RESUMEN DE TESIS DOCTORAL

Restricted Conceptual Clustering Algorithms based on Seeds

Algoritmos Conceptuales Restringidos basados en Semillas

Graduated: Irene Olaya Ayaquica Martínez
National Institute of Astrophysics, Optics and Electronics
Luis Enrique Erro # 1, Santa María Tonantzintla, C.P. 72840, Puebla, México.
Graduated in July 19, 2007

Advisor: José Francisco Martínez Trinidad
National Institute of Astrophysics, Optics and Electronics
Luis Enrique Erro # 1, Santa María Tonantzintla, C.P. 72840, Puebla, México.
fmartine@inaoep.mx

Advisor: Jesús Ariel Carrasco Ochoa
National Institute of Astrophysics, Optics and Electronics
Luis Enrique Erro # 1, Santa María Tonantzintla, C.P. 72840, Puebla, México.
ariel@inaoep.mx

Abstract

The non-supervised classification algorithms determine clusters such that objects in the same cluster are similar among them, while objects in different clusters are less similar. However, there are some practical problems where, besides determining the clusters, the properties that characterize them are required. This problem is known as conceptual clustering.

There are different methods that allow to solve the conceptual clustering problem, one of them is the conceptual k-means algorithm, which is a conceptual version of the k-means algorithm; one of the most studied and used algorithms for solving the restricted non-supervised classification problem (when the number of clusters is specified *a priori*). The main characteristic of the conceptual k-means algorithm is that it requires generalization lattices for the construction of the concepts. In this thesis, an improvement of the conceptual k-means algorithm and a new conceptual k-means algorithm that does not depend on generalization lattices for building the concepts are proposed.

Finally, in this thesis, two fuzzy conceptual clustering algorithms, which are fuzzy versions of the proposed hard conceptual clustering algorithms, are introduced.

Keywords: Conceptual Clustering, Fuzzy Conceptual Clustering, Similarity Functions, Mixed Data.

Resumen

El estudio de la clasificación no supervisada ha sido enfocado principalmente a desarrollar métodos que determinen agrupamientos tales que objetos en el mismo agrupamiento sean similares entre ellos, mientras que objetos de diferentes agrupamientos sean poco similares. Sin embargo, para algunos problemas prácticos se requiere, además de determinar los agrupamientos, conocer las propiedades que describan cómo son dichos agrupamientos. A este problema se le conoce como agrupamiento conceptual.

Existen diversos algoritmos que permiten resolver el problema de agrupamiento conceptual, entre los que se encuentra el algoritmo k-means conceptual, el cual es una versión conceptual del algoritmo k-means; uno de los algoritmos más estudiados y utilizados para resolver el problema de clasificación no supervisada restringida (cuando se especifica *a priori* el número de agrupamientos). La principal característica del algoritmo k-means conceptual es que requiere retículos de generalización para la construcción de los conceptos. En esta tesis se proponen dos algoritmos k-means conceptuales, el primero de ellos es una mejora del algoritmo k-means conceptual y el segundo es un algoritmo k-means conceptual que no requiere retículos de generalización para la construcción de los conceptos.

Finalmente, en esta tesis se proponen dos algoritmos conceptuales difusos, los cuales son versiones difusas de los algoritmos conceptuales duros propuestos.

Palabras Clave: Agrupamiento Conceptual, Agrupamiento Conceptual Difuso, Funciones de Similaridad, Datos Mezclados.

1 Introduction

The clustering algorithms determine clusters such that objects in the same cluster are similar among them, while objects in different clusters are less similar. However, there are some situations where, besides determining the clusters, the properties that characterize them are required. This problem is known as conceptual clustering.

The first conceptual clustering algorithms were proposed by Michalski (Michalski and Diday, 1981; Michalski and Stepp, 1983; Stepp and Michalski, 1986) and starting from these, other conceptual clustering algorithms have been developed (Lebowitz, 1986; Hanson, 1990; Fisher, 1990; Gennari et al., 1990; McKusick and Thompson, 1990; Béjar and Cortés, 1992; Ralambondrainy, 1995; Martínez-Trinidad and Sánchez-Díaz, 2001; Pons-Porrata et al., 2002; Seeman and Michalski, 2006). These conceptual clustering algorithms can be divided in two types: restricted and non restricted. The restricted conceptual algorithms are those where the number of clusters, and concepts, to build is specified *a priori*, and usually they require seeds for working; while the non restricted conceptual algorithms are those where the number of clusters is not specified *a priori*. In this thesis, the restricted conceptual clustering problem based on seeds was addressed.

In this thesis, two extensions of the conceptual k-means algorithm (Ralambondrainy, 1995), which allow working with mixed and incomplete data using any object comparison function were proposed. Also, two fuzzy conceptual clustering algorithms, which are fuzzy versions of the proposed conceptual clustering algorithms, were proposed.

2 Restricted Conceptual Algorithms

The conceptual k-means algorithm (Ralambondrainy, 1995) is a conceptual version of the k-means algorithm, one of the most studied and used algorithms for solving the clustering problem when the number of clusters is specified *a priori*. The conceptual k-means algorithm consists of two phases: an aggregation phase, where the clusters are built and a characterization phase, where the properties or concepts are generated. This algorithm allows working with objects described by mixed (quantitative and qualitative) features; and it does not allow missing data.

In the aggregation phase, a distance for measuring similarities among objects is defined. The distance function is defined as the sum of the Euclidean distance, for quantitative features; and the Chi-square distance, for the qualitative features. In order to apply the Chi-square distance, a transformation of each qualitative feature into a set of Boolean features, must be done. This codification does not transform the representation space in $\Re^n$, where means can be computed, because the 1's y 0's associated to the new features are codes not numbers; therefore, the obtained centroids (means) do not have an interpretation in $\Re^n$.

In the characterization phase, a generalization lattice for each feature is needed. For qualitative features, the generalization lattice must be given *a priori*. For quantitative features, a codification into qualitative features is done. The codification function, for each quantitative feature is the following:

$$c(r) = \begin{cases} inf & if \quad r < \mu_x - \sigma_x \\ typical & if \quad \mu_x - \sigma_x \leq r \leq \mu_x + \sigma_x \\ sup & if \quad \mu_x + \sigma_x < r \end{cases} \tag{1}$$

where r is a value of the feature x; μ_x is the *mean* of the feature x in the cluster A_i and σ_x is the *standard deviation* of x in the cluster A_i. Using this codification, the following generalization lattice was proposed by Ralambondrainy:

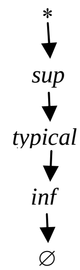

Fig. 1. Generalization lattice for the quantitative features proposed by Ralambondrainy

Before presenting the conceptual clustering algorithms proposed in this thesis, a formal outline of the conceptual clustering problems is given.

2.1 Restricted Conceptual Clustering

Let $X = \{O_1,\ldots,O_n\}$ be a set of objects. Each object O_j is described by a set of features $R = \{x_1,\ldots,x_m\}$. Each feature x_s takes values in a set of admissible values D_s, $x_s(O_j) \in D_s$, $s = 1,\ldots,m$, $j = 1,\ldots,n$; with $x_s(O_j)$ the value of the feature x_s in the object O_j. The features can be of any nature (qualitative: Boolean, multi-valued, etc. or quantitative: integer, real). Also, it is assumed that D_s contains the symbol "?" which denotes missing data; thus, incomplete object descriptions can be considered.

For each feature x_s, a comparison function $FC_s : D_s \times D_s \rightarrow L_s$ $s=1,2,...,m$, is defined, where L_s is a completely ordered set. The FC_s function gives an evaluation of the similarity degree between two values of the feature x_s. In addition, let $\Gamma : (D_1 \times \ldots \times D_m)^2 \rightarrow [0,1]$ be a similarity function, which allows evaluating the similarity degree between two object descriptions.

The restricted conceptual clustering problem consists in structuring the objects in k clusters $\{A_1,\ldots,A_k\}$, $k>1$, and generating properties or concepts, C_i, for characterizing the clusters A_i, $i = 1,\ldots,k$.

A concept C_i will be represented as a disjunction of predicates $P = (x_1,a_1) \wedge \ldots \wedge (x_m,a_m)$, with $x_s \in R$ and $a_s \in D_s$. The predicate P covers an object O_j if $x_s(O_j) = a_s$ or a_s is more general than $x_s(O_j)$ according to the generalization lattice of x_s, $s = 1,\ldots,m$. Also, a concept C_i covers an object O_j if at least one predicate P in the concept C_i covers the object O_j.

A concept C_i for the cluster A_i must satisfy that if the object O_j belongs to the cluster A_i then the object O_j should be covered by the concept C_i; and if the object O_j does not belong to the cluster A_i then the object O_j should not be covered by the concept C_i.

2.2 Quality Function

For comparing the algorithms that solve the conceptual clustering problem, a function for evaluating the quality of the concepts is required. In order to measure this quality, some characteristics of the concepts that can be taken into account are: the percentage of objects belonging to a cluster that are covered by the concept, the number of objects outside the cluster covered by the concept, the size of the concepts, or the simplicity of the concepts.

Ralambondrainy (1995) proposed to take the percentage of objects belonging to the cluster that are covered by the concept as quality measure. However, it is also necessary to take into account the objects outside the cluster that are covered by the concept; because this allows to evaluate not only how the concepts characterize the clusters, but also how much the concepts differentiate objects of a cluster from objects in other clusters.

The quality function that we propose in this thesis takes into account the number of objects in a cluster covered by the concept (examples) as well as the number of objects outside the cluster covered by the concept (counterexamples). A concept will have better quality if it recognizes more examples and less counterexamples. The maximum quality is reached when the concepts cover to all the objects in their clusters and any object outside of them.

The proposed quality function is the following:

$$quality(C_1,...,C_k) = \frac{1}{k}\sum_{i=1}^{k}\frac{examples(C_i)}{|A_i| + counterexamples(C_i)} \quad (2)$$

where:

k	**is the number of clusters.**
C_i	**is the concept associated to the cluster A_i, $i = 1,\ldots,k$.**
$examples(C_i)$	**is the number of objects in the cluster A_i covered by the concept C_i.**
$counterexamples(C_i)$	**is the number of objects outside the cluster A_i covered by the concept C_i.**

This function takes higher values if the number of covered examples increases and the number of covered counterexamples decreases. This function takes 1.0 when the concept C_i covers all the objects in the cluster A_i and it does not cover any object outside of A_i.

2.3 Conceptual K-means Algorithm based on Similarity Functions

The first algorithm proposed in this thesis is the conceptual k-means algorithm based on similarity functions (CKMSF), which is a modification of the conceptual k-means algorithm (CKM) (Ralambondrainy, 1995). The CKMSF algorithm consists of two phases: a clustering phase and a characterization phase.

2.3.1 Clustering Phase

In this phase, we propose to use the k-means with similarity functions algorithm (KMSF) (García-Serrano and Martínez-Trinidad, 1999) for building the clusters. This algorithm, opposite to the CKM algorithm, allows using any comparison function to compare feature values and any similarity function to compare objects. Also, in the KMSF algorithm objects of the sample are selected as centroids instead of using means.

2.3.2 Characterization Phase

In this phase, a generalization lattice for each feature is required. The generalization lattices for qualitative features must be given *a priori* by the user; while, for quantitative features, the same codification than CKM and the generalization lattice proposed by Pons-Porrata (1999) (Figure 2) were used.

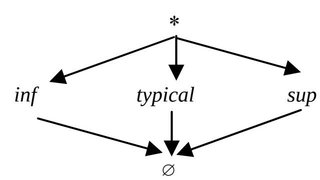

Fig. 2. Generalization lattice for quantitative features proposed by Pons-Porrata

First, an initial predicate $P = (x_1,a_1) \wedge \ldots \wedge (x_m,a_m)$ for each object $O_j \in A_i$, $j = 1,\ldots,|A_i|$ is built, where a_s is the value of the feature x_s in the object O_j.

Starting from these predicates and based on the generalization lattices generalized predicates are generated. Two predicates $P_1 = (x_1,a_1) \wedge \ldots \wedge (x_m,a_m)$ and $P_2 = (x_1,b_1) \wedge \ldots \wedge (x_m,b_m)$ will be generalized if, for each feature x_s, $s = 1,\ldots,m$, the values a_s and b_s are equal or they can be generalized in the generalization lattice defined for the feature x_s. If the values a_s and b_s can not be generalized, then P_1 and P_2 will not be generalized.

The generalization of the values a_s and b_s of each feature x_s is made as follows: if the values a_s and b_s are equal then, the generalized predicate will take this value for the feature x_s; if the values are different then the value for the feature x_s in the generalized predicate will be the generalization of a_s and b_s given by the lattice. If a value is more general than the other then the value for the feature x_s that the generalized predicate will take is the most general value of them.

A generalized predicate is stored if it is α-discriminating (the number of objects outside of A_i covered by the predicate is smaller or equal than α) and β-characterizing (the number of objects in A_i covered by the predicate is greater o equal than β), in other case it is eliminated. If a new predicate is stored then the predicates, starting from which this predicate was generated, are eliminated. This generalization process is repeated until no more generalized predicates can be generated.

The obtained set of predicates can contain predicates that recognize the same objects; therefore, this set can be reduced (eliminating those predicates recognizing the same objects than another predicate). This reduction is made using the strategy proposed by Ralambondrainy (1995) which works as follows: the predicates are descendently ordered according to the number of objects that each one covers. The first predicate is stored. For the remaining predicates, if a predicate covers any object not covered by the stored predicates then this predicate is added to the concept; otherwise, it is eliminated. Finally, the concept will be formed by the disjunction of the stored predicates.

In a generalization lattice the symbol * indicates that the feature can take any value; therefore, in order to simplify a concept we can eliminate from the predicates those features that contain *; which are not useful as descriptors for the concept.

The main problem of using generalization lattices is that they could be difficult to define; moreover, there are not automatic methods to build these lattices, therefore this task must be done by the user. For this reason, in the next section, we propose a new k-means conceptual algorithm that does not depend on generalization lattices for the construction of the concepts.

2.4 Conceptual K-means Algorithm based on Complex Features

In this section, a conceptual k-means algorithm based on complex features (CKMCF) which, as CKM and CKMSF algorithms, has a clustering phase and a characterization phase is proposed.

2.4.1 Clustering Phase

In this phase, as in the CKMSF, we use the k-means with similarity functions algorithm for building the clusters.

2.4.2 Characterization Phase

The complex features (De-la-Vega-Doria, 1994) are combinations of values for a subset of features such that these values appear in objects of the cluster A_i and, they do not appear in objects of other clusters. These values characterize objects in the cluster A_i and they do not characterize objects outside of A_i. Therefore, the complex features can be used for generating concepts. A complex feature is defined as follows:

Definition 2.1*:** Let $\Omega = \{x_{s_1},...,x_{s_p}\}$ be a set of features and let $(a_1,...,a_p)$ be values associated to the features $x_{s_1},...,x_{s_p}$ taken from an object of the cluster A_i, then $\{x_{s_1},...,x_{s_p}\}$-$(a_1,...,a_p)$ is a ***complex feature (De-la-Vega-Doria, 1994) of the cluster A_i, if and only if:

1) $\sum_{O_j \in A_i} \Gamma(\Omega O_j, (a_1,...,a_p)) \geq \beta_i$

2) $\sum_{O_j \notin A_i} \Gamma(\Omega O_j, (a_1,...,a_p)) < \lambda_i$

where ΩO_j is the subdescription of the object O_j taking into account only the features of Ω; β_i is the minimum similarity that the objects of the cluster A_i should have with the subdescription $(a_1,...,a_p)$ and λ_i is the maximum similarity that the objects outside the cluster should have with $(a_1,...,a_p)$.

In order to obtain the complex features, subsets of features Ω that indicate the subdescriptions of the objects where the complex features will be searched; these subsets are called support sets. The following support sets are used for the CKMCF algorithm:

1. Γ-discriminating support sets (Alba-Cabrera, 1997); which are subsets of features such that the difference among objects from different clusters is greater than the difference considering all the features.
2. Γ-characterizing support sets (Alba-Cabrera, 1997); which are subsets of features such that the similarity among objects in the same cluster is greater than the similarity considering all the features.
3. Γ-testors support sets (Alba-Cabrera, 1997); which are subsets of features satisfying the properties Γ-discriminating and Γ-characterizing at the same time.

The support sets are obtained through a genetic algorithm, which is described in (Guevara-Cruz, 2004) and the complex features are computed using these support sets and verifying the definition 2.1.

In order to generate the concepts, a predicate P is associated to each complex feature. The predicate P is a conjunction of feature values, where the features that appear in the complex feature take the value a_s and the features that do not appear in the complex feature take the value *. The symbol * means "any value is possible".

The set of predicates obtained from the complex features can contain predicates that recognize the same objects; therefore, this set of predicates can be reduced (eliminating predicates that recognize the same objects than another predicate). This reduction is made using the same strategy used for reducing the predicates in the CKMSF algorithm.

2.5 Experimental Results

In order to illustrate the performance of the proposed algorithms (Sections 2.3 and 2.4), the results obtained by applying the CKMSF and CKMCF algorithms on different databases are presented in this section. The databases used for the experiments were taken from the UCI databases repository (Blake et al., 1998). For these experiments, we ignored the labels of the classes. In addition, a comparison among our algorithms and the conceptual k-means (CKM) algorithm is shown.

In the experiments, the following similarity function was used:

$$\Gamma(O_p, O_j) = \frac{\sum_{x_s \in R} FC_s(x_s(O_p), x_s(O_j))}{m}$$

where $FC_s(x_s(O_p),x_s(O_j))$ is the comparison function used for comparing values of the feature x_s.

The comparison functions used for the experiments were the following:

1. For quantitative features:

$$FC_s(x_s(O_i), x_s(O_j)) = \begin{cases} 0 & \text{if } \; x_s(O_i) = ? \vee x_s(O_j) = ? \vee \left| x_s(O_i) - x_s(O_j) \right| \geq \sigma \\ 1 & \text{in other case} \end{cases}$$

where σ is the standard deviation of the feature x_s in the sample.

2. For qualitative features:

$$FC_s(x_s(O_i), x_s(O_j)) = \begin{cases} 0 & \text{if } \; x_s(O_i) = ? \vee x_s(O_j) = ? \vee x_s(O_i) \neq x_s(O_j) \\ 1 & \text{in other case} \end{cases}$$

The treatment given, in this thesis, to missing data is the following: when a value of the feature x_s is missing ("?") then it is considered as different from any other value, even from another missing value.

In Table 1, the quality and the number of predicates of the concepts obtained by each algorithm (CKM, CKMSF and CKMCF) are shown.

In Figure 3 the results of Table 1 are shown in a graph. For the CKMCF algorithm, only the results obtained with the Γ-discriminating support sets, which were the support sets that obtained the best results, are depicted.

In Table 1 and Figure 3, we can observe that, in average, the best quality is obtained with the CKMSF and CKMCF algorithms. In average, the CKMCF algorithm using Γ-discriminating support sets obtained concepts with a slightly lower quality than the quality of concepts obtained by the CKMSF algorithm; however, the CKMCF does not require generalization lattices and it obtains concepts with less predicates.

Table 1. Quality and number of predicates of the concepts obtained by the CKM, CKMSF and CKMCF algorithms

Databases	**CKM Algorithm**		**CKMSF Algorithm**		**CKMCF Algorithm**					
					Γd		**Γc**		**Γt**	
	Quality	**# Pred.**	**Quality**	**# Pred.**	**Quality**	**# Pred.**	**Quality**	**# Pred.**	**Quality**	**# Pred.**
Diabetes	0.53	261	0.83	218	0.89	44	0.89	44	0.89	44
Glass	0.54	67	0.89	83	0.66	21	0.66	20	0.66	21
Iris	0.85	52	0.92	10	0.88	3	0.88	3	0.88	3
Wine	0.29	47	1.00	40	1.00	30	1.00	27	1.00	29
Hayes	1.00	18	0.99	17	1.00	13	1.00	13	1.00	13
Lenses	1.00	5	0.95	8	1.00	8	1.00	8	1.00	8
Zoo	1.00	9	1.00	14	1.00	21	1.00	17	1.00	19
Auto-mpg	0.75	164	0.61	136	0.62	43	0.62	43	0.62	43
Echocardiogram	0.40	43	0.88	89	0.94	48	0.94	40	0.95	53
Hepatitis	0.53	50	0.99	46	0.98	68	0.92	39	0.99	97
Import85	0.47	57	0.98	63	0.98	113	0.86	46	0.98	114
Tae	0.89	30	0.97	71	0.95	40	0.95	40	0.95	40
Average	**0.69**	**67**	**0.92**	**66**	**0.91**	**38**	**0.89**	**28**	**0.91**	**40**

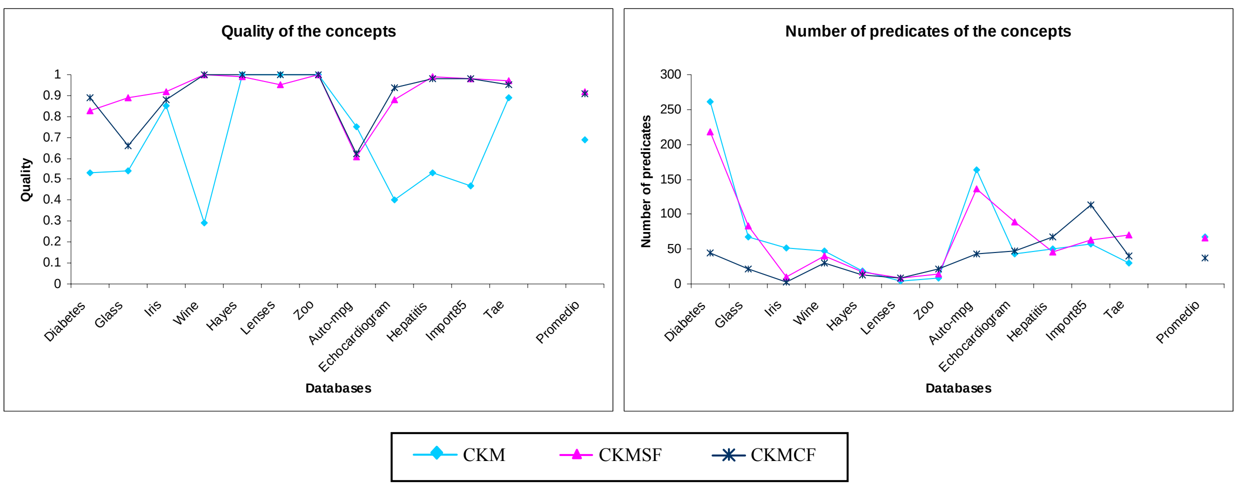

Figure 3. Quality and number of predicates of the concepts obtained by the CKM, CKMSF and CKMCF (using Γ-discriminating support sets) algorithms

3 Fuzzy Conceptual Algorithms

In some practical problems, to determine the membership degree of an object to a cluster instead of determining only if the object belongs or not to the cluster is required. In addition, to obtain fuzzy concepts that provide us a description of the objects belonging with a certain degree to the clusters is very important. This problem is called Fuzzy Conceptual Clustering.

There are some works where the fuzzy conceptual clustering problem has been faced (Martínez-Trinidad and Ruiz-Shulcloper, 1998; Martínez-Trinidad, 2000; Quan et al., 2004a; Quan et al., 2004b). These works address the problem when the number of clusters is not specified *a priori*. However, there are not algorithms for solving the restricted fuzzy conceptual clustering problem; i.e., when the number of clusters is specified *a priori*. Therefore, in this thesis two restricted fuzzy conceptual clustering algorithms, which are fuzzy versions of the conceptual algorithms proposed in Section 2 are introduced.

3.1 Fuzzy Restricted Conceptual Clustering

The fuzzy restricted conceptual clustering problem is similar to the restricted conceptual clustering problem, but the clusters and the concepts are fuzzy.

The fuzzy restricted conceptual clustering problem consists in structuring the objects in k fuzzy clusters $\{A_1,\ldots,A_k\}$, $k>1$, and generating fuzzy properties or concepts, C_i, $i = 1,\ldots,k$.

In this thesis, a fuzzy predicate will be a pair (P,μ_P), where P is a hard predicate that describe some objects and μ_P is a value that will be asociated to the objects covered by P, i.e., P describes the objects that belong with degree μ_P to the fuzzy cluster A_i.

The degree in that each object is covered by the concept is determined as follows: If an object O_j is covered by only one fuzzy predicate (P,μ_P) of the concept C_i, the degree in that the object O_j is covered by the fuzzy concept C_i will be the value μ_P. If the object O_j is covered by more than one fuzzy predicate of C_i, the degree in that the object O_j is covered by the fuzzy concept C_i will be the maximum of the μ_P associated to the predicates that cover O_j; on the other hand, if any predicate covers the object O_j, then the fuzzy concept C_i covers the object O_j with degree 0.

3.2 Quality Funtion

A fuzzy concept C_i for a fuzzy cluster A_i must satisfy: if the object O_j belongs with high degree to the fuzzy cluster A_i then it should be covered with high degree by the concept C_i and if the object O_j belongs with low degree to the fuzzy cluster A_i then it should be covered with low degree by the concept C_i. Therefore, for evaluating the quality of the concepts obtained by the proposed fuzzy conceptual clustering algorithms, a generalization of the quality function for the hard algorithms (2.2) is defined as follows:

$$quality(C_1,\ldots,C_k) = \frac{1}{k}\sum_{i=1}^{k}\frac{\sum_{O\in A_{C_i}}\left(1-\left|\mu_{A_i}(O_j)-\mu_{C_i}(O_j)\right|\right)}{\left|A_{C_i}\right| + \sum_{O\notin A_{C_i}}\left|\mu_{A_i}(O_j)-\mu_{C_i}(O_j)\right|} \tag{3}$$

where:

$A_{C_i} = \left\{O_j \,\middle|\, \max_{p=1,\ldots,k}\{\mu_{A_p}(O_j)\} = \mu_{A_i}(O_j)\right\}$ **is the set of objects that belong more to the cluster A_i than to other clusters.**

k **is the number of clusters.**
C_i **is the concept associated to the cluster A_i.**
$\mu_{A_i}(O_j)$ **is the membership degree of the object O_j to the cluster A_i.**
$\mu_{C_i}(O_j)$ **is the degree in which the object O_j is covered by the concept C_i.**

The function (3.1) takes high values if the difference between the membership degree of the objects to the clusters and the degree in which they are covered by the concepts is small. The function takes 1.0 when the concepts cover all the objects in the same degree that they belong to the cluster.

If the membership degrees are hard (0's y 1's) then this function is the same quality function defined for the hard conceptual clustering problem.

3.3 Fuzzy Conceptual K-means Algorithm based on Similarity Functions

The fuzzy conceptual k-means algorithm based on similarity functions (FCKMSF) is an extension of the conceptual k-means algorithm based on similarity functions (CKMSF). The FCKMSF algorithm consists of two phases: a clustering phase and a characterization phase.

3.3.1 Clustering Phase

In this phase, we propose to use the fuzzy k-means with similarity functions algorithm (FKMSF) (Ayaquica-Martínez and Martínez-Trinidad, 2001) to build the fuzzy clusters. The FKMSF algorithm is a fuzzy version of the KMSF algorithm (García-Serrano and Martínez-Trinidad, 1999), which was used in the clustering phase of the proposed hard algorithms.

3.3.2 Characterization Phase

In this phase, a generalization of the CKMSF characterization phase is proposed. This generalization allows generate fuzzy concepts starting from fuzzy clusters.

It is important to remind that, due to the FCKMSF is based on the KMSF algorithm; a generalization lattice for each feature is required. For qualitative features, the generalization lattice must be given *a priori*, while for quantitative features, the codification function (2.1) and the generalization lattice showed in Figure 2 are used.

In order to build the initial predicates, for each fuzzy cluster A_i, only the objects that belong more to the cluster A_i than to other clusters are taken into account. A fuzzy predicate (P,μ_P) is associated to each object O_j. The predicate P is built as in the hard case and μ_P takes as value the membership degree of the object O_j to the cluster A_i.

Starting from these predicates and based on the generalization lattices generalized fuzzy predicates are generated. Two fuzzy predicates $\left(P_1,\mu_{P_1}\right)$ and $\left(P_2,\mu_{P_2}\right)$ will be generalized if $\left|\mu_{P_1}-\mu_{P_2}\right|<\varepsilon$, with ε ∈ [0,1] and P_1 and P_2 can be generalized. Thus, only predicates with similar value of μ_P are generalized. The predicate P for the generalized fuzzy predicate, will be the generalization of P_1 and P_2, and the value of μ_P for the generalized fuzzy predicate will be the average of μ_{P_1} and μ_{P_2}.

It is important to note that if the value of ε is close to 1, then we will obtain fuzzy concepts with low quality, because we allow to generalize fuzzy predicates which represent objects with no similar membership degrees; while if the value of ε is close to 0, the fuzzy concepts obtained will have high quality, because we allow to generalize only fuzzy predicates with very similar membership degrees; if $\varepsilon = 0$ the final fuzzy predicates will be all the initial fuzzy predicates and each predicate will cover only one object.

A generalized fuzzy predicate is stored if it is α-discriminating (the number of objects outside of A_i covered by the predicate is smaller or equal than α) and β-characterizing (the number of objects in A_i covered by the predicate is greater o equal than β), in other case it is eliminated. If a generalized fuzzy predicate is stored then the fuzzy predicates, starting from which this predicate was generated, are eliminated. This generalization process is repeated until no more generalized fuzzy predicates can be generated.

The α-discriminating and β-characterizing properties are verified as in the hard case; therefore, it is necesary hardening the clusters. In order to harden a cluster A_i, only the objects that belong more to the cluster A_i than to other clusters are taken into account.

The obtained set of fuzzy predicates can contain predicates that do not contribute to improve the quality of the concepts; therefore, this set can be reduced. This reduction is made using a generalization of the strategy proposed by Ralambondrainy (1995), which works as follows: the fuzzy predicates are descendently ordered according to μ_P. The first predicate is stored. For the remaining predicates, if a predicate improves the quality of the concept (measured with the expression (3.1)), then the predicate is added to the concept; otherwise, it is eliminated. Finally, the concept will be formed by the disjunction of the stored fuzzy predicates.

3.4 Fuzzy Conceptual K-means Algorithm based on Complex Features

In this section, a fuzzy version (FCKMCF) of the conceptual k-means algorithm based on complex features (CKMCF) is proposed. The FCKMCF algorithm consists of a clustering phase and a characterization phase.

3.4.1 Clustering Phase

In this phase, as in the FCKMSF algorithm, the fuzzy k-means with similarity functions algorithm is used for building the fuzzy clusters.

3.4.2 Characterization Phase

In this phase, for generating the concepts, the fuzzy complex features are used. A fuzzy complex feature is defined as follows:

Definition 3.1*:** Let $\Omega = \{x_{s_1},...,x_{s_p}\}$ be a set of features and let $(a_1,...,a_p)$ be values associated to the features $x_{s_1},...,x_{s_p}$ taken from an object of the cluster A_i, then $\{x_{s_1},...,x_{s_p}\}$-$(a_1,...,a_p)$ is a ***fuzzy complex feature (De-la-Vega-Doria, 1994) of the cluster A_i, if and only if:

1) $\sum_{O_j \in X} \left[\Gamma\left(\Omega O_j, (a_1,...,a_p)\right) \mu_{A_i}(O_j)\right] \geq \beta_i$

2) $\sum_{O_j \in X} \left[\Gamma\left(\Omega O_j, (a_1,...,a_p)\right) \left(1 - \mu_{A_i}(O_j)\right)\right] < \lambda_i$

where $\mu_{A_i}(O_j)$ is the membership degree of O_j to the cluster A_i ; ΩO_j, β_i y λ_i are defined in the same way as in the hard case.

In order to obtain the fuzzy complex features, support sets are needed. In this thesis, besides the Γ-discriminating, Γ-characterizing and Γ-testors, the fuzzy Φ-testors are used. The Γ-discriminating, Γ-characterizing and Γ-testors support sets can be obtained only for hard clusters; in a similar way for evaluating the α-discriminating and β-characterizing properties the clusters were hardening. In order to calculate the Γ-discriminating, Γ-characterizing and Γ-testors support sets a genetic algorithm, which is described in (Guevara-Cruz, 2004) was used; and for calculating the fuzzy Φ-testors the genetic algorithm proposed by Santos-Gordillo et al. (2003) was used.

In order to generate the concepts, a fuzzy predicate (P,μ_P) is associated to each complex feature. The predicate P is built as in the hard case and the value of μ_P is the average of the membership degrees of the objects covered by the predicate P.

The set of fuzzy predicates obtained from the complex features can contain predicates that do not contribute to improve the quality of the concepts. Thus, this set of predicates can be reduced using the same strategy used for reducing the predicates in the FCKMSF algorithm.

3.5 Experimental Results

In order to show the performance of the proposed algorithms (Sections 3.3 and 3.4), in this section the results obtained by applying the FCKMSF and FCKMCF algorithms on different databases are presented. These databases are the same that in the hard case. Also, the similarity function and the comparison functions are the same that in the hard case.

For the fuzzy case, the FCKMSF and FCKMCF algorithms were compared only between them because there are not restricted fuzzy conceptual algorithms for comparing with.

In Table 2 the quality and the number of predicates of the concepts obtained by the FCKMSF and FCKMCF algorithms are shown.

Table 2. Quality and number of fuzzy predicates of the concepts obtained by the FCKMSF and FCKMCF algorithms

Databases	**FCKMSF Algorithm**		**FCKMCF Algorithm**							
			Γd		**Γc**		**Γt**		**Φt**	
	Quality	**# Pred.**	**Quality**	**# Pred.**	**Quality**	**# Pred.**	**Quality**	**# Pred.**	**Quality**	**# Pred.**
Diabetes	0.66	230	0.68	72	0.68	72	0.68	72	0.70	105
Glass	0.69	69	0.65	21	0.65	21	0.65	21	0.68	40
Iris	0.78	37	0.75	3	0.75	3	0.75	3	0.78	10
Wine	0.62	99	0.57	99	0.57	94	0.57	112	0.60	99
Hayes	0.71	66	0.63	23	0.63	23	0.63	23	0.65	25
Lenses	0.77	11	0.66	14	0.66	14	0.66	14	0.70	11
Zoo	0.64	41	0.59	17	0.50	13	0.58	15	0.60	22
Auto-mpg	0.60	96	0.72	22	0.72	22	0.72	22	0.75	60
Echocardiogram	0.64	84	0.57	58	0.57	58	0.56	70	0.63	70
Hepatitis	0.74	43	0.45	40	0.58	73	0.48	49	0.62	60
Import85	0.60	112	0.45	18	0.50	37	0.47	31	0.58	90
Tae	0.72	96	0.52	21	0.52	21	0.52	21	0.60	50
Average	**0.68**	**82**	**0.60**	**34**	**0.61**	**38**	**0.61**	**38**	**0.66**	**54**

In Figure 4, the results of Table 2 are shown in a graph. For the FCKMCF algorithm only the results obtained with the Φ-testors support sets, which were the support sets that obtained the best results, are depicted.

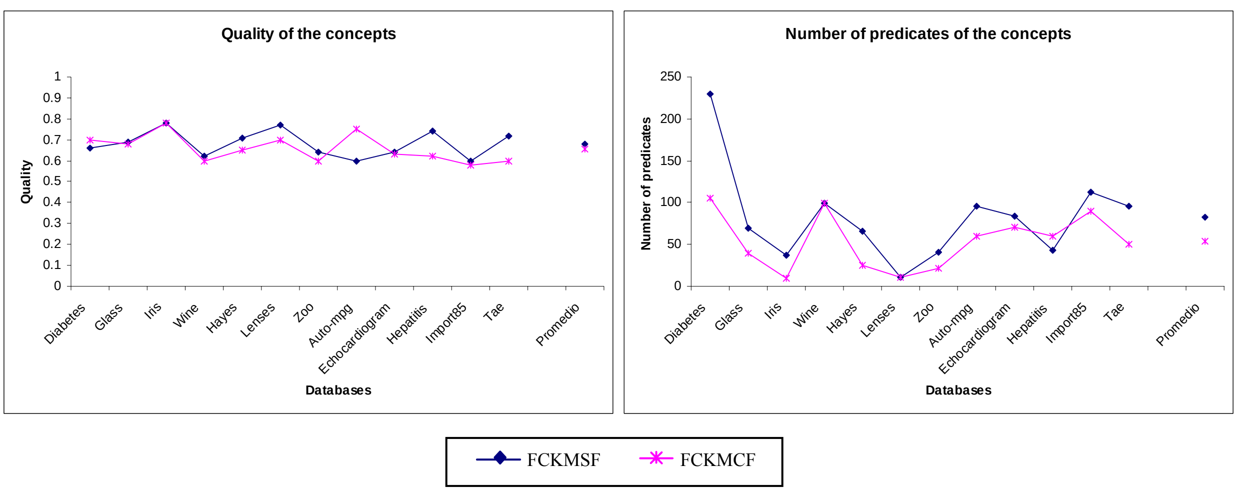

Fig. 4. Quality and number of fuzzy predicates of the concepts obtained by the FCKMSF and FCKMCF (with Φ-testors support sets) algorithms

In Table 2 and Figure 4, we can observe that, the FCKMCF algorithm using Φ-testors support sets obtained, in average, concepts with a slightly lower quality than the quality of the concepts obtained by the FCKMSF algorithm; however, the FCKMCF does not require generalization lattices and it obtains concepts with less predicates.

4 Conclusions and Future Work

4.1 Conclusions

In this thesis, two restricted conceptual clustering algorithms (CKMSF and CKMCF) and fuzzy versions of them (FCKMSF and FCKMCF) were proposed.

We observed, from the experimentation, that the CKMSF and CKMCF algorithms obtained concepts with better quality than those obtained by the CKM algorithm. Also, the CKMCF algorithm generated concepts with less number of predicates than those obtained by the CKM and CKMSF algorithms.

We can conclude that the CKMCF algorithm is a good alternative for solving restricted conceptual clustering problems when the objects are described by mixed and missing data. On the other hand, when the generalization lattices are known, the CKMSF algorithm is a good alternative for solving this kind of problems.

In the experimentation with the proposed fuzzy conceptual clustering algorithms we observed that the FCKMCF algorithm using Φ-testors support sets obtained, in average, concepts with a slightly lower quality than the quality of the concepts obtained by the FCKMSF algorithm; however, The FCKMCF algorithm does not require generalization lattices and it obtains concepts with less number of predicates.

Finally, we can conclude that the FCKMSF and FCKMCF algorithms are a first approximation for solving fuzzy restricted conceptual clustering problems where the objects are described by mixed and missing data.

4.2 Future Work

Based on the experimental results we observed that the proposed algorithms obtain concepts with good quality; nevertheless, these qualities can be improved. For that reason, we propose to find new strategies, in the characterization phase, for generating better concepts.

The proposed function for evaluating the quality of the concepts takes into account only the number of objects covered by the concepts, without taking into account their size. As future work we propose to define a quality function that takes into account both characteristics.

References

1. **Alba-Cabrera E. (1997),** *Nuevas extensiones del concepto de testor para diferentes tipos de funciones de semejanza*. Tesis para obtener el grado de Doctor en Ciencias Matemáticas, ICIMAF, Cuba.
2. **Ayaquica-Martínez I.O., Martínez-Trinidad J.F. (2001),** *Fuzzy c-means algorithm to analyze mixed data*. VI Iber-american Symposium on Pattern Recognition. Florianopolis, Brazil, pp. 27-33.
3. **Béjar J., Cortés U. (1992),** *LINNEO+: Herramienta para la adquisición de conocimiento y generación de reglas de clasificación en dominios poco estructurados*. En las memorias del 3er. Congreso Iberoamericano de Inteligencia Artificial. La Habana Cuba, pp. 471-481.
4. **Blake C., Keogh E., Merz C.J. (1998),** *UCI Repository of Machine Learning Databases*. http://www.ics.uci.edu/~mlearn/MLRepository.html. Irvine CA: University of California, Department of Information and Computer Science.
5. **De-la-Vega-Doria L.A. (1994),** *Extensión al caso difuso del algoritmo de clasificación Kora-3*. Tesis para obtener el grado de Maestro en Ciencias en especialidad en Ingeniería Eléctrica, CINVESTAV, México.
6. **Fisher D. (1990),** *Knowledge acquisition via incremental conceptual clustering*. Shavlik and Dietterich editors. Readings in Machine Learning, pp. 267-283.
7. **García-Serrano J.R., Martínez-Trinidad J.F. (1999),** *Extension to c-means algorithm for the use of similarity functions*. 3rd European Conference on Principles of Data Mining and Knowledge Discovery Proceedings. Prague, Czech. Republic, pp 354-359.
8. **Gennari J.H., Langley P., Fisher D. (1990),** *Model of incremental concept formation*. In J. Cabonell. MIT/Elsevier Machine Learning, paradigms and methods, pp. 11-61.

9. **Guevara-Cruz M. E. (2004),** *Genetic Algorithm for feature selection and informational weight computation using the fuzzy FS testor concept*. Tesis para obtener el grado de Maestro en Ciencias de la Computación, Facultad de Computación, BUAP, México.
10. **Hanson S.J. (1990),** *Conceptual clustering and categorization: bridging the gap between induction and causal models*. In Y. Kodratoff and R.S. Michalski, editors. Machine Learning: an artificial intelligence approach, vol. 3, Morgan Kaufmann, Los Altos CA, pp. 235-268.
11. **Lebowitz M. (1986),** *Concept learning in a rich input domain: Generalization based memory*. In R.S. Michalski, J.G. Carbonell and T.M. Mitchell, editors. Machine Learning: an artificial intelligence approach, vol.2, Morgan Kaufmann, Los Altos, CA, pp. 193-214.
12. **Martínez-Trinidad J.F. (2000),** *Herramientas para la Estructuración Conceptual de Espacios*. Tesis para obtener el grado de Doctor en Ciencias de la Computación, CIC, IPN, México.
13. **Martínez-Trinidad J.F., Ruiz-Shulcloper J. (1998),** *Fuzzy LC conceptual algorithm*. In proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing. Aache, Germany, pp. 20-24.
14. **Martínez-Trinidad J.F., Sánchez-Díaz G. (2001),** *LC a conceptual clustering algorithm*. International Workshop on Machine Learning and Data Mining in Pattern Recognition. Leipzig, Germany, pp. 117-127.
15. **McKusick K., Thompson K. (1990),** *Cobweb/3: A portable implementation*. Technical report FIA-90-6-18-2, NASA Ames Research Center.
16. **Michalski R.S. (1980),** *Knowledge adquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts,* (special issue on knowledge acquisition and induction). Policy Analysis and Information Systems 3, pp. 219-244.
17. **Michalski R.S. (1983),** *Automated construction of classifications: conceptual clustering versus numerical taxonomy*. IEEE transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-5 4.
18. **Michalski R.S. (1986),** *A theory and methodology of inductive learning*. In R.S. Michalski, J. G. Carbonell and T. M. Mitchell, editors. Machine Learning: An artificial intelligence approach, volume 2, Morgan Kaufmann, Los Altos, CA, pp. 83-129.
19. **Michalski R.S., Diday E. (1981),** *A recent advance in data analysis: Clustering objects into classes characterized by conjunctive concepts*. Progress in Pattern Recognition L.N. Kanal and A. Rosenfeld. North Holland Publishing Company, pp. 33-56.
20. **Michalski R.S., Stepp R.E. (1983),** *Learning from observation: Conceptual clustering*. In R.S. Michalski, J.G. Carbonell and T.M. Mitchell, editors. Machine Learning: An artificial intelligence approach 1, pp. 331-363.
21. **Pons-Porrata A. (1999),** *RGC: Un nuevo algoritmo de caracterización conceptual*. Tesis para obtener el grado de Maestro en Ciencias de la Computación, Universidad de Oriente, Cuba.
22. **Pons-Porrata A., Ruiz-Shulcloper J., Martínez-Trinidad J.F. (2002),** *RGC: a new conceptual clustering algorithm for mixed incomplete data sets*. In Mathematical and Computer Modelling 36, pp. 1375-1385.
23. **Quan T. T., Hiu S. C., Cao T. H. (2004),** *A Fuzzy FCA-based Approach to Conceptual Clustering for Automatic Generation of Concept Hierarchy on Uncertainty Data*, CLA 2004, pp. 1-12.
24. **Quan T. T., Hui S. C. Cao T. H. (2004),** *FOGA: A Fuzzy Ontology Generation Framework for Scholarly Semantic Web*. In Proceedings of the Knowledge Discovery and Ontologies Workshop, Pisa, Italy.
25. **Ralambondrainy H. (1995),** *A conceptual version of the K-means algorithm*. Pattern Recognition Letters 16, pp. 1147-1157.
26. **Santos-Gordillo J. A., Carrasco-Ochoa J. A., Martínez-Trinidad J. F. (2003),** Computing Fuzzy Φ-Testors using a genetic algorithm, WSEAS Transactions on Systems 4/2 pp. 1068-1072.
27. **Seeman W. D., Michalski R. S. (2006),** *The CLUSTER/3 system for goal-oriented conceptual clustering: method and preliminary results*. Proceedings of The Data Mining and Information Engineering 2006 Conference, Prague, Czech Republic, vol. 37, pp. 81-90.
28. **Stepp R.E., Michalski R.S. (1986),** *Conceptual clustering: inventing goal oriented classifications of structured objects*. In R.S. Michalski, J.G. Carbonell and T.M. Mitchell, editors. Machine Learning: an artificial intelligence approach, vol.2, Morgan Kaufmann, Los Altos, CA, pp. 471-498.

Irene Olaya Ayaquica Martínez *received her BS degree in Computer Science from the Computer Science School of the Autonomous University of Puebla (BUAP), Mexico in 1998; her MSc degree in Computer Science from the Center for Computing Research of the National Polytechnic Institute (CIC, IPN), Mexico in 2002; and her PhD degree in Computer Science from the National Institute of Astrophysics, Optics and Electronics (INAOE), Mexico in 2007. Her research interests include: Logical Combinatorial Pattern Recognition, Clustering, Fuzzy Clustering, Conceptual Clustering and Fuzzy Conceptual Clustering.*

José Francisco Martínez Trinidad *received his B.S. degree in Computer Science from Physics and Mathematics School of the Autonomous University of Puebla (BUAP), Mexico in 1995, his M.Sc. degree in Computer Science from the faculty of Computers Science of the Autonomous University of Puebla, Mexico in 1997 and his Ph.D. degree in the Center for Computing Research of the National Polytechnic Institute (CIC, IPN), Mexico in 2000. Professor Martinez-Trinidad edited/authored four books and over fifty papers, on subjects related to Pattern Recognition.*

Jesús Ariel Carrasco Ochoa *received his PhD degree in Computer Science form the Center for Computing Research of the National Polytechnic Institute (CIC-IPN), Mexico, in 2001. He works as full time researcher at the National Institute for Astrophysics, Optics and Electronics of Mexico. His current research interests include, Logical Combinatorial Pattern Recognition, Testor Theory, Feature and Prototype Selection, and Clustering.*