Objective Function Based Fuzzy Clustering in Air Traffic Management

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieurin
(Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

Annette Keller

Magdeburg, im November 2002

Objective Function Based Fuzzy Clustering in Air Traffic Management

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieurin
(Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von: Dipl.-Inform. Annette Keller

geb. am 26.10.1972 in Braunschweig

Gutachter:
Prof. Dr. Rudolf Kruse
Prof. Dr. Frank Klawonn
Prof. Dr. Erich Peter Klement

Magdeburg, den 08.11.2002

Danksagung

An erster Stelle möchte ich mich bei Herrn Prof. Dr. Frank Klawonn und Herrn Dr. Johannes Reichmuth bedanken: An jenem Tag, als ich meine Diplomarbeit über Fuzzy-Clusteringmethoden in der Abteilung Luftverkehrssysteme des Instituts für Flugführung am Deutschen Zentrum für Luft- und Raumfahrt in Braunschweig Herrn Dr. Reichmuth vorstellte, existierte nur eine vage Idee, Fuzzy-Clusteringverfahren weiter zu entwickeln und auf Aufgaben im Air Traffic Management anzuwenden.

Meinem Doktorvater Herrn Prof. Dr. Frank Klawonn danke ich in besonderem Maße für die vielen Anregungen, wertvolle Gespräche, fachliche Diskussionen und die Durchsicht der Veröffentlichungen, die im Rahmen dieser Arbeit entstanden sind.

Herr Dr. Reichmuth schaffte es innerhalb kürzester Zeit nach meinem Vorstellungsgespräch, eine Doktorandenstelle am Institut für Flugführung einzurichten und stand mir mit seinen Mitarbeitern jederzeit fachlich zur Seite. Zudem möchte ich mich für das gute Teamwork, das lockere Arbeitsklima und den fruchtbaren Ideenaustausch bedanken. Hierzu haben die Mitarbeiter und anderen Doktoranden der Abteilung Luftverkehrssysteme beigetragen, von denen ich besonders Dr. Thorsten Mühlhausen für die gute Zusammenarbeit bei der Passagierdatenanalyse, Ingrid Gerdes und Franz Knabe für die vielen Informationen über Air Traffic Management, Reinhard Schmitz für die Aufbereitung der Radardaten und Birka von Schmidt für die Zusammenarbeit im Arbeitskreis Softcomputing und die fachlichen Diskussionen nennen möchte.

Besonders bedanken möchte ich mich auch bei Herrn Prof. Dr. Rudolf Kruse, der das Erstreferat für meine Arbeit übernommen hat und sich besonders in der Anfangsphase im Arbeitskreis Softcomputing mit den theoretischen Ideen seiner Doktoranden auseinandergesetzt hat. Herrn Prof. Dr. Erich Peter Klement danke ich für die bereitwillige Übernahme des dritten Gutachtens.

Viele Graphiken dieser Arbeit, die Zugehörigkeitsgrade, Distanzen und Datengewichte veranschaulichen, sind mit dem ’FC’-package von Frank Höppner entstanden.

Neben Vorträgen auf einigen Innereuropäischen Konferenzen, die mir das Institut für Flugführung ermöglicht hat, konnte ich mit einer Förderung der Deutschen Forschungsgemeinschaft (DFG) an der NAFIPS 2000 in Atlanta teilnehmen.

Am meisten habe ich jedoch der Unterstützung meiner Familie besonders meines Mannes Marco Temme zu verdanken, ohne dessen Geduld und Rücksicht auf die privaten Einschränkungen, unsere fachlichen Diskussionen und seine mentale Unterstützung diese Arbeit nicht entstanden wäre.

Braunschweig, November 2002 Annette Keller

Kurzfassung

Ein wichtiger Teil der Datenanalyse ist die Unterteilung von vorgegebenen Daten in Gruppen. Den sogenannten Clusterverfahren liegen mathematische Modelle zu Grunde, die einen Ähnlichkeitsbegriff für die Einteilung der Daten definieren. Fuzzy-Clustering Verfahren ermöglichen nicht nur die Unterteilung der Daten in eine bestimmte Anzahl von Gruppen, sondern bestimmen für jedes einzelne Datum zu jeder Gruppe einen Zugehörigkeitsgrad. Dieser Zugehörigkeitsgrad spiegelt die Repräsentativität des einzelnen Datums zur ganzen Datengruppe wieder. Spricht man von "Objective Function based Fuzzy Clustering", so lässt sich das zu Grunde liegende mathematische Modell in Form einer Bewertungsfunktion beschreiben. Diese Bewertungsfunktion beurteilt die Einteilung der Daten in Teilgruppen unter Berücksichtigung der Zugehörigkeitsgrade und des verwendeten Distanz- bzw. Ähnlichkeitsmaßes. Die Nebenbedingungen, die bei der Berechnung der Zugehörigkeitsgrade berücksichtigt werden, führen zu verschiedenen Clusteringkonzepten. Wenn die Bewertungsfunktion differenzierbar ist, lassen sich über die partiellen Ableitungen notwendige Bedingungen für die Zugehörigkeitsgrade und die anderen Parameter des Clusterverfahrens ermitteln, die für das Distanz- bzw. Ähnlichkeitsmaß verwendet werden. Diese notwendigen Bedingungen werden als Berechnungsvorschriften benutzt, um die Bewertungsfunktion zu optimieren. Die Berechnungsvorschriften liefern einen Algorithmus zur Berechnung der Dateneinteilung, in dem die Gleichungen für die einzelnen Parameter iterativ bestimmt werden.

In dieser Arbeit werden das probabilistische, possibilistische und noise Fuzzy-Clustering Konzept vorgestellt. Dabei handelt es sich um bekannte Bewertungsfunktionen, die mit einer Reihe von Distanz- oder Ähnlichkeitsmaßen kombiniert werden können. Die Grenzen und Probleme dieser Verfahren haben zu der Entwicklung eines neuen Fuzzy-Clusteringkonzeptes geführt, das ebenfalls mit unterschiedlichsten Distanzmaßen kombiniert werden kann. Mit diesem neuen Verfahren lässt sich mit einem der auftretenden Probleme – die Handhabung von Meßfehlern oder Ausreißern innerhalb der Daten – besser umgehen. Dazu wird der Einfluss einzelner Daten auf die Gesamteinteilung während der Clusterberechnung angepasst. Am Beispiel der Analyse von Umsteigern am Flughafen Frankfurt wird gezeigt, wie diese Technik genutzt werden kann, um Ausreißer zu identifizieren und ihren Einfluss auf die Gruppierung zu reduzieren.

Bisher werden in Kombination mit den Bewertungsfunktionen im wesentlichen die euklidische und die mit Hilfe von Matrizen transformierte euklidische Distanz verwendet. Hier werden neue Techniken vorgestellt, mit denen die bekannten Verfahren erweitert und ihnen mehr Flexibilität verliehen werden kann. Die Basisfunktionen werden so modifiziert, dass sich die Clusterverfahren besser an die Struktur der einzelnen Teilgruppen anpassen können. Die einzelnen Cluster können sich dann z.B. an die Form, die Größe, den Einfluss einzelner Attribute oder den Einfluss einer ganzen im Kontext zusammengefassten Gruppe von Attributen anpassen. Am Beispiel der Analyse von Flugradardaten wird deutlich, wie wichtig eine Anpassung der Clustergröße

und -form sein kann. In diesem Beispiel werden die Anflugrouten im Luftraum des Flughafens Zürich analysiert, um Unterschiede zwischen den vorgegebenen und tatsächlich praktizierten Anflugrouten aufzuzeigen. Die Gruppierung von Daten anhand des Vergleichs von Attributgruppen ist besonders für die Bildverarbeitung von Bedeutung. Dabei geht es häufig darum, ähnliche Regionen zu erkennen, wie anhand eines Bildes des Forschungsflugzeuges "ATTAS" des Deutschen Zentrums für Luft- und Raumfahrt e.V. demonstriert wird.

Ein Anwendungsgebiet der Fuzzy-Clustering Verfahren ist die Beschreibung eines Systemverhaltens in Form von Fuzzy-Regeln. Möglichkeiten, aus Clustereinteilungen und den dabei berechneten Zugehörigkeitsgraden Fuzzy-Regeln abzuleiten, werden in dieser Arbeit erläutert. Bei der Analyse der umsteigenden Passagiere wird die Regelgenerierung genutzt, um das Passagierverhalten zu beschreiben.

Auch wenn diese nachträgliche Beschreibung der berechneten Cluster in Form von Fuzzy-Regeln durchaus anwendbar ist, kann damit immer nur eine Näherungslösung für die Einteilung der Daten in Gruppen bestimmt werden. Deshalb wird in dieser Arbeit eine Bewertungsfunktion eingeführt, die eine Einteilung der Daten in Teilgruppen ermöglicht, die äquivalent zu der Beschreibung von Bereichen des Analyseraumes mit Fuzzy-Regeln ist. Da die partiellen Ableitungen dieser Bewertungsfunktion nicht überall existieren, werden ein heuristischer Ansatz und ein auf Evolutionären Algorithmen basierendes Clusterverfahren eingeführt. Mit diesen Verfahren lässt sich eine Clustereinteilung finden, die diese Bewertungsfunktion optimiert und von vornherein darauf ausgelegt ist, eine Beschreibung in Form von Fuzzy-Regeln für den Untersuchungsraum zu finden.

Abstract

Determining a partition of given sample data is an important part in data analysis tasks. Clustering methods use a mathematical model based on a similarity measure to determine a suitable partition of the data set. In fuzzy clustering the data is not only partitioned in a number of subgroups, but each datum is assigned a degree of membership for each subgroup. In this way a representativeness of each datum for the single subgroups is determined during the analysis. In objective function based clustering the mathematical model is stated in form of an objective function that evaluates the partition of data with respect to the membership degrees and the underlying similarity or distance measure. Different assumptions and constraints lead to a variety of basic clustering concepts. If the objective function is differentiable, necessary conditions for the membership degrees and other cluster parameters used in the distance or similarity measure can be derived in order to optimise the objective function. The resulting equations are then alternatingly applied in an algorithm to determine the data partition.

In this work, the basic fuzzy clustering concepts of probabilistic, possibilistic, and noise clustering are explained. These well-known concepts can be applied to a variety of distance or similarity measures. The restrictions and disadvantages of these techniques led to the development of a new basic clustering concept. This approach can be applied to the presented distance or similarity measures and is well suited to handle the problem of outliers within the data set. Therefore, the influence of single data vectors on the partition is estimated and adapted during the clustering procedure. How this technique is used to identify outliers and reduce their influence on a partition is shown for the example of transfer passengers at Frankfurt airport.

Usually the Euclidean or a transformed Euclidean distance is applied as distance measures to the basic clustering concepts. Here, alternative fuzzy clustering algorithms are introduced that expand well-known techniques and give them a greater flexibility. Therefore, modifications of the similarity measures are introduced that enhance the adaptation possibilities of the clustering procedure to the subgroups' structures, e.g. to a structure's volume, the single attribute's influence, or the influence of a whole group of data attributes combined in some context. The importance of volume and shape adaptation becomes obvious for the example of the analysis of flight radar data. In this example, arrival routes of Zurich airport are analysed to point out differences between pre-determined and practiced routes. Using the comparison of attribute groups for clustering is esp. significant for pattern recognition. Often similar regions have to be identified as is shown for an image of the research aircraft "ATTAS" of the German Aerospace Center.

One application field of fuzzy clustering techniques is the derivation of fuzzy rules, e.g. to describe system behaviour. In this work, the possibilities to derive rules from fuzzy clustering results are illustrated. Rule generation is used for the analysis of transfer passenger to describe passenger behaviour.

Although the derivation of fuzzy rules based on clustering results is quite applicable, this form of rule derivation is only an approximation of the clus-

tering partition. Therefore, an objective function that is based on the idea to estimate the data partition in a form of groups related to fuzzy rules is introduced in this work. Since this objective function does not have partial derivatives a heuristic solution and an evolutionary algorithm based fuzzy clustering technique are introduced to cope with this objective function. These techniques are able to find a cluster partition optimising the objective function and are specifically designed for describing a domain of interest in the form of fuzzy rules.

Contents

Chapter 1

Introduction

The idea to use the empirical knowledge of a control engineer to construct a controller lead in the early 70's to the development of fuzzy control [115]. To use the data by itself to derive control rules is an alternative approach, which became more popular in the last years as expert knowledge is not required [73]. Other techniques like evolutionary or neural computation are approaches applied to learning fuzzy rules from data often with the intention to optimise certain parameters of a fuzzy controller.

Fuzzy clustering techniques aim at finding a suitable fuzzy partition for a given data set. For a fuzzy partition a datum is not necessarily assigned to a unique class or cluster, but has membership degrees between zero and one to each cluster. Fuzzy clustering algorithms are applied for various reasons:

- The membership degrees give information about the ambiguity of the classification.
- Fuzzy clustering can adapt to noisy data and classes that are not well separated.
- Since most fuzzy clustering approaches are based on optimising an objective function, membership degrees represent continuous parameters so that a continuous optimisation problem has to be solved.
- Fuzzy clustering can be applied to learning fuzzy rules from data.

The set of cluster parameters, that determine the size and the shape of a cluster, depends on the specific application field. We mainly distinguish between fuzzy clustering as an explorative data analysis method, especially for unsupervised classification tasks, techniques for rule extraction (for instance for fuzzy controllers), and shell clustering algorithms, that are designed for boundary detection in image recognition.

In this work we review objective function-based fuzzy clustering approaches in section 3 in general and demonstrate their applicability to the field of air traffic management. First, we introduce basic concepts including a new clustering approach, especially tailored for noisy data, in section 3.1. These basic approaches rely all on the choice of suitable dissimilarity or distance measures.

Various modifications of the Euclidean distance function have been proposed in order to model different cluster forms.

Standard fuzzy clustering methods like the fuzzy c-means algorithm are based on the idea of optimising an objective function. This objective function depends on the distances of the data to the cluster centres weighted by the membership degrees. By taking the first derivative of the objective function with respect to the cluster parameters, one obtains necessary conditions for the objective function to receive an optimum. These conditions are then applied in an iteration procedure and define a clustering algorithm. Numerous approaches have been developed to detect different forms of cluster shapes in data sets. The more flexible the clustering algorithms are in general, the more they depend on a suitable cluster initialisation. Also with the flexibility of cluster structures the complexity of the proposed algorithms highly increases.

In section 3.2 we review a number of well-known and frequently used distance measures. They can all be combined with the basic clustering approaches presented in section 3.1.

Before presenting several new dissimilarity measures tailored for special clustering tasks, so-called validity measures are reviewed in section 4. All described basic clustering approaches need the number of clusters or subgroups to be predefined for the calculation of a partition. The presented validity measures enable us to evaluate a whole partition of data into subgroups in a way (nearly) independent of the number of clusters. Optimising the validity of a partition for varying number of clusters enables us to determine a (in some way) optimal number.

In the following sections new modifications for the distance measures and their applicability to the basic concepts of section 3.1 on the one hand and for real applications on the other hand are introduced. Again, the validity measures described in section 4 are applied to determine the number of groups.

In section 5.1 we present a new angle-based distance measure that is suitable for data sets with a smaller number of extreme values and a large number of ‘normal’ values. Section 5.2 modifies this approach and we obtain a clustering algorithm to detect lines and (hyper-)planes that can be applied to line recognition as well as to constructing Takagi-Sugeno fuzzy rule systems that describe a function in terms of local linear models.

In section 6 we present an extension that can be applied to well-known simple and fast clustering techniques enabling these to adapt to the cluster sizes without highly increasing the computational effort. One approach no longer considering points as cluster centres but using circles with adapted radii as cluster representatives is illustrated in this section.

Another important question is the influence of certain attributes. In the worst case, some attributes even depend on each other or are just ’randomly’ distributed. In chapter 7 we present a modification of the basic clustering techniques that enables us to determine the importance of certain attributes or variables.

Fuzzy clustering has been shown to be a valuable technique not only for data analysis but also for image recognition, see e.g. [20], [17], [15], [16], or [24]. The methods presented in chapter 3 have been successfully applied to image recognition. In chapter 8 we introduce a new clustering technique especially tailored to identify similar regions in a data set, being essential in image recognition. Our approach does not only compare the regions pixel-wise but can handle predefined sets of attributes within the (dis-)similarity measure. Therefore, the similarity of whole sets rather than single attributes is used for a classification.

Because of the close connection between fuzzy clusters and fuzzy rules, fuzzy clustering seems to be a very promising method for generating rules from data. Intuitively, each if-then rule of a Mamdani-type fuzzy controller or in case of a classification task the rule's premise 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 modelling the linguistic terms appearing in a rule. If, for example, triangular membership functions are used, the coordinates of the tips of the triangles define a vector or point that can be interpreted as a 'typical' point of the control function. Points in the neighbourhood with increasing distance are less 'typical' and therefore have a decreasing membership degree defined by the Cartesian product of the fuzzy sets appearing in the rule. The similarity to many fuzzy clustering strategies is evident: A typical element - usually the cluster centre or prototype - represents the cluster and the membership degree of a datum to the cluster is decreasing with increasing distance, which could even be a transformed distance. In section 9 we review how rules can be extracted from fuzzy clusters and show how the fuzzy clustering techniques presented in this work can be used for rule extraction.

Some drawbacks in rule generation from solid clusters in general lead us to the heuristic clustering approach of grid clustering, introduced in section 9.2. This approach seems to be well-suited for the task of rule learning.

In principal any kind of prototype parameter set and distance function can be chosen in order to have flexible cluster shapes. However, the alternating optimisation scheme can only be applied, when the corresponding distance function is differentiable. But even for differentiable distance functions we usually obtain equations for the prototypes that have no analytical solution (for instance [33]). This means that we have to cope with numerical problems and need in each iteration step a numerical solution of a coupled system of non-linear equations. Other approaches try to optimise the objective function directly by evolutionary algorithms as reviewed in chapter 11. For our grid clustering it is impossible to design a differentiable objective function leading to simple update equations. Therefore, we review some approaches to apply evolutionary strategies to fuzzy cluster analysis in section 11.3 and introduce a clustering algorithm with a non-differential objective function that leads to similar results as the heuristic grid clustering algorithm.

1.1 Data analysis and fuzzy clustering

Fuzzy Clustering is only a small part of data analysis as a whole. It can be categorised as a part of pattern recognition, where data is partitioned or

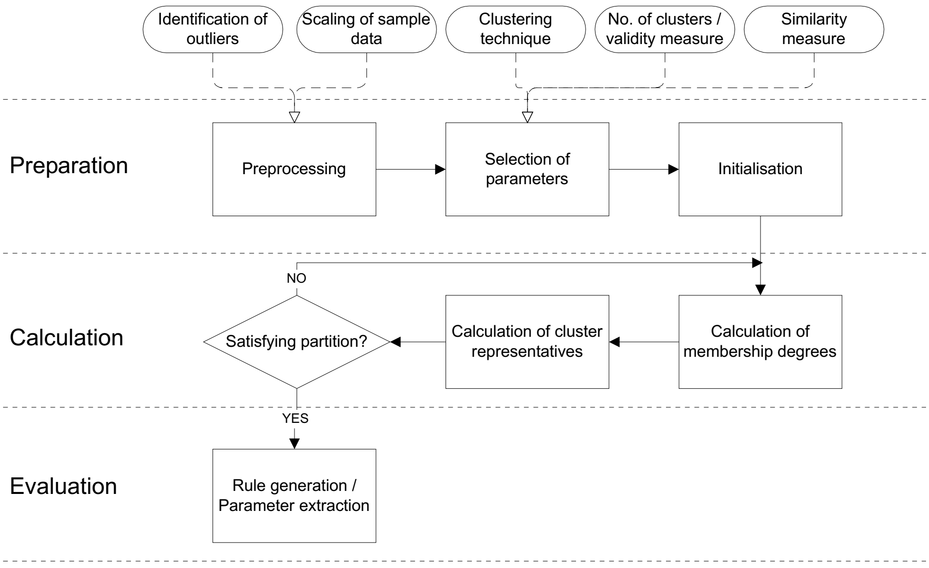
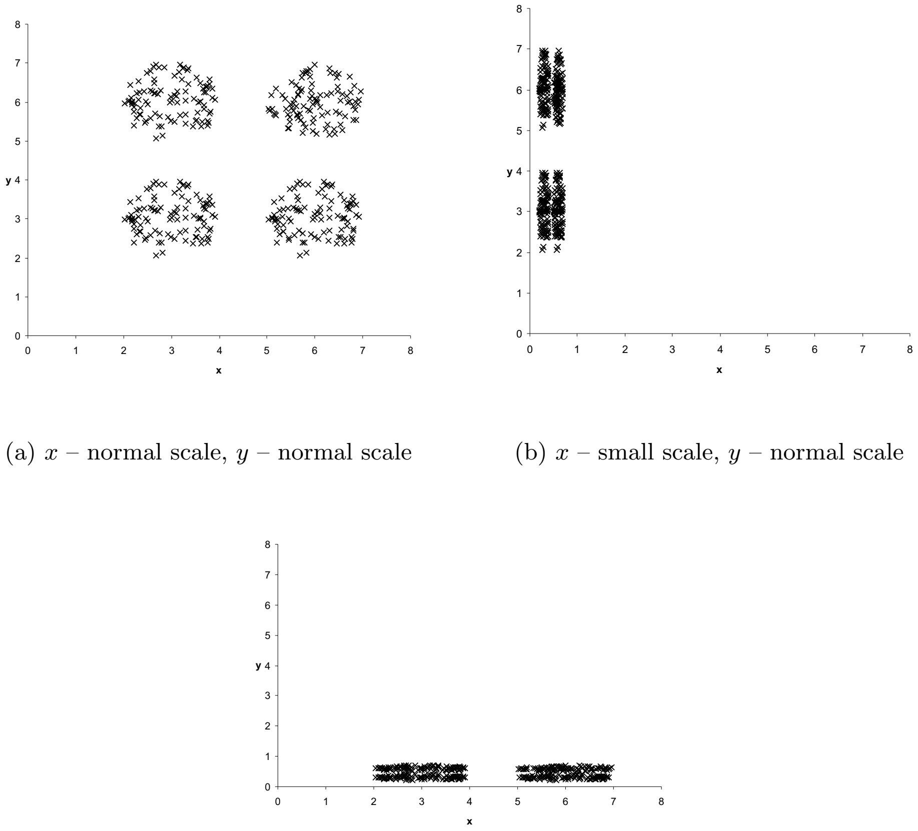

Identification of outliers
Scaling of sample data
Clustering technique
No. of clusters / validity measure
Similarity measure
Preparation
Preprocessing
Selection of parameters
Initialisation
NO
Calculation
Satisfying partition?
Calculation of cluster representatives
Calculation of membership degrees
YES
Evaluation
Rule generation / Parameter extraction

Figure 1.1: Fuzzy clustering and data analysis

grouped without implying a mathematical model. Knowledge extraction e.g. in form of fuzzy rule generation is usually based on previously determined subgroups of the data. Figure 1.1 illustrates the data analysis task using fuzzy clustering to perform the data partition. In this scheme the data analysis process is divided into three phases: preparation, calculation, and evaluation. The preparation phase can be considered as pre-processing for the clustering task. Knowledge about the data set under consideration or statistical pre-processing is used to simplify the fuzzy clustering task.

Some of the described clustering techniques are not robust against outliers and most use distance measures that highly depend on the scaling of the data variables or attributes. The influence of attribute scaling is illustrated in figure 1.2. If we have to partition the example data into two groups, we are not sure how to partition the data in the first figure (1.2(a)). The other two examples in figure 1.2(b) and 1.2(c) would lead to contrary partitions. We have to handle or at least keep in mind these problems if we prepare a data set for clustering tasks.

Once we have finished the data pre-processing, we have to select the fuzzy clustering parameters. A suitable clustering technique as e.g. detection of solid clusters, outliers, or contours has to be chosen. In general, these basic clustering techniques aim at optimising the single clusters, depending on the (gradually) assigned data. Therefore, the number of clusters either has to be predefined or a so-called validity measure has to be specified. On the contrary to the basic clustering scheme, the validity measure is tailored to evaluate the partition of the data as a whole. Carrying out the clustering task for a varying number of clusters and evaluate the resulting partitions with a suitable validity measure helps to identify the 'optimal' number of clusters. Most basic



(a) x – normal scale, y – normal scale

(b) x – small scale, y – normal scale

(c) x – normal scale, y – small scale

Figure 1.2: Influence of attribute scaling

clustering techniques can be combined with varying similarity/dissimilarity measures. The dissimilarity measure often has the form of a distance measure between the cluster representative and a data vector.

The last step of the preparation phase is the initialisation of the fuzzy clustering procedure. We have to determine either an initial gradual partition of the data – so-called membership degrees – or an initial solution for the cluster representatives, often in form of cluster centres. In general it is easier to select single data vectors as cluster centres by chance than to determine suitable membership degrees for the data set without given cluster representatives.

After the initialisation we enter the calculation phase of the selected clustering algorithm. In general we determine alternately the membership degrees of all data to all clusters and the cluster representatives. The alternating optimisation is carried out until we are satisfied with the resulting partition, e.g. when the iterative calculation no longer leads to significant changes in the partition. In the most simple algorithms, the cluster representatives (also called prototypes) consist only of centre vectors. Other parameters that influence the partition may belong to the cluster representative, e.g. attribute, cluster, or data weights, matrices or scaling factors used for the distance measure.

Once the calculation phase has been finished, we can evaluate the partition. It depends on the goal of the clustering task, if further studies based on the clustering parameters are carried out. If we are interested in functional dependencies in our sample data, fuzzy rules might be the form of result we are looking for. Considering the task of image recognition, the combination of data vectors in form of clusters might be the result. Also certain parameters determined during the alternating optimisation could be of interest for further studies or by themselves. Indifferently to the results we are looking for, we have to check and sometimes validate our conclusions.

This process is not only suitable for fuzzy clustering tasks but also for other clustering techniques, see e.g. [90], or even data analysis methods in general. See e.g. [84] where a method to deal with missing values in classification tasks is described. In the following we restrict ourselves to fuzzy clustering, considering crisp data only. For an introduction in fuzzy data analysis see e.g. [10, 9].

1.2 Fuzzy clustering notation and basics

The mathematical problem to divide collected data into meaningful and interpretable subgroups is considered since the beginnings of statistical research. We generally try to build data parts in a way that similar samples are joint together in one group and dissimilar samples are divided into different groups. In a mathematical notation the sample data can be written as $X \in \mathbb{R}^{p \times n}$, where n is the number of collected data and p denotes the number of feature attributes, variables or measured quantities. One feature vector or *datum* is denoted as the vector $x_k \in \mathbb{R}^p$ with $k \in \{1, \ldots, n\}$. With this definition, we can write the data as set of feature vectors: $X = \{x_1, \ldots, x_n\}$. The goal is to partition the sample data into subgroups, so-called *clusters*. They are denoted by $\mathcal{C} = \{\mathsf{v}_1, \ldots, \mathsf{v}_c\}$, with c denoting the number of subgroups. Each cluster representative or *prototype* $\mathsf{v}_i, i \in \{1, \ldots, c\}$ consists of the cluster parameters, generally the centre vector $v_i \in \mathbb{R}^p$ and other parameters, describing the clusters size, form, or influence of attributes. Some parameters as e.g. the *covariance matrix* influence the dissimilarity or *distance measure* $d^2(\mathsf{v}_i, x_k)$. Each data vector $x_k \in X$ is gradually assigned to each prototype $\mathsf{v}_i \in \mathcal{C}$. The grade assigning a datum x_k to a cluster v_i is called *membership degree* and denoted by u_{ik}. The matrix of all membership degrees

$$U = \begin{pmatrix} u_{11} & \ldots & u_{1c} \\ \vdots & \ddots & \vdots \\ u_{n1} & \ldots & u_{nc} \end{pmatrix}$$

determines the partition of the data X into groups $\mathcal{C}$. Classical techniques admit only crisp membership degrees – u_{ik} either 0 or 1, $u_{ik} \in \{0, 1\}$ – whereas in fuzzy clustering all degrees between 0 and 1 are possible, leading to fuzzy membership degrees – $u_{ik} \in [0, 1]$.

In case of fuzzy clustering, often a parameter to influence the fuzziness of the calculated partition is used. This parameter is called *fuzzifier* or *fuzziness index* and denoted by $m \in \mathbb{R}_{>1}$. During the calculation m is used as exponent for the membership degrees u_{ik}.

The task of clustering is to determine a partition where similar data is grouped in one cluster – $u_{ik} \to 1$ for small dissimilarity $d^2(\mathsf{v}_i, x_k)$ – and dissimilar data is partitioned in different subgroups – $u_{ik} \to 0$ for large dissimilarity. In summing up the dissimilarity weighted by the membership degrees we are able to define a general objective function that has to be minimised:

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

In order to avoid the trivial solution $u_{ik} = 0$, additional assumptions have to be made leading to probabilistic (section 3.1.1, [15]), possibilistic (section 3.1.2, [79]), noise (section 3.1.3, [34]), or outlier (section 3.1.4) clustering. Partial derivatives with respect to the prototype parameters and the membership degrees lead to necessary conditions for the objective function to have a minimum. For $m \to 1$, the membership degrees are either 0 or 1, i.e. $u_{ik} \to 0/1$, so the classification tends to be crisp. If $m \to \infty$, then $u_{ik} \to \frac{1}{c}$. In this case each datum is totally split among the clusters and assigned with the same degree to each cluster.

Before we take a closer look at objective function based fuzzy clustering in general, the applications in air traffic management are described in the next chapter.

Chapter 2

Fuzzy Clustering in Air Traffic Management

Several approaches to apply fuzzy clustering and softcomputing techniques to traffic problems and aerospace applications in general have been described in the literature, see e.g. [27, 30, 32, 11].

Increasing mobility of the individual leads to growing flight rates at the main airports around the world. Most airports are located in urban areas where no (or at least not suitable) areas are left to expand the airport. In addition, the ecological risks and the strain on humans of expanding the airport are not tenable in a number of cases. The search for alternative solutions is a very complex task and for each airport individual. Usually the airport is analysed under current conditions to identify critical parts and develop schemes for improvement. The main task is to increase the airport capacity without raising delay times. Thereby security and social as well as ecological problems have to be taken into account.

Increasing traffic at airports without adapting the airports infrastructure leads to more delay. Problems arise not only from security regulations but also from the higher amount of work load for the airport staff in general. The demand for increasing airport capacity leads to the necessity of assistance systems to reduce the human workload. To determine critical components where assistance systems can be effectively introduced, the particular airport has to be analysed and simulated. Therefore, data e.g. flight information data or radar data combined with aircraft type and size, time, weather, and airline dependent information are studied. The results lead to a better understanding of the airport as a whole system and indicate problematic influence factors. The aim of applying clustering techniques is to assist the analyst with mathematical methods not only to focus on the obvious problem factors but indicate additional items to include in airport simulations. In general, the available data is graphically or statistically evaluated and interpreted. The aim is to identify critical technical or operational configurations and introduce new procedures or assistance systems reducing the work load as well as the ecological or social burden.

Two main air traffic management examples have been selected for this work to demonstrate how fuzzy clustering is suited for analysis problems in air traffic management. In the first example we look for rules describing the transfer passenger behaviour. Detecting current flight routes from radar data is the second task described in this work. A third air traffic example illustrates the possibilities of fuzzy clustering techniques for image recognition.

2.1 Analysis of Transfer Passenger Information

One analysis task is based on the airports landside. The airside restricts an airports capacity, but we can only take advantage of the airside capacity limit if landside components are suited to handle the corresponding volume of traffic. Therefore, the passengers have to reach their flight in time and the airport staff has to be able to load, unload, maintain, etc. the aircraft in time, too.

To simulate the airport as a complete airside and landside system the amount of data needed is too huge to build a complete microscopic airport model where each single passenger and staff member would be represented in the simulation. The idea for a total airport simulation system is to use microscopic models in areas of specific interest and complete other regions by macroscopic models. For a new large aircraft it is e.g. of interest, in which way the passengers can be loaded and unloaded in the fastest way, but for the way to and from the gate statistic mean values might be sufficient. The macroscopic models are based e.g. not on single movements of one passenger but of the passenger flow from one terminal area to another. The aim in this case is to generate rules describing the passenger flow in the terminal area. To verify the extracted rules, an interpretable set of rules has to be identified. Using fuzzy clustering methods enables us to extract in a first step a rule system based solely on available data. This way it is possible to identify influence factors that might be underestimated or overlooked by experts. At the moment only standard rules such as "at a hub airport a large number of passengers are transfer passengers" and "long-haul flights carry a significant number of transfer passengers arriving from other airports" are known. The amount of transfer passengers continuing on a short-haul flight depends e.g. on the departure time. Airports usually have detailed statistics about the number of transfer passengers for departing and arriving flights, flight destinations, gates, terminals, and apron positions. For a macroscopic simulation model developed in our department it is useful to have a kind of rule set describing the passenger flow (amount of departing and arriving transfer passengers) in dependence on the flight time, type (long-haul, medium-haul, short-haul), passenger amount, etc. Such a rule set enables us to simulate the effect of changes in the airports landside or airside architecture or in the landside connection of the airport. The amount of transfer passengers implicitly describes the number of passengers arriving from the landside. Such a rule system helps also to identify the interface between land- and airside if additional information e.g. total number of passengers in relation to transfer passengers is available. Flight specific passenger data is usually highly sensitive data in the best case provided by the airlines with restricted access. In addition to the advantages for macroscopic simulation, a more or less general airport dependent rule system is extremely helpful for further airport studies to avoid the use of sensitive data material.

For the rule system under development, it is not suitable to generate strict rules. Vagueness resulting from the influence factors has to be handled. Rules identified by experts are usually not strict. Terms as "large aircraft", "significant amount of passengers", and others are frequently used and sufficient for our macroscopic model. The circumstances under which passengers choose a connecting flight are at least not recorded in the available data material, because they are usually unknown. Also occurrences in the business world – e.g. a strike of an airline carrier – have effects on the amount of transfer passengers but are unpredictable.

Under this considerations we look for a rule system that gives us rules in the form "under certain conditions usually an amount of about $x\%$ passengers are transfer passengers". Some expert knowledge is also available that can be used to develop a corresponding fuzzy rule system. However the aim is to see whether additional parameters – at the moment not considered by experts – influence the transfer passenger behaviour. Therefore, we have chosen to use fuzzy clustering techniques to identify such a rule system, see chapter 10. The chosen clustering techniques have to be suited for the task of rule learning. Having expert knowledge available enables us to determine to what extend the rule system reflects this knowledge.

In this work, the generation of a rule system explaining the transfer passenger rate in dependence on the flight time, destination, and aircraft size is presented. The rule system is developed for departures and arrivals at Frankfurt airport. In future studies these two rule sets are incorporated in an expert system together with available expert knowledge.

2.2 Analysis of Radar Data

Another important task is the analysis of radar data. The radar plots are collected for each aircraft arriving, departing, and flying through. The position – above ground coordinates and height – are stored for each single aircraft. Generally, the two-dimensional data – leaving out height – is visualised for a certain area around the airport to determine frequently used flight paths. These plots are e.g. used to estimate the noise pollution of surrounding urban areas and introduce new routes for noise abatement procedures. However, the flight level of aircraft is important for the separation and highly influences the noise level, so that flight routes including flight height are more informative than only the ground path determined by visual analysis.

For most airports departure and approach procedures define flight routes that have to be used by aircraft in the surrounding airspace. In low traffic as well as high traffic situations the air traffic controller advises the pilot to use a deviating flight pass. Written approach procedures are regularly but seldom updated, see e.g. [29].

In practice also other than the official procedures are used. For airport analysis and capacity studies it is important to know the "state-of-the-art"

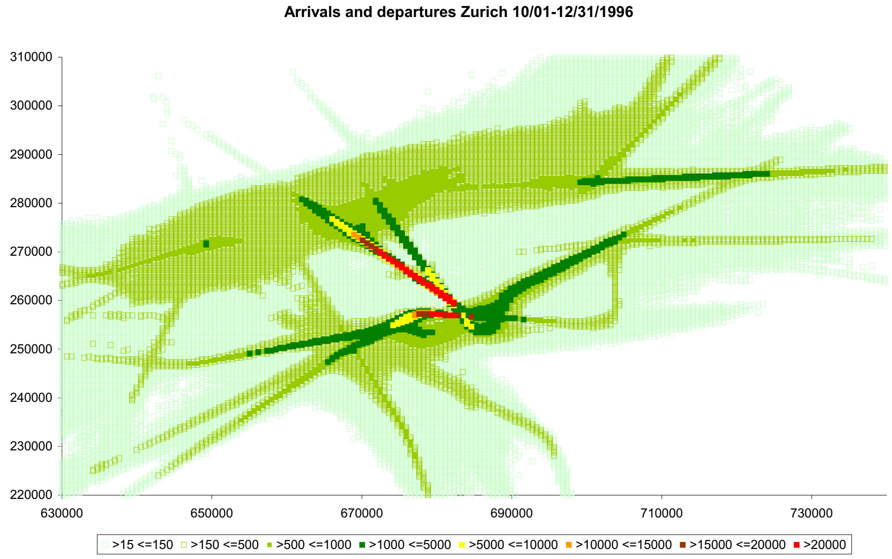


Figure 2.1: Arrivals and departures at Zurich airport

airport procedures to validate the simulations and derive improved procedures. Commonly the flight paths of approaching and departing aircraft are evaluated. Therefore, radar data giving the position of an aircraft is recorded every four seconds. Often the data is analysed visually. The ground area around an airport is divided into small segments and the number of aircraft recorded in each segment is counted for a fixed time period. Then the segments are coloured to show the frequency of aircraft in each segment for the analysed time period. An example of such an analysis is shown in figure 2.2 for the Zurich airport with data from 1996. A problem for capacity analysis of this method is the neglect of flight height. The airspace is divided in height-levels where aircraft are allowed to act independently of each other. The visual analysis and description of 3-dimensional flight routes is often impossible.

For this application, the aim of this work is to use a method that is able to detect line segments in the 3-dimensional elliptical data clouds with different sizes that are described in form of radar data. Therefore, the size-adaptable centre-based clustering technique combined with the algorithm introduced by Gustafson and Kessel [48] – introduced in section 6.1 as GK-sized – is used to extract flight route segments, see section 6.5. This technique is suited to detect elliptical structures of different sizes. The advantage of a fuzzy clustering technique in this case is the possibility of recorded data points to belong to more than one of the identified ellipsoidal structures. Since the aim is to detect route segments these segments usually overlap for continuous flight procedures. For two overlapping parts it is necessary to incorporate the radar points in the two elliptical structures describing the route segments. Once we have obtained a description of flight routes, it can be used for further studies to identify main flight paths and improve the determination of noise polluted areas as well as further airport simulations. This work is restricted to the demonstration how flight routes can be identified with fuzzy clustering techniques. Further studies at the German Aerospace Center in cooperation with airport industry are carrying out pre-processing and deeper analysis of the results described in this work.

Chapter 3

Objective Function Based Fuzzy Clustering

In this chapter the principle ideas of objective function based fuzzy clustering and some well known and often applied techniques are described. For a more thorough overview on fuzzy clustering see for example [51, 15]. Most objective function based fuzzy clustering algorithms aim at minimising an objective function that evaluates the partition of data into a given number of clusters.

Other clustering approaches based on this general objective function clustering concept have been developed by different authors for special tasks. One problem that often has to be handled in business management is the interconnection between the data groups. In [31] an objective function based clustering technique has been developed that extends the general objective function by a binary relation representing cluster interconnections. The problem of incomplete data and solutions to handle missing values are e.g. described in [50].

3.1 Basic Objective Functions

Before discussing several special clustering techniques, general forms of objective functions for fuzzy clustering are introduced that still depend on the choice of a suitable distance measure. Two very common basic clustering techniques are *probabilistic* and *possibilistic clustering* . Both depend on a distance or dissimilarity measure weighted by the membership degrees. Probabilistic clustering [15] uses a constraint ensuring that all data points totally belong to the partition, whereas possibilistic clustering [79] considers outliers with small membership degrees to all groups of data. A third approach related to possibilistic clustering is called *noise clustering* [34]. The idea of this approach is to assign outliers to a special group of data called noise cluster and reduce the influence of this group on the whole partition. Selim and Ismail [101] introduced other approaches to avoid the drawback of probabilistic clustering. They suggest to let a datum belong to a maximum number of clusters, to set the membership degrees to zero if a predefined maximal distance is exceeded, or to define a minimum threshold for the membership degrees. At last, a new approach related to noise clustering is presented. In this approach weights for each datum are adapted during the clustering that indicates if single data points can be seen as outliers. This approach is called *fuzzy clustering with outliers*.

3.1.1 Probabilistic Clustering

In case of probabilistic fuzzy clustering the objective function is of the form

$$J^{prob}(X,U,\mathsf{v}) = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^{m} \cdot d^{2}(\mathsf{v}_i, x_k), \tag{3.1}$$

where $X = \{x_1, \ldots, x_n\} \in \mathbb{R}^{n \times p}$ is the data set, n the number of data points, c denotes the number of fuzzy clusters, $u_{ik} \in [0,1]$ is the membership degree of datum x_k to cluster i, v_i is the prototype or the vector of parameters for cluster i, and $d(\mathsf{v}_i, x_k)$ is the distance between prototype v_i and datum x_k. The parameter $m \in \mathbb{R}_{>1}$ is called *fuzziness index* . For $m \to 1$ the clusters tend to be crisp, i.e. either $u_{ik} \to 1$ or $u_{ik} \to 0$, for $m \to \infty$ we have $u_{ik} \to 1/c$, i.e. the same membership degree for all data to all clusters. Often a value about 2 is chosen for m.

To avoid the trivial solution in minimising the objective function 3.1 that all membership degrees u_{ik} are 0, constraints have to be taken into account. In this case the constraints are

$$\sum_{k=1}^{n} u_{ik} > 0 \qquad \text{for all } i \in \{1, \ldots, c\} \tag{3.2}$$

and

$$\sum_{i=1}^{c} u_{ik} = 1 \qquad \text{for all } k \in \{1, \ldots, n\}. \tag{3.3}$$

Constraint (3.2) guarantees that only non-empty clusters are admitted in the partition. Constraint (3.3) ensures that the sum of all membership degrees for one datum equals 1. This can be interpreted as ”each datum is fully divided among the clusters and belongs totally to the partition of the data set”.

Theorem 3.1 (Probabilistic membership degrees)

Differentiating (3.1) and taking the constraints into account by a Lagrange function leads to the necessary condition

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d^2(\mathsf{v}_i, x_k)}{d^2(\mathsf{v}_j, x_k)}\right)^{\frac{1}{m-1}}} \tag{3.4}$$

for (3.1) to have a (local) minimum if the distance between datum x_k and prototype v_i is not 0, i.e. $d^2(\mathsf{v}_i, x_k) \neq 0$. Otherwise the datum x_k has to be *shared equally among the cluster centres whose prototypes have distance* 0 *to* x_k*:*

$$\text{if } d^2(\mathsf{v}_i, x_k) \;=\; 0 \text{ then } \begin{cases} u_{ik} = \frac{1}{|\mathfrak{I}_x|} & \text{for } \mathsf{v}_i \in \mathfrak{I}_x, \\ u_{ik} = 0 & \text{else} \end{cases} \tag{3.5}$$

where $\mathfrak{I}_x = \{\mathsf{v}_j \mid d^2(\mathsf{v}_j, x_k) = 0 \text{ and } j \in \{1, \ldots, c\}\}$*. Normally the distance of a datum is only* 0 *to one cluster. Otherwise the prototypes of at least two clusters and therefore the clusters itself would be identical. Identical clusters or groups of data are meaningless in case of data partitions. The proof for non-zero distances* $d^2(\mathsf{v}_i, x_k)$ *is shown in Proof 3.1. The constraint 3.2 is fulfilled by equation 3.4 as long as not all data points have distance* 0 *to the prototype of one cluster.*

Proof 3.1 (Probabilistic membership degrees)

Considering constraint 3.3 leads to the Lagrange function

$$J_\lambda^{prob}(X, U, \mathsf{v}) \;=\; \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^m \cdot d^2(\mathsf{v}_i, x_k) - \sum_{k=1}^{n} \lambda_k \cdot \left(\sum_{i=1}^{c} u_{ik} - 1\right).$$

Calculating the partial derivative w.r.t. u_{ik} *leads to*

$$\frac{\partial J_\lambda^{prob}(X, U, \mathsf{v})}{\partial u_{ik}} \;=\; m \cdot u_{ik}^{m-1} \cdot d^2(\mathsf{v}_i, x_k) - \lambda_k \overset{!}{=} 0$$

and therefore

$$u_{ik} \;=\; \left(\frac{\lambda_k}{m \cdot d^2(\mathsf{v}_i, x_k)}\right)^{\frac{1}{m-1}}. \tag{1}$$

Using constraint 3.3 gives us

$$\begin{aligned} 1 &= \sum_{j=1}^{c} u_{jk} \\ &= \sum_{j=1}^{c} \left(\frac{\lambda_k}{m \cdot d^2(\mathsf{v}_j, x_k)}\right)^{\frac{1}{m-1}} \\ &= \left(\frac{\lambda_k}{m}\right)^{\frac{1}{m-1}} \cdot \sum_{j=1}^{c} \left(\frac{1}{d^2(\mathsf{v}_j, x_k)}\right)^{\frac{1}{m-1}} \end{aligned}$$

and therefore

$$\left(\frac{\lambda_k}{m}\right)^{\frac{1}{m-1}} \;=\; \frac{1}{\sum_{j=1}^{c} \left(\frac{1}{d^2(\mathsf{v}_j, x_k)}\right)^{\frac{1}{m-1}}}.$$

with (1) *we receive*

$$\begin{aligned} u_{ik} &= \frac{1}{\sum_{j=1}^{c}\left(\frac{1}{d^2(\mathsf{v}_j,x_k)}\right)^{\frac{1}{m-1}}} \cdot \left(\frac{1}{d^2(\mathsf{v}_i,x_k)}\right)^{\frac{1}{m-1}} \\ &= \frac{1}{\sum_{j=1}^{c}\left(\frac{d^2(\mathsf{v}_i,x_k)}{d^2(\mathsf{v}_j,x_k)}\right)^{\frac{1}{m-1}}} \quad \diamond \end{aligned}$$

Therefore, equation (3.4) is used in an iteration procedure for updating the membership degrees u_{ik}. If a suitable distance function and parameter form is chosen, equations for the prototypes can be derived analogously, assuming the membership degrees are fixed. The alternating optimisation scheme starts with a random initialisation and applies the equations for the u_{ik} and the prototypes until the norm of the membership matrices $\left(U^{old}\right)$ and (U^{new}) in two succeeding iterations is smaller than a given bound ε. The basic algorithm scheme is shown in Algorithm 3.1.

Algorithm 3.1 (Basic Probabilistic Case)

```
Choose()
{
   $m \in \mathbb{R}_{>1}$;
   $c \in \{2, \ldots, n-1\}$;
   $\epsilon > 0$;
}

Initialise()
{
   $\mathsf{v}_i$ for all $i \in \{1, \ldots, c\}$;
   for all $i \in \{1, \ldots, c\} \wedge k \in \{1, \ldots, n\}$
      CalculateMembership $(u_{ik}^{(new)})$;
}

CalculateMembership($u_{ik}$)
{
   $\mathfrak{I}_x := \{\mathsf{v}_j \mid d^2(\mathsf{v}_j, x_k) = 0 \wedge j \in \{1, \ldots, c\}\}$;
   $u_{ik} := \begin{cases} \frac{1}{|\mathfrak{I}_x|} & \text{if } \mathsf{v}_i \in \mathfrak{I}_x \wedge \mathfrak{I}_x \neq \varnothing; \\ 0 & \text{if } \mathsf{v}_i \notin \mathfrak{I}_x \wedge \mathfrak{I}_x \neq \varnothing; \\ \frac{1}{\sum_{j=1}^{c} \left( \frac{d^2(\mathsf{v}_i, x_k)}{d^2(\mathsf{v}_j, x_k)} \right)^{\frac{1}{m-1}}} & \text{if } \mathfrak{I}_x = \varnothing; \end{cases}$
}

CalculatePartition()
{
   do
   {
      for all $i \in \{1, \ldots, c\}$
         Calculate($\mathsf{v}_i$);
      for all $i \in \{1, \ldots, c\} \wedge k \in \{1, \ldots, n\}$
         $u_{ik}^{(old)} := u_{ik}^{(new)}$;
      for all $i \in \{1, \ldots, c\} \wedge k \in \{1, \ldots, n\}$
         CalculateMembership($u_{ik}^{(new)}$);
   }while $\left( \left( \sum_{i=1}^{c} \sum_{k=1}^{n} \mid u_{ik}^{(new)} - u_{ik}^{(old)} \mid \right) < \epsilon \right)$;
}
```

3.1.2 Possibilistic Clustering

In probabilistic clustering the strong constraint (3.3) possibly leads to undesirable membership degrees of some data. Assume a data point in great distance to all clusters exists in the data set. This outlier would be assigned approximately the same membership degree $\frac{1}{c}$ to all c clusters and therefore would have a greater influence on the partition than desired. This effect is illustrated in Figure 3.1. The two groups of data points denote two clusters with the red dots as cluster centre and the points between and above the data groups as outliers. In probabilistic clustering, all outliers would be assigned a membership degree of about 0.5 to both clusters. In case of the points between both circles such a membership degree can be interpreted in the way that it is not possible to assign the point clearly to one cluster, but it belongs as an outlier to the partition. For the points above and beneath the circles a small membership degree to both clusters would help to identify this point as noise.

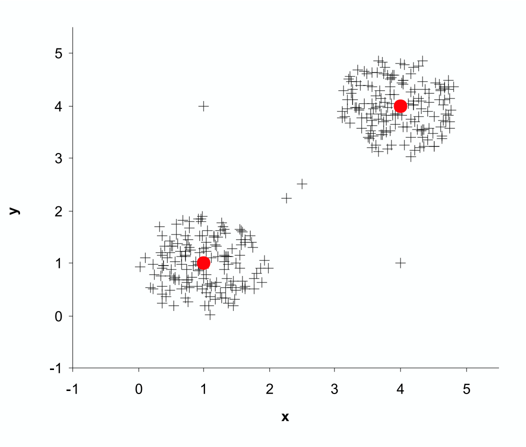


Figure 3.1: Two circular groups with outliers

To avoid such a drawback the approach of *possibilistic clustering* [79] was introduced with remaining constraint (3.2) but released constraint (3.3):

$$\sum_{i=1}^{c} u_{ik} \; > \; 0 \qquad \text{for all } k \in \{1, \ldots, n\}. \tag{3.6}$$

With these constraints the membership degree u_{ik} could be interpreted as a degree of representativeness of datum x_k for cluster i. To avoid the trivial solution all $u_{ik} \to 0$ by minimising equation (3.1) considering constraint (3.6) the objective function has to be modified as well (3.7).

$$J^{poss}\left(X, U, \mathsf{v}\right) \; = \; \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^{m} \cdot d^{2}\left(\mathsf{v}_i, x_k\right) \; + \; \sum_{i=1}^{c} \eta_i \; \sum_{k=1}^{n} \left(1 - u_{ik}\right)^{m} \tag{3.7}$$

The additional parameter $\eta_i > 0$ determines the permissible extension of cluster i.

Theorem 3.2 (Possibilistic membership degrees)

Differentiating (3.7) considering the constraints (3.6) and (3.2) leads to

$$u_{ik} = \frac{1}{1 + \left(\frac{d^2(\mathsf{v}_i, x_k)}{\eta_i}\right)^{\frac{1}{m-1}}}. \tag{3.8}$$

Proof 3.2 (Possibilistic membership degrees)

Differentiating the possibilistic objective function

$$J^{poss}(X, U, \mathsf{v}) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^m \cdot d^2(\mathsf{v}_i, x_k) + \sum_{i=1}^{c} \eta_i \cdot \sum_{k=1}^{n} (1 - u_{ik})^m$$

with respect to u_{ik} leads to

$$\frac{\partial J^{poss}(X, U, \mathsf{v})}{\partial u_{ik}} = m \cdot u_{ik}^{m-1} \cdot d^2(\mathsf{v}_i, x_k) - \eta_i \cdot m \cdot (1 - u_{ik})^{m-1} \overset{!}{=} 0$$

and therefore

$$u_{ik}^{m-1} \cdot d^2(\mathsf{v}_i, x_k) = \eta_i \cdot (1 - u_{ik})^{m-1}.$$

This gives us

$$\frac{d^2(\mathsf{v}_i, x_k)}{\eta_i} = \left(\frac{1 - u_{ik}}{u_{ik}}\right)^{m-1} = \left(\frac{1}{u_{ik}} - 1\right)^{m-1}$$

leading to

$$\frac{1}{u_{ik}} = \left(\frac{d^2(\mathsf{v}_j, x_k)}{\eta_i}\right)^{\frac{1}{m-1}} + 1.$$

So we finally obtain

$$u_{ik} = \frac{1}{\left(\frac{d^2(\mathsf{v}_j, x_k)}{\eta_i}\right)^{\frac{1}{m-1}} + 1} \qquad \diamond$$

Constraint (3.6) is fulfilled, since $u_{ik} \to 0$ only if $d^2(\mathsf{v}_i, x_k) \to \infty$ or $\eta_i \to 0$. To illustrate the influence of η_i, assume $\eta_i = d^2(\mathsf{v}_i, x_k)$. The resulting membership degrees are $u_{ik} = \left(1 + 1^{\frac{1}{m-1}}\right)^{-1} = 0.5$. Defining a membership degree of 0.5 as lower bound for assigning a data point x_k to cluster i gives parameter η_i the mentioned meaning. The permissible extension of cluster i is in some way defined by η_i. If the cluster shapes are known in advance, η_i could be estimated for all $i = 1, \ldots, c$ easily. Otherwise additional assumptions have to be made. One possible approach is to assume clusters containing about the same number of data points and estimate

$$\eta_i = \frac{\sum_{k=1}^{n} u_{ik}^m \cdot d^2(\mathsf{v}_i, x_k)}{\sum_{k=1}^{n} u_{ik}^m}. \tag{3.9}$$

Together with this approach, Krishnapuram and Keller [79] have also proposed other methods to estimate the parameters η_i.

One has to be careful with the possibilistic clustering algorithm in choosing a suitable parameter m. In general, for a relatively large value for m – $m \to \infty$ – the membership degrees of each datum tend towards 0.5 to each cluster in calculating the membership degrees. This effect can be observed in probabilistic as well as possibilistic clustering, but probabilistic clustering seems to be more robust regarding the fuzziness index. In the worst case, no realistic partition of the data is carried out any more in possibilistic clustering. In most cases a value for the fuzzifier between 1.0 and 2.0 will work.

As first described by Davé [34], the clusters are independent of each other in possibilistic clustering since the membership degrees only depend on the single clusters distances. Here the predefined number of clusters c is more comparable to an upper bound of groups than the exact number. In probabilistic clustering the groups 'compete' with each other for the data. Therefore, we first carry out an initialisation with the probabilistic clustering algorithm before starting the possibilistic clustering. The basic algorithm scheme is shown in Algorithm 3.2.

Algorithm 3.2 (Basic Possibilistic Case)

```
Choose()
{
  $m \in \mathbb{R}_{>1}$;
  $c \in \{2, \dots, n-1\}$;
  $\epsilon > 0$;
  $count := 1$;
}
Initialise()
{
  for all $i \in \{1, \dots, c\} \wedge k \in \{1, \dots, n\}$
    $u_{ik}^{(new)} := u_{ik}$ result from Algorithm 3.1;
}

CalculatePartition()
{
  for ($count \leq 2$)
  {
    for all $i \in \{1, \dots, c\}$
      $\eta_i := \frac{\sum_{k=1}^{n} \left(u_{ik}^{(new)}\right)^m \cdot d^2(\mathsf{v}_i, x_k)}{\sum_{k=1}^{n} \left(u_{ik}^{(new)}\right)^m}$ ;
    do
    {
      for all $i \in \{1, \dots, c\}$
        Calculate($\mathsf{v}_i$);
      for all $i \in \{1, \dots, c\} \wedge k \in \{1, \dots, n\}$
        $u_{ik}^{(old)} := u_{ik}^{(new)}$;
      for all $i \in \{1, \dots, c\} \wedge k \in \{1, \dots, n\}$
        $u_{ik}^{(new)} := \frac{1}{\left(\frac{d^2(\mathsf{v}_i, x_k)}{\eta_i}\right)^{\frac{1}{m-1}} + 1}$ ;
    }while $\left(\left(\sum_{i=1}^{c} \sum_{k=1}^{n} \mid u_{ik}^{(new)} - u_{ik}^{(old)} \mid\right) < \epsilon\right)$ ;
    $count := count + 1$;
  }
}
```

3.1.3 Noise Clustering

Possibilistic clustering is one approach to deal with noisy data. Another related technique is called *noise clustering*, see e.g. [34, 36, 35, 102] and the references therein. The principle idea is to add one noise cluster to the set of clusters. Since the objective function considers only the distance function and the membership degrees, the noise cluster can be represented by the weighted membership degrees of the data to this cluster. The second term in equation (3.10) expresses the noise cluster.

$$J^{noise}\left(X,U,\mathsf{v}\right) = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^{m}\, d^{2}\left(\mathsf{v}_i, x_k\right) + \sum_{k=1}^{n} \delta^{2}\left(1-\sum_{i=1}^{c} u_{ik}\right)^{m} \tag{3.10}$$

Parameter $\delta >> 0$ has to be chosen in advance and is supposed to be the (large) constant distance of each datum to the noise cluster.

Theorem 3.3 (Noise clustering membership degrees)

As in possibilistic clustering the constraint (3.2) respectively (3.6) has to be considered in order to derive equations for the membership degrees

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d^{2}(\mathsf{v}_i,x_k)}{d^{2}(\mathsf{v}_j,x_k)}\right)^{\frac{1}{m-1}} + \left(\frac{d^{2}(\mathsf{v}_i,x_k)}{\delta^{2}}\right)^{\frac{1}{m-1}}} \tag{3.11}$$

as necessary conditions for (3.10) to have a minimum. How the membership degrees are derived is explained in proof 3.3.

Proof 3.3 (Noise clustering membership degrees)

$$J^{noise}\left(X,U,\mathsf{v}\right) = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^{m}\cdot d^{2}\left(\mathsf{v}_i, x_k\right) + \sum_{k=1}^{n} \delta^{2}\cdot\left(1-\sum_{i=1}^{c} u_{ik}\right)^{m}$$

where δ^2 is the constant distance of each datum x_k to the noise cluster $c+1$ with cluster centre $\mathsf{v}_{(c+1)}$. Defining this distance and corresponding membership degrees gives us

$$d^{2}(\mathsf{v}_{(c+1)}, x_k) = \delta^{2} \text{ and } u_{(c+1)k}^{m} = \left(1-\sum_{i=1}^{c} u_{ik}\right)^{m},$$

leading to

$$J^{noise}\left(X,U,\mathsf{v}\right) = \sum_{k=1}^{n}\sum_{i=1}^{c+1} u_{ik}^{m}\cdot d^{2}\left(\mathsf{v}_i, x_k\right).$$

Equivalent to proof 3.1 the membership equations can be derived

$$u_{ik} = \frac{1}{\sum_{j=1}^{c+1}\left(\frac{d^2(\mathsf{v}_i,x_k)}{d^2(\mathsf{v}_j,x_k)}\right)^{\frac{1}{m-1}}} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d^2(\mathsf{v}_i,x_k)}{d^2(\mathsf{v}_j,x_k)}\right)^{\frac{1}{m-1}} + \left(\frac{d^2(\mathsf{v}_i,x_k)}{\delta^2}\right)^{\frac{1}{m-1}}} \diamond$$

An interesting result is that

$$\sum_{i=1}^{c} u_{ik} < 1 \qquad \text{for all } k \in \{1,\dots,n\} \tag{3.12}$$

in general, unless $x_k = v_i$ for some i. This illustrates that each datum belongs at least with a small membership degree to the noise cluster. In 1984 Ohashi [95] already made an attempt to consider noise in data. Davé and Krishnapuram [35] showed that the minimisation of Ohashi's objective function is equivalent to the presented approach introduced by Davé [34].

Parameter δ is often estimated as follows

$$\delta^2 = \frac{2}{c \cdot n} \cdot \left(\sum_{k=1}^{n}\sum_{i=1}^{c} d^2(\mathsf{v}_i, x_k)\right).$$

This estimation is used in all presented examples.

In the section 3.2, the difference between noise, possibilistic, and outlier clustering becomes clear in the examples. Figure 3.3 in section 3.2.1 shows the membership degrees for a data set with two spherical clusters and normal distributed data. We can see the problems that arise with probabilistic clustering (the sum of the membership degrees of one datum to all clusters equals one) and the tendency of the other clustering techniques to classify data points at the outer border of the two data groups as outliers. If we assume data points with a membership degree greater than 0.5 to a cluster v to belong to that cluster, we see that in possibilistic clustering a non-negligible number of data does not belong to the partition at all. In noise clustering the membership degrees tend to be smaller as in probabilistic clustering but larger as in possibilistic clustering, reducing both drawbacks. Fuzzy clustering with outliers is comparable to noise clustering, if we see the overall membership degrees as the original membership degrees weighted by the adapted weighting factors. The weights can also be directly used to determine potential outliers. Nevertheless, it depends on a particular data set to choose an appropriate basic fuzzy clustering technique.

The basic algorithm for noise clustering is shown in Algorithm 3.3.

Algorithm 3.3 (Basic Noise Clustering Case)

```
Choose()
{
  $m \in \mathbb{R}_{>1}$;
  $c \in \{2, \ldots, n-1\}$;
  $\epsilon > 0$;
  $\delta \gg 0$;
}

Initialise()
{
  $\mathsf{v}_i$ for all $i \in \{1, \ldots, c\}$;
  for all $i \in \{1, \ldots, c\} \wedge k \in \{1, \ldots, n\}$
    CalculateMembership ($u_{ik}^{(new)}$);
}

CalculateMembership($u_{ik}$)
{
  $\mathfrak{I}_x := \{\mathsf{v}_j \mid d^2(\mathsf{v}_j, x_k) = 0 \wedge j \in \{1, \ldots, c\}\}$;
  $u_{ik} := \begin{cases} \frac{1}{|\mathfrak{I}_x|} & \text{if } \mathsf{v}_i \in \mathfrak{I}_x \wedge \mathfrak{I}_x \neq \varnothing; \\ 0 & \text{if } \mathsf{v}_i \notin \mathfrak{I}_x \wedge \mathfrak{I}_x \neq \varnothing; \\ \frac{1}{\sum_{j=1}^{c} \left(\frac{d^2(\mathsf{v}_i, x_k)}{d^2(\mathsf{v}_j, x_k)}\right)^{\frac{1}{m-1}} \left(\frac{d^2(\mathsf{v}_i, x_k)}{\delta^2}\right)^{\frac{1}{m-1}}} & \text{if } \mathfrak{I}_x = \varnothing; \end{cases}$
}

CalculatePartition()
{
  do
  {
    for all $i \in \{1, \ldots, c\}$
      Calculate($\mathsf{v}_i$) ;
    for all $i \in \{1, \ldots, c\} \wedge k \in \{1, \ldots, n\}$
      $u_{ik}^{(old)} := u_{ik}^{(new)}$;
    for all $i \in \{1, \ldots, c\} \wedge k \in \{1, \ldots, n\}$
      CalculateMembership($u_{ik}^{(new)}$);
  }while $\left(\left(\sum_{i=1}^{c} \sum_{k=1}^{n} \mid u_{ik}^{(new)} - u_{ik}^{(old)} \mid\right) < \epsilon\right)$ ;
}
```

3.1.4 Fuzzy Clustering with Outliers

In this section a modified objective function with an additional weighting factor for each datum is introduced. This approach is related to the former described basic objective functions for probabilistic, possibilistic, and noise clustering. The aim of our approach is not only to assign fuzzy membership degrees to the data points, but also to determine a kind of representativeness of each datum for the whole data distribution. This technique is especially suited to detect data or regions in the sample data that are not well covered by the actual classification. Relatively rare extreme situations are often critical in case of control tasks. The approach presented here enables the expert to determine and separate the critical data from the whole sample data to further study each part separately from each other. In the following we refer to this approach as *outlier clustering*. We have presented this approach in [54]. Other approaches to deal with outliers in case of fuzzy models have been presented in [13, 35, 36, 63]. Clustering algorithms to deal with data belonging to some but not necessarily all clusters are introduced in [101].

Also, outliers could have a disrupting effect on a cluster calculation. If we are able to reduce the influence of outliers in the classification task the resulting methods would be more robust.

Some approaches to add weighting factors to the single data points in order to take the number of data points per cluster into account have been studied in [106]. Here, the aim is not to weight the number of data points per cluster but to assign a kind of influence factor to the single data points. In that sense this approach is related to noise clustering, presented in the previous section. In adapting the weight during the clustering procedure, we are able to detect outliers in the data set.

Since no special distance measure is used here, this approach can be seen as a general basic clustering algorithm. This concept is a modification of the probabilistic fuzzy clustering scheme, described in section 3.1.1. Only an additional weighting factor is added to the probabilistic objective function, as can be seen in

$$J^{outlier}(X, U, \mathsf{v}) \;=\; \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m} \cdot \frac{1}{\omega_k^q} \cdot d^2(\mathsf{v}_i,\; x_k) \tag{3.13}$$

where the factor ω_k represents the weight for the k'th datum. With constant real-valued parameter q the influence of the weighting factor can be controlled.

Theorem 3.4 (Fuzzy Clustering with Outliers – Weighting Parameters)

Considering the constraint

$$\sum_{k=1}^{n} \omega_k = \omega, \tag{3.14}$$

where ω is a constant real valued parameter and assuming that the membership degrees are fixed, leads to the following update equation for the weighting parameters

$$\omega_k \;=\; \frac{\left(\sum_{i=1}^{c} u_{ik}^{m}\, d^2(\mathsf{v}_i,\; x_k)\right)^{\frac{1}{q+1}}}{\sum_{l=1}^{n}\left(\sum_{i=1}^{c} u_{il}^{m}\, d^2(\mathsf{v}_i,\; x_l)\right)^{\frac{1}{q+1}}} \cdot \omega, \tag{3.15}$$

see proof 3.4. It should be mentioned, that as long as not all clusters collapse, we obtain non-zero values for the weighting parameters ω_k.

Proof 3.4 (Fuzzy Clustering with Outliers – Weighting Parameters)

$$J^{outlier}\left(X, U, \mathsf{v}\right) \;=\; \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^{m} \cdot \frac{1}{\omega_k^q} \cdot d^2\left(\mathsf{v}_i, x_k\right)$$

Considering constraint (3.14) leads to the Lagrange function

$$J_\lambda^{outlier}\left(X, U, \mathsf{v}\right) \;=\; \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^{m} \cdot \frac{1}{\omega_k^q} \cdot d^2\left(\mathsf{v}_i, x_k\right) + \lambda \cdot \left(\sum_{k=1}^{n} \omega_k - \omega\right) \tag{1}$$

Differentiating (1) *with respect to ω_k leads to*

$$\frac{\partial J_\lambda^{outlier}\left(X, U, \mathsf{v}\right)}{\partial \omega_k} \;=\; -q \cdot \frac{1}{w_k^{q+1}} \cdot \sum_{i=1}^{c} u_{ik}^{m} \cdot d^2\left(\mathsf{v}_i, x_k\right) + \lambda \overset{!}{=} 0$$

and therefore to

$$\lambda \;=\; q \cdot \frac{1}{w_k^{q+1}} \cdot \sum_{i=1}^{c} u_{ik}^{m} \cdot d^2\left(\mathsf{v}_i, x_k\right).$$

Resolving for ω_k gives us

$$\omega_k \;=\; \left(\frac{q \cdot \sum_{i=1}^{c} u_{ik}^{m} \cdot d^2\left(\mathsf{v}_i, x_k\right)}{\lambda}\right)^{\frac{1}{q+1}}. \tag{2}$$

With constraint (3.14) we obtain

$$\omega \;=\; \sum_{k=1}^{n}\left(\frac{q \cdot \sum_{i=1}^{c} u_{ik}^{m} \cdot d^2\left(\mathsf{v}_i, x_k\right)}{\lambda}\right)^{\frac{1}{q+1}}$$

leading to

$$\lambda^{\frac{1}{q+1}} \;=\; \sum_{k=1}^{n}\left(q \cdot \sum_{i=1}^{c} u_{ik}^{m} \cdot d^2\left(\mathsf{v}_i, x_k\right)\right)^{\frac{1}{q+1}} \cdot \frac{1}{\omega}$$

and therefore

$$\lambda = \left(\sum_{k=1}^{n} \left(q \cdot \sum_{i=1}^{c} u_{ik}^{m} \cdot d^{2}\left(\mathsf{v}_i, x_k\right) \right)^{\frac{1}{q+1}} \cdot \frac{1}{\omega} \right)^{q+1} . \tag{3}$$

Inserting (3) *in* (2) *leads to*

$$\begin{aligned} \omega_k &= \frac{\left(q \cdot \sum_{i=1}^{c} u_{ik}^{m} \cdot d^{2}\left(\mathsf{v}_i, x_k\right)\right)^{\frac{1}{q+1}}}{\sum_{l=1}^{n} \left(q \cdot \sum_{i=1}^{c} u_{il}^{m} \cdot d^{2}\left(\mathsf{v}_i, x_l\right)\right)^{\frac{1}{q+1}}} \cdot \omega \\ &= \frac{\left(\sum_{i=1}^{c} u_{ik}^{m} \cdot d^{2}\left(\mathsf{v}_i, x_k\right)\right)^{\frac{1}{q+1}}}{\sum_{l=1}^{n} \left(\sum_{i=1}^{c} u_{il}^{m} \cdot d^{2}\left(\mathsf{v}_i, x_l\right)\right)^{\frac{1}{q+1}}} \cdot \omega \quad \diamond \end{aligned}$$

Again, additional assumptions have to be made to derive update equations for the membership degrees. In the probabilistic case

$$\sum_{k=1}^{n} u_{ik} > 0 \qquad \forall\, i \in \{1, \ldots, c\}$$

and

$$\sum_{i=1}^{c} u_{ik} = 1 \qquad \forall\, k \in \{1, \ldots, n\}$$

have to be taken into account. Assuming the weighting factors as constant, the distance measure could be combined with ω_k and so defined as

$$d^2(\mathsf{v}_i,\ x_k,\ \omega_k) = \frac{1}{\omega_k^q} \cdot d^2(\mathsf{v}_i,\ x_k).$$

Using this distance measure, leads to the probabilistic equation for the membership degrees.

$$\begin{aligned} u_{ik} &= \frac{1}{\sum_{j=1}^{c} \left(\frac{d^2(\mathsf{v}_i,\, x_k,\, \omega_k)}{d^2(\mathsf{v}_j,\, x_k,\, \omega_k)} \right)^{\frac{1}{m-1}}} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\frac{1}{\omega_k^q} \cdot d^2(\mathsf{v}_i,\, x_k)}{\frac{1}{\omega_k^q} \cdot d^2(\mathsf{v}_j,\, x_k)} \right)^{\frac{1}{m-1}}} \\ &= \frac{1}{\sum_{j=1}^{c} \left(\frac{d^2(\mathsf{v}_i,\, x_k)}{d^2(\mathsf{v}_j,\, x_k)} \right)^{\frac{1}{m-1}}} \end{aligned}$$

To derive update equations for the prototypes we can combine the weighting factor with the membership degrees, since both are assumed to be fixed for the prototype parameter estimation. Thus,

$$\tilde{u}_{ik}^{m} = \frac{u_{ik}^{m}}{\omega_k^{q}} \tag{3.16}$$

is defined. Now all of the following proofs for the prototypes in case of probabilistic clustering remain valid for the fuzzy clustering with outliers. Only the membership degrees u_{ik}^m have to be replaced by $\tilde{u}_{ik}^m$.

Based on this approach an alternating optimisation scheme for fuzzy clustering is derived, using the weighting factor in combination with a chosen distance measure, see algorithm 3.4. As distance measure the Euclidean distance used for the fuzzy c-means as well as the distance of the Gustafson-Kessel algorithm or its axes parallel version can be used. Also the new defined distance measures of sections 5, 6.1, 7, or 8 are well suited for this approach.

The aim of this fuzzy clustering with outliers is to add small weighting factors ω_k (large values for $\frac{1}{\omega_k^q}$) to data points fitting well to at least one of the clusters. Outliers often have a relatively large distance to all of the data groups and are equally shared among the groups. In this approach they are assigned a large weight ω_k, so $\frac{1}{\omega_k^q}$ is small in this case. In equation (3.15) the influence of parameter q becomes obvious. For $q \to \infty$, all parameter $\omega_k \to \frac{\omega}{n}$ and therefore gain the same influence on the partition, whereas for $q \to 0$ the weighting influence reaches its maximum.

Algorithm 3.4 (Basic Clustering with Outliers)

```
Choose()
{
    $m \in \mathbb{R}_{>1}$;
    $c \in \{2, \dots, n-1\}$;
    $\epsilon > 0$;
    $q \in \mathbb{R}_{>0}$;
    $\omega \in \mathbb{R}_{>0}$;
}

Initialise()
{
    $\mathsf{v}_i$   for all $i \in \{1, \dots, c\}$;
    CalculateMembership($u_{ik}^{(new)}$);
}
```

Algorithm 3.4 (Basic Clustering with Outliers – continued)

```
CalculateMembership($u_{ik}$)
{
    $\Im_x := \{\mathsf{v}_j \mid d^2(\mathsf{v}_j, x_k) = 0 \wedge j \in \{1, \dots, c\}\};$
    $u_{ik} := \begin{cases} \frac{1}{|\Im_x|} & \text{if } \mathsf{v}_i \in \Im_x \wedge \Im_x \neq \varnothing; \\ 0 & \text{if } \mathsf{v}_i \ni \Im_x \wedge \Im_x \neq \varnothing; \\ \frac{1}{\sum_{j=1}^{c} \left( \frac{d^2(\mathsf{v}_i, x_k)}{d^2(\mathsf{v}_j, x_k)} \right)^{\frac{1}{m-1}}} & \text{if } \Im_x = \varnothing; \end{cases}$
}

CalculateWeightingParameter($\omega_k$)
{
    $\omega_k \; := \; \frac{\left( \sum_{i=1}^{c} u_{ik}^m \cdot d^2\left(v_i, x_k\right) \right)^{\frac{1}{q+1}}}{\sum_{l=1}^{n} \left( \sum_{i=1}^{c} u_{il}^m \cdot d^2\left(v_i, x_l\right) \right)^{\frac{1}{q+1}}} \cdot \omega;$
}

CalculatePartition()
{
    do
    {
        for all $k \in \{1, \dots, n\}$
            CalculateWeightingParameter($\omega_k$);
        for all $i \in \{1, \dots, c\}$
            Calculate($\mathsf{v}_i$) using $\tilde{u}_{ik}^m \; = \; \frac{\left( u_{ik}^{(new)} \right)^m}{\omega_k^q};$
        for all $i \in \{1, \dots, c\} \wedge k \in \{1, \dots, n\}$
            $u_{ik}^{(old)} := u_{ik}^{(new)};$
        for all $i \in \{1, \dots, c\} \wedge k \in \{1, \dots, n\}$
            CalculateMembership($u_{ik}^{(new)}$);
    }while $\left( \left( \sum_{i=1}^{c} \sum_{k=1}^{n} \mid u_{ik}^{(new)} - u_{ik}^{(old)} \mid \right) < \epsilon \right);$
}
```

3.2 Distance measures and algorithms

In the previous section some general clustering concepts have been described. All techniques rely on the definition of suitable distance measures. Choosing a certain dissimilarity measure defines the structure which is searched for in the sample data. Different distance measures are able to describe varying forms or shapes of clusters. In general we can say that the more flexible one algorithm is in detecting different cluster forms, the more susceptible is this algorithm towards local optima. In the following we start with a simple form – the Euclidean distance – and change gradually to more complex forms. One possibility to overcome local optimal solutions is to use the clustering result of a simpler and therefore less susceptible fuzzy clustering algorithm as initialisation for a more adaptable algorithm. Another possibility is to use statistic clustering methods to calculate initial prototypes, see for instance [90, 10], instead of random points.

3.2.1 The Fuzzy c-Means Algorithm

One simple fuzzy clustering technique is the *fuzzy c-means algorithm* (*FCM*), see e.g. [15, 39, 40], where the distance $d^2(\mathsf{v}_i, x_k)$ is chosen as the squared Euclidean distance

$$d^2(\mathsf{v}_i, x_k) \; = \; \mathcal{D}_{FCM} \; = \| \; x_k - v_i \; \|^2 = \sum_{\nu=1}^{p} \left(x_k^{(\nu)} - v_i^{(\nu)} \right)^2 \tag{3.17}$$

where the prototypes are vectors $v_i \in \mathbb{R}^p$, with p the dimensionality or number of attributes of the data. $x_k^{(\nu)}$ $\left(v_i^{(\nu)}\right)$ denotes the ν'th coordinate of the data vector (cluster centre representative). In [62] an approach using the median ($l_1 - norm$) instead of the mean is introduced.

Due to the Euclidean distance measure, this technique searches for spherical clusters of approximately the same size. See figure 3.2 for an illustration of the Euclidean distance to the cluster centre $v^\top = (0, 0)$ and varying data $x^\top = (x_0, x_1)$. The distance is indicated by colour and contour lines.

Theorem 3.5 (Fuzzy c-Means Prototypes)

By differentiating (3.1), (3.7) or (3.10) we obtain the necessary condition

$$v_i \; = \; \frac{\sum_{k=1}^{n} u_{ik}^m \cdot x_k}{\sum_{k=1}^{n} u_{ik}^m} \tag{3.18}$$

as prototype calculation instruction for the objective functions to have a (local) minimum using $\mathcal{D}_{FCM}$, see Proof 3.5. In the following, $v_i^{(\nu)}$, $x_k^{(\nu)}$ denote the ν'th component of v_i and x_k, respectively.

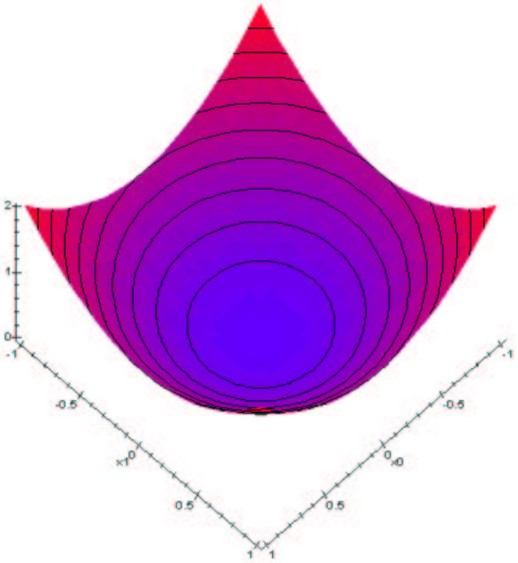


Figure 3.2: Euclidean distance to $v^\top = (0, 0)$

Proof 3.5 (Fuzzy c-Means Prototypes)

$$
\begin{aligned}
J^{prob}(X, U, v) &= \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^{m} \cdot d^{2}(\mathsf{v}_i, x_k) \\
&= \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^{m} \cdot \parallel x_k - v_i \parallel^{2} \\
\frac{\partial J^{prob}(X, U, v)}{\partial v_i^{(\nu)}} &= -2 \cdot \sum_{k=1}^{n} u_{ik}^{m} \cdot \left(x_k^{(\nu)} - v_i^{(\nu)}\right) \overset{!}{=} 0 \\
\Rightarrow \sum_{k=1}^{n} u_{ik}^{m} \cdot v_i^{(\nu)} &= \sum_{k=1}^{n} u_{ik}^{m} \cdot x_k^{(\nu)} \\
\Rightarrow v_i^{(\nu)} &= \frac{\sum_{k=1}^{n} u_{ik}^{m} \cdot x_k^{(\nu)}}{\sum_{k=1}^{n} u_{ik}^{m}} \\
\Rightarrow v_i &= \frac{\sum_{k=1}^{n} u_{ik}^{m} \cdot x_k}{\sum_{k=1}^{n} u_{ik}^{m}} \quad \diamond
\end{aligned}
$$

Since the first summand is identical in the three objective functions from Section 3.1 and the second term in equations (3.7) and (3.10) does not depend on a certain distance measure, the derived prototype equation holds in all three cases. These prototypes could be used alternating with (3.4), (3.8), or (3.11) in the algorithms 3.1, 3.2, 3.3, and 3.4 as step Calculate(v_i), see Algorithm 3.5. The update equation for the membership degrees depends on the chosen basic objective function as described in the previous section.

Algorithm 3.5 (Prototype Calculation for FCM)

```
Calculate($\mathsf{v}_i$)
{
    $v_i = \frac{\sum_{k=1}^{n} u_{ik}^m \cdot x_k}{\sum_{k=1}^{n} u_{ik}^m}$;
}
```

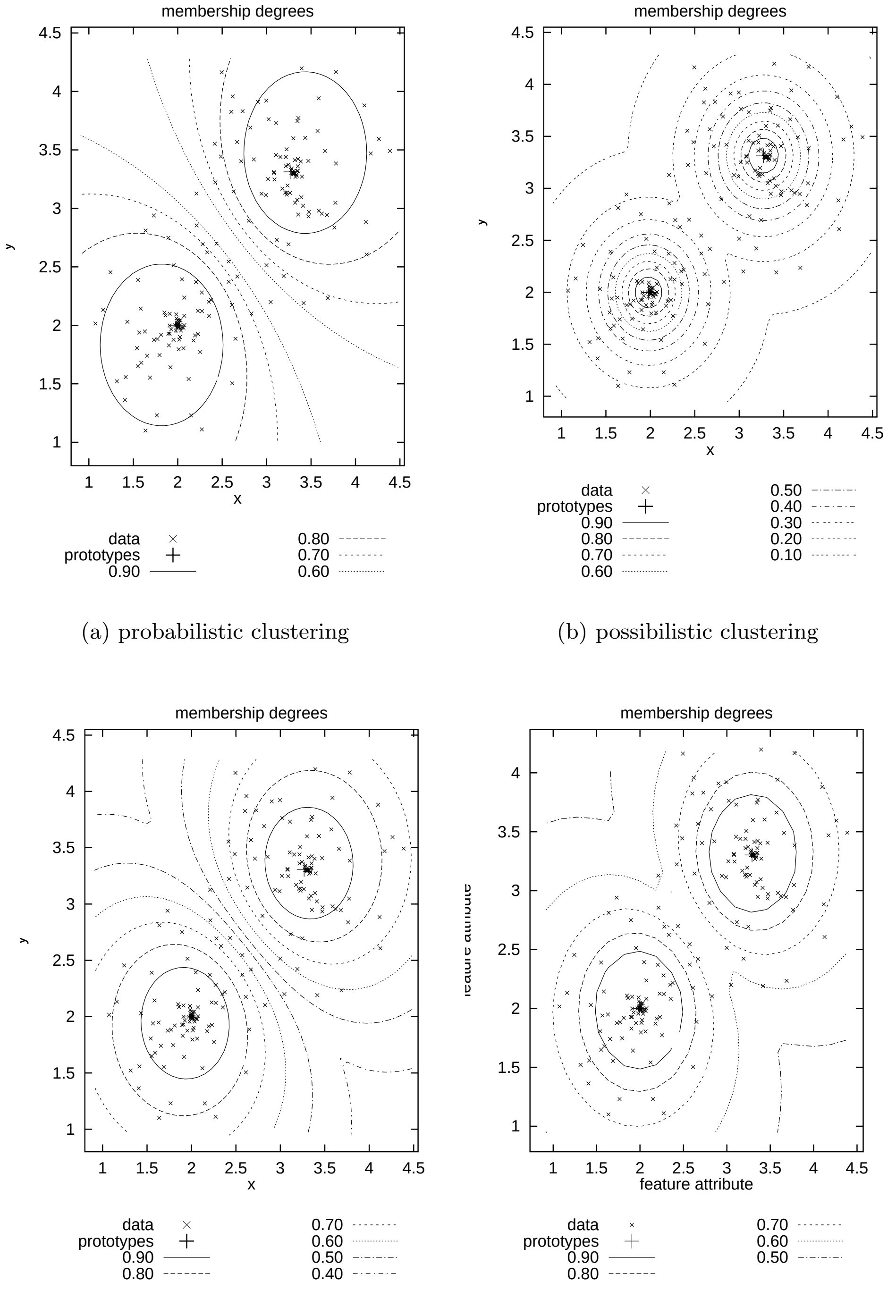
In figure 3.3 an example for a data set with two circular groups is shown. The clustering was calculated for the four described basic objective functions from the previous section, probabilistic (see figure 3.3(a)), possibilistic (see figure 3.3(b)), noise (see figure 3.3(c)), and outlier clustering (see figure 3.3(d)). The parameters for the basic clustering algorithms were set as follows: fuzzifier $m = 2$, number of clusters $c = 2$, the lower bound to end the calculation $\epsilon = 0.0001$, and additionally for noise clustering parameter δ^2 as weighted medium distance

$$\delta^2 = \frac{2}{c \cdot n} \cdot \left(\sum_{k=1}^{n} \sum_{i=1}^{c} d^2(\mathsf{v}_i, x_k) \right).$$

For fuzzy clustering with outliers the weighted membership degrees are illustrated for $q = 0.5$ and $\omega = 200$, i.e.

$$\tilde{u}_{ik} = \left(\frac{u_{ik}^m}{\omega_k^q} \right)^{\frac{1}{m}}.$$

These figures show the differences of the objective functions introduced in section 3.1. In case of probabilistic clustering (fig. 3.3(a)), data points with the same distance to both clusters are assigned a membership degree of about 0.5 to both clusters. This leads to the effect, that data points in the middle of the illustration in between both clusters are assigned a smaller membership degree than data points with the same distance to one cluster's centre but on the opposite side of that centre, e.g. in the upper right corner of this figure. In possibilistic clustering (fig. 3.3(b)), the membership degrees decrease rapidly with increasing distance of the data points to the cluster centres in all directions. This way, data points in between both cluster centres as well as those in the upper right and lower left corners of fig. 3.3(b) are assigned a



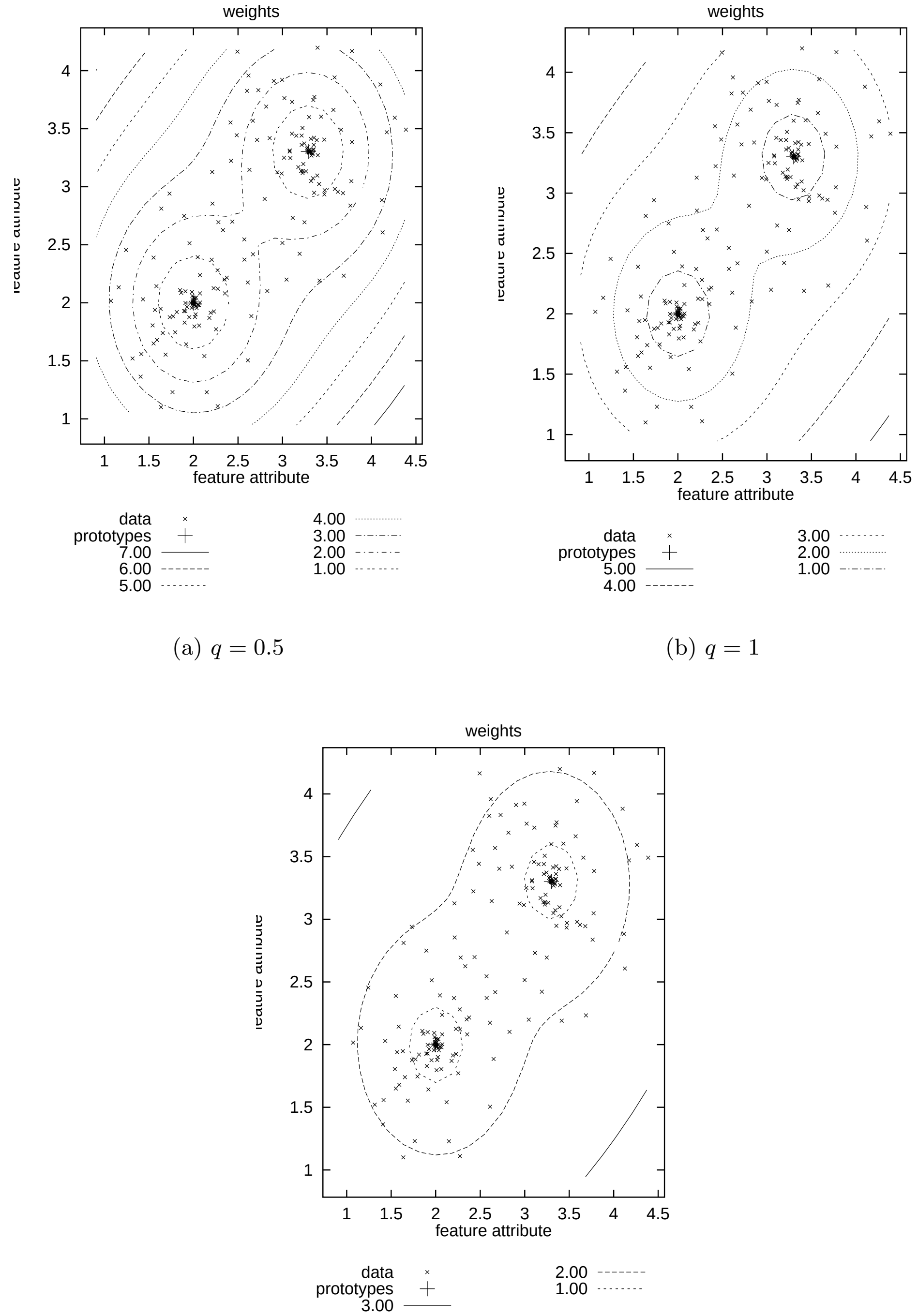
(a) probabilistic clustering

(b) possibilistic clustering

(c) noise clustering

(d) outlier clustering

Figure 3.3: Clustering Results for FCM ($m = 2$, $c = 2$, $\epsilon = 0.0001$) – membership degrees



(a) $q = 0.5$

(b) $q = 1$

(c) $q = 2$

Figure 3.4: Clustering Results for FCM ($m = 2$, $c = 2$, $\epsilon = 0.0001$) – weights for outlier clustering

small membership degree of about 0.2 to both clusters. Noise clustering and outlier clustering to not lead to so rapidly decreasing membership degrees as possibilistic clustering but are able to assign smaller membership degrees to data points in the upper right and lower left corners (see fig. 3.3(c) and 3.3(d)). In noise clustering the "medium" distance δ^2 is used to reduce the membership degrees for data points at the edge of the cluster's range. Individual weights that are adapted during the cluster calculation are used in outlier clustering to identify outliers. In figure 3.4 the calculated weights ω_k are shown for $q = 0.5$, $q = 1$, and $q = 2$. Here we see the effect of parameter q. The smaller q, the greater is the emphasis on the weight adaptation. For $q = 2$ the single data points are assigned similar weights, fig. 3.4(c), whereas $q = 0.5$ (3.4(a)) leads to weights between about 0 and 4.0. The wider distribution allows a more precise distinction whether data points do belong to a certain partition.

In [14] and [23] Bezdek and others have shown that the alternating optimisation for probabilistic FCM converges to a saddle point or a (local) minimum. A general convergence analysis for the algorithms based on the probabilistic basic algorithm scheme (Algorithm 3.1) does not exist. Only some clustering algorithms have been analysed concerning convergence, see e.g. [21]. Nevertheless, the great success of these techniques suggests their use in practical operations.

3.2.2 The Algorithm by Gustafson and Kessel

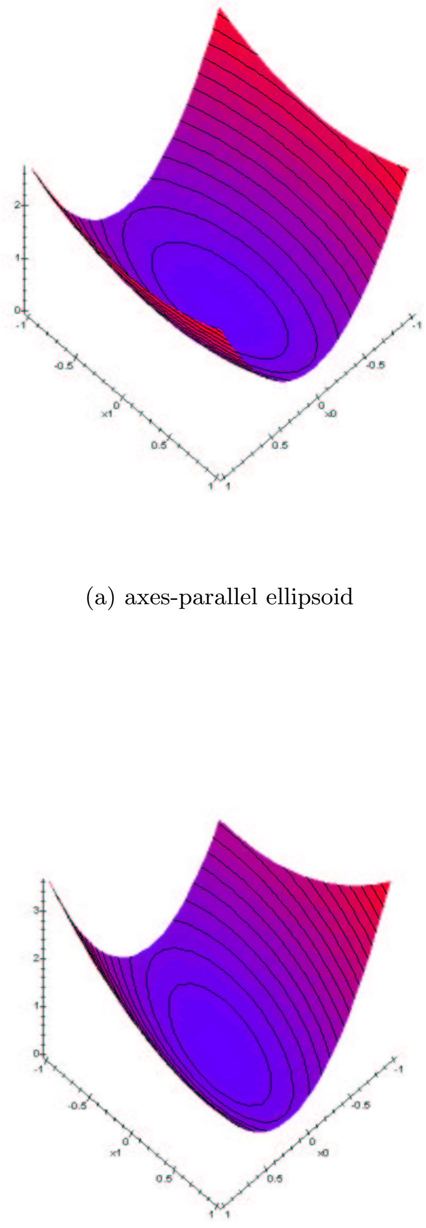
Gustafson and Kessel [48] designed a fuzzy clustering method that is able to adapt to hyper-ellipsoidal forms. Therefore, a transformed Euclidean distance of the form $(x_k - v_i)^\top C_i^{-1}(x_k - v_i)$ is used, where C_i is a symmetric positive definite matrix. This distance measure is illustrated in Figure 3.5. Colour and contour lines denote the distance of a point $x^\top = (x_0, x_1)$ to the cluster centre $v^\top = (0, 0)$. In Figure 3.5(a) the diagonal matrix

$$C_i = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix} \Rightarrow C_i^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 0.2 \end{pmatrix} \text{ and } \det C_i = 5$$

has been used to transform the Euclidean distance. The diagonal elements of matrix C_i determine the axis-parallel expansion of the region having the same distance to the structures centre v_i. Figure 3.5(b) illustrates the effect of non-axes-parallel matrix elements

$$C_i = \begin{pmatrix} 0.595 & 0.476 \\ 0.476 & 2.381 \end{pmatrix} \Rightarrow C_i^{-1} = \begin{pmatrix} 2 & -0.4 \\ -0.4 & 0.5 \end{pmatrix} \text{ and } \det C_i = 1.091.$$

The region with the same distance to the structure's centre is expanded as in the case of diagonal matrices C_i and additionally rotated around the centre v_i. Both matrices are positive definite since all eigenvalues are positive ($>$ 0). The corresponding clustering algorithm uses the cluster centres and the transforming matrices as cluster parameters. Both are alternately updated using the updated membership degrees for the calculation. So the prototypes consist of the cluster centres v_i as in FCM and (positive definite) *covariance*



(a) axes-parallel ellipsoid

(b) rotated ellipsoid

Figure 3.5: Transformed Euclidean distances to $v^\top = (0, 0)$

matrices C_i. The *Gustafson-Kessel algorithm* (*GK*) replaces the Euclidean distance by the transformed Euclidean distance

$$d^2(\mathsf{v}_i, x_k) \;=\; \mathcal{D}_{GK} \;=\; (\rho_i \det C_i)^{1/p} \cdot (x_k - v_i)^\top C_i^{-1}(x_k - v_i). \tag{3.19}$$

The factor $(\rho_i \det C_i)^{1/p}$ in $\mathcal{D}_{GK}$ guarantees the volume for all clusters to be constant. Factor ρ_i can be used to determine the size of cluster i and is not changed during the alternating optimisation. If the sizes cannot be estimated in advance, the parameters ρ_i might be set to one.

Theorem 3.6 (Gustafson-Kessel Covariance Matrices)

The covariance matrices C_i are computed using equation (3.20), see Proof 3.6.

$$C_i = \sum_{k=1}^{n} u_{ik}^m \cdot (x_k - v_i)(x_k - v_i)^\top. \tag{3.20}$$

Proof 3.6 (Gustafson-Kessel Covariance Matrices)

Inserting the distance measure 3.19 into the general objective function leads to

$$\begin{aligned} J^{prob}(X, U, v) \;&=\; \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^m \cdot d^2(\mathsf{v}_i, x_k) \\ &= \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^m \cdot (\rho_i \det C_i)^{1/p} \cdot (x_k - v_i)^\top C_i^{-1}(x_k - v_i). \end{aligned}$$

Setting $\rho_i \;=\; 1$ for all $i \in \{1, \ldots, c\}$ without loss of generality and

$$G_i \;=\; \sqrt[p]{det(C_i)} \cdot C_i^{-1}$$

leads to

$$J^{prob}(X, U, v) \;=\; \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^m \cdot (x_k - v_i)^\top \cdot G_i \cdot (x_k - v_i).$$

Under constraints

$$det(G_i) \;=\; 1 \text{ for all } i \in \{1, \ldots, c\}. \tag{1}$$

In this way the Lagrange function is as follows

$$J_\lambda^{prob}(X, U, v) \;=\; \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^m \cdot (x_k - v_i)^\top \cdot G_i \cdot (x_k - v_i) - \sum_{i=1}^{c} \lambda_i \cdot det(G_i).$$

Consider G_i as regular matrices $\in \mathbb{R}^{p\times p}$

$$\begin{aligned}\frac{\partial J_\lambda^{prob}(X,U,v)}{\partial G_i} &= \sum_{k=1}^{n} u_{ik}^m \cdot (x_k - v_i)^\top (x_k - v_i) - \lambda_i \cdot det(G_i) \cdot G_i^{-1} \overset{!}{=} 0 \\ &\Rightarrow \sum_{k=1}^{n} u_{ik}^m \cdot (x_k - v_i)^\top (x_k - v_i) \;=\; \lambda_i \cdot det(G_i) \cdot G_i^{-1}.\end{aligned}$$

Let $\mathbf{I} \in \mathbb{R}^{p\times p}$ denote the identity matrix and take into account that G_i is invertible. With (1) we then receive

$$G_i \cdot \left(\sum_{k=1}^{n} u_{ik}^m \cdot (x_k - v_i)^\top (x_k - v_i) \right) = \lambda_i \cdot \mathbf{I}. \tag{2}$$

Leading to

$$\lambda_i^p \;=\; \det\left(G_i \cdot \sum_{k=1}^{n} u_{ik}^m \cdot (x_k - v_i)^\top (x_k - v_i) \right)$$

and therefore

$$\begin{aligned}\Rightarrow \lambda_i &= \sqrt[p]{\det G_i \cdot \det\left(\sum_{k=1}^{n} u_{ik}^m \cdot (x_k - v_i)^\top (x_k - v_i) \right)} \\ &= \sqrt[p]{\det\left(\sum_{k=1}^{n} u_{ik}^m \cdot (x_k - v_i)^\top (x_k - v_i) \right)}.\end{aligned}$$

Inserting in (2) gives us

$$G_i = \sqrt[p]{\det\left(\sum_{k=1}^{n} u_{ik}^m \cdot (x_k - v_i)^\top (x_k - v_i) \right)} \cdot \left(\sum_{k=1}^{n} u_{ik}^m \cdot (x_k - v_i)^\top (x_k - v_i) \right)^{-1},$$

so that we finally obtain

$$C_i \;=\; \sum_{k=1}^{n} u_{ik}^m \cdot (x_k - v_i)^\top (x_k - v_i) \qquad \diamond$$

The prototype calculation instruction can be derived analogously to equation (3.18), so again we obtain

$$v_i \;=\; \frac{\sum_{k=1}^{n} u_{ik}^m \cdot x_k}{\sum_{k=1}^{n} u_{ik}^m}$$

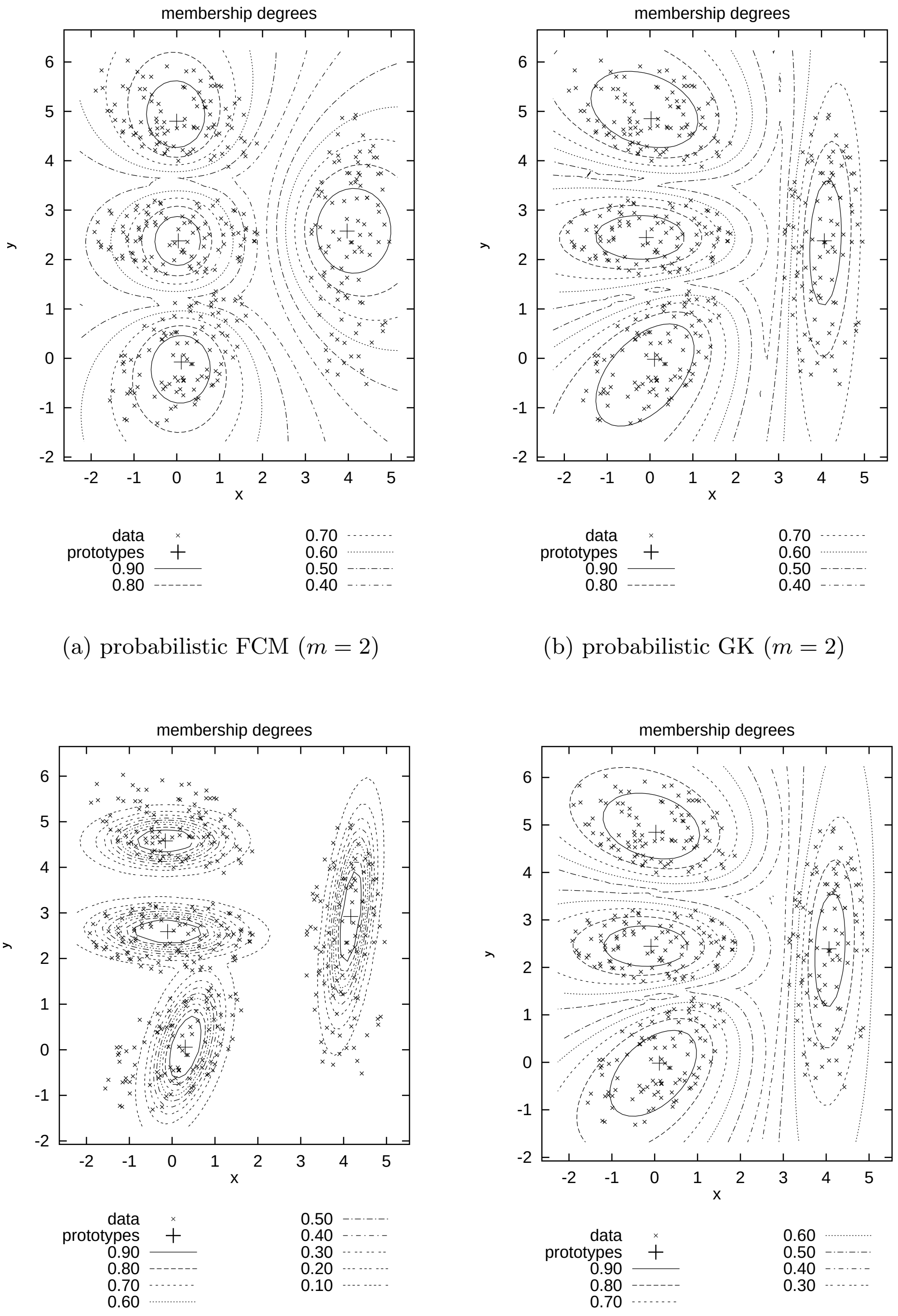
as a necessary condition for the objective functions (3.1), (3.7), or (3.10) to have a (local) minimum. With these equations the alternating iteration procedure for the Gustafson-Kessel algorithm is defined. In the corresponding update equation for the membership degrees, (3.4), (3.8), or (3.11), respectively, the distance measure $\mathcal{D}_{GK}$, see equation (3.19), has to be used for $d^2(\mathsf{v}_i, x_k)$. The step Calculate(v_i) of the general algorithms from Section 3.1 is shown in Algorithm 3.6.

Algorithm 3.6 (Prototype Calculation for GK)

```
Calculate($\mathsf{v}_i$)
{
    $v_i = \frac{\sum_{k=1}^{n} u_{ik}^m \cdot x_k}{\sum_{k=1}^{n} u_{ik}^m};$
    $C_i = \sum_{k=1}^{n} u_{ik}^m \cdot (x_k - v_i)(x_k - v_i)^\top;$
}
```

This general form of the Gustafson-Kessel algorithm searches for hyper-ellipsoidal forms in the domain of interest.

In figure 3.6 the results for the basic probabilistic objective function with distance $\mathcal{D}_{FCM}$ (figure 3.6(a)) and $\mathcal{D}_{GK}$ (figure 3.6(b)) are shown. For all classifications, the parameters have been set as follows: $c = 4$, $\epsilon = 0.001$ and m as denoted in the figures. The limitations of the distance $\mathcal{D}_{FCM}$ can be seen on the membership degrees of fig. 3.6(a) and 3.6(b). In case of the FCM, the algorithm is not able to adapt to the ellipsoidal structures whereas the transformed distance of the GK adapts well to the cluster structure. Figures 3.6(b), 3.6(c), and 3.6(d) show the clustering result for distance measure $\mathcal{D}_{GK}$ in combination with the probabilistic, possibilistic, and noise clustering algorithm, respectively. Since one has to be careful with the possibilistic clustering algorithm in choosing a suitable parameter m, see section 3.10, here a slightly smaller value $m = 1.5$ has been selected. The capabilities of the different basic objective functions in combination with $\mathcal{D}_{GK}$ are equivalent to the combinations with $\mathcal{D}_{FCM}$. Whereas in probabilistic clustering, fig. 3.6(b), data points with similar distances to all clusters are assigned a membership degree of about $\frac{1}{c}$ to each cluster, the emphasis in assigning membership degrees in case of possibilistic clustering, fig. 3.6(c), lies on the distance to a single cluster's centre. In noise clustering, fig. 3.6(d), again an overall medium distance is used to identify outliers. The results for clustering with outliers – membership degrees $\tilde{u}_{ik}^m$ and weighting parameters ω_k – are illustrated in figure 3.7 for different values of q. As constraint parameter $\omega = 200$ was chosen. The membership degrees illustrate the influence of the weighting parameter on the extension of the clusters. The lager the weighting parameter q the greater the cluster's range with high membership degrees, see fig. 3.7(b) and 3.7(a). The



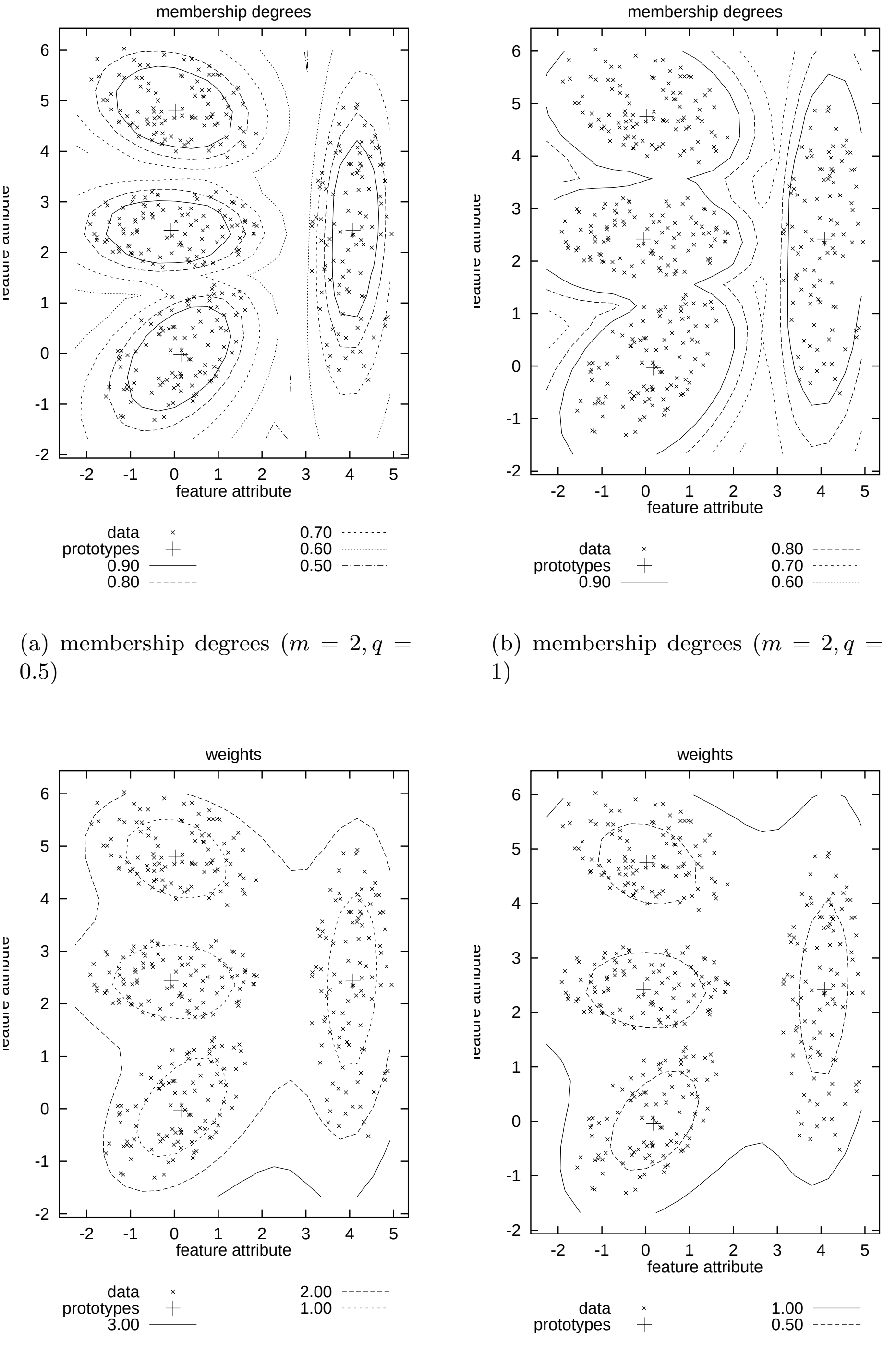
(a) probabilistic FCM ($m = 2$)

(b) probabilistic GK ($m = 2$)

(c) possibilistic GK ($m = 1.5$)

(d) noise GK clustering ($m = 2$)

Figure 3.6: Clustering results for an elliptical test data set ($c = 4$, $\epsilon = 0.001$)



(a) membership degrees ($m = 2, q = 0.5$)

(b) membership degrees ($m = 2, q = 1$)

(c) weights ($m = 2, q = 0.5$)

(d) weights ($m = 2, q = 1$)

Figure 3.7: Results for outlier clustering ($c = 4$, $\epsilon = 0.001$, $\omega = 200$)

single weights of the data points are inversely proportional to the membership degrees. In figures 3.7(c) and 3.7(d) we see again the influence of the weighting parameter q on the separation.

Other clustering techniques dealing with cluster-specific scatter matrices that enable the clustering methods to describe elliptical clusters with different orientations are presented in [97, 98].

The axes parallel Gustafson-Kessel algorithm

Considering e.g. the task of rule learning, where fuzzy clusters are projected to single dimensions (see section 9.1 for an illustration) non-axes parallel ellipsoids would lead in general to a serious loss of information caused by the construction of the fuzzy sets for the single domains. One approach to reduce this drawback is to restrict the covariance matrices C_i to diagonal matrices resulting in axes-parallel hyper-ellipsoids [70, 73]. The resulting clustering algorithm is less flexible than the original GK. Instead of scaling the axes and rotating the resulting ellipsoid, only the axes are scaled. This way the axes parallel Gustafson-Kessel algorithm avoids inverting of matrices and calculating of determinants. In comparison to the fuzzy-c-means algorithm the axes parallel Gustafson-Kessel algorithm has still a better performance. It transforms the cluster structure from uniform (hyper-) balls into axes parallel (hyper-) ellipsoids. The corresponding algorithm is called *axes-parallel Gustafson-Kessel algorithm* (*AGK*). The distance measure in this case can be rewritten as

$$d^2(\mathsf{v}_i, x_k) \;=\; \mathcal{D}_{AGK} \;=\; \left(\rho_i \prod_{\nu=1}^{p} c_i^{(\nu)}\right)^{1/p} \cdot \left(\sum_{\nu=1}^{p} (x_k^{(\nu)} - v_i^{(\nu)})^2 \cdot \frac{1}{c_i^{(\nu)}}\right). \tag{3.21}$$

Here, p is the dimensionality of the data vectors and $x_k^{(\nu)}, v_i^{(\nu)}$ denote the ν'th component of the k'th data point, i'th cluster centre, respectively. For the alternating optimisation the calculation instruction for the covariance matrices can be simplified in the following way

$$c_i^{(\nu)} = \sum_{k=1}^{n} u_{ik}^m \cdot (x_k^{(\nu)} - v_i^{(\nu)})^2, \tag{3.22}$$

where $c_i^{(\nu)}$ denotes the ν'th diagonal element of the covariance matrix. $\mathcal{D}_{AGK}$ can be used as distance measure in the membership update equations from section 3.1. The resulting prototype calculation instruction is outlined in Algorithm 3.7.

This technique avoids not only loss of information in case of rule learning tasks but needs computationally less effort than the original Gustafson-Kessel algorithm since no reverse covariance matrix has to be calculated. If an application does not depend on the rotation of the ellipsoidal clusters as performed by the Gustafson-Kessel algorithm, the axes parallel Gustafson-Kessel algorithm should be preferred. The better performance is significant, because the reverse covariance matrices would have to be calculated for each cluster in each iteration of the clustering algorithm.

Algorithm 3.7 (Prototype Calculation for AGK)

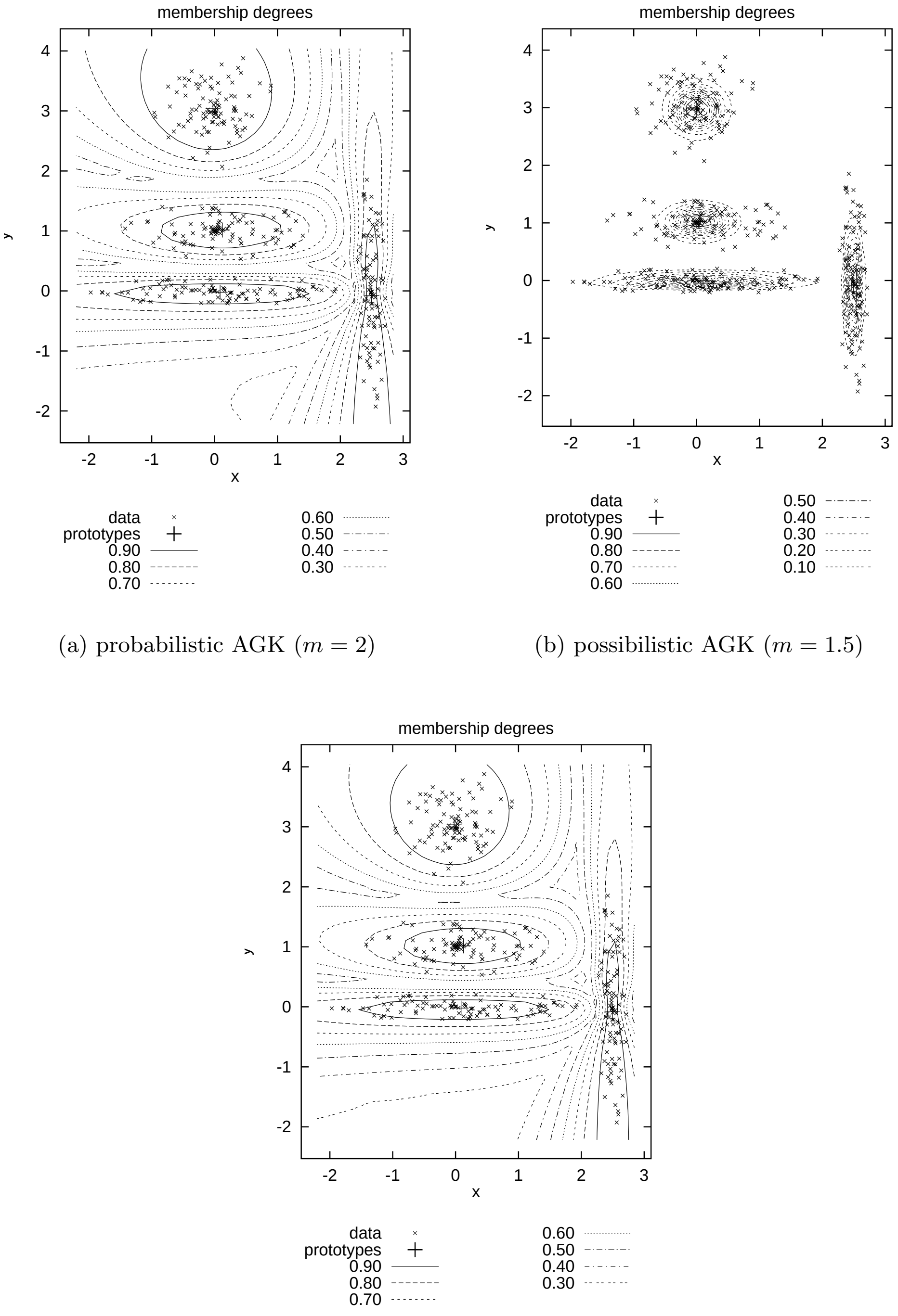
```
Calculate($\mathsf{v}_i$)
{
    $v_i = \frac{\sum_{k=1}^{n} u_{ik}^{m} \cdot x_k}{\sum_{k=1}^{n} u_{ik}^{m}}$;

    for all $\gamma \in \{1, \dots, p\}$
        $c_i^{(\gamma)} = \sum_{k=1}^{n} u_{ik}^{m} \cdot (x_k^{(\gamma)} - v_i^{(\gamma)})^2$;
}
```

Figure 3.8 shows the results for the basic probabilistic (figure 3.8(a)), possibilistic (figure 3.8(b)), and noise clustering (figure 3.8(c)) objective function with distance $\mathcal{D}_{AGK}$. For all classifications, the parameters have been set as follows: $c = 4$, $\epsilon = 0.001$ and m as denoted in the figures. For possibilistic clustering, a smaller value for m as for the other objective functions has been chosen. The membership degrees and the corresponding weights for the basic outlier clustering objective function with $\omega = n$ in combination with $\mathcal{D}_{AGK}$ are illustrated in figure 3.9. Here again we see the differences of the basic objective functions. The constraint that the sum of all membership degrees for one data point has to be one in case of probabilistic clustering, fig. 3.8(a), leads to more widespread clusters than in case of possibilistic clustering, fig. 3.8(b). The difference between probabilistic and noise clustering, fig. 3.8(c), is not so obvious in this example, but the lower empty area is more separated in case of noise clustering, i.e. smaller membership degrees are assigned in this area. In case of outlier clustering for $q = 0.5$ all data points are assigned about the same weights, fig. 3.9(c), leading to membership degrees, fig. 3.9(a), similar to the probabilistic clustering results. In case of $q = 1.5$ the weights, fig. 3.9(d), are more differentiated and the membership degrees decrease faster with increasing distance, fig. 3.9(b).

3.2.3 The Algorithm by Gath and Geva

Another clustering technique, the *Gath-Geva algorithm* (*GG*), was designed by Gath and Geva [42]. This extension of the Gustafson-Kessel algorithm is in some way able to adapt the cluster size and like the GK adapts to hyperellipsoidal forms. Actually this approach is not based on an objective function optimiser. Instead the GG is a heuristic method derived from the fuzzification of a maximum likelihood estimator. The principle idea is to assume that the data points are part of a p-dimensional normal distribution. Assuming a crisp partition of the n data points x_k, $k \in \{1, \dots, n\}$ on the c normal distributions N_i, $i \in \{1, \dots, c\}$ we have to consider hard membership degrees

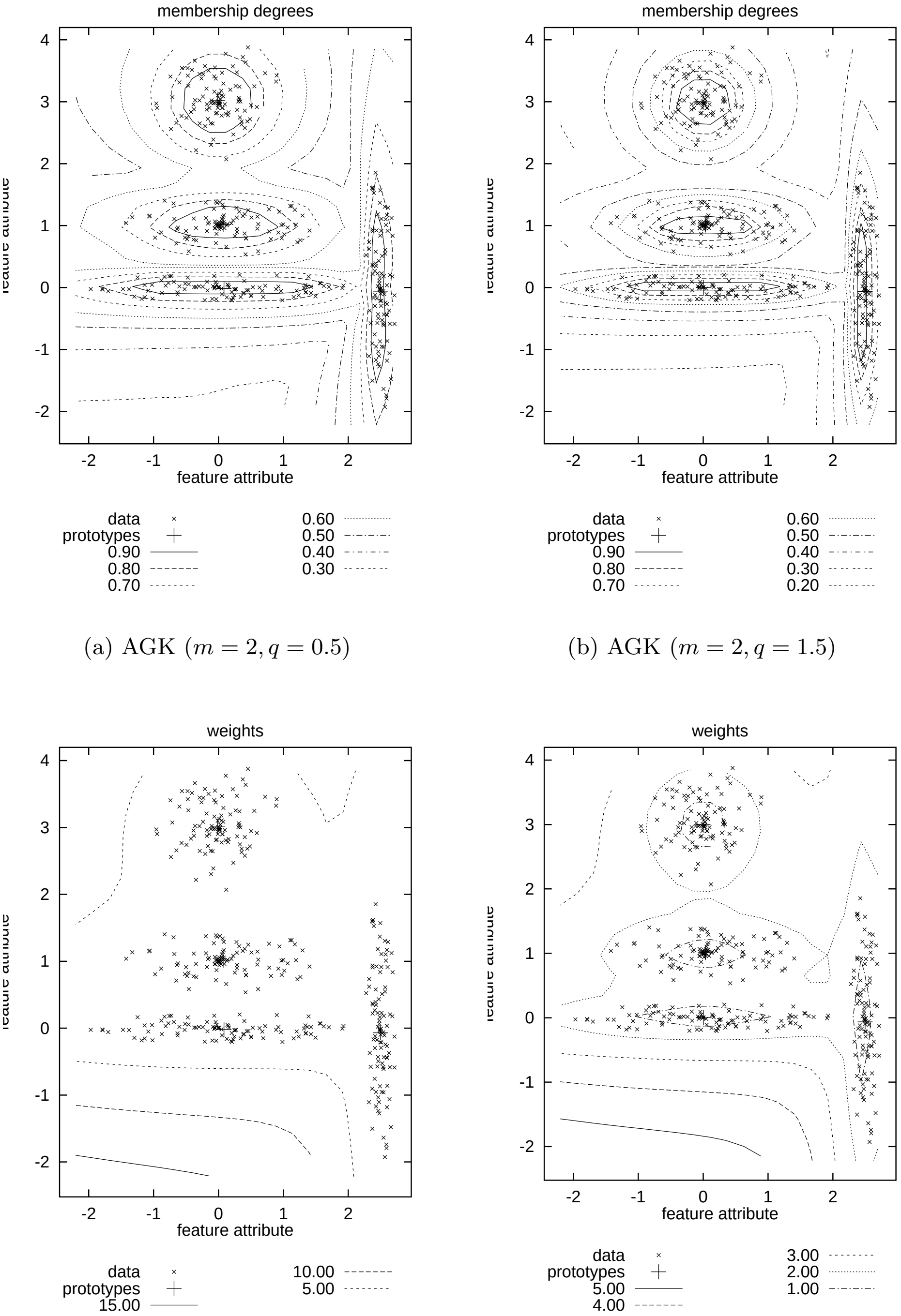


(a) probabilistic AGK ($m = 2$)

(b) possibilistic AGK ($m = 1.5$)

(c) noise AGK clustering ($m = 2$)

Figure 3.8: Clustering results for an axes-parallel elliptical test data set ($c = 4$, $\epsilon = 0.001$)



(a) AGK ($m = 2, q = 0.5$)

(b) AGK ($m = 2, q = 1.5$)

(c) AGK ($m = 2, q = 0.5$)

(d) AGK ($m = 2, q = 1.5$)

Figure 3.9: Membership degrees $\tilde{u}_{ik}^m$ and weights ω_k for outlier clustering ($c = 4$, $\epsilon = 0.001$, $\omega = n$)

$\mathrm{u}_{ik} \in \{0,\ 1\}$. In this case, known from statistics, the mean value of the i'th normal distribution is

$$v_i \;=\; \frac{\sum_{k=1}^{n} \mathrm{u}_{ik} \cdot x_k}{\sum_{k=1}^{n} \mathrm{u}_{ik}}$$

and the corresponding covariance matrix is

$$\mathrm{A}_i \;=\; \frac{\sum_{k=1}^{n} \mathrm{u}_{ik} (x_k - v_i)(x_k - v_i)^\top}{\sum_{k=1}^{n} \mathrm{u}_{ik}}.$$

These equations are similar to those obtained from Gustafson and Kessel. Therefore, a generalisation of the results from probability theory for fuzzy membership degrees seems obvious. If the *a-priori probability* to choose the i'th normal distribution for calculation of a datum is denoted by P_i, the *(non-normalised) a-posteriori probability* can be calculated as follows (Likelihood)

$$\frac{P_i}{(2\pi)^{\frac{p}{2}} \sqrt{det(\mathrm{A}_i)}} \;\cdot\; e^{-\frac{1}{2}(x_k - v_i)^\top \,\mathrm{A}_i^{-1}\, (x_k - v_i)}.$$

For the Gath and Geva algorithm, the distance measure is chosen inversely proportional to the a-posteriori probability. So the distance measure is of the following form

$$d^2(\mathsf{v}_i, x_k) \;=\; \mathcal{D}_{GG} \;=\; \frac{1}{\pi_i} \;\cdot\; \sqrt{\det(A_i)} \;\cdot\; \exp^{\left(\frac{1}{2}\cdot(x_k - v_i)^\top A_i^{-1} (x_k - v_i)\right)}. \qquad (3.23)$$

The parameter π_i denotes the a-priori probability for a datum to belong to the i'th normal distribution. π_i is estimated as described by equation (3.24), i.e. *"number of data belonging to cluster i in relation to total number of data"*.

$$\pi_i \;=\; \frac{\sum_{k=1}^{n} u_{ik}^m}{\sum_{j=1}^{c} \sum_{k=1}^{n} u_{jk}^m} \qquad (3.24)$$

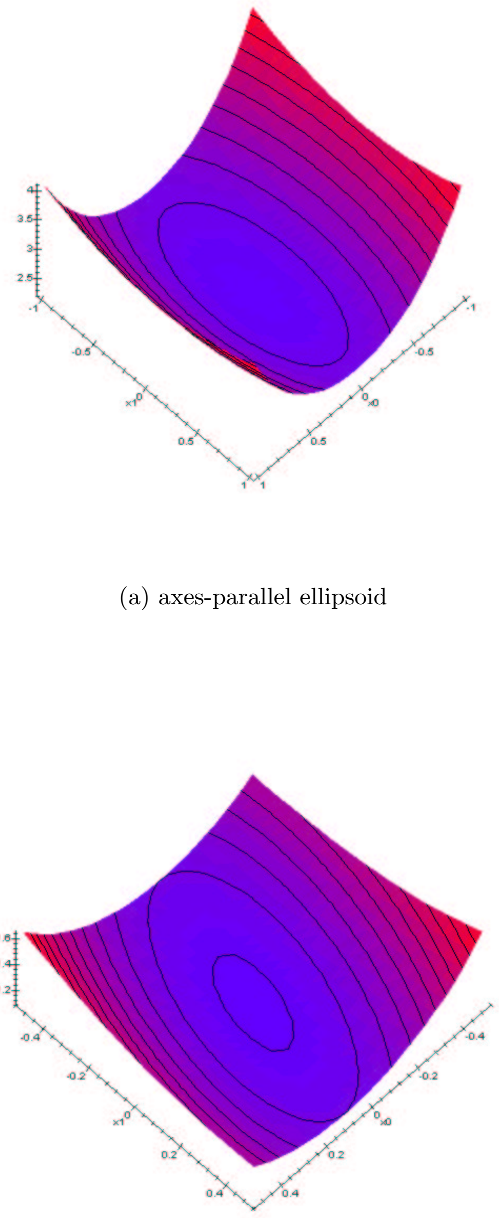
The covariance matrix of the i'th normal distribution is denoted by A_i, where (3.25) is the calculation instruction for estimating matrix A_i equivalent to the statistic covariance matrix, except that we here handle fuzzy membership degrees $u_{ik} \in [0; 1]$ instead of crisp partitions $\mathrm{u}_{ik} \in \{0, 1\}$. Additionally the fuzziness index m is assigned to u_{ik}.

$$A_i \;=\; \frac{\sum_{k=1}^{n} u_{ik}^m \cdot (x_k - v_i)(x_k - v_i)^\top}{\sum_{k=1}^{n} u_{ik}^m} \qquad (3.25)$$

An illustration of the distance measure $\mathcal{D}_{GG}$ is given in Figure 3.10. Colour and contour lines denote the distance of a point $x^\top = (x_0, x_1)$ to the cluster centre $v^\top = (0, 0)$. Again the restriction to diagonal matrices A_i leads to axes-parallel ellipsoidal structures, see Figure 3.10(a). The difference to $\mathcal{D}_{GK}$ becomes obvious, if more than one data group with varying sizes have to be detected.

In Figure 3.10(a) the diagonal matrix

$$A_i = \begin{pmatrix} 1 & 0 \\ 0 & 5 \end{pmatrix} \Rightarrow A_i^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 0.2 \end{pmatrix} \text{ and } \sqrt{\det A_i} = \sqrt{5}.$$



(a) axes-parallel ellipsoid

(b) rotated ellipsoid

Figure 3.10: Distance of the Gath-Geva algorithm to $v^{\top} = (0,0)$

has been used to calculate the distance. Figure 3.5(b) illustrates the effect of non-axes-parallel matrix elements

$$A_i = \begin{pmatrix} 0.595 & 0.476 \\ 0.476 & 2.381 \end{pmatrix} \Rightarrow A_i^{-1} = \begin{pmatrix} 2 & -0.4 \\ -0.4 & 0.5 \end{pmatrix} \text{ and } det A_i = 1.091.$$

In both cases $\pi_i = 1$ has been chosen. The matrices are positive definite since all eigenvalues are positive (< 0).

The prototype coordinates are the estimated expected values of the assumed normal distribution for cluster i, using fuzzy membership degrees and the fuzziness index m. Again the calculation of the prototypes can be done using equation (3.18) as in the FCM or GK, respectively

$$v_i \; = \; \frac{\sum_{k=1}^{n} u_{ik}^m \cdot x_k}{\sum_{k=1}^{n} u_{ik}^m}.$$

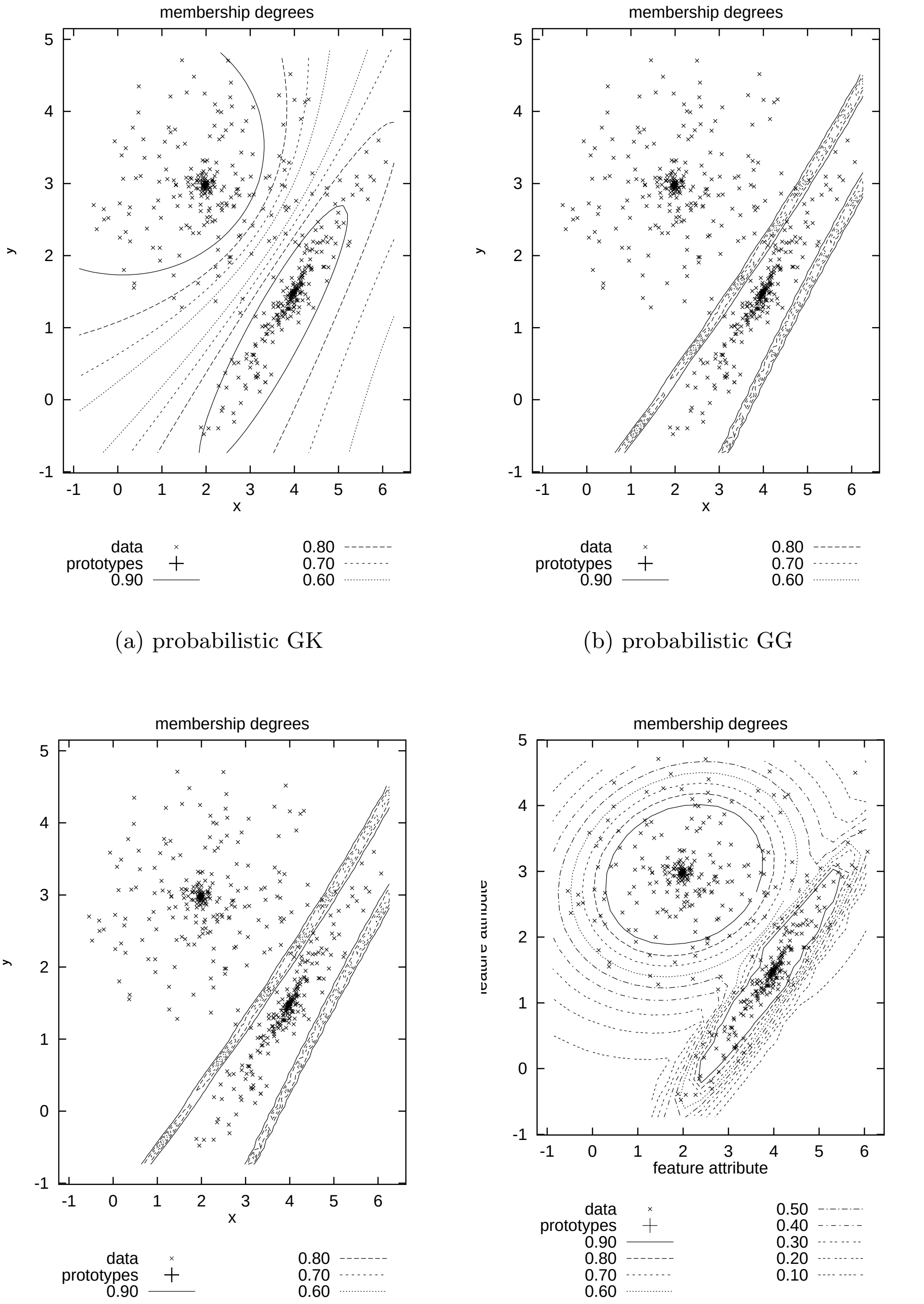
Now the equations for the alternating iteration procedure of the Gath-Geva algorithm are complete. The alternating optimisation schemes from Section 3.1 can be adapted in replacing the distance measure $d^2(\mathsf{v}_i, x_k)$ by $\mathcal{D}_{GG}$ and using Algorithm 3.8 as calculation step for the prototypes.

Algorithm 3.8 (Prototype Calculation for GG)

```
Calculate($\mathsf{v}_i$)
{
    $v_i = \frac{\sum_{k=1}^{n} u_{ik}^m \cdot x_k}{\sum_{k=1}^{n} u_{ik}^m};$
    $\pi_i = \frac{\sum_{k=1}^{n} u_{ik}^m}{\sum_{j=1}^{c} \sum_{k=1}^{n} u_{jk}^m};$
    $A_i = \frac{\sum_{k=1}^{n} u_{ik}^m \cdot (x_k - v_i)(x_k - v_i)^\top}{\sum_{k=1}^{n} u_{ik}^m};$
}
```

To obtain equations for the prototypes as a necessary condition for the optimisation function having a local minimum, the objective functions described in section 3.1 would have to be differentiated. Using $\mathcal{D}_{GG}$ as distance measure would lead to equations for which no analytical solution exists. Therefore, the estimation analogous to probability theory provides a good heuristic method.

In figure 3.11 the results for the basic probabilistic objective function with distance $\mathcal{D}_{GK}$ (figure 3.11(a)) and $\mathcal{D}_{GG}$ (figure 3.11(b)) are shown. For all classifications, the parameters have been set as follows: $c = 2$, $\epsilon = 0.001$



(a) probabilistic GK

(b) probabilistic GG

(c) noise GG clustering

(d) outlier GG ($\omega = n$, $q = 1$)

Figure 3.11: Clustering results for an elliptical test data set ($c = 2$, $\epsilon = 0.001$, $m = 2$)

and $m = 2$. The limitations of the distance $\mathcal{D}_{GK}$ can be seen on the membership degrees. In case of the GK, the algorithm is not able to adapt to the ellipsoidal structures of different sizes whereas the GG adapts well to the cluster structure. Figures 3.11(b) and 3.11(c) show the clustering result for distance measure $\mathcal{D}_{GG}$ in combination with the probabilistic and noise clustering algorithm, respectively. The GG separates the two clusters within a very small range, i.e. the range from membership degree 1 to one cluster over a small membership degree to both clusters too a membership degree of 1 to the second cluster is very small and can be interpreted as a separating line. In chapter 6 a clustering algorithm related to the GK but able to adapt to clusters of different extends is introduced. This new algorithm leads to smoother transitions as the GK membership degrees between clusters. In figure 3.11(d) the weighted membership degrees, i.e. $\tilde{u}_{ik}^m$ are illustrated for $q = 1$. The constraint parameter ω was in all cases chosen as the number of data. It can be seen, that outlier clustering is a possibility to overcome the strict cluster separation performed by GG in combination with the probabilistic or noise objective function.

The axes parallel Gath-Geva algorithm

Since GG is able to adapt to hyper-ellipsoidal forms as well as to different cluster sizes, the same problems as with the Gustafson-Kessel algorithm arise in the task of rule learning. As with GK it is possible to restrict this approach to detect axes-parallel hyper-ellipsoids [70, 73]. The corresponding algorithm is called *axes-parallel Gath-Geva algorithm* (*AGG*). Nevertheless, the axes-parallel version of the algorithm based on the technique introduced by Gath and Geva is able to adapt to different cluster sizes. In this case the distance measure, illustrated in Figure 3.10(a) can be rewritten as

$$d^2(\mathsf{v}_i, x_k) \;=\; \mathcal{D}_{AGG} \;=\; \frac{1}{\pi_i}\sqrt{\prod_{\nu=1}^{p} a_i^{(\nu)}} \cdot exp^{\frac{1}{2}\cdot\left(\sum_{\nu=1}^{p}(x_k^{(\nu)}-v_i^{(\nu)})^2\cdot\frac{1}{a_i^{(\nu)}}\right)}. \qquad (3.26)$$

Equivalent to the AGK, $a_i^{(\nu)}$ denotes the ν'th diagonal element of A_i and all other elements of A_i are 0 in case of axes-parallel (hyper-)ellipsoids. Again, p denotes the dimensionality of the data, $x_k^{(\nu)}$ and $v_i^{(\nu)}$ designate the ν'th component of the k'th data point, i'th cluster centre, respectively. In the alternating optimisation the covariance matrices parameter calculation can be simplified in the following way

$$a_i^{(\gamma)} = \frac{\sum_{k=1}^{n} u_{ik}^m \cdot (x_k^{(\gamma)} - v_i^{(\gamma)})^2}{\sum_{k=1}^{n} u_{ik}^m}, \qquad (3.27)$$

where $a_i^{(\gamma)}$ denotes the γ'th diagonal element of the covariance matrix. Parameter π_i is estimated again as denoted in (3.24). $\mathcal{D}_{AGG}$ could be used as distance measure in the membership update equations from section 3.1. The resulting prototype calculation instruction can be outlined as in Algorithm 3.9. For the technique introduced by Gustafson and Kessel the covariance matrices of the single clusters have to be reversed in each iteration of the algorithm. The same applies to GG. Again the axes parallel Gath-Geva algorithm is not only better suited to cope with the task of rule learning than the original GG but also drastically reduces the computationally effort. In case of rule learning the membership degrees are projected to the single domains in order to determine fuzzy sets for the single domains. A reverse projection of fuzzy sets for single attributes into the multidimensional domain usually leads to angular cluster structures. Axes parallel (hyper-) ellipsoids coincide better with this angular structure than rotated ellipsoids usually received by GG.

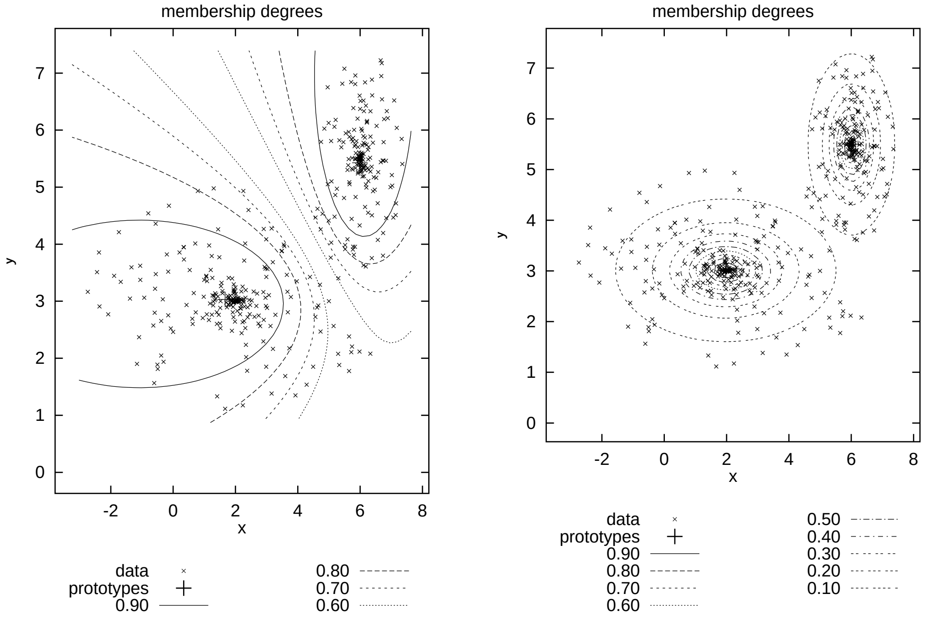
Algorithm 3.9 (Prototype Calculation for AGG)

```
Calculate($\mathsf{v}_i$)
{
    $v_i = \frac{\sum_{k=1}^{n} u_{ik}^m \cdot x_k}{\sum_{k=1}^{n} u_{ik}^m};$

    $\pi_i = \frac{\sum_{k=1}^{n} u_{ik}^m}{\sum_{j=1}^{c} \sum_{k=1}^{n} u_{jk}^m};$

    for all $\gamma \in \{1, \dots, p\}$
        $a_i^{(\gamma)} = \frac{\sum_{k=1}^{n} u_{ik}^m \cdot (x_k^{(\gamma)} - v_i^{(\gamma)})^2}{\sum_{k=1}^{n} u_{ik}^m};$
}
```

The results for the basic probabilistic objective function with distance $\mathcal{D}_{AGK}$ (figure 3.12(a)) and with distance $\mathcal{D}_{AGG}$ (figure 3.13(a)) are shown. For all classifications, the parameters have been set as follows: $c = 2$, $\epsilon = 0.001$ and $m = 2$. The limitations of the distance $\mathcal{D}_{AGK}$ can be seen on the membership degrees. In case of the probabilistic AGK, the algorithm is not able to adapt to the ellipsoidal structures of different sizes whereas the possibilistic AGK classifies a great number of data as outliers – see the outer membership degree line for each ellipsis in figure 3.12(b). The AGG adapts well to the cluster structure. Figures 3.13(a), 3.13(b) and 3.13(c) show the clustering result for distance measure $\mathcal{D}_{AGG}$ in combination with the probabilistic, noise and outlier clustering algorithm, respectively. As we have seen for the GG, the AGG separates the two clusters within a very small range, i.e. the range from membership degree 1 to one cluster over a small membership degree to both clusters to a membership degree of 1 to the second cluster is very small and can be interpreted as a separating line or curve. In figure 3.13(c) the weighted membership degrees $\tilde{u}_{ik}^m$ calculated by outlier clustering are illustrated. Parameter ω was chosen as the number of clusters. Again the combination of



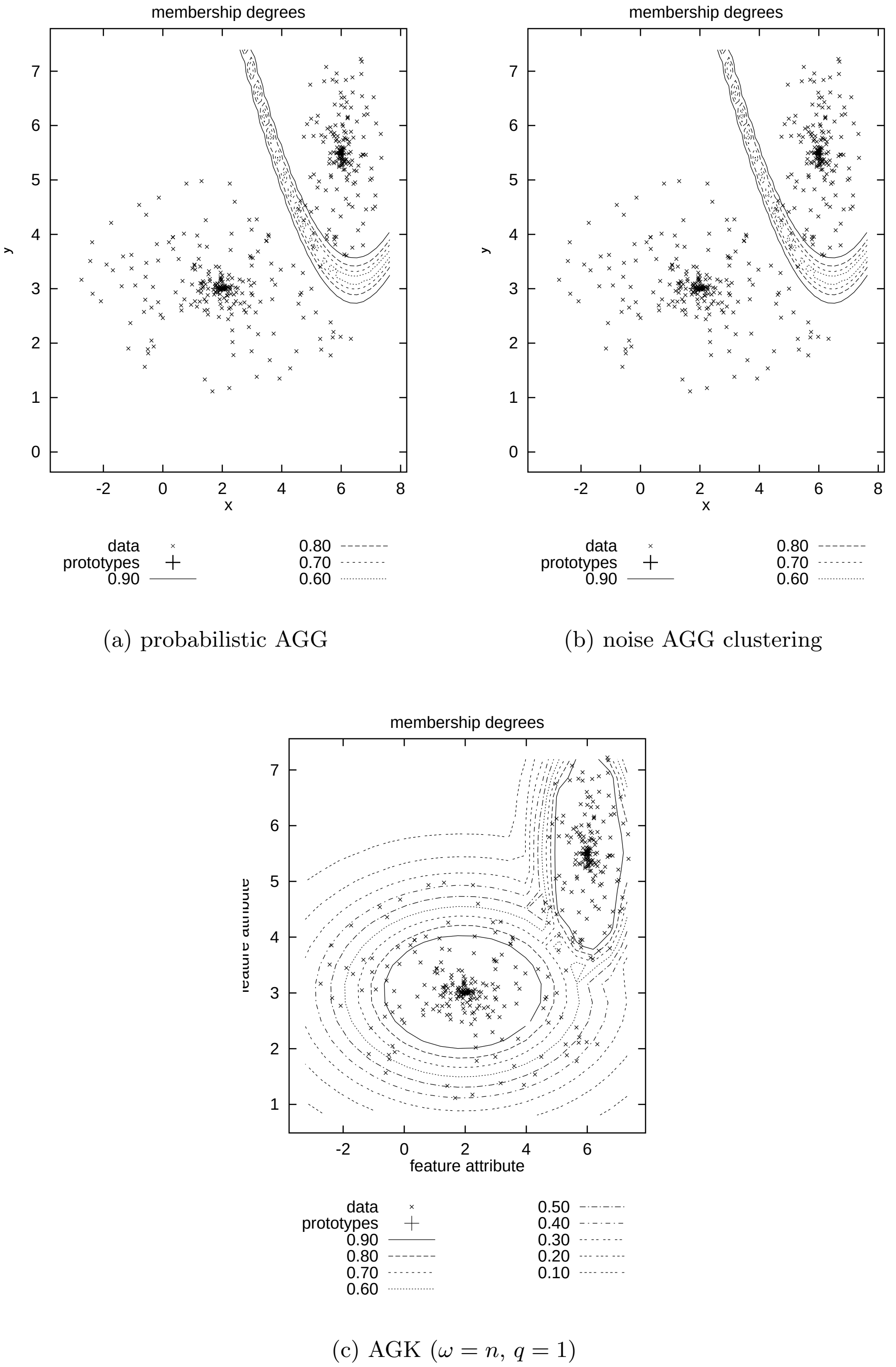
(a) probabilistic AGK

(b) possibilistic AGK

Figure 3.12: Clustering results for an elliptical test data set with GK ($c = 2$, $\epsilon = 0.001$, $m = 2$)

$\mathcal{D}_{AGK}$ with the basic outlier clustering technique is a possibility to cope with clusters of different sizes and overcome the strict separation of clusters in case of combining $\mathcal{D}_{AGK}$ with the probabilistic or noise objective function.

In [110] other clustering methods based on the maximum likelihood principle are described.



(a) probabilistic AGG

(b) noise AGG clustering

(c) AGK ($\omega = n$, $q = 1$)

Figure 3.13: Clustering results for an elliptical test data set with GG ($c = 2$, $\epsilon = 0.001$, $m = 2$)

3.3 Other Clustering approaches

In this section some fuzzy clustering techniques related to the methods described in Section 3.1 are briefly described. The scope of these methods differs from this work, but they have been the basis for some of the further on described new clustering techniques.

Fuzzy c-varieties

Besides the spherical or ellipsoidal cluster shapes described in this chapter also other forms can be detected by choosing a suitable distance function. For instance, the prototypes of the fuzzy c-varieties algorithm (FCV) describe linear subspaces, i.e. lines, planes and hyperplanes [15, 19, 28]. The equations for the prototypes of this algorithm require the computation of eigenvalues and eigenvectors of (weighted) covariance matrices. FCV can be applied to image recognition (line detection) and to construct local linear (fuzzy) models.

Alternating Cluster Estimation

In order to increase the influence of the user in extracting functional models from data, Runkler and Bezdek, see e.g. [99] and the references therein, have developed alternative approaches based on the presented basic ideas. They call the general clustering form with interchanging update equations for prototypes and membership degrees as presented above *alternating optimisation*. In one approach the expert has to specify the input space components in form of prototype parameters. In case of the fuzzy c-means algorithm (see section 3.2.1) the expert has to state prototype coordinates for each input domain. For other clustering algorithms also a suitable distance measure has to be chosen. The components for the output space are alternatingly updated during the optimisation phase of that algorithm. Runkler and Bezdek call this form of alternating optimisation *regular alternating optimisation, rAO*. Since some parameters are defined by the user and do not have to be updated during the alternating optimisation the computational effort is reduced.

By *alternating cluster estimation, ACE*, Runkler and Bezdek denote a clustering method where the expert has to select suitable membership function shapes and thereby defines the update equations for the cluster parameters.

The combination of both approaches where the expert has to specify suitable prototype parameters for the input domain as well as to choose suitable membership function shapes is called *regular alternating cluster estimation, rACE*. The algorithm generates a partition of the data and then evaluates the projections of the cluster centres into the output space.

Although the resulting functional models are easy to understand and reflect the experts interpretation of the modelled system, they are not necessarily based on objective functions. Problems may arise if the expert associates a system behaviour with the data and assigns suitable parameters for the clustering algorithm but the so defined model has a different basis. The greater the influence of the expert the greater are the restrictions of the associated functional model. Knowledge about unknown dependencies in the data is difficult to extract with these models, but under assumptions about the system behaviour these are computationally fast methods resulting in interpretable and easily understandable functional models.

Shell-Clustering

In contrast to the methods that are designed for solid clusters, the so-called *shell-clustering* algorithms are tailored for clusters in the form of boundaries of circles, ellipses, parabolas etc. (For an overview on shell clustering see [51, 78].) Davé [33] developed one of the first shell clustering algorithms for the detection of circles. Each prototype consists of the cluster centre v_i and the radius r_i. The distance function for the *fuzzy c-shells algorithm* (*FCS*) is

$$d^2((v_i, r_i), x_k) \; = \; \mathcal{D}_{FCS} \; = \; (\| \, x_k - v_i \, \| - r_i)^2$$

so that exactly those data have zero distance to the cluster that are located on the circle with radius r_i and centre v_i. Unfortunately, this distance function leads to a set of coupled non-linear equations for the v_i and r_i that cannot be solved in an analytical way. Thus an additional numerical iteration procedure to solve non-linear equations is necessary in each iteration step of the clustering algorithm which causes a high computational effort. Although this specific problem is solved for circles by the *fuzzy c-spherical shells algorithm* (*FCCS*) [81] using the distance function

$$d^2((v_i, r_i), x_k) \; = \; \mathcal{D}_{FCCS} \; = \; (\| \, x_k - v_i \, \|^2 - r_i^2)^2,$$

the general problem for other shell shapes remains.

Chapter 4

Unsupervised Fuzzy Clustering

For all fuzzy clustering approaches described in section 3 the number of clusters has to be specified in advance. If the *optimal* number is not known, a classification has to be calculated for each possible number of groups. How can we decide which classification represents the *best* partition? In the case of classified sample data it is possible to count the data that were assigned to the wrong cluster and choose that classification with the smallest *error rate* as optimal solution. Even in this case it has to be specified how the corresponding class of a particular cluster is determined. If we have to handle unclassified data, we have to search for other measures that say something about the classification's quality. Measures that try to value the whole classification are so-called global validity measures.

4.1 Global Validity measures

In this section some *global validity measures* that are especially suited for solid fuzzy clustering algorithms are presented. The basic idea of unsupervised fuzzy clustering is to define an upper bound for the number of clusters c_{max} and carry out the clustering for each number of clusters $c \in \{2, \ldots, c_{max}\}$. The global validity measures indicates the optimal number of clusters. Algorithm 4.1 indicates the alternating optimisation scheme that can be used in of unsupervised fuzzy clustering. The optimal number of clusters is indicated by c_{best}. It is useful to save not only the best number of clusters but to examine the curve of the validity measure over the number of clusters. Local extrema indicate good results. Reliable results are often found at local extrema with a great gradient to the neighbouring cluster numbers. Whether minima or maxima have to be considered depends on the chosen validity measure. In the following, $\mathcal{A}_*$ symbolises any of the basic clustering algorithms from section 3.1 that has to be combined with a suitable distance measure $\mathcal{D}_*$, whereas $\mathcal{V}_*$ denotes one of the following validity measures.

Algorithm 4.1 (Basic Unsupervised clustering algorithm)

```
Choose()
{
   $c_{max} \in \{2, \ldots, n-1\}$;
   validity measure $\mathcal{V}_*$;
   basic clustering algorithm $\mathcal{A}_*$;
   $\mathcal{A}_*$ :: Choose();
   distance measure $\mathcal{D}_*$;
}

Initialise()
{
   $c \in \{2, \ldots, c_{max}\}$;
   $c_{best} := c$;
}

Calculate()
{
   do
   {
      $\mathcal{A}_*$ :: CalculatePartition();
      $\mathcal{V}_*(c)$;
      if $\mathcal{V}_*(c)$ better than $\mathcal{V}_*(c_{best})$
         $c_{best} := c$;
      $c = c + 1$;
   }while $(c \leq c_{max})$;
}
```

In this section validity measures $\mathcal{V}_*$ that could be used to find the *optimal* number of clusters with the basic clustering algorithms from section 3.1 are described. For an overview on validity measures see also [25]. Some of the here presented validity measures rely on characteristics of special clustering algorithms, others can be used for solid classifications in general. In the following, $J^*(X, U, v)$ denotes any of the previously and further on described objective functions.

4.1.1 Error Rate

The error rate can be used in case of classified sample data to determine the percentage of wrong classified data. To determine the *optimal* number of clusters, the error rate is calculated using the sample data. To validate the clustering results, cross-validation with varying test data has to be carried out. In this case, the whole data set is split in two parts, the sample data and the test data. The clustering is then carried out for the sample data alone. After the calculation is finished and a suitable number of clusters has been found, the test data is used to determine resulting classification validity.

It has to be specified, how the class of each cluster $C(v_i)$ and the class determining cluster for each datum can be calculated. One possibility to obtain the class of a datum x_k – $C(x_k)$ – in a partition is, to choose that cluster v_i with the highest membership degree $u_k^{max} = \{u_{ik} \mid \forall i, j \in \{1, \ldots, k\} : u_{jk} < u_{ik}\}$ and assign the class of v_i to x_k. In case of outliers it would not always make sense to assign a partition class $C(x_k)$ to each datum. Therefore, a minimum membership degree u_k^{min} can be defined, where each datum with a smaller maximal membership degree than the given bound $u_k^{max} < u_k^{min}$ is assigned to a outlier class. There are other possibilities to determine $C(x_k)$, e.g. the single cluster's classes – $C(v_i)$ – can be weighted with the membership degrees u_{ik} and that class with the greatest sum of weights can be used as $C(x_k)$.

The remaining problem is to determine the class of a single cluster. Therefore, the classes given in the sample data – $SC(x_k)$ – have to be considered. Again these classes can be weighted with the membership degrees u_{ik} for each cluster i and the class with the highest over all membership degree can be used as the cluster's class $C(v_i)$. Let us assume, that the sample data consists of s classes. For each cluster, the class can be defined in the following way Let $U(s, i)$ be the sum of membership degrees of the sample data with $SC(x_k) = s$ to cluster i, i.e.

$$U(l, i) = \sum_{k \in \{1, \ldots, n\};\ SC(x_k) = l} u_{ik}.$$

Using this definition, the class of a cluster is defined as follows

$$C(v_i) = f \quad \text{where} \quad U(f, i) = \max_{l \in \{1, \ldots, s\}} \{U(l, i)\}.$$

It is not guaranteed that this definition is definite. If $U(l, i)$ is equal for different classes l we choose the first of the ordered classes $s \in \{1, \ldots, l\}$. A random choice between the equally suited classes is also possible. Since we determine the wrong classified data, we have to look for *minimal* values for the error rate. A method with less computational effort is to assign the class of the datum x_k with the greatest membership degree u_{ik} – i.e. u_i^{max} – to cluster i. In this way $C(v_i) = u_i^{max}$.

With these definitions, the error rate is defined as follows

$$\mathcal{V}_{ER} = \frac{\sum_{k=1}^{n} C(x_k)}{n} \tag{4.1}$$

with

$$C(x_k) = C(v_i) \quad \text{where} \quad u_{ik} = u_k^{max}.$$

Figure 4.1 shows the error rate for the iris data [26] and the fuzzy c-means clustering algorithm. The iris data was split in two parts: 90% has been

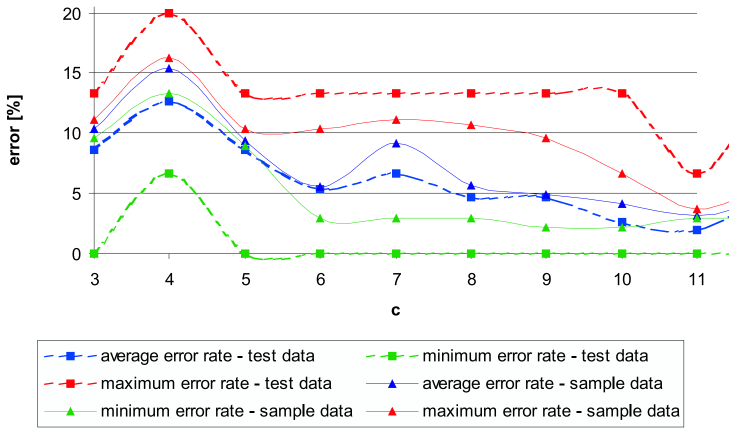


Figure 4.1: Error rate for the Iris Data and FCM

used as sample data and 10% was left as test data. The wrong classified data points have been determined separately for the test data and the sample data after the clustering algorithm terminated for a particular number of clusters. Since the iris data set is classified in three sub-species, we started the fuzzy c-means clustering with $c = 3$ clusters. Only the sample data was used for the clustering task. In figure 4.1 the results for the sample and test data are illustrated. We carried out FCM 10 times for each number of clusters. The best, the worst, and the average error rates are shown for $c = 3, \ldots, 11$ clusters in the figures.

4.1.2 Partition Coefficient

This validity measure rates the crispness of a classification. The more crisp the membership degrees the better the classification [15]. The partition coefficient has to be *maximised*.

$$\mathcal{V}_{PC} = \frac{\sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^2}{n} \tag{4.2}$$

An example for the partition coefficient is illustrated in figure 4.2. For the figure the fuzzy c-means algorithm has been used to partition the data set from section 3.2.1 for a varying number of clusters.

4.1.3 Partition Entropy

The partition entropy is similar to the partition coefficient. The idea for this measure was derived from Shannon's information theory [15]. An optimal classification provides a *minimal* value for the partition entropy.

$$\mathcal{V}_{PE} = -\frac{\sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik} \cdot \ln(u_{ik})}{n} \tag{4.3}$$

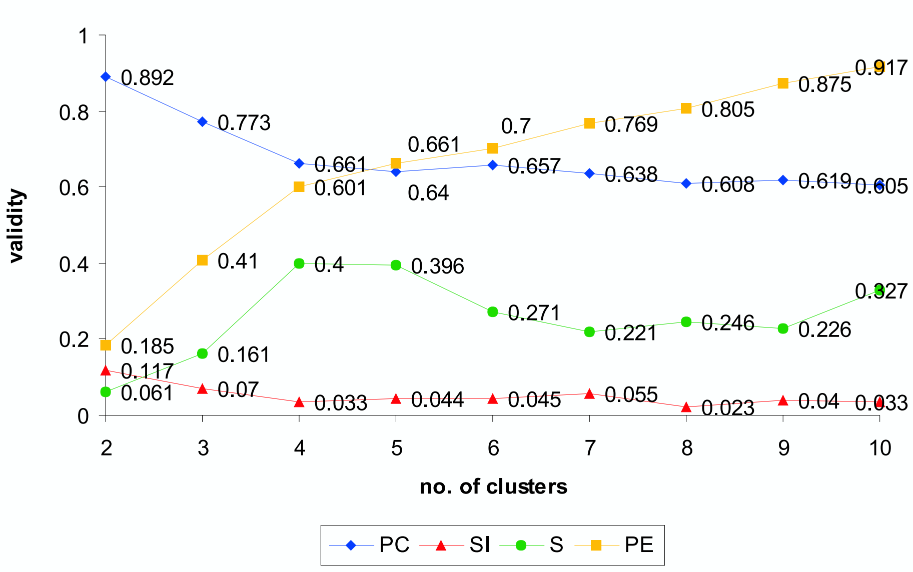


Figure 4.2: Validity measures for FCM and the sample dataset with two clusters

The partition entropy is illustrated in figure 4.2 for the data set from section 3.2.1.

4.1.4 Separation Measure

This validity measure was introduced in [113]. This measure depends not only on the membership degrees but also on the distance measure corresponding to the chosen clustering algorithm. The separation measure has to be *minimised*.

$$\mathcal{V}_S = \frac{\sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^2 \cdot \mathcal{D}_*(v_i, x_k)}{n \cdot \min\{\mathcal{D}_*(v_i, v_j) \mid i, j \in \{1, \ldots, c\}\}} \tag{4.4}$$

The separation measure is illustrated in figure 4.2.

4.1.5 Separation Index

The separation index evaluates the relation between the smallest distance between two clusters and the maximal diameter of all clusters [40]. Originally this measure was designed for hard partitions. The distance is weighted with the membership degrees to use this measure for fuzzy partitions. The separation index has to be *maximised*.

$$\mathcal{V}_{SI} = \min_{k \in \{1,\ldots,c\}} \{ \min_{l \in \{1,\ldots,c\} \wedge l \neq k} \{ \frac{\min\limits_{k,l \in \{1,\ldots,k\}} \{\frac{\mathcal{D}_*(x_k, x_l)}{u_{ik} \cdot u_{jl}}\}}{\max\limits_{h \in \{1,\ldots,c\}} \{ \max\limits_{k,l \in \{1,\ldots,k\}} \{u_{hk} \cdot u_{hl} \cdot \mathcal{D}_*(x_k, x_l)\}\}} \}\} \tag{4.5}$$

An example for the separation index is shown in figure 4.2.

4.1.6 Fuzzy Hypervolume

Together with their clustering algorithm, Gath and Geva have proposed three validity measures that make use of the algorithms covariance matrix, see section 3.2.3 or [42].

A *minimum* of the first measure called fuzzy hypervolume denotes small compact clusters. In probabilistic clustering it is guaranteed that the whole data set is covered by clusters, so a small sum of all cluster sizes is desirable.

$$\mathcal{V}_{FHV} = \sum_{i=1}^{c} \sqrt{det(A_i)}. \tag{4.6}$$

An example for the fuzzy hypervolume is illustrated in figure 4.3. For the figure the Gustafson-Kessel algorithm has been applied to partition the data set from section 3.2.2 for a varying number of clusters.

To calculate the covariance matrix only the data vectors, cluster centres and membership degrees are used. These parameters are used in all described clustering techniques so that the validity measures using the covariance matrix can be applied to all presented algorithms.

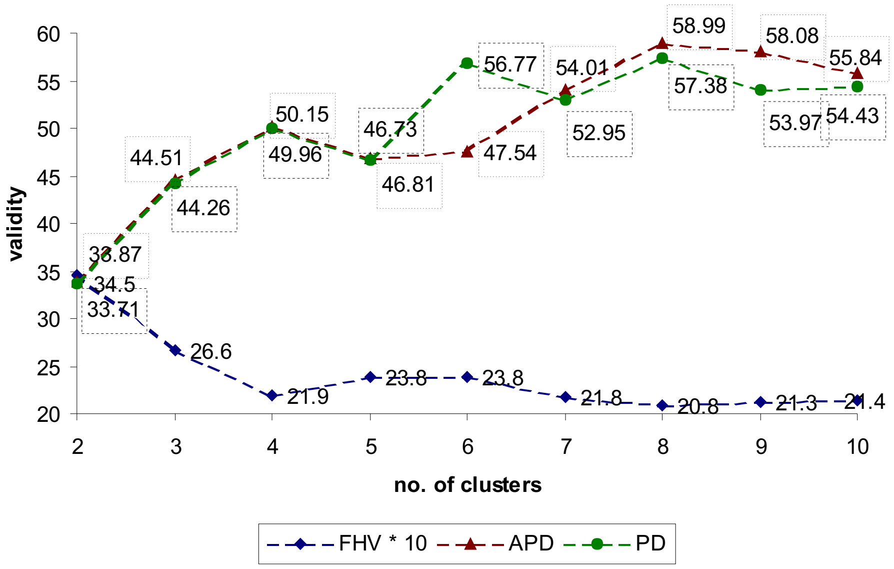


Figure 4.3: Validity measures for GK

4.1.7 Average Partition Density

The second measure defined by Gath and Geva determines the average physical density of the clusters. It can be described as the average number of data in the cluster's centre weighted by the cluster's volume.

$$\mathcal{V}_{APD} = \frac{1}{c} \cdot \sum_{i=1}^{c} \frac{\sum_{k \in R_i} u_{ik}}{\sqrt{det(A_i)}}, \tag{4.7}$$

where $R_i = \{k \in \mathbb{N}_{\leq n} \mid (x_k - v_i)^\top A_i^{-1} (x_k - v_i) < 1\}$.

The average partition density is illustrated in figure 4.3 for the data set from section 3.2.2 and a varying number of clusters.

4.1.8 Partition Density

The third measure is equivalent to the average partition density with the exception that the densities are simply summed up and the average is not evaluated.

$$\mathcal{V}_{PD} = \frac{\sum_{i=1}^{c} \sum_{k \in R_i} u_{ik}}{\sum_{i=1}^{c} \sqrt{det(A_i)}}, \tag{4.8}$$

where $R_i = \{k \in \mathbb{N}_{\leq n} \mid (x_k - v_i)^\top A_i^{-1} (x_k - v_i) < 1\}$.

In figure 4.3 the partition density is illustrated for the data set from section 3.2.2.

4.2 Local Validity measures

In contrast to global validity measures, where the whole partition of data into groups is evaluated, local validity measures rate single clusters. The idea is to reduce the computational effort necessary in unsupervised clustering with global validity measures.

Especially shell-clustering techniques tend to result in local optimal solutions. Here, local validity measures help to separate satisfying clusters from not well covered data. The poor classified data can be identified for further analysis.

The idea of local validity measures is to start with a relatively large number of clusters c_{max}. Here, this upper bound is chosen much larger than the number of clusters in the resulting partition. The clustering algorithm is carried out for c_{max} clusters leading to a small number of data per cluster. The aim is to avoid that large amounts of data coming from different groups are covered by one (e.g. very large) cluster. To estimate the in some way *optimal* number of clusters, the single groups are compared to identify similar ones. One way to reduce the number of clusters is to merge similar clusters where possible. Additionally, good clusters can be removed from the data before the remaining data set is further analysed.

Compatible cluster merging (*CCM*) was introduced by Krishnapuram and Freg [76] to detect an unknown number of lines. The Gustafson-Kessel algorithm as well as the Gath-Geva algorithm or other line detecting algorithms are well suited for CCM.

The basic concept of CCM is to first carry out the chosen clustering algorithm for the predefined maximal number of clusters. In a following iteration procedure the clusters are grouped regarding a defined *compatibility relation*. The clusters grouped together are replaced by one resulting cluster and equivalently the total number of clusters is reduced. Then the clustering algorithm is carried out for the resulting number of clusters and using the merged clusters as initialisation. If no more compatible clusters are detected, the calculation is finished. In case of GK or GG, compatible clusters should belong to the same hyperplane, i.e. be part of the same line in the 2-dimensional case, and correspondingly to their expansion be close to each other. A compatibility relation can be defined using the eigenvalues and -vectors of the covariance matrix and additionally the cluster centres, see e.g. [51, 76]. The approach of robust competitive cluster merging (RCM) is based on CCM and can be applied to the FCM family of algorithms, see [41].

Other local validity measures for line or circle detection especially used in image recognition are described in [81, 78]. A comparison of these approaches can be found in [51]. For some techniques further improvements can be achieved thinning the resulting lines or line segments as shown in [77].

Chapter 5

Scalar Product-Based Distance Measures

The distance measures described in section 3.2 assume that all data points are equally meaningful. If we think e.g. about a technical system where especially abnormal behaviour is of interest, most of the sample data could be classified as normal and only that small data part, indicating difficulties in the system state, should be divided in meaningful groups. Often these unusual states can be identified as outliers in the data set. In this chapter a modified distance measure is introduced, that is applied to the basic objective functions from section 3.1. The aim is to locate the usual data points in one group, therefore distances in the centre of gravity of the whole data set should be small, whereas other data points should obtain a greater distance from the centre as well as from one another. A suitable distance measure is derived from the scalar product [68].

5.1 Clustering with Angle-Based Distances for Normalised Data

The idea of this approach is very similar to the original neural network competitive learning approach as it is for instance described in [93]. Instead of the Euclidean distance between a class representative and a given datum used by Kohonen's self organising feature maps, the simple competitive learning approach computes the scalar product of these vectors.

In the following this approach is referred to as *normalised angle-based clustering* (*NAB*). For normalised vectors the scalar product is simply the cosine of the angle between the two vectors, i.e. the scalar product is one if and only if the (normalised) vectors are identical, otherwise we obtain values between -1 and 1. Therefore, we define as the (modified) distance between a normalised prototype vector v and a normalised data vector x

$$d^2(\mathsf{v}_i, x_k) \;=\; \mathcal{D}_{NAB} \;=\; 1 - v_i^\top x_k. \tag{5.1}$$

Thus we have $0 \leq \mathcal{D}_{NAB} \leq 2$ and, in case of normalised vectors, $\mathcal{D}_{NAB} = 0 \Leftrightarrow x_k = v_i$.

Let us for the moment assume that the data vectors are already normalised. How we actually carry out the normalisation will be discussed later on. With the distance function (5.1) the objective function (3.1) becomes

$$J^{prob}(X,U,\mathsf{v}) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m}(1 - v_i^{\top} x_k) \tag{5.2}$$

$$= \sum_{i=1}^{c}\sum_{k=1}^{n}\left(u_{ik}^{m} - u_{ik}^{m}\sum_{\nu=1}^{p} v_i^{(\nu)} x_k^{(\nu)}\right) \tag{5.3}$$

where $v_i^{(\nu)}$ and $x_k^{(\nu)}$ is the ν'th coordinate/component of vector v_i and x_k, respectively.

Theorem 5.1 (Normalised Angle-Based Prototypes)

By taking into account the constraint that the prototype vectors v_i have to be normalised, i.e.

$$\| v_i \|^2 = \sum_{\nu=1}^{p}\left(v_i^{(\nu)}\right)^2 = 1, \tag{5.4}$$

in differentiating the objective function (5.2) we can derive prototype update equations, see 5.1.

$$v_i^{(\nu)} = \frac{\sum_{k=1}^{n} u_{ik}^{m} x_k^{(\nu)}}{\sqrt{\sum_{\gamma=1}^{p}\left(\sum_{k=1}^{n} u_{ik}^{m} x_k^{(\gamma)}\right)^2}}. \tag{5.5}$$

Proof 5.1 (Normalised Angle-Based Prototypes)

Considering constraint (5.4), we obtain the Lagrange function

$$J_\lambda^{prob}(X,U,\mathsf{v}) = \sum_{i=1}^{c}\sum_{k=1}^{n}\left(u_{ik}^{m} - u_{ik}^{m}\sum_{\nu=1}^{p} v_i^{(\nu)} x_k^{(\nu)}\right) + \sum_{j=1}^{c}\lambda_j\left(\sum_{\nu=1}^{p}\left(v_j^{(\nu)}\right)^2 - 1\right).$$

The partial derivative of J_λ^{prob} w.r.t. $v_i^{(\nu)}$ yields

$$\frac{\partial J_\lambda}{\partial v_i^{(\nu)}} = -\sum_{k=1}^{n} u_{ik}^{m} x_k^{(\nu)} + 2\lambda_i v_i^{(\nu)}.$$

Since the first derivative has to be zero in a minimum, we obtain

$$v_i^{(\nu)} = \frac{1}{2\lambda_i}\sum_{k=1}^{n} u_{ik}^{m} x_k^{(\nu)}.$$

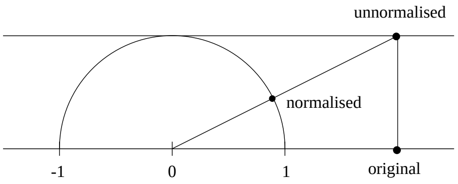


Figure 5.1: Normalisation of a datum

Making use of the constraint (5.4), we have

$$1 \;=\; \frac{1}{4\lambda_i^2}\sum_{\nu=1}^{p}\left(\sum_{k=1}^{n} u_{ik}^m x_k^{(\nu)}\right)^2 ,$$

which gives us

$$2\lambda_i \;=\; \sqrt{\sum_{\ell=1}^{p}\left(\sum_{k=1}^{n} u_{ik}^m x_k^{(\nu)}\right)^2}$$

so that we finally obtain

$$v_i^{(\nu)} \;=\; \frac{\sum_{k=1}^{n} u_{ik}^m x_k^{(\nu)}}{\sqrt{\sum_{\gamma=1}^{p}\left(\sum_{k=1}^{n} u_{ik}^m x_k^{(\gamma)}\right)^2}}. \quad \diamond$$

For this formula we have assumed that the data vectors are normalised. When we simply normalise the data vectors, we loose information, since collinear vectors are mapped to the same normalised vector. In order to avoid this effect we extend the data vectors by one component which is set to 1 for all data vectors and normalise these $(p+1)$-dimensional data vectors. In this way, the data vectors in $\mathbb{R}^p$ are mapped to the upper half of the unit sphere in $\mathbb{R}^{p+1}$. Figure 5.1 illustrates the normalisation for one-dimensional data. Algorithm 5.1 shows the procedure *Calculate*(v_i) where in a first step the mapping of the data vectors $x_k \in \mathbb{R}^p$ to the $(p+1)$-dimensional space is carried out. After the calculation of the normalised cluster centres v_i, these prototypes are mapped to the p-dimensional space in a last step, e.g. for illustration purposes. To calculate the membership degrees, the $(p+1)$-dimensional normalised data vectors $\tilde{x}_k$ and cluster centres $\tilde{v}_i$ have to be used in the distance measure.

Algorithm 5.1 (Prototype Calculation for NAB)

```
Calculate($\mathsf{v}_i$)
{
  for all $k \in \{1, \ldots, n\}$
  {
    for all $\nu \in \{1, \ldots, p\}$
      $\tilde{x}_k^{(\nu)} = \frac{x_k^{(\nu)}}{\sqrt{1 + \sum_{\gamma=1}^{p} (x_k^{(\gamma)})^2}}$;
      $\tilde{x}_k^{(p+1)} = \frac{1}{\sqrt{1 + \sum_{\gamma=1}^{p} (x_k^{(\gamma)})^2}}$;
  }
  for all $\nu \in \{1, \ldots, p+1\}$
    $\tilde{v}_i^{(\nu)} = \frac{\sum_{k=1}^{n} u_{ik}^m \cdot \tilde{x}_k^{(\nu)}}{\sqrt{\sum_{\gamma=1}^{p+1} \left( \sum_{k=1}^{n} u_{ik}^m \cdot \tilde{x}_k^{(\gamma)} \right)^2}}$;
  for all $\nu \in \{1, \ldots, p\}$
    $v_i^{(\nu)} = \frac{\tilde{v}_i^{(\nu)}}{\tilde{v}_i^{(p+1)}}$;
}
```

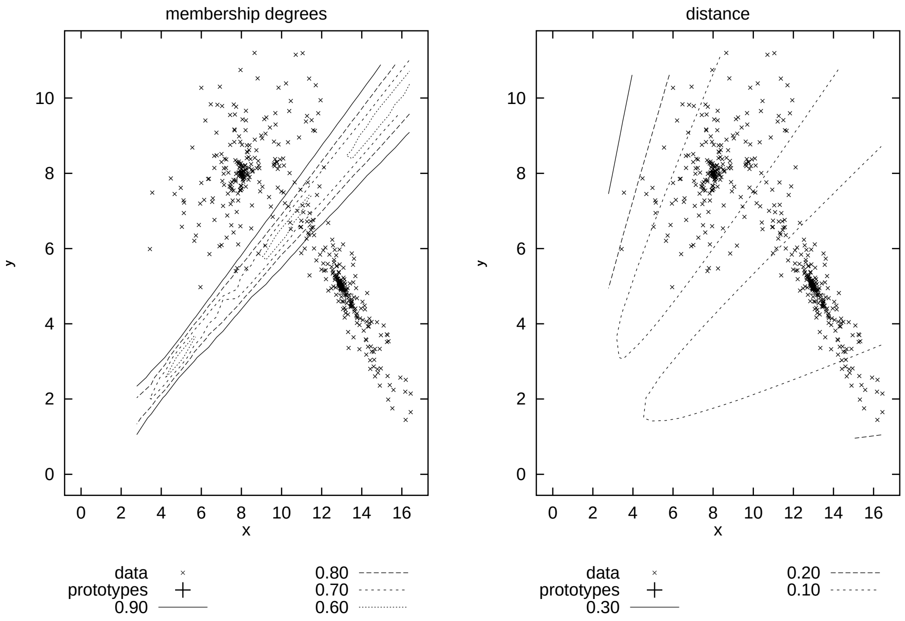
Figure 5.2 shows a clustering result for a two-dimensional data set (i.e. the clustering is actually carried out on the normalised three-dimensional data). The membership degrees are illustrated together with the sample data. In this case, the fuzzifier m has been set to 1.5.

It has to be noted that the distance function is not affine invariant. We can already see in figure 5.1 that vectors near zero keep almost their Euclidean distance when we normalise them, whereas very long vectors are all mapped to the very lower part of the semi-circle.

Figure 5.3 shows distance values of two one-dimensional vectors. (The distance is computed for the normalised two-dimensional vectors.) Of course, the distance is zero at the diagonal and increases when we go away from the diagonal. But the distance is increasing very quickly with the distance to the diagonal near zero, whereas it increases slowly, when we are far away from the origin.

Figures 5.4 and 5.5 also illustrate this effect. In Figure 5.4 the distance to the (non-normalised) two-dimensional vector (cluster centre) $(0,0)^\top$ is shown. It is a symmetrical distance function. However, when we replace the cluster centre $(0,0)^\top$ by the vector $(1,0)^\top$, we obtain the function in figure 5.5.

Here we can see that the distance is asymmetrical in the sense that it



(a) probabilistic membership degrees (b) probabilistic clustering distance

Figure 5.2: A two-dimensional data set and the partition in two clusters

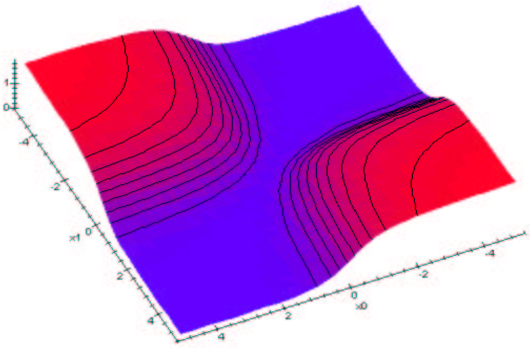


Figure 5.3: The one-dimensional distance from $x0$ to $x1$

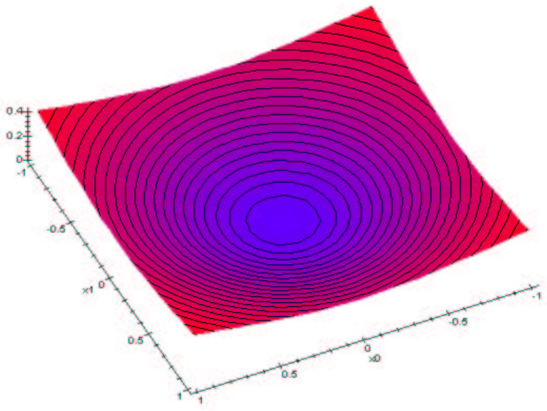


Figure 5.4: Distance to the point $v^\top = (0, 0)$

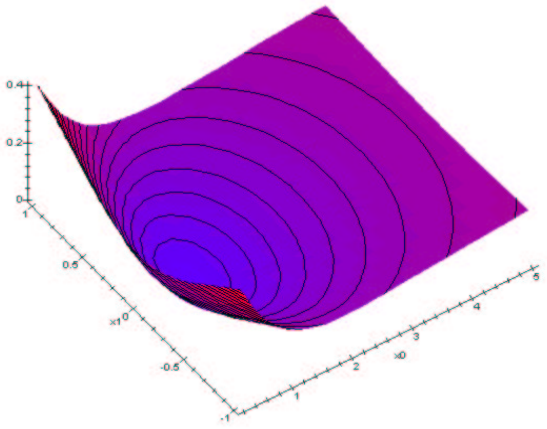


Figure 5.5: Distance to the point $(1, 0)^\top$

increases faster when we look in the direction of $(0,0)^\top$. This can be an undesired effect for certain data sets. But there are also data sets for which this effect has a positive influence on the clustering result. Consider for instance data vectors with the annual salary of a person as one component. When we simply normalise each component, the effect is that a few outliers (persons with a very high income) force that almost all data are normalised to values very near to zero. This means that the great majority simply collapses to one cluster (near zero) and few outliers build single clusters. Instead of a standard normalisation, we can also choose a logarithmic scale in order to avoid this effect. But the above mentioned clustering approach offers an interesting alternative.

5.2 Clustering with Angle-Based Distances for Non-normalised Data

In the previous section we have assumed that the data vectors are normalised or that we normalise them for the clustering. In this section we discuss what happens, when we refrain from normalising the data vectors and the cluster centres. Using non-normalised data vectors and cluster centres leads to a clustering technique with completely different capabilities. The resulting technique is able to detect clusters in form of lines in a 2-dimensional domain respectively hyperplanes in multi-dimensional domains. Thereby the needed computational effort is less than for other comparable techniques. This approach is called *angle-based clustering* (*AB*) in the following. In order to avoid negative distances, we have to modify the distance function to

$$d^2(\mathsf{v}_i, x_k) \;=\; \mathcal{D}_{AB} \;=\; (1 - v_i^\top x_k)^2. \tag{5.6}$$

The geometrical meaning of this distance function is the following. A datum x_k has distance zero to the cluster v_i, if and only if $v_i^\top x_k = 1$ holds. This equation describes a hyperplane, i.e. the hyperplane of all $x_k \in \mathbb{R}^p$ of the form

$$\frac{v_i}{\| v_i \|^2} + \sum_{\nu=1}^{p-1} \lambda_\nu w_\nu$$

where the vectors $w_1, \ldots, w_{p-1} \in \mathbb{R}^p$ span the hyperplane perpendicular to v_i and $\lambda_1, \ldots, \lambda_{p-1} \in \mathbb{R}$.

This means that we can find clusters in the form of linear varieties like the FCV algorithm [15, 28]. We will return to a comparison of FCV and this approach later on. Figure 5.6 shows the distance to the prototype $v_i^\top = (0.5, 0)$. This prototype describes the line

$$\begin{pmatrix} 2 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

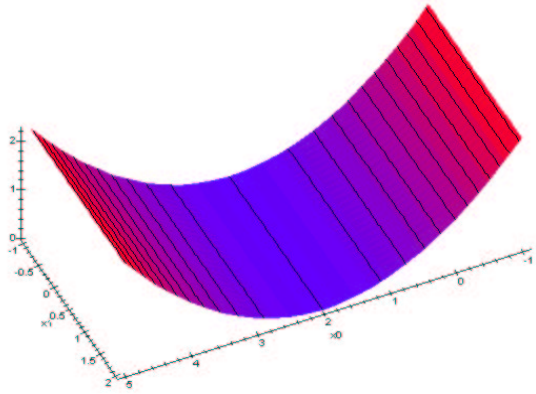


Figure 5.6: Distance between $v^\top = (0.5, 0)$ and $x^\top = (x0, x1)$

In order to derive equations for the prototypes we insert the distance function (5.6) into the objective function (3.1). This leads to

$$J^{prob}(X, U, \mathsf{v}) \;=\; \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m}(1 - v_i^\top x_k)^2.$$

Theorem 5.2 (Angle-Based Prototypes)

Prototype update equations for the alternating optimisation scheme are derived due to proof 5.2

$$v_i \;=\; \sum_{k=1}^{n} u_{ik}^{m} x_k \cdot \left(\sum_{k=1}^{n} u_{ik}^{m} x_k x_k^\top\right)^{-1}. \tag{5.7}$$

Proof 5.2 (Angle-Based Prototypes)

The first derivative w.r.t. $v_i^{(\nu)}$ is taken

$$\frac{\partial J^{prob}(X, U, \mathsf{v})}{\partial v_i^{(\nu)}} \;=\; -2\sum_{k=1}^{n} u_{ik}^{m}(1 - v_i^\top x_k)x_k^{(\nu)} \overset{!}{=} 0.$$

These derivatives have to be zero at a minimum and we obtain the system of linear equations

$$\sum_{k=1}^{n} u_{ik}^{m}(1 - v_i^{\top} x_k)x_k \;=\; 0.$$

Note that the matrix $\sum_{k=1}^{n} u_{ik}^{m} x_k x_k^{\top}$ *is the (weighted) covariance matrix and can therefore be inverted unless the data are degenerated. Making use of the fact that* $(v_i^{\top} x_k)x_k = (x_k x_k^{\top})v_i$ *holds, we obtain for the prototypes*

$$v_i \;=\; \sum_{k=1}^{n} u_{ik}^{m} x_k \cdot \left(\sum_{k=1}^{n} u_{ik}^{m} x_k x_k^{\top} \right)^{-1} . \quad \diamond$$

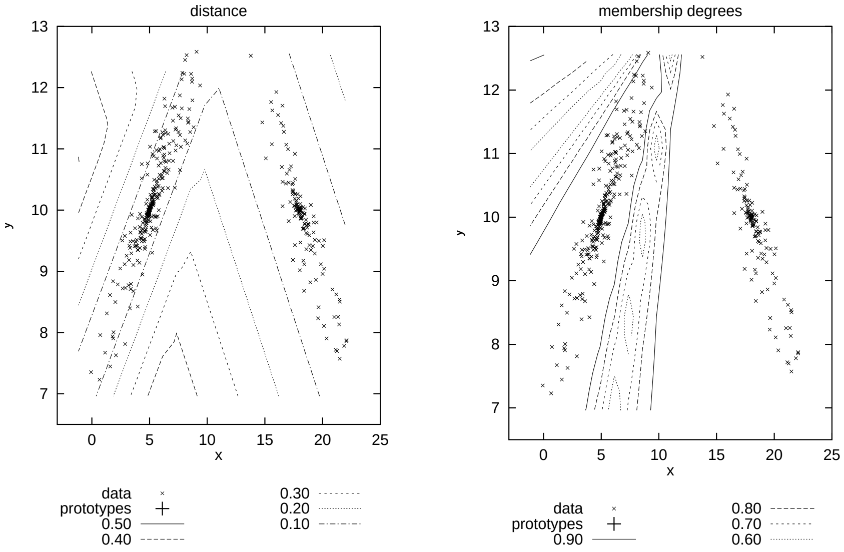
The advantage of this approach in comparison to the FCV algorithm is in the computing scheme that requires inverting a matrix whereas for the FCV algorithm all eigenvalues and eigenvectors have to be computed. Another difference is caused by the non-Euclidean distance function used here that is again not affine invariant. Problems can arise when lines are near to $(0,0)^{\top}$, since then the corresponding prototype vector v is very large, and even small deviations from the linear cluster lead to large distances. These problems are well known for other fuzzy clustering algorithms with non-Euclidean distance functions [78] and have to be treated in a similar way.

Algorithm 5.2 shows the step *Calculate*(v_i) that can be used with the basic probabilistic objective function from section 3.1.

Algorithm 5.2 (Prototype Calculation for AB)

```
Calculate($\mathsf{v}_i$)
{
    $v_i = \sum_{k=1}^{n} u_{ik}^{m} x_k \cdot \left( \sum_{k=1}^{n} u_{ik}^{m} x_k x_k^{\top} \right)^{-1}$;
}
```

An example of the detection of two linear clusters is shown in figure 5.7 for the probabilistic algorithm. For a comparison with the possibilistic, noise, and outlier clustering algorithms see A.1. As fuzzifier $m = 1.5$ has been chosen for all angle-based clustering tasks. The first figure (5.7(a)) shows the distance of the data to the cluster centres, whereas the corresponding membership degrees are illustrated in figure 5.7(b). The line that is defined by the prototype coordinates is illustrated in figure 5.8 for both clusters.



(a) probabilistic clustering - distance

(b) probabilistic clustering - membership degrees

Figure 5.7: Results for an elliptical test data set and angle-based clustering ($m = 1.5, \epsilon = 0.001$)

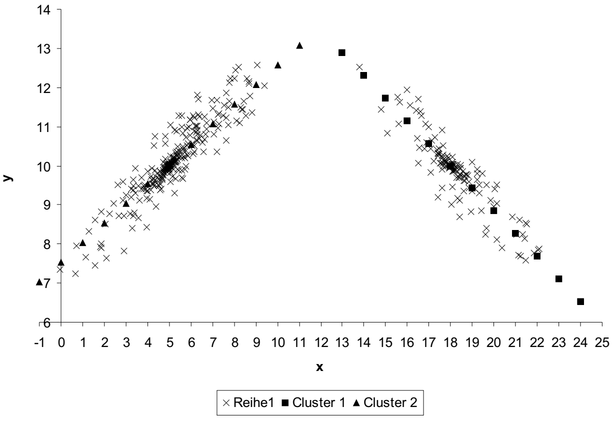


Figure 5.8: Lines defined by the cluster centres of probabilistic clustering

Chapter 6

Adaptation of Cluster Volumes

The objective function based fuzzy clustering algorithms from section 3.2 assume that all groups in one partition have at least nearly the same size. The algorithm by Gath and Geva, see 3.2.3 (GG), is in someway able to adapt to the cluster sizes, but is heuristically derived from statistics and no longer objective function based.

This chapter discusses new approaches in objective function based fuzzy clustering extending some of the clustering techniques described in 3 by a supplementary component, see also [61]. The resulting new clustering techniques are able to adapt single clusters to the expansion of the corresponding group of data in an iterative optimisation procedure. A new approach based on volume centres as cluster representatives with varying radii for individual groups is described in section 6.2. The corresponding objective functions are presented and alternating optimisation schemes are derived. Experimental results demonstrate the significance of the presented techniques.

In the first section of this chapter a general approach to adapt to clusters with different expansions or (hyper-)volumes of the corresponding data groups is presented. This approach was presented in [57] to reduce the loss of information in rule learning. Therefore, only small modifications of some algorithms discussed in section 3 have to be made. The principle of objective function based fuzzy clustering with cluster representatives in form of real-valued vector prototypes remains unchanged. The second presented method does no longer use multidimensional centre-points as representatives but centre-volumes. As we will see, this approach has a non-negligible drawback. A combination of the presented methods seems to eliminate this lack and is presented in section 6.3. Another approach using volume prototypes to enable the fuzzy c-means to detect clusters with different densities was presented by M. Setnes and U. Kaymak [104]. Some remarks regarding GK have been made in [80].

Our approach seems to be well suited to adapt to different sizes of clusters. One remaining problem concerning the original versions of the algorithms presented in section 3 is that all these approaches presuppose uniformly distributed data over all clusters, i.e. the number of data points per cluster are assumed to be equal for all clusters. To cluster data with varying sizes and differences regarding the number of data points per structure correctly, adaptation to the density has to be taken into account. Some heuristics attempts to handle this problem have been developed in [106].

The higher flexibility of the algorithms requires a better initialisation of the cluster centres. Often the standard versions of the size-adaptable algorithms, see 3, are a good choice for the initialisation. Otherwise the presented approaches tend towards local optima.

6.1 Centre-Based Clustering

In this section one approach that is able to adapt to different expansions of the corresponding clusters using cluster centres as cluster representatives is introduced. This approach is called *size-adaptable centre-based clustering* (SACB). For each cluster an additional parameter τ_i is introduced to the objective function in order to enable the clustering algorithm to adapt the cluster volumes. τ_i can be considered as the (relative) radius of the corresponding cluster. The resulting probabilistic objective function is shown in (6.1), with constant real-valued parameter $l \in \mathbb{R}_{>0}$.

$$J^{prob}(X, U, \mathsf{v}) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} \cdot \frac{1}{\tau_i^l} \cdot d^2(\mathsf{v}_i, x_k) \tag{6.1}$$

To avoid the trivial solution that all $\tau_i \to \infty$, the constraint

$$\sum_{i=1}^{c} \tau_i = \tau \tag{6.2}$$

has to be taken into account, where $\tau \in \mathbb{R}_{>0}$ is a predefined constant parameter, e.g. $\tau = c$ or $\tau = 1$.

This approach is related to the basic clustering technique described in section 3.1.4. Here an additional parameter for each cluster is introduced whereas in section 3.1.4 the influence of a single datum on the partition is measured with parameter ω_k. In both cases an additional constraint has to be taken into account. Here the sum over all clusters and in section 3.1.4 the sum over all data for the additional parameter is restricted to be constant during the clustering.

Since the objective function (6.1) does not require special properties of the distance measure $d^2(\mathsf{v}_i, x_k)$, most of the described distance measures need only small modifications to use the advantages of the proposed objective function. Let us define

$$d_\tau^2(x_k, \mathsf{v}_i) = \mathcal{D}_{SACB,*} = \frac{1}{\tau_i^l} \cdot d^2(\mathsf{v}_i, x_k) \tag{6.3}$$

as a new group of distance measures. Then the objective function (6.1) can be rewritten as

$$J^{prob}(X, U, \mathsf{v}) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} \cdot d_\tau^2(\mathsf{v}_i, x_k).$$

Considering constraints (3.2) and (3.3) from section 3.1 we obtain the same equations for the membership degrees as in (3.1), except that we have to replace the old distance $d^2(\mathsf{v}_i, x_k)$ by $d^2_\tau(\mathsf{v}_i, x_k)$, i.e. in the probabilistic case the membership degrees can be derived analogously to proof 3.1 and evaluate to

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d^2_\tau(\mathsf{v}_i,x_k)}{d^2_\tau(\mathsf{v}_j,x_k)}\right)^{\frac{1}{m-1}}}.$$

For the possibilistic and noise clustering approach the membership update equations are derived again as in proof 3.2 and 3.3, respectively. Only the distance measure has to be replaced. The modified distance measure for FCM is shown in equation (6.4) and figure 6.1 for different l-values.

$$\mathcal{D}_{SACB,FCM} = \frac{1}{\tau_i^l}\cdot(x_k - v_i)^T(x_k - v_i) \tag{6.4}$$

For small values for l – $l = 0.25$ in figure 6.1(a) – the overall distance increases more rapid with increasing Euclidean distance, whereas the overall distance increases slowly for larger l-values, see figure 6.1(b), in comparison to $l = 1$ in figure 6.1(c). The same effect occurs with the transformed Euclidean distances used in section 3.2.2 for GK.

Let us define

$$\tilde{u}_{ik} = \frac{u_{ik}}{\tau^{\frac{l}{m}}}.$$

Equivalent to the objective function from section 3.2 minimising (6.1) leads to the necessary condition (3.18)

$$v_i = \frac{\sum_{k=1}^{n}\tilde{u}_{ik}^m\cdot x_k}{\sum_{k=1}^{n}\tilde{u}_{ik}^m} = \frac{\sum_{k=1}^{n}u_{ik}^m\cdot x_k}{\sum_{k=1}^{n}u_{ik}^m}$$

for the evaluation of the prototype coordinates in FCM, GK, and AGK and

$$C_i = \sum_{k=1}^{n}u_{ik}^m\cdot(x_k - v_i)(x_k - v_i)^\top$$

for the covariance matrices (GK)

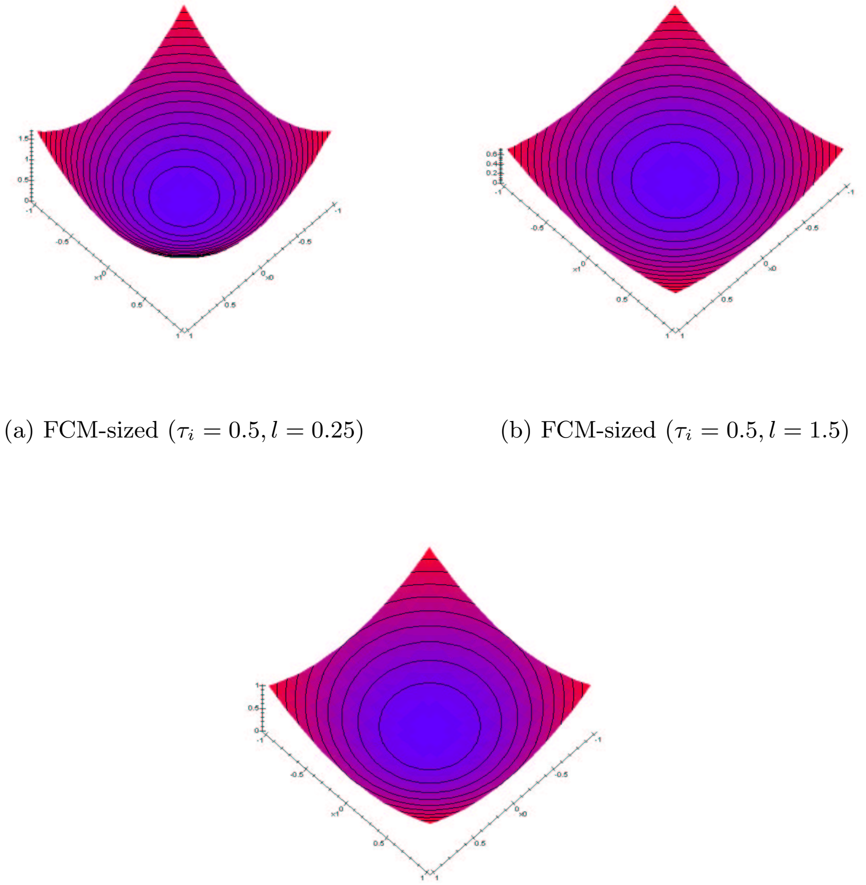
$$c_i^{(\nu)} = \sum_{k=1}^{n}u_{ik}^m\cdot(x_k^{(\nu)} - v_i^{(\nu)})^2,$$

respectively for AGK.

Theorem 6.1 (Centre-Based Size Parameter)

Considering constraint 6.2 leads to equation 6.5 as necessary condition for the objective function to have a minimum, see proof 6.1.

$$\tau_i = \frac{\left(\sum_{k=1}^{n}u_{ik}^m\cdot d^2(\mathsf{v}_i,x_k)\right)^{\frac{1}{l+1}}}{\sum_{j=1}^{c}\left(\sum_{k=1}^{n}u_{jk}^m\cdot d^2(\mathsf{v}_j,x_k)\right)^{\frac{1}{l+1}}}\cdot\tau. \tag{6.5}$$



(a) FCM-sized ($\tau_i = 0.5, l = 0.25$)

(b) FCM-sized ($\tau_i = 0.5, l = 1.5$)

(c) FCM-sized ($\tau_i = 0.5, l = 1$)

Figure 6.1: Distance for size-adaptable FCM to $v^\top = (0, 0)$

Proof 6.1 (Centre-Based Size Parameter)

Assuming that the parameters $l > 0$ and $\tau > 0$ are fixed, we have to take constraint (6.2) into account, to determine the values τ_i with predefined parameters $l > 0$ and $\tau > 0$ that are fixed during the iteration procedure. So we obtain the Lagrange function

$$J_\lambda^{prob}(X,U,\mathsf{v}) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m \cdot \frac{1}{\tau_i^l} \cdot d^2(\mathsf{v}_i, x_k) + \lambda\left(\sum_{i=1}^{c}\tau_i - \tau\right). \qquad (1)$$

Note that the last term of (1) *does neither depend on u_{ik} nor on v_i so that the formulae for the optimal choices of the u_{ik} and the v_i remain valid. Since the distance measure is independent of τ_i, differentiating* (1) *yields*

$$\frac{\partial J_\lambda^{prob}(X,U,\mathsf{v})}{\partial \tau_i} = -\frac{l}{\tau_i^{l+1}} \cdot \sum_{k=1}^{n} u_{ik}^m \cdot d^2(\mathsf{v}_i, x_k) + \lambda \overset{!}{=} 0$$

and therefore

$$\tau_i = \left(\frac{l \cdot \sum_{k=1}^{n} u_{ik}^m \cdot d^2(\mathsf{v}_i, x_k)}{\lambda}\right)^{\frac{1}{l+1}}. \qquad (2)$$

With (6.2) λ evaluates to

$$\lambda = \frac{\left(\sum_{j=1}^{c}\left(l \cdot \sum_{k=1}^{n} u_{jk}^m \cdot d^2(\mathsf{v}_j, x_k)\right)^{\frac{1}{l+1}}\right)^{l+1}}{\tau^{l+1}}. \qquad (3)$$

After inserting (3) *in* (2)*, we obtain the resulting calculation instruction for the τ_i*

$$\tau_i = \frac{\left(\sum_{k=1}^{n} u_{ik}^m \cdot d^2(\mathsf{v}_i, x_k)\right)^{\frac{1}{l+1}}}{\sum_{j=1}^{c}\left(\sum_{k=1}^{n} u_{jk}^m \cdot d^2(\mathsf{v}_j, x_k)\right)^{\frac{1}{l+1}}} \cdot \tau \quad \diamond$$

Equation 6.5 is used in an alternating iteration procedure, see algorithm 6.1.

The parameter $l > 0$ plays a similar role as the fuzzifier m. When we choose a small value for l, a strong emphasis is put on adapting to the cluster size. Too small values for l can have a negative effect on algorithms as the GK, since the priority is put on the cluster expansion instead of the cluster shape. For $l \to \infty$, no adaptation of cluster volume is carried out any more, and we obtain the original algorithms.

For an illustration of the influence of parameter l and the results of this approach, see in addition to this section also section 6.4.

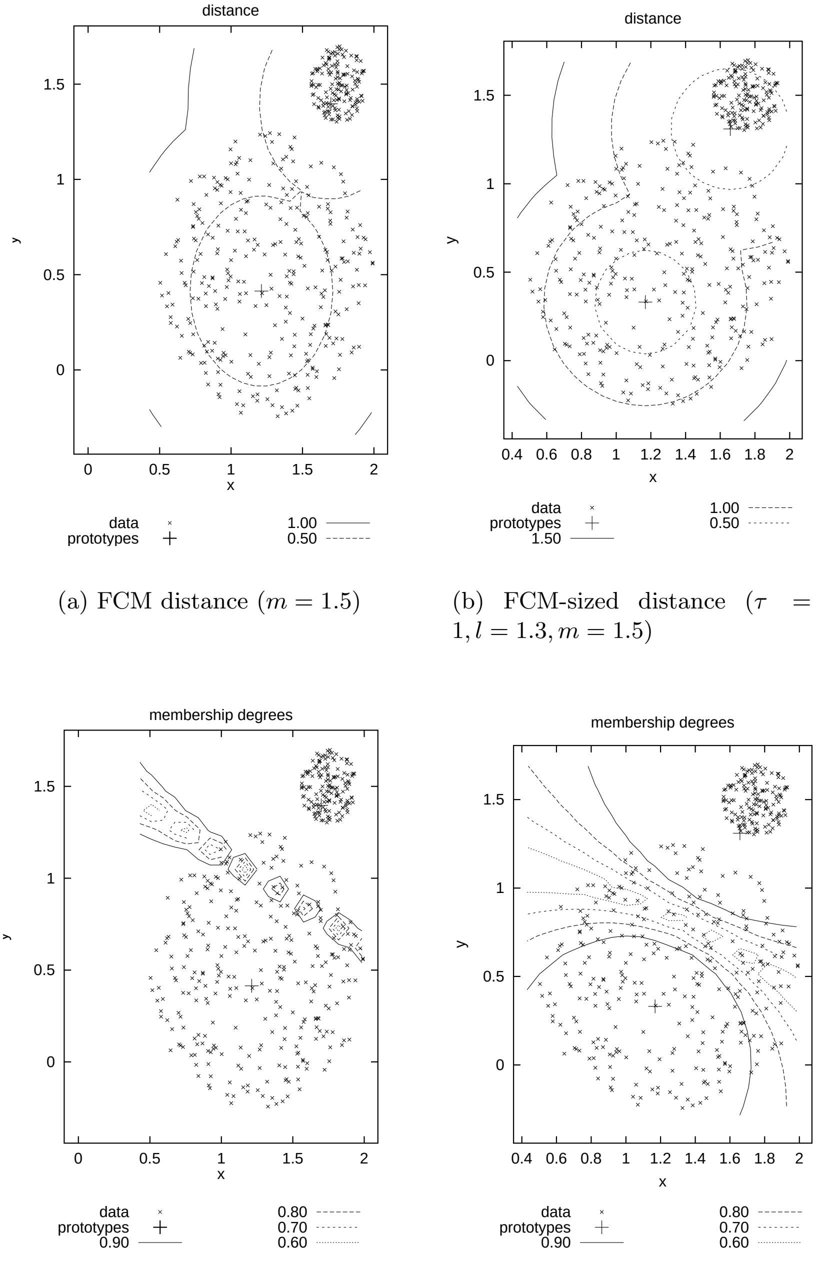
Equation (6.5) can be used alternatingly with one of the basic clustering algorithms from section 3.1 and a suitable distance measure for fuzzy clustering algorithms, see section 3.2. We call this group of clustering techniques *Size-Adaptable Centre-Based clustering algorithms* (SACB). Applying our results to the described FCM or GK algorithms enables these algorithms to detect clusters of different expansions.

In case of FCM, rule generation only results in a small loss of information. Adapting the sizes of the detected spherical structures has no influence on the precision of the resulting fuzzy rules. Also the axes-parallel version of GK, i.e. AGK, does not lead to a significant loss of information in rule-learning. Not only considering the task of rule learning this approach can in combination with GK as well as AGK be an objective function based alternative to GG or AGG, respectively.

In comparison to GG and AGG the presented approach has shown to be more robust and reliable regarding the clustering results. For not well-separated data GG or AGG tend towards a few large clusters covering the whole domain and the rest of the clusters have a negligible size. This behaviour depends on the choice of the fuzzifier m, but it is impossible to give a value for m that prevents the building of extremely sized clusters. In our approach the emphasis that is put upon size adaptation can be controlled with parameter l as described above. However, even for relative small values of l, where the emphasis on size adaptation is high, usually no deformed clusters occur. In addition the clustering result depends not so severely on small changes of l as GG can depend on changes of m.

It is possible to combine this approach with the objective function approaches of possibilistic (section 3.1.2) or noise clustering (section 3.1.3). The difference of these methods compared to the probabilistic objective function of section 3.1.1 does not depend on a special distance measure. In case of possibilistic clustering, equations (3.8) for the membership degrees u_{ik}, (6.5) for the size parameters τ_i, and the necessary conditions derived from the chosen distance measure ($\mathcal{D}_{FCM}$, $\mathcal{D}_{GK}$ or $\mathcal{D}_{AGK}$), e.g. the equations for cluster centres or covariance matrices can be used in an alternating optimisation procedure. Applying noise clustering to the size-adapting clustering approach, equation (3.11) has to be used to calculate the corresponding noise membership degrees. The other parameters are equivalent to probabilistic and possibilistic clustering, see algorithm 6.1. It has to be considered that the choice of τ is related to an appropriate choice of δ in noise clustering. Both determine in some way the 'overall size' of the clusters. Whereas δ is the same for all clusters in noise clustering, here τ is only an upper bound for the 'sum of sizes' $\sum_{i=1}^{c} \tau_i$. Nevertheless, the distances $d_\tau^2(\mathsf{v}_i, x_k)$ with initialised parameters τ_i have to be used to estimate δ in case of noise clustering.

In the alternating optimisation, $\mathcal{A}_*$ takes the place for one of the algorithms FCM, GK, or AGK, respectively and $\mathcal{A}$ itself stands for one of the basic objective function algorithms from section 3.1. The procedure $Calculate(\mathsf{v}_i)$ has to be extended as denoted in the algorithm scheme. Additionally the procedure $Choose()$ of the chosen basic algorithm has to be extended by $l \in \mathbb{R}_{>0}$ and $\tau \in \mathbb{R}_{>1}$, as shown in the algorithm scheme.



(a) FCM distance ($m = 1.5$)

(b) FCM-sized distance ($\tau = 1, l = 1.3, m = 1.5$)

(c) FCM membership degrees ($m = 1.5$)

(d) FCM-sized membership degrees ($\tau = 1, l = 1.3, m = 1.5$)

Figure 6.2: Comparison of probabilistic FCM and FCM-sized clustering

Algorithm 6.1 (Prototype Calculation for SACB)

```
Choose()
{
    $\mathcal{A}$ :: Choose();
    $\tau \in \mathbb{R}_{>0}$;
    $l \in \mathbb{R}_{>0}$;
}

Calculate($\mathsf{v}_i$)
{
    $\tau_i \;=\; \dfrac{\left(\sum_{k=1}^{n} u_{ik}^{m} \cdot d^2\left(\mathsf{v}i, x_k\right)\right)^{\frac{1}{l+1}}}{\sum_{j=1}^{c}\left(\sum_{k=1}^{n} u_{jk}^{m} \cdot d^2\left(\mathsf{v}_j, x_k\right)\right)^{\frac{1}{l+1}}} \cdot \tau;$
    $\mathcal{A}_*$ :: Calculate($\mathsf{v}_i$);
}
```

We call the corresponding alternating optimisation incorporating cluster size adaptation the sized algorithm (FCM-sized, GK-sized etc.).

In figure 6.2 clustering results for probabilistic FCM and FCM-sized are shown. Fo FCM-sized, parameter τ was set to 1 and $l = 1.3$ was chosen. The fuzzifier m has been set to 1.5 in both cases. The membership degrees as well as the distance of the data points to the cluster centres are illustrated. It can be seen that the FCM-sized is able to adapt the cluster size of the smaller cluster in the upper right corner. In figure 6.2(b), the circles denoting a distance of 0.5 to the cluster centres are more apart than in figure 6.2(a). The membership degrees have only a small range with degrees less than 1 in case of FCM (fig. 6.2(c)) whereas this range is increased in case of FCM-sized (fig. 6.2(d)). To adapt the cluster size even more, a smaller value for l has to be chosen.

Figures A.4 to A.13 in section A.2 illustrate the differences between the original Gustafson-Kessel algorithm and the size-adaptable centre-based clustering algorithm using the same transformed Euclidean distance as GK. The GK-sized partitions are shown for two different values of the influence parameter l, $l = 0.5$ and $l = 5$ and the basic objective functions introduced in section 3.1.

6.2 Volume-Based Clustering

In the previously described fuzzy clustering techniques the clusters are characterised by a vector, consisting of real-valued attributes, and a distance mea-

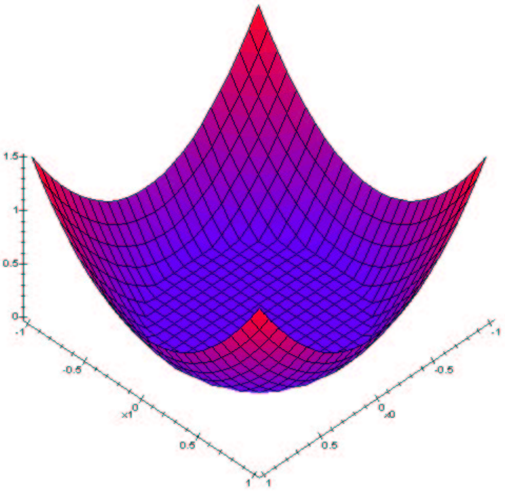


Figure 6.3: Volume-Based Distance for $\tau_i = 0.5$ to $v^\top = (0, 0)$

sure. Only the data points that coincide with a prototype may be assigned to the corresponding cluster with a membership degree of 1.0. Let us imagine dense spherical clusters. Instead of having just one ideal prototype for each cluster to which we calculate the distances of the data points, we now assume that we have a complete circle or (hyper-)ball as the cluster centre. This means that data within this area have distance zero to the cluster. This idea was proposed in [104]. However, there it was not based on an objective function, but on pure heuristic considerations. Here we want to derive an alternating optimisation scheme for this approach as well, called *volume-based clustering* (SAVB).

Taking these considerations into account, we obtain a probabilistic objective function (6.6) reflecting the idea of volume prototypes using (6.7) as the distance function. The distance $\mathcal{D}_{SAVB}$ is illustrated in figure 6.3 for $\tau_i = 0.5$.

$$J^{prob}(X, U, \mathsf{v}) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^m \cdot \ \max\{0,\ (x_k - v_i)^\top (x_k - v_i) - \tau_i\} \tag{6.6}$$

$$d^2\left(\mathsf{v}_i,\ x_k\right) \ = \ \mathcal{D}_{SAVB} \ = \ \max\{0,\ (x_k - v_i)^\top (x_k - v_i) - \tau_i\} \tag{6.7}$$

If the clusters' radii τ_i are known in advance, these values should be used directly. Otherwise the τ_i have to be adapted during the alternating optimisation taking constraint (6.8) into account, to avoid the trivial solution $\tau_i \rightarrow \infty$ for all $i \in \{1, \ \cdots, c\}$ in minimising the objective function (6.6).

$$\sum_{i=1}^{c} \tau_i^2 = \tau \tag{6.8}$$

Here τ is a predefined constant parameter. Assigning 0 to τ (all τ_i are 0) leads to the previously described fuzzy c-means (FCM) clustering technique, see section 3.2.1.

Theorem 6.2 (Volume-Based Cluster Parameters)

Differentiating the objective function 6.6 leads to necessary conditions for the cluster centres v_i, the centre radii τ_i and – depending on the basic algorithm chosen from section 3.1 – the membership degrees. The resulting update equations that can be used in an alternating optimisation approach are 6.9 and 6.10, see proof 6.2.

$$v_i = \frac{\sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m \cdot x_k}{\sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m} \tag{6.9}$$

$$\tau_i = \frac{\sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m}{\sqrt{\sum_{i=1}^c \left(\sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m\right)^2}} \cdot \sqrt{\tau} \tag{6.10}$$

The membership update equations have to be chosen from section 3.1, depending on the choice of probabilistic, possibilistic, noise, or outlier clustering. There the distance measure has to be replaced by $\mathcal{D}_{SAVB}$.

Proof 6.2 (Volume-Based Cluster Parameters)

To derive equations for prototype coordinates and radii values respectively, the partial derivatives of the objective function (6.6) considering constraint 6.8 have to be computed.

$$J_\lambda^{prob}(X,U,\mathsf{v}) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m \cdot \mathcal{D}_{SAVB} + \lambda \cdot \left(\sum_{i=1}^{c} \tau_i^2 - \tau\right).$$

$$\frac{\partial J_\lambda^{prob}(X,U,\mathsf{v})}{\partial v_i} = 2 \cdot \sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m \cdot (v_i - x_k) \overset{!}{=} 0$$

leading to

$$v_i = \frac{\sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m \cdot x_k}{\sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m}$$

$$\frac{\partial J_\lambda^{prob}(X,U,\mathsf{v})}{\partial \tau_i} = -\sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m + 2 \cdot \lambda \cdot \tau_i \overset{!}{=} 0$$

leads to the update equation for parameters τ_i

$$\tau_i = \frac{\sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m}{\sqrt{\sum_{i=1}^c \left(\sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m\right)^2}} \cdot \sqrt{\tau} \quad \diamond$$

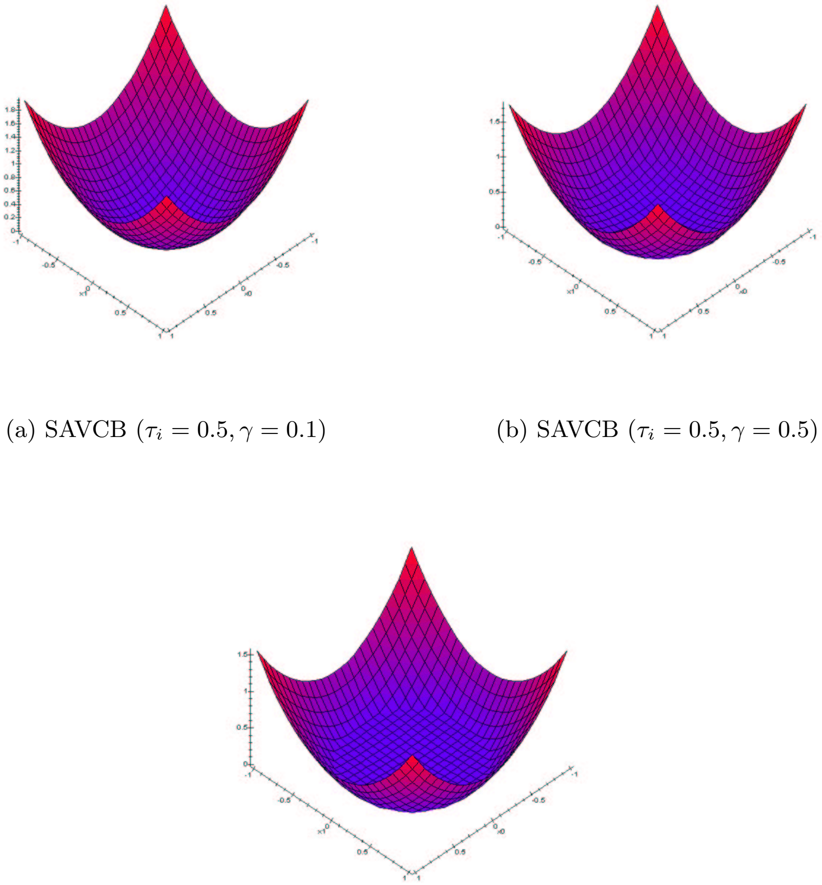
In the basic alternating optimisation scheme the distance measure has to be replaced by (6.7). Depending on the chosen clustering technique (probabilistic, possibilistic, noise, or outlier) the corresponding update equations of the membership degrees u_{ik} (3.4, 3.8, 3.11, or again 3.4) have to be used.

Note that the objective function is not differentiable in some points. The necessary conditions (6.9) and (6.10) lead to a local minimum, if no data points leave a volume centre or wander into a volume centre. This is why in equations (6.9) and (6.10) only data points with Euclidean distance greater τ_i to the cluster centres v_i have influence on the next alternating parameters τ_i^{new} and v_i^{new} for cluster i. Imagine two well separated spherical clusters are given. In the first alternating optimisation steps the structures are identified correctly. The τ_i are assigned the correct radius values of the circles containing the data points. In the next step each prototype and each radius is only calculated on the basis of the data points assigned to the opposite cluster. So the cluster parameters are alternatingly interchanged. Even if the τ_i are smaller than the correct radius values, convergence is neither guaranteed nor plausible. Nevertheless, the algorithm scheme is denoted in algorithm 6.2. $\mathcal{A}$ denotes the chosen basic objective function, probabilistic, possibilistic, noise, or outlier clustering, respectively.

Algorithm 6.2 (Prototype Calculation for SAVB)

```
Choose()
{
  $\mathcal{A}$ :: Choose();
  $\tau \in \mathbb{R}_{>0}$;
}

Calculate($\mathsf{v}_i$)
{
  $\tau_i = \frac{\sum_{k:(x_k-v_i)^\top (x_k-v_i)>\tau_i} u_{ik}^m}{\sqrt{\sum_{i=1}^{c} \left(\sum_{k:(x_k-v_i)^\top (x_k-v_i)>\tau_i} u_{ik}^m\right)^2}} \cdot \sqrt{\tau};$
  $v_i = \frac{\sum_{k:(x_k-v_i)^\top (x_k-v_i)>\tau_i} u_{ik}^m \cdot x_k}{\sum_{k:(x_k-v_i)^\top (x_k-v_i)>\tau_i} u_{ik}^m};$
}
```



(a) SAVCB ($\tau_i = 0.5, \gamma = 0.1$)

(b) SAVCB ($\tau_i = 0.5, \gamma = 0.5$)

(c) SAVCB ($\tau_i = 0.5, \gamma = 0.9$)

Figure 6.4: Distance for the SAVCB to $v^\top = (0, 0)$

6.3 Volume-Centre-Based Clustering

To avoid drawbacks as in case of the volume-based clustering technique (SAVB) from section 6.2 objective function (6.6) has been modified so as to combine distance measure $\mathcal{D}_{SAVB}$, equation (6.7), with the Euclidean distance used for the fuzzy c-means algorithm. The resulting objective function for the *volume-centre-based clustering* (*SAVCB*) is shown in (6.11). Parameter $0 < \gamma < 1$ determines the influence of each summand in the distance function (6.12). The distance function is illustrated in figure 6.4 for $\tau_i = 0.5$ and varying values for γ.

$$J^{prob}(X, U, \mathsf{v}) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m} \cdot \Big(\gamma \cdot \ \max\{0,\ (x_k - v_i)^{\top}(x_k - v_i) - \tau_i\} + (1-\gamma) \cdot (x_k - v_i)^{\top}(x_k - v_i)\Big) \tag{6.11}$$

$$d^2(\mathsf{v}_i,\ x_k) \ = \ \mathcal{D}_{SAVCB} \ = \gamma \cdot \ \max\{0,\ (x_k - v_i)^{\top}(x_k - v_i) - \tau_i\} + (1-\gamma) \cdot \ (x_k - v_i)^{\top}(x_k - v_i) \tag{6.12}$$

Theorem 6.3 (Volume-Centre-Based Cluster Parameters)

To adapt the cluster radii during the alternating optimisation, again constraint (6.8) has to be considered, leading to equation (6.13) as update equation for the cluster centres v_i and 6.14 for the 'radii' τ_i, see proof 6.3.

$$v_i \ = \ \frac{\sum_{k=1}^{n} u_{ik}^{m} \cdot \ x_k \ - \ \gamma \cdot \ \sum_{k:(x_k - v_i)^{\top} \cdot (x_k - v_i) \leq \tau_i} u_{ik}^{m} \cdot \ x_k}{\sum_{k=1}^{n} u_{ik}^{m} \ - \ \gamma \cdot \ \sum_{k:(x_k - v_i)^{\top} \cdot (x_k - v_i) \leq \tau_i} u_{ik}^{m}} \tag{6.13}$$

$$\tau_i = \frac{\sum_{k:(x_k - v_i)^{\top}(x_k - v_i) > \tau_i} u_{ik}^{m}}{\sqrt{\sum_{i=1}^{c}\left(\sum_{k:(x_k - v_i)^{\top}(x_k - v_i) > \tau_i} u_{ik}^{m}\right)^2}} \cdot \ \sqrt{\tau} \cdot \ \gamma \tag{6.14}$$

The greater the influence of the Euclidean distance ($\gamma \rightarrow 0$), the smaller are the calculated center radii τ_i.

Proof 6.3 (Volume-Centre-Based Cluster Parameters)

Considering constraint 6.8 leads to the Lagrange function

$$J_{\lambda}^{prob}(X, U, \mathsf{v}) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^{m} \cdot \Big(\gamma \cdot \ \max\{0,\ (x_k - v_i)^{\top}(x_k - v_i) - \tau_i\} + (1-\gamma) \cdot \ (x_k - v_i)^{\top}(x_k - v_i)\Big) + \lambda \cdot \left(\sum_{i=1}^{c} \tau_i^2 \ - \ \tau\right).$$

The partial derivative for v_i is as follows

$$\frac{\partial J_{\lambda}^{prob}(X, U, \mathsf{v})}{\partial v_i} = 2 \cdot \ (1-\gamma) \cdot \sum_{k=1}^{n} u_{ik}^{m} \cdot \ (v_i - x_k) + \ 2 \cdot \ \gamma \cdot \sum_{k:(x_k - v_i)^{\top}(x_k - v_i) > \tau_i} u_{ik}^{m} \cdot \ (v_i - x_k) \overset{!}{=} 0,$$

so that we obtain

$$v_i \;=\; \frac{\sum_{k=1}^{n} u_{ik}^m \cdot\; x_k \;-\; \gamma \cdot\; \sum_{k:(x_k-v_i)^\top\cdot(x_k-v_i)\le\tau_i} u_{ik}^m \cdot\; x_k}{\sum_{k=1}^{n} u_{ik}^m \;-\; \gamma \cdot\; \sum_{k:(x_k-v_i)^\top\cdot(x_k-v_i)\le\tau_i} u_{ik}^m}.$$

To obtain the update equation for parameter τ_i we have to determine the partial derivative with respect to τ_i

$$\frac{\partial J_\lambda^{prob}(X,U,\mathsf{v})}{\partial\tau_i} = -\,\gamma\cdot \sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m + 2\cdot\;\lambda\cdot\;\tau_i \overset{!}{=} 0,$$

so that we finally obtain

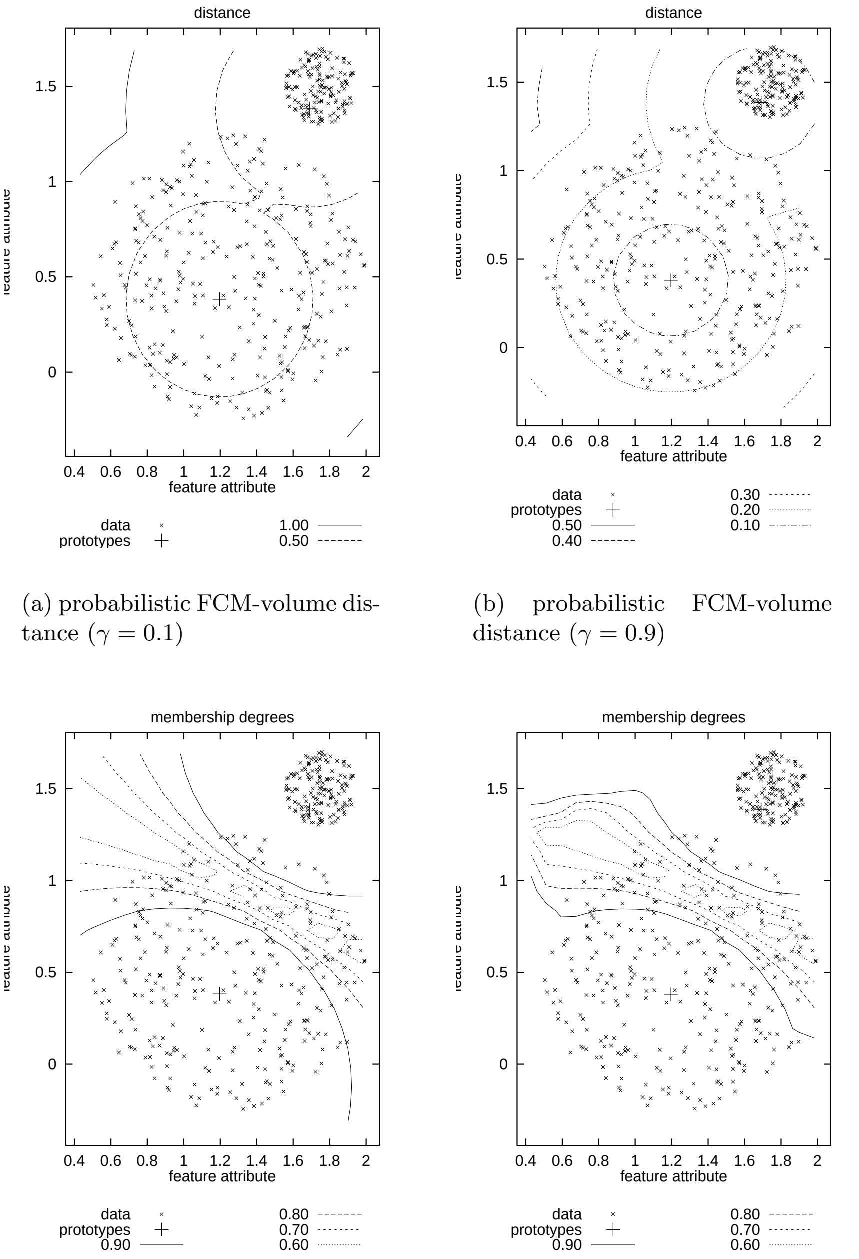
$$\tau_i \;=\; \frac{\sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m}{\sqrt{\sum_{i=1}^{c}\left(\sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m\right)^2}} \cdot\; \sqrt{\tau}\cdot\;\gamma \quad \diamond$$

Depending on the chosen basic clustering technique (probabilistic, possibilistic, noise, or outlier) the adequate calculation instruction for the membership degrees u_{ik} has to be chosen. There the distance measure has to be replaced by $\mathcal{D}_{SAVCB}$ (6.12). Even if the influence of the Euclidean distance is rather small ($\gamma \approx 0.99$), the alternating optimisation converges reliably in the experiments.

To the corresponding alternating optimisation scheme is referred as the FCM-volume algorithm. The procedure *Calculate*(v_i) is denoted in algorithm 6.3 together with the extension of procedure *Choose*(). Again, $\mathcal{A}$ symbolises one of the basic algorithms, probabilistic, possibilistic, noise, or outlier clustering.

Figure 6.5 illustrates the clustering results for the FCM-volume clustering algorithm in combination with the probabilistic basic objective function from section 3.1. Figures A.14 to A.18 in the appendix illustrate the clustering results for the FCM-volume clustering algorithm in combination with the possibilistic, noise, and outlier objective functions from section 3.1. The same data set used here was also used for the FCM-sized example in figure 6.2.

The influence of parameter γ on the clusters centre radii is visible in the illustration of the distance to the cluster centres (fig. 6.5(a) and 6.5(b)). The larger γ the smaller are the parameters τ_i.



(a) probabilistic FCM-volume distance ($\gamma = 0.1$)

(b) probabilistic FCM-volume distance ($\gamma = 0.9$)

(c) probabilistic FCM-volume membership degrees ($\gamma = 0.1$)

(d) probabilistic FCM-volume membership degrees ($\gamma = 0.9$)

Figure 6.5: Comparison of probabilistic FCM-volume clustering for different values of γ, $m = 1.5$, and $\tau = 1$

Algorithm 6.3 (Prototype Calculation for SAVCB)

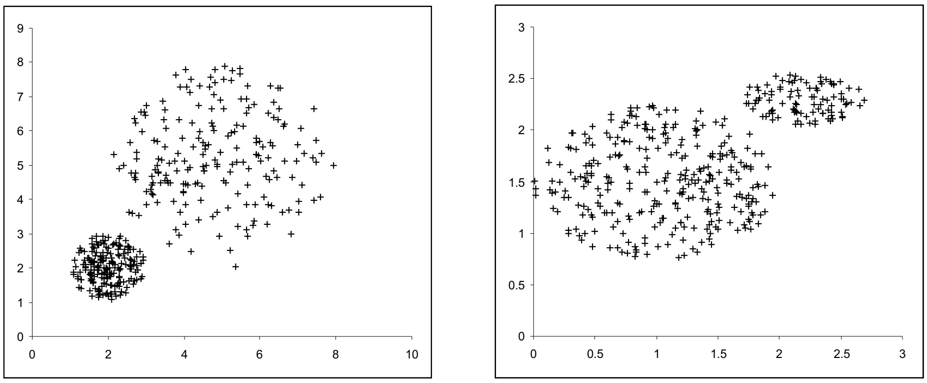
```
Choose()
{
    $\mathcal{A}$ :: Choose();
    $\tau \in \mathbb{R}_{>0}$;
}

Calculate($\mathsf{v}_i$)
{
    $\tau_i \;=\; \frac{\sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m}{\sqrt{\sum_{i=1}^{c}\left(\sum_{k:(x_k-v_i)^\top(x_k-v_i)>\tau_i} u_{ik}^m\right)^2}} \cdot \sqrt{\tau} \cdot \gamma;$
    $v_i \;=\; \frac{\sum_{k=1}^{n} u_{ik}^m \cdot x_k \;-\; \gamma \cdot \sum_{k:(x_k-v_i)^\top\cdot(x_k-v_i)\le\tau_i} u_{ik}^m \cdot x_k}{\sum_{k=1}^{n} u_{ik}^m \;-\; \gamma \cdot \sum_{k:(x_k-v_i)^\top\cdot(x_k-v_i)\le\tau_i} u_{ik}^m};$
}
```

6.4 Illustrative Examples

To demonstrate the properties of the approaches of this chapter two additional artificial test data sets have been designed. They are shown in figure 6.6. Part (a) of figure 6.6 shows two spherical clusters with uniformly distributed data points for both clusters but different radii. Here the number of data points for each cluster is the same. In part (b) of figure 6.6 two elliptical clusters with uniformly distributed data points but different extents are displayed. The larger cluster has about twice as many data points as the smaller one.

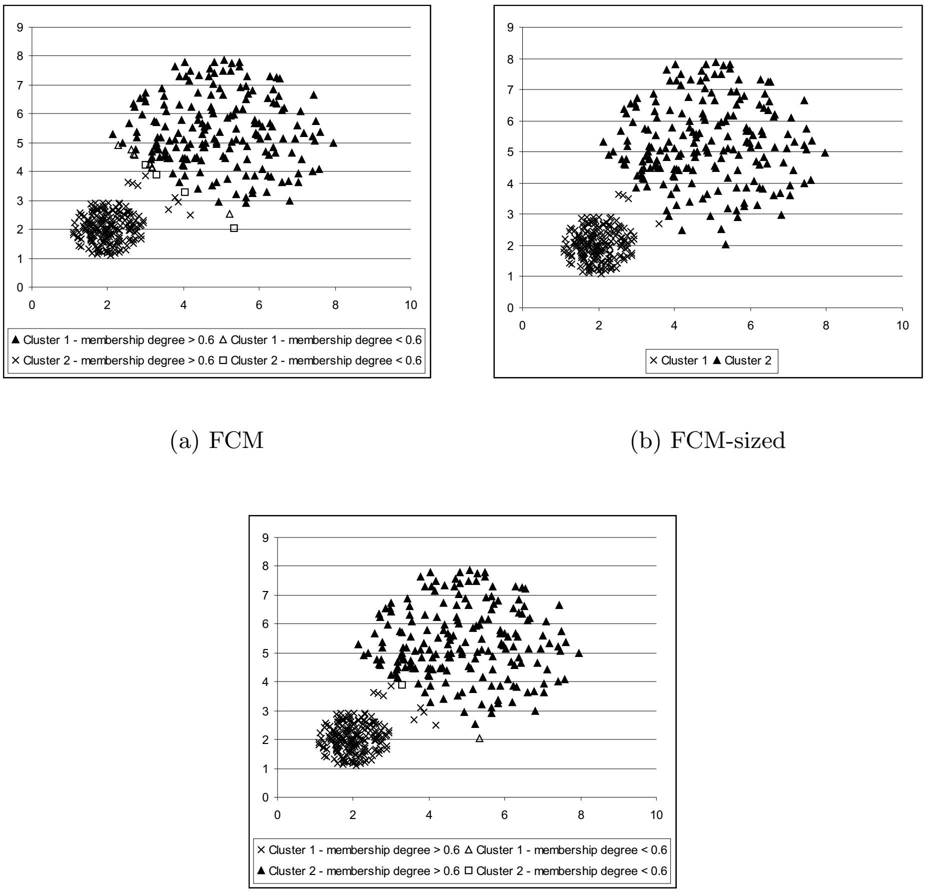
In figure 6.7 the results for the data set from figure 6.6(a) with the algorithms using the Euclidean distance measure are compared. The fuzzifier m was in all cases set to 2.0. The constraint parameter τ was set to 1.0 in both cases, FCM-sized and FCM-volume. For the size adaptable version of the fuzzy c-means algorithm the exponent l was set to 0.5. For the influence of the radii part in case of the FCM-volume approach, $\gamma = 0.99$ was chosen. The original fuzzy c-means algorithm has difficulties in assigning the data to the correct clusters (see figure 6.7 (a)). A datum is assigned to the cluster with the highest membership degree. The approach using the Euclidean distance combined with volume centres is in a position to adapt the volume centres and therefore yields slightly better results than the original FCM (see part (c) of figure 6.7). Only the size adaptable approach (part (b)) has the ability to assign most data points correctly.



(a) Two spherical clusters

(b) Two elliptical clusters

Figure 6.6: Artificial test data sets

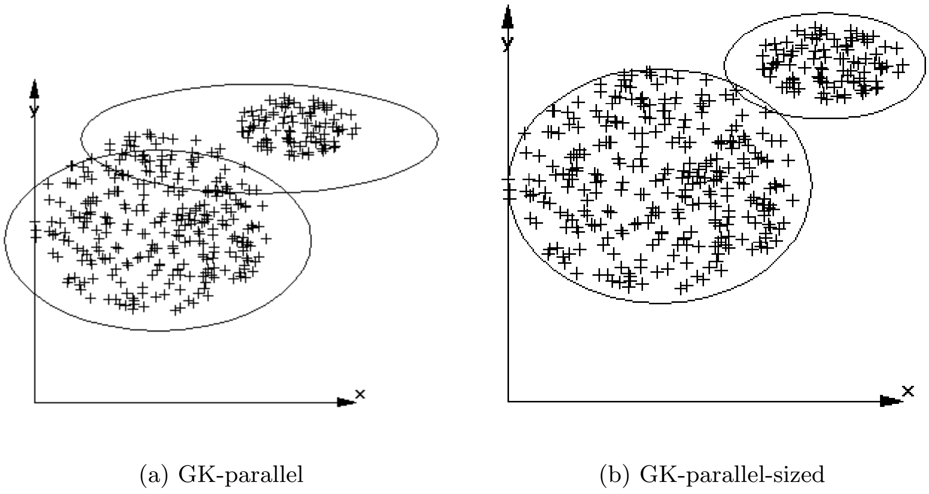


(a) FCM

(b) FCM-sized

(c) FCM-volume

Figure 6.7: Classification results for the circular test data set



(a) GK-parallel

(b) GK-parallel-sized

Figure 6.8: Classification results for the elliptical test data set

In figure 6.8 the results for the ellipsoidal test data set from figure 6.6(b) are shown. As clustering algorithms the axes-parallel versions of the Gustafson-Kessel algorithm and the new size-adaptable version of that algorithm have been chosen. The fuzzifier m was assigned the value 2.0 in both cases. For the size-adaptable approach constraint parameter τ has been set to 1 and the exponent l was assigned 0.4. This new approach is able to adapt to the ellipses' content (figure 6.8 part (b)) whereas the result in part (a) of figure 6.8 shows that the Gustafson-Kessel algorithm searches for groups of about the same size. Our approach can be further improved if a smaller value for parameter l, e.g. $l = 0.3$, is chosen.

As another example, we used the Wisconsin Breast Cancer Database [112, 26] to test our new approaches with the probabilistic objective function. This classified data set originally contains 699 data points with 9 attributes and a classification attribute. 16 data points with missing values have been deleted from the data set for our tests [109]. From the remaining 683 data points 444 were classified as benign and 239 as malignant. In Figure 6.9 the results for the original fuzzy c-means algorithm (FCM) are compared to our size adaptable version of this algorithm (FCM-sized) and the combination of the FCM with the volume-center-based approach (FCM-volume). In figure 6.9 the percentage of wrong classified data for two to ten clusters is displayed. The fuzzifier m was in all cases set to 2.0. The values for the other parameters are $\tau = 1.0$ for FCM-sized as well as for FCM-volume, $l = 0.8$ for FCM-sized and $\gamma = 0.9$ for FCM-volume. In this case the priority of FCM-volume lies upon the volume-based component. The best results are obtained with our new algorithms. The FCM-sized algorithm yields the best classification where 2.6% of the data entries are misclassified with four clusters. A similar good result (2.8% misclassified data points) is reached in case of FCM-volume at 5 clusters. The best result for the FCM (3.1% wrong classified data) is obtained with 4 clusters. Our approaches seem both to improve the results for the Wisconsin

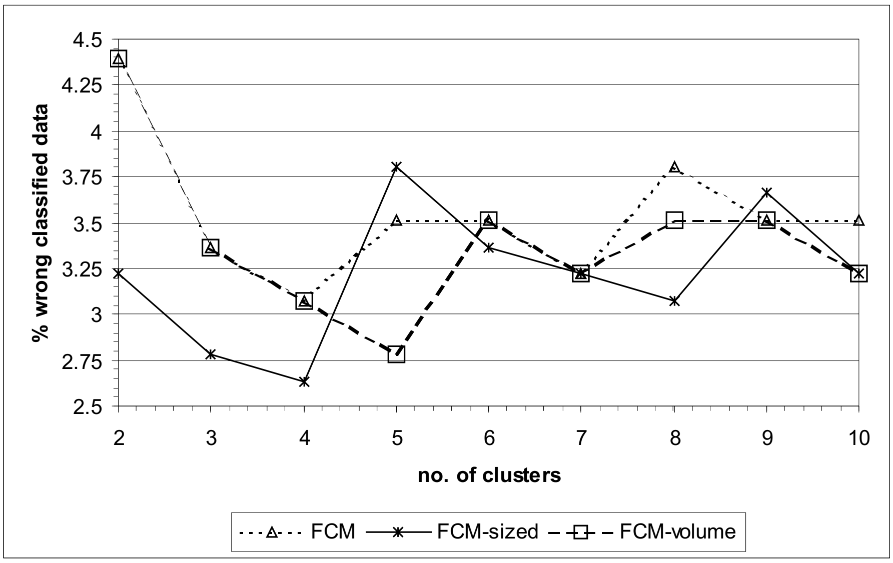


Figure 6.9: Classification results for Wisconsin Breast Cancer

breast cancer database. For further analysis, cross validation would have to be carried out for the Wisconsin breast cancer data set. Therefore, a small part of the data set (e.g. 10% has to be selected randomly as test data for validation and neglected from the sample data. The classification has to be calculated for the sample data and afterwards the membership degrees and the classification for the test data has to be evaluated on the basis of the sample data clustering result. Carrying out this analysis for several times with different test data sets leads to reliable results whether a clustering technique is suited for a special classification task. Such tests are important if the classification of additional data should be determined using classification results for data with known classification attributes. For the illustration purposes of this section, we restrict to the classification of the whole sample data set.

In table 6.1 clustering results in case of the size-adaptable fuzzy c-means for different values of parameter l are shown. The values denote the percentage of misclassified data. To calculate this value first for all clusters c the class which is represented by one particular cluster is determined. Then the data points corresponding to that cluster but originally belonging to a different class than the cluster's are counted. The sum of misclassified data over all clusters in ratio to the total number of data gives the percentage of misclassified data, also called error rate. It can be seen, that the result depends on the choice of the exponent l. For figure 6.9 the value for $l = 0.8$ obtaining best results has been chosen.

For table 6.2 we have chosen those c-partitions for each l-value from table 6.1, where the least error values occurred, and calculated the maximal membership degree for each datum separately. In table 6.2 the average of these maximal membership degrees is shown for each partition. It is obvious that this value in probabilistic clustering depends on the total number of clusters

Table 6.1: Percentage of misclassified data for varying number of clusters and different l-values with FCM-sized

c	**l = 0.2**	**l = 0.5**	**l = 0.8**	**l = 1.0**	**l = 3.0**	**l = 5.0**
2	25.5	*2.8*	3.2	3.1	4.0	4.2
3	30.9	*2.8*	2.8	2.9	3.2	3.2
4	36.6	2.9	*2.6*	*2.6*	*2.9*	*3.1*
5	47.4	4.4	3.8	3.8	3.7	3.5
6	34.8	2.9	3.4	3.2	3.5	3.5
7	13.6	2.9	3.2	3.4	3.2	3.2
8	17.4	*2.8*	3.1	3.2	3.8	3.8
9	22.3	4.2	3.7	3.5	3.5	3.5
10	34.6	3.4	3.2	3.2	3.2	3.4
11	*12.2*	2.9	3.8	3.8	3.8	3.8
12	13.3	3.1	3.5	3.5	3.5	3.5

Table 6.2: Maximal membership degrees for the best results found in table 6.1 for specific l-values and FCM-sized

l	u_{ik}^{max}
0.2	0.5964
0.5	0.9078
0.8	0.7045
1.0	0.7105
3.0	0.7112
5.0	0.7119

for the partition, e.g. the value for $l = 0.2$ and $c = 11$ clusters is less than the result for $l = 0.5$ and $c = 2$ clusters. The last four entries illustrate the influence of parameter l on the membership degrees, see section 6.1. Here, for the l-values $0.8, 1.0, 3.0$ and 5.0 the number of clusters was in all cases $c = 4$. The calculated values are slightly increasing for increasing l-values.

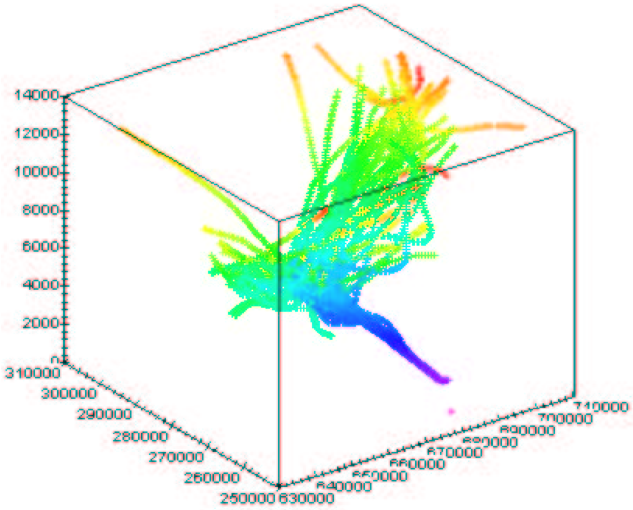


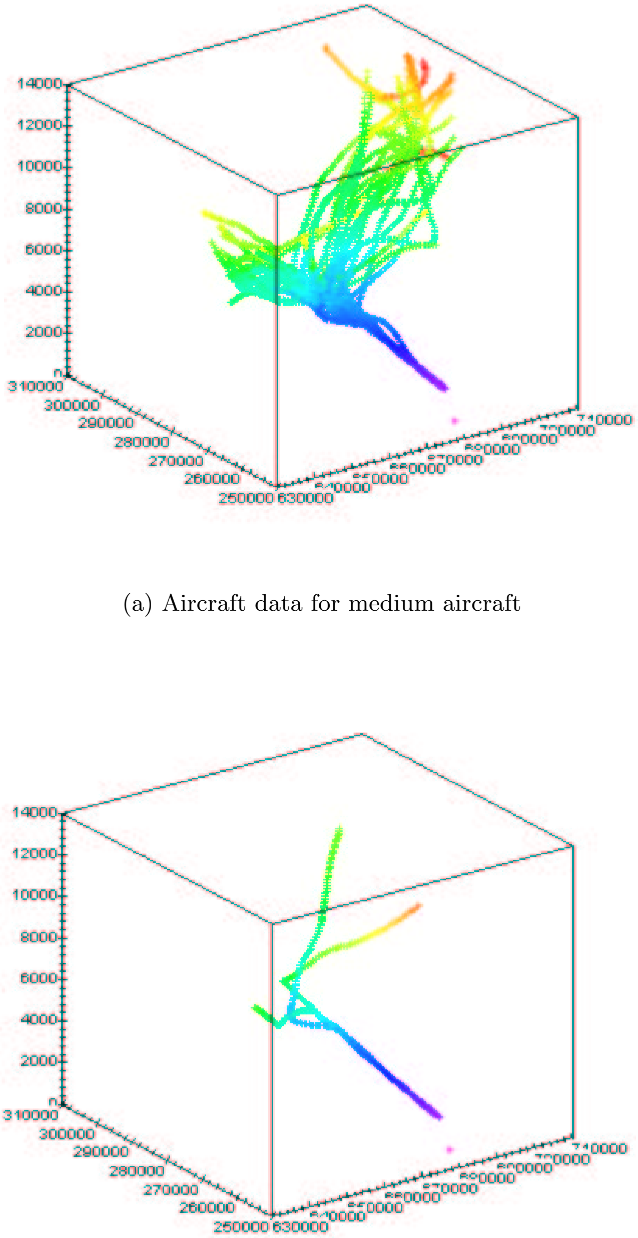
Figure 6.10: Air traffic data for two high traffic hours at Zurich airport

6.5 Flight Route Detection

In this section we demonstrate how the adaptation of cluster volumes in fuzzy clustering can be applied to the flight route detection problem described in section 2.2. The idea of this example is to develop a technique that enables us to describe practiced routes in a way comparable to pre-defined routes. Therefore, the widespread radar plots of a number of aircrafts have to be replaced by a description in form of route segments for "average" routes.

An example of a two-dimensional radar plot is illustrated in figure 2.2 in section 2.2. In this figure, the arrivals and departures at Zurich airport in 1996 are shown. An invisible grid was laid over the airport surrounding area and the number of aircraft passing one field of this grid were counted. This way the colours separate the differently frequented areas around Zurich airport.

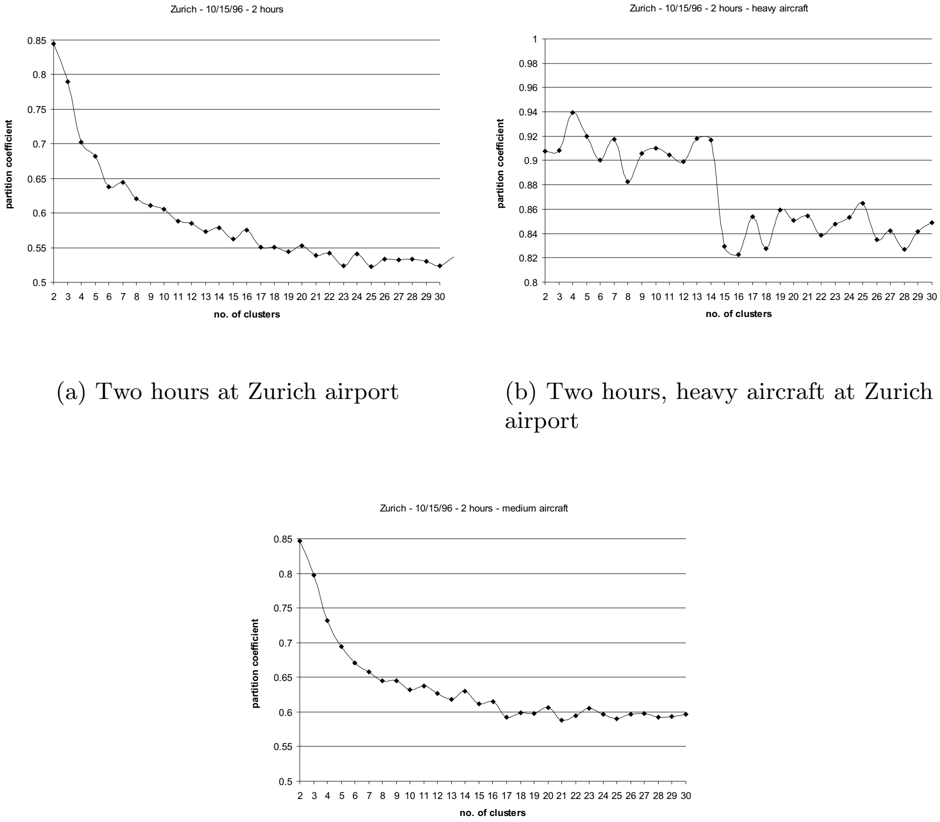
Here we show how GK-sized can be used to identify and describe flight paths in the airport surrounding airspace. Since the data set is too large to analyse and handle as a whole, we have to restrict ourselves to a subpart of the data. The data set for all arriving and departing aircraft at Zurich airport in three month has a size of about $175MB$ in ASCII form. Even if we restrict ourselves to arriving aircraft the ASCII file with the data for all three month has a size of about $110MB$. Data for one week and arriving aircraft leads to a ASCII file-size of about $9MB$. Selecting only two high traffic hours, e.g. 4 to 6 p.m., leads to a size of about $1.6MB$ or 77.000 data vectors per week. The number of data vectors is not equivalent to the number of aircraft, each aircraft's position is recorded every four seconds. Not only the data set is too large to analyse as a whole, but also differing results for certain subparts are of great interest, e.g. if the time of day or aircraft type influences the flight routes. To reduce the computational effort or split the data in meaningful



(a) Aircraft data for medium aircraft

(b) Aircraft data for heavy aircraft

Figure 6.11: Parts of air traffic data for two high traffic hours at Zurich airport



(a) Two hours at Zurich airport

(b) Two hours, heavy aircraft at Zurich airport

(c) Two hours, medium aircraft at Zurich airport

Figure 6.12: Validity (partition coefficient) for air traffic data

subparts we

- choose the analysis time or part, e.g. one day, certain aircraft type, time of day,
- extract coordinate data, and
- generate a data set with aircraft coordinates.

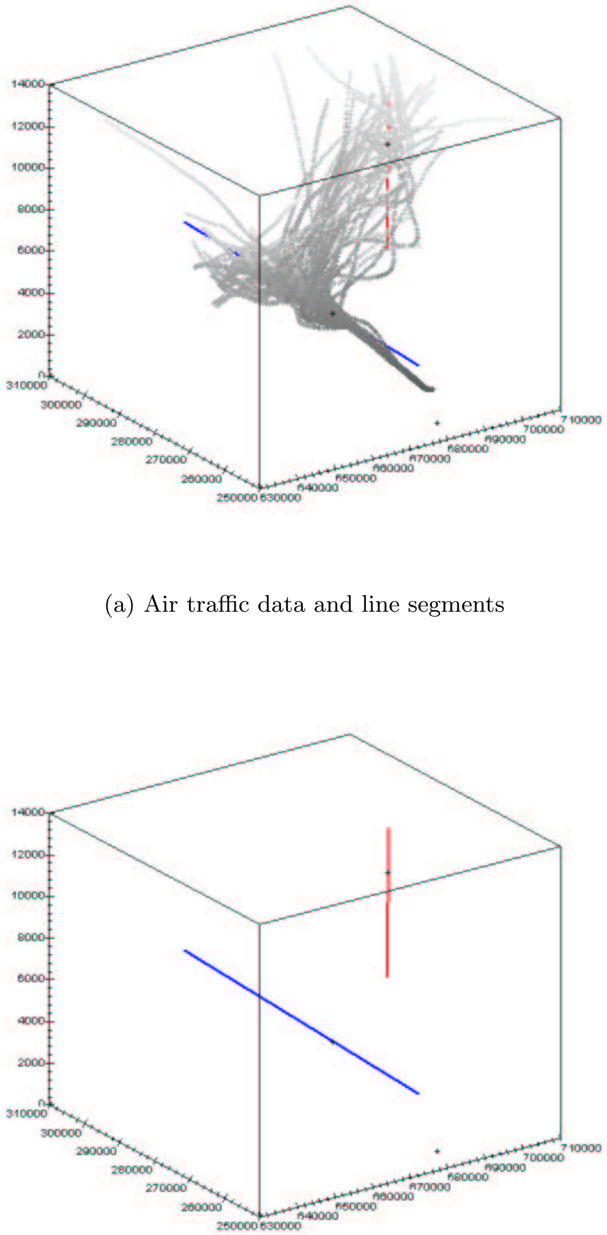
In figure 6.10 the radar data for arriving aircraft at Zurich airport is shown for two high-traffic hours, i.e. $210KB$ ASCII data or about 10.500 data vectors. This data is further split in one part containing the radar points of heavy-sized aircraft ($10KB$ ASCII data or about 470 data vectors), see 6.11(b), and another part consisting of the radar points of medium-sized aircraft ($120KB$ ASCII data or about 5.900 data vectors), see 6.11(a).

For the selected data part we have to determine a suitable number of clusters. Therefore global validity measures as described in chapter 4 can be chosen. Figures 6.12(a), 6.12(b), and 6.12(c) illustrate the validity for our three test data parts. Here the measure partition coefficient – see section 4.1.2 – was chosen, for the coordinate data from Zurich airport clustered with GK-sized. The parameters have been set as follows: fuzzifier $m = 2$, termination bound $\epsilon = 0.001$, constraint parameter $\tau = 1$, and parameter $l = 0.5$ to determine the emphasis that is put upon size adaptation.

To illustrate the results we analysed the partition for each of our three sample data sets for a small number of clusters, i.e. 2 clusters for the complete and the medium data part and 4 for the heavy data part, and a larger local optimum, i.e. 16 for the complete data set, 13 for the heavy aircraft part and 14 for the part with medium aircraft. One covariance matrix for each cluster is used in GK-sized to estimate the cluster form. Each matrix describes an ellipsis whose main axis represents one line segment. In this way the partition of the flight data in ellipsoidal clusters can be used to estimate parts of common flight paths. We have to determine the smallest eigenvalue of the cluster's covariance matrix and the corresponding eigenvector. This eigenvector gives us the direction of the line corresponding to the analysed cluster in our p-dimensional analysis space. Equivalently we could use the greatest $p-1$ eigenvalues to describe lines in the p-dimensional space. In this case the corresponding eigenvectors are all orthogonal to the resulting line. To determine the line segments corresponding to one cluster, we have to

- determine the covariance matrix' eigenvalues,
- sort the eigenvalues e_{is}^{val} in increasing order, where i is the number of the corresponding cluster and s the number of the ordered eigenvalue,
- calculate the eigenvector e_{is}^{vec} corresponding to the smallest (the first in our order) eigenvalue
- determine the line's equation using the eigenvector for the line's direction and the cluster centre as a line point, i.e. the resulting line equation is of the form $g := v_i + \rho \cdot e_{is}^{vec}$ where v_i is the cluster centre, and
- determine the line's extent.

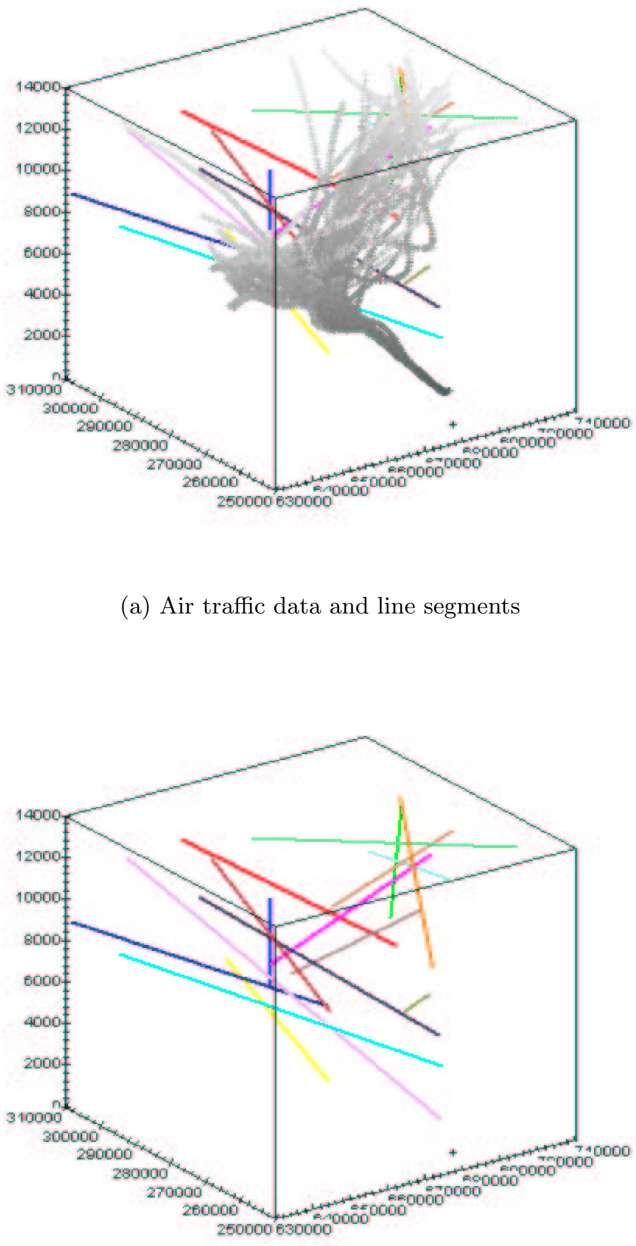
The line equation gives us the direction of a part of a flight path. To estimate the extent of a single part, the membership degree of the data points to a particular cluster are used. We determine the partition of the data – each datum is assigned to that cluster to which it has the highest membership degree. To avoid that outliers determine the length of a line segment, we exclude data with a small membership degree to its assigned cluster from further analysis, e.g. in probabilistic clustering we choose $u_{ik} < \frac{1+\beta}{c}$, where $\beta < 1$ is a small real number that determines "the percentage that the membership degree of a datum has to be greater than that of a datum equally shared among the clusters". For data assigned to the cluster under consideration, we calculate the minimal and maximal coordinates in each domain. The minimal and maximal coordinates whose corresponding – not identical – data vectors have the greatest membership degrees to the cluster under consideration are used to estimate the lines start- and endpoint. The eigenvector of the covariance matrix together with the cluster centre determine an equation that defines a line



(a) Air traffic data and line segments

(b) Line segments

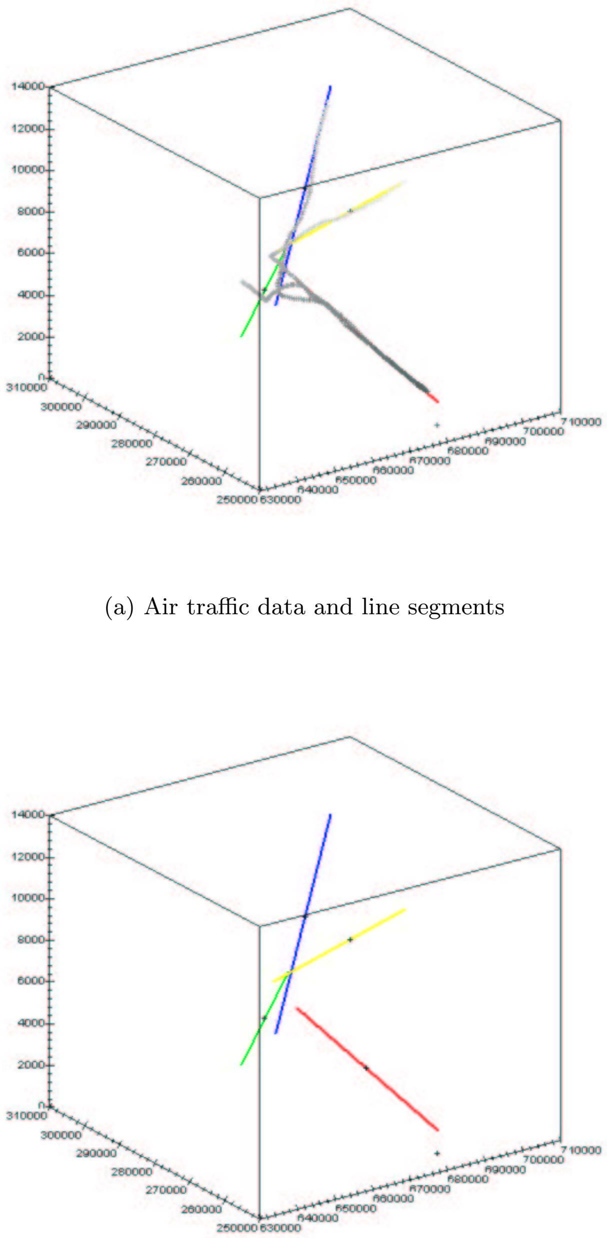
Figure 6.13: Air traffic data and line segments for 2 clusters and the complete data set



(a) Air traffic data and line segments

(b) Line segments

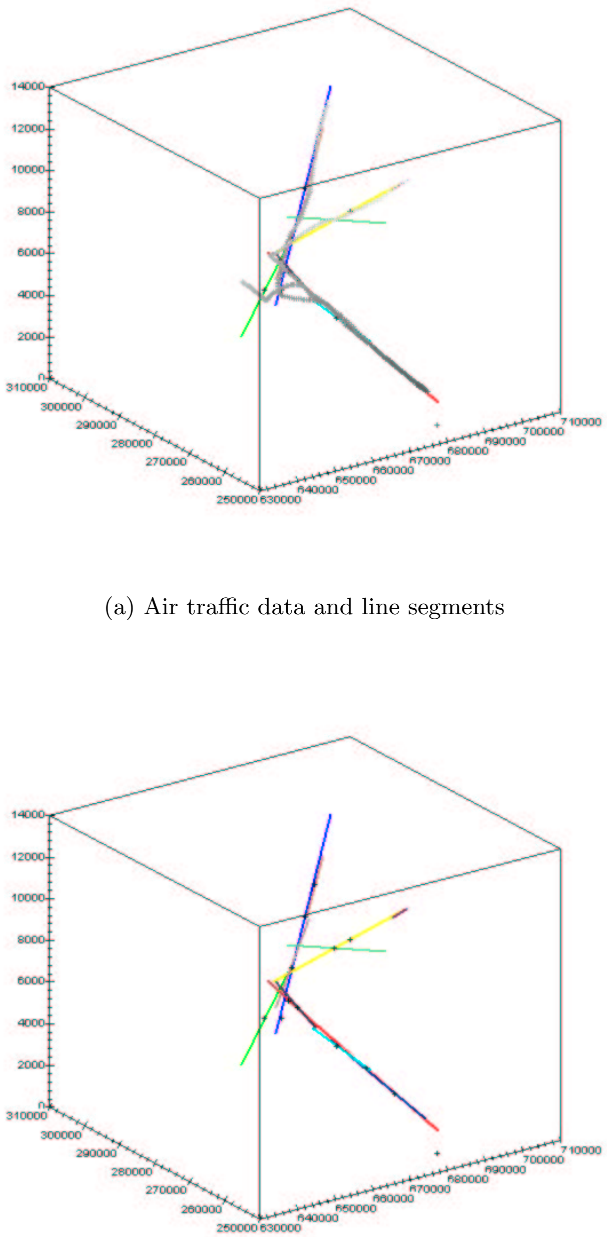
Figure 6.14: Air traffic data and line segments for 16 clusters and the complete data set



(a) Air traffic data and line segments

(b) Line segments

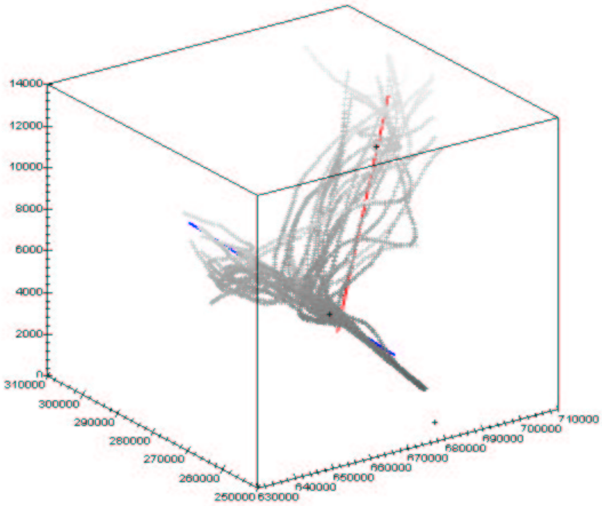
Figure 6.15: Air traffic data and line segments for 4 clusters and the heavy data part



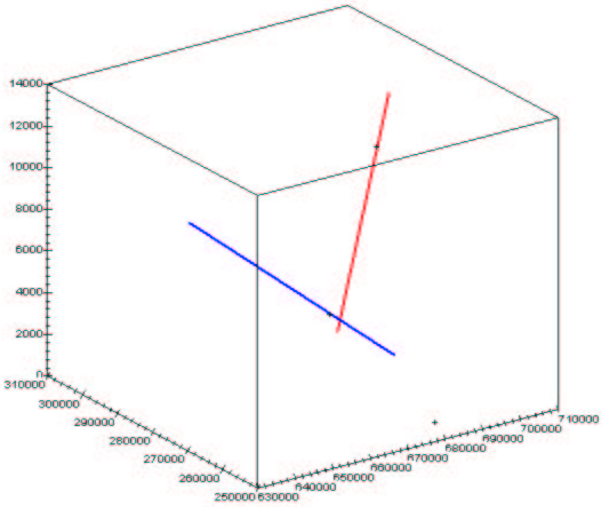
(a) Air traffic data and line segments

(b) Line segments

Figure 6.16: Air traffic data and line segments for 13 clusters and the heavy data part

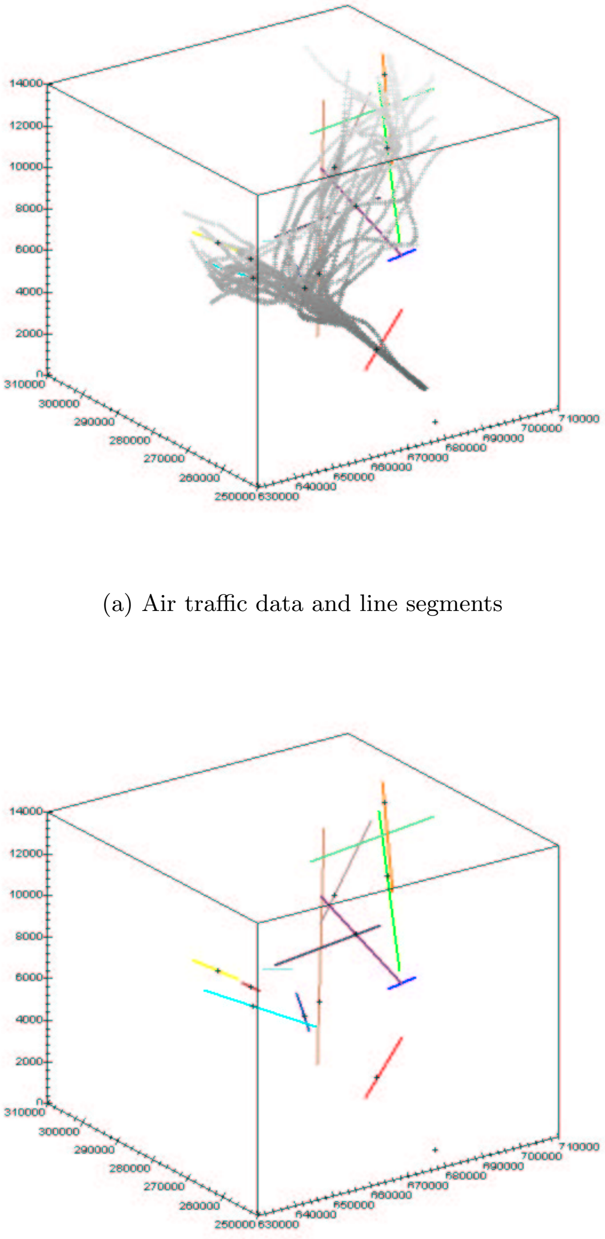


(a) Air traffic data and line segments



(b) Line segments

Figure 6.17: Air traffic data and line segments for 2 clusters and the medium data part



(a) Air traffic data and line segments

(b) Line segments

Figure 6.18: Air traffic data and line segments for 14 clusters and the medium data part

in the multidimensional space – in our case part of the flight route. With the estimated coordinates, we can solve this equation and obtain multidimensional points that are part of the straight line described by the covariance matrix. Density based clustering techniques are used in large databases to identify related geographical structures, see e.g. [5] and the references therein. However, the distance measure used for GK based on a cluster's covariance matrix enables us to identify line-segments corresponding to the (hyper-) ellipsoidal structures described by the clusters.

In this way, one line segment is described for each cluster, see figures 6.13 to 6.18. Part (a) of each figure shows the data with underlying line segments, whereas part (b) illustrates the estimated line segments. The smaller the data part the better are the results. A significant larger number of clusters could improve the results for the two larger data sets, but we have to take into account that we are interested in an "average" flight route for a certain situation. If the cluster number is to large, we would try to rebuild nearly each single route. Thereby, it is not guaranteed, that the direction of our line segments corresponds to the flight direction. If we use only the radar points for cluster analysis, not all clusters led to meaningful line segments, see e.g. the red line in figure 6.18(a) and 6.18(b). Although the cluster was correct from the mathematical and data analysis point of view ("group data with small distances together"), it does not lead to the correct flight direction. To assure that our line segments have the orientation of the flight routes, we can generate an additional artificial data attribute that contains this direction, e.g. the positive angle to the airport if the aircraft flies towards the airport or the negative angle for the opposite direction.

Let us briefly describe how the clustering results are further processed. First we have to carry out a kind of plausibility check and neglect clusters that do not fulfil this test. The part of data corresponding to the neglected clusters has eventually to be analysed separately. For other clusters background knowledge tells us to neglect the results, e.g. from a certain height, routes for arriving aircraft lead usually directly to the airport. This way, the red segment of figure 6.18 would be neglected. Additionally, similar segments can be combined to a single segment. To determine whether two line segments are similar enough to combine, the segments begin and end points as well as the segments direction have to be compared. For approach procedures usually the segments starting point will have a greater height value whereas for departing flights the end point will have a greater height value. One possibility to determine the similarity of two segments is to calculate the area determined by the segments and their combined starting (resp. end) points. If the enclosed area is small, it has to be checked whether we cope with sequencing (or overlapping) segments. A comparison of one segments endpoint and length with the second segments starting point determines if both segments can be brought together. To identify flight routes the single line segments have to be connected to a full flight path. If we restrict ourselves to arriving or departing aircraft, either the start or the endpoint has to be near to the airport's coordinate. Therefore overlapping line segments have to be identified and combined. Where gaps in the flight route occur, we have to estimate the full path, e.g.

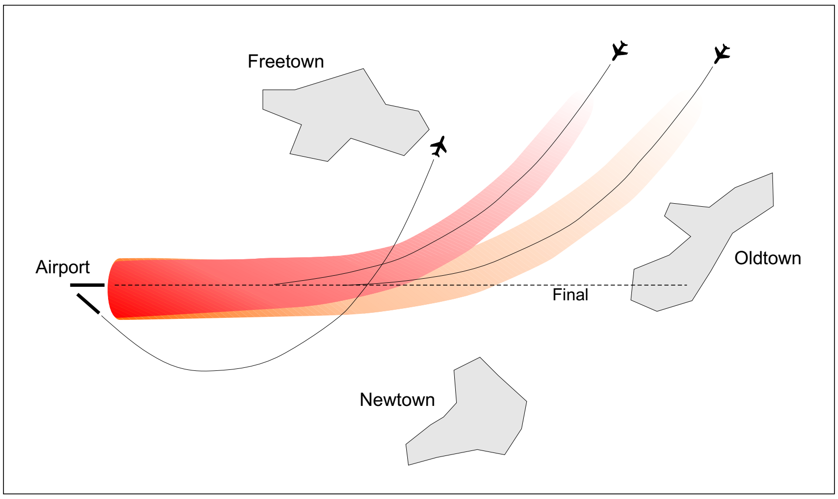


Figure 6.19: Noise reducing routes

enlarge the detected line segments or add additional segments. Thereby special demands of aircraft have to be taken into account. The crossing points have to meet suitable angles between lines in all directions.

This method can be applied to compare flight routes, e.g. under different flight, weather, or traffic conditions or with existing procedure paths. Finding routes that minimise air traffic delay and the pollution of the airport surrounding area, see e.g. figure 6.19, are an important task in data analysis for air traffic management. In figure 6.19 an artificial airport and surrounding cities are shown. The direct approach route *final* and two alternatives are shown with the corresponding noise pollution. In some cases the social demands have let to improved noise reducing flight routes that have not been described in official arrival or departure procedures. It is important to analyse the actual flight situation e.g. before further building land is declared or airport improvements are realised. The methods developed in this work are new approaches to meet these demands. One actual routing problem is the definition of a new airspace structure. Data analysis is a possibility to identify the real situation and in this way indicate workable solutions.

Chapter 7

Attribute Weighting Fuzzy Clustering

This chapter describes a new fuzzy clustering method developed to determine the influence of particular features on particular clusters and detect structures or groups in unevenly over the structure's single domains distributed data [60, 59]. In sample data single clusters are often defined by only a few attributes of the full attribute set. The aim of this weighing fuzzy clustering approach is to determine these attributes. Therefore influence parameters for each single data feature for each cluster are introduced. These parameters are added to the Euclidean distance used by FCM, see section 3.2.1. In this way attribute weighting fuzzy clustering generalises the FCM approach.

The presented clustering technique gives information about the influence of particular variables or attributes of the data set on special clusters. This knowledge can be used e.g. in classification tasks to determine or detect class defining attributes. Without ignoring one data attribute for the whole classification it is possible to reduce the influence of that attribute on only some clusters. In that way, attribute weights could help to partition the whole data set into smaller data parts depending on the same attributes. Analysing the smaller parts with a reduced number of attributes would reduce the computational effort. Real data sets soon get immense large. If we would e.g. consider all flight specific attributes for flight data analysis together, we would have to cope with about $255,000$ data vectors with 15 attributes for arrivals and departures, 5 additional weather attributes, and other derived variables. Attribute weighting could not only be helpful in reducing computation time but also to reduce the future expense of measuring.

7.1 Formal Definition

Especially in data where few variables determine particular clusters other variables may disguise the structure and should therefore not be considered to find these clusters. This can be done by weighting single attributes for each cluster as is expressed in the new distance measure (7.1). The resulting *attribute weighting fuzzy clustering* technique is denoted by *AWFCM*.

The distance between a datum x_k and a cluster (vector) v_i is defined by

$$d^2(\mathsf{v}_i, x_k) \;=\; \mathcal{D}_{AWFCM} \;=\; \sum_{s=1}^{p} \alpha_{is}^{t} \cdot \left(x_k^{(s)} \;-\; v_i^{(s)}\right)^2. \tag{7.1}$$

$x_k^{(s)}$ and $v_i^{(s)}$ indicate the s'th coordinates of the vectors x_k and v_i, respectively. The number of variables or attributes is denoted by p. α_{is} is a parameter determining the influence of attribute (coordinate) s for cluster i. $t \in \mathbb{R}_{>1}$ is a real-valued parameter that enables us to define the strongness of the emphasis that is put on the attribute weighting task. The distance function is illustrated in figure 7.1 for $t = 1.05$ (figure 7.1(a)) and $t = 5$ (figure 7.1(b)).

The parameters α_{is} can be considered as fixed or adapted individually for each cluster during clustering. If pre-knowledge about the attributes' significance is available, the weighting parameters can be specified by the user. Otherwise the α_{is} can be estimated during the clustering due to the constraint

$$\sum_{s=1}^{p} \alpha_{is} = 1 \qquad \forall i \in \{1, \cdots, c\}. \tag{7.2}$$

More generally, a constant $a \in \mathbb{R}$ can be introduced instead of 1 in the constraint, e.g. $a = 1$ or $a = c$. If we would neglect this constraint, we would obtain the trivial solution $\alpha_{is} = 0$ for all i and s.

The exponent $t \in \mathbb{R}_{>1}$ in equation (7.1) has a similar influence on the parameters α_{is} as the fuzzifier m on the membership degrees u_{ik}. For $t \longrightarrow 1$ the α_{is} tend to be 1 or 0 – either one attribute has unrestricted influence or no influence at all. On the other hand, if $t \longrightarrow \infty$, all attributes get the same influence on the cluster structure, i.e. $\alpha_{is} \longrightarrow \frac{1}{p}$ for all i and s.

Based on this approach we can derive an alternating optimisation scheme for fuzzy clustering using distance measure (7.1). The basic probabilistic objective function using $\mathcal{D}_{AWFCM}$ as distance measure is shown in equation 7.3.

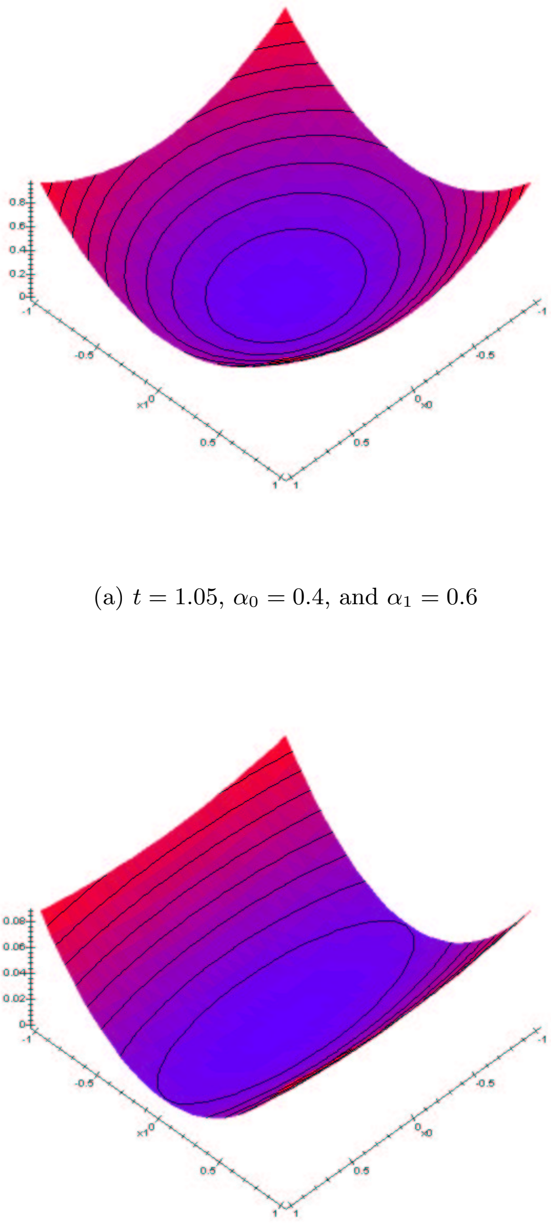
$$J^{prob}(X, U, \mathsf{v}) \;=\; \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^{m} \cdot \sum_{s=1}^{p} \alpha_{is}^{t} \left(x_k^{(s)} - v_i^{(s)}\right)^2 \tag{7.3}$$

To adapt the influence parameters α_{is} we have to determine a necessary condition for the values α_{is} so that the objective function achieves an optimum value.

Theorem 7.1 (Attribute Weights)

Differentiating (7.3) leads to equation (7.4) for the parameter α_{is} as a necessary condition for the objective function to have a minimum, see proof 7.1. The resulting equation can be used for updating α_{is} during the alternating clustering procedure.

$$\alpha_{is} = \frac{1}{\sum_{\gamma=1}^{p} \left(\frac{\sum_{k=1}^{n} u_{ik}^{m} \cdot \left(x_k^{(s)} - v_i^{(s)}\right)^2}{\sum_{k=1}^{n} u_{ik}^{m} \cdot \left(x_k^{(\gamma)} - v_i^{(\gamma)}\right)^2} \right)^{\frac{1}{t-1}}}. \tag{7.4}$$



(a) $t = 1.05$, $\alpha_0 = 0.4$, and $\alpha_1 = 0.6$

(b) $t = 5$, $\alpha_0 = 0.4$ and $\alpha_1 = 0.6$

Figure 7.1: Attribute weighting distance of $(x0, x1)^\top$ to $v^\top(0, 0)$

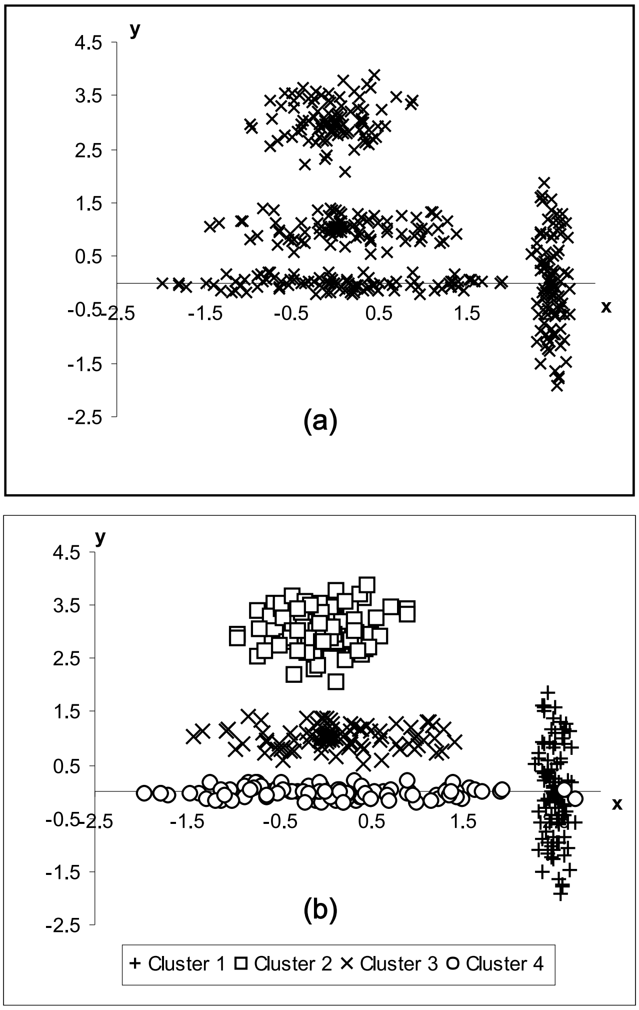


Figure 7.2: Ellipsoidal clusters

Proof 7.1 (Attribute Weights)

With condition (7.2) we obtain the Lagrange function

$$J_\lambda(X,U,\mathsf{v}) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m \cdot \sum_{s=1}^{p} \alpha_{is}^t \left(x_k^{(s)} - v_i^{(s)}\right)^2 - \sum_{i=1}^{c} \lambda_i \left(\sum_{s=1}^{p} \alpha_{is} - 1\right).$$

leading to the partial derivation

$$\frac{\partial J_\lambda\left(X,U,\mathsf{v}\right)}{\partial \alpha_{is}} \;=\; \sum_{k=1}^{n} u_{ik}^m \cdot t \cdot \alpha_{is}^{t-1} \left(x_k^{(s)} - v_i^{(s)}\right)^2 - \lambda_i \overset{!}{=} 0. \qquad (1)$$

So we obtain from (1)

$$\lambda_i \;=\; t \cdot \alpha_{is}^{t-1} \sum_{k=1}^{n} u_{ik}^m \cdot \left(x_k^{(s)} - v_i^{(s)}\right)^2$$

and therefore

$$\alpha_{is} \;=\; \left(\frac{\lambda_i}{t \cdot \sum_{k=1}^{n} u_{ik}^m \cdot \left(x_k^{(s)} - v_i^{(s)}\right)^2}\right)^{\frac{1}{t-1}}. \qquad (2)$$

With constraint 7.2 this leads to

$$\begin{aligned} 1 \;&=\; \sum_{s=1}^{p} \left(\frac{\lambda_i}{t \cdot \sum_{k=1}^{n} u_{ik}^m \cdot \left(x_k^{(s)} - v_i^{(s)}\right)^2}\right)^{\frac{1}{t-1}} \\ &=\; \left(\frac{\lambda_i}{t}\right)^{\frac{1}{t-1}} \cdot \sum_{s=1}^{p} \left(\frac{1}{\sum_{k=1}^{n} u_{ik}^m \cdot \left(x_k^{(s)} - v_i^{(s)}\right)^2}\right)^{\frac{1}{t-1}} \end{aligned}$$

and so λ_i evaluates to

$$\lambda_i \;=\; \frac{t}{\left(\sum_{s=1}^{p} \left(\frac{1}{\sum_{k=1}^{n} u_{ik}^m \cdot \left(x_k^{(s)} - v_i^{(s)}\right)^2}\right)^{\frac{1}{t-1}}\right)^{t-1}}.$$

Together with (2) we obtain the equation for parameter α_{is}

$$\alpha_{is} \;=\; \frac{1}{\sum_{\gamma=1}^{p} \left(\frac{\sum_{k=1}^{n} u_{ik}^m \cdot \left(x_k^{(s)} - v_i^{(s)}\right)^2}{\sum_{k=1}^{n} u_{ik}^m \cdot \left(x_k^{(\gamma)} - v_i^{(\gamma)}\right)^2}\right)^{\frac{1}{t-1}}}. \quad \diamond$$

Theorem 7.2 (Cluster Centres for Attribute Weighting Clustering)

In a similar way we obtain a necessary condition for the cluster centres, see proof 7.2.

$$v_i^{(s)} = \frac{\sum_{k=1}^{n} u_{ik}^m \cdot x_k^{(s)}}{\sum_{k=1}^{n} u_{ik}^m} \Rightarrow v_i = \frac{\sum_{k=1}^{n} u_{ik}^m \cdot x_k}{\sum_{k=1}^{n} u_{ik}^m},$$

as in FCM, see section 3.2.1.

Proof 7.2 (Cluster Centres for Attribute Weighting Clustering)

$$J(X, U, \mathsf{v}) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^m \cdot \sum_{s=1}^{p} \alpha_{is}^t \left(x_k^{(s)} - v_i^{(s)} \right)^2$$

$$\begin{aligned} \frac{\partial J(X, U, \mathsf{v})}{\partial v_i^{(s)}} &= -2 \cdot \sum_{k=1}^{n} u_{ik}^m \cdot \alpha_{is}^t \left(x_k^{(s)} - v_i^{(s)} \right) \overset{!}{=} 0 \\ \Rightarrow \sum_{k=1}^{n} u_{ik}^m x_k^{(s)} &= v_i^{(s)} \cdot \sum_{k=1}^{n} u_{ik}^m \\ \Rightarrow v_i^{(s)} &= \frac{\sum_{k=1}^{n} u_{ik}^m x_k^{(s)}}{\sum_{k=1}^{n} u_{ik}^m} \qquad \diamond \end{aligned}$$

The membership update equations are derived analogously to section 3.1, depending on the chosen basic clustering technique, probabilistic, possibilistic, or noise clustering, respectively. To calculate the membership degrees, distance measure $\mathcal{D}_{AWFCM}$ has to be used for $d^2(\mathsf{v}_i, x_k)$ in the corresponding basic clustering algorithm.

It should be noted that this approach is also related to the axes parallel version of the Gustafson-Kessel algorithm (AGK) described in section 3.2.2 and [73]. AGK uses a diagonal matrix for each cluster that determines the axes-parallel extensions of that cluster. The diagonal elements can be seen as weights for the attributes in the same way as we use them here, except for our exponent t. However, the constraint for AGK is that the determinant is constant, i.e. the sum in equation (7.2) is replaced by a product. In case of AGK the constant determinant for each cluster guarantees that the volume of a cluster is constant during the iterative clustering procedure. The sum for AWFCM allows a modification of the cluster volume during the alternating cluster estimation but prevents the trivial solution where all weights are 0. Another advantage of our approach is that the strongness of the influence of single variables can be controlled by the parameter t as described above.

The algorithm scheme 7.1 can be used as procedure $Calculate(\mathsf{v}_i)$ in the chosen basic clustering Algorithm $\mathcal{A}$, see section 3.1.

Algorithm 7.1 (Prototype Calculation for AWFCM)

```
Calculate($prototype_i$)
{
    $v_i = \frac{\sum_{k=1}^{n} u_{ik}^{m} \cdot x_k}{\sum_{k=1}^{n} u_{ik}^{m}}$;
    for all  $s \in \{1, \dots, p\}$
        $\alpha_{is} = \frac{1}{\sum_{\gamma=1}^{p} \left( \frac{\sum_{k=1}^{n} u_{ik}^{m} \cdot \left( x_k^{(s)} - v_i^{(s)} \right)^2}{\sum_{k=1}^{n} u_{ik}^{m} \cdot \left( x_k^{(\gamma)} - v_i^{(\gamma)} \right)^2} \right)^{\frac{1}{t-1}}}$;
}
```

7.2 Illustrative Examples

In Figure 7.2 an artificial test data set consisting of four ellipsoidal groups is shown. Part (a) presents the original data set and part (b) represents the clustering result obtained by the attribute weighting clustering technique (AWFCM) where a datum is assigned to the cluster to which it has the highest membership degree (maximum defuzzification). In this case we have set both, the fuzzifier m and the exponent t, to 2.0. However, the clustering result depends more on a suitable initialisation of cluster centres than the choice of parameters m and t. Table 7.1 lists the minimum and maximum attribute values for all clusters.

Table 7.1: Minimum/maximum feature values for each cluster

	attributes			
	x		y	
cluster no.	min	max	min	max
1	2.28	2.71	−1.93	1.85
2	−0.96	0.90	2.07	3.88
3	−1.42	1.40	0.54	1.40
4	−1.97	1.93	−0.21	0.21

In table 7.2 cluster 1 represents the ellipsoidal group with greatest x-values in the right part of figure 7.2. From top to bottom in the left part of figure 7.2 are the clusters 2, 3, and 4. The scale values α_{is} were adapted during the clustering procedure. It is obvious, that for each cluster the more the data coordinates are scattered around the corresponding prototype's coordinate,

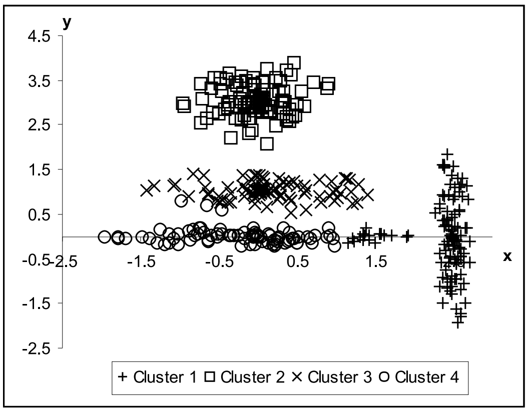


Figure 7.3: Results for ellipsoidal clusters with FCM

the less is the influence of the corresponding attribute for that cluster. In our example in figure 7.2 the two attribute influence parameters α_{is} for cluster 2 have nearly the same value. The data coordinates are approximately uniformly distributed for the two domains of this cluster. For clusters 3 and 4, the data values for attribute x are scattered widely whereas the values for attribute y have a small range – so the influence parameters α_{ix} are small in comparison to α_{iy} for clusters 3 and 4. In case of cluster 1 the data values for attribute y are scattered widely, resulting in a high value for influence parameter α_{1x}.

Table 7.2: Attribute weights for ellipsoidal data set

	attributes	
cluster no.	α_{ix}	α_{iy}
1	0.99	0.01
2	0.49	0.51
3	0.08	0.92
4	0.01	0.99

Figure 7.3 presents the clustering result for the example data set generated by FCM clustering technique with fuzzifier $m = 2$ as above. Using the Euclidean distance measure, FCM is not well suited to detect ellipsoidal structures in data. One indication for the suitability of a clustering result is the following value. Of the c membership degrees associated with each datum, we only consider the highest membership degree (i.e. the membership degree to the cluster to which we would assign the datum by maximum defuzzification) and the mean value of these membership degrees is computed. Here, the mean value for FCM is 0.81 in comparison to 0.96 for the AWFCM clustering technique. Nevertheless the methods by Gustafson and Kessel [48] or Gath and Geva [42] are also well suited to detect the structures of our example data, but GK and GG lead to a higher computational effort since the covariance matrices have to be estimated and inverted for each cluster in each iteration

step.

The AWFCM fuzzy clustering approach is also well suited for deriving rules from the clusters. Since the weighting of the attributes for each cluster provides information about the importance of the variables, we can neglect variables with very small weighting factors in the rules. For the presented artificial test data set we can e.g. consider to derive a fuzzy rule from cluster 2 invoking only the variable y.

Note that this approach differs from the idea to carry out a cluster analysis first and then apply something like a principal component analysis to each cluster. This would mean that the clustering has to take all attributes into account, whereas here the selection of relevant variables is already carried out during the clustering.

7.3 Attribute Weighting for transfer passenger data

We show the main clustering results and rule learning for the transfer passenger problem (described in section 2.1) in chapter 10. In that chapter the FCM-sized algorithm 6.1 based on the outlier objective function 3.1.4 as well as the original FCM based on outlier clustering are used as clustering techniques. The rule base is deduced for arrival and departure data for both clustering techniques in section 10.3. For the departure data and FCM-sized based on outlier clustering a rule-base is derived from a clustering result with 18 clusters. Therefore, attribute weighting clustering was carried out for the departure transfer passenger data and 18 clusters. The resulting weights are shown in table 7.3 for all 18 clusters and each attribute, i.e. maximal number of passengers that can be carried by a certain aircraft type, range of a flights destination (short-, medium-, or long-haul), time of departure, and percentage of passengers that previously arrived with another flight. Comparable results can be obtained for the arrival data set. The weighting exponent t was set to 2.5 and the constraint parameter $a = 1$ was chosen. Note that the cluster numbers are not correlated to the fuzzy set numbers in figure 10.3.

The weights indicate that the first attribute ”maximal number of passengers in a specific aircraft” has – except for cluster 13 and 16 – only a small influence on the cluster partition. The destination range either defines a clusters structure, e.g. cluster 3, 5, 6, 7, 9, and 10, or has a relative small influence on a particular cluster. Such information can be used to analyse data in separate parts. Data belonging to clusters with a large weight for one attribute can be analysed whether the output domain (amount of transfer passengers) is also defined by that attribute (and in that sense one attribute is sufficient) or if additional attributes are needed. Evenly weighted attributes indicate that the corresponding cluster is well defined with the attributes under consideration. The relatively high influence of the destination attribute indicates a problem that might arise with categorical variables. Categorical attributes have no real variation. The distances of similar attributes are zero – the attribute values are identical – whereas the distances of differing attributes are relatively large in comparison to average distances of other attributes.

Table 7.3: Attribute weights for departure transfer passenger data set

weights	attributes			
cluster	Pax Max	Destination	Time	Transfer Pax
0	0.051	0.507	0.293	0.149
1	0.023	0.148	0.745	0.084
2	0.002	0.073	0.916	0.009
3	0.001	0.939	0.048	0.012
4	0.039	0.063	0.870	0.028
5	0.000	1.000	0.000	0.000
6	0.000	1.000	0.000	0.000
7	0.022	0.946	0.016	0.015
8	0.104	0.259	0.423	0.215
9	0.000	0.994	0.003	0.003
10	0.000	1.000	0.000	0.000
11	0.038	0.660	0.132	0.170
12	0.003	0.759	0.172	0.065
13	0.247	0.306	0.229	0.217
14	0.014	0.685	0.288	0.013
15	0.004	0.008	0.986	0.002
16	0.216	0.290	0.286	0.208
17	0.085	0.152	0.144	0.619

This clustering technique is able to indicate for clustering tasks not based on classification problems whether the selected attributes are suited to describe the selected output attribute. For classification problems the error rate can be used to validate a classification. For other approximations with non-discrete output values, validity measures evaluate the whole classification independently whether an attribute is an output or input domain. If attribute weighting clustering is used and the weighting factors are very small for the output attribute(s) in all clusters, the chosen input attributes might be not sufficient to describe the output behaviour.

Chapter 8

Context Sensitive Fuzzy Clustering

In this chapter an objective function-based fuzzy clustering technique that incorporates linear combinations of attributes in the distance function is introduced [58]. The scope of this method is to develop a clustering technique that is able to classify a data set comparing not single attribute vectors of the sample data but some kind of regions. Thus this method can be applied in image segmentation or texture classification. Consider e.g. a grey-scale image as in figure 8.5. Pixel-wise comparison of these parts would lead to a countless number of groups in the best case. Adding up the grey-values of the whole region and comparing this value in-between the groups would be more effective. The main application field of this method is image processing where a comparison pixel by pixel is usually not adequate, but the environment of a pixel or groups of pixels characterise important properties of an image or parts of it. This clustering method is referred to as *context sensitive clustering* (*CS*).

Therefore, a new distance measure is defined. Formally the resulting algorithm is a generalisation of the FCM, see section 3.2.1, as well as the Gustafson-Kessel algorithm restricted to diagonal fuzzy covariance matrices, see the axes parallel Gustafson-Kessel algorithm in section 3.2.2.

This clustering technique seems to be well suited to determine groups of similar images. Problems may arise if no diverging areas for the groups or classes of images (describing a special class) can be found.

8.1 Formal Definition

Let us define the distance between a datum x_k and a cluster (vector) v_i by

$$d^2(\mathsf{v}_i, x_k) = \mathcal{D}_{CS} = \sum_{I \in \mathcal{I}} \alpha_I \left(\sum_{s \in I} x_k^{(s)} - \sum_{s \in I} v_i^{(s)} \right)^2 . \tag{8.1}$$

$x_k^{(s)}$ and $v_i^{(s)}$ indicate the s'th coordinates of the vectors x_k and v_i, respectively. $\mathcal{I}$ is a set of sets of indices (coordinates), i.e. $\mathcal{I} \subseteq 2^{\{1,\ldots,p\}}$ when we have to deal with p-dimensional data vectors. The parameters α_I can be considered as fixed or adapted during clustering individually for each cluster subject to the constraint

$$\prod_{I\in\mathcal{I}} \alpha_I = 1. \tag{8.2}$$

The distance measure is illustrated in figure 8.1. For figure 8.1(a) the subsets $I \in \mathcal{I}$ containing 1 variable each have been chosen for α_I, with $\alpha_{\{0\}} = 0.5$ and $\alpha_{\{1\}} = 2$, whereas the subsets $\{0\}, \{0\}$ and $\{0,1\}$ with $\alpha_{\{0\}} = 0.5$, $\alpha_{\{1\}} = 1$ and $\alpha_{\{0,1\}} = 2$ have been used for figure 8.1(b).

The idea of this context sensitive clustering is that certain subsets of the variables of data vectors yield similar values when we sum them up instead of comparing them one by one. A typical application of this approach is image recognition, where two similar images or regions might not correspond to each other pixel by pixel, but for instance the sum of the grey values in smaller parts $\mathcal{I}$ might almost coincide.

Based on this approach we can derive an alternating optimisation scheme for fuzzy clustering using this distance measure. The objective function that has to be minimised is of the following form.

$$J(X,U,\mathsf{v}) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m \cdot \sum_{I\in\mathcal{I}} \alpha_I \left(\sum_{s\in I} x_k^{(s)} - \sum_{s\in I} v_i^{(s)}\right)^2. \tag{8.3}$$

Using $\mathcal{D}_{CS}$ as distance measure, the objective function 8.3 can be written as

$$J(X,U,\mathsf{v}) = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m \cdot \mathfrak{D}_{CS}.$$

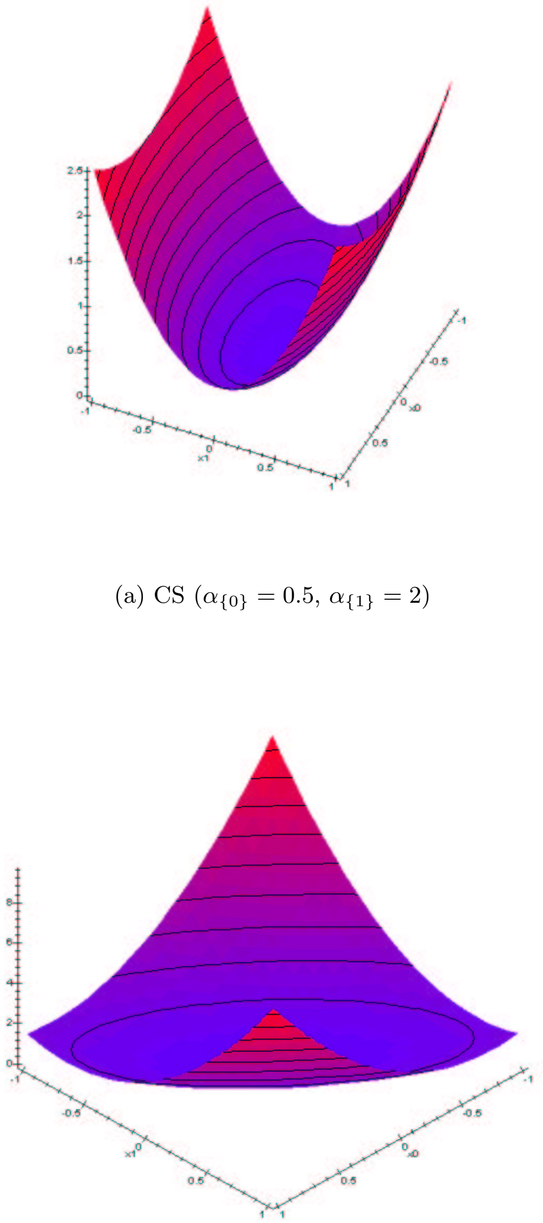
Therefore, choosing one of the basic objective functions described in section 3.1 gives us the update equations for the membership degrees. $\mathcal{D}_{CS}$ has to be inserted as distance measure in any of the basic clustering algorithms 3.1, 3.2, 3.3, or 3.4, respectively.

Theorem 8.1 (Context Sensitive Cl., Coordinate Set Parameter)

If the parameter α_I should be adapted during the iteration procedure, differentiating (8.3) leads to a calculation instruction (8.4) for the parameter α_I as a necessary condition for the objective function to adopt a minimal value. With card($\mathcal{I}$) the cardinality (number of elements) of the set $\mathcal{I}$ we obtain

$$\alpha_I = \frac{\sqrt[card(\mathcal{I})]{\prod_{J\in\mathcal{I}} \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m \cdot \left(\sum_{s\in J}\left(x_k^{(s)} - v_i^{(s)}\right)\right)^2}}{\sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m \cdot \left(\sum_{s\in I}\left(x_k^{(s)} - v_i^{(s)}\right)\right)^2}, \tag{8.4}$$

see proof 8.1.



(a) CS ($\alpha_{\{0\}} = 0.5$, $\alpha_{\{1\}} = 2$)

(b) CS ($\alpha_{\{0\}} = 0.5$, $\alpha_{\{1\}} = 1$, $\alpha_{\{0,1\}} = 2$)

Figure 8.1: Distance for the CS to $v^\top = (0, 0)$

Proof 8.1 (Context Sensitive Cl., Coordinate Set Parameter)

With condition (8.2) we obtain the Lagrange function

$$\begin{aligned} J_\lambda(X,U,v) &= \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m \cdot \sum_{I\in\mathcal{I}}\alpha_I\left(\sum_{s\in I}x^{(s)} - \sum_{s\in I}v^{(s)}\right)^2 \\ &\quad -\lambda\cdot\left(\prod_{L\in\mathcal{I}}\alpha_L - 1\right). \end{aligned}$$

Differentiating the Lagrange function leads to

$$\frac{\partial J_\lambda(X,U,v)}{\partial\alpha_I} = \sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}^m\cdot\left(\sum_{s\in I}x_k^{(s)} - \sum_{s\in I}v_i^{(s)}\right)^2 - \lambda\cdot\prod_{L\in\mathcal{I}-\{I\}}\alpha_L \overset{!}{=} 0$$

with $\prod_{L\in\mathcal{I}-\{I\}}\alpha_L = \frac{1}{\alpha_I}$, *see 8.2, this leads to*

$$\frac{\lambda}{\alpha_I} = \sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}^m\cdot\left(\sum_{s\in I}x_k^{(s)} - \sum_{s\in I}v_i^{(s)}\right)^2. \tag{1}$$

Then the product of (1) *for all* $I\in\mathcal{I}$ *leads to*

$$\lambda^{card(\mathcal{I})} = \prod_{I\in\mathcal{I}}\sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}^m\cdot\left(\sum_{s\in I}x_k^{(s)} - \sum_{s\in I}v_i^{(s)}\right)^2$$

and therefore

$$\lambda = \sqrt[card(\mathcal{I})]{\prod_{I\in\mathcal{I}}\sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}^m\cdot\left(\sum_{s\in I}x_k^{(s)} - \sum_{s\in I}v_i^{(s)}\right)^2}.$$

Inserting λ *in* (1) *gives us the update equation for* α_I

$$\alpha_I = \frac{\sqrt[card(\mathcal{I})]{\prod_{J\in\mathcal{I}}\sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}^m\cdot\left(\sum_{s\in J}\left(x_k^{(s)} - v_i^{(s)}\right)\right)^2}}{\sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}^m\cdot\left(\sum_{s\in I}\left(x_k^{(s)} - v_i^{(s)}\right)\right)^2} \qquad \diamond$$

Theorem 8.2 (Context Sensitive Clustering, Cluster Centres)

In a similar way we obtain a necessary condition for the cluster centres (8.5), see proof 8.2.

$$\sum_{I\in\mathcal{I}|r\in I}\left(\alpha_I\cdot\sum_{s\in I}v_i^{(s)}\right) = \frac{\sum_{k=1}^{n}\left(u_{ik}^m\cdot\sum_{I\in\mathcal{I}|r\in I}\left(\alpha_I\cdot\sum_{s\in I}x_k^{(s)}\right)\right)}{\sum_{k=1}^{n}u_{ik}^m} \tag{8.5}$$

Equation (8.5) is a system of linear equations, but variable and highly dependent on the choice of the sets in $\mathcal{I}$. So the following heuristics (8.6) is chosen to estimate the parameters $v_i^{(s)}$ in (8.5).

$$\begin{aligned} v_i^{(r)} \;=\; & \frac{\sum_{k=1}^{n}\left(u_{ik}^{m}\cdot\sum_{I\in\mathcal{I}|r\in I}\left(\alpha_I\cdot\sum_{s\in I}x_k^{(s)}\right)\right)}{\left(\sum_{k=1}^{n}u_{ik}^{m}\right)\cdot\left(\sum_{I\in\mathcal{I}|r\in I}\alpha_I\right)} \\ & -\frac{\left(\sum_{k=1}^{n}\left(u_{ik}^{m}\right)\right)\cdot\left(\sum_{I\in\mathcal{I}|r\in I}\alpha_I\cdot\sum_{s\in I\setminus\{r\}}v_i^{(s)}\right)}{\left(\sum_{k=1}^{n}u_{ik}^{m}\right)\cdot\left(\sum_{I\in\mathcal{I}|r\in I}\alpha_I\right)} \end{aligned} \tag{8.6}$$

In (8.6) we have to make sure that none of the parameters $v_i^{(r)}$ is allowed to be placed outside the domain of the corresponding data set's attribute. If values outside the domain would be allowed, the error of the former placed cluster centres would be neglected in placing the next coordinate $v_i^{(r)}$ in great distance of all observed data coordinates. In calculating $v_i^{(r')}$ the former calculated parameters $v_i^{(r)}$ $(r < r')$ are used in equation (8.6). Otherwise each prototype coordinate would adapt the whole error between the sum of corresponding data coordinates and the coordinates of the former prototype. A similar heuristic approach to determine the prototypes is also used for the fuzzy c-ellipses [43] and fuzzy c-rings [88] clustering techniques.

Proof 8.2 (Context Sensitive Clustering, Cluster Centres)

$$\begin{aligned} J(X,U,\mathsf{v}) &= \sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}^{m}\cdot\sum_{I\in\mathcal{I}}\alpha_I\cdot\left(\sum_{s\in I}x_k^{(s)}-\sum_{s\in I}v_i^{(s)}\right)^2 \\ &= \sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}^{m}\cdot\sum_{I\in\mathcal{I}}\alpha_I\Bigg(\sum_{s\in I}\left(x_k^{(s)}\right)^2+2\cdot\sum_{s\in I}\sum_{r\in I|r>s}x_k^{(s)}\cdot x_k^{(r)} \\ &\quad -2\cdot\sum_{s\in I}\sum_{r\in I}x_k^{(s)}\cdot v_i^{(r)}+\sum_{s\in I}\left(v_i^{(s)}\right)^2+2\cdot\sum_{s\in I}\sum_{r\in I|r>s}v_i^{(s)}\cdot v_i^{(r)}\Bigg) \end{aligned}$$

Computing the partial derivative of $J(X,U,\mathsf{v})$ with respect to $v_i^{(l)}$ leads to

$$\frac{\partial J(X,U,v)}{\partial v_i^{(l)}} = \sum_{k=1}^{n}u_{ik}^{m}\cdot\sum_{I\in\mathcal{I}|l\in I}\alpha_I\cdot\left(2\cdot v_i^{(l)}+2\cdot\sum_{s\in I-\{l\}}v_i^{(s)}-2\cdot\sum_{s\in I}x_k^{(s)}\right)\overset{!}{=}0.$$

With $v_i^{(l)}+\sum_{s\in I-\{l\}}v_i^{(s)}=\sum_{s\in I}v_i^{(s)}$ we get

$$\sum_{k=1}^{n}u_{ik}^{m}\cdot\sum_{I\in\mathcal{I}|l\in I}\left(\alpha_I\cdot\sum_{s\in I}v_i^{(s)}\right)=\sum_{k=1}^{n}\left(u_{ik}^{m}\cdot\sum_{I\in\mathcal{I}|l\in I}\alpha_I\cdot\sum_{s\in I}x_k^{(s)}\right)$$

and therefore

$$\sum_{I\in\mathcal{I}|l\in I}\left(\alpha_I\cdot\sum_{s\in I}v_i^{(s)}\right) = \frac{\sum_{k=1}^n\left(u_{ik}^m\cdot\sum_{I\in\mathcal{I}|l\in I}\alpha_I\cdot\sum_{s\in I}x_k^{(s)}\right)}{\sum_{k=1}^n u_{ik}^m}. \quad \diamond$$

The parameter α_I determines the influence of one particular subset of attributes. If e.g. the class determining areas of an image are known in advance, it is not necessary to adapt the α_I (assuming that each I contains the variables of one significant area) during the clustering procedure. In the case that no supplementary information about the data set is given, it is possible to define more subsets I than are expected to be necessary for the task of pattern recognition and adapt the α_I in order to adapt the influence of certain subsets.

This approach can be seen as a generalisation of the axes-parallel version of the Gustafson-Kessel algorithm, see section 3.2.2 and [73]. The AGK is based on the distance function $\mathcal{D}_{AGK} = (x-v)^\top C(x-v)$ where C is a diagonal matrix with determinant 1. Therefore, if $c_1,\dots,c_p$ are the diagonal elements of C, the distance function can be written as

$$\mathcal{D}_{AGK} = \sum_{s=1}^p c_s(x_s-v_s)^2 \tag{8.7}$$

with the constraint

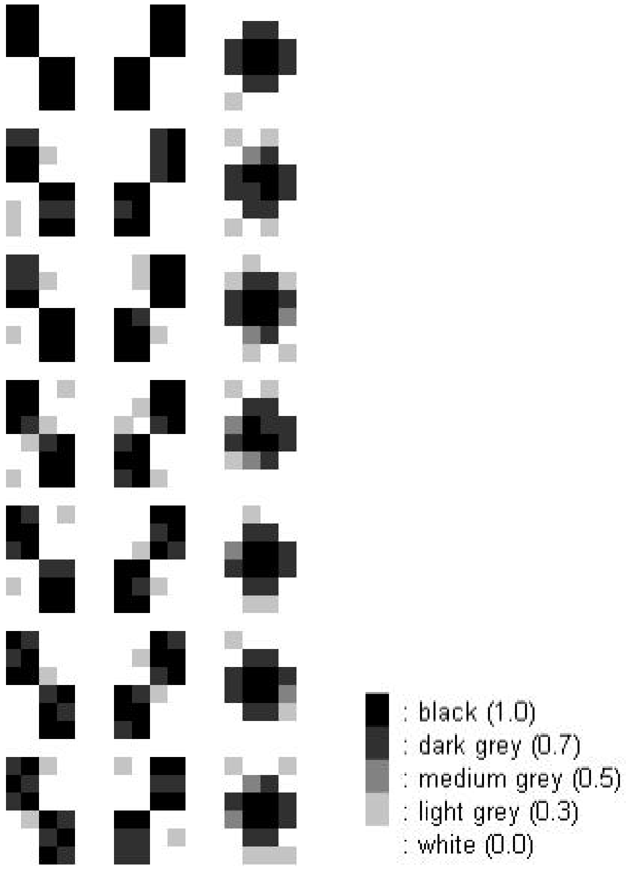
$$\prod_{s=1}^p c_s = 1. \tag{8.8}$$

When we choose $\mathcal{I}=\{\{1\},\dots,\{p\}\}$, then our initial equations (8.1) and (8.2) correspond to equations (8.7) and (8.8), respectively. In this case, we define the comparable regions as the single attributes. However, context sensitive clustering allows to define whatever combination of attributes.

8.2 Illustrative Example

Figure 8.2 shows an artificial example with three different kinds of grey-scale images. In figure 8.2(a) the data set is shown. The grey scales are equivalent to real numbers as denoted in figure 8.2(b). During the clustering procedure we adapted the values for the α_I and set the number of clusters to three. The set of sets $\mathcal{I}$ is constructed of all sets containing one element – one particular point of the picture's area – and four sets with six points each (the top-left, top-right, bottom-left and bottom-right regions of the images). The resulting prototypes for the three clusters are shown in figure 8.3(a). The highest membership degrees of the data points to the clusters are presented together with the corresponding clusters in table 8.1. Table 8.3 contains the adapted values for the parameters α_I. The set notation is illustrated in figure 8.4.

In figure 8.3 results for the data set from figure 8.2 are shown. In the second example the set of sets $\mathcal{I}$ contains only the subsets with six elements each – as described above. Figure 8.3(b) shows the resulting prototypes and



(a) Box data set

(b) Box grey values

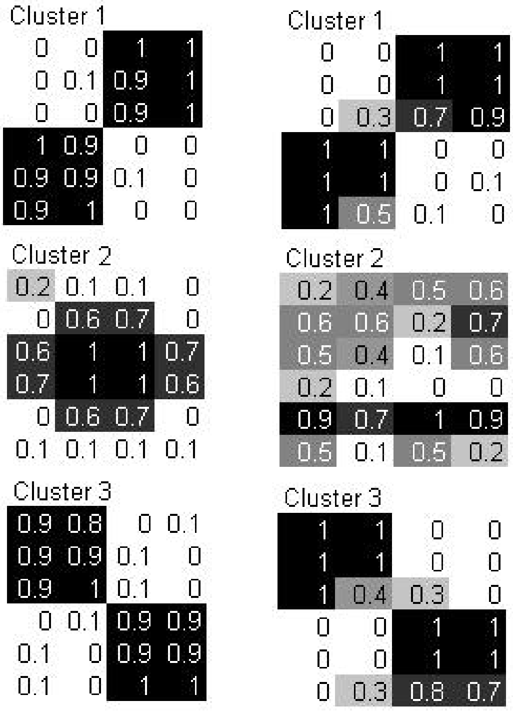
Figure 8.2: Box Images

Table 8.1: Class determining membership degrees for Box Images

	maximal membership degrees		
datum	4 and 1-elemental subsets	4-elemental subsets	cluster
1	0.979	0.976	3
2	0.971	0.994	3
3	0.967	0.978	3
4	0.963	0.987	3
5	0.975	0.998	3
6	0.975	0.995	3
7	0.948	0.983	3
8	0.984	0.982	1
9	0.975	0.983	1
10	0.976	0.987	1
11	0.962	0.995	1
12	0.951	0.997	1
13	0.973	0.997	1
14	0.942	0.974	1
15	0.980	0.992	2
16	0.967	0.995	2
17	0.936	0.987	2
18	0.964	0.995	2
19	0.974	0.996	2
20	0.975	0.996	2
21	0.952	0.985	2

Table 8.2: Influence values α_I for the result in figure 8.3(b)

set I	α_I
$\{a1, a2, b1, b2, c1, c2\}$	0.82
$\{a3, a4, b3, b4, c3, c4\}$	1.19
$\{d1, d2, e1, e2, f1, f2\}$	0.93
$\{d3, d4, e3, e4, f3, f4\}$	1.11



Cluster 1

0	0	1	1
0	0.1	0.9	1
0	0	0.9	1
1	0.9	0	0
0.9	0.9	0.1	0
0.9	1	0	0

Cluster 2

0.2	0.1	0.1	0
0	0.6	0.7	0
0.6	1	1	0.7
0.7	1	1	0.6
0	0.6	0.7	0
0.1	0.1	0.1	0.1

Cluster 3

0.9	0.8	0	0.1
0.9	0.9	0.1	0
0.9	1	0.1	0
0	0.1	0.9	0.9
0.1	0	0.9	0.9
0.1	0	1	1

(a) Prototypes (all subsets I with 4 and 1 elements each)

Cluster 1

0	0	1	1
0	0	1	1
0	0.3	0.7	0.9
1	1	0	0
1	1	0	0.1
1	0.5	0.1	0

Cluster 2

0.2	0.4	0.5	0.6
0.6	0.6	0.2	0.7
0.5	0.4	0.1	0.6
0.2	0.1	0	0
0.9	0.7	1	0.9
0.5	0.1	0.5	0.2

Cluster 3

1	1	0	0
1	1	0	0
1	0.4	0.3	0
0	0	1	1
0	0	1	1
0	0.3	0.8	0.7

(b) Prototypes (subsets I with 4 elements each)

Figure 8.3: Box Image clustering results

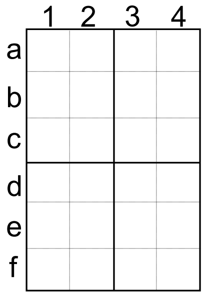


Figure 8.4: Attribute notation for the Box Image data set

Table 8.3: Influence values α_I for the result in figure 8.3(a)

set I	α_I
$\{a1, a2, b1, b2, c1, c2\}$	0.19
$\{a3, a4, b3, b4, c3, c4\}$	0.26
$\{d1, d2, e1, e2, f1, f2\}$	0.22
$\{d3, d4, e3, e4, f3, f4\}$	0.25
$\{a1\}$	0.81
$\{a2\}$	0.86
$\{a3\}$	1.08
$\{a4\}$	1.08
$\{b1\}$	1.49
$\{b2\}$	0.91
$\{b3\}$	1.09
$\{b4\}$	2.03
$\{c1\}$	1.17
$\{c2\}$	1.95
$\{c3\}$	0.85
$\{c4\}$	3.88
$\{d1\}$	2.71
$\{d2\}$	0.92
$\{d3\}$	1.29
$\{d4\}$	1.65
$\{e1\}$	0.85
$\{e2\}$	1.68
$\{e3\}$	1.20
$\{e4\}$	1.08
$\{f1\}$	0.75
$\{f2\}$	1.34
$\{f3\}$	1.32
$\{f4\}$	1.52

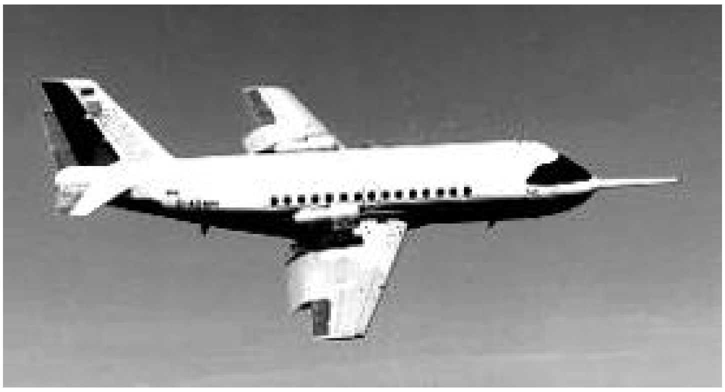


Figure 8.5: ATTAS image with 256 grey values and 276 x 147 pixel

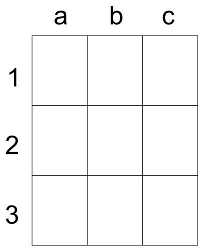


Figure 8.6: Block notation for the ATTAS image

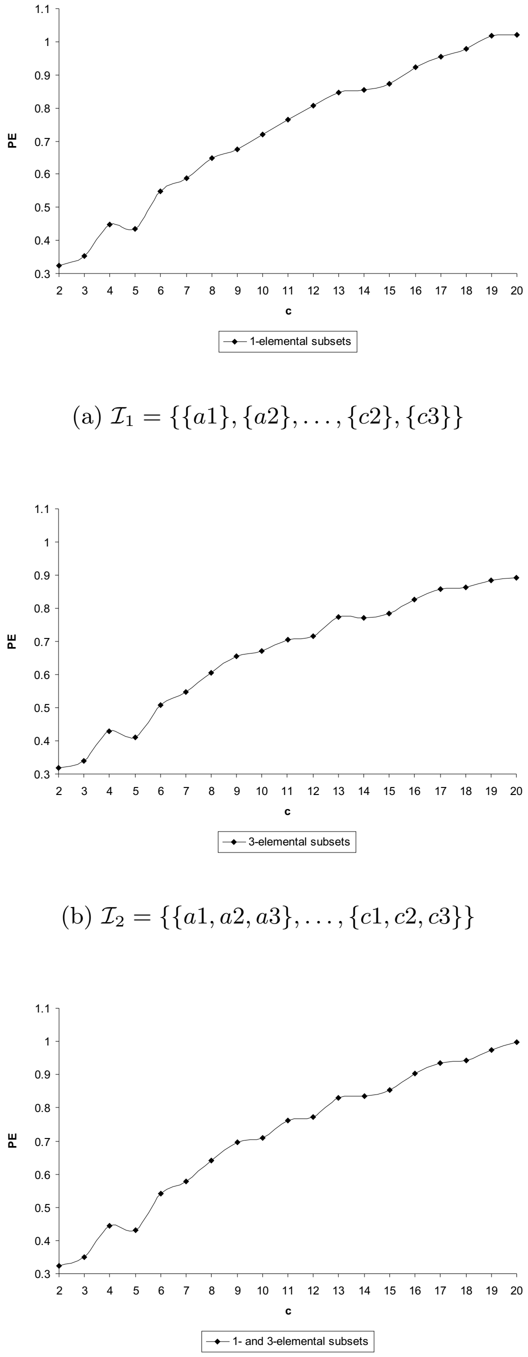
table 8.1 presents the highest membership degrees of the data points to the clusters together with the corresponding clusters. The adapted values for the parameters α_I are denoted in table 8.2. The corresponding clusters of the data points are determined correctly, also the prototypes for cluster one and three. Since all grey values are present in each particular subset I of $\mathcal{I}$ for the third group of data points (figure 8.2(a)), the corresponding prototype for cluster two in figure 8.3(b) is not able to reproduce the data images correctly – nevertheless the data points are correctly assigned.

8.3 Context Sensitive Analysis of ”ATTAS” image

Let us now illustrate the properties of the presented approach with a second more realistic example. In figure 8.5 the aircraft ”ATTAS” of the German Aerospace Centre is shown. The colours where reduced to 256 grey values and the size was set to 276 x 147 pixel. To generate a data set for the context sensitive fuzzy clustering approach, the picture was divided into blocks of 3 x 3 pixels leading to 4,508 data vectors with 9 attributes each. The attributes are the grey values of pixels included in the 3 x 3 block. Figure 8.6 illustrates the notation in one 3 x 3 block that is used for the sets α_I.

Three cases have been analysed for a varying number of clusters:

- $\mathcal{I} = \{\{a1\}, \{a2\}, \ldots, \{c2\}, \{c3\}\}$



(a) $\mathcal{I}_1 = \{\{a1\}, \{a2\}, \ldots, \{c2\}, \{c3\}\}$

(b) $\mathcal{I}_2 = \{\{a1, a2, a3\}, \ldots, \{c1, c2, c3\}\}$

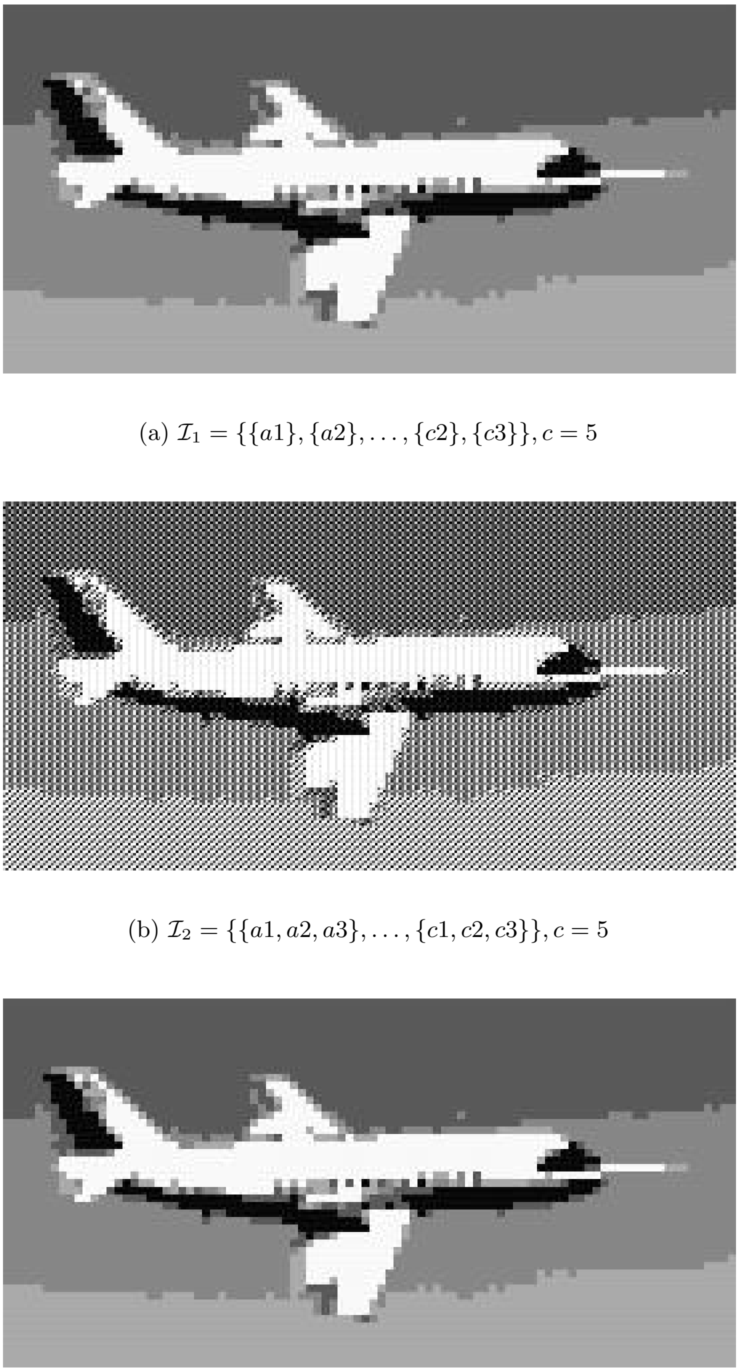
(c) $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2$

Figure 8.7: PE for ATTAS image and different $\mathcal{I}$

- $\mathcal{I} = \{\{a1, a2, a3\}, \dots, \{c1, c2, c3\}\}$
- $\mathcal{I} = \{\{a1\}, \{a2\}, \dots, \{c2\}, \{c3\}, \{a1, a2, a3\}, \dots, \{c1, c2, c3\}\}$

To determine the best number of clusters the partition entropy, see section 4.1.3, has been chosen as validity measure. The development of the validity functions are shown in figure 8.7.

The results for the local minimal PE-values are illustrated in figure 8.8. Here a local minimum was obtained at 5 clusters for each $\mathcal{I}$. To generate the figures, each data vector has been assigned to that cluster to which it had the highest membership degree. Afterwards, the original figure's grey values have been exchanged for the corresponding clusters prototype values. The real-valued prototype coordinates have been rounded to integer values in order to represent grey values.



(a) $\mathcal{I}_1 = \{\{a1\}, \{a2\}, \ldots, \{c2\}, \{c3\}\}, c = 5$

(b) $\mathcal{I}_2 = \{\{a1, a2, a3\}, \ldots, \{c1, c2, c3\}\}, c = 5$

(c) $\mathcal{I} = \mathcal{I}_1 \cup \mathcal{I}_2, c = 5$

Figure 8.8: Resulting ATTAS image for different $\mathcal{I}$

Chapter 9

Fuzzy Clustering for Rule Learning

Fuzzy rules are usually obtained from fuzzy clusters by projecting the clusters to the coordinate spaces. Especially in applying fuzzy clustering techniques to the task of rule learning it is not necessary to implement highly form-adaptable algorithms since more flexible clustering algorithms admitting complex cluster shapes generally result in a higher loss of information by rule generation. As long as we stay with such simple clustering algorithms as FCM or the parallel version of GK, loss of information in case of rule learning can be mostly avoided, see fig. 9.5. This is also valid for the size adaptable versions of these algorithms, see section 6 since the form describing distance measure is not changed. In case of rule learning the proposed modified versions of the described algorithms are a good alternative to more complex algorithms like the method introduced by Gath and Geva, see section 3.2.3. However, a loss of information by the process of rule generation is unavoidable and most of the algorithms presented in the preceding chapters can be applied to learn fuzzy rules from data for classification problems [44, 72, 70, 114, 37] or function approximation [71, 73, 107, 83]. The loss of information could be minimised if the clusters had the shapes of rectangles or hyperboxes. Unfortunately, such shapes lead to non-differentiable objective functions. One method to avoid a major part of this information loss is described in section 9.2 and [64]. There we start with partitions of the single domains and try to find a suitable partition for the data under consideration. Especially suited for the task of rule learning are the algorithms presented in section 6.1 where the algorithms are made more flexible with respect to the cluster shape without increasing the existing loss of information in case of rule learning. Namely the size-adaptable FCM or AGK version are appropriate.

Another possibility is to simplify the rule base after the rules have been extracted from the clustering results. Some methods to reduce the number of fuzzy sets in a rule base and use a similarity measure to simplify the rule base are described in [103, 105]. A method for classification tasks to tune and split the rule base has been described in [1, 2].

Fuzzy clusters are often described in form of a cluster representative and

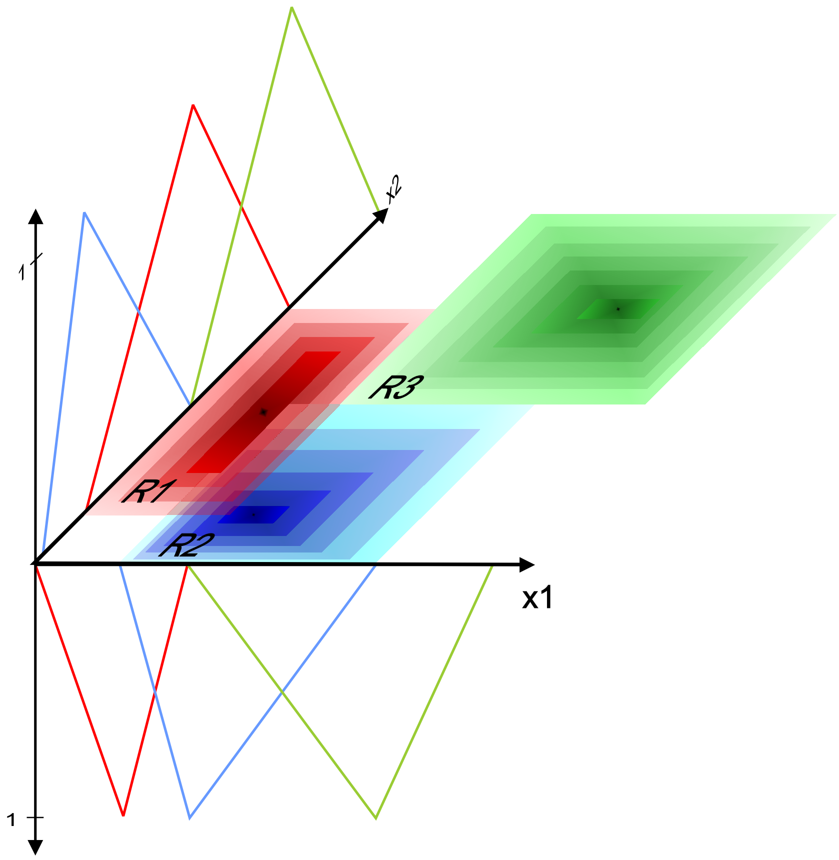


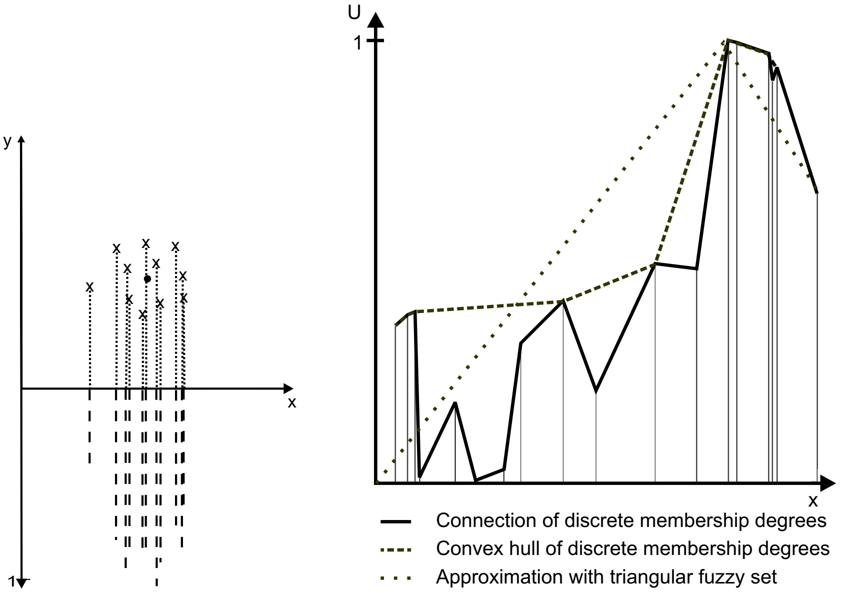
Figure 9.1: Vague areas induced by fuzzy rules

the membership degrees of the sample data points to the clusters. For one cluster the membership degrees are decreasing with increasing distance. If we think about solid clusters, a vague area with the cluster's prototype as centre is described. Now assume that we have a fuzzy rule consisting of triangular fuzzy sets for the single attributes. If we represent these fuzzy sets in the multidimensional input-output space of the fuzzy rule, a vague area similar to a cluster is described. This representation and the described vague area are illustrated in figure 9.1.

9.1 Rule Generation from Solid Clusters

The principle 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 [71, 107, 56]. The projection of a cluster to the s'th domain is obtained by taking the s'th coordinate of each data point and assigning to it the membership degree of the original data point to the cluster, see figure 9.2(a).

In this way a discrete fuzzy set is defined on the s'th coordinate space. To extend this fuzzy set to the whole s'th domain, a piecewise linear fuzzy set



(a) Projection of membership degrees

(b) Generating a parameterised fuzzy set

Figure 9.2: Generating a parameterised fuzzy set from a solid cluster

can be defined on the basis of these discrete points, an enveloping fuzzy set or a suitable approximation by a parameterised fuzzy set like a triangular or trapezoidal membership function can be chosen to simplify the handling of the resulting rule system [107]. The way to generate a parameterised fuzzy set is illustrated in figure 9.2(b) for triangular fuzzy sets.

One possibility to generate a trapezoidal fuzzy set is described in [107]. First the convex hull of the discrete fuzzy set is determined before a trapezoidal fuzzy set estimating the convex set is calculated. The formal calculation is done as denoted in algorithm 9.1. To calculate the convex hull and thereof the trapezoidal fuzzy set, the data vectors have to be ordered with respect to attribute s, the attribute whose fuzzy set corresponding to cluster i has to be determined. The membership degrees have to be sorted corresponding to the data vectors.

To obtain a fuzzy rule system of a fuzzy clustering/classification result, one has to carry out algorithm 9.1 for each cluster i and each domain s separately.

Algorithm 9.1 (Parameterisation (domain s, cluster i))

```
Choose()
{
   $u_{min} \in [0, 1[$;
   $\kappa \in \mathbb{R}_{>0}$ and $\kappa < \max_{k \in \{1,\ldots,n\}} \{x_k^{(s)}\} - \min_{k \in \{1,\ldots,n\}} \{x_k^{(s)}\}$;
}

Initialise()
{
   sort $X$w.r.t. attribute $s$;
   sort $U$w.r.t. new order of $X$;
}

MakeConvex()
{
   $Y := \varnothing$
   $u_{actual} := u_{min}$;
   for all $k \in \{1, \ldots, n\}$
      $if(u_{ik} \geq u_{min}) \wedge (u_{ik} \geq u_{actual})$
         {
            $Y := Y \cup \{x_k\}$;
            $u_{actual} := u_{ik}$;
         }
   $u_{actual} := u_{min}$;
   for all $k \in \{n, \ldots, 1\}$
      $if(u_{ik} \geq u_{min}) \wedge (u_{ik} \geq u_{actual})$
         {
            $Y := Y \cup \{x_k\}$;
            $u_{actual} := u_{ik}$;
         }
}
```

Algorithm 9.1 (Parameterisation – continued)

```
ParameteriseConvexSet()
{
  $p_1 := x_k^{(s)}$ where $x_k \in Y$ and $u_{ik} = \min_{l \in \{1,\ldots,n\} \wedge x_l \in Y} \{u_{il}\}$;
  $p_4 := x_k^{(s)}$ where $x_k \in Y$ and $u_{ik} = \max_{l \in \{1,\ldots,n\} \wedge x_l \in Y} \{u_{il}\}$;
  $p_2 := p_1 + \frac{p_4 - p_1}{3}$;
  $p_3 := p_1 + \frac{2 \cdot (p_4 - p_1)}{3}$;
  do
  {
    for $j \in \{1, \ldots, 4\}$
      if $((j > 1) \wedge ((p_j - \delta) < p_{j-1}))$
        $p_j^{(1)} := p_{j-1}$;
      else
        $p_j^{(1)} := p_j - \kappa$;
      if $((j < 4) \wedge ((p_j + \kappa) > p_{j+1}))$
        $p_j^{(2)} := p_{j+1}$;
      else
        $p_j^{(2)} := p_j + \kappa$;
      $p_j := q \in \{p_j^{(1)}, p_j, p_j^{(2)}\}$
        where the smallest squared error between the convex hull
        – described by $Y$ and $u_{ik}$– and the trapezoidal fuzzy set
        – described by $\{p_1, p_2, p_3, p_4\}$ – is obtained.
  }while(trapezoidal set not satisfying)
}
```

Sugeno and Yasukawa evaluated for a fixed number of 20 iteration steps the iteration in procedure *ParamteriseConvexSet*() of algorithm 9.1. They selected κ as 5% of the attribute (domain) corresponding to the projected cluster. The parameters of the trapezoidal fuzzy sets are illustrated in figure 9.3.

Similar to the derivation of trapezoidal fuzzy sets, other parameterised fuzzy sets e.g. triangular can be estimated. The aim is to let the squared error between the convex hull of the discrete membership degrees and the parameterised fuzzy set as small as possible.

Considering the projection of each cluster for each single domain a cluster

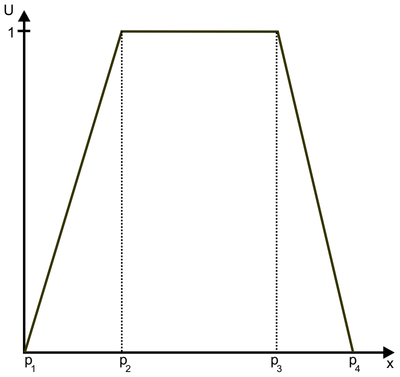


Figure 9.3: Trapezoidal fuzzy set defined by four parameters

induces the rule

$$\text{if } \xi_1 \text{ is } \mu_1 \text{ and} \ldots \text{and } \xi_{r_i-1} \text{ is } \mu_{r_i-1} \text{ then } \xi_{r_i} \text{ is } \mu_{r_i} \text{ and} \ldots \text{and } \xi_{r_o} \text{ is } \mu_{r_o}. \quad (9.1)$$

Here, μ_l denotes the (extension of the) l'th projection of the considered cluster. The symbols $\xi_1, \ldots, \xi_{r_i-1}$ are input variables and $\xi_{r_i}, \ldots, \xi_{r_o}$ are output variables. The attributes $\{1, \ldots, p\}$ of the fuzzy clustering task are ordered in such a way that $p_o \geq p_i$, $p_i \in \{1, \ldots, p\}$ and $p_o = p$. In this way a Mamdani-type fuzzy controller is defined [71, 107]. For restrictions on max-min rules in multidimensional classification problems see [69] and [94] for the influence of different t-norms on the cluster shapes. A number of variants of this principle were proposed by different authors to solve control, function approximation and classification problems with fuzzy rules (for a brief overview see [64]).

To derive classification rules where discrete classes appear in the conclusions of the rules only the input variables have to be taken into account to build the fuzzy sets. Instead of the fuzzy set belonging to the output variable the discrete class of the data having the highest membership degree to the cluster under consideration is used as output [72]. Other assignments for the fuzzy rules output as e.g. the weighted class attribute of the data assigned to the corresponding cluster are possible. In that way, Mamdani-type classification rules, see e.g. [87], can be derived similar to the task of function approximation. Instead of the output variables $\xi_{r_i}, \ldots, \xi_{r_o}$ and the corresponding parameterised fuzzy sets, the class of the rule representing cluster – c_i – has to be determined, see section 4.1.1. This parameter is used as output value for the fuzzy rule, see equation (9.2).

$$\text{if } \xi_1 \text{ is } \mu_1 \text{ and} \ldots \text{and } \xi_{r_i-1} \text{ is } \mu_{r_i-1} \text{ then class is } c_i. \quad (9.2)$$

For function approximation also the fuzzy rule system first stated by Takagi and Sugeno [108] is commonly used. They build the fuzzy rules premise in the same way as for the Mamdani-type fuzzy controller but evaluate the single rule's output values with (usually linear) functions depending on the input variables, see equation (9.3).

$$\text{if } \xi_1 \text{ is } \mu_1 \text{ and} \ldots \text{and } \xi_{r_i-1} \text{ is } \mu_{r_i-1} \text{ then} \quad (9.3)$$
$$\xi_{r_i} = f_{r_i}(\xi_1, \ldots, \xi_{r_i-1}) \text{ and} \ldots \text{and } \xi_{r_o} = f_{r_o}(\xi_1, \ldots, \xi_{r_i-1}).$$

The function $f_{R_q}(\xi_1, \ldots, \xi_{r_i-1})$ is often of the following form for each output domain and each fuzzy rule R_q

$$f_{R_q}(\xi_1, \ldots, \xi_{r_i-1}) \;=\; \beta_0 + \sum_{l=1}^{r_i-1} \beta_l \cdot \xi_l.$$

Up to now we have seen how single fuzzy rules can be generated from solid clustering algorithms and which kind of rules are suited for the results of fuzzy clustering tasks. To further illustrate the difficulties in stating complete fuzzy rule systems using fuzzy clustering results, we have first to show how fuzzy rule systems are evaluated.

If we denote the fuzzy rules by R_q, the overall output value (for one domain) is evaluated in case of Takagi-Sugeno-type fuzzy controllers due to equation (9.4).

$$\xi_{out} \;=\; \frac{\sum_{R_q} \mu_{R_q} \cdot f_{R_q}(\xi_1, \ldots, \xi_{r_i-1})}{\sum_{R_q} \mu_{R_q}} \quad (9.4)$$

Here, μ_{R_q} is the *firing degree* of the rule R_q. The firing degree is used to determine the representativeness of a particular rule for certain input values. It is calculating considering the membership degrees that single attributes receive for the corresponding rule's input fuzzy set. The firing degree is determined using a *t-norm* over the single membership degrees for all input attributes/domains, often the minimum is used as t-norm.

In case of Mamdani-type rule systems, the evaluation of a rule system is illustrated in figure 9.4. As for Takagi-Sugeno-type fuzzy controllers, the firing degree of a fuzzy rule has to be determined first. Then the fuzzy sets corresponding to the output values are cut off at the firing degree level. In this way reduced fuzzy sets are evaluated for each single rule. To combine the output fuzzy sets, the *t-conorm* – corresponding to the t-norm used to calculate the firing degree – has to be evaluated. In case of the minimum as t-norm, the corresponding t-conorm is the maximum. In this way a new complex output fuzzy set is build as denoted in figure 9.4. With a *defuzzification strategy* – e.g. the centre of gravity or the mean of maxima – the crisp output value is determined.

The approaches of deriving fuzzy rule systems from clustering results have to face the problem that the fuzzy clustering algorithm yields a fuzzy partition 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

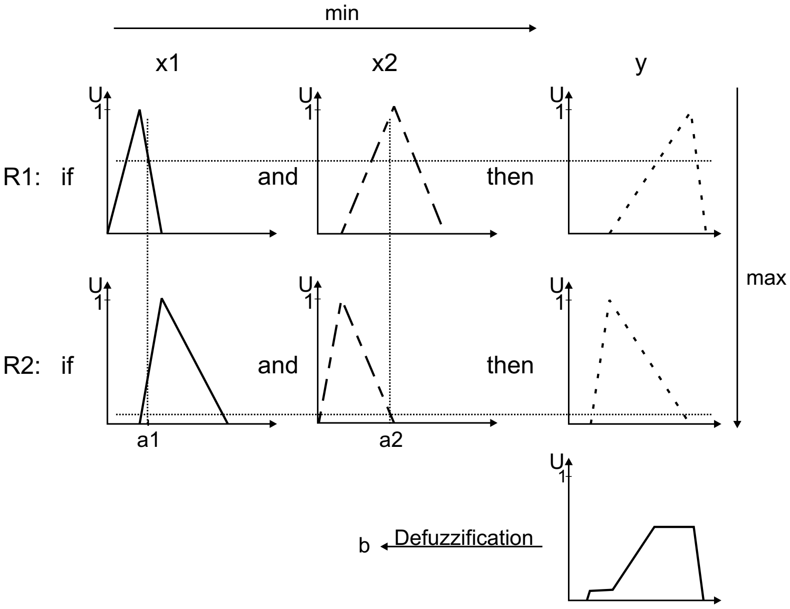


Figure 9.4: Mamdani fuzzy rule evaluation

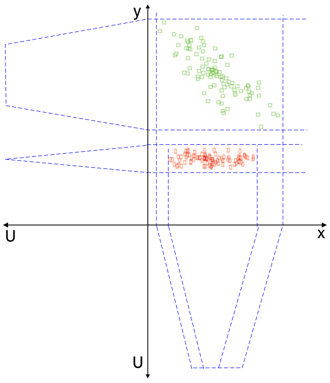


Figure 9.5: Overlaying and overlapping fuzzy sets

original fuzzy cluster cannot be reconstructed from the fuzzy sets appearing in the if-then rule derived from the cluster, see figure 9.5.

The aim of a fuzzy rule system is easy interpretation but if too many fuzzy sets are in one partition overlapping and overlaying one another even the expert has difficulties reading such a description of system behaviour. Since most cluster algorithms detecting solid clusters where created without the intention of finding rules, problems in rule generation occur. Often, each vague area described by a fuzzy cluster leads to one fuzzy rule. In this approach the fuzzy partitions of the single domains contain one fuzzy set for each of the clusters. It is not possible to reuse one fuzzy set in other rules than the rule depending on this cluster. To avoid these drawbacks, the grid clustering algorithm was designed as a top down approach.

There are other approaches to reduce this loss of information: [70] recom-

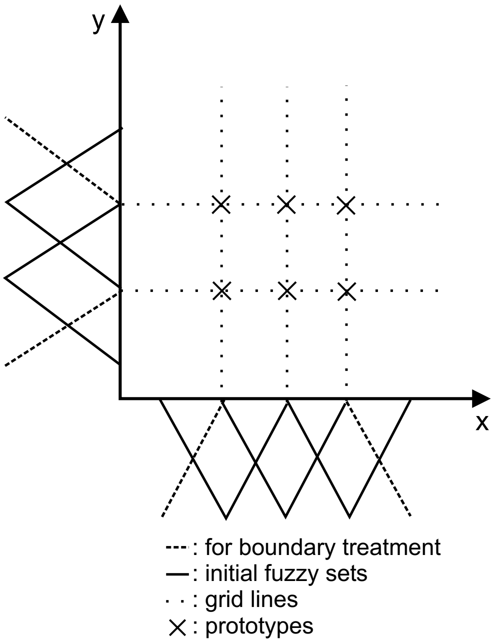


Figure 9.6: Initial fuzzy sets for grid clustering

mends to restrict to diagonal matrices C_i when using the GK, see section 3.2.2, or GG, see section 3.2.3, 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 [107] 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. For a short overview on the ability of fuzzy rule-based classifiers to match classification boundaries see e.g. [82].

9.2 Grid Clustering

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. These considerations lead to the following approach which is also described in [64, 55, 67, 66]. We call this approach *grid clustering* or shortly *Grid*. In [100] a technique called Grid-Based data analysis is presented. The aim of this BANG-Clustering is to cluster large data sets hierarchically and is not intended for rule generation. Assume a data set

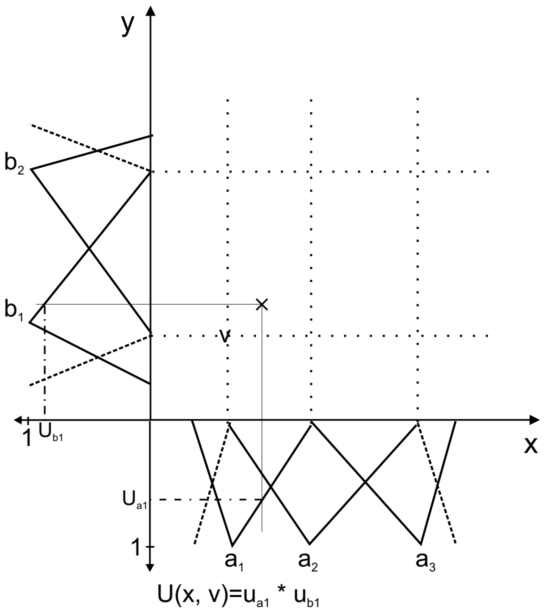


Figure 9.7: Membership degree evaluation for grid clustering

$X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$ is given. We are looking for fuzzy partitions on the single domains consisting of triangular membership functions 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, see figure 9.6.

In this case we have to determine a suitable grid in the multidimensional space as is already illustrated in figure 9.6.

The membership degree of a data point to a cluster represented by a grid point is defined as the product of the membership degrees of the triangular membership functions whose tips are the projections of the grid point, see figure 9.7. It is possible to choose any other t-norm than the product to determine the overall membership degree of one datum to one particular cluster.

Assuming the number c_s of triangular membership functions as predefined for each domain $s \in \{1, \ldots, p\}$ we start the fuzzy clustering procedure 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 the triangular membership functions are updated by computing new tips as the cluster centres, i.e. as

$$v_i^{(s)} = \frac{\sum_{k=1}^{n} u_{iks}^m \cdot x_k^{(s)}}{\sum_{k=1}^{n} u_{iks}^m} \tag{9.5}$$

where $v_i^{(s)}$ symbolises the actualised tip and $x_k^{(s)}$ denotes the s'th projection of datum x_k. u_{iks} is the membership degree of datum x_k to the triangular fuzzy set with its tip at $\hat{v}_i^{(s)}$, the tip of the last iteration step.

To handle the fuzzy sets at the left and right boundary in each dimension we 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, see figure 9.6.

The algorithm scheme for the grid clustering algorithm is shown in algorithm 9.2. Note that the initialisation procedure guarantees the prototype coordinates for one dimension to be ordered. In the following, c_s denotes the number of grid coordinates in domain s.

Algorithm 9.2 (Grid Clustering)

```
Choose()
{
  $m \in \mathbb{R}_{>1}$;
  for all $s \in \{1, \dots, p\}$
    $c_s \in \mathbb{N}_{\geq 2}$;
  $\epsilon > 0$;
}

Initialise()
{
  for all  $s \in \{1, \dots, p\}$
  {
    $x_{min}^{(s)} := \min_{x_k \in X} \{x_k^{(s)}\}$;
    $x_{max}^{(s)} := \max_{x_k \in X} \{x_k^{(s)}\}$;
    $v_1^{(s)} := x_{min}^{(s)} + \frac{x_{max}^{(s)} - x_{min}^{(s)}}{2 \cdot c_s}$;
    for all $i \in \{2, \dots, c_s\}$
      $v_i^{(s)} := v_{i-1}^{(s)} + \frac{x_{max}^{(s)} - x_{min}^{(s)}}{c_s}$;
    $v_0^{(s)} := x_{min}^{(s)} - \frac{x_{max}^{(s)} - x_{min}^{(s)}}{2 \cdot c_s}$;
    $v_{i+1}^{(s)} := x_{max}^{(s)} + \frac{x_{max}^{(s)} - x_{min}^{(s)}}{2 \cdot c_s}$;
  }
  CalculateMembershipDegrees();
}
```

Algorithm 9.2 (Grid Clustering – continued)

```
Neighbour($v_i^{(s)}$)
{
  $result := \{x_k^{(s)} | (v_i^{(s)} > x_k^{(s)} > v_{i-1}^{(s)}) \vee (v_i^{(s)} < x_k^{(s)} < v_{i+1}^{(s)})\};$
}

CalculateMembershipDegrees()
{
    for all  $k \in \{1, \ldots, n\}, s \in \{1, \ldots, p\}, i \in \{1, \ldots, c_s\}$
    {
      if $x_k^{(s)} \in$ Neighbour($v_i^{(s)}$)
      {
        if $x_k^{(s)} \in$ Neighbour($v_{i-1}^{(s)}$)
          $u_{iks}^{(new)} := 1 - \frac{|v_i^{(s)} - x_k^{(s)}|}{|v_i^{(s)} - v_{i-1}k^{(s)}|};$
        else
          $u_{iks}^{(new)} := 1 - \frac{|v_i^{(s)} - x_k^{(s)}|}{|v_{i+1}^{(s)} - v_i k^{(s)}|};$
      }
      else
        $u_{iks}^{(new)} := 0;$
    }
}

CaculatePartition()
{
  do
  {
    for all  $k \in \{1, \ldots, n\}, s \in \{1, \ldots, p\}, i \in \{1, \ldots, c_s\}$
      $u_{iks}^{(old)} := u_{iks}^{(new)};$
    CalculateMembershipDegrees();
    for all  $s \in \{1, \ldots, p\}, i \in \{1, \ldots, c_s\}$
      $v_i^{(s)} := \frac{\sum_{k=1}^{n} (u_{iks}^{(new)})^m \cdot x_k^{(s)}}{\sum_{k=1}^{n} (u_{iks}^{(new)})^m};$
  }while $\left( \left( \sum_{k=1}^{n} \sum_{s=1}^{p} \sum_{i=1}^{c_s} |u_{iks}^{(new)} - u_{iks}^{(old)}| \right) < \epsilon \right);$
}
```

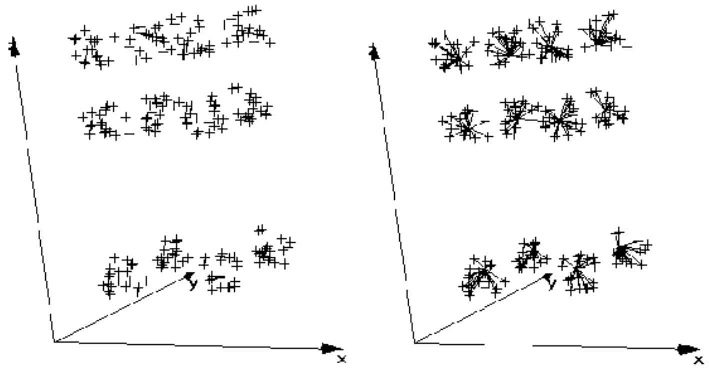


Figure 9.8: An example of grid clustering

Figure 9.8 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 this case, the membership degree of a 3-dimensional data point has been computed 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 objective function based algorithm, but provides clusters with cluster centres on the grid that are very well suited for rule induction for classification tasks as well as for function approximation. 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. Only non-empty clusters are allowed to induce a fuzzy rule. In opposition to the usual probabilistic, possibilistic, or noise 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 - another advantage of this grid clustering algorithm. Although the number of grid points was fixed for the example, it is possible to determine the number automatically on the basis of suitable validity measures as they are described in section 4 or [15, 42, 51, 107].

The number of grid points is determined applying the chosen validity measure to the whole classification and optimising the number of grid points dimension-wise. The unsupervised clustering algorithm used for the grid clustering algorithm is denoted in algorithm 9.3. In the shown way, the optimal number of coordinates for each domain is heuristically estimated. For exact calculation of the optimal number of clusters, the grid clustering algorithm would have to be evaluated for all possible combinations of grid coordinates. This would lead to $\prod_{s=1}^{p}(c_s^{max} - c_s^{min})$ times of starting the grid clustering algorithm and validity evaluation. Here, c_s^{max} (c_s^{min}) is the chosen upper (lower) bound for the number of grid coordinates in domain s. The shown heuristic method lead to satisfying results in several experiments.

Algorithm 9.3 (Unsupervised Grid Clustering)

```
Choose()
{
  $\mathcal{A}_{Grid}$ :: Choose();
  for all   $s \in \{1, \ldots, p\} : c_s^{max} \in \mathbb{N}_{\geq 2}$ where $\prod_{s=1}^{p} c_s^{max} < n$
  validity measure $\mathcal{V}_*$;
}

Initialise()
{
  $c := 1$;
  for all   $s \in \{1, \ldots, p\}$
  {
      $c_s \in \{2, \ldots, c_s^{max} - 1\}$;
      $c := c \cdot c_s$;   $c_s^{best} := c_s$;
  }
  $c_{best} := c$;
  $\mathcal{A}_{Grid}$ :: Initialise();
}

Calculate()
{
  for 2 times, for all $s \in \{1, \ldots, p\}$
    do
    {
      $\mathcal{A}_{Grid}$ :: CalculatePartition();
      $\mathcal{V}_*(c)$;
      if $\mathcal{V}_*(c)$ better than $\mathcal{V}_*(c_{best})$
      {
        $c_{best} := c$;   $c_s^{best} := c_s$;
        $c := \frac{c}{c_s} \cdot (c_s + 1)$;   $c_s := c_s + 1$;
      }
    }while($(c_s \leq c_s^{max}) \wedge (c \leq c_{max})$);
    $c := \frac{c}{c_s} \cdot c_s^{best}$;
    $c_s := c_s^{best}$;
}
```

In case of classification tasks a possible validity measure is the error rate, the percentage of wrong classified data. Otherwise the further described measures from section 4 can be used.

As described in section 4.1.1, an assignment of the data set to predefined classes is given and only the input variables have to be taken into account for the clustering algorithm in case of classification tasks. Each cluster is assigned to the class of the datum with the highest membership degree to this cluster. It is possible that more than one cluster belongs to the same class.

In most cases the projections of the clusters lead to a lot of similar overlapping fuzzy sets for each input variable. For each cluster p fuzzy sets are constructed, if p is the number of dimensions of the data. The grid clustering algorithm avoids this problem that often leads to non-interpretable fuzzy rules. During the classification this algorithm assumes a fuzzy partition of the variables and constructs clusters depending on the fuzzy partitions. So only a few fuzzy sets are needed for the variables and are used in more than one fuzzy rule. Although FCM, GK, or GG usually need less clusters to satisfy the validity criterion, the problem with a high number of fuzzy sets for each single partition still remains. This will lead to difficulties in interpreting the resulting fuzzy rules. The grid clustering algorithm helps to generate rule systems that are easier to understand and interpret by experts, although this method relies not on an objective function.

Chapter 10

Transfer Passenger Analysis

To know the amount of transfer passengers in an aircraft in time would enable the air traffic controller to adapt the sequence of arriving aircraft accordingly. This way, a belated flight with a high amount of passengers that have to reach a connecting flight can be preferred for landing. Real-time assistant systems take a long time to be developed and introduced at airports. Additionally, personal passenger data is confidential available for the airline but usually not for air traffic controllers. Therefore, developing a generalised rule system to describe and identify transfer passenger amounts under certain conditions can be used in combination with simulation models to show the effects of changes in the airports landside on the one hand and enable airside simulations to evaluate the performance of passenger adapted sequencing techniques on the other.

The aim of this work is to illustrate how a rule set can be developed that describes the passenger flow (amount of departing or arriving transfer passengers) in dependence on the flight time, flight range (long-haul, medium-haul, short-haul), and passenger amount. Such a rule set will be used in further studies to simulate the effect of changes in the airports landside or airside architecture or in the landside traffic connections of the airport. The system helps also to identify the interface between land- and airside if additional information – e.g. total number of passengers in relation to transfer passengers – is available. The aim is to combine a rule system with a passenger flow model developed in [91]. The analysis described in this chapter is based on data recorded at Frankfurt airport in 1999. In this work, the applicability of fuzzy clustering techniques is shown. For confidentially reasons we restrict ourselves for analysis purposes to a small amount of data and do not include the original data material in this work.

10.1 Analysis of Flight Information

The transfer passenger amount is especially interesting for a combination of long- and short-haul flights at one airport. Long-distance flights usually start from so-called hub airports. Airlines use a few airports for their long-haul flights that are operated by large aircraft carrying a high amount of passengers, e.g. Airbus 340 or Boeing 747. Complementing spoke airports are usually used to take passengers to their connecting flights. For additional information on hub and spoke systems see e.g. [6] or [38].

Frankfurt airport is one of two hub airports used by Lufthansa. Additionally, Lufthansa is the only carrier using Frankfurt as a main hub airport. For the analysis we restrict ourselves to Lufthansa passenger flights departing (resp. arriving) on weekends in July at Frankfurt airport. For analysis purposes we use the maximal amount of passengers that can be carried by a certain aircraft type. The actual amount of passengers is confidential airline data. For booked up flights both amounts are the same. Usually, flights are nearly booked up during summer. After a plausibility check of the available data and deleting implausible data sets, e.g. flights where the number of transfer passengers is higher than the total number of passengers, an amount of about 900 for each, departures and arrivals, remains for analysis purposes. Errors in the available data are common because part of the data is recorded manually. Typing errors are unavoidable.

For all flights the aircraft type is available. From the literature, e.g. [74], [96], or [86], we derive the maximum number of passengers that can be seated in a certain type of aircraft. The values used here are based on the seating configuration usually used by Lufthansa.

In a pre-processing step the categorical data has to be transformed into real-valued data. To meet the mathematical needs of fuzzy clustering techniques, the distance of the resulting attribute values should reflect the similarity between the categorical values.

A categorical attribute used here is the distance information for flights. Usually the destination or origin airport is known. At Frankfurt airport all destinations and origins are separated in three categories: short-haul, medium-haul, and long-haul. The definition of this categories as described in [85] is used to transfer the corresponding data into real values, see table 10.1.

Table 10.1: Distance Classification

Code	Category	Distance [NM]
1	short-haul	< 1000
2	medium-haul	< 3000
3	long-haul	≥ 3000

To simplify the transformation from resulting real values after a fuzzy classification has been calculated, the departure (arrival) time given in $hh : mm$ is transformed to real values in the following way: $dep_{time} = hh + \frac{mm}{60}$.

The result is the percentage of transfer passengers. This attribute is calculated from the actual amount of passengers in an aircraft and the number of transfer passengers, given in the sample data. This way the direct use of confidential passenger information is avoided.

To use a fuzzy clustering algorithm based on a Euclidean distance measure, the attribute ranges have to be comparable. In the test data we have four attributes with the following ranges: maximal amount of passengers – 36 to 410 (103 to 410 for arrivals), range of destination or origin – 1 to 3, departure time – 6:45 a.m. to 10:40 p.m., i.e. 6.75 to 22.67, (arrival time 5:20 a.m. to 11:30 p.m., i.e. 5.33 to 23.5), and % transfer passengers – 0 to 100. The attributes are transformed into more comparable ranges in the sense of the Euclidean distance. The maximal amount of passengers is divided by 10 leading to values between 3.6 and 41 (resp. 10.3 to 41). The departure time is transformed into real values as described above, the destination (resp. origin) range is transformed to the values 10, 20, and 30. The percentage of transfer passengers is divided by 3 leading to values in the range 0.0 to 33.33. Examples of the resulting departure dataset are shown in table 10.2.

Table 10.2: Transfer Passenger Dataset for Departures

Max Amount of Pax	Range of Dest.	Dep. Time	% Transfer Pax
18.2	30	13.67	21.10
12.3	20	17.67	30.11
...	...	...	...
39.5	30	13.42	28.63
22.2	10	13.42	9.80
3.6	10	16.17	22.99

10.2 Fuzzy Clustering Results

FCM with probabilistic as well as outlier basic objective functions, see chapter 3, and the size adaptive FCM, see section 6.1, have been chosen as clustering techniques for this example. A short test with the axes-parallel GK, see section 3.2.2, has shown, that the resulting clusters predominantly separate the categorical data of the destination or origin range. The resulting cluster shapes have the form of long ellipsoidal structures. Categorical attributes are a problem for form-adaptable clustering techniques because the distance of non-identical neighbouring data points is relatively large compared to other attributes. In case of FCM, the Euclidean distance is used. Here the distance is not adapted for each attribute as in case of the axes-parallel GK. Non-axes-parallel form adaptable techniques have not been tested because the goal of this task is to generate a fuzzy rule system. Generating fuzzy rules from solid fuzzy clusters leads to a loss of information that increases dramatically if non-axes-parallel techniques are chosen, see section 9.1.

The outlier basic function has been chosen in order to enable the algorithm to cope with rare combinations of attributes. The combination of very large aircraft and very few transfer passengers might occur but will happen only for special occasions. The outlier constraint parameter ω was set to 1 and as outlier exponent $t = 0.5$ was chosen. The size adaptation of single clusters can help to reduce the number of necessary rules. Each cluster leads to one fuzzy rule. More specialised small clusters might be necessary for reliable results.

In case of the FCM as described in section 3.2.1, all clusters have the same diameter or size. We use the size-adaptable FCM as developed in 6.1 in order to be able to handle specialised small clusters together with larger clusters that would else be split into different clusters. The parameters additionally needed for FCM-sized have been set as follows: size exponent parameter $l = 0.5$ and size constraint $\tau = 1$. As fuzzifier $m = 1.8$ was selected in all cases.

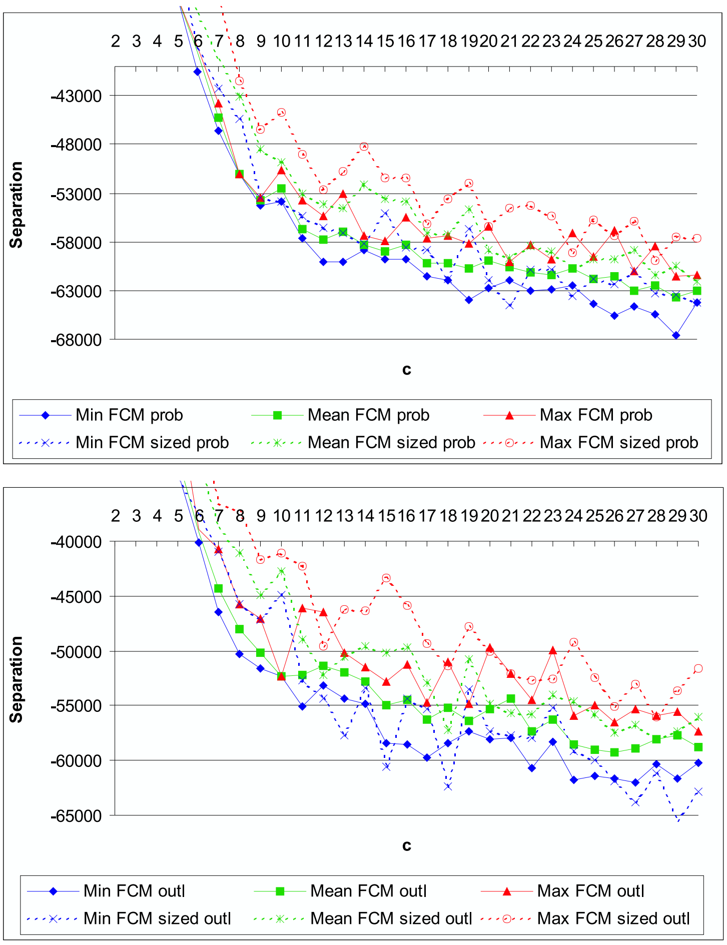


Figure 10.1: Validity for departure clustering results with FCM

A measure related to the separation measure described in section 4.1.4 has been chosen as validity criterion: $\sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik}^{m} \cdot (\mathcal{D}_*(v_i, x_k) - \mathcal{D}_*(v_i, x_\oslash))$. Here, $x_\oslash$ is a vector consisting of the average data values for each attribute. This criterion takes the membership degrees as well as the distance measure into account. This way the size influence factor used for the size-adaptable FCM is taken into account. The separation measure values distances inside the clusters – weighted with the membership degrees – in dependence on the minimal distance between two cluster centres and the number of clusters. Validity criterions based on the form-describing covariance matrix have been especially developed for algorithms as the GK and GG using this matrix for cluster calculations. To use these measures for FCM, the covariance matrix

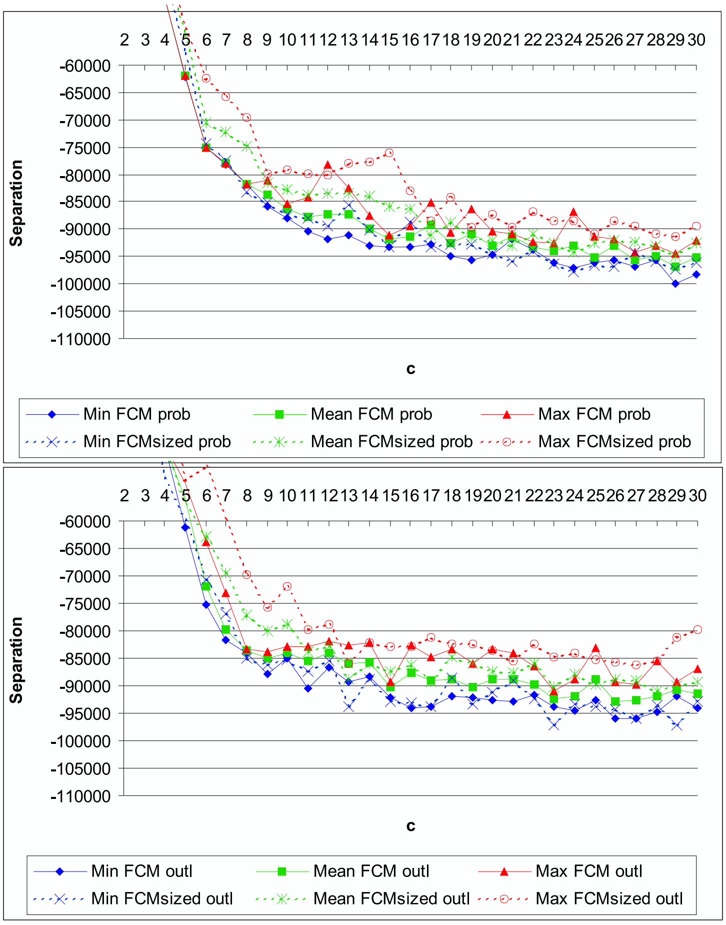


Figure 10.2: Validity for arrival clustering results with FCM

would have to be evaluated for validity purposes only. The resulting separation for departures is shown in figure 10.1 for FCM and FCM-sized based on probabilistic as well as outlier basic clustering. The validity for arrivals is illustrated in figure 10.2. Each cluster partition has been calculated five times for the four combinations FCM probabilistic, FCM outlier, FCM-sized probabilistic, and FCM-sized outlier. In figures 10.1 and 10.2 the maximum, mean, and minimum values of the separation are shown for 5 to 30 clusters.

In general the deviation between the maximum, mean, and minimum separation values is smaller in case of the probabilistic FCM than for the outlier FCM. In the beginning, from 2 to about 9 clusters, the separation measures decreases very strong. The tendency is the same for more clusters but local minima of the validity function can be identified. Most validity measures tend to depend on the number of clusters, although the cluster number is considered in the calculation instruction of the criterion. In the separation measure the number of clusters is taken into account as divisor in the corresponding equation. Therefore, the separation decreases with increasing number of clusters. Local optima of the validity function indicate good results. We look especially for cluster numbers with a relatively high deviation of the validity criterion

Table 10.3: Transfer passenger scale values for departure data

cluster no.	scale value
0	0.0016
1	0.0036
2	0.0069
3	0.0104
4	0.0143
5	0.0200
6	0.0247
7	0.0302
8	0.0385
9	0.0459
10	0.0536
11	0.0636
12	0.0767
13	0.0894
14	0.1048
15	0.1224
16	0.1386
17	0.1549

from criterion results of neighbouring cluster numbers. Here we choose the results for FCM-sized in combination with the outlier basic objective function and 18 clusters for departures and 13 clusters for arrivals. In table 10.3 (resp. 10.4) the scale values for FCM-sized based on outlier clustering for departure (resp. arrival) data and 18 (resp. 13) clusters are shown. We can see that the scales for the single clusters are in a range between 0.0016 and 0.1549 for departures and 0.0033 to 0.1998 for arrivals. The first clusters have a smaller extension than the last ones, since a cluster's extension corresponds to that cluster's scale value. The resulting fuzzy sets reflect the extensions of the clusters in the single domains.

Tables 10.5 and 10.6 show the average maximal membership degrees for a partition of the passenger transfer data into 18 (resp. 13) clusters for both, FCM-sized based on probabilistic and outlier clustering. Considering the membership degrees of one data vector to all clusters the maximal membership degree defines to which cluster a data vector is associated. As we see in tables 10.5 and 10.6 the average maximal degree for all data vectors is about the same for probabilistic as well as possibilistic clustering. For outlier clustering we have a higher amount of data that has a maximal membership degree of less than 0.15 for departures, resp. 0.2 for arrivals. Data vectors with very small maximal membership degrees can be considered as outliers.

The weights for FCM sized based on outlier clustering are in a range between 0.000005 as minimal weight and 0.0045 as maximal weight for departures and 0.000014 as minimal and 0.0034 as maximal weight for arrivals. Because

Table 10.4: Transfer passenger scale values for arrival data

cluster no.	scale value
0	0.0033
1	0.0074
2	0.0156
3	0.0241
4	0.0350
5	0.0499
6	0.0644
7	0.0776
8	0.0951
9	0.1204
10	0.1389
11	0.1686
12	0.1998

Table 10.5: Average maximal membership degree and amount of data for different maximal membership degrees – departure data and 18 clusters

	FCM-sized probabilistic	FCM-sized outlier
Avg. max. memb. degree	0.55	0.53
data with max degree > 0.75	249	219
data with max degree > 0.5	485	439
data with max degree > 0.25	752	745
data with max degree < 0.15	8	13

1 was chosen for τ – the outlier constraint parameter – the sum of all data weights gives 1 as result. Therefore, the single weights for each of the 888 data vectors (resp. 909 for arrivals) is a small real value. The difference between the minimal and maximal weight indicate that weighting occurred for our data set. If all data vectors are assigned the same weight, they would receive a weight value of about 0.001. Here, about 550 data vectors of both data sets – arrivals and departures – have a weight smaller than 0.001 and about 15 data vectors are assigned a weight smaller than 0.0001. Again, data vectors with such small weights can be considered as outliers.

10.3 Fuzzy Rule Generation

Fuzzy sets for the single partitions are derived from the clustering result with FCM-sized based on the outlier objective function for 18 (departures) resp. 13 (arrivals) clusters. Additionally, the resulting fuzzy rules for FCM based on outlier clustering for 22 (departures) resp. 15 (arrivals) clusters are developed.

Table 10.6: Average maximal membership degree and amount of data for different maximal membership degrees – arrival data and 13 clusters

	FCM-sized probabilistic	FCM-sized outlier
Avg. max. memb. degree	0.61	0.63
data with max degree > 0.75	321	349
data with max degree > 0.5	583	575
data with max degree > 0.25	855	843
data with max degree < 0.2	4	20

First the membership degrees for one cluster are projected onto the single domains. Then the convex hull over the membership degrees is build for each domain. In the last step, a trapezoidal fuzzy set approximating the convex hull is calculated as described in section 9.1. The resulting fuzzy sets are illustrated in figure 10.3 for FCM sized based on outlier clustering and the departure data set. In figure 10.4 the resulting fuzzy sets for the same clustering technique applied to arrival data and 13 clusters is shown. Since each fuzzy cluster leads to one fuzzy rule, we obtain 18 fuzzy sets for each attribute for departures, resp. 13 sets for arrivals – one for each rule.

This form of rule generation leads to a rule base with overlapping and overlaying fuzzy sets for the single attributes. Such a rule base is extremely difficult to interpret. Therefore we use the attempt proposed in [103] to simplify the extracted rule base. In an iterative procedure, similar fuzzy sets for each attribute are identified, combined, and the rule base is updated. Afterwards, fuzzy sets similar to the universal fuzzy set are neglected from the rule base. If all data points of the universe of discourse have the membership degree one to a fuzzy set this is called universal fuzzy set. In the last step of rule reduction rules with the same input sets are combined. Contradictory rules might occur in this step. If contradictory rules occur, the corresponding data has to be analysed in more detail. One reason might be additional influence factors that have not been taken into account. However, in our transfer passenger example we did not have to cope with contradictory rules. As similarity measure for fuzzy sets we used a measure based on the set-theoretic operations of intersection and union, see equation (10.1).

$$Sim(\mu_A, \mu_B) = \frac{\sum_x Min(\mu_A(x), \mu_B(x))}{\sum_x Max(\mu_A(x), \mu_B(x))} \tag{10.1}$$

Here, μ_A and μ_B are the fuzzy sets for which the similarity has to be determined. x denotes the corresponding attribute values of the sample data. Two fuzzy sets are combined if $Sim(\mu_A, \mu_B) \geq 0.5$. A fuzzy set is identified as universal fuzzy set $\mu_{universal}$ if $Sim(\mu_A, \mu_{universal}) \geq 0.8$.

The resulting fuzzy sets for the single attributes are illustrated in figures 10.5 for departures and 10.6 for arrivals. Tables 10.7 and 10.8 show the resulting fuzzy rule bases. Especially for the attributes *departure resp. arrival* *time* and *% transfer passenger* the number of fuzzy sets has been reduced to about a third of the original fuzzy sets. Also for the other two attributes in our example only about half the number of fuzzy sets remains. The separation of the input and output domains into fuzzy sets after simplification allows an interpretation of the rule base that can be discussed with experts.

Table 10.7: Fuzzy rules for departure data and FCM sized outlier – combined fuzzy sets – for corresponding fuzzy sets see fig. 10.5

Rule	Max. no. of Pax	Destination	Departure Time	% Transfer Pax
1	Pax Max 1	Range 1	Time 1	Transfer Pax 1
2	Pax Max 2	Range 1	Time 2	Transfer Pax 2
3	Pax Max 3	Range 1	Time 3	Transfer Pax 3
4	Pax Max 4	Range 1	Time 4	Transfer Pax 4
5	Pax Max 5	Range 5	Time 1	Transfer Pax 5
6	Pax Max 6	Range 1	Time 3	Transfer Pax 4
7	Pax Max 7	Range 7	Time 5	Transfer Pax 4
8	Pax Max 8	Range 8	Time 5	Transfer Pax 6
9	Pax Max 5	Range 9	Time 6	Transfer Pax 5
10	Pax Max 5	Range 5	Time 6	Transfer Pax 7
11	Pax Max 9	Range 7	Time 6	Transfer Pax 4
12	Pa Max 10	Range 7	Time 5	Transfer Pax 4
13	Pax Max 5	Range 11	Time 6	Transfer Pax 4
14	Pax Max 5	Range 11	Time 5	Transfer Pax 4
15	Pax Max 5	Range 8	Time 5	Transfer Pax 4
16	Pax Max 5	Range 11	Time 5	Transfer Pax 6
17	Pax Max 5	Range 8	Time 6	Transfer Pax 4
18	Pax Max 5	Range 11	Time 6	Transfer Pax 5

Let us give an explanation how the rule base for departures generated from FCM-sized clustering based on the outlier objective function can be interpreted:

- Rules 1 and 5 can be interpreted as follows: Aircraft with a relatively small number of maximal passengers (80 to 200), a short- to medium-haul destination, and departing late at night (about 9.00 p.m.) usually have a high amount (about 80 to 90%) of transfer passengers.

- Rules 2 tells us, that medium-haul flights with relatively small aircraft (about 150) starting about noon carry a large number of passengers who arrived by plane in Frankfurt (about 70%).

- About 50 to 80% transfer passengers are in smaller aircraft (about 120 passengers) used for medium-haul flights starting about noon as we can see in rules 3 and 6.

- Small to medium size aircraft with a destination in short to medium and large aircraft with medium- or long-haul distance departing in the

Table 10.8: Fuzzy rules for arrival data and FCM sized outlier – combined fuzzy sets – for corresponding fuzzy sets see fig. 10.6

Rule	Max. no. of Pax	Origin	Arrival Time	% Transfer Pax
1	Pax Max 1	Range 1	Time 1	Transfer Pax 1
2	Pax Max 2	Range 2	Time 2	Transfer Pax 2
3	Pax Max 3	Range 3	Time 2	Transfer Pax 3
4	Pax Max 4	Range 4	Time 3	Transfer Pax 4
5	Pax Max 5	Range 3	Time 2	Transfer Pax 3
6	Pax Max 5	Range 3	Time 2	Transfer Pax 4
7	Pax Max 4	Range 5	Time 4	Transfer Pax 4
8	Pax Max 6	Range 6	Time 4	Transfer Pax 4
9	Pax Max 5	Range 3	Time 2	Transfer Pax 5
10	Pax Max 5	Range 3	Time 2	Transfer Pax 4
11	Pax Max 5	Range 3	Time 2	Transfer Pax 6
12	Pax Max 5	Range 7	Time 4	Transfer Pax 4
13	Pax Max 5	Range 8	Time 4	Transfer Pax 4

afternoon carry about 50 to 80% transfer passengers, see rules 9, 10, 11, 13, 17, and 18.

- An amount of 50 to 80% transfer passengers is carried in aircraft with 150 to 200 passengers departing in the morning hours to a medium-haul destination, see rule 4.

In general we see that all rules with result *Transfer Pax* 4 are relatively indifferent. In this case additional attributes might be of interest as e.g. a more detailed separation of the destinations. The last 6 rules with *Pax Max* 5 in the premise result from the clusters with larger scale values. The less specified fuzzy sets corresponding to these rules reflect the greater cluster extension indicated by the scale values.

Similar to the departure rule base the arrival rule base can be interpreted as follows:

- Rule 1 denotes that relatively large aircraft (about 250 passengers) arriving from a long-haul origin in the morning carry a relatively high amount of transfer passengers (about 70%).

- Few transfer passengers arrive by small aircraft from short- to medium-haul origins in the afternoon, see rule 2.

- Large to very large aircraft arriving in the afternoon bring a relatively large amount of transfer passengers to the airport independent of the origin's distance, see rules 3 and 4.

- Aircraft arriving in the early morning hours carry a large number of transfer passengers. From rules 7, 8, 12, and 13 we see that this is independent of the aircraft size and the origins distance.

Rules 5, 6, and 9 to 11 have the same fuzzy sets in their premises. The resulting fuzzy sets altogether show the indifference of these rules. To separate the amount of transfer passengers in these cases, additional attributes are necessary. Again, the more indifferent rules 9 to 11 have a relatively large extension, see table 10.4.

To estimate the influence of size adaptation in fuzzy clustering, FCM based on outlier clustering without size adaptation has been carried out for 22 (departures) and 15 (arrivals) clusters. These cluster numbers reflect local minima in the validity curves in figures 10.1 and 10.2 for FCM based on the outlier objective function. The single clusters in one partition have the same extension. Therefore, a cluster previously representing a huge multi-dimensional range of sample data might now – without size adaptation – be split into separate clusters. This explains that local optima occur for FCM for a larger cluster number than for FCM-sized based on outlier clustering as well as probabilistic clustering.

Figures 10.7 and 10.8 show the resulting fuzzy sets for FCM outlier and the departure resp. arrival dataset. Again, rule generation leads to overlapping fuzzy sets that are extremely difficult to interpret. The approach described above has been used to reduce the rule bases. Resulting fuzzy sets for the single attributes are illustrated in figures 10.9 for departures and 10.10 for arrivals. Tables 10.9 and 10.10 show the resulting fuzzy rule bases. For the attribute *departure resp. arrival time* the number of fuzzy sets has been reduced to about a fourth of the original fuzzy sets. Also for the other three attributes in the departure example only about a third of the number of fuzzy sets remains. The amount of fuzzy sets for the arrival example is reduced to about half of the fuzzy sets. Again, the simplified separation of the input and output domains into fuzzy sets gives us an interpretable rule base that can be discussed with experts.

Let us give a few examples for departure rule base generated from fuzzy clustering based on the Euclidean distance and the outlier objective function (table 10.9):

- Rules 2, 7, 18, and 19 tell us, that relatively small aircraft with short- to medium-haul destination departing in the late afternoon or night carry a relatively high amount of transfer passengers.
- Small aircraft departing around noon to a medium-haul destination have about 40 to 80% transfer passengers, see rules 1 and 22.
- In contrast, small aircraft departing around noon to a short-haul destination carry only about 0 to 60% transfer passengers, see rules 15 and 16.

Equivalently, the arrival rule base can be interpreted as follows:

- Rules 2, 3, 4, 6, 8, and 9 denote that aircraft arriving the early morning carry a relatively high amount of transfer passengers (about 60 to 90%).

Table 10.9: Fuzzy rules for departure data and FCM outlier – combined fuzzy sets – for corresponding fuzzy sets see fig. 10.9

Rule	Max. no. of Pax	Destination	Departure Time	% Transfer Pax
1	Max Pax 1	Range 1	Time 1	Transfer Pax 1
2	Max Pax 1	Range 1	Time 2	Transfer Pax 2
3	Max Pax 2	Range 2	Time 1	Transfer Pax 1
4	Max Pax 1	Range 1	Time 3	Transfer Pax 3
5	Max Pax 1	Range 3	Time 3	Transfer Pax 4
6	Max Pax 3	Range 3	Time 1	Transfer Pax 1
7	Max Pax 1	Range 4	Time 4	Transfer Pax 1
8	Max Pax 4	Range 3	Time 3	Transfer Pax 1
9	Max Pax 1	Range 5	Time 3	Transfer Pax 1
10	Max Pax 5	Range 6	Time 1	Transfer Pax 1
11	Max Pax 6	Range 2	Time 4	Transfer Pax 1
12	Max Pax 1	Range 4	Time 3	Transfer Pax 1
13	Max Pax 7	Range 5	Time 4	Transfer Pax 1
14	Max Pax 2	Range 2	Time 5	Transfer Pax 1
15	Max Pax 1	Range 7	Time 1	Transfer Pax 5
16	Max Pax 1	Range 4	Time 1	Transfer Pax 3
17	Max Pax 1	Range 5	Time 4	Transfer Pax 3
18	Max Pax 1	Range 7	Time 2	Transfer Pax 2
19	Max Pax 1	Range 4	Time 2	Transfer Pax 1
20	Max Pax 8	Range 5	Time 4	Transfer Pax 3
21	Max Pax 1	Range 1	Time 4	Transfer Pax 1
22	Max Pax 1	Range 5	Time 1	Transfer Pax 3

- Very large aircraft arriving in the afternoon from a medium- to long-haul origin bring a relatively large amount of transfer passengers to the airport, see rule 13.

The interpretation of the more precise rules coincides in both examples (departure and arrival data) for both clustering techniques (FCM based on outlier clustering with and without size adaptation) and reflects our expert knowledge. Some of the identified rules reflect the description in air traffic management literature, see e.g. [6] or [7]. They are typical for the way Lufthansa and other carriers use slots, i.e. time intervals assigned to airlines for departures and arrivals to perform certain flights.

A crosscheck of the resulting rules with data not used for rule generation, e.g. flight data for August, shows the performance of the extracted rules. The derived rules – and first of all the more precise rules – are a proper description of August flight data not only for weekends. Additionally, the rule base meets our needs for the macroscopic passenger movement model in general, see [4] and [12]. We will extend our analysis for additional attributes, e.g. terminal information and more precise destination resp. origin information, and study

Table 10.10: Fuzzy rules for arrival data and FCM outlier – combined fuzzy sets – for corresponding fuzzy sets see fig. 10.10

Rule	Max. no. of Pax	Origin	Arrival Time	% Transfer Pax
1	Pax Max 1	Range 1	Time 1	Transfer Pax 1
2	Pax Max 2	Range 2	Time 2	Transfer Pax 1
3	Pax Max 3	Range 3	Time 2	Transfer Pax 1
4	Pax Max 1	Range 1	Time 2	Transfer Pax 2
5	Pax Max 3	Range 2	Time 3	Transfer Pax 3
6	Pax Max 4	Range 4	Time 2	Transfer Pax 1
7	Pax Max 3	Range 1	Time 3	Transfer Pax 1
8	Pax Max 5	Range 5	Time 2	Transfer Pax 1
9	Pax Max 3	Range 1	Time 2	Transfer Pax 2
10	Pax Max 2	Range 1	Time 3	Transfer Pax 4
11	Pax Max 3	Range 6	Time 3	Transfer Pax 5
12	Pax Max 3	Range 1	Time 3	Transfer Pax 3
13	Pax Max 4	Range 5	Time 1	Transfer Pax 1
14	Pax Max 3	Range 1	Time 3	Transfer Pax 6
15	Pax Max 3	Range 1	Time 3	Transfer Pax 1

the less specific clusters in more detail in future analysis. This way a more precise rule base can be developed.

The grid clustering algorithm described in section 9.2 and [53] might be seen as an alternative to FCM used in this example. This algorithm is designed in order to derive a rule base where all intersections of fuzzy sets are at a membership degree of 0.5. The numerical complexity of the grid clustering algorithm increases exponentially with an increasing number of fuzzy sets for the single domains. A partition in fuzzy sets similar to the results obtained for arrival data and FCM-sized based on the outlier objective function would result in a partition into 1152 clusters – $6 \cdot 8 \cdot 4 \cdot 6$ attributes for the single domains. During rule generation, the ”clusters” describing areas in the multi-dimensional universe of discourse without any sample data can be neglected but this does not reduce the computational effort during clustering. If only a few attributes are considered and only a few fuzzy sets are needed for each domain, grid clustering is a promising technique.

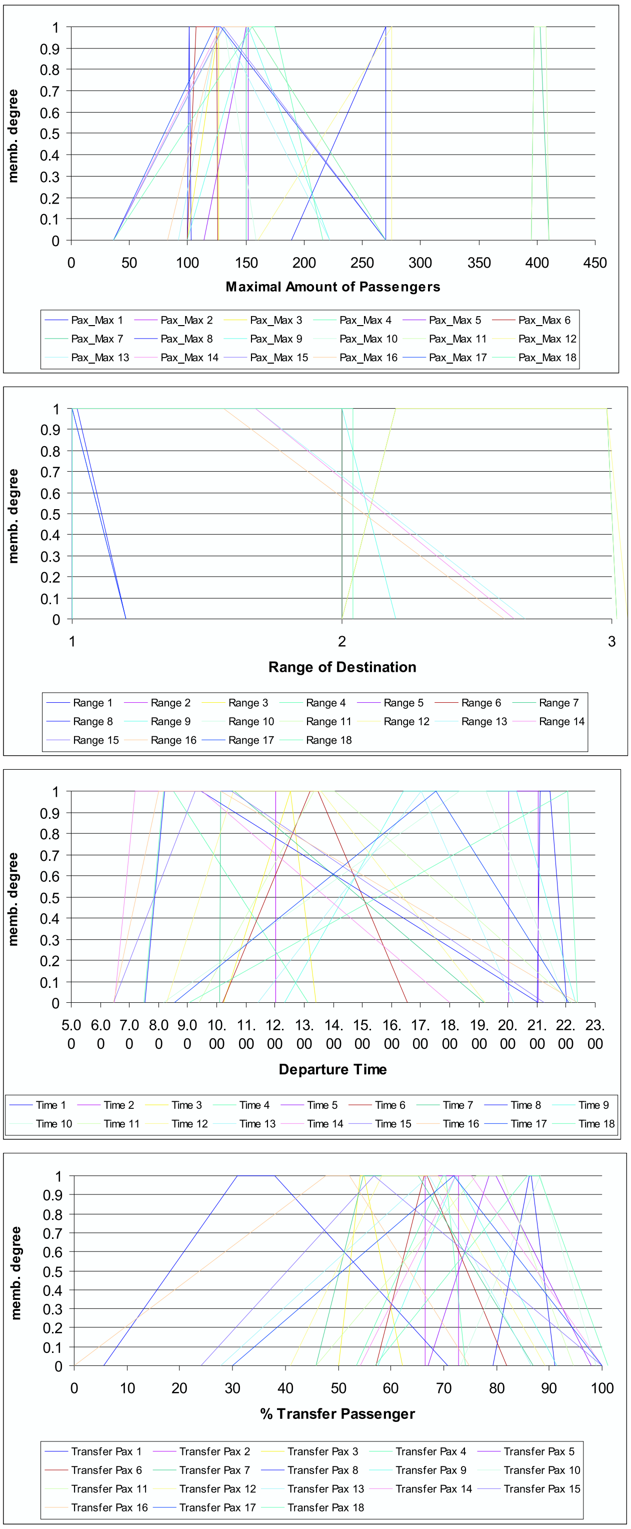


Figure 10.3: Generated fuzzy rules for departure data and FCM sized outlier with 18 clusters

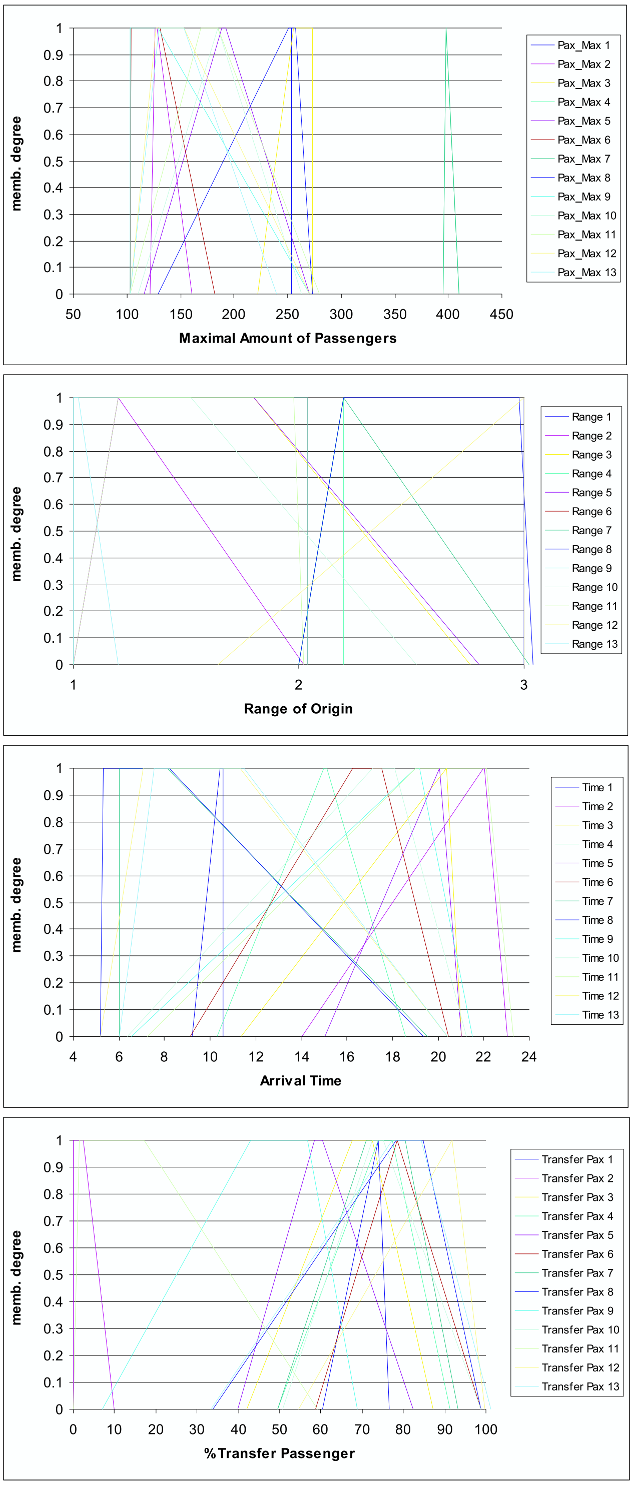


Figure 10.4: Generated fuzzy rules for arrival data and FCM sized outlier with 13 clusters

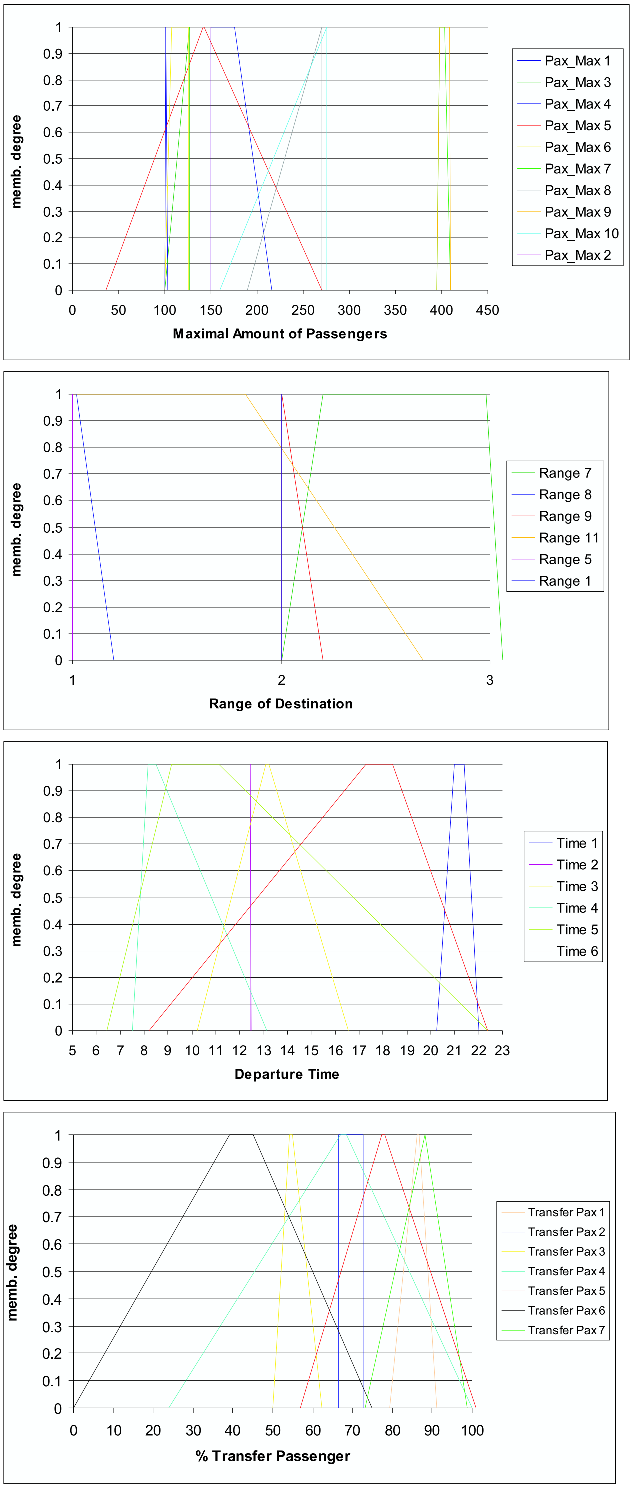


Figure 10.5: Combined and simplified fuzzy rule sets for departure data and FCM sized outlier with 18 clusters

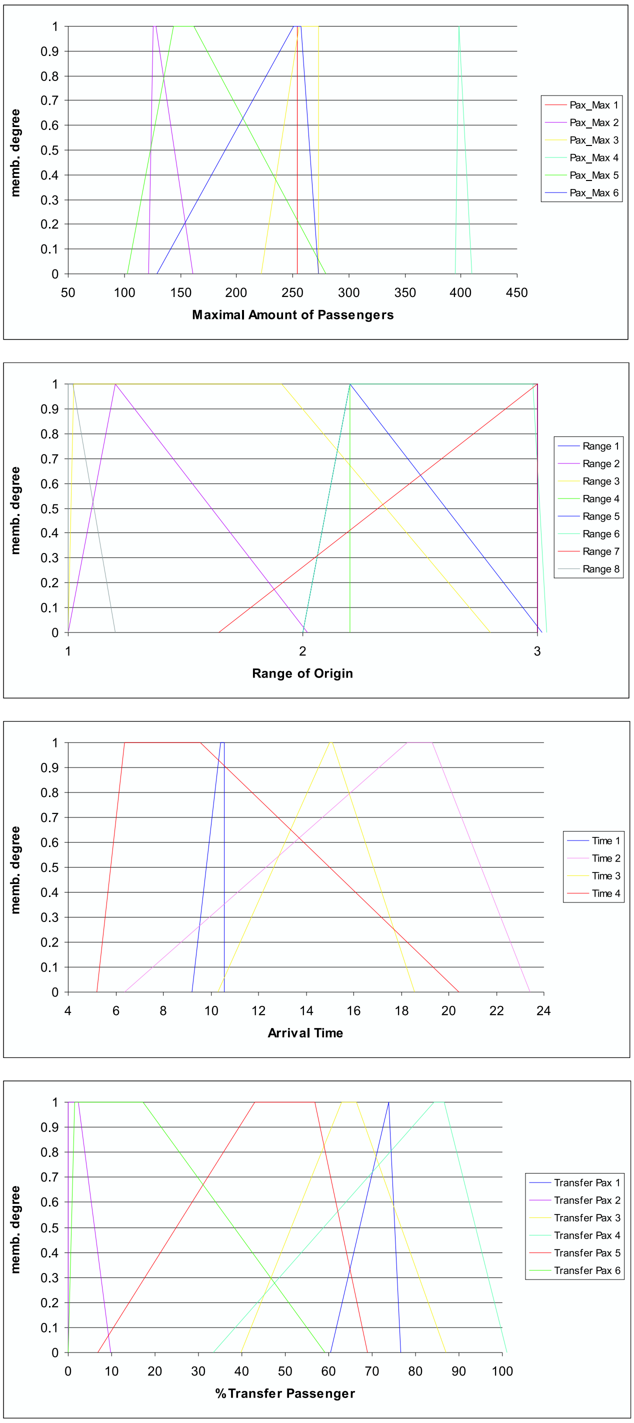


Figure 10.6: Combined and simplified fuzzy rule sets for arrival data and FCM sized outlier with 13 clusters

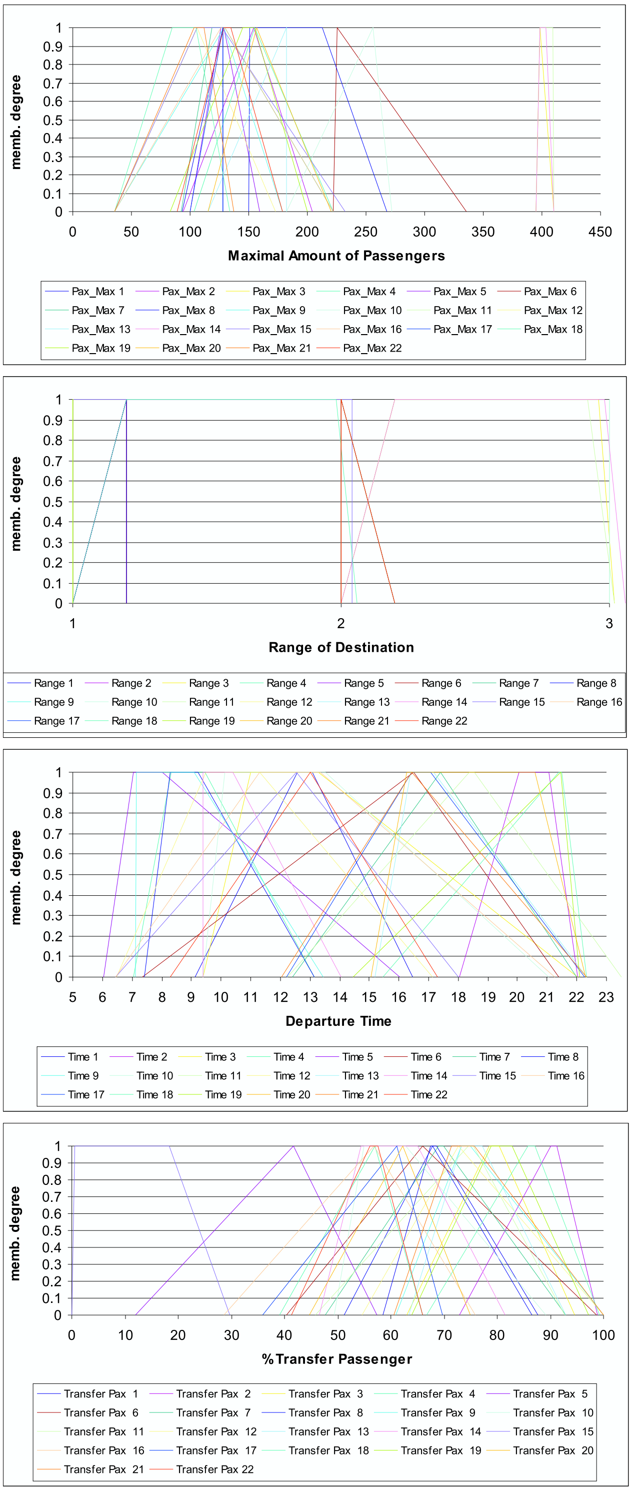


Figure 10.7: Generated fuzzy rules for departure data and FCM outlier with 22 clusters

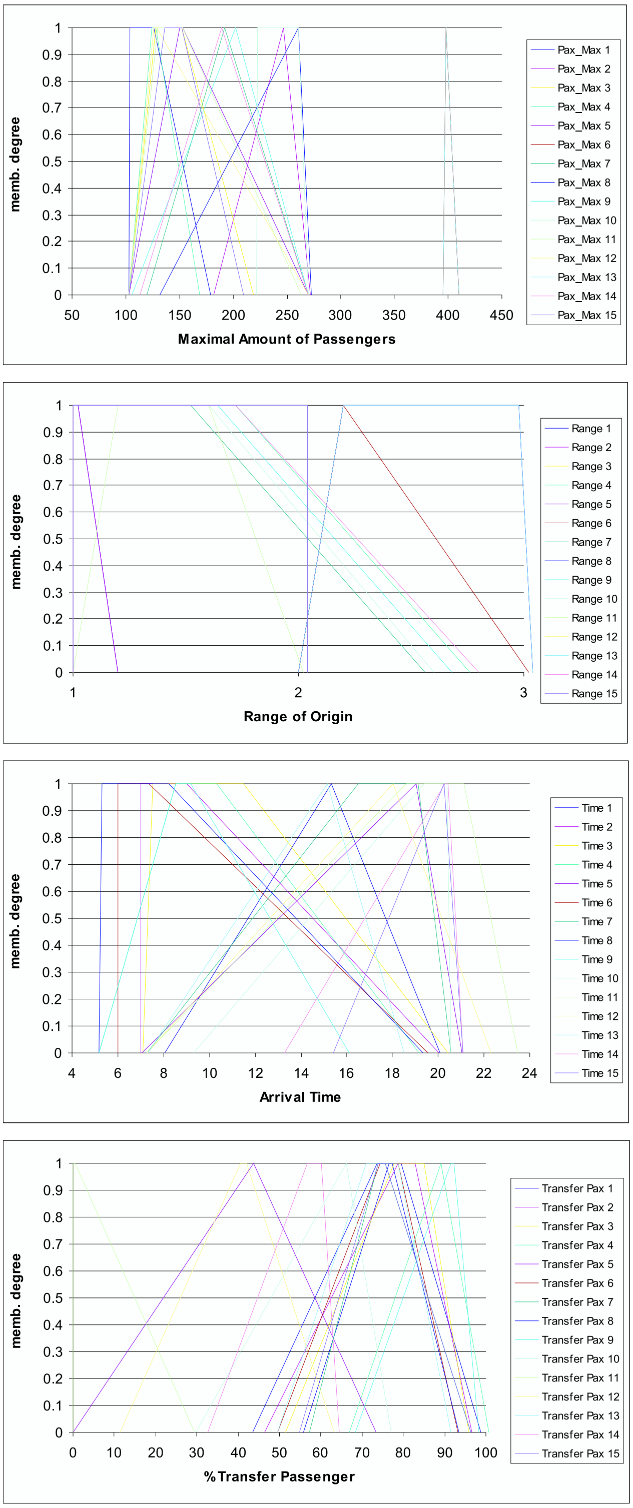


Figure 10.8: Generated fuzzy rules for arrival data and FCM outlier with 15 clusters

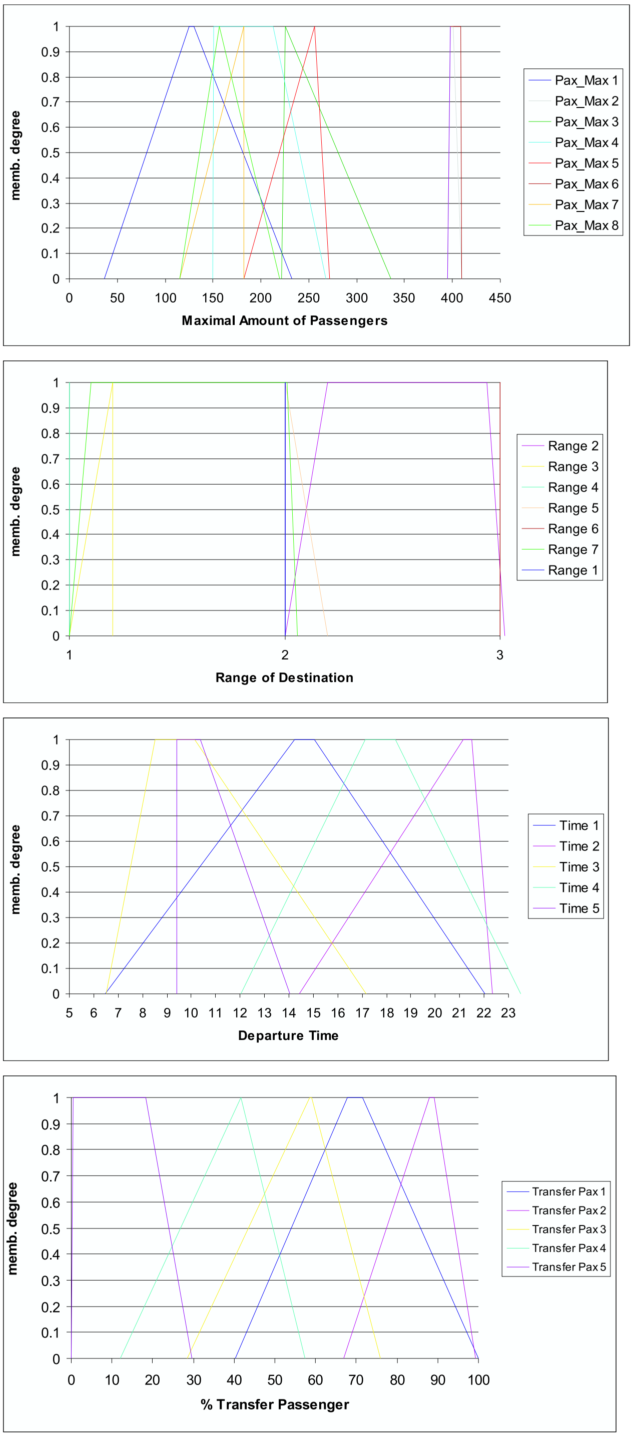


Figure 10.9: Combined and simplified fuzzy rule sets for departure data and FCM outlier with 22 clusters

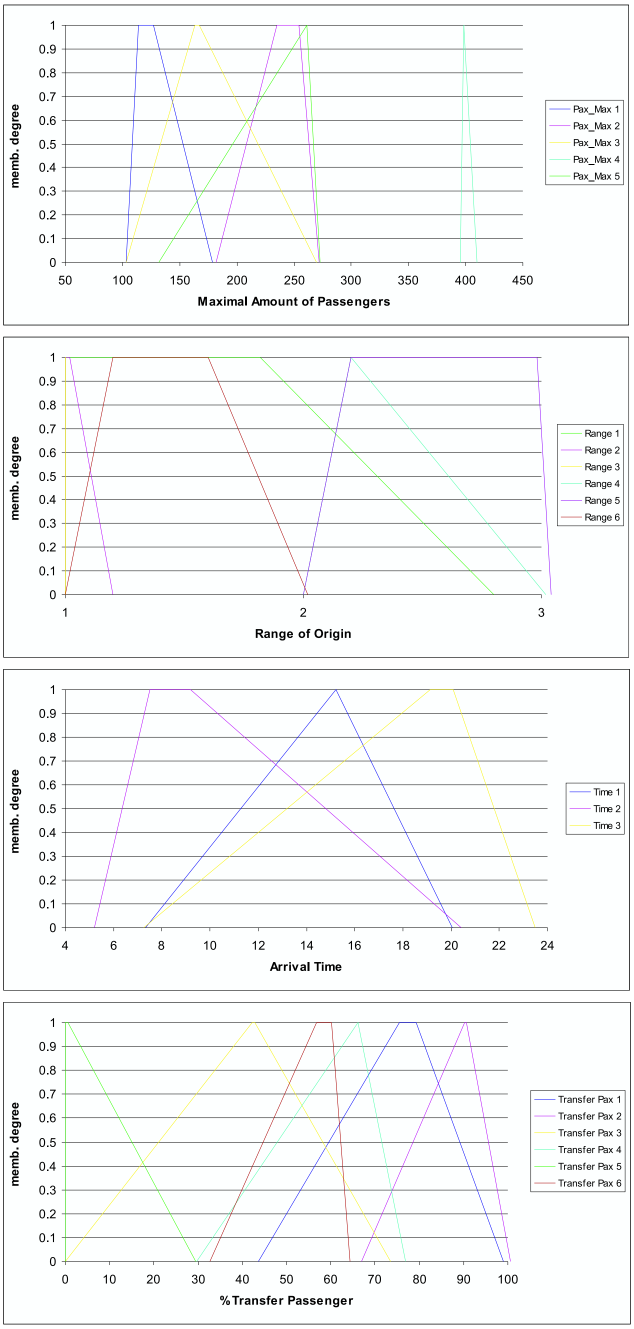


Figure 10.10: Combined and simplified fuzzy rule sets for arrival data and FCM outlier with 15 clusters

Chapter 11

Evolutionary Algorithm-Based Fuzzy Clustering

In the former chapters we have seen that objective function based fuzzy clustering aims at finding a fuzzy partition by optimising a function evaluating a (fuzzy) assignment of a given data set to clusters, that are characterised by a set of parameters, the prototypes. The iterative optimisation technique usually requires the objective function not only to be differentiable, but prefers also an analytical solution for the equations of necessary conditions for local optima. Evolutionary algorithms are known to be an alternative robust optimisation technique which are applicable to quite general forms of objective functions. In this chapter the possibility of making use of evolutionary algorithms in fuzzy clustering is investigated. Experiments and theoretical investigations show that the application of evolutionary algorithms to shell clustering, where the clusters are in the form of geometric contours, is not very promising due to the shape of the objective function, whereas they can be helpful in finding solid clusters that are not smooth, for example rectangles or cubes. These types of clusters play an important role for fuzzy rule extraction from data, as we have seen in chapter 9.

In section 3.1 basic concepts for objective function based fuzzy clustering have been introduced. To derive an alternating optimisation scheme, the first derivative of the objective function with respect to the cluster parameters has to be computed. The resulting necessary conditions for the objective function to have a minimum are then used in an iteration procedure and define a clustering algorithm. On the one hand, this requires the objective function to be differentiable and on the other hand, the iteration procedure can be computationally efficient only if the derived conditions lead to explicit equations for the cluster parameters.

Unfortunately, in many cases the restrictions that are enforced on the cluster parameters to be able to derive the iteration procedure are too narrow and exclude a lot of interesting possibilities. Thus it is desirable to have an alternative optimisation technique that allows for more freedom in the choice of the cluster parameters.

In this chapter evolutionary algorithms are shown to be a possible solution to this problem. After a short introduction on evolutionary strategies, we discuss the principal approach to use evolutionary algorithms for fuzzy clustering (for an overview see [49, 65, 75]).

We will see that shell clustering with evolutionary algorithms seems to be quite problematic, since there exist a lot of local optima and the correct solution often looks like a very narrow optimum. The situation is better for solid clusters. It should however be noted that evolutionary algorithms require a much longer computation time to solve the problem compared to the standard iteration procedure so that the application of evolutionary algorithms seems to be suited only if the standard iteration procedure cannot be applied according to a non-differentiable distance function or when an analytical solution for the single iteration steps cannot be found. Even in that case a good heuristic algorithm can lead to results almost as good as the ones obtained by evolutionary algorithms in a much shorter time, as the comparison with grid clustering shows.

We restrict our investigations to probabilistic clustering, requiring that the membership degrees of a datum to the clusters add up to one. In principal, we can as well use the possibilistic version of the objective function, see section 3.1.2, dropping the probabilistic constraint. Even the noise clustering (section 3.1.3) or clustering with outliers (section 3.1.4) approaches can be used in principle.

Another important question is a strategy for determining the number of clusters. This can be done in the usual way on the basis of suitable validity measures. However, it requires to carry out the clustering for a varying number of clusters and increases the computation time even more.

We obtain the most promising results for clusters suitable for constructing fuzzy rules. Of course, there are other techniques of learning rules, like neuro-fuzzy approaches (for an overview see [93]). However, it turns out that most of these approaches are well suited for tuning the fuzzy sets, but not so for detecting rules.

11.1 Evolutionary Algorithms and Fuzzy Clustering

Evolutionary algorithms are a class of optimisation methods that are inspired by the process of biological evolution. The principal idea is to have a collection or *population* of possible problem solutions encoded as parameter vectors – the *chromosomes* – that define a solution. From this population a new population – the next *generation* – is generated by first producing offspring from the old chromosomes by changing some components of the chromosomes, the *genes*, randomly and sometimes also by a mixing of genes of different chromosomes (*crossover*), and then by selecting the best chromosomes for the next generation. For details we refer to books like [8, 89, 45].

As a quite general optimisation strategy, evolutionary algorithms might be

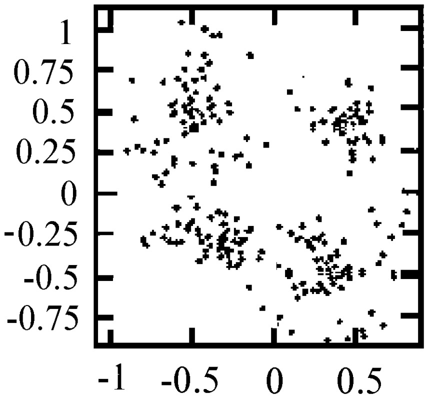
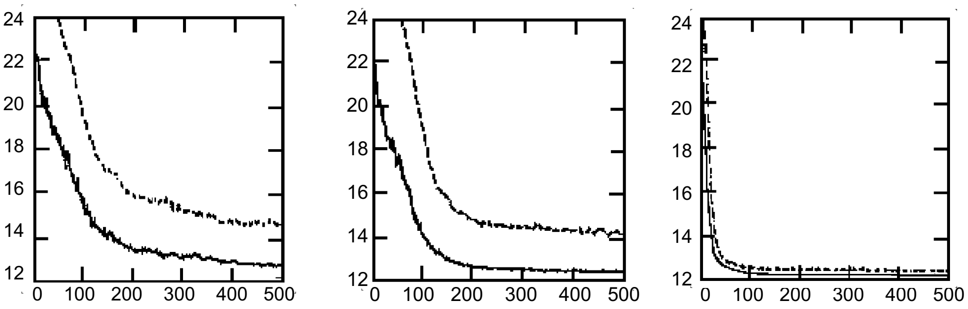


Figure 11.1: A test data set for the FCM

applicable to objective function based fuzzy clustering. Thus it is necessary to find a suitable coding of the parameters to be determined in fuzzy clustering. Obviously, the parameters to be optimised are the prototypes v_i and the membership degrees u_{ik}. In [18] it was proposed to perform hard clustering (i.e. $u_{ik} \in \{0, 1\}$), with genetic algorithms by taking the u_{ik} as the parameters for the evolutionary algorithm. For fuzzy clustering this does not seem to be suitable, since this means that besides the prototypes $c \cdot n$ real-valued parameters have to be optimised, where c is the number of clusters and n the number of data vectors. Since the probabilistic equation for calculation of the u_{ik} from section 3.1 is generally valid independent of the structure of the prototype, it is not necessary to optimise the parameters u_{ik} by an evolutionary algorithm. In addition, the problems for the standard iterative optimisation procedure are not caused by the membership degrees, but by the choice of the prototype parameters. Thus we restrict ourselves here to optimise only the prototypes by an evolutionary algorithm. The corresponding membership degrees are computed as in the usual algorithm on the basis of equation (3.4) from section 3.1.1.

The aim of fuzzy clustering depends on the application domain. In the case of data analysis and classification tasks as well as for rule extraction it is important to find an appropriate (gradual) assignment of the data to suitable prototypes. In this case, the emphasis is on the assignment and not on the exact prototype parameters. On the other hand, the exact prototype parameters might be the terms to extract from a classification as e.g. in shell clustering algorithms. With this group of algorithms, geometrical objects in images are detected. Therefore, it is not sufficient just to assign the data points to the correct circle represented by one cluster, but that we have to determine the centre and the radius of the circle.



(a) Roulette wheel

(b) Remainder stochastic sampling

(c) Tournament

Figure 11.2: Comparison of selection strategies

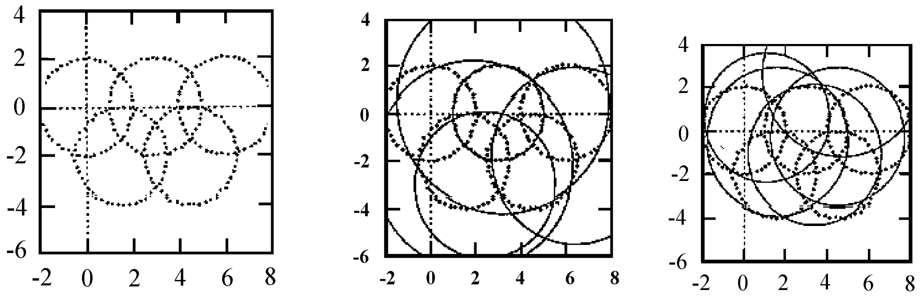
11.2 Experimental and Theoretical Results

Before we apply evolutionary algorithms to fuzzy clustering with non-standard prototypes, we first test their performance by two standard fuzzy clustering techniques, namely FCM as an example for the search after solid clusters and the shell clustering algorithm FCS. Similar experiments were also carried out in [92] for FCM. As in our approach, only the prototypes, i.e. for FCM the cluster centres, are determined by the evolutionary algorithm in [92]. [92] also reports good results when the probabilistic objective function (3.1) from section 3.1.1 is replaced by the partition coefficient, a validity measure that is sometimes used for FCM to determine the number of clusters, see section 4.1.2.

Figure 11.1 shows a test data set for FCM with four clusters. Evolutionary algorithms with different parameters were all able to solve the problem of finding suitable prototypes. A comparison of different selection strategies (left to right: roulette wheel, remainder stochastic sampling, tournament) in Figure 11.2 shows that tournament selection has the best performance. In the figure the dashed line indicates the average fitness and the other line the best fitness in each generation, averaged over 18 runs of the evolutionary algorithm.

Looking at these good results for FCM we were quite optimistic also for shell clustering. However, the results were more or less disappointing. Figure 11.3 shows a test data set with five circles for FCS and two results of the evolutionary algorithm. In one case no circle was detected correctly, in the other only one of the five. These results could not be improved, neither by experimenting with the mutation or crossover rate nor by introducing techniques like controlling the mutation rate on the basis of a span measure [89].

The typical results of the optimisation of the objective function of FCS by an evolutionary algorithm tend to yield larger circles – an observation which was also made in [22] where rectangular shells were considered. Thus we tested the evolutionary algorithm with a data set representing only one circle with centre $(0,0)^\top$ and radius 2. In one case we limited the search space for the



(a) Test data

(b) FCS Result

(c) FCS Result

Figure 11.3: FCS: Test data and results

radius and the coordinates to the interval $[-2.2, 2.2]$, in the other case to the interval $[-22, 22]$. In the first case the evolutionary algorithm computed the correct radius and centre in all test runs after about 30 generations. For the second case the circle was detected correctly only in about 60% of the cases after approximately 125 generations.

This motivated us to take a closer look at the fitness function. We consider again a circle with centre $(0, 0)^\top$ and radius 2 as the data set. In Figure 11.4 the evaluation of a chromosome is shown whose y-coordinate for the circle is the correct value 0, whereas the x-value is shifted to the right between 0 and 22. The different curves were drawn for radius values between 1 and 10. The middle and lower diagram are just magnified scalings of the upper diagram.

From this figure it is obvious that the correct radius 2 gets the best evaluation only as long as the shifted circle centre is not shifted too far away from the original circle centre. The smaller radius 1 does never yield better values than a larger radius. And with increasing distance of the shifted centre to the original centre, a larger radius gets a better evaluation. This implies that in early generations where the random circle centres are still far away from the correct circle centres, chromosomes with smaller radius values are not selected. But when these values are missing in the population, a random mutation to the correct radius without mutating also the centre to the correct value leads to a very bad evaluation. Thus these genes are so strongly dependent on each other that only a simultaneous random jump of all genes to the correct values can lead to good results. Obviously this is rarely possible.

When we applied the evolutionary algorithm again to the data set of Figure 11.1 and limited the search range for radius genes to the interval $[0, 2.2]$, the results were satisfactory. In all test runs the circles were detected correctly after about 80 generations in average.

Our theoretical considerations and experiments show that the applicability of evolutionary algorithms to shell clustering is not very promising except when the parameter range can be restricted to quite limited bounds.

As mentioned in section 11.1, the standard iteration procedure is difficult or impossible not only for certain types of shell clusters, but also for instance

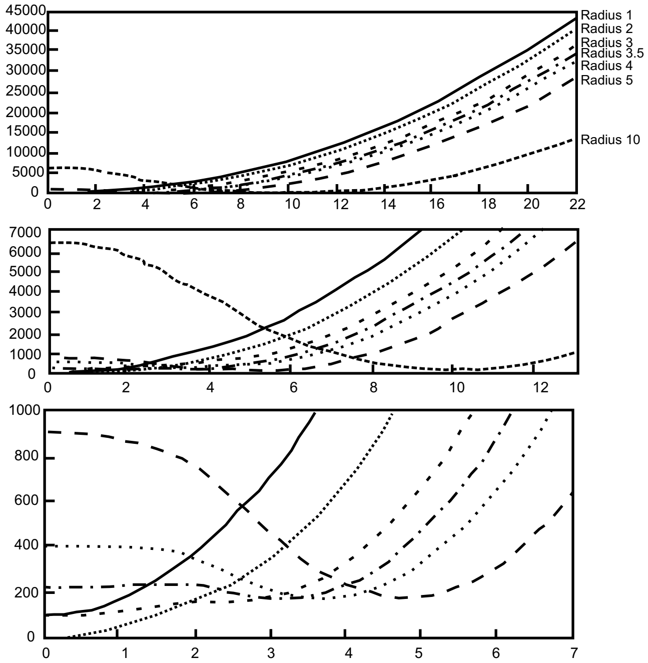


Figure 11.4: Evaluation of chromosomes with shifted circle centres

for solid rectangular clusters. Such clusters would be ideal for fuzzy rule extraction, especially when they are restricted to axes parallel rectangles or cubes.

In order to obtain clusters of this type we define as prototypes for p-dimensional data cluster centres $v_i \in \mathbb{R}^p$ and diagonal matrices A_i. As distance function we choose

$$\begin{aligned} d^2(x_k, \mathsf{v}_i) &= \| A_i(v_i - x_k) \|_\infty^2 \\ &= \left(\max_{1 \le s \le p} \{ a_i^{(s)} \cdot |v_i^{(s)} - x_k^{(s)}| \} \right)^2 . \end{aligned}$$

In opposition to [22] we use the supremum norm $\| \;.\; \|_\infty$ instead of the 1-norm. In order to avoid the undesired solution $a_i^{(s)} = 0$ for all i, s, we have to enforce a restriction on the matrices A_i. Analogously to the Gustafson and Kessel algorithm we require the matrix A_i to have a fixed value ϱ_i for the determinant, which determines the size or the volume of the cluster i. If nothing is known about the data, we simply choose $\varrho_i = 1$ for all $i = 1, \ldots, c$.

All results were quite satisfactory. In 90-100 percent of the test runs the evolutionary algorithm was able to assign the data to the correct clusters and the cluster centres were detected approximately correct. Figures 11.5 and 11.6 show two 2-dimensional examples. In both cases we used two data sets – one in which data points were only placed on the edges of the rectangles and one where the rectangles were filled with data points.

It is worth noticing that the best chromosome in Figure 11.6 for the data set with points only on the edges had the vectors $(0, -0.1)^\top$ and $(0, -5.1)^\top$ for

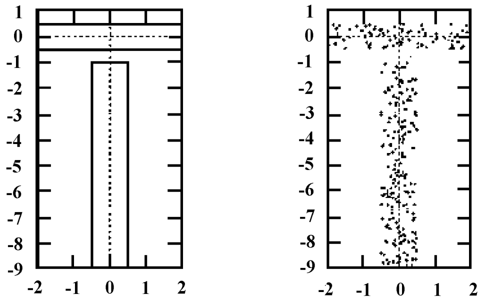


Figure 11.5: Two separated rectangular clusters (not filled and filled)

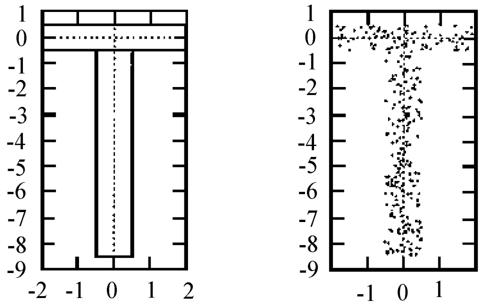


Figure 11.6: Two contiguous rectangular clusters (not filled and filled)

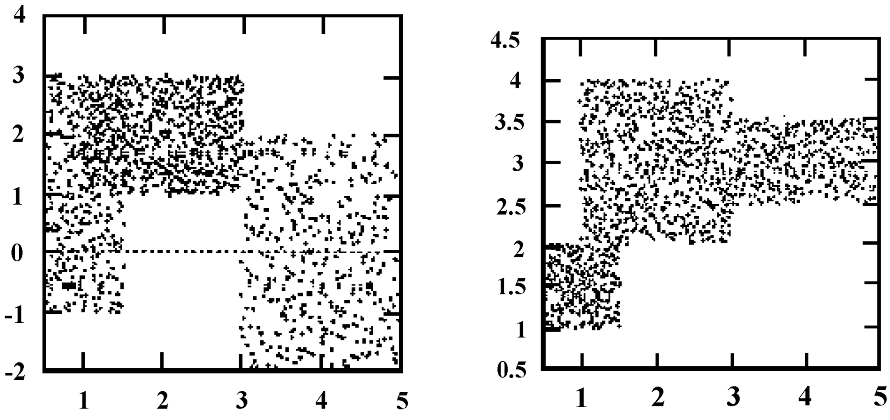
the cluster centres and the values 0.6 and 1.6 for the entree in the upper left corner of the diagonal matrices. The other non-zero value of the 2-dimensional diagonal matrix can then be calculated, since the determinants are fixed. This chromosome was assigned the error value 626.5 whereas the 'desired' solution with centres $(0,0)^\top$ and $(0,-4.5)^\top$ and matrix values 0.5 and 2.0 gets a larger error value of 740.8. Thus this method is not suited for computing the correct parameters of rectangular shells. Nevertheless, the assignment of the data to the clusters was satisfactory even if data points were only present on the edges. Also the 3-dimensional case in Figure 11.7(a) and 11.7(b) caused no problems.

The good results on rectangular clusters described in the previous section can be applied to deriving fuzzy rules from data.

11.3 EA-Based Clustering for Rule Learning

In this section we take a closer look at evolutionary algorithm based fuzzy clustering (*EA-based fuzzy clustering*) and learning fuzzy rules. 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 as described in chapter 9.

Fuzzy clusters are usually not bounded in the sense, that even data very far away from one cluster centre have non-zero membership degree to that cluster. Normally these degrees tend to be small as long as the number of clusters is not too small. In [52] an evolutionary algorithm based fuzzy clustering algorithm



(a) x/y-projections (b) x/z-projections

Figure 11.7: Projections of three 3-dimensional clusters

was proposed that constructs membership functions in the form of hyper-cones for the clusters. In this way the membership degree to a cluster becomes zero if the data point is located outside a hyper-ellipsoidal region defined by the hypercones. Nevertheless, the more serious problem appearing when extracting rules from fuzzy clusters is not solved by this approach, namely the problem of a certain loss of information enforced by the projection of a multidimensional cluster.

The approach described in [111] constructs for each cluster a collection of triangular fuzzy sets – one for each dimension – with an evolutionary algorithm and avoids the projection of the clusters in this way. However, there are no further restrictions for the fuzzy sets than the triangular membership functions. It is possible that one set might be contained in another or that two fuzzy sets strongly overlap.

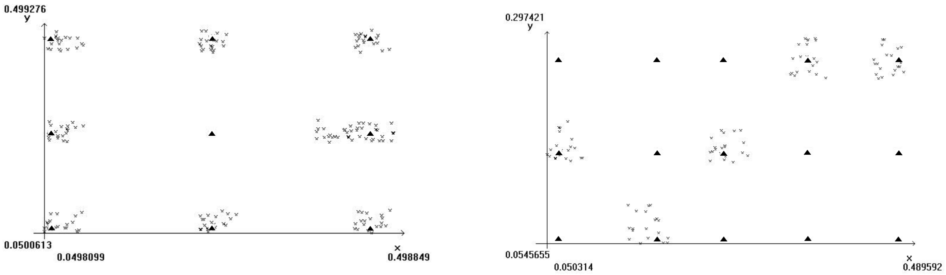
A method for constructing rules by fuzzy clustering that are restricted to well-behaved triangular membership functions (in the sense that the membership degrees at each point add up to 1) is the grid clustering, described in section 9.2 and [64]. It is a fuzzy clustering algorithm that aims at finding fuzzy partitions for the single domains on the basis of multidimensional data. For the grid clustering we assume that we are given a data set $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$. We are looking for fuzzy partitions on the single domains consisting of triangular membership functions with the restrictions 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. In that sense we have to determine a suitable grid in the multidimensional 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.

The grid clustering algorithm introduced in section 9.2 has the same concept. It is not based on an objective function, but relies on a heuristic strategy for constructing the clusters. In order to improve this grid clustering algorithm, a suitable objective function that can be optimised by an evolutionary algorithm was designed.

The aim of this strategy is to place the prototypes on a suitable grid in the multidimensional space in order to get fuzzy partitions on the single domains consisting of triangular membership functions. Therefore our objective function should not depend on the order in which we analyse the single dimensions. The coordinates of the prototypes should only be influenced by those coordinates of the data that are relatively near (i.e. have a small Euclidean distance) to the prototype's coordinate. A simple way in single dimensions is to let the data between two prototypes only influence these two. On a grid there are a couple of prototypes with the same coordinates in a single dimension, see figure 9.6 in section 9.2. In the objective function each coordinate has only once to be taken into account. These considerations led us to the following objective function

$$\begin{aligned} J^{grid}(X,U,v) \;=\; & \sum_{s=1}^{p}\sum_{j=1}^{c_r}\Bigg(\sum_{\substack{k\in\{1,\dots,n\}:\\ v_{j-1}^{(s)}<x_k^{(s)}<v_j^{(s)}}} \left(\frac{v_j^{(s)}-x_k^{(s)}}{v_j^{(s)}-v_{j-1}^{(s)}}\right)^m \\ & + \sum_{\substack{\ell\in\{1,\dots,n\}:\\ v_j^{(s)}<x_\ell^{(s)}<v_{j+1}^{(s)}}} \left(\frac{x_\ell^{(s)}-v_j^{(s)}}{v_{j+1}^{(s)}-v_j^{(s)}}\right)^m\Bigg). \end{aligned} \tag{11.1}$$

$x_k^{(s)}$ and $x_\ell^{(s)}$ are the s'th coordinate of the data vectors x_k and x_ℓ, respectively ($k,\ell \in \{1,\dots,n\}$). We assume that we have c_s triangular fuzzy sets ($s \in \{1,\dots,p\}$) in the s'th dimension. Each triple $v_{j-1}^{(s)}, v_j^{(s)}, v_{j+1}^{(s)}$ induces a triangular membership function with the interval $(v_{j-1}^{(s)}, v_{j+1}^{(s)})$ as the support and $v_j^{(s)}$ as the point with the membership degree one. Thus the fractions in the sums (without the power m) provide the value one minus the membership degree to the triangular membership function with the tip at $v_j^{(s)}$ of the data (or better: their s'th projection) that lie on the support of the membership function. Since we add up the values for all triangular fuzzy sets (and the sum of the membership degrees of a datum to neighbouring fuzzy sets yields one), we obtain the smallest considerable value of $J^{grid}(X,U,v)$, when all the membership degrees are 0.5 (as long as $m > 1$ is chosen) and the largest considerable value, when the membership degrees are either zero or one. The scope is to locate the $v_j^{(s)}$ in the centre of the data clusters, i.e. the membership degree is near one for data in the cluster and near zero for data in other clusters. Thus we aim at maximising $J^{grid}(X,U,v)$. Note that $J^{grid}(X,U,v)$ is a measure very similar to the partition coefficient, see section 4.1.2 and [15].



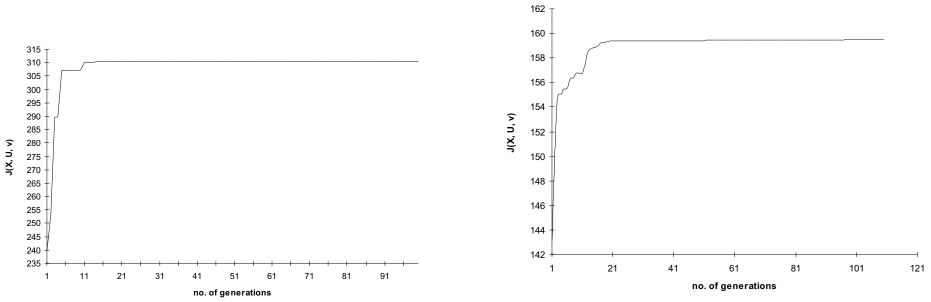
(a) Example 1

(b) Example 2

Figure 11.8: EA-based grid clustering results

A special treatment is needed for the data on the left/right of the leftmost/rightmost prototype (the values $v_1^{(s)}$ and $v_{c_s}^{(s)}$). In the beginning, we assume that the $v_j^{(s)}$ are uniformly distributed, i.e. equidistant. We add in each dimension two additional prototypes $v_0^{(s)}$ and $v_{c_s+1}^{(s)}$, again equidistant to the left and right of $v_1^{(s)}$ and $v_{c_s}^{(s)}$, see figure 9.6 in section 9.2. The values $v_0^{(s)}$ and $v_{c_s+1}^{(s)}$ are assumed to be fixed and must not be changed by the evolutionary algorithm. Nevertheless, we have to take these additional prototypes into account in the objective function so that the data at the edge of the domain have the same influence as the data in the centre. This means that the second sum in $J^{grid}(X, U, v)$ actually goes from $j = 0$ to $j = c_s + 1$. For the construction of the prototypes we only need the grid coordinates in each dimension. In the evolutionary algorithm every single dimension has a population and descendants that contain the coordinates of this domain. The size of the population and the descendants can differ for each dimension. The objective function is simply the sum over all dimensions, so the best populations are derived independently of the other domains. To determine the best prototypes, the best coordinate sets for the single domains are evaluated. Therefore, the descendants are calculated. Depending on the chosen evolution strategy all possible combinations of coordinates, either of the descendants or the descendants and the population, are evaluated. An ordinary evolution strategy (i.e. with normally distributed mutation and no crossover) is applied. Both, the +-strategy (survival of the fittest of the parents and the children) and the ,-strategy (parents are extinguished, survival of the fittest for the children) have been tested.

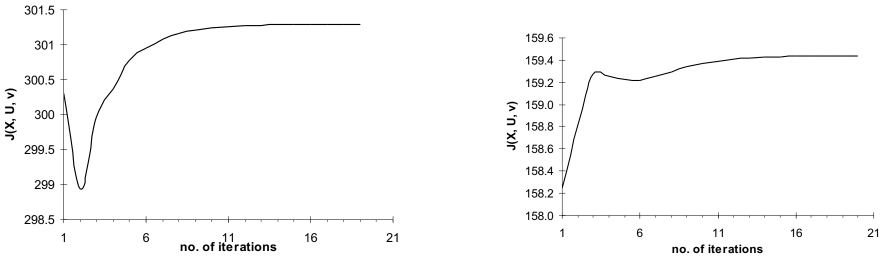
In most cases the initialisation with equidistant $v_j^{(s)}$ is good enough that the +-strategy does not fall into local optima and leads to satisfying results. In case of the ,-strategy the population of descendants has to be much larger than the population so that the evaluation time increases drastically. All combinations of coordinates have to be computed in each dimension. If P is the



(a) Objective function values for figure 11.8(a)

(b) Objective function values for figure 11.8(b)

Figure 11.9: Objective function values for the evolutionary algorithm



(a) Objective function values for figure 11.8(a)

(b) Objective function values for figure 11.8(b)

Figure 11.10: Objective function values for the heuristic grid clustering algorithm

size of the population for one dimension and N is the number of descendants

$$\binom{N}{P} \text{ (,-strategy), respectively } \binom{N+P}{P} \text{ (+-strategy)}$$

possible combinations have to be evaluated. Heuristics like tabu search, see [46, 47] or [3], could help to reduce the evaluation time. The chosen strategy for the examples is the +-strategy with variance 0.9 and 100 Iterations. Figures 11.8(a) and 11.8(b) illustrate the results for two examples (showing only the grid points).

To compare the evolutionary algorithm with the original heuristic grid clustering algorithm described in section 9.2, the results of the latter one were evaluated with the objective function (11.1). The best result of the evolution strategy (Figure 11.8(a): 310.5, Figure 11.8(b): 159.5) is in both examples better than the results of the original grid clustering algorithms (for the data set in Figure 11.8(a): 301.3, in Figure 11.8(b): 159.4). Using the +-strategy, the evolutionary algorithm has the advantage of monotonously increasing objective function values. The development of the objective function in case of the evolutionary algorithm based on the +-strategy is illustrated in figure 11.9. 11.9(a) shows the objective function values for the data set from figure 11.8(a), whereas the objective function values for the second example (see figure 11.8(b)) are illustrated in figure 11.9(b). For the heuristic grid clustering algorithm, the objective function values are denoted in figure 11.10(a) and 11.10(b), respectively.

In case of the heuristic clustering algorithm the best result for the first example is obtained after the first iteration, see figure 11.10(a). Faster evaluation is an advantage of the grid clustering algorithm. The values of the objective function for the heuristic grid clustering are rather worse than the results obtained from the evolution strategy. It has to be taken into account that the grid clustering algorithm is not objective function based and therefore not tailored for the particular objective function of the evolutionary algorithm. Nevertheless, the results show that it is a successful heuristic method being much faster than the evolution strategy.

Chapter 12

Conclusions and Future Perspectives

In this work we have shown how objective function based fuzzy clustering techniques are used for analysis problems in air traffic management tasks. Therefore, well-known algorithms have been presented and suitable extensions have been developed.

We have introduced extensions that allow the clustering techniques to adapt to special structures in data, e.g. the size of single clusters, separating unusual data from the usual case, reducing the influence of outliers on the whole partition, or comparing a kind of context sensitive regions.

However, the more flexible a clustering technique is designed the more tends the resulting partition towards local optima. In the transfer passenger example this behaviour is reflected in the validity functions. For the more flexible size-adaptive and outlier based fuzzy c-means algorithm the deviation between minimal and maximal validity values for the same number of clusters is higher than for the FCM based on probabilistic clustering. A suitable initialisation, e.g. with the result of a less flexible clustering technique, helps to overcome this drawback. In addition a technique's parameters can be chosen in a way to receive more strict results. A somewhat smaller value for the fuzzifier m in case of more flexible clustering techniques is a well-known possibility – e.g. for possibilistic clustering – to obtain reliable results.

The possibilities of context sensitive clustering have been illustrated for the task of image recognition with an image of the DLR research aircraft "*ATTAS*". We have seen, that this technique is suited to detect similar regions in an image and can e.g. be used to reduce image size. However, the scope of this approach is not to reduce the size of an image without optical deterioration. This problem is handled by numerous other approaches based e.g. on Fourier transformation. The presented approach was developed to cope with data where groups of attributes instead of single attributes have to be compared. This enables us to distinguish e.g. background parts from regions of specific interest in an image.

The flight route example has shown that a size-adaptable extension of clustering techniques based on a cluster's covariance matrix to transform the

Euclidean distance is suited to detect ellipsoidal structures of different extensions in data. We have shown how line segments can be extracted from (hyper-) ellipsoidal structures. Techniques based on fuzzy clustering are able to handle a certain amount of scattering in data due to the assigned gradual membership degrees. A radar data set for a number of weeks at a large airport is too huge to handle as a whole. Additionally, scattering of a complete data set is too high to identify any structures. Especially in low traffic situations or under extreme conditions (e.g. heavy rain or snow storms) other flight routes than the usual are used. Our results can be improved if expert knowledge for a better pre-processing of data is available. Also additional attributes than the used geographical coordinates, e.g. a flights direction in relation to its actual position, lead to a more realistic result. Density based clustering techniques are an alternative to the presented approach.

We have shown the process of rule learning for the example of transfer passenger analysis. Here, we have developed a rule-system describing the changes of the amount of transfer passengers during a day in dependence on time, aircraft size, and a destinations or origins distance. Our size-adaptable and outlier based clustering techniques are well-suited for rule learning. As long as we stay with basic clustering techniques suited for rule learning, it is possible to apply the developed extensions. We have seen that the capabilities of some clustering techniques to adapt to special cluster forms, esp. (hyper-) ellipsoidal structures, lead to a significant loss of information in rule learning. The capabilities of our extensions for size-adaptation, attribute weighting, and handling outliers do not change a (hyper-) ellipsoidal cluster's position in the multi-dimensional space. Therefore, these techniques have no influence on a techniques general suitability for rule learning. However, in our transfer passenger example we have seen, that size-adaptation can lead to some clusters with very large extensions. The resulting fuzzy sets cover a wide range of the single attributes. This way it is possible that a fuzzy rule resulting from a very large cluster is meaningless. This result can be an indication that the attributes under analysis are not sufficient to describe a certain behaviour. Without size-adaptation a group of rules seen together in our transfer passenger example leads to results comparable to a rule resulting from a larger cluster with size-adaptation. In general the resulting rule base is quite capable to describe transfer passenger behaviour.

The idea of rule learning led to the development of the grid clustering technique. The aim is to partition data in a way that meets the demand of an ideal rule system. However, the resulting objective function is not suited to derive necessary conditions by differentiation suited for a clustering algorithm. Therefore, an evolutionary algorithm-based fuzzy clustering techniques has been developed that optimises the corresponding objective function. Our experiments have shown that the heuristic grid clustering approach needs less computational effort than the technique based on an evolutionary algorithm and leads to comparable results. Where heuristic techniques are not suitable and it is not possible to use a differentiable objective function, techniques based on evolutionary algorithms can be a good alternative.

To cope with discrete variables is in general difficult to solve with fuzzy clustering techniques. Special care has to be taken to select a suitable distance measure or to find a suitable scaling of a data sets attributes in a preprocessing step. Especially clustering algorithms based on distance measures that adapt to attribute depending extensions of the clusters tend to partition a data set into the categories corresponding with the discrete variables values. Comparable scaling of the single attributes and using the Euclidean distance measure help to overcome these problems. For our transfer passenger example we transformed the attributes values into comparable ranges and used the fuzzy c-means as basic clustering technique.

Data recorded on airports usually is not restricted to radar plots and flight specific information but contains further information. In addition weather, waypoints, and several derived variables – e.g. weight class, necessary separation of aircraft, and runway length – are used for delay analysis. The overall goal of tasks in air-traffic management is to increase airport capacity and reduce delay times, if possible in combination with reducing noise and environmental pollution. Capacity studies are performed using simulation tools and varying parameters that are known to influence the airport capacity. Despite generating rules to describe under which weather conditions significant delays agglomerate in air traffic, it is useful to know which weather attributes have the most influence on dispatching flights. It is obvious that staggering of aircraft has to be increased in bad weather conditions. For arrival or flight time prognosis purposes the delay time has to be predicted. The less attributes have to be taken into account in a reliable simulation or decision support tool the less computational as well as technical effort is needed. Some parameters as e.g. the aircraft separation are fixed and known to define the maximum aircraft capacity. Analysing the dependencies of the recorded or derived variables and their influence on aircraft delay can help to improve airport procedures and reduce delay times. Data analysis in general and especially the developed attribute weighting fuzzy clustering technique is a possibility to identify additional attributes influencing the real situation and in this way indicate improvements for further assistance systems or airport extensions. Analysis of delay data and available weather information with the developed attribute weighting clustering technique have shown that the available weather information without further knowledge is not suited to predict delay times. Similar results have been obtained with classified data where the delay times have been replaced by delay classes. In this case extremely high error rates indicated that additional information has to be considered for analysis purposes. One problem is the calculation of delay times. Available for analysis purposes have been the actual and planned times of arrivals at Frankfurt airport. However, no information if the delay occurred already at the destination airport or on the flight route has been available. This task is intended to be further studied in cooperation with Frankfurt airport to develop a delay prediction system.

Appendix A

Illustrations

In this appendix additional illustrations are included to demonstrate the effects of certain parameters and the use of specific basic objective functions described in chapter 3 on the clustering results.

A.1 Scalar Product-Based Distance Measures

For an explanation of the scalar product-based clustering technique and the algorithm of scalar product-based clustering see section 5.2.

As fuzzifier $m = 1.5$ has been chosen for all angle-based clustering tasks. Additionally the constraint parameter ω for clustering with outliers has been chosen as the number of data, i.e. $\omega = n$, and the weighting exponent q has been set to 0.5. The four figures (A.2(a), A.2(b), A.2(c)), and A.2(d) show the distance of the data to the cluster centres, whereas the corresponding membership degrees are illustrated in figures A.3(a), A.3(b), A.3(c), and A.3(d). The weights ω_k are displayed in figure A.1.

A.2 Centre-Based Clustering

For an explanation of the centre-based clustering technique and the algorithm of centre-based clustering see section 5.2.

Figures A.4 to A.13 illustrate the differences between the original Gustafson-Kessel algorithm and the size-adaptable centre-based clustering algorithm using the same transformed Euclidean distance as GK. For all GK-sized clustering results, parameter τ was set to 1.

The GK-sized partitions are shown for two different values of the influence parameter l, $l = 0.5$ and $l = 5$. In figure A.4 and A.5 the probabilistic basic objective function from section 3.1 has been used, whereas the results in figure A.6 and A.7 were obtained with the possibilistic objective function. We see, that the choice of GK-sized has not such a significant influence on the membership degrees for possibilistic clustering as for the probabilistic case. Noise clustering results are shown in figure A.8 and A.9. Here δ has been

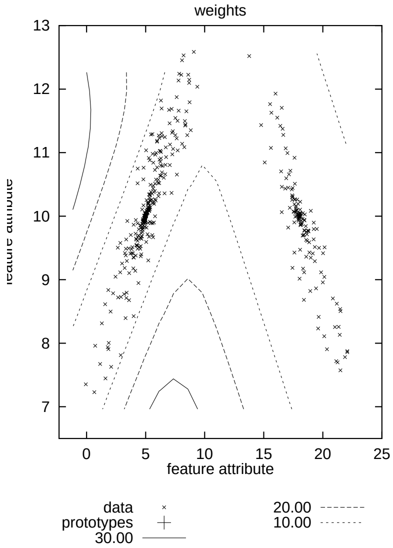


Figure A.1: Results for angle-based clustering – weights for fuzzy clustering with outliers ($m = 1.5, \epsilon = 0.001, \omega = n, q = 0.5$)

estimated as described in section 3.1.3

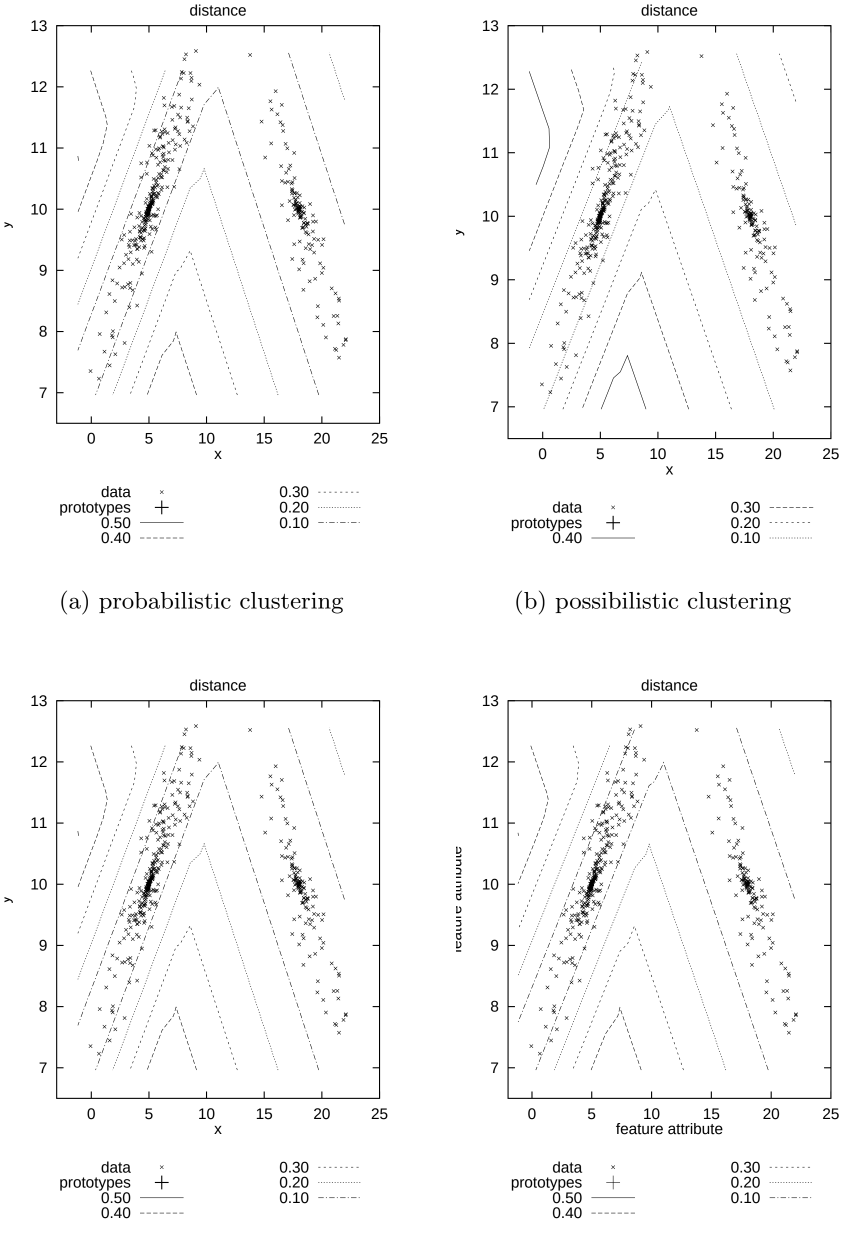
$$\delta^2 = \frac{2}{c \cdot n} \cdot \left(\sum_{k=1}^{n} \sum_{i=1}^{c} d^2(\mathsf{v}_i, x_k) \right).$$

In this case the results are similar to those obtained with probabilistic clustering.

Figures A.10 to A.13 show the membership degrees, distances, and weights for GK combined with clustering for outliers. The constraint parameter $\omega = n$ was chosen for all clustering with outliers results. The membership degrees, distances, and weights are compared for different values of the weighting exponent and the size parameters exponent. In figure A.11 the weighted membership degrees $\tilde{u}_{ik}^m$ for $l = 0.5$ and $l = 5$ are illustrated for $q = 0.5$ and $q = 2$. The distances shown in figure A.12 depend more on the choice of l, whereas q has a greater influence on the weights displayed in figure A.13. Although only two contour lines are shown for the weights in case of $q = 0.5$, the weight values have a larger range of values than for $q = 2$. So the smaller q the more emphasis is put upon weight adaptation as explained in section 3.1.4. The adapted scale values for GK-sized combined with the basic objective functions are shown in table A.1. We see that the scale adaptation depends on influence parameter l – the exponent parameter for centre-based size adaptation. The resulting scale values are similar for all basic clustering techniques.

A.3 Volume-Centre-Based Clustering

For an explanation of the volume-centre-based clustering technique and the algorithm of volume-centre-based clustering see section 5.2.



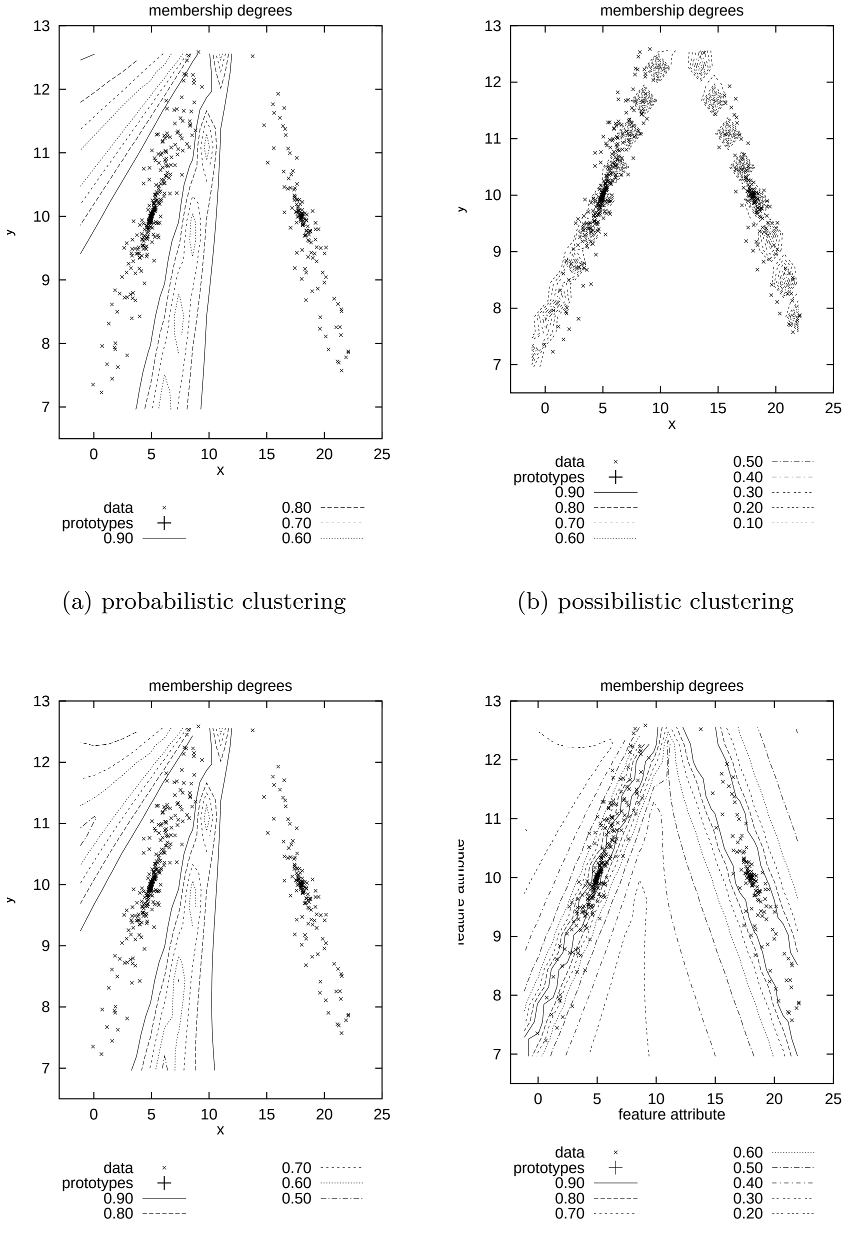
(a) probabilistic clustering

(b) possibilistic clustering

(c) noise clustering

(d) outlier clustering ($\omega = n$, q = 0.5)

Figure A.2: Results for an elliptical test data set and angle-based clustering – distance ($m = 1.5, \epsilon = 0.001$)



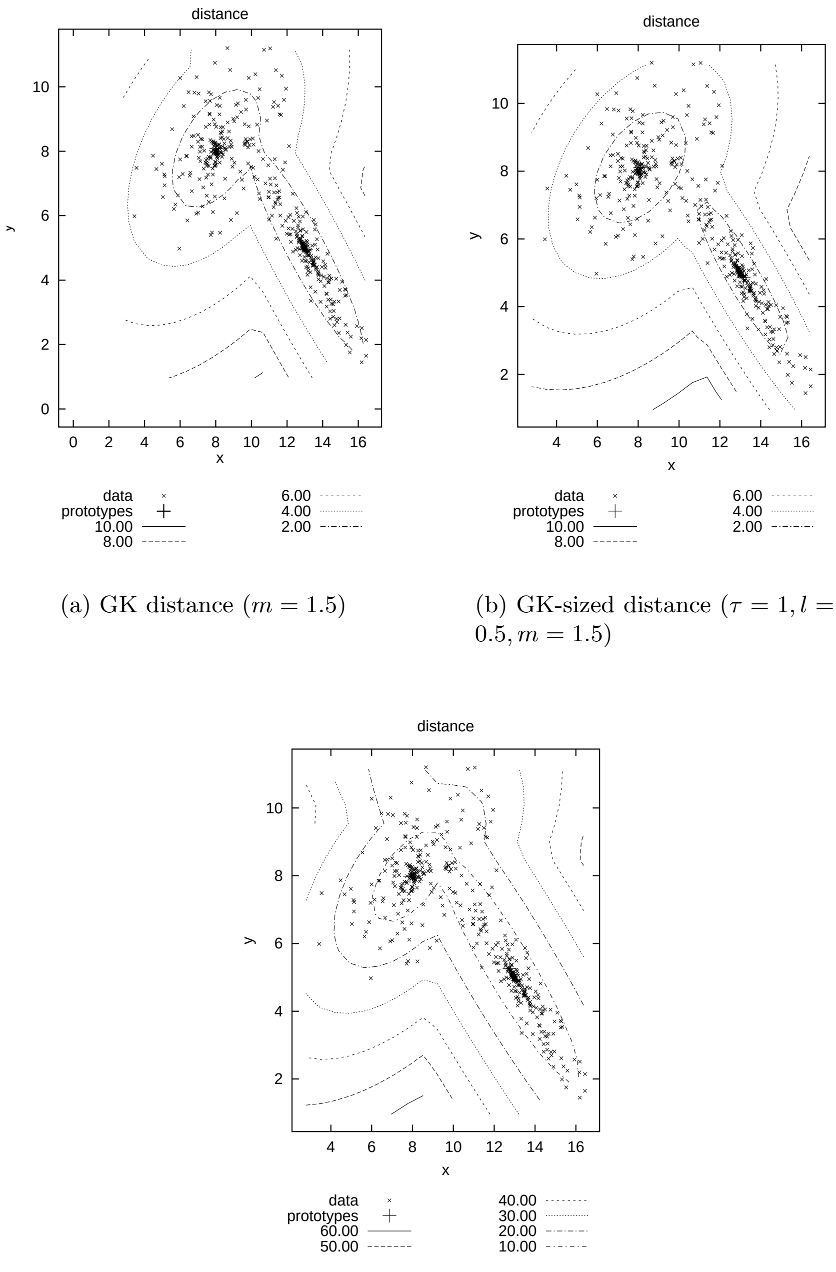
(a) probabilistic clustering

(b) possibilistic clustering

(c) noise clustering

(d) outlier clustering ($\omega = n, q = 0.5$)

Figure A.3: Results for an elliptical test data set and angle-based clustering – membership degrees ($m = 1.5, \epsilon = 0.001$)

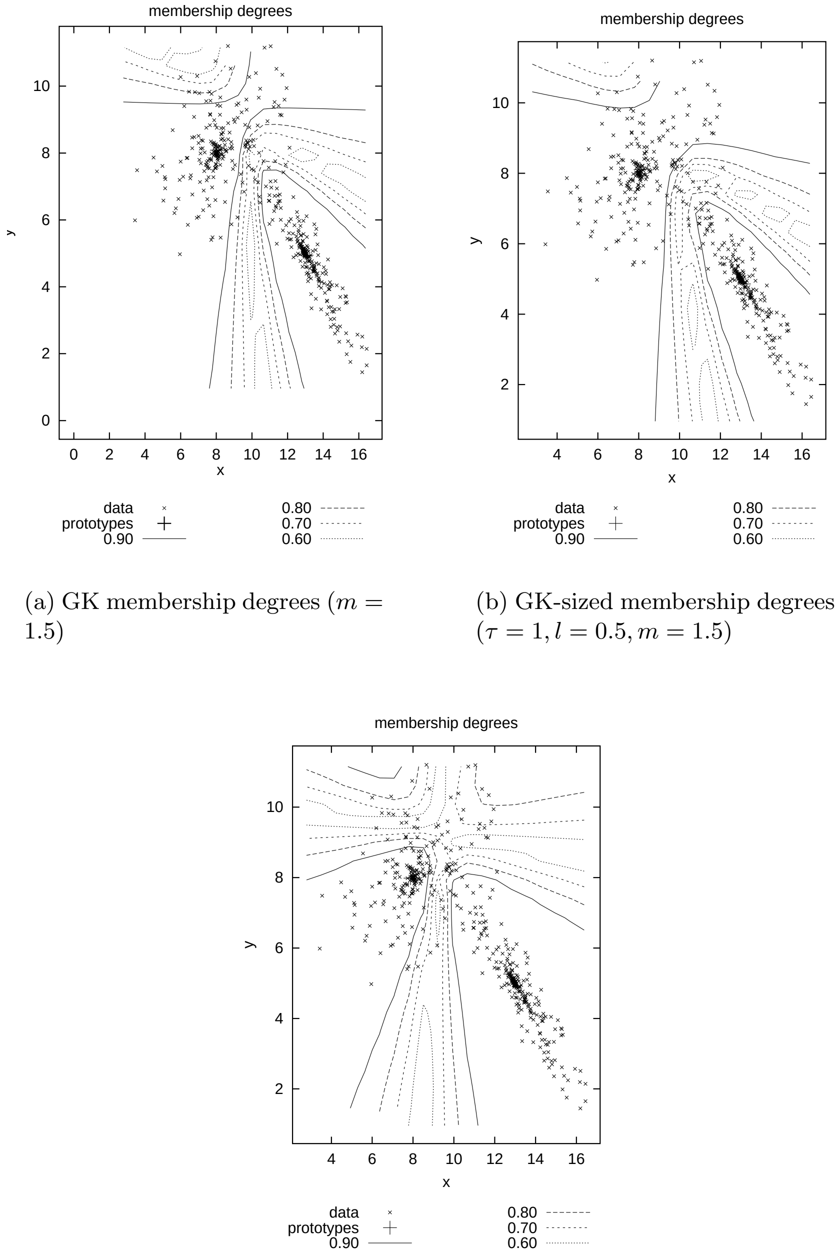


(a) GK distance ($m = 1.5$)

(b) GK-sized distance ($\tau = 1, l = 0.5, m = 1.5$)

(c) GK-sized distance ($\tau = 1, l = 5, m = 1.5$)

Figure A.4: Comparison of probabilistic GK and GK-sized clustering – distance

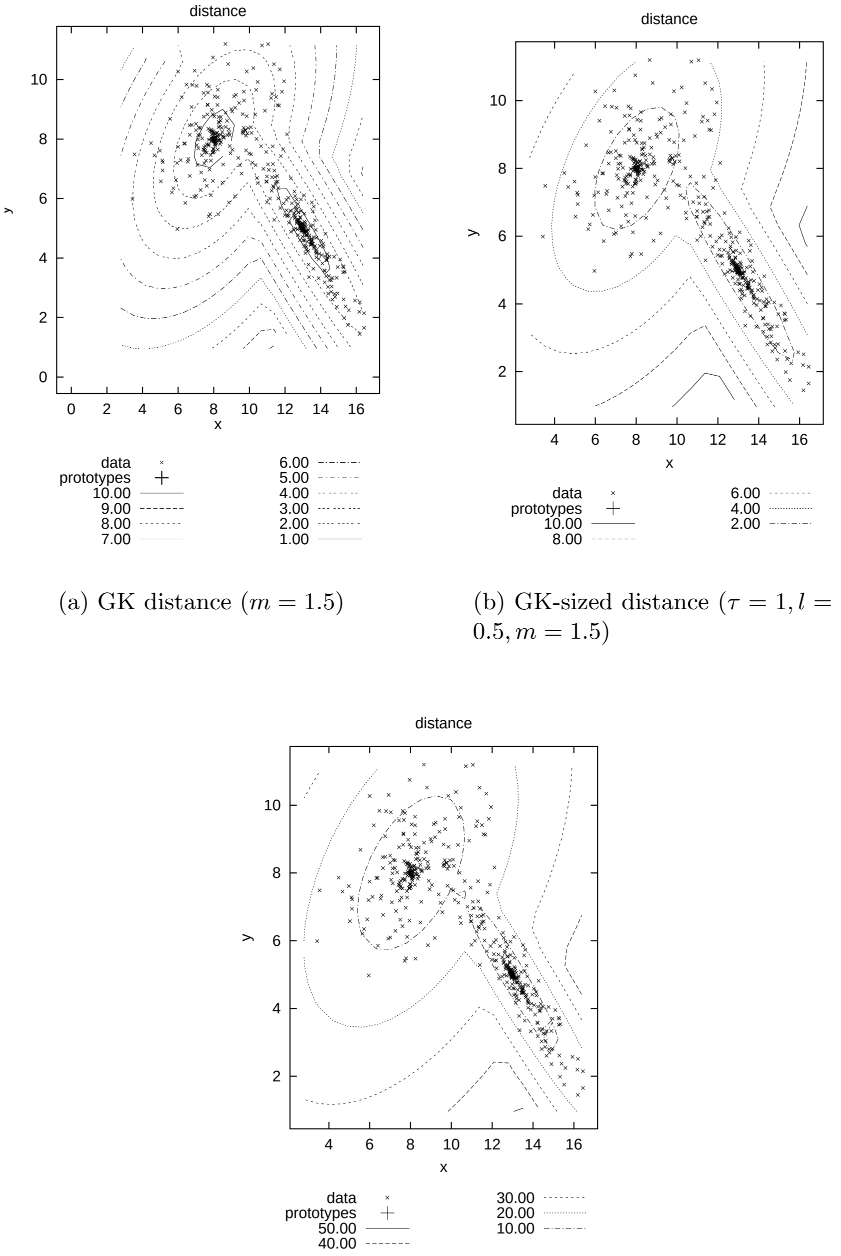


(a) GK membership degrees ($m = 1.5$)

(b) GK-sized membership degrees ($\tau = 1, l = 0.5, m = 1.5$)

(c) GK-sized membership degrees ($\tau = 1, l = 5, m = 1.5$)

Figure A.5: Comparison of probabilistic GK and GK-sized clustering – membership degrees

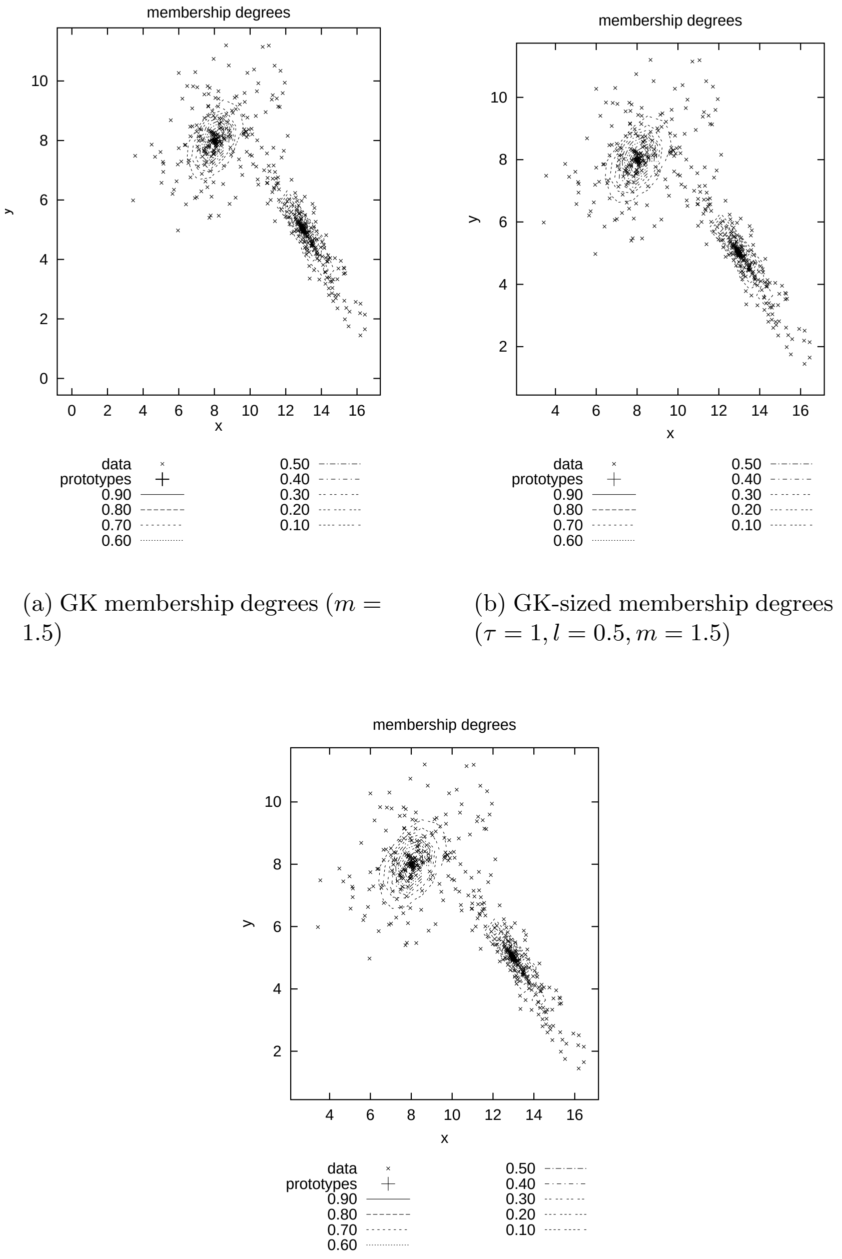


(a) GK distance ($m = 1.5$)

(b) GK-sized distance ($\tau = 1, l = 0.5, m = 1.5$)

(c) GK-sized distance ($\tau = 1, l = 5, m = 1.5$)

Figure A.6: Comparison of possibilistic GK and GK-sized clustering – distance

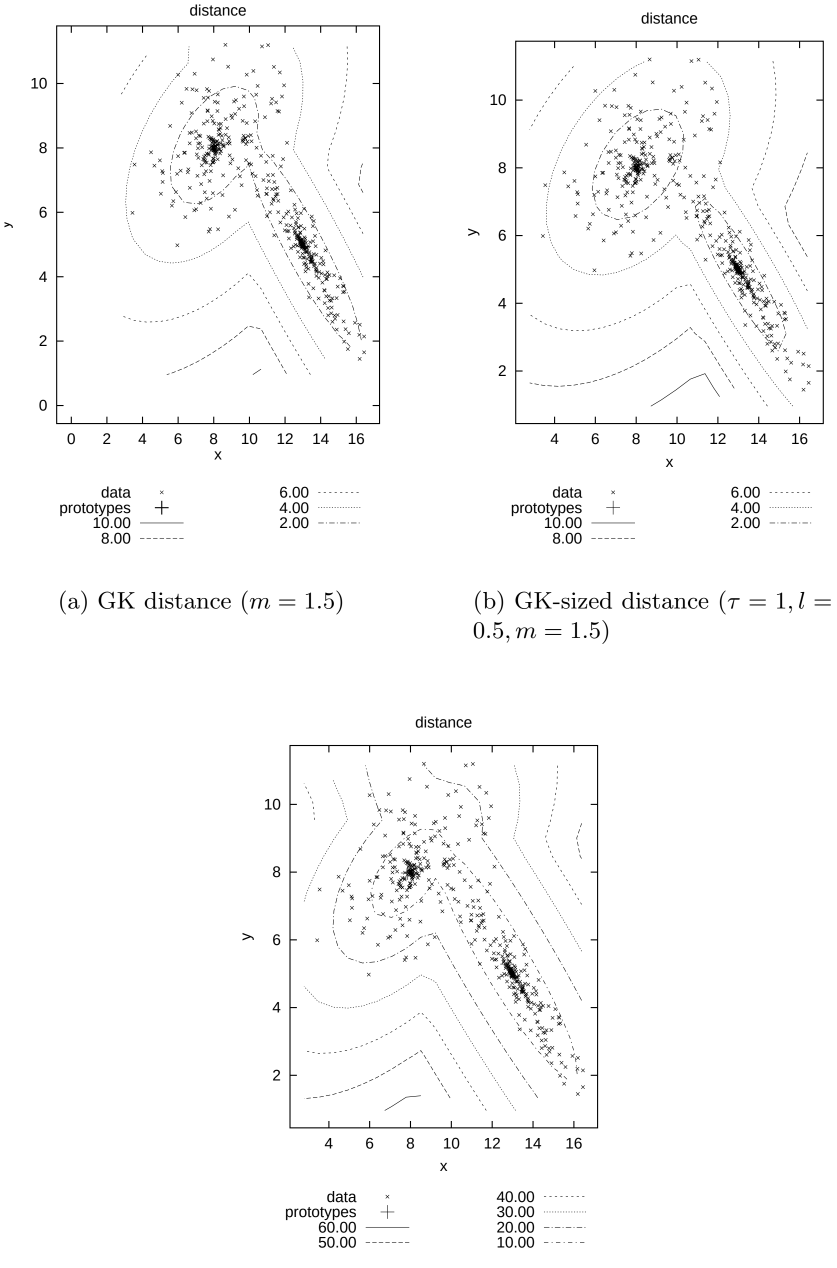


(a) GK membership degrees ($m = 1.5$)

(b) GK-sized membership degrees ($\tau = 1, l = 0.5, m = 1.5$)

(c) GK-sized membership degrees ($\tau = 1, l = 5, m = 1.5$)

Figure A.7: Comparison of possibilistic GK and GK-sized clustering – membership degrees

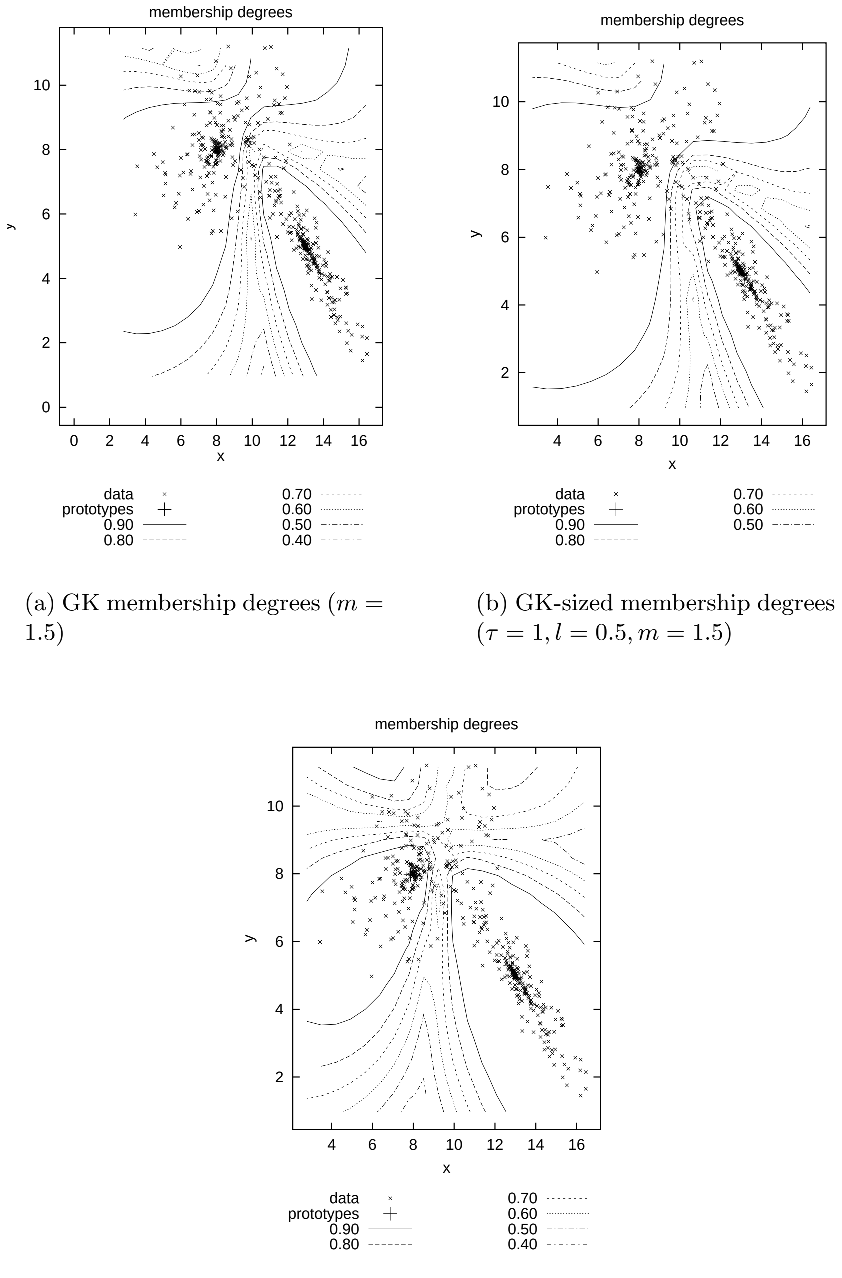


(a) GK distance ($m = 1.5$)

(b) GK-sized distance ($\tau = 1, l = 0.5, m = 1.5$)

(c) GK-sized distance ($\tau = 1, l = 5, m = 1.5$)

Figure A.8: Comparison of GK and GK-sized noise clustering – distance

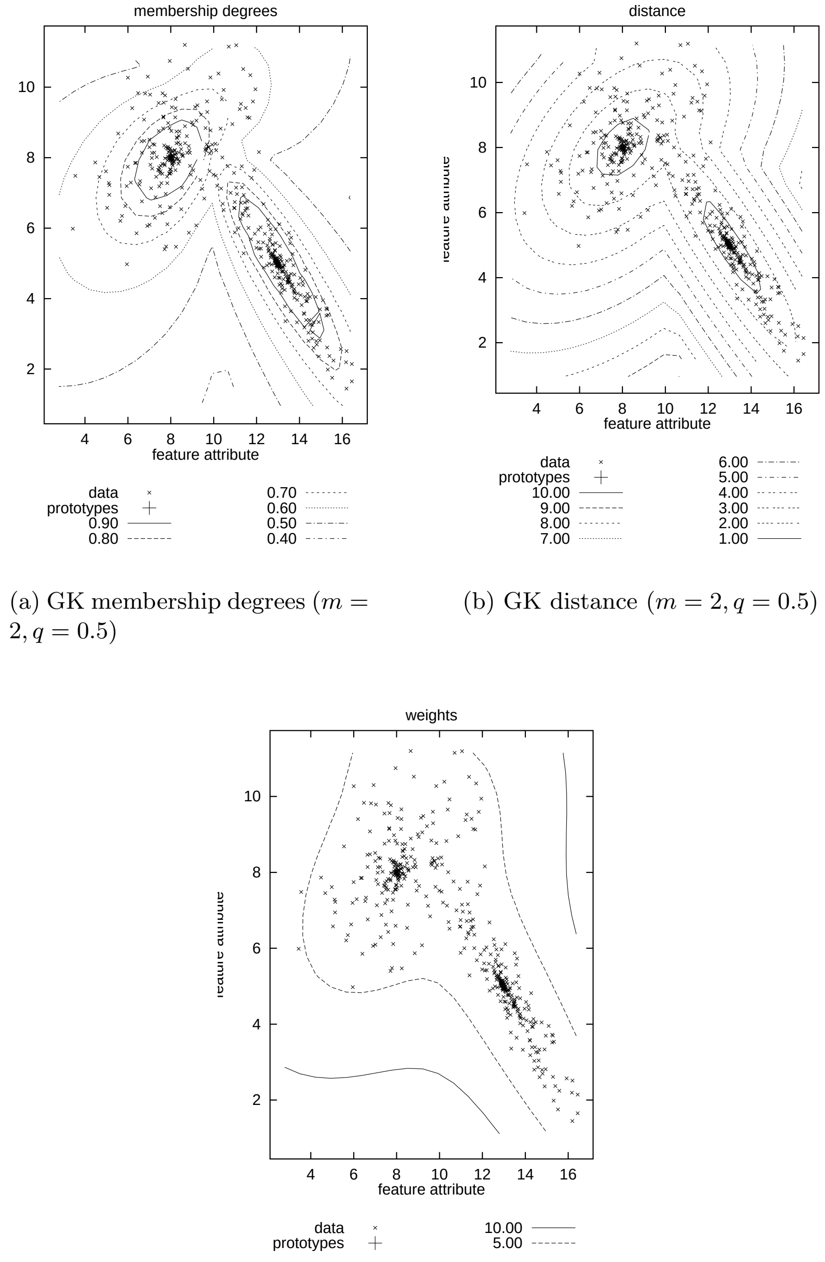


(a) GK membership degrees ($m = 1.5$)

(b) GK-sized membership degrees ($\tau = 1, l = 0.5, m = 1.5$)

(c) GK-sized membership degrees ($\tau = 1, l = 5, m = 1.5$)

Figure A.9: Comparison of GK and GK-sized noise clustering – membership degrees

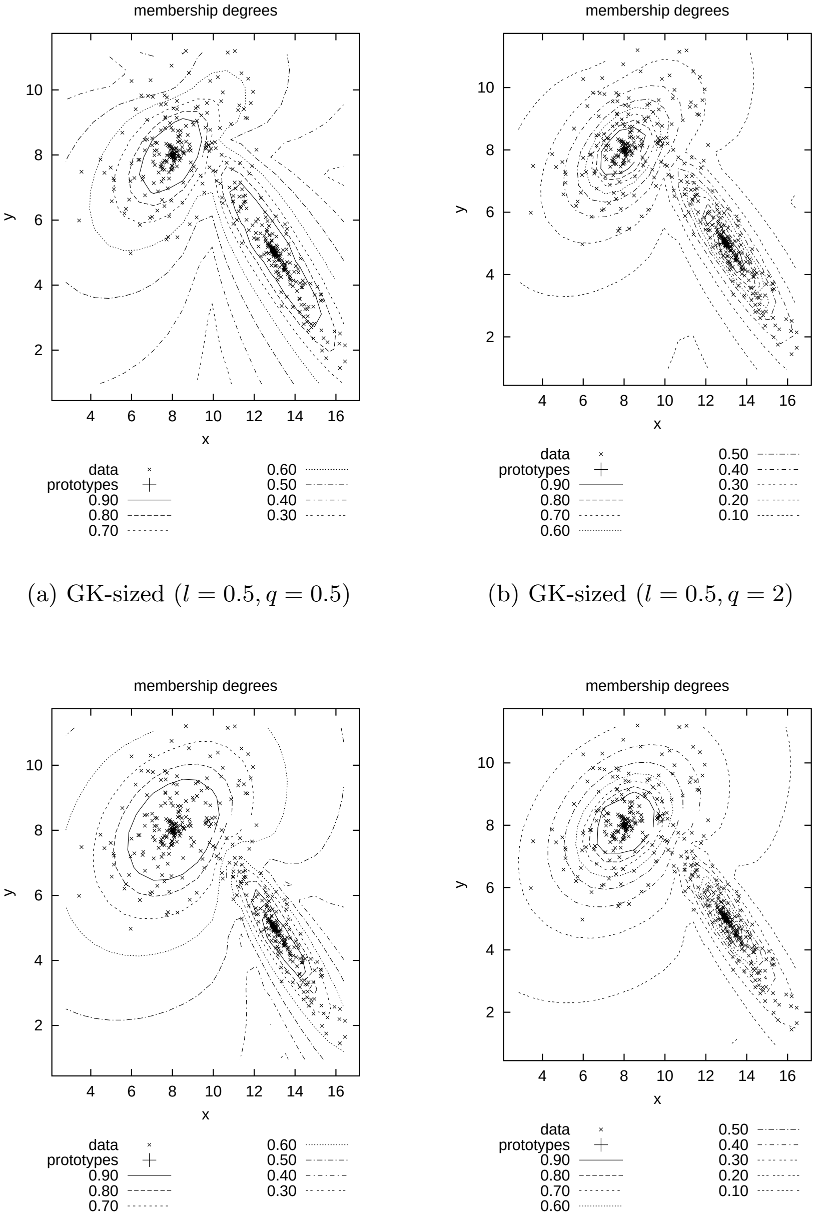


(a) GK membership degrees ($m = 2, q = 0.5$)

(b) GK distance ($m = 2, q = 0.5$)

(c) GK weights ($m = 2, q = 0.5$)

Figure A.10: Comparison of GK and GK-sized outlier clustering – GK clustering results



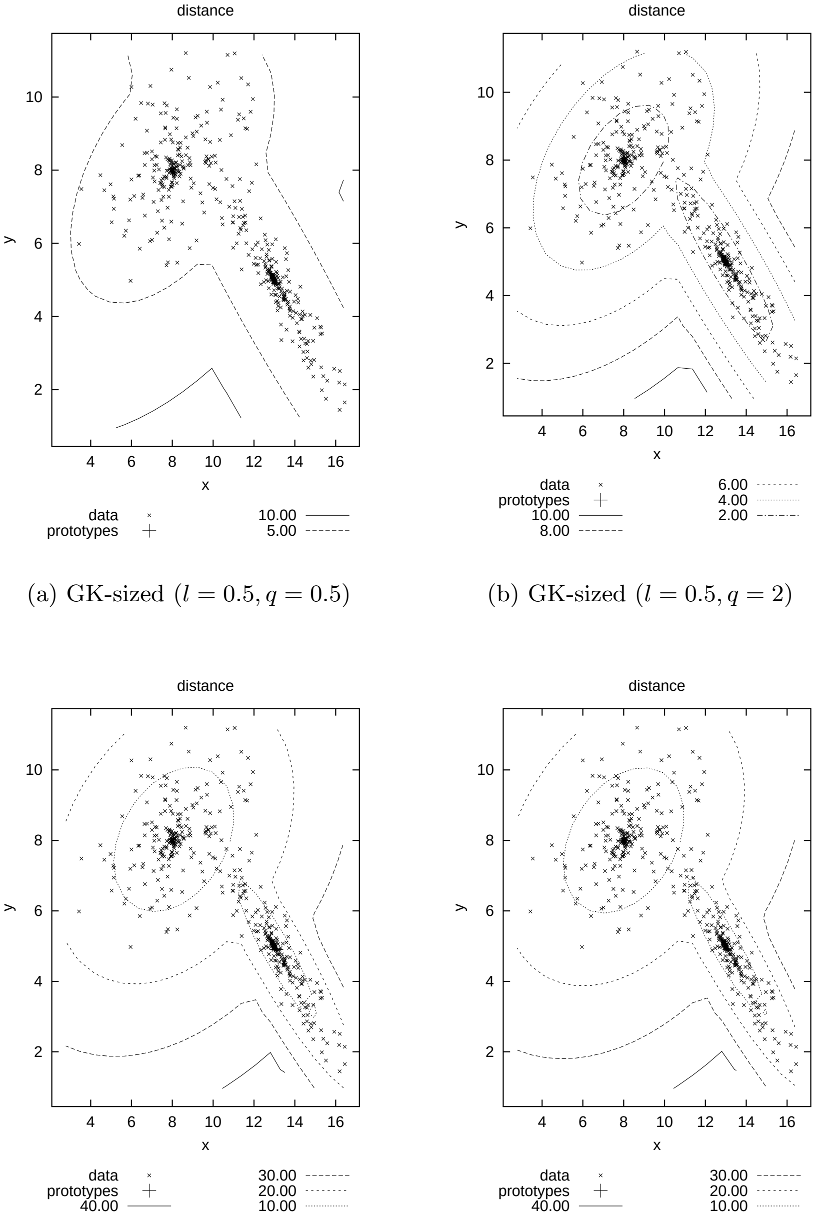
(a) GK-sized ($l = 0.5, q = 0.5$)

(b) GK-sized ($l = 0.5, q = 2$)

(c) GK-sized ($l = 5, q = 0.5$)

(d) GK-sized ($l = 5, q = 2$)

Figure A.11: Comparison of GK and GK-sized outlier clustering – GK-sized membership degrees for $m = 1.5$, $\omega = n$, and $\tau = 1$



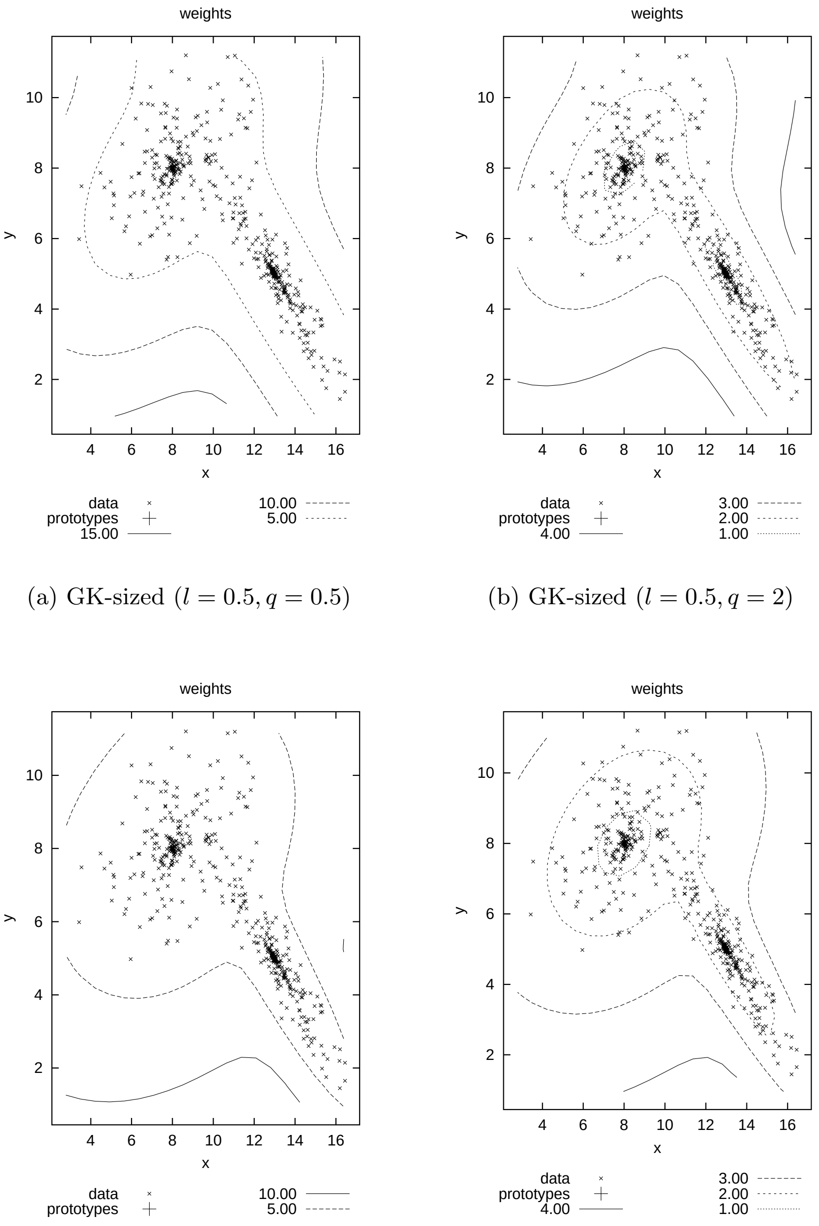
(a) GK-sized ($l = 0.5, q = 0.5$)

(b) GK-sized ($l = 0.5, q = 2$)

(c) GK-sized ($l = 5, q = 0.5$)

(d) GK-sized ($l = 5, q = 2$)

Figure A.12: Comparison of GK and GK-sized outlier clustering – GK-sized distances for $m = 1.5$, $\omega = n$, and $\tau = 1$



(a) GK-sized $(l = 0.5, q = 0.5)$

(b) GK-sized $(l = 0.5, q = 2)$

(c) GK-sized $(l = 5, q = 0.5)$

(d) GK-sized $(l = 5, q = 2)$

Figure A.13: Comparison of GK and GK-sized outlier clustering – GK-sized weights for $m = 1.5$, $\omega = n$, and $\tau = 1$

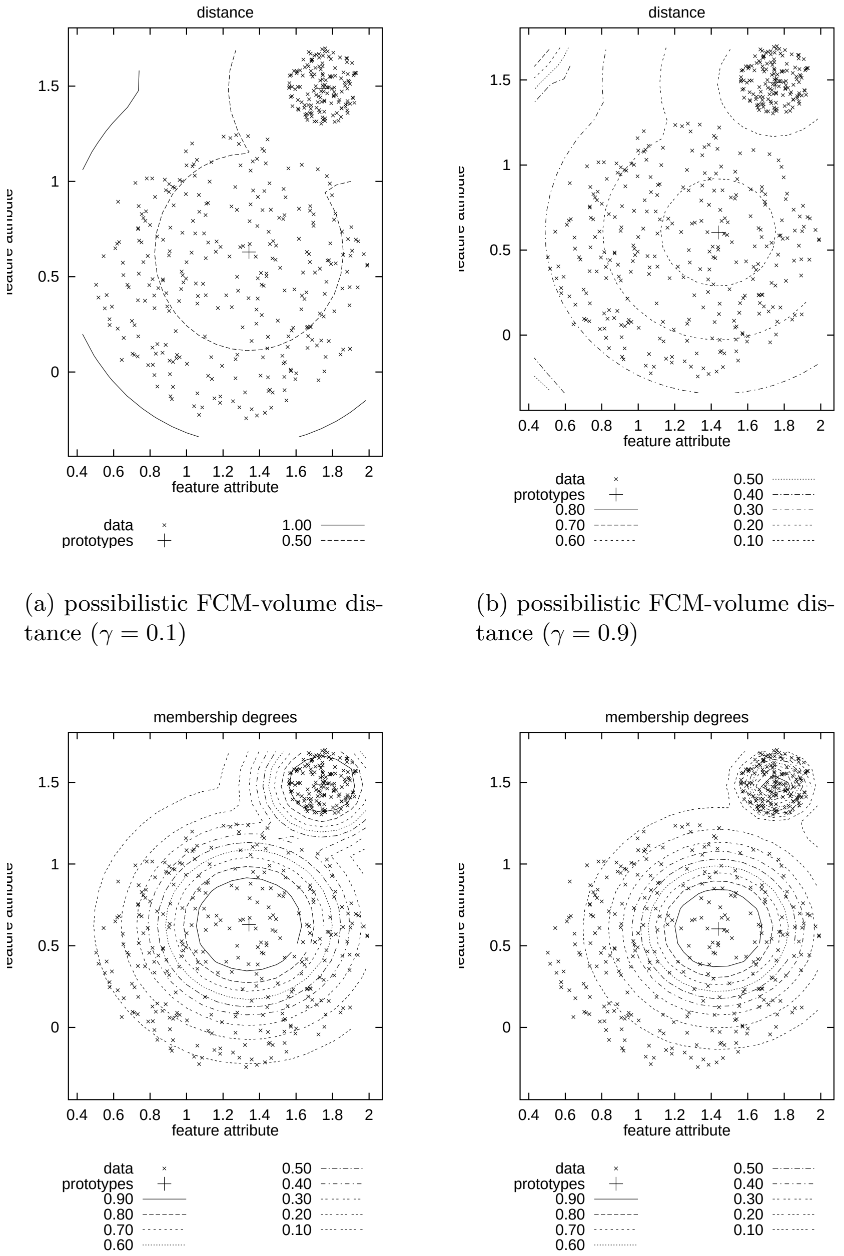
Table A.1: Scale values for GK-sized

basic obj. function	l	cluster 1	cluster 2
probabilistic	0.5	0.310	0.690
probabilistic	5	0.461	0.539
possibilistic	0.5	0.318	0.682
possibilistic	5	0.446	0.554
noise	0.5	0.312	0.688
noise	5	0.458	0.542
outlier $- q = 0.5$	0.5	0.393	0.607
outlier $- q = 0.5$	5	0.436	0.564
outlier $- q = 2$	0.5	0.369	0.631
outlier $- q = 2$	5	0.435	0.565

Figures A.14 to A.18 illustrate the clustering results for the FCM-volume clustering algorithm in combination with the possibilistic, noise, and outlier objective functions from section 3.1. The same data set used here was also used for the FCM-sized example in figure 6.2.

Figures A.16 to A.18 show the distances, membership degrees, and weights for FCM-volume combined with clustering for outliers. The constraint parameter $\omega = n$ was chosen for all clustering with outliers results. The membership degrees, distances, and weights are compared for different values of the weighting exponent q and the influence parameter γ. In figures A.17 to A.17 the weighted membership degrees $\tilde{u}_{ik}^m$ for $\gamma = 0.1$ and $\gamma = 0.9$ are illustrated for $q = 0.5$ and $q = 2$.

The influence of parameter γ on the clusters centre radii is visible in the illustration of the distance to the cluster centres. The larger γ the smaller are the parameters τ_i. The crispier transition from one cluster to another in possibilistic (resp. noise) clustering in comparison to probabilistic clustering becomes obvious in the illustration of the membership degrees for the basic objective functions.



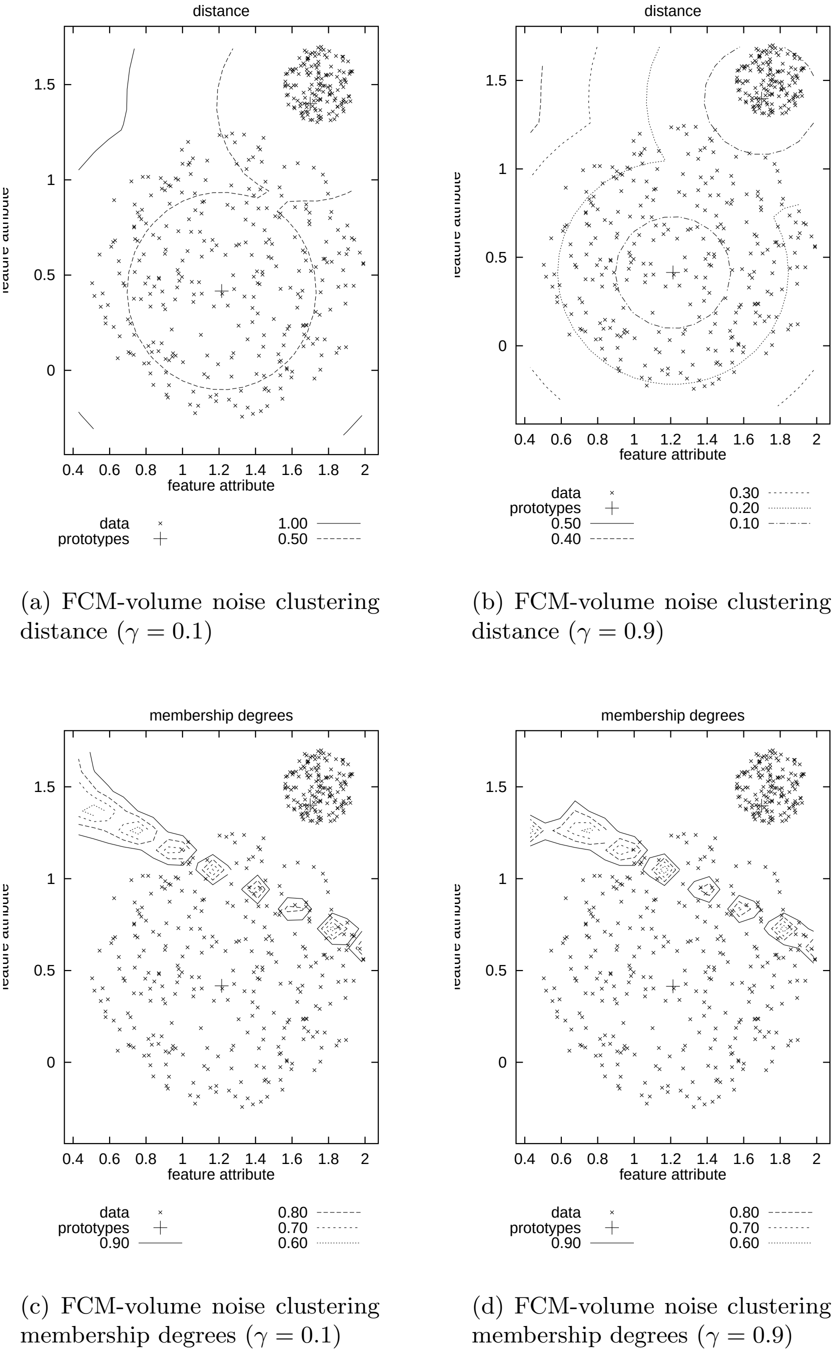
(a) possibilistic FCM-volume distance ($\gamma = 0.1$)

(b) possibilistic FCM-volume distance ($\gamma = 0.9$)

(c) possibilistic FCM-volume membership degrees ($\gamma = 0.1$)

(d) possibilistic FCM-volume membership degrees ($\gamma = 0.9$)

Figure A.14: Comparison of possibilistic FCM-volume clustering for different values of γ, $m = 1.5$, and $\tau = 1$



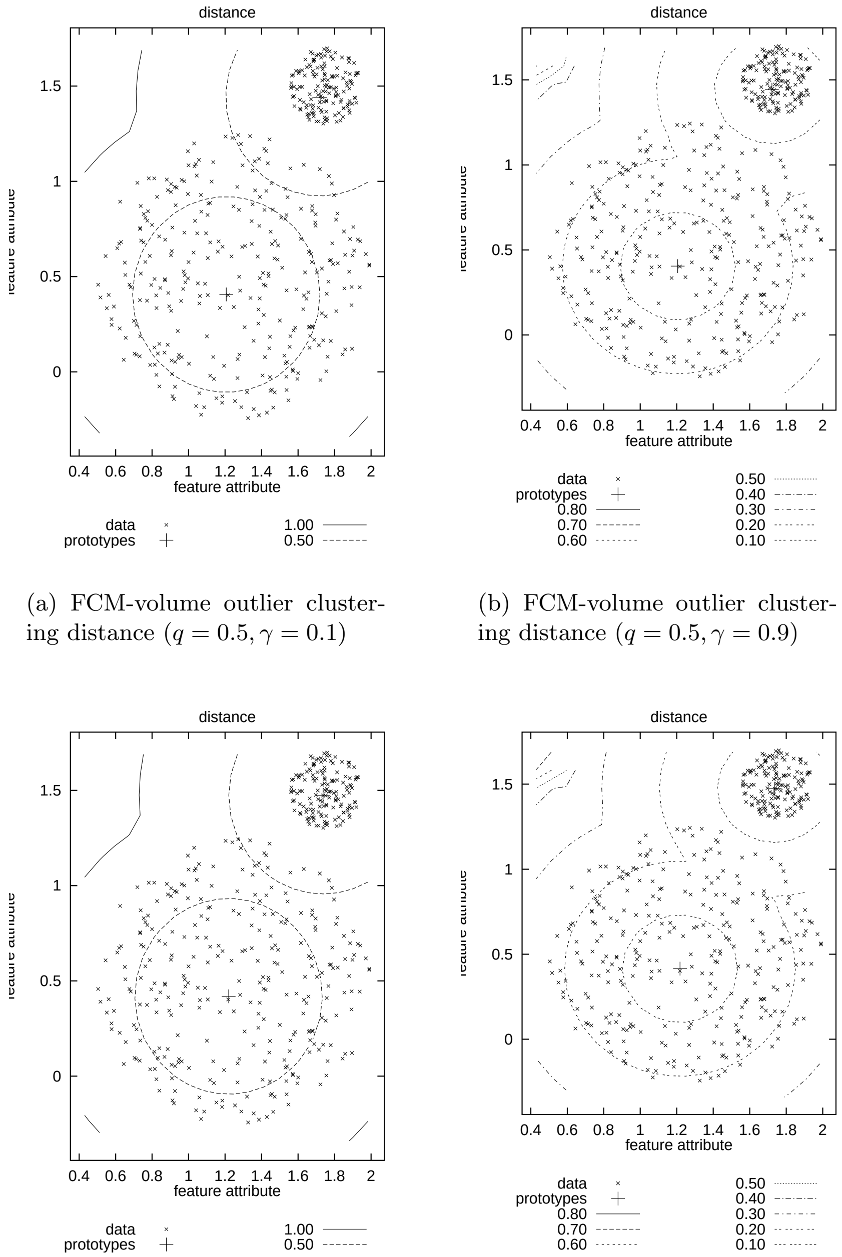
(a) FCM-volume noise clustering distance ($\gamma = 0.1$)

(b) FCM-volume noise clustering distance ($\gamma = 0.9$)

(c) FCM-volume noise clustering membership degrees ($\gamma = 0.1$)

(d) FCM-volume noise clustering membership degrees ($\gamma = 0.9$)

Figure A.15: Comparison of FCM-volume noise clustering for different values of γ, $m = 1.1$, and $\tau = 1$



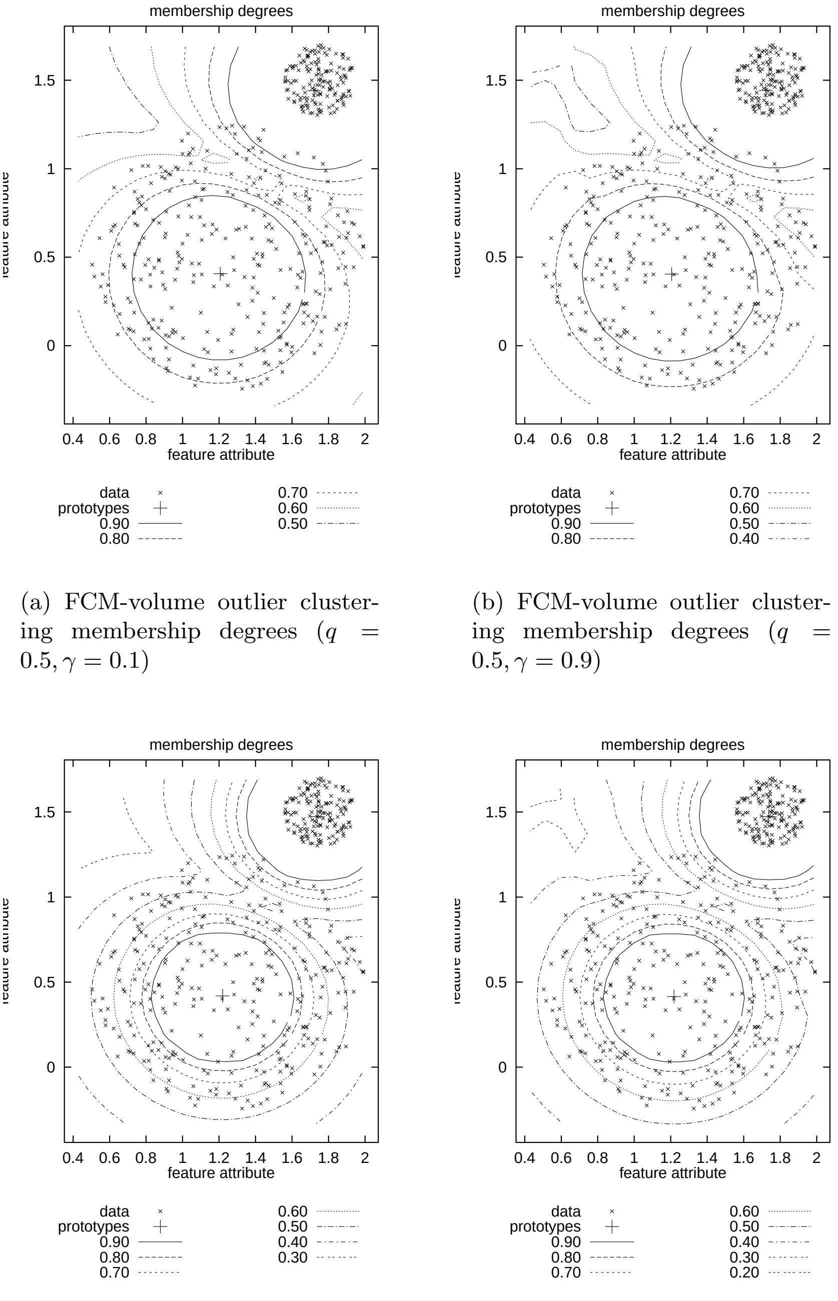
(a) FCM-volume outlier clustering distance $(q = 0.5, \gamma = 0.1)$

(b) FCM-volume outlier clustering distance $(q = 0.5, \gamma = 0.9)$

(c) FCM-volume outlier clustering distance $(q = 2, \gamma = 0.1)$

(d) FCM-volume outlier clustering distance $(q = 2, \gamma = 0.9)$

Figure A.16: Comparison of FCM-volume outlier clustering for different values of γ, q, $m = 1.5$, and $\tau = 1$ – distances



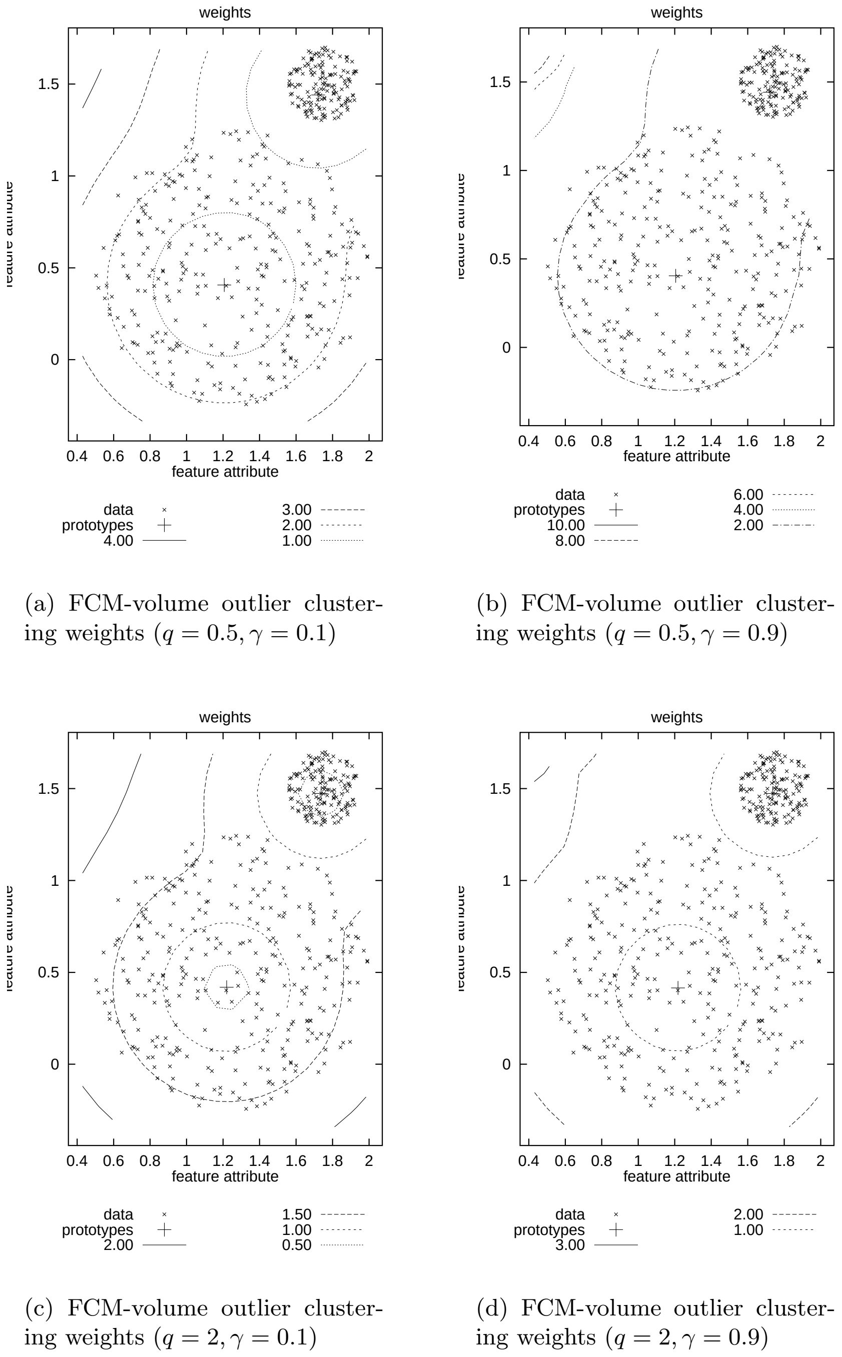
(a) FCM-volume outlier clustering membership degrees ($q = 0.5, \gamma = 0.1$)

(b) FCM-volume outlier clustering membership degrees ($q = 0.5, \gamma = 0.9$)

(c) FCM-volume outlier clustering membership degrees ($q = 2, \gamma = 0.1$)

(d) FCM-volume outlier clustering membership degrees ($q = 2, \gamma = 0.9$)

Figure A.17: Comparison of FCM-volume outlier clustering for different values of γ, q, $m = 1.5$, and $\tau = 1$ – membership degrees



(a) FCM-volume outlier clustering weights ($q = 0.5, \gamma = 0.1$)

(b) FCM-volume outlier clustering weights ($q = 0.5, \gamma = 0.9$)

(c) FCM-volume outlier clustering weights ($q = 2, \gamma = 0.1$)

(d) FCM-volume outlier clustering weights ($q = 2, \gamma = 0.9$)

Figure A.18: Comparison of FCM-volume outlier clustering for different values of γ, q, $m = 1.5$, and $\tau = 1 -$ weights

List of Figures

List of Tables

Abbreviations

FCM	fuzzy c-means algorithm	32
GK	Gustafson-Kessel algorithm	39
AGK	axes parallel Gustafson-Kessel algorithm	44
GG	Gath-Geva algorithm	45
AGG	axes parallel Gath-Geva algorithm	52
FCS	fuzzy c-shells algorithm	57
FCCS	fuzzy c-spherical shells algorithm	57
ER	error rate	60
PC	partition coefficient	62
PE	partition entropy	62
S	separation measure	63
SI	separation index	63
FHV	fuzzy hypervolume	64
APD	average partition density	64
PD	partition density	65
CCM	Compatible cluster merging	65
NAB	normalised angle-based clustering	67
AB	angle-based clustering	73
SACB	size-adaptable centre-based clustering	78
SAVB	size-adaptable volume-based clustering	85
SAVCB	size-adaptable volume-centre-based clustering	88
AWFCM	attribute weighting fuzzy clustering	109
CS	context sensitive clustering	119
Grid	grid clustering	141
EA	Evolutionary Algorithm	177

Symbols

$x_k \in \mathbb{R}^p$	data vector	6
$X = \{x_1, \ldots, x_n\}$	data set	6
n	number of data vectors	6
$\mathcal{C} = \{\mathsf{v}_1, \ldots, \mathsf{v}_c\}$	set of clusters	7
$\mathsf{v}_i, i \in \{1, \ldots, c\}$	cluster prototype	7
$v_i \in \mathbb{R}^p$	cluster centre vector	7
$d^2(\mathsf{v}_i, x_k)$	distance between cluster centre v_i and datum x_k	7
u_{ik}	membership degree of datum x_k to cluster v_i	7
U	matrix of membership degrees	7
$m \in \mathbb{R}_{>1}$	fuzzifier or fuzziness index	7
$J^* (X, U, \mathsf{v})$	basic objective function, $*$ symbolises probabilistic, possibilistic, noise, or outlier	16
$\mathfrak{I}_x$	set of clusters that have distance 0 to x_k	17
ε	termination bound for clustering algorithms	18
$\eta_i > 0$	permissible extension of cluster i in possibilistic clustering	20
$\delta >> 0$	(large) constant distance of each datum to the noise cluster in noise clustering	24
ω_k	weight for datum x_k	27
ω	constraint parameter for weights ω_k	27
$q \in \mathbb{R}_{>0}$	determines weighting influence in outlier clustering	27
$\tilde{u}_{ik}^m = \frac{u_{ik}^m}{\omega_k^q}$	overall membership degree in outlier clustering	30
$\mathcal{D}_{FCM}$	squared Euclidean distance used in FCM	32
p	dimensionality of data set X	32
$x_k^{(\nu)}$	ν'th component of data vector x_k	32
$v_i^{(\nu)}$	ν'th component of vector v_i	32
δ^2	weighted medium distance	34

C_i	symmetric positive definite covariance matrix	37
$\mathcal{D}_{GK}$	squared transformed Euclidean distance for GK	39
ρ_i	predefined size of cluster i for GK	39
$\mathbf{I}$	identity matrix	40
$\mathcal{D}_{AGK}$	squared transformed Euclidean distance for AGK	44
$c_i^{(\nu)}$	ν'th diagonal element of the covariance matrix	44
N_i	normal distribution corresponding to cluster i	48
$\mathrm{u}_{ik} \in \{0,1\}$	hard membership degree of datum x_k to cluster i	48
A_i	covariance matrix for hard partition	48
P_i	a-priori probability to choose N_i to calculate a new datum	48
$\mathcal{D}_{GG}$	squared distance for GG	48
π_i	a-priori probability for a datum to belong to N_i	48
A_i	fuzzy covariance matrix	48
$\mathcal{D}_{AGG}$	squared distance of the AGG	52
$a_i^{(\nu)}$	ν'th diagonal element of A_i	52
r_i	radius of cluster i	57
$\mathcal{D}_{FCS}$	squared distance of the FCS	57
$\mathcal{D}_{FCCS}$	squared distance of the FCCS	57
c_{max}	maximal number of clusters in unsupervised fuzzy clustering	59
c_{best}	optimal number of fuzzy clusters in unsupervised fuzzy clustering	59
$\mathcal{A}_*$	basic clustering algorithm '$*$'	59
$\mathcal{D}_*$	squared distance of algorithm '*'	59
$\mathcal{V}_*$	validity measure '$*$' for unsupervised clustering	59
u_k^{max}	maximal membership degree of one datum	61
$C(x_k)$	class of datum x_k in a calculated partition	61
$C(v_i)$	class of cluster i	61
$SC(x_k)$	class of datum x_k given in the classified sample data	61
$U(l,i)$	sum of membership degrees for sample data with given class l to cluster i	61
u_i^{max}	maximal membership degree of one cluster	61
$\mathcal{V}_{ER}$	validity measure "error rate"	61

$\mathcal{V}_{PC}$	validity measure ”partition coefficient”	62
$\mathcal{V}_{PE}$	validity measure ”partition entropy”	62
$\mathcal{V}_{S}$	validity measure ”separation measure”	63
$\mathcal{V}_{SI}$	validity measure ”separation index”	64
$\mathcal{V}_{FHV}$	validity measure ”fuzzy hypervolume”	64
$\mathcal{V}_{APD}$	validity measure ”average partition density”	65
$\mathcal{V}_{PD}$	validity measure ”partition density”	65
$\mathcal{D}_{NAB}$	squared distance of normalised angle-based clustering algorithm	67
$\mathcal{D}_{AB}$	squared distance for angle-based clustering	73
τ_i	(relative) radius of cluster i	78
$l \in \mathbb{R}_{>0}$	determines the emphasis that is put upon size adaptation in SACB	78
$\tau \in \mathbb{R}_{\geq 1}$	constant predefined constraint parameter for size-adaptable algorithms	78
$\mathcal{D}_{SACB,*}$	squared distance used for size-adaptable centre-based clustering algorithms	78
$\mathcal{D}_{SAVB}$	squared distance used for size-adaptable volume-based clustering	85
$\gamma \in [0,1]$	defines influence of volume-based part in SAVCB	88
e_{is}^{val}	s'th eigenvalue for covariance matrix of cluster i in increasing order	100
e_{is}^{vec}	eigenvector corresponding to e_{is}^{val}	100
$g := v_i + \rho \cdot e_{is}^{vec}$	equation determined by the eigenvector corresponding to the smallest eigenvalue and the cluster centre v_i	100
$\mathcal{D}_{AWFCM}$	squared distance for attribute weighting fuzzy clustering	110
α_{is}	weight for attribute s and cluster i in AWFCM	110
$t \in \mathbb{R}_{>1}$	parameter to influence the strongness of parameter adaptation	110
$\mathcal{D}_{CS}$	squared distance for CS	119
$\mathcal{I} \subseteq 2^{\{1,\ldots,p\}}$	set of sets of coordinates used by the context sensitive clustering distance	120
$I \in \mathcal{I}$	one set of coordinates	120
α_I	influence of coordinate set I on the context sensitive distance	120
μ_l	fuzzy set corresponding to the projection of a cluster to the l'th domain	138
$\xi_1, \ldots, \xi_{r_i - 1}$	input variables for fuzzy rules	138
$\xi_{r_i}, \ldots, \xi_{r_o}$	output variables for fuzzy rules	138
c_i	class assigned to cluster i in classification tasks	138

u_{iks}	membership degree of datum x_k to cluster i for domain s, used for Grid	142
c_s	number of grid coordinates in domain s	142
c_s^{max}	upper bound for the number of grid coordinates in domain s	145
c_s^{min}	lower bound for the number of grid coordinates in domain s	145

Algorithms

Bibliography

[1] S. Abe. Dynamic cluster generation for a fuzzy classifier with ellipsoidal regions. *IEEE Trans. on Systems, Man and Cybernetics - Part B: Cybernetics*, 28(6):869–876, 1998.

[2] S. Abe, R. Thawonmas, and Y. Kobayashi. Feature selection by analyzing class regions approximated by ellipsoids. *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 28(2):282–287, 1998.

[3] A. Amberg, W. Domschke, and S. Voß. Capacitated minimum spanning trees: Algorithms using intelligent search. *Combinatorial Optimization: Theory and Praxis*, pages 9–39, 1996.

[4] H. Angermann. Entwicklung eines Umsteigermodells für das SYSTEM DYNAMICS Passagierflussmodell (in german). Master's thesis, Technische Universität Berlin, Berlin, Fachgebiet Flugführung und Luftverkehr, 2001.

[5] M. Ankerst, M. Breunig, H.-P. Kriegel, and J. Sander. OPTICS: Ordering points to identify the clustering structure. In *ACM SIGMOD'99 Int. Conf. on Management of Data*, Philadelphia, 1999.

[6] N. Ashford, H. P. S. Stanton, and C. A. Moore. *Airport Operations.* McGraw Hill, 2 edition, 1997.

[7] N. Ashford and P. H. Wright. *Airport Engineering.* John Wiley & Sons, Inc., 3 edition, 1992.

[8] T. Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, Oxford, 1996.

[9] H. Bandemer and S. Gottwald. *Einführung in Fuzzy-Methoden (in German).* Akademie Verlag, Berlin, 1993.

[10] H. Bandemer and W. Näther. *Fuzzy Data Analysis.* Theory and Decision Library, Series B: Mathematical and Statistical Methods. Kluwer Academic Publishers, 1992.

[11] J. Beasley, J. Sonander, and P. Havelock. Scheduling aircraft landings at londeon heatzrow using a population heuristic. *Journ. of the Operational Research Society*, 52:483–493, 2001.

[12] T. Beier. Makroskopische Simulation von Passagierströmen am Frankfurter Flughafen, Studienarbeit (in german). Fachhochschule Braunschweig/Wolfenbüttel, Salzgitter, 2001.

[13] M. Berthold. Fuzzy models and potential outliers. In R. N. Davé and T. Sudkamp, editors, *18'th Int. Conf. of the North American Fuzzy Information Processing Society - Nafips*, pages 532–535, New York, USA, 1999.

[14] J. Bezdek. A convergence theorem for the fuzzy isodata clustering algorithms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2:1–8, 1980.

[15] J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Plenum Press, New York, 1981.

[16] J. Bezdek. *Analysis of Fuzzy Information.* CRC Press, Boston, 1987.

[17] J. Bezdek, editor. *Advances in Artificial Intelligence : Applications and Theory.* World Scientific Pub Co, 1990.

[18] J. Bezdek, S. Boggavarapu, L. Hall, and A. Bensaid. Genetic algorithm guided clustering. In *Proc. First IEEE Conf. on Evolutionary computation*, pages 34–38, Orlando, 1994.

[19] J. Bezdek, C. Coray, R. Gunderson, and J. Watson. Detection and characterization of cluster substructure - part i + ii. *SIAM Journal on Applied Mathematics*, 40(2):339–372, 1981.

[20] J. Bezdek, D. Dubois, and H. Prade, editors. *Fuzzy Sets in Approximate Reasoning and Information Systems.* The Handbooks of Fuzzy Sets Series, Fshs 5. Kluwer Academic Publishers, 1999.

[21] J. Bezdek and R. Hathaway. Numerical convergence and interpretation of the fuzzy c-shells clustering algorithm. *IEEE Trans. on Neural Networks*, 3(5):787–793, 1992.

[22] J. Bezdek, R. Hathaway, and N. Pal. Norm-induced shell-prototypes (nisp) clustering. *Neural, Parallel & Scientific Computation*, 3:431–450, 1995.

[23] J. Bezdek, R. Hathaway, M. Sabin, and W. Tucker. Convergence theory for fuzzy c-means: Counterexamples and repairs. *IEEE Trans. Systems, Man, and Cybernetics*, 17:873–877, 1987.

[24] J. Bezdek, J. Keller, R. Krishnapuram, and N. Pal. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, volume 4 of *The Handbooks of Fuzzy Sets.* Kluwer Academic Publishers, New York, 1999.

[25] J. Bezdek and N. Pal. Some new indexes of cluster validity. *IEEE Trans. on Systems, Man and Cybernetics - Part B: Cybernetics*, 28(3):301–315, 1998.

[26] C. Blake, E. Keogh, and C. Merz. Uci repository of machine learning databases. URL: http://www.ics.uci.edu/˜mlearn/MLRepository.html. University of California, Irvine, Dept. of Information and Computer Sciences, 1998.

[27] A. L. Blumel, E. J. Hughes, and B. A. White. Design of robust fuzzy controllers for aerospace applications. In R. N. Davé and T. Sudkamp, editors, *18'th Int. Conf. of the North American Fuzzy Information Processing Society - Nafips*, pages 438–442, New York, USA, 1999.

[28] H. H. Bock. Clusteranalyse mit unscharfen partitionen (in german). In H. Bock, editor, *Klassifikation und Erkenntnis: Vol. III: Numerische Klassifikation*, pages 137–163. INDEKS, Frankfurt, 1979.

[29] Bundesamt für Zivilluftfahrt. Aeronautical Information Publication – AIP Switzerland. Bundesamt für Zivilluftfahrt, CH-3003 Bern.

[30] S. Cafiso and V. Cutello. A fuzzy model for road accidents analysis. In R. N. Davé and T. Sudkamp, editors, *18'th Int. Conf. of the North American Fuzzy Information Processing Society - Nafips*, pages 139–143, New York, USA, 1999.

[31] V. Chepoi and D. Dumitrescu. Fuzzy clustering with structural constraints. *Fuzzy Sets and Systems*, 105:91–97, 1999.

[32] S. Das and B. A. Bowles. Simulations of highway chaos using fuzzy logic. In R. N. Davé and T. Sudkamp, editors, *18'th Int. Conf. of the North American Fuzzy Information Processing Society - Nafips*, New York, USA, 1999.

[33] R. Davé. Fuzzy shell clustering and application to circle detection in digital images. *Intern. Journ. General Systems*, 16:343–355, 1990.

[34] R. Davé. Characterization and detection of noise in clustering. *Pattern Recognition Letters*, 12:657–664, 1991.

[35] R. Davé and R. Krishnapuram. Robust clustering methods: A unified view. *IEEE Transactions on Fuzzy Sets and Systems*, 5(2):270–293, 1997.

[36] R. Davé and S. Sen. On generalizing the noise clustering algorithms. In *Proc. 7th Intern. Fuzzy Systems Association World Congress (IFSA'97)*, volume III, pages 205–210, Prague, 1997. Academia.

[37] M. Delgado and A. Gòmez-Skarmeta. Learning fuzzy systems using fuzzy clustering. In *7th Intern. Fuzzy Systems Association World Congress (IFSA'97)*, Prague, 1997. Academia.

[38] P. S. Dempsey. *Airport planning and development handbook: a global survey.* Mc Graw Hill, 1999.

[39] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[40] J. Dunn. A fuzzy relative of the isodata process and its use in detecting compact, well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

[41] H. Frigui and R. Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):450–465, 1999.

[42] I. Gath and A. Geva. Unsupervised optimal fuzzy clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11:773–781, 1989.

[43] I. Gath and D. Hoory. Fuzzy clustering of elliptic ring-shaped clusters. *Pattern Recognition Letters*, 16:727–741, 1995.

[44] H. Genther and M. Glesner. Automatic generation of a fuzzy classification system using fuzzy clustering methods. In *Proc. ACM Symposium on Applied Computing (SAC'94)*, pages 180–183, Phoenix, 1994.

[45] I. Gerdes, F. Klawonn, R. Kruse, and M. Schröder. *Evolutionäre Algorithmen. Ihre Kopplung mit Fuzzy-Systemen und Neuronalen Netzen. (in German).* Vieweg, Wiesbaden, 2000.

[46] F. Glover. Tabu search – part i. *ORSA Journal on Computing*, 1:190–206, 1989.

[47] F. Glover. Tabu search – part ii. *ORSA Journal on Computing*, 2:4–32, 1990.

[48] D. Gustafson and W. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Proc. IEEE CDC*, pages 761–766, San Diego, 1979.

[49] L. Hall, B. Ozyurt, and J. Bezdek. The case for genetic algorithms in fuzzy clustering. In *7th Intern. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU)*, pages 288–295, Paris, 1998.

[50] R. Hathaway and J. Bezdek. Fuzzy c-means clustering of incomplete data. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 31(5):735–744, 2001.

[51] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis.* Wiley, Chichester, 1999.

[52] H. Inoue, K. Kamei, and K. Inoue. Automatic generation of fuzzy rules using hyper elliptic cone membership function by genetic algorithms. In *Proc. 7th Intern. Fuzzy Systems Association World Congress (IFSA'97)*, volume II, pages 383–388, Prague, 1997. Academia.

[53] A. Keller. Regelerzeugung mit Fuzzy-Datenanalyse (in german). Master's thesis, Technische Universität Braunschweig, Braunschweig, Institut für Betriebssysteme und Rechnerverbund, 1997.

[54] A. Keller. Fuzzy clustering with outliers. In T. Whalen, editor, *PeachFuzz 2000, 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS)*, pages 143–147, Atlanta, 2000.

[55] A. Keller and F. Klawonn. Generating classification rules by grid clustering. In R. Felix, editor, *Proc. Third European Workshop on Fuzzy Decision Analysis and Neural Networks for Management, Planning and Optimization (EFDAN'98)*, pages 113–121, Dortmund, 1998.

[56] A. Keller and F. Klawonn. Regelerzeugung mit Fuzzy-Clusteranalyse (in german). In R. Kruse and G. Saake, editors, *GI-Jahrestagung 1998*, pages 119–130, Magdeburg, 1998.

[57] A. Keller and F. Klawonn. Clustering with volume adaptation for rule learning. In H. Zimmermann, editor, *Proc. 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*, page 215, Aachen, 1999.

[58] A. Keller and F. Klawonn. Context sensitive fuzzy clustering. In R. N. Davé and T. Sudkamp, editors, *Proc. 18th Intern. Conf. of the North American Fuzzy Information Processing Society - NAFIPS*, pages 347–351, Piscataway, NJ, 1999. IEEE.

[59] A. Keller and F. Klawonn. Fuzzy clustering with weighting of data variables. In *Proc. 1999 Eusflat-Estylf Joint Conference*, pages 497–500, Palma de Mallorca, 1999.

[60] A. Keller and F. Klawonn. Fuzzy clustering with weighting of data variables. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 8:735–746, 2000.

[61] A. Keller and F. Klawonn. Adaptation of cluster sizes in objective function based fuzzy clustering. In T. Leondes, editor, *Intelligent Systems: Techniques and Applications - Database and Learning Systems*, volume 4, pages 181–191. CRC Press, 2002.

[62] P. R. Kersten. Implementation issues in the fuzzy c-median clustering algorithms. In *IEEE 6th Int. Conf. on Fuzzy Systems (FUZZ-IEEE'97)*, pages 957–962, Barcelona, Spain, 1997.

[63] Y. Kharin. Robustness of clustering under outliers. In X. Liu, P. Cohen, and M. Berthold, editors, *Advances in Intelligent Data Analysis*, pages 501–511. Springer-Verlag, Berlin, 1997.

[64] F. Klawonn and A. Keller. Fuzzy clustering and fuzzy rules. In *Proc. 7th Intern. Fuzzy Systems Association World Congress (IFSA'97)*, volume I, pages 193–198, Prague, 1997. Academia.

[65] F. Klawonn and A. Keller. Fuzzy clustering with evolutionary algorithms. *Intern. Journ. of Intelligent Systems*, 13:975–991, 1998.

[66] F. Klawonn and A. Keller. Grid clustering for generating fuzzy rules. In *Proc. 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT'98)*, pages 1365–1369, Aachen, 1998.

[67] F. Klawonn and A. Keller. Learning fuzzy rules from data. In J. Dix and S. Hölldobler, editors, *Inference Mechanisms in Knowledge-Based Systems: Theory and Applications (Proc. KI'98)*, pages 60–74, Koblenz, 1998.

[68] F. Klawonn and A. Keller. Fuzzy clustering based on modified distance measures. In D. Hand, J. Kok, and M. Berthold, editors, *Advances in Intelligent Data Analysis*, pages 291–301, Berlin, 1999. Springer-Verlag.

[69] F. Klawonn and E.-P. Klement. Mathematical analysis of fuzzy classifiers. In X. Liu, P. Cohen, and M. Berthold, editors, *Advances in Intelligent Data Analysis*, pages 359–370. Springer-Verlag, Berlin, 1997.

[70] F. Klawonn and R. Kruse. Automatic generation of fuzzy controllers by fuzzy clustering. In *1995 IEEE Intern. Conference on Systems, Man, and Cybernetics*, pages 2040–2045, Vancouver, 1995.

[71] F. Klawonn and R. Kruse. Clustering methods in fuzzy control. In W. Gaul and D. Pfeifer, editors, *From Data to Knowledge: Theoretical and Practical Aspects of Classification, Data Analysis and Knowledge Organization.*, pages 195–202. Springer-Verlag, Berlin, 1995.

[72] F. Klawonn and R. Kruse. Derivation of fuzzy classification rules from multidimensional data. In G. Lasker and X. Liu, editors, *Advances in Intelligent Data Analysis*, pages 90–94. The International Institute for Advanced Studies in Systems Research and Cybernetics, Windsor, Ontario, 1995.

[73] F. Klawonn and R. Kruse. Constructing a fuzzy controller from data. *Fuzzy Sets and Systems*, 85:177–193, 1997.

[74] U. Klee. *jp airline-fleets international 2000/01*. Bucher & Co., 2000.

[75] K. Krishna and M. Murty. Genetic k-means algorithm. *IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics*, 29(3):433–439, 1999.

[76] K. Krishnapuram and C. Freg. Fitting an unknown number of lines and planes to image data through compatible cluster merging. *Pattern Recognition*, 25:385–400, 1992.

[77] R. Krishnapuram and L. Chen. Implementation of parallel thinning algorithms using recurrent neural networks. *IEE Trans. Neural Networks*, 4:142–147, 1993.

[78] R. Krishnapuram, H. Frigui, and O. Nasraoui. Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation - part 1 & 2. *IEEE Trans. on Fuzzy Systems*, 3:29–60, 1995.

[79] R. Krishnapuram and J. Keller. A possibilistic approach to clustering. *IEEE Trans. on Fuzzy Systems*, 1:98–110, 1993.

[80] R. Krishnapuram and J. Kim. A note on the gustafson-kessel and adaptive fuzzy clustering algorithms. *IEEE Transactions on Fuzzy Systems*, 7(4):453–461, 1999.

[81] R. Krishnapuram, O. Nasraoui, and H. Frigui. The fuzzy c spherical shells algorithm: A new approach. *IEEE Trans. on Neural Networks*, 3:663–671, 1992.

[82] L. Kuncheva. How good are fuzzy if-then classifiers? *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 30(4):501–509, 2000.

[83] D. Lisin and M. A. Gennert. Optimal function approximation using fuzzy rules. In R. N. Davé and T. Sudkamp, editors, *18'th Int. Conf. of the North American Fuzzy Information Processing Society - Nafips*, pages 184–188, New York, USA, 1999.

[84] W. Liu, A. White, S. Thompson, and M. Bramer. Techniques for dealing with missing values in classification. In X. Liu, P. Cohen, and M. Berthold, editors, *Advances in Intelligent Data Analysis*, pages 527–536. Springer-Verlag, Berlin, 1997.

[85] S. Lorenz. Erstellung eines Flugplangenerators (in german). Master's thesis, Technische Universität Dresden, Dresden, Institut für Luftfahrt, 1999.

[86] Lufthansa. Lufthansa fleet. URL: http://www.lufthansa-financials.de/english/ir/combpany/flotte_b.htm, 2001.

[87] E. Mamdani and B. Gaines. *Fuzzy Reasoning and its Applications.* Academic Press, London, 1981.

[88] Y. Man and I. Gath. Detection and separation of ring-shaped clusters using fuzzy clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16:855–861, 1994.

[89] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer, Berlin, 1992.

[90] B. Mirkin. *Mathematical Classification and Clustering*, volume 11 of *Nonconvex Optimization and Its Applications.* Kluwer Academic Publishers, Dordrecht, 1996.

[91] T. Mühlhausen. Ein Beitrag zur makroskopischen Simulation von Passagierströmen zwischen kooperierenden Flughäfen unter Nutzung des SYSTEM DYNAMICS Zuganges nach Forrester (in german). URL: http://hsss.slub-dresden.de/hsss/servlet/hsss.urlmapping.MappingServlet?id= 993202531468-5970. Technische Universität Dresden, Dresden, Institut für Verkehrsinformationssysteme, 1999.

[92] S. Nascimento and F. Moura-Pires. A genetic approach to fuzzy clustering with a validity measure fitness function. In X. Liu, P. Cohen, and M. Berthold, editors, *Advances in Intelligent Data Analysis*, pages 325–335. Springer, Berlin, 1997.

[93] D. Nauck, F. Klawonn, and R. Kruse. *Neuro-Fuzzy Systems*. Wiley, Chichester, 1997.

[94] A. Nürnberger, A. Klose, and R. Kruse. Discussing cluster shapes of fuzzy classifiers. In R. N. Davé and T. Sudkamp, editors, *Nafips*, pages 546–550, New York, 1999.

[95] Y. Ohashi. Fuzzy clustering and robust estimation. In *9th Meeting SAS Users Group Int.*, Hollywood Beach, Fl., 1984.

[96] C. Pschierer. Civil aviation database (in german). URL:http://cip.physik.uni-wuerzburg.de/˜pschirus/aviation/, 2001.

[97] P. Rousseeuw. Discussion: Fuzzy clustering at the intersection. *TECHNOMETRICS*, 37(3):283–286, 1995.

[98] P. Rousseeuw, L. Kaufmann, and E. Trauwaeert. Fuzzy clustering using scatter matrices. *Computational Statistics and Data Analysis*, 23:135–151, 1996.

[99] T. Runkler and J. C. Bezdek. Alternating cluster estimation: A new tool for clustering and function approximation. *IEEE Transactions on Fuzzy Systems*, 7(4):377–393, 1999.

[100] E. Schikuta and M. Erhart. The bang-clustering system: Grid-based data analysis. In X. Liu, P. Cohen, and M. Berthold, editors, *Advances in Intelligent Data Analysis*, pages 513–524. Springer-Verlag, Berlin, 1997.

[101] S. Z. Selim and M. A. Ismail. Soft clustering of multidimensional data: A semi-fuzzy approach. *Pattern Recognition*, 17(5):559–568, 1984.

[102] S. Sen and R. N. Davé. Application of noise clustering in group technology. In R. N. Davé and T. Sudkamp, editors, *18'th Int. Conf. of the North American Fuzzy Information Processing Society - Nafips*, pages 366–370, New York, USA, 1999.

[103] M. Setnes, R. Babuska, U. Kaymak, and H. R. v. N. Lemke. Similarity measures in fuzzy rule base simplification. *IEEE Trans. on Systems, Man and Cybernetics - Part B: Cybernetics*, 28(3):376–386, 1998.

[104] M. Setnes and U. Kaymak. Extended fuzzy c-means with volume prototypes and cluster merging. In *EUFIT'98*, pages 1360–1364, Aachen, 1998.

[105] M. Setnes and H. Roubos. Transparent fuzzy modeling using fuzzy clustering. In R. N. Davé and T. Sudkamp, editors, *NAFIPS*, pages 198–202, New York, USA, 1999.

[106] C. Stütz. *Anwendungsspezifische Fuzzy-Clustermethoden (in German).* Ph.d., Universität München, 1999.

[107] M. Sugeno and T. Yasukawa. A fuzzy-logic-based approach to qualitative modelling. *IEEE Trans. on Fuzzy Systems*, 1:7–31, 1993.

[108] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Trans. on Systems, Man, and Cybernetics*, 15:116–132, 1985.

[109] H. Timm and F. Klawonn. Different approaches for fuzzy cluster analysis with missing values. In *EUFIT99: 7th European Congress on Intelligent Techniques & Soft Computing*, pages 177–178, Aachen, Germany, 1999. Verlag Mainz GmbH, Aachen.

[110] E. Trauwaert, L. Kaufmann, and P. Rousseeuw. Fuzzy clustering algorithms based on the maximum likelihood principle. *Fuzzy Sets and Systems*, 42:213–227, 1991.

[111] M. Turhan. Genetic fuzzy clustering by means of discovering membership functions. In X. Liu, P. Cohen, and M. Berthold, editors, *Advances in Intelligent Data Analysis*, pages 383–393. Springer, Berlin, 1997.

[112] W. H. Wolberg and O. L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. In *National Academy of Sciences*, volume 87, pages 9193–9196, USA, 1990.

[113] X. L. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):841–847, 1991.

[114] Y. Yoshinari, W. Pedrycz, and K. Hirota. Construction of fuzzy models through clustering techniques. *Fuzzy Sets and Systems*, 54:157–165, 1993.

[115] L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

Index