Aplicação do Algoritmo de Categorização FCM e avaliação das Medidas de Validação *ICC* e *CS*

Adriano Martins Moutinho[1] Viviane Soares Rodrigues Silva[2]

Núcleo de Computação Eletrônica / IM – Instituto de Matemática

UFRJ – Universidade Federal do Rio de Janeiro

Resumo

Este trabalho tem como objetivo apresentar as características principais do algoritmo de categorização de dados FCM (Fuzzy C-Means) e avaliar o desempenho de sua aplicação através das medidas de validação denominadas *ICC* (Inter Class Contrast) e *CS* (Compactness and Separation). Os algoritmos foram utilizados sobre três amostras de dados criadas artificialmente com distribuições distintas e prévio conhecimento do número ideal de grupos pertinentes às amostras. Foram feitos dez experimentos de categorização para cada número de categorias variando entre 2 e 10, para cada uma das três amostras, sendo computados os valores de *CS* e *ICC* para cada distribuição. É também apresentada uma comparação entre as duas medidas de validação, mostrando qual experimento obteve a melhor capacidade de categorização.

Abstract

The goal of this paper is to show the main characteristics of FCM (Fuzzy C-Means) and verify the performance of its application by using the cluster's validation measures called *ICC* (Inter Class Contrast) and *CS* (Compactness and Separation). FCM algorithm was applied over three different artificial-created distributions where the number of clusters was previously known. Ten simulations were done for each different number of categories from 2 to 10, and the validation measures *ICC* and *CS* were calculated. A comparison between *CS* and *ICC* is present by showing which experiment achieved the best categorization.

[1] Mestrando do NCE, UFRJ. adrianomm@posgrad.nce.ufrj.br
[2] Mestrando do NCE, UFRJ. vivianesrs@posgrad.nce.ufrj.br

1. Introdução:

"O nível menos apurado do processo cognitivo reside no simples reconhecer do objeto. O nível mais elevado consubstancia-se na capacidade do homem de ver as coisas como parte de um sistema".

Platão.

O ser humano é munido da capacidade de reconhecer e classificar padrões, o que o leva a ter uma percepção única do mundo [1].

Embora o reconhecer de um objeto e classificá-lo como pertencente a um conjunto pré-definido ou até mesmo criar uma nova classe para incluí-lo dentro do conhecimento prévio faça parte de diversas tarefas do cotidiano do ser humano, a complexidade de tal processo é percebida a medida que máquinas tentam reproduzi-la [2].

Geralmente, em problemas reais, o número de classes dentro de um conjunto de dados é desconhecido previamente, sendo necessário ter critérios para a identificação da melhor escolha dos grupos [1]. Por exemplo, nos negócios, seria útil descobrir os grupos distintos de clientes existentes e as características que devem ser observadas para melhor de agrupá-los. [3].

No processo de automatização do reconhecimento de padrões, pode-se dizer que existem dois tipos de algoritmos: os de aprendizado supervisionado e os de aprendizado não-supervisionado.

Na aprendizagem supervisionada, ou aprendizagem com “professor”, fornece-se ao algoritmo uma resposta correta (saída) para cada padrão de entrada. Os parâmetros do algoritmo são então ajustados a fim de permitir que sejam produzidas respostas tão próximas quanto possíveis das respostas corretas. Este tipo de aprendizagem é utilizado para experimentos de classificação.

Em contraste, a aprendizagem não-supervisionada, ou aprendizagem sem “professor”, não é necessária uma resposta correta associada com cada padrão de entrada no conjunto de dados avaliado. Ela explora a estrutura subjacente dos dados, ou correlações entre padrões dos dados e organiza os mesmos em categorias a partir destas correlações. Este tipo de aprendizagem é utilizado para experimentos de categorização.

A categorização (Cluster Analysis) deve avaliar se os dados dentro de um grupo são similares e dados de grupos disjuntos são, ao máximo, diferentes. A métrica mais usada nos métodos de categorização é o calculo da distância euclideana (eq. 2.2)

entre as características ou coordenadas dos pontos da amostra.

Em problemas reais dificilmente consegue-se incluir um elemento em apenas um grupo de maneira rígida. Por esta razão a categorização nebulosa especificará o grau de inclusão de um membro da amostra de dados em um dado grupo.

De fato, o que interessará após aplicação de um algoritmo de categorização sobre uma amostra de dados, será a avaliação do resultado de tal algoritmo, verificando se o número de grupos encontrado é realmente o número ideal.

Encontra-se na literatura algumas medidas que se propõem a encontrar de maneira eficaz este número ideal de divisões dentro de uma amostra de dados, que normalmente estão definidas num espaço de dimensão maior que três, tornando impossível sua visualização. [2]

Na abordagem deste trabalho, o método de categorização FCM (Fuzzy C-Means) é implementado, e computou-se a medida de validação dos resultados através dos algoritmos *ICC* (Inter Class Contrast) e *CS* (Compactness and Separation). Esses algoritmos, bem como as medidas de validação, são mostradas nas seções 2 e 3 respectivamente.

2. Método Nebuloso de Categorização: Fuzzy C-Means

Pode-se considerar que o algoritmo FCM é uma versão nebulosa do conhecido método k-means, empregado para classificar um universo de amostras em categorias nebulosas de acordo com a sua disposição no espaço euclidiano.

$$J_m(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n}(\mu_{ij})^m (d_{ij})^2 \quad \textbf{(2.1)}$$

$$d_{ij} = d(x_j - v_i)$$

$$d_{ij} = \left\| x_j - v_i \right\| = \left[\sum_{l=1}^{p} (x_{jl} - v_{il})^2 \right]^{1/2} \quad \textbf{(2.2)}$$

O algoritmo FCM tem como finalidade minimizar a função objetivo J_m (equação 2.1). A distância euclidiana, que é normalmente utilizada como métrica, é mostrada na equação 2.2, onde x_i é um vetor com *p* características de *n* amostras. O algoritmo possui *V = {v1, v2, ..., vc}* como um conjunto de vetores que representa os *c* centros das categorias. *U* representa a matriz de graus de inclusão nebulosos, onde μ_{ij} é o grau de inclusão do

ponto *j* na categoria *i*. A matriz *U* deve seguir as seguintes condições:

$$\sum_{i=1}^{c} \mu_{ij} = 1, \forall j \qquad \textbf{(2.3)}$$

$$0 < \sum_{j=1}^{n} \mu_{ij} < n, \forall j \qquad \textbf{(2.4)}$$

Obedecendo a eq. 2.3, garante-se a condição de que o somatório dos graus de inclusão de uma amostra em todas as categorias deve ser igual a 1 e a eq 2.4 garante que nenhuma categoria pode estar vazia ou conter todos os elementos.

A variável *m* representa o fator de nebulosidade e define a faixa de nebulosidade que existe entre uma categoria e outra. Quando m tende a 1 a matriz *U* tende a ser rígida, ou seja, um elemento poderá pertencer somente a uma categoria com grau 1. No entanto, a literatura sugere que *m* seja igual a 1,25.

O critério de parada do algoritmo deve estar relacionado ao momento em que se chega num estado onde as posições dos centros das categorias calculadas num dado instante praticamente não diferem das posições num instante anterior.

O algoritmo FCM apresenta-se da seguinte forma [3] [1] [5]:

- Fixar **c** (número de categorias nebulosas, $1<c<n$ onde n é o número de amostras);
- Atribuir um valor à m (fator de nebulosidade);
- Estabelecer condição de parada **ε** (**ε**>0);
- Inicializar aleatoriamente **U** (observando as eqs. 2.3 e 2.4);
- Inicializar o passo **r = 0**;
- Repetir até que o erro < **ε**
  - Calcular o conjunto **V** dos **c** centros das categorias:

$$v_i = \frac{\sum_{j=1}^{n} \mu_{ij}^{m} x_j}{\sum_{j=1}^{n} \mu_{ij}^{m}}$$

  - Calcular a distância de cada padrão ao centro de cada categoria de acordo com a eq. 2.2;
  - Recalcular a matriz **U** para os novos centros das categorias:

    Se $\mathbf{d_{ij}} > 0$ então

$$\mu_{ij} = \left(\sum_{l=1}^{c} \left(\frac{d_{ij}}{d_{lj}} \right)^{\frac{2}{m-1}} \right)^{-1}$$

    Senão se $d_{ij} = 0$ **então** $\mu_{ij} = 1$

    senão $\mu_{ij} = 0$

3. Medidas de Validação:

Após a aplicação de algoritmos não supervisionados freqüentemente surgem perguntas do tipo:

- Quantos grupos existem no conjunto de dados?
- Este resultado de categorização representa meu conjunto de dados?
- Esta é a melhor maneira de particionar os dados?

Para responder a estas perguntas aplica-se uma medida de validação. O objetivo é medir a qualidade dos agrupamentos resultantes da aplicação dos algoritmos de categorização de acordo com a variação dos parâmetros de inicialização.

Dentre as medidas de validação apresentadas em [1] pode-se destacar a medida *CS* (Compacidade e Separação) e a *ICC* (Inter Class Contrast) que serão objeto de avaliação nas próximas seções.

3.1. *CS* (Compactness and Separation):

Avalia a compacidade das categorias geradas e a qualidade da separação entre estas, não perdendo a exatidão mesmo quando o grau de sobreposição das categorias é alto.

Quanto menor o valor de *CS* (eq 3.1) melhor a disposição das categorias. Minimizar *CS* significa minimizar a função J_m (eq 2.1), que é a finalidade do método FCM [1].

$$CS = \frac{J_m}{n * (d_{\min})^2} \qquad (3.1)$$

Expandida a equação 3.1, mostrada na eq. 3.2, a medida *CS* pode passar a avaliar métodos nebulosos.

$$CS = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n} \mu_{ij}^{\ 2} \left\| v_i - x_j \right\|^2}{n * \min \left\| v_i - x_j \right\|^2} \qquad (3.2)$$

3.2. A Medida de Validação *ICC* (Inter Class Contrast)

A medida de validação *ICC* foi proposta em [1] e considera a compacidade e separação das categorias. Assim como a *CS*, também pode ser aplicado tanto a categorizações nebulosas quanto a rígidas. Seu calculo e dado pela eq. 3.3.

$$ICC = \frac{S_{be}}{n} * D_{\min} * \sqrt{c} \qquad (3.3)$$

onde seus componentes podem ser descritos da forma descrita no conjunto de equações 3.4:

$$S_{be} = \sum_{i=1}^{c} \sum_{j=1}^{nn} \mu_{ij} (m_{ei} - m)(m_{ei} - m)^T$$

$$D_{\min} = {}_{1<i<c} \min \left[{}_{i+1 \le j \le c-1} \min \left\| m_{ei-} m_{ej} \right\| \right]$$

$$m_{ei} = \frac{\sum_{j=1}^{n} \mu_{ij} x_j}{\sum_{j=1}^{n} \mu_{ij}}$$

$$m = \frac{1}{n} \sum_{j=1}^{n} x \qquad (3.4)$$

Na eq. 3.3 utiliza-se o termo S_{be} que é definido no primeiro passo do conjunto de equações 3.4, e que mede a proximidade dos centros das categorias. Quando o valor S_{be} é baixo, os centros das categorias estão muito próximos e isto significa que o particionamento não é adequado. Portanto, é interessante verificar que o maior valor da *ICC* é que deve ser tomado como resultado ideal de particionamento. Porém, deve-se ter cuidado quando o número de categorias e maior que o número ideal de classes, pois o valor do S_{be} cresce junto com o valor da *ICC*, levando a um falso resultado [1].

O termo D_{min} representa a distância mínima entre os centróides de todas as categorias. Seu objetivo é evitar o crescimento da *ICC*, pois quando duas ou mais categorias são associadas a uma mesma classe, a menor distância D_{min} decresce abruptamente. Desta forma, D_{min} evita que a *ICC* atinja um maior valor para uma quantidade de categorias *c* que seja maior que a ideal. Em contrapartida, quando uma ou mais categorias englobam mais de uma classe, a distância *Dmin* entre os centros aumenta, aumentando assim o valor da *ICC* [1].

O termo $\sqrt{c}$ evita que a *ICC* atinja seu valor máximo para um valor de *c* menor que o ideal. Isto pode ocorrer quando existem categorias que representam mais de uma classe e seus centros estão distantes uns dos outros. Nesta situação seriam gerados valores altos tanto para o termo D_{min} quanto para a medida *ICC*.

O fator *1/n*, onde *n* e o número de amostras, é utilizado apenas como um fator de normalização. Seu objetivo é compensar a influência do número de pontos no termo S_{be} [1].

4. Aplicação do Algoritmo Fuzzy C-Means, Problema propostos:

Como caso de estudo, para avaliar as medidas de validação *ICC* e *CS* citadas nas seções 3.1 e 3.2, são propostos 3 problemas de clusterização com dados criados artificialmente, estes são descritos a seguir:

- **Problema 1:** Um conjunto de 2000 dados bidimensionais divididos em 5 classes iguais com 500 pontos cada, distribuição normal e desvio padrão de 0,3. As classes são centradas nos pontos (1:2), (6:2), (1:6), (6:6) e (3,5:9).

- **Problema 2:** Um conjunto de 2000 dados bidimensionais divididos em 5 classes iguais com 500 pontos cada, distribuição normal e desvio padrão de 0,7. As classes são centradas nos pontos (2:2,5), (4:2,5), (3:7), (2:5) e (4:5)

- **Problema 3:** Um conjunto de 2000 dados bidimensionais divididos em 5 classes iguais com 500 pontos cada, distribuição normal e desvio padrão de 0,3. Os centros são centradas em (2:2), (4:4), (6:6), (8:8) e (10:10) de forma que os centros estão linha reta e afastados.

As figuras 1, 2 e 3 mostram as distribuições dos pontos dos problemas 1, 2 e 3 respectivamente.

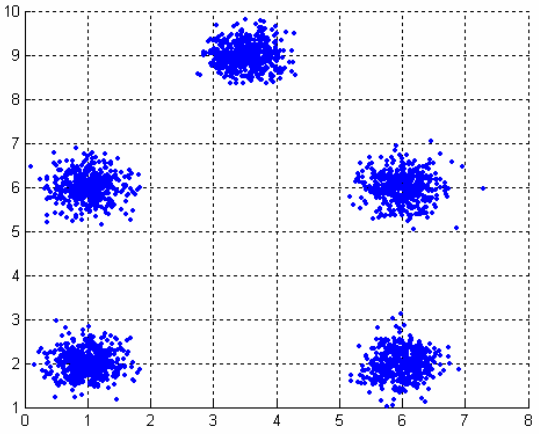

Figura 1 - Distribuição do problema 1

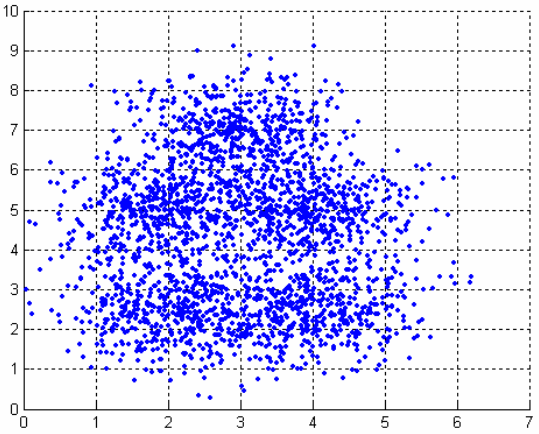

Figura 2 - Distribuição do problema 2

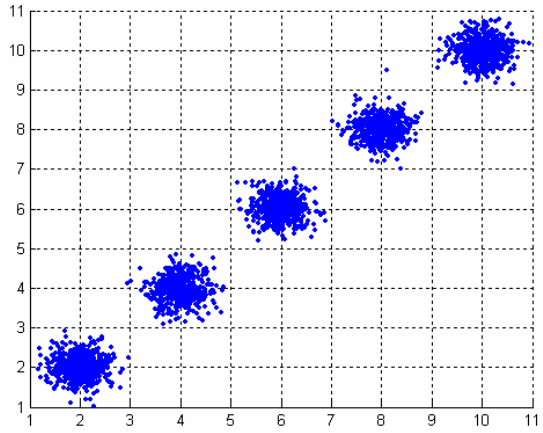

Figura 3 - Distribuição do problema 3

Os algoritmos foram implementados no software Matlab [4], bem como todos os gráficos apresentados neste trabalho.

4.1. **Algoritmo de clusterização**:

Foi usado um algoritmo C-means, descrito na seção 3, em conjunto com as medidas *ICC* e *CS*, descritas na seção 3.1 e 3.2, a fim de avaliar a melhor distribuição e particionamento dos dados criados artificialmente.

Embora se conheça, antes da aplicação das medidas de validação, o melhor número de clusters para cada caso nos problemas apresentados na seção 4, em uma aplicação real este dado não estaria disponível.

Uma metodologia que permitiria descobrir o melhor particionamento seria executar o algoritmo c-means um número grande de vezes para diferentes números de clusters, e aplicar as medidas de validação procurando o melhor resultado.

Propõe-se, então, executar o particionamento 10 vezes para cada vez que for modificado o número de clusters, que será variado de 2 a 10, totalizando 90 simulações para cada problema.

4.2. Algoritmos de validação:

Em cada uma destas simulações são aplicadas as duas técnicas de validação, tanto o *CS* como o *ICC*, totalizando duas tabelas com 90 resultados para cada problema.

5. Testes e resultados:

Para cada problema descrito na seção 4 será feita uma análise de resultados nas seções que se seguem:

5.1. Testes e resultados do problema 1:

O problema 1 é o mais simples de todos, as classes se encontram bastante separadas. Os resultados do valor de *ICC* e *CS* do problema 1 são mostrados nas tabelas 1 e 2, os melhores resultados de *ICC* e *CS* são mostrados em vermelho.

Tabela 1 – Valores do *ICC* para o problema 1.

Tentativas	Valores do *ICC* de 2 a 10 clusters								
	2	3	4	5	6	7	8	9	10
1	42,2206	80,3726	85,4089	105,708	14,1652	15,2947	16,3821	19,0237	17,5558
2	42,2954	56,8927	82,2441	105,708	14,351	16,9244	16,3867	17,6607	19,5048
3	29,4292	80,405	85,4004	105,6604	15,0405	16,8923	16,8594	17,6558	18,3486
4	42,2937	80,3838	85,4004	105,708	14,6124	15,7735	16,6324	17,626	17,5611
5	29,4131	80,4149	85,1578	105,708	14,5784	15,3134	15,7523	13,047	19,2387
6	42,2447	80,3691	82,572	105,708	14,8948	15,3143	16,3468	17,5878	18,3437
7	42,2702	80,2348	82,1709	105,708	14,1661	16,079	17,2297	17,3907	19,4726
8	42,2231	56,8219	85,3905	105,7081	14,9709	15,5152	16,3854	17,6547	17,7299
9	29,2004	80,3677	85,3879	11,6831	14,0356	15,2985	16,6045	18,1952	18,6245
10	42,2927	56,6748	85,3576	105,7075	14,1613	15,4572	15,7455	16,723	18,2937

Tabela 2 – Valores do *CS* para o problema 1.

Tentativas	Valores do *CS* de 2 a 10 clusters								
	2	3	4	5	6	7	8	9	10
1	313,5886	103,2929	64,411	5,8592	298,8579	229,8332	191,0662	138,1114	144,4901
2	313,2602	203,5782	71,5333	5,8592	291,6144	199,9546	179,63	136,3124	116,1505
3	445,5055	103,0248	64,4485	5,8623	264,1654	200,6564	183,3078	149,862	118,6551
4	313,2684	103,1779	64,4482	5,8592	277,8214	216,1951	174,3466	148,7147	144,2252
5	445,7722	103,1486	64,3894	5,8592	281,5494	231,2838	226,684	326,3776	118,1279
6	313,4817	103,3229	71,5556	5,8592	267,761	230,3085	181,0406	147,6394	119,7775
7	313,3692	103,4991	71,5278	5,8592	298,8316	207,6878	173,9856	153,4256	116,5755
8	313,5692	204,654	64,4479	5,8592	266,7924	224,0194	179,2269	148,2244	143,0053
9	444,2833	103,2578	64,4484	3396,759	303,231	230,1888	189,4824	138,2937	116,4802
10	313,2717	206,411	64,459	5,8592	299,052	225,1977	206,9087	152,1498	120,2955

Neste exemplo, onde as distâncias entre as classes são relativamente grandes, os valores de *ICC* e *CS* resultaram na mesma análise. Em 9 das 10 tentativas, o melhor particionamento foi efetuado utilizando 5 clusters atingindo o mínimo de *CS* com aproximadamente 5.85 e o máximo de *ICC* com aproximadamente 105.

A figura 4 mostra o melhor particionamento (5 clusters) obtido pelo algoritmo FCM.

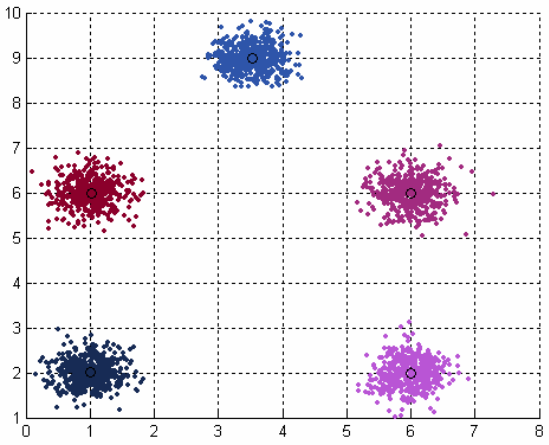

Figura 4 – Melhor distribuição mostrada pelas medidas de validação *ICC* e *CS*

5.2. Testes e resultados do problema 2:

O problema 2 é mais complexo que o problema 1, as classes se encontram bastante juntas, e mesmo visualmente não se consegue verificar que existem 5 clusters diferentes. Os resultados dos valores de *ICC* e *CS* do problema 2 são mostrados nas tabelas 3 e 4, os melhores resultados de *ICC* e *CS* são mostrados em vermelho.

Tabela 3 – Valores do *ICC* para o problema 2.

Tentativas	Valores do *ICC* de 2 a 10 clusters								
	2	3	4	5	6	7	8	9	10
1	11.5777	11.3592	13.7612	18.1664	13.0085	14.3743	15.5171	15.068	15.4894
2	11.5805	11.9553	16.4016	18.1722	14.9335	15.3362	15.7163	15.34	15.6735
3	11.5813	11.615	16.37	18.1713	14.2548	12.5057	13.9122	14.8745	15.2606
4	11.5825	11.695	14.2037	18.1647	14.4342	14.5746	14.7066	15.0744	16.1449
5	11.5818	12.0793	16.3978	18.173	13.9235	14.7041	16.4022	16.6218	16.5129
6	11.5776	11.3663	16.3883	18.1642	15.7031	14.4572	13.89	15.5318	15.5677
7	11.5827	11.7031	13.8124	18.1573	15.532	14.346	14.6396	14.7847	15.337
8	11.5755	12.0594	14.1037	18.181	13.3298	15.3004	15.4738	16.2307	15.3644
9	11.5818	12.1998	14.084	18.1801	14.2113	15.0492	16.1095	16.8873	15.3649
10	11.5773	12.0951	16.3584	18.1697	13.8582	14.6637	13.3574	15.5715	15.2956

Tabela 4 – Valores do *CS* para o problema 2.

Tentativas	Valores do *CS* de 2 a 10 clusters								
	2	3	4	5	6	7	8	9	10
1	244.6806	257.9836	162.2192	82.5901	147.5647	109.134	86.0885	83.3303	73.2449
2	244.6549	233.7474	119.2933	82.5756	111.6756	95.9694	83.7934	80.6252	71.9617
3	244.6225	246.8257	119.9912	82.5236	122.6994	144.3135	106.1899	85.5	74.8871
4	244.6236	251.7787	155.6345	82.6068	119.5234	105.9866	96.4737	83.2636	67.1981
5	244.6369	228.029	119.3748	82.5137	127.5015	104.8303	76.6435	69.4805	64.4084
6	244.6944	258.9765	119.3746	82.5519	100.8052	107.7639	107.2367	78.5395	72.4063
7	244.6058	251.7548	163.1505	82.6142	103.0445	109.3884	96.5681	86.3857	74.5185
8	244.724	228.7917	156.9192	82.5433	140.2383	96.3253	85.8074	73.063	74.0595
9	244.6348	223.0854	157.7882	82.5189	123.3773	100.1798	79.4986	67.3404	74.2814
10	244.6908	227.4313	119.9236	82.5273	129.2473	104.8123	116.1656	78.4784	75.0847

Neste exemplo, onde os clusters são bastante próximos, não sendo possível definir visualmente o número e o centro dos clusters, os valores de *ICC* e *CS* obtiveram resultados diferentes. O ICC obteve o resultado correto de acordo com o definido na distribuição criada artificialmente. Já o CS obteve uma melhor divisão em torno de 9 e 10 clusters, sendo este resultado não condizente com a forma com que foram criados os dados.

A figura 5 mostra o melhor particionamento (5 clusters) obtido pelo algoritmo C-means. A figura 6 mostra o resultado incorreto apontado pela medida de validação *CS*.

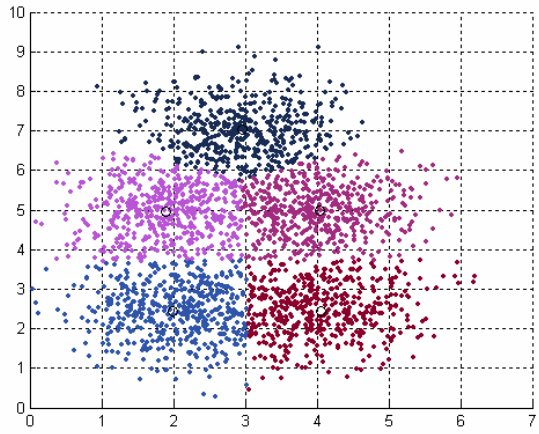

Figura 5 – Melhor distribuição (correta) do problema 2, mostrada pela medida de validação *ICC*.

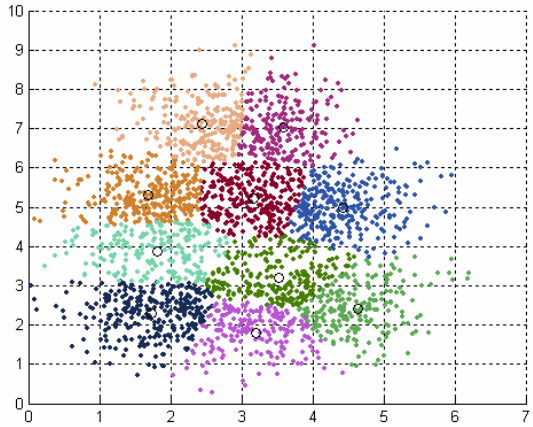

Figura 6 – Melhor distribuição (incorreta) do problema 2, mostrada pela medida de validação *CS*.

5.3. Testes e resultados do problema 3:

O problema 3 é simples como o problema 1, no entanto, sabe-se [1] que quando os centros dos clusters da distribuição artificial estão distribuídos em uma reta, ocorrem problemas nas medidas de validação, sendo o interesse verificar estes defeitos.

Os resultados dos valores de ICC e CS do problema 3 estão nas tabelas 5 e 6 respectivamente.

Neste exemplo, mesmo com os clusters são bastante separados, sendo possível definir visualmente o número e o centro dos clusters, os valores de *ICC* e *CS* obtiveram resultados diferentes. O ICC obteve o resultado incorreto mostrando que o melhor particionamento é feito com apenas dois clusters. Já o CS obteve que a melhor divisão é feita corretamente com 5 clusters.

A figura 7 mostra o particionamento de 2 clusters segundo apontado como melhor pelo algoritmo *ICC*. A figura 8 mostra o resultado correto apontado pela medida de validação *CS*.

Tabela 5 – Valores do *ICC* para o problema 3.

Tentativas	Valores do *ICC* de 2 a 10 clusters								
	2	3	4	5	6	7	8	9	10
1	122.6489	107.172	86.6959	99.8841	19.0368	20.584	22.1588	25.3469	24.1085
2	122.6805	106.9693	86.6062	99.8841	18.5046	20.0517	21.4088	23.3624	22.9965
3	122.6747	107.039	84.8773	99.8842	19.1527	20.0123	21.5315	23.3793	24.1035
4	122.674	106.9848	84.9099	99.8836	18.5122	20.082	21.4475	22.821	23.1189
5	122.6359	107.1557	86.7329	99.8842	19.0185	20.0651	20.6593	22.8432	24.1094
6	122.6719	107.0041	84.8385	99.8831	18.6118	20.0534	21.5382	22.1858	24.6539
7	122.6751	107.0451	86.6602	99.8473	19.0452	20.0336	22.1592	22.8668	24.0211
8	122.6753	106.9506	84.9499	99.8841	18.6105	22.3026	21.4386	22.8437	24.0544
9	122.6772	106.9306	85.0735	99.8473	18.5096	20.0474	22.0302	22.8681	24.1638
10	122.6823	107.1308	86.6816	99.884	19.0421	19.6248	22.1694	22.7541	25.3468

Tabela 6 – Valores do *CS* para o problema 2.

Tentativas	Valores do *CS* de 2 a 10 clusters								
	2	3	4	5	6	7	8	9	10
1	89.3934	76.6251	70.8291	11.4669	288.438	222.8964	184.9323	134.6289	121.3139
2	94.392	77.0525	70.9215	11.4669	305.5468	235.5891	198.0565	148.0409	148.3821
3	89.3506	76.9208	75.6268	11.4669	284.7331	236.0581	182.2951	147.9426	119.9437
4	89.4032	77.0073	75.5845	11.4669	305.2899	234.9135	183.8815	154.1358	147.5423
5	89.3929	76.6545	70.7906	11.4669	288.9127	234.2722	231.6824	154.7846	119.9645
6	89.4056	76.9782	75.6763	11.4665	301.4695	235.5499	182.7402	151.3127	126.8277
7	94.3119	76.8132	70.866	11.4683	288.1896	236.0104	185.0118	154.4027	134.6784
8	94.3142	77.0279	75.5316	11.4669	301.5146	202.883	183.8599	153.4264	120.6631
9	94.3426	76.7681	75.3691	11.4669	305.3747	235.1667	173.3065	141.6806	120.8689
10	94.4175	76.6921	70.8438	11.467	288.2757	247.1356	184.5669	155.8135	118.4478

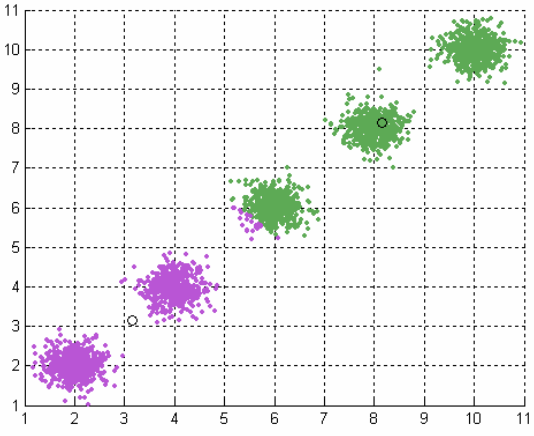

Figura 7 – Melhor distribuição (incorreta) do problema 3, mostrada pela medida de validação *ICC*.

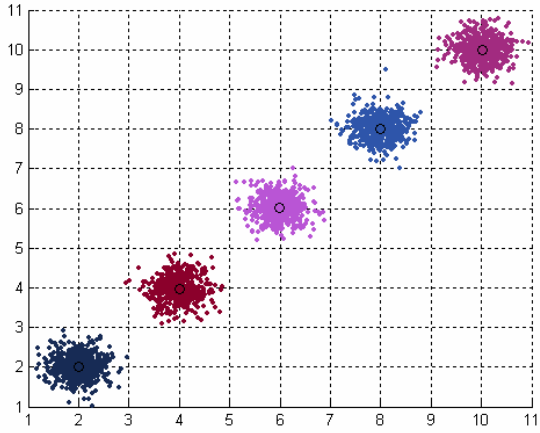

Figura 8 – Melhor distribuição (correta) do problema 3, mostrada pela medida de validação *CS*.

6. Conclusões:

As medidas de validação CS e ICC são bastante necessárias pois, numa aplicação real, não seria conhecido o número de clusters do algoritmo FCM que melhor particionaria os pontos.

No entanto, os resultados parecem não mostrar superioridade de um método sobre o outro, considerando que o melhor resultado foi obtido pelo *ICC* no problema 2 e pelo *CS* no problema 3.

Porém, a correta observação mostra que o erro do algoritmo *ICC* no problema 3 poderia ser evitado se ao invés de basear a decisão no valor absoluto obtido pelo algoritmo, fosse verificado o ponto onde a derivada do mesmo, em relação ao número de clusters, é mínima (máxima negativa).

Esta constatação pode ser verificada pelos valores da tabela 5. As distribuições apresentam valores altos em 2, 3, 4 e 5 clusters, ocorrendo redução brusca a partir de 6 clusters. Detectar o ponto onde ocorre a redução brusca é equivalente a detectar o ponto de mínimo da derivada dos resultados.

A tabela mostra os resultados da aplicação da derivada do *ICC*, onde aponta-se corretamente o número de clusters como 5.

Tabela 7 – Valores da derivada do *ICC* para o problema 3.

Tentativas	Valores da derivada do *ICC* de 2/3 a 9/10 clusters							
	2	3	4	5	6	7	8	9
1	-15.4769	-20.4761	13.1882	-80.8473	1.5472	1.5748	3.1881	-1.2384
2	-15.7112	-20.3631	13.2779	-81.3795	1.5471	1.3571	1.9536	-0.3659
3	-15.6357	-22.1617	15.0069	-80.7315	0.8596	1.5192	1.8478	0.7242
4	-15.6892	-22.0749	14.9737	-81.3714	1.5698	1.3655	1.3735	0.2979
5	-15.4802	-20.4228	13.1513	-80.8657	1.0466	0.5942	2.1839	1.2662
6	-15.6678	-22.1656	15.0446	-81.2713	1.4416	1.4848	0.6476	2.4681
7	-15.63	-20.3849	13.1871	-80.8021	0.9884	2.1256	0.7076	1.1543
8	-15.7247	-22.0007	14.9342	-81.2736	3.6921	-0.864	1.4051	1.2107
9	-15.7466	-21.8571	14.7738	-81.3377	1.5378	1.9828	0.8379	1.2957
10	-15.5515	-20.4492	13.2024	-80.8419	0.5827	2.5446	0.5847	2.5927

7. Referências:

[1] FRANCO, Cláudia Rita de. Novos Métodos de Classificação Nebulosa e de Validação de Categorias e suas Aplicações a Problemas de Reconhecimento de padrões. Rio de Janeiro, UFRJ/IM-DCC-NCE, 2002. Dissertação (Mestrado em Informática).

[2] Silva, Eugênio. Uso de técnicas de validação em clusterização. Trabalho da disciplina de sistemas nebulosos, UFRJ/IM-DCC-NCE. 2002.

[3] de Oliveira, Adriano Joaquim Cruz. Lógica nebulosa, apostila do curso de

lógica nebulosa. UFRJ/IM-DCC-NCE. Dezembro de 2003.

[4] Matlab software versão 6.0.0.8. Mathworks Inc. www.mathworks.com. 2004.

[5] FRANCO, Cláudia Rita de. A Validity Measure for Hard and Fuzzy Clustering derived from Fisher's Linear Discriminant, IEEE International Conference on Fuzzy Systems. World Congress on Computational Intelligence (WCCI), Honolulu, Maio, 2002.