



Granular Computing:

Algorithms of Information

Granulation



Index

1. Fuzzy C-Means clustering
2. Exclusion/Inclusion Fuzzy Classification Network
3. Granular prototyping in Fuzzy Clustering
4. A model of granular data: Tchebyshev distance based clustering

From Data to Information Granules

Where do fuzzy sets come from?

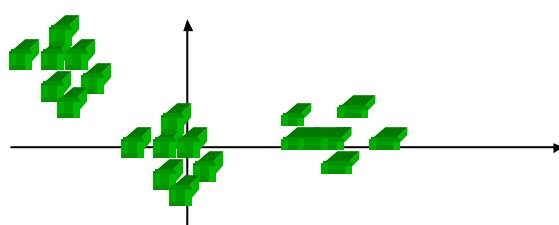
experiencial knowledge

numeric data \Rightarrow information granules

Clustering (unsupervised learning)

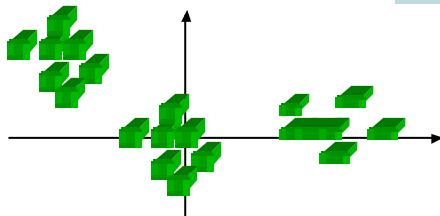
(Kaufman, Rousseeuw, 1990)

... cluster analysis is the art of finding groups in data.



Clustering Representation issues

how to represent clusters(groups)?



data \Rightarrow partition matrix

Partition matrix

N data points

c clusters

$$U = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

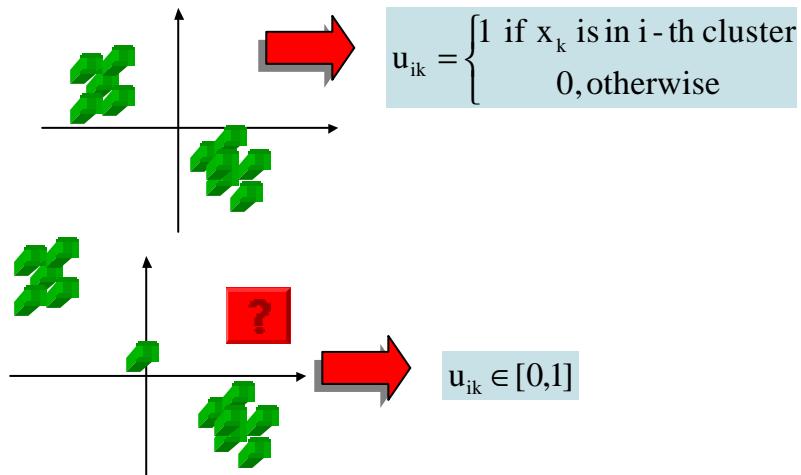
Partition Matrix

$$U = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$



cluster-1: {1,4,5,8}
cluster-2:{2,3}
cluster-3:{6,7}

Fuzzy Partition



Fuzzy Partition

U is a partition matrix if

$$- \sum_{i=1}^c u_{ik} = 1 \text{ for all } k = 1, 2, \dots, N$$

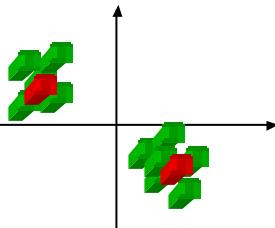
$$- 0 < \sum_{k=1}^N u_{ik} < N \text{ for all } i = 1, 2, \dots, c$$

$$U = \{ U \mid 0 < \sum_{k=1}^N u_{ik} < N, \quad 0 < \sum_{i=1}^c u_{ik} < 1 \}$$

Objective Function-based Clustering (Fuzzy C-Means)

$$Q = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m \| \mathbf{x}_k - \mathbf{v}_i \|^2, \quad m > 1$$

\mathbf{v}_i : prototypes (centres of clusters)
 $\mathbf{U} = [u_{ik}]$: partition matrix



$$Q = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m \| \mathbf{x}_k - \mathbf{v}_i \|^2 \Rightarrow \text{Min}_{\text{prototypes } \mathbf{U} \in \mathbf{U}} Q$$

Fuzzy C-Means: Detailed Calculations

Constrained optimization converted into unconstrained optimization using Lagrange multipliers

$$V = \sum_{i=1}^c u_{ik}^m d_{ik}^2 - \lambda \left(\sum_{i=1}^c u_{ik} - 1 \right)$$

$$\frac{\partial V}{\partial u_{st}} = 0, \quad \frac{\partial V}{\partial \lambda} = 0 \quad s = 1, 2, \dots, c, \quad t = 1, 2, \dots, N$$

Fuzzy C-Means: Detailed Calculations

$$\frac{\partial V}{\partial u_{st}} = mu_{st}^{m-1}d_{st}^2 - I = 0$$

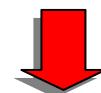
$$\sum_{i=1}^c u_{it} = 1$$



$$u_{st} = \left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}} \left(\frac{1}{d_{st}}\right)^{\frac{2}{m-1}}$$



$$\left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}} \sum_{i=1}^c \left(\frac{1}{d_{it}}\right)^{\frac{2}{m-1}} = 1$$



Fuzzy C-Means: Detailed Calculations

$$mu_{st}^{m-1}d_{st}^2 = \lambda$$

$$\left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}} = \frac{1}{\sum_{i=1}^c \left(\frac{1}{d_{it}}\right)^{\frac{2}{m-1}}}$$

$$u_{st} = \left(\frac{\lambda}{m}\right)^{\frac{1}{m-1}} \frac{1}{(d_{st})^{\frac{2}{m-1}}}$$



$$u_{st} = \frac{1}{\sum_{i=1}^c \left(\frac{d_{st}}{d_{it}}\right)^{\frac{2}{m-1}}}$$

Fuzzy C-Means: Detailed Calculations

$$\min_{\text{prototypes}} \sum_{i=1}^c \sum_{k=1}^N u_{ik}^m \| \mathbf{x}_k - \mathbf{v}_i \|^2$$

$$\frac{\partial}{\partial \mathbf{v}_i} \sum_{k=1}^N u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i)^T (\mathbf{x}_k - \mathbf{v}_i) = -2 \sum_{k=1}^N u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i)$$

↓

$$\sum_{k=1}^N u_{ik}^m (\mathbf{x}_k - \mathbf{v}_i) = 0$$

→

$$\mathbf{v}_i = \frac{\sum_{k=1}^N u_{ik}^m \mathbf{x}_k}{\sum_{k=1}^N u_{ik}^m}$$

Fuzzy C-Means: The Algorithm

Given : Define the number of clusters (c), fix the distance function and decide upon the value of the power factor (m) in the objective function

(iterative computation) compute the prototypes and update the partition matrix (U) based upon the first-order conditions of the minimized objective function. The computations are stopped while some termination criterion is satisfied.

Result : partition matrix and prototypes

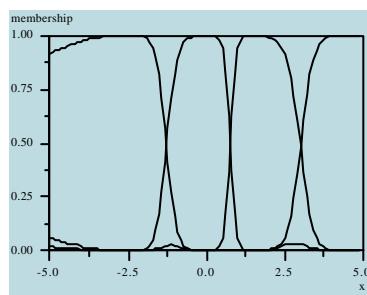
Fuzzy C-Means: The Algorithm

$$u_{st} = \frac{1}{\sum_{i=1}^c \left(\frac{d_{st}}{d_{it}} \right)^{\frac{2}{m-1}}}$$

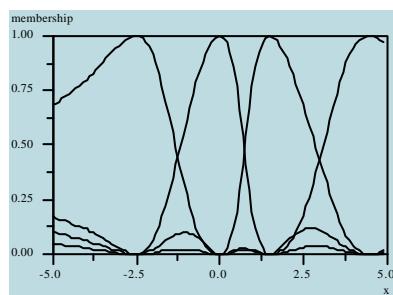
$$d_{st} = \| \mathbf{x}_t - \mathbf{v}_s \|$$

$$\mathbf{v}_i = \frac{\sum_{k=1}^N u_{ik}^m \mathbf{x}_k}{\sum_{k=1}^N u_{ik}^m}$$

Fuzzy C-Means: Fuzzification coefficient (m)

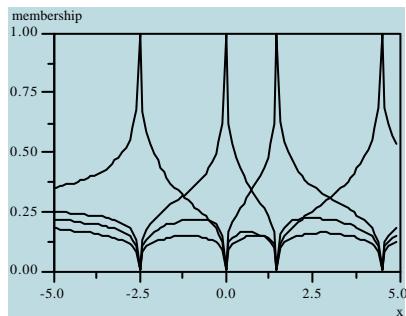


$m = 1.5$



$m = 2.0$

Fuzzy C-Means: Fuzzification coefficient (m)



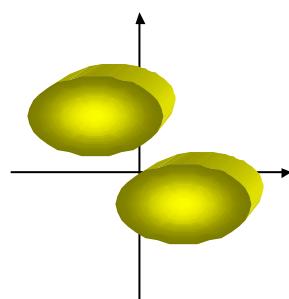
$m = 3.0$

From Fuzzy C-Means to Membership Functions

Given prototypes $\{\mathbf{v}_i\}, i = 1, 2, \dots, c$

Induction of fuzzy relations R_i

$$R_i(\mathbf{x}) = \frac{1}{\sum_{j=1}^c \|\mathbf{x} - \mathbf{v}_j\|^{2/m-1}}$$



From Fuzzy C-Means to Membership Functions

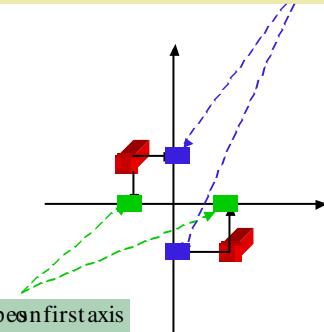
Given prototypes $\{v_i\}, i = 1, 2, \dots, c$

Projection of prototypes on individual variables

Induction of fuzzy sets for each variable A_i

$$R_i(x) = \frac{1}{\sum_{j=1}^c \frac{\|x - v_{i(j)}\|^{2/m-1}}{\|x - v_{j(1)}\|^{2/m-1}}}$$

projection of prototype on second axis



projection of prototype on first axis

From Fuzzy C-Means to Membership Functions

A Compact Gaussian Representation of Fuzzy Information Granules, *

G. Castellano, A.M. Fanelli, C. Mencar, Proc. SCIS 2002, Tsukuba, Japan

Generation of interpretable fuzzy granules by a double-clustering technique,

G. Castellano, A.M. Fanelli, C. Mencar, *Archives of Control Sciences*, Vol.12, 4, 397-410.

Objective

- For each cluster discovered by the clustering algorithm, find a set of Gaussian representations, corresponding to the functional form:

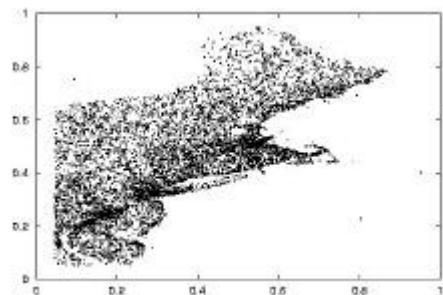
$$m_{[w,C]}(\mathbf{x}) := \exp\left(-(\mathbf{x} - \mathbf{w}) C (\mathbf{x} - \mathbf{w})^T\right)$$

- where:

$$C := \text{diag } \mathbf{c} = \text{diag}(c_1, c_2, \dots, c_n), c_i > 0$$

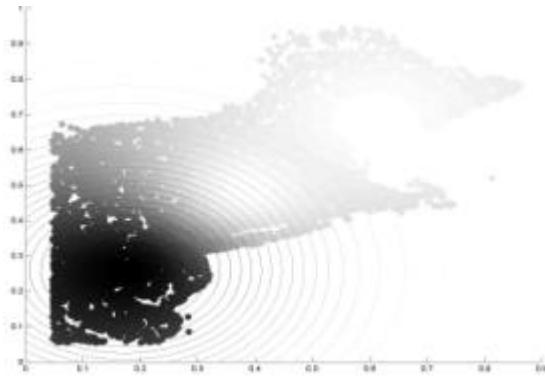
Simulation Results

- North East Dataset
 - 123,593 postal addresses
 - Three metropolitan areas (New York, Boston, Philadelphia)
- Fuzzy clustering with Fuzzy C-Means
 - 3 clusters found (one for each metropolitan area)



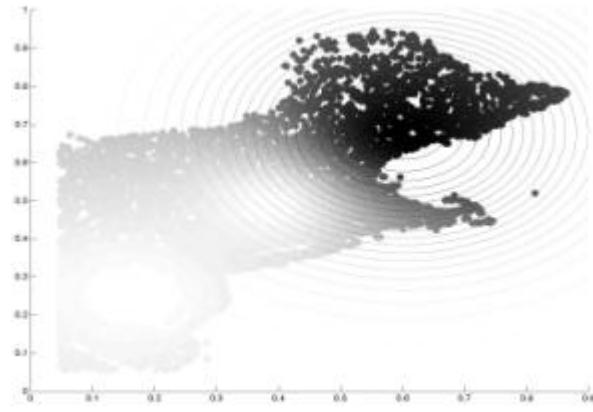
Philadelphia Cluster

Shades = values of partition matrix
Lines = level curves of derived Gaussian clusters



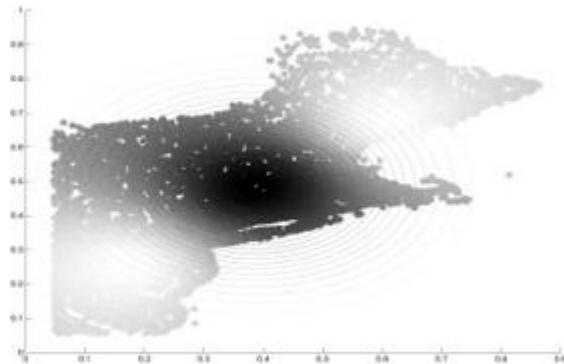
Boston Cluster

Shades = values of partition matrix
Lines = level curves of derived Gaussian clusters



New York Cluster

Shades = values of partition matrix
Lines = level curves of derived Gaussian clusters

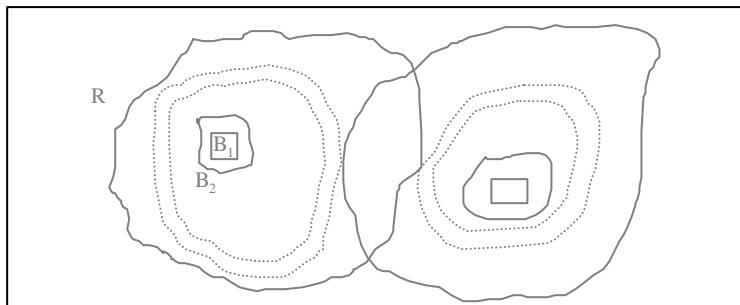


A model of granular data: Tchebyshev distance based clustering

Presentation Plan

- Standard (L_2) FCM
- L_{\inf} FCM
 - Bobrowski, Bezdek, 1991
 - Groenen, Jajuga, 2001
- Gradient-based L_{\inf} FCM
- Hierarchical data model
- Experimental results

Structure of data



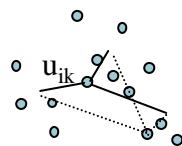
$$D = B_1 \cup B_2 \cup B_3 \cup \dots \cup B_m \cup R$$

Standard FCM

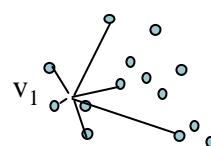
$$Q = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^2 d_{ik}$$

$$\sum_{i=1}^c u_{ik} = 1 \quad k=1,2,\dots,N; \quad 0 < \sum_{k=1}^N u_{ik} < N \quad i=1,2,\dots,c \quad (*)$$

Min Q w.r. U



Min Q w.r. v₁, v₂,...,v_c



FCM with L₂-distance

$$V = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^2 d_{ik} + I \left(\sum_{i=1}^c u_{ik} - 1 \right)$$

Min V w.r. U

$$u_{st} = \frac{1}{\sum_{j=1}^c \frac{d_{st}}{d_{jt}}}$$

Min Q w.r. v

$$v_s = \frac{\sum_{k=1}^N u_{sk}^2 x_k}{\sum_{k=1}^N v_k^2}$$

Distance measure

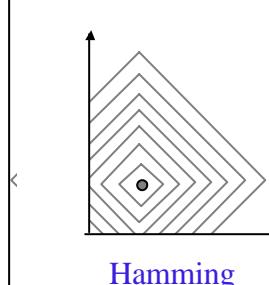
Euclidean

$$d_{ik} = \sqrt{(x_k - v_i)^2}$$

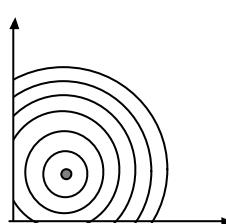
Minkovski

$$d_{ik} = \sqrt[p]{(x_k - v_i)^p}, \quad p > 0$$

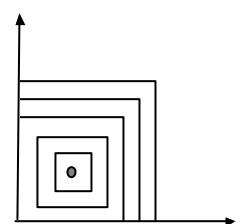
Distance measure



Hamming



Euclidean



Tchebyshev

FCM with L_{inf} distance

$$Q = \sum_{i=1}^c \sum_{k=1}^N u_{ik}^2 \max_{j=1,2,\dots,n} |x_{kj} - v_{ij}| \quad \text{s.t. (*)}$$

$$u_{st} = \frac{1}{\sum_{j=1}^c \frac{d_{st}}{d_{jt}}}$$

$$v_{st}(iter + 1) = v_{st}(iter) - \alpha \frac{\partial Q}{\partial v_{st}}$$

$$\frac{\partial Q}{\partial v_{st}} = \sum_{k=1}^N u_{sk}^2 \frac{\partial}{\partial v_{st}} \left\{ \max_{j=1,2,\dots,n} |x_{kj} - v_{sj}| \right\}$$

j = t

FCM with L_{inf} distance

$$\frac{\partial Q}{\partial v_{st}} = \sum_{k=1}^N u_{sk}^2 \frac{\partial}{\partial v_{st}} \left\{ \max(A_{kst}, |x_{kt} - v_{st}|) \right\}$$

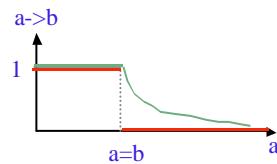
$$A_{kst} = \max_{\substack{j=1,2,\dots,n \\ j \neq t}} |x_{kj} - v_{sj}|$$

$$\frac{\partial Q}{\partial v_{st}} = \sum_{k=1}^N u_{sk}^2 \begin{cases} -1 & \text{if } A_{kst} \leq |x_{kt} - v_{st}| \text{ and } x_{kt} > v_{st} \\ 1 & \text{if } A_{kst} \leq |x_{kt} - v_{st}| \text{ and } x_{kt} \leq v_{st} \\ 0 & \text{otherwise} \end{cases}$$

FCM with L_{\inf} distance

Degree(a is included in b)

$$a \rightarrow b = \begin{cases} 1 & \text{if } a \leq b \\ b/a & \text{otherwise} \end{cases}$$



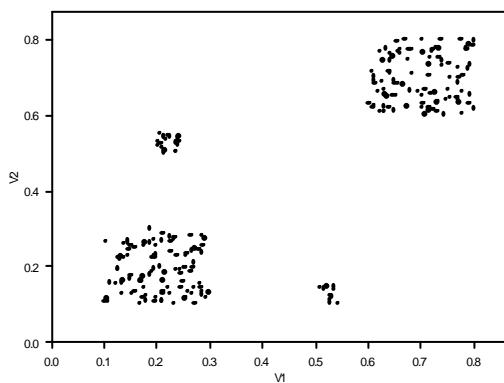
residuation operation →

$$\frac{\partial Q}{\partial v_{st}} = \sum_{k=1}^N u_{sk}^2 \begin{cases} -(A_{kst} \rightarrow |x_{kt} - v_{st}|) & \text{if } x_{kt} > v_{st} \\ (A_{kst} \rightarrow |x_{kt} - v_{st}|) & \text{if } x_{kt} \leq v_{st} \end{cases}$$

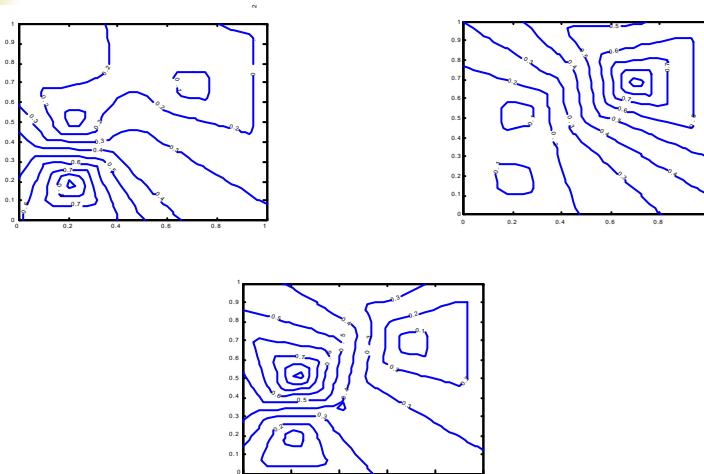
Summary of the algorithm

- Random initialization of U
- Repeat:
 - Compute partition matrix U;
 - Compute prototypes v (the U is ‘frozen’ during calculation of prototypes); this step is most time consuming;
- Until standard FCM termination criterion satisfied

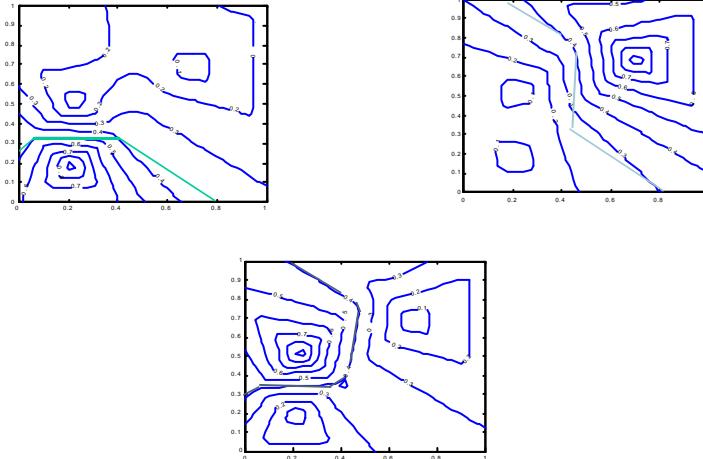
Experiments



Membership grades



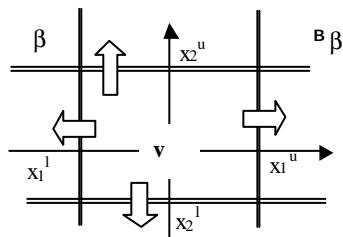
Classification boundaries



Capture of the structure of data

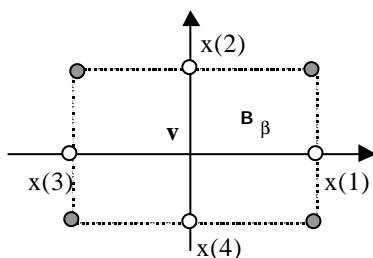
No. of FCM clusters	Core areas in data groups (FCM-L ₂)	Core areas in data groups (FCM-L _{inf})
2	1, 2	1, 2
3	1, 2	1, 2, 3
4	1, 2	1, 2, 3
5	1, 2	1, 2, 3
6	1, 2	1, 2, 3

Generation of granular prototypes



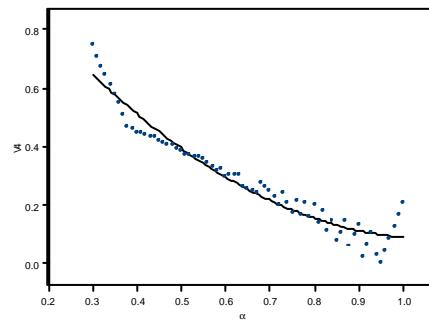
$${}^C \mathbf{b} = \{\mathbf{x} \in \{x_1^l, x_1^u\} \times \dots \times \{x_n^l, x_n^u\} \mid u(\mathbf{x}, \mathbf{v}) = \mathbf{b}\}$$

Hyperbox deformation index



$$D = |\beta - u(1)| + |\beta - u(2)| + |\beta - u(3)| + |\beta - u(4)|$$

Hyperbox deformation index



Residual structure

$$u_i(x) = \frac{1}{\sum_{j=1}^c \frac{d(x, v_i)}{d(x, v_j)}}$$

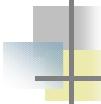


Conclusions

- Data characterisation through core segments (identified as hyperboxes)
- New way of optimising prototypes in FCM with Tchebyschev distance (gradient-based technique with logic-oriented mechanism of gradient determination)
- Deformation of hyperboxes as a criterion for establishing the extent of core area



Granular prototyping in Fuzzy Clustering



Objectives

- Sequential granulation with globally significant prototypes identified first
- Ascertainment of good explorative capabilities of the logic-based granular prototyping
- Achievement of characterization of prototypes that guides formation of interpretable granules



Problem formulation

- N data points in n-dimensional hypercube $[0,1]^n$
- Selection of prototypes:
 - Match the data to the highest extent
 - Ensure distinctiveness of prototypes
- Fuzzy set interpretation of data elements in a unit-hypercube => similarity (equality) between the membership grades

Similarity between fuzzy sets

Similarity:

$$\text{sim}(x, v) = \sum_{i=1}^n w_i^2 s(x_i \equiv v_i)$$

Matching:

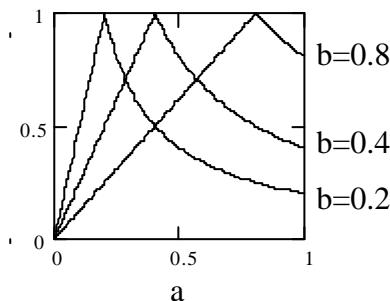
$$a \equiv b = (a \rightarrow b)t(b \rightarrow a)$$

(a is included in b) and (b is included in a)

Implication:

$$a \rightarrow b = \sup\{c \in [0,1] \mid atc \leq b\}$$

Similarity between fuzzy sets



The similarity index $a \equiv b$ regarded as a function of a for selected values of b ; the residuation is induced by the product operation, $a \rightarrow b = \min(1, b/a)$

Performance index

First prototype:

$$\sum_{k=1}^N \text{sim}(x_k, v_1; w_1) \Rightarrow \text{Max}_{v_1, w_1}$$

Second prototype:

$$(1 - \text{sim}(v_2, v_1; \mathbf{0})) \sum_{k=1}^N \text{sim}(x_k, v_2; w_2)$$

L-th prototype:

$$Q(L) = (1 - \text{sim}(v_L, v_{L-1}; \mathbf{0})) (1 - \text{sim}(v_L, v_{L-2}; \mathbf{0}) \dots \\ \dots (1 - \text{sim}(v_L, v_1; \mathbf{0})) \sum_{k=1}^N \text{sim}(x_k, v_L; w_L)$$

Prototype optimization

$$\max Q(L) = G \sum_{k=1}^N \text{sim}(x_k, v_L; w_L)$$

where

$$G = (1 - \text{sim}(v_L, v_{L-1}; \mathbf{0})) (1 - \text{sim}(v_L, v_{L-2}; \mathbf{0}) \dots (1 - \text{sim}(v_L, v_1; \mathbf{0}))$$

Subject to: $\sum_{j=1}^n w_{Lj} = 1$

Prototype optimization

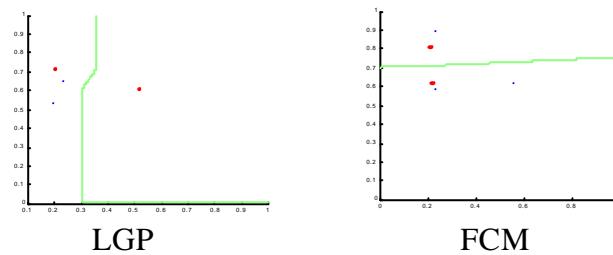
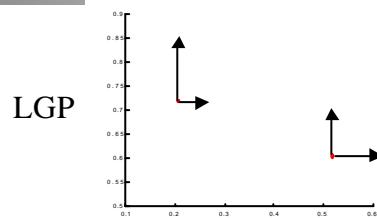
$$V = G \sum_{k=1}^N \left\{ \sum_{j=1}^n T(w_j^2 s(x_{kj} \equiv v_{Lj})) \right\} - I \left(\sum_{j=1}^n w_{Lj} - 1 \right)$$

$$\frac{dV}{dw_{Ls}} = 0 \quad \frac{dV}{dI} = 0$$

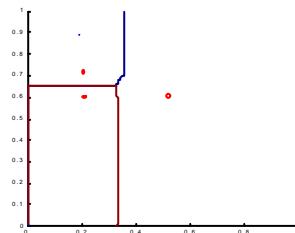
Using product and probabilistic sum for t- and s-norms we obtain

$$w_s = \frac{1}{\sum_{j=1}^c \frac{\sum_{k=1}^N A_{ks}(1-u_{ks})}{\sum_{k=1}^N A_{kj}(1-u_{kj})}}$$

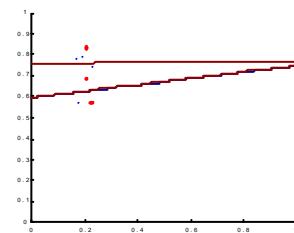
Example – 2 prototypes



Example – 3 prototypes



LGP



FCM

Conclusions

- New logic-based granular prototyping algorithm:
 - Sequential identification of prototypes; from global to detailed view
 - Fuzzy set based optimization
 - Indication of the relative importance of features of individual prototypes by the weight vector
- Computational complexity
- Complement to other techniques



Exclusion/Inclusion fuzzy classification network



Overview

- Background
- General fuzzy Min-Max NN
- Exclusion/Inclusion NN
- Example

Background

Fuzzy sets abstraction:

- Bellman, R.E., Kalaba, R., Zadeh, L., 1966, Abstraction and pattern classification, *J. Math. Anal. Appl.*, 13, 1-7

Neural networks + fuzzy sets:

- Pedrycz, W., 1992, Fuzzy neural networks with reference neurons as pattern classifiers, *IEEE Trans. Neural Networks*, 5, 770-775
- P.K. Simpson, 1992, Fuzzy min-max neural networks, *IEEE Trans. Neural Networks*, 5, 776-786
- Gabrys, B., Bargiela, A., 2000, General fuzzy min-max neural network for clustering and classification, *IEEE Trans. Neural Networks*, 11, 769-783

Basic definitions

Input:

- Points: $\{X_h, d_h\}$;
- Hyperboxes: $\{X_h, d_h\}$;
 $X_h = [X_h^l \ X_h^u] \subset I^n; X_h^l, X_h^u \subset R^n; d_h \in \{0, 1, 2, \dots, p\}$

Basic definitions

■ Hyperbox Fuzzy Set

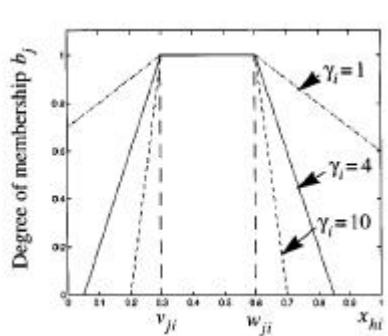
$$B_j = \{\mathbf{X}_h, \mathbf{V}_j, \mathbf{W}_j, b_j(\mathbf{X}_h, \mathbf{V}_j, \mathbf{W}_j)\}$$

■ Fuzzy Hyperbox Membership Function

$$b_j(\mathbf{X}_h) = \min_{i=1 \dots n} (\min([1 - f(x_{hi}^u - w_{ji}, \gamma_i)], [1 - f(v_{ji} - x_{hi}^l, \gamma_i)]))$$

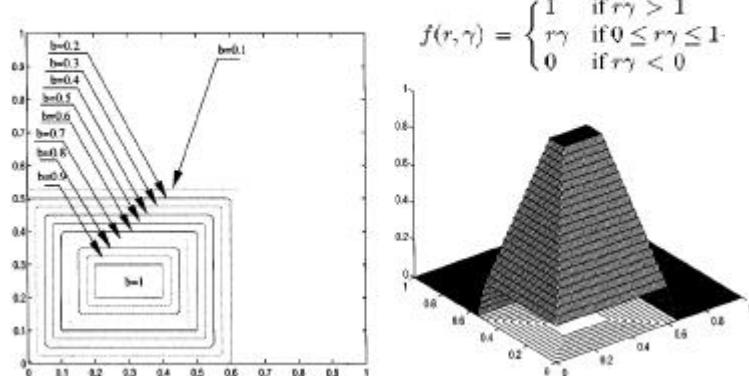
$$f(r, \gamma) = \begin{cases} 1 & \text{if } r\gamma > 1 \\ r\gamma & \text{if } 0 \leq r\gamma \leq 1 \\ 0 & \text{if } r\gamma < 0 \end{cases}$$

Basic definitions



GFMM algorithm

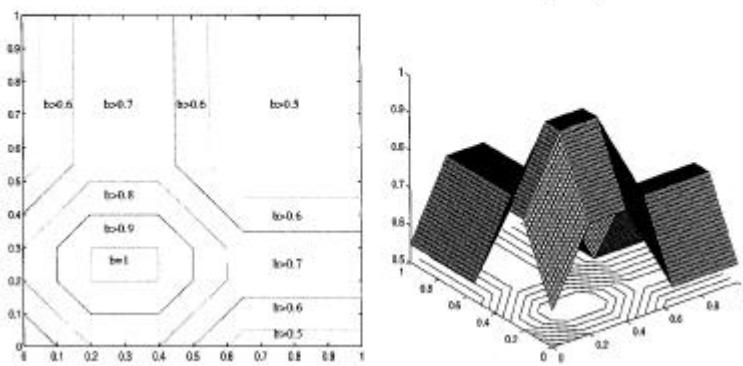
$$b_j(X_h) = \min_{i=1,N} (\min ([1-f(x_{hi}^u - w_{ji}\gamma_i)], [1-f(v_{ji} - x_{hi}^l\gamma_i)]))$$



$$f(r, \gamma) = \begin{cases} 1 & \text{if } r\gamma > 1 \\ r\gamma & \text{if } 0 \leq r\gamma \leq 1 \\ 0 & \text{if } r\gamma < 0 \end{cases}$$

GFMM algorithm

$$b_j(A_h) = \frac{1}{2n} \sum_{i=1}^n [\max(0, 1 - \max(0, \gamma \min(1, a_{hi} - w_{ji})) + \max(0, 1 - \max(0, \gamma \min(1, v_{ji} - a_{hi})))]$$



GFMM algorithm

■ Initialisation

$$V_j = \mathbf{0} \quad \text{and} \quad W_j = \mathbf{0},$$

this means that when the hyperbox is adjusted for the first time using input pattern X_h

the hyperbox becomes

$$V_j = X_h^l \quad \text{and} \quad W_j = X_h^u$$

GFMM algorithm

■ Hyperbox expansion

■ expansion criterion

$$\forall_{i=1,\dots,n} (\max(w_{ji}, x_{hi}^u) - \min(v_{ji}, x_{hi}^l)) \leq \Theta$$

*if $d_h = 0$ then adjust B_j
else*

$$\text{if } \text{class}(B_j) = \begin{cases} 0 \Rightarrow \text{adjust } B_j \\ d_h \Rightarrow \text{adjust } B_j \\ \text{else} \Rightarrow \text{take another } B_j \end{cases}$$

GFMM algorithm

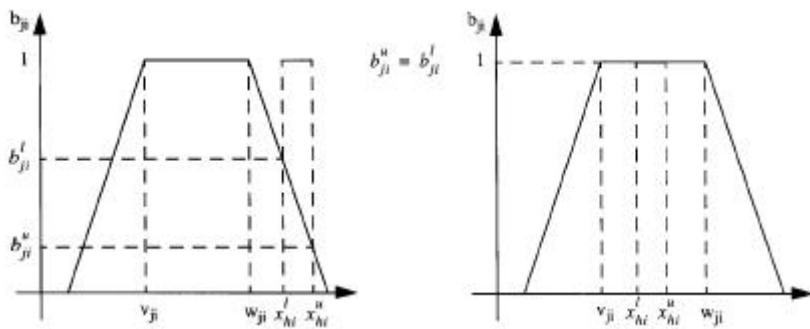
- Hyperbox expansion
 - hyperbox adjustment

$$v_{ji}^{\text{new}} = \min(v_{ji}^{\text{old}}, x_{hi}^l), \quad \text{for each } i = 1, \dots, n$$

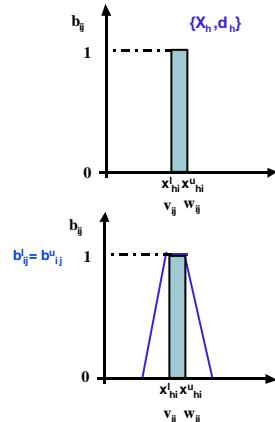
$$w_{ji}^{\text{new}} = \max(w_{ji}^{\text{old}}, x_{hi}^u), \quad \text{for each } i = 1, \dots, n.$$

GFMM algorithm

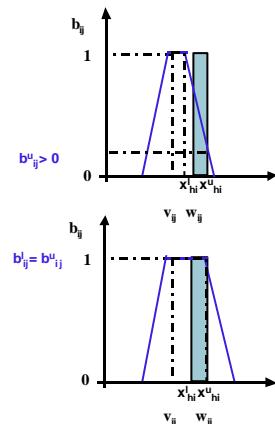
- Hyperbox expansion (parameters: γ, Θ)



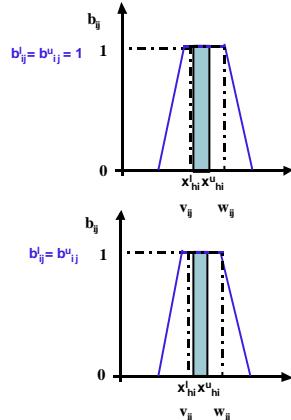
GFMM algorithm



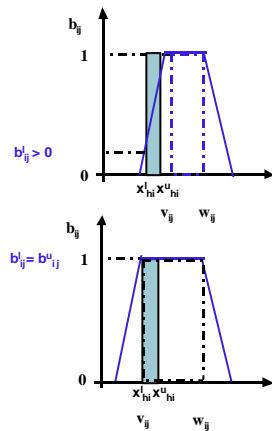
GFMM algorithm



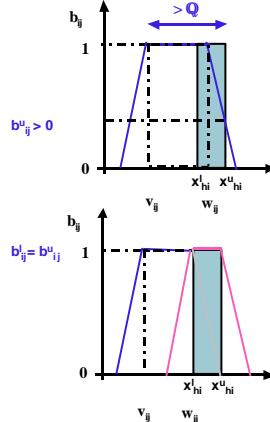
GFMM algorithm



GFMM algorithm



GFMM algorithm



GFMM algorithm

Overlap test

Assuming that hyperbox B_j was expanded in the previous step, test for overlapping with B_k if

$$class(B_j) = \begin{cases} \text{test for overlapping} \\ \text{with all the other} \\ \text{hyperboxes} \\ 0 \Rightarrow \\ \text{else} \Rightarrow \text{only if} \\ class(B_j) \neq class(B_k). \end{cases}$$

GFMM algorithm

Overlap test

Case 1: $v_{ji} < v_{ki} < w_{ji} < w_{ki}$

$$\delta^{\text{new}} = \min(w_{ji} - v_{ki}, \delta^{\text{old}}).$$

Case 2: $v_{ki} < v_{ji} < w_{ki} < w_{ji}$

$$\delta^{\text{new}} = \min(w_{ki} - v_{ji}, \delta^{\text{old}}).$$

Case 3: $v_{ji} < v_{ki} \leq w_{ki} < w_{ji}$

$$\delta^{\text{new}} = \min(\min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{\text{old}}).$$

Case 4: $v_{ki} < v_{ji} \leq w_{ji} < w_{ki}$

$$\delta^{\text{new}} = \min(\min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{\text{old}}).$$

GFMM algorithm

Hyperbox contraction

Case 1: $v_{j\Delta} < v_{k\Delta} < w_{j\Delta} < w_{k\Delta}$

$$v_{k\Delta}^{\text{new}} = w_{j\Delta}^{\text{new}} = \frac{v_{j\Delta}^{\text{old}} + w_{j\Delta}^{\text{old}}}{2} \text{ or alternatively } (w_{j\Delta}^{\text{new}} = v_{k\Delta}^{\text{old}}).$$

Case 2: $v_{k\Delta} < v_{j\Delta} < w_{k\Delta} < w_{j\Delta}$

$$v_{j\Delta}^{\text{new}} = w_{k\Delta}^{\text{new}} = \frac{v_{j\Delta}^{\text{old}} + w_{k\Delta}^{\text{old}}}{2} \text{ or alternatively } (v_{j\Delta}^{\text{new}} = w_{k\Delta}^{\text{old}}).$$

Case 3: $v_{j\Delta} < v_{k\Delta} \leq w_{k\Delta} < w_{j\Delta}$

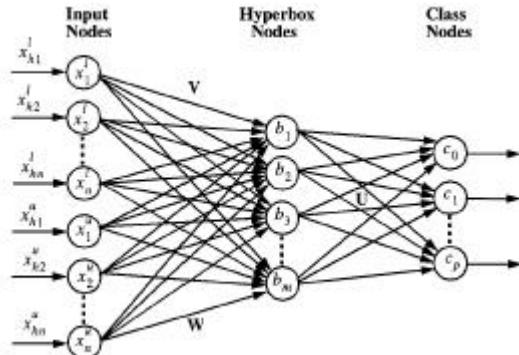
$$\text{if } w_{k\Delta} - v_{j\Delta} < w_{j\Delta} - v_{k\Delta} \text{ then } w_{j\Delta}^{\text{new}} = w_{k\Delta}^{\text{old}} \text{ otherwise } w_{j\Delta}^{\text{new}} = v_{k\Delta}^{\text{old}}.$$

Case 4: $v_{k\Delta} < v_{j\Delta} \leq w_{j\Delta} < w_{k\Delta}$

$$\text{if } w_{k\Delta} - v_{j\Delta} < w_{j\Delta} - v_{k\Delta} \text{ then } w_{k\Delta}^{\text{new}} = v_{j\Delta}^{\text{old}} \text{ otherwise } v_{k\Delta}^{\text{new}} = w_{j\Delta}^{\text{old}}.$$

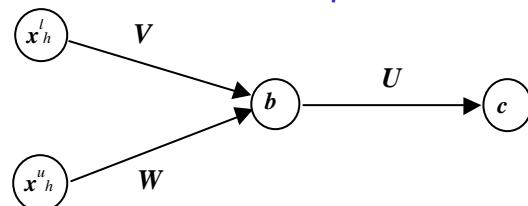
GFMM algorithm

- Neural network implementation



GFMM algorithm

- Neural network implementation



$$\mathbf{x}_h^l = [x_{h1}^l, \dots, x_{hn}^l], \mathbf{x}_h^u = [x_{h1}^u, \dots, x_{hn}^u]$$

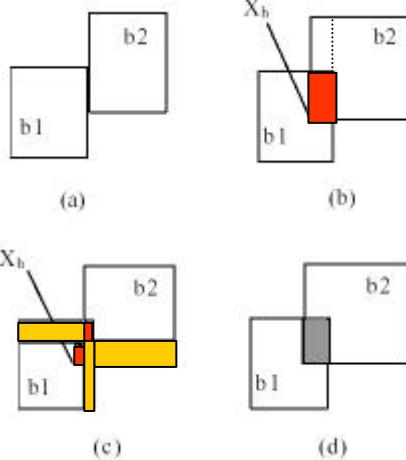
$$\mathbf{V} = [v_1, \dots, v_m], \mathbf{W} = [w_1, \dots, w_m]$$

$$\mathbf{b} = [b_1, \dots, b_m]$$

$$\mathbf{U} = [u_{10}, \dots, u_{1p}; \dots; u_{m0}, \dots, u_{mp}]$$

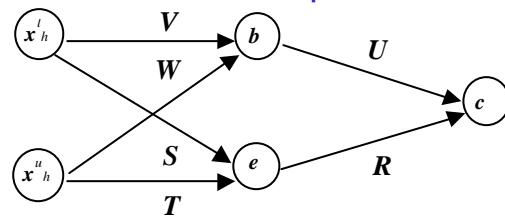
$$\mathbf{c} = [c_0, c_1, \dots, c_p]$$

GFMM → EIFC



EIFC algorithm

■ Neural network implementation



$$\begin{aligned}
 \mathbf{x}_h^l &= [x_{h1}^l, \dots, x_{hn}^l], \mathbf{x}_h^u = [x_{h1}^u, \dots, x_{hn}^u] \\
 \mathbf{V} &= [v_1, \dots, v_m], \mathbf{W} = [w_1, \dots, w_m], \mathbf{S} = [s_1, \dots, s_m], \mathbf{T} = [t_1, \dots, t_m] \\
 \mathbf{b} &= [b_1, \dots, b_m], \mathbf{e} = [e_1, \dots, e_q] \\
 \mathbf{U} &= [u_{10}, \dots, u_{1p}, u_{1(p+1)}, \dots, u_{m0}, \dots, u_{mp}, u_{m(p+1)}] \\
 \mathbf{R} &= [r_{10}, \dots, r_{1p}, r_{1(p+1)}, \dots, r_{q0}, \dots, r_{qp}, r_{q(p+1)}] \\
 \mathbf{c} &= [c_0, c_1, \dots, c_p, c_{p+1}]
 \end{aligned}$$

EIFC algorithm

■ Initialisation

$$\{\mathbf{X}_h, \mathbf{d}_h\};$$

$$\mathbf{V}_j = \mathbf{0} \quad \text{and} \quad \mathbf{W}_j = \mathbf{0}.$$

$$B_j = \{\mathbf{X}_h, \mathbf{V}_j, \mathbf{W}_j, b_j(\mathbf{X}_h, \mathbf{V}_j, \mathbf{W}_j)\}$$

$$b_j(\mathbf{X}_h) = \min_{i=1,\dots,n} (\min([1 - f(x_{hi}^u - w_{ji}, \gamma_i)], [1 - f(v_{ji} - x_{hi}^l, \gamma_i)]))$$

EIFC algorithm

■ Inclusion hyperboxes – training/expansion

for $h=1,\dots,N$

$$\forall_{i=1,\dots,n} (\max(w_{ij}, x_{hi}^u) - \min(v_{ij}, x_{hi}^l)) \leq \Theta$$

if $\text{class}(B_j) = \begin{cases} d_h & \text{adjust } B_j \\ \text{else} & \text{take } B_j \text{ with next biggest } b_j(X_h) \\ & \text{or create new } B_j \text{ for class } d_h \text{ if needed} \end{cases}$



EIFC algorithm

- Exclusion hyperboxes – training/overlap test

$\forall_{j,k=1,\dots,h} \text{ if } \text{class}(B_j) \neq \text{class}(B_k) \Rightarrow$

test for overlap among B_j and B_k



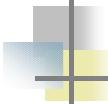
EIFC algorithm

- Exclusion hyperboxes – training/overlap test

$$\forall_{i=1,\dots,n} \begin{cases} \text{if } v_i < x_{hi}^l < w_i \Rightarrow f_i^1 = 1 \text{ and } \bar{v}_i = \max(\bar{v}_i, x_{hi}^l) \\ \text{if } v_i < x_{hi}^u < w_i \Rightarrow f_i^1 = 1 \text{ and } \bar{w}_i = \min(\bar{w}_i, x_{hi}^u) \\ \text{if } x_{hi}^l < v_i < x_{hi}^u \Rightarrow f_i^2 = 1 \text{ and } \bar{v}_i = \max(\bar{v}_i, v_i) \\ \text{if } x_{hi}^l < w_i < x_{hi}^u \Rightarrow f_i^2 = 1 \text{ and } \bar{w}_i = \min(\bar{w}_i, w_i) \end{cases}$$

$$\text{if } \max\left(\prod_i f_i^1, \prod_i f_i^2\right) = 1 \Rightarrow$$

create new exclusion hyperbox $E_l = \{\bar{V}, \bar{W}\}, 0\}$



EIFC algorithm

- Association of classes to “inclusion hyperboxes”

$$u_{jk} = \begin{cases} 1, & \text{if } b_j \text{ is a hyperbox for class } c_k \\ 0, & \text{otherwise} \end{cases}$$

$$c_k = \max b_j u_{jk}$$



EIFC algorithm

- Association of classes to “exclusion hyperboxes”

$$r_{lk} = \begin{cases} 1, & \text{if } e_l \text{ is a hyperbox overlapping with class } c_k \text{ and } 1 < k < p \\ 0, & \text{otherwise} \end{cases}$$

$$c_k = \max(0, \max_{j=1}^m b_j u_{jk} - \max_{i=1}^q e_i r_{ik})$$

Example

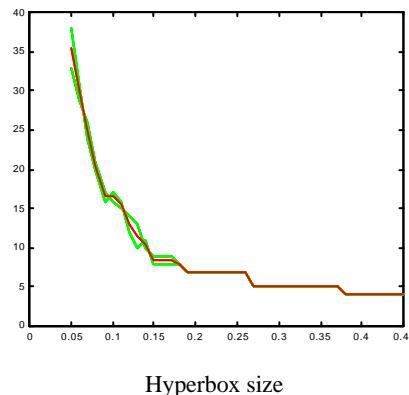
The “Iris data set”

(<http://www.ics.uci.edu/~mlearn/MLRepository.html>)

```
%title 'Canonical discriminant analysis: Fisher (1936)';  
%   iris data';  
%   INPUT SEPALLEN SEPALWID PETALLEN PETALWID SPEC_NO;  
%   IF SPEC_NO=1 THEN SPECIES='SETOSA      ';  
%   IF SPEC_NO=2 THEN SPECIES='VERSICOLOR';  
%   IF SPEC_NO=3 THEN SPECIES='VIRGINICA ';  
%   LABEL SEPALLEN=Sepal length  
%         SEPALWID=Sepal width  
%         PETALLEN=Petal length *****  
%         PETALWID=Petal width *****;
```

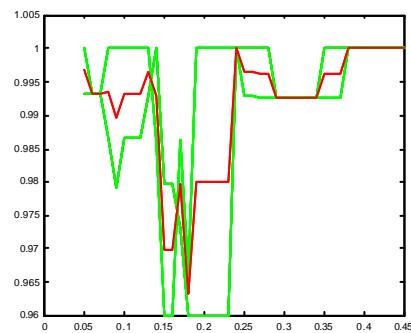
Example

Number of hyperboxes



Example

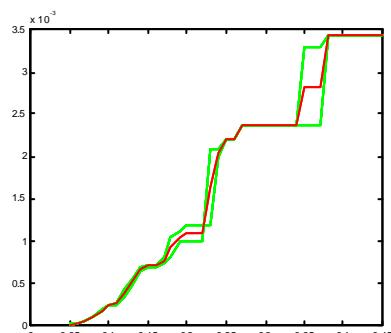
Classification accuracy



Hyperbox size

Example

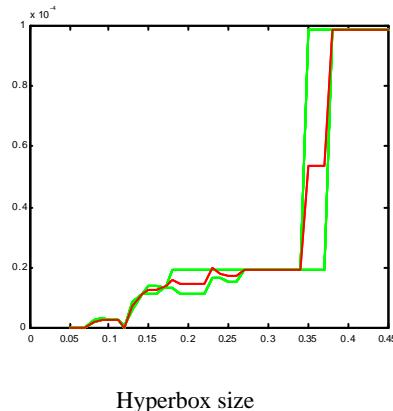
Volume of all hyperboxes



Hyperbox size

Example

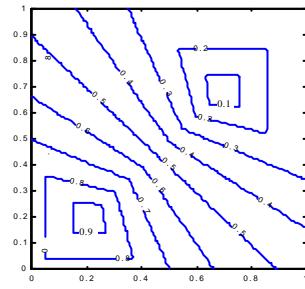
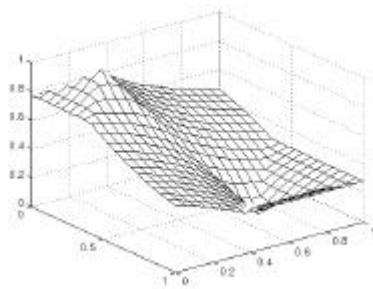
Volume of exclusion hyperboxes



Example

Algorithm / Performance criterion	FMM	GFMM	EIFC
Recognition rate (range)	97.33-92%	100-92%	100-96%
Number of hyperboxes (max. size 0.05)	48	42	35
Number of hyperboxes (max. size 0.1)	28	21	16

Membership grades



Cluster centred around (0.2088, 0.1998)