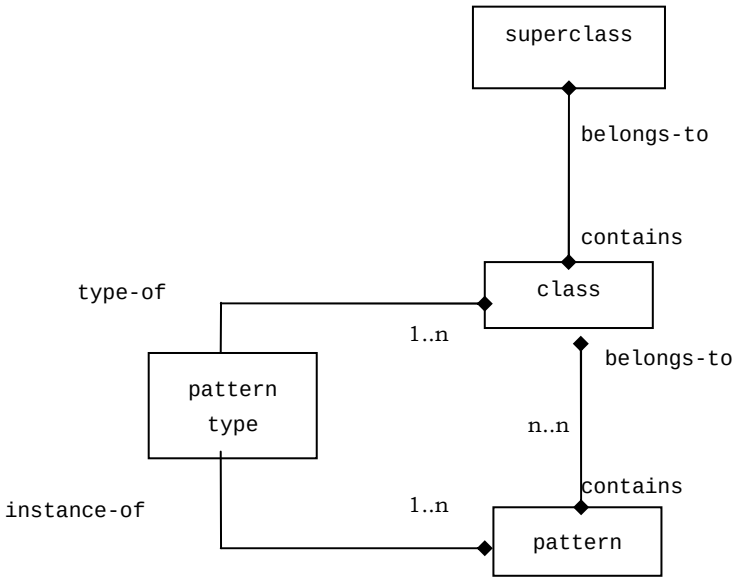


Figure 6-11 Pattern Base logical model

Furthermore, the concept of *superclass* is defined, which is a set of classes, probably of different pattern types. Thus, patterns belonging to different pattern types can be grouped together. For instance, a user might want to group all association rules related to seismic activity in the summer of 2003 and the clusters of faults that gave earthquakes of magnitude M>3 during the same time period. The link between the two types would be the magnitude of earthquakes. In other words, we are interested in studying the relation between earthquakes and geological faults, thus the grouping of classes of different pattern types is necessary.

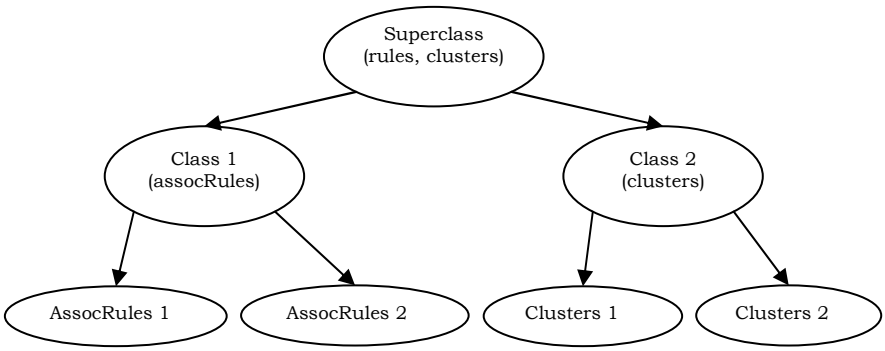


Figure 6-12 Class and Superclass relation

Ontologies are stored in external files and are written in OWL (Horrocks & Patel-Schneider, 2003).

Since the ontology represents the domain of interest, it has to be well designed. In this way, pattern evaluation can be more accurate and may give useful results to domain experts.

Using PatternMiner and domain ontologies the whole KDD process is covered. The extracted patterns can be evaluated before storing them in the pattern base (or in any other time later), where query posed by the user can be evaluated before running in the pattern-base. Queries containing concepts that according to the ontology are irrelevant could be discarded – notifying - the user decreasing in this way the processing time and load of the system.

6.4 Synopsis

In this chapter we presented PatternMiner an open source Pattern Base Management System, PatternMiner, which integrates data mining tasks with pattern storage and management providing advanced pattern operations such as pattern comparison and pattern monitoring over time. PatternMiner consists of the following components: WEKA data mining engine, Oracle XMLDB pattern base and PANDA comparison, metamining and monitoring framework. We described the system architecture and its components. We presented a short demo of the system with a its basic functions providing some of the system screenshots.

Finally we presented a way to extend PBMS concept an PatternMiner prototype to incorporate domain knowledge through the use of ontologies, in order to integrate the pattern evaluation step of the KDD process providing the end-users a more powerful tool of pattern extraction, manipulation and evaluation.

As shown from the application scenarios, PatternMiner can provide an integrated environment for extracting, storing and comparing patterns of every kind, saving valuable time from the experts.

7 Conclusions

In this chapter we summarize the contributions of this thesis and we suggest topics for future work.

7.1 Thesis Contributions

Due to the huge amount of data that is nowadays collected and stored in databases in every scientific or commercial domain, data mining applications and techniques are used more often in order to discover hidden information, groups and associations. The amount of patterns though that are extracted from these databases are also huge and in a lot of cases its management is not an easy task. End users cannot cope with all the different type of patterns that are produced with a variety of software over heterogeneous data sources.

Facing this challenge, we deal with the management of patterns in a Pattern Base Management System (PBMS). A PBMS treats patterns the way a DBMS treats raw data and using a pattern-base and a pattern-representation specific query language. Patterns are rich in semantics compact representation of raw data and they can be simple or complex (defined over simple patterns). The variety of existing pattern types is big but all patterns share common characteristics in the way that they are defined. The unified pattern management along with advanced operations over patterns, such as pattern comparison, results in a variety of interesting applications. In particular, cluster comparison, in the case of this thesis, can be used to classify or retrieve images in the context of a Content-based image retrieval system.

Another important issue when dealing with patterns that have automatically extracted using data mining techniques is their evaluation, as not all patterns extracted are important or interesting to the end-users. The evaluation of extracted patterns is an important but also difficult task. Although, using a unified pattern management system along with domain knowledge ontologies this task can be supported by software.

In this thesis, we dealt with the above issues of pattern management. In particular:

- We studied the most proper representation model for a pattern-base, based on the pattern definition of the PANDA project. Through a qualitative evaluation of three models, the relational, the object-relational and the semi-structured (XML) model, we concluded that the XML model is more appropriate for a pattern-base as it is very extensible and provides query effectiveness among other characteristics.
- We dealt with the comparison of crisp clusters, defining new similarity measures for density-based clusters, produced by the EM algorithm.. We defined a methodology for the comparison of various types of data/objects (e.g. images), that includes four steps; feature extraction from raw data, clustering of the extracted features, Pattern Instantiation and Computation of Pattern Similarities. We evaluated these measures and methodology in content-based image retrieval applications with very satisfactory results.
- We dealt also with the comparison of Fuzzy Clusters and in particular with intuitionistic fuzzy clustering. More specifically, we introduced a novel variant of the Fuzzy C-Means (FCM) clustering algorithm that copes with uncertainty in the localization of feature vectors due to imprecise measurements and noise and a novel similarity measure between intuitionistic fuzzy sets, which is appropriately integrated in the clustering algorithm. We also introduced an intuitionistic fuzzy representation of color digital images as a paradigm of intuitionistic fuzzification of data. To evaluate our approach, we described an intuitionistic fuzzification of color digital images upon which we applied the proposed scheme. The experimental evaluation of the proposed scheme shows that it can be more efficient and more

effective than the well established FCM algorithm, especially as the number of clusters increases, opening perspectives for various applications.

- We presented PatternMiner, an open-source PBMS prototype, that provides an integrated environment for pattern management using modules for pattern extraction, storage, retrieval and comparison. WEKA data mining tool has been chosen to provide the algorithms for pattern extraction. Enhanced PMML XML documents are used to store patterns in the XMLDB pattern-base and the extended PANDA comparison framework is used to support operations based on pattern comparison.
- We studied the use of ontologies to support the pattern evaluation step of the Knowledge Discovery process. Ontologies represent the domain knowledge and in this way can be used to evaluate patterns extracted with data mining algorithms. We proposed a methodology to support this task and we presented a preliminary validation study, while we analyzed the way that PatternMiner prototype can be extended to support the ontology-based pattern evaluation process.

7.2 Future work

Dealing with issues such as pattern management, pattern comparison and evaluation, various research challenges arise. More specifically:

- Future perspectives of the work presented in 3, in combination with the pattern evaluation scheme, include the integration of the proposed scheme with ontology-based information extraction and data mining techniques for the retrieval of medical images using heterogeneous data sources. By storing the semantically rich patterns along with low-level features in a unified way according to the PANDA framework will enable the extension of the CBIR methodologies with knowledge representation techniques for semantic processing and analysis.
- Future perspectives of the work presented in 4 include the systematic evaluation of the proposed scheme in comparison with other clustering schemes for the clustering of various kinds of datasets after appropriately representing them in terms of intuitionistic fuzzy sets

theory; it is worth noting that currently, no reference intuitionistic fuzzy dataset is available to benchmarking clustering algorithms. A challenging issue is the enhancement of the proposed clustering scheme so as to take into account not only the membership, but also the non-membership of each data vector to a cluster.

- Regarding the PatternMiner prototype, new components can be added, like a visualization module for better interpretation of the results or a pattern monitoring module for monitoring and change detection over patterns extracted from a dynamic population.

- Integrating ontologies to the data mining process is not an easy task and a lot of issues have to be addressed. Things are complicated due to the fact that scientists and companies create ontologies according to their needs instead of adopting a universal ontology. There is a large number of ontology languages most of them designed for the semantic web like RDF (Beckett, 2004), SHOE (Luke & Heflin, 2000), DAML, DAML+OIL (Harmelen et al., 2001), OWL (McGuinness, & Harmelen, 2005). New ontologies are constructed for various fields and applications without centralized guidance and common agreement. This is getting even more complex as recent studies have indicated semantic and syntactic conflicts between these languages, especially between DAML+OIL and OWL (Horrocks & Patel-Schneider, 2003) (Patel-Schneiderand & Fensel, 2002). Therefore, building a system that uses ontologies in the data mining process requires choosing a specific ontology language to support.

- Another important theoretical issue concerns the evaluation of various pattern types using ontologies. It is very hard to define general rules that apply to all pattern types. We have defined filters for association rule mining but depending on the application, filters for each pattern type separately have to be defined in order to build a system to support the majority of pattern types.

8 References

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data (pp. 207-216). Washington, DC: ACM.

Ardizzone E., Chella A., Pirrone R., Gambino O., (2004). An image retrieval system for artistic database on cultural heritage. Atti Conferenza Italiana sui Sistemi Intelligenti, (CISI 2004), Perugia, Italia, 2004.

Atanassov, K.T. (1986). Intuitionistic fuzzy sets. Fuzzy Sets and Systems, Vol. 20, pp. 87–96.

Atanassov, K.T. (1989). More on intuitionistic fuzzy sets. Fuzzy Sets Systems, Vol. 33, pp. 37–45.

Atanassov, K.T. (1994). New operations defined over the intuitionistic fuzzy sets. Fuzzy Sets and Systems, Vol. 61, pp. 137–142.

Atanassov, K.T. (1994). Operators over interval valued intuitionistic fuzzy sets. Fuzzy Sets Systems, Vol. 64, pp. 159–174.

Atanassov, K.T. (1999). Intuitionistic Fuzzy Sets: Theory and Applications. Studies in Fuzziness and Soft Computing, Vol. 35, Physica-Verlag, Heidelberg.

Bartolini I., Ciaccia P., Ntoutsi I., Patella M., and Theodoridis Y. (2004). A Unified and Flexible Framework for Comparing Simple and Complex Patterns. In Proceedings of 8th Eur. Conf. on Principles and Practice of Knowledge Discovery in Database, PKDD'04, Pizza, Italy. pp. 496-499, 2004.

Beckett, D. (2004). RDF/XML Syntax Specification (Revised), W3C Recommendation, 2004. http://www.w3.org/TR/rdf-syntax-grammar/.

Berman A., and Shapiro L. G. (1997). Efficient Image Retrieval with Multiple Distance Measures. In Proceedings of SPIE Conf. on Storage and Retrieval for Image and Video Databases, pp. 12-21, 1997.

Bezdek, J.C., Ehrlich, R., and Full, W. (1984). FCM: the Fuzzy c-Means clustering algorithm. Computers and Geosciences, Vol. 10, pp. 191-203.

Bezdek, J.C. (1981). Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, NewYork, 1981.

Bongard, F.S., Sue, D.Y. (2002). Current Critical Care: Diagnosis and Treatment 2nd Ed. McGraw-Hill/Appleton and Lange, 2002.

CINQ (Consortium on Discovering Knowledge with Inductive Queries). (2001). http://www.cinq-project.org.

CWM (Common Warehouse Model) (2001) homepage. http://www.omg.org/cwm.

Cai, W., Feng, D. D., Fulton, R. (2000). Content Based Retrieval of Dynamic PET Functional Images. IEEE Trans. Inf. Tech. Biomed., vol. 4, no. 2, pp. 152-158, 2000.

Carson, C., Belongie, S., Greenspan, H., and Malik, J. (2002). Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, no.8, pp. 1026-1038, 2002.

Caruana R., Elhawary, M., Nguyen, N., and Smith, C. (2006). Meta Clustering. In Proceedings of ICDM 2006.

Catania B., & Maddalena A. (2006). Pattern Management: Practice and Challenges. In J. Darmont & O. Boussaid, (Eds). Processing and Managing Complex Data for Decision Support. Idea Group Publishing.

Catania B., Maddalena A., & Mazza M. (2005). PSYCHO: A Prototype System for Pattern Management. In Proceedings of the International Conference on Very Large Data Bases 2005.

Chen C.C, Wactlar H., Wang J., and Kiernan K. (2004). Digital Imagery for Significant Cultural and Historical Materials - An Emerging Research Field Bridging People, Culture, and Technologies. Int. J. Digital Libraries, 2004.

Chen X., Zhou X., Scherl R., & Geller J. (2003). Using an interest ontology for improved support in rule mining. In DaWaK 2003. pp. 320-329.

Chen, S.M. (1995). Measures of similarity between vague sets. Fuzzy Sets Systems, Vol. 74 No. 2, pp. 217–223.

Chen, S.M. (1997). Similarity measures between vague sets and between elements. IEEE Trans. Syst. Man Cybernet, Vol. 27, No. 1, pp. 153–158.

Chumsamrong, W., Thitimajshima, P. and Rangsanseri, Y. (2000). Synthetic Aperture Radar (SAR) Image Segmentation Using a New Modified Fuzzy C-Means Algorithm. Geoscience and Remote Sensing Symposium, IGARSS 2000. IEEE 2000 International, Vol. 2, pp. 624 – 626.

Cohen, J. (1988). Statistical power analysis for the behavioral sciences (2nd ed.). Hillsdale, NJ: Lawrence Earlbaum Associates.

Cristianini N., Shawe-Taylor J. (2000). An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, (2000).

Cui Z., Jones D., & O'Brien P. (2002). Semantic B2B Integration: Issues in Ontology-based Approaches. ACM SIGMOD Record archive Vol. 31, Issue 1 (March 2002) SPECIAL ISSUE: Data management issues in electronic commerce table of contents. pp. 43–48

DMG - PMML, http://www.dmg.org/pmml-v3-1.html.

Davis, J. and Goadrich, M. (2006). The Relationship Between Precision-Recall and ROC Curves. In Proceedings of 23rd Int. Conf. on Machine Learning (ICML), Pittsburgh, USA, 2006, pp. 233-240.

Dempster, A., Laird, N. and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B, vol. 39, no.1, pp.1–38, 1977.

Dengfeng, L., Chuntian, C. (2002). New similarity measure of intuitionistic fuzzy sets and application to pattern recognitions. Pattern Recognition Letters, Vol. 23, pp. 221–225.

Deselaers T., Keysers D., and Ney H. (2004). FIRE - Flexible Image Retrieval Engine: ImageCLEF 2004 Evaluation. LNCS 3491, pp. 688-698, 2004.

Deselaers, T., Keysers, D., Ney, H. (2004). Features for Image Retrieval-A Quantitative Comparison. In Proceedings of 26th DAGM Symp., LNCS, pp. 228-236, 2004.

Doan, A., Madhavan, J., Domingos, P., & Halevy, A. (2003). Ontology Matching: A Machine Learning Approach. In S. Staab and R. Studer (Eds), Handbook on Ontologies in Information Systems. Springer-Velag.

Dowd, S.B., Wilson, B.G. (1995). Encyclopedia of Radiographic Positioning: Volume 2, Saunders, 1995.

Dunn, J.C. (1973). A Fuzzy Relative of the ISODATA process and its Use in Detecting Compact Well-Separated Cluster. Journal Cybernetics Vol. 3, No. 3, pp. 32-57.

El-Naqa, I., Yang, Y., Galatsanos, N.P., Nishikawa, R.M., Wernick, M.N. (2004). A similarity learning approach to content-based image retrieval: application to digital mammography. IEEE Trans. Med Imaging, vol. 23, no. 10, pp. 1233-1244, Oct. 2004.

FCD/fcd-datamining-2001-05.pdf, 2001.

FHW, Foundation of the Hellenic World, http://www.fhw.gr/index_en.html.

Faloutsos C., Barber R., Flickner M., Hafner J., Niblack W., Petkovic D., and Equitz W., (1994), Efficient and Effective Querying by Image Content, J. Intell. Inf. Systems, vol. 3, pp. 231-262, 1994.

Fan, L., Zhangyan, X. (2001). Similarity measures between vague sets. J. Software Vol. 12, No.6, pp. 922–927 (in Chinese).

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery, an overview. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, (Eds.), Advances in Knowledge Discovery and Data Mining. (pp. 1–30). Menlo Park, Calif. AAAI/MIT Press.

Fayyad, U., Haussler, D., & Stolorz, P. (1996). Mining scientific data. Communications of the ACM, Vol. 39, Issue 11, 51-57.

Freitas, A.A. (1999). On rule interestingness measures. Knowledge-Based Systems, Vol. 12. number 5-6. pp. 309–315, October 1999. Elsevier.

GAIA, ESA project. http://www.esa.int/science/gaia 2009.

Greenspan, H., Pinhas, A.T. (2007). Medical Image Categorization and Retrieval for PACS Using the GMM-KL Framework. IEEE Trans. Inf. Tech. Biomed., vol.11, no.2, pp.190-202, Mar. 2007.

Grosky W.I. and Mehrotra R. (1990). Indexed-Based Object Recognition in Pictorial Data Management. Computer Vision, Graphics, Image Processing, vol. 52, pp. 416-436, 1990.

Gruber, T. R. (1993). A translation approach to portable ontologies. Knowledge Acquisition, 5(2):199-220.

Guarino, N., & Giaretta, P. (1995). Ontologies and knowledge bases: Towards a terminological clarification. In N. Mars (Ed.), Towards very large knowledge bases: Knowledge building and knowledge sharing (pp.25-32). Amsterdam, IOS Press.

Hampapur A., Gupta A., Horowitz B., Shu C. F., Fuller C., Bach J., Gorkani M., and Jain R. (1997). Virage video engine. In Proceedings ofSPIE: Storage and Retrieval for Image and Video Databases V, pp. 188-197, San Jose, February 1997.

Harmelen, F.V., Patel-Schneider, P.F., & Horrocks, I. (2001). Reference Description of the DAML+ OIL Ontology Markup Langauge. http://www.daml.org/2001/03/daml+oil-index.html.

Hartigan, J. A. (1975). Clustering algorithms. New York: John Wiley, 1975.

Hong, D.H., Kim, C. (1999). A note on similarity measures between vague sets and between elements. Inform. Science Vol. 115, pp. 83–96.

Hore, P., Hall, L.O. and Goldgof, D.B. (2007). Single Pass Fuzzy C-means. IEEE International Conference on Fuzzy Systems, London, 2007.

Horrocks, I., & Patel-Schneider, P.F. (2003). Three theses of representation in the semantic web. In Proceedings of the Twelfth International Conference on World Wide Web.

Hotho, A., Maedche, A., Staab, S., & Zacharias, V. (2002). On knowledgeable unsupervised text mining. In Proceedings of the DaimlerChrysler Workshop on Text Mining, Ulm, April 26–27 2002. Springer.

Hung, W.-L., Yang, M.-S. (2004). Similarity measures of intuitionistic fuzzy sets based on Hausdorff distance. Pattern Recognition Lett. Vol. 25, pp. 1603–1611.

ISO SQL/MM Part 6. http://www.sql-99.org/SC32/WG4/Progression_Documents/ (2001)

Iakovidis, D.K., Maroulis D.E., Karkanis S.A. (2005). A Comparative Study of Color-Texture Image Features. In Proceedings of IEEE Int. Workshop on Systems, Signal and Image Processing (IWSSIP), Halkida, Greece, 2005, pp. 205-209.

Iakovidis, D.K., Kotsifakos, E.E., Pelekis, N., Karanikas, H., Kopanakis, I., and Theodoridis, Y. (2007). Pattern-Based Retrieval of Cultural Heritage Images. 11th Panhellenic Conference on Informatics (PCI'2007), Patras, Greece, 2007.

Iakovidis, D.K., Pelekis, N., Karanikas, H., Kotsifakos, E.E., Kopanakis, I., and Theodoridis, Y. (2006). A Pattern Similarity Scheme for Medical Image Retrieval, ITAB 2006, Proc of the 7th Annual IEEE Conf on International Technology Applications in Biomedicine, Ioannina, Greece.

Iakovidis, D.K., Pelekis, N., Karanikas, H., Kotsifakos, E.E., Kopanakis, I., and Theodoridis, Y. (2009). A Pattern Similarity Scheme for Medical Image

Retrieval. IEEE Transactions on Information Technology in Biomedicine Journal, Volume: 13 Issue: 4, July 2009.

Iakovidis, D.K., Pelekis, N., Kotsifakos, E.E. and Kopanakis, I. (2008). Intuitionistic Fuzzy Clustering with Applications in Computer Vision. In the Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS'08), LNCS 5259, pp. 764–774, Juan-les-Pins, France, 2008

Iqbal Q., and Aggarwal J. K. (2002). CIRES: A System for Content-based Retrieval in Digital Image Libraries. In Proceedings of Int. Conf. Control, Automation, Robotics and Vision (ICARCV), pp. 205-210, December 2-5, 2002.

Java Data Mining API homepage (2003), http://www.jcp.org/jsr/detail/73.prt.

Jawahar, C.V., Ray, A. K. (1996). Fuzzy statistics of digital images. Pattern Recognition Letters, Vol. 17, pp. 541–546.

Jeng W.-M., and Hsiao J.-H. (2005). An Efficient Content Based Image Retrieval System Using the Mesh-of-Trees Architecture. J. Inf. Science Eng., vol. 21, pp. 797-808, 2005.

Jia L. and Wang J. Z. (2004). Studying Digital Imagery of Ancient Paintings by Mixtures of Stochastic Models. IEEE Trans. on Image Processing., vol.12, no.2, pp., 2004.

Karkanis, S.A., Iakovidis, D.K., Maroulis, D.E., Karras, D.A. and Tzivras, M. (2003). Computer Aided Tumor Detection in Endoscopic Video using Color Wavelet Features. IEEE Trans. Inf. Tech.Biomed., vol. 7, pp. 141-152, 2003.

Kopanakis, I. and Theodoulidis, B. (2003). Visual Data Mining Modelling Techniques for The Visualization of Mining Outcomes. J. Visual Languages and Computing, Special Issue on Visual Data Mining, vol. 14, no.6, pp. 543-589, 2003.

Kotsifakos, E.E., Marketos, G., and Theodoridis, Y. (2007). A framework for integrating ontologies and pattern-bases. In Nigro, H.O., Cisaro, S.G. and

Xodo, D. (eds), Data Mining with Ontologies: Implementations, Findings,and Frameworks. Idea Group Inc., Hershey, 2007.

Kotsifakos, E.E., Ntoutsi, I. and Theodoridis, Y. (2005). Database Support for Data Mining Patterns. 10th Panhellenic Conference on Informatics (PCI'2005), Volos, Greece, 2005. Advances in Informatics - Springer-Verlag LNCS #3746,2005

Kotsifakos, E.E., Ntoutsi, I., Vrahoritis, Y., and Theodoridis Y. (2008). Monitoring Patterns through an Integrated Management and Mining Tool. In Proceedings of Machine Learning and Knowledge Discovery in Databases, European Conference (ECML/PKDD'08), Antwerp, Belgium, 2008.

Kotsifakos, E.E., Ntoutsi, I., Vrahoritis, Y., and Theodoridis Y. (2008) PATTERN-MINER: Integrated Management and Mining over Data Mining Models. In Proceedings of 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'08), Las Vegas, USA, 2008.

Kwak, D.-M., Kim, B.-S., Yoon, O.-K., Park, C.-H., Won, J.-U., Park, K.-H. (2002). Content-Based Ultrasound Image Retrieval Using a Coarse to Fine Approach. Annals of the New York Academy of Sciences, vol. 980, pp.212-224, 2002.

Laaksonen J., Koskela M., Laakso S., and Oja E. (2000). PicSOM - Content-Based Image Retrieval with Self-Organizing Maps.Pattern Recognition Letters, vol. 21, pp. 1199-1207, 2000.

Lehmann, T.M., Guld, M.O., Thies, C., Plodowski, B., Keysers, D., Ott, B., Schubert, H. (2004). IRMA - Content-based image retrieval in medical applications. In Proceedings of 14th World Congress on Medical Informatics (Medinfo), IOS Press, Amsterdam, vol. 2, pp. 842-848, 2004.

Lehmann, T.M., Schubert, H., Keysers, D., Kohnen, M., Wein, B.B. (2003). The IRMA Code for Unique Classification of Medical Images. In Proc SPIE 2003, vol. 5033, pp. 109–17, 2003.

Li, Y., Olson D.L., Qin Z. (2007). Similarity measures between vague sets: A comparative analysis. Pattern Recognition Letters Vol. 28, pp. 278-285

Li, Y., Zhongxian, C., Degin,Y. (2002). Similarity measures between vague sets and vague entropy. J.Computer Sci. Vol. 29, No.12, pp. 129–132.

Lin, C.-Y., Yin, J.-X., Gao, X., Chen, J.-Y. and Qin, P. (2006). A Semantic Modelling Approach for Medical Image Semantic Retrieval Using Hybrid Bayesian Networks. In Proceedings of 6th Int. Conference on Intelligent Systems Design and Applications (ISDA), pp. 482-487, 2006.

Linacre, J.M. (1996). Overlapping Normal Distributions. Rasch Measurement Transactions, vol. 10, no. 1 pp. 487-488.

Littau, D. (2003). Using Low-Memory Approximations to Cluster Very Large Data Sets. In Proceedings of 3rd SIAM Int. Conf. on Data Mining (SDM), 2003.

Liu, B., Hsu W., Chen S., & Ma, Y. (2000). Analyzing the subjective interestingness of association rules. IEEE Intelligent Systems, 15(5):47-55.

Luke, S., & Heflin, J. (2000). SHOE 1.01 Proposed Specification, SHOE Project, http://www.cs.umd.edu/projects/plus/SHOE/spec.html.

Ma W.Y., and Manjunath B.S. (1999). Netra: A Toolbox for Navigating Large Image Databases,.Multimedia System, vol. 7, pp. 184-198, 1999.

MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. In Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297.

Maedche, A., Motik, B., Stojanovic, L., Studer, R., & Volz, R. (2003). Ontologies for enterprise knowledge management. IEEE Intelligent Systems, 18(2):26–33, March/April 2003.

Maenpaa, T., and Pietikainen, M. (2004). Classification with color and texture: Jointly or separately?, Pattern Recognition, Vol. 37, No. 8, pp. 1629–1640.

Mallat, S. (1999). A Wavelet Tour of Signal Processing. Acad. Press, 2nd ed., 1999.

Maroulis, D.E., Savelonas, M., Iakovidis, D.K., Karkanis, S.A., Dimitropoulos, N. (2007). Variable Background Active Contour Model for Computer-Aided Delineation of Nodules in Thyroid Ultrasound Images. IEEE Trans. Inf. Tech. Biomed., vol. 11, no. 5, pp. 537-543, 2007.

McGuinness, D.L., & Harmelen, F.V. (2005). OWL Web Ontology Language Overview. http://www.w3.org/TR/owl-features/ (current Feb. 2005).

Mitchell, H.B. (2003). On the Dengfeng–Chuntian similarity measure and its application to pattern recognition. Pattern Recognition Lett. Vol. 24, pp. 3101–3104.

Muller, H., Michoux, N., Bandon, D., Geissbuhler, A. (2004). A Review of Content-Based Image Retrieval Systems in Medicine - Clinical Benefits and Future Directions. Int.J.of Med.Informatics, vol.73, pp.1-23, 2004.

Niles, I., & Pease, A. (2001). Toward a Standard Upper Ontology. In Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001).

Niles, Ian & Terry, Allan. (2004). The MILO: A general-purpose, mid-level ontology. In 2004 International Conference on Information and Knowledge Engineering (IKE'04).

Ntoutsi, I. (2008). Similarity Issues in Data Mining – Methodologies and Techniques. PhD thesis, University of Piraeus, June 2008.

Ntoutsi, I., Pelekis, N., and Theodoridis, Y. (2007). Pattern Comparison in Data Mining: a survey. In D.Taniar, editor, Research and Trends in Data Mining Technologies and Applications (Advances in Data Warehousing and Mining), pages 86 – 120. Idea Group Publishing, 2007.

Ohta, Y., Kanade, T., Sakai, T. (1980). Color information for region segmentation. ComputerVision, Graphics, and Image Processing, Vol. 13, pp. 222-241.

Ojala T., Pietikainen M., Harwood D. (1996). A Comparative Study of Texture Measures with Classification based on Feature Distributions. Pattern Recognition, vol. 29, 1996, pp. 51-59.

Omelayenko B. (2000). Integration of Product Ontologies for B2B Marketplaces: A Preview. In ACM SIGecom Exchanges. Vol. 2, issue 1, pp. 19-25.

Oracle Corp. Berkeley DB XML. Available at http://www.oracle.com/database/berkeley-db/xml/index.html

PANDA (Patterns for Next-generation Database Systems). (2001). project homepage. http://dke.cti.gr/panda.

PBMS (2006) homepage. http://www.pbms.org.

PMML (Predictive Model Markup Language) (2009). http://www.dmg.org/v4-0/GeneralStructure.html.

PQL, Information Discovery Data Mining Suite. http://www.patternwarehouse.com/ dmsuite.htm.

Padmanabhan B. & Tuzhilin, A. (1998). A belief-driven method for discovering unexpected patterns. In Proceedings of the International Conference on Knowledge Discovery and Data Mining, pages 94–100, August 1998.

Patel-Schneiderand, P.F., & Fensel, D. (2002). Layering the semantic web: Problems and Directions. In Proceedings of the 1st International Semantic Web Conference. LNCS 2342, Springer.

Pelekis, N., Iakovidis, D.K., Kotsifakos, E.E, Karanikas, H., and Kopanakis, I. (2007). Intuitionistic Fuzzy Clustering to Information Retrieval from Cultural Databases. 22nd European Conference on Operational Research, EURO XXII, Prague, 2007.

Pelekis, N., Iakovidis, D.K., Kotsifakos, E.E. and Kopanakis, I. (2008). Fuzzy Clustering of Intuitionistic Fuzzy Data. International Journal of Business Intelligence and Data Mining, 3(1), 45-65, 2008

Petrakis E. G.M., Faloutsos C. (1997). Similarity Searching in Medical Image Databases. IEEE Trans. Knowl. Data Eng., vol. 9, no. 3, pp. 435-447, 1997.

Piatetsky-Shapiro G. & Matheus C.J. (1994). The interestingness of deviations. In Proceedings of KDD-94: AAAI-94 Knowledge Discovery in Databases Workshop, pages 25–36. AAAI Press, July 1994.

Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. In G. Piatetsky-Shapiro & W.J. Frawley (Eds). Knowledge Discovery in Databases, pp. 229–248. AAAI/MIT Press, Cambridge, MA.

Piatetsky-Shapiro, G. (2000). Knowledge discovery in databases: 10 years after. SIGKDD Explorations, Vol. 1, no 2. pp. 59–61, January 2000.

Pinto, H.S., & Martins, J.P. (2001) A methodology for ontology integration. 1st International conference on Knowledge Captur. Pp 131-138.

Pohle, C. & Spiliopoulou, M. (2002). Building and exploiting ad hoc concept hierarchies for web log analysis. In Y. Kambayashi, W. Winiwarter, & M. Arikawa (Eds). Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery, DaWaK 2002, volume 2454 of Lecture Notes in Computer Science, pages 83–93, Aix en Provence, France, September 4–6 2002. Springer-Verlag.

Pohle, C. (2003). Integrating and updating domain knowledge with data mining. VLDB PhD Workshop.

R-project, (2009). The R-tool. http://www.r-project.org/ 2009

Rahman, Md. M., Bhattacharya, P. and Desai, B.C. (2007). A Framework for Medical Image Retrieval Using Machine Learning and Statistical Similarity Matching Techniques With Relevance Feedback. IEEE Trans. Inf. Tech. Biomed., vol. 11, no. 1, pp. 58-67, Jan. 2007.

Rizzi, S., Bertino, E., Catania, B., Golfarelli, M., Halkidi, M., Terrovitis, M., Vassiliadis, P., Vazirgiannis, M., & Vrahnos, E. (2003). Towards a logical model for patterns. Proceedings of ER'03 conference, Chicago, IL, USA, 2003..

Ruspini, E. H. (1969). A New Approach to Clustering. Information Control Vol. 15, No. 1, pp. 22-32.

SUMO, Suggested Upper Merged Ontology (2009) http://www.ontologyportal.org/.

Schnorrenberg, F., Pattichis, C. S., Schizas, C. N., Kyriacou, K. (2000). Content-Based Retrieval of Breast Cancer Biopsy Slides. Technology and Health Care, vol. 8, pp. 291-297, 2000.

Seismo-Surfer, A WebGIS application for integrating, visualizing and analyzing seismic data. http://www.seismo.gr.

Setia, L., Teynor, A., Halawani, A. and Burkhardt, H. (2006). Image Classification using Cluster-Cooccurrence Matrices of Local Relational Features. In Proceedings of 8th ACM Int. Workshop on Mult. Inf. Retrieval, 2006.

Shyu, C. R., Brodley, C. E., Kak, A. C., Kosaka, A., Aisen, A. M., and Broderick, L. S. (1999). ASSERT: A Physician-in-the-loop Content-Based Image Retrieval System for HRCT Image Databases. Computer Vision and Image Understanding, pp. 111-131, 1999.

Silberschatz A. & Tuzhilin, A. (1996). What makes patterns interesting. In knowledge discovery systems. IEEE Transactions on Knowledge and Data Engineering, 8(6):970–974, December 1996.

Smith J.R., and Chang S.F. (1996). Visualseek: A Fully Automated Content-Based Image Query System. In Proceedings of ACM Int'l Multimedia Conf., pp. 87-98, 1996.

Spiliopoulou M., Ntoutsi, I., Theodoridis, Y., and Schult, R. (2006). MONIC: Modelling and monitoring cluster transitions. KDD, 2006.

Spiliopoulou M., and Roddick J. F. (2000). Higher Order Mining: Modellingand Mining the Results of Knowledge Discovery. In Proceedings of Int. Conf. Data Mining, vol. 2, pp. 309-320, 2000.

Stehling R.O., Falcao A.X., and Nascimento M.A. (2001). An Adaptive and Efficient Clustering-Based Approach for Content-Based Image Retrieval in Image Databases. In Proceedings of Int. Symp. Database Eng. & App. (IDEAS 01), pp. 56-365, 2001.

Stehling R.O., Nascimento M.A., and Falcao A.X. (2002). A Compact and Efficient Image Retrieval Approach based on Border/Interior Pixel Classification. In Proceedings of 11th Int. Conf. Inf. Knowledge Man. (CIKM'02), ACM Press, NY, pp. 102-109.

Stone, M. (1974). Cross-validation choice and assessment of statistical predictions. Journal of the Royal Statistical Society, B-36, 111–147 1974.

Stonebraker, M. (1997). Object-Relational DBMS: The Next Wave. Informix Software, CA Feb, 1997.

Stonebraker, M., Brown, P., and Moore, D. (1999). Object-Relational DBMSs: Tracking the Next Great Wave 2e. Morgan-Kauffman Publishers. San Francisco, 1999.

Swain, M.J. and Ballard, D.H. (1991). Color Indexing. Int. J. Computer Vision, Vol. 7, No. 1, pp. 11–32, Nov. 1991.

Terrovitis , P., Skiadopoulos, S., Bertino, E., Catania, B., Maddalena, A., and Rizzi, S. (2007). Modeling and language support for the management of pattern-bases, Data Knowl. Eng. 62, 2 (Aug. 2007).

Terrovitis M., Vassiliadis P., Skiadopoulos S., Bertino E., Catania B., Maddalena A. (2004). Modelling and Language Support for the Management of Pattern-Bases. In Proceedings of SSDBM Conference, Santorini, Greece, 2004.

Theodoridis, S. and Koutroumbas K. (2006). Pattern Recognition, Elsevier.

Theodoridis, Y., Marketos, G., & Kalogeras, I.S. (2004). Collecting and Mining Seismic Data in Greek Territory - The Seismo-Surfer Tool. In Proceedings of 7th Panhellenic Geographical Conference of the Hellenic Geographical Association (HGA'04), Mytilene, Lesvos, Greece.

Theodoridis, Y., Vazirgiannis, M., Vassiliadis, P., Catania, B., and Rizzi, S. (2003) A Manifesto for Pattern Bases, PANDA TR-2003-03. Available at http://www.pbms.org/papers/TR-2003-03.pdf

Thitimajshima, P. (2000). A New Modified Fuzzy C-Means Algorithm for Multispectral Satellite Images Segmentation. Geoscience and Remote

Sensing Symposium, IGARSS 2000. IEEE 2000 International, Vol. 4, pp. 1684 – 1686.

Valle E., Cord M., and Philipp-Foliguet S. (2006). Content-Based Retrieval Of Images For Cultural Institutions Using Local Descriptors, Int. Conf. on Geometric Modeling and Imaging, London, 2006.

Vazirgiannis M., Halkidi M., Tsatsaronis G., Vrachnos E. (2003). A Survey on Pattern Application Domains and Pattern Management Approaches. PANDA Technical Report TR-2003-01, 2003. Available at http://dke.cti.gr/panda.

Veltcamp R., and Tanase M. (2000). Content--Based Image Retrieval Systems: A Survey. Tech. Report UU-CS-2000-34, Dept. of Comp. Sci., Utrecht Univ., 2000.

Vlachos, I.K. and Sergiadis G.D. (2005). Towards Intuitionistic Fuzzy Image Processing. Proceedings of the International Conference on Computational Intelligence for Modelling. Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vol. 1, pp. 2-7.

Vlachos, I.K. and Sergiadis, G.D. (2006). Intuitionistic fuzzy information – Applications to pattern recognition. Pattern Recognition Letters, Vol. 28, pp. 197–206.

Wang J.Z., Li J., and Wiederhold G. (2001). SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries. IEEE Trans. Patt. Anal. Machine Intell., vol. 23, no. 9, pp. 947-963, Sept. 2001.

Wang, J.Z., Wiederhold, G., Firschein, O. and Wei, S.X. (1998). Content-Based Image Indexing and Searching Using Daubechies' Wavelets. Int. Journal of Digital Libraries, vol. 1, no. 4, pp. 311–328, 1998.

Wang, J.Z. (2001). Wavelets and imaging informatics: A review of the literature. Jour. of Biomedical Informatics, vol. 34, pp. 129-141, 2001.

Weber, R., Schek, H.-J. and Blott, S. (1998). A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional

Spaces. In Proceedings of Very Large Data Bases Conference (VLDB), pp. 194-205 1998.

Witten I. H., and Frank E. (2005). Data Mining: Practical machine learning tools and techniques. 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

Wu J.K., and Narasimhalu A.D. (1994). Identifying Faces Using Multiple Retrievals. IEEE Multimedia, vol. 1, no. 2, pp. 20-38, 1994.

XQuery 1.0 An XML Query Language. *XQuery 1.0 An XML Query Language*. W3C Working Draft 12 November 2003. http://www.w3.org/TR/2003/WD-xquery-20031112

Yaoa, J., Antani, S., Longb, R., Thoma, G. and Zhanga, Z. (2006). Automatic Medical Image Annotation and Retrieval Using SECC. In Proceedings of 19th Int. Symp. Computer-Based Medical Systems (CBMS), Utah, June 2006.

Yixin Chen, Wang, J.Z., Krovetz, R. (2005). CLUE: Cluster-Based Retrieval of Images by Unsupervised Learning. IEEE Trans. on Image Processing, vol. 14, no. 8, pp. 1187- 1201, Aug. 2005.

Yong, Y., Chongxun, Z., Pan, L. (2004). A Novel Fuzzy C-Means Clustering Algorithm for Image Thresholding. Measurement Science Review, Vol. 4, Sec. 1.

Zadeh, L.A. (1965). Fuzzy sets. Information Control Vol. 8, pp. 338–356.

Zhang R., Zhang Z.M. (2002). A Clustering Based Approach to Efficient Image Retrieval. In Proceedings of 14th IEEE Int. Conf. Tools Artif. Intel. (ICTAI'02), p. 339, 2002.

Zheng, L., Wetzel, A.W., Gilbertson, J., Becich, M.J. (2003). Design and Analysis of a Content-Based Pathology Image Retrieval System IEEE Trans.Inf, Tech. Biomed., vol.7, no.4, pp.249-255, 2003.

Zhizhen, L., Pengfei, S. (2003). Similarity measures on intuitionistic fuzzy sets. Pattern Recognition Lett. Vol. 24, pp. 2687–2693.