



SCC5895 – Análise de Agrupamento de Dados

Algoritmos Particionais: Parte II

Prof. Eduardo R. Hruschka

PPG-CCMC / ICMC / USP



Créditos

- Este material consiste de adaptações e extensões dos originais elaborados por Ricardo J. G. B. Campello e Eduardo R. Hruschka

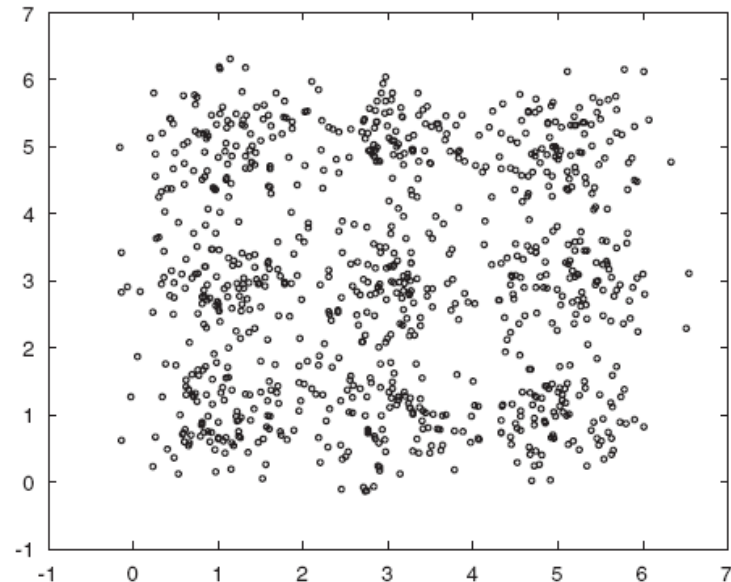


Aula de Hoje

- Partições com Sobreposição
 - Soft, Fuzzy e Probabilísticas
- Algoritmos Particionais com Sobreposição
 - Fuzzy C-Means (FCM)
 - Gustafson-Kessel (GK)
 - Expectation Maximization (EM)
- Algoritmos Baseados em Densidade
 - DBSCAN

Algoritmos de Partição Com Sobreposição

- Algoritmos particionais como o k-means, k-medóides e diversos outros produzem uma **partição rígida** da base de dados:
 - Cada objeto pertence a um único grupo, de forma integral
 - Usualmente refere-se a esse tipo de partição como **Hard** ou **Crisp**
- No entanto, muitos problemas envolvem grupos mal delineados, que não podem ser separados adequadamente dessa maneira
- Em outras palavras, existem situações nas quais os dados compreendem categorias que se sobrepõem umas às outras em diferentes níveis
- Por exemplo:



Algoritmos de Partição Com Sobreposição

- Métodos de agrupamento com sobreposição, ou **overlapping clustering algorithms**, são concebidos para lidar com tais situações
- Podem gerar partições de 3 tipos:
 - **Soft**: Objetos podem pertencer a mais de um grupo, porém necessariamente de forma integral
 - **Fuzzy**: Objetos pertencem a todos os grupos, com diferentes graus ou níveis de pertinência (possivelmente nulo)
 - **Probabilísticas**: Objetos possuem probabilidades de pertinência associadas a cada grupo
- Vamos discutir representantes clássicos dos últimos dois tipos, que são os mais comumente utilizados

Definição de Partição Soft de Dados

- Consideremos um conjunto de N objetos a serem agrupados: $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$
- **Partição Soft:** coleção de k grupos, dada por $\mathbf{P} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_k\}$, tal que:

$$\mathbf{C}_1 \cup \mathbf{C}_2 \cup \dots \cup \mathbf{C}_k = \mathbf{X}$$

$$\mathbf{C}_i \neq \emptyset$$

- Exemplo: $\mathbf{P} = \{ (\mathbf{x}_1, \mathbf{x}_4), (\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6), (\mathbf{x}_2, \mathbf{x}_5) \}$

Matriz de Partição Soft

- **Matriz de Partição** (revisão): matriz com k linhas (no. de grupos) e N colunas (no. de objetos) na qual cada elemento μ_{ij} indica o *grau de pertinência* do j -ésimo objeto (\mathbf{x}_j) ao i -ésimo grupo (\mathbf{C}_i)

$$\mathbf{U}(\mathbf{X}) = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2N} \\ \vdots & & \ddots & \vdots \\ \mu_{k1} & \mu_{k2} & \cdots & \mu_{kN} \end{bmatrix}$$

- **Matriz de Partição Soft**: matriz de partição **binária**, ou seja, $\mu_{ij} \in \{0,1\}$, sem qualquer outra restrição, exceto colunas não nulas

Matriz de Partição Soft

- **Exemplo:**

- $\mathbf{P} = \{ (\mathbf{x}_1, \mathbf{x}_4), (\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6), (\mathbf{x}_2, \mathbf{x}_5) \}$

$$\mathbf{U}(\mathbf{X}) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Partições Fuzzy e Probabilísticas

- **Matriz de Partição Fuzzy:** elementos assumem valores contínuos de pertinência, ao invés de binários, i.e., $\mu_{ij} \in [0,1]$

$$\mathbf{U}(\mathbf{X}) = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2N} \\ \vdots & & \ddots & \vdots \\ \mu_{k1} & \mu_{k2} & \cdots & \mu_{kN} \end{bmatrix}$$

- **Matriz de Partição Probabilística:** matriz de partição fuzzy particular, tal que pertinências representam probabilidades
 - Logo, a seguinte restrição deve ser satisfeita: $\sum_i(\mu_{ij}) = 1 \quad \forall j$

Partições Fuzzy e Probabilísticas

- **Exemplo (Fuzzy):**

$$\mathbf{U}(\mathbf{X}) = \begin{bmatrix} 1 & 0.1 & 0.5 & 0.1 \\ 0 & 0.1 & 0.5 & 0.9 \end{bmatrix}$$

- **Exemplo (Fuzzy / Probabilística):**

$$\mathbf{U}(\mathbf{X}) = \begin{bmatrix} 1 & 0.7 & 0.5 & 0.1 \\ 0 & 0.3 & 0.5 & 0.9 \end{bmatrix}$$

Algoritmo Fuzzy c-Means

- **Modelo de Otimização (Dunn 1973, Bezdek 1981):**

$$\min_{\mu_{ij}, \mathbf{v}_i} J = \sum_{j=1}^N \sum_{i=1}^c \mu_{ij}^m \left\| \mathbf{x}_j - \mathbf{v}_i \right\|^2$$

$$s. a \quad 0 \leq \mu_{ij} \leq 1$$

$$\sum_{i=1}^c \mu_{ij} = 1 \quad \forall j \in \{1, 2, \dots, N\}$$

$$0 < \sum_{j=1}^N \mu_{ij} < N \quad \forall i \in \{1, 2, \dots, c\}$$

onde $m > 1$ (parâmetro) e $\mathbf{v}_i \in \mathcal{R}^n$ (protótipos)

Algoritmo Fuzzy c-Means

NOTAS:

- Qual a interpretação do critério de custo J ?
- Qual a interpretação das restrições ?
- Qual o papel do “fator de fuzzificação” m ?

Algoritmo Fuzzy c-Means

- **Algoritmo Básico** ($m = 2, \mathbf{v}_i \neq \mathbf{x}_j$):

1 – Selecionar centros iniciais : $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c$

2 – Calcular μ_{ij} :

$$\mu_{ij} = \left[\sum_{l=1}^c \frac{\|\mathbf{x}_j - \mathbf{v}_i\|^2}{\|\mathbf{x}_j - \mathbf{v}_l\|^2} \right]^{-1}$$

3 – Atualizar os centros :

$$\mathbf{v}_i = \frac{\sum_{j=1}^N \mu_{ij}^2 \mathbf{x}_j}{\sum_{j=1}^N \mu_{ij}^2}$$

4 – Parar em caso de convergência ou voltar ao passo 2

Algoritmo Fuzzy c-Means

- **Algoritmo Completo:**

1 – Seleccionar os centros iniciais : $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c$

2 – Calcular μ_{ij} : Para cada $j \in \{1, \dots, N\}$, se $\|\mathbf{x}_j - \mathbf{v}_i\| > 0$ para $i = 1, \dots, c$ então

$$\mu_{ij} = \left[\sum_{l=1}^c \left(\frac{\|\mathbf{x}_j - \mathbf{v}_i\|}{\|\mathbf{x}_j - \mathbf{v}_l\|} \right)^{\frac{2}{m-1}} \right]^{-1}$$

Se $\|\mathbf{x}_j - \mathbf{v}_i\| = 0$ para $i \in I \subseteq \{1, \dots, c\}$, então definir μ_{ij} para $i \in I$ como \forall número real não negativo que satisfaça $\sum_{i \in I} \mu_{ij} = 1$ e definir $\mu_{ij} = 0$ para $i \in \{1, \dots, c\} - I$

3 – Atualizar os centros :

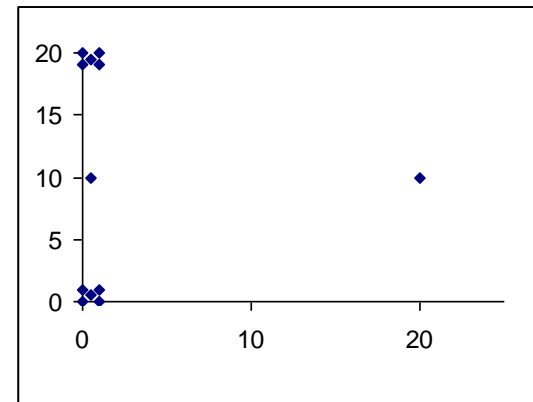
$$\mathbf{v}_i = \frac{\sum_{j=1}^N \mu_{ij}^m \mathbf{x}_j}{\sum_{j=1}^N \mu_{ij}^m}$$

4 – Parar em caso de convergência ou voltar ao passo 2

Algoritmo Fuzzy c-Means

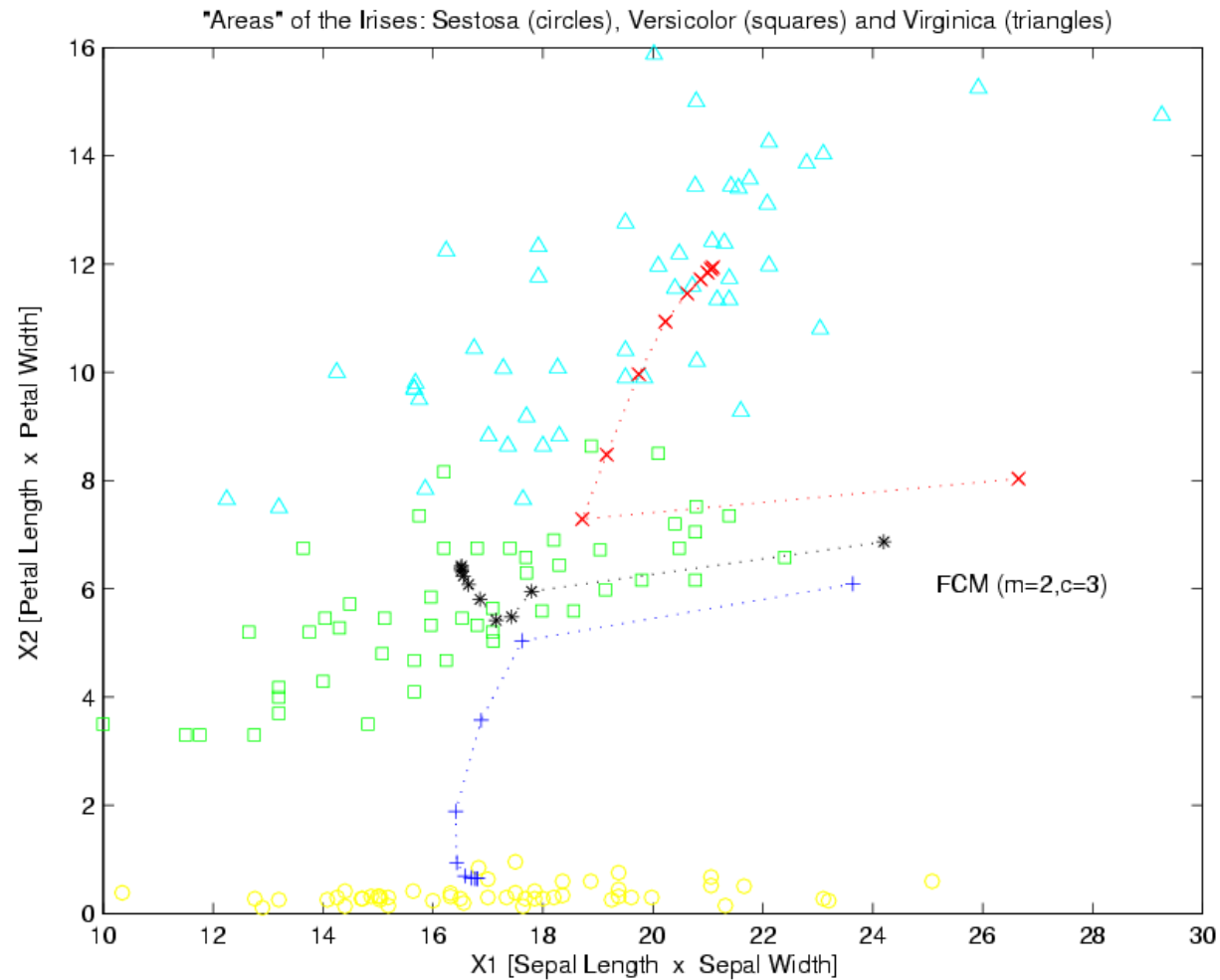
Notas:

- Valor ótimo de m desconhecido. Usualmente $m = 2$
- Trata-se de uma extensão de k-means para o domínio fuzzy
 - Como tal \Rightarrow apenas garantia de convergência para soluções locais !
 - Ou seja, também é susceptível a mínimos locais da função objetivo J
 - depende da inicialização dos protótipos
 - esquemas de múltiplas execuções podem ser utilizados...
- Existem dezenas de variantes
 - e.g. **versão possibilística**
 - considere a figura ao lado com $c = 2$
 - pertinência dos dois outliers... ?

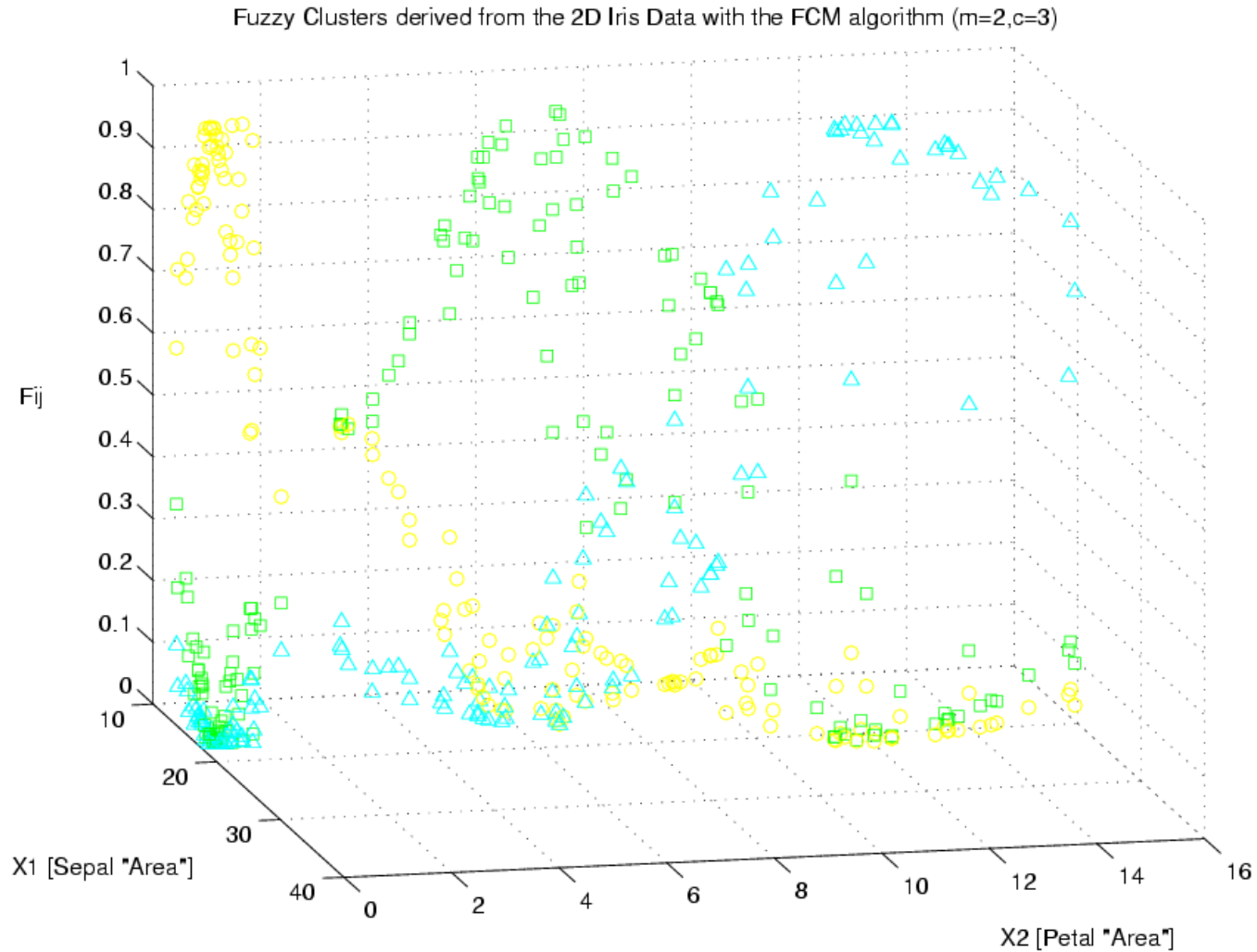


FCM

Exemplo (“Iris” 2D):

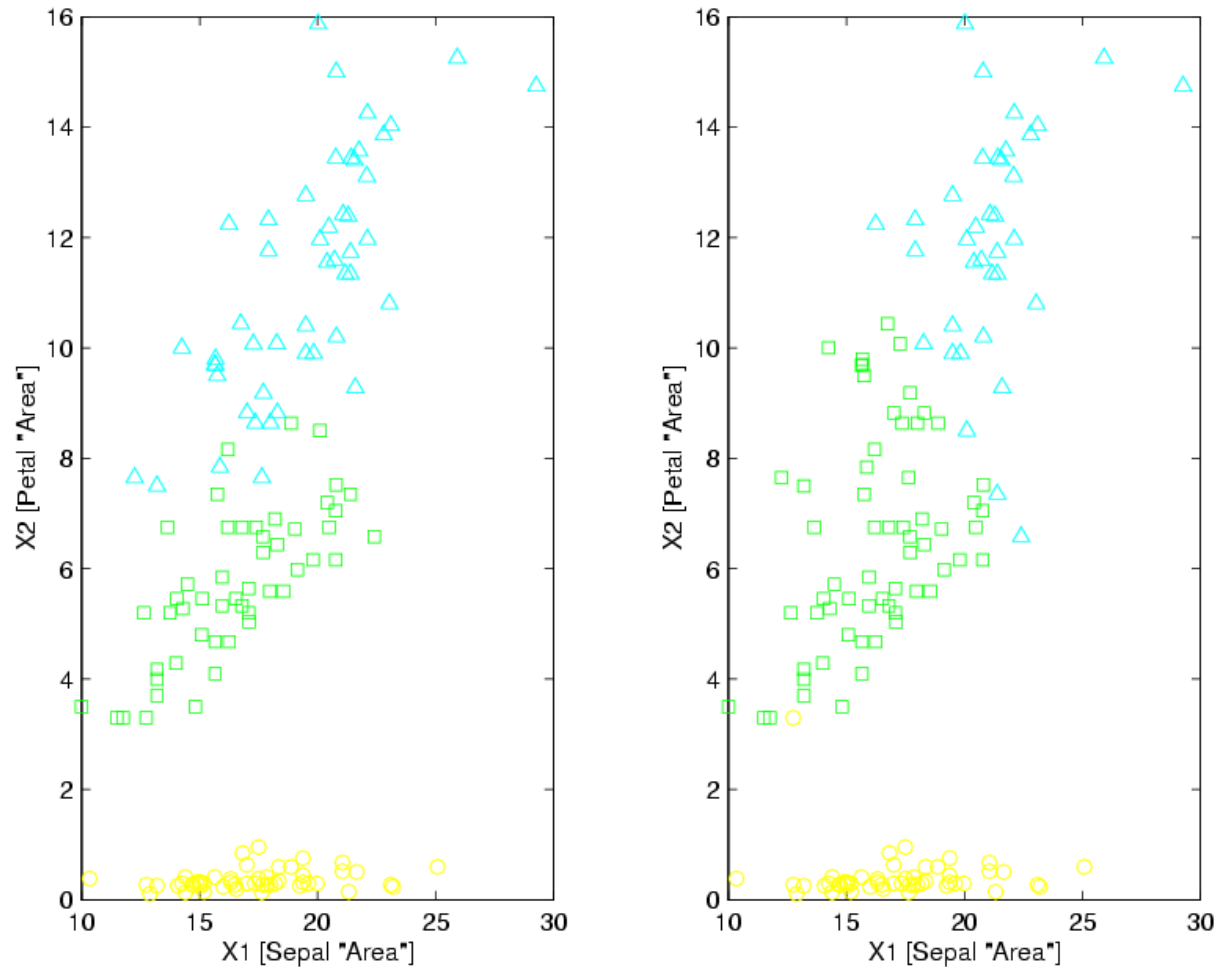


FCM



FCM

Original 2D Iris Data (Left) and Rough Fuzzy C-Means Classification (Right)



Determinação do No. de Grupos...

- Múltiplas execuções ordenadas de FCM, com uso da função objetivo J

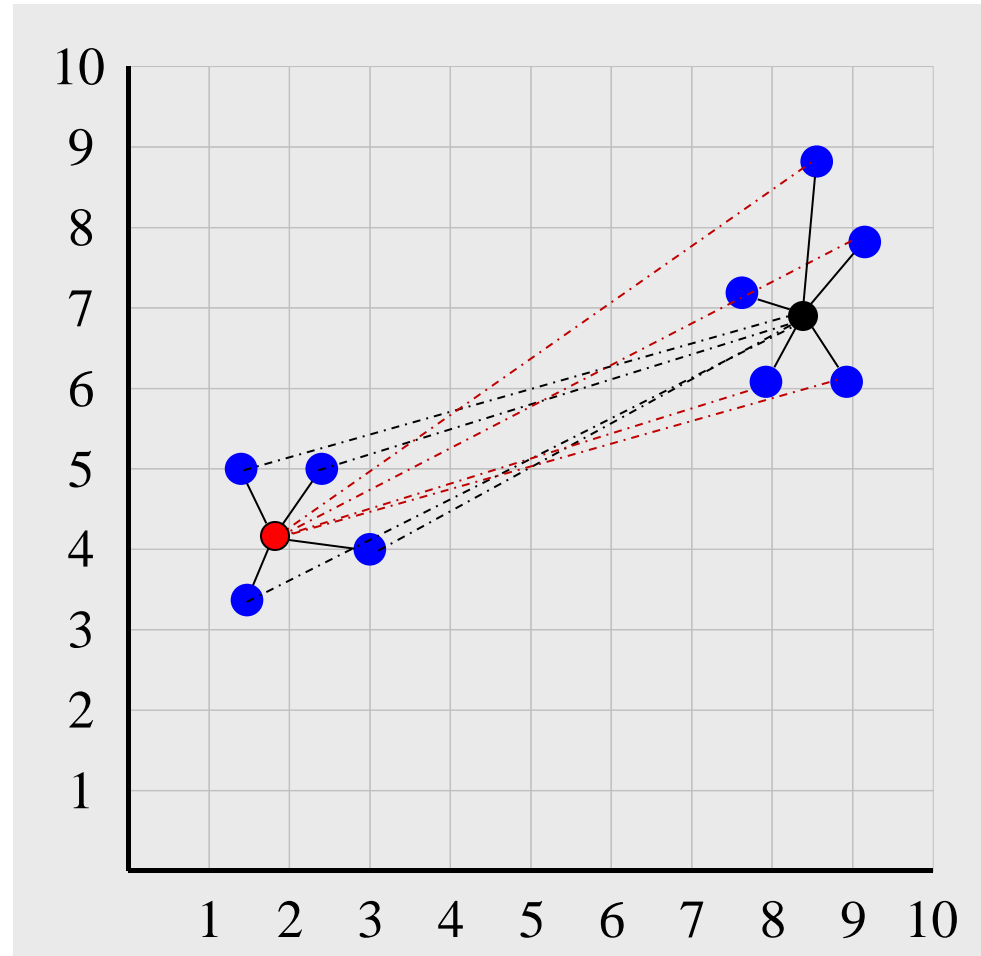
Erro Quadrático

Ponderado:

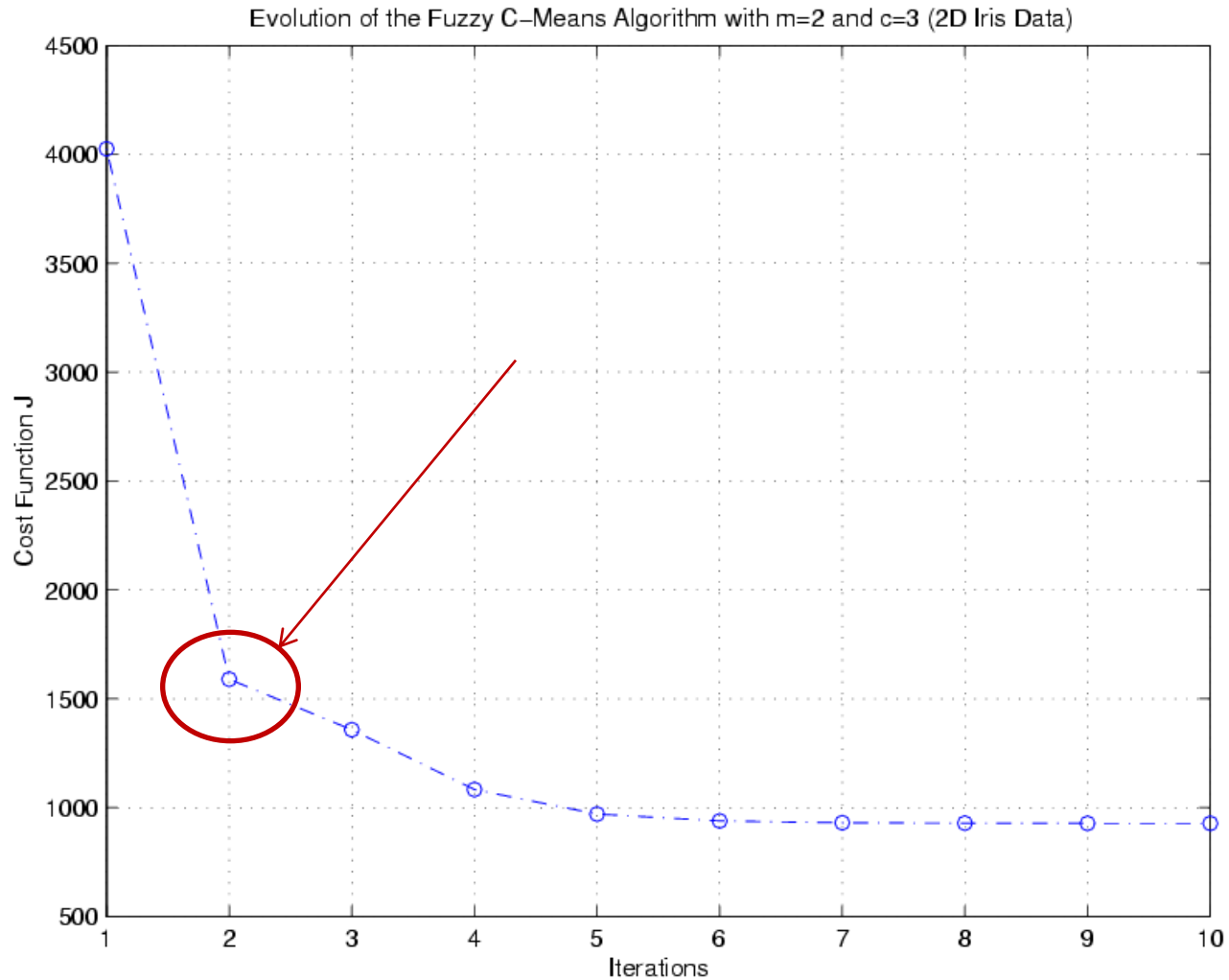
$$J = \sum_{j=1}^N \sum_{i=1}^c \mu_{ij}^m d(\mathbf{x}_j, \mathbf{v}_i)^2$$



Função Objetivo



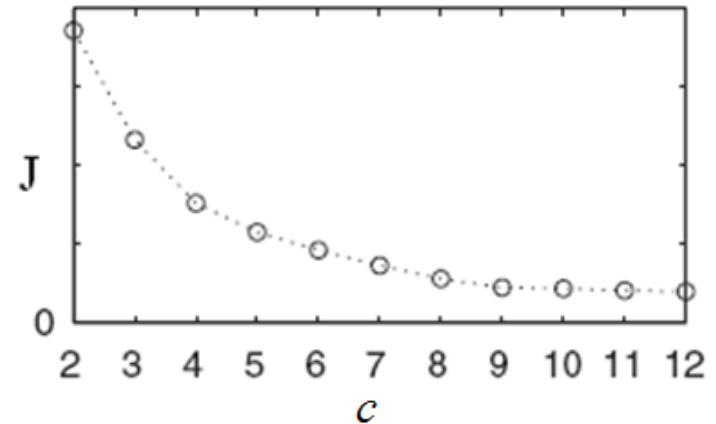
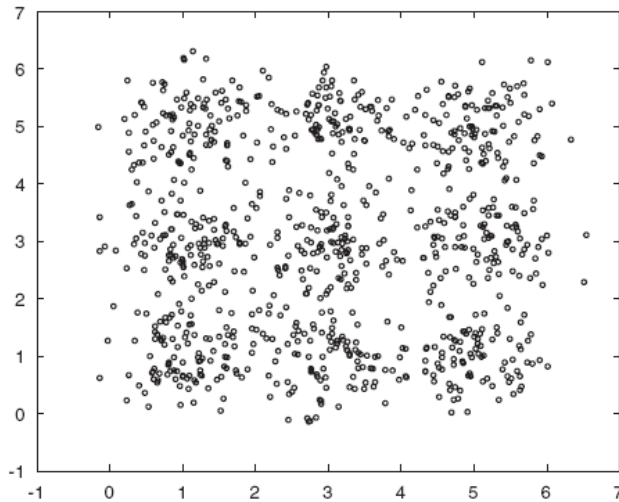
Exemplo (IRIS 2D)



Limitações

- Mesmas já vistas para o k-means:

- 1. Casos Ambíguos:



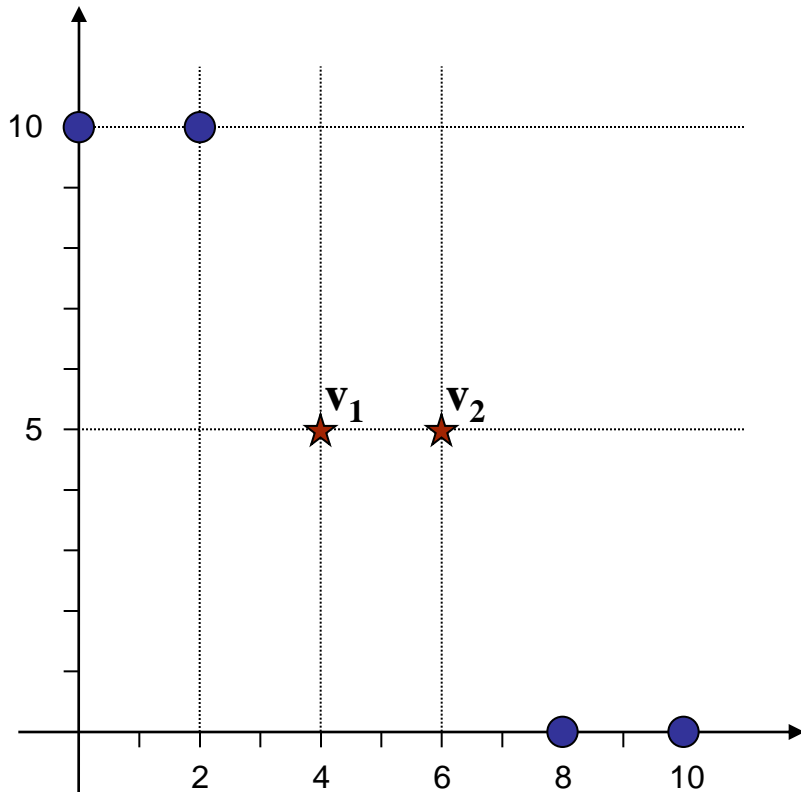
- 2. Busca Guiada:

- FCM evolutivo, ...

- Discussões complementares: aulas sobre **validação**

Exercício

- Agrupar os dados em azul na figura abaixo através do método FCM com 2 clusters, $m=2$ e centros iniciais assinalados em vermelho. Apresentar os centros dos grupos e a matriz de valores de pertinência para cada iteração.



Computacional \Rightarrow 5 iterações.

Manual \Rightarrow 1ª iteração.

FCM Paralelo / Distribuído

- ◆ Conceitos análogos àqueles usados para k-means paralelo/distribuído;
- ◆ Ver, por exemplo: Coletta, Vendramin, Hruschka, Campello, Pedrycz, Collaborative Fuzzy Clustering Algorithms: Some Refinements and Design Guidelines. *IEEE Transactions on Fuzzy Systems*, v. 20, p. 444-462, 2012.

Principal Variante do FCM

◆ Algoritmo Gustafson-Kessel (GK)

- Variante do FCM equipada com métrica adaptativa capaz de induzir clusters elipsoidais / circulares com diferentes distribuições espaciais
- Possui “a mesma” função objetivo e segue os mesmos passos do FCM, exceto que a distância entre o j-ésimo objeto e o i-ésimo protótipo é calculada em função de uma matriz adaptativa independente A_i , i.e.:

$$\|\mathbf{x}_j - \mathbf{v}_i\|_{A_i}^2 = (\mathbf{x}_j - \mathbf{v}_i)^T A_i (\mathbf{x}_j - \mathbf{v}_i)$$

- FCM com dist. Euclidiana é obtido com $A_i = I_{n \times n}$ para todo $i = 1, \dots, c$

Gustafson-Kessel (GK)

- ◆ Matrizes A_i são obtidas adaptativamente, em função das **matrizes de covariância fuzzy** dos clusters:

$$A_i = \sqrt[n]{\det(\Phi_i)} \Phi_i^{-1}$$

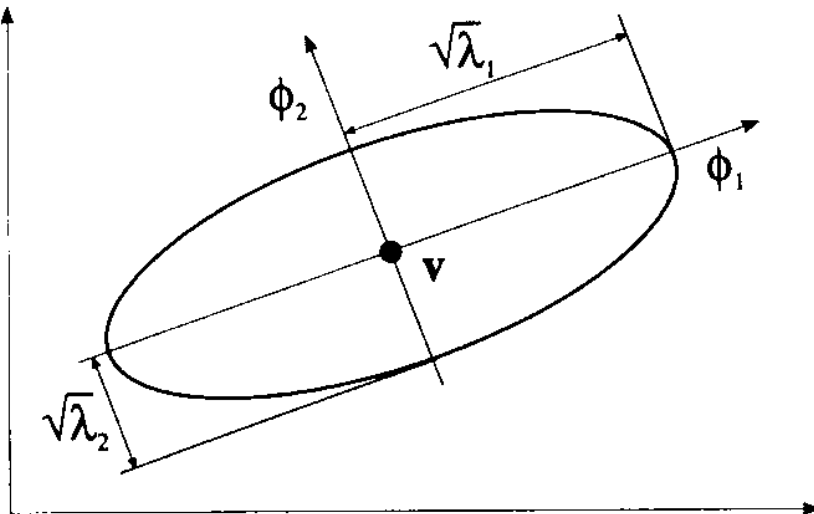
onde:

$$\Phi_i = \frac{\sum_{j=1}^N \mu_{ij}^m (\mathbf{x}_j - \mathbf{v}_i)(\mathbf{x}_j - \mathbf{v}_i)^T}{\sum_{k=1}^N \mu_{ik}^m}$$

Gustafson-Kessel (GK)

- ◆ **Nota:** A equação abaixo define um hiper-elipsóide

$$(\mathbf{x} - \mathbf{v})^T \Phi_i^{-1} (\mathbf{x} - \mathbf{v}) = 1$$

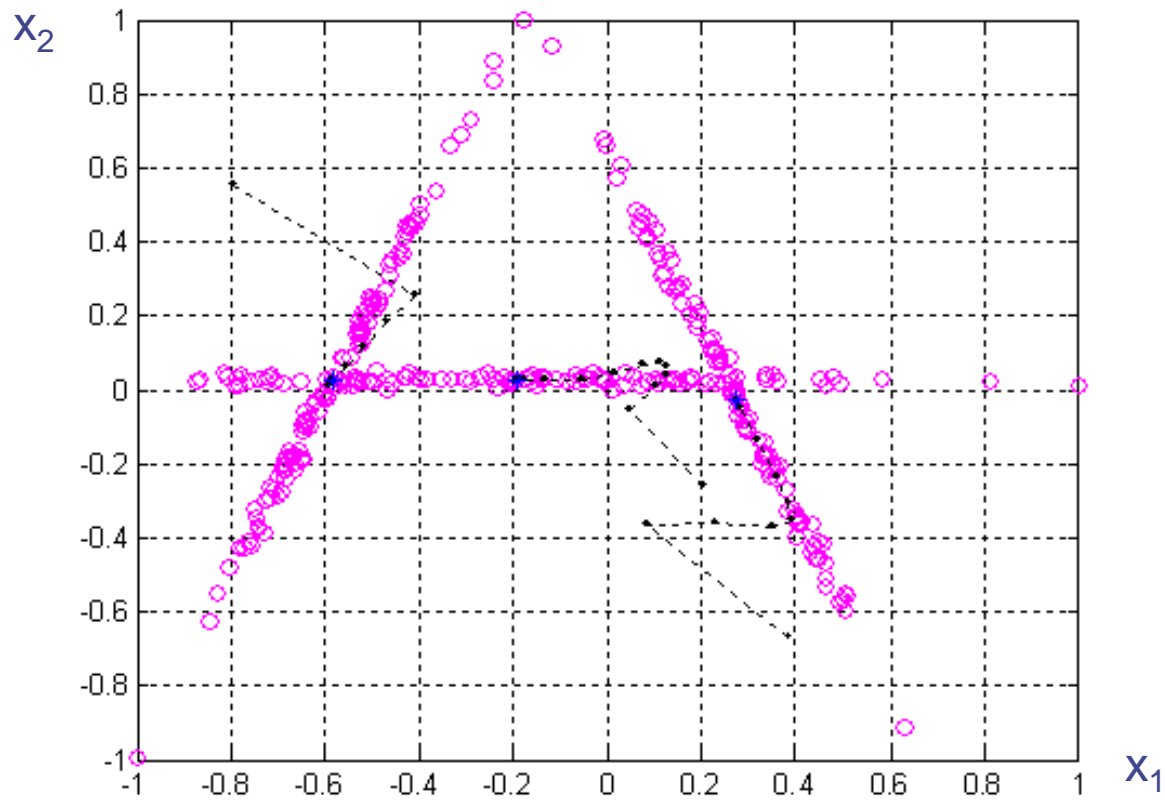


A matriz de covariância fuzzy concentra informação sobre a distribuição espacial do cluster: Autovetores ϕ_k e autovalores λ_k parametrizam a distância hiper-elipsoidal

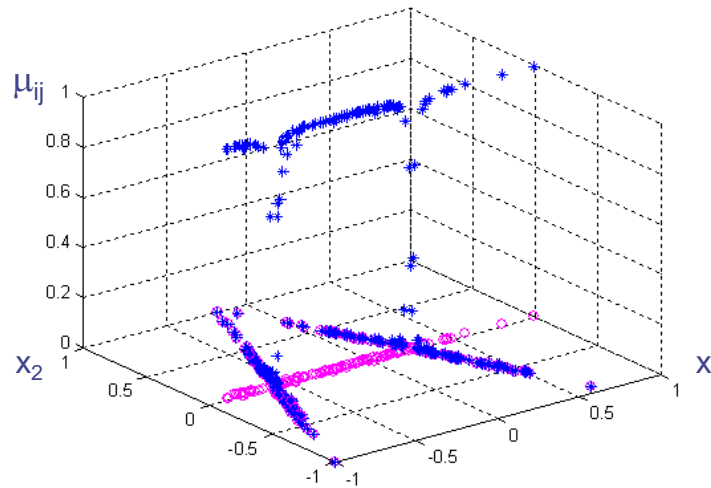
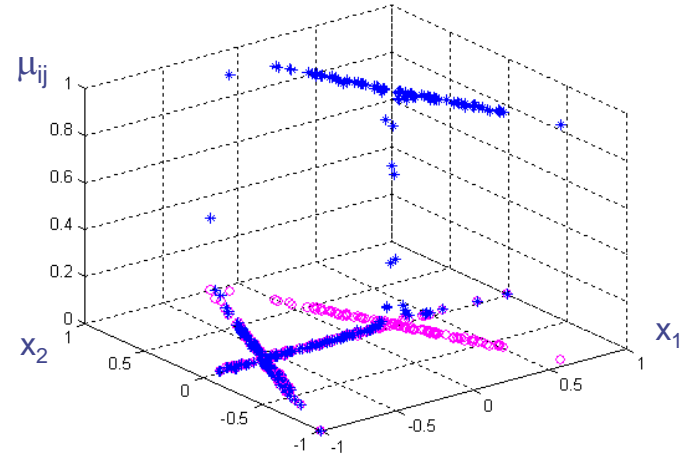
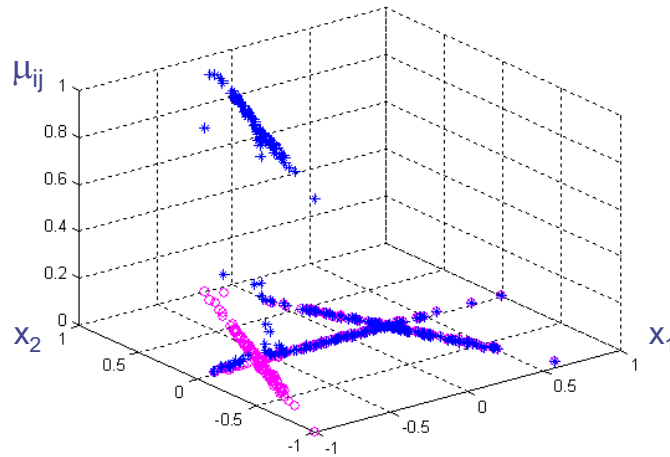
- maior dificuldade está no cálculo da inversa, Φ^{-1} : $\mathcal{O}(n^3)$

Gustafson-Kessel (GK)

◆ Exemplo:



Gustafson-Kessel (GK)



Expectation Maximization (EM)

- O Algoritmo **EM (Expectation Maximization)** é um procedimento genérico para a modelagem **probabilística** de um conjunto de dados
- Basicamente, **EM** otimiza os parâmetros de uma função de distribuição de probabilidades (p.d.f.) de forma que esta represente os dados da forma mais verossímil possível
- Modelo mais utilizado: **Mistura de Gaussianas**

EM – Mistura de Gaussianas

- Representada pela seguinte *p.d.f*:

$$p(\mathbf{x}) = \sum_{i=1}^k \pi_i \mathcal{N}(\mathbf{x} | \mathbf{v}_i, \Sigma_i)$$

- \mathbf{x} é um objeto
- \mathcal{N} é uma Gaussiana (da mesma dimensão dos objetos)
 - \mathbf{v}_i é o centro da i -ésima Gaussiana (vetor da mesma dimensão de \mathbf{x})
 - Σ_i é a matriz de covariância da i -ésima Gaussiana
- π_i são os coeficientes da mistura
- k é o número de Gaussianas/componentes da mistura

EM – Mistura de Gaussianas

- Para compreender $p(\mathbf{x})$, seja uma var. aleatória binária k-dimensional \mathbf{z} , tal que:
 - $\mathbf{z} = [z_1 \dots z_k]^T$ assume apenas valores em representação 1-de-k:
 - $z_i = 1$ para um dado $i \in \{1, \dots, k\}$; todos os demais são nulos, ou seja: $z_i \in \{0, 1\}$ e $\sum_i z_i = 1$
 - Define-se $\pi_i = p(z_i = 1)$ como a **probabilidade a priori** de que $z_i = 1$, $0 \leq \pi_i \leq 1$, $\sum_i \pi_i = 1$.
 - A distribuição de probabilidades $p(\mathbf{z})$ é tal que:

$$p(\mathbf{z}) = \prod_{i=1}^k \pi_i^{z_i}$$

EM – Mistura de Gaussianas

- Note que a i -ésima Gaussiana corresponde à distribuição condicional de \mathbf{x} dado um valor particular de \mathbf{z} , i.e.:

- $p(\mathbf{x} | z_i = 1) = \mathcal{N}(\mathbf{x} | \mathbf{v}_i, \Sigma_i)$, ou (equivalentemente)

- $p(\mathbf{x} | \mathbf{z}) = \mathcal{N}(\mathbf{x} | \mathbf{v}_i, \Sigma_i)$ para a realização de \mathbf{z} tal que $z_i = 1$

- Das distribuições $p(\mathbf{z})$ e $p(\mathbf{x} | \mathbf{z})$ temos:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z}) p(\mathbf{x} | \mathbf{z})$$

- A distribuição $p(\mathbf{x})$ é obtida então como:

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^k \pi_i \mathcal{N}(\mathbf{x} | \mathbf{v}_i, \Sigma_i)$$

EM – Mistura de Gaussianas

- Responsabilidade do componente i para explicar \mathbf{x}_j :

$$\mu_{ij} = p(z_i = 1 | \mathbf{x}_j) = \frac{\pi_i \mathcal{N}(\mathbf{x}_j | \mathbf{v}_i, \Sigma_i)}{\sum_{l=1}^k \pi_l \mathcal{N}(\mathbf{x}_j | \mathbf{v}_l, \Sigma_l)}$$

$$(p(\mathbf{z}_j | \mathbf{x}_j) = p(\mathbf{x}_j, \mathbf{z}_j) / p(\mathbf{x}_j) \rightarrow p(\mathbf{z}_j | \mathbf{x}_j) = p(\mathbf{z}_j) p(\mathbf{x}_j | \mathbf{z}_j) / p(\mathbf{x}_j) \rightarrow p(z_{ij} = 1 | \mathbf{x}_j) = p(z_{ij} = 1) p(\mathbf{x}_j | z_{ij} = 1) / p(\mathbf{x}_j))$$

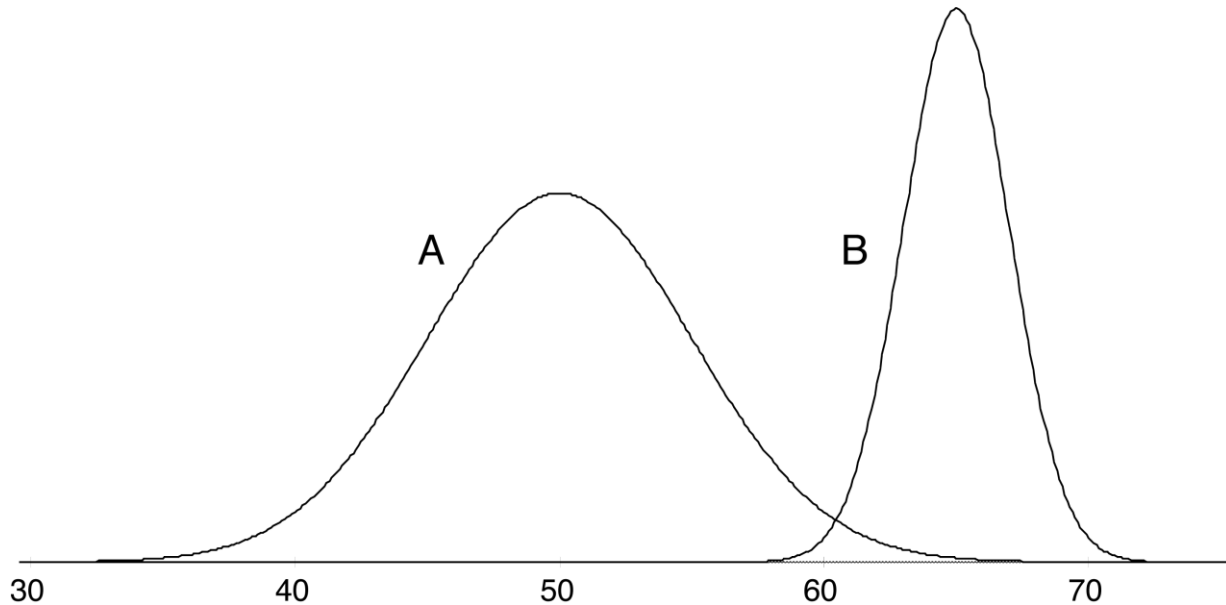
- É a probabilidade **a posteriori** de $z_i = 1$ dado que se observou \mathbf{x}_j
 - Compare com π_i (probabilidade a priori)
- Probabilidade a posteriori de \mathbf{x}_j ter sido produzido pela i -ésima Gaussiana

$p(x)=?$ $\mu_{ij}=?$

Objetos:

A	51	B	62	B	64	A	48	A	39	A	51
A	43	A	47	A	51	B	64	B	62	A	48
B	62	A	52	A	52	A	51	B	64	B	64
B	64	B	64	B	62	B	63	A	52	A	42
A	45	A	51	A	49	A	43	B	63	A	48
A	42	B	65	A	48	B	65	B	64	B	64
A	46	A	48	B	62	B	66	A	48	A	41
A	45	A	49	A	43	B	65	B	64		
A	45	A	46	A	40	A	46	A	48		

Modelo:



$$\mu_A=50, \sigma_A=5, p_A=0.6$$

$$\mu_B=65, \sigma_B=2, p_B=0.4$$

EM – Mistura de Gaussianas

- Dado $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ de N observações *i.i.d.*, temos:

$$p(\mathbf{X}) = p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \prod_{j=1}^N p(\mathbf{x}_j) = \prod_{j=1}^N \sum_{l=1}^k \pi_l \mathcal{N}(\mathbf{x}_j | \mathbf{v}_l, \Sigma_l)$$

- $\Sigma = \{\Sigma_1, \dots, \Sigma_k\}$, $\mathbf{v} = \{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ e $\pi = \{\pi_1, \dots, \pi_k\}$
- Refere-se a esta distribuição por $p(\mathbf{X} | \pi, \Sigma, \mathbf{v})$
- Por conveniência matemática, utiliza-se da **log-verossimilhança**:

$$\ln(p(\mathbf{X} | \pi, \Sigma, \mathbf{v})) = \sum_{j=1}^N \ln \left(\sum_{l=1}^k \pi_l \mathcal{N}(\mathbf{x}_j | \mathbf{v}_l, \Sigma_l) \right)$$

EM – Mistura de Gaussianas

- Maximizar a verossimilhança pode ser visto como maximizar a compatibilidade entre as N observações e o modelo
- EM (Dempster et al., 1977) é um algoritmo de otimização que visa maximizar a (log) verossimilhança em dois passos:
 - **Passo E** (Expectation)
 - Avalia as probabilidades a posteriori μ_{ij} ($i = 1, \dots, k; j = 1, \dots, N$)
 - a partir das N observações $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ e do modelo corrente, dado pelos parâmetros $\Sigma = \{\Sigma_1, \dots, \Sigma_k\}$, $\mathbf{v} = \{\mathbf{v}_1, \dots, \mathbf{v}_k\}$ e $\pi = \{\pi_1, \dots, \pi_k\}$
 - **Passo M** (Maximization)
 - Ajusta os parâmetros do modelo visando maximizar a log-verossimilhança

EM – Mistura de Gaussianas

- **Passo E** (Expectation):

- Avalia as probabilidades a posteriori μ_{ij} ($i = 1, \dots, k; j = 1, \dots, N$)

$$\mu_{ij} = \frac{\pi_i \mathcal{N}(\mathbf{x}_j | \mathbf{v}_i, \Sigma_i)}{\sum_{l=1}^k \pi_l \mathcal{N}(\mathbf{x}_j | \mathbf{v}_l, \Sigma_l)}$$

$$\mathcal{N}(\mathbf{x}_j | \mathbf{v}_i, \Sigma_i) = \frac{1}{(2\pi)^{n/2} \det(\Sigma_i)^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}_j - \mathbf{v}_i)^T \Sigma_i^{-1} (\mathbf{x}_j - \mathbf{v}_i) \right\}$$

EM – Mistura de Gaussianas

- **Passo M** (Maximization):

- Ajusta o modelo visando maximizar a verossimilhança

$$\left\{ \begin{array}{l} \mathbf{v}_i = \frac{1}{N_i} \sum_{j=1}^N \mu_{ij} \mathbf{x}_j \quad \rightarrow \text{centróide ponderado} \\ \boldsymbol{\Sigma}_i = \frac{1}{N_i} \sum_{j=1}^N \mu_{ij} (\mathbf{x}_j - \mathbf{v}_i)(\mathbf{x}_j - \mathbf{v}_i)^T \quad \rightarrow \text{covariância ponderada} \\ \pi_i = \frac{N_i}{N} \quad \rightarrow \text{Coeficientes da mistura / prob. a priori do i-ésimo componente} \\ N_i = \sum_{j=1}^N \mu_{ij} \quad \rightarrow \text{Número efetivo de pontos atribuídos ao i-ésimo grupo} \end{array} \right.$$

EM – Mistura de Gaussianas

■ Algoritmo:

1. Inicialização (via k-means)

- protótipos \mathbf{v}_i = centróides finais do k-means
- covariâncias Σ_i = matrizes de covariância finais (dos grupos)
- probabilidades μ_{ij} (para N_i e π_i) = matriz de partição rígida final

2. Passo E

3. Passo M

4. Avaliação do Critério de Parada

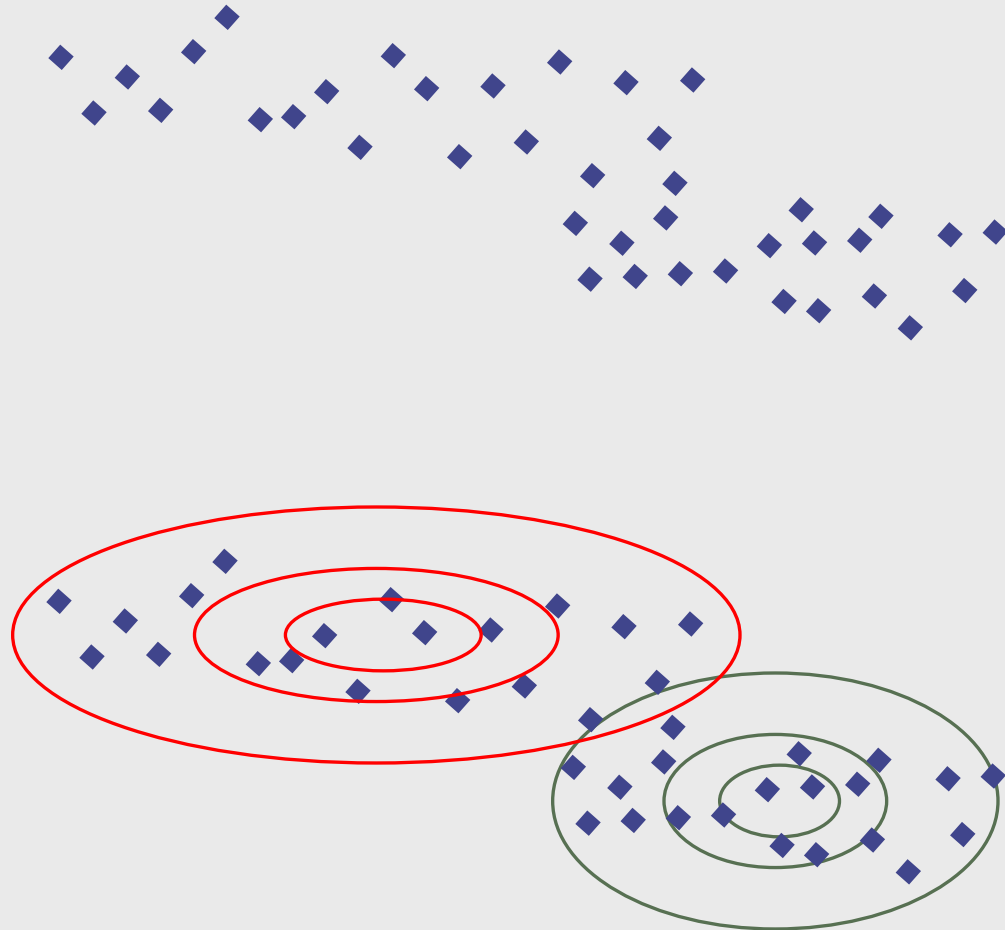
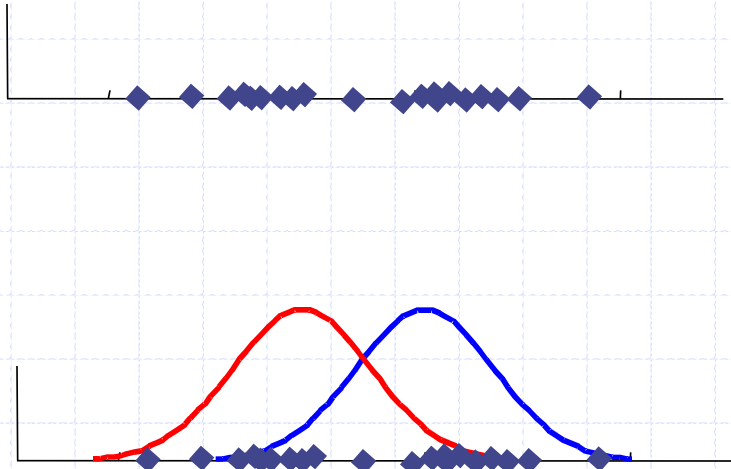
- e.g. função de log-verossimilhança

5. Interrupção ou Retorno ao Passo 2

EM × k-Means

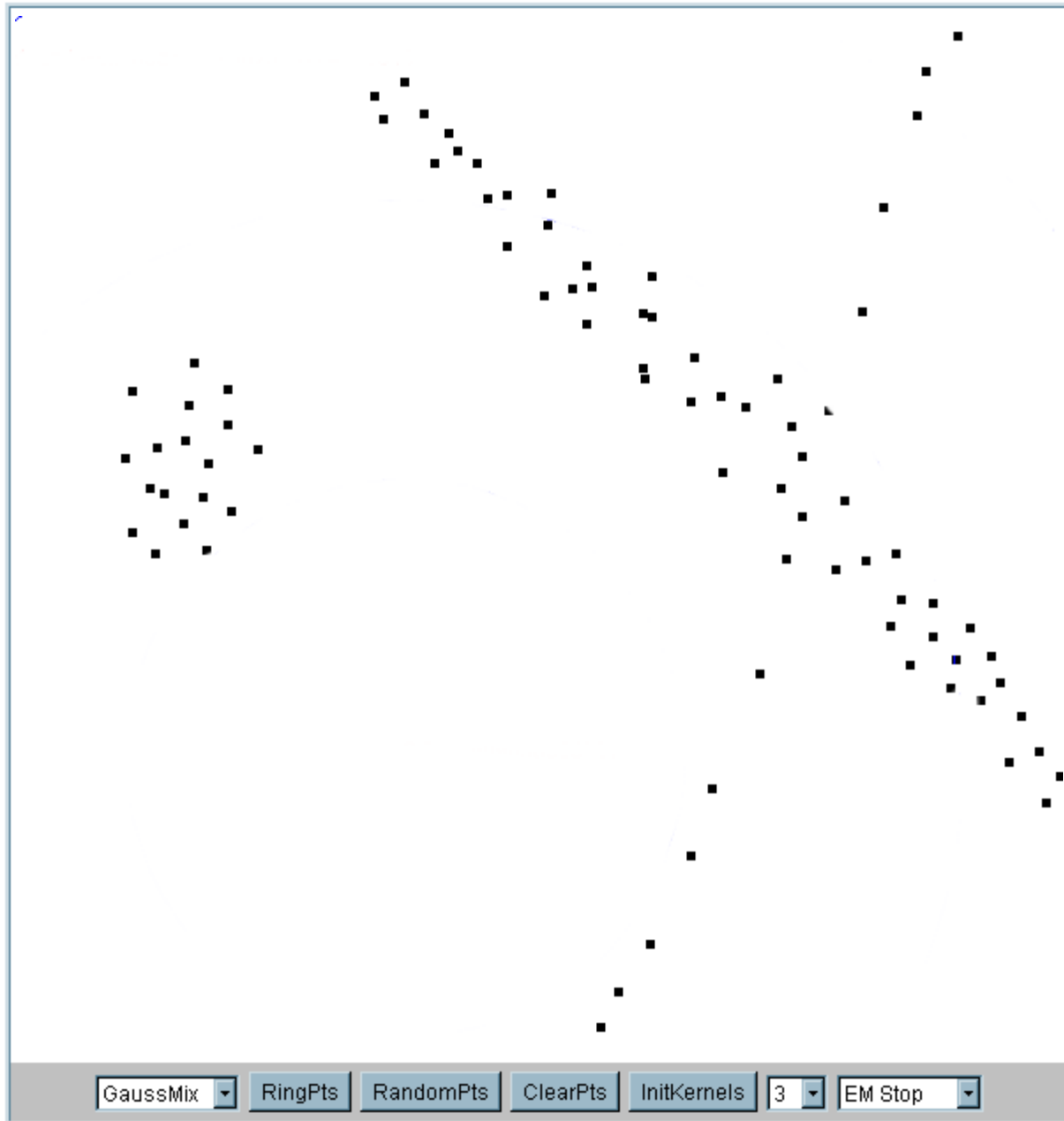
- EM produz informação muito mais rica sobre os dados
 - Probabilidades associadas a cada padrão / cluster
- Probabilidades produzidas por EM podem facilmente ser convertidas em uma partição rígida, caso desejado
 - Via projeção do maior valor
 - Essa partição é capaz de representar clusters alongados, elipsoidais, com atributos correlacionados
- No entanto, todas as vantagens acima vêm com um elevado custo computacional associado...
 - Cálculo das Normais Multi-Dimensionais \mathcal{N} demanda as inversas das matrizes de covariância Σ_i , que é $O(n^3)$
- k-means é um caso particular de EM. Ambos estão sujeitos a mínimos locais

EM (Mistura de Gaussianas – Exemplos)

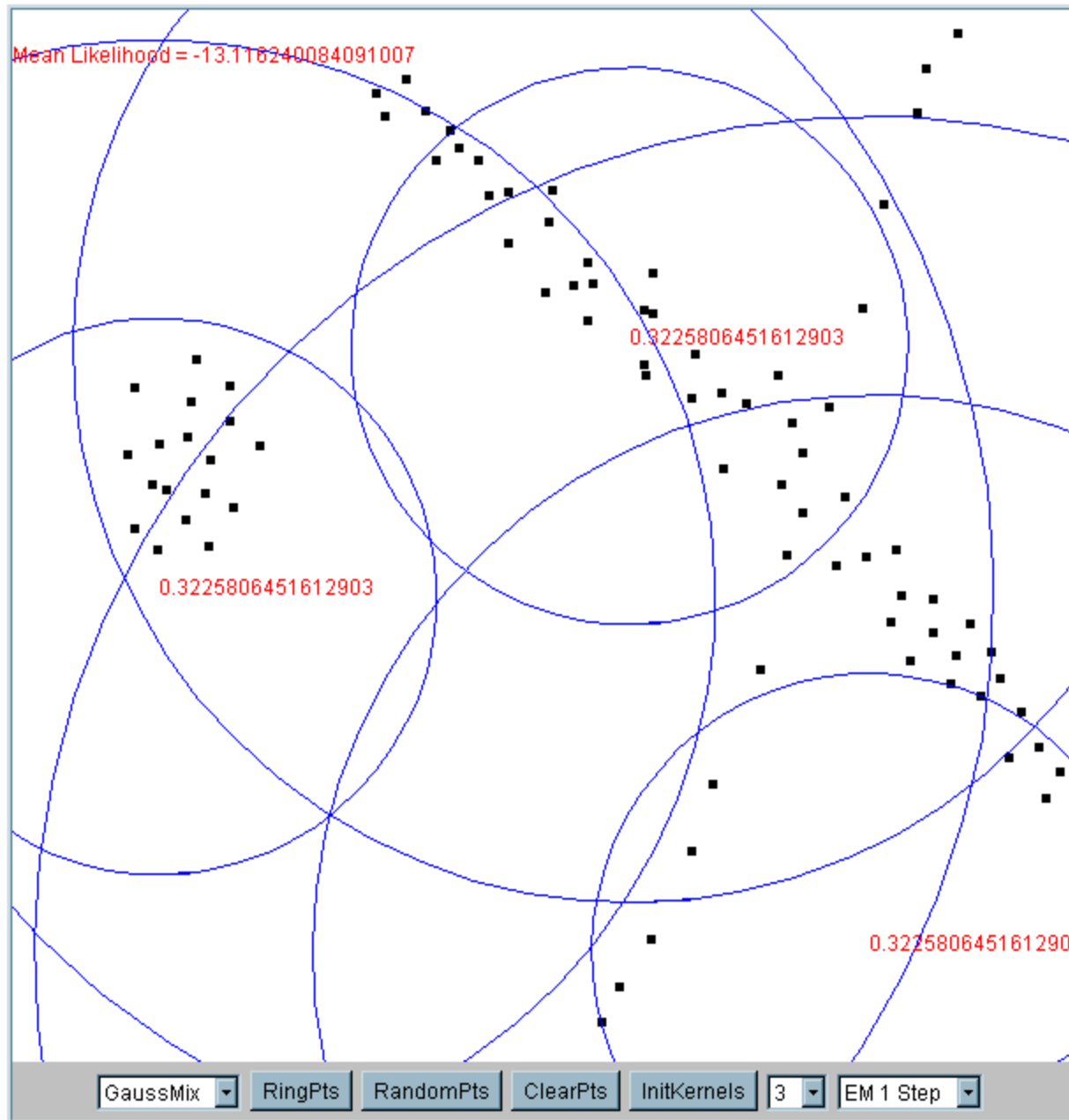


Exemplo

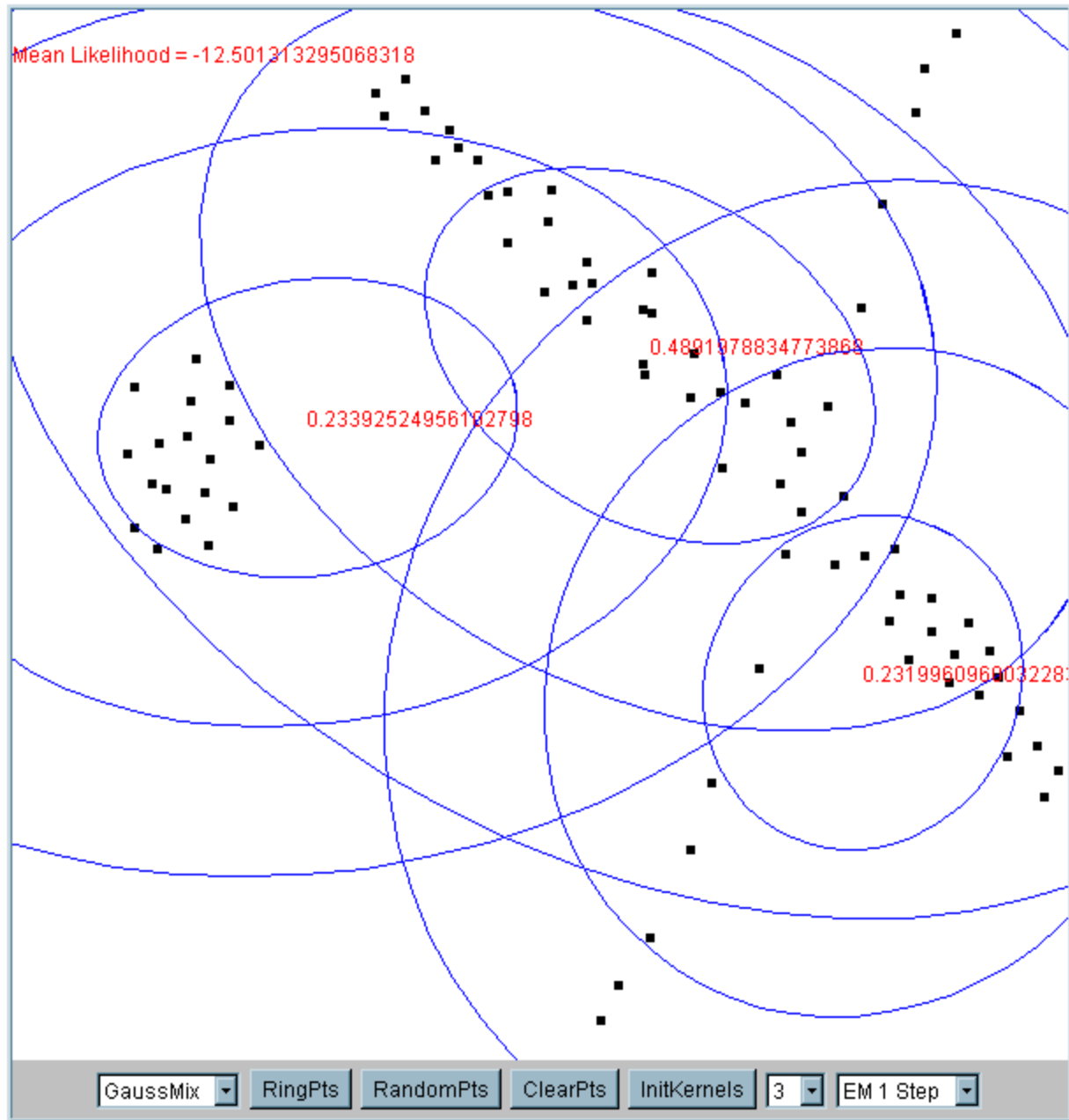
(passo-a-passo)



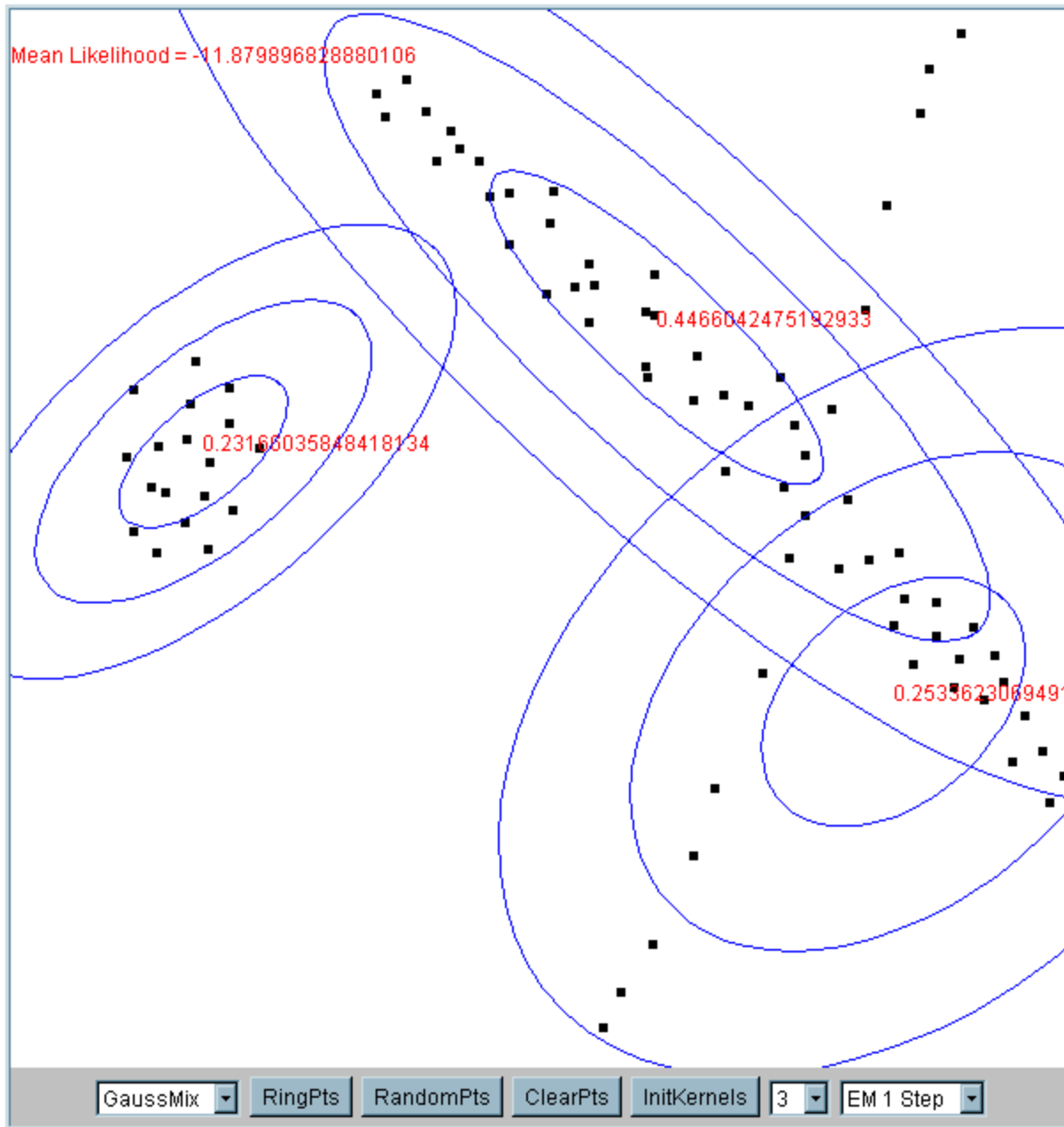
Iteração 1



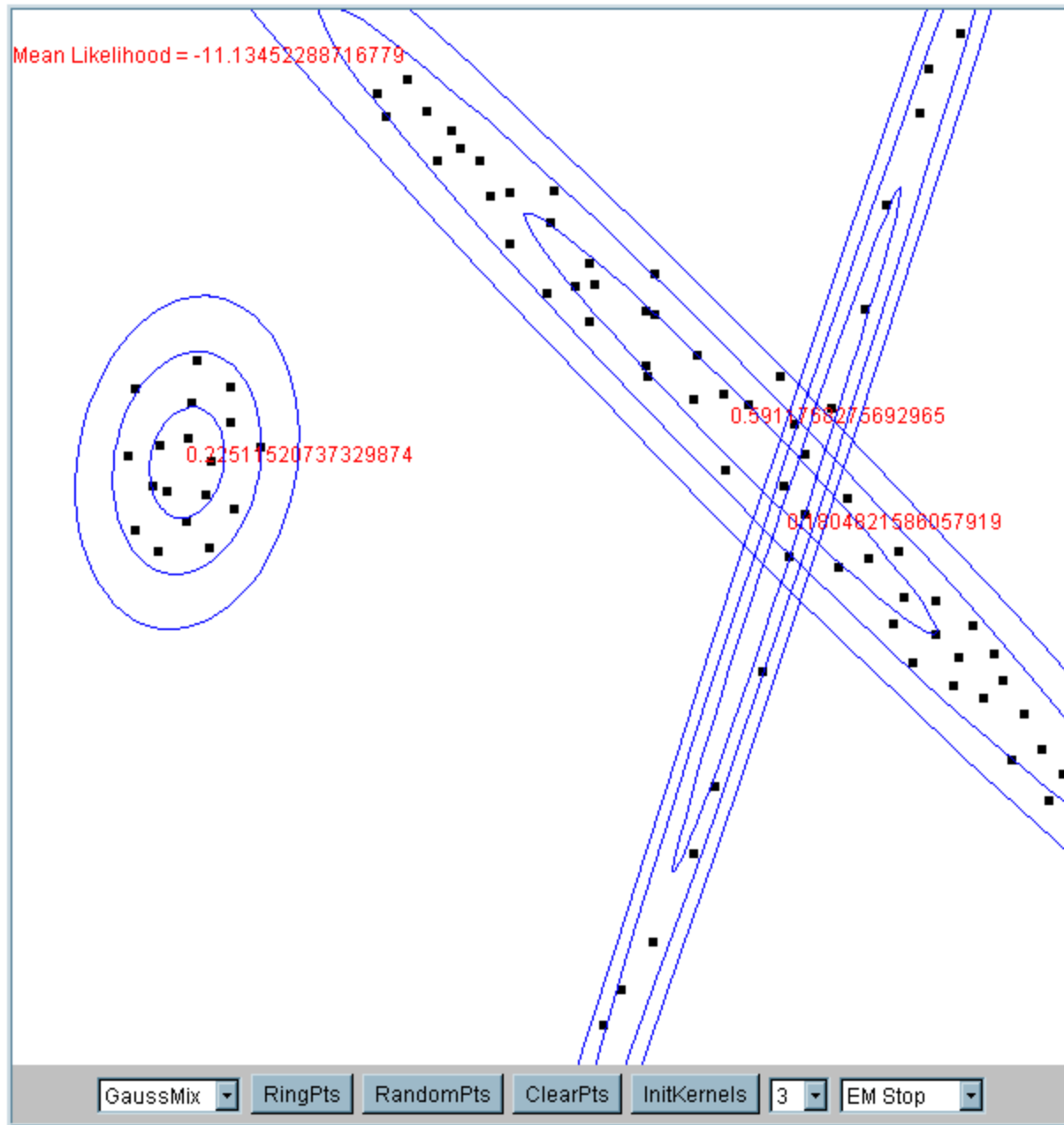
Iteração 2



Iteração 5



Iteração 25



Exercício

Objeto	x
1	-1.31
2	-0.43
3	0.34
4	3.57
5	2.76
6	0.30
7	9.06
8	4.45
9	2.87
10	4.42

- Execute manualmente iterações do EM na base de dados ao lado ($n = 1, N = 10$), com $k = 2$. Tome protótipos iniciais arbitrários e os demais parâmetros inicializados a partir destes, de maneira análoga à inicialização via k-means
- Ilustre o resultado obtido de forma gráfica

Algoritmos Baseados em Densidade

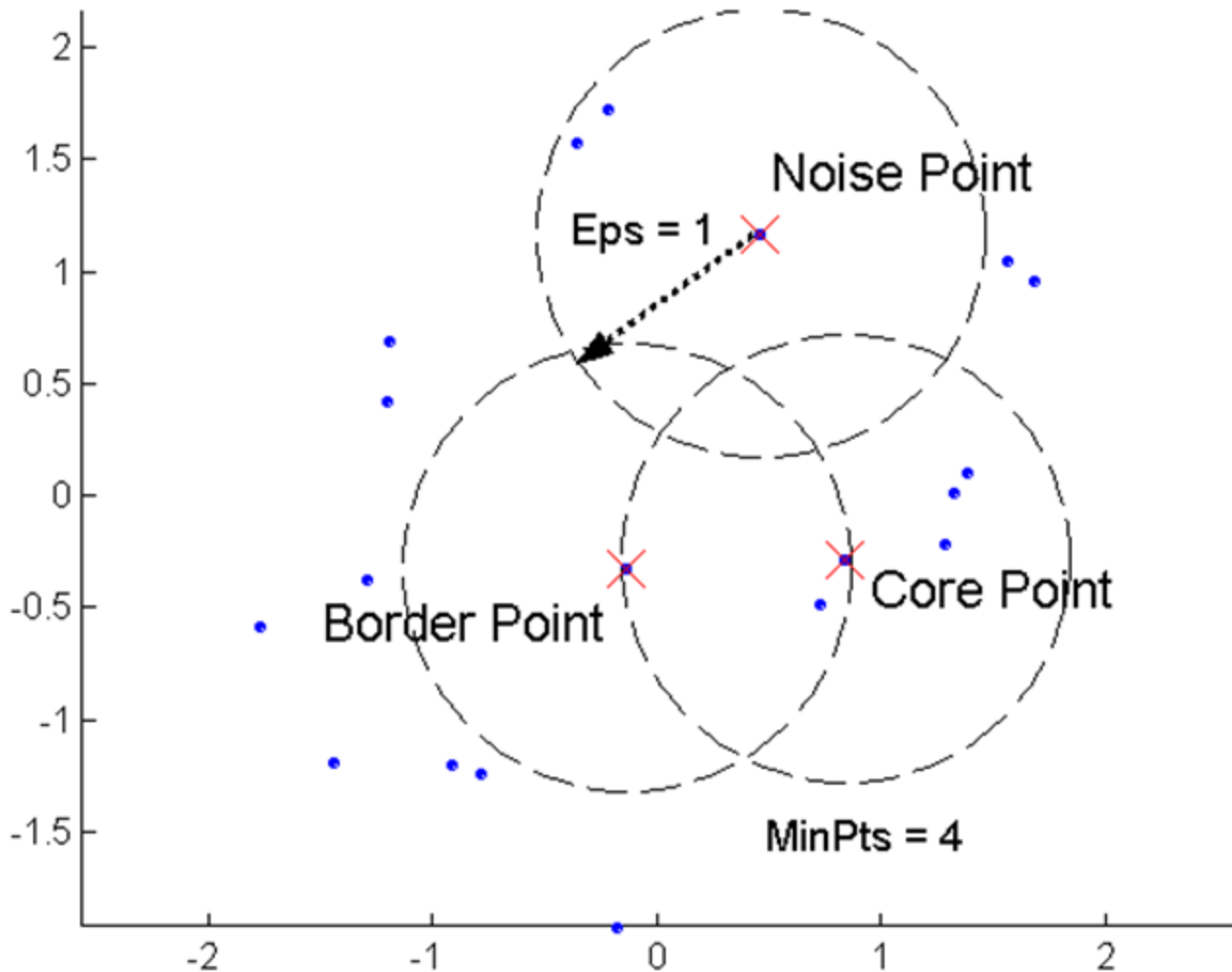
- **Paradigma de Agrupamento por Densidade**
 - Clusters como regiões de alta concentração de objetos
 - separadas por regiões de baixa concentração de objetos
 - Paradigma alternativo àquele baseado em protótipos
 - K-means e variantes, FCM, GK, EM, etc
- Existem vários algoritmos
- Veremos a seguir um dos mais conhecidos: **DBSCAN**

DBSCAN

- DBSCAN is a density-based algorithm
 - It uses the concept of **Center-Based Density**
 - ◆ number of points within a specified radius (**Eps**)

- Definitions:
 - A point is a **core point** if it has at least a specified number of points (**MinPts**) within the radius **Eps** (including the point itself)
 - ◆ these are points that are in the interior of a cluster
 - A **border point** has fewer than MinPts within Eps, but is in the neighborhood (within the radius) of at least 1 core point
 - A **noise point** is neither a core point nor a border point

DBSCAN: Core, Border, and Noise Points



DBSCAN

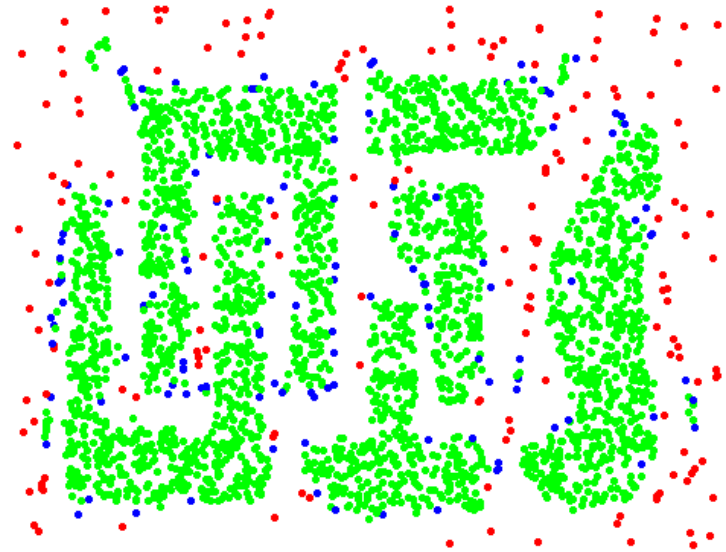
▪ Algoritmo Conceitual:

1. Percorra a BD e rotule os objetos como **core**, **border** ou **noise**
2. Elimine aqueles objetos rotulados como **noise**
3. Insira uma aresta entre cada par de objetos **core** vizinhos
 - 2 objetos são vizinhos se um estiver dentro do raio ϵ do outro
4. Faça cada componente conexo resultante ser um cluster
5. Atribua cada **border** ao cluster de um de seus **core** associados
 - Resolva empates se houver objetos **core** associados de diferentes clusters

DBSCAN: Core, Border and Noise Points



Original Points



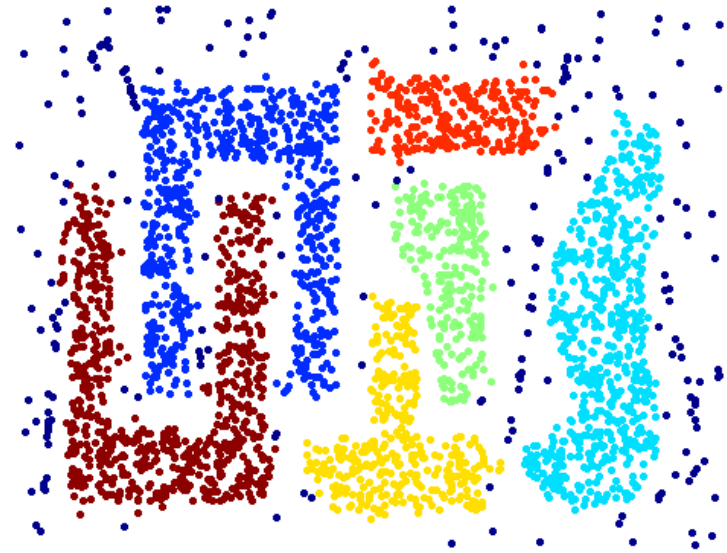
Point types: **core**,
border and **noise**

Eps = 10, MinPts = 4

When DBSCAN Works Well



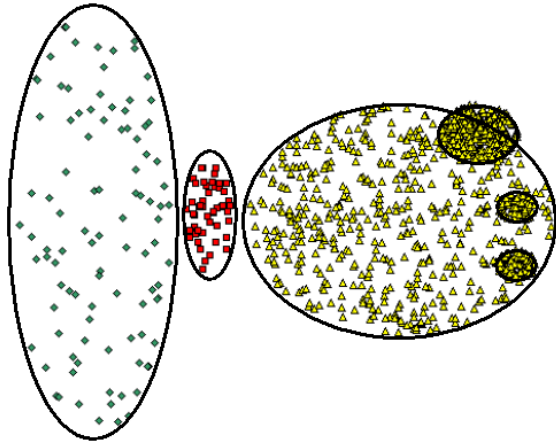
Original Points



Clusters

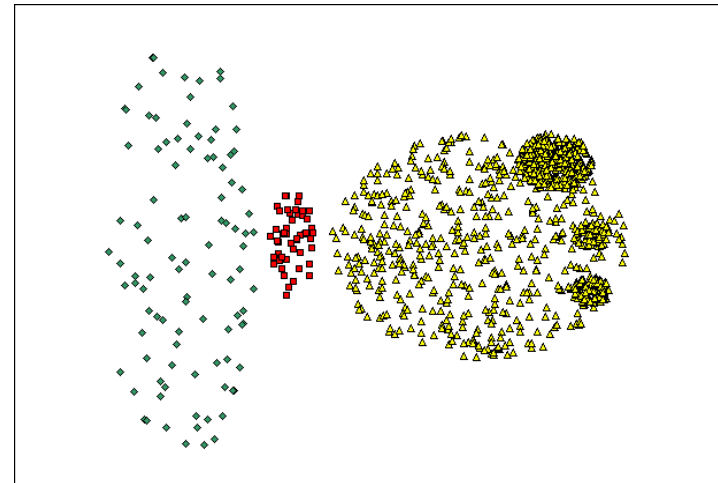
- **Resistant to Noise**
- **Can handle clusters of different shapes and sizes**

When DBSCAN Does NOT Work Well

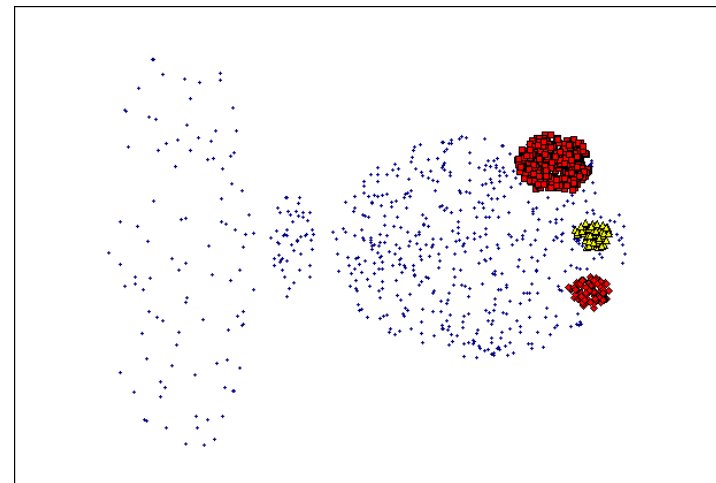


Original Points

- Varying densities
- High-dimensional data



(MinPts=4, Eps=9.75).



(MinPts=4, Eps=9.92)



Referências

- Höppner, F., Klawonn, F., Kruse, R., Runkler, T., *Fuzzy Cluster Analysis*, 1999
- Bezdek, J. C., *Pattern Recognition with Fuzzy Objective Function Algorithm*, Plenum Press, 1981
- Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, 2006
- I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd Edition, Morgan Kaufmann, 2005
- Tan, P.-N., Steinbach, M., and Kumar, V., *Introduction to Data Mining*, Addison-Wesley, 2006