Probabilistic and More General Uncertainty-Based (e.g., Fuzzy) Approaches to Crisp Clustering Explain the Empirical Success of the K-Sets Algorithm

Vladik Kreinovich, Olga Kosheleva, Shahnaz Shabazova, and
Songsak Sriboonchitta

Abstract Recently, a new empirically successful algorithm was proposed for crisp clustering: the K-sets algorithm. In this paper, we show that a natural uncertainty-based formalization of what is clustering automatically leads to the mathematical ideas and definitions behind this algorithm. Thus, we provide an explanation for this algorithm's empirical success.

1 Clustering by Similarity: Formulation of the Practical Problem

Clustering is important. In many practical situations, there are so many different objects that it is not possible to come up with an individual approach to each of these objects. In such situations, a reasonable idea is to divide all the objects into a small number of groups (clusters), so that we will be able to develop a reasonable strategy of dealing with each group.

For example, it is not (yet) practically possible to come up with a completely individualized medicine, so a natural idea is:

- to divide all the patients into groups corresponding to different diseases, and then
- to develop treatments for each of these diseases.

Vladik Kreinovich and Olga Kosheleva
University of Texas at El Paso, 500 W. University, El Paso, TX 79968, USA
e-mail: vladik@utep.edu, olgak@utep.edu

Shahnaz Shabazova
Azerbaijan Technical University, Baku, Azerbaijan
e-mail: shahbazova@gmail.com

Songsak Sriboonchitta
Faculty of Economics, Chiang Mai University, Chiang Mai, Thailand
e-mail: songsakecon@gmail.com

Scientific clustering: by origin. In science, it is often desirable to bring together *related* objects, i.e., objects which are related by a small change from each other. For example, in astronomy, there are different sequences of star's evolution, so we group together all the stars corresponding to the same sequence – even when they are at completely different parts of this sequence. Similarly, in biology, it is sometimes reasonable group together animals on different stages of their development.

For such situations, there are many well-defined clustering methods. In these methods, we usually mark sufficiently close objects as belonging to the same cluster, and then apply the transitive closure, i.e., consider objects a and b belonging to the same cluster if there is a sequence of objects $a = a_0, a_1, \ldots, a_{n-1}, a_n = b$ in which each object a_i is sufficiently close to the next one a_{i+1}.

In the scientific classification, objects from the same cluster are not necessarily similar to each other. In this approach, belonging to the same cluster does not necessarily mean similarity. For example, in this approach, each butterfly gets grouped together with its caterpillar, but, of course, there is much more similarity between different butterflies than between the butterfly and its caterpillar.

Practical clustering: by similarity. In most practical situations, it is desirable to classify objects by their similarity.

For example, for medical purposes, it makes much more sense to group together patients who have flu than to group them by diseases they had in childhood. Such clustering-by-similarity is the main topic of this paper.

Formulation of the problem. One of the main problems with clustering is that it is not clear what exactly we want. It is often difficult to come up with a reasonable objective measure of which clustering is better.

In some cases, we know the desired result: e.g., we have a DNA-based classification of plants, and we want to come up with clusters which are the closest to this classification. However, in general, this is not clear.

K-sets algorithm: a recent approach. Recently, a new approach to clustering has been proposed [2]. This approach leads to an empirically successful algorithm that the authors call *K-sets*.

Remaining problem and what we do in this paper. The K-sets approach optimizes some seemingly reasonable criterion, but it is not clear why specifically this criterion has been chosen – there are, in principle, many possible similar criteria, and it is not clear why namely this one has been empirically successful.

In this paper, we show that a natural fuzzy-based formalization of clustering can explain this criterion – and can, thus, explain the empirical success of the K-sets algorithm.

Comment. At this stage, we are only considering *crisp* clustering, when each object is assigned to a single cluster. In practice, it is often useful to have a probabilistic or fuzzy clustering, in which each object can be assigned to two or more different clusters, with different degrees of belonging. For example, in medical analysis, based on some patients' symptoms, these patients may be in between flu and cold;

in such cases, it is reasonable to consider such patients as somewhat belonging to two clusters at the same time.

It is desirable to extend our analysis to such clustering, but as of now, we only know how to apply this analysis to crisp clustering.

2 The Main Idea Behind Clustering-by-Similarity

The main idea of clustering by similarity is that once we divide the objects into clusters, then:

- objects within each cluster are similar to each other, while
- objects assigned to different clusters are not similar to each other.

3 Probabilistic Approach: Towards the Precise Formulation of the Problem

Probabilistic case: a brief description. In this section, we assume that for every two objects a and b, we know the probability $p(a,b)$ that these objects are similar. This may be a subjective probability evaluated by an expert, or this may be a probability estimated based on some objective criterion.

In this case, the probability that a and b are not similar is equal to $1-p(a,b)$.

From similarity between objects to quality of clustering. It is reasonable to assume that for different pairs, similarities are independent events.

This assumptions makes sense if we have no information about the relation between these events. In this case, we do not know the full probability distribution, we have many different probability distributions consistent with the given information. In such situations, a reasonable idea is to use the Maximum Entropy Approach (see, e.g., [3]) to select the most reasonable probability distribution. For situations in which we know the probabilities of individual events, but we do not know the correlation between these events, the Maximum Entropy approach selects a distribution in which these events are independent [3].

Under this independence assumption, for each clustering, i.e., for each subdivisions of the set Ω of all objects into K disjoint sets $S_1,\ldots,S_K$, the probability $P(S_1,\ldots,S_K)$ that this clustering is consistent with the available information about similarity is equal to the product of the corresponding probabilities:

$$P(S_1,\ldots,S_k)=\prod_{a\sim b} p(a,b)\cdot\prod_{a\not\sim b}(1-p(a,b)),$$

where $a\sim b$ means that the objects a an b belong to the same cluster:

$$a\sim b\Leftrightarrow\exists k\,(a\in S_k\,\&\,b\in S_k).$$

How do we select the best clustering. In the probabilistic case, a natural idea is to use the Maximum Likelihood approach (see, e.g., [7]), and select the clustering $S_1,\dots,S_K$ for which the probability $P(S_1,\dots,S_K)$ is the largest possible.

4 Probabilistic Approach: Precise Formulation of the Problem, Resulting Clustering Algorithm, and Their Relation to K-Sets Algorithm and Its Foundations

Precise formulation of the problem. For every two objects a and b, we know the probability $p(a,b)$ that these two objects are similar. We then need to find a clustering, i.e., a subdivision of the set Ω of all objects into disjoint subsets $S_1,\dots,S_k$ for which the product

$$P(S_1,\dots,S_K)=\prod_{a\sim b} p(a,b)\cdot\prod_{a\not\sim b}(1-p(a,b))$$

attains the largest possible value, where $a\sim b$ means that $\exists k\,(a\in S_k\,\&\,b\in S_k)$.

Let us simplify this optimization problem. The above formulation is mathematically precise, but from the computational viewpoint, it is not perfect.

First of all, to compute the above value, we need to consider all possible pairs (a,b). Often, there are many possible objects, so the number of pairs is huge. So, it is desirable to cut down on the number of pairs for which we perform the computations when gauging the quality of different clusterings.

This can be done, e.g., if we take into account that for every constant $C>0$, maximizing the objective function $F(x)$ is equivalent to maximizing the objective function $\dfrac{F(x)}{C}$. In our case, we can take $C=\prod\limits_{a,b}(1-p(a,b))$, where the product is taken over all possible pairs. If we divide the above objective function by this constant C, we get an equivalent objective function in which we only need to consider pairs belonging to the same cluster:

$$\prod_{a\sim b}\frac{p(a,b)}{1-p(a,b)}.$$

This product can be, in its turn, subdivided into products corresponding to each individual cluster S_k:

$$\prod_{k=1}^{K}\prod_{a,b\in S_k}\frac{p(a,b)}{1-p(a,b)}.$$

In practice, we may have hundreds and thousands of objects, and multiplying hundreds of numbers smaller than 1 lead us to close to 0 real fast, to values below the machine zero. This problem can be avoided if we take into account that logarithm

is a strictly increasing function, and the logarithm of the product is equal to sum of the logarithms. Thus, maximizing the above objective function is equivalent to maximizing the sum

$$\sum_{k=1}^{K} \sum_{a,b \in S_k} \gamma(a,b),$$

where we denoted

$$\gamma(a,b) \stackrel{\text{def}}{=} \ln\left(\frac{p(a,b)}{1-p(a,b)}\right) = \ln(p(a,b)) - \ln(1-p(a,b)).$$

Thus, we arrive at the following equivalent formulation:

- we are given a function $\gamma(a,b)$, and
- we need to find a clustering for which the above sum takes the largest possible value.

Relation to the K-sets approach. In the above formula, we count each un-ordered pair exactly once. Instead, we can consider all possible ordered pairs – meaning that we will count each un-ordered pair twice. This will simply increase the value of the above objective function by a factor or 2, without changing which clustering is better and which is worse. With this re-arrangement, the above formula takes the form

$$\sum_{k=1}^{K} \sum_{a \in S_k} \sum_{b \in S_k} \gamma(a,b),$$

i.e., the form

$$\sum_{k=1}^{K} \gamma(S_k, S_k),$$

where we denoted

$$\gamma(A,B) = \sum_{a \in A} \sum_{b \in B} \gamma(a,b).$$

This is exactly the *modularity index Q* which is used in [2] to find the best clustering – now we have an explanation for this formula.

Towards a natural algorithm for finding the best clustering. How can we find the best clustering? When the number of clusters is known, a natural idea is to iteratively optimize.

Namely, once we have some subdivision into clusters $S_1, \ldots, S_K$, we then, or each element a, decide to which of these clusters this element should belong so as to maximize the value of the objective function.

If the element a is assigned to the cluster S_k, then it contributes the sum $\sum_{b \in S_k} \gamma(a,b)$ to the objective function. Thus, we assign each the element to the cluster k for which this sum is the largest. In other words, we arrive at the following algorithm.

Resulting iterative algorithm. We start with some clustering $S_1, \ldots, S_K$. Then, for each element a and for each cluster k, we compute the sum $\sum_{b \in S_k} \gamma(a,b)$, and we assign the element a to the cluster k for which this sum is the largest.

Thus, we get a new clustering, so we repeat this procedure until the process converges.

Comment. Instead of computing the sum over all the elements $b \in S_k$, we can use the Monte-Carlo techniques and estimate this sum by selecting a few random points from the set S_k.

Relation to K-sets algorithm. What we described is exactly the iterative procedure in the K-sets algorithm. Thus, we have a natural explanation for this empirically successful algorithm.

Comment. So far, we have only considered the case when the number of clusters K is fixed. However, it is also possible to use the same criterion to select the optimal value K. For this purpose, we can check whether the value of the objective function increases if we merge two clusters – or, vice versa, split one of the clusters into two; see [2] for more details.

5 Towards A More General Uncertainty-Based Approach

Need to go beyond probabilities. In the above text, we assumed that for every two objects a and b, we know the probability $p(a,b)$ that these objects are similar. In particular, this can be subjective probability, describing the expert's degree of confidence that the two objects are similar.

In practice, however, the experts often cannot express their degrees of confidence in probabilistic terms. What we can do in such situations is, e.g., ask an expert to mark, on a scale from 0 to 10, his or her degree of confidence that the objects a and b are similar. If the expert marks 7, we can then assign the degree $s(a,b) = 7/10$. In general, if an expert selects m on a scale from 0 to n, we select $s(a,b) = m/n$.

There cab other ways to gauge the expert's degree of confidence that the objects a and b are similar. In all these cases, it is convenient to scale the corresponding degrees $s(a,b)$ to the interval $[0,1]$, so that:

- the value $s(a,b) = 1$ means absolutely similar,
- the value $s(a,b) = 0$ means not similar at all, and
- intermediate values $s(a,b) \in (0,1)$ correspond to some similarity.

How to describe degree of non-similarity in such a general case. If the two objects are similar with degree $s(a,b) < 1$, then, since perfect similarity corresponds to degree 1, it is reasonable to gauge the degree of non-similarity by the remaining value $1 - s(a,b)$.

From similarity between objects to quality of clustering: need for "and"-operations. Based on the degrees of similarity between different pairs of objects,

we would like to estimate how well a given clustering corresponds to the desired quality, i.e., to what extent objects within each cluster are similar to each other, and objects from different clusters are not similar to each other.

In other words, we are interested, for a given clustering $S_1 = \{a, b, \ldots\}$, $S_2 = \{c, d, \ldots\}$, in a degree to which a and b are similar *and* c and d are similar, *and* a and c are not similar, etc.

An ideal solution would be to do what we did when we looked for degrees $s(a,b)$ of similarity between objects: ask experts. However, even for a reasonably small class of objects (e.g., 30) and for just two possible clusters ($K = 2$), there are $2^{30} \approx 10^9$ possible subdivisions into two clusters, and there is no way to ask a billion questions to an expert.

Since we cannot ask the expert to gauge his/her degree of confidence in every possible "and"-combination of individual statements, we have to estimate this degree of confidence based on the degrees of confidence of individual statements. This is one of the main ideas behind *fuzzy logic* (see, e.g., [4, 6, 8]): we need an estimating algorithm $f_\&(a,b)$ that, given the expert's degree of confidence a and b in two statements A and B, provides an estimate $f_\&(a,b)$ for the expert's degree of confidence in the "and"-combination $A \,\&\, B$.

By using an appropriate "and"-operation, we can then estimate the degree to which a given clustering is good as

$$f_\&(s(a,b), s(c,d), \ldots, 1 - s(a,c), \ldots),$$

where:

- for objects a and b from the same cluster, we combine the degrees of similarity $s(a,b)$, while
- for objects a and c from different clusters, we combine the degrees of non-similarity $1 - s(a,c)$.

Which "and"-operations should we use in computing this degree? To answer this question, let us consider natural requirements on possible "and"-operations.

Natural properties of "and"-operations. What are the natural properties of an "and"-operation $f_\&(a,b)$?

First of all, $A \,\&\, B$ means the same as $B \,\&\, A$, so we should expect that applying our estimating function $f_\&$ to these two different expressions would lead to the same result: $f_\&(a,b) = f_\&(b,a)$. Thus, the "and"-operation $f_\&(a,b)$ must be commutative.

Similarly, since $A \,\&\, (B \,\&\, C)$ and $(A \,\&\, B) \,\&\, C$ mean the same, we expect that $f_\&(a, f_\&(b,c)) = f_\&(f_\&(a,b), c)$ for all a, b. and c. In other words, the "and"-operation should be associative.

If we increase our degree of confidence in A, this should increase our degree of confidence in $A \,\&\, B$ – so the "and"-operation should be increasing in both variables.

In particular, small changes in degree of confidence in A should lead to small changes in degree of confidence in $A \,\&\, B$ – so the "and"-operation must be continuous.

Finally, if A is absolutely true (i.e., if our degree of confidence in the statement A is equal to 1), then the only reason why we may be not fully confident in $A\,\&\,B$ is because we are not fully confident in B. In other words, in this case, our degree of confidence in $A\,\&\,B$ should be equal to the degree of confidence in B: $f_{\&}(1,b) = b$.

There is a known classification of all "and"-operations that satisfy all these properties. "And"-operations satisfying all these properties are known as *t-norms*. There exists a full classification of all possible t-norms. In particular, it is known [5] that, for any given accuracy $\varepsilon > 0$, any t-norm can be ε-approximated by an *Archimedean t-norm*, i.e., by a t-norm of the type $f_{\&}(a,b) = g^{-1}(g(a)\cdot g(b))$ for some strictly increasing continuous function $g(a)$; here, as usual, $g^{-1}(x)$ denotes the inverse function.

Thus, without losing generality, we can safely assume that our "and"-operation is Archimedean.

Which Archimedean t-norm should we choose? Different functions $g(x)$ describe, in general, different "and"-operations. Which one should we choose for our comparison of different clusterings?

To have the most adequate comparison, out of all possible "and"-operations, we must select the one which best describes the reasoning of the corresponding experts. For this purpose, we need to compare the expert's degrees of confidence in different "and"-combinations $A\,\&\,B$ with the estimates $f_{\&}(a,b)$ predicted by different t-norms, and select the t-norm which is the best fit for a given expert. Such t-norm-fitting has a long history: it was first done by researchers who designed the world's first successful expert system MYCIN; see, e.g., [1].

The resulting formula for the quality of a given clustering. For an Archimedean "and"-operation, the above formula for the degree of quality of clustering takes the following form:

$$g^{-1}\left(\prod_{a\sim b} g(s(a,b))\cdot \prod_{a\not\sim b} g(1-s(a,b))\right),$$

where $a\sim b$ means that the objects a and b belong to the same cluster.

Let us simplify this formula. Since the function $g(x)$ is strictly increasing and continuous, its inverse $g^{-1}(x)$ is also strictly increasing, i.e., $g^{-1}(x) < g^{-1}(y)$ if and only if $x < y$.

Our goal is to compare different clusterings. Thus, instead of the above degrees x, we can consider the values $g(x)$: for every two degrees x and y, the comparison between x and y will lead to the same result as the comparison between $g(x)$ and $g(y)$. The advantage of using $g(x)$ is that, since the original degrees x have the form $x = g^{-1}(E)$ for some expression E, we thus have $g(x) = g(g^{-1}(E)) = E$, i.e., a simpler expression that the original degree $g^{-1}(E)$. Thus, instead of comparing the original degrees, we can now compare simpler expressions

$$\prod_{a\sim b} g(s(a,b))\cdot \prod_{a\not\sim b} g(1-s(a,b)).$$

So, we arrive at the following precise formulation of the problem.

Case of general uncertainty: precise formulation of the problem. For every two objects a and b, we know the degree of similarity $s(a,b)$ between these objects. We need to also determine, by comparing the expert's opinions on individual statements and on their "and"-combinations, the t-norm (and thus, the corresponding function $g(x)$) that most adequately describes the expert's reasoning. Then, we need to find a clustering, i.e., a subdivision of the set Ω of all objects into disjoint subsets $S_1, \ldots, S_k$ for which the product

$$\prod_{a \sim b} g(s(a,b)) \cdot \prod_{a \not\sim b} g(1 - s(a,b))$$

attains the largest possible value, where $a \sim b$ means that $\exists k\,(a \in S_k \,\&\, b \in S_k)$.

Let us simplify this expression. Similarly to the probabilistic case, we can simplify this optimization problem if we take into account that for every constant $C > 0$, maximizing the objective function $F(x)$ is equivalent ti maximizing the objective function $\dfrac{F(x)}{C}$. In our case, we can take $C = \prod\limits_{a,b} g(1 - s(a,b))$, where the product is taken over all possible pairs. If we divide the above objective function by this constant C, we get an equivalent objective function in which we only need to consider pairs belonging to the same cluster:

$$\prod_{a \sim b} \frac{g(s(a,b))}{g(1 - s(a,b))}.$$

This product can be, in its turn, subdivided into products corresponding to each individual cluster S_k:

$$\prod_{k=1}^{K} \prod_{a,b \in S_k} \frac{g(s(a,b))}{g(1 - s(a,b))}.$$

Again, similarly to the probabilistic case, it is computationally beneficial to optimize the logarithm of this expression, i.e., a function

$$\sum_{k=1}^{K} \sum_{a,b \in S_k} \gamma(a,b),$$

where we denoted

$$\gamma(a,b) \stackrel{\text{def}}{=} \ln\left(\frac{g(s(a,b))}{g(1 - s(a,b))}\right) = \ln(g(s(a,b))) - \ln(g(1 - s(a,b))).$$

Thus, we arrive at the following equivalent formulation:

- we are given a function $\gamma(a,b)$, and
- we need to find a clustering for which the above sum takes the largest possible value.

Algorithms and their relation to the K-sets approach. Similarly to the probabilistic case, if we consider all possible ordered pairs – meaning that we count each un-ordered pair twice – then we get the equivalent objective function

$$\sum_{k=1}^{K} \sum_{a \in S_k} \sum_{b \in S_k} \gamma(a,b),$$

i.e., the function

$$\sum_{k=1}^{K} \gamma(S_k, S_k),$$

where we denoted

$$\gamma(A,B) = \sum_{a \in A} \sum_{b \in B} \gamma(a,b).$$

This is exactly the *modularity index Q* which is used in [2] to find the best clustering – now we have a general uncertainty-based explanation for this formula.

Thus, this approach – and the corresponding iterative clustering algorithm – have been justified not only for the probabilistic case, but also for the case of general (not necessarily probabilistic) uncertainty.

6 What If We Have Disproportionate Clusters?

Formulation of the problem. The above descriptions work well if all the clusters are of the same order of magnitude. However, sometimes, one (or more) of the clusters is much smaller than the others. In this case, in the above formulation, the probabilities (or, more generally, degrees of confidence) corresponding to a small cluster will "drown" in the preponderance of degrees corresponding to larger clusters. What shall we do in this case?

How to deal with this problem: an informal idea. A natural idea is that, instead of considering similarity between all possible pairs, we consider "average" similarity between clusters:

- on average, two objects picked from the same cluster should be similar, while
- on average, two objects picked from two different clusters, should not be similar.

Example. For example, if we classify dogs to one cluster and cats to a different cluster, then the above idea means that:

- two randomly selected dogs should be similar to each other,
- two randomly selected cats should be similar to each other, but
- a randomly selected dog should not be similar to a randomly selected cat.

How can we describe this idea in precise terms?

7 Disproportionate Clusters: Probabilistic Case

Formulation of the problem. For every two objects a and b from the same cluster S_k, we have the probability $p(a,b)$ that these objects are similar. How can we describe the "average" probability that the objects from this cluster are similar to each other?

First idea and its formalization. The probabilities of similarities between objects from the same cluster enter the objective function as a product term $\prod\limits_{a,b\in S_k} p(a,b)$. What does it mean to replace all these probabilities by an "average" one? A natural idea is to come with a single probability p_k so that when we use p_k instead of all the different probabilities $p(a,b)$, we get the exact same result. In other words, we should have $\prod\limits_{a,b\in S_k} p(a,b) = \prod\limits_{a,b\in S_k} p_k$, i.e., equivalently,

$$p_k^{P_k} = \prod_{a,b\in S_k} p(a,b),$$

where P_k is the number of pairs (a,b).

If we count each pair twice and denote the number of element in the k-th cluster by N_k, then we get N_k^2 pairs, and thus,

$$p_k = \left(\prod_{a\in S_k}\prod_{b\in S_k} p(a,b)\right)^{1/N_k^2}.$$

Similarly, for every two clusters S_k and S_ℓ, the terms corresponding to pairs $a\in S_k$ and $b\in S_\ell$ form the product $\prod\limits_{a\in S_k}\prod\limits_{b\in S_\ell}(1-p(a,b))$. So, it is reasonable to pick up an "average" probability $p_{k\ell}$ of non-similarity as the value for which

$$\prod_{a\in S_k}\prod_{b\in S_\ell}(1-p(a,b)) = \prod_{a\in S_k}\prod_{b\in S_\ell} p_{k\ell}.$$

This formula leads to

$$p_{k\ell} = \left(\prod_{a\in S_k}\prod_{b\in S_\ell}(1-p(a,b))\right)^{1/(N_k\cdot N_\ell)}.$$

In these terms, we want to optimize the overall probability that:

- within each cluster, the objects are, on average, similar, and
- for every two different clusters, the objects are non-similar.

Thus, we want to find the clustering that maximizes the following probability

$$\prod_{k=1}^{K} p_k \cdot \prod_{k<\ell} p_{k\ell} =$$

$$\prod_{k=1}^{K}\left(\prod_{a\in S_k}\prod_{b\in S_k}p(a,b)\right)^{1/N_k^2}\cdot\prod_{k<\ell}\left(\prod_{a\in S_k}\prod_{b\in S_\ell}(1-p(a,b))\right)^{1/(N_k\cdot N_\ell)}.$$

Let us simplify this expression. Just like before this objective function can be simplified if we consider its logarithm. Thus, the original optimization problem is equivalent to maximizing the expression

$$\sum_{k=1}^{K}\frac{1}{N_k^2}\cdot\sum_{a,b\in S_k}d^+(a,b)+\sum_{k<\ell}\frac{1}{N_k\cdot N_\ell}\cdot\sum_{a\in S_k}\sum_{b\in S_\ell}d^-(a,b),$$

where we denoted $d^+(a,b)\stackrel{\text{def}}{=}\ln(p(a,b))$ and $d^-(a,b)\stackrel{\text{def}}{=}\ln(1-p(a,b))$. We can describe it in the following equivalent form, as the need to maximize

$$\sum_{k=1}^{K}d^+(S_k,S_k)-\frac{1}{2}\cdot\sum_{k=1}^{K}\sum_{\ell\neq k}d^-(S_k,S_\ell),$$

where for every two sets S_k and S_ℓ and for every function $f(a,b)$, we denote

$$f(S_k,S_\ell)\stackrel{\text{def}}{=}\frac{1}{N_k\cdot N_\ell}\cdot\sum_{a\in S_k}\sum_{b\in S_\ell}f(a,b).$$

In particular, for two clusters, $S_1=S$ and $S_2=S^c\stackrel{\text{def}}{=}\Omega-S$, we need to minimize the expression

$$d^+(S,S)+d^+(S^c,S^c)+d^-(S,S^c).$$

Comment. It is worth noticing that this expression is similar to expressions from [2].

Iterative algorithm. Similarly to the above case, we can have an iterative algorithm for minimizing the above expression. We start with some subdivision into clusters, then we assign each object a to a cluster k for which the a's contribution to the distances should be the largest, i.e., for which the following expression is the largest possible:

$$\frac{1}{N_k}\cdot\left(d^+(a,S_k)+\sum_{\ell\neq k}d^-(a,S_\ell)\right),$$

where, for each function $f(a,b)$, we denote

$$f(a,S_k)\stackrel{\text{def}}{=}\frac{1}{N_k}\cdot\sum_{b\in S_k}f(a,b).$$

Once we assign all the elements to appropriate clusters, we get new clusters $S_1,\ldots,S_K$, so we can re-assign the elements, etc., until the process converges.

Second idea. The second idea comes from the fact that the original probability-based objective function

$$\prod_{a\sim b} p(a,b)\cdot\prod_{a\not\sim b}(1-p(a,b)).$$

Instead of using this expression, we can divide the objective function by the product $\prod_{a,b}\sqrt{p(a,b)\cdot(1-p(a,b))}$, where the product is over all possible pairs, and get an equivalent objective function

$$\prod_{a\sim b}\sqrt{\frac{p(a,b)}{1-p(a,b)}}\cdot\prod_{a\not\sim b}\sqrt{\frac{1-p(a,b)}{p(a,b)}},$$

i.e., equivalently,

$$\left(\prod_{k=1}^{K}\prod_{a,b\in S_k}\sqrt{\frac{p(a,b)}{1-p(a,b)}}\right)\cdot\left(\prod_{k<\ell}\prod_{a\in S_k}\prod_{b\in S_\ell}\sqrt{\frac{1-p(a,b)}{p(a,b)}}\right).$$

Instead of individual values, we can now consider the average factors. Thus, it makes sense to consider the average values f_k and $f_{k\ell}$ for which

$$\prod_{a,b\in S_k} f_k = f_k^{P_k} = \prod_{a,b\in S_k}\sqrt{\frac{p(a,b)}{1-p(a,b)}}$$

and

$$\prod_{a\in S_k}\prod_{b\in S_\ell} f_{k\ell} = f_{k\ell}^{N_k\cdot N_\ell} = \prod_{a\in S_k}\prod_{b\in S_\ell}\sqrt{\frac{p(a,b)}{1-p(a,b)}}.$$

Thus,

$$f_k = \left(\prod_{a,b\in S_k}\sqrt{\frac{p(a,b)}{1-p(a,b)}}\right)^{1/P_k} = \left(\prod_{a\in S_k}\prod_{b\in S_k}\sqrt{\frac{p(a,b)}{1-p(a,b)}}\right)^{1/N_k^2}$$

and

$$f_{k\ell} = \left(\prod_{a\in S_k}\prod_{b\in S_\ell}\sqrt{\frac{1-p(a,b)}{p(a,b)}}\right)^{1/(N_k\cdot N_\ell)}.$$

We therefore maximize the expression

$$\prod_{k=1}^{K} p_k\cdot\prod_{k<\ell} p_{k\ell} =$$

$$\prod_{k=1}^{K}\left(\prod_{a\in S_k}\prod_{b\in S_k}\sqrt{\frac{p(a,b)}{1-p(a,b)}}\right)^{1/N_k^2}\cdot\prod_{k<\ell}\left(\prod_{a\in S_k}\prod_{b\in S_\ell}\sqrt{\frac{1-p(a,b)}{p(a,b)}}\right)^{1/(N_k\cdot N_\ell)}.$$

Taking the logarithm of this expression, and taking into account that

$$\ln\left(\sqrt{\frac{1-p(a,b)}{p(a,b)}}\right)=-\ln\left(\sqrt{\frac{p(a,b)}{1-p(a,b)}}\right),$$

we conclude that the optimizing function has the form

$$\sum_{k=1}^{K}d(S_k,S_k)-\sum_{k=1}^{K}\sum_{\ell\neq k}d(S_k,S_\ell),$$

where we denoted

$$d(a,b)\stackrel{\text{def}}{=}\ln\left(\sqrt{\frac{p(a,b)}{1-p(a,b)}}\right)=\frac{1}{2}\cdot\ln(p(a,b))-\frac{1}{2}\cdot\ln(1-p(a,b)).$$

In particular, for $K=2$, we get the expression

$$d(S,S)+d(S^c,S^c)-d(S,S^c),$$

which is similar to the expression used in [2] to define a cluster.

Comment. We can get the exact expression $d(S,S)+d(S^c,S^c)-2d(S,S^c)$ from [2] if we consider the property of objects from the same cluster to be less important than the property that objects from different clusters be different. This can increase the weight given to the corresponding different-objects term $d(S,S^c)$; in particular, we can get the weight equal to 2, as in [2].

8 Disproportionate Clusters: Case of General Uncertainty

Formulation of the problem. For every two objects a and b from the same cluster S_k, we have the expert's degree of confidence $s(a,b)$ that these objects are similar. How can we describe the "average" probability that the objects from this cluster are similar to each other?

First idea and its formalization. The degrees of similarities between objects from the same cluster enter the objective function via a term $f_{\&}(s(a,b):a,b\in S_k)$. What does it mean to replace all these degree of confidence $s(a,b)$ by an "average" one? A natural idea is to come with a single degree s_k so that when we use s_k instead of all the different degrees of confidence $s(a,b)$, we get the exact same result. In other words, we should have $f_{\&}(s_k:a,b\in S_k)=f_{\&}(s(a,b):a,b\in S_k)$. For an Archimedean "and"-operation $f_{\&}(a,b)=g^{-1}(g(a)\cdot g(b))$ this means that

$$g^{-1}\left(\prod_{a,b\in S_k} g(s_k))\right) = g^{-1}\left((g(s_k))^{P_k}\right) = g^{-1}\left(\prod_{a,b\in S_k} g(s(a,b))\right).$$

By applying the function $g(x)$ to both sides, we get a simpler equality

$$\prod_{a,b\in S_k} g(s_k) = (g(s_k))^{P_k} = \prod_{ab\in S_k} g(s(a,b)),$$

based on which we can compute

$$g(s_k) = \left(\prod_{a,b\in S_k} g(s(a,b))\right)^{1/P_k}.$$

If we count each pair twice and denote the number of element in the k-th cluster by N_k, then we get N_k^2 pairs, and thus,

$$g(s_k) = \left(\prod_{a\in S_k}\prod_{b\in S_k} g(s(a,b))\right)^{1/N_k^2}.$$

Similarly, for every two clusters S_k and S_ℓ, the values corresponding to pairs consisting of $a \in S_k$ and $b \in S_\ell$ form the term $f_\&(1-s(a,b) : a \in S_k, b \in S_\ell)$. A natural idea is to come with a single degree $s_{k\ell}$ so that when we use $s_{k\ell}$ instead of all the different degrees of confidence $1-s(a,b)$, we get the exact same result. So, it is reasonable to pick up an "average" degree of confidence $s_{k\ell}$ that $a \in S_k$ and $b \in S_\ell$ are not similar as the value for which

$$f_\&(s_{k\ell} : a \in S_k, b \in S_\ell) = f_\&(1-s(a,b) : a \in S_k, b \in S_\ell).$$

For the Archimedean "and"-operation $f_\&(a,b) = g^{-1}(g(a)\cdot g(b))$ this means that

$$g^{-1}\left(\prod_{a\in S_k}\prod_{b\in S_\ell} g(s_{k\ell}))\right) = g^{-1}\left((g(s_{k\ell}))^{N_k\cdot N_\ell}\right) = g^{-1}\left(\prod_{a\in S_k}\prod_{b\in S_\ell} g(1-s(a,b))\right).$$

By applying the function $g(x)$ to both sides, we get a simpler equality

$$(g(s_{k\ell}))^{N_k\cdot N_\ell} = \prod_{a\in S_k}\prod_{b\in S_\ell} g(1-s(a,b)),$$

based on which we can compute

$$g(s_{k\ell}) = \left(\prod_{a\in S_k}\prod_{b\in S_\ell} g(1-s(a,b))\right)^{1/(N_k\cdot N_\ell)}.$$

In these terms, we want to optimize the overall degree of confidence that:

- within each cluster, the objects are, on average, similar, and
- for every two different clusters, the objects are non-similar.

Thus, we want to find the clustering that maximizes the following degree of confidence

$$s = f_{\&}(s_1, \ldots, s_K, s_{12}, s_{13}, \ldots, s_{K-1,K}).$$

Substituting the expression for the corresponding t-norm, we conclude that

$$s = g^{-1}\left(\prod_{k=1}^{K} g(s_k) \cdot \prod_{k<\ell} g(s_{k\ell})\right) =$$

$$g^{-1}\left(\prod_{k=1}^{K}\left(\prod_{a\in S_k}\prod_{b\in S_k} g(s(a,b))\right)^{1/N_k^2} \cdot \prod_{k<\ell}\left(\prod_{a\in S_k}\prod_{b\in S_\ell} g((1-s(a,b))\right)^{1/(N_k\cdot N_\ell)}\right).$$

Let us simplify this expression. Since the function $g(x)$ is strictly increasing, maximizing s is equivalent to maximizing the value $g(s)$ for which the expression is somewhat simpler:

$$g(s) = \prod_{k=1}^{K} g(s_k) \cdot \prod_{k<\ell} g(s_{k\ell}) =$$

$$\prod_{k=1}^{K}\left(\prod_{a\in S_k}\prod_{b\in S_k} g(s(a,b))\right)^{1/N_k^2} \cdot \prod_{k<\ell}\left(\prod_{a\in S_k}\prod_{b\in S_\ell} g((1-s(a,b))\right)^{1/(N_k\cdot N_\ell)}.$$

Similar to the probabilistic case, we can be simplify this expression if we consider its logarithm

$$\sum_{k=1}^{K} \frac{1}{N_k^2} \cdot \sum_{a\in S_k}\sum_{b\in S_k} d^+(a,b) + \sum_{k<\ell} \frac{1}{N_k\cdot N_\ell} \cdot \sum_{a\in S_k}\sum_{b\in S_\ell} d^-(a,b),$$

where we denoted $d^+(a,b) \stackrel{\text{def}}{=} \ln(g(s(a,b)))$ and $d^-(a,b) \stackrel{\text{def}}{=} \ln(g(1-s(a,b)))$. We can describe it in the following equivalent form:

$$\sum_{k=1}^{K} d^+(S_k,S_k) - \frac{1}{2}\cdot\sum_{k=1}^{K}\sum_{\ell\neq k} d^-(S_k,S_\ell),$$

where for every two sets S_k and S_ℓ and for every function $f(a,b)$, we denote

$$f(S_k,S_\ell) \stackrel{\text{def}}{=} \frac{1}{N_k\cdot N_\ell}\cdot\sum_{a\in S_k}\sum_{b\in S_\ell} f(a,b).$$

In particular, for two clusters, $S_1 = S$ and $S_2 = S^c \stackrel{\text{def}}{=} \Omega - S$, we need to minimize the expression

$$d^+(S,S) + d^+(S^c,S^c) + d^-(S,S^c).$$

Comment. As we have mentioned when we analyzed the probabilistic case, this expression is similar to expressions from [2].

Iterative algorithm. Similarly to the probabilistic case, we can have an iterative algorithm for minimizing the above expression. We start with some subdivision into clusters, then we assign each object a to a cluster k for which the a's contribution to the distances should be the largest, i.e., for which the following expression is the largest possible:

$$\frac{1}{N_k}\cdot\left(d^+(a,S_k)+\sum_{\ell\neq k} d^-(a,S_\ell))\right),$$

where

$$f(a,S_k)\stackrel{\text{def}}{=}\frac{1}{N_k}\cdot\sum_{b\in S_k} f(a,b).$$

Once we assign all the elements, we get new clusters $S_1,\ldots,S_K$, so we can re-assign them, etc.

Second idea. The second idea comes from the fact that maximizing the original objective function is equivalent to maximizing the product

$$\prod_{a\sim b} g(s(a,b))\cdot\prod_{a\not\sim b} g(1-s(a,b)).$$

Instead of using this expression, we can divide the objective function by the product $\prod\limits_{a,b}\sqrt{g(s(a,b))\cdot g(1-s(a,b))}$, where the product is over all possible pairs, and get an equivalent objective function

$$\prod_{a\sim b}\sqrt{\frac{g(s((a,b))}{g(1-s(a,b))}}\cdot\prod_{a\not\sim b}\sqrt{\frac{g(1-s(a,b))}{g(s(a,b))}},$$

i.e., equivalently,

$$\left(\prod_{k=1}^{K}\prod_{a,b\in S_k}\sqrt{\frac{g(s((a,b))}{g(1-s(a,b))}}\right)\cdot\left(\prod_{k<\ell}\prod_{a\in S_k}\prod_{b\in S_\ell}\sqrt{\frac{g(1-s(a,b))}{g(s(a,b))}}\right).$$

Instead of individual values, we can now consider the average factors. Thus, it makes sense to consider the average values f_k and $f_{k\ell}$ for which

$$\prod_{a,b\in S_k} f_k = f_k^{P_k} = \prod_{a,b\in S_k}\sqrt{\frac{g(s(a,b))}{g(1-s(a,b))}}$$

and

$$\prod_{a\in S_k}\prod_{b\in S_\ell} f_{k\ell} = f_{k\ell}^{N_k\cdot N_\ell} = \prod_{a\in S_k}\prod_{b\in S_\ell}\sqrt{\frac{g(s(a,b))}{g(1-s(a,b))}}.$$

Thus,

$$f_k = \left(\prod_{a,b\in S_k}\sqrt{\frac{g(s(a,b))}{g(1-s(a,b))}}\right)^{1/P_k} = \left(\prod_{a\in S_k}\prod_{b\in S_k}\sqrt{\frac{g(s(a,b))}{g(1-s(a,b))}}\right)^{1/N_k^2}$$

and

$$f_{k\ell} = \left(\prod_{a\in S_k}\prod_{b\in S_\ell}\sqrt{\frac{g(1-s(a,b))}{g(s(a,b))}}\right)^{1/(N_k\cdot N_\ell)}.$$

We therefore maximize the expression

$$\prod_{k=1}^{K} p_k \cdot \prod_{k<\ell} p_{k\ell} =$$

$$\prod_{k=1}^{K}\left(\prod_{a\in S_k}\prod_{b\in S_k}\sqrt{\frac{g(s(a,b))}{g(1-s(a,b))}}\right)^{1/N_k^2}\cdot\prod_{k<\ell}\left(\prod_{a\in S_k}\prod_{b\in S_\ell}\sqrt{\frac{g(1-s(a,b))}{g(s(a,b))}}\right)^{1/(N_k\cdot N_\ell)}.$$

Taking a logarithm of this expression, and taking into account that

$$\ln\left(\sqrt{\frac{g(1-s(a,b))}{g(s(a,b))}}\right) = -\ln\left(\sqrt{\frac{g(s((a,b))}{g(1-s(a,b))}}\right),$$

we conclude that the optimizing function has the form

$$\sum_{k=1}^{K} d(S_k,S_k) - \frac{1}{2}\cdot\sum_{k=1}^{K}\sum_{\ell\neq k} d(S_k,S_\ell),$$

where we denoted

$$d(a,b) \stackrel{\text{def}}{=} \ln\left(\sqrt{\frac{g(s(a,b))}{g(1-s(a,b))}}\right) = \frac{1}{2}\cdot\ln(g(s(a,b)) - \frac{1}{2}\cdot\ln(g(1-s(a,b))).$$

In particular, for $K=2$, we get the expression

$$d(S,S) + d(S^c,S^c) - d(S,S^c),$$

which is similar to the expression used in [2] to define a cluster.

9 Conclusion

Recently, a new empirically successful algorithm was proposed for crisp clustering: the K-sets algorithm.

In this paper, we show that a natural uncertainty-based formalization of what is clustering automatically leads to the mathematical ideas and definitions behind this algorithm.

Thus, we provide an explanation for this algorithm's empirical success.

Acknowledgments

This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by an award "UTEP and Prudential Actuarial Science Academy and Pipeline Initiative" from Prudential Foundation.

References

1. B. G. Buchanan and E. H. Shortliffe, *Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*, Addison-Wesley, Reading, Massachusetts, 1984.
2. C.-S. Chang, W. Liao, Y.-S.Chen. and L.-H. Liou, "A mathematical theory for clustering in metric spaces", *IEEE Transactions on Network Science and Engineering*, 2016, Vol. 3, No. 1, pp. 2–16.
3. E. T. Jaynes and G. L. Bretthorst, *Probability Theory: The Logic of Science*, Cambridge University Press, Cambridge, UK, 2003.
4. G. Klir and B. Yuan, "Fuzzy Sets and Fuzzy Logic", Prentice Hall, Upper Saddle River, New Jersey, 1995.
5. H. T. Nguyen, V. Kreinovich, and P. Wojciechowski, "Strict Archimedean t-norms and t-conorms as universal approximators", *International Journal of Approximate Reasoning*, 1998, Vol. 18, Nos. 3–4, pp. 239–249.
6. H. T. Nguyen and E. A. Walker, *A First Course in Fuzzy Logic*, Chapman and Hall/CRC, Boca Raton, Florida, 2006.
7. D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman and Hall/CRC, Boca Raton, Florida, 2011.
8. L. A. Zadeh, "Fuzzy sets", *Information and Control*, 1965, Vol. 8, pp. 338–353.