Clusterização

1. Visão geral

O desafio envolvido na tarefa conhecida como **clusterização** pode ser enunciado da seguinte forma:

- Dado um conjunto de observações $\mathbf{x}(i) \in \mathbb{R}^{K\times 1}$, $i = 1, \ldots, N$, desejamos separá-las, utilizando alguma medida de similaridade, em grupos distintos denominados *clusters*, os quais correspondem a subconjuntos disjuntos da coleção completa de dados. Ao final, espera-se que cada grupo (*cluster*) identificado exiba alguma homogeneidade ou regularidade interna.

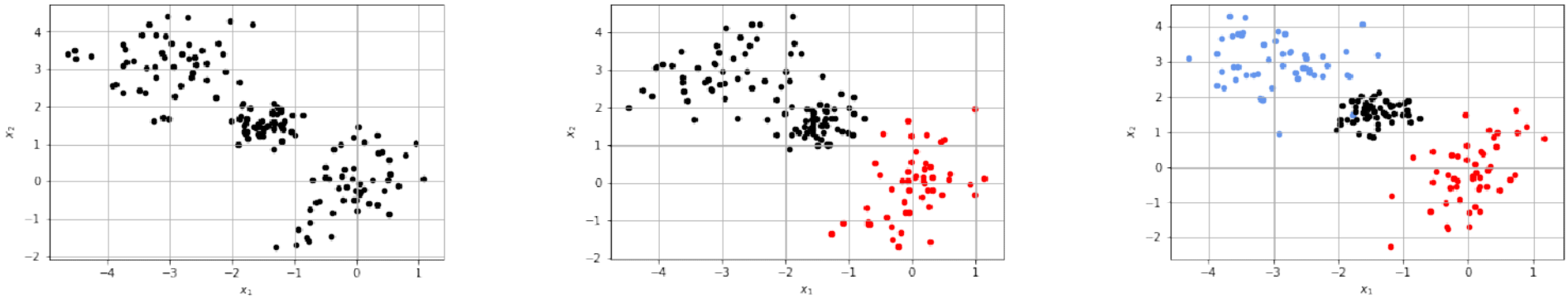
O papel da **clusterização** consiste, portanto, em revelar uma estrutura natural existente no conjunto de dados ao agrupar objetos de acordo com suas

similaridades mútuas. Em outras palavras, o objetivo é dividir um conjunto de observações em k subconjuntos disjuntos, cada um composto de elementos com certo grau de similaridade.

Diferentemente dos problemas de regressão e classificação, a tarefa de **clusterização** está associada à ideia de **aprendizado não-supervisionado:**

- Agora, não há mais uma saída conhecida a ser aproximada para cada padrão de entrada.
- Na realidade, neste paradigma diferente de aprendizado, o que se busca é inferir uma função que descreva de maneira adequada a estrutura dos dados não-rotulados.

Questão: Para um mesmo conjunto de amostras, dadas múltiplas propostas de agrupamento, existem formas automáticas de ordená-las de acordo com a qualidade do agrupamento?



(a) Dados disponíveis. (b) Dois *clusters*. (c) Três *clusters*.

Figura. Diferentes possibilidades de agrupamento para o mesmo conjunto de observações.

- É preciso ressaltar que, no problema de clusterização, a análise dos *clusters* obtidos inevitavelmente carrega certo grau de subjetividade. No fundo, o mesmo conjunto de dados pode ser dividido de diversas maneiras, dependendo do ponto de vista e/ou das hipóteses consideradas com respeito à natureza dos dados.
- Não obstante, é possível utilizar alguns índices matemáticos para expressar a qualidade da clusterização, ainda que eles não sejam definitivos e que não haja um consenso na literatura quanto ao mais apropriado.

2. Métricas de avaliação

2.1. *Akaike Information Criterion* (AIC)

O critério AIC (AKAIKE, 1974) foi concebido com base numa estimativa da divergência de Kullbäck-Leibler entre a distribuição do modelo estimado e a distribuição verdadeira associada aos dados. Em essência, trata-se de um estimador da qualidade relativa de modelos estatísticos para um determinado conjunto de dados.

$$\mathrm{AIC} = 2P - 2\ln \mathcal{L}, \tag{1}$$

onde P denota o número de parâmetros ajustáveis do modelo e $\mathcal{L}$ corresponde ao valor (máximo) da função de verossimilhança para este modelo. O primeiro termo em (1) representa uma penalização que, implicitamente, busca desencorajar a opção por modelos mais susceptíveis a *overfitting*.

2.2. *Bayesian Information Criterion* (BIC)

O critério BIC (SCHWARZ ET AL., 1978) foi criado com base em uma estimativa da probabilidade das observações **x** condicionadas ao modelo em estudo ($\mathcal{M}$), $p(\mathbf{x}|\mathcal{M})$. Assim como o AIC, ele leva em conta o valor da função de verossimilhança e o número de parâmetros do modelo. Porém, acrescenta também uma penalidade proporcional ao número de amostras:

$$\mathrm{BIC} = 2P \ln N - 2 \ln \mathcal{L}. \qquad (2)$$

Nota: Os critérios AIC e BIC podem ser usados para a seleção de modelos em tarefas de regressão e de classificação.

2.3. **Coeficiente Silhouette**

O coeficiente silhouette (ROUSSEEUW, 1987) dá origem a um método de interpretação e validação da consistência dentro dos *clusters* de dados. A técnica oferece uma representação gráfica sucinta do quão bem cada objeto se situa dentro do seu *cluster*.

O coeficiente silhouette é uma medida do quão similar um objeto é com relação ao próprio *cluster* (coesão) quando comparado com outros clusters (separação). Seu valor está restrito ao intervalo $[-1,1]$, sendo que um valor elevado indica que o objeto apresenta um bom casamento com o seu *cluster* e, ao mesmo tempo, um casamento pobre com *clusters* vizinhos.

Se a maior parte dos objetos apresenta um valor elevado, então a clusterização obtida é considerada como apropriada. Se muitos pontos possuem um valor baixo ou negativo para o coeficiente silhouette, então a clusterização pode ter sido feita com um número excessivo ou insuficiente de *clusters*.

Considere que os dados disponíveis tenham sido clusterizados em k clusters.

Então, para cada padrão $\mathbf{x}(i), i = 1, \dots, N$, $a(i)$ denota a distância média entre $\mathbf{x}(i)$ e todos os outros padrões pertencentes ao mesmo *cluster*. É possível enxergar $a(i)$ como uma medida da qualidade da atribuição do padrão $\mathbf{x}(i)$ ao seu *cluster*: quanto

menor o valor, melhor a atribuição, pois há forte similaridade (média) com todos os padrões pertencentes àquele *cluster*.

Seja $b(i)$ a menor distância média entre $\mathbf{x}(i)$ e todos os *clusters* dos quais i não é membro. O *cluster* com menor dissimilaridade média em relação a $\mathbf{x}(i)$ é dito ser o *cluster* vizinho, porque ele é o próximo melhor agrupamento para o padrão $\mathbf{x}(i)$.

Sendo assim, define-se o coeficiente silhouette $s(i)$ para o padrão $\mathbf{x}(i)$ da seguinte forma:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}, \tag{3}$$

ou, escrito de outra forma:

$$s(i) = \begin{cases} 1 - \dfrac{a(i)}{b(i)}, & \text{se } a(i) < b(i) \\ 0, & \text{se } a(i) = b(i) \\ \dfrac{b(i)}{a(i)} - 1, & \text{se } a(i) > b(i) \end{cases} \tag{4}$$

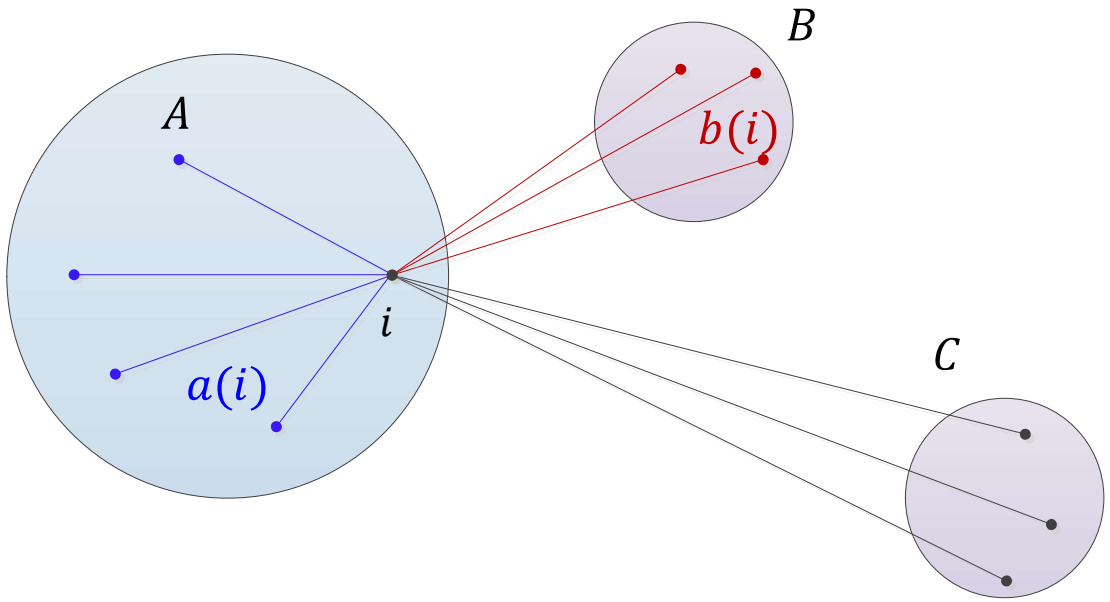


Figura. Ilustração do cálculo do coeficiente silhouette para o padrão i.

Para $s(i)$ assumir valores próximos de $+1$, é necessário que $a(i) \ll b(i)$. Uma vez que $a(i)$ indica o quão dissimilar é o padrão i com respeito a seu próprio *cluster*, um valor pequeno significa que ele se encaixa bem nas características do *cluster*. Além disso, um valor elevado de $b(i)$ implica que i não se encaixa bem no *cluster* vizinho. Portanto:

- $s(i)$ próximo de $+1$ significa que o dado foi clusterizado de forma adequada.
- $s(i)$ próximo de -1 indica que seria mais apropriado o padrão i pertencer ao *cluster* vizinho.
- $s(i)$ próximo de zero significa que o dado está próximo da fronteira entre dois *clusters* naturais.

Análise:

- Podemos tomar o **valor médio do coeficiente silhouette** para todos os padrões $\mathbf{x}(i), i = 1, \ldots, N$ como uma medida do quão apropriadamente os dados foram clusterizados.
- Gráficos do coeficiente silhouette para cada padrão podem ser utilizados para determinar o número natural de *clusters* dentro de um conjunto de dados.

2.4. **Índice de Davies-Bouldin**

Suponha que um conjunto de dados $\{\mathbf{x}_i\}_{i=1}^{N}$ já foi clusterizado em k *clusters* $C_j, j = 1, \ldots, k$. Cada *cluster* C_j possui T_j pontos e um protótipo $\boldsymbol{\mu}_j$, de modo que $\sum_j T_j = N$.

Espalhamento dentro de um *cluster*:

$$S_j = \left(\frac{1}{T_j} \sum_{i \in C_j} \left| \mathbf{x}_i - \boldsymbol{\mu}_j \right|^p \right)^{\frac{1}{p}}, \tag{5}$$

para $j = 1, \ldots, k$.

É importante mencionar que a métrica de distância explorada em S_j deve casar com aquela utilizada pelo esquema de clusterização para termos resultados significativos.

Separação entre clusters:

Considere os clusters C_i e C_q. Então,

$$M_{iq} = \left\| \boldsymbol{\mu}_i - \boldsymbol{\mu}_q \right\|_p = \left(\sum_{k=1}^{K} \left| \mu_{ki} - \mu_{kq} \right|^p \right)^{\frac{1}{p}} \quad (6)$$

denota a distância entre os protótipos $\boldsymbol{\mu}_i$ e $\boldsymbol{\mu}_q$.

Seja R_{iq} uma medida da qualidade de um esquema de clusterização. Esta medida, por definição, deve levar em consideração M_{iq}, a separação entre os *clusters* i e q, a qual deve ser a maior possível, e também S_i, o espalhamento intra-*cluster*, o qual deve ser o menor possível.

Sendo assim, o índice de Davies-Bouldin (DAVIES & BOULDIN, 1979) (DBI) é definido de tal modo que as seguintes propriedades sejam conservadas:

- $R_{iq} \geq 0$.
- $R_{iq} = R_{qi}$.
- Se $S_z \geq S_q$ e $M_{iz} = M_{iq}$, então $R_{iz} > R_{iq}$.
- Se $S_z = S_q$ e $M_{iz} \leq M_{iq}$, então $R_{iz} > R_{iq}$.

Com esta formulação, quanto menor o valor de R_{iq}, maior a separação dos *clusters* e mais concentrada é a dispersão dentro dos *clusters*. Uma solução que satisfaz estes requisitos corresponde a:

$$R_{iq} = \frac{S_i + S_q}{M_{iq}}. \tag{7}$$

Esta medida é usada para definir D_i como:

$$D_i = \max_{q \neq i} R_{iq}. \tag{8}$$

O valor de D_i corresponde ao cenário de pior caso, e este valor é igual a R_{iq} para o *cluster* mais similar ao *cluster* i.

Finalmente, o índice de Davies-Bouldin é dado por:

$$\text{DBI} = \frac{1}{k}\sum_{i=1}^{k} D_i \tag{9}$$

O índice DBI é simétrico e não-negativo: quanto menor o seu valor, melhor é a clusterização.

3. Algoritmo *k-means*

Um dos métodos mais simples e utilizados para encontrar agrupamentos em coleções de dados é o famoso algoritmo *k-means* (MACQUEEN, 1967; LLOYD, 1982).

Hipótese: as distâncias entre os pontos pertencentes ao mesmo *cluster* devem ser menores do que as distâncias em relação a pontos fora do *cluster*.

No algoritmo *k-means*, cada *cluster* será representado por um protótipo $\boldsymbol{\mu}_i \in \mathbb{R}^{K\times 1}$. O desafio é encontrar uma atribuição dos dados aos *k clusters*, bem como um conjunto "ótimo" de protótipos $\{\boldsymbol{\mu}_i\}_{i=1}^{k}$, tais que a soma das distâncias de cada amostra $\mathbf{x}_n$ ao protótipo mais próximo seja minimizada.

Função objetivo:

Seja

$$r_{n,i} = \begin{cases} 1, & \text{se } \mathbf{x}_n \text{ foi atribuído ao cluster } i \\ 0, & \text{caso contrário} \end{cases}.$$

Para cada padrão $\mathbf{x}_n$, somente um $r_{n,i}$ será não-nulo.

O critério explorado pelo algoritmo *k-means* é definido pela seguinte função:

$$J = \sum_{n=1}^{N} \sum_{i=1}^{k} r_{n,i} \|\mathbf{x}_n - \boldsymbol{\mu}_i\|^2 \qquad (10)$$

Parâmetros a determinar

O procedimento para obter a solução envolve dois passos:

- No primeiro passo, dadas as posições atuais dos protótipos $\{\boldsymbol{\mu}_i\}_{i=1}^{k}$, minimizamos J com respeito a $r_{n,i}$; ou seja, fazemos a atribuição de cada padrão $\mathbf{x}_n$ a um dos *clusters* existentes.
- No segundo passo, dada a atribuição dos dados aos *clusters*, minimizamos J com respeito a $\boldsymbol{\mu}_i$; ou seja, atualizamos as posições dos protótipos.

Primeiro passo:

$$\min_{r_{n,i}} J = \sum_{n=1}^{N} \sum_{i=1}^{k} r_{n,i} \|\mathbf{x}_n - \boldsymbol{\mu}_i\|^2 \qquad (11)$$

Como J é linear com respeito a $r_{n,i}$, a otimização pode ser feita de forma independente para cada n. Assim, dado um padrão $\mathbf{x}_n$, a opção que minimiza o termo $\sum_{i=1}^{k} r_{n,i}\|\mathbf{x}_n - \boldsymbol{\mu}_i\|^2$ corresponde a atribuir o padrão ao *cluster* cujo protótipo apresente a menor distância em relação ao dado. Ou seja:

$$r_{n,i} = \begin{cases} 1, & \text{se } i = \arg\min_j \left\|\mathbf{x}_n - \boldsymbol{\mu}_j\right\|^2 \\ 0, & \text{caso contrário} \end{cases}. \tag{12}$$

Segundo passo:

$$\min_{\boldsymbol{\mu}_i} J = \sum_{n=1}^{N}\sum_{i=1}^{k} r_{n,i}\|\mathbf{x}_n - \boldsymbol{\mu}_i\|^2 \tag{13}$$

Aplicando a condição de otimalidade:

$$\frac{\partial J}{\partial \boldsymbol{\mu}_j} = \sum_{n=1}^{N} 2\, r_{n,j}\left(\mathbf{x}_n - \boldsymbol{\mu}_j\right) = 0 \tag{14}$$

Logo,

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} r_{n,j}\mathbf{x}_n}{\sum_{n=1}^{N} r_{n,j}}, \tag{15}$$

para $j = 1, \ldots, k$. Note que o denominador equivale ao número de padrões $\mathbf{x}_n$ atribuídos ao *cluster* j. Sendo assim, $\boldsymbol{\mu}_j$ nada mais é do que o vetor resultante da média aritmética de todos os padrões pertencentes ao *cluster* j (daí o nome *k-means*).

O algoritmo *k-means* (versão batelada) aplica sucessivamente os dois passos anteriores até que um critério de parada (e.g., variação nas posições dos protótipos inferior a um limiar pequeno) seja atingido.

Características:

- O método seguramente converge, mas pode ser para um mínimo local de J.
- Além disso, o resultado é fortemente dependente da condição inicial. Pode, também, ser sensível a *outliers*, uma vez que o critério explora a norma L_2 do vetor diferença.

Há, também, uma versão padrão-a-padrão (*online*) do algoritmo *k-means*, semelhante a um método de gradiente descendente:

$$\boldsymbol{\mu}_j^{(\text{NOVO})} = \boldsymbol{\mu}_j^{(\text{ANTIGO})} + \eta_n \underbrace{\left(\mathbf{x}_n - \boldsymbol{\mu}_j^{(\text{ANTIGO})}\right)}_{\text{Vetor gradiente}}, \tag{16}$$

onde η_n é a taxa de aprendizagem (tipicamente, decresce de forma monotônica conforme n aumenta). Neste caso, somente o protótipo mais próximo ao padrão $\mathbf{x}_n$ é que experimenta a atualização. Implicitamente, há uma espécie de disputa entre os protótipos para ver qual será escolhido para representar o padrão de entrada. Neste sentido, ocorre um processo de aprendizado competitivo.

O número de *clusters* / protótipos k pode ser escolhido com base em índices de qualidade para clusterização.

3.1. **Extensão: *k-medoids***

Explora uma medida de dissimilaridade mais geral entre cada padrão $\mathbf{x}_n$ e cada protótipo $\boldsymbol{\mu}_i$, denotada por $v(\mathbf{x}_n, \boldsymbol{\mu}_i)$, levando à seguinte função custo:

$$J = \sum_{n=1}^{N} \sum_{i=1}^{k} r_{n,i} v(\mathbf{x}_n, \boldsymbol{\mu}_i) \tag{17}$$

O procedimento para se obter a solução do algoritmo *k-medoids* segue os dois passos básicos utilizados pelo *k-means*.

Primeiro passo: novamente, atribuímos $\mathbf{x}_n$ ao *cluster* cujo protótipo minimize $v(\mathbf{x}_n, \boldsymbol{\mu}_i)$.

Surge, porém, um problema no segundo passo: dependendo da medida de dissimilaridade adotada, pode não ser possível obter uma solução simples e em forma fechada para $\boldsymbol{\mu}_j$.

Uma alternativa barata consiste em definir $\boldsymbol{\mu}_j$ como sendo um dos padrões $\mathbf{x}_n$ atribuídos ao *cluster* j. Usualmente, opta-se pelo padrão que, sendo visto como protótipo, minimiza a soma das dissimilaridades em relação a todos os demais pontos pertencentes àquele *cluster*.

Caso particular: quando $v(\mathbf{x}_n, \boldsymbol{\mu}_i)$ corresponde à norma L_1 do vetor diferença, *i.e.*, $v(\mathbf{x}_n, \boldsymbol{\mu}_i) = \|\mathbf{x}_n - \boldsymbol{\mu}_i\|_1$, a solução para $\boldsymbol{\mu}_j$ equivale à mediana dos pontos pertencentes ao *cluster* j, e o algoritmo é conhecido como *k-medians*.

Exemplo:

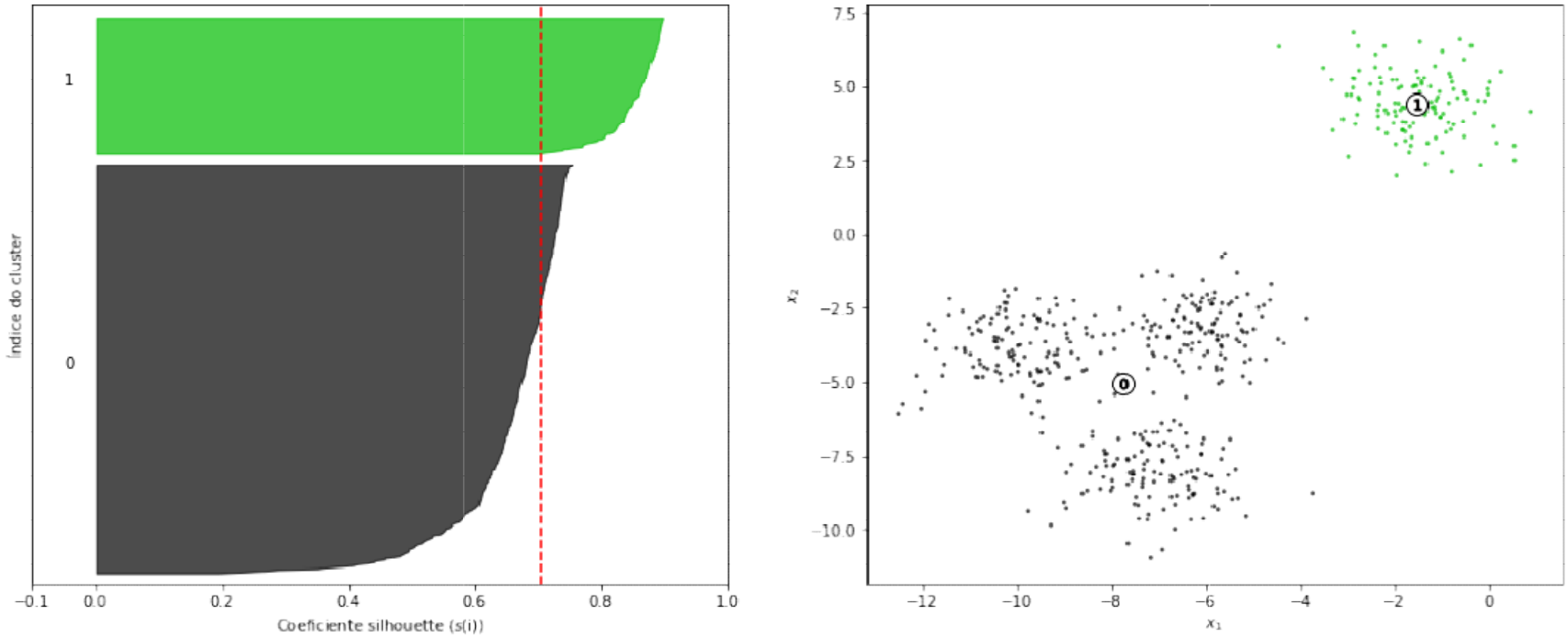


Figura. À esquerda, são mostrados os valores do coeficiente silhouette para cada padrão de entrada. O valor médio (em vermelho) foi igual a 0,7049. À direita, observamos os protótipos obtidos pelo *k-means* para o caso em que $k = 2$, bem como a separação dos dados nos dois *clusters*.

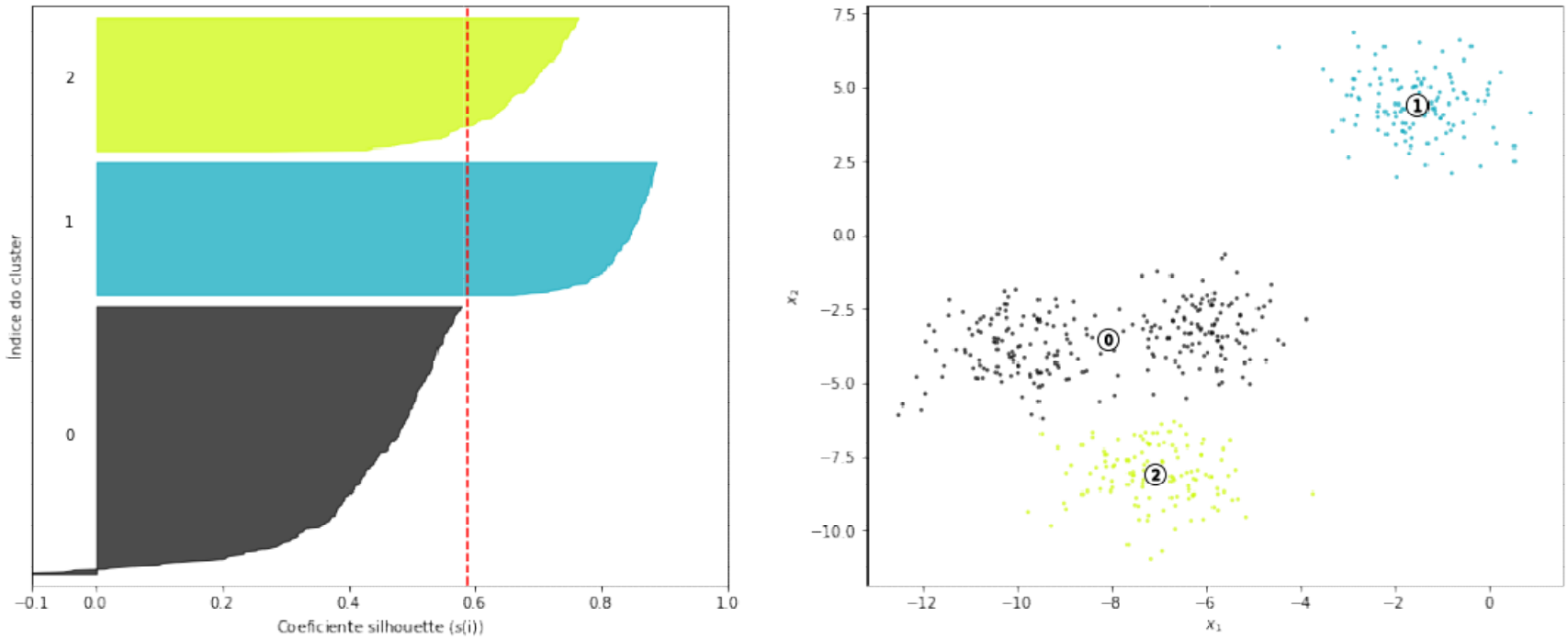


Figura. À esquerda, são mostrados os valores do coeficiente silhouette para cada padrão de entrada. O valor médio (em vermelho) foi igual a 0,5882. À direita, observamos os protótipos obtidos pelo *k-means* para o caso em que $k = 3$, bem como a separação dos dados nos três *clusters*.

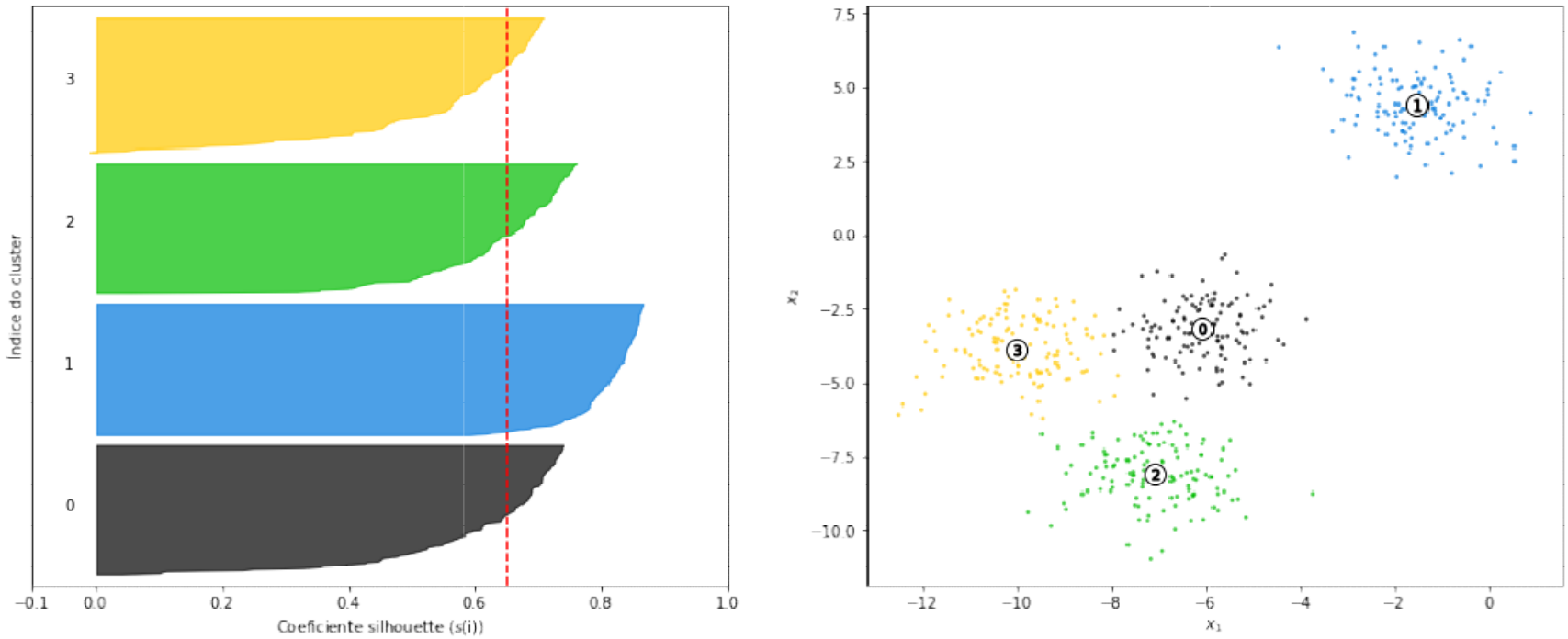


Figura. À esquerda, são mostrados os valores do coeficiente silhouette para cada padrão de entrada. O valor médio (em vermelho) foi igual a 0,6505. À direita, observamos os protótipos obtidos pelo *k-means* para o caso em que $k = 4$, bem como a separação dos dados nos quatro *clusters*.

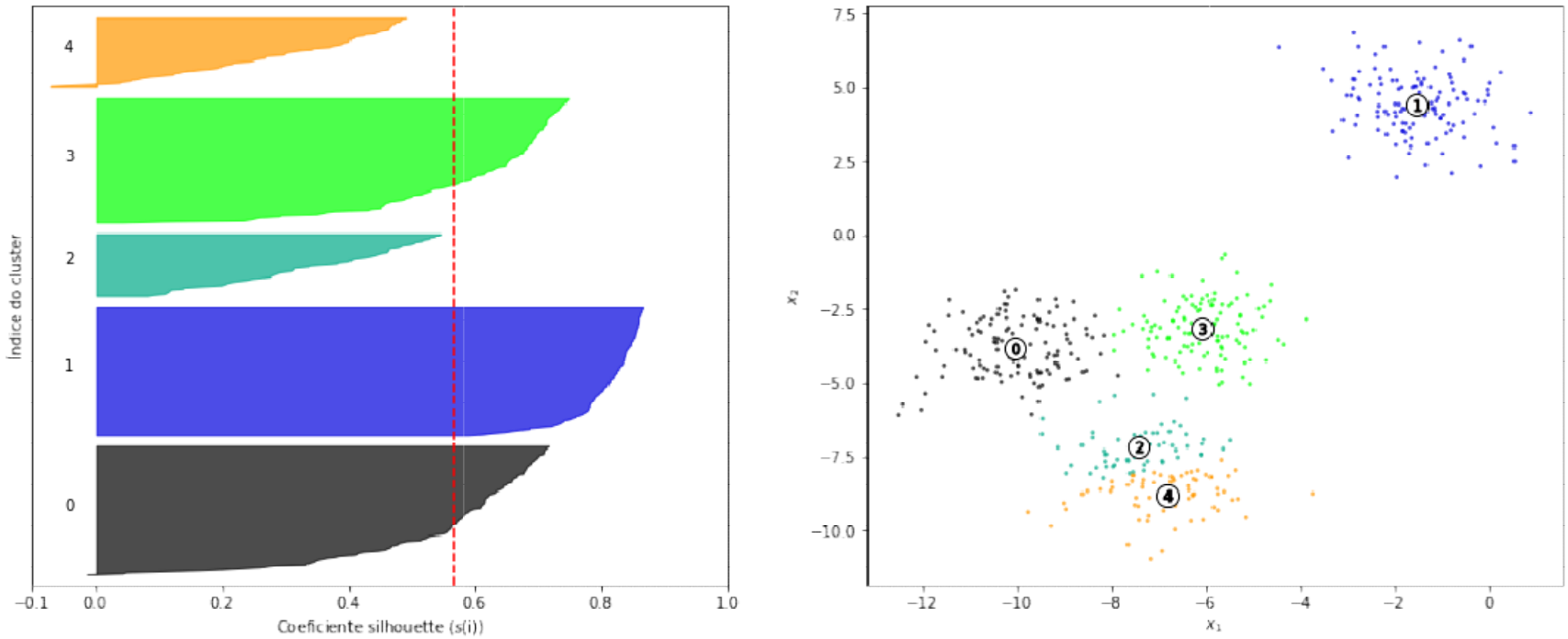


Figura. À esquerda, são mostrados os valores do coeficiente silhouette para cada padrão de entrada. O valor médio (em vermelho) foi igual a 0,5637. À direita, observamos os protótipos obtidos pelo *k-means* para o caso em que $k = 5$, bem como a separação dos dados nos cinco *clusters*.

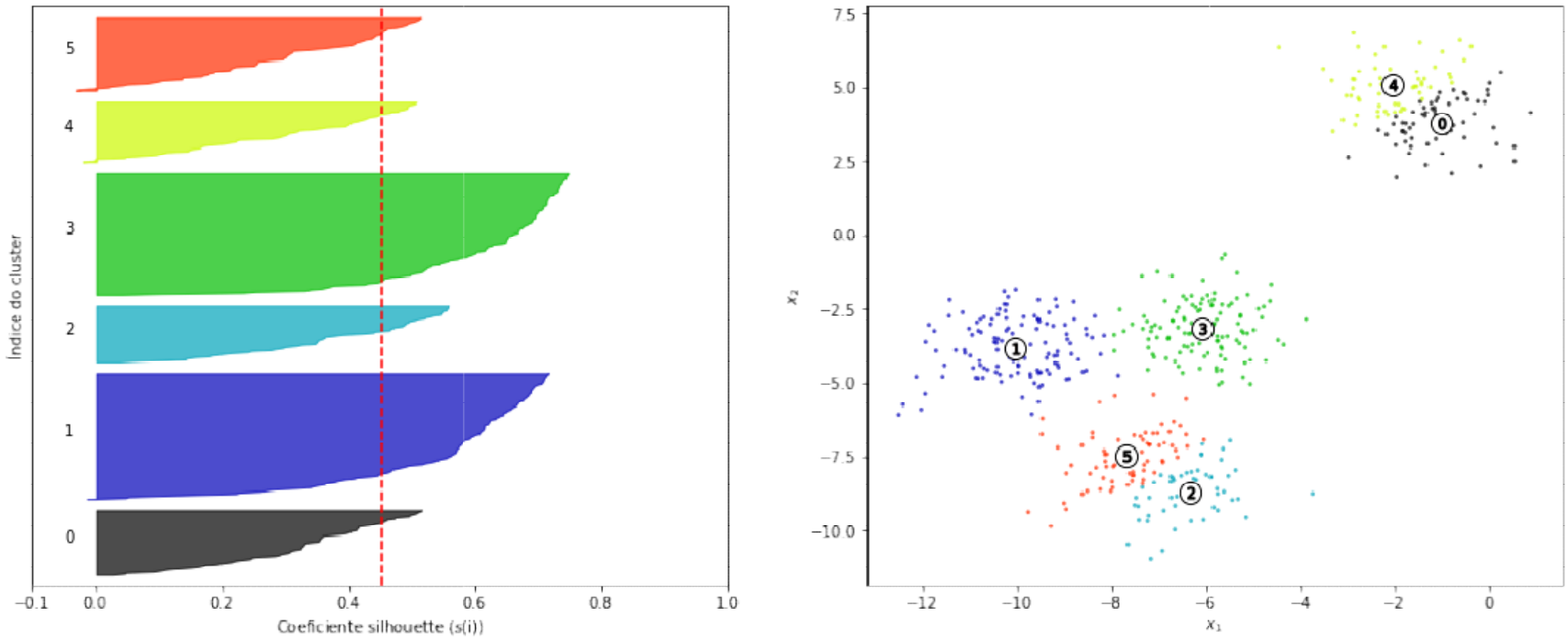


Figura. À esquerda, são mostrados os valores do coeficiente silhouette para cada padrão de entrada. O valor médio (em vermelho) foi igual a 0,4504. À direita, observamos os protótipos obtidos pelo *k-means* para o caso em que $k = 6$, bem como a separação dos dados nos seis *clusters*.

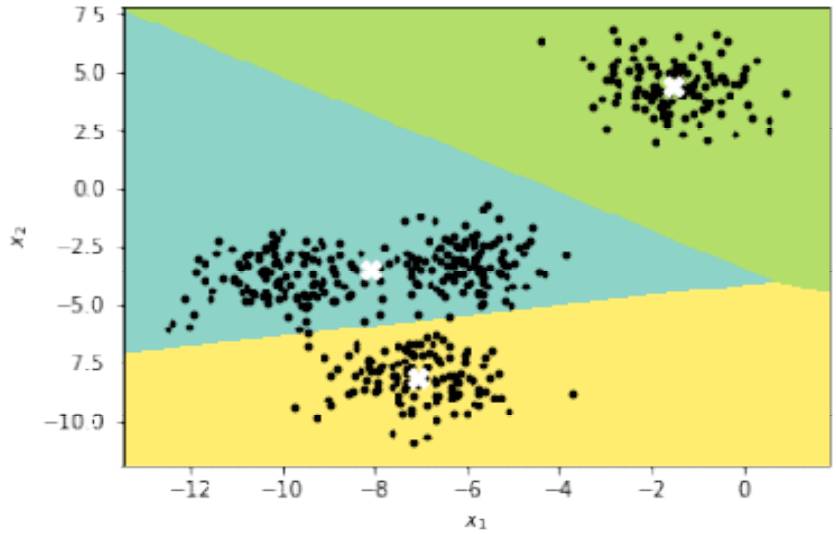


Figura. Regiões referentes a cada um dos protótipos (*clusters*) obtidos pelo algoritmo *k-means*.

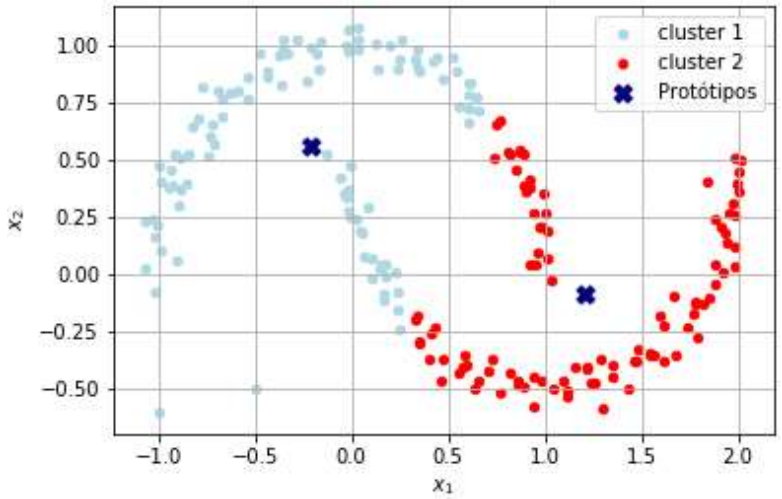


Figura. Divisão do conjunto de dados em dois *clusters* feita pelo *k-means*.

Observação: no conjunto de algoritmos do tipo *k-medoids*, cada padrão de entrada $\mathbf{x}_n$ é atribuído a um, e somente um, *cluster*. Este tipo de decisão rígida, conhecida como *hard assignment*, pode ser razoável para dados situados a uma distância relativamente pequena do respectivo protótipo. Contudo, para dados que ficam aproximadamente no meio do caminho entre dois *clusters* diferentes, este tipo de decisão não parece ser tão apropriada.

4. Algoritmo *fuzzy k-means*

Esta técnica relaxa a condição de *hard-assignment* presente no *k-means* tradicional, permitindo que cada padrão $\mathbf{x}_n$ possua um nível de pertinência gradual (também chamado de nebuloso, ou *fuzzy*) em relação a cada *cluster*. Agora, $r_{n,i} \in [0,1]$ é uma variável contínua e $\sum_{i=1}^{k} r_{n,i} = 1, n = 1, \ldots, N$. Nesta condição, a atribuição feita é do tipo *soft-assignment*.

A função custo do *fuzzy k-means* é definida como (BEZDEK, 1981):

$$J_{\text{fuzzy}} = \sum_{n=1}^{N} \sum_{i=1}^{k} {r_{n,i}}^{\beta} \|\mathbf{x}_n - \boldsymbol{\mu}_i\|^2 \tag{18}$$

O parâmetro β controla o quanto os padrões podem se espalhar por vários *clusters*:

- Quanto maior o valor de β, mais *fuzzy* podem ser os *clusters*.
- $\beta = 1$ nos leva ao *k-means* tradicional.

Condição de otimalidade:

$$\frac{\partial J_{\text{fuzzy}}}{\partial \boldsymbol{\mu}_j} = \sum_{n=1}^{N} 2{r_{n,j}}^{\beta} (\mathbf{x}_n - \boldsymbol{\mu}_i) = 0 \tag{19}$$

Assim,

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} {r_{n,j}}^{\beta} \mathbf{x}_n}{\sum_{n=1}^{N} {r_{n,j}}^{\beta}}. \tag{20}$$

Na análise da derivada parcial com respeito a $r_{n,j}$, também podemos tratar cada padrão de forma isolada. Contudo, temos de incorporar a restrição de que $\sum_{i=1}^{k} r_{n,i} = 1$.

Multiplicadores de Lagrange:

$$\min \mathcal{L}(r_{n,i}, \lambda) = \min \sum_{i=1}^{k} {r_{n,i}}^{\beta} (d_{n,i})^2 - \lambda \left(\sum_{i=1}^{k} r_{n,i} - 1 \right) \tag{21}$$

Calculando as derivadas parciais e igualando a zero:

$$\frac{\partial \mathcal{L}(r_{n,i}, \lambda)}{\partial \lambda} = \left(\sum_{i=1}^{k} r_{n,i} - 1 \right) = 0 \tag{22}$$

e

$$\frac{\partial \mathcal{L}(r_{n,i}, \lambda)}{\partial r_{n,j}} = \beta {r_{n,j}}^{\beta-1} (d_{n,j})^2 - \lambda = 0 \tag{23}$$

Com isso,

$$r_{n,j} = \left(\frac{\lambda}{\beta \left(d_{n,j} \right)^2} \right)^{1/(\beta-1)} \tag{24}$$

Usando (22):

$$\sum_{i=1}^{k} r_{n,i} = 1 \tag{25}$$

Substituindo (24) em (25):

$$\sum_{i=1}^{k} \left(\frac{\lambda}{\beta} \right)^{1/(\beta-1)} \left(\frac{1}{\left(d_{n,i} \right)^2} \right)^{1/(\beta-1)} = \left(\frac{\lambda}{\beta} \right)^{1/(\beta-1)} \sum_{i=1}^{k} \left(\frac{1}{\left(d_{n,i} \right)^2} \right)^{1/(\beta-1)} = 1 \tag{26}$$

Logo,

$$\left(\frac{\lambda}{\beta} \right)^{1/(\beta-1)} = \frac{1}{\sum_{i=1}^{k} \left(\frac{1}{\left(d_{n,i} \right)^2} \right)^{1/(\beta-1)}} \tag{27}$$

Substituindo este resultado em (24), obtemos:

$$r_{n,j} = \frac{1}{\sum_{i=1}^{k} \left(\frac{1}{\left(d_{n,i}\right)^2} \right)^{1/(\beta-1)}} \left(\frac{1}{\left(d_{n,j}\right)^2} \right)^{1/(\beta-1)} \tag{28}$$

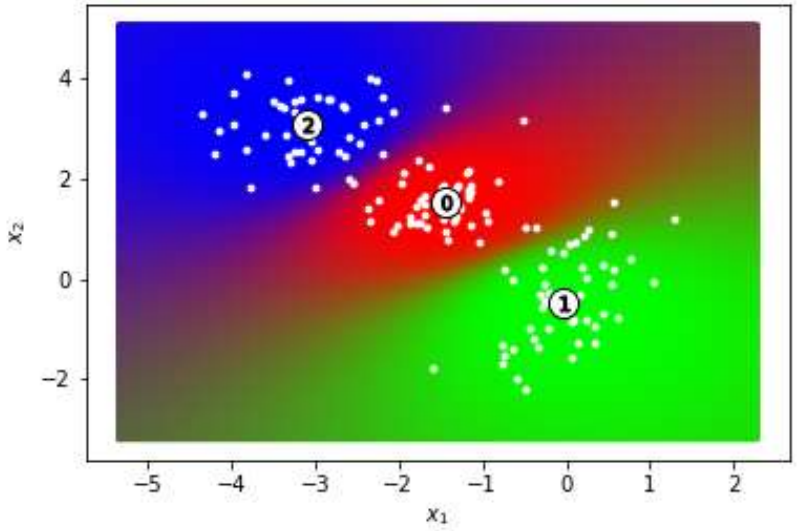
Finalmente, podemos escrever $r_{n,j}$ como:

$$r_{n,j} = \frac{\left(1/d_{n,j}^{\,2} \right)^{1/(\beta-1)}}{\sum_{i=1}^{k} \left(1/d_{n,i}^{\,2} \right)^{1/(\beta-1)}} = \frac{1}{\sum_{i=1}^{k} \left(d_{n,j}/d_{n,i} \right)^{2/(\beta-1)}} \tag{29}$$

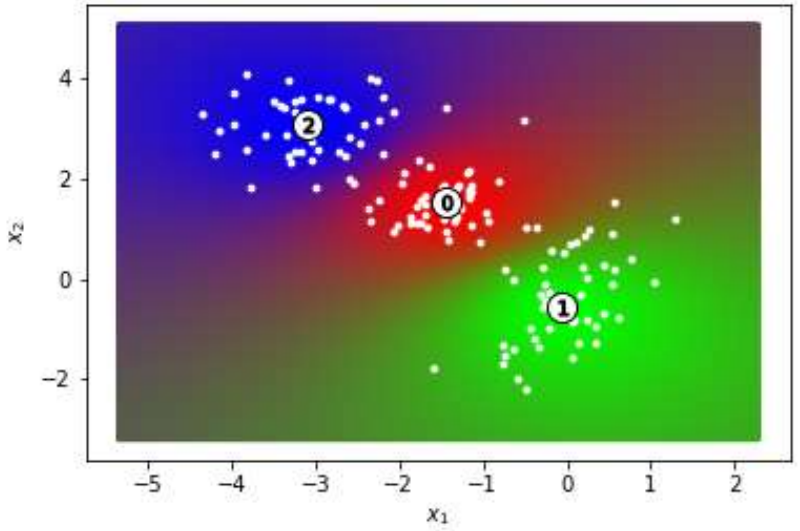
Com isto, o algoritmo *fuzzy k-means* pode ser resumido através do seguinte pseudo-código:

Algoritmo *fuzzy k-means*

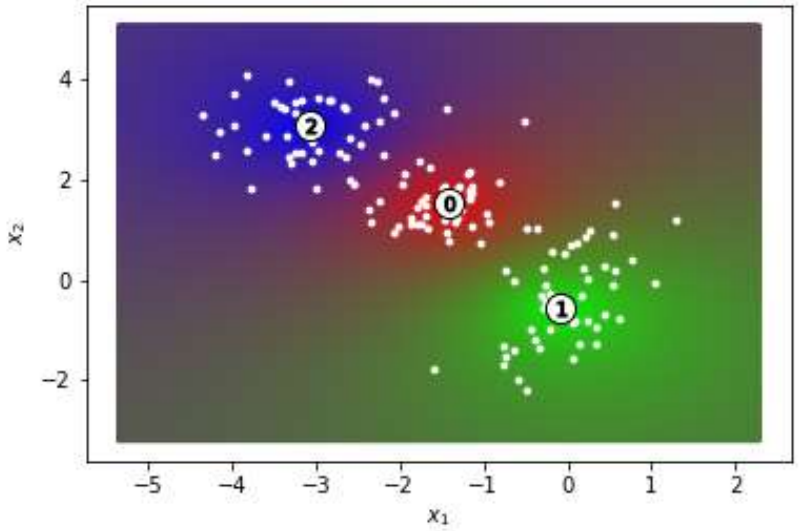
1. Inicialize $k, \boldsymbol{\mu}_i, \ldots, \boldsymbol{\mu}_k, \beta$ e $\mathbf{R} = [r_{n,i}] \in \mathbb{R}^{N \times k}$.
2. Normalize $r_{n,i}$.
3. **Faça**
   3.1. Atualize $\boldsymbol{\mu}_j$ usando (20).
   3.2. Atualize $r_{n,i}$ usando (28).
4. **Até** ocorrerem mudanças suficientemente pequenas em $\boldsymbol{\mu}$ e $\mathbf{R}$.



(a) $\beta = 1{,}5$



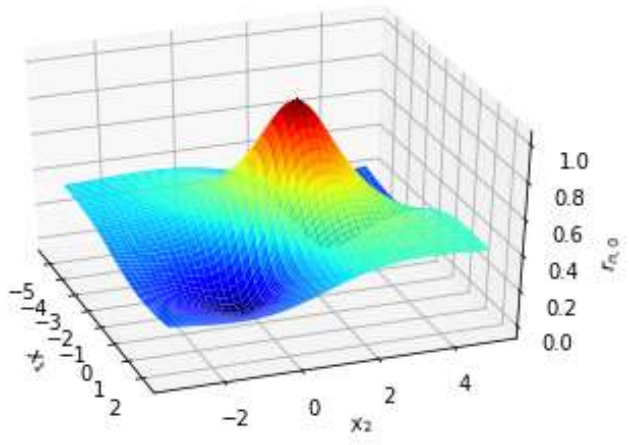
(b) $\beta = 2$



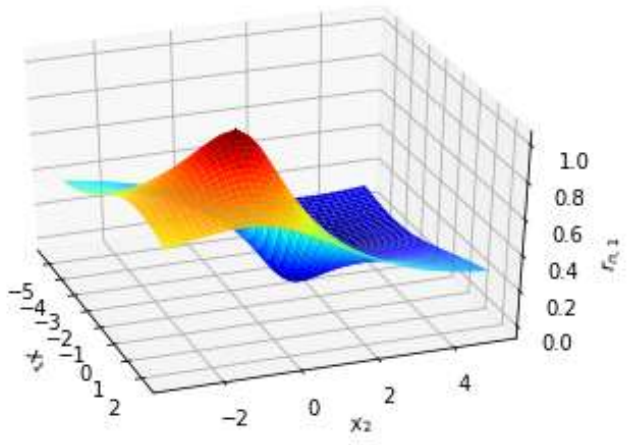
(c) $\beta = 3$

Figura. Os círculos numerados representam os protótipos ($\boldsymbol{\mu}_i, i = 1, \ldots, 3$) obtidos ao final da execução do algoritmo *fuzzy k-means*. Em cada ponto do espaço de entrada, a cor codifica o nível de pertinência daquele dado em relação aos *clusters*.

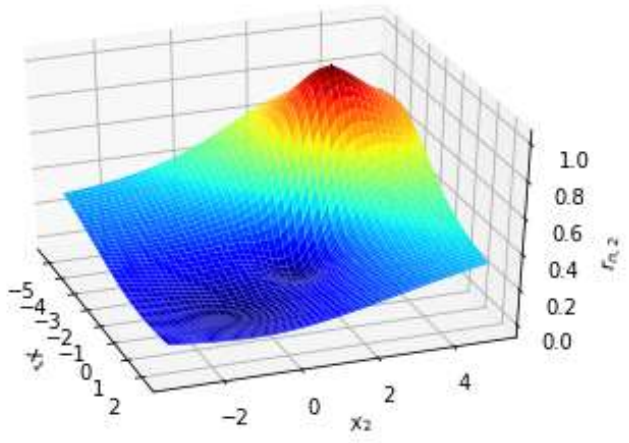
Funções de pertinência ($\beta = 2$):



(a) *Cluster* 0



(b) *Cluster* 1



(c) *Cluster* 2

5. Misturas de gaussianas

Outra possível abordagem para o problema de clusterização baseia-se na ideia de que cada *cluster* no espaço de atributos pode ser bem representado por uma função densidade de probabilidade (PDF, do inglês *probability density function*) própria. Deste modo, considera-se que o conjunto completo de observações pode ser descrito, do ponto de vista estatístico, por uma soma de modelos probabilísticos, em que cada modelo descreve um *cluster* de dados.

O modelo de mistura de gaussianas (GMM, do inglês *Gaussian mixture model*) explora esta ideia supondo que os dados $\{\mathbf{x}_i\}_{i=1}^{N}$ foram amostrados com base numa PDF dada por uma combinação ponderada de gaussianas:

$$p(\mathbf{x}) = \sum_{k=1}^{L} \pi_k G(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{30}$$

onde $\mathbf{x} = [x_1 \quad \ldots \quad x_K]^T$,

$$G(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{K/2}} \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}-\boldsymbol{\mu}_k)}, \tag{31}$$

π_k denota o peso associado a cada componente gaussiana na mistura, $\boldsymbol{\mu}_k \in \mathbb{R}^{K\times 1}$ e $\boldsymbol{\Sigma}_k \in \mathbb{R}^{K\times K}$ representam o vetor de média e a matriz de covariâncias da k-ésima componente gaussiana, $|\mathbf{M}|$ indica o determinante de uma matriz $\mathbf{M}$ e L define o número de componentes na mistura, o qual, no caso de clusterização, representa também o número de *clusters*.

Restrições:

- $\int p(\mathbf{x}) d\mathbf{x} = 1 \Rightarrow \sum_{k=1}^{L} \pi_k = 1.$
- $p(\mathbf{x}) \geq 0 \Rightarrow \pi_k \geq 0.$

O conjunto de pesos $\{\pi_k\}_{k=1}^{L}$ satisfaz os requisitos para ser uma função probabilidade de massa. Sendo assim, cada π_k corresponde a $p(k)$, *viz.*, a probabilidade *a priori* associada à k-ésima componente gaussiana.

5.1. **Probabilidade *a posteriori***

Também conhecida como a responsabilidade que a k-ésima componente tem de explicar a observação **x.**

$$P(k|\mathbf{x}) = \frac{P(\mathbf{x}|k)P(k)}{P(\mathbf{x})}. \tag{32}$$

Ora:

- $P(\mathbf{x}|k) = G(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.
- $P(k) = \pi_k$.
- $P(\mathbf{x}) = \sum_{j=1}^{L} P(j)P(\mathbf{x}|j) = \sum_{j=1}^{L} \pi_j G\left(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)$.

Portanto:

$$P(k|\mathbf{x}) = \frac{\pi_k G(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{L} \pi_j G\left(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)}. \tag{33}$$

5.2. **Formulação com variáveis latentes**

Seja $\mathbf{z} = [z_1 \ \ldots \ z_L]^T$ o vetor de variáveis latentes do tipo *one-hot encoding*.

- Somente um elemento $z_k = 1$, enquanto os demais são nulos.
- Cada $z_k \in \{0,1\}$ e $\sum z_k = 1$.

Há L configurações possíveis para o vetor $\mathbf{z}$ (e.g., [1 0 ... 0], [0 1 ... 0], *etc*.), sendo que cada uma representa uma atribuição específica de um padrão $\mathbf{x}$ a uma das componentes da mistura, *i.e.*, a um dos *clusters*.

- $P(z_k = 1) = \pi_k$.
- $P(\mathbf{z}) = \prod_{k=1}^{L} {\pi_k}^{z_k}$.

Assim, $P(\mathbf{x}|z_k = 1) = P(\mathbf{x}|k) = G(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$.

Logo,

$$P(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{L} G(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}. \tag{34}$$

Com isto,

$$P(\mathbf{z}) = \sum_{\mathbf{z}} P(\mathbf{z})P(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{L} \pi_k G(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \quad (35)$$

Para cada vetor $\mathbf{x}$ observado, existe uma única variável latente $\mathbf{z}$ correspondente.

5.3. **Ajuste dos parâmetros da mistura**

Seja $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ o conjunto de observações disponíveis. Desejamos modelar estes dados usando uma mistura de gaussianas, de maneira que cada componente gaussiana represente um agrupamento (*cluster*).

Dados: $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1{}^T \\ \vdots \\ \mathbf{x}_N{}^T \end{bmatrix} \in \mathbb{R}^{N \times K}$.

Supondo que as amostras coletadas são *i.i.d.*, podemos escrever a função de verossimilhança como:

$$P(\mathbf{X}|\boldsymbol{\Pi}, \mathbf{M}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} \left(\sum_{k=1}^{L} \pi_k G(\mathbf{x}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right). \quad (36)$$

Aplicando o logaritmo:

$$\ln P(\mathbf{X}|\mathbf{\Pi}, \mathbf{M}, \mathbf{\Sigma}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{L} \pi_k G(\mathbf{x}_n; \boldsymbol{\mu}_k, \mathbf{\Sigma}_k) \right). \quad (37)$$

O objetivo é determinar os parâmetros $\boldsymbol{\mu}_k$, $\mathbf{\Sigma}_k$ e π_k de cada componente gaussiana de modo a maximizar o logaritmo da verossimilhança (*log-likelihood*).

Problema: caso uma componente gaussiana tenha um vetor de média idêntico a uma das amostras $\mathbf{x}_n$, sua contribuição na função de verossimilhança se reduz a:

$$G(\mathbf{x}_n; \mathbf{x}_n, \mathbf{\Sigma}_k) = \frac{1}{(2\pi)^{K/2}} \frac{1}{{\sigma_k}^K}, \quad (38)$$

supondo, por simplicidade, que $\mathbf{\Sigma}_k = {\sigma_k}^2 \mathbf{I}$ (diagonal). Para $\sigma_k \to 0$, este termo cresce indefinidamente, fazendo com que $\ln P(\mathbf{X}|\mathbf{\Pi}, \mathbf{M}, \mathbf{\Sigma}) \to \infty$.

É preciso, portanto, criar mecanismos para evitar a obtenção destas configurações indesejáveis.

Por exemplo, durante a otimização da função de verossimilhança, é possível detectar se uma componente gaussiana está colapsando para uma das amostras e, então, reinicializar sua média para um vetor aleatório, e sua matriz de covariâncias para que tenha elementos de magnitude mais elevada.

Observação: explorando uma abordagem bayesiana para o ajuste do GMM, associada à ideia de inferência variacional, este problema é naturalmente contornado.

Desafio:

$$\max_{\mathbf{\Pi},\mathbf{M},\mathbf{\Sigma}} \ln P(\mathbf{X}|\mathbf{\Pi},\mathbf{M},\mathbf{\Sigma}) \tag{39}$$

Um caminho possível para resolver (39) consiste em lançar mão de métodos baseados no vetor gradiente.

5.4. **Cálculo do gradiente da verossimilhança**

Vamos aplicar as condições de otimalidade a fim de obter expressões para a atualização dos parâmetros do GMM.

5.4.1. Derivada com respeito a $\boldsymbol{\mu}_k$

$$\frac{\partial \ln P(\mathbf{X}|\boldsymbol{\Pi}, \mathbf{M}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}_k} = \sum_{n=1}^{N} \frac{1}{\sum_{l=1}^{L} \pi_l G(\mathbf{x}_n; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} \frac{\partial \sum_{l=1}^{L} \pi_l G(\mathbf{x}_n; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}{\partial \boldsymbol{\mu}_k} = 0 \qquad (40)$$

Então:

$$\frac{\partial \ln P(\mathbf{X}|\boldsymbol{\Pi}, \mathbf{M}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}_k} = \sum_{n=1}^{N} \underbrace{\frac{\pi_k G(\mathbf{x}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{L} \pi_l G(\mathbf{x}_n; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}}_{P(k|\mathbf{x}_n)} \boldsymbol{\Sigma}_k{}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0 \qquad (41)$$

O primeiro fator dentro do somatório é, na verdade, a probabilidade *a posteriori* da k-ésima componente gaussiana, $P(k|\mathbf{x}_n)$.

$$\frac{\partial \ln P(\mathbf{X}|\mathbf{\Pi}, \mathbf{M}, \mathbf{\Sigma})}{\partial \boldsymbol{\mu}_k} = \sum_{n=1}^{N} P(k|\mathbf{x}_n)\boldsymbol{\Sigma}_k{}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0 \tag{42}$$

Multiplicando ambos os lados por $\boldsymbol{\Sigma}_k$ (definida positiva, logo não-singular):

$$\sum_{n=1}^{N} P(k|\mathbf{x}_n)(\mathbf{x}_n) = \sum_{n=1}^{N} P(k|\mathbf{x}_n)(\boldsymbol{\mu}_k) \tag{43}$$

Com isto,

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} P(k|\mathbf{x}_n)(\mathbf{x}_n)}{\sum_{n=1}^{N} P(k|\mathbf{x}_n)} = \frac{\sum_{n=1}^{N} P(k|\mathbf{x}_n)(\mathbf{x}_n)}{N_k}, \tag{44}$$

onde $N_k = \sum_{n=1}^{N} P(k|\mathbf{x}_n)$ representa o número efetivo de amostras atribuídas à componente (ou ao *cluster*) k.

5.4.2. Derivada com respeito a $\boldsymbol{\Sigma}_k{}^{-1}$

Para facilitar um pouco o desenvolvimento, vamos calcular a derivada com respeito a $\boldsymbol{\Sigma}_k{}^{-1}$.

$$\frac{\partial \ln P(\mathbf{X}|\mathbf{\Pi},\mathbf{M},\mathbf{\Sigma})}{\partial {\mathbf{\Sigma}_k}^{-1}} = \sum_{n=1}^{N} \frac{1}{\sum_{l=1}^{L} \pi_l G(\mathbf{x}_n;\boldsymbol{\mu}_l,\mathbf{\Sigma}_l)} \frac{\partial \sum_{l=1}^{L} \pi_l G(\mathbf{x}_n;\boldsymbol{\mu}_l,\mathbf{\Sigma}_l)}{\partial {\mathbf{\Sigma}_k}^{-1}} = 0 \qquad (45)$$

Ora,

$$\frac{\partial \sum_{l=1}^{L} \pi_l G(\mathbf{x}_n;\boldsymbol{\mu}_l,\mathbf{\Sigma}_l)}{\partial {\mathbf{\Sigma}_k}^{-1}} = \pi_k \frac{\partial \left(\frac{1}{(2\pi)^{K/2}} \frac{1}{|\mathbf{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_n-\boldsymbol{\mu}_k)^T {\mathbf{\Sigma}_k}^{-1}(\mathbf{x}_n-\boldsymbol{\mu}_k)} \right)}{\partial {\mathbf{\Sigma}_k}^{-1}} \qquad (46)$$

Para facilitar este cálculo, vamos utilizar a seguinte propriedade:

$$\frac{\partial \ln f}{\partial {\mathbf{\Sigma}_k}^{-1}} = \frac{1}{f} \frac{\partial f}{\partial {\mathbf{\Sigma}_k}^{-1}} \qquad (47)$$

Então,

$$\frac{\partial f}{\partial {\mathbf{\Sigma}_k}^{-1}} = f \frac{\partial \ln f}{\partial {\mathbf{\Sigma}_k}^{-1}}. \qquad (48)$$

Seja $f = G(\mathbf{x}_n;\boldsymbol{\mu}_k,\mathbf{\Sigma}_k) = \frac{1}{(2\pi)^{K/2}} \frac{1}{|\mathbf{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_n-\boldsymbol{\mu}_k)^T {\mathbf{\Sigma}_k}^{-1}(\mathbf{x}_n-\boldsymbol{\mu}_k)}$.

Então, $\ln f = -\left(\frac{K}{2}\right) \ln 2\pi - \frac{1}{2} \ln|\mathbf{\Sigma}_k| - \frac{1}{2}(\mathbf{x}_n-\boldsymbol{\mu}_k)^T {\mathbf{\Sigma}_k}^{-1}(\mathbf{x}_n-\boldsymbol{\mu}_k)$.

Assim, aplicando a derivada com respeito a $\boldsymbol{\Sigma}_k{}^{-1}$ termo a termo, obtemos:

$$\frac{\partial \ln f}{\partial \boldsymbol{\Sigma}_k{}^{-1}} = -\frac{1}{2}\frac{\partial \ln|\boldsymbol{\Sigma}_k|}{\partial \boldsymbol{\Sigma}_k{}^{-1}} - \frac{1}{2}\frac{\partial (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k{}^{-1}(\mathbf{x}_n - \boldsymbol{\mu}_k)}{\partial \boldsymbol{\Sigma}_k{}^{-1}} \tag{49}$$

Propriedades:

a) $|\boldsymbol{\Sigma}_k| = \frac{1}{|\boldsymbol{\Sigma}_k{}^{-1}|} \Rightarrow \ln|\boldsymbol{\Sigma}_k| = \ln\frac{1}{|\boldsymbol{\Sigma}_k{}^{-1}|} = -\ln\left|\boldsymbol{\Sigma}_k{}^{-1}\right|$.

b) $\frac{\partial \ln|\mathbf{M}|}{\partial \mathbf{M}} = (\mathbf{M}^{-1})^T$.

Combinando as propriedades a) e b), obtemos:

$$-\frac{1}{2}\frac{\partial \ln|\boldsymbol{\Sigma}_k|}{\partial \boldsymbol{\Sigma}_k{}^{-1}} = \frac{1}{2}\frac{\partial \ln\left|\boldsymbol{\Sigma}_k{}^{-1}\right|}{\partial \boldsymbol{\Sigma}_k{}^{-1}} = \frac{1}{2}\boldsymbol{\Sigma}_k{}^{T} = \frac{1}{2}\boldsymbol{\Sigma}_k. \tag{50}$$

c) $\frac{\partial \mathbf{a}^T\mathbf{M}\mathbf{b}}{\partial \mathbf{M}} = \mathbf{a}\mathbf{b}^T \Rightarrow \frac{\partial (\mathbf{x}_n-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k{}^{-1}(\mathbf{x}_n-\boldsymbol{\mu}_k)}{\partial \boldsymbol{\Sigma}_k{}^{-1}} = (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$.

Portanto:

$$\frac{\partial \ln f}{\partial \boldsymbol{\Sigma}_k{}^{-1}} = \frac{1}{2}\boldsymbol{\Sigma}_k - \frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T. \tag{51}$$

Usando (48), obtemos:

$$\frac{\partial f}{\partial \boldsymbol{\Sigma}_k{}^{-1}} = \frac{f}{2}(\boldsymbol{\Sigma}_k - (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T) \tag{52}$$

Retornando a (45):

$$\frac{\partial \ln P(\mathbf{X}|\boldsymbol{\Pi}, \mathbf{M}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}_k{}^{-1}} = \sum_{n=1}^{N} \frac{\pi_k G(\mathbf{x}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{L} \pi_l G(\mathbf{x}_n; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} \frac{1}{2}(\boldsymbol{\Sigma}_k - (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T) = 0$$

Mais uma vez, o primeiro fator dentro do somatório corresponde a $P(k|\mathbf{x}_n)$. Assim,

$$\frac{\partial \ln P(\mathbf{X}|\boldsymbol{\Pi}, \mathbf{M}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}_k{}^{-1}} = \sum_{n=1}^{N} P(k|\mathbf{x}_n) \frac{1}{2}(\boldsymbol{\Sigma}_k - (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T) = 0 \tag{53}$$

Então,

$$\sum_{n=1}^{N} P(k|\mathbf{x}_n)\boldsymbol{\Sigma}_k = \sum_{n=1}^{N} P(k|\mathbf{x}_n)(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \tag{54}$$

Finalmente,

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} P(k|\mathbf{x}_n)(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T. \tag{55}$$

5.4.3. Derivada com respeito a π_k

No tocante aos pesos π_k, queremos maximizar $\ln P(\mathbf{X}|\boldsymbol{\Pi}, \mathbf{M}, \boldsymbol{\Sigma})$ e, ao mesmo tempo, satisfazer a restrição de que $\sum_{l=1}^{L} \pi_l = 1$.

Para isto, vamos utilizar o método dos multiplicadores de Lagrange.

$$\max_{\boldsymbol{\Pi},\mathbf{M},\boldsymbol{\Sigma},\lambda} \mathcal{L}(\mathbf{X}, \boldsymbol{\Pi}, \mathbf{M}, \boldsymbol{\Sigma}, \lambda) = \max \ln P(\mathbf{X}|\boldsymbol{\Pi}, \mathbf{M}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{l=1}^{L} \pi_l - 1 \right) \tag{56}$$

Derivando $\mathcal{L}(\mathbf{X}, \boldsymbol{\Pi}, \mathbf{M}, \boldsymbol{\Sigma}, \lambda)$ com respeito a λ:

$$\frac{\partial \mathcal{L}(\mathbf{X}, \boldsymbol{\Pi}, \mathbf{M}, \boldsymbol{\Sigma}, \lambda)}{\partial \lambda} = \sum_{l=1}^{L} \pi_l - 1 \Rightarrow \sum_{l=1}^{L} \pi_l = 1. \tag{57}$$

Derivando com respeito a π_k**:**

$$\frac{\partial \ln P(\mathbf{X}|\boldsymbol{\Pi},\mathbf{M},\boldsymbol{\Sigma})}{\partial \pi_k} = \sum_{n=1}^{N} \frac{1}{\sum_{l=1}^{L} \pi_l G(\mathbf{x}_n;\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l)} \frac{\partial \sum_{l=1}^{L} \pi_l G(\mathbf{x}_n;\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l)}{\partial \boldsymbol{\Sigma}_k{}^{-1}}$$
$$= \sum_{n=1}^{N} \frac{1}{\sum_{l=1}^{L} \pi_l G(\mathbf{x}_n;\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l)} G(\mathbf{x}_n;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k). \quad (58)$$

Por sua vez,

$$\frac{\partial \lambda\left(\sum_{l=1}^{L} \pi_l - 1\right)}{\partial \pi_k} = \lambda. \quad (59)$$

Então:

$$\frac{\partial \mathcal{L}(\mathbf{X},\boldsymbol{\Pi},\mathbf{M},\boldsymbol{\Sigma},\lambda)}{\partial \pi_k} = \sum_{n=1}^{N} \frac{1}{\sum_{l=1}^{L} \pi_l G(\mathbf{x}_n;\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l)} G(\mathbf{x}_n;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k) + \lambda = 0 \quad (60)$$

Multiplicando ambos os lados por π_k:

$$\sum_{n=1}^{N} \frac{\pi_k G(\mathbf{x}_n;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{l=1}^{L} \pi_l G(\mathbf{x}_n;\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l)} + \lambda \pi_k = 0 \quad (61)$$

Tomando a $\sum_{k=1}^{L}$:

$$\sum_{n=1}^{N} \underbrace{\frac{\sum_{k=1}^{L} \pi_k G(\mathbf{x}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{L} \pi_l G(\mathbf{x}_n; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}}_{=1} + \lambda \underbrace{\sum_{k=1}^{L} \pi_k}_{=1} = 0 \tag{62}$$

Com isso,

$$\sum_{n=1}^{N} 1 + \lambda = 0 \Rightarrow \lambda = -N. \tag{63}$$

Substituindo o valor de λ em (61), obtemos:

$$\pi_k = {}^{1}/_{N} \sum_{n=1}^{N} \frac{\pi_k G(\mathbf{x}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{L} \pi_l G(\mathbf{x}_n; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} = \frac{\sum_{n=1}^{N} P(k|\mathbf{x}_n)}{N} = \frac{N_k}{N}. \tag{64}$$

5.5. **Algoritmo *Expectation-Maximization* (EM)**

As expressões obtidas para $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ e π_k não formam soluções fechadas para os parâmetros, pois elas dependem das responsabilidades $P(k|\mathbf{x}_n)$, as quais, por sua vez, dependem dos próprios parâmetros da mistura de gaussianas.

Neste contexto, surge a ideia de se utilizar um procedimento iterativo para buscar a solução de máxima verossimilhança, fazendo a atualização sucessiva dos valores de $P(k|\mathbf{x}_n)$ e, em seguida, dos parâmetros $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ e π_k. A formalização desta ideia dá origem ao método conhecido como *expectation-maximization* (EM).

Algoritmo EM

1. Inicialize $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ e π_k, $k = 1, \ldots, L$ e calcule a *log-likelihood*.
2. **Passo E:** determine as probabilidades *a posteriori*:

$$P(k|\mathbf{x}_n) = \frac{\pi_k G(\mathbf{x}_n; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{L} \pi_j G(\mathbf{x}_n; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

3. **Passo M:** re-estime os parâmetros da mistura usando $P(k|\mathbf{x}_n)$:

$$N_k = \sum_{n=1}^{N} P(k|\mathbf{x}_n) \qquad \boldsymbol{\Sigma}_k{}^{(\text{novo})} = \frac{1}{N_k} \sum_{n=1}^{N} P(k|\mathbf{x}_n)\left(\mathbf{x}_n - \boldsymbol{\mu}_k{}^{(\text{novo})}\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_k{}^{(\text{novo})}\right)^T$$

$$\boldsymbol{\mu}_k{}^{(\text{novo})} = \frac{1}{N_k} \sum_{n=1}^{N} P(k|\mathbf{x}_n)(\mathbf{x}_n) \qquad \pi_k{}^{(novo)} = \frac{N_k}{N}$$

4. Avalie a *log-likelihood*:

$$\ln P(\mathbf{X}|\boldsymbol{\Pi}, \mathbf{M}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left(\sum_{k=1}^{L} \pi_k{}^{(novo)} G\left(\mathbf{x}_n; \boldsymbol{\mu}_k{}^{(\text{novo})}, \boldsymbol{\Sigma}_k{}^{(\text{novo})}\right) \right)$$

5. Repita os passos 2 a 4 caso não tenha ocorrido convergência da *log-likelihood* e/ou dos parâmetros.

Comentários:

- A cada passo do algoritmo EM, seguramente o valor da *log-likelihood* não diminui (BISHOP, 2006).
- Assim como o algoritmo *k-means*, o resultado final do EM depende da inicialização; além disso, como possivelmente a função de verossimilhança possui múltiplos ótimos locais, a convergência do algoritmo se dá com garantias para um ótimo local.
- Assim como ocorre no caso do algoritmo *fuzzy k-means*, cada padrão de entrada $\mathbf{x}$ possui uma probabilidade $P(k|\mathbf{x})$ de pertencer a cada um dos *clusters*, ou, equivalentemente, de ter sido gerado por cada uma das componentes gaussianas. Deste modo, o GMM também realiza uma atribuição suave (*soft-assignment*).
- Podemos usar o *k-means* para obter uma inicialização mais adequada para os vetores de média $\boldsymbol{\mu}_k$ das componentes gaussianas. Neste caso, as matrizes de

covariâncias podem ser inicializadas com base nas estimativas amostrais, usando, para a *k-ésima* componente, os dados atribuídos ao *cluster* k. Por fim, π_k pode ser inicializado como a fração dos dados que foi atribuída ao *cluster* k.

Exemplo:

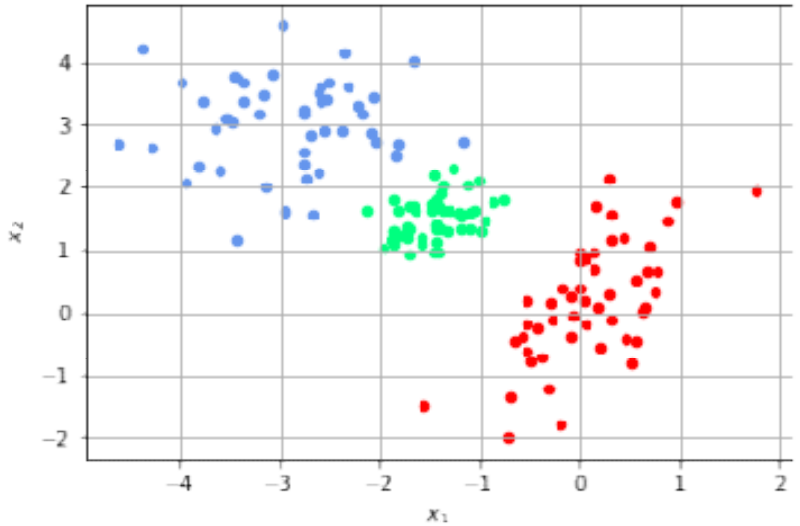


Figura. Distribuição dos dados disponíveis, os quais, por construção, estão associados a três diferentes agrupamentos (*clusters*), gerados por distribuições gaussianas com diferentes matrizes de covariâncias.

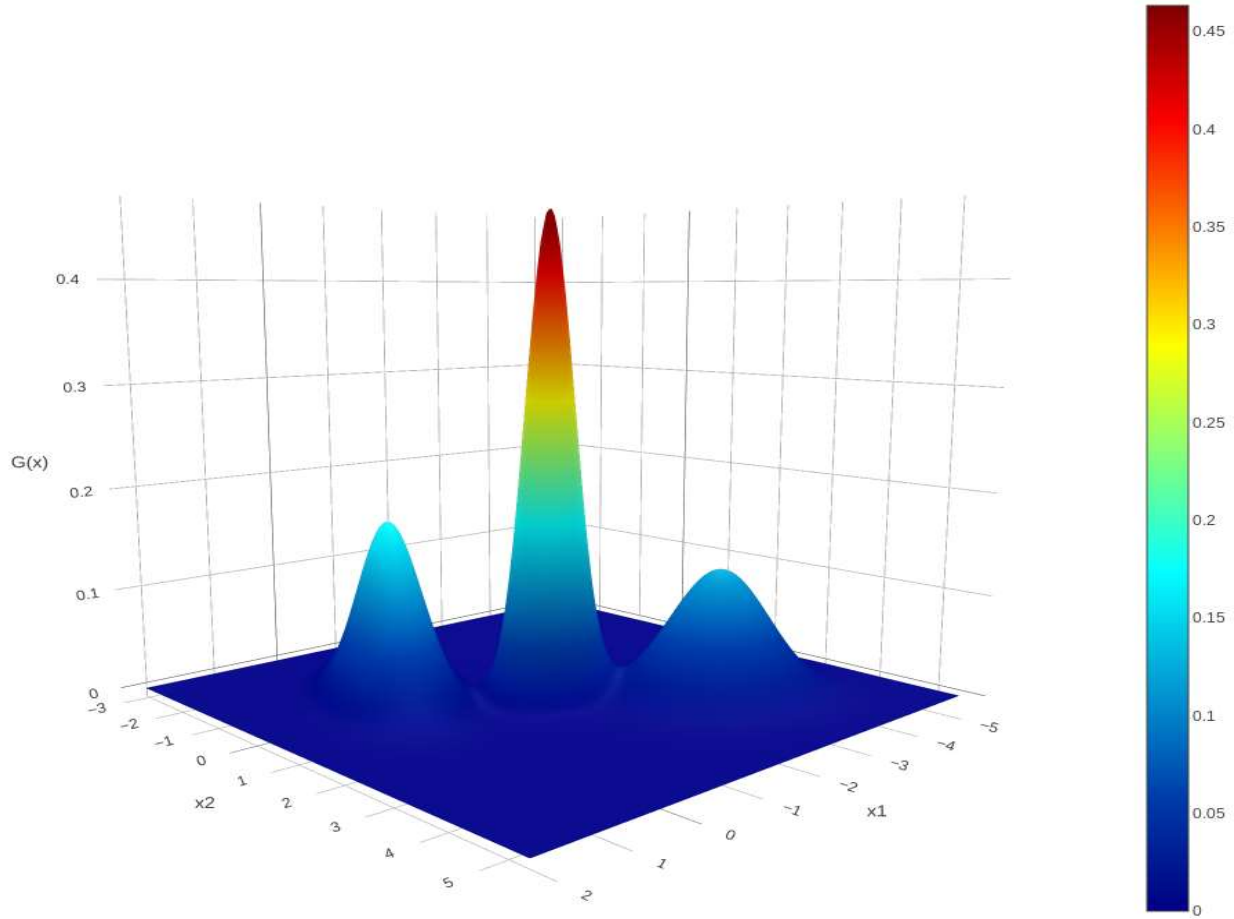


Figura. PDF construída pelo GMM considerando três componentes gaussianas.

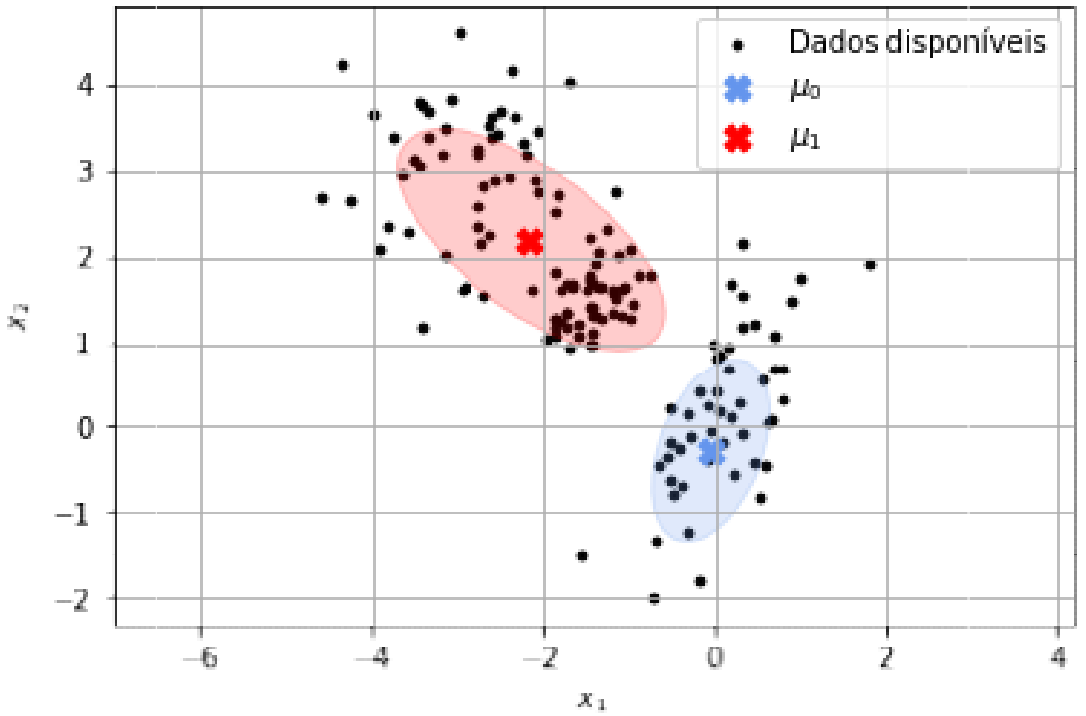


Figura. Configuração final de uma mistura de duas componentes gaussianas. As elipses indicam o perfil das curvas de nível associadas a cada componente gaussiana, sendo um reflexo da matriz de covariâncias obtida pelo algoritmo EM.

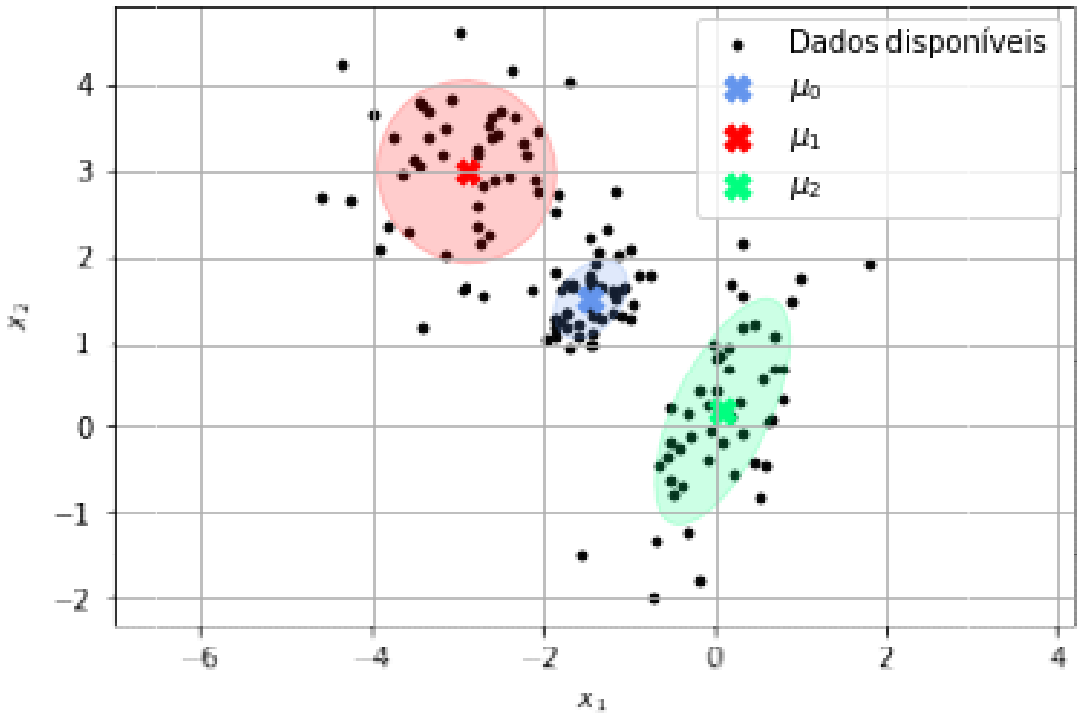


Figura. Configuração final de uma mistura de três componentes gaussianas.

6. Algoritmo DBSCAN

O algoritmo intitulado *density-based spatial clustering of applications with noise* (DBSCAN), proposto em (Ester et. al., 1996), é um método que usa a informação da densidade de pontos em torno de cada padrão para definir os *clusters*.

O DBSCAN é constituído de duas etapas: (*i*) na primeira etapa, cada amostra $\mathbf{x}(i)$ é rotulada como sendo um **ponto central**, um **ponto de fronteira** ou, então, **ruído**; (*ii*) na segunda etapa, os **pontos centrais** e de **fronteira** são agrupados em *clusters*.

- Um padrão $\mathbf{x}(i)$ (ou, simplesmente, i) é considerado um **ponto central** se há, pelo menos, M_p padrões com distância inferior a ε em relação a ele (incluindo ele próprio). Todos os pontos dentro deste raio de vizinhança são ditos diretamente alcançáveis por i.

- Um padrão i é considerado um **ponto de fronteira** se ele possui menos que M_p vizinhos dentro de um raio ε, e ele próprio está a uma distância inferior a ε em relação a um ponto central.
- Todos os pontos que não são centrais nem de fronteira são considerados como **ruído**.

O algoritmo DBSCAN possui, portanto, dois **parâmetros**: o limiar de similaridade (ou raio de vizinhança) ε e o número mínimo de pontos M_p.

Uma vez que todos os padrões foram rotulados, o algoritmo DBSCAN constrói os *clusters* da seguinte maneira:

- Cada ponto central forma um *cluster* juntamente com todos os pontos (centrais ou de fronteira) que são alcançáveis por ele.

- A extensão de um *cluster* é definida com base na noção de pontos conectados por densidade (*density-connected*): dois pontos p e q são conectados por densidade se existe algum outro ponto r tal que tanto p quanto q são alcançáveis por r.
- Assim, um cluster contém todos os pontos que satisfazem as seguintes propriedades:
  - Todos os pontos designados ao mesmo *cluster* são mutuamente conectados por densidade.
  - Se um ponto arbitrário do espaço de entrada pode ser conectado por densidade com qualquer ponto de um *cluster*, então ele também deve fazer parte deste *cluster*.

Vantagens:

- Não requer que o número de *clusters* seja pré-especificado; além disso, pode encontrar *clusters* com formatos arbitrários.

- Nem todos os padrões da base de dados são atribuídos aos *clusters*: os pontos classificados como **ruído** não pertencem a qualquer *cluster*. Assim, o DBSCAN possui robustez a *outliers*.

Desvantagens:

- A qualidade da clusterização feita pelo DBSCAN depende da métrica de distância utilizada para definir a vizinhança em torno de cada padrão.
- DBSCAN pode não clusterizar bem conjuntos de dados com diferenças expressivas nas densidades dos agrupamentos.
- Escolher apropriadamente o valor de ε pode não ser fácil se os dados e as escalas envolvidas não são bem compreendidos.

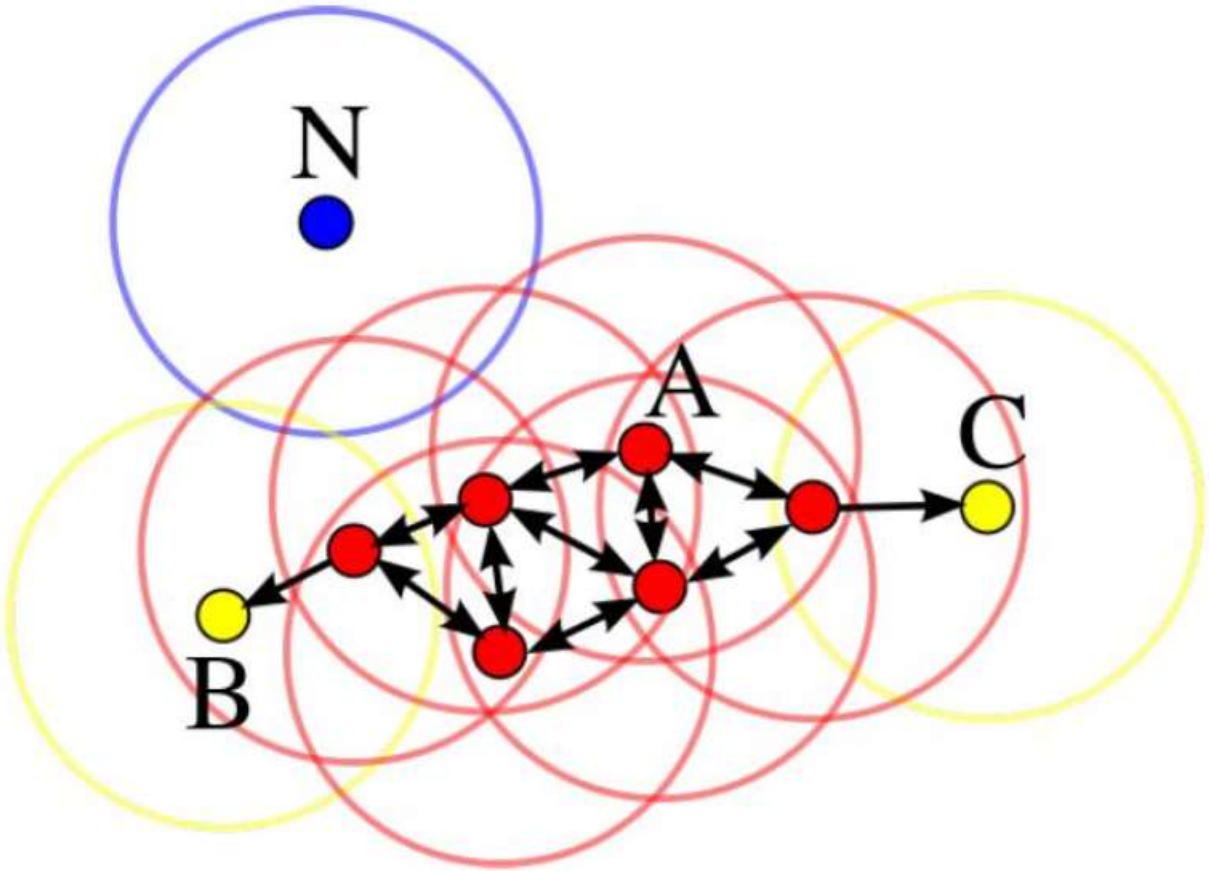
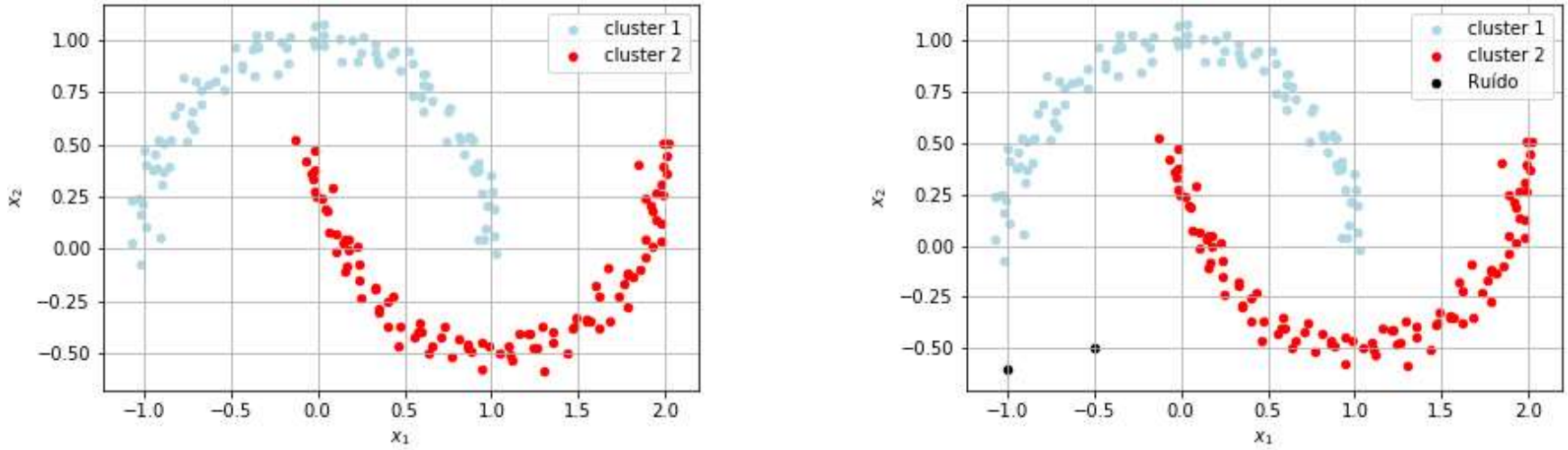


Figura. Neste diagrama, foi considerado o valor de $M_p = 4$. O ponto A e todos os demais pontos em vermelho representam pontos centrais, pois os círculos de raio ε em torno destes pontos englobam, no mínimo, M_p pontos. Uma vez que todos os pontos em vermelho são conectados por densidade, eles formam um único *cluster*. Os pontos B e C não são pontos centrais, mas são alcançáveis por A (diretamente ou através dos outros pontos centrais). Assim, B e C também pertencem a este *cluster*. Finalmente, o ponto N é considerado como ruído, pois não é um ponto central nem é alcançável por qualquer um dos pontos centrais.

Exemplo:



(a) Dados originais. (b) Dados com *outliers*.

Figura. Clusterização realizada pelo algoritmo DSBCAN considerando $M_p = 5$ e $\varepsilon = 0{,}2$.

7. Outras técnicas de clusterização

7.1. **Clusterização hierárquica: abordagens aglomerativas**

Considere um conjunto de dados $\{\mathbf{x}_i\}_{i=1}^{N}, \mathbf{x}_i \in \mathbb{R}^{K\times 1}$. No início do processo de clusterização aglomerativo, cada padrão é visto como um *cluster* por si só. Os *clusters* são, então, sequencialmente combinados em *clusters* maiores, até que todos os dados acabem pertencendo a um único *cluster*.

Em cada etapa, os dois *clusters* separados pela menor distância são combinados. A definição da distância entre *clusters*, denotada por $d(C_i; C_j)$, é o que diferencia os variados métodos de clusterização aglomerativa.

Algumas opções:

- **Ligação simples** (*single-linkage*): a distância entre dois *clusters* é determinada a partir de um único par de elementos, a saber, os dois elementos (um em cada *cluster*) que estão mais próximos entre si.

$$d(C_i; C_j) = \min\{d(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j\}$$

- **Ligação completa** (*complete-linkage*): a distância máxima entre os elementos de cada *cluster* define a distância entre os respectivos *clusters*.

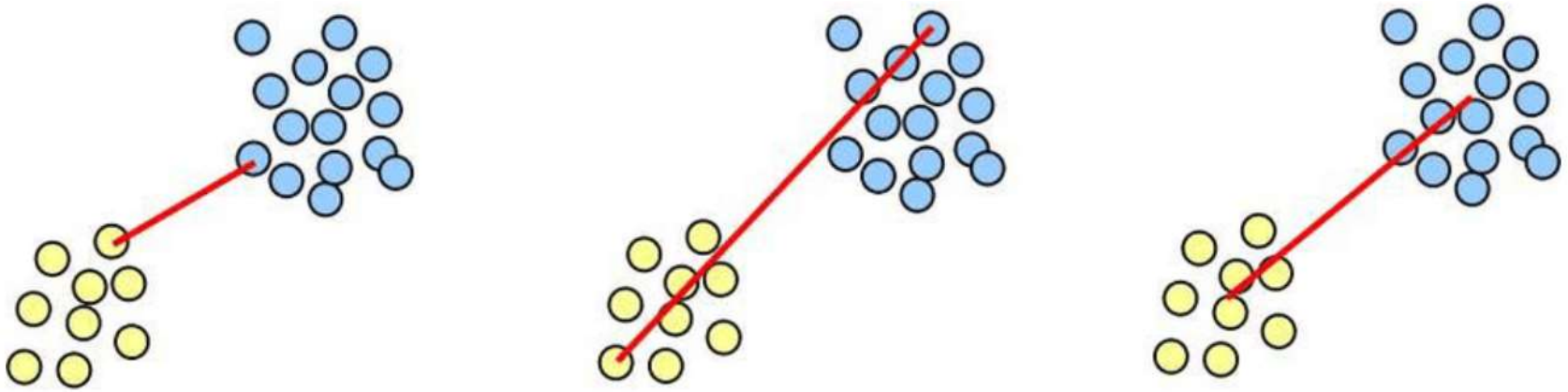
$$d(C_i; C_j) = \max\{d(\mathbf{x}_i, \mathbf{x}_j) : \mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j\}$$

- **Ligação média** (*average-linkage*): $d(C_i; C_j)$ corresponde à distância média entre os elementos de cada *cluster*.

$$d(C_i; C_j) = \frac{1}{n_{C_i}} \frac{1}{n_{C_j}} \sum_{\mathbf{x}_i \in C_i} \sum_{\mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$$

Os dois *clusters* que apresentarem a menor distância entre si serão fundidos em um único *cluster*.

O resultado da clusterização hierárquica pode ser visualizado na forma de um dendograma, o qual mostra a sequência de combinações de *clusters* e a distância envolvida em cada fusão.



(a) *single-linkage* (b) *complete-linkage* (c) *average-linkage*

Exemplo: dados de renda anual e de tendência de consumo referentes a 200 clientes.

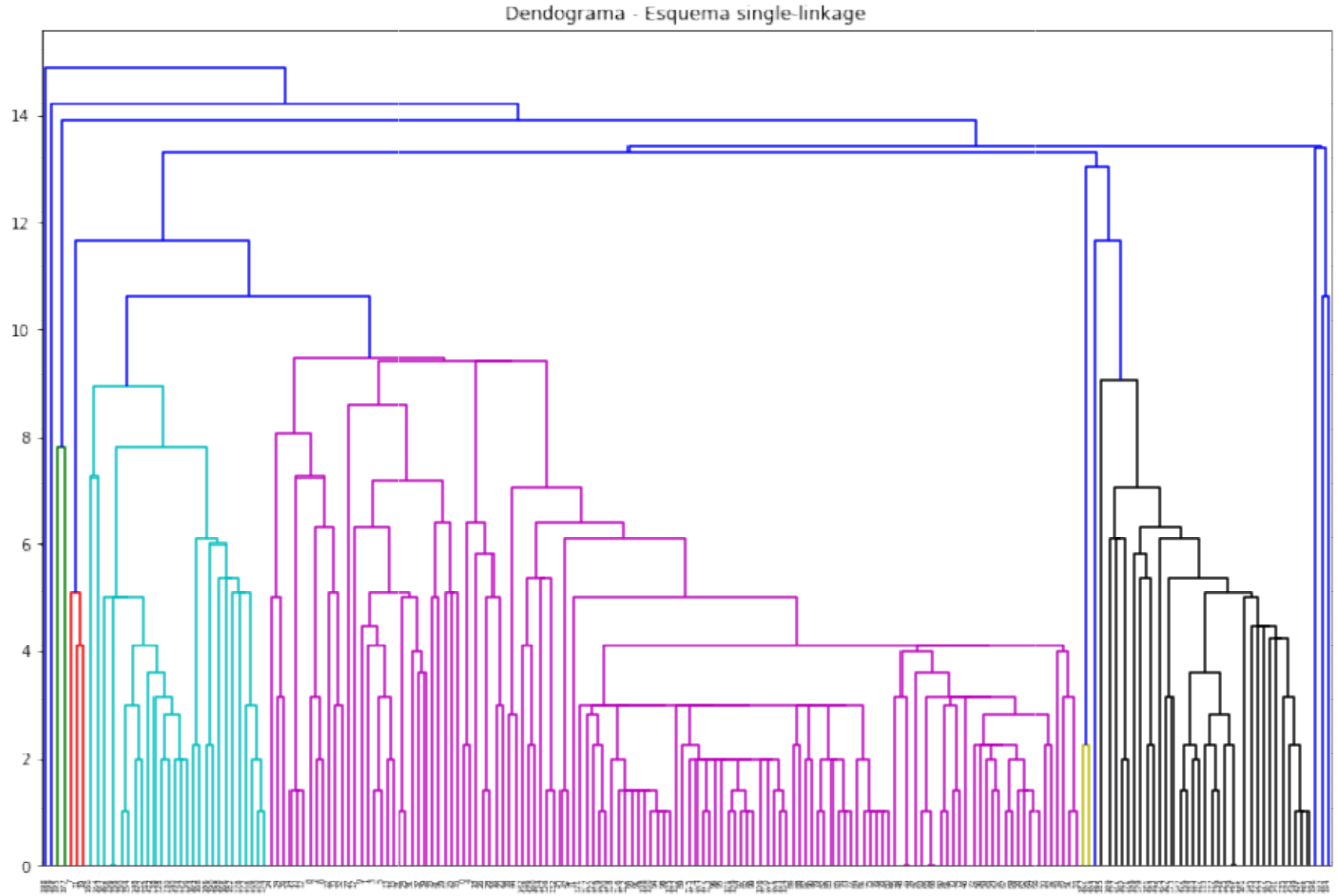


Figura. Dendograma associado à clusterização hierárquica aglomerativa com *single-linkage*.

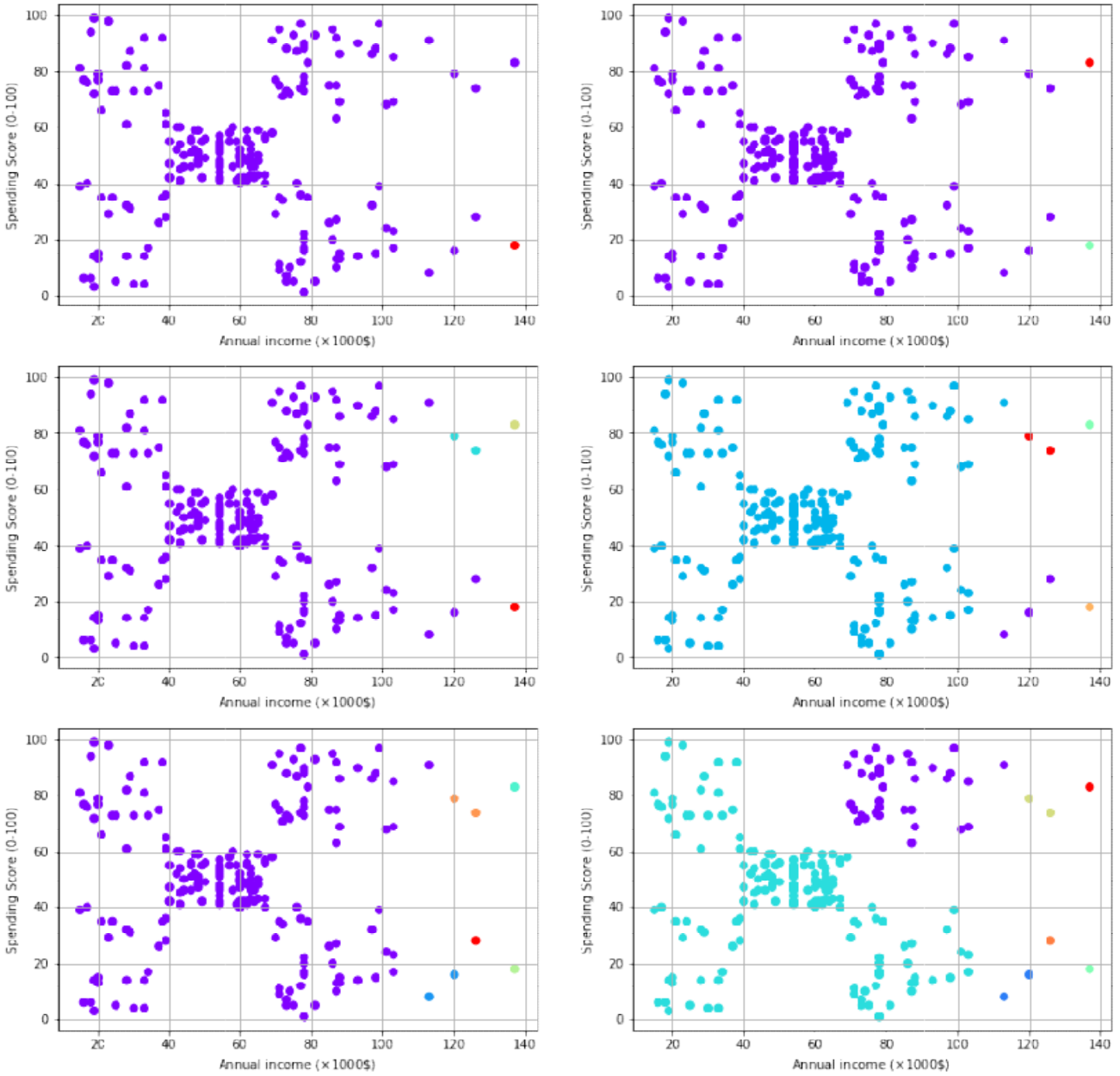


Figura. Visualização dos *clusters* formados pela abordagem aglomerativa com *single-linkage*.

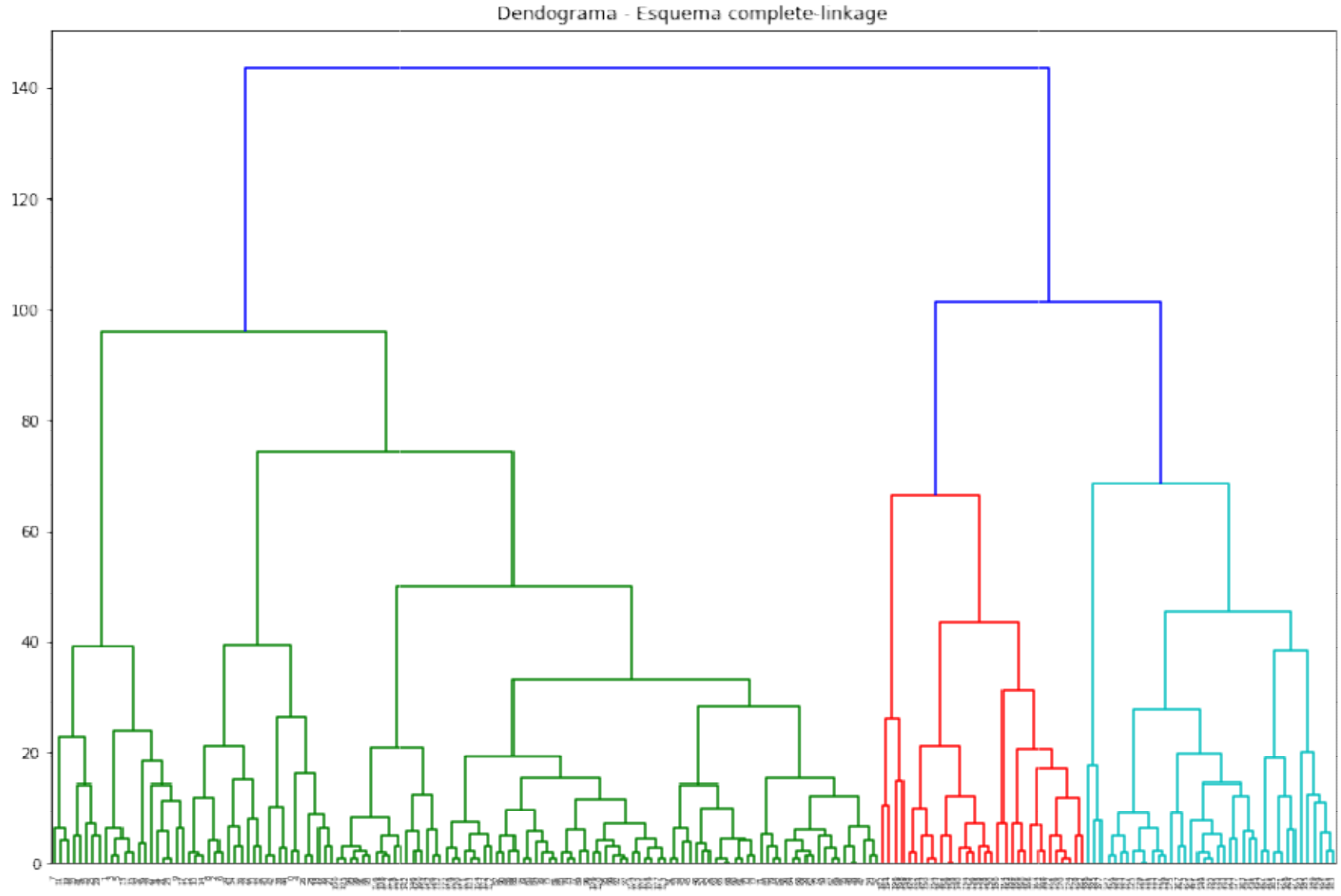


Figura. Dendograma associado à clusterização hierárquica aglomerativa com *complete-linkage*.

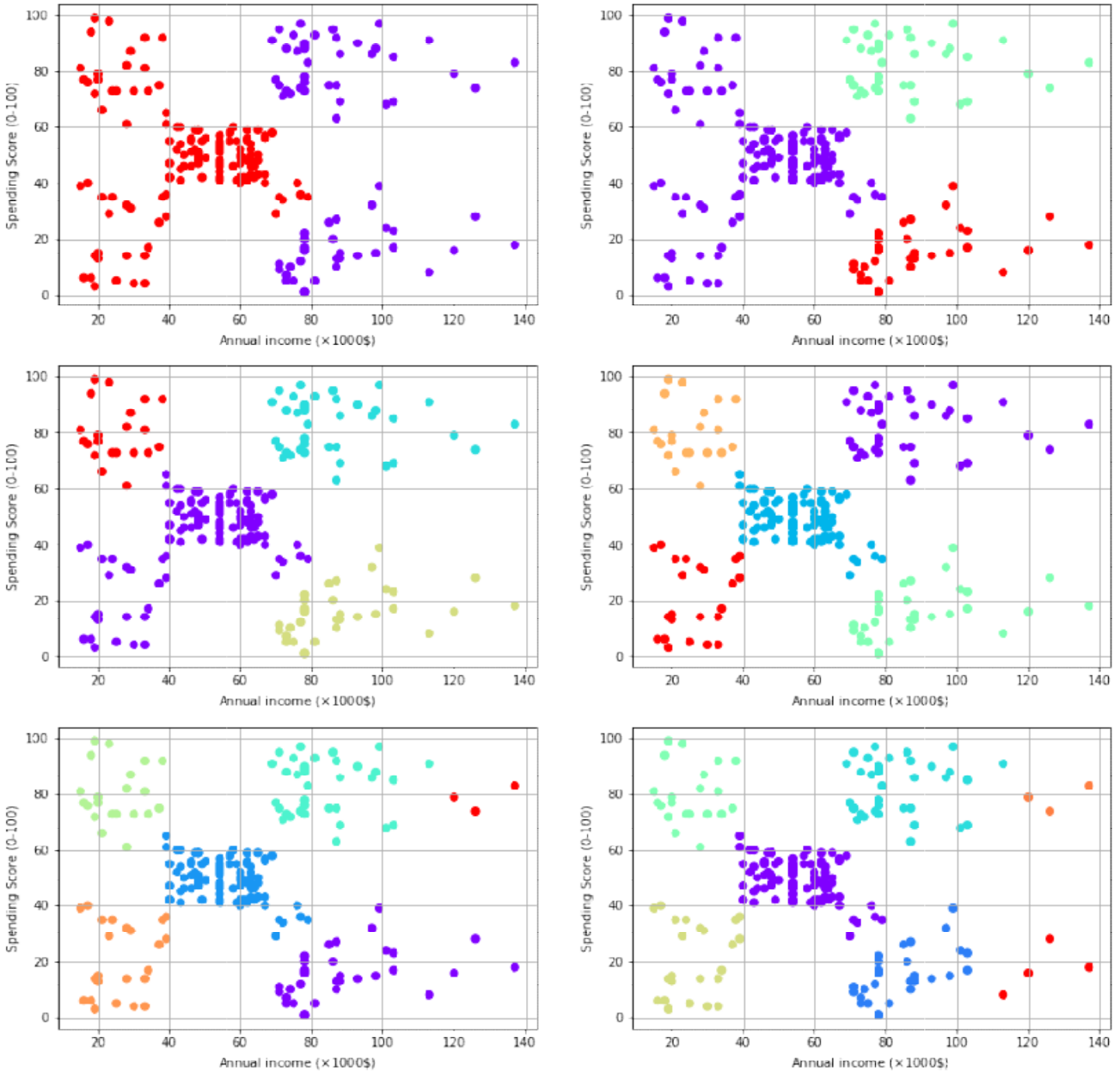


Figura. Visualização dos *clusters* formados pela abordagem aglomerativa com *complete-linkage*.

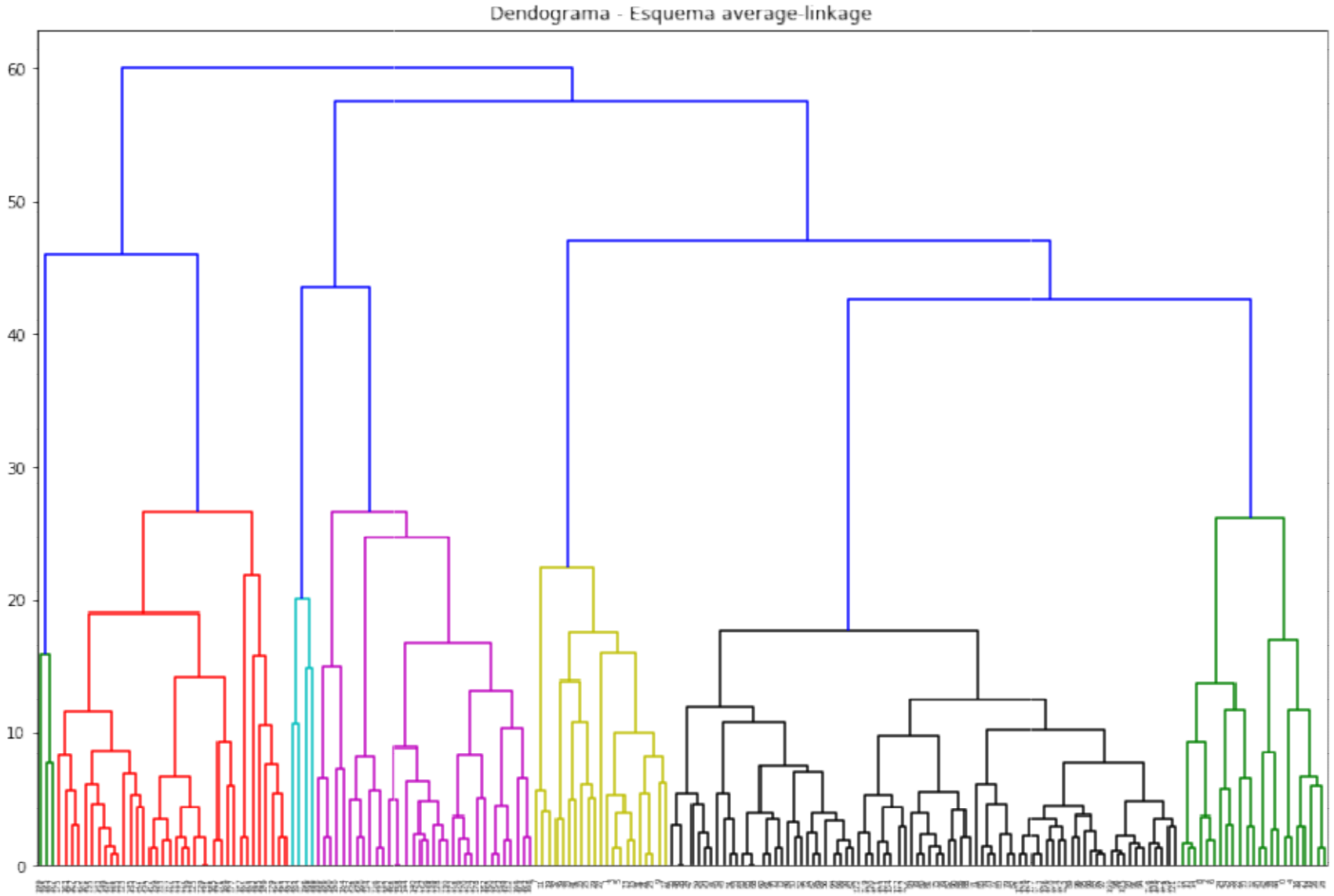


Figura. Dendograma associado à clusterização hierárquica aglomerativa com *average-linkage*.

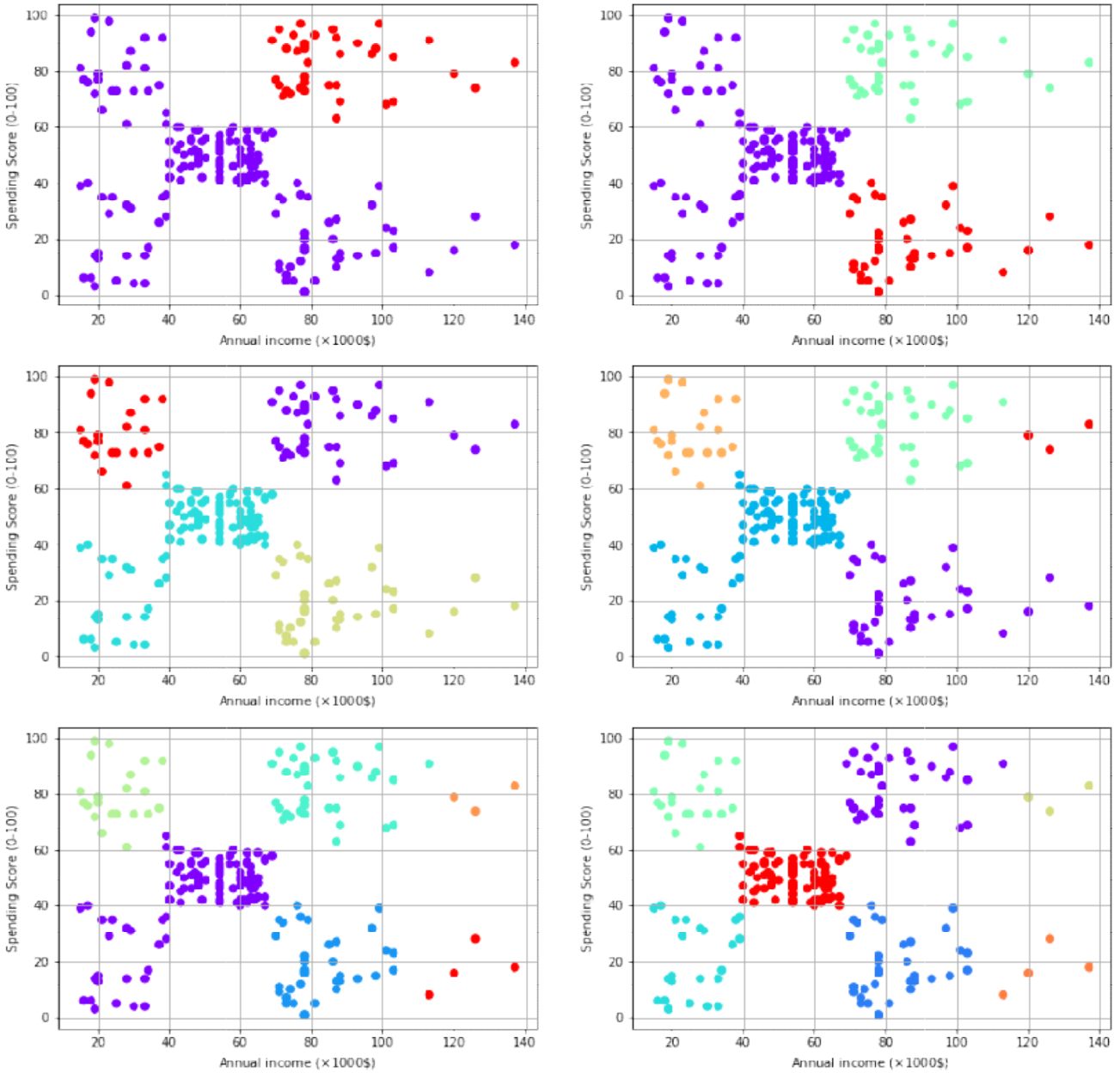
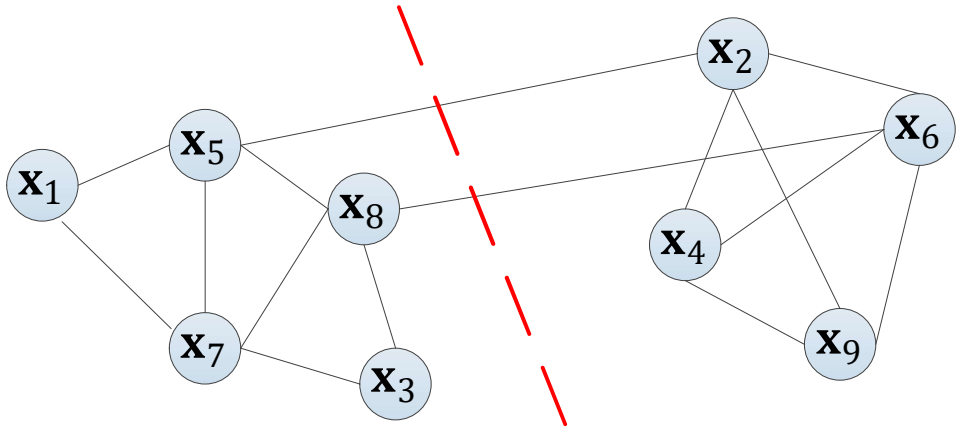


Figura. Visualização dos *clusters* formados pela abordagem aglomerativa com *average-linkage*.

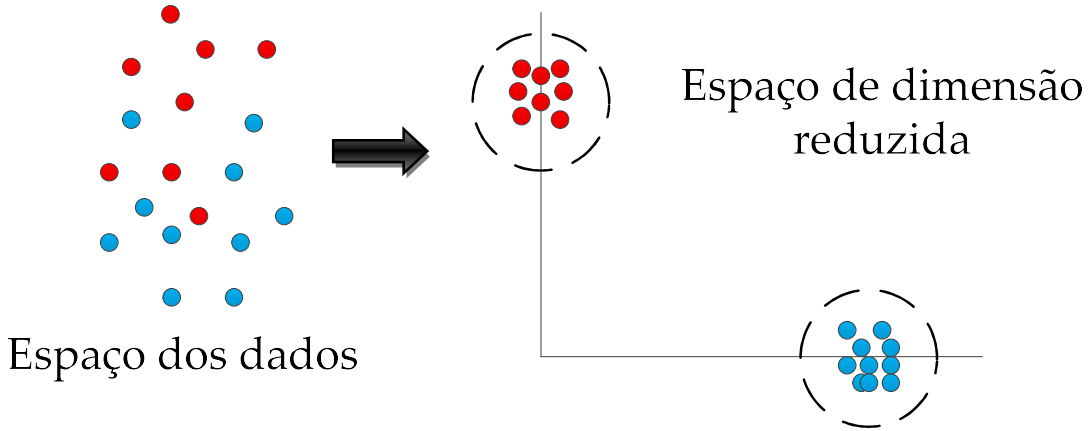
7.2. **Clusterização espectral**

As técnicas de clusterização espectral (VON LUXBURG, 2007) podem ser compreendidas a partir de dois pontos de vista distintos.

Do ponto de vista da **teoria de grafos**, primeiramente se constrói um grafo de similaridades entre os pontos $\mathbf{x}_1, \ldots, \mathbf{x}_N$ tendo como base as medidas de afinidade entre os dados, $a_{ij} = a(\mathbf{x}_i; \mathbf{x}_j)$, onde $a(\cdot)$ denota uma função de afinidade. Com isto, clusterizar os dados equivale a determinar pontos de corte no grafo.



Do ponto de vista de **redução de dimensionalidade**, os dados originais são projetados em um novo espaço, com dimensão reduzida, no qual a tarefa de clusterização é mais facilmente resolvida.



Interessantemente, os dois pontos de vista estão interligados: o espaço de dimensão reduzida é determinado pelos dados. Com efeito, a partir de informações espectrais (m autovetores) extraídas de uma matriz Laplaciana relacionada ao grafo de similaridade construído, os dados são mapeados em um espaço de dimensão reduzida, o qual tende a exibir agrupamentos mais facilmente distinguíveis (possivelmente radiais).

8. Mapas auto-organizáveis de Kohonen

Um mapa de Kohonen é um arranjo de neurônios, geralmente restrito a espaços de dimensão 1 ou 2, que procura estabelecer e preservar noções de vizinhança (preservação topológica).

Se estes mapas apresentarem propriedades de auto-organização, então eles podem ser aplicados a problemas de clusterização e ordenação espacial de dados.

Auto-organização: emergência de estrutura e organização em um sistema sem que isto seja imposto externamente.

Toda redução de dimensão (relativa à dimensão intrínseca dos dados) pode implicar na perda de informação (por exemplo, violação topológica). Sendo assim, este mapeamento deve ser tal que minimize a perda de informação.

Um mapa de Kohonen unidimensional é dado por uma sequência ordenada de neurônios lineares, sendo que o número de pesos de cada neurônio é igual ao número de entradas.

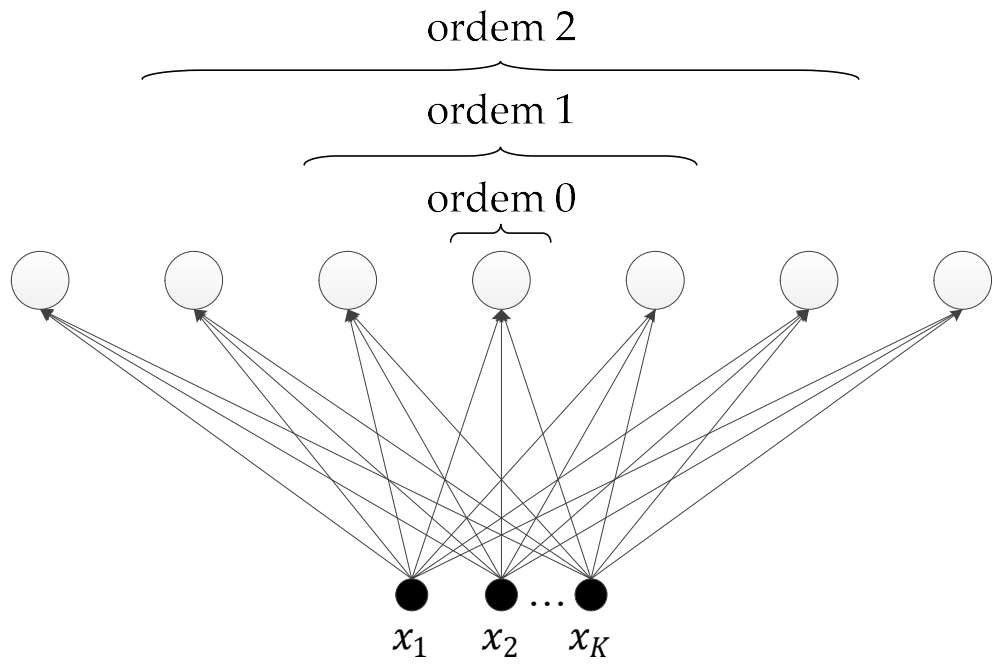


Figura. Rede de Kohonen em arranjo unidimensional: ênfase na vizinhança.

Há uma relação de vizinhança entre os neurônios (no espaço unidimensional vinculado ao arranjo), mas há também uma relação entre os pesos dos neurônios no

espaço de dimensão igual ao número de entradas. Para entender a funcionalidade dos mapas de Kohonen, é necessário considerar ambas as relações.

Exemplos bidimensionais:

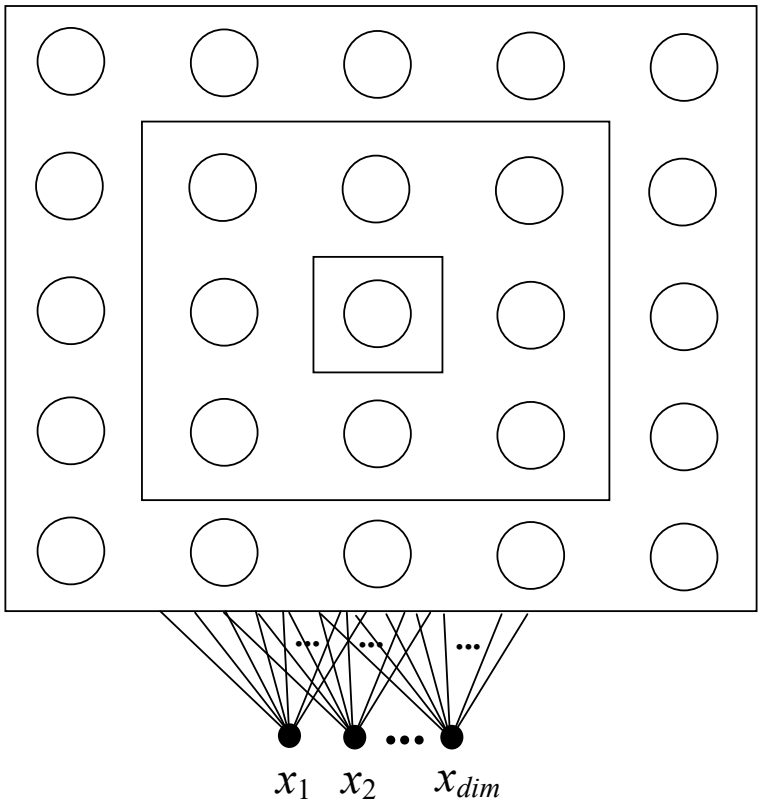


Figura. Rede de Kohonen em arranjo bidimensional: ênfase na vizinhança.

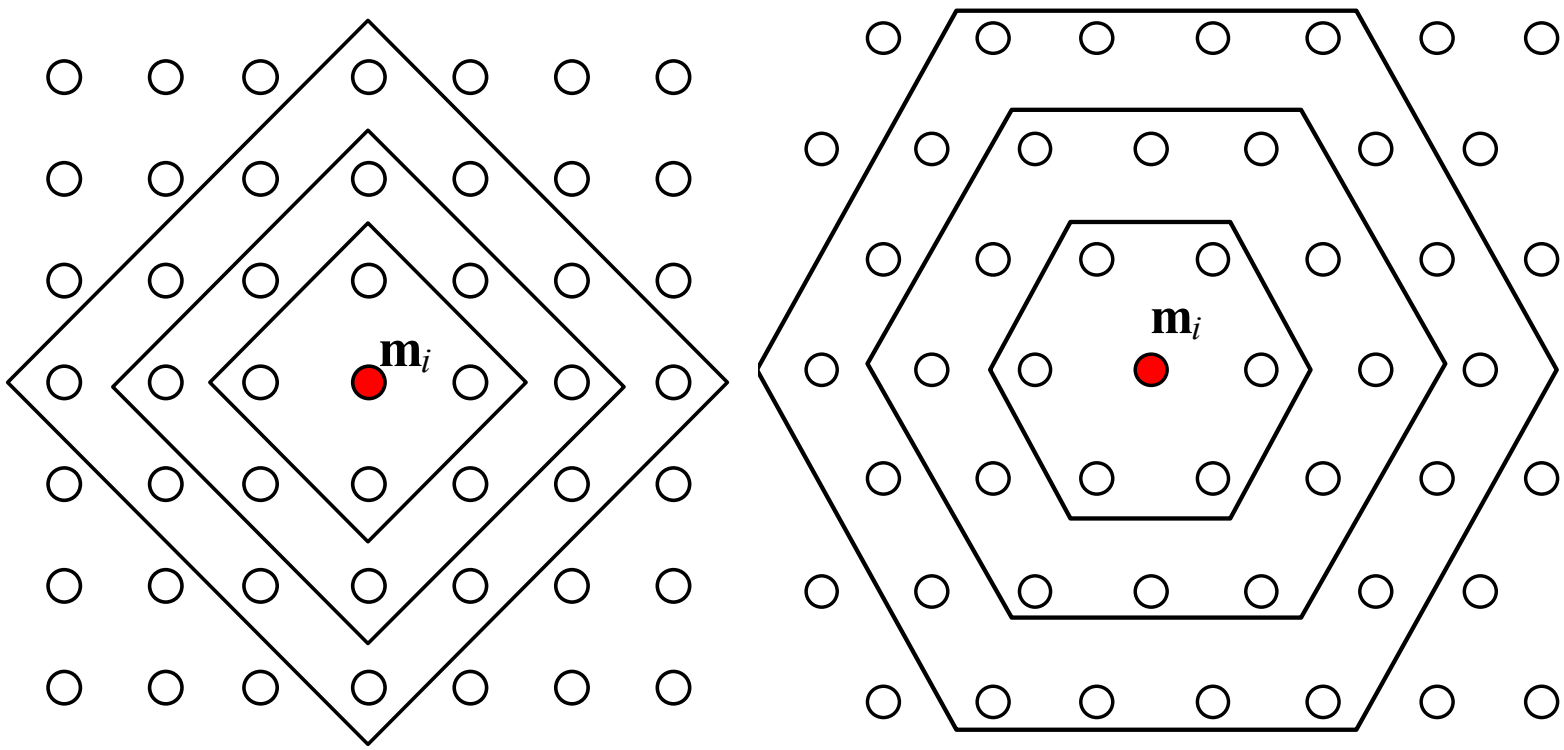


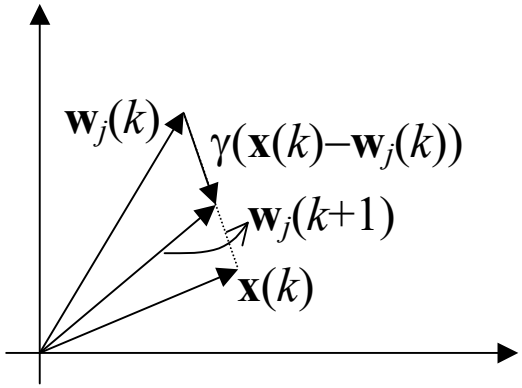
Figura. Outras configurações de mapas e de vizinhança (figuras extraídas de ZUCHINI, 2003).

O mapa de Kohonen é construído explorando a ideia de **aprendizado competitivo**: para cada padrão de entrada $\mathbf{x}(i) \in \mathbb{R}^K$, o neurônio cujo vetor de pesos apresenta a menor distância Euclidiana com respeito a $\mathbf{x}(i)$ é considerado o vencedor (e é chamado de *best matching unit*, BMU). Então, o seu vetor de pesos é ajustado na

direção daquele padrão, de maneira que ele se especialize ainda mais em responder ativamente para padrões similares a $\mathbf{x}(i)$.

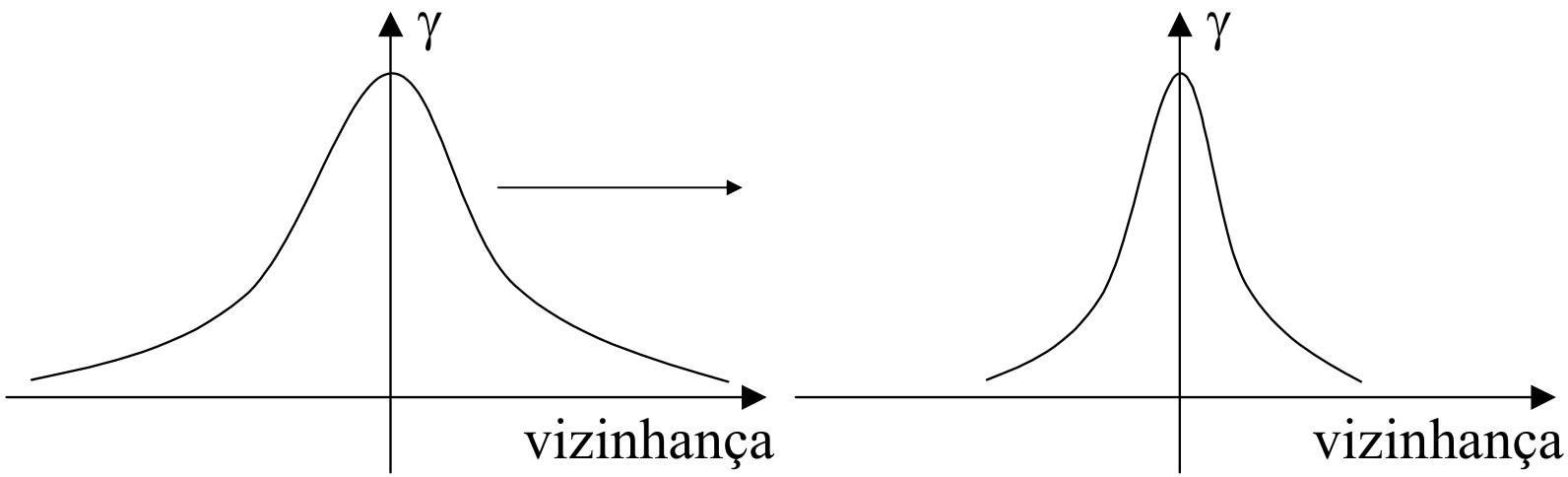
Seja j o neurônio vencedor. Então, uma opção para o processo de aprendizado não-supervisionado consiste em ajustar somente os pesos deste neurônio na forma:

$$\mathbf{w}_j(k+1) = \mathbf{w}_j(k) + \gamma\left(\mathbf{x}(k) - \mathbf{w}_j(k)\right)$$



Caso existam múltiplos representantes para cada agrupamento de dados, então é interessante ajustar o neurônio vencedor juntamente com seus vizinhos mais próximos.

Implementação



É importante que a influência de cada neurônio vencedor seja ampla no início do processo e sofra uma redução continuada com o decorrer das iterações.

8.1. **Algoritmo de ajuste dos pesos**

Algoritmo de ajuste do mapa auto-organizável de Kohonen

Enquanto uma condição de parada não é atingida:

1. Ordene aleatoriamente os N padrões de entrada.
2. Para $k = 1$ até N, faça:
   a. $j = \arg\min_J \left\| \mathbf{x}(k) - \mathbf{w}_J \right\|$
   b. $\forall\, J \in$ vizinhança(j), faça:
      i. $\mathbf{w}_J = \mathbf{w}_J + \gamma(d(j;J))\left(\mathbf{x}(k) - \mathbf{w}_J\right)$
3. Atualize a taxa de aprendizagem γ.
4. Atualize a vizinhança.

8.2. **Ajuste de pesos com restrição de vizinhança**

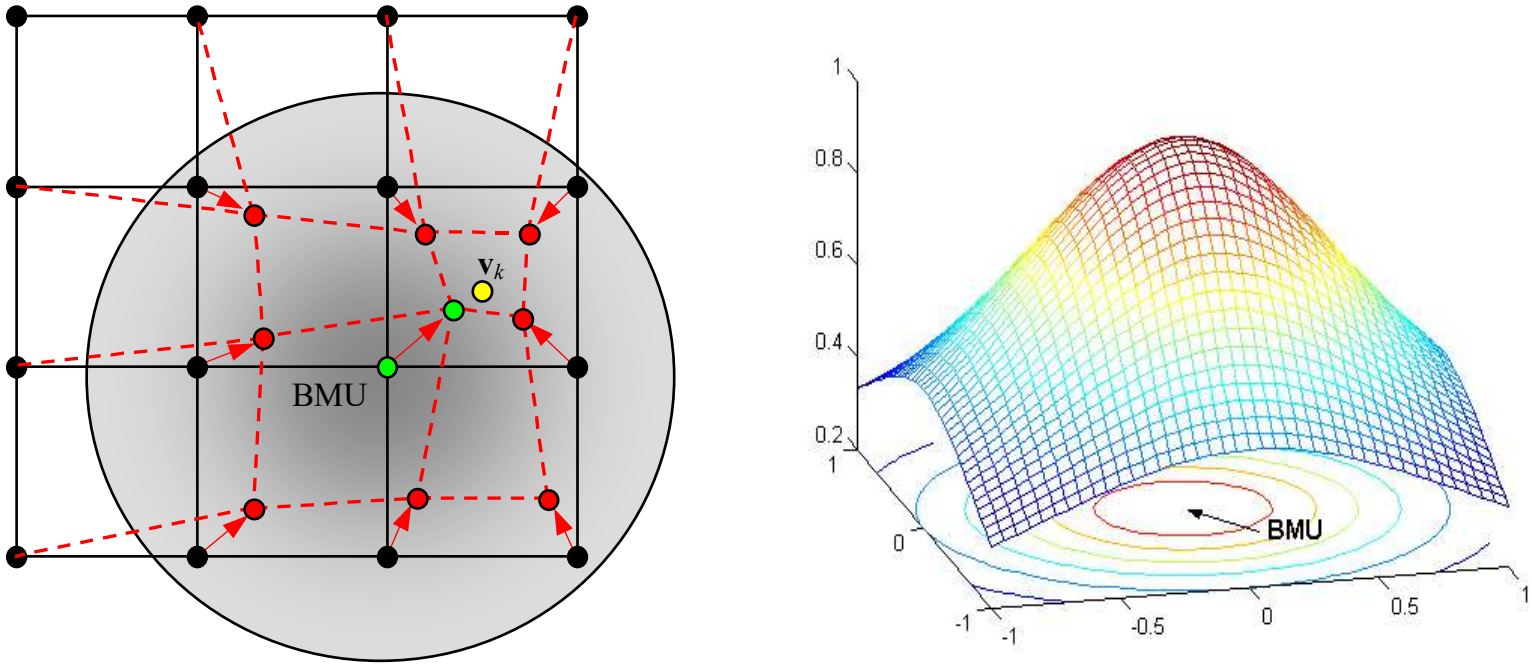


Figura. BMU (*Best Matching Unit*) e seus vizinhos (figuras extraídas de ZUCHINI, 2003).

- O neurônio que venceu para uma dada amostra é o que sofre o maior ajuste. No entanto, dentro de uma vizinhança, todos os neurônios vizinhos também sofrerão um ajuste de pesos, embora de menor intensidade.

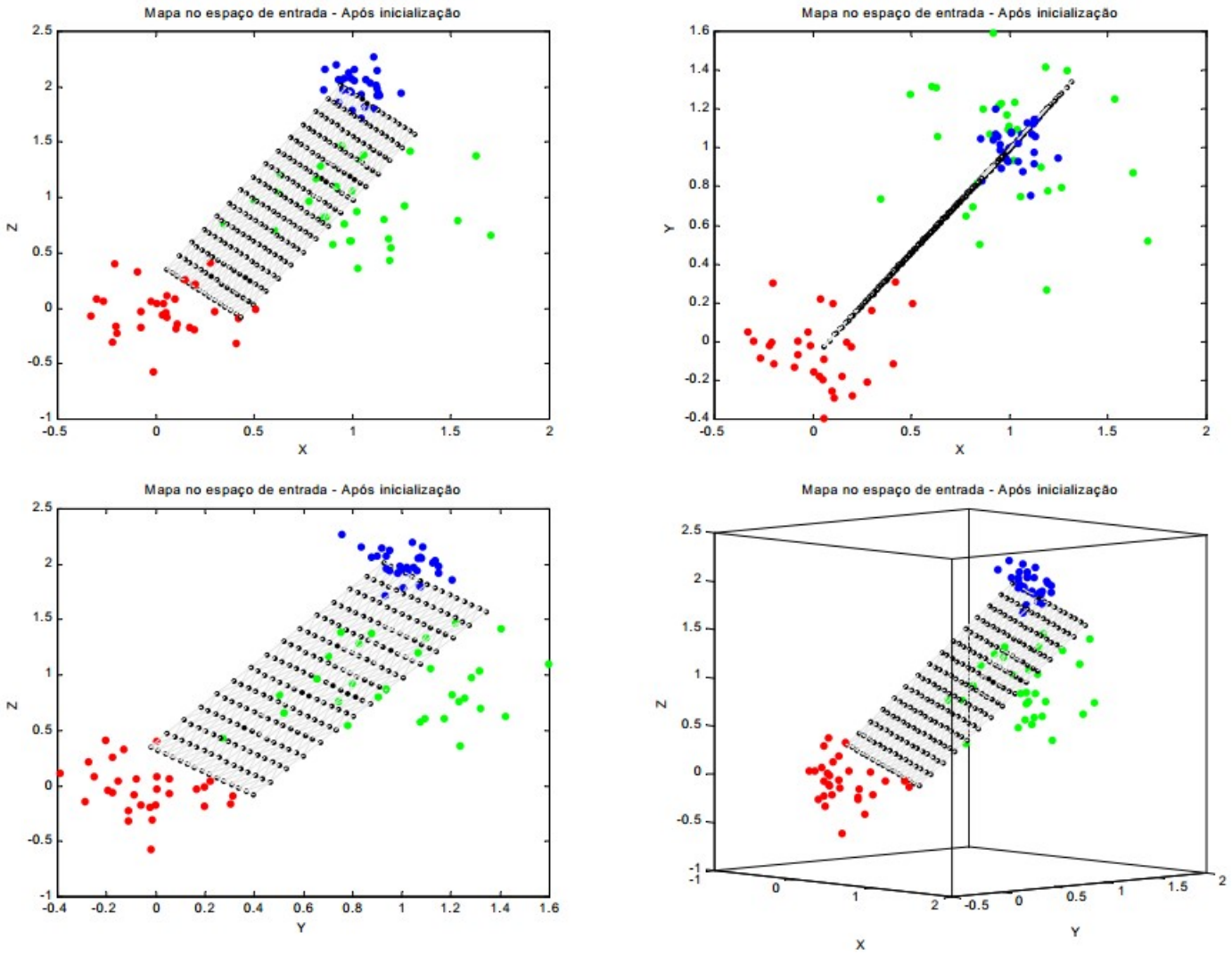


Figura. Inicialização linear dos vetores de pesos sobre um conjunto de dados artificiais.

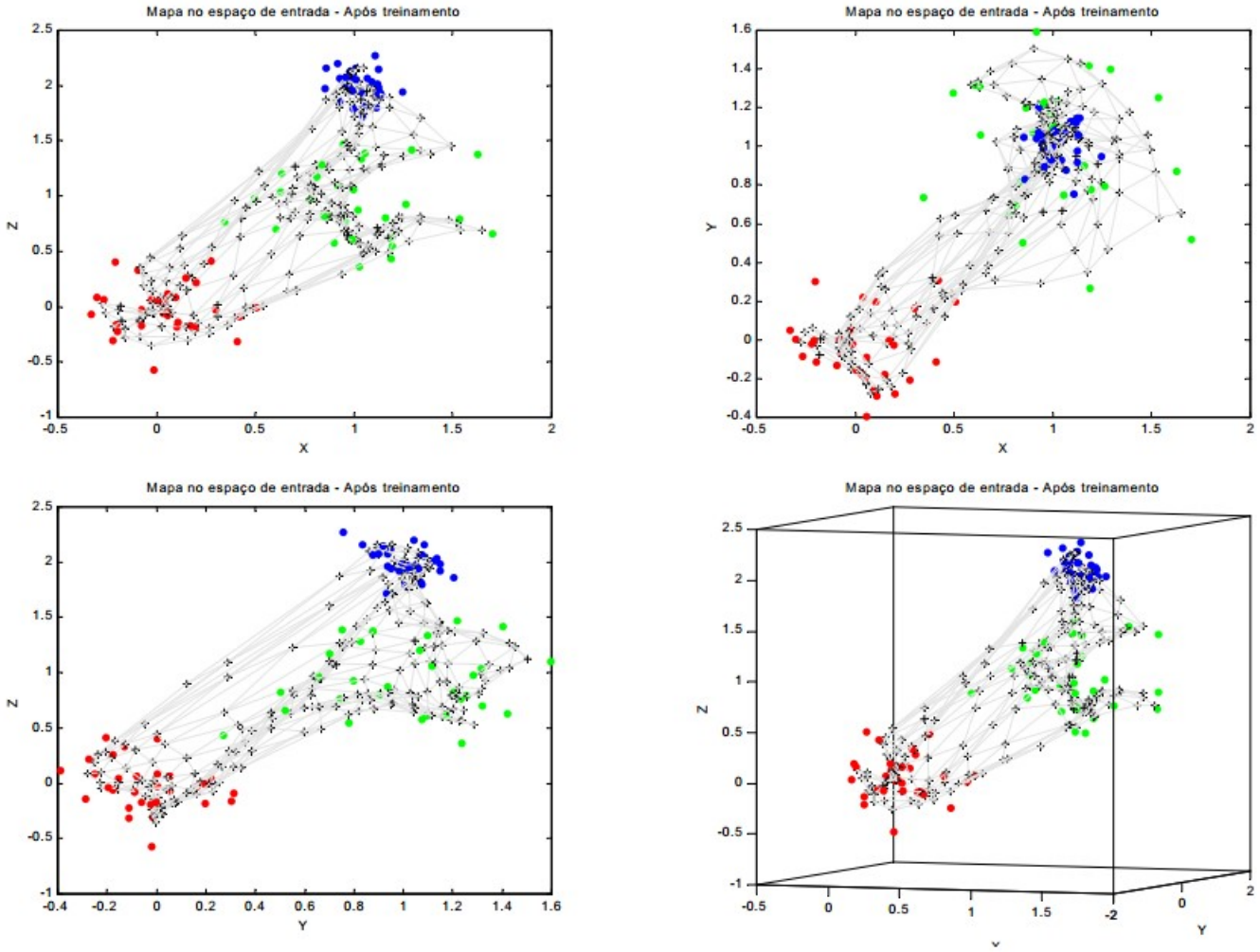


Figura. Conformação do mapa após o processo de auto-organização.

8.3. **Discriminação dos agrupamentos**

Dada a conformação final dos neurônios (não rotulados), como realizar a clusterização, ou seja, definição de agrupamentos e atribuição do mesmo rótulo a todos os neurônios pertencentes a um dado agrupamento?

Solução: matriz(vetor)-U (ULTSCH, 1993; COSTA, 1999).

Aspecto a ser explorado: após o processo de auto-organização, dados de entrada com características semelhantes passam a promover reações semelhantes da rede neural. Assim, comparando-se as reações da rede neural treinada, é possível agrupar os dados pela análise do efeito produzido pela apresentação de cada um à rede.

A matriz-U é uma ferramenta que permite realizar a discriminação dos agrupamentos, a partir de uma medida do grau de similaridade entre os pesos de

neurônios adjacentes na rede. O perfil apresentado pelas distâncias relativas entre neurônios vizinhos representa uma forma de visualização de agrupamentos.

Topologicamente, as distâncias entre neurônios vizinhos refletem os agrupamentos, pois uma "depressão" ou um "vale" da superfície de relevo representa neurônios pertencentes a um mesmo agrupamento. Neurônios que têm uma distância grande em relação ao neurônio adjacente, a qual é representada por um pico da superfície de relevo, são neurônios discriminantes de agrupamentos.

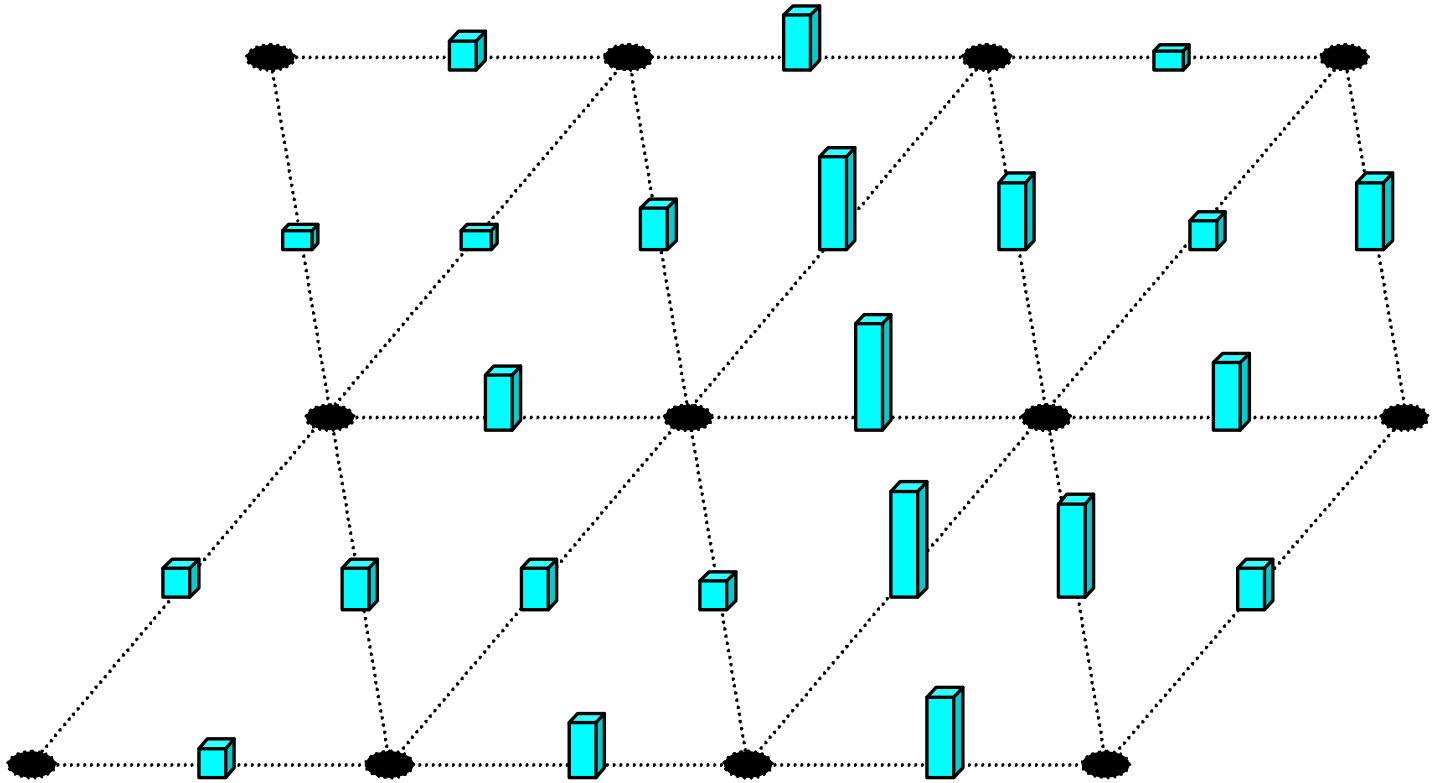


Figura. Exemplo de matriz-U para arranjo hexagonal (figura extraída de ZUCHINI, 2003).

8.4. **Mapa conceitual**

Animal		Dove	Hen	Duck	Goose	Owl	Hawk	Eagle	Fox	Dog	Wolf	Cat	Tiger	Lion	Horse	Zebra	Cow
is	small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	feathers	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
likes to	hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	fly	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
	swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Figura. Conjunto de *features* associadas a cada espécie de animal presente no conjunto de dados Extraída de (HAYKIN, 2008).

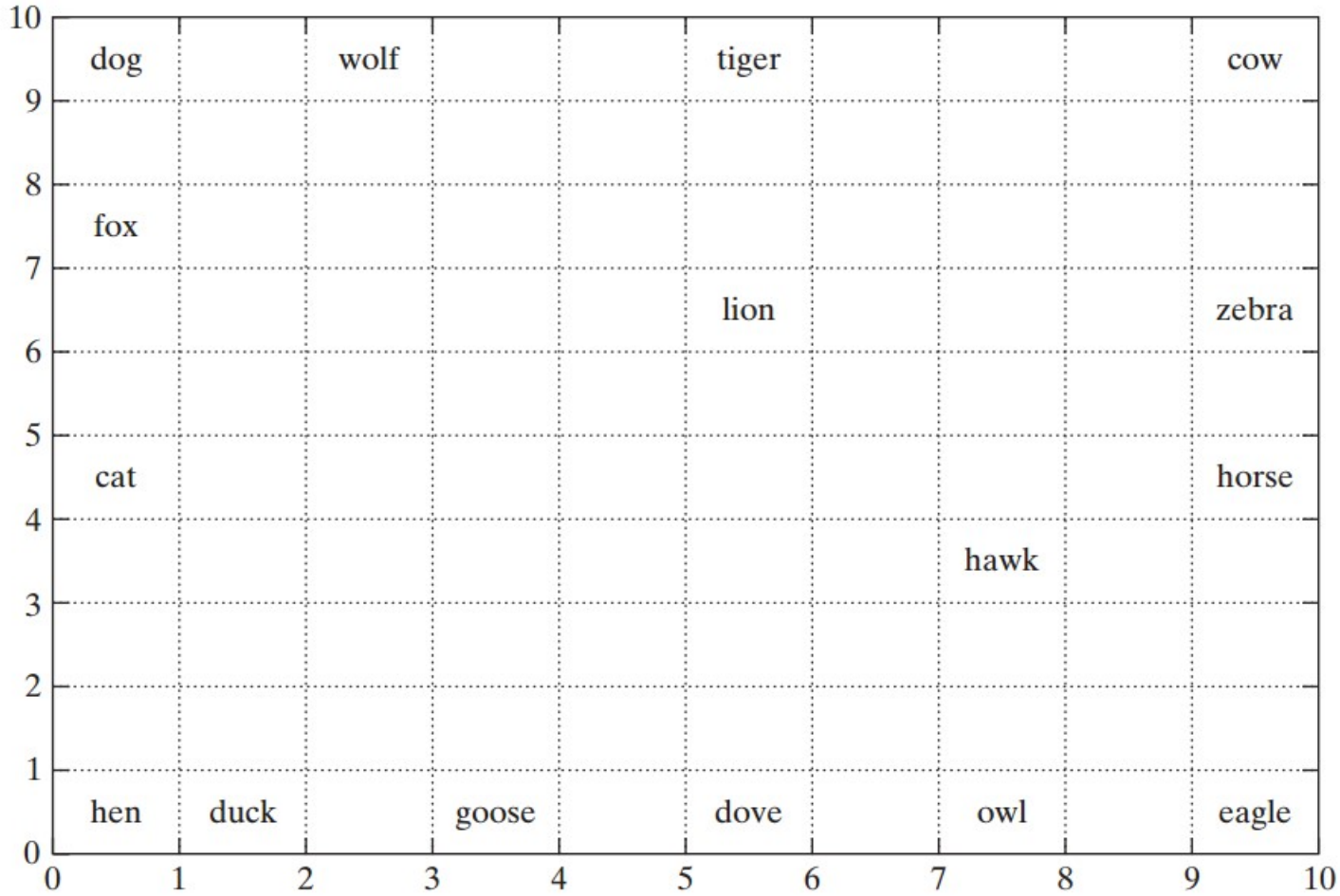


Figura. Mapa auto-organizável 10 x 10: os neurônios que responderam mais ativamente aos padrões de entrada (*features*) receberam o respectivo rótulo. Figura extraída de (HAYKIN, 2008).

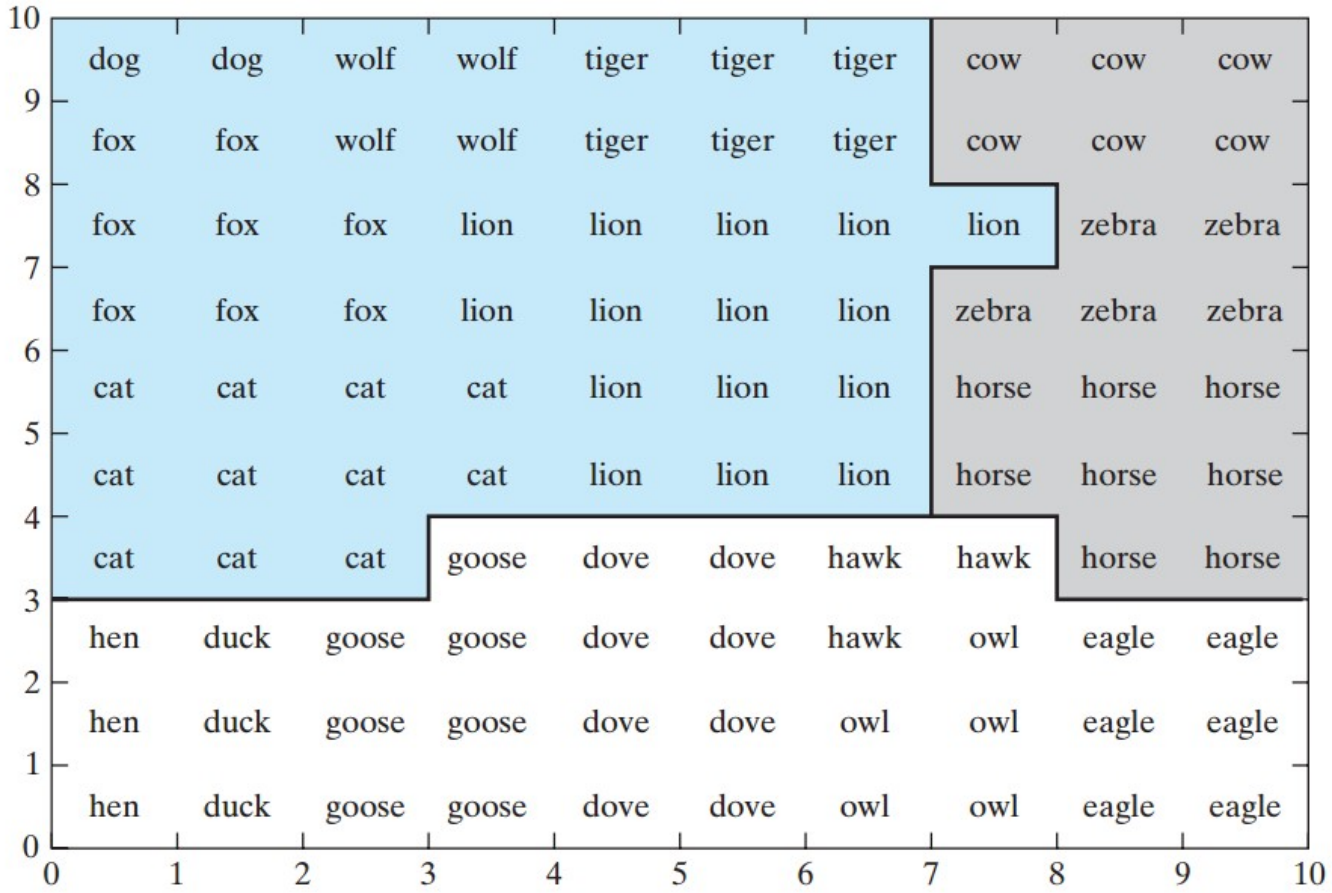


Figura. Cada neurônio foi marcado com o rótulo associado ao padrão de entrada (consequentemente, ao animal) para o qual ele é mais fortemente ativado. Percebe-se claramente que o mapa conseguiu capturar relações intrínsecas às próprias espécies – 3 *clusters*: em branco, temos os pássaros; em cinza, as espécies "pacíficas"; em azul, os predadores. Extraída de (HAYKIN, 2008).

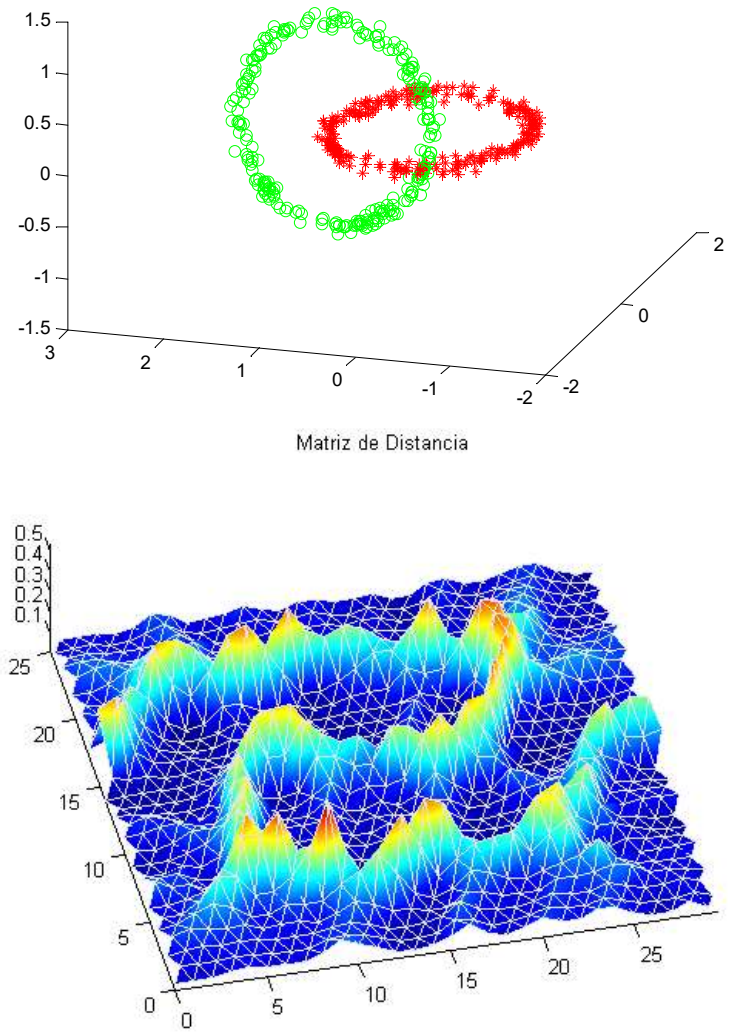


Figura. Matriz-U para grid hexagonal. Extraída de (ZUCHINI, 2003).

9. Referências bibliográficas

AKAIKE, H. A. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716-723, 1974.

ALPAYDIN, E. **Introduction to Machine Learning**. MIT Press. 3rd edition. 2014.

BEZDEK, J. C. **Pattern Recognition with Fuzzy Objective Function Algorithms**. Springer. 1981.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. Springer, 2006.

DAVIES, D. L.; BOULDIN, D. W. A Cluster Separation Measure. I*EEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 224-227, 1979.

DESGRAUPES, B. Clustering Indices. *Technical Report*, University Paris Ouest, Lab Modal'X, 2017.

DUDA, R. O., HART, P. E., STORK, D. G. **Pattern Classification**. John Wiley & Sons. 2nd edition, 2001.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, AAAI Press, pp. 226-231, 1996.

HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference and Prediction**. Springer. 2nd edition, 2006.

HAYKIN, S. **Neural Networks and Learning Machines**. Prentice Hall, 3rd edition, 2008.

JAIN, A.K., MURTY, M.N. & FLYNN, P.J. Data Clustering: A Review, *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.

KOHONEN, T. Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, vol. 43, pp. 59-69, 1982.

KOHONEN, T. **Self-Organization and Associative Memory**, Springer, 3rd edition, 1989 (1st. edition, 1984).

KOHONEN, T. The Self-Organizing Map, *Proceedings of the IEEE*, 78:1464-1480, 1990.

KOHONEN, T. **Self-Organizing Maps**, Springer, 3rd edition, 2001.

LUENBERGER, D.G. **Linear and Nonlinear Programming**. 2nd edition, Addison-Wesley Publishing Company, 1984.

LLOYD, S. P. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129-137, 1982.

MACQUEEN, J. Some Methods for Classification and Analysis of Multivariate Observations. In L. M. LeCam e J. Neyman (Eds.), *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Volume I, pp. 281-297, University of California Press, 1967.

ROUSSEEUW, P.J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53-65, 1987.

SCHWARZ, G. et. al. Estimating the Dimension of a Model. *The Annals of Statistics, Institute of Mathematical Statistics*, vol. 6, no. 2, pp. 461-464, 1978.

VON LUXBURG, U. A Tutorial on Spectral Clustering, *Statistics and Computing*, vol. 17, no. 4, pp.395-416, 2007.

ZUCHINI, M.H. "Aplicações de Mapas Auto-Organizáveis em Mineração de Dados e Recuperação de Informação", *Tese de Mestrado*, Faculdade de Engenharia Elétrica e de Computação (FEEC/Unicamp), Setembro 2003.