Journal of AI and Data Mining
Vol 6, No 1, 2018, 163-175

Grouping Objects to Homogeneous Classes Satisfying Requisite Mass

M. Manteqipour[1*], A. Ghaffari Hadigheh[1], R. Mahmoudvand[2] and A. Safari[3]

1. Mathematics, Azarbaijan Shahid Madani University, Tabriz, Iran.
2. Department of Statistics, Bu-Ali Sina University, Hamedan, Iran.
3. Insurance Research Center (Affiliated to the central insurance of Iran), Tehran, Iran.

Received 01 September 2016; Revised 25 January 2017; Accepted 15 March 2017
**Corresponding author: manteqipour@azaruniv.edu (M. Manteqipour).*

Abstract
Grouping datasets play an important role in many scientific research works. Depending on the data features and applications, different constrains are imposed on groups, while having groups with similar members is always a main criterion. In this paper, we propose an algorithm for grouping the objects with random labels, nominal features having too many nominal attributes. In addition, the size constraint on groups is necessary. These conditions lead to a mixed integer optimization problem that is neither convex nor linear. It is an NP-hard problem, and exact solution methods are computationally costly. Our motivation to solve such a problem comes along with grouping the insurance data, which is essential for fair pricing. The proposed algorithm includes two phases. First, we rank random labels using fuzzy numbers. Afterwards, an adjusted K-means algorithm is used to produce the homogenous groups satisfying a cluster size constraint. Fuzzy numbers are used to compare random labels, in both the observed values and their chance of occurrence. Moreover, an index is defined to find the similarity of multi-valued attributes without perfect information with those accompanied with perfect information. Since all ranks are scaled into the interval [0,1], the result of ranking random labels does not require rescaling techniques. In the adjusted K-means algorithm, the optimum number of clusters is found using the coefficient of variation instead of the Euclidean distance. Experiments demonstrate that our proposed algorithm produces fairly homogenous and significantly different groups having the requisite mass.

Keywords: *Classification, Clustering, Fuzzy Numbers, Homogenous Groups, K-means Algorithm.*

1. Introduction

Generally, data grouping provides a powerful tool for managers and researchers. Here, a large number of objects are divided into a few groups such that the members of each group are similar as much as possible, while they have the most dissimilarity with the members of the other groups. Exploiting members' common behavior of a homogenous group empowers mangers in decision-making in many practical cases as well as in scientific studies. In this work, we aim to group a dataset into homogenous groups satisfying requisite of mass that enables using the law of large numbers. The objects of this dataset are labeled with random variables.

Profitability and retaining a certain level of market share are two main goals in policy pricing strategies of all insurance industries [1, 2]. For devising a plan to accomplish these goals, having homogenous and insurable risk groups is important, and an optimum pricing plan usually follows the average behavior of such groups in several aspects such as claim cost, effect of premium variation on customers' retention rates, and demand functions.

Consequently, we are motivated to find a method to group the database of cargo insurance policies to insurable homogenous risk groups. Here, policies are objects, and the cost of claims is their labels that can be assumed as random variables with non-negative outcomes. Our dataset contains a large number of nominal attributes, while the attributes of some features can be merged to produce a new integrated attribute. For instance, when a policy-holder uses airplane, lorries, and trains altogether for transportation, these three attributes are incorporated to construct a multi-

valued attribute, and consequently, such an attribute must be considered different from its components. In other words, the transporting mean of a policy as a feature might have multi-valued attributes, while the type of commodity is a feature with simple attributes.

Homogeneity in a group of this dataset may be defined as the similarity of different policies on the basis of cost of claims, which are considered as the random variables. Furthermore, insurability imposes the condition of having enough members in each group [2].

Two main approaches exist for constructing a homogenous collection of objects, classification and clustering. While classification is a supervised learning algorithm [3], clustering methods are categorized as unsupervised learning methods [4]. Clearly speaking, the clustering algorithms are used for the unlabeled objects, while the classification methods are applied to predict the labels of new objects using the pattern of some other samples.

It is worth mentioning that the labels of objects in this work are random variables, while the popular classification methods are applied for objects with crisp labels. Further, the classification methods such as the SVM [3] techniques divide the dataset with respect to these labels, and as a main characteristic, the classification methods typically have no control on the size of classes.

Having objects with nominal attributes may tempt one to apply one of the existing clustering methods such as *K*-modes [5, 6] and *K*-prototype [7]. These methods only consider two possibilities for the nominal attributes, similar and dissimilar. Therefore, by minimizing the intra-dispersion in all clusters, objects are included in a group when most of their features are the same. By this consideration, the impacts of dissimilar attributes on random labels are ignored. Furthermore, when the possible number of values for nominal variables is too many, it may lead to clusters with a low similarity between objects.

Another option to deal with too many nominal attributes might be clustering the attributes of each feature. For example, this idea has been applied in a heuristic fashion on the automobile policies with 4 features; each categorized into 3 attributes [8]. As a result, 3^4 risk groups are identified, while some of them are not insurable. Besides, labels (cost of claims in our case) do not play any role on specifying the clusters in this method.

Recall that the value of claim on each policy as a label can be considered as a random variable. This imposes a sort of uncertainty on the problem. In order to deal with such a dataset, the idea of probabilistic databases [9] strikes. For ranking the tuples with respect to scores, claim rates in this work, Li et. al [10, 11] introduced a method using a parameterized ranking function, defined as follows:

$$\gamma_w(t) = \sum_{i>0} w(i,t)\Pr(r(t)=i),$$

in which, $\Pr(r(t)=i)$ is the probability where t stands in position i and $w(i,t)$ is a weight function. The higher the value for $|\gamma_w(t)|$, the higher the rank. In this approach, the authors devised a method to determine the priorities of tuples according to scores, while evaluation of the distance between tuples was not their concern. Consider the situation where there are tuples $t_1,\ldots,t_n$ with uniform distributions such that with certainty they sit in positions 1 to n, respectively. Recalling the notion of support of a probability distribution f as $\{x \in D_f ; f(x) \neq 0\}$, this means that supports of their distribution probabilities have empty intersections. Subsequently, $\gamma_w(t) = w(i,t_i)$. If w does not depend on the distribution of scores of t as well as the location of values of scores, as defined in [10, 11], no matter how far the scores of tuples are from each other, the values for $|\gamma_w(t)|$ are determined as the same. Since our final aim is to cluster the objects respecting their distances, applying this model emerges an additional challenge on determining a suitable w , which makes it potentially impractical for our goal.

Since distance plays a vital role in clustering algorithms, distance of probability distributions defined in [12] could be another option for determining the distance between random variables. As a measure, f-divergences are not symmetric, and do not need to satisfy triangular inequality. Recall that they are applied for finding the distance of probability distributions with common supports, and consequently, they are independent from the values of random variables. For example, consider the case of having uniform random variables $U(l_i,u_i)$ $i=1,2,\ldots,n$ with $u_i < l_{i+1}$ $i=1,\ldots,n-1$; f-divergence functions take equal values regardless of the distances between the support sets. As a result, the f-divergence function is a weaker notion than the distance, in addition to their other properties, which make them less effective in addressing our problem.

For clustering a database with random labels and nominal attributes, we need a method to compare the random variables considering both the location of their values and their chance for occurrence. For simultaneously respecting these two criteria, fuzzy numbers capability is an appropriate option. The idea of fuzzy sets was introduced by Zadeh [13] to deal with the uncertainty by extending the concept of membership to a set. The membership function helps to consider the trustworthy of the claim cost occurrence. Moreover, by comparing α-cuts, the location of claim rate values can be considered in calculation of the distance between random variables.

Since we use fuzzy numbers for labels with random values, their ranking necessitates a method for fuzzy numbers' ranking. Several methods exist in the literatures for this purpose, each of which has some advantages and disadvantages. Their ability in distinguishing different fuzzy numbers can be considered as the most important criterion to choose one of them in practical applications.

The proposed method in [14] is applied in this research work that uses α-cuts instead of local information. Recall that in the clustering methods, the range of variables must be comparable. Since all the ranks obtained are in [0,1], rescaling the data, as suggested in [15, 16], is superfluous. Our experiments denote that this method intuitively produces more consistent and sensible results.

K-means is one of the most popular clustering algorithms for ordinal data [17].

According to [18], it is the second top algorithm among the top-10 data mining algorithms. As an achievement, it produced a more satisfying outcome in a case study for claim cost prediction in the automobile insurance industry [2]. In this work, we need to have clusters satisfying requite mass, while the number of clusters is not known in advance.

There are some research works aimed to limit the number of members in each group (e.g. [19]). Their modification is based on sorting the distance of objects from centers in ascending order, and including an object in a group with the least distance while not violating the size restriction.

This adapted algorithm maintains the mass of groups less than a pre-determined volume; however, after identifying the centroid, there is no control on dispersion of objects in each group. Here, we adjust the K-means algorithm in another sense; it identifies the number of clusters such that the ratio of intra-dispersion index to the centers-dispersion index is minimized.

The rest of the paper is organized as what follows. In Section 2, the understudied problem is stated. Section 3 explains our proposed method. It is applied in a motivating problem with real data in Section 4, and the results obtained are analyzed. The final section concludes with findings and future related works.

2. Problem statement

To clarify the problem, let A denote a dataset defined as $A=\{a^1,a^2,\ldots,a^m\}$, where $a^i \in B_1 \times B_2 \times \ldots \times B_n \quad i=1,\ldots,m$ and B_j is a set of nominal attributes available for the j^{th} dimension of an object. Therefore, each object is a vector with nominal components. In addition, the object a^i has a label $l(a^i)=r^i, \quad i=1,\ldots,m,$ which is considered as a random variable such as cost of claims in our motivating problem. Our goal is to partition A into disjoint subsets $A^j, \quad j=1,\ldots,k$ while k is not known beforehand. These subsets should satisfy the following conditions:

$$I.\left|A^j\right|>\gamma_0; \tag{1}$$
$$II.A^j \cap A^l = \varnothing \quad j,l=1,\ldots,k,\ j \neq l;$$
$$III.A=\bigcup_{j=1}^{k} A^j.$$

The sets $A^j, j=1,\ldots,k$ are called clusters in the literature. The corresponding clustering problem can be formulated as the optimization problem (2) (See [20, 21, 22]). In which, l assigns the unique random variables to the objects, $\overline{l}_j, j=1,\ldots,k$ is the center of the j^{th} cluster, and $dist(.,.)$ denotes dissimilarities of two random variables. In this problem, $\overline{l}_j$ and w_{ij} are decision variables. In an optimal solution, the binary variable w_{ij} is 1 if a^i belongs to the j^{th} cluster and 0 otherwise. For an exact definition of the distance between two random variables, their properties should be identified.

$$\min \sum_{i=1}^{m}\sum_{j=1}^{k} w_{ij}\, dist(l(a^i),\overline{l}_j)$$
$$s.t$$
$$\sum_{j=1}^{k} w_{ij}=1 \quad i=1,\ldots,m, \tag{2}$$
$$\sum_{i=1}^{m} w_{ij} \geq \gamma_0 \quad j=1,\ldots,k,$$
$$w_{ij} \in \{0,1\} \quad i=1,\ldots,m, \quad j=1,\ldots,k$$

Recall that B_j is the set of nominal attributes available for the j^{th} dimension of an object. Since there are not sufficient observations of the outcomes of $l(a^i)$, we consider the following random variables of all a^i, where their j^{th} dimension is b_j.

$$l(a^i \mid a^i_j = b_j), b_j \in B_j, j = 1,\ldots,n.$$

For the sake of simplicity, $l(a^i \mid a^i_j)$ is denoted by $l(. \mid a^i_j)$, hereafter. Therefore, the distance of two random variables is defined as follows:

$$dist(l(a^i), \overline{l_j}) := dist\left(\left(l(., a^i_1),\ldots,l(., a^i_n)\right), \overline{l_j}\right) \quad (3)$$

To evaluate the distance of two vectors with random variable components, we first construct fuzzy numbers $fuz\left(l(.,b)\right)$ from the outcomes of random variables $l(.,b), b \in B_j, j = 1,\ldots,n$. Then for all $j = 1,\ldots,n$, members of the produced fuzzy numbers set $\{fuz(l(.,b)); b \in B_j\}$ are ranked. Let $R(b) \in \mathbf{R}$ denote the rank of $b \in B_j$. In this way, the Euclidean distance can be applied as a measure for the specified ordinal values.

$$dist\left(l(a^i), \overline{l_j}\right) := dist\left((R(a^i_1),\ldots,R(a^i_n)), \overline{R_j}\right) := \left\|((R(a^i_1),\ldots,R(a^i_n)) - \overline{R_j}\right\|^2 \quad (4)$$

By considering such a distance function, problem (2) identifies a mixed integer problem that is neither linear nor convex.

In some cases, the size of B_j might be too large, and as a consequence, there are not enough observations of some $l(.,b)$ for $b \in B_j$, and the calculated $R(b)$ might be not reliable. In such cases, finding the similarity of $b \in B_j$ for those members of B that their related $l(.,b)$ has been observed enough may help to estimate the rank of $l(.,b)$. Structure of elements in B_j plays an important role in determining such similarity. In our case study, we observed two kinds of B_j's; with multi-valued, and with simple attributes. Recall that members of B_j are subsets of possible attributes, and these subsets may be non-singleton only for the multi-valued attribute. Here, a heuristic method is proposed to measure these similarities, and specify $R(b)$ for the b's with not enough observations.

After execution of the above procedure, a rank is assigned to each attribute. At this stage, we need to solve problem (2) with distance function defined in (4). If the number of clusters, k, was identified, while the third constraint in problem (2) was not satisfied, the K-means algorithm could be useful to find a local optimum. Even so, for solving problem (2) with unknown k, we propose an adjusted K-means algorithm, explained in the next section.

3. Methodology details

Our proposed method includes two main phases; ranking random labels, and data clustering.

3.1. Phase 1: Ranking random labels

In this phase, the uncertain label of each policy is first described as a triangular fuzzy number, and then these fuzzy numbers are ranked. The steps of this phase for all $B_j, j = 1,\ldots,n$ are as what follow.

Step1: Divide the members of B_j into two disjoint sets, B^e_j and B^l_j, where B^e_j is the set of members with enough large numbers of observations, more than a threshold t_0, and $B^l_j = B_j \setminus B^e_j$.

Step2: Construct the fuzzy numbers, $fuz(l(.,b))$ from the outcome $l(.,b)$, $b \in B^e_j$.

Step3: Rank all the fuzzy numbers $fuz(l(.,b))$ $b \in B^e_j$, denoted by $R(fuz(l(.,b)))$.

Determine the rank of $b \in B^l_j$ as follows:

For all $b \in B^l_j$, where B_j is a set of simple attributes, let:

$$R\left(l(.,b)\right) = \frac{\sum_{b \in B^e_j} R\left(l(.,b)\right)}{\left|B^e_j\right|}.$$

For all $b \in B^l_j$, where B_j is a set of multi-valued attributes, let:

$$R\left(l(.,b)\right) = \sum_{b \in B^e_j} w_b R(l(.,b)).$$

where, $\sum_{b \in B^e_j} w_b = 1$.

Steps 2-4 are explained in details in the sequel.

3.1.1. Step 2: Constructing fuzzy numbers

Fuzzy number construction from uncertain data mainly depends on the application. Employing the outcomes of random labels, the fuzzy numbers could be associated such that the outcomes with a higher possibility are assigned a higher membership degree, and more reliable results are expected from more accurate and informative fuzzy numbers [23]. Here, the triangular fuzzy numbers are used. A triangular fuzzy number is shown by a triple $[a_1, a_2, a_3]$ with the membership function μ defined as:

$$\mu(x) = \begin{cases} \dfrac{x-a_1}{a_2-a_1} & if\ a_1 < x < a_2 \\ \dfrac{a_3-x}{a_3-a_2} & if\ a_2 < x < a_3 \\ 1 & if\quad x = a_2 \\ 0 & otherwise \end{cases}$$

Let F_b for $b \in B_j^e, j = 1,\ldots,n$ be the crisp set of the observed outcomes of label $l(.,b)$. The fuzzy number associated with b is defined as:

$fuz(l(.,b)) := [\min(F_b), mean(F_b), \max(F_b)]$.

Depending on the application, definition of a triangular fuzzy number can be varied. For instance, the minimum and maximum values can be substituted by percentiles to ignore the outlier data.

3.1.2. Step 3: Ranking fuzzy numbers

The ranking method used in this work is due to [14]. Let X and Y be the intervals for an uncertain variable with uniform distributions $p_x(x)$ and $p_y(y)$, respectively. In order to compare these intervals, the probability $P(X \succeq Y)$ is evaluated as:

$$P(X \succeq Y) = \int_{-\infty}^{\infty} p_x(x)[\int_{-\infty}^{x} p_y(y)dy]dx. \qquad (5)$$

Calculating (5) is simplified by considering all the six different possible relative positions of X and Y [14].

For an $\alpha \in (0,1]$, the α-cut of a fuzzy number A, denoted by A_α, is the crisp set $A_\alpha = \{x \in \mathbf{R}; \mu_A \geq \alpha\}$, where μ_A is the membership function of A. Obviously, any α-cut is an interval. For comparing the two fuzzy numbers A and B based on their α-cuts by (5), an index called comparison relation is defined as:

$$P(A \succeq B) = \int_0^1 P(A_\alpha \succeq B_\alpha)d\alpha. \qquad (6)$$

Let $S = \{A_1, A_2, \ldots, A_N\}$ be a set of fuzzy numbers. The fuzzy target number T associated with S is a number with the membership function $\mu_T : \mathbf{R} \to [0,1]$ satisfying the following properties:

1. μ_T is a piecewise continuous function, and $supp(T) = \{x \in \mathbf{R}; \mu_A(x) > 0\}$ is bounded.
2. For any i, $supp(A_i) \subseteq supp(T)$.
3. T is non-empty, i.e. $\int_{-\infty}^{\infty} \mu_T(x)dx > 0$.

For the target number T, values of $E_T(A_i) = P(A_i \succeq T), i = 1,2,\ldots,N$ are first calculated and then normalized as:

$$R_T(A_i) = \frac{E_T(A_i)}{\max_{A_j \in S}\{E_T(A_j)\}} \qquad (7)$$

Recall that $R_T(A_i) \in [0,1]$, referred to as the relative index of the fuzzy number $A_i \in S$ with respect to the target T.

In [14], three different triangular target fuzzy numbers are introduced according to the objective of decision-maker; T_{pes}, T_{net}, and T_{opt} for pessimistic, neutral, and optimistic ones, respectively.

Let us define:

$$x_{\min} = \min\{x; \exists A_i \in S; \mu_{A_i}(x) > 0\},$$
$$x_{\max} = \max\{x; \exists A_i \in S; \mu_{A_i}(x) > 0\}.$$

In this way, the two fuzzy triangular numbers T_{pes} and T_{opt} are defined as:

$$T_{pes} = [x_{\min}, x_{\min}, x_{\max}],$$
$$T_{opt} = [x_{\min}, x_{\max}, x_{\max}].$$

Furthermore, T_{net} is defined as a fuzzy number with the membership function that attains 1 over the interval $[x_{\min}, x_{\max}]$ and 0 elsewhere.

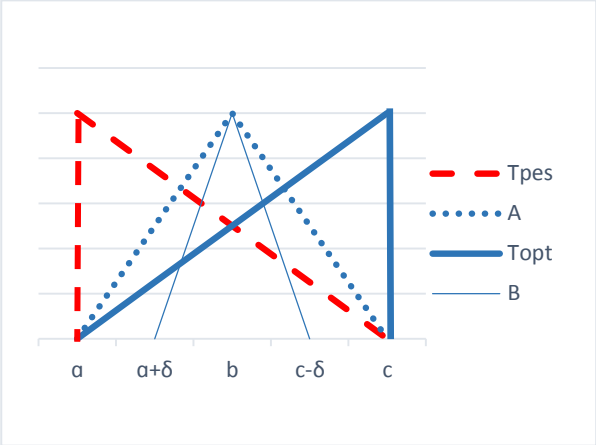

Figure 1. Two symmetric fuzzy numbers $A=[a,b,c]$ and $B=[a+\delta,b,c-\delta]$ and their associated fuzzy targets T_{pes} and T_{opt} are, respectively, depicted by dashed and solid lines.

An appropriate fuzzy target number for a problem is the one that assigns identical ranks to identical random labels, and different ranks to the others. A further helpful criterion for a suitable target fuzzy number is its behavior on comparing two symmetric fuzzy numbers. The result of rating two symmetric fuzzy numbers with identical mean should satisfy the sense of decision-makers. In some applications, the outer number, i.e. the more deviated one, may be preferred to have a higher rank. However, the inner one's rank is preferred to be larger in some other cases. Simply, it can be seen that if $[a,b,c]$ and $[a+\delta,b,c-\delta]$ are two triangular fuzzy numbers with $\delta>0$, then:

$$R_{T_{net}}[a,b,c]=R_{T_{net}}[a+\delta,b,c-\delta], \quad (8)$$

$$R_{T_{pes}}[a,b,c]<R_{T_{pes}}[a+\delta,b,c-\delta], \quad (9)$$

$$R_{T_{opt}}[a,b,c]>R_{T_{opt}}[a+\delta,b,c-\delta], \quad (10)$$

Thus, the target number can be selected respecting (8) -(10). As it can be seen in Figure 1, A and B are two symmetric fuzzy numbers with identical means, while A is more deviated. In light of (8), the algorithm does not distinguish A and B, if one uses T_{net} as the target number, no matter how large the value for δ is (See Figure 1). If the decision-maker does not care about the value for deviation, T_{net} could be an appropriate target number. On the other hand, if assigning a lower rank to random labels with lower minimum outcomes is preferred, T_{pes} might be considered as the target number (9), while T_{opt} seems significant when the decision-maker prefers to assign larger ranks to fuzzy numbers with a higher maximum outcome.

3.1.3. Step 4: Ranking random labels with not enough observations

Recall that random labels $l(.,b), b\in B_j^l$ are ranked in Step 4. As mentioned earlier, some random variables may have not enough observation of outcomes. For these cases, we approximate their rank by the average of the $R\big(fuz(l(.,b))\big), b\in B_j^e$, based on the existing similarity. If B_k is a set of single attributes, there is no explicit similarity between the members since they are only nominal attributes. Therefore, we simply approximate their ranks as the average of $R\big(fuz(l(.,b))\big), b\in B_j^e$.

$$R\big(fuz(l(.,b))\big):=\frac{\sum_{b\in B_j^e}R(b)}{\left|B_j^e\right|}. \quad (11)$$

When B_k is a set of multi-valued attributes, some members of B_k^l and B_k^e have some common codes. We will make use of this property. Consider the set of multi-valued attributes $B_k=B_k^l\bigcup B_k^e$, where

$$B_k^l=\{C_1^l,C_2^l,\ldots,C_{m_k}^l\}$$

$$B_k^e=\{C_1^e,C_2^e,\ldots,C_{m'_k}^e\},$$

and C_i^l and C_i^e are subsets of a set of all options for the k^{th} features. For $j=1,\ldots,m'_k$ and $i=1,\ldots,m_k$ define:

$$w(C_i^l,C_j^e):=\frac{2\times\left|C_i^l\cap C_j^e\right|}{\left|C_i^l\right|+\left|C_j^e\right|}.$$

The weighted mean is defined as follows:

$$S(C_i^l):=\frac{\sum_{j=1}^{m'_k}w(C_i^l,C_j^e)\times R\Big(fuz\big(l(.,C_j^e)\big)\Big)}{\sum_{j=1}^{m'_k}w(C_i^l,C_j^e)} \quad (12)$$

Let α_i represent the validity of the index $S(C_i^l)$, as the rank of the i^{th} attribute C_i^l. Let us define the value for α_i as follows:

$$\alpha_i:=\frac{\left|C_i^l\cap(\cup_{d\in B_k^e}d)\right|+\sum_{d\in B_k^e}\left|C_i^l\cap d\right|}{\sum_{d\in B_k^e}|d|+\left|C_i^l\right|}. \quad (13)$$

Observe that α_i can be considered as a generalization of $w(C_i^l, C_j^e)$, i.e. $w(C_i^l, C_j^e)$ is the value for α_i when $B_k^e = \{C_j^e\}$.

Now the rank of $R(l(.,C_i^l))$ is determined as follows:

$$R(l(.,C_i^l)) = \alpha_i S(C_i^l) + (1-\alpha_i)M, \qquad (14)$$

in which M is the average of the objects' ranks in B_j^e. A larger value for α_i may refer to more reliability of $S(C_i)$. When C_i^l has no intersection with all members of B_k^e, we have $\alpha_i = 0$ and $R\left(l(.,C_i^l)\right) = M$.

3.2. Phase 2: Clustering objects

In this phase, we are looking for an optimal solution of the following problem that is a transformation of (2).

$$\begin{aligned} \min & \sum_{i=1}^{m}\sum_{j=1}^{k} w_{ij} \left\| R(a^i) - \overline{R_j} \right\|^2 \\ s.t. & \sum_{j=1}^{k} w_{ij} = 1 \quad i = 1,\ldots,m \\ & \sum_{i=1}^{m} w_{ij} \geq \gamma_0 \quad j = 1,\ldots,k \\ & w_{ij} \in \{0,1\}, \quad i = 1,\ldots,m \quad j = 1,\ldots,k \end{aligned} \qquad (15)$$

where, k is the number of clusters, $\overline{R_j}$ is the center of the j^{th} cluster, and w_{ij} are decision variables. To do this end, an adjusted K-means algorithm is proposed.

3.2.1 Adjusted K-means algorithm

Problem (15), without satisfying a size constraint on clusters, is a well-known NP-hard problem [24]. When the number of clusters is known, it can be solved in time $O(m^{nk+1}\log m)$, [25], where m is the number of objects, k is the number of clusters, and n is the dimension of objects. In such a case, several heuristic methods have been proposed to find a local minimum solution. The standard K-means algorithm can be summarized as follows: First, the initial k centers are selected. Each point in the dataset is then assigned to the closest center, and a cluster is identified as the points with identical centers. In each step, the centers are updated with respect to the elements of clusters. This process is iterated until no point changes the cluster center. This algorithm is usually fast, and therefore, running it for several times with different numbers of clusters to find the best one is common [15, 16]. Our adjusted K-means algorithm to solve problem (15) is based on this property. It should be mentioned that this method strongly depends on the selection of the initial centers. Different initial centers may lead to different solutions. Therefore, we execute this method 1000 times, each time with different initial centers, and select the best solution obtained among these runs.

To find the best number of clusters, several indices are defined [15, 16]; while there are two main ideas behind them, having clusters with lower intra-dispersion or clusters with a higher differentiate between them. Most of these indices apply the Euclidean norm between the objects. We use the coefficient of variation, which is a better index to evaluate dispersion.

Recall that in our approach, in addition to the constraint on the size of clusters, the number of clusters is also unknown. The following adjusted algorithm is proposed to approximate k^* as the best solution for the number of clusters and β_{k^*} as the clustering solution.

1. Set the number of clusters $k = 2$ and $T = 0.$
2. Apply the K-means algorithm for 1000 times, leading to $E_i^k \quad i = 1,\ldots,1000$ as their solutions. Let E be the set of such solutions among these 1000 that satisfy the cluster constraint size. If $E = \varnothing$ go to Step 5.
3. Let $\beta_k = \arg\min_{E_i^k \in E} \dfrac{IDI(E_i^k)}{ODI(E_i^k)}$ and $T = 1$.
4. $k = k + 1,$ and go to Step 2.
5. If $T = 1$, let:
   $k^* = \arg\min_{2,\ldots,k-1} \dfrac{IDI(\beta_j)}{ODI(\beta_j)}$,
   otherwise there is no clustering solution with clusters satisfying the requisite mass constraint.

In Step 3, $IDI(E_i^k)$ denotes the intra-dispersion index of clusters, defined as the sum of coefficient of variation of cluster's members, and $ODI(E_i^k)$ is the sum of coefficient of variation of centers. When the underlying data includes multiple vectors, the sum of the coefficient of variations over all dimensions are considered. The best

number of clusters is the value that minimizes the ratio of these two indices, which is a number between 2 and $\left\lfloor \frac{m}{\gamma_0} \right\rfloor$, where m is the number of objects, and γ_0 is the minimum number of elements in each cluster. Thus, the algorithm terminates after finite iterations. The algorithm may obtain no solution; for example, $\gamma_0 > m$. Such conditions are controlled by the variable T.

4. Experimental results

Here, the proposed method is applied to a real data of the cargo insurance policies.

4.1. Describing real world data

For nine years, the number of written cargo insurance policies has been 354820. For each of them, there are 6 features including different kinds of parceling, transportation means, special conditions, origin, destination, and commodity name, which, respectively, have 190, 1100, 2121, 177, 287, and 1063 different nominal values. In details, the first three variables are constructed from 65, 21, and 36 different possible values, accordingly. Therefore, the number of related attributes is the number of these multi-valued attributes. Table 1 shows two examples of such registered policies.

Special conditions are added to some policy contracts. They are texts, each of which is replaced by a code for summarizing the information. Analogously, for an easier programming, the codes are associated with the attributes. As a result, for problem (2) in our case, $m = 354820, n = 6$, and random labels stand for the cost of claims. To remove the effect of inflation rates and make the costs comparable, the ratio of cost of claims over insured capital is considered as the label instead. Hereafter, we refer to this value as the claim ratio.

4.2. Application of proposed algorithm

Let us explain executing the proposed algorithm on this real-case data step by step. Here, γ_0 in (2) is fixed to 5000.

4.2.1. Phase 1

This phase includes 4 steps. In the first step $B_j \quad j = 1, \ldots, 6$ are partitioned into B_j^e and B_j^l. The threshold value is set to 200. It means that for an accurate conclusion on the claims rate, at least 200 policies from each attribute must be observed. This value has been determined empirically. Only 304, 69, and 70 codes from total codes for commodity, origin, and destination have more than 200 records, and the triangular fuzzy numbers are constructed just for the related random claim values of these simple attributes. Further, there are 76, 29, and 204 multi-valued attributes of parceling; transportation means and special conditions have enough observations.

Table 1. Two different sample policies and their features.

Feature	Attributes	
	Policy1	Policy2
Origin	North Korea	Dubai
Destination	Tehran	Tehran
Commodity	Polyethylene	Diesel Generator
Vehicle	Cargo ship	Cargo ship, Lorry
Parceling	Pallet, Container, bag	Container
Conditions	5, 8, 9, 24	5, 9, 11, 20

In Step 2, fuzzy numbers related to random labels for members of $B_j^e \quad j = 1, \ldots, 6$ are constructed from the outcomes of random labels. For all members of B_j^e, the claim ratios are calculated annually; these ratios are presented by r_{iy} for $i = 1, 2, \ldots, n_{B_j^e}$, and $y = 1, 2, \ldots, 9$. These indices help us to remove the effect of inflation or any other price variation over time. As a result, a single rate is calculated for each code in each year. By dividing these values by the number of policies having i^{th} attribute in the y^{th} year, we have per-capita rate of claims in each year, shown by γ_{iy}.

For each code, $i = 1, 2, \ldots, n_{B_j^e}$, the values for minimum, mean, and maximum are evaluated.

$$\begin{aligned} \min_i &= \min_{y=1,\ldots,9} \{\gamma_{iy}\}, \\ mean_i &= mean_{y=1,\ldots,9} \{\gamma_{iy}\}, \\ \max_i &= \max_{y=1,\ldots,9} \{\gamma_{iy}\}. \end{aligned} \qquad (16)$$

We restate that using other kinds of fuzzy numbers may lead to a higher accuracy but with costly computation in the ranking phase. Here, the triangular fuzzy numbers are chosen to simplify the calculation in this study. Considering (16), the i^{th} triangular fuzzy number is $[\min_i, mean_i, \max_i]$.

These fuzzy numbers are ranked using the method explained in Section 3.1.2. Nature of the motivating problem, which deals with claim rates, encourages us to be more conservative assuming

T_{opt} as the target fuzzy number. In table 2, some of the existing multi-valued attributes for parceling, their related fuzzy numbers, and associated ranks are mentioned.

Table 2. Ranks of fuzzy numbers for some multi-valued attributes for parceling.

Multi-valued attributes	Fuzzy number	Rank
{Carton}	[0, 0.000025, 0.000059]	0.000850
{Pallet}	[5.16-08, 0.0001. 0.000298]	0.003859
{Carton, Container}	[0, 7.91-06, 0.000025]	0.000310
{Pallet, Container}	[3.85659e-06, 0.00003,7.54141e-05]	0.001150
{Carton, Bag}	[0, 2.244 e-05, 0.00020]	0.000326
{Pallet, Bag}	[0, 0.00018, 0.00047]	0.002596

The results of this ranking coincide with the expectations. A lower rank denotes a higher safety in parceling. As it can be seen in table 2, {Carton} stands lower in the rank than {Pallet}. This means that Carton is safer than Pallet. Further, a joint use of different kinds of parceling reasonably saves the order of safety. For instance, when pallet and carton are mixed with container, {carton, container} is safer. Analogously, expectation shows more safety for {Carton, Bag} than for {Pallet, Bag}.

Another intuition observed in this table is the ranks of {Carton, Bag} and {Carton, Container}, meaning that the Container provided a safer parceling than the Bag. An analogous conclusion is also valid in comparing {Pallet, Bag} and {Pallet, Container}. Another intuition observed in this table is the rank of {Carton, Bag} and {Carton, Container}, meaning that the Container provided a safer parceling than the Bag. An analogous conclusion is also valid in comparing {Pallet, Bag} and {Pallet, Container}. Another accordance with the rational expectation is that such attributes including more kinds of parceling are safer. Similar intuitions are observed in other multi-valued attributes.

The final task in this phase is the ranking of $b \in B_j^l$. In our database, three simple attributes are origin, destination, and commodity, in addition to three multi-valued attributes; parceling, transportation means, and special conditions. The rank of single attributes in members of B_1^l, B_2^l, B_3^l (the first three) are approximated by simple average of B_j^e members' rank using (11).

Table 3. Attributes having common codes with $C_i^l = \{21, 25\}$.

C_j^e	$w(C_i^l, C_j^e)$	$\lvert C_i^l \cap C_j^e \rvert$	C_j^e
{25}	0.667	1	1
{21}	0.667	1	1
{13,25}	0.5	1	2
{13,21}	0.5	1	2
{3,25}	0.5	1	2
{2,21}	0.5	1	2
{3,13,21}	0.4	1	3
{1,25}	0.5	1	2
{3,21}	0.5	1	2
{4,13,21}	0.4	1	3
{2,13,21}	0.4	1	3
{1,21}	0.5	1	2

For the ones with multi-valued attributes, denoted as members of B_4^l, B_5^l, B_6^l, the weighted average seems more accurate. A sample calculation of such weighted average is summarized as what follows. Consider the multi-valued attribute $C_i^l = \{21, 25\}$ of a parceling feature. Members of sets are representatives for different kinds of parceling, used instead of texts. In table 3, different attributes having non-empty intersection with C_i^l and their related weights, $w(C_i^l, C_j^e)$, (defined in Section 3.1.3) are calculated. Using these weights, the weighted average $S(C_i^l)$ is computed as defined in (12). Then α_i in definition (13), which estimates the validation of $S(C_i^l)$ as the rank of C_i^l, is obtained. The value for M is the simple average of $R(b)$ for all $b \in B_j^e$. At last, the rank of C_i^l is approximated by replacing these values in (14). The calculation leads to:

$$\alpha_i = \frac{12+2}{25+2} = 0.51852 \quad S(C_i^l) = 0.1119$$

$$M = 0.014896 \quad R(C_i^l) = 0.0652$$

4.2.1. Phase 2

In this phase, for clustering the objects having ordered features, a local solution of problem (15) is obtained. Figure 2 depicts the results of the K-means algorithm. In these graphs, the horizontal axes show the number of clusters. The vertical axis in the first graph stands for the number of clustering solutions having insurable clusters among 1000 different solutions.

As it is seen, all clustering solutions have insurable groups when the number of clusters is 2, while there is no solution with insurable groups with 12 clusters. Henceforth, the algorithm of clustering is terminated for $k=12$.

The optimum value of β_i for $2,\ldots,12$ is obtained where $k=7$. The second graph denotes the minimum value of intra-dispersion index among 1000 solutions for different numbers of clusters, and the last one denotes the deviation between centers.

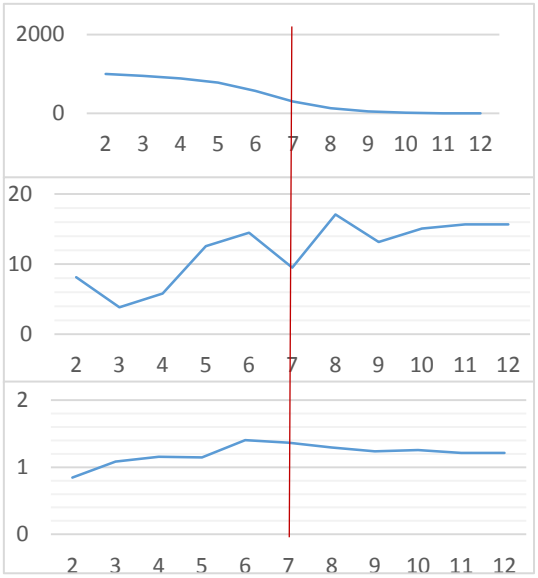

Figure 2. First graph, from above, illustrates number of clustering solutions having insurable groups among 1000 solutions for different number of clusters. Second one shows minimum value of intra-dispersion index over 1000 solutions, and third one denotes dispersion of centers.

4.3. Validity of results

In table 4, the number of elements in each cluster is reported. As it can be seen, the second and third clusters are so massive in comparison with the others. For keeping consistency with other clusters, these clusters are considered as self-stand databases, and the operation is carried out on them once again. This process is named as re-clustering in table 4.

Re-clustering of data in the third cluster determines 2 as the best number of sub-clusters; however, having the ability of comparison, the number of sub-clusters are fixed at 3. Even by this consideration, the third sub-cluster is still highly populated, while one more time re-clustering leads to no insurable groups. This phenomenon may refer to the structure of data, that some attributes are very popular. For instance, about one third of data have the same transportation mean; {Lorry} and {Cargo Ship}.

Table 4. Number of members of obtained cluster.

No.	First clustering	Re-clustering
1	6034	
2	76913	42629
		34284
3	175007	23149
		9690
		142168
4	5902	
5	14890	
6	21607	
7	54467	

Table 5. Distance of centers from origin.

No.	$\|C_i\|$	No.	$\|C_i\|$
1	1.001073		
2	0.019112	2.1	0.027325
		2.2	0.009190
3	1.00017	3.1	1.008535
		3.2	1.000058
		3.3	1.000017
4	1.064107		
5	1.414227		
6	0.130789		
7	0.061332		

The center of a cluster is considered as its representative. Being a centroid in far distance from the origin may be a sign of higher expected claim of the associated cluster. In table 5, the Euclidean distance of centers from origin is presented. As it can be seen, Cluster 2 is the nearest one to the origin, and Cluster 5 is the farthest. Re-clustering leads to some close sub-clusters. Table 5 shows that by re-clustering cluster 3 to 2 sub-clusters, the optimal solution of the proposed algorithm is preferable.

Recall that nominal attributes are converted to the ordinal ones and then clustered. Analyzing the difference between the clusters obtained with respect to the nominal attributes is useful. In order to do this comparison, the Chi-square test is applied to identify how distribution of nominal attributes differs [26]. Therefore, $\binom{10}{2}$ pairwise comparisons are carried out between clusters for each attribute. Table 6 presents the results, where x_1 - x_6 , respectively, denotes the features origin, destination, parceling, special condition, commodity, and transportation mean.

Table 6. Chi-square test results for pair wise comparison of clusters.

pairs	x_1	x_2	x_3	x_4	x_5	x_6
1×2.1	9791	12353	48653	13427	21970	35038
2.1×2.2	15556	20188	23606	27743	24277	72466
2.2×3.1	6588	7360	11928	19150	57433	57433
3.1×3.2	3435	3096	27639	4035	32839	0
3.2×3.3	3796	5416	145687	7538	18643	0
3.3×4	14820	8101	21509	148519	38213	96434
4×5	12398	6819	20792	20687	15981	12103
5×6	23112	11999	36497	27726	34571	36497
6×7	20464	10405	13790	14658	76074	40359

The critical values for the Chi-squared distribution with respect to degrees of freedom are given in table 7 as well. Comparing the values in tables 6 and 7 shows that the Chi-squared values are high enough to conclude that most groups are significantly dissimilar with respect to all attributes. However, there are a few exceptions. For example, Clusters 3.1, 3.2, 3.3, and 5 are exactly similar with respect to the attribute x_6 . The transportation means for all of their members are the multi-valued attribute {Lorry, Cargo Ship}. Besides, some groups are totally dissimilar. For instance, there are no policies with identical transportation mean or commodity in Clusters 2.2 and 3.1. In addition, all the three pairwise comparisons of Clusters 3.1, 3.2, and 3.3 reveal that they are not significantly dissimilar with respect to the attribute x_4 .

Finally, the dissimilarity measure proposed in [27], which is designed to compare two populations with nominal attributes, is calculated. The dissimilarity measure for two clusters, say C and C', is defined as:

$$G_{C,C'} = \sum_{i=1}^{r} \sum_{k=1}^{n_i} \frac{(C_{i,k} - C'_{i,k})^2}{P_{i,k}},$$

where, r is the number of features (here 6), n_i is the number of attributes for the i^{th} feature, and $C_{i,k}$ and $C'_{i,k}$ are proportions in the k^{th} attribute for the i^{th} feature in Clusters C and C', respectively, and $P_{i,k} = 1/2(C_{i,k} + C'_{i,k})$.

Table 7. Critical values for Chi-squared with respect to degree of freedom.

variable	critical value (0.99)	critical value (0.95)	Degree of freedom
x_1	1054	950	880
x_2	1557	1519	1430
x_3	5640	5567	5395
x_4	10876	10775	10535
x_5	5553	548	5310
x_6	1049	1018	945

Table 8. Dissimilarity measure between any two clusters.

No.	2.1	2.2	3.1	3.2	3.3	4	5	6	7
1	12	11	16	15	14	13	13	9	12
2.1		10	14	14	11	11	18	8	11
2.2			11	11	8	11	16	9	11
3.1				9	5	12	14	9	16
3.2					6	13	14	15	15
3.3						11	12	13	13
4							17	10	11
5								19	19
6									9

Comparing the values reported in table 8 says that pairs of Clusters 6, 5 and Clusters 2.1, 5 have the highest dissimilarity measure, while the lowest dissimilarity relates to pairs of Cluster 3.1, 3.2, and Clusters 3.2, 3.3.

As a conclusion, it can be said that the groups obtained are significantly dissimilar. However, it may be better to merge clusters 3.1, 3.2, and 3.3 since centers of these clusters are positioned very close to each other according to table 5, and additionally, they are not significantly dissimilar in some features. This consistency might be an evidence that our proposed algorithm satisfyingly specifies the number of clusters.

5. Conclusion

In this work, we modeled a real-world problem with a non-linear, non-convex optimization model, in which the objective function includes distances of random variables. For solving the problem obtained, an algorithm was proposed, which included two phases, ranking random labels, and clustering.

To define the distance of random labels, fuzzy numbers were used, which enabled us to consider the value of random variables and their chance of occurrence into ranking process simultaneously.

To produce homogenous clusters satisfying a requisite mass, an adjusted K-means algorithm was applied. In this algorithm, to determine the best number of clusters, an index applying

coefficient of variation of the objects was defined. The optimum number of clusters is chosen so that the clusters obtained have members with lower intra-dispersing and higher differentiate with the members of other clusters; simultaneously, the constraint of the requisite mass for each cluster is satisfied.

The proposed algorithm was applied on real data of nine-year cargo insurance policies including more than 354000 objects. The results obtained show that the seven obtained clusters are significantly different and homogenous, while they satisfy cluster constraint size.

Here, triangular fuzzy numbers were used to rank the random labels. More accurate fuzzy numbers may produce more reliable results, though the complexity of computations may increase. The rank of not enough observed attributes, and weighted average of the rank for all enough frequency ones were estimated. By selecting a proper collection of enough frequency attributes instead of all of them in Section 3.1.3, the accuracy of the ranks, and henceforth, the final results can be improved, which could be a further research direction. The proposed method can be considered as a main approach in the future works by testing some other fuzzy ranking methods, and other similarity indices.

Acknowledgment

This paper is financially supported by Azarbaijan Shahid Madani University. We are particularly grateful for the assistances given by Masoud Manteqipour, and we thank the review referees of the paper for their bright feedbacks.

References

[1] Yeo, A. C., Smith, K. A., Willis R. J., Brooks, M. (2001). Modeling the effect of premium changes on motor insurance customer retention rates using neural networks. Computational Science-ICCS 2001, pp. 390-399.

[2] Yeo A. C., Smith K. A., Willis R. J, Brooks, M. (2003). A comparison of soft computing and traditional approaches for risk classification and claim cost prediction in the automobile insurance industry. In Soft computing in measurement and information acquisition, Springer Berlin Heidelberg, pp. 249-261.

[3] Alpaydin, E. (2010). Introduction to Machine Learning. Cambridge, Mass.: MIT Press.

[4] Shmueli, G., Patel, N. R., and Bruce, P. C. (2010). Data mining for business intelligence. Hoboken. NJ: John Wiley & Sons, Inc.

[5] Huang, Z. (1997). A fast clustering algorithm to cluster very large categorical data sets in data mining, In: DMKD.

[6] Huang, Z. (1998). Extensions to the K-means algorithm for clustering large data sets with categorical values. Data mining and knowledge discovery, vol. 2, no. 3, pp. 283-304.

[7] Huang, Z. (1997). Clustering large data sets with mixed numeric and categorical values, In: Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining, (PAKDD), pp. 21-34.

[8] Samson, D., Thomas, H. (1987). Linear models as aids in insurance decision making: the estimation of automobile insurance claims. Journal of Business Research, vol. 15, no. 3, pp. 247-256.

[9] Dalvi, N., Suciu, D. (2007). Efficient query evaluation on probabilistic databases. The VLDB Journal-The International Journal on Very Large Data Bases, vol.16, no. 4, pp. 523-544.

[10] Li, J., Deshpande, A. (2010). Ranking continuous probabilistic data sets, Proceedings of the VLDB Endowment, vol. 3, no. 1-2, pp. 638-649.

[11] Li, J., Saha, B., Deshpande, A. (2009). A Unified Approach to Ranking in Probabilistic Databases. Proceedings of the VLDB Endowment, vol. 2, no. 1, pp. 502-513.

[12] Martos Venturini, G. A. (2015). Statistical distances and probability metrics for multivariate data, ensembles and probability distributions.

[13] Lotfi, A. Z., (1965). Fuzzy sets. Information and control, vol. 8, no.3, pp. 338-353.

[14] Huynh, V. N., Nakamori, Y., Lawry, J. (2008). A probability-based approach to comparison of fuzzy numbers and applications to target-oriented decision making. IEEE Transactions on Fuzzy Systems, vol. 16, no. 2, pp. 371-387.

[15] Murat Y.S. (2012). Fuzzy Clustering Approach for Accident Black Spot Centers Determination, INTECH Publishing.

[16] Murat, Y.Ş., Şekerler A. (2009), Modelling Traffic Accident data by Clustering Approaches. Technical Journal of Turkish Chamber of Civil Engineers, Vol 20, no. 3, pp. 4759-4777.

[17] Wu, J. (2012). Advances in K-means clustering: a data mining thinking. Springer Science & Business Media.

[18] Wu, X., Kumar,V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Philip, S. Y. and others. (2008). Top 10 algorithms in data mining, Knowledge and Information Systems. vol. 14, no. 1, pp. 1-37.

[19] Ganganath, N., Cheng, C. T., Chi, K. T. (2014). Data clustering with cluster size constraints using a modified k-means algorithm, In: Cyber-Enabled Distributed Computing and Knowledge Discovery *(CyberC)* 2014 International Conference on, IEEE, pp. 158-161.

[20] Bagirov, A. M. (2010). Modified global k-means algorithm for minimum sum-of-squares clustering problems. Pattern Recognition, vol. 41, no. 10, pp. 3192-3199.

[21] Bock, H. H. (1998). Clustering and neural networks. In: Advances in data science and classification. Springer-Verlag. Berlin. pp. 265-277.

[22] Spath, H. (1980). Cluster analysis algorithms for data reduction and classification of objects. Ellis Horwood Chichester.

[23] Safaee, B., Kamaleddin Mousavi Mashhadi, S. (2017). Optimization of fuzzy membership functions via PSO and GA with application to quad rotor. Journal of AI and Data Mining, vol. 5, no. 1, pp. 1-10.

[24] Mahajan, M., Nimbhorkar, P., Varadarajan, K. (2009). The Planar k-Means Problem is NP-Hard. In: International Workshop on Algorithms and computation, Springer, pp. 274–285.

[25] Inaba, M., Katoh, N., Imai, H. (1994). Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. In: Proceedings of 10th ACM Symposium on Computational Geometry, pp. 332-339.

[26] McDonald, J. H. (2009). Hand book of biological statistics. Sparky House Publishing, vol. 2. pp. 6-59.

[27] Balakrishnan, V., Sanghvi, L. D. (1968). Distance between populations on the basis of attribute data. Biometrics. pp. 859-865.

نشریه هوش مصنوعی و داده کاوی

گروه‌بندی اشیاء به کلاس‌های همگن با حداقل تعداد اعضا

مهناز منطقی پور*[۱]، علیرضا غفاری حدیقه[۱]، رحیم محمودوند[۲] و امیر صفری[۳]

[۱] دانشکده علوم پایه، گروه ریاضی کاربردی، دانشگاه شهید مدنی آذربایجان، تبریز ،ایران.

[۲] دانشکده علوم پایه، گروه آمار، همدان، ایران.

[۳] پژوهشکده بیمه وابسته به بیمه مرکزی ج.ا.ا، تهران، ایران.

ارسال ۲۰۱۶/۰۹/۰۱؛ بازنگری ۲۰۱۷/۰۱/۲۵؛ پذیرش ۲۰۱۷/۰۳/۱۵

چکیده:

گروه‌بندی داده‌ها نقش بسیار مهمی در تحقیقات علمی پیاده می‌کنند. بر اساس نوع داده‌ها و کاربرد موضوع قیود متفاوتی برای گروه‌ها در نظر گرفته می‌شود، در حالی‌که داشتن گروه‌هایی شامل اعضای مشابه یکی از مهمترین معیارها می‌باشد. در این مقاله ما الگوریتمی برای گروه‌بندی اشیاء با برچسب‌های تصادفی و خصوصیات اسمی ارائه می‌نمائیم. همچنین برای هر گروه، قید حداقل تعداد عضو در نظر گرفته شده‌است. مدلسازی این مسئله منجر به یک مسئله بهینه‌سازی عدد صحیح مختلط شده است که نه خطی است و نه محدب. یافتن جوابهای دقیق این مسئله از نظر محاسباتی بسیار پر هزینه است. انگیزه ما از حل این مساله، گروه‌بندی ریسکهای بیمه باربری بوده است که برای تعیین نرخ منصفانه مفید است. الگوریتم پیشنهادی شامل دو فاز است. اول رتبه‌بندی برچسب‌های تصادفی با استفاده از اعداد فازی و پس از آن استفاده از الگوریتم اصلاح شده کا-میانگین‌ها برای ساختن گروه‌های همگن و دارای حداقل تعداد عضو. برای مقایسه مقادیر و احتمال رخداد برچسب‌های تصادفی از اعداد فازی استفاده کرده‌ایم. علاوه بر آن به منظور یافتن رتبه خصوصیات اسمی که به اندازه کافی مشاهده نشده‌اند شاخص شباهت آنها به سایر خصوصیات را تعریف کرده‌ایم. در الگوریتم کا-میانگین‌های معرفی شده، تعداد بهینه خوشه‌ها با استفاده از ضریب تغییرات داده‌ها به دست آمده است. به‌کارگیری الگوریتم پیشنهادی بر داده‌های واقعی منجر به تولید خوشه‌هایی همگن، به طور معنادار متفاوت و شامل حداقل اعضا شده است.

کلمات کلیدی: اعداد فازی، الگوریتم کا-میانگین‌ها، خوشه‌بندی، کلاس‌بندی، گروه‌های همگن.