OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

INF FAKULTÄT FÜR INFORMATIK

Fuzzy Systems

Fuzzy Clustering 2

Prof. Dr. Rudolf Kruse **Christoph Doell**

{kruse,doell}@ovgu.de
Otto-von-Guericke University of Magdeburg
Faculty of Computer Science
Institute for Intelligent Cooperating Systems

Outline

1. **Possibilistic c-means**

    Comparison of FCM and PCM

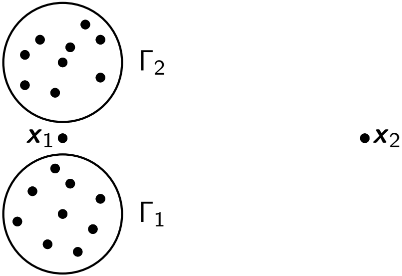
2. **Distance Function Variants**

3. **Objective Function Variants**

4. **Cluster Validity**

5. **Example: Transfer Passenger Analysis**

Problems with Probabilistic c-means



$\boldsymbol{x}_1$ has the same distance to Γ_1 and $\Gamma_2 \Rightarrow \mu_{\Gamma_1}(\boldsymbol{x}_1) = \mu_{\Gamma_2}(\boldsymbol{x}_1) = 0.5$.

The same degrees of membership are assigned to $\boldsymbol{x}_2$.

This problem is due to the normalization.

A better reading of memberships is "If $\boldsymbol{x}_j$ must be assigned to a cluster, then with probability u_{ij} to Γ_i".

Problems with Probabilistic c-means

The normalization of memberships is a problem for noise and outliers.

A fixed data point weight causes a high membership of noisy data, although there is a large distance from the bulk of the data.

This has a bad effect on the clustering result.

Dropping the normalization constraint

$$\sum_{i=1}^{c} u_{ij} = 1, \quad \forall j \in \{1, \ldots, n\},$$

we obtain more intuitive membership assignments.

Possibilistic Cluster Partition

Definition

Let $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be the set of given examples and let c be the number of clusters $(1 < c < n)$ represented by the fuzzy sets $\mu_{\Gamma_i}, (i = 1, \ldots, c)$. Then we call $U_p = (u_{ij}) = (\mu_{\Gamma_i}(\boldsymbol{x}_j))$ a *possibilistic cluster partition* of X if

$$\sum_{j=1}^{n} u_{ij} > 0, \quad \forall i \in \{1, \ldots, c\}$$

holds. The $u_{ij} \in [0, 1]$ are interpreted as degree of representativity or typicality of the datum $\boldsymbol{x}_j$ to cluster Γ_i.

now, u_{ij} for $\boldsymbol{x}_j$ resemble possibility of being member of corresponding cluster

Possibilistic Fuzzy Clustering

J_f is not appropriate for possibilistic fuzzy clustering.

Dropping the normalization constraint leads to a minimum for all $u_{ij} = 0$.

Thus is, data points are not assigned to any Γ_i. Thus all Γ_i are empty.

Hence a penalty term is introduced which forces all u_{ij} away from zero.

The objective function J_f is modified to

$$J_p(X, U_p, C) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} (1 - u_{ij})^m$$

where $\eta_i > 0 (1 \leq i \leq c)$.

The values η_i balance the contrary objectives expressed in J_p.

Optimizing the Membership Degrees

The update formula for membership degrees is

$$u_{ij} = \frac{1}{1 + \left(\frac{d_{ij}^2}{\eta_i}\right)^{\frac{1}{m-1}}}.$$

The membership of $\boldsymbol{x}_j$ to cluster i depends only on d_{ij} to this cluster.

A small distance corresponds to a high degree of membership.

Larger distances result in low membership degrees.

So, u_{ij}'s share a typicality interpretation.

Interpretation of η_i

The update equation helps to explain the parameters η_i.

Consider $m = 2$ and substitute η_i for d_{ij}^2 yields $u_{ij} = 0.5$.

Thus η_i determines the distance to Γ_i at which u_{ij} should be 0.5.

η_i can have a different geometrical interpretation:

- the hyperspherical clusters (*e.g.* PCM), thus $\sqrt{\eta_i}$ is the mean diameter.

Estimating η_i

If such properties are known, η_i can be set a priori.

If all clusters have the same properties, the same value for all clusters should be used.

However, information on the actual shape is often unknown a priori.

- So, the parameters must be estimated, *e.g.* by FCM.
- One can use the fuzzy intra-cluster distance, *i.e.* for all $\Gamma_i,\ 1 \leq i \leq n$

$$\eta_i = \frac{\sum_{j=1}^{n} u_{ij}^m d_{ij}^2}{\sum_{j=1}^{n} u_{ij}^m}.$$

Optimizing the Cluster Centers

The update equations j_C are derived by setting the derivative of J_p *w.r.t.* the prototype parameters to zero (holding U_p fixed).

The update equations for the cluster prototypes are identical.

Then the cluster centers in the PCM algorithm are re-estimated as

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^{n} u_{ij}^m}.$$

Revisited Example: The Iris Data

© **Iris Species Database** `http://www.badbear.com/signa/`

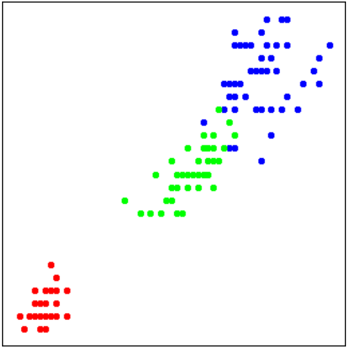

Iris setosa


Iris versicolor


Iris virginica

Collected by Ronald Aylmer Fischer (famous statistician).
150 cases in total, 50 cases per Iris flower type.
Measurements: sepal length/width, petal length/width (in cm).
Most famous dataset in pattern recognition and data analysis.

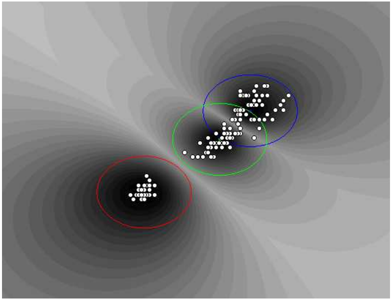
Example: The Iris Data

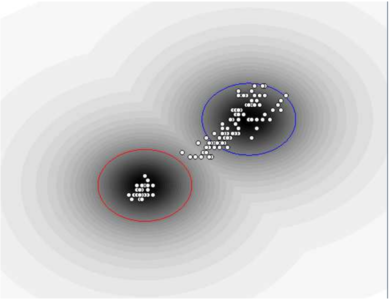


Shown: sepal length and petal length.

Iris setosa (red), Iris versicolor (green), Iris virginica (blue)

Comparison of FCM and PCM





FCM (left) and PCM (right) of Iris dataset into 3 clusters.

FCM divides space, PCM depends on typicality to closest clusters.

FCM and PCM divide dataset into 3 and 2 clusters, resp.

- This behavior is specific to PCM.
- FCM drives centers apart due to normalization, PCM does not.

Cluster Coincidence

characteristic	FCM	PCM
data partition	exhaustively forced to	not forced to
membership degr.	distributed	determined by data
cluster interaction	covers whole data	non
intra-cluster dist.	high	low
cluster number c	exhaustively used	upper bound

Clusters can coincide and might not even cover data.

PCM tends to interpret low membership data as outliers.

A better coverage obtained by

- using FCM to initialize PCM (*i.e.* prototypes, η_i, c),
- after 1st PCM run, re-estimate η_i again,
- then use improved estimates for 2nd PCM run as final solution.

Cluster Repulsion I

J_p is truly minimized only if all cluster centers are identical.

Other results are achieved when PCM gets stuck in a local minimum.

PCM can be improved by modifying J_p:

$$J_{rp}(X, U_p, C) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m d_{ij}^2 + \sum_{i=1}^{c} \eta_i \sum_{j=1}^{n} (1 - u_{ij})^m + \sum_{i=1}^{c} \gamma_i \sum_{k=1, k \neq i}^{c} \frac{1}{\eta d(\mathbf{c}_i, \mathbf{c}_k)^2}.$$

γ_i controls the strength of the cluster repulsion.

η makes the repulsion independent of normalization of data attributes.

Cluster Repulsion II

The minimization conditions lead to the update equation

$$\mathbf{c}_i = \frac{\sum_{j=1}^{n} u_{ij}^m \mathbf{x}_j - \gamma_i \sum_{k=1, k\neq i}^{c} \frac{1}{d(\mathbf{c}_i, \mathbf{c}_k)^4} \mathbf{c}_k}{\sum_{j=1}^{n} u_{ij}^m - \gamma_i \sum_{k=1, k\neq i}^{c} \frac{1}{d(\mathbf{c}_i, \mathbf{c}_k)^4}}.$$

This equation shows an effect of the repulsion between clusters:

- A cluster is attracted by data assigned to it.
- It is simultaneously repelled by other clusters.

The update equation of PCM for membership degrees is not modified.

It yields a better detection of shape of very close or overlapping clusters.

Recognition of Positions and Shapes

Possibilistic models do not only carry problematic properties.

The cluster prototypes are more intuitive:

- The memberships depend only on the distance to one cluster.

Shape & size of clusters better fit data clouds than with FCM.

- They are less sensitive to outliers and noise.
- This is an attractive tool in image processing.

Outline

Distance Function Variants

So far, only Euclidean distance leading to standard FCM and PCM

Euclidean distance only allows spherical clusters

Several variants have been proposed to relax this constraint

- fuzzy Gustafson-Kessel algorithm
- fuzzy shell clustering algorithms
- kernel-based variants

Can be applied to FCM and PCM

Gustafson-Kessel Algorithm

[Gustafson and Kessel, 1979] replaced Euclidean distance by cluster-specific Mahalanobis distance

For cluster Γ_i, its associated Mahalanobis distance is defined as

$$d^2(\boldsymbol{x}_j, C_j) = (\boldsymbol{x}_j - \boldsymbol{c}_i)^T \Sigma_i^{-1} (\boldsymbol{x}_j - \boldsymbol{c}_i)$$

where Σ_i is covariance matrix of cluster

Euclidean distance leads to $\forall i : \Sigma_i = I$, *i.e.* identity matrix

Gustafson-Kessel (GK) algorithm leads to prototypes $C_i = (\boldsymbol{c}_i, \Sigma_i)$

Gustafson-Kessel Algorithm

Specific constraints can be taken into account, *e.g.*

- restricting to axis-parallel cluster shapes
- by considering only diagonal matrices
- usually preferred when clustering is applied for fuzzy rule generation

Cluster sizes can be controlled by $\varrho_i > 0$ demanding $\det(\Sigma_i) = \varrho_i$

Usually clusters are equally sized by $\det(\Sigma_i) = 1$

Objective Function

Identical to FCM and PCM: J, update equations for c_i and U

Update equations for covariance matrices are

$$\Sigma_i = \frac{\Sigma_i^*}{\sqrt[p]{\det(\Sigma_i^*)}}$$

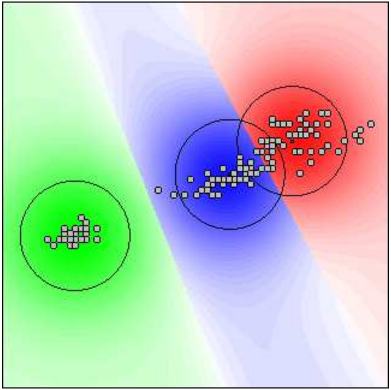
where

$$\Sigma_i^* = \frac{\sum_{j=1}^n u_{ij}(\boldsymbol{x}_j - \boldsymbol{c}_i)(\boldsymbol{x}_j - \boldsymbol{c}_i)^T}{\sum_{j=1}^n u_{ij}}$$

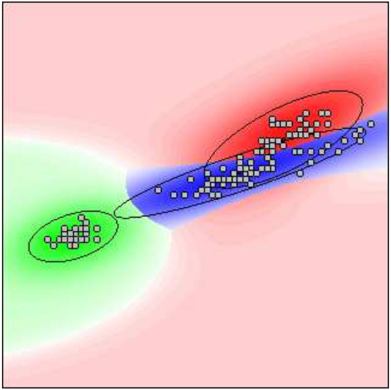
Covariance of data assigned to cluster i

Σ_i are modified to incorporate fuzzy assignment

Fuzzy Clustering of the Iris Data



Fuzzy c-Means



Gustafson-Kessel

Summary: Gustafson-Kessel

Extracts more information than standard FCM and PCM

More sensitive to initialization

Recommended initializing: few runs of FCM or PCM

Compared to FCM or PCM: due to matrix inversions GK is

- computationally costly
- hard to apply to huge datasets

Restriction to axis-parallel clusters reduces computational costs

Fuzzy Shell Clustering

Up to now: searched for convex "cloud-like" clusters

Corresponding algorithms = **solid clustering** algorithms

Especially useful in data analysis

For image recognition and analysis:
variants of FCM and PCM to detect lines, circles or ellipses

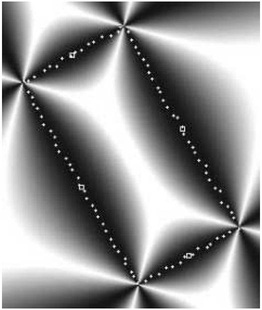
shell clustering algorithms

replace Euclidean by other distances

Fuzzy c-varieties Algorithm

Fuzzy c-varieties (FCV) algorithm recognizes lines, planes, or hyperplanes

Each cluster is affine subspace characterized by point and set of orthogonal unit vectors, $C_i = (\mathbf{c}_i, \mathbf{e}_{i1}, \ldots, \mathbf{e}_{iq})$ where q is dimension of affine subspace



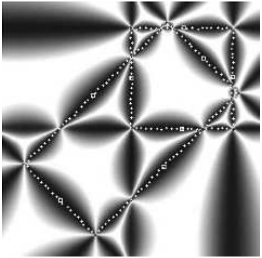
Distance between data point $\mathbf{x}_j$ and cluster i

$$d^2(\mathbf{x}_j, \mathbf{c}_i) = \|\mathbf{x}_j - \mathbf{c}_i\|^2 - \sum_{l=1}^{q} (\mathbf{x}_j - \mathbf{c}_i)^T \mathbf{e}_{il}$$

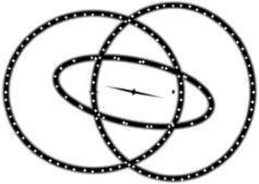
Also used for locally linear models of data with underlying functional interrelations

Other Shell Clustering Algorithms

Name	Prototypes
adaptive fuzzy c-elliptotypes (AFCE)	line segments
fuzzy c-shells	circles
fuzzy c-ellipsoidal shells	ellipses
fuzzy c-quadric shells (FCQS)	hyperbolas, parabolas
fuzzy c-rectangular shells (FCRS)	rectangles



AFCE



FCQS

FCRS

Kernel-based Fuzzy Clustering

Kernel variants modify distance function to handle non-vectorial data, *e.g.* sequences, trees, graphs

Kernel methods [Schölkopf and Smola, 2001] extend classic linear algorithms to non-linear ones without changing algorithms

Data points can be vectorial or not $\Rightarrow x_j$ instead of $\boldsymbol{x}_j$

Kernel methods: based on mapping $\phi : \mathcal{X} \to \mathcal{H}$

Input space $\mathcal{X}$, feature space $\mathcal{H}$ (higher or infinite dimensions)

$\mathcal{H}$ must be Hilbert space, *i.e.* dot product is defined

Principle

Data are not handled directly in $\mathcal{H}$, only handled by dot products

Kernel function

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, \forall x, x' \in \mathcal{X} : \langle \phi(x), \phi(x') \rangle = k(x, x')$$

No need to known ϕ explicitly

Scalar products in $\mathcal{H}$ only depend on k and data $\Rightarrow$ **kernel trick**

Kernel methods = algorithms with scalar products between data

Kernel Fuzzy Clustering

Kernel framework has been applied to fuzzy clustering

Fuzzy shell clustering extracts prototypes, kernel methods do not

They compute similarity between $x, x' \in \mathcal{X}$

Clusters: no explicit representation

Kernel variant of FCM [Wu et al., 2003] transposes J_f to $\mathcal{H}$

Centers $c_i^\phi \in \mathcal{H}$ are linear combinations of transformed data

$$c_i^\phi = \sum_{r=1}^{n} a_{ir} \phi(x_r)$$

Kernel Fuzzy Clustering

Euclidean distance between points and centers in $\mathcal{H}$ is

$$d_{\phi_{ir}}^2 = \left\| \phi(x_r) - c_i^\phi \right\|^2 = k_{rr} - 2\sum_{s=1}^{n} a_{is} k_{rs} + \sum_{s,t=1}^{n} a_{is} a_{it} k_{st}$$

whereas $k_{rs} \equiv k(x_r, x_s)$

Objective function becomes

$$J_\phi(X, U_\phi, C) = \sum_{i=1}^{c} \sum_{r=1}^{n} u_{ir}^m d_{\phi_{ir}}^2$$

Minimization leads to update equations:

$$u_{ir} = \frac{1}{\sum_{l=1}^{c} \left(\frac{d_{\phi_{ir}}^2}{d_{\phi_{lr}}^2} \right)^{\frac{1}{m-1}}}, \quad a_{ir} = \frac{u_{ir}^m}{\sum_{s=1}^{n} u_{is}^m}, \quad c_i^\phi = \frac{\sum_{r=1}^{n} u_{ir}^m \phi(x_r)}{\sum_{s=1}^{n} u_{is}^m}$$

Summary: Kernel Fuzzy Clustering

Update equations (and J_ϕ) are expressed by k

For Euclidean distance, membership degrees are identical to FCM

Cluster centers: weighted mean of data (comparable to FCM)

Disadvantage of kernel methods:

- choice of proper kernel and its parameters
- similar to feature selection and data representation
- cluster centers belong to $\mathcal{H}$ (no explicit representation)
- only weighting coefficients a_{ir} are known

Outline

Objective Function Variants

So far, variants of FCM with different distance functions

Now, other variants based on modifications of J

Aim: improving clustering results, *e.g.* noisy data

Many different variants:

- explicitly handling noisy data
- modifying fuzzifier m in objective function
- new terms in objective function (*e.g.* optimize cluster number)
- improving PCM *w.r.t.* coinciding cluster problem

Noise Clustering

Noise clustering (NC) adds to c clusters one noise cluster

- shall group noisy data points or outliers
- not explicitly associated to any prototype
- directly associated to distance between implicit prototype and data

Center of noise cluster has constant distance δ to all data points

- all points have same "probability" of belonging to noise cluster
- during optimization, "probability" is adapted

Noise Clustering

Noise cluster: added to objective function as any other cluster

$$J_{nc}(X, U, C) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2} + \sum_{k=1}^{n} \delta^2 \left(1 - \sum_{i=1}^{c} u_{ik}\right)^{m}$$

Added term: similar to terms in first sum

- distance to cluster prototype is replaced by δ
- outliers can have low membership degrees to standard clusters

J_{nc} requires setting of parameter δ, *e.g.*

$$\delta = \lambda \frac{1}{c \cdot n} \sum_{i=1}^{c} \sum_{j=1}^{n} d_{ij}^{2}$$

λ user-defined parameter: if low λ, then high number of outliers

Fuzzifier Variants

Fuzzifier m introduces problem:

$$u_{ij} = \begin{cases} \{0, 1\} & \text{if } m = 1, \\ ]0, 1[& \text{if } m > 1 \end{cases}$$

Disadvantage for noisy datasets (to be discussed in the exercise)

Possible solution: convex combination of hard and fuzzy c-means

$$J_{hf}(X, U, C) = \sum_{i=1}^{c} \sum_{j=1}^{n} \left[\alpha u_{ij} + (1 - \alpha) u_{ij}^2 \right] d_{ij}^2$$

where $\alpha \in [0, 1]$ is user-defined threshold

Outline

Problems with Fuzzy Clustering

What is optimal number of clusters c?

Shape and location of cluster prototypes: not known a priori $\Rightarrow$ initial guesses needed

Must be handled: different data characteristics, *e.g.* variabilities in shape, density and number of points in different clusters

Cluster Validity for Fuzzy Clustering

Idea: each data point has c memberships

Desirable: summarize information by single criterion indicating how well data point is classified by clustering

Cluster validity: average of any criteria over entire data set

"good" clusters are actually not very fuzzy!

Criteria for definition of "optimal partition" based on:

- clear separation between resulting clusters
- minimal volume of clusters
- maximal number of points concentrated close to cluster centroid

Judgment of Classification by Validity Measures

Validity measures can be based on several criteria, *e.g.*

membership degrees should be $\approx 0/1$, *e.g.* **partition coefficient**

$$\mathrm{PC} = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^2$$

Compactness of clusters, *e.g.* **average partition density**

$$\mathrm{APD} = \frac{1}{c}\sum_{i=1}^{c}\frac{\sum_{j\in Y_i} u_{ij}}{\sqrt{|\Sigma_i|}}$$

where $Y_i = \{j \in \mathbb{N},\ j \le n \mid (\mathbf{x}_j - \boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_j - \boldsymbol{\mu}_i) < 1\}$

especially for FCM: **partition entropy**

$$\mathrm{PE} = -\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}\log u_{ij}$$

Outline

Example: Transfer Passenger Analysis
[Keller and Kruse, 2002]

German Aerospace Center (DLR) developed macroscopic passenger flow model for simulating passenger movements on airport's land side

For passenger movements in terminal areas: distribution functions are used today

Goal: build fuzzy rule base describing transfer passenger amount between aircrafts

These rules can be used to improve macroscopic simulation

Idea: find rules based on probabilistic fuzzy c-means (FCM)

Attributes for Passenger Analysis

Maximal amount of passengers in certain aircraft (depending on type of aircraft)

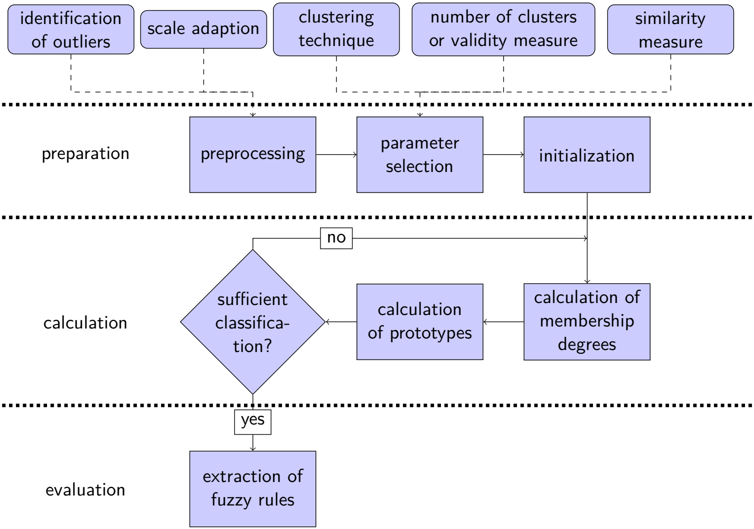
Distance between airport of departure and airport of destination (in three categories: short-, medium-, and long-haul)

Time of departure

Percentage of transfer passengers in aircraft

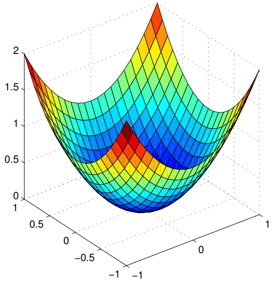
FAKULTÄT FÜR INFORMATIK

General Clustering Procedure

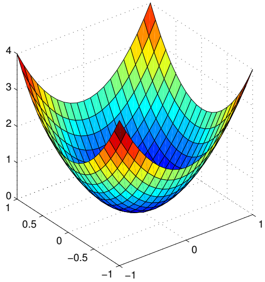


Distance Measure

distance between $\boldsymbol{x} = (x_1, x_2)$ and $\boldsymbol{c} = (0, 0)$



$$d^2(\boldsymbol{c}, \boldsymbol{x}) = \|\boldsymbol{c} - \boldsymbol{x}\|^2$$



$$d_\tau^2(\boldsymbol{c}, \boldsymbol{x}) = \frac{1}{\tau^p} \|\boldsymbol{c} - \boldsymbol{x}\|^2$$

Distance Measure with Size Adaption

$$d_{ij}^2 = \frac{1}{\tau_i^p} \cdot \|\mathbf{c}_i - \mathbf{x}_j\|^2$$

$$\mathbf{c}_i = \frac{\sum_{j=1}^n u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^n u_{ij}^m}$$

$$\tau_i = \frac{\left(\sum_{j=1}^n u_{ij}^m d_{ij}^2\right)^{\frac{1}{p+1}}}{\sum_{k=1}^c \left(\sum_{j=1}^n u_{kj}^m d_{kj}^2\right)^{\frac{1}{p+1}}} \cdot \tau$$

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

p determines emphasis put on size adaption during clustering

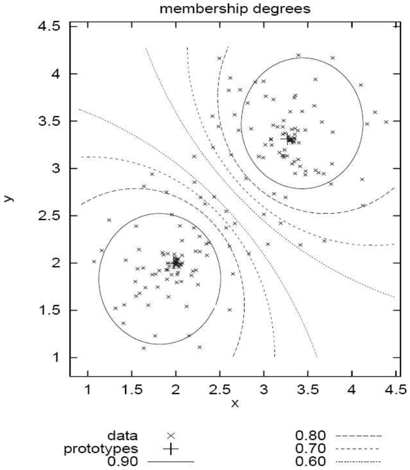
Constraints for the Objective function

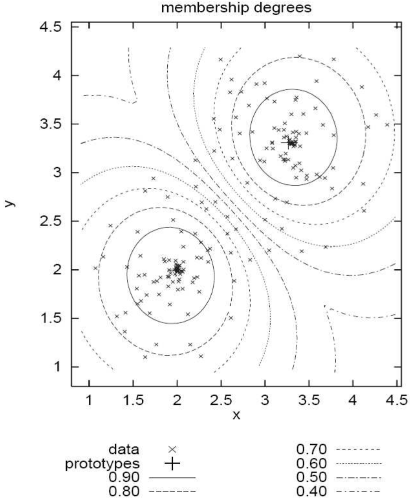
Probabilistic clustering

Noise clustering

Influence of outliers

Probabilistic and Noise Clustering





Influence of Outliers

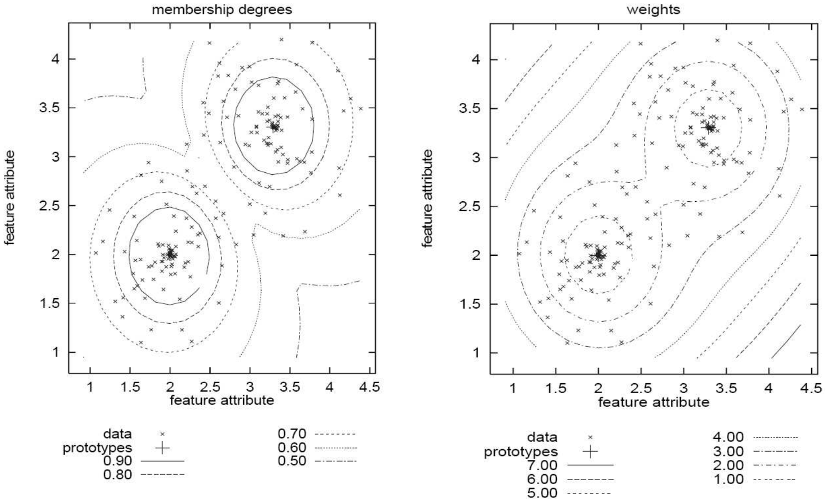
A weighting factor ω_j is attached to each datum $\boldsymbol{x}_j$

Weighting factors are adapted during clustering

Using concept of weighting factors:

- outliers in data set can be identified and
- outliers' influence on partition is reduced

Membership Degrees and Weighting Factors



Influence of Outliers

Minimize objective function

$$J(X, U, C) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} \cdot \frac{1}{\omega_j^q} \cdot d_{ij}^2$$

subject to

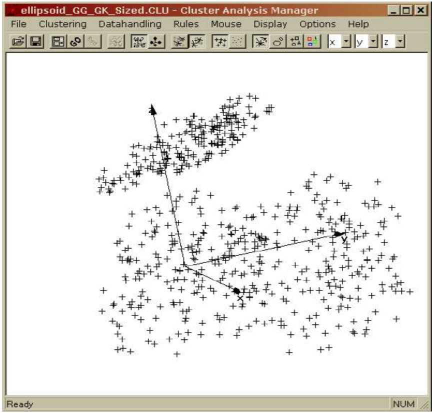
$$\forall j \in [n] : \sum_{i=1}^{c} u_{ij} = 1, \quad \forall i \in [c] : \sum_{j=1}^{n} u_{ij} > 0, \quad \sum_{j=1}^{n} \omega_j = \omega$$

q determines emphasis put on weight adaption during clustering

Update equations for memberships and weights, resp.

$$u_{ij} = \frac{d_{ij}^{\frac{2}{1-m}}}{\sum_{k=1}^{c} d_{kj}^{\frac{2}{1-m}}}, \qquad \omega_j = \frac{\left(\sum_{i=1}^{c} u_{ij}^m d_{ij}^2\right)^{\frac{1}{q+1}}}{\sum_{k=1}^{n} \left(\sum_{i=1}^{c} u_{ik}^m d_{ik}^2\right)^{\frac{1}{q+1}}} \cdot \omega$$

Determining the Number of Clusters



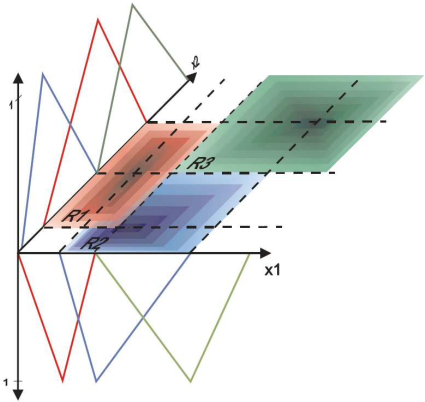
Here, validity measures evaluating whole partition of data

Getting: global validity measures

Clustering is run for varying number of clusters

Validity of resulting partitions is compared

Fuzzy Rules and Induced Vague Areas

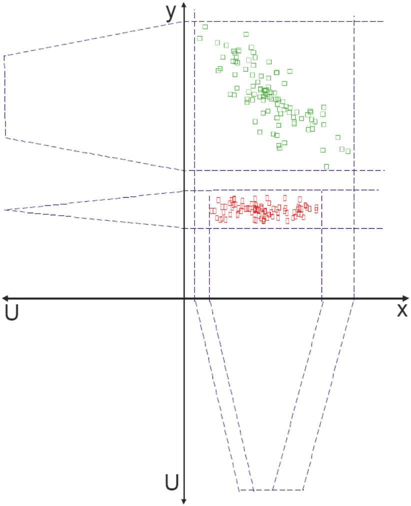


Intensity of color indicates firing strength of specific rule

Vague areas = fuzzy clusters where color intensity indicates membership degree

Tips of fuzzy partitions in single domains = projections of multidimensional cluster centers

Simplification of Fuzzy Rules



Similar fuzzy sets are combined to one fuzzy set

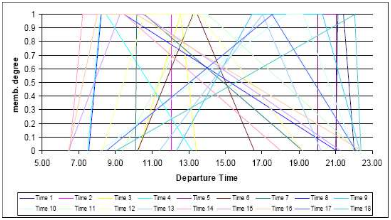
Fuzzy sets similar to universal fuzzy set are removed

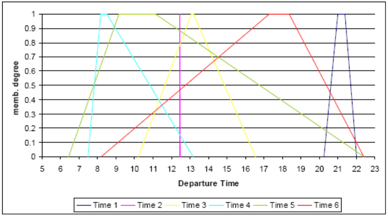
Rules with same input sets are

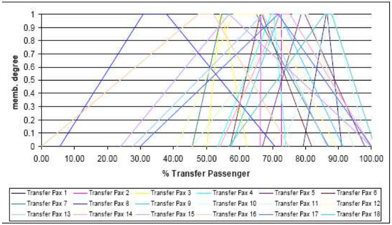
- Combined if they also have same output set(s) or
- Otherwise removed from rule set

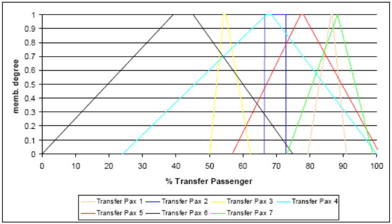
Results

FCM with $c = 18$, outlier and size adaptation, Euclidean distance:









resulting fuzzy sets — simplified fuzzy sets

Evaluation of the Rule Base

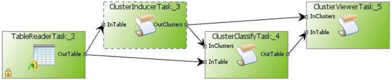
rule	max. no. of pax	De st.	depart.	% transfer pax
1	paxmax1	R1	time1	tpax1
2	paxmax2	R1	time2	tpax2
3	paxmax3	R1	time3	tpax3
4	paxmax4	R1	time4	tpax4
5	paxmax5	R5	time1	tpax5
...	...	...	...	...

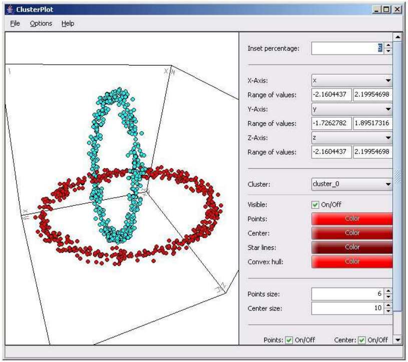
rules 1 and 5: aircraft with relatively small amount of maximal passengers (80-200), short- to medium-haul destination, and departing late at night usually have high amount of transfer passengers (80-90%)

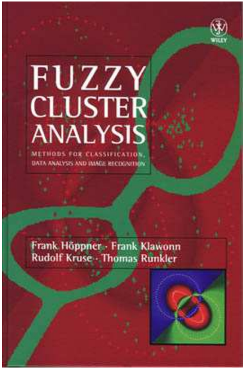
rule 2: flights with medium-haul destination and small aircraft (about 150 passengers), starting about noon, carry relatively high amount of transfer passengers (ca. 70%)

Software and Literature

“Information Miner 2” and “Fuzzy Cluster Analysis”







References I

Gustafson, E. E. and Kessel, W. C. (1979).
Fuzzy clustering with a fuzzy covariance matrix.
In *Proceedings of the IEEE Conference on Decision and Control*, pages 761–766, Piscataway, NJ, USA. IEEE Press.

Keller, A. and Kruse, R. (2002).
Fuzzy rule generation for transfer passenger analysis.
In Wang, L., Halgamuge, S. K., and Yao, X., editors, *Proceedings of the 1st International Conference on Fuzzy Systems and Knowledge Discovery (FSDK'02)*, pages 667–671, Orchid Country Club, Singapore.

Schölkopf, B. and Smola, A. J. (2001).
Learning With Kernels: Support Vector Machines, Regularization, Optimization and Beyond.
MIT Press, Cambridge, MA, USA.

Wu, Z., Xie, W., and Yu, J. (2003).
Fuzzy c-means clustering algorithm based on kernel method.
In *Proceedings of the Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA)*, pages 1–6.