Conceptual K-Means Algorithm with Similarity Functions [*]

I.O. Ayaquica-Martínez, J.F. Martínez-Trinidad, and J.A. Carrasco-Ochoa

National Institute of Astrophysics, Optics and Electronics,
Computer Science Department, Luis Enrique Erro # 1,
Santa María Tonantzintla, Puebla, Mexico, C.P. 72840
{ayaquica,fmartine,ariel}@inaoep.mx

Abstract. The conceptual k-means algorithm consists of two steps. In the first step the clusters are obtained (aggregation step) and in the second one the concepts or properties for those clusters are generated (characterization step). We consider the conceptual k-means management of mixed, qualitative and quantitative, features is inappropriate. Therefore, in this paper, a new conceptual k-means algorithm using similarity functions is proposed. In the aggregation step we propose to use a different clustering strategy, which allows working in a more natural way with object descriptions in terms of quantitative and qualitative features. In addition, an improvement of the characterization step and a new quality measure for the generated concepts are presented. Some results obtained after applying both, the original and the modified algorithms on different databases are shown. Also, they are compared using the proposed quality measure.

1 Introduction

The conceptual clustering concept surged at 80's with the Michalski's works [1]. The conceptual clustering consists on finding, from a data set, not only the clusters but an interpretation of such clusters.

There are some algorithms to solve the conceptual clustering problem [1,2]; one of them is the conceptual k-means algorithm [2].

The conceptual k-means algorithm, proposed by Ralambondrainy in 1995, is a method that integrates two algorithms: 1) an extended version of the well known k-means clustering algorithm for determining a partition of a set of objects described in terms of mixed features (*aggregation step*), and 2) a conceptual characterization algorithm for the intentional description of the clusters (*characterization step*).

In the aggregation step, a distance function to simultaneously deal with qualitative and quantitative features is defined. The distance between two objects is evaluated as a weighted sum of the distance among the quantitative features, using the normalized Euclidean distance, and the distance among the qualitative features, using the chi-square distance.

[*] This work was financially supported by CONACyT (México) through the project J38707-A.

M. Lazo and A. Sanfeliu (Eds.): CIARP 2005, LNCS 3773, pp. 368 – 376, 2005.

This way to define the distance function is inappropriate because it requires transforming of each qualitative feature in a set of Boolean features. The values of these new features are codes but they are deal as numbers, which is incorrect. For this reason, in this paper, we propose to use a different strategy to obtain the clusters.

In the characterization step, it is necessary to define, for each feature, a generalization lattice, which defines a relation among the values of the feature. We consider that the lattice defined for the quantitative features is incorrect, because it does not satisfy the definition of a generalization lattice. Therefore, we propose a new generalization lattice. The definition of a generalization lattice is given in section 2.2.

This paper is structured in the following way: in section 2, a description of the conceptual k-means algorithm is presented. In section 3, a new conceptual k-means algorithm using similarity functions is proposed. In section 4, the results obtained after applying both algorithms over different databases are shown. In section 5, conclusions and future work are presented.

2 Conceptual K-Means Algorithm (CKM)

In this section, a description of the Ralambondrainy's conceptual k-means algorithm is presented. As we have mentioned above, the CKM algorithm consists of two steps: the aggregation and the characterization steps. These steps are described in sections 2.1 and 2.2 respectively.

2.1 Aggregation Step

The goal of the aggregation step is to find a partition $\{C_1,...,C_k\}$ in *k* clusters of the data set Ω. This algorithm is based on an extension of the well known k-means algorithm in order to allow working with objects described in terms of mixed features.

As a comparison function between objects, a distance function is defined, which is given by a weighted sum of the normalized Euclidean distance (for quantitative features):

$$\delta^2_{1/\sigma^2}(O,O') = \sum_{1\le i\le p} \frac{(x_i(O)-x_i(O'))^2}{\sigma_i^2}$$

where σ_i denotes the standard deviation of the *i*th feature. And the chi-square distance (for qualitative features). In order to apply the chi-square distance, a transformation of each qualitative feature in a set of Boolean features to deal them as numbers, is carried out. Therefore, the chi-square distance is given as follows:

$$\delta^2_{\chi^2}(O,O') = \sum_{1\le j\le q} \frac{(x_j(O)-x_j(O'))^2}{\eta_j}$$

where $q=\sum_{i=1}^{s}|D_i|$ and η_j is the number of objects taking the value 1 in the *j*th modality. This distance gives more importance to rare modalities than to frequent ones.

Then, in order to work with mixed data, the following distance was proposed:

$$d^2(O,O')=\pi_1\delta^2_{1/\sigma^2}(O,O')+\pi_2\delta^2_{\chi^2}(O,O')$$

where π_1 y π_2 are weights for balancing the influence of quantitative and qualitative features. In [3] a strategy to select the values for the weights π_1 and π_2 is shown.

This algorithm requires transforming the qualitative features in sets of Boolean features. The values of these new features are codes (0 or 1) but they are dealt as numbers, which is incorrect because the codes 0 and 1 are not in the real space.

In addition, this algorithm always uses this distance to manipulate mixed data, not giving the flexibility of using the comparison function which is more suitable for the problem that is being solved.

On the other hand, the centroids obtained by the algorithm are elements that cannot be represented in the same space in which the objects of the sample are represented, the averages obtained by k-means for the qualitative features are real values not 0's and 1's, so that, it is not possible to return to the original representation space.

For this reasons, we propose to use a different strategy in the aggregation step, which is presented in section 3.1.

2.2 Characterization Step

In order to apply the characterization step, a generalization lattice is associated to each feature. A generalization lattice is defined as follows: a generalization lattice is a structure $L = (E,\leq,\vee,\wedge,*,\varnothing)$, where E is a set of elements called *the search space*, $\leq$ is a partial order relation "*is less general than*", which redefines the inclusion relation as follows: $\forall e,f\in E, e\leq f \Rightarrow e\subseteq f$, the symbol * denotes the greatest member of E and it is interpreted as "*all values are possible*" and $\varnothing$ denote the minimal element of E and it is interpreted as "*impossible value*". Every (e,f) has a least upper bound that is denoted by $e\vee f$ called also the generalization of e and f, and a greatest lower bound of e and f denoted by $e\wedge f$ [2].

The generalization lattice for the qualitative features is defined by the user from the available background knowledge. While for the quantitative features, a code or transformation into qualitative features through a partition of the values domain is carry out.

For each cluster C obtained in the aggregation step, a value r of x is typical for this cluster if it satisfies:

$$\mu_x-\sigma_x\leq r\leq\mu_x+\sigma_x$$

where μ_x is the mean of the feature x in the cluster C and σ_x is the standard deviation of x in the cluster C.

Therefore, a coding function $c:\Re\rightarrow\{inf, typical, sup\}$ is defined as:

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

The generalization lattice for the quantitative features, associated to the search space $E = \{inf, typical, sup\}$, is shown in figure 1 a).

The values of μ_x and σ_x are calculated with the objects of the cluster. This fact originates a problem when the predicate $\hat{A}_C$ is compared with the counterexamples, because the values *inf*, *typical* and *sup* are syntactically similar but semantically different among clusters. In other words, these values represent different intervals depending on the cluster that is analyzed. For this reason, in this paper, some modifications to this step are proposed. These modifications are presented in the section 3.2.

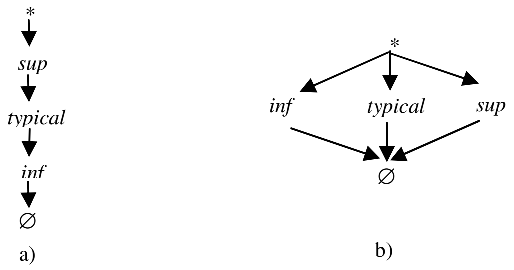


Fig. 1. a) Generalization lattice for the conceptual k-means, b) generalization lattice for the conceptual k-mans with similarity functions

3 Conceptual K-Means Algorithm with Similarity Functions (CKMSF)

In this section, some modifications to the CKM algorithm are presented. In section 3.1, the new strategy to obtain clusters is described and in section 3.2, we propose a new generalization lattice for the quantitative features in characterization step.

3.1 Aggregation Step

In this paper, we propose to use the k-means with similarity functions algorithm (KMSF) [4], for the aggregation step, instead of the original strategy used by the CKM algorithm.

This algorithm allows working in a more natural way (without transforming the space) with mixed features. For each feature, in order to compare its values, a comparison function is defined, which is denoted by $C : D_i \times D_i \rightarrow [0,1]$. This function is given in dependence of the nature of the feature.

The similarity function, used in this paper, to compare two objects is:

$$\Gamma(O_i, O_j) = \sum_{x_t \in R} C_t\left(x_t(O_i), x_t(O_j)\right) \Big/ |R|$$

where $C_t\left(x_t(O_i), x_t(O_j)\right)$ is the comparison function defined for the feature x_t.

The similarity functions are more appropriate than the defined distance function for the aggregation step of the CKM algorithm, because this function does not require transforming features. In addition, they could be defined in terms of comparison functions, which allow expressing how the values of the features are compared in the problem to solve. It is more reasonable than using a fixed function for all the problems.

On the other hand, the KMSF algorithm, selects the centroids of the clusters as objects of the sample instead of centroids, which are in a different representation space, as occur in the k-means algorithm. Considering an object of the sample as the centroid of the cluster is more reasonable than using an element that cannot be represented in the same space.

3.2 Characterization Step

The generalization lattice given by Ralambondrainy [2], for the qualitative features does not satisfy $\forall e, f \in E, e \leq f \Rightarrow e \subseteq f$, because *inf* ≤ *typical* does not imply that the interval of values represented by the *inf* label are contained in the interval of values represented by the *typical* label and *typical* ≤ *sup* does not imply that the interval of values represented by the *typical* label are contained in the interval of values represented by the *sup* label; because these intervals are excluding (see expression (1)). Then, the concepts obtained with this generalization lattice do not represent appropriately the objects in the clusters. Therefore, we propose to use the generalization lattice shown in figure 1 b).

This generalization lattice satisfies that $\forall e, f \in E, e \leq f \Rightarrow e \subseteq f$, because *inf* ≤ * ⇒ *inf* ⊆ *, *typical* ≤ * ⇒ *typical* ⊆ * and *sup* ≤ * ⇒ *sup* ⊆ *, which allows working in a more appropriate way with the quantitative features.

As we have mentioned above, the values of μ_x and σ_x depend of the cluster. Therefore, it is not appropriate to only take the labels *inf*, *typical* and *sup*, but it is also necessary to verify if the value for the feature *x*, for the object that is being analyzed, is inside the range of values for the label of the feature *x* into the cluster.

Another way to define the coding function for the quantitative features is using the mean μ_x and the standard deviation σ_x as global. This allows measuring the range of values of the feature with respect to the total sample of objects. In addition, in this case verifying if an object is covered by the generated concept; it is equivalent to take the labels or the ranges of the labels, because these values do not depend on the cluster.

We consider that taking μ_x global and σ_x local or μ_x local and σ_x global does not make sense, because this values would be evaluating in different levels, i.e., one value is evaluated with respect to the cluster and the other one is evaluated with respect to the whole sample of objects.

Therefore, the proposed conceptual k-means algorithm uses, in the aggregation step, a similarity function given in terms of comparison functions which allows expressing how the features are compared depending on the problem to solve. Also, the centroids are objects in the sample and not elements that cannot be represented in the same space where the objects of the sample are represented.

On the other hand, in the characterization step a new generalization lattice for the quantitative features was introduced.

Finally, we consider that it is important to have a way to evaluate the quality of the concepts. Ralambondrainy [2] proposed to take as quality measure for the concepts the percentage of objects of the cluster that are recognized by the concept. However, it is also necessary to take into account the objects outside the cluster that are recognized by the concept. Therefore, we propose the following quality measure:

$$quality = \sum_{i=1}^{c} examples_i \Bigg/ total + \sum_{i=1}^{c} counterexamples_i$$

where:

$examples_i$: is the number of objects in the cluster C_i that are covered by the concept;

$counterexamples_i$: is the number of objects outside of the cluster C_i that are covered by the concept;

$total$: is the number of objects in the sample.

This function obtains higher values if the number of examples covered by the concept increases and the number of counterexamples covered by the concept decreases. And vice versa, the function obtains lower values if the number of examples covered by the concept decreases and the number of counterexamples covered by the concept increases.

4 Experimentation

Initially, some tests with the aggregation step were carried out. The k-means algorithm and the KMSF were applied over different databases and the obtained results were compared. The Iris, Glass, Ecoli, Tae, Hayes, Lenses and Zoo databases were used for the tests; these databases are supervised and they were taken from the UCI repository [5].

Table 1. Percentages of classification obtained by both algorithms over different databases

		k-means algorithm		KMSF algorithm	
Data-bases	**Number of objects**	**% of objects well classified**	**% of objects bad classified**	**% of objects well classified**	**% of objects bad classified**
Iris	150	85.33%	14.67%	90.67%	9.33%
Glass	214	34.11%	65.89%	45.79%	54.21%
Ecoli	336	33.04%	66.96%	42.26%	57.74%
Tae	151	36.42%	63.58%	61.59%	38.41%
Hayes	132	36.36%	63.64%	46.97%	53.03%
Lenses	24	41.67%	58.33%	41.67%	58.33%
Zoo	101	71.29%	28.71%	79.21%	20.79%

In table 1, the results obtained in the aggregation step after applying the k-means and the KMSF algorithms over the databases are shown. The obtained clusters were compared against the original classification.

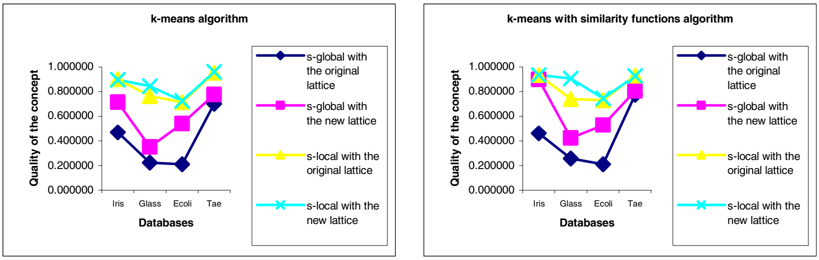
In table 1, we can observe that the classification obtained by the KMSF algorithm has more well classified objects, in most of the cases, than the classification obtained by the k-means algorithm. This is due to the form in which the objects are compared with the centroids, also that the centroids are objects in the sample instead of being in a different representation space.

Later some tests with the characterization step, using the Iris, Glass, Ecoli and Tae databases were carried out. These databases contain quantitative data. The tests were made taking the global mean and standard deviation and taking the local mean and standard deviation. The characterization step was applied over the clusters obtained in the aggregation step by the k-means and the KMSF algorithms

In addition, an analysis of the parameters α (maximum number of counterexamples that could be covered by a predicate) and β (minimum number of examples that must be covered by a predicate) was carried out. In this analysis, we observed that for small values of β more objects of the cluster are covered by the concept but the concepts are larger and therefore, more difficult to understand. While, for big values of β it could happen that for some clusters any concept could be generated.

In figure 2 a), the results obtained after applying the characterization step over the clusters created by the k-means algorithm taking σ_x global and σ_x local, and both lattices, the original and the new, are shown. In figure 2 b), the results obtained after applying the characterization step over the clusters obtained by the new conceptual k-means algorithm taking σ_x global and σ_x local, and both lattices, the original and the new, are shown. The results shown in figure 2 are for those values of α and β, which obtain the highest concept quality.

In figure 2, we can observe that for σ_x global, the obtained concepts using the new lattice are better than the obtained concepts using the original lattice, according to the proposed quality measure.



a) b)

Fig. 2. Results obtained in the characterization step: a) for the CKM algorithm and b) for the CKMSF

When the σ_x local is taking, this improvement in the concept quality is not so clear (see figure 2). However, with the new lattice, the concept quality does not depend so much of the parameters α and β, because for any value of α and β the concepts obtain a high quality (see figure 3), which does not occur when the original lattice is used (see figure 4). In that case, it was necessary to do a good selection of the parameters α and β, to obtain concepts with high quality.

Only the results obtained using the Iris database are shown (figures 3 and 4). However, the Glass, Ecoli and Tae databases have a similar behavior.

In addition, some tests with the Hayes, Lenses and Zoo databases, that contain only qualitative information, were carried out. In this case, we only compare the concept quality obtained by the CKM and the CKMSF algorithms because the new generalization lattice for quantitative features does not influence in the qualitative features. Only the results obtained with the Hayes database are shown (figure 5). However, the Lenses and Zoo databases have a similar behavior.

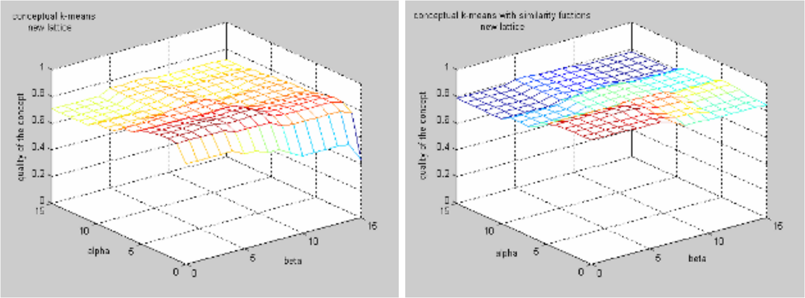


Fig. 3. Results obtained by the characterization step of both algorithms applied over the Iris database, using the new lattice and for values of α and β between 0 and 15

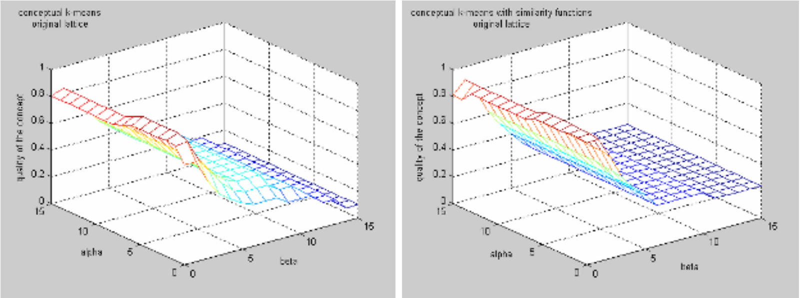


Fig. 4. Results obtained by the characterization step of both algorithms applied over the Iris database, using the original lattice and for values ovf α and β between 0 and 15

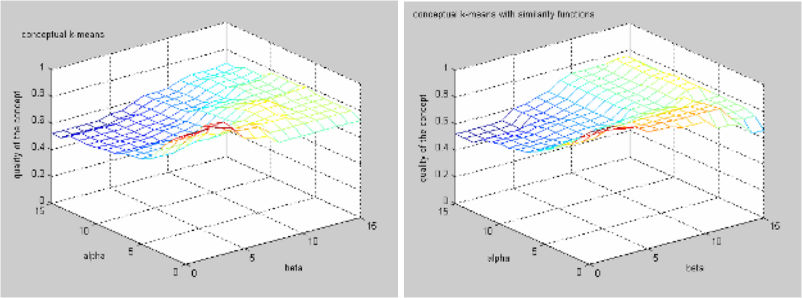


Fig. 5. Results obtainedin th e characterization step, applied over both algorithms, using the Hayes database, for values of α and β between 0 and 15

In figure 5, we can observe that the concepts obtained for the CKMSF have similar quality than those obtained by the CKM even when the clusters obtained, in the aggregation step, by the k-means algorithm are different than the clusters obtained by the KMSF.

5 Conclusions and Future Work

In this paper, we proposed a new conceptual k-means algorithm using similarity functions, which allows dealing in a more natural way with objects described in terms of mixed features.

This algorithm uses, in the aggregation step, the k-means algorithm with similarity functions (KMSF). The KMSF uses a similarity function defined in terms of comparison functions for features, which allow expressing how the values for the features are compared, in the problem to solve. Also, this function does not require transforming the features.

In addition, the centroids of the clusters are objects in the sample instead of elements that cannot be represented in the same space where the objects of the sample are represented.

On the other hand, in the characterization step, we proposed a new generalization lattice, which allows dealing with quantitative features in a more appropriate way.

Besides, we proposed a function to evaluate the quality of the concepts. This function takes into account both the objects into the cluster that are covered by the concept and the objects outside the cluster that are covered by the concept.

Based on the experimentation, we observed that using the new lattice we obtained concepts with a high quality, independently of the values for the parameters α and β, which did not happen when the original lattice was used. In this case, it is necessary to do a good selection of the parameters to obtain concepts with high quality.

As future work, we are working in other way to obtain the characterization of the clusters and in a fuzzy version of the conceptual k-means algorithm with similarity functions.

References

1. Hanson S.J. 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, pp. 235-268. Morgan Kaufmann, Los Altos CA. (1990).
2. Ralambondrainy H., A conceptual version of the K-means algorithm, Pattern Recognition Letters 16, pp. 1147-1157 (1995).
3. Ralambondrainy H., A clustering method for nominal data and mixture of numerical and nominal data. Proc. First Conf. Internat. Federation of Classification Societies, Aachen (1987).
4. García Serrano J.R. and Martínez-Trinidad J.F., Extension to k-means algorithm for the use of similarity functions. 3rdEuropean Conference on Principles of Data Mining and Knowledge Discovery Proceedings. Prague, Czech. Republic, pp 354-359. (1999).
5. http://www.ics.uci.edu/pub/machine-learning-databases/