**UNIVERSIDADE FEDERAL DE MINAS GERAIS - UFMG**
**ELE075 - Sistemas Nebulosos - 2018/2**

**Professor:** Cristiano Leite de Castro

# Trabalho Computacional 2 - Inferência Nebulosa

**Aluno:** Rafael Carneiro de Castro **Matrícula:** 2013030210
**Curso:** Engenharia de Sistemas

## 1 - Introdução:

Este trabalho consiste do estudo da inferência nebulosa a partir de dois exercícios. O primeiro deles é a aproximação da função *cosseno* entre $-\pi/2$ e $3\pi/2$ utilizando o mecanismo de inferência de Sugeno, conforme visto em sala de aula. O segundo exercício é o projeto de um classificador binário baseado em regras nebulosas. O número de regras e as funções de pertinência para os antecedentes e consequentes das regras devem ser definidas com base no algoritmo de agrupamento *Fuzzy K-Means*.

## 2 - Aproximação da Função *cosseno*:

Como já mencionado, o primeiro exercício consiste na aproximação da função *cosseno* no intervalo $[-\pi/2;\ 3\pi/2]$. Esta função está representada na Figura 1.

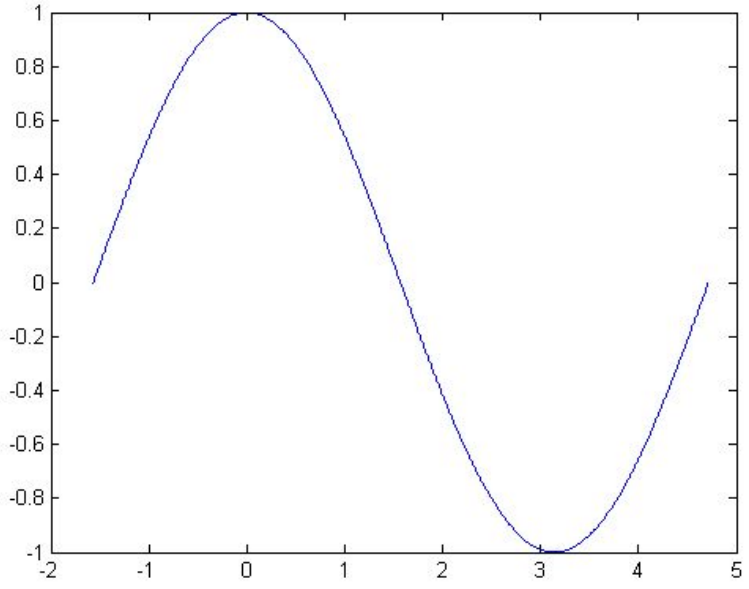


**Figura 1: Função cosseno**

O primeiro passo para se fazer a aproximação pela inferência de Sugeno é encontrar os consequentes das regras *fuzzy*. Neste caso serão usados consequentes de primeira ordem. Por inspeção à Figura 1, pode-se notar que é possível usar três funções de primeira ordem para uma possível aproximação. São elas:

$$(1)\quad f1 = \frac{2}{\pi}x + 1$$
$$(2)\quad f2 = \frac{-2}{\pi}x + 1$$
$$(3)\quad f3 = \frac{2}{\pi}x - 3$$

sendo que a equação (1) é aplicada no intervalo $[-\pi/2;\ 0]$, a equação (2) no intervalo $[0;\ \pi]$ e a equação (3) no intervalo $[\pi;\ 3\pi/2]$. A Figura 2 representa visualmente as três funções em seu intervalos, bem como a função cosseno para contraste.

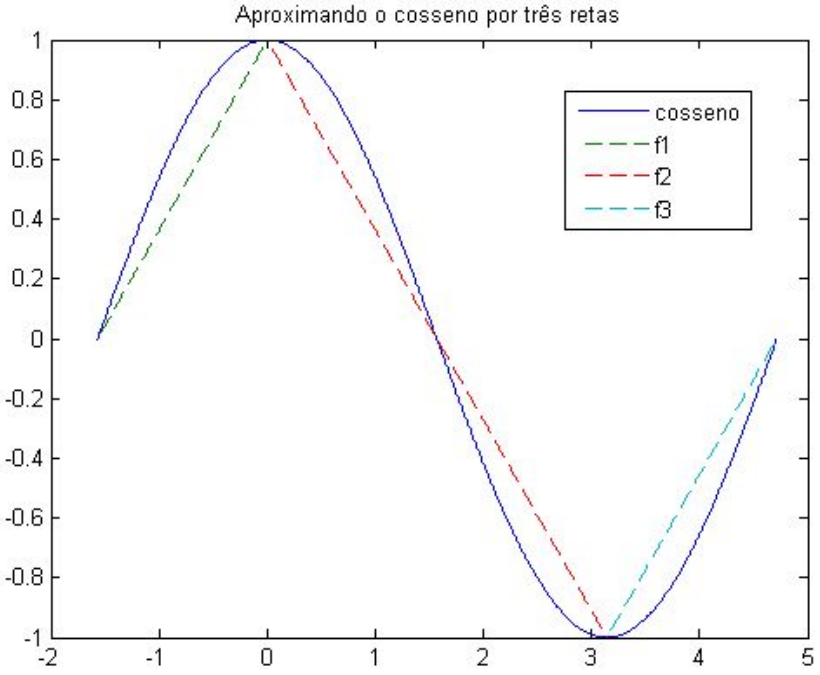


**Figura 2: Aproximando o cosseno por três retas**

Agora é possível se definir as regras *fuzzy*, que possuem tais funções como consequente. As regras são:

- *Se x é A1, y é f1*
- *Se x é A2, y é f2*
- *Se x é A3, y é f3*

As funções de pertinência A1, A2 e A3 nas regras devem ser "*fuzzyficações"* da variável x, e conforme sugerido nas especificações, pode-se usar funções de pertinência do

tipo triangular. As funções utilizadas aqui estão representadas na Figura 3 e são definidas por:

- *A1 = trimf(x, [-pi/2, -pi/2, pi/2])*
- *A2 = trimf(x, [-pi/2, pi/2, 3*pi/2])*
- *A3 = trimf(x, [pi/2, 3*pi/2, 3*pi/2])*

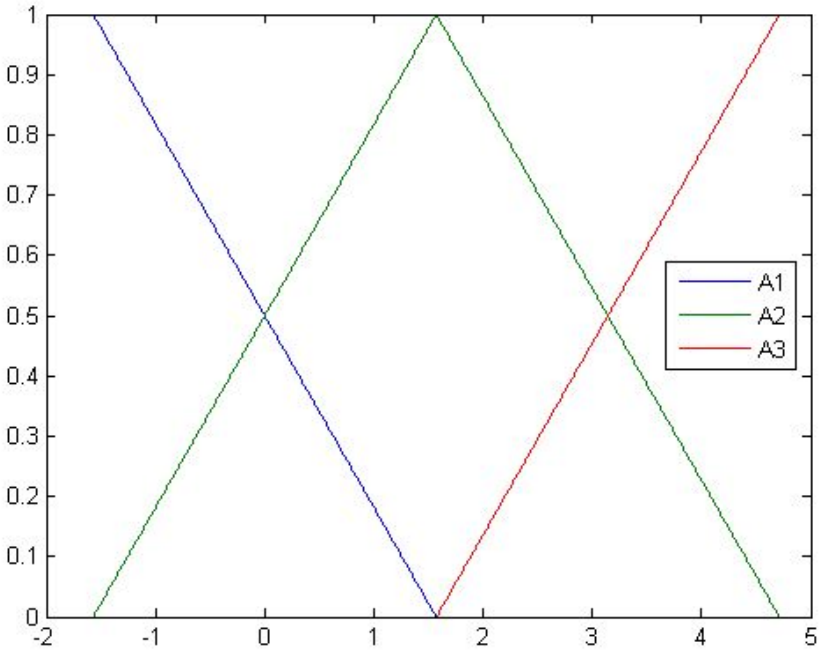


**Figura 3: Funções de pertinência triangulares**

Pela inferência de Sugeno, é possível utilizar as funções de pertinência A1, A2 e A3 para se obter o valor dos pesos w1, w2 e w3, respectivamente, de cada uma delas para cada valor de x, e combinar com as equações (1), (2) e (3) para se obter a aproximação final, conforme a equação (4) a seguir:

$$(4) \quad z = \frac{w1f1 + w1f2 + w2f3}{w1 + w2 + w3}$$

Todos estes passos foram feitos no *MATLAB*, conforme script *main.m* anexo junto a esta documentação, e o resultado gráfico da aproximação final pode ser visto na Figura 4. Nela é mostrada a função *cosseno,* bem como a aproximação obtida pelo método de inferência de Sugeno utilizando os consequentes e funções de pertinência aqui apresentados. Como se pode notar, a aproximação foi bem satisfatória.

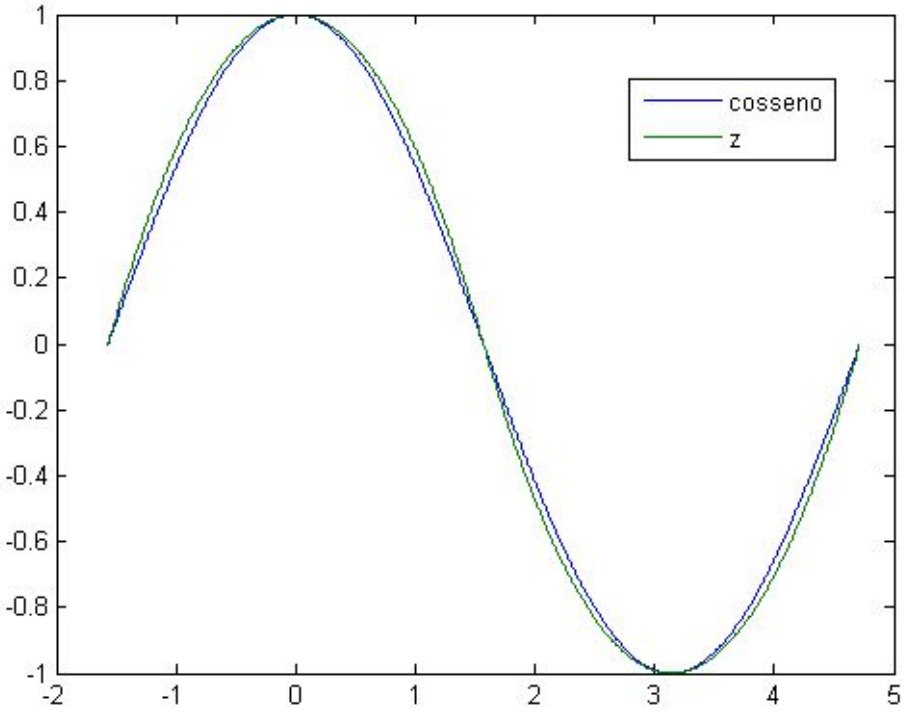


**Figura 4: Aproximação final da função cosseno**

Para uma análise quantitativa dos resultados, pode-se calcular o Erro Quadrático Médio, dado pela fórmula:

$$EQM = \frac{1}{N}\sum_{i=1}^{N}(y_i - z_i)^2$$

onde $y_i$ é a saída real da função *cosseno* e $z_i$ é a saída obtida pelo sistema nebuloso, ou seja, a aproximação. Este cálculo foi feito com o auxílio do *MATLAB*, também no script *main.m*, e o resultado obtido foi:

$$EQM = 0.001283$$

que é um valor pequeno, representado o baixo erro da aproximação final.

## 3 - Classificador *Fuzzy*:

O próximo exercício consiste do projeto de um classificador binário baseado em regras nebulosas. O algoritmo de agrupamento *Fuzzy K-Means* será utilizado para construir as regras nebulosas. O classificador será testado, primeiramente, com uma base de dados disponibilizada pelo professor, contida no arquivo *dataset_2d.mat*. Em seguida ele também será testado com uma base real, a do problema *Pima Diabetes*, contida no arquivo *diabetes.csv*. Ambos os arquivos das bases de dados estão anexos a este relatório.

### *3.1 - Separação do Conjunto de Dados:*

O primeiro passo para a criação e teste do algoritmo é a separação dos dados. Conforme especificação, o conjunto de dados $X$ com dimensões $n \times d$ (onde $n$ é o número de amostras e $d$ é a dimensão do espaço de entrada) e a matriz $Y$ com dimensões $n \times 1$, correspondendo aos rótulos das amostras, deve ser dividido entre treinamento e validação. 70% dos dados do conjunto devem ser utilizados para o treinamento, enquanto que os outros 30% devem ser utilizados para validar o modelo obtido.

Neste trabalho optou-se por fazer uma separação dos dados de forma balanceada. Desta forma, tanto o conjunto de treinamento quanto o conjunto de validação terão aproximadamente a mesma proporção de amostras de cada classe. Este ponto é importante para evitar viés de treinamento ou de validação provocado pelo fato de um dos grupos ter apenas amostras de uma classe, ou proporcionalmente muito mais amostras de uma classe do que de outra. Vale salientar que o conjunto de dados é separado de forma aleatória entre treinamento e validação.

### *3.2 - Definição das Regras pelo Fuzzy K-Means:*

As regras nebulosas deste classificador são definidas pelo algoritmo de agrupamento *Fuzzy K-Means*. As amostras de treinamento são submetidas à *clusterização*, com o auxílio da função *fcm* do MATLAB. Vale salientar que a quantidade de grupos $K$ é um parâmetro do classificador, e neste trabalho o valor de $K$ irá variar entre 2 e 8.

Após a definição de grupos pelo *Fuzzy K-Means,* o centróide do *j*-ésimo grupo deve corresponder a uma regra do tipo:

$$regra\ j:\ se\ x_1 \text{ é } A_{1j}\ e\ x_2 \text{ é } A_{2j}\ e\ ...\ e\ x_d \text{ é } A_{dj}\ então\ y_j = c$$

Na regra, o antecedente $A_{ij}$ é a função de pertinência Gaussiana com centro $c_{ij}$ igual à projeção do centróide do grupo $j$ na variável de entrada $i$. A dispersão $\sigma_{ij}$ da Gaussiana será explicada no próximo tópico. O valor de $c$ no consequente é definido como sendo a classe (0 ou 1) que fornece o valor máximo para a soma dos valores de pertinência do grupo $j$. Para obtê-lo, basta somar os valores de pertinência do grupo $j$ por classe e usar a matriz de treinamento $Y$ para descobrir qual é a classe do padrão.

### *3.3 - Definição da Dispersão das Funções de Pertinência:*

Para a definição da dispersão $\sigma_{ij}$, será utilizado o desvio padrão dos pontos do grupo $j$ em relação ao centróide daquele grupo. Desta maneira, a dispersão de cada Gaussiana pode ser calculada por:

$$\sigma_{ij} = \sqrt{\frac{(\sum_{k=1}^{d_j} x_k - c_j)^2}{d_j}}$$

Nesta equação, $d_j$ é a quantidade de pontos no grupo $j$, $x_k$ são os pontos que fazem parte daquele grupo e $c_j$ é o centróide do grupo.

### *3.4 - Saída do Classificador:*

Para definir a saída do classificador, ou seja, a classe para as amostras, basta fazer as seguintes operações:

- Para cada regra $j$, calcular os valores de pertinência $\mu A_{ij}(x_i)$ para cada variável de entrada $x_i$ a partir das Gaussianas definidas;
- Para cada regra $j$, obter o grau de ativação $\omega_j = \prod_{i=1}^{d} \mu A_{ij}(x_i)$;
- Agregar os graus de ativação das regras que possuem o mesmo consequente (mesma classe) com o operador soma probabilística e escolher como saída para a amostra a classe que fornece o maior valor agregado. No MATLAB, a soma probabilística pode ser calculada com a função *probor.*

### *3.5 - Resultados:*

O algoritmo do classificador está presente no arquivo *fuzzy_classifier.m*. Este arquivo define uma função que recebe como parâmetro as variáveis de entrada $x$, os rótulos $y$, a quantidade de agrupamentos $K$ que deve ser usada no *Fuzzy K-Means* (e é também a quantidade de regras *fuzzy*) e uma variável $plotGraphs$ que pode ser usada para problemas com amostras de duas variáveis, para plotar os gráficos dos grupos divididos pelo *Fuzzy K-Means* e o resultado final da classificação do grupo de validação. Esta função retorna a acurácia do classificador.

No arquivo *main.m*, a função do classificador foi chamada sete vezes, para $K$ indo de 2 até 8, utilizando a base de dados *dataset_2d.mat*, disponibilizada pelo professor. A acurácia dos sete classificadores treinados foi calculada e plotada. Pode ser vista na Figura 5. Como se pode notar, para $K = 2$, o classificador obteve uma precisão baixa, na faixa de 40%. Para $K = 3$ o classificador já deu um salto de precisão, classificando cerca de 75%

das amostras corretamente. A partir de $K = 4$, a precisão alcança seus maiores níveis, chegando inclusive a 100% em alguns casos, como em $K = 8$, para a base testada.

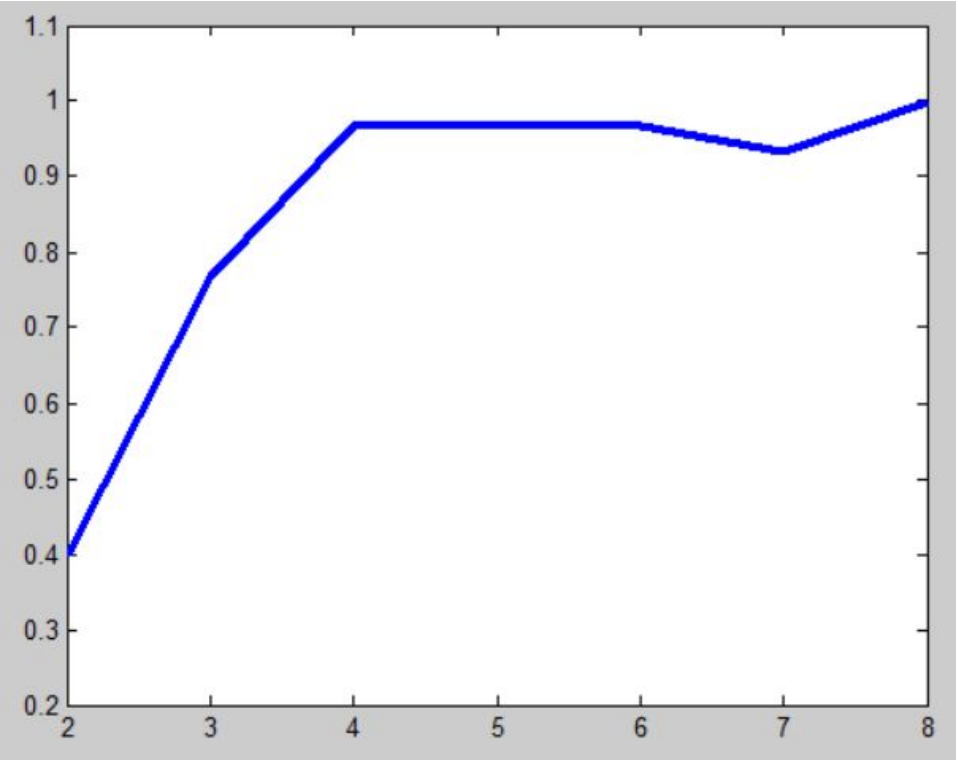


**Figura 5 - Acurácia dos classificadores com K indo de 2 até 8, para a base *dataset_2d***

Na Figura 6 é possível ver a separação em grupos feita pelo *Fuzzy K-Means* (à esquerda) e a classificação executada pelo classificador desenvolvido neste trabalho (à direita), para os valores de $K$ iguais a 2, 3 e 8, de cima para baixo, nesta ordem, mostrando com um x os pontos classificados erroneamente. Como se pode notar, para $K = 2$, o classificador tentou fazer uma separação aproximadamente linear, tendo dois grupos separados, aproximadamente, por uma reta, o que provocou uma grande quantidade de erros. Com $K = 3$, a classificação já se tornou um pouco mais complexa, conseguindo atingir um nível de precisão maior. Já para $K = 8$, com precisão de 100%, é possível notar que a classificação final pode ser dividida por quatro grande grupos, sendo dois deles de uma classe, e os outros dois de outra classe. Por isso, a partir de $K = 4$, a precisão do modelo se tornou maior.

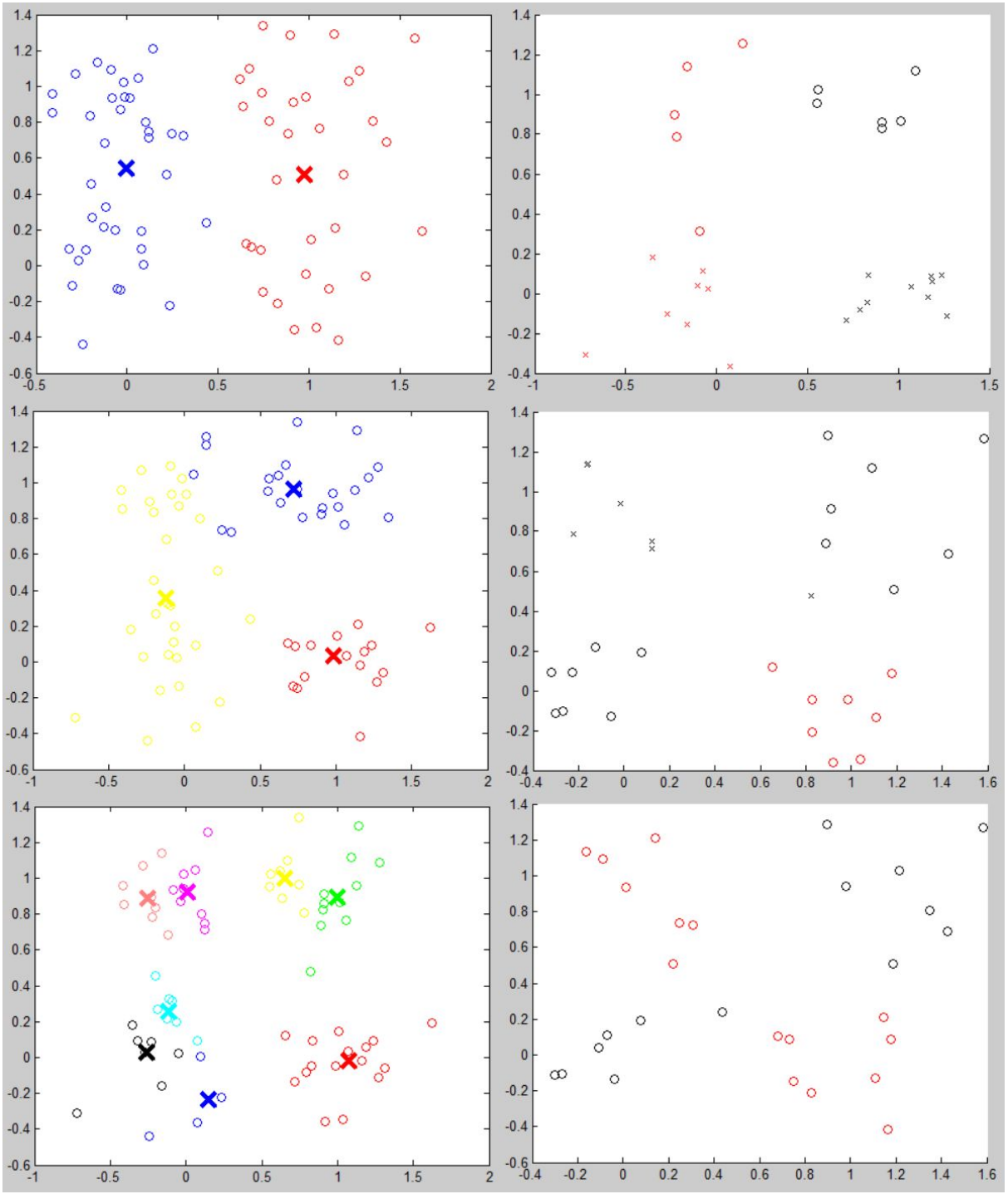

**Figura 6 - Separação dos K grupos e exibição da classificação final**

O modelo também foi testado para a base *Pima Diabetes,* contida no arquivo *diabetes.csv*, também com $K$ variando de 2 até 8. Tal procedimento está contido no arquivo *main.m.* A precisão para os classificadores pode ser vista na Figura 7. Como se pode notar, para $K = 2$, o classificador obteve sua menor precisão, aproximadamente 55%. A partir de $K = 3$ os valores de precisão ficaram entre 65% e 70%, não variando muito entre um valor de $K$ e outro.

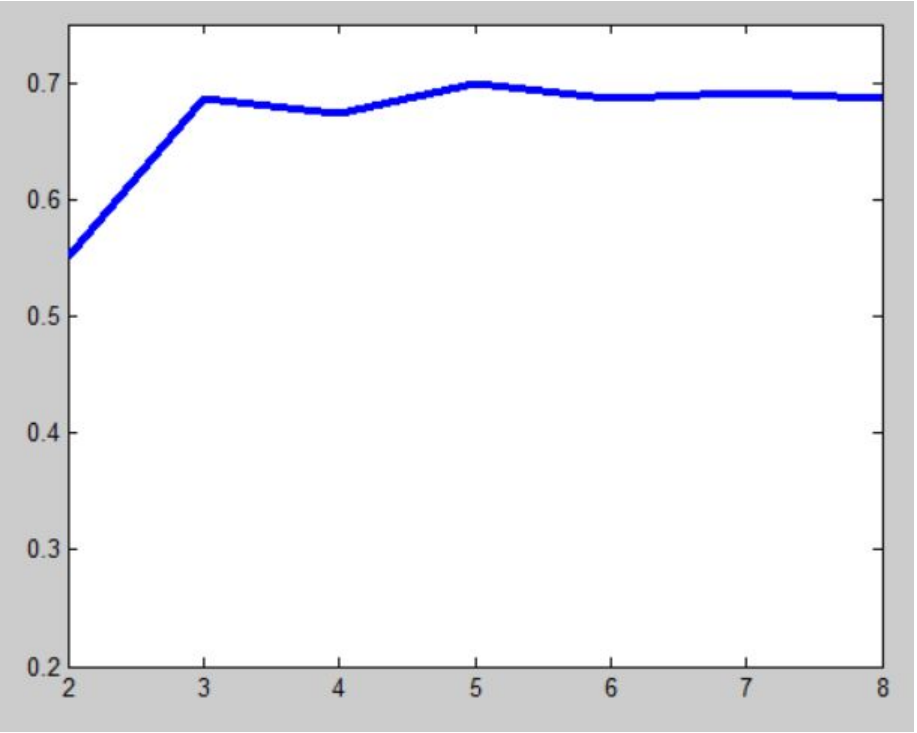


**Figura 7 - Acurácia dos classificadores com K indo de 2 até 8, para a base *Pima Diabetes***

## 4 - Conclusão:

Neste trabalho foi apresentada a aproximação da função *cosseno* a partir dos conhecimentos obtidos na matéria Sistemas Nebulosos, com lógicas *fuzzy*. A partir destes conhecimentos também foi apresentado e testado um classificador binário *fuzzy*. Tais exercícios foram importantes para trabalhar os conceitos vistos em sala de aula, tornando-os mais palpáveis em aplicações práticas.

Alguns desafios foram encontrados, como na definição de funções de pertinência para a aproximação do *cosseno*, e na proposição de uma dispersão para as funções de pertinência no classificador *fuzzy*. Após a superação de obstáculos como estes, conclui-se que o objetivo do trabalho foi alcançado, colocando em prática muitos dos conceitos vistos em sala de aula. O erro quadrático médio para a aproximação do *cosseno* foi bem baixo, o que mostra que a aproximação alcançou um nível aceitável. As precisões alcançadas nos testes do classificador *fuzzy* também obtiveram níveis aceitáveis. Para a base *Pima Diabetes* classificada aqui, utilizando estratégias de Aprendizado de Máquina, como a Regressão Logística combinada com o SVM Linear, é possível alcançar acurácia de 78%. Conclui-se então que os 70% alcançados neste trabalho são um valor de acurácia razoável, enquadrado com os objetivos do trabalho.