Feature Weighting and Feature Selection in Fuzzy Clustering

Christian Borgelt

Abstract—**This paper studies the problem of weighting and selecting attributes and principal axes in fuzzy clustering. Its main contribution is a selection method that is not based on simply applying a threshold to computed feature weights, but directly assigns zero weights to features that are not informative enough. This has the important advantage that the clustering result that can be obtained on the selected subspace coincides with the projection (to the selected subspace) of the clustering result that is obtained on the full data space.**

I. Introduction

A serious problem in distance-based clustering is that the more dimensions (attributes) a datasets has, the more the distances between data points—and thus also the distances between data points and constructed cluster centers—tend to become uniform. This, of course, impedes the effectiveness of clustering, as distance-based clustering exploits that these distances *differ*. In addition, in practice often only a subset of the available attributes is relevant for forming clusters, even though this may not be known beforehand. In such cases it is desirable to have a clustering algorithm that automatically weights the attributes or even selects a proper subset.

In general, there are three principles to do feature selection for clustering. The first is a *filter* approach (see e.g. [7], [12]), which tries to assess and select features without any explicit reference to the clustering algorithm to be employed. The second is a *wrapper* approach (see e.g. [6], [8], [5]), which uses a clustering algorithm as an evaluator for chosen feature subsets and may employ different search strategies for choosing the subsets to evaluate. The final approach tries to combine clustering and feature selection by pushing the feature selection method into the clustering algorithm (see e.g. [18], [16]). It should also be noted that any feature weighting scheme (which may, in itself, employ any of these three principles) can be turned into a feature selection method by simply applying a weight threshold to the computed feature weights.

In this paper I study the problem of weighting and selecting features in clustering [1], [2], [11], [4]. Apart from reviewing straighforward modifications of Gustafson–Kessel fuzzy clustering [10] and attribute weighting fuzzy clustering [13] that lead to attribute weighting schemes, it introduces a new feature selection method by applying the idea of an alternative to the fuzzifier [14] to the latter scheme. The resulting combined feature weighting and selection method has the advantage that the obtained clustering result on the chosen subspace coincides with the projection of the result obtained on the full data space. Finally extensions to principal axes selection are discussed.

II. Preliminaries and Notation

Throughout this paper I assume that as input we are given an m-dimensional data set $\mathbf{X}$ that consists of n data points $\vec{x}_j = (x_{j1}, \ldots, x_{jm})$, $1 \le j \le n$. This data set may also be seen as a data matrix $\mathbf{X} = (x_{jk})_{1\le j\le n, 1\le k\le m}$, the rows of which are the data points. The objective is to group the data points into c clusters, which are described by m-dimensional cluster centers $\vec{\mu}_i = (\mu_{i1}, \ldots, \mu_{im})$, $1 \le i \le c$. These cluster centers as well as the feature weights that will be derived (as they can be interpreted as cluster shape and size parameters) are jointly denoted by the parameter set $\mathbf{C}$. The (fuzzy) assignment of the data points to the cluster centers is described by a (fuzzy) membership matrix $\mathbf{U} = (u_{ij})_{1\le i\le c, 1\le j\le n}$.

III. Attribute Weighting

This section reviews two basic methods to compute attribute weights in fuzzy clustering that can be derived in a straightforward manner from known algorithms. Its main purpose is to contrast these closely related methods and to set the stage for the attribute selection approach developed in this paper.

A. Axes-parallel Gustafson–Kessel Fuzzy Clustering

A very direct way to obtain attribute weights is to apply axes-parallel Gustafson–Kessel fuzzy clustering [15] with one global set of variances instead of the usual c cluster-specific sets. In this case we have to minimize the objective function

$$J(\mathbf{X}, \mathbf{C}, \mathbf{U}) = \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) \sum_{k=1}^{m} \sigma_k^{-2} (x_{jk} - \mu_{ik})^2$$

subject to $\prod_{k=1}^{m} \sigma_k^{-2} = 1$ (equivalent to $\prod_{k=1}^{m} \sigma_k^2 = 1$) and again the standard constraints $\forall j, 1 \le j \le n : \sum_{i=1}^{c} u_{ij} = 1$ and $\forall i, 1 \le i \le c : \sum_{j=1}^{n} u_{ij} > 0$. The inverse variances σ_k^{-2} are the desired attribute weights, which have to be found by optimizing the objective function. The membership transformation function h is a convex function on the unit interval. Usually $h(u_{ij}) = u_{ij}^{\alpha}$ with a user-specified *fuzzifier* α (most often $\alpha = 2$) is chosen, but there are also other suggestions, for example [14] (see also Section IV-B). As the methods discussed in this paper work with any choice of the function h, its exact form will be left unspecified in the following.

Incorporating the constraint on the variances σ_k^2 into the objective function yields the Lagrange functional (with the Lagrange multiplier λ)

$$\mathcal{L}(\mathbf{X}, \mathbf{C}, \mathbf{U}, \lambda) = \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) \sum_{k=1}^{m} \sigma_k^{-2} (x_{jk} - \mu_{ik})^2 + \lambda \Big(1 - \prod_{k=1}^{m} \sigma_k^{-2}\Big).$$

Christian Borgelt is with the European Center for Soft Computing, Campus Mieres, Edificio Científico-Tecnológico, c/ Gonzalo Gutiérrez Quirós s/n, 33600 Mieres, Asturias, Spain (email: christian.borgelt@softcomputing.es).

We therefore obtain as necessary conditions for a minimum

$$\frac{\partial}{\partial \sigma_k^{-2}} \mathcal{L}(\mathbf{X}, \mathbf{C}, \mathbf{U}, \lambda) = s_k^2 - \lambda \prod_{\substack{r=1\\ r\neq k}}^{m} \sigma_r^{-2} = s_k^2 - \lambda \sigma_k^2 \overset{!}{=} 0,$$

where

$$s_k^2 \overset{\text{def}}{=} \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij})(x_{jk} - \mu_{ik})^2$$

and the second step follows from the fact that, since it is $\prod_{r=1}^{m} \sigma_r^{-2} = 1$, we have $\prod_{r=1;r!=k}^{m} \sigma_r^{-2} = \sigma_k^2$. From these conditions (one for each value of k, $1 \leq k \leq m$) it follows

$$\lambda \sigma_k^2 = s_k^2 \qquad \text{and thus} \qquad \sigma_k^2 = \lambda^{-1} s_k^2.$$

In order to determine λ, we exploit the variance constraint:

$$\prod_{k=1}^{m} \lambda \sigma_k^2 = \lambda^m \prod_{k=1}^{m} \sigma_k^2 = \lambda^m = \prod_{k=1}^{m} s_k^2,$$

which leads to

$$\lambda = \Big(\prod_{k=1}^{m} s_k^2 \Big)^{\frac{1}{m}} \qquad \text{and thus} \qquad \sigma_k^2 = s_k^2 \Big(\prod_{r=1}^{m} s_r^2 \Big)^{-\frac{1}{m}}$$

as the resulting update rule for the variances σ_k^2. The desired feature weights can now easily be found by inverting the σ_k^2.

B. Attribute Weighting Fuzzy Clustering

An alternative attribute weighting scheme was suggested in [13]. Again the original suggestion employed cluster-specific attribute weights, while here I am using only one global weight set $\{w_1, \ldots, w_m\}$. The objective function to minimize is

$$J(\mathbf{X}, \mathbf{C}, \mathbf{U}) = \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) \sum_{k=1}^{m} w_k^v (x_{jk} - \mu_{ik})^2,$$

subject to $\sum_{k=1}^{m} w_k = 1$ and again the standard constraints $\forall j, 1 \leq j \leq n \colon \sum_{i=1}^{c} u_{ij} = 1$ and $\forall i, 1 \leq i \leq c \colon \sum_{j=1}^{n} u_{ij} > 0$. The difference to the approach in the preceding section consists in the (user-specified) exponent v that controls the influence of the attribute weights (which is analogous to the fuzzifier in the standard transformation of the membership degrees) and the different weight constraint (the sum of the weights, instead of their product, has to be equal to 1).

Incorporating the constraint on the attribute weights into the objective function yields the Lagrange functional

$$\begin{aligned} &\mathcal{L}(\mathbf{X}, \mathbf{C}, \mathbf{U}, \lambda) \\ &= \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) \sum_{k=1}^{m} w_k^v (x_{jk} - \mu_{ik})^2 + \lambda \Big(1 - \sum_{k=1}^{m} w_k \Big). \end{aligned}$$

We therefore obtain as necessary conditions for a minimum

$$\nabla_{w_k} \mathcal{L}(\mathbf{X}, \mathbf{C}, \mathbf{U}, \lambda) = v w_k^{v-1} s_k^2 - \lambda \overset{!}{=} 0$$

with s_k^2 defined as above. It follows

$$\lambda = v w_k^{v-1} s_k^2 \qquad \text{and thus} \qquad w_k = \Big(\frac{\lambda}{v} s_k^{-2} \Big)^{\frac{1}{v-1}}$$

In order to determine λ, we exploit the weight constraint:

$$1 = \sum_{k=1}^{m} w_k = \sum_{k=1}^{m} \Big(\frac{\lambda}{v} s_k^{-2} \Big)^{\frac{1}{v-1}} = \Big(\frac{\lambda}{v} \Big)^{\frac{1}{v-1}} \sum_{k=1}^{m} s_k^{\frac{2}{1-v}},$$

which leads to

$$\lambda = v \Big(\sum_{k=1}^{m} s_k^{\frac{2}{1-v}} \Big)^{1-v}.$$

The resulting update rule for the attribute weights is therefore

$$w_k = \frac{s_k^{\frac{2}{1-v}}}{\sum_{r=1}^{m} s_r^{\frac{2}{1-v}}}, \qquad \text{or} \qquad w_k = \frac{s_k^{-2}}{\sum_{r=1}^{m} s_r^{-2}} \quad \text{if } v = 2.$$

Note how this update rule compares to that of axes-parallel Gustafson–Kessel fuzzy clustering, which may be written as

$$\sigma_k^{-2} = \frac{s_k^{-2}}{(\prod_{r=1}^{m} s_r^{-2})^{-\frac{1}{m}}}.$$

The difference resides in the normalization factor and the exponent v that is used in attribute weighting fuzzy clustering.

IV. Attribute Selection

The methods reviewed in the preceding section yield attribute weights, either as inverse variances σ_k^{-2} or directly as weights w_k, $1 \leq k \leq m$. It is important to note that in both cases it is impossible that any attribute weight vanishes (which, for axes-parallel Gustafson–Kessel fuzzy clustering this is already obvious from the constraint $\prod_{i=1}^{m} \sigma_k^{-2} = 1$, since a vanishing weight would lead to a violation of this constraint). Therefore an additional step or a modification of the approach is necessary in order to select attributes (which may be achieved by allowing attribute weights to become 0). This section discusses two possible approaches, the second of which is the main contribution of this paper.

A. Attribute Weight Threshold

The most straightforward way of selecting attributes with the two methods reviewed above is, of course, to simply apply a weight threshold: select those attributes that were assigned a weight greater than the threshold. The advantage of such an approach is that one may also keep the threshold flexible, choosing it dynamically in such a way that the best r attributes (with r to be specified by a user) are selected.

However, a severe disadvantage of this approach is that the weights of the attributes that are discarded are actually not zero. Hence the projection of the result of the clustering algorithm, as it is obtained on the full data space, will usually *not* coincide with the clustering result that is obtained on only the subspace. The reason is simply that the discarded attributes, even if their weight is small, influence the membership degrees in the clustering result on the full data space and thus the cluster parameters. In contrast to this, there is no such influence when clustering the projected data set, because the discarded attribute are never accessed. As a consequence it is desirable to have a method that assigns attribute weights that may be 0, so that their influence is actually removed from the clustering result (as it is obtained on the whole data set).

B. Alternative Convex Transformation

The core idea of the attribute selection method introduced in this section is to transfer the analysis of the effect of the fuzzifier (the exponent of the membership degrees) and its possible alternatives, as it was carried out in [14], to attribute weights. As [14] showed, it is necessary to apply a convex function $h(\cdot)$ to the membership degrees in order to rule out a crisp assignment. Raising the membership degrees u_{ij} to a user-specified power (namely the fuzzifier) is, of course, such a convex function, but has the disadvantage that it forces all assignments to be fuzzy (that is, to differ from 0 and 1). The reason is that the derivative of this function vanishes at 0. If we want to maintain the possibility of crisp assignments, we rather have to choose a function h with $h'(0) > 0$.

With the approach of attribute weighting fuzzy clustering it becomes possible to transfer this idea to the transformation of the attribute weights. That is, instead raising them to the power v as in [13], we may transform the attribute weights by

$$g(x) = \alpha x^2 + (1-\alpha)x \qquad \text{with } \alpha \in (0, 1].$$

The same function was suggested as an alternative transformation of the membership degrees in [14], and a fuzzy clustering algorithm was derived that allowed for crisp memberships in case the distances of a data point to different clusters differed considerably. Here the idea is that the same method applied to attribute weights should allow us to derive a fuzzy clustering algorithm that assigns zero weights to some attributes, thus effectively selecting attributes during the clustering process.

However, as was also discussed in [14], the above function has the disadvantage that its parameter α is difficult to interpret and thus difficult to choose adequately. Fortunately, [14] also provided a better parameterization, namely

$$g(x) = \frac{1-\beta}{1+\beta}\,x^2 + \frac{2\beta}{1+\beta}\,x \qquad \text{with } \beta \in [0, 1).$$

Transferred to attribute weights the underlying rationale is as follows: suppose that the data set to cluster has only two dimensions. Then the objective function to minimize is

$$J(\mathbf{X}, \mathbf{C}, \mathbf{U}) = g(w)s_1^2 + g(1-w)s_2^2$$

with s_k^2 defined as above and w the weight of the first attribute. Taking the partial derivative of this function w.r.t. the attribute weight w yields as a necessary condition for a minimum

$$\frac{\partial}{\partial w} J(\mathbf{X}, \mathbf{C}, \mathbf{U}) = g'(w)s_1^2 - g'(1-w)s_2^2 \overset{!}{=} 0.$$

Suppose now, without loss of generality (as the dimensions can always be exchanged), that $s_1^2 > s_2^2$ and $w = 1$. Then

$$\frac{s_1^2}{s_2^2} = \frac{g'(0)}{g'(1)} = \frac{\frac{2\beta}{1+\beta}}{2\frac{1-\beta}{1+\beta} + \frac{2\beta}{1+\beta}} = \beta.$$

That is, β is the smallest ratio of total intra-cluster variances (since $\sigma_1^2/\sigma_2^2 = s_1^2/s_2^2$) at which the dimension having the larger total intra-cluster variance will be suppressed. This is fairly intuitive and thus reasonably easy to choose.

Generally, we have to minimize the objective function

$$J(\mathbf{X}, \mathbf{C}, \mathbf{U}) = \sum_{i=1}^{c}\sum_{j=1}^{n} h(u_{ij}) \sum_{k=1}^{m} g(w_k)(x_{jk} - \mu_{ik})^2$$

subject to $\sum_{k=1}^{m} w_k = 1$ and again the standard constraints $\forall j, 1 \le j \le n\colon \sum_{i=1}^{c} u_{ij} = 1$ and $\forall i, 1 \le i \le c\colon \sum_{j=1}^{n} u_{ij} > 0$. Incorporating the constraint on the attribute weights into the objective function yields the Lagrange functional

$$\begin{aligned}&\mathcal{L}(\mathbf{X}, \mathbf{C}, \mathbf{U}, \lambda)\\ &\quad= \sum_{i=1}^{c}\sum_{j=1}^{n} h(u_{ij}) \sum_{k=1}^{m} g(w_k)(x_{jk} - \mu_{ik})^2 + \lambda\Big(1 - \sum_{k=1}^{m} w_k\Big).\end{aligned}$$

We therefore obtain as necessary conditions for a minimum

$$\nabla_{w_k} \mathcal{L}(\mathbf{X}, \mathbf{C}, \mathbf{U}, \lambda) = \left(2\frac{1-\beta}{1+\beta}\,w_k + \frac{2\beta}{1+\beta}\right) s_k^2 - \lambda \overset{!}{=} 0$$

with s_k^2 defined as above. It follows

$$w_k = \frac{1+\beta}{2(1-\beta)}\left(\lambda s_k^{-2} - \frac{2\beta}{1+\beta}\right).$$

In order to determine λ, we exploit the weight constraint:

$$\begin{aligned}1 = \sum_{\substack{k=1\\ w_k>0}}^{m} w_k &= \frac{1+\beta}{2(1-\beta)} \sum_{\substack{k=1\\ w_k>0}}^{m} \left(\lambda s_k^{-2} - \frac{2\beta}{1+\beta}\right)\\ &= -\frac{2\beta m_\oplus}{2(1-\beta)} + \frac{1+\beta}{2(1-\beta)} \sum_{\substack{k=1\\ w_k>0}}^{m} \lambda s_k^{-2},\end{aligned}$$

where $m_\oplus$ denotes the number of attributes that have a positive weight (that is, for which $w_k > 0$). This leads to

$$\lambda = \frac{2(1+\beta(m_\oplus - 1))}{(1+\beta)\sum_{k=1; w_k>0}^{m} s_k^{-2}}$$

The resulting update rule for the attribute weights is therefore

$$\begin{aligned}w_k &= \frac{1+\beta}{2(1-\beta)}\left(\frac{2(1+\beta(m_\oplus - 1))}{(1+\beta)\sum_{r=1; w_r>0}^{m} s_r^{-2}}\,s_k^{-2} - \frac{2\beta}{1+\beta}\right)\\ &= \frac{1}{1-\beta}\left(\frac{1+\beta(m_\oplus - 1)}{\sum_{r=1; w_r>0}^{m} s_r^{-2}}\,s_k^{-2} - \beta\right)\end{aligned}$$

The needed value $m_\oplus$ can be determined as follows: if an attribute weight w_k does not vanish, it is obviously proportional to the value of the corresponding s_k^{-2}. Hence we sort the attributes descendingly w.r.t. the values s_k^{-2}, $1 \le k \le m$. Let ς describe the index permutation that sorts the attributes into this order (that is, let $\varsigma(r) = 1$ if s_r^{-2} is largest among all s_k^{-2}, $\varsigma(r) = 2$ if s_r^{-2} is the second largest etc). Then $m_\oplus$ can be determined from the fact that the second factor in the update rule for the weight w_k must be positive, namely as

$$m_\oplus = \max\left\{k \;\middle|\; s_{\varsigma(k)}^{-2} > \frac{\beta}{1+\beta(k-1)} \sum_{r=1}^{k} s_{\varsigma(r)}^{-2}\right\}.$$

Note that these results are completely analogous to the results obtained in [14] for membership degrees.

V. Principal Axes Weighting

A standard problem of attribute weighting and selection approaches is that often attributes that are highly correlated will receive very similar weights or will both be selected, even though they are obviously redundant: one of them contains already almost all of the relevant information. In order to cope with this problem, an approach in the spirit of principal component analysis may be employed: instead of weighting and selecting attributes, one may try to find (and weight) linear combinations of the attributes, and thus (principal) axes of the data set. This section shows how the methods of Section III can be extended to principal axes weighting. In the case of Gustafsson–Kessel fuzzy clustering this is trivial (see Section V-A), while the extension of attribute weighting fuzzy clustering can be achieved by reformulating Gustafsson–Kessel fuzzy clustering so that the specification of the (principal) axes and their weights is separated.

A. Gustafson–Kessel Fuzzy Clustering

Gustafson-Kessel fuzzy clustering uses a Mahalanobis distance, which, in the standard form of this algorithm, is based on cluster-specific covariance matrices $\mathbf{\Sigma}_i$, $i = 1, \ldots, c$. Here, however, since I am interested in a global weighting of (principal) axes of the data space, I use a only a single covariance matrix $\mathbf{\Sigma}$. That is, I consider the objective function

$$J(\mathbf{X}, \mathbf{C}, \mathbf{U}) = \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) (\vec{x}_j - \vec{\mu}_i)^\top \mathbf{\Sigma}^{-1} (\vec{x}_j - \vec{\mu}_i),$$

which is to be minimized subject to $|\mathbf{\Sigma}^{-1}| = 1$ (equivalent to $|\mathbf{\Sigma}| = 1$; intuitive interpretation: fixed cluster volume) and the standard constraints $\forall j, 1 \leq j \leq n : \sum_{i=1}^{c} u_{ij} = 1$ and $\forall i, 1 \leq i \leq c : \sum_{j=1}^{n} u_{ij} > 0$. Incorporating the constraint on the covariance matrix $\mathbf{\Sigma}$ into the objective function yields the Lagrange functional (with the Lagrange multiplier λ)

$$\begin{aligned} &\mathcal{L}(\mathbf{X}, \mathbf{C}, \mathbf{U}, \lambda) \\ &= \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) (\vec{x}_j - \vec{\mu}_i)^\top \mathbf{\Sigma}^{-1} (\vec{x}_j - \vec{\mu}_i) + \lambda(1 - |\mathbf{\Sigma}^{-1}|). \end{aligned}$$

We therefore obtain as a necessary condition for a minimum

$$\nabla_{\mathbf{\Sigma}^{-1}} \mathcal{L}(\mathbf{X}, \mathbf{C}, \mathbf{U}, \Lambda) = \mathbf{S} - \lambda |\mathbf{\Sigma}^{-1}| \mathbf{\Sigma} \overset{!}{=} \mathbf{0}$$

where $\mathbf{0}$ is an $m \times m$ zero matrix and

$$\mathbf{S} \overset{\text{def}}{=} \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) (\vec{x}_j - \vec{\mu}_i)(\vec{x}_j - \vec{\mu}_i)^\top.$$

From this condition it follows that

$$\lambda |\mathbf{\Sigma}^{-1}| \mathbf{\Sigma} = \lambda \mathbf{\Sigma} = \mathbf{S} \qquad \text{and thus} \qquad \mathbf{\Sigma} = \lambda^{-1} \mathbf{S}.$$

In order to determine λ, we look at the determinant:

$$|\lambda \mathbf{\Sigma}| = \lambda^m |\mathbf{\Sigma}| = \lambda^m = |\mathbf{S}| \qquad \text{and thus} \qquad \lambda = |\mathbf{S}|^{\frac{1}{m}}.$$

The resulting update rule for the covariance matrix $\mathbf{\Sigma}$ is

$$\mathbf{\Sigma} = \mathbf{S} |\mathbf{S}|^{-\frac{1}{m}}.$$

In order to obtain explicit weights for (principal) axes, we observe that, since $\mathbf{\Sigma}$ is a symmetric and positive definite matrix, it possesses an eigenvalue decomposition

$$\mathbf{\Sigma} = \mathbf{R} \mathbf{D}^2 \mathbf{R}^\top \qquad \text{with} \qquad \mathbf{D} = \operatorname{diag}(\sigma_1, \ldots, \sigma_m)$$

(i.e., eigenvalues σ_1^2 to σ_m^2) and an orthogonal matrix $\mathbf{R}$, the columns of which are the corresponding eigenvectors.[1] This enables us to write the inverse of the covariance matrix $\mathbf{\Sigma}$ as

$$\mathbf{\Sigma}^{-1} = \mathbf{T} \mathbf{T}^\top \qquad \text{with} \qquad \mathbf{T} = \mathbf{R} \mathbf{D}^{-1}.$$

As a consequence, we can rewrite the objective function as

$$\begin{aligned} &J(\mathbf{X}, \mathbf{C}, \mathbf{U}) \\ &= \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) (\vec{x}_j - \vec{\mu}_i)^\top \mathbf{T} \mathbf{T}^\top (\vec{x}_j - \vec{\mu}_i) \\ &= \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) ((\vec{x}_j - \vec{\mu}_i)^\top \mathbf{R} \mathbf{D}^{-1}) ((\vec{x}_j - \vec{\mu}_i)^\top \mathbf{R} \mathbf{D}^{-1})^\top \\ &= \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) \sum_{k=1}^{m} \sigma_k^{-2} \Big(\sum_{l=1}^{m} (x_{jl} - \mu_{il}) r_{lk} \Big)^2, \end{aligned}$$

In this form the scaling and the rotation of the data space that are encoded in the covariance matrix $\mathbf{\Sigma}$ are nicely separated: the former is represented by the variances σ_k^2, $k = 1, \ldots, m$ (or their inverses σ_k^{-2}), the latter by the orthogonal matrix $\mathbf{R}$. In other words: the inverse variances σ_k^{-2} (the eigenvalues of $\mathbf{\Sigma}^{-1}$) provide the desired axes weights, while the corresponding eigenvectors (the columns of $\mathbf{R}$) indicate the axes.

B. Reformulation of Gustafson–Kessel Fuzzy Clustering

In order to transfer the approach of [13] and the one developed in Section IV-B, we start from the rewritten objective function, in which the scaling and the rotation of the data space are separated and thus can be treated independently. Deriving the update rule for the scaling factors σ_k^{-2} is trivial, since basically the same result is obtained as for axes-parallel Gustafson–Kessel fuzzy clustering (see Section III-A), namely

$$\sigma_k^2 = s_k^2 \Big(\prod_{r=1}^{m} s_r^2 \Big)^{-\frac{1}{m}},$$

with the only difference that now we have

$$s_k^2 \overset{\text{def}}{=} \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) \Big(\sum_{l=1}^{m} (x_{jl} - \mu_{il}) r_{lk} \Big)^2.$$

Note that this update rule reduces to the update rule for axes-parallel Gustafson–Kessel clustering derived in Section III-A if $\mathbf{R} = \mathbf{1}$ (where $\mathbf{1}$ is an $m \times m$ unit matrix), which provides a simple sanity check of this rule.

In order to derive an update rule for the orthogonal matrix $\mathbf{R}$, we have to take into account that in contrast to how the covariance matrix $\mathbf{\Sigma}$ is treated in normal Gustafson–Kessel fuzzy clustering, there is an additional constraint, namely that $\mathbf{R}$ must be orthogonal, that is, $\mathbf{R}^\top = \mathbf{R}^{-1}$. This constraint

[1] Note that the eigenvalues of a symmetric and positive definite matrix are all positive and thus it is possible to write them as squares.

can conveniently be expressed by requiring $\mathbf{R}\mathbf{R}^\top = \mathbf{1}$. Incorporating this constraint[2] into the objective function yields the Lagrange functional

$$\begin{aligned}
&\mathcal{L}(\mathbf{X}, \mathbf{C}, \mathbf{U}, \mathbf{\Lambda}) \\
&= \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) ((\vec{x}_j - \vec{\mu}_i)^\top \mathbf{R}\mathbf{D}^{-1}) ((\vec{x}_j - \vec{\mu}_i)^\top \mathbf{R}\mathbf{D}^{-1})^\top \\
&+ \mathit{trace}\big(\mathbf{\Lambda}(\mathbf{1} - \mathbf{R}\mathbf{R}^\top)\big),
\end{aligned}$$

where $\mathbf{\Lambda}$ a symmetric $m \times m$ matrix of Lagrange multipliers and $\mathit{trace}(\cdot)$ is the trace operator, which for an $m \times m$ matrix $\mathbf{M}$ is defined as $\mathit{trace}(\mathbf{M}) = \sum_{k=1}^{m} m_{kk}$. We thus obtain as a necessary condition for a minimum (see the appendix for detailed computations of the derivatives)

$$\nabla_{\mathbf{R}} \mathcal{L}(\mathbf{X}, \mathbf{C}, \mathbf{U}, \lambda) = 2\mathbf{S}\mathbf{R}\mathbf{D}^{-2} - 2\mathbf{\Lambda}\mathbf{R} \overset{!}{=} \mathbf{0},$$

where $\mathbf{0}$ is an $m \times m$ zero matrix and $\mathbf{S}$ is defined as in standard Gustafson–Kessel fuzzy clustering (see above). It follows

$$\mathbf{\Lambda} = \mathbf{S}\mathbf{R}\mathbf{D}^{-2}\mathbf{R}^\top = \mathbf{S}\mathbf{\Sigma}^{-1}.$$

Since $\mathbf{S}$ is clearly a symmetric and positive definite matrix, it possesses an eigenvalue decomposition

$$\mathbf{S} = \mathbf{O}\mathbf{E}^2\mathbf{O}^\top \qquad \text{with} \qquad \mathbf{E} = \operatorname{diag}(e_1, \ldots, e_m)$$

(positive eigenvalues e_1^2 to e_m^2) and an orthogonal matrix $\mathbf{O}$. Furthermore we observe that $\mathbf{R}\mathbf{D}^{-2}\mathbf{R}^\top = \mathbf{\Sigma}^{-1}$ is also symmetric (and positive definite) and that the product $\mathbf{S}\mathbf{\Sigma}^{-1}$ is symmetric (as it equals $\mathbf{\Lambda}$, which is symmetric by definition). As a consequence, $\mathbf{S}$ and $\mathbf{\Sigma}^{-1}$ commute and have the same eigenspaces [9], which immediately yields the update rule[3]

$$\mathbf{R} = \mathbf{O}.$$

As a sanity check I show that first updating the orthogonal matrix $\mathbf{R}$ and then applying the update rule for the variances σ_k^2 (using the updated matrix $\mathbf{R}$) yields the same update rule for the covariance matrix $\mathbf{\Sigma}$ as a direct derivation (see Section V-A). To do so, observe that

$$\mathbf{E}^2 = \mathbf{O}^\top\mathbf{S}\mathbf{O} = \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) \mathbf{O}^\top (\vec{x}_j - \vec{\mu}_i)(\vec{x}_j - \vec{\mu}_i)^\top \mathbf{O}$$

and therefore (since $\mathbf{E}^2 = \operatorname{diag}(e_1^2, \ldots, e_m^2)$)

$$e_k^2 = \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) \Big(\sum_{l=1}^{m} (x_{jl} - \mu_{il}) o_{lk} \Big)^2,$$

which coincides with the definition of s_k^2 (see above), since $\mathbf{R} = \mathbf{O}$. Hence we can write the update rule for the σ_k^2 as

$$\mathbf{D}^2 = \mathbf{E}^2 |\mathbf{E}^2|^{-\frac{1}{m}}.$$

Therefore the new value of $\mathbf{\Sigma} = \mathbf{R}\mathbf{D}^2\mathbf{R}^\top$ is

$$\mathbf{\Sigma} = \mathbf{O}\mathbf{E}^2\mathbf{O}^\top |\mathbf{E}^2|^{-\frac{1}{m}} = \mathbf{O}\mathbf{E}^2\mathbf{O}^\top |\mathbf{O}\mathbf{E}^2\mathbf{O}^\top|^{-\frac{1}{m}} = \mathbf{S}|\mathbf{S}|^{-\frac{1}{m}},$$

which is the update rule of standard Gustafson–Kessel fuzzy clustering (with joint treatment of scaling and rotation).

C. Alternative Weighting

With the reformulation of Gustafson–Kessel fuzzy clustering, as it was obtained in the preceding section, it becomes possible to replace the weight update while keeping the update of the (principal) axes (the update of the orthogonal matrix $\mathbf{R}$). In particular, we can use the update rule for the w_k of attribute weighting fuzzy clustering instead of the Gustafsson–Kessel rule for the σ_k^{-2}. The resulting update rule (the derivation follows exactly the same lines as above) is

$$w_k = \frac{s_k^{\frac{2}{1-v}}}{\sum_{r=1}^{m} s_r^{\frac{2}{1-v}}}, \qquad \text{or} \qquad w_k = \frac{s_k^{-2}}{\sum_{r=1}^{m} s_r^{-2}} \quad \text{if } v = 2,$$

with s_k^2 defined as for Gustafson–Kessel clustering, that is, as

$$s_k^2 \overset{\text{def}}{=} \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) \Big(\sum_{l=1}^{m} (x_{jl} - \mu_{il}) r_{lk} \Big)^2.$$

Note that $\operatorname{diag}(w_1, \ldots, w_m)$ corresponds to the diagonal matrix $\mathbf{D}^{-2}$ in normal Gustafson–Kessel fuzzy clustering.

VI. Principal Axes Selection

In analogy to the transition from attribute weighting (Section III) to attribute selection (Section IV), it is possible to make the transition from (principal) axes weighting (Section V) to (principal) axes selection (this section).

A. Axis Weight Threshold

Again the most straightforward approach to select a (principal) axes based on a weighting scheme is to use a weight threshold, which is completely in line with the normal procedure in principal component analysis. It has the same advantages and disadvantages that were already discussed in Section IV-A. Therefore a selection method that directly assigns vanishing weights to irrelevant axes is desirable.

B. Alternative Convex Transformation

With the reformulated Gustafson–Kessel fuzzy clustering algorithm derived in Section V-B a (principal) axes selection method can easily be obtained: we simply replace the update rule for the weights with the one obtained in Section IV-B. This leads (after a fairly straightforward and analogous derivation, which I do not spell out here) to the update rule

$$w_k = \frac{1}{1-\beta} \left(\frac{1 + \beta(m_\oplus - 1)}{\sum_{r=1; w_r > 0}^{m} s_r^{-2}} s_k^{-2} - \beta \right),$$

again with the modified definition of the s_k, that is,

$$s_k^2 \overset{\text{def}}{=} \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) \Big(\sum_{l=1}^{m} (x_{jl} - \mu_{il}) r_{lk} \Big)^2$$

and $m_\oplus$ defined as described at the end of Section IV-B.

[2] Note that, in principle, the orthogonality constraint alone is not enough as it is compatible with $|\mathbf{R}| = -1$, while we need $|\mathbf{R}| = 1$. However, we will see in the following that the unit determinant constraint is automatically satisfied by the solution and thus we can avoid incorporating it. This is similar to the treatment of the covariance matrix $\mathbf{\Sigma}$ in standard Gustafson–Kessel clustering, where, in principle, we need constraints ensuring that it is symmetric and positive definite. However, since the result automatically satisfies these constraints, they are neglected.

[3] Note that this rule satisfies $|\mathbf{R}| = 1$ as claimed in the preceding footnote.

TABLE I
RESULTS ON THE IRIS DATA WITH 2 CLUSTERS.

attribute	gk	$v = 2$	$\beta = .126$	$\beta = .235$	$\beta = .500$	$\beta = .662$
sepal length	0.7367	0.1501	0.0901	0.0000	0.0000	0.0000
sepal width	0.4698	0.0937	0.0000	0.0000	0.0000	0.0000
petal length	2.0011	0.4447	0.5618	0.6461	0.7859	1.0000
petal width	1.4437	0.3115	0.3481	0.3539	0.2141	0.0000

TABLE II
RESULTS ON THE IRIS DATA WITH 3 CLUSTERS.

attribute	gk	$v = 2$	$\beta = .049$	$\beta = .095$	$\beta = .300$	$\beta = .530$
sepal length	0.5666	0.0788	0.0420	0.0000	0.0000	0.0000
sepal width	0.3019	0.0427	0.0000	0.0000	0.0000	0.0000
petal length	2.7300	0.4826	0.5296	0.5529	0.5989	1.0000
petal width	2.1413	0.3959	0.4284	0.4471	0.4011	0.0000

VII. EXPERIMENTS

In order to test the methods suggested above I implemented them as part of my fuzzy and probabilistic clustering toolbox.[4] In all experiments reported in the following I used the standard membership degree transformation $h(u_{ij}) = u_{ij}^2$.

Tables I and II show the results of the attribute weighting and selection methods on the well-known Iris data, with 2 and 3 clusters, respectively. Table III shows the results on the equally well-known wine data set from the UCI machine learning repository [3]. Each table lists, in the first column, the attributes used for clustering the data (the class attribute was, of course, not used in both cases). The following columns state the attribute weights obtained with (different parameterizations of) the suggested feature weighting and selection methods.

The second column of each table refers to axes-parallel Gustafson–Kessel fuzzy clustering and states the σ_k^{-2} as the attribute weights. Note that this is the only column in which the sum of the entries does not equal 1, since here the constraint is that their product must be 1. The third column refers to attribute weighting fuzzy clustering with a weight exponent $v = 2$ and states the obtained w_k. Note that the weights in this column are identical to those that would be obtained with the attribute selection method introduced in this paper and $\beta = 0$, since then $g(w_k) = w_k^2$.

All following columns refer to the new attribute selection method, with different values for the parameter β. Except for the columns with $\beta = 0.5$ in Table I and $\beta = 0.3$ in Table II, which have been added to provide an additional impression of the effect on attribute weighting, the chosen values for β are the smallest ones that yield the number of non-vanishing weights in the corresponding column. Clearly, the larger the value of β, the fewer attributes get selected.

Experiments with principal axes weighting and selection are still under way (since the implementation is not quite finished yet) and will be included in the final version of the paper.

TABLE III
RESULTS ON THE WINE DATA WITH 3 CLUSTERS.

attribute	gk	$v = 2$	$\beta = .109$	$\beta = .120$	$\beta = 0.153$	$\beta = .374$
att01	0.9667	0.0649	0.0000	0.0000	0.0000	0.0000
att02	0.8749	0.0563	0.0000	0.0000	0.0000	0.0000
att03	0.7449	0.0493	0.0000	0.0000	0.0000	0.0000
att04	0.8471	0.0553	0.0000	0.0000	0.0000	0.0000
att05	0.7819	0.0520	0.0000	0.0000	0.0000	0.0000
att06	1.2341	0.1024	0.2008	0.2067	0.2057	0.0000
att07	1.6027	0.1515	0.4504	0.4768	0.5415	1.0000
att08	0.8760	0.0589	0.0000	0.0000	0.0000	0.0000
att09	0.9410	0.0690	0.0424	0.0344	0.0000	0.0000
att10	0.9102	0.0633	0.0090	0.0000	0.0000	0.0000
att11	1.0407	0.0763	0.0401	0.0304	0.0000	0.0000
att12	1.3766	0.1247	0.2478	0.2516	0.2528	0.0000
att13	1.1272	0.0760	0.0095	0.0000	0.0000	0.0000

VIII. CONCLUSIONS

In this paper I reviewed feature weighting schemes that can be derived in a fairly straightforward manner from known algorithms and introduced a powerful feature selection method by modifying the weight transformation of attribute weighting fuzzy clustering. In addition, by reformulating Gustafson–Kessel fuzzy clustering so that the rotation (orthogonal matrix $\mathbf{R}$) and the axes weights (inverse eigenvalues), which are encoded in the covariance matrix, are separated, it was possible to transfer the approach from attribute selection to (principal) axes selection. Future work includes making the feature weights cluster-specific (again), which paves the way to apply the method to subspace clustering (as in [17]).

REFERENCES

[1] J.C. Bezdek and N. Pal. *Fuzzy Models for Pattern Recognition.* IEEE Press, New York, NY, USA 1992

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

[3] C.L. Blake and C.J. Merz. *UCI Repository of Machine Learning Databases.* University of California, Irvine, CA, USA 1998 http://www.ics.uci.edu/~mlearn/MLRepository.html

[4] C. Borgelt. *Prototype-based Classification and Clustering.* Habilitation thesis, University of Magdeburg, Germany 2005

[5] R. Butterworth, G. Piatetsky-Shapiro, and D.A. Simovici. On Feature Selection Through Clustering. *Proc. 5th IEEE Int. Conf. on Data Mining (ICDM 2005, Houston, TX)*, 581–584. IEEE Press, Piscataway, NJ, USA 2005

[6] M. Dash and H. Liu. Feature Selection for Clustering. *Proc. 4th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD 2000, Kyoto, Japan)*, 110–121. Springer-Verlag, London, United Kingdom 2000

[7] M. Dash, K. Choi, P. Scheuermann, and H. Liu. Feature Selection for Clustering: A Filter Solution. *Proc. 2nd IEEE Int. Conf. on Data Mining (ICDM 2002, Maebashi, Japan)*, 51–58.

[8] J.G. Dy and C.E. Brodley. Visualization and Interactive Feature Selection for Unsupervised Data. *Proc. 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2000, Boston, MA)*, 360–364. ACM Press, Ney York, NY, USA 2000

[9] G.H. Golub and C.F. Van Loan. *Matrix Computations*, 3rd edition. The Johns Hopkins University Press, Baltimore, MD, USA 1996

[10] E.E. Gustafson and W.C. Kessel. Fuzzy Clustering with a Fuzzy Covariance Matrix. *Proc. of the IEEE Conf. on Decision and Control (CDC 1979, San Diego, CA)*, 761–766. IEEE Press, Piscataway, NJ, USA 1979. Reprinted in [1], 117–122

[11] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis.* J. Wiley & Sons, Chichester, England 1999

[4] This toolbox is a set of command line programs written in C, but there also exists a graphical user interface written in Java. The toolbox as well as the graphical user interface can be downloaded from http://www.borgelt.net/software.html. The current version does not yet include the methods described here, but the extended version will be available soon.

[12] P.-E. Jouve and N. Nicoloyannis. A Filter Feature Selection Method for Clustering. *Proc. 15th Int. Symp. on Foundations of Intelligent Systems (ISMIS 2005, Saratoga Springs, NY)*, 583–593. Springer-Verlag, Heidelberg, Germany 2005

[13] A. Keller and F. Klawonn. Fuzzy Clustering with Weighting of Data Variables. *Int. J. of Uncertainty, Fuzziness and Knowledge-based Systems* 8:735-746. World Scientific, Hackensack, NJ, USA 2000

[14] F. Klawonn and F. Höppner. What is Fuzzy about Fuzzy Clustering? Understanding and Improving the Concept of the Fuzzifier. *Proc. 5th Int. Symposium on Intelligent Data Analysis (IDA 2003, Berlin, Germany)*, 254–264. Springer-Verlag, Berlin, Germany 2003

[15] F. Klawonn and R. Kruse. Constructing a Fuzzy Controller from Data. *Fuzzy Sets and Systems* 85:177–193. North-Holland, Amsterdam, Netherlands 1997

[16] M.H.C. Law, M.A.T. Figueiredo, and A.K. Jain. Simultaneous Feature Selection and Clustering Using Mixture Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)* 26(9):1154–1166. IEEE Press, Piscataway, NJ, USA 2004

[17] L. Parsons, E. Haque, and H. Liu. Subspace Clustering for High-Dimensional Data: A Review. *ACM SIGKDD Explorations Newsletter* 6(1):90-105. ACM Press, New York, NY, USA 2004

[18] V. Roth and T. Lange. Feature Selection in Clustering Problems. *Advances in Neural Information Processing 16: Proc. 17th Ann. Conf. (NIPS 2003, Vancouver, Canada)*. MIT Press, Cambridge, MA, USA 2004

[19] X. Wang, Y. Wang, and L. Wang. Improving Fuzzy *c*-Means Clustering based on Feature-Weight Learning. *Pattern Recognition Letters* 25(10):1123–1132. Elsevier, New York, NY, USA 2004

Appendix

1. Derivative of the objective function in the reformulation of Gustafson–Kessel fuzzy clustering w.r.t. the orthogonal matrix $\mathbf{R}$: it is

$$\begin{aligned}
&\frac{\partial}{\partial r_{ab}} J(\mathbf{X}, \mathbf{C}, \mathbf{U}) \\
&\quad = \frac{\partial}{\partial r_{ab}} \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) \sum_{k=1}^{m} \sigma_k^{-2} \Big(\sum_{l=1}^{m} (x_{jl} - \mu_{il}) r_{lk} \Big)^2 \\
&\quad = \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) \sigma_b^{-2} \frac{\partial}{\partial r_{ab}} \Big(\sum_{l=1}^{m} (x_{jl} - \mu_{il}) r_{lb} \Big)^2 \\
&\quad = 2 \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) \sigma_b^{-2} \Big(\sum_{l=1}^{m} (x_{jl} - \mu_{il}) r_{lb} \Big) (x_{ja} - \mu_{ia})
\end{aligned}$$

and therefore

$$\begin{aligned}
&\nabla_{\mathbf{R}} J(\mathbf{X}, \mathbf{C}, \mathbf{U}) \\
&\quad = 2 \sum_{i=1}^{c} \sum_{j=1}^{n} h(u_{ij}) (\vec{x}_j - \vec{\mu}_i)(\vec{x}_j - \vec{\mu}_i)^\top \mathbf{R} \mathbf{D}^{-2}.
\end{aligned}$$

2. Derivative of the orthogonality constraint w.r.t. $\mathbf{R}$: it is

$$\begin{aligned}
&\frac{\partial}{\partial r_{ab}} \mathit{trace}(\mathbf{\Lambda}(\mathbf{1} - \mathbf{R}\mathbf{R}^\top)) \\
&\quad = \frac{\partial}{\partial r_{ab}} \sum_{i=1}^{m} \sum_{k=1}^{m} \lambda_{ik} \sum_{l=1}^{m} (\delta_{ki} - r_{kl} r_{il}) \\
&\quad = - \sum_{\substack{k=1 \\ k \neq a}}^{m} \lambda_{ak} r_{kb} - \sum_{\substack{k=1 \\ k \neq a}}^{m} \lambda_{ka} r_{kb} - 2\lambda_{aa} r_{ab} \\
&\quad = -2 \sum_{k=1}^{m} \lambda_{ak} r_{kb},
\end{aligned}$$

since $\lambda_{ka} = \lambda_{ak}$ as $\mathbf{\Lambda}$ is symmetric, and therefore

$$\nabla_{\mathbf{R}}\, \mathit{trace}(\mathbf{\Lambda}(\mathbf{1} - \mathbf{R}\mathbf{R}^\top)) = -2\mathbf{\Lambda}\mathbf{R}.$$