FUZZY CLUSTERING AND FUZZY RULES

Frank KLAWONN[a], Annette KELLER[b]

[a] FB Elektrotechnik und Informatik, FH Ostfriesland, Constantiaplatz 4, D-26723 Emden, Germany

[b] Institut für Betriebssysteme und Rechnerverbund, Technische Universität Braunschweig, Germany

Abstract. Fuzzy clustering offers various possibilities for learning fuzzy if–then rules from data for classification tasks as well as for function approximation problems like in fuzzy control. In this paper we review approaches for deriving rules from data by fuzzy clustering and discuss some of their common problems. As a consequence, we propose a new method which is specifically tailored for the task of learning rules.

Keywords. Fuzzy clustering, rule extraction, fuzzy control, fuzzy classification

1 Introduction

The development of fuzzy control was initiated in the early 70's by the idea to construct a controller based on the empirical knowledge of an operator or control engineer. Deriving control rules from data is an alternative approach, which became ever more interesting during the last years since it does not require the difficult knowledge acquisition task from scratch.

There are of course a number of approaches to learning fuzzy rules from data based for instance on techniques of neural (for an overview see [18]) or evolutionary computation (see for example [9]), mostly aiming at optimizing certain parameters of a fuzzy controller. However, fuzzy clustering seems to be a very appealing method for learning rules since there is a close and canonical connection between fuzzy clusters and fuzzy rules. Intuitively, each if–then rule of a Mamdani–type fuzzy controller specifies a vague point of the graph of the control function in the sense that it can be identified with the Cartesian product of the membership functions modeling the linguistic terms appearing in a rule. If for instance triangular membership functions are used, the point or vector with the coordinates of the tips of the triangles is a 'typical' point on the control function. With increasing distance, points in its neighbourhood are less 'typical' and have therefore a decreasing membership degree to the vague point defined by the Cartesian product of the fuzzy sets appearing in the rule. In this sense, a fuzzy controller can be characterized by a typical point (on the control function) and membership function that is decreasing with increasing (transformed [10]) distance to the typical points. (For a more formal presentation of this idea we refer to [16]). Many fuzzy clustering algorithms are exactly pursuing the same strategy: A fuzzy cluster is represented by a typical element – usually the cluster center – and the membership degree of a datum to the cluster is decreasing with increasing, sometimes transformed distance to the cluster center.

In the following section we briefly review the background of fuzzy clustering we need to explain how rules can be derived from fuzzy clusters. In Section 3 we present a variety of methods for applying fuzzy clustering to obtain different kinds of fuzzy rules. Section 4 is devoted to some problems that are caused by these specific approaches and proposes new algorithms that are specifically tailored for the task of learning rules.

2 Fuzzy Clustering and Global Partitions

Many fuzzy clustering approaches characterize each cluster by a set of parameters, the so–called prototype. Given a set $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$ of sample data, the aim of objective function based fuzzy clustering [2] is to determine the prototypes in such a way that the objective function

$$J(X, U, v) = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m d^2(v_i, x_k) \qquad (1)$$

is minimized. u_{ik} stands for the membership degree of datum x_k to cluster i, $d(v_i, x_k)$ is the distance of datum x_k to the cluster i, represented by the prototype v_i. c is the number of clusters which is fixed or can be determined by using a suitable cluster validity measure (see for instance [2, 3, 20]). The choice of the parameter $m > 1$ – the fuzzifier – determines whether the clusters tend to be more crisp or fuzzy, i.e., for $m \to 1$ we have $u_{ik} \to 1$ or $u_{ik} \to 0$, whereas $m \to \infty$ implies $u_{ik} \to 1/c$.

To avoid the trivial minimum $u_{ik} = 0$ for all i, k,

either the probabilistic constraint

$$\sum_{i=1}^{c} u_{ik} = 1 \qquad \text{for all } k \in \{1, \ldots n\} \quad (2)$$

has to be assumed or the term

$$\sum_{i=1}^{c} \eta_i \sum_{k=1}^{n} (1 - u_{ik})^m$$

has to be added to (1) leading to the concept of possibilistic clustering [15]. It is obvious that in order to minimize (1) data points x_k with a small distance $d(v_i, x_k)$ to the cluster i should be assigned a high membership degree whereas data with larger distances should have low membership degrees.

A standard approach to obtain a concrete algorithm is to derive necessary conditions for the membership degrees u_{ik} and the prototypes v_i in order to obtain a (local) minimum of the objective function (1). The clustering algorithm simply starts with a random initialization and updates the u_{ik} and the v_i alternatingly in an iterative procedure. We do not go into the details of the algorithms, since they are not of importance for our considerations in this paper.

The most simple algorithm is the fuzzy c–means algorithm (FCM) whose prototypes are simply the cluster centers in the form of vectors $v_i \in \mathbb{R}^p$ and the distance $d(v_i, x_k)$ of datum x_k to cluster i is the Euclidean distance between x_k and the cluster center. Gustafson and Kessel [5] enriched each prototype with a symmetric, positive definite matrix C_i and compute the distance $d(v_i, x_k)$ by the formula

$$d^2(v_i, x_k) = (\det C_i)^{1/p} \cdot (x_k - v_i)^\top C^{-1} (x_k - v_i).$$

This resulting Gustafson–Kessel algorithm (GK) allows in contrast to the FCM, that is tailored for spherical clusters, the detection of hyperellipsoidal clusters whose axes determined by the eigenvectors of the matrix C_i.

The method designed by Gath and Geva (GG) [3] introduces an additional parameter P_i for each prototype that allows the algorithm in connection with the matrices C_i to adapt to clusters of different sizes. However, the GG is not a proper objective function algorithm, since it is based on a fuzzification of a maximum likelihood estimator.

The prototypes of the fuzzy c–varieties algorithm (FCV) [2] are tuples of the form $(v_i, e_i^{(1)}, \ldots, e_i^{(r)}) \in (\mathbb{R}^p)^{r+1}$ and induce r–dimensional linear varieties

$$\left\{ y \in \mathbb{R}^p \;\middle|\; y = v_i + \sum_{j=1}^{r} t_j e_i^{(j)},\ t_1, \ldots, t_r \in \mathbb{R} \right\},$$

i.e. lines for $r = 1$ and planes for $r = 2$. The distance function

$$d^2(v_i, x_k) = \| x_k - v_i \|^2 - \sum_{j=1}^{r} \left((x_k - v_i)^\top e_i^{(j)} \right)^2 \quad (3)$$

assigns the distance 0 exactly to those data that are lying in the linear variety determined by the prototype.

The fuzzy c–elliptotypes algorithm (FCE) [2] computes the distance as a convex combination of the distance (3) of the FCV and the Euclidean distance between the cluster center v_i and the datum x_k in order to avoid that, for instance for $r = 1$, collinear, separated, short lines are lumped together in one cluster.

For an overview on various fuzzy clustering algorithms see [6].

3 Deriving Rules from Fuzzy Clusters

The principal idea to apply fuzzy clustering in order to derive if–then rules from data is that each cluster induces a rule by projecting the cluster to the corresponding coordinate spaces [11, 20]. The projection of a cluster to the i–th domain is obtained by taking the i–th coordinate of each data point and assigning to it the membership degree of the original data point to the cluster. In this way a discrete fuzzy set is defined on the i–th coordinate space. To extend this fuzzy set to the whole i–th domain, a piecewise linear fuzzy set can be defined on the basis of these discrete points, an enveloping fuzzy set or a suitable approximation by a parameterized fuzzy set like a triangular or trapezoidal membership function can be chosen [20].

In this way a cluster induces the rule

> If ξ_1 is μ_1 and ... and ξ_{p-1} is μ_{p-1} then ξ_p is μ_p.

where μ_i denotes the (extension of the) i–th projection of the considered cluster and $\xi_1, \ldots, \xi_{p-1}$ are input variables and ξ_p is the output variable. In this way a Mamdani–type fuzzy controller is defined [11, 20]. In [19] a method based on the FCE is proposed to derive Takagi–Sugeno type controllers from fuzzy clusters. For the premise in the rule, the fuzzy sets are again defined by the projections of the cluster, whereas the conclusion must now be a (linear) function in the input variables. This function is simply the linear function constituting the prototype of the corresponding cluster. A similar approach to obtain local linear approximations of data is described in [21]. In [1, 7] as well as in [17] Takagi–Sugeno type controllers are generated on the basis of the GK where the linear functions

are defined on the basis of the eigenvalues and eigenvectors of the matrices C_i.

Besides these techniques that are from a mathematical point of view tailored for function approximation or regression, similar methods can be applied to derive classification rules where discrete classes appear in the conclusions of the rules [12].

All these approaches have to face the problem that the fuzzy clustering algorithm yields a 'fuzzy partition'[1] of the product space of all data whereas fuzzy if–then rules are usually defined on the basis of fuzzy partitions of the single domains. This means that in addition to the loss of information caused by the approximation of the discrete fuzzy sets the projection of the fuzzy cluster can lead to unusual fuzzy partitions on the single domains and enforces again a loss of information, since the original fuzzy cluster can not be reconstructed from the fuzzy sets appearing in the if–then rule derived from the cluster.

There are approaches to reduce this loss of information: [13] recommends to restrict to diagonal matrices C_i when using the GK or GG for rule induction. In this way, the fuzzy clusters are forced to be in the form of axis–parallel hyperellipsoids. Since from the projections of the clusters only the smallest hyperbox containing the corresponding hyperellipsoid can be reconstructed, the loss of information is kept smaller in comparison to arbitrary hyperellipsoids. One approach in [20] clusters only the output data and induces the rules by computing the projections to the input domains of the cylindrical extensions of the fuzzy clusters.

Nevertheless, the fuzzy partitions of the single domains cannot be guaranteed to be in the form of usual fuzzy partitions defined by experts.

4 Single Domain Partitions

In the previous section we have seen that although fuzzy clustering is an important contribution to data analysis in general, it is not fully accurate for inducing if–then rules. On the one hand, the shapes of the membership functions tend to be unusual, and on the other hand, fuzzy clustering is designed for partitions of product spaces and not of single domains that are usually considered for fuzzy rules.

In the following we describe two approaches that concentrate on finding fuzzy partitions of the single domains and are therefore well–suited for inducing fuzzy rules. Both approaches construct fuzzy partitions with triangular membership function with the restriction that for each domain at most two supports of different fuzzy sets have a non–empty intersection and the sum of the membership degrees is one at any point of the domain. Although these kinds of fuzzy partitions seem to be very restrictive, they are very reasonable from a semantic point of view [14].

The first approach was developed to construct a Takagi–Sugeno type controller from data with constant functions in the conclusions of the rules. Thus taking the above restrictions for the fuzzy partitions into account, the parameters that have to be determined are the constant output values in the conclusions of the rules and the tips (kernels) of the triangular membership functions in each input domain. The algorithm starts with equidistant tips of the triangular membership function. As error function we simply take the sum of the quadratic difference between the desired outputs given by the data set and the output computed by our controller. In order to minimize this error, the output values in the rules can be determined by a simple linear regression technique when we fix the fuzzy partitions [8].

Now we have to adjust the distances between the tips of the triangular membership functions. For this task we determine for each domain X_ν and each area the error of the regression function. By an area we mean the interval between the tips of two neighbouring triangular membership function.

$$\mathrm{error}(\mathrm{area}_i) \;=\; \sum_{(x_1^{(\ell)},\ldots,x_n^{(\ell)}):x_\nu^{(\ell)}\in \mathrm{area}_i} \Big(y^{(\ell)} - f(x_1^{(\ell)},\ldots,x_n^{(\ell)})\Big)^2 \tag{4}$$

$(x_1^{(\ell)},\ldots,x_n^{(\ell)})$ represent the input values in the data set and $y^{(\ell)}$ the corresponding desired output values. $f(x_1^{(\ell)},\ldots,x_n^{(\ell)})$ denotes the actual output of the fuzzy controller for input $(x_1^{(\ell)},\ldots,x_n^{(\ell)})$. area_i is the interval between the tips of $(i-1)$th and ith triangular membership function in the domain X_ν.

Now the area_i should be contracted, when the error is relatively high, whereas it can be stretched when the error is small. In this way we refine the partitions in these areas where the error is high and can better approximate the function. If L_i^{old} is the length of area_i then we define the new (relative) length of the ith area by

$$L_i^{\mathrm{rel}} \;=\; \frac{L_i^{\mathrm{old}}}{\mathrm{const} + \mathrm{error}(\mathrm{area}_i)}$$

where const is a positive constant that first of all avoids devision by zero when the error is zero for a certain area. const also determines how strong the contraction or stretching of the corresponding area is depending on the error. If const is small in comparison to the error, this will result in drastic changes whereas a large constant allows only very small variations.

[1] We do not want to dive into a discussion on a formal definition of a fuzzy partition. We use this terminology in a naive and intuitive way.

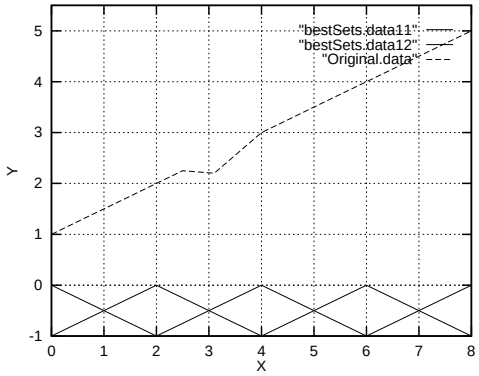

Figure 1: Initial approximation

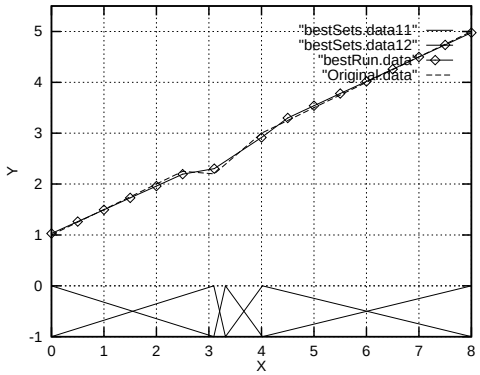

Figure 2: Approximation with adjusted fuzzy partitions

Finally, it is necessary to normalize the relative length of each area so that the overall length is the same as before, i.e., the length of the interval X_ν:

$$L_i \;=\; L_i^{\mathrm{rel}} \cdot \frac{\sum_j L_i^{\mathrm{old}}}{\sum_j L_j^{\mathrm{rel}}}. \qquad (5)$$

In this way, we obtain new fuzzy partitions for the domains and can then compute new constant outputs for the rule by linear regression, derive again new fuzzy partitions and so on. We iterate this procedure until the error of the regression function does not improve sufficiently anymore.

The idea to refine the fuzzy partitions in those areas where the error is relative high is illustrated by a simple example of a piecewise linear function. Figure 1 shows such a function, which would be approximated by a single line when we start the regression with equidistant triangular membership functions. Figure 2 illustrates the good result with adjusted fuzzy partitions. It is interesting to note that the fuzzy sets concentrate on the area where the function is changing its course.

The good results of this approach also for multi–dimensional inputs motivated us to develop a fuzzy clustering algorithm that aims at finding fuzzy partitions for the single domains on the basis of multi–dimensional data. We assume that we are given a data set $\{x_1, \ldots, x_n\} \subset \mathbb{R}^p$. We are looking for fuzzy partitions on the single domains consisting of triangular membership function with the properties mentioned before. This means that we have to determine a suitable grid in the multi–dimensional space. We assume that for each domain the number of triangular membership functions is predefined. We define the membership degree of a data point to a cluster represented by a grid point as the minimum of the membership degrees of the triangular membership functions whose tips are the projections of the grid point. We start the fuzzy clustering with equidistant triangular membership functions on the domains. In order to rearrange the grid, we compute the projections of the data and the membership degrees of these projections to the triangular membership functions. Then we update the triangular membership functions by computing new tips as the cluster centers, i.e., as

$$t_{\mathrm{new}}^{(\nu)} \;=\; \frac{\sum_{k=1}^{n} \mu_{t_{\mathrm{old}}^{(\nu)}}^{m}(x_{k,\nu}) \cdot x_{k,\nu}}{\sum_{k=1}^{n} \mu_{t_{\mathrm{old}}^{(\nu)}}^{m}(x_{k,\nu})}$$

where $t_{\mathrm{new}}^{(\nu)}$ and $t_{\mathrm{old}}^{(\nu)}$ stand for the actualized, respectively old tip and $x_{k,\nu}$ denotes the ν–th projection of datum x_k. $\mu_{t_{\mathrm{old}}^{(\nu)}}$ is the triangular fuzzy set with its tip at $t_{\mathrm{old}}^{(\nu)}$.

The fuzzy sets at the left and right boundary in each dimension require a special treatment, since it is not clear how membership degrees on the left and right side of the left-most, respectively right-most tip of the triangular membership function have to be computed. A very simple approach which does not yield good results would put a triangular fuzzy set at the left and right boundary (given by the smallest, respectively greatest, value of the data in the corresponding dimension) and would not allow to change these fuzzy sets. Therefore, cluster centers of clusters at the boundary would tend to be at the boundary of the clusters and not in their middle. To avoid this effect, we allow changes of the left-most and right-most fuzzy sets in each dimension and extend the triangular membership function in the direction of the corresponding boundary in such a way that the data points at the very boundary obtain a membership degree of 0.5.

Figure 3 shows a result obtained by this grid clustering technique. Each data point is connected to the prototype (grid point) whose associated cluster assigns the greatest membership degree to the data point. (In his case, we have computed the membership degree of a 3–dimensional data point by taking the product of the membership degrees of its coordinates to the corresponding fuzzy sets. Of course, another t–norm than the product is also possible.)

This grid clustering method is of course not an

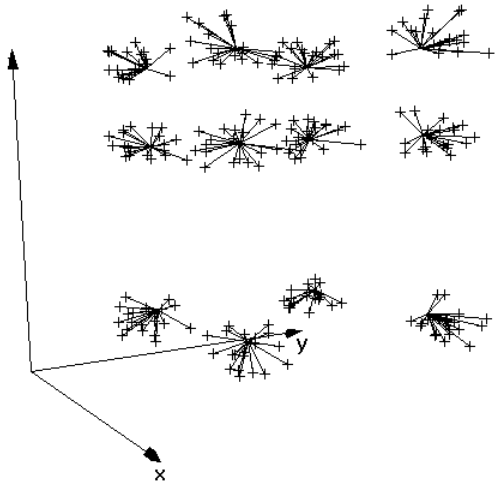

Figure 3: An example of grid clustering

objective function based algorithm, but provides clusters with cluster centers on the grid that are very well suited for rule induction for function approximation as well as for classification tasks. It should be noted that empty clusters, i.e., a cluster corresponding to a grid point whose entourage does not contain any data points, should be neglected when the rules are stated. This means that only those clusters are allowed to induce a rule that are non–empty.

An advantage of this grid clustering method is also that in opposition to the usual probabilistic or possibilistic clustering algorithm, clusters do not have an infinite range, thus data points that are covered by other clusters far away from one cluster do not have any influence on this cluster.

Although we assumed the number of grid points to be fixed, we can also automatically determine their number on the basis of suitable validity measures like they are described in [2, 3, 6, 20]. We determine the number of grid points applying the corresponding validity measure separately in each dimension and optimizing the number of grid points dimension–wise.

References

[1] R. Babuška, H.B. Verbruggen, A New Identification Method for Linguistic Fuzzy Models. Proc. Intern. Joint Conferences of the Fourth IEEE Intern. Conf. on Fuzzy Systems and the Second Intern. Fuzzy Engineering Symposium, Yokohama (1995), 905–912.

[2] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981).

[3] I. Gath, A.B. Geva, Unsupervised Optimal Fuzzy Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (1989), 773–781.

[4] H. Genther, M. Glesner, Automatic Generation of a Fuzzy Classification System Using Fuzzy Clustering Methods. Proc. ACM Symposium on Applied Computing (SAC'94), Phoenix (1994), 180–183.

[5] D.E. Gustafson, W.C. Kessel, Fuzzy Clustering with a Fuzzy Covariance Matrix. Proc. IEEE CDC, San Diego (1979), 761–766.

[6] F. Höppner, F. Klawonn, R. Kruse, Fuzzy–Clusteranalyse: Verfahren für die Bilderkennung, Klassifikation und Datenanalyse (in German). Vieweg, Braunschweig (1997).

[7] U. Kaymak, R. Babuška, Compatible Cluster Merging for Fuzzy Modelling. Proc. Intern. Joint Conferences of the Fourth IEEE Intern. Conf. on Fuzzy Systems and the Second Intern. Fuzzy Engineering Symposium, Yokohama (1995), 897-904.

[8] V. Kecman, B.–M. Pfeiffer, Exploiting the Structural Equivalence of Learning Fuzzy Systems and Radial Basis Function Neural Networks. Proc. Second European Congress on Intelligent Techniques and Soft Computing (EUFIT'94), Aachen (1994), 58–66.

[9] J. Kinzel, F. Klawonn, R. Kruse, Modifications of Genetic Algorithms for Designing and Optimizing Fuzzy Controllers. Proc. IEEE Conference on Evolutionary Computation, IEEE, Orlando (1994), 28–33.

[10] F. Klawonn, Fuzzy Sets and Vague Environments. Fuzzy Sets and Systems 66 (1994), 207–221.

[11] F. Klawonn, R. Kruse, Clustering Methods in Fuzzy Control. In: W. Gaul, D. Pfeifer (eds.), From Data to Knowledge: Theoretical and Practical Aspects of Classification, Data Analysis and Knowledge Organization. Springer–Verlag, Berlin (1995), 195–202.

[12] F. Klawonn, R. Kruse, Derivation of Fuzzy Classification Rules from Multidimensional Data. In: G.E. Lasker, X. Liu (eds.), Advances in Intelligent Data Analysis. The International Institute for Advanced Studies in Systems Research and Cybernetics, Windsor, Ontario (1995), 90–94.

[13] F. Klawonn, R. Kruse, Automatic Generation of Fuzzy Controllers by Fuzzy Clustering. Proc.

1995 IEEE Intern. Conf. on Systems, Man and Cybernetics, Vancouver (1995), 2040–2045.

[14] F. Klawonn, V. Novák, The Relation between Inference and Interpolation in the Framework of Fuzzy Systems. Fuzzy Sets and Systems 81 (1996), 331–354.

[15] R. Krishnapuram, J. Keller, A Possibilistic Approach to Clustering. IEEE Trans. on Fuzzy Systems 1 (1993), 98–110.

[16] R. Kruse, J. Gebhardt, F. Klawonn, Foundations of Fuzzy Systems. Wiley, Chichester (1994).

[17] Y. Nakamori, M. Ryoke, Identification of Fuzzy Prediction Models Through Hyperellipsoidal Clustering. IEEE Trans. Systems, Man, and Cybernetics 24 (1994), 1153-1173.

[18] D. Nauck, F. Klawonn, R. Kruse, Neuronale Netze und Fuzzy–Systeme. Braunschweig (1996) (English translation to be published by Wiley).

[19] T. Runkler, R. Palm, Identification of Nonlinear Systems Using Regular Fuzzy c–Elliptotypes Clustering. Proc. IEEE Intern. Conf. on Fuzzy Systems, New Orleans (1996), 1026–1030.

[20] M. Sugeno, T. Yasukawa, A Fuzzy–Logic–Based Approach to Qualitative Modeling. IEEE Transactions on Fuzzy Systems 1 (1993), 7–31.

[21] Y. Yoshinari, W. Pedrycz, K. Hirota, Construction of Fuzzy Models Through Clustering Techniques. Fuzzy Stes and Systems 54 (1993), 157–165.