5-1-2009

Towards Dynamical Systems Approach to Fuzzy Clustering

Vladik Kreinovich
University of Texas at El Paso, vladik@utep.edu

Olga Kosheleva
University of Texas at El Paso, olgak@utep.edu

Comments:
Technical Report: UTEP-CS-09-15
Published in: Dmitri A. Viattchenin (ed.), *Developments in Fuzzy Clustering*, Vever Publ., Minsk, Belarus, 2009, pp. 10-35.

Towards Dynamical Systems Approach to Fuzzy Clustering

Vladik Kreinovich and Olga Kosheleva
University of Texas at El Paso
El Paso, TX 79968, USA
vladik@utep.edu, olgak@utep.edu

Abstract

In many application areas, there is a need for clustering, and there is a need to take fuzzy uncertainty into account when clustering. Most existing fuzzy clustering techniques are based on the idea that an object belongs to a certain cluster if this object is close to a typical object from this cluster. In some application areas, however, this idea does not work well. One example of such application is clustering in education that is used to convert a detailed number grade into a letter grade.

In such application, it is more appropriate to use clustering techniques which are based on a different idea: that an object tends to belong to the same cluster as its nearest neighbor. In this paper, we explain the relationship between this idea and dynamical systems, and we discuss how fuzzy uncertainty can be taken into account in this approach to clustering.

1 Formulation of the Problem

Clustering is important. In many applications areas we have a large number of different objects, with different behavior. For example, in biology, we have many different plants, animals, bacteria, etc. In chemistry, we have many different substances. In astronomy, we have a large number of different objects (planets, stars, galaxies, etc.). In all these situations, the number of objects is so huge and the behavior of these objects is so different that it is not possible to study each object individually.

In such cases, it is often possible to group objects into *clusters* in such a way that all the objects within the same cluster are similar to each other (in some reasonable sense). Clustering enables us to replace a practically impossible task of studying all individual objects with a more doable task of studying the behavior of *typical* objects from different clusters. Since all the objects within a cluster are similar to each other, the analysis of typical objects provides us with a good description of how all the objects behave.

For example, in biology, instead of studying each of millions of animals individually, we classify them into species, subspecies, etc., and study typical

behavior of each species. In astronomy, we classify galaxies into groups (elliptic, spiral, etc.), and then study typical galaxies from each group.

Clustering is not just a new computer technique: clustering is how we humans deal with the challenge of having to deal with a large amount of information. For example, in everyday life, instead of dealing with the details of meteorological information, we classify it into a few meaningful groups such as "sunny", "cloudy", etc. Similarly, instead of dealing individually with hundreds of co-workers, we classify them by their function and by their attitude to us ("a friendly guy from human resources"). Marketers divide the population into groups by age, education, and social status, and design marketing strategies aimed at each resulting cluster.

Since clustering is an important practical problem, numerous techniques have been invented to solve clustering problems. Some of these techniques successfully use fuzzy methods.

In this paper, we will describe an important class of clustering techniques for which fuzzy analogues are needed, and outline how fuzzy techniques can be added to the corresponding techniques.

Grading: educational example of clustering. A simple but important example of clustering comes from grading. A conscientious professor, after analyzing in detail tests, labs, homeworks, quizzes, and other samples of a student's work, can come with a number from 0 to 100 that described the student's knowledge of the class material. This detailed information is definitely important for the student.

This information is also important for others who make decisions affecting the life of this student. For example, a professor sometimes needs to make a decision whether to accept a student in a class; this decision is usually made based on the student's grade for previous classes, especially classes that are important for the class that will be taught. A graduate committee must make a decision: which students should be accepted into a graduate program. A stipend committee must decide which students will be given stipends, etc.

In many of these cases, the actual number grade for each class constitutes too much information; it is desirable to replace this detailed individual information by assigning each student's grade to a certain class, and by dealing with the student based on the class to which his or her grade belongs. In the US, such classes are known as *letter grades*, with A meaning excellent, B meaning good, C meaning satisfactory, D meaning deficient, and F meaning failing.

A letter grade is the only information about the student's performance in the class that goes into the student's transcript, and thus. It is the only information about this performance that is used to make important (often life-changing) decisions about this student. Thus, the question of translating number grades into letter grades is of great importance.

In some cases, this translation is already defined for a given school. In many cases, a grade of 90 and higher means an A, a grade 80 and higher (but below 90) means a B, a grade between 70 and 80 means a C, a grade between 60 and

70 means a D, and a grade below 60 means an F.

However, due to the importance of grading, most schools leave the letter grading policy to individual professors, so that the instructors will be able to adjust their grading policy to each individual subject and each individual class.

Grading: uncertainty and need for clustering. While it may be tempting to simply use the standard grading policy, this is not always the best approach. The main reason for this is that no matter how meticulous the professor, the number grades are inevitably approximate. It is not possible to test all the details of the student's knowledge in a few tests and homeworks. So, each test, in effect, selects a "random" (unpredictable) sample of subtopics and estimates the student's overall knowledge based on this sample.

Students whose knowledge is perfect are not affected by this selection: they will show perfect knowledge for all possible selections. However, for students who have not yet fully mastered all the topics, the number grade depends on which exactly topics are covered in the tests:

- the grade will be somewhat higher if it so happens that the tests cover none of the subtopics in which the student's knowledge is deficient, and
- the grade will be somewhat lower if it so happens that the actual tests contain a significant number of such topics.

Because of this randomness, the number grade reflects not only the student's knowledge, but also the "luck" of matching the tests' questions with the student's knowledge. As a result, while the difference between 80 and 90 usually reflects the true difference in knowledge, the difference between, say, 91 and 92 (and for sure between 91.0 and 91.1) may indicate simply the result of different luck. A student with a number grade of 91 may turn out to actually have a better knowledge of the material than a student with a number grade of 92.

Because of this uncertainty, it may not be a good idea to use a pre-defined number like 90 to serve as a separator between A and B grades. Indeed, in this case, a student with a number grade 89.9 will be given a B, while a student with a number grade of 90.0 will be given a much better letter grade A – while in reality, the level of knowledge of the two students is approximately the same, and it may even be that a student with the number grade of 89.9 has a slightly higher level of knowledge. Since, as we mentioned, a letter grade is used in many important decisions, it is not fair to base the letter grade on a distinction which may not reflect the actual difference in the students' knowledge.

Instead, it is desirable to classify number grades of actual students into clusters corresponding to A, B, etc. – in such a way that number grades from different clusters truly reflect the difference in students' knowledge.

Natural idea: how number grades are clustered now. At present, the number grades are usually clustered as follows: we find a relatively large "gap" separating two grades, a gao which is close to 90, and make this gap a threshold

for A. For example, if students have grades 90.0, 89.8, 89.6, and the next grades (in descending order) are 86.5, 85.9, etc., then it natural to assign A to all the students with the grade 89.6 and above, and B starting with 86.5.

Similarly, we find a threshold for B, etc.

General comment: it is desirable to take uncertainty into account when clustering objects. The need to take uncertainty into account occurs not only when clustering grades, it is a typical problem in clustering.

This need comes from the fact that clustering is based on the known properties of different objects, and the numerical values of the corresponding quantities come either from measurements or from expert estimates. Expert estimates are imprecise, and measurements are also inevitably imprecise. As a result, when we assign an object to the cluster, we need to take into account the original uncertainty.

For example, a medical doctor makes a potentially life-saving decision – whether to perform a surgery or not – based on classifying a tumor as malicious or benign. For many tumors, measurements are very imprecise, so the resulting classification may turn out to be incorrect. To make the right decision, we must know not only how the patient was classified, but also how confident we are about this classification. For example, if we are 90% confident that a patient has an operable cancer, then the surgery is in order. On the other hand, if we are only 30% confident, then it is probably necessary to undertake further tests.

Basic types of uncertainty: interval, probabilistic, fuzzy. In order to describe how to deal with uncertainty, let us briefly describe the main types of uncertainty.

As we have mentioned, the values used for clustering come either from measurements or from expert estimates, and both measurements and expert estimates are imprecise.

In measurement, due to the imperfection of measuring instruments, the result $\widetilde{x}$ of measuring is, in general, different from the actual (unknown) value x of the measured property. In most measurement situations, we know the upper bound Δ on the (absolute value) $|\Delta x|$ of the measurement error $\Delta x \stackrel{\text{def}}{=} \widetilde{x} - x$. This upper bound is usually provided by the manufacturer of the measuring instrument.

Indeed, if we do not even know the upper bound on the measurement error, this means that the actual value x can be as different from the measured value $\widetilde{x}$ as possible – so, in effect, the value $\widetilde{x}$ is a wild guess, not a measurement result.

Once we know the upper bound Δ, we can conclude that the actual (unknown) value of the measured quantity belongs to the interval $[\widetilde{x} - \Delta, \widetilde{x} + \Delta]$. In some measurement situations, the upper bound Δ is the only information that we have about the accuracy of the measuring instrument. In such situations, the only information that we have about x is that x belongs to the above interval. This situation is called *interval uncertainty*; see, e.g., [15, 18].

In other measurement situations, in addition to the upper bound on the measurement error Δx, we also know the probabilities of different values of Δx. In this case, once we know the measurement result $\widetilde{x}$, we can determine not only the interval of possible values of x, but also the probabilities of different values from this interval. Such situations are called *probabilistic uncertainty*; see, e.g., [21].

In case of expert estimates, we have an additional type of uncertainty caused by the fact that experts usually describe their knowledge by using words from a natural language, words like "small" whose meaning is imprecise (fuzzy). Fuzzy logic and other fuzzy techniques provide a way to deal with such *fuzzy uncertainty*; see, e.g., [12, 19].

Uncertainty of classification. In clustering, uncertainty comes not only from the uncertainty in the input data, it only naturally appears in the output data, i.e., in how we classify different objects.

If a new object is well inside one of the clusters, we are absolutely sure that the object belongs to this cluster. However, if an object is in between the two clusters, we can only classify it with uncertainty.

Important reminder: uncertainty techniques are often useful even in the absence of data uncertainty. In the previous paragraphs, we mentioned that uncertainty must be taken into account when clustering. It is important to also mention that uncertainty-related techniques are often useful even if the data uncertainty is negligible, and when the classification output is certain – but when the corresponding mathematical description of the classification problem is computationally difficult to solve.

For example, it is well known that even in the absence of probabilistic uncertainty, probability-based methods (such as Monte-Carlo methods) are useful in solving problems like the computation of multi-dimensional intervals: in such methods, probabilities reflect not our uncertainty about the data, but rather the probabilities (frequencies) of different solutions.

Similarly, fuzzy techniques are often used to solve difficult problem with negligible data uncertainty. A good example of such applications is fuzzy control, which is applicable even when there is no significant data uncertainty – actually, we can make an even stronger statement: the overwhelming majority of applications of fuzzy control are in situations in which there is no significant data uncertainty. In such situations, we ask an expert how to solve the corresponding problem, and we use fuzzy techniques to formalize the answer that the expert gives by using (informal) words from a natural language. In such applications, fuzzy methods are used to formalize not the expert's uncertainty about the data, but rather the expert's fuzziness in describing how the corresponding problem can be solved; see, e.g., [12, 19].

Comparison with typical elements: an idea behind most fuzzy clustering techniques. At present, most fuzzy clustering techniques are based

on the following idea: once we select a "typical" element t_c in each cluster c, we can then assign a new object n to the cluster c for which n is the closest to the corresponding typical object t_c.

For example, if we are trying to classify pets into cats and dogs,

- we classify a new object as a dog if it resembles a typical dog more than it resembles a typical cat, and
- we classify this object as a cat if it is closer to the typical cat.

With this idea, the typical objects uniquely determine how to classify each object, i.e., uniquely determine the clusters. Thus, we can have the following iterative approach to clustering:

- we start with a reasonable first approximation to the typical objects;
- based on the known approximations to typical objects, we classify all the objects into clusters;
- for each of the new clusters, we form a new typical element which better reflects all the elements in this cluster – e.g., as an arithmetic average of all the objects from this cluster;
- based on these new typical elements, we build new clusters, etc.

This idea leads to clustering techniques such as K-means.

In the fuzzy techniques of this type, instead of assigning each object to a certain class, we assign a difficult-to-classify object to several different classes – with the corresponding degrees of uncertainty. In this case, when we estimate the new typical elements, we do not simply take the arithmetic average of all the objects within a class – we take into account the degree with which each belongs to this class (e.g., by using a weighted average).

Such fuzzy clustering methods are known as fuzzy c-means; see, e.g., [5].

Comment. It should be mentioned that there exists other approaches to fuzzy clustering. For example, a very promising approach based on fuzzy tolerance has been proposed by D. Viatchenin; see, e.g., [22, 23].

In some clustering problems, the idea of typical elements does not lead to an adequate clustering. In some application areas, the above typical idea does not lead to a meaningful clustering.

For example, in the above grade clustering problem, the typical idea would mean that we classify a number grade as an A grade as opposed to a B grade if it is closer to the average A-grade (e.g., 95) than to the average B-grade (e.g., 85). Since the midpoint between 85 and 95 is 90, this would simply mean that all the grades of 90 and above are classified as As, while all the grades below 90 are classified as Bs. This threshold classification is exactly what we have shown to be faulty.

In such application areas, a different clustering idea is needed.

What we do in this paper. In this paper, we describe a different clustering idea that can be used in problems like grades clustering, and we discuss how fuzzy uncertainty can be taken into account in such clustering.

2 Clustering Beyond K-Means and Fuzzy c-Means: Main Ideas

Main idea. Since the idea of comparing each object with typical elements of different clusters does not always work well, let us describe an alternative clustering idea.

This idea is as simple and as intuitive as the idea of comparing with typical elements. This idea is as follows. Suppose that a student has just moved to a new town, and has therefore enrolled in a new high school. In high schools, students tend to cluster into groups by interest. Which group will the new student join? If one of the groups contains several students just like him, with exactly the same interests (e.g., love of computers), it is natural to expect that the student will most probably join this group – irrespective of whether a typical student from this group is similar to him or not.

Similarly, when a faculty is deciding between the two offers from the two universities, she may want to prefer a position at the liberal arts college where most computer science faculty are closer to her areas of interest rather than a position at a technical school where the typical faculty may be closer to computer science – but where computer science folks are further away from her interest area.

In general, it is natural to classify an object to the same class as the nearest neighbor – provided, of course, that this nearest neighbor is not too far from the object.

Nearest neighbor idea: a more precise description. Let us describe the above idea in precise terms. In this description, we will assume that there is no uncertainty; the effects of uncertainty will be discussed in the next section. Let us denote by n the number of objects that we need to classify, and let us denote these objects by $a_1, \ldots, a_n$.

The closeness between two objects a and b is usually described by a numerical value $d(a, b) > 0$ characterizing the "distance" (degree of difference) between a and b. This distance is usually assumed to be symmetric, i.e., $d(a, b) = d(b, a)$ for all objects a and b. There is no difference between the object a and itself, so we have $d(a, a) = 0$. For different objects $a \neq b$, we have $d(a, b) > 0$.

So, we assume that we have a symmetric matrix $d(a_i, a_j) = d(a_j, a_i)$ consisting of the corresponding non-negative distances. The diagonal elements $d(a_i, a_i)$ of this matrix are all zeros, all the other elements are positive.

For each object a_i, its nearest neighbor is defined as an object a_j ($j \neq i$) with the smallest possible distance $d(a_i, d_j)$. We fix a threshold $d_0 > 0$, and we

claim that when the distance between the object a_i and its nearest neighbor a_j does not exceed d_0, then a_i and a_j should belong to the same cluster.

We also assume that this is the only criterion for assigning objects to the same cluster. Of course, the notion of belonging to the same cluster is *transitive*: if a_i and a_j belong to the same cluster, and a_j and a_k belong to the same cluster, then a_i and a_k should also belong to the same cluster. This notion is also symmetric: if a_i and a_j belong to the same cluster, then a_j and a_i belong to the same cluster.

Thus, the equivalence relation $\sim$ corresponding to division into resulting clusters (i.e., relation $a \sim b$ meaning that a and b belong to the same cluster) is the transitive symmetric closure of the relation "a_i is the nearest neighbor to a_j, and $d(a_i, a_j) \le d_0$".

Example: clustering of number grades. Let us illustrate this idea on the number grades example. In this example, if the difference between the two neighboring grades is $< d_0$, it is possible that these grades will be assigned to the same cluster. However, once we have a gap of size $> d_0$, we are thus guaranteed that objects on the two sides of this gap belong to two different clusters.

Comment. It is worth mentioning that while the nearest neighbor idea explains the heuristic method of looking for a gap, it does not fully explain this heuristic – because sometimes, it classifies into too many classes.

Indeed, let us assume that we have grade 92.0, 90.1, 90.0, 89.1, 89.0, 86.1, 86.0, 84.1, etc. In this case, the grade 90.0 is the nearest grade to 90.1, and the grade 90.1 is the nearest grade to 90.0. Thus, the two grades 90.0 and 90.1 form a separate cluster. Similarly, the grades 89.1 and 89.0 form a separate cluster, as well as the grades 86.0 and 86.1.

Yes, there is a gap between 86.1 and 89.0, so that every grade of 89.0 and above belongs to a different cluster than every grade of 86.1. However, in addition to this gap, there are several other cluster-separating gaps as well.

Nearest neighbor clustering: algorithm. The nearest neighbor idea leads to the following natural algorithm – the algorithm that uses the standard algorithms of finding the smallest value and for computing the transitive closure; see, e.g., [6].

First, we form the list of nearest neighbors for each element a_i. For that, for each element a_i, we find the smallest of the distances $d(a_i, a_j)$. If this smallest distance does not exceed the threshold d_0, we add the corresponding indices j to the list of nearest neighbors. Due to the need for symmetry, once we add j as a nearest neighbor to i, we also add i to the list of nearest neighbors of j.

After n lists are formed, we start assigning the corresponding n elements to different classes. For that, we first mark all n objects $1, \ldots, n$ as not yet assigned to clusters.

We then start with the cluster containing the element a_1. We mark 1 as assigned to this cluster, then add all nearest neighbors of a_1 to this cluster,

then add all nearest neighbors of all nearest neighbors, etc.

At some stage, no new objects will be added to the first cluster. At this stage, we take the first un-assigned element, and start a new cluster with this element: we take the nearest neighbors, nearest neighbors of nearest neighbors, etc.

Nearest neighbor clustering: computational complexity of the algorithm and of the problem itself. Let us estimate the computational complexity (= running time) of this algorithm and show that this algorithm is indeed (at least asymptotically) the fastest possible.

For each object a_i, finding the smallest of n numbers $d(a_i, a_j)$ requires n steps. Thus, for all n objects, we need $n \cdot n = O(n^2)$ steps.

After the nearest neighbors are found, at each further step, we either add one object to one of the clusters, or we start a new cluster – if no object can be added to the previously formed cluster. At the end, we classify all n objects to $C \le n$ clusters. Thus, the total number of these further steps cannot exceed the sum of the number of objects and of the number of clusters, i.e., cannot exceed $n + C \le 2n$. Hence, the total number of computational steps of the above algorithm is $\le n^2 + 2n = O(n^2)$.

Let us explain that we cannot have an algorithm that uses fewer than $\dfrac{n^2 - n}{4}$ steps. Indeed, if an algorithm uses fewer than $(n^2 - n)/4$ steps, then, since at each step (be it addition, multiplication, etc.) we can process at most two numbers, we can thus process $< (n^2 - n)/2$ numbers. The original symmetric distances matrix $d(a_i, a_j)$ with 0 diagonal contains $n^2 - n$ non-zero elements. Due to symmetry $d(a_i, a_j) = d(a_j, a_i)$, it is sufficient to only describe the $(n^2 - n)/2$ elements corresponding to $i < j$.

Since the algorithm processes fewer than $(n^2 - n)/2$ elements, this means that it never takes into account some distance $d(a_i, a_j)$. However, without knowing this distance, we sometimes cannot correctly classify the objects. For example, in the situation when all other distances exceed the threshold, the correct classification depends on whether the non-processed distance $d(a_i, a_j)$ exceeds this threshold or not:

- if $d(a_i, a_j) > d_0$, then we have no non-trivial clusters, i.e., each of n elements is its own cluster;
- on the other hand, if $d(a_i, a_j) \le d_0$, then objects a_i and a_j form a cluster.

This argument shows that every clustering algorithm requires at least $n^2/4 = O(n^2)$ steps. Since the above algorithm requires $O(n^2)$ steps, this means that the above algorithm is indeed asymptotically optimal.

What if some objects have been pre-classified? In some practical situations, some objects have been pre-classified into clusters.

For example, in the number grade example, usually, a professor promises students, from the very beginning, that grades 90 and higher will lead to the A grade.

It is easy to modify the above algorithm to take pre-classified objects into account. Indeed, after we describe the nearest neighbors, instead of starting with the object a_1, we start with the list of all the objects which have been pre-classified into the first cluster. Then, we add their neighbors, the neighbors of their neighbors, etc. After this first cluster is filled, we take the second pre-classified cluster, add neighbors, neighbors of neighbors, etc.

The computation time remains asymptotically the same $O(n^2)$.

Caution. One needs to be cautious in this approach since it may happen that two objects which were pre-classified into different classes turn out to be nearest neighbors – and thus, according to our algorithm, should be classified into the same cluster. In this case, we a proper classification which correctly assigns all pre-classified classes is not possible.

This may happen

- either when our pre-classification is not correct,
- or when the observations are imprecise, and, as a result, the nearest neighbors are erroneously determined.

Comment. In the next section, we will discuss how to deal with the imprecision of the observations.

Application to grade clustering. For number grades, in addition to pre-classifying all the grades of 90 and higher as As, we may also have more complex promises. For example, a professor may also promise:

- that grades 80 and higher will always lead to A or B,
- that grades 70 and higher will always means A, B, or C,
- etc.

Such more complex promises can also be easily included in the nearest neighbor algorithm. Indeed, first, we find the nearest neighbors. Then, we start forming a new cluster for As: we start with grades 90 and higher that were pre-classified as As, add nearest neighbors, nearest neighbors of nearest neighbors, etc., until we fill this cluster.

Then, we start filling the B cluster. Due to our promise of As or Bs for an grade of 80 or above, all the values 80 and higher which have not been classified as As will thus have to be classified as Bs. To these grades, we add nearest neighbors, nearest neighbors or nearest neighbors, etc., until we fill this cluster too.

Then, we start filling the C cluster. We start will all the grades 70 and above which have not yet been classified as As or Bs, and classify them as Cs. Then, we add nearest neighbors, etc., until we fill in the C cluster as well.

After that, we perform a similar procedure with the D cluster. All remaining grades will now be classified as Fs.

Nearest neighbor algorithm reformulated in terms of dynamical systems. After discussing possible modifications, let us go back to the original nearest neighbor algorithm.

The above description of this algorithm does not provide a clear mathematical description of the resulting clusters. To get such a description, let us take into account that in many practical problems, we have a large number of objects that need to be classified. When we have a large number of objects, instead of listing individual objects, we can list the *density* of such objects in different areas. This use of density is similar to the fact that to describe a macroscopic body, we do not list the coordinates of all individual molecules; instead, we list the densities at different spatial locations.

The density at each location x can be determined if we:

- take a small neighborhood of x,
- count the number of objects in this neighborhood, and
- divide this number by the volume of this neighborhood.

It may also be beneficial to take objects from different parts of this neighborhood with different weights – so that objects closer to x get larger weights than objects which are further away from x.

In the 1-D case, the density $\rho(x)$ at a location x means that on average, we have $\rho(x) \cdot \Delta x$ objects in an area of size Δx. The distance from an object to its nearest neighbor can be estimated as the value Δx for which the interval of size Δx contains one more element, i.e., as the value for which $\rho(x) \cdot \Delta x = 1$. From this equation, we conclude that $\Delta x \sim 1/\rho(x)$.

Similarly, in the 2-D case, we have $\rho(x) \cdot (\Delta x)^2 \sim 1$ hence $\Delta x \sim 1/(\rho(x))^{1/2}$. In the 3-D case, we have $\rho(x) \cdot (\Delta x)^3 \sim 1$ hence $\Delta x \sim 1/(\rho(x))^{1/3}$, etc.

In general, the larger the density $\rho(x)$, the closer the corresponding nearest neighbor.

According to the nearest neighbor algorithm, an object belongs to the same class at its nearest neighbor – as long as the distance to the nearest neighbor does not exceed a certain threshold d_0. This threshold can be translated into the corresponding threshold ρ_0 for the density: once the density becomes smaller than this threshold value ρ_0, objects are no longer classified into clusters.

When the density is at or above the threshold value, each object has neighbors in all directions. Since the distance to the nearest neighbor decreases with the increase in density, the nearest neighbor is located in the direction in which the density is the largest. Thus, the nearest neighbor to a point x is located in the direction in which the density is the largest.

A direction can be described by a unit vector $\vec{e}$ with coordinates $(e_1, \ldots, e_d)$, where d is the dimension of the corresponding space. The change of density in the direction $\vec{e}$ can be described if we expand the expression $\rho(x + \Delta x \cdot \vec{e})$ into Taylor series in Δx and retaining only the main (linear) terms in this expansion:

$$\rho(x_1 + \Delta x \cdot e_1, \ldots, x_d + \Delta x \cdot e_d) = \rho(x_1, \ldots, x_n) + \sum_{i=1}^{d} \frac{\partial \rho}{\partial x_i} \cdot \Delta x \cdot e_i + o(\Delta x).$$

In vector terms, the sum in this formula is simply a scalar (dot) product of the vector $\Delta x \cdot \vec{e}$ and the gradient vector

$$\nabla \rho \stackrel{\text{def}}{=} \left(\frac{\partial \rho}{\partial x_1}, \ldots, \frac{\partial \rho}{\partial x_n} \right).$$

Thus,

$$\rho(\vec{x} + \Delta x \cdot \vec{e}) = \rho(\vec{x}) + \Delta x \cdot (\vec{e} \cdot \nabla \rho).$$

The scalar product of the two vectors is equal to the product of their lengths and the cosine of the angle between them.

In our case, the unit vector $\vec{e}$ has length 1. The length of the gradient vector is also fixed. So the scalar product – and hence, the density – is the largest when the cosine is the largest. Cosine attains its largest value 1 when the angle is 0, i.e., when the unit vector $\vec{e}$ is parallel to the direction of the gradient.

Thus, if we go from a point x in the direction of the gradient, we remain in the same cluster. In precise terms, going in the direction of the gradient means following the trajectories of the following dynamical system:

$$\frac{dx}{dt} = \nabla \rho.$$

This system is called the *gradient ascent* system.

Thus, a point x belongs to the same cluster as all the points on this trajectory – and as the point with the largest density that we encounter along this trajectory.

In terms of dynamical systems, the corresponding limit points are *attraction points*, and the domain of the values of x is divided into *attraction basins* that end up in different attraction points.

Thus, from the dynamical systems viewpoint, the nearest neighbor clustering can be described as follows:

- first, we select only the points at which the density exceeds as certain threshold ρ_0;
- then, we form a gradient ascent dynamical system corresponding to the density function; the attraction basins of this dynamical system are exactly the desired clusters.

What is known about solving the corresponding dynamical systems problem. The gradient ascent system is well-studied, since gradient ascent is one of the basic methods of numerical optimization.

In particular, for this problem, efficient algorithms are known for the following *watershed* formulation of this problem. We have a map that described the height h as a function of a spatial location x. Water floes in the direction of steepest descent of the height function (i.e., equivalently, in the direction of steepest ascent of opposite function $-h(x)$). We need to separate all the locations depending on where the water at location x ends up.

For example, in the continental US, rainwater ends up either in the Atlantic Ocean or in the Pacific Ocean. From the mathematical viewpoint, we this need to describe the attraction basins of the corresponding gradient ascent system. Corresponding algorithms are overviewed, e.g., in [8, 9]; see also [3, 24].

Comment. Sometimes, the clusters appear even without the search for attraction basins, simply when we delete all the points with low density. In some cases, points with high density then form disconnected clusters.

This is true, e.g., for many astronomical images, where individual stars can be selected as objects whose brightness exceeds the background level. In this example, only for a few double objects, the dynamical systems algorithm leads to further clustering; for most objects, each connected component of the set $\{x : \rho(x) \geq \rho_0\}$ is its own attraction basin.

There exist several algorithms that simply find connected components of the region in which the density of data objects exceed a given threshold; see, e.g., DBSCAN [2, 7] and OPTICS [1].

3 Need to Take Uncertainty into Account in Dynamical Systems Approach to Clustering

Need to take uncertainty into account: reminder. As we have mentioned, the data often come with uncertainty. In this case, it is desirable to take uncertainty into account when clustering.

In particular, it is desirable to take uncertainty into account when we use the dynamical systems approach to clustering. Indeed, in the nearest neighbor approach, we base our assignment on the fact that some object b is the nearest neighbor of the object a, i.e., that we) have $d(a, b) < d(a, c)$ for all other objects c.

In the idealized case when the distances are known exactly, the inequality $d(a, b) < d(a, c)$ indeed means that b is the nearest neighbor – and thus, that objects a and b should be assigned to the same cluster. However, due to the uncertainty of measurements and expert estimates, the measured (or estimated) distances are, in general, somewhat different from the actual ones. Thus, when the measured distances $\widetilde{d}$ satisfy the inequality $\widetilde{d}(a, b) < \widetilde{d}(a, c)$ and the difference between these measured distances is small, it may be that the actual

distances satisfy the opposite inequality $d(a,b) > d(a,c)$. In this case, in reality, a belongs to the same class as c, but we erroneously classify it with b.

To avoid such misclassifications, it is desirable to only classify into the same cluster objects a and b for which we are sure that a and b are probably nearest neighbors.

Existing heuristic ideas of how to take uncertainty into account. One possible way to avoid misclassifications is to use the *shared nearest neighbors approach* in which, for each object a, we find k nearest neighbors, and classify a and b into the same cluster if at least one object appears in both lists. This algorithm was proposed by Jarvis and Patrick in 1973 [11] and has been successfully used since then.

Several other similar ideas have been proposed, ideas that have implemented in several modifications of the shared nearest neighbors algorithm.

Comment. A similar idea is used in the mathematical morphology approach to image processing; see, e.g., [10, 17]. Specifically, in this algorithm, we divide the points of a black-and-white image into points that belong to the object (i.e., should be white) and points that correspond to the background (i.e., should be black).

The original image most contains the correct points, but some points are corrupted by noise:

- some points which should be shown as belonging to the object (white) are erroneously shown as belonging to the background (black); these points are called *pepper* noise;
- on the other hand, some points which should be shown as belonging to the background (black) are erroneously shown as belonging to the object (white); these points are called the *salt* noise.

To eliminate the pepper noise, we can use the idea that a point for which most nearest neighbors are white should also be white. Similarly, to eliminate the salt noise, we can use the idea that a point fir which most nearest neighbors are black should also belong to the black cluster.

Possible way of taking interval uncertainty into account. Let us consider the situation in which we measure the quantities characterizing each object with interval uncertainty, with some accuracy Δ. In this case, we can also find the bound on the uncertainty with which we know the distance $d(a,b)$ based on the imprecise measurement results.

For example, for clustering number grades, the distance is simply equal to $d(a,b) = |a-b|$. In this case, if we know a and b with accuracy Δ, i.e., if we know the grades $\widetilde{a}$ and $\widetilde{b}$ which may be different from the actual (unknown) values a and b describing the students' knowledge ($|\widetilde{a}-a| \le \Delta$ and $|\widetilde{b}-b| \le \Delta$),

then the distance is known with accuracy 2Δ:

$$|d(\widetilde{a},\widetilde{b})-d(a,b)|\le 2\Delta.$$

When we sort all the measured grades in descending order into a sequence $\widetilde{g}_1\ge\widetilde{g}_2\ge\ldots$, the grades $\widetilde{g}_i$ and $\widetilde{g}_{i+1}$ are classified to different clusters if they are not nearest neighbors to each other. In this case, the grade $\widetilde{g}_{i-1}$ is the nearest neighbor for $\widetilde{g}_i$, and the grade $\widetilde{g}_{i+2}$ is the nearest neighbor for $\widetilde{g}_{i+1}$, i.e., $\widetilde{g}_{i-1}-\widetilde{g}_i<\widetilde{g}_i-\widetilde{g}_{i+1}$ and $\widetilde{g}_i-\widetilde{g}_{i+1}>\widetilde{g}_{i+1}-\widetilde{g}_{i+2}$.

Since the grades g_i are only approximately known, we would like to classify them into different clusters only if we are absolutely sure that these inequalities are satisfied

- not only for the measured grades,
- but also for the actual (unknown) grades g_i for which $|\widetilde{g}_i-g_i|\le\Delta$.

One can check that this absolute assurances happens when the difference between the gaps exceeds 4Δ:

$$(\widetilde{g}_i-\widetilde{g}_{i+1})-(\widetilde{g}_{i-1}-\widetilde{g}_i)>4\Delta;\quad(\widetilde{g}_i-\widetilde{g}_{i+1})-(\widetilde{g}_{i+1}-\widetilde{g}_{i+2})>4\Delta.$$

Comment. A similar idea can be used in the more general situations of interval uncertainty.

Possible ways to take fuzzy uncertainty into account. When we only have fuzzy information about the objects, we thus have fuzzy values of the distances $d(a,b)$. A fuzzy set is, in effect, a nested family of intervals – the alpha-cuts of this fuzzy set. Thus, to handle fuzzy uncertainty, and can pick some level α, and use the corresponding alpha-cut intervals to appropriately modify the clustering algorithm.

Possible ways of taking probabilistic uncertainty into account. Due to uncertainty, some points may be misplaced. Let us assume that we know the portion of misplaced points, e.g., we know that at most 5% of the points are misplaced.

A single misplaced point can bias the clustering. For example, a point erroneously placed in the gap between the two clusters can lead to their merger into a single cluster. It is therefore desirable to make sure that the clusters that we come up with are robust with respect to such mis-placements.

For that, we can use the Monte-Carlo approach. Specifically, in addition to classifying the original data, we delete randomly selected 5% of the points and repeat the clustering procedure. We repeat this delete-and-cluster procedure several times, and get several clusterings.

Now, we only assign two points a and b to the same cluster, if they are assigned to the same cluster in the majority of the resulting clusterings – after which, we apply the transitive closure to restore the equivalence relation.

4 Computational Complexity of Clustering under Uncertainty: Probabilistic vs. Fuzzy Approaches

Formulation of the problem. What is the computational complexity of the problem of clustering under uncertainty? In this section, we compare the complexity of this problem for probabilistic and fuzzy uncertainty.

This comparison does now always make sense. For example, if we know the exact probabilities of different measurement errors, then we should use probabilistic methods. However, in many practical situations, our uncertainty is subjective, it can be, in principle, described both in the probabilistic and in fuzzy terms. In situations when we have to classify a large number of objects and thus computation time is an issue, it makes sense to select a techniques which leads to faster (and thus, more feasible) computations.

In this section, we consider the simplified problem of producing the most probable clustering – without computing the uncertainty of this classification. The main reason why we make this restriction is that while clustering itself is straightforward – we classify each object into a certain cluster – there are many different ways of estimating the clustering uncertainty, with drastically different computational complexity.

We will show that for this simplified problem,

- the probabilistic clustering problem is, in general, NP-hard (computationally difficult) already for the case of two clusters, while
- the corresponding fuzzy clustering problem is reasonably easy to solve.

Thus, if both techniques can be applied (and if computation time is an issue), we should use fuzzy techniques.

Clustering under probabilistic uncertainty: formulation of the problem. Let us assume that for every two objects a and b, we know the probability $p_+(a,b)$ that a and b should belong to the same cluster – and, correspondingly, the probability $p_-(a,b) = 1 - p_+(a,b)$ that a and b should belong to different clusters.

Let us also assume that the probabilities corresponding to different pairs are independent. In this case, for each subdivision of n original objects into two clusters A and B, with $A \cap B = \emptyset$ and $A \cup B = \{a_1, \ldots, a_n\}$, the probability $P(A)$ that this clustering is consistent with our knowledge is equal to the product of the corresponding probabilities:

$$P(A) = \prod_{a,a' \in A} p_+(a,a') \cdot \prod_{b,b' \in B} p_+(a,b') \cdot \prod_{a \in A, b \in B} (1 - p_+(a,b)).$$

Our objective is to find the set A for which this probability is the largest possible.

Clustering under probabilistic uncertainty is NP-hard: a proof. Let us prove that the above problem of clustering under probabilistic uncertainty is NP-hard [13]. (The fact that clustering itself is NP-hard is well know; see, e.g., [4, 25].)

Crudely speaking, NP-hard means that (unless P=NP, which most computer scientists believe to be impossible), it is not possible to have a feasible algorithm that solves all the particular cases of this problem in feasible time; for exact definitions, see [16, 20].

To be more precise, a problem is NP-hard if all the problems from a certain class NP can be reduced to it. The possibility of such a reduction shows that this problem is indeed the hardest of the problems from this class NP.

In view of this definition, a usual way to prove that a problem $\mathcal{P}$ is NP-hard is to reduce a known NP-hard problem $calP_0$ to this problem $\mathcal{P}$. In this case, every problem from the class NP can be reduced to $\mathcal{P}_0$, and since $\mathcal{P}_0$ can be reduced to $\mathcal{P}$, every problem from NP can be reduced to $\mathcal{P}$ as well – by transitive of reduction. Thus, by definition of NP-hardness, the problem $\mathcal{P}$ is indeed NP-hard.

In our proof, as the known NP-hard problem, we will use the following *subset sum* problem: given n numbers $s_1, \ldots, s_n$, find a subset $A \subseteq \{1, \ldots, n\}$ for which

$$\sum_{i \in A} s_i = \sum_{j \notin A} s_j,$$

i.e., in other words,

$$\sum_{i \in A} s_i = \frac{1}{2} \cdot S,$$

where we denoted

$$S \stackrel{\text{def}}{=} \sum_{i=1}^{n} s_i.$$

To reduce this problem to our probabilistic clustering problem, we take

$$p_+(a_i, a_j) = \frac{1}{1 + \exp(s_i \cdot s_j)}.$$

In this case,

$$p_-(a_i, a_j) = 1 - p_+(a_i, a_j) = \frac{\exp(s_i \cdot s_j)}{1 + \exp(s_i \cdot s_j)}.$$

For these expressions, the product of the denominators in the formula for $P(A)$ is always the same: it simply equals to the product of all the terms $1+\exp(s_i \cdot s_j)$ for all possible pairs $i < j$. Thus, the expression $P(A)$ attains its largest possible value if and only if its numerator is the largest.

This numerator only comes from the terms $a \in A$, $b \in B$ and is thus equal to $\prod_{i \in A, j \notin A} \exp(s_i \cdot s_j)$. This product is the largest if and only if its logarithm L

takes the largest value. Since the logarithm of the product is equal to the sum of the corresponding logarithms, this logarithm L takes the form

$$L = \sum_{i \in A, j \notin A} s_i \cdot s_j,$$

or, equivalently,

$$L = \sum_{i \in A} s_i \cdot \sum_{j \notin A} s_j.$$

By using the notation $S = \sum\limits_{i=1}^{n} s_i$, we conclude that $L = S(A) \cdot (S - S(A))$, where we denoted $S(A) \stackrel{\text{def}}{=} \sum\limits_{i \in A} s_i$. The product $z \cdot (S - z)$ attains its largest possible value when $z = S/2$. Thus, the expression L attains its largest possible value when $S(A) = S/2$. So, if we find a clustering that maximizes the corresponding probability $P(A)$, we can thus find the set A that solves the original subset sum problem. The reduction is complete, so the probabilistic clustering problem is indeed NP-hard.

Comment. In this reduction, we use a somewhat unusual clusterization problem: e.g., if the values s_i and s_j are both large, the probability $p_+(a_i, a_j)$ should be small. So even two very similar objects with large $s_i \approx s_j$ should be assigned to different clusters. This corresponds more to the use of clustering in grouping people (so that they are most productive) than to the more traditional problem of dividing objects into clusters of similar ones.

In a more traditional setting, when we have the probabilities $p_A(a)$ and $p_B(a) = 1 - p_A(a)$ of each object belonging to the corresponding class, classification is much easier. Indeed, the corresponding probability

$$P(A) = \prod_{i \in A} p_A(a_i) \cdot \prod_{j \notin A} p_B(a_j)$$

is equal to the product $\prod\limits_{i=1}^{n} p_B(a_i)$ (which does not depend on our choice of the set a) times the auxiliary product

$$N(A) = \prod_{i \in A} \frac{p_A(a_i)}{p_B(a_i)}.$$

Thus, maximizing the probability $P(A)$ is equivalent to maximizing this auxiliary product.

For each object i, the contribution to the auxiliary product $N(A)$ is

- either equal to 1 (if $i \notin A$)
- or equal to the ratio $p_A(a_i)/p_B(a_i)$ (if $i \in A$).

Thus, to maximize the product $N(A)$, we should take, for every i, the largest of these two values:

- we take $i \in A$ is $p_A(a_i) > p_B(a_i)$, i.e., if $p_A(a_i) > 1/2$;
- we take $j \notin A$ if $p_A(a_i) \le 1/2$.

In this case, classification is easy.

Similarly, a classification into three or more groups is easy: we assign each object a to the cluster A for which this object's probability $p_A(a)$ of belonging to this cluster is the largest.

However, in a slightly different formulation, when we only have partial information about probabilities, we again get an NP-hard problem. For example, let us assume that for cluster 1,

- we know the exact mean – which is equal to the arithmetic average of all the objects, and
- we know that exactly one half of the elements must belong to this cluster.

In this selecting such a cluster is also equivalent to solving the subset sum problem (we can always add 0s to make sure that we can have exactly half of elements in the class A).

Clustering under fuzzy uncertainty: formulation of the problem. Let us now consider a similar problem under fuzzy uncertainty.

We assume

- that for every two objects a and b, we know the degree $d_+(a,b)$ with which the experts believe that a and b should belong to the same cluster,
- and that we also know the degree $d_-(a,b)$ that a and b should belong to different clusters.

For the sake of generality, in addition to the case when $d_-(a,b) = 1 - d_+(a,b)$, we also allow for the case of partial knowledge, when $d_-(a,b) + d_+(a,b) < 1$ (this possibility is usually considered in so-called *intuitionistic fuzzy sets*).

In the probabilistic case, we did not aim for this generality, since we were proving a negative result – and a negative result for subproblem implies the negative result for any more general problem. However, in the fuzzy case, we are interested in producing an algorithm, so we should consider as general a situation as possible.

We want to make sure that all the expert knowledge is satisfied, i.e., that the statements about all the pairs are satisfied as much as possible. The resulting combined statement is the result of applying "and" to the statements about individual pairs. The simplest "and"-operation in fuzzy logic is min. Thus, for each cluster A, the degree $d(A)$ with which this cluster satisfies all the conditions is equal to $d(A) = \min(d_A, d_B, d_{AB})$, where B is a complement to A, and

$$d_A \stackrel{\text{def}}{=} \min_{a,a' \in A} d_+(a,a'), \quad d_B \stackrel{\text{def}}{=} \min_{b,b' \in B} d_+(b,b'), \quad d_{AB} \stackrel{\text{def}}{=} \min_{a \in A, b \in B} d_-(a,b).$$

It makes sense to only consider clusters for which this degree of satisfaction is larger than the degree $1-d(A)$ of non-satisfaction, i.e., for which $d(A) > 1/2$.

Clustering under fuzzy uncertainty: an efficient algorithm. Let us show:

- that the existence of a clustering A with $d(A) > 1/2$ can be checked by a simple algorithm, and
- a similarly simple algorithm can provide a clustering for which the degree $d(A)$ is the largest possible.

Indeed, from the fact that the minimum $d(A)$ of several degree is larger than $1/2$, we can conclude that each of these degree is larger than $1/2$. Thus,

- when a and $a' \in A$, we get $d_+(a, a') > 1/2$, and
- when $a \in A$ and $b \in B$, we get $d_-(a, b) > 1/2$ and therefore,

$$d_+(a, b) \le 1 - d_-(a, b) < 1/2.$$

So, let us pick an arbitrary element a. Let A denote the cluster than contains this element. Then, we have the following simple algorithm for checking, for every other object x, whether $x \in A$:

- if $d_+(a, x) > 1/2$, then we should have $x \in A$ (since $x \notin A$ would imply $d_+(a, x) < 1/2$); and
- if $d_+(a, x) < 1/2$, then we should have $x \notin A$ (since $x \in A$ would imply $d_+(a, x) > 1/2$).

Once we formed A, we can then check whether $d_+(a, a') > 1/2$ for all the elements a, a' of this newly formed set A, and whether indeed $d_+(a, b) < 1/2$ for all $a \in A$ and all $b \notin A$.

If we only classify into two clusters, then we automatically get thbe second cluster B as the complement to A, we just need to check that $d_+(b, b') > 1/2$ for all objects b and b' from this cluster B.

If we need more clusters, we pick an arbitrary object which is not in A, call it b, and similarly form the new cluster B as the set of all the objects x for which $d_+(b, x) > 1/2$, etc.

So, clustering under fuzzy uncertainty is indeed a straightforward problem.

Comment. It should be emphasized that the existence of the fast clustering algorithm was made possible by our choice of min as the fuzzy "and"-operation. For the product, we would be able to have the same NP-hardness proof as for the case of the probabilistic uncertainty.

5 Open Problems and Future Work

Main open problem: adding fuzzy uncertainty to dynamical system clustering techniques. In this paper, we argued that in many practical applications, clustering techniques based on the dynamical system approach lead to much more adequate clustering than the more traditional clustering methods based on typical samples.

It is therefore desirable to add fuzzy uncertainty to such dynamical system clustering techniques.

Beyond dynamical system clustering techniques. While dynamical systems techniques are often more adequate than techniques based on typical elements, dynamical system techniques are not perfect either.

For example, in image processing, when we want to recognize a small object on top of the large one: e.g., when the smaller star from a double-star system passing in front of its larger companion. We will be able to recognize the smaller object if it is significantly brighter than the background. In this case, we have a local maximum which will (hopefully) be recognized by the gradient ascent techniques.

Similarly, we will be able to recognize a group of similar object against a background of a more general group.

However, when a smaller object passed at the edge of the larger one, where the density of the larger object decreases, this decrease may compensate for the increase caused by the passing. In this case, there is no local maximum and thus, the gradient ascent method will not notice the new object – while visually, we can often clearly see it.

It is therefore desirable to come up with techniques that will recognize the smaller objects (or smaller clusters) in such cases as well.

Acknowledgments

This work was supported in part by the National Science Foundation grant HRD-0734825, by Grant 1 T36 GM078000-01 from the National Institutes of Health, and by Grant MSM 6198898701 from MŠMT of Czech Republic.

The authors are greatly thankful to Misha Koshelev for his help and advise on statistical clustering (and for his NP-hardness proof), and to Dmitri Viatchenin for his interest and encouragement.

References

[1] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering Points To Identify the Clustering Structure", *Proceedings of the ACM SIGMOD 1999 International Conference on Management of Data*, ACM Press, 1999, pp. 49–60.

[2] D. Arlia and M. Coppola, "Experiments in Parallel Clustering with DB-SCAN", in *Proceedings of the 7th International Parallel Processing Conference Euro-Par'2001*, Manchester, U.K., August 28–31, 2001, Spinger-Verlag.

[3] D. Austin, A review of [9], *Notices of the American Mathematical Society*, 2009, Vol. 56, No. 2, pp. 237–239.

[4] M. Bern and D. Eppstein, "Approximation algorithms for geometric problems", in: *Approximation Algorithms for NP-Hard Problems*, PWS Publishing, 1996, pp. 296–345.

[5] J. C. Bezdek, N. Pal. J Keller, and R. Krishnapuram, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, Kluwer Academic Publ., Dordrecht, 1999.

[6] C. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Boston, Massachusetts, 2001.

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining KDD'96*, AAAI Press, 1996, pp. 226–231.

[8] B. Hayes, "Dividing the continent", *American Scientist*, 2000, Vol. 88, No. 6, pp. 481–485.

[9] B. Hayes, *Group Theory in the Bedroom, and Other Mathematical Diversions*, Hill and Wang Publ., 2008.

[10] H. J. A. M. Heijnmans, *Morphological Image Operators*, Academic Press, Boston, Massachusetts, 1994.

[11] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared nearest neighbours", *IEEE Transasction on Computers*, 1973, Vol. C-22, pp. 1025–1034.

[12] G. Klir amd B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applicatins*, Prentice Hall, Upper Saddle River, New Jersey, 1995.

[13] M. Koshelev, unpublished result.

[14] O. M. Kosheleva and M. Ceberio, "Processing Educational Data: From Traditional Statistical Techniques to an Appropriate Combination of Probabilistic, Interval, and Fuzzy Approaches", *Proceedings of the International Conference on Fuzzy Systems, Neural Networks, and Genetic Algorithms FNG'05*, Tijuana, Mexico, October 13–14, 2005, pp. 39–48.

[15] V. Kreinovich, "Interval Computations and Interval-Related Statistical Techniques: Tools for Estimating Uncertainty of the Results of Data Processing and Indirect Measurements", In: F. Pavese and A. B. Forbes (eds.), *Data Modeling for Metrology and Testing in Measurement Science*, Birkhauser-Springer, Boston, 2009, pp. 117–145.

[16] V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl, *Computational Complexity and Feasibility of Data Processing and Interval Computations*, Kluwer Academic Publishers, Dordrecht, 1998.

[17] G. Matheron, *Randm Sets and Integral Geometry*, Wiley, New York, 1975.

[18] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*, SIAM Press, Philadelphia, Pennsylvania, 2009.

[19] H. T. Nguyen and E. A. Walker, *A First Course in Fuzzy Logic*, CRC Press, Boca Raton, Florida, 2006.

[20] C. Papadimitriou, *Computational Complexity*, Addison Welsey, Reading, Massachusetts, 1994.

[21] S. Rabinovich, *Measurement Errors and Uncertainties: Theory and Practice*, Springer Verlag, New York, 2005.

[22] D. A. Viatchenin, "Construction of the formation solutions in the spaces of fuzzy tolerances as the approach to the solution of the task of the fuzzy cluster - analysis: conceptual aspect of a problem", *Proceedings of the International Symposium on Reflexive Processes and Control RPC'2001*, Moscow, Russia, October 8–10, 2001.

[23] D. A. Viattchenin, "A Heuristic Approach to Possibilistic Clustering for Fuzzy Data", *Journal of Information and Organizational Sciences*, 2008, Vol. 32, No. 2.

[24] L. Vincent and P. Soille, "Watersheds in digital spaces: an effective algorithm based on immersion simulations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991, Vol. 13, pp. 583–598.

[25] W. J. Welch, "Algorithmic complexity: three NP-hard problems in computational statistics", *Journal Statist. Comput. Simul.*, 1981, No. 15, pp. 17–25.