Why u^m and $u \cdot \log(u)$ Are the Most Effective Nonlinear Functions in Fuzzy Clustering: Theoretical Explanation of the Empirical Fact

Olga Kosheleva[1], Vladik Kreinovich[2], and Yuchi Kanzawa[3]

[1] Department of Teacher Education, University of Texas at El Paso, 500 W. University, El Paso, Texas 79968, USA, olgak@utep.edu, https://www.cs.utep.edu/vladik/olgavita.html

[2] Department of Computer Science, University of Texas at El Paso, 500 W. University, El Paso, Texas 79968, USA, vladik@utep.edu, https://www.cs.utep.edu/vladik/

[3] School of Engineering, Shibaura Institute of Technology, 3-7-5 Toyosu, Koto, Tokyo 1358548, Japan, kanzawa@shibaura-it.ac.jp

Abstract. In fuzzy clustering, we need to have non-linear functions of the membership degrees. Different nonlinear functions have been tried. Empirical evidence shows that for fuzzy clustering, the most effective nonlinear functions are u^m and $u \cdot \log(u)$. In this paper, we provide a theoretical explanation for this empirical fact.

Keywords: fuzzy clustering, normalization invariance, theoretical explanation

1 Formulation of the Problem

Why clustering: a brief reminder. Often, we feel that the objects form several clusters: e.g., pets can be divided into dogs and cats, people are divided by race, by ethnicity, etc. – so that with respect to some characteristics, objects from the same cluster are closer to each other than objects from different clusters; see, e.g., [2].

Division into clusters often helps. For example, in medicine, since objects from the same cluster are similar to each other, it is natural to expect that the same medicine and/or the same medical procedure can help all the people from this cluster. Thus, instead of looking for individual cure for each patient, it is sufficient to find cure for each cluster. From this viewpoint, it is desirable to have an automatic procedure for dividing objects into clusters.

k-means clustering: a brief reminder. Once we have found typical elements $t_1, \ldots, t_c$ in each cluster, a natural way to assign objects to clusters is to assign, to each object x, the cluster i for which x is the closest to the corresponding typical element, i.e., for which the distance $d(x, t_i)$ is the smallest possible. This way, the distance $d(x) \stackrel{\text{def}}{=} d(x, t_i)$ from the object x to the typical object t_i from the resulting cluster is equal to

$$d(x) = \min_i d(x, t_i).$$

By definition of a cluster, all objects within the same cluster should be close to each other. Thus, all the value $d(x)$ should be small – and the smaller all these values, the more we believe that our division into clusters is correct. In other words, we want to have $d(x) \approx 0$ for all objects x. This means that we want the tuple $(d(x), \ldots)$ formed by the values $d(x)$ to be as close to the tuple $(0, \ldots, 0)$ as possible.

A natural way to measure the distance d between the two tuples is to interpret each tuple as a point in the corresponding multi-D space, and to use the usual Euclidean formula for the distance between the two points in this space. For our tuples, this means

$$d = \sqrt{\sum_x d^2(x)}.$$

The usual way to minimize an expression is to differentiate it and equate the derivative to 0. This is difficult to do for the above expression, since the square root function $\sqrt{v}$ is not differentiable for $v = 0$. To avoid this problem, we can use the fact that minimizing the distance d is equivalent to minimizing its square, so we minimize the sum

$$d^2 = \sum_x d^2(x).$$

Each $d(x)$ is the smallest of the values $d(x, t_i)$, so we can conclude that

$$d^2 = \sum_x \min_i d^2(x, t_i).$$

Our goal is then to find the tuples $t_1, \ldots, t_c$ for which the expression d^2 is the smallest possible. The resulting clustering method is known as *k-means*.

To optimize d^2, we can use the following iterative procedure. First, we randomly select the typical elements $t_1, \ldots, t_n$. On each iteration, we start with some tuple of typical elements t_i. Then:

- First, we assign, to each object x, a cluster i for which the distance $d(x, t_i)$ is the smallest.
- Then, for each i, we re-calculate the typical element t_i by minimizing the sum $\sum d^2(x, t_i)$, where the sum is taken overall all the objects x that are, at this moment, assigned to cluster i. In particular, if $d(x, t_i)$ is the usual Euclidean distance, then the minimizing elements t_i are simply the mean values of all the objects x from the cluster.

On each iteration, the sum d^2 decreases or stays the same, so hopefully, the process will converge, We stop when on some iteration, typical elements do not change, i.e., in precise terms, when the distance $d(t_i, t'_i)$ between the values t_i before and this iteration is smaller than some fixed small threshold $\varepsilon > 0$.

Need for fuzzy clustering. K-means assigns each object to a cluster. In practice, however, we can have objects that belongs – to some extent – to several clusters. For example, we can have people of mixed racial and/or ethnic origin. It is therefore desirable, instead of assigning an object x to a single cluster, to assign, for each object x and for each cluster i, a degree u_{xi} to which the object x belongs to the cluster i. These degrees split the object between several clusters, so it is reasonable to require that for each object x, the corresponding degrees add up to 1:

$$\sum_i u_{xi} = 1.$$

Resulting fuzzy clustering method: need for a nonlinear function. In this scheme, we need, given the objects, to define the corresponding values u_{xi}. We thus need to generalize the above objective function d^2 to the case when we have degrees d_{xi}. At first glance, this looks easy to do: the classical situation corresponds to the case when $u_{xi} = 1$ for the cluster i containing the object x, and $u_{xj} = 0$ for all other clusters j. In these terms, the expression

$$\min_i d^2(x, t_i)$$

takes the form

$$\sum_i u_{xi} \cdot d^2(x, t_i),$$

and the objective function takes the following form:

$$\sum_x \sum_i u_{xi} \cdot d^2(x, t_i).$$

We need to minimizing this expression under the condition that $u_{xi} \geq 0$ and

$$\sum_i u_{xi} = 1.$$

The problem with this approach is that in this problem we optimize a linear combination of the unknowns u_{xi} under a linear constraint. Such problems are known as *linear programming* problems, and it is known that for each such problem, one of the optimal solutions is located at one of the vertices of the domain over which we optimize; see, e.g., [3]. The vertices correspond to the case when as many inequalities as possible become equalities. In this case, we have $n-1$ inequalities $u_{xi} \geq 0$ become equalities, i.e., we have a tuple u_{xi} in which $n-1$ degrees are 0s and the remaining degree is equal to 1. Thus, we will always get a crisp solution.

So, to get truly fuzzy solutions, we cannot use the objective function whose dependence on u_{xi} is linear, we must have a nonlinear dependence.

Fuzzy clustering: resulting general formulas. We can get a desired nonlinear expression if we replace u_{xi} in our objective function with a nonlinear functions of u_{xi}. In general, replacing u_{xi} with a nonlinear function $A(u_{xi})$ of

u_{xi} and/or add a term $B(u_{xi})$ non-linear in u_{xi} to the objective function, we get the following objective function:

$$\sum_{x,i}(A(u_{xi})\cdot J_{xi}+B(u_{xi})),$$

where we denoted $J_{xi}\stackrel{\text{def}}{=}d^2(x,t_i)$.

Empirical fact. Different nonlinear functions $A(u)$ and $B(u)$ have been tried. It turns out that in the most effective pairs, we have $A(u)=u^m$ for some m and $B(u)$ has either the same form or the form $u\cdot\log(u)$. In this paper, we provide a theoretical explanation for this empirical fact.

2 Our Explanation

Main idea: normalization-invariance. Usually, we consider normalized fuzzy sets, i.e., fuzzy sets for which at some point, membership is equal to 1. Usually, the membership degree $m(x)$ increases up to a certain value x_0 at which it is equal to 1, and then it start decreasing again.

What if our prior knowledge is that the value x has a property described by the corresponding membership function – e.g., that x is small, in which case $x_0=0$. Suppose now that we have received an additional information that x is larger than or equal to some threshold $t>x_0$. What will then be the resulting knowledge about x? We know that x is small *and* that $x\ge t$, so the resulting fuzzy set is an intersection of fuzzy sets corresponding to small and to $x\ge t$. For values $x<t$, the membership function $m_\cap(x)$ of this intersection is 0, and for value $x\ge t$, it is smaller than or equal to $m(t)<1$. This membership function never reaches the value 1 and is, thus, not normalized – which is a typical situation for intersections.

Many algorithm for processing fuzzy information assume that the membership functions are normalized. So, to be able to apply these algorithms to the intersection membership function $m_\cap(x)$, we need to first transform it into a normalized one $m_{\text{norm}}(x)$. This transformation is known as *normalization.* The usual normalization means multiplying all the values of this membership by an appropriate constant, namely

$$m_{\text{norm}}(x)=\lambda\cdot m_\cap(x),\ \text{where}\ \lambda=\frac{1}{\max\limits_y m_\cap(y)}.$$

From this viewpoint, a membership function is defined modulo multiplication by a constant $m(x)\to\lambda\cdot m(x)$.

Since this normalization does not change the meaning of the membership function, it is reasonable to require that whatever conclusions we can make should not change if we simply multiply all membership degrees by a constant λ. We will call this property – a detailed description will be given later in this paper – *normalization-invariance.*

For this particular example, the constraint – that the sum of all degrees u_{xi} corresponding to each object x is equal to 1 – has to change, since after we multiply all the values $u_{x,t}$ by the same normalizing constant λ, the sum is also multiplied by λ. Thus, the constraint should be that all these sums are equal to some constant λ. So, we arrive at the following definitions.

Definition 1.

- *By a* fuzzy clustering method*, we mean a pair of measurable functions* $(A(u), B(u))$*.*
- *We say that a clustering method* $(A(u), B(u))$ *is* non-trivial *if* $A(u)$ *is not a constant function.*
- *We say that two clustering methods* $(A(u), B(u))$ *and* $(A'(u), B'(u))$ *are* equivalent *if* $A'(u) = c \cdot A(u)$ *and* $B'(u) = c \cdot B(u) + c_0 + c_1 \cdot u$ *for some coefficients* $c > 0$*,* c_0 *and* c_1*.*
- *We say that a fuzzy clustering method is* normalization-invariant *if for each* $\lambda > 0$ *and for every four sequences of values* u_{xi}*,* J_{xi}*,* u'_{xi}*,* J'_{xi} *for which*

$$\sum_i u_{xi} = \sum_i u'_{xi} \quad for \quad all \quad x,$$

$$if \quad \sum_{x,i} (A(u_{xi}) \cdot J_{xi} + B(u_{xi})) \geq \sum_{x,i} (A(u'_{xi}) \cdot J'_{xi} + B(u_{xi})),$$

$$then \quad \sum_{x,i} (A(\lambda \cdot u_{xi}) \cdot J_{xi} + B(\lambda \cdot u_{xi})) \geq \sum_{x,i} (A(\lambda \cdot u'_{xi}) \cdot J'_{xi} + B(\lambda \cdot u_{xi})).$$

Comment. The term "non-trivial" comes from the fact that if $A(u)$ is a constant function, then, since the term depending on the degrees u_{xi} does not depend on the distances J_{xi} – and thus, on the input data, the resulting "optimal" degrees u_{xi} do not depend on the data at all – while we want clustering determined by the data.

The term "equivalent" is explained by the following simple result:

Proposition 1. *When two fuzzy clustering methods* $(A(u), B(u))$ *and* $(A'(u), B'(u))$ *are equivalent, then for every four sequences of values* u_{xi}*,* J_{xi}*,* u'_{xi}*,* J'_{xi} *for which*

$$\sum_i u_{xi} = \sum_i u'_{xi} \quad for \quad all \quad x,$$

$$if \quad \sum_{x,i} (A(u_{xi}) \cdot J_{xi} + B(u_{xi})) \geq \sum_{x,i} (A(u'_{xi}) \cdot J'_{xi} + B(u_{xi})),$$

$$then \quad \sum_{x,i} (A'(u_{xi}) \cdot J_{xi} + B'(u_{xi})) \geq \sum_{x,i} (A'(u'_{xi}) \cdot J'_{xi} + B'(u_{xi})).$$

Proof of Proposition 1. We can get an inequality corresponding to the c-based change in the function $A(u)$ if we multiply both sides of the original inequality by c. Terms proportional to c_0 simply lead to the same constant on both sides – so by subtracting this constant, we get an equivalent inequality. Terms proportional to c_1 cancel each other, since they are proportional to the sum of all the degrees u_{xi}, and the condition is that the sum of the values u_{xi} is the same as the sum of all the values u'_{xi}. The proposition is proven.

Proposition 2. *A non-trivial fuzzy clustering method is normalization-invariant if and only it is equivalent to the following method: $A(u) = u^m$ for some m, and the function $B(u)$ has the following form:*

- *when $m \neq 1$, we have $B(u) = c_m \cdot u^m$ for some c_m;*
- *when $m = 1$, we have $B(u) = c \cdot u \cdot \log(u)$ for some c.*

Proof of Proposition 2. It is easy to show that both above examples are normalization-invariant. So, to prove the proposition, it is sufficient to prove that every normalization-invariant fuzzy clustering has the desired form. To prove this, let us consider the case when we have only one object x and two clusters $i = 1$ and $i = 2$. In this case, we will skip the subscript x and write u_i, J_i and u'_i instead of u_{xi}, J_{xi}, and u'_{xi}. Then, the above constraint takes the form $u_1 + u_2 = u'_1 + u'_2$. Let us denote the sum $u_1 + u_2 = u'_1 + u'_2$ by u. Then, $u_2 = u - u_1$ and $u'_2 = u - u'_1$. In this case and in these notations, normalization invariance takes the following form: if

$$A(u_1) \cdot J_1 + A(u - u_1) \cdot J_2 + B(u_1) + B(u - u_1) \geq$$

$$A(u'_1) \cdot J'_1 + A(u - u'_1) \cdot J'_2 + B(u'_1) + B(u - u'_1), \tag{1}$$

then

$$A(\lambda \cdot u_1) \cdot J_1 + A(\lambda \cdot (u - u_1)) \cdot J_2 + B(\lambda \cdot u_1) + B(\lambda \cdot (u - u_1)) \geq$$

$$A(\lambda \cdot u'_1) \cdot J'_1 + A(\lambda \cdot (u - u'_1)) \cdot J'_2 + B(\lambda \cdot u'_1) + B(\lambda \cdot (u - u'_1)). \tag{2}$$

Since $a = b$ means $a \geq b$ and $b \geq a$, if we have equality in equation (1), then we have both inequality (1) and the opposite inequality. From normalization invariance, we can now conclude that we have both inequality (2) and the opposite inequality. Thus, we have equality in (2). In other words, if:

$$A(u_1) \cdot J_1 + A(u - u_1) \cdot J_2 + B(u_1) + B(u - u_1) =$$

$$A(u'_1) \cdot J'_1 + A(u - u'_1) \cdot J'_2 + B(u'_1) + B(u - u'_1), \tag{3}$$

then

$$A(\lambda \cdot u_1) \cdot J_1 + A(\lambda \cdot (u - u_1)) \cdot J_2 + B(\lambda \cdot u_1) + B(\lambda \cdot (u - u_1)) =$$

$$A(\lambda \cdot u'_1) \cdot J'_1 + A(\lambda \cdot (u - u'_1)) \cdot J'_2 + B(\lambda \cdot u'_1) + B(\lambda \cdot (u - u'_1)). \tag{4}$$

By moving all the terms containing J_i and J'_i into the left-hand side and other terms into the right-hand side, we conclude that if

$$A(u_1) \cdot J_1 + A(u - u_1) \cdot J_2 - A(u'_1) \cdot J'_1 - A(u - u'_1) \cdot J'_2 =$$

$$B(u_1') + B(u - u_1') - B(u_1) - B(u - u_1), \tag{5}$$

then

$$A(\lambda \cdot u_1) \cdot J_1 + A(\lambda \cdot (u - u_1)) \cdot J_2 - A(\lambda \cdot u_1') \cdot J_1' - A(\lambda \cdot (u - u_1')) \cdot J_2' =$$

$$B(\lambda \cdot u_1') + B(\lambda \cdot (u - u_1')) - B(\lambda \cdot u_1) - B(\lambda \cdot (u - u_1)). \tag{6}$$

Let J_i and J_i' satisfy the equation (5). Then, if we have a set of values ΔJ_i and $\Delta J_i'$ for which

$$A(u_1) \cdot \Delta J_1 + A(u - u_1) \cdot \Delta J_2 - A(u_1') \cdot \Delta J_1' - A(u - u_1') \cdot \Delta J_2' = 0, \tag{7}$$

then for $\widetilde{J}_i \stackrel{\text{def}}{=} J_i + \Delta J_i$ and $\widetilde{J}_i' \stackrel{\text{def}}{=} J_i' + \Delta J_i'$, by adding (5) and (7), we get

$$A(u_1) \cdot \widetilde{J}_1 + A(u - u_1) \cdot \widetilde{J}_2 - A(u_1') \cdot \widetilde{J}_1' - A(u - u_1') \cdot \widetilde{J}_2' =$$

$$B(u_1') + B(u - u_1') - B(u_1) - B(u - u_1). \tag{8}$$

By normalization invariance to (8), we have

$$A(\lambda \cdot u_1) \cdot \widetilde{J}_1 + A(\lambda \cdot (u - u_1)) \cdot \widetilde{J}_2 - A(\lambda \cdot u_1') \cdot \widetilde{J}_1' - A(\lambda \cdot (u - u_1')) \cdot \widetilde{J}_2' =$$

$$B(\lambda \cdot u_1') + B(\lambda \cdot (u - u_1')) - B(\lambda \cdot u_1) - B(\lambda \cdot (u - u_1)). \tag{9}$$

Subtracting (6) from (9), we conclude that

$$A(\lambda \cdot u_1) \cdot \Delta J_1 + A(\lambda \cdot (u - u_1)) \cdot \Delta J_2 - A(\lambda \cdot u_1') \cdot \Delta J_1' - A(\lambda \cdot (u - u_1')) \cdot \Delta J_2' = 0. \tag{10}$$

So, for every set of values ΔJ_i and $\Delta J_i'$, equality (7) implies equality (10). The left-hand side of each of these two equalities can be described as a dot (scalar) products of two vectors, namely as $\Delta \cdot V = 0$ and, correspondingly, as $\Delta \cdot V' = 0$, where

$$\Delta \stackrel{\text{def}}{=} (\Delta J_1, \Delta J_2, \Delta J_1', \Delta J_2'),$$

$$V \stackrel{\text{def}}{=} (A(u_1), A(u - u_1), -A(u_1'), -A(u - u_1')), \text{ and}$$

$$V' \stackrel{\text{def}}{=} (A(\lambda \cdot u_1), A(\lambda \cdot (u - u_1)), -A(\lambda \cdot u_1'), -A(\lambda \cdot (u - u_1'))).$$

Thus, every vector orthogonal to V is also orthogonal to V'.

Let us prove that this implies that the vector V' is parallel to V. Indeed, V' can be represented as the sum of two components $V' = V_\parallel' + V_\perp$, where

$$V_\parallel' \stackrel{\text{def}}{=} \frac{V' \cdot V}{|V|} \cdot V$$

(where $|V| = \sqrt{V \cdot V}$ is the length of the vector V) is the component parallel to V, and $V_\perp'$ is the component which is orthogonal to V. Since $V_\perp'$ is orthogonal to V, it should also be orthogonal to V', i.e., we should have $0 = V' \cdot V_\perp' = V_\parallel \cdot V_\perp' + V_\perp' \cdot V_\perp'$. Since $V_\parallel'$ and $V_\perp$ are orthogonal to each other, we have $V_\parallel \cdot V_\perp' = 0$, thus $V_\perp' \cdot V_\perp' = 0$, hence $V_\perp' = 0$, i.e., V' is indeed parallel to V.

Since V' is parallel to V, the vector V' can be obtained from V by multiplying it by a constant c that, in general, may depends on λ, u_1, u, and u'_1:

$$V' = c(\lambda, u_1, u, u'_1) \cdot V. \tag{11}$$

For the first components of the vectors, this means that

$$A(\lambda \cdot u_1) = c(\lambda, u_1, u, u'_1) \cdot A(u_1). \tag{12}$$

The value c is equal to the ratio $A(\lambda \cdot u_1)/A(u_1)$. This ratio does not depend on u or on u'_1, so c does not depend on them either: $c = c(\lambda, u_1)$. Similarly, for the third components of the vectors, the equality (11) takes the form

$$-A(\lambda \cdot u'_1) = -c(\lambda, u_1) \cdot A(u'_1).$$

We can conclude that $c(\lambda, u_1)$ is equal to the ratio $A(\lambda \cdot u'_1)/A(u'_1)$. This ratio does not depend on u_1, so c does not depend on u_1 either: $c = c(\lambda)$. So, the equation (12) takes the following simplified form:

$$A(\lambda \cdot u_1) = c(\lambda) \cdot A(u_1). \tag{13}$$

It is known (see, e.g., [1]) that every measurable solution of this functional equation has the form $A(u) = c_A \cdot u^m$ for constants c_A and m. This is equivalent to $A(u) = u^m$. So, we proved that the function $A(u)$ has the desired form.

Let us now prove that the function $B(u)$ also has the desired form. Indeed, substituting the above expression for $A(u)$ into the formulas (5) and (6), we conclude that if

$$c_A \cdot u_1^m \cdot J_1 + c_A \cdot (u - u_1)^m \cdot J_2 - c_A \cdot (u'_1)^m \cdot J'_1 - c_A \cdot (u - u'_1)^m \cdot J'_2 =$$
$$B(u'_1) + B(u - u'_1) - B(u_1) - B(u - u_1), \tag{14}$$

then

$$c_A \cdot (\lambda \cdot u_1)^m \cdot J_1 + c_A \cdot (\lambda \cdot (u - u_1))^m \cdot J_2 - c_A \cdot (\lambda \cdot u'_1)^m \cdot J'_1 - c_A \cdot (\lambda \cdot (u - u'_1))^m \cdot J'_2 =$$
$$B(\lambda \cdot u'_1) + B(\lambda \cdot (u - u'_1)) - B(\lambda \cdot u_1) - B(\lambda \cdot (u - u_1)). \tag{15}$$

Multiplying both sides of the equality (14) by λ^m, we get an equivalent equality

$$c_A \cdot (\lambda \cdot u_1)^m \cdot J_1 + c_A \cdot (\lambda \cdot (u - u_1))^m \cdot J_2 - c_A \cdot (\lambda \cdot u'_1)^m \cdot J'_1 - c_A \cdot (\lambda \cdot (u - u'_1))^m \cdot J'_2 =$$
$$\lambda^m \cdot B(u'_1) + \lambda^m \cdot B(u - u'_1) - \lambda^m \cdot B(u_1) - \lambda^m \cdot B(u - u_1). \tag{16}$$

Since (16) is equivalent to (14), we can conclude that if we have (16), then we have (15) as well. The left-hand sides of (15) and (16) are the same. For each right-hand side of (16), we can always find J_1 and J_2 for which (16) is true – and thus, (15) is true as well. Since the left-hand sides of (15) and (16) are equal to each other, this means that the right-hand sides of these equalities are equal to each other too. Thus, we have the following equality.

$$B(\lambda \cdot u'_1) + B(\lambda \cdot (u - u'_1)) - B(\lambda \cdot u_1) - B(\lambda \cdot (u - u_1)) =$$

$$\lambda^m \cdot B(u_1') + \lambda^m \cdot B(u - u_1') - \lambda^m \cdot B(u_1) - \lambda^m \cdot B(u - u_1). \quad (17)$$

Moving all terms to the left-hand sides and grouping similar terms together, we conclude that for

$$b(u) \stackrel{\text{def}}{=} B(\lambda \cdot u) - \lambda^m \cdot B(u), \quad (18)$$

we get

$$b(u_1) + b(u - u_1) - b(u_1') - b(u - u_1') = 0. \quad (19)$$

i.e., equivalently, that

$$b(u_1) + b(u - u_1) = b(u_1') + b(u - u_1') \quad (20)$$

for all u_1, u, and u_1'. In particular, for $u_1' = 0$, we get

$$b(u_1) + b(u - u_1) = b(0) + b(u). \quad (21)$$

If we subtract $2b(0)$ from both sides of this equality, we conclude that

$$(b(u_1) - b(0)) + (b(u - u_1) - b(0)) = b(u) - b(0). \quad (22)$$

Thus, for $t(u) \stackrel{\text{def}}{=} b(u) - b(0)$, we have

$$t(u_1) + t(u - u_1) = t(u) \quad (23)$$

for all u_1 and u. It is known (see, e.g., [1]) that every measurable solution to this functional equation is $t(u) = c_t \cdot u$ for some c_t. Thus, $b(u) = t(u) + b(0) = c_t \cdot u + c_b$, where we denoted $b(0)$ by c_b. Here, the coefficients c_t and c_b, in general, depend on λ. So, by definition of $b(u)$, we get the following equality:

$$B(\lambda \cdot u) - \lambda^m \cdot B(u) = c_t(\lambda) \cdot u + c_b(\lambda). \quad (24)$$

If we multiply both sides of the equality (24) by μ^m, we get

$$\mu^m \cdot B(\lambda \cdot u) - \mu^m \cdot \lambda^m \cdot B(u) = \mu^m \cdot c_t(\lambda) \cdot u + \mu^m \cdot c_b(\lambda). \quad (25)$$

On the other hand, if, in the formula (24), we replace λ with μ, and replace u with $\lambda \cdot u$, we get the following:

$$B(\lambda \cdot \mu \cdot u) - \mu^m \cdot B(\lambda \cdot u) = c_t(\mu) \cdot \lambda \cdot u + c_b(\mu). \quad (26)$$

If we add the equalities (25) and (26), we get the following equality:

$$B(\lambda \cdot \mu \cdot u) - \lambda^m \cdot \mu^m \cdot B(u) = c_t(\mu) \cdot \lambda \cdot u + c_b(\mu) + \mu^m \cdot c_t(\lambda) \cdot u + \mu^m \cdot c_b(\lambda). \quad (27)$$

The left-hand side of this equality does not change if we swap λ and μ, so the value of the right-hand side should also not change under this swap. In other words, the following equality must hold:

$$c_t(\mu) \cdot \lambda \cdot u + c_b(\mu) + \mu^m \cdot c_t(\lambda) \cdot u + \mu^m \cdot c_b(\lambda) =$$

$$c_t(\lambda)\cdot\mu\cdot u + c_b(\lambda) + \lambda^m\cdot c_t(\mu)\cdot u + \lambda^m\cdot c_b(\mu). \tag{28}$$

This equality of two linear functions of u must be true for all u, so for the two expressions both the free terms and the coefficients at u must be equal.

Equating the free terms, we get the following equality:

$$c_b(\mu) + \mu^m\cdot c_b(\lambda) = c_b(\lambda) + \lambda^m\cdot c_b(\mu). \tag{29}$$

By moving all the terms proportional to $c_b(\mu)$ to the left-hand side and all other terms to the right-hand side, we conclude that

$$c_b(\mu)\cdot(1-\lambda^m) = c_b(\lambda)\cdot(1-\mu^m). \tag{30}$$

For non-trivial fuzzy clustering methods, $m\ne 0$. So, we can divide both sides of the equality (30) by the product $(1-\lambda^m)\cdot(1-\mu^m)$ and get

$$\frac{c_b(\lambda)}{1-\lambda^m} = \frac{c_b(\mu)}{1-\mu^m}. \tag{31}$$

This equality holds for all possible λ and μ. This means that this expression is a constant, not depending on λ at all. Let us denote this constant by c_b. Then, we have

$$\frac{c_b(\lambda)}{1-\lambda^m} = c_b,$$

and thus,

$$c_b(\lambda) = c_b\cdot(1-\lambda^m). \tag{32}$$

By equating coefficients at u at both sides of the equality (28), we conclude that

$$c_t(\mu)\cdot\lambda + \mu^m\cdot c_t(\lambda) = c_t(\lambda)\cdot\mu + \lambda^m\cdot c_t(\mu). \tag{33}$$

By moving all the terms proportional to $c_t(\mu)$ to the left-hand side and all other terms to the right-hand side, we conclude that

$$c_t(\mu)\cdot(\lambda-\lambda^m) = c_t(\mu)\cdot(\mu-\mu^m). \tag{34}$$

Let us first consider the general case when $m\ne 1$; the case when $m=1$ will be considered separately later. In this case, we can divide both sides of the equality (34) by the product $(\lambda-\lambda^m)\cdot(\mu-\mu^m)$ and get

$$\frac{c_t(\lambda)}{\lambda-\lambda^m} = \frac{c_t(\mu)}{\mu-\mu^m}. \tag{35}$$

This equality holds for all possible λ and μ. This means that this expression is a constant, not depending on λ at all. Let us denote this constant by c_t. Then, we have

$$\frac{c_t(\lambda)}{\lambda-\lambda^m} = c_t,$$

and thus,

$$c_t(\lambda) = c_t\cdot(\lambda-\lambda^m). \tag{36}$$

Substituting the expressions (32) and (36) into the formula (24), we conclude that

$$B(\lambda \cdot u) - \lambda^m \cdot B(u) = c_t \cdot (\lambda - \lambda^m) \cdot u + c_b \cdot (1 - \lambda^m). \tag{37}$$

For $u = 1$ and $\lambda = x$, we conclude that

$$B(x) = x^m \cdot b + c_t \cdot (x - x^m) + c_b \cdot (1 - x^m), \tag{38}$$

where $b \stackrel{\text{def}}{=} B(1)$. Grouping together terms proportional to 1, x, and x^m, we conclude that

$$B(x) = c_0 + c_1 \cdot x + c_m \cdot x^m, \tag{39}$$

for $c_0 = c_b$, $c_1 = c_t$, and $c_m = b - c_b$. This is exactly the form that we wanted to derive.

To complete the prove, we need to consider the special cases $m = 1$. In this case, if we plug in the formula (32) for $c_b(\lambda)$ into the equality (24) and move the term $\lambda^m \cdot B(u) = \lambda \cdot B(u)$ to the right-hand side, we get the following equality:

$$B(\lambda \cdot u) = \lambda \cdot B(u) + c_t(\lambda) \cdot u + c_b \cdot (1 - \lambda). \tag{40}$$

If we swap λ and u, then we get the following equality in which only the right-hand side changes:

$$B(\lambda \cdot u) = u \cdot B(\lambda) + c_t(u) \cdot \lambda + c_b \cdot (1 - u). \tag{41}$$

Since both equalities have the same left-hand side, their right-hand sides should also be equal:

$$\lambda \cdot B(u) + c_t(\lambda) \cdot u + c_b - c_b \cdot \lambda = u \cdot B(\lambda) + c_t(u) \cdot \lambda + c_b - c_b \cdot u. \tag{42}$$

The terms c_b in both sides of (42) cancel each other. If move all the terms proportional to λ to the left-hand side and all other terms to the right-hand side, we get the following equality:

$$\lambda \cdot (B(u) - c_t(u) - c_b) = u \cdot (B(\lambda) - c_t(\lambda) - c_b). \tag{43}$$

If we divide both sides by $\lambda \cdot u$, we get the following:

$$\frac{B(u) - c_t(u) - c_b}{u} = \frac{B(\lambda) - c_t(\lambda) - c_b}{\lambda}. \tag{44}$$

This equality holds for all possible values λ and u. Thus, the corresponding ratio does not depend on u, this ratio is a constant. Let us denote this constant by c, then we have

$$\frac{B(u) - c_t(u) - c_b}{u} = c \text{ and } B(u) - c_t(u) - c_b = c \cdot u. \tag{45}$$

Thus, we have

$$c_t(u) = B(u) - c_b - c \cdot u. \tag{46}$$

Substituting this expression or $c_t(u)$ into the formula (40), we conclude that

$$B(\lambda \cdot u) = \lambda \cdot B(u) + B(\lambda) \cdot u - c_b \cdot u - c \cdot \lambda \cdot u + c_b - c_b \cdot \lambda. \quad (47)$$

Let us show that this equality can be simplified if instead of the function $B(u,$ we use, for some c_0 and c_1, an equivalent function

$$\widetilde{B}(u) = B(u) + c_0 + c_1 \cdot u, \quad (48)$$

for which

$$B(u) = \widetilde{B}(u) - c_0 - c_1 \cdot u. \quad (49)$$

Indeed, from (47) and (48), we conclude that

$$\widetilde{B}(\lambda \cdot u) = \lambda \cdot B(u) + B(\lambda) \cdot u - c_b \cdot u - c \cdot \lambda \cdot u + c_b - c_b \cdot \lambda + c_0 + c_1 \cdot \lambda \cdot u, \quad (50)$$

i.e., that

$$\widetilde{B}(\lambda \cdot u) = \lambda \cdot B(u) + B(\lambda) \cdot u - c_b \cdot (u + \lambda) + (c_1 - c) \cdot \lambda \cdot u + (c_0 + c_b). \quad (51)$$

Substituting the expression (49) for $B(u)$ into this equality, we conclude that

$$\widetilde{B}(\lambda \cdot u) = \lambda \cdot \widetilde{B}(u) + \widetilde{B}(\lambda) \cdot u - (c_b + c_0) \cdot (u + \lambda) - (c_1 + c) \cdot \lambda \cdot u + (c_0 + c_b). \quad (21)$$

So, for $c_0 = -c_b$ and $c_1 = -c$, we get the simplified equality

$$\widetilde{B}(\lambda \cdot u) = \lambda \cdot \widetilde{B}(u) + \widetilde{B}(\lambda) \cdot u. \quad (53)$$

If we divide both sides of this equality by $\lambda \cdot u$, we conclude that

$$\frac{\widetilde{B}(\lambda \cdot u)}{\lambda \cdot u} = \frac{\widetilde{B}(u)}{u} + \frac{\widetilde{B}(\lambda)}{\lambda}. \quad (54)$$

So, for $b(u) \stackrel{\text{def}}{=} B(u)/u$, we have $b(\lambda \cdot u) = b(\lambda) + b(u)$. According to [1], any measurable solution to this functional equation has the form $b(u) = c \cdot \log(u)$ for some constant c. Thus, we have $\widetilde{B}(u) = u \cdot b(u) = c \cdot u \cdot \log(u)$. This is exactly what we wanted to prove for $m = 1$. The proposition is proven.

Acknowledgments

This work was supported in part by the AT&T Fellowship in Information Technology, by a grant from the Hungarian National Research, Development and Innovation Office (NRDI), and by the Institute for Risk and Reliability, Leibniz Universitaet Hannover, Germany.

References

1. J. Aczél and J. Dhombres, *Functional Equations in Several Variables*, Cambridge University Press, 2008.
2. J. C. Bezdek, *Elementary Cluster Analysis: Four Basic Methods that (Usually) Work*, River Publishers, Aarhus, Denmark, 2024.
3. R. J. Vanderbei, *Linear Programming: Foundations and Extensions*, Springer, New York, 2014.