1. Introdução

O presente trabalho visa codificar, aplicar e avaliar uma técnica de agrupamento e algumas técnicas já consagradas para a resolução de problemas de classificação classificação. Para tanto uma base de dados foi gerada artificialmente, a partir do Toolbox do Matlab, obedecendo distribuições normais bivariadas, com parâmetros pré-definidos.

O trabalho contemplou a codificação do algoritmo *fuzzy* de agrupamento FCM-DFCV e a análise de seus resultados. Foi objeto ainda deste estudo, a codificação e aplicação do algoritmo EM (*Expectation-Maximization*), da estimação por máxima verossimilhança, da estimação de densidades pelo método Janela de Parzen, a codificação e uso do método de estimação de probabilidades a posteriori K-NN (K Vizinhos), bem como a combinação, uso e avaliação destes classificadores. Os resultados foram analisados visando aferir resultados e subsidiar escolhas por algoritmos ou combinações.

A seção 1 do presente documento contempla a base de dados utilizada. A seção 2 apresenta o FCM-DFCV e a análise de seus resultados. A seção 3 engloba os algoritmos de classificação utilizados, dentro do paradigma supervisionado de aprendizagem, a combinação dos classificadores e os resultados alcançados.

2. Dados Utilizados

Os dados utilizados foram artificialmente gerados a partir do toolbox do Matlab. Foram gerados duas classes, sendo uma delas uma mistura de gaussianas bivariadas. A parametrização dos dados está descrita na tabela 1.

Tabela 1 – Parametrização dos dados gerados.

Classe \ Parâmetros	**μ1**	**μ2**	**σ1**	**σ2**
Classe1 (componente 1)	0.00	0.00	2.00	1.00
Classe1 (componente 2)	4.00	3.00	2.00	1.00
Classe 2	0.00	3.00	0.50	0.50

Foram dadas ainda as covariâncias entre as componentes da classe e 1 e a classe 2, a partir de onde se extrai os respectivos coeficientes de correlação. Um total de 300 padrões foi gerado, dos quais 200 são pertencentes à classe 1 e 100 à classe 2. A classe 1 foi gerada com igual probabilidade de componentes, de forma que foram produzidos 100 padrões para cada componente da classe.
O gráfico de dispersão das classes encontra-se na figura 2 adiante.

3. FCM-DFCV

O algoritmo utilizado para a geração dos clusters é uma derivação do algoritmo padrão, o FCM (*Fuzzy C-Means*), onde este último busca minimizar da distância entre padrões e respectivos centros, considerando, para tal, os graus de pertinência de um padrão a cada cluster. O FCM permite, desta forma, que um dado padrão pertença a mais de um cluster, a partir da noção *Fuzzy* de pertinência. O FCM atua como uma versão nebulosa do algoritmo *K-means,* sendo utilizado para dispor um universo de amostras em categorias nebulosas de acordo com sua disposição no espaço euclidiano.

O FCM-DFCV (FCM *Diagonal Fuzzy Covariance Matrix*) é baseado em distância quadrática adaptativa. Busca minimizar o critério de adequação entre *clusters* e seus protótipos, que é dado pela expressão:

$$J5 = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m d^2_{\mathbf{M}_i}(\mathbf{x}_k, \mathbf{g}_i) = \sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m (\mathbf{x}_k - \mathbf{g}_i)^{\mathrm{T}}\mathbf{M}_i(\mathbf{x}_k - \mathbf{g}_i),$$

Onde:

$\mathbf{x_k}$ é um padrão da amostra (um vetor de características);

$\mathbf{g_i}$ é um protótipo. Cada *cluster* é definido por uma partição, P_i (i= 1, ..., c) e possui um protótipo correspondente g_i (i = 1,...,c);

$\mathbf{u_{ik}}$ representa o grau de pertinência do padrão k no cluster P_i. Cada valor u_{ik} estará contido numa matriz u (c x n), a matriz de graus de pertinência;

m representa um parâmetro o grau de nebulosidade das pertinências de cada padrão;

$\mathbf{M_i}$ é a matriz de covariância *fuzzy*, representada por uma matriz diagonal positiva associada a cada *cluster* (i = 1,...,c).

A inicialização do algoritmo FCM-DFCV se dá a partir da escolha aleatória dos centros, o vetor de protótipos $\mathbf{g_i}$. A matriz $\mathbf{M_i}$ é inicializada como uma matriz diagonal unitária. A matriz de pertinência **u** (c x n) é inicializada, com base nos valores de $\mathbf{g_i}$ e $\mathbf{M_i}$, de acordo com a expressão seguinte

$$u_{ik} = \left[\sum_{h=1}^{c} \left\{ \frac{\sum_{j=1}^{p} (x_k^j - g_i^j)^2}{\sum_{j=1}^{p} (x_k^j - g_h^j)^2} \right\}^{1/(m-1)} \right]^{-1}$$

onde 'c' é o número de *clusters* e 'p' é a dimensão do espaço de características. A partir deste ponto, o algoritmo se alterna entre dois passos, onde no primeiro, chamado Representação, o vetor de protótipos é atualizado de acordo com a expressão

$$\mathbf{g}_i = \frac{\sum_{k=1}^{n} (u_{ik})^m \mathbf{x}_k}{\sum_{k=1}^{n} (u_{ik})^m}.$$

O segundo passo é composto por duas etapas. Na primeira etapa os elementos da diagonal principal (λ_{ij}) da matriz $\mathbf{M_i}$ são atualizados de acordo com a expressão seguinte.

$$\lambda_i^j = \frac{\{\prod_{h=1}^{p} [\sum_{k=1}^{n} (u_{ik})^m (x_k^h - g_i^h)^2]\}^{1/p}}{\sum_{k=1}^{n} (u_{ik})^m (x_k^j - g_i^j)^2}$$

Na etapa seguinte do segundo passo, chamada Alocação, o grau de pertinência de cada padrão a um dado *cluster* é atualizado a partir da atualização da matriz **u** (c x n), conforme expressão já descrita. O algoritmo prossegue com uma alternância entre os dois padrões até que um número máximo de iterações seja atingido, ou, antecipadamente, segundo algum critério de convergência.

Foi fixado um número máximo de 300 iterações no algoritmo ($t_{máx}$ = 300) cuja parada pode ser antecipada ao se obter um valor mínimo de $|J^t - J^{t+1}|$: $|J^t - J^{t+1}| < \varepsilon$, para $\varepsilon << 1$, onde no presente trabalho ε foi fixado em 10^{-10}. Este procedimento foi aplicado 100 vezes aos padrões gerados, com 2 *cluster*s.

Dado que c, o número de *clusters*, foi fixado em 2, a partição *hard* é encontrada diretamente a partir da análise da matriz de pertinência u, após o processamento realizado pelo algoritmo. A um dado padrão $\mathbf{x_k}$, a matriz de pertinência, u, associa dois valores, u(1, k) e u(2, k), e associa o

padrão ao cluster de cuja linha na matriz u, para um k fixo, possui maior valor. Uma melhor visualização pode ser obtida analisando um trecho da matriz u^T:

$$\begin{matrix} \vdots & \vdots \\ 0.4938 & 0.5062 \\ 0.8694 & 0.1306 \\ 0.9670 & 0.0330 \\ 0.9834 & 0.0166 \\ \vdots & \vdots \end{matrix}$$

Desta forma, o primeiro padrão exibido será associado ao *cluster* 2, pois 0.5262 > 0.4938, enquanto os demais serão associados ao *cluster* 1.

3.2 Resultados do Algoritmo FCM-DFCV

Após a aplicação do algoritmo à base de dados, foi selecionada a matriz de pertinência a partir do melhor resultado alcançado na categorização, tomando como parâmetro de escolha a iteração que gerou o menor valor de J entre todas as experimentações realizadas para a base. A figura 1 exibe o agrupamento gerado pelo algoritmo. Desta forma, um *cluster* agrupou 201 padrões (padrões em vermelho) e o outro ficou com os 99 remanescentes (padrões em verde). Na figura 1, cada x circulado representa um centro dado pelo vetor final de protótipos g_i (i = 1, 2).

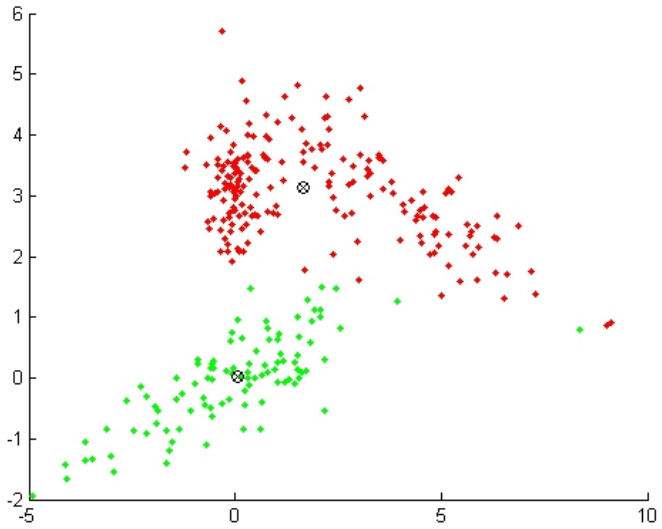

Figura 1 – Agrupamento realizado pelo algoritmo FCM-DFCV. Os padrões agrupados em dois *clusters*, com 201 padrões em verde e 99 em vermelho. Os centros estão destacados com um 'x' circulado.

As classes originais estão exibidas na figura 2. Os elementos marcados em azul, correspondem aos padrões da classe 1, e os vermelhos representam a classe 2.

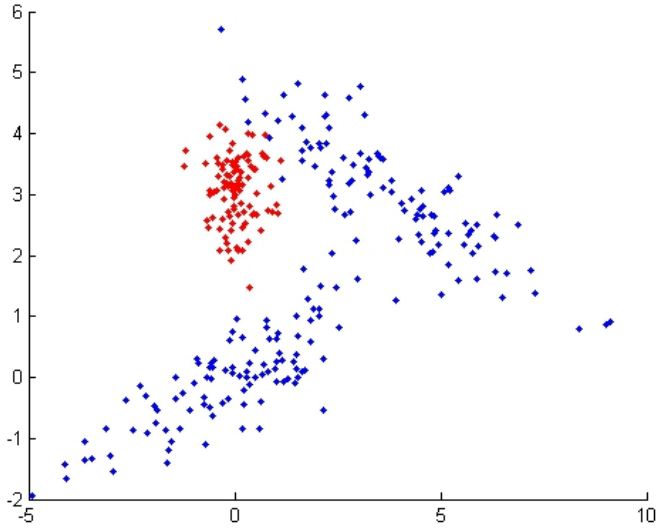

Figura 2 – Rotulação das classes. A classe 1, com 200 padrões está marcada em azul, a classe 2, com 100 padrões, em vermelho.

Os resultados, relativos e índices e taxas estão descritos na tabela 1.

Tabela 1 – Índices de avaliação.

Índice	**Valor**
Índice de Rand Corrigido	0.084004
Erro classificação global	0.3433
Erro classificação classe 01	0.49
Erro classificação classe 02	0.01

O índice de rand corrigido é um índice externo de classificação, que toma como critério de cálculo o quantitativo de elementos pertencentes a cada cluster e a cada classe, ou seja, os rótulos de classe e de grupo. O índice está definido no intervalo [-1,1], onde valores próximos a 1 indicam uma grande similaridade entre os *clusters* gerados e as rotulações de classe, ao passo em que valores negativos, ou mesmo próximos a 0, apontam para um grau relativamente elevado de aleatoriedade nos *clusters* gerados. Entretanto, é intuitivo perceber que esta citada aleatoriedade será fruto da rotulação de classe, onde uma classe rotulada a partir de um atributo com fraca correlação com os demais atributos da base de dados terá maiores chances de gerar valores baixos para o índice de rand corrigido.

O índice de rand corrigido é dado pela expressão:

RC = {(a + d) – (X /M)} / {M – (X /M)}, com X = [(a + b)*(a + c) + (b + d)*(c + d)] e M = a + b + c + d, onde:

a representa a quantidade de combinações, 2 a 2, de padrões rotulados com mesma classe e cluster;

b representa a quantidade de combinações, 2 a 2, de padrões rotulados com mesma classe e diferentes rótulos de cluster;

c representa a quantidade de combinações, 2 a 2, de padrões com diferentes rotulações de classe e mesmas rotulações de *cluster*;

d representa a quantidade de combinações, 2 a 2, de padrões rotulados com diferentes rotulações de classe e diferentes rótulos de *cluster*.

O índice atingirá o valor 1 quando a + d for igual a M, o que em outras palavras significa que cada *cluster* conterá elementos de apenas uma das duas classe, maximizando assim os termos ‘a’ e ‘d’, ao passo em que ‘b’ e ‘c’ assumem valor 0.

No presente trabalho, o índice de rand corrigido apresentou valor baixo (0.084004), o que se justifica pelo fato de a classe 1 possuir duas componentes com características significativamente distintas, o que faz o algoritmo FCM-DFCV agrupar grande parte dos dados gerados por uma das componentes da classe 1 no *cluster* que englobou grande parte dos padrões da classe 2. Este fato pode ser mais bem visualizado a partir da inspeção das figuras 1 e 2.

O erro de global de classificação foi obtido mediante a comparação entre as rotulações geradas pelo *cluster* e os rótulos de classe originais. O número de vezes em que os rótulos coincidiram foi dividido pela quantidade de padrões. Este valor é subtraído de 1, caso maior que 0.5, gerando assim o valor de erro global de classificação. Procedimento análogo foi utilizado para o cálculo do erro por classe.