Density-Based Fuzzy Clustering as a First Step to Learning Rules: Challenges and Solutions

Gözde Ulutagay[1] and Vladik Kreinovich[2]

[1][1]Department of Industrial Engineering, Izmir University, Uckuyular-Izmir, Turkey,
gozde.ulutagay@gmail.com
[2]University of Texas at El Paso, El Paso, TX 79968, USA, vladik@utep.edu

Abstract. In many practical situations, it is necessary to cluster given situations, i.e., to divide them into groups so that situations within each group are similar to each other. This is how we humans usually make decisions: instead of taking into account all the tiny details of a situation, we classify the situation into one of the few groups, and then make a decision depending on the group containing a given situation. When we have many situations, we can describe the probability density of different situations. In terms of this density, clusters are connected sets with higher density separated by sets of smaller density. It is therefore reasonable to define clusters as connected components of the set of all the situations in which the density exceeds a certain threshold t. This idea indeed leads to reasonable clustering. It turns out that the resulting clustering works best if we use a Gaussian function for smoothing when estimating the density, and we select a threshold in a certain way. In this paper, we provide a theoretical explanation for this empirical optimality. We also show how the above clustering algorithm can be modified so that it takes into account that we are not absolutely sure whether each observed situation is of the type in which we are interested, and takes into account that some situations "almost" belong to a cluster.

1 Formulation of the Challenges

Clustering is how we humans make decisions. Most algorithms for control and decision making take, as input, the values of the input parameters, and transform them into the optimal decision (e.g., into an optimal control value). Humans rarely do that. When facing a need to make a decision – e.g., where to go eat, which car or which house to buy, which job offer to accept – we rarely write down all the corresponding numbers and process them. Most of the time, for each input variable, instead of its exact known value, we only use a category to which this value belongs. For example, to decide where to eat, instead of the exact prices of different dishes, we usually base our decision on whether the restaurant is cheap, medium, expensive, or very expensive. Instead of taking into account all the menu details, we base our decision on whether this restaurant can be classified as Mexican, Chinese, etc. Similarly, when we select a hotel to stay during a conference, instead of taking into account all the possible features, we base our decision on how many stars this hotel has and whether it is walking distance, close, or far away from the conference site.

In all such cases, before we make decisions, we *cluster* possible situations, i.e., divide them into a few groups – and then make a decision based on the group to which the current situation belongs.

Clustering is a natural first step to learning the rules. Humans are often good at making decisions. In many situations – such as face recognition – we are much better than the best of the known computer programs, in spite of the fact that computers process data much faster than we humans. To improve the ability of computers to solve problems, it is therefore reasonable to emulate the way we humans make the corresponding decisions. This means, in particular, that a reasonable way to come up with a set of good-quality decision rules is to first cluster possible situations, and then make a decision based on the cluster containing the current situation.

Clustering: ideal case. How shall we cluster? In order to cluster, we need to have a set of situations, i.e., vectors $x = (x_1, \ldots, x_n)$ consisting of the values of n known quantities that characterize each situation.

Let us first consider the case when we have so many examples that in the vicinity of each situation $x = (x_1, \ldots, x_n)$, we can meaningfully talk about the *density* $d(x)$ of situations in this vicinity – i.e., the number of situations per unit volume.

In the ideal case, when all situations belong to several clearly distinct clusters, there are no examples outside the clusters – so the density outside the clusters is 0. Within each cluster, the density $d(x)$ is positive. Different clusters can be distinguished from each other because each cluster is connected. So, in this ideal case, one we know the density $d(x)$ at each point x, we can find each cluster as the connected component of the set $\{x : d(x) > 0\}$.

Clustering: a more realistic case. In practice, in addition to objects and situations which clearly belong to different clusters, there are also "weird" situations that do not fall under any meaningful clusters. For example, when we make a medical decision, we classify all the patients into a few meaningful groups – e.g., coughing and sneezing patients can be classified into patients with cold, patients with allergy, patients with flu, etc. However, there may be some exotic diseases which also cause sneezing and coughing, diseases which are not present in the current sample in sufficient numbers.

Such not-easy-to-classify examples can occur for every x. Let d_a be the average density of such examples. In this case, if at some point, the observed density $d(x)$ is smaller than or equal to d_a, then most probably all examples with these parameters are not-easy-to-classify, so they do not belong to any of the clusters that we are trying to form. On the other hand, if for some point x, the observed density $d(x)$ is much larger than d_a, this means that all these examples cannot come from not-easy-to-classify cases: some of these example come from one of the clusters that we are trying to form.

This idea leads us to the following clustering algorithm: we select a threshold t, and then find each cluster as the connected component of the set $\{x : d(x) \ge t\}$.

How to estimate the density $d(x)$. In practice, we only have a finite set of examples $x^{(1)}$, $x^{(2)}$, …, $x^{(N)}$. In order to apply the above approach, we must use the observed values $x^{(j)}$ to estimate the density $d(x)$ at different values x.

One possible answer to this question comes from the fact that usually, dues to inevitable measurement inaccuracy, the measured values $x^{(j)} = (x_1^{(j)}, \ldots, x_n^{(j)})$ are not exactly equal to the actual (unknown) values $x_1^{(j),\mathrm{act}}, \ldots, x_n^{(j),\mathrm{act}}$ of the corresponding quantities; see, e.g., [7]. If we know the probability density function $\rho(\Delta x)$ describing the measurement errors $\Delta x \stackrel{\mathrm{def}}{=} x - x^{\mathrm{act}}$, then for each j, we know the probability density of the corresponding actual values: $\rho(x^{(j)} - x^{(j),\mathrm{act}})$.

So, if we only have one observation $x^{(1)}$, it is reasonable to estimate the density of different situations x as $d(x) = \rho(x^{(1)} - x)$. When we have N observations $x^{(1)}$, $x^{(2)}$, …, $x^{(N)}$, it is reasonable to consider them all equally probable, i.e., to assume that each of these observations occurs with probability $p(x^{(j)}) = 1/N$. Thus, due to the formula of the full probability, the probability $d(x)$ of having the actual situation x can be computed as

$$d(x) = p(x^{(1)}) \cdot \rho(x^{(1)} - x) + \ldots + p(x^{(N)}) \cdot \rho(x^{(N)} - x) = \frac{1}{N} \cdot \sum_{j=1}^{N} \rho(x^{(j)} - x).$$

The above formula is known as the *Parzen window*; see, e.g., [8]. The corresponding function $\rho(x)$ is known as a *kernel.* As a result, we arrive at the following algorithm.

Resulting clustering algorithm. At first, we select a function $\rho(x)$. Then, based on the observed examples $x^{(1)}$, $x^{(2)}$, …, $x^{(N)}$, we form a density function $d(x) = (1/N) \cdot \sum\limits_{j=1}^{N} \rho(x^{(j)} - x)$. After that, we select a threshold t, and we find the clusters as the connected components of the set $\{x : d(x) \ge t\}$.

Algorithmic comment. In practice, we can only handle a finite number of possible points x, so we perform computations only for finitely many x – e.g., for all points x from a dense grid. In the discrete case, the subdivision into connected components is equivalent to finding a transition closure of the direct neighborhood relation – and it is well known how to efficiently compute the transitive closure of a given relation; see, e.g., [1].

Discussion: beyond probabilities and measurements. In the above text, we describe a statistical motivation for this algorithm. It turns out (see, e.g., [2, 3]) that a fuzzy approach leads, in effect, to the same algorithm – in this case, instead of the probability density $\rho(x)$, we must take the membership function describing the neighborhood relation, and the Parzen window formula for $d(x)$ describes not the total probability, but rather (modulo an irrelevant $1/N$ factor) the fuzzy cardinality of the fuzzy set of all examples in the neighborhood of the given point x. This fuzzy approach makes sense, e.g., when the sample values $x^{(j)}$ come not from measurements, but from expert observations.

Empirical results. By testing different possible selections, we found out [2, 3] that empirically:

- The best kernel is the Gaussian function $\rho(x) \sim \exp(-\text{const} \cdot x^2)$.
- To describe the best threshold, we must describe, for each possible threshold t, the interval formed by all the threshold values t' that lead to the same clustering of the original points $x^{(j)}$ as t. Then, we select the threshold t for which this interval is the widest – i.e., for which clustering is the most robust to the threshold selection.

1st challenge: explain the above empirical results. Our 1st challenge is to provide a theoretical explanation for these empirical results.

2nd challenge: some observations may be erroneous. In the above analysis, we assumed that all the situations that we observed and/or measured are exactly of the type in which we are interested. In other words, we assume that the only uncertainty is that the measurement values are imprecise. In reality, about some measurements, we are not sure whether the corresponding situations are of the desired type or not.

For example, when we analyze the animals that we observed in the wild, not only are our measurements not absolutely accurate, but in addition to this, in some cases, we are not sure whether we actually observed an animal or it was just a weird combination of shadows that made it look like an animal.

It is desirable to take this additional uncertainty into account during clustering.

Comment. This additional uncertainty was recently emphasized by L. Zadeh, when he promoted the idea of a *Z-number* [9], a number for which there are two types of uncertainty: an uncertainty in *value* – corresponding to the accuracy of the measuring instrument, and an uncertainty in whether we did measure anything meaningful at all – corresponding, e.g., to *reliability* of the measuring instrument.

3rd challenge: need for fuzzy clustering results. The above algorithm provides a crisp (non-fuzzy) division into clusters. In real life, in some cases, we may indeed be certain that a given pair of objects belongs to the same cluster (or to different clusters); however, in many other cases, we are not 100% sure about it. It is desirable to modify the above clustering algorithm in such a way that it reflects this uncertainty. In other words, we *fuzzy* clusters, clusters in which some situations x are assigned to different clusters with different degrees of certainty.

The existing fuzzy-techniques-based clustering algorithms provide such classification (see, e.g., [4, 6]); it is therefore reasonable to modify the above density-based fuzzy-motivated algorithm so that it will produce a similar fuzzy clustering.

4th challenge: need for hierarchical clustering. In practice, our classification is hierarchical. For example, to make a decision about how to behave when we see an animal in the forest, we first classify animals into dangerous and harmless ones. However, once we get more experience, we realize that different dangerous animals require different strategies, so we sub-classify them into subgroups with a similar behavior: snakes, bears, etc.

It is therefore desirable that our clustering algorithm have the ability to produce such a hierarchical clustering: once we subdivided the original situations into clusters, we should be able to apply the same clustering algorithm to all the situations within each cluster c and come up with relevant sub-clusters. Alas, this does not always happen in the above density-based clustering algorithm. Indeed, we select a threshold for which the corresponding interval is the widest. Thus, it is highly possible that within a cluster, we will have the same intervals – so the new sub-classification will be based on the same threshold and thus, it will return the exact same cluster. The 4th – and the last – challenge is to modify the above algorithm so that it will enable us to produce the desired hierarchical clustering.

In this paper, we propose possible solutions to all these challenges.

2 Solutions to Challenges

A Solution to the 1st Part of the 1st Challenge. We need to explain why empirically, the Gaussian membership functions – or, equivalently, the Gaussian kernels $\rho(\Delta x)$ – are empirically the best.

Case of measurements. This empirical fact is reasonably easy to explain in the case when the values $x^{(j)}$ come from measurements, and the probability density $\rho(\Delta x)$ corresponds to the probability density of the measurement errors.

In this case, the empirical success of the Gaussian kernels can be easily explained by another (better known) empirical fact: that the Gaussian distribution of the measurement error is indeed frequently occurring in practice. This new empirical fact, in its turn, has a known explanation:

- a measurement error usually consists of a large number of small independent components, and,
- according to the Central Limit theorem, the distribution of the sum of a large number of small independent components is indeed close to Gaussian (see, e.g., [8]) – the more components, the closer the resulting distribution is to Gaussian.

Case of expert estimates. As we have mentioned, the values $x^{(j)}$ often come not from measurements, but from expert estimates. In this case, the inaccuracy of these estimates is also caused by a large number of relatively small independent factors. Thus, we can also safely assume that the corresponding estimation errors are (approximately) normally distributed.

Alternative explanation. The whole idea of the above density estimates can be reformulated as follows: we start with the discrete distribution $d_N(x)$ in which we get N values $x^{(j)}$ with equal probability, and then we "smoothen" this original distribution – by taking a convolution between $d_N(x)$ and a kernel function $\rho(x)$: $d(x) = \int d_N(y) \cdot \rho(x-y)\,dy$.

In the previous text, we mentioned that the empirically best choice of the kernel function is a Gaussian function, but what we did not explicitly mention is that even after we fix the class of Gaussian functions, we still to find an appropriate parameter – the half-width of the corresponding Gaussian distribution. An appropriate selection of this parameter is important if we want to achieve a reasonable clustering:

- on the one hand, if we select a very narrow half-width, then each original point $x^{(j)}$ becomes its own cluster;
- alternative, if we select a very wide half-width, then all the density differences will be smoother out, and we will end up with a single cluster.

The choice of this half-width is usually performed empirically: we start with a small value of half-width and gradually increase it.

In principle, every time we slightly increase the half-width, we could go back to the original discrete distribution and apply the new slightly modified kernel function. However, since the kernel functions are close to each other, the resulting convolutions are also close to each other. So, it is more computationally efficient, instead of starting with the original discrete distribution, to apply a small modifying convolution to the previous convolution result.

In this approach, the resulting convolution is the result of applying a large number of minor convolutions, with modification kernel functions which change the function very slightly – i.e., which are close to the delta-function convolution with which does not change the original function at all. How can we describe the composition of such large number of convolutions?

From the mathematical viewpoint, each modification kernel function $K(x)$ can be viewed as a random variable whose probability density function proportional to $K(x)$. The delta-function kernel – that does not change anything – corresponds to the random variable which is equal to 0 with probability 1. A kernel corresponding to a small change thus corresponds to a random variable which is close to 0 – i.e., which is small.

It is well known that the probability distribution $\rho(X)$ of the sum $X = X_1 + X_2$ of two independent random variables is equal to the convolution of their probability density functions $\rho_1(X_1)$ and $\rho_2(X_2)$:

$$\rho(X) = \int \rho_1(X_1) \cdot \rho_2(X - X_1)\,dX_1.$$

Thus, applying several convolutions – corresponding to several small random variables – is equivalent to applying one convolution corresponding to the sum of these small random variables. Due to the Central Limit theorem, this sum is (almost) normally distributed. So, the corresponding probability density function is (almost) Gaussian, and the resulting convolution is (almost) the convolution with the Gaussian kernel.

A Solution to the 2nd Part of the 1st Challenge. The above clustering algorithm depends on the selection of an appropriate threshold t. It turns out that empirically, the following method of selecting this threshold works best: we select a threshold t for which the results of clustering the sample situations $x^{(j)}$ are the most robust with respect to this selection, i.e., for which the interval $\mathbf{t}(t)$ consisting of all threshold values t' that lead to the same clustering of the original situations as t is the widest.

How can we explain the empirical efficiency of this method?

Analysis of the problem. The clustering of the sample situations is based on comparing the corresponding values $d(x^{(j)})$ with the threshold t. Thus, crudely speaking, the interval $\mathbf{t}(t)$ consists of all the values t' between the two sequential values $d(x^{(j)})$.

For simplicity, let us consider 1-D case. In this case, locally, the density function is monotonic, so the consequent values of the density $d(x^{(j)})$ are, most probably, attained at the two neighboring points $x^{(j)}$. (In multi-D case, if we use the local coordinates in which the gradient of the density is one of the directions, there are also additional dimensions that do not affect the density.)

In a sample of N points, the distance Δx to the next point can be found from the condition that there should be one point on this interval. By definition of the probability density, the probability to find the point on the intervals is equal to $d(x) \cdot \Delta x$. The total number of points is N, so the average number of points on the interval is $N \cdot d(x) \cdot \Delta x$. Thus, we have $N \cdot d(x) \cdot \Delta x \approx 1$ hence $\Delta x \approx 1/(N \cdot d(x))$. When we move from the original point x to the new point $x+\Delta x$, the density changes from $d(x)$ to $d(x)+\Delta x \cdot d'(x) \approx d(x)+d'(x)/(N \cdot d(x))$. Thus, the different between the two values if the threshold that lead to different clusterings – the desired gap – is proportional to the ratio $|d'(x)|/d(x)$. In multi-D case, we similarly have $\|\nabla d(x)\|_2/d(x)$, where $\nabla d \stackrel{\text{def}}{=} \left(\dfrac{\partial d}{\partial x_1}, \ldots, \dfrac{\partial d}{\partial x_n}\right)$ is the gradient vector, and for every vector $z = (z_1, \ldots, z_n)$, $\|z\|_2 \stackrel{\text{def}}{=} \sqrt{z_1^2 + \ldots + z_n^2}$ denotes its length.

After this reformulation, the question becomes: why, as an objective function, the ratio $\|\nabla d(x)\|_2/d(x)$ works the best? To answer this question, we will consider general reasonable optimality criteria which can be formulated in terms of the density function $d(x)$ and its gradient $\nabla d(x)$.

What we need is a preference relation. We do not necessarily need a numerical objective function that would enable us to compare two points x with two different values of $d(x)$ and $\nabla d(x)$. All we need is a preference relation $(d, z) \succeq (d', z')$ allowing us to compare two pairs (d, z) consisting of a real number d and an n-dimensional vector z. The meaning of this relation is that the pair (d, z) is better than (or of the same quality as) (d', z') – as a point whose value $d(x)$ is used as a threshold.

Let us enumerate natural properties of this relation, and then see which relations satisfy all these properties.

Natural algebraic properties of the preference relation. To be able to always make a decision, we must require that for every two pairs (d, z) and (d', z'), we have either $(d, z) \succeq (d', z')$ or $(d', z') \succeq (d, z)$. In mathematical terms, this means that the relation is *linear* or *total*.

Of course, since any pair (d, z) is of the same quality as itself, we must have $(d, z) \succeq (d, z)$ for all d and z. In mathematical terms, this means that the relation is *reflexive*.

This relation must also be *transitive*: indeed, if (d, z) is better than (or of the same quality as) (d', z'), and (d', z') is better than (or of the same quality as) (d'', z''), then (d, z) should better than (or of the same quality as) (d'', z'').

Closeness of the preference relation. Let us assume that $(d_n, z_n) \to (d, z)$, $(d'_n, z'_n) \to (d', z')$, and $(d_n, z_n) \succeq (d'_n, z'_n)$ for all n.

Since all the measurements are imprecise, this implies that for any given measurement error, for sufficiently large n, the pair (d, z) is indistinguishable from (d_n, z_n): $(d, z) \approx (d_n, z_n)$. Similarly, for sufficiently large n, the pair (d', z') is indistinguishable from the pair (d'_n, z'_n): $(d', z') \approx (d'_n, z'_n)$.

Thus, no matter how accurately we perform measurements, for the pairs (d, z) and (d', z'), there are indistinguishable pairs $(d_n, z_n) \approx (d, z)$ and $(d'_n, z'_n) \approx (d', z')$ for which $(d_n, z_n) \succeq (d'_n, z'_n)$. Hence, from the practical viewpoint, we will never be able to empirically conclude, based on measurement results, that $(d, z) \not\succeq (d', z')$. So, it is reasonable to conclude that $(d, z) \succeq (d', z')$.

In mathematical terms, this means that the relation $\succeq$ is *closed* in the topological sense.

Rotation invariance. The components x_i of each situation $x = (x_1, \ldots, x_n)$ describe, e.g., spatial coordinates of some object, or components of the 3-D vector describing the velocity of this object. In all these cases, the specific numerical representation of the corresponding vector depends on the choice of the coordinate system. In most practical situations, the choice of a coordinate system is arbitrary: instead of the original system, we could select a new one which is obtained from the previous one by rotation. It is therefore reasonable to require that the preference relation not change if we simply rotate the coordinates. In other words, it is reasonable to require that if $(d, z) \succeq (d', z')$, and T is an arbitrary rotation in n-dimensional space, then $(d, T(z)) \succeq (d', T(z'))$.

First result. From closeness and rotation invariance, we can already make an important conclusion about the preference relation. Let us formulate this first result in precise terms.

Definition 1.

- *A relation $\succeq$ on a set A is called:*

- linear *(or* total*) if for every two elements $a, a' \in A$, we have $a \succeq a'$ or $a' \succeq a$.*
- reflexive *if $a \succeq a$ for all $a \in A$;*
- transitive *if $a \succeq a'$ and $a' \succeq a''$ imply that $a \succeq a''$.*

– *Let $n \geq 1$ be an integer. By a* preference relation*, we mean a linear reflexive transitive relation $\succeq$ on the set of all pairs (d, z), where d is a non-negative real number and z is an n-dimensional vector.*

– *We say that a preference relation $\succeq$ is* closed *if for every two sequences $(d_n, z_n) \to (d, z)$ and $(d'_n, z'_n) \to (d', z')$ for which $(d_n, z_n) \succeq (d'_n, z'_n)$ for all n, we have $(d, z) \succeq (d', z')$.*

– *We say that a preference relation $\succeq$ is* rotation-invariant *if for every two pairs (d, z) and (d', z') and for every rotation T in n-dimensional space, $(d, z) \succeq (d', z')$ implies that $(d, T(z)) \succeq (d', T(z'))$.*

Proposition 1. *For every closed rotation-invariant preference relation $\succeq$, whether there is a relation $(d, z) \succeq (d', z')$ between the two pairs (d, z) and (d', z') depends only on the values d and d' and on the lengths $\|z\|_2$ and $\|z'\|_2$ of the vectors z and z', i.e., if $(d, z) \succeq (d', z')$, $\|z\|_2 = \|t\|_2$, and $\|z'\|_2 = \|t'\|_2$, then $(d, t) \succeq (d', t')$.*

Proof. Let us start with notations. Let us denote $a \equiv b$ if $a \succeq b$ and $b \succeq a$. This relation is clearly symmetric. Since the original relation $\succeq$ is reflexive and transitive, the new relation is also reflexive and transitive. In mathematical terms, reflexive symmetric transitive relations are called *equivalence relations*; thus, the above relation $\equiv$ is an equivalence relation.

One can easily check that $a \succeq b$ and $b \equiv c$ imply that $a \succeq c$, and that $a \equiv b$ and $b \succeq c$ also implies $a \succeq c$.

We plan to prove that for for every number d, for every vector z, and for every rotation T, we have $(d, z) \equiv (d, T(z))$.

Let us show that if we succeed in proving this, then the proposition will be proven. Indeed, since every two vectors of equal length can be transformed into each other by an appropriate rotation, this will mean that if $(d, z) \succeq (d', z')$, $\|z\|_2 = \|t\|_2$, and $\|z'\|_2 = \|t'\|_2$, then $(d, z) \equiv (d, t)$ and $(d', z') \equiv (d', t')$. From $(d, t) \equiv (d, z)$, $(d, z) \succeq (d', z')$, and $(d', z') \equiv (d', t')$, we will now be able to conclude that $(d, t) \succeq (d', t')$, i.e., exactly what we want to conclude in Proposition 1.

So, to complete our proof, it is sufficient to prove that for every axis ℓ and for every angle φ, the property $(d, z) \equiv (d, T_{\ell,\varphi}(z))$ holds, where $T_{\ell,\varphi}$ denoted a rotation by the angle φ around the axis ℓ.

We will first prove this statement for the case when $\varphi = 2\pi/k$ for some integer $k \geq 2$, i.e., when $k \cdot \varphi = 2\pi$.

Indeed, due to linearity of the preference relation $\succeq$, we have $(d, z) \succeq (d, T_{\ell,\varphi}(z))$ or $(d, T_{\ell,\varphi}(z)) \succeq (d, z)$. Without losing generality, let us consider the first case, when $(d, z) \succeq (d, T_{\ell,\varphi}(z))$.

In this case, rotation invariance implies that $(d, T_{\ell,\varphi}(z)) \succeq (d, T_{\ell,2\varphi}(z))$, that $(d, T_{\ell,2\varphi}(z)) \succeq (d, T_{\ell,3\varphi}(z))$, $\ldots$, and that $(d, T_{\ell,(k-1)\cdot\varphi}(z)) \succeq (d, T_{\ell,k\cdot\varphi}(z)) = (d, T_{\ell,2\pi}(z)) = (d, z)$.

Transitivity, when applied to $(d, T_{\ell,\varphi}(z)) \succeq (d, T_{\ell,2\varphi}(z)) \succeq \ldots \succeq (d, z)$, implies that $(d, T_{\ell,\varphi}(z)) \succeq (d, z)$. Since we already know that $(d, z) \succeq (d, T_{\ell,\varphi}(z))$, we conclude that $(d, z) \equiv (d, T_{\ell,\varphi}(z))$. The statement is proven.

Let us now prove the desired statement $(d, z) \equiv (d, T_{\ell,\varphi}(z))$ for the case when $\varphi = 2\pi \cdot (p/q)$ for some integers p and q.

Indeed, in this case, $\varphi = p \cdot \varphi(q)$, where we denoted $\varphi(q) \stackrel{\text{def}}{=} (2\pi)/q$. We already know, from Part 3.1 of this proof, that equivalence is preserved when we rotate by the angle $\varphi(q)$, i.e., that $(d, z) \equiv (d, T_{\ell,\varphi(q)}(z))$. Similarly, $(d, T_{\ell,\varphi(q)}(z)) \equiv (d, T_{\ell,2\varphi(q)}(z))$, $\ldots$, and, finally, that $(d, T_{\ell,(p-1)\cdot\varphi(q)}(z)) \equiv (d, T_{\ell,p\cdot\varphi(q)}(z)) = (d, T_{\ell,\varphi}(z))$. Thus, by transitivity of the equivalence relation, we conclude that indeed $(d, z) \equiv (d, T_{\ell,\varphi}(z))$. The statement is proven.

Let us now prove the desired statement $(d, z) \equiv (d, T_{\ell,\varphi}(z))$ for an arbitrary angle φ.

Indeed, every real number can be represented as a limit of rational numbers – e.g., its approximations of higher and higher accuracy. By applying this statement to the ratio $\varphi/(2\pi)$, we conclude that an arbitrary angle φ can be represented as a limit of the angles φ_n each of which has a form 2π times a rational number. For such angles, in Part 3.2 of our proof, we already proved that $(d, z) \succeq (d, T_{\ell,\varphi_n}(z))$ and $(d, T_{\ell,\varphi_n}(z)) \succeq (d, z)$. Due to closeness of the preference relation, we can now conclude that in the limit $\varphi_n \to \varphi$, we also have $(d, z) \succeq (d, T_{\ell,\varphi}(z))$ and $(d, T_{\ell,\varphi}(z)) \succeq (d, z)$, thus $(d, z) \equiv (d, T_{\ell,\varphi}(z))$.

The statement is proven, and so is the proposition.

Discussion. Based on Proposition 1, when we describe a preference relation, it is not necessarily to consider pairs consisting of a real number $d \geq 0$ and a vector z. Instead, it is sufficient to only consider two non-negative numbers: d and the length $l \stackrel{\text{def}}{=} \|z\|_2$ of the vector z. So now, we have a preference relation defined on the set of pairs of non-negative numbers d and l.

Monotonicity. If we have a homogeneous zone, i.e., a zone in which the density is constant and its gradient is 0, then this whole zone should belong to the same cluster. Selecting a threshold corresponding to this zone would mean cutting through this zone, which contradicts to the idea of clustering as bringing similar situations into the same cluster. From this viewpoint, it makes sense to dismiss pairs (d, l) for which $l = 0$: the optimal cut should never be at such pairs.

Similarly, points x with small gradient are probably not the best placed to cut. In other words, everything else being equal, situations with higher gradient (i.e., with larger values of l) are preferable as points used to determine a threshold.

With respect to density, as we have mentioned, the higher the density, the more probable it is that the corresponding values belong to the same cluster. Thus, everything else being equal, situations with lower density (i.e., with smaller values of d) are preferable to cut. From this viewpoint, it makes sense to dismiss pairs (d, l) for which $d = 0$: if such ideal pairs are present, we have an ideal (no-noise) clustering (as we mentioned in the beginning of this paper), so there is no need for all these sophisticated methods. Thus, we arrive at the following definitions.

Definition 2.

- *By a* non-zero preference relation*, we mean a linear reflexive transitive relation $\succeq$ on the set of all pairs (d, l) of positive real numbers.*
- *We say that a non-zero preference relation is* monotonic *if the following two conditions hold:*
  - *for every d and for every $l < l'$, $(d, l') \succeq (d, l)$ and $(d, l) \not\succeq (d, l')$;*
  - *for every l and for every $d < d'$, $(d, l) \succeq (d', l)$ and $(d', l) \not\succeq (d, l)$.*

Comment. The notion of closeness can be easily extended to this new definition.

Sub-samples. Instead of considering all possible situations, we may want to consider only part of them – this often happens in data processing when we want to decrease computation time. Of course, we need to select sub-populations in such a way that within each cluster, the relative density does not change. However, it is OK to select different fractions of sample in different clusters. For example, if some cluster contains a large number of different situations, it makes sense to select only some of them, while for another cluster which consists of a few situations, we cannot drastically decrease this number since otherwise, we will not have enough remaining elements to make statistically meaningful estimates (e.g., estimates of the probability density $d(x)$).

When we select only a portion of elements at a location x, the density $d(x)$ in the vicinity of this location is multiplied by the ratio λ of the following two proportions: the proportion of this cluster in the original sample, and the proportion of this cluster in the new sample. Since the values of x_j do not change, the gradient z – and hence, its length l – is also multiplied by the same constant λ. In the vicinity of another cluster, the corresponding values of d' and l' are similarly multiplied by a different constant λ'. It is reasonable to require that the relative quality of different possible thresholds does not change under this transition to a sub-sample. Thus, we arrive at the following definition.

Definition 3. *We say that a non-zero preference relation $\succeq$ is* sub-sampling invariant *if for every two pairs (d, l) and (d', l') and for every two positive real numbers $\lambda > 0$ and $\lambda' > 0$, $(d, l) \succeq (d', l')$ implies that*

$$(\lambda \cdot d, \lambda \cdot l) \succeq (\lambda' \cdot d', \lambda' \cdot l').$$

Proposition 2. *For every closed monotonic sub-sampling invariant non-zero preference relation $\succeq$, $(d, l) \succeq (d', l')$ if and only if $l/d \geq l'/d'$.*

Discussion. Thus, as a threshold, we should select a value of the density $d(x)$ corresponding to the point where the ratio $\dfrac{\|\nabla d\|}{d}$ attains its largest possible value. This result explains the above empirical rule.

Proof of Proposition 2. Since a preference relation is reflexive, we have $(d,l) \succeq (d,l)$ for every d and l. If we apply invariance with respect to sub-sampling with $\lambda = 1$ and $\lambda' = 1/d$, we get $(d,l) \succeq (1, l/d)$. If we apply invariance with respect to sub-sampling with $\lambda = 1/d$ and $\lambda' = 1$, we get $(1, l/d) \succeq (d,l)$. Thus, $(d,l) \equiv (1, l/d)$. Similarly, $(d', l') \equiv (1, l'/d')$. Thus, $(d,l) \succeq (d', l')$ if and only if $(1, l/d) \succeq (1, l'/d')$. For pairs $(1,l)$, due to monotonicity, $(1,l) \succeq (1,l')$ if and only if $l \geq l'$. Thus, indeed, $(d,l) \succeq (d',l')$ if and only if $l/d \geq l'/d'$. The proposition is proven.

A Solution to the 2nd Challenge. In the above algorithm, we implicitly assume that, while there is some inaccuracy in the measurement results corresponding to each observed situation $x^{(j)}$, each measurement indeed represents the situation of the type in which we are interested. In practice, we are not always sure that what we measured in necessarily one of such situations.

A natural way to describe this uncertainty is to assign, to each observed situation j, a probability p_j (most probably, subjective probability) that this situation is indeed of the desired type.

It is desirable to take these probabilities into account during clustering.

Idea. The main algorithm is based on the Parzen formula

$$d(x) = p(x^{(1)}) \cdot \rho(x^{(1)} - x) + \ldots + p(x^{(N)}) \cdot \rho(x^{(N)} - x) = \frac{1}{N} \cdot \sum_{j=1}^{N} \rho(x^{(j)} - x).$$

In deriving formula, we assumed that all observations $x^{(j)}$ are equally probable, i.e., that they have the same probability $p(x^{(j)})$ to be observed. Now that we know the probability p_j that each observation is real, these observations are not equally probable: the probability $p(x^{(j)})$ of the j-th observation is proportional to p_j: $p(x^{(j)}) = k \cdot p_j$ for some constant k.

This constant can be found from the condition that the overall probability is 1, i.e., that $\sum\limits_{j=1}^{N} p(x^{(j)}) = k \cdot \sum\limits_{j=1}^{N} p_j = 1$. Thus, we get $k = \dfrac{1}{\sum\limits_{j=1}^{N} p_j}$, and instead of the original Parzen formula, we get a new formula:

$$d(x) = p(x^{(1)}) \cdot \rho(x^{(1)} - x) + \ldots + p(x^{(N)}) \cdot \rho(x^{(N)} - x) = k \cdot d_0(x),$$

where $d_0(x) \stackrel{\text{def}}{=} \sum\limits_{j=1}^{N} p_j \cdot \rho(x^{(j)} - x)$.

Comment. Clusters are then determined based on the set of all the situations x that satisfy the inequality $d(x) \geq t$ for some threshold t. Since $d(x) = k \cdot d_0(x)$, this inequality is equivalent to the inequality $d_0(t) \geq t_0$, where $t_0 \stackrel{\text{def}}{=} \dfrac{t}{k}$.

Thus, instead of considering the actual density $d(x)$ and selecting an appropriate threshold t, we could as well consider a simpler function $d_0(x)$ and select an appropriate threshold t_0 for this simpler function. As a result, we arrive at the following modification of the above algorithm:

Resulting algorithm. Based on the observations $x^{(j)}$ and on the probabilities p_j, we form an auxiliary function $d_0(x) \stackrel{\text{def}}{=} \sum\limits_{j=1}^{N} p_j \cdot \rho(x^{(j)} - x)$. Then, we select an appropriate threshold t_0 and find clusters as connected components of the set $\{x : d(x) \geq t_0\}$.

Comment. For selecting t_0 we can use the same algorithm as before since, as one can easily see, this algorithm does not change if we simply multiply all the values of $d(x)$ by the same constant $(1/k)$.

A Solution to the 3rd Challenge. In the above algorithm, we assign each situation to a definite cluster; crudely speaking, to the cluster which is most probable to contain this situation. Because of the probabilistic character of the assignment procedure, the resulting "most probable" assignment is not necessarily always the correct one – it is just the assignment which is correct in more cases than other possible assignments.

In reality, it is quite possible that each cluster also contains situations which were not assigned to it – and vice versa, that some situations that were assigned to this cluster actually belong to a different cluster. It is therefore desirable to estimate, for each current cluster c and for each situation x which is currently outside this cluster, the degree to which it is possible that x actually belongs to c.

An idea on how to solve this challenge. In the above algorithm, the clusters were built based on the choice of a threshold t: each cluster c is a connected component of the set $\{x : d(x) \geq t\}$, where $d(x)$ is a probability density function based on the observed situations $x^{(j)}$.

If a situation x does not belong to the given cluster, this means that x cannot be connected to elements of c by points y for which $d(y) \geq t$. In other words, whatever connection we make between the point x and a point x_c from c (e.g., a curve connecting x and x_c), there will be a point y on this connection at which $d(y) < t$.

If for some situation x, there is a connection at which all these intermediate values $d(y)$ are close to t – e.g., exceed $t - \varepsilon$ for some small $\varepsilon > 0$ – this means that the corresponding situation y "almost" belongs to the cluster: it would belong to the cluster if we changed the threshold a little bit. In this case, if we assign degree of confidence 1 to situations originally assigned to the cluster c, it makes sense to assign a degree close to 1 to this situation c. For example, we can simply take the ratio $\dfrac{t - \varepsilon}{t}$ as the desired degree of confidence that the situation x belongs to the cluster c.

On the other hand, if no matter how we connect x with some $x_c \in c$, we have to go through some points with very low probability density – e.g., density 0 – this means that no matter how much we decrease the threshold, this situation x will not end up in the cluster c. To such situations x, we should assign low degree of confidence that this situation x belongs to the given cluster c.

Thus, we arrive at the following natural definition.

Resulting definition. For each situation x and for each cluster c, we estimate the degree $d_c(x)$ to which x belongs to c as the ratio $d_c(x) = \dfrac{t_c(x)}{t}$, where t is the threshold used for the original clustering, and $t_c(x)$ is the largest value $s \le t$ for which both the situation x and the original cluster c belong to the same connected component of the set $\{y : d(y) \ge s\}$.

Discussion. For elements x that were originally assigned to the cluster c, the degree $d_c(x)$ as defined above is equal to 1.

For elements x that can be connected to c by situations y for which $d(y) \ge t - \varepsilon$, the above-defined degree $d_c(x)$ is larger than or equal to $\dfrac{t-\varepsilon}{t}$.

Finally, if we have a situation x for which, no matter how we connect it to c, there will always be situations y on this connection for which $d(y) = 0$, then the above-defined degree $d_c(x)$ is equal to 0.

An Alternative Solution to the 3rd Challenge. In the above approach, all the situations x which have been originally assigned to a cluster c are automatically assigned degree $d_c(x) = 1$. An alternative approach is to assign different degrees $d_c(x)$ to different such situations x.

To assign such degrees, we can use the same idea that we used when we assigned degrees $d_c(x)$ to situations x which are *outside* the original cluster c. Namely, the original assignment of a situation x to different clusters is based on the value $d(x)$: situations with $d(x) \ge t$ were assigned to different clusters, while situations with $d(x) < t$ were not assigned to any clusters. If $d(x) = t$, then a minor change in $d(x)$ can move this situation outside the clusters. On the other hand, if $d(x) \gg t$, this means that even after a reasonable change in the value of $d(x)$, the situation x will still be assigned to a cluster. Thus, the larger the value $d(x)$, the larger our confidence that the situation x will be assigned to the cluster. It is therefore reasonable to take $d(x)$ as a degree of confidence that the situation x belongs to the cluster c.

Of course, this value needs to be normalized so that the largest degree will be 1. Thus, we arrive at the following alternative definition.

Alternative definition. For each situation x and for each cluster c, we estimate the degree $d_c(x)$ to which x belongs to c as follows:

If $d(x) \ge t$, then, as $d_c(x)$, we take the ratio $\dfrac{d(x)}{d_{\max}}$, where $d_{\max} \stackrel{\text{def}}{=} \sup\limits_y d(y)$ is the largest possible value of the density $d(x)$.

If $d(x) < t$, then, as the desired degree, we take the ratio $d_c(x) = \dfrac{t_c(x)}{d_{\max}}$, where $t_c(x)$ is the largest value $s \le t$ for which both the situation x and the original cluster c belong to the same connected component of the set $\{y : d(y) \ge s\}$.

A Solution to the 4th Challenge. Once the original clusters are established, then, for each cluster c, it is desirable to be able to apply the clustering algorithm only to the situations from this cluster – and come up with sub-clusters of the cluster c.

A possible solution is to use the *fuzzy clusters*, i.e., to produce the degrees of belonging (that we produced as a solution to the 3rd challenge), and then to use these degrees when clustering all the situations from the cluster c – as we did in our solution to the second challenge.

Because of the degree of belonging, the resulting density function is different from what we had based on the original sample. As a result, hopefully, we will not simply produce the original cluster (as in the original algorithm) – we will divide this cluster into reasonable sub-clusters.

Implementations. Most of the above solution have been implemented and applied to real-life problems [2, 3]; the resulting clustering is indeed closer to the expert-generated clustering than the clustering performed by the usual fuzzy clustering algorithms.

Acknowledgments. This work was supported in part by NSF grants HRD-0734825 and HRD-1242122, by Grants 1 T36 GM078000-01 and 1R43TR000173-01 from NIH, by a grant N62909-12-1-7039 from ONR, and by a grant 111T273 from TUBITAK. This work was partially performed when Gözde Ulutagay was visiting the University of Texas at El Paso.

The authors are greatly thankful to Professor Zadeh for inspiring discussions and to the anonymous referees for valuable suggestions.

References

1. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stain, *Introduction to Algorithms*, MIT Press, Cambridge, Massachusetts, 2009.
2. E. N. Nasibov and G. Ulutagay, "A new unsupervised approach for fuzzy clustering", *Fuzzy Sets and Systems*, 2007, Vol. 158, pp. 2118–2133.
3. E. N. Nasibov and G. Ulutagay, "Robustness of density-based clustering methods with various neighborhood relations", *Fuzzy Sets and Systems*, 2009, Vol. 160, pp. 3601–3615.
4. N. R. Pal and J. C. Bezdek, "On the cluster validity for the fuzzy c-means model", *IEEE Transactions on Fuzzy Systems*, 1995, Vol. 3, No. 3, pp. 370–379.
5. E. Parzen, "On the estimation of a probability density function and the mode", *Annals of Mathematical Statistics*, 1962, Vol. 33, pp. 1065–1076.
6. W. Pedrycz and F. Gomide, *An Introduction to Fuzzy Sets*, MIT Press, Cambridge, Massachusetts, 1998.
7. S. Rabinovich, Measurement Errors and Uncertainties: Theory and Practice, Springer Verlag, New York, 2005.
8. D. J. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, Chapman & Hall/CRC, Boca Raton, Florida, 2007.
9. L. A. Zadeh, "A Note on Z-numbers", *Information Sciences*, 2011, Vol. 181, pp. 2923–2932.