Applying Fuzzy Data Mining for Soaring Area Selection

A. Salguero[1], F. Araque[1], R.A. Carrasco[1], M.A. Vila[2], and L. Martínez[3]

[1] Dpto. Lenguajes y Sistemas Informáticos, ETSIIT
Universidad de Granada (Andalucía), Spain
[2] Dpto. Ciencias de la Computación e IA, ETSIIT
Universidad de Granada (Andalucía), Spain
[3] Dpto. de Informática, EPS
Universidad de Jaén (Andalucía), Spain

Abstract. Soaring is a recreational activity and competitive sport where individuals fly un-powered aircrafts known as gliders. Soaring place selection process depends on a number of factors, resulting in a complex decision-making task. In this paper, we propose the use of the dmFSQL language for fuzzy queries as one of the techniques of Data Mining, which can be used to solve the problem of offering the better place for soaring given the environment conditions and customer characteristics. After doing a process of clustering and characterization of a Customers Database in a Data Warehouse we are able of classify next customer in a cluster and offer an answer according it.

1 Introduction

We can define *Data Mining* (DM) as the process of extraction of interesting information from the data in databases. According to [6] a discovered knowledge is interesting when it is novel, potentially useful and non-trivial to compute. A series of new functionalities exist in DM, which reaffirms that it is an independent area [6]: high-level language on the discovered knowledge and for showing the results of the user's requests for information (e.g. queries); efficiency on large amounts of data; handling of different types of data; etc. In this paper we discuss the implementation of two prototypes for Data Mining purposes: we have used a combination of DAPHNE which was initially designed for clustering on numeric data types [2] and *dmFSQL* which was designed for fuzzy (or flexible) queries [4].

Soaring is a recreational activity and competitive sport where individuals fly un-powered aircrafts known as gliders. The selection of the best zone to fly is directly related to pilot's skill. Soaring pilots get their lift from atmospheric instability. The more instability, the more height gained. The problem is that instability implies turbulences and novel pilots can result injured. We use the data recorded by the GPS devices of pilots for discovering regularities and patterns to predict and select worthy zones to fly for pilots depending of their characteristics.

A *Decision Support System* (DSS) for adventure practice recommendation can be offered as a post-consumption value-added service by travel agencies to their customers [1]. Therefore, once a customer makes an on-line reservation, the travel agency can offer advice about adventure practices available in the area that customer may be

F. Sandoval et al. (Eds.): IWANN 2007, LNCS 4507, pp. 597–605, 2007.
© Springer-Verlag Berlin Heidelberg 2007

interested in. Due to the high risk factor accompanying most adventure sports a more sophisticated system is required. In this way, the customer can be provided with true helpful assistance to be aided in the decision-making process.

This paper is organized as follows: in Section 2 we introduce the dmFSQL language and the dmFSQL server. In Section 3 we introduce a new version of DAPHNE which incorporates the dmFSQL Server to do effective clustering, characterization and classification. In Section 3 we use the proposed system to solve a particular problem of soaring place selection. Finally, we suggest some conclusions.

2 dmFSQL: A Language for Flexible Queries

The dmFSQL language [11] extends the SQL language to allow flexible queries. We show an abstract with the main extensions added to SELECT command:

- **Linguistic Labels:** They represent a concrete value of the fuzzy attribute. dmFSQL works with any kind of attributes therefore, by example, a label can have associated: a trapezoidal possibility, a scalar, a text, a XML document, etc.
- **Fuzzy Comparators:** In addition to common comparators (=, >, etc.), dmFSQL includes fuzzy comparators [11].
- **Fulfillment Thresholds γ:** For each simple condition a Fulfillment threshold may be established with the format <*condition*> *THOLD* γ, indicating that the condition must be satisfied with a minimum degree γ in [0,1].
- **CDEG(<attribute>) function:** This function shows a column with the Fulfillment degree of the condition of the query for a specific attribute, which is expressed in brackets as the argument.
- **Fuzzy Constants:** We can use all of the fuzzy constants in dmFSQL [11].

At present, we have a dmFSQL Server available for Oracle© Databases, mainly, programmed in PL/SQL [10]. The architecture of the Fuzzy Relational Database with the dmFSQL Server is made up by: data and dmFSQL Server. The data can be classified in two categories:

- **Traditional Database**: They are data from our relations with a special format to store the fuzzy attribute values. They are classified by the system in 4 types:
  - **Fuzzy Attributes Type 1:** These attributes are totally crisp, but they have some linguistic trapezoidal labels defined on them, which allow us to make the query conditions for these attributes more flexible.
  - **Fuzzy Attributes Type 2:** These attributes admit crisp data as well as possibility distributions over an ordered underlying domain.
  - **Fuzzy Attributes Type 3:** These attributes have not an ordered underlying domain. On these attributes, some labels are defined and on these labels, a similarity relation has yet to be defined. With these attributes, we can only use the fuzzy comparator FEQ, as they have no relation of order.
  - **Attributes Type 4:** There are different kinds of data in a database used in diverse applications (text, XML, etc.) therefore, it would be desirable that a DM system would carry out its work in an effective way. It is a generic type (fuzzy or crisp), which admits some fuzzy treatment.

- **Fuzzy Meta-knowledge Base** (FMB)**:** It stores information for the fuzzy treatment of the fuzzy attributes in order to define the:
- **Representation Functions:** These functions are used to show the fuzzy attributes in a comprehensible way for the user and not in the internally used format.
- **Fuzzy Comparison Functions:** They are utilized to compare the fuzzy values and to calculate the compatibility degrees (CDEG function).

The dmFSQL Server has been programmed mainly in PL/SQL. It carries out a lexical, syntactic and semantic analysis of the dmFSQL query. If there are no errors, the dmFSQL query is translated into a standard SQL sentence. The resulting SQL sentence includes reference to the Representation and Fuzzy Comparison Functions.

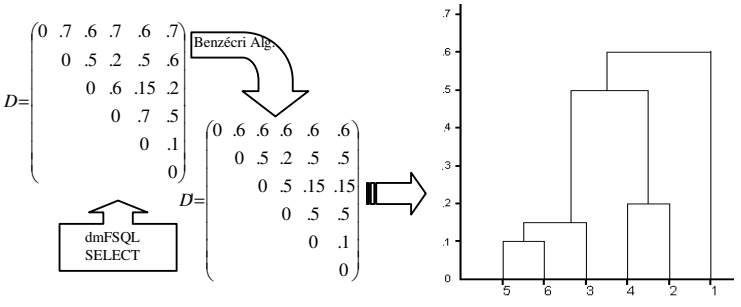

Fig. 1. Computing an ultrametric distance matrix (dendograme) for six elements

3 dmFSQL for Clustering, Characterization and Classification

In this section, we show a new version of a prototype called DAPHNE [2] which incorporate the dmFSQL Server to do effective clustering, characterization and classification of customer database. Following we explain the full process.

Operation of DAPHNE: In the first step, the relevant features of the customers for the classification are chosen using the user's knowledge. Therefore, the user inserts a new project for clustering in the meta-database of the prototype specifying the table or view with the source data *id_tabla_orig_cla* and the attributes, which DAPHNE will use for classification (col_clu_1, col_clu_2,…, col_clu_m). Normally this is a table included in the Data Warehouse (DW) with several million of records. Besides the user specify the weight of each attributes in the process (w_clu_1, w_clu_2,…, w_clu_m such that $w\_clu_r \in [0,1]$ with r=1..m and verifying $\Sigma\ w\_clu_r$ =1). Subsequently the main processes of DAPHNE are explained.

1. Computing Ultrametric Distance Matrix (see Figure 1)**:** This process attempts to obtain the population's ultrametric distance matrix. Since the results by Dunn, Zadeh y Bezdek [7] it has been well known that there is equivalence between hierarchical

clustering, max-min transitive fuzzy relation, and ultrametric distances. This process contains the following treatments:

- Computing population's normalized (in [0,1]) distance matrix (by example, the matrix D in Figure 1). For each pair of the population's individuals (i, j) the distance that separates both (dij) is obtained using dmFSQL as following:

```
SELECT A1.ROW_ID AS i, A2.ROW_ID AS j,
       1-(CDEG(A1.col_clu1)* wclu1 +…+   CDEG(A1. col_clum)* w_clum) AS dij,
FROM id_table_clustering A1,  id_table_clustering A2
WHERE A1.ROW_ID < A2.ROW_ID AND (A1.col_clu1 fuzzy_ecomp1 A2.col_clu1 THOLD 0
   AND … AND
       (A1.col_clum fuzzy_ecompm A2.col_clum THOLD 0
```

  where fuzzy_ecomp$_r$ is the fuzzy equal comparator (FEQ or NFEQ) chosen for the user for the fuzzy attribute *col_clu$_r$*.
- Computing population's ultrametric distance matrix (matrix *D'* in Figure 1). In the distance matrix, each of the three elements verifies the triangle inequality. The matrix is transformed so that each of the three elements of the ultra metric inequality is also verified. An algorithm based on the method of Benzécri [9] is used.

2. Computing possible α-cuts: Since the ultra metric matrix is finite, it contains only a finite set of different values. Thus, for the hierarchical clustering or ultra metric matrix we can always determine unequivocally the set of all possible different α-cuts, that is, the set of all different equivalence relations associated with the matrix. By example, in the Figure 1 the possible α-cuts are 0.1, 0.15, 0.2, 0.5 and 0.6.

3. Clustering: This process assigns each individual in the population to a certain cluster. The problem consists of choosing an α-cut among the possible α-cuts already obtained. The partition can be obtained in different ways according to the user's choice:

- Absolute good partition. Partition determined by the α-cut 0.5 [8]. By example, in the Figure 1 the α-cut 0.5 determines the classes {5, 6, 3} and {4, 2, 1}.
- A good partition. We use an unsupervised learning procedure based on fuzzy-set tools. This procedure determines a good partition as the minimum value of a measure denned on the set of all possible α-cuts [7].
- Partition that determines a certain number of groups. By means of a binary search algorithm on all possible α-cuts, we obtain the α-cut which implies a number of groups which are closest to the user's request.

The table *id_table_result_clu* is created as the result of the clustering process. In this table is included for each rows of *id_table_orig_project* the cluster that the row has been assigned. The structure of this table is identical to *id_table_orig_project* but with two additional attributes: ROW_ID unique identifier of each row of the table and CLUSTER_ID, identification of the cluster that the row has been assigned. Both attributes are numerical, they belong to the interval [1..*num_reg_tab*] where *num_reg_tab* is the number of rows of the table *id_table_orig_project*.

4. Computing Cluster Centroids: This process carries out a characterization of each cluster by means of a tuple called centroid (the tuple that identifies it for each group). For each attribute *col_clu$_r$* we can specify following abstraction levels:

- LABEL: the linguistic label label_avg_cdegrlc (defined in the FMB for the fuzzy column) that better describe the cluster. We choose the label with greatest

fuzzy equal average to the cluster column values using the following fuzzy select:

SELECT AVG(CDEG(*)) **AS** *label_avg_cdeg*$_{rlc}$
FROM *id_tabla_result_clu*
WHERE (*col_clu*$_r$ **fuzzy_ecomp**$_r$ *label*$_{rl}$ **THOLD** 0

- VALUE: the col_clur column value of the tuple identified with ROWID v (among all the existing) that better describe the cluster. We choose the value with greatest fuzzy equal average to the cluster column values using the following fuzzy select:

SELECT AVG(CDEG(*)) **AS** *value_avg_cdeg*$_{rvc}$
FROM *id_tabla_result_clu* A1, *id_tabla_result_clu* A2
WHERE (A1.*col_clu*$_r$ **fuzzy_ecomp**$_r$ A2.*col_clu*$_r$ **THOLD** 0
AND A2.*cluster_id* = *cluster_id*$_c$ **AND** A1.*row_id* = *v*;

- AVG: the value that better describe the cluster. This is only possible for attributes with orderly referential. In this case we use averages for crisp columns or the above mentioned methods (better label and/or better value) for fuzzy attributes.

The table *id_table_result_cen* is created as the result of the characterization process. The structure of this table is identical to *id_table_resul_clu* but with an additional attribute: AVG_CDEG cluster representativeness degree.

5. Fuzzy Classification based on Centroids: Centroids describe the clusters. The problem is the assignment to a particular cluster of the rest of the database and the new inserted tuples in such database. Usually a solution is to use a classification algorithm to obtain the rules (e.g. decision trees) which describe each group. In our approach we use the centroids in a dmFSQL query. Starting from the centres stored in *id_table_result_cen* we create a dmFSQL query with the following format:

SELECT A2.*, CDEG(*) **AS** *cdeg_cluster*
FROM *id_tabla_result_cen* A1, *id_tabla_orig_cla* A2
WHERE (A1.*col_clu*$_1$ **fuzzy_ecomp**$_1$ A2.*col_clu*$_1$ **THOLD** γ
AND … **AND**
(A1.*col_clu*$_m$ **fuzzy_ecomp**$_m$ A2.*col_clu*$_m$ **THOLD** γ **AND** A1.*cluster_id* = *cluster_id*$_c$;

With this efficient query we retrieve the objects belonging to the cluster *cluster_id*$_c$ with a membership degree of *cdeg_cluster* greater that γ.

4 Experimental Results

Site selection process for soaring depends on a number of factors, resulting in a complex decision-making task. It is common for the decision makers to use their subjective judgment and previous experience when selecting the most appropriate place for soaring or gliding (soaring is the correct term to use when the craft gains altitude from air movements during the flight). To solve this problem, the integration of a DW and a DSS seems to be efficient to extract and analyze data from different databases and sources in order to provide useful and explicit information.

Regularities and patterns for pilots and places for soaring can be discovered from the database to predict and select worthy places for soaring depending on customer's characteristics. This system has been applied to obtain such regularities and patterns

thought a process of clustering, characterization and classification of places for soaring in Andalucía (Spain) in real life situations. Here we show a simplified example of this process. Firstly, relevant attributes identified by the soaring expert have been:

- Pilot's skill (skill): is a binary attribute that indicates if the client is a novel pilot (value 0) or not (value 1). We decide define this attribute as Type 4 specifying a FEQ comparator in the FMB based in the Sokal and Michener distance. Also we define a LABEL as abstraction level for this attribute.
- Height gained (Hgained): it is the total height, in meters, gained in the flight. Pilots gain height using turbulences, so it can be seen as a potential danger indicator of the day. The more meters gained, the more dangerous was the conditions. This is a crisp attribute but we decide define this as Type 1 in the FMB using the fuzzy constants value #n = 500 (approximately n). Also we define a value (AVG) as abstraction level for this attribute.
- Satisfaction (satisfaction): after the flight, the pilots express their opinion through a survey. A value of 0 indicates that the flight conditions do not fit the pilot's skill.
- Places for soaring (place): there are four areas in the study (figure 2). Sierra Nevada (SN) and Alfacar correspond to places for soaring near Granada. Pegalajar is a place for soaring near Jaén. Obviously, this is a scalar attribute (Type 3), therefore we define a similarity relationship for the FEQ comparator in the FMB (see Table 3). Also we define a LABEL as abstraction level for this attribute.

Table 3. Similarity relationship defined for *area*

Place	S.N.	Alfacar	Pegalajar	Rest of World
S.N.	1	0.9	0.6	0
Alfacar		1	0.6	0
Pegalajar			1	0
Rest of World				1

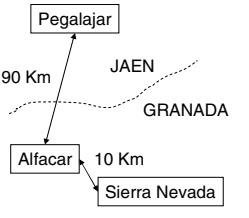

Fig. 2. Geographic distribution

Every time a flight is performed by a pilot a new row is added to a DW table. Now we must specify the weight of each attributes in the clustering process in order to better focus the customers clustering according to the user criteria. The weights chosen are 0.4 for *place* and *novel* and 0.2 for *Hgained*. By means of a sample of a few tuples includes in the table *flight_examples* the system here proposed (step 1, 2 and 3 above explained) has obtained six clusters as the optimum number in the population. This clustering results are includes in the table *flight_clu*. In the step 4 the system obtains the cluster centroids and includes it in the table *flight_cen* (table 4).

When a new customer need advise for selecting the best place for soaring we can measure how his characteristics, weather conditions and place desired for soaring fits in the corresponding cluster. It is not a good idea to recommend Sierra Nevada area to a novel pilot if we expect to have a high height gain. We can estimate the height gain knowing the weather conditions. To facilitate this estimation we can use linguistic trapezoidal labels: low, average, high.

Table 4. Results of clustering and characterization

Table *flight_clu*						Table *flight_cen*				
Id_ flight	**Place**	**no vel**	**Hgai-ned**	**Satis-fac-**	**id_ clus-**	**place**	**no vel**	**Hgai-need**	**Satis-faction**	**avg_cdeg id_cluster**
93036	Rest of World	0	20	1	1	Rest of World	0	18,67	1	.5
60932	Rest of World	0	1	1	1					
65940	Rest of World	0	35	1	1					
07788	Sierra Nevada	0	310	0	4	Sierra Ne-vada	0	275,50	0	.6
87992	Sierra Nevada	0	241	0	4					
67476	Sierra Nevada	1	1	1	2	Sierra Ne-vada	1	351,17	1	.6
44596	Sierra Nevada	1	16	1	2					
14160	Alfacar	1	141	1	2					
11281	Alfacar	1	353	1	2					
65532	Sierra Nevada	1	631	1	2					
74188	Sierra Nevada	1	965	1	2					
18096	Pegalajar	0	6	1	5	Pega-lajar	0	5	1	.8
45700	Pegalajar	0	0	1	5					
21184	Pegalajar	0	5	1	5					
10427	Pegalajar	0	9	1	5					
49867	Pegalajar	1	0	0	6	Pega-lajar	1	3,50	0	.9
01384	Pegalajar	1	7	0	6					
50392	Pegalajar	1	1580	1	3	Pega-lajar	1	1852,75	1	.8
55689	Pegalajar	1	1831	1	3					
87752	Pegalajar	1	1989	1	3					
23952	Pegalajar	1	2011	1	3					

Moreover, in order to solve a focused marketing campaign, the adventure tourism enterprise may wants to retrieve, for instance, the customers (in the entire DW database, i.e, table *flights*) who belong to cluster 2 with a minimum degree of 0.7. Therefore the dmFSQL sentence will be:

SELECT A2.*, CDEG(*) **AS** *cdeg_cluster*
FROM *flight_cen* A1, *flight* A2
WHERE A1.*place* **FEQ** A2.place **THOLD** .7
AND A1.*novel* **FEQ** A2. *novel* **THOLD** .7
AND A1.*Hgained* **FEQ** A2. *Hgained* **THOLD** .7
AND A1.*cluster_id* = 2 ;

5 Conclusions

dmFSQL Server has been defined to handling of different types of data and used as a useful tool for certain Data Mining process [3][5]. We have applied it for Tourism

Management. Besides the specific requirements of the area, the prototype has been designed considering the above-mentioned desirable functionalities of DM systems:

- Handling of Different Types of Data: The possibility of combination any type of data for the clustering process is considered novel in this area.
- Efficiency and Interactive Mining Knowledge: The prototype has been designed to be interactive with the user and to give the answer in real time in order to obtain the wanted population's classification of the entire very large database.
- Accuracy: The use of the classic method of Benzécri to obtain the hierarchy of parts has guaranteed the goodness of such a partition. In addition, the procedure to obtain a good partition based on fuzzy sets and the fuzzy classification has given excellent results during the tests.
- Friendly Interface: The interface of DAPHNE is graphic and completely user guided. Like-wise, the prototype includes a meta-database, in such a way that the management of a clustering project can become quick and easy for the user.

This work has been supported by the Andalucía Research Program under project GR050/06 and the Spanish Research Program under project TIN2005-09098-C05-03.

References

1. Araque, F., Salguero, A., Abad-Grau, M.M.: Application of data warehouse and decision support system in Soaring site recommendation. In: Hitz, M., Sigala, M., Murphy, J. (eds.) Proceedings of the the Thirteenth Conference on Information and Communication Technologies in Tourism, ENTER 2006, pp. 308–319. Springer-Verlag Computer Science, Wien-NewYork (2006)
2. Carrasco, R.A., Galindo, J., Vila, M.A., Medina, J.M.: Clustering and Fuzzy Classification in a Financial Data Mining Environment. In: 3rd International ICSC Symposium on Soft Computing, SOCO'99, pp. 713–720, Genova, Italy (June 1999)
3. Carrasco, R.A., Vila, M.A., Galindo, J.: Using dmFSQL for Financial Clustering. In: 7th International Conference on Enterprise Information Systems, ICEIS'2005, Miami (USA), Vol. II, pp. 135–141 (2005)
4. Carrasco, R.A., Vila, M.A., Galindo, J.: FSQL: a Flexible Query Language for Data Mining. In: Piattini, M., Filipe, J., Braz, J. (eds.) Enterprise Information Systems IV., pp. 68–74. Kluwer Academic Publishers, Boston (2002) ISBN: 1-4020-1086-9
5. Carrasco, R.A., Vila, M.A., Galindo, J., Cubero, J.C.: FSQL: a Tool for Obtaining Fuzzy Dependencies. In: 8th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'2000, Madrid, Spain, pp. 1916–1919 (July 2000)
6. Frawley, W.J., Piatetsky-Shapiro, G., Matheus, C.J.: Knowledge Discovery in Databases: An Overview. In: Piatetsky-Shapiro, G., Frawley, W.J. (eds.) Knowledge Discovery in Databases, pp. 1–31. The AAAI Press, Stanford.
7. Delgado, M., Gómez-Skarmeta, A.F., Vila, A.: On the Use of Hierarchical Clustering, In Fuzzy Modelling. International Journal of Approximate Reasoning 14, 237–257 (1996)
8. Vila, M.A.: Nota sobre el cálculo de particiones óptimas obtenidas a partir de una clasificación con jerárquica. Actas de la XI Reunión Nacional de I.O., Sevilla, España (1979)
9. Benzécri, P, et al.: L'analyse des données; Tomo I: La Taxinomie; Tomo II: L'analyse des correspondences, Paris, Dunod (1976)

10. Carrasco, R.A., Vila, M.A., Araque, F., Salguero, A., Aguilar, M. Á.: dmFSQL: a Server for Data Mining. Workshop on Data Mining and Business Intelligence (DMBI2007) in conjunction with the 23rd International Conference on Data Engineering, ICDE'07. April 15, 2007, IEEE. Istanbul, Turkey (To appear 2007)
11. Carrasco, R.A., Vila, M.A., Araque, F.: dmFSQL: a Language for Data Mining. XVI Database and Experts Systems Aplications, DEXA, IEEE, pp. 440–444, (2006) ISBN-13: 978-0-7695-2641-6 ISBN-10:0-7695-2641-1 ISSN:1529-4188