

# Clusterização

Profa. Dra. Sarajane Marques Peres

Prof. Dr. Clodoaldo A. M. Lima

Escola de Artes, Ciências e Humanidades

<http://each.uspnet.usp.br/sarajane>



# Clusterização

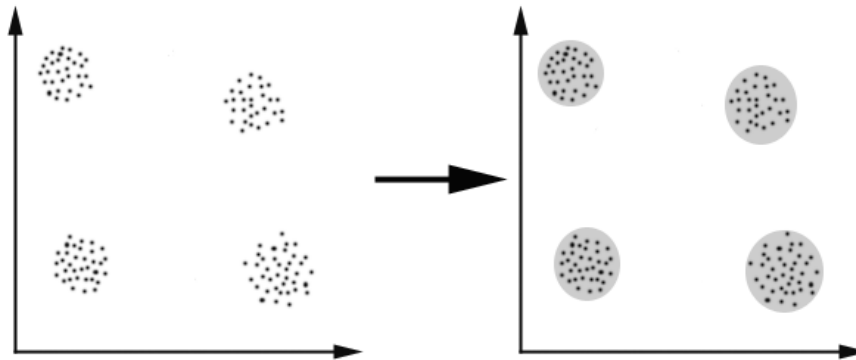
---

- ▶ A tarefa de Clusterização, também chamada de Agrupamento, é usada para **particionar os registros** de uma base de dados em subconjuntos ou clusters, de tal forma que elementos de um clusters compartilhem um conjunto de propriedades comuns que os distingam dos elementos de outros clusters.
  - ▶ Objetivo: maximizar **similaridade intracluster** e minimizar **similaridade intercluster**.
  - ▶ Também chamada de indução não supervisionada (pois a informação de classe não existe).
  - ▶ Ela auxilia o usuário (de Data Mining) a realizar agrupamentos naturais de registros.
- ▶ Envolve a **organização de um conjunto de padrões** (usualmente representados na forma de vetores de atributos ou pontos em um espaço multidimensional – espaço de atributos) em clusters, **de acordo com alguma medida de similaridade**.
  - ▶ Intuitivamente, padrões pertencentes a um dado cluster devem ser mais similares entre si (compartilham um conjunto de propriedades comuns) do que em relação a padrões pertencentes a outros clusters.



# Clusterização

---

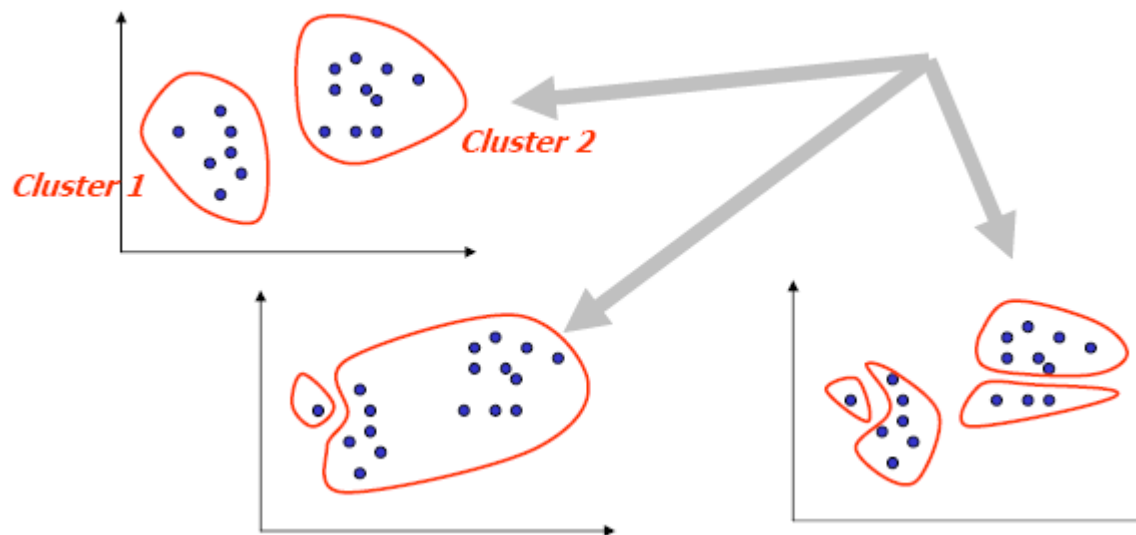


- Encontrar K cluster (ou uma classificação que consiste de K cluster) tal que os objetos de um cluster são similares entre si ao passo que os objetos de cluster diferentes são dissimilares (Bacher, 1996)
  - Clusterização **pode ser considerado como o problema de aprendizado não supervisionado mais importante**;
  - Visa encontrar uma estrutura em um coleção de dados não rotulados.
  - Uma definição vaga de clusterização poderia ser "o processo de organização de objetos em grupos cujos membros são similares de alguma forma".
- 



# Clusterização

- Agrupar os pontos em um conjunto de dados não rotulados a fim de **descobrir estruturas nos dados** (aprendizado não supervisionado)
- Nós gostaríamos que os indivíduos no mesmo grupo fossem similares entre si e dissimilares para indivíduos de outros grupos
- **Freqüentemente dissimilaridade = distância**
- Se os grupos são compactos e bem separados, então eles podem ser rotulados e suas características interpretadas



# O objetivo de Clusterização

---

- Determinar o agrupamento intrínseco em um conjunto de dados não rotulados
  - Em que consiste uma boa clusterização?
  - Pode ser mostrado que não há critério absoluto melhor que seja independente do objetivo da clusterização.
  - Conseqüentemente, o usuário deve fornecer este critério, de forma que o resultados da clusterização irá satisfazer suas necessidades.
  - Todos os algoritmos de clusterização produzem clusters, embora os dados possa não possuir.
  - Não há um critério excelente, depende do objetivo
    - Redução dos dados - encontrar representantes para grupos homogêneos
    - “Clusters natural” - descreve suas propriedades naturais (tipos dados naturais)
    - “Clusters úteis” - encontrar clusters úteis e convenientes (classe de dados úteis)
    - Detecção de outliers - encontrar objetos de dados não usuais
- 



# Clusterização

---

- ▶ Na **maioria dos casos**, o processo de clusterização **requer que o usuário determine qual é o número de grupos a ser considerado**.
- ▶ Uma vez tendo os grupos é possível fazer uma análise dos elementos que compõem cada um deles, identificando as características comuns aos seus elementos e, desta forma, podendo criar um rótulo que represente cada grupo.
- ▶ **A presença de dados distribuídos em um espaço de grande dimensionalidade (muitos atributos) dificulta a detecção de clusters.**
  - ▶ Os clusters podem estar imersos em algum subespaço do espaço de dados original.



# Clusterização

---



(a) Original



(b) Dois grupos



(c) Quatro grupos



(d) Seis grupos

Diferentes grupos formados sobre um mesmo conjunto de dados.  
(KUMAR; TAN; STEINBACH, 2005)



# Clusterização

---

## ► Formalizando ...

No processo de clusterização supõe-se a existência de  $n$  dados  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$  tais que cada ponto pertença a um espaço  $d$  dimensional  $\mathbf{R}^d$ . A tarefa de clusterização desses dados, separando-os em  $k$  clusters consiste em encontrar  $k$  pontos  $\mathbf{m}_j$  em  $\mathbf{R}^d$  de tal forma que a expressão:

$$\frac{\sum_i \min_j d^2(\mathbf{x}_i, \mathbf{m}_j)}{n}$$

seja minimizada, onde  $d^2(\mathbf{x}_i, \mathbf{m}_j)$  denota uma distância entre  $\mathbf{x}_i$  e  $\mathbf{m}_j$ . Os pontos  $\mathbf{m}_j$  são denominados centróides ou médias dos clusters.



# Clusterização – Estruturas de Dados

---

## Matriz de Dados

$X =$

$x_{11}$	$x_{12}$	$x_{13}$	...	$x_{1p}$
$x_{21}$	$x_{22}$	$x_{23}$	...	$x_{2p}$
$x_{31}$	$x_{32}$	$x_{33}$	...	$x_{3p}$
...	...	...	...	...
$x_{n1}$	$x_{n2}$	$x_{n3}$	...	$x_{np}$

## Matriz de Similaridade

$D =$

0				
$d(1,2)$	0			
$d(3,1)$	$d(3,2)$	0		
...	...	...	...	
$d(n,1)$	$d(n,2)$	$d(n,3)$	...	0



# Clusterização

---

- ▶ Requisitos desejáveis por algoritmos de clusterização:
  - ▶ Descoberta de clusters com formas arbitrárias
  - ▶ Identificar clusters de tamanhos variados
  - ▶ Aceitar diversos tipos de variáveis possíveis
  - ▶ Ser insensível à ordem de apresentação dos objetos (dados)
  - ▶ Trabalhar com objetos com qualquer número de atributos
  - ▶ Ser escalável para lidar com qualquer quantidade de objetos
  - ▶ Fornecer resultados interpretáveis e utilizáveis
  - ▶ Ser robusto na presença de ruídos
  - ▶ Exigir o mínimo de conhecimento para determinar os parâmetros de entrada
  - ▶ Aceitar restrições
  - ▶ Encontrar o número adequado de clusters



# Categorização dos métodos de agrupamento

(Han & Kamber, 2006)

---

- ▶ Métodos de particionamento
- ▶ Métodos hierárquicos
- ▶ Métodos baseados em densidade
- ▶ Métodos baseados em gride
- ▶ Métodos baseados em modelos
- ▶ Agrupamento de dados de alta-dimensão
- ▶ Agrupamento baseado em restrições



# Categorização dos métodos de agrupamento

---

- ▶ Método por particionamento: dado um conjunto de dados com  $n$  instâncias, um método por particionamento constrói  $k$  partições dos dados, onde cada partição representa um grupo e  $k \leq n$ . O método cria uma partição inicial e, então usa uma técnica de realocação iterativa que tenta melhorar o particionamento. Exemplo: c-Means.
- ▶ Métodos hierárquicos: cria uma decomposição hierárquica de um conjunto de dados. Os métodos hierárquicos podem ser aglomerativos ou divisivos, dependendo de como a decomposição hierárquica é formada – juntando decomposições ou dividindo decomposições. A cada passo, divisões ou junções são feitas. Podem representar seus resultados em dendogramas. Exemplo: BIRCH.



# Categorização dos métodos de agrupamento

---

- ▶ Métodos baseados em densidade: A maioria dos métodos de particionamentos são baseados na distância entre objetos. No caso dos métodos baseados em densidade, os grupos formados crescem de acordo com a densidade de dados em um “potencial” grupo. Para cada dado dentro de um dado grupo, a vizinhança em um dado raio tem que conter pelo menos um numero mínimo de pontos. Exemplo: DBSCAN.
- ▶ Métodos baseados em grid: esses métodos quantizam o espaço de dados em um numero finito de celular que forma uma estrutura em grid. Exemplo: STING
- ▶ Métodos baseados em modelos: criam uma hipótese sobre um modelo para cada um dos grupos e encontram o melhor ajuste dos dados ao modelo. Exemplo: SOM.



# Métodos de particionamento

---

(Han & Kamber, 2006)

- ▶ Métodos de particionamento:
  - ▶ dado um conjunto de dados **com  $n$  objetos**, um método de particionamento **constrói  $k$  partições dos dados**, onde cada partição representa um *cluster* e  $k \leq n$ .
  - ▶ os  $k$  grupos satisfazem os seguintes requisitos:
    - ▶ cada grupo deve conter pelo menos um objeto;
    - ▶ cada objeto deve pertencer a exatamente um grupo.

O segundo requisito pode ser relaxado em técnicas de partições *fuzzy*.

- ▶ dado  $k$  – o número de partições a serem construídas -, um método de particionamento **cria um particionamento inicial** e, usa uma **técnica de realocação iterativa que tenta melhorar tal particionamento**, trocando objetos de um grupo para outro.

... ,o que pode ser feito sob diferentes estratégias!

---



# Métodos de particionamento

(Han & Kamber, 2006)

- ▶ Os clusters são encontrados por meio da otimização de critérios tais como a **função de dissimilaridade baseada em distância**.
- ▶ Para alcançar a otimalidade global, pe necessário enumerar,exaustivamente,todas as possíveis partições.
- ▶ Porém,dado o custo de tal procedimento,métodos heurísticos são usados:
  - ▶ algoritmo **k-means**: onde cada cluster é representado pelo **valor médio** dos objetos no cluster
  - ▶ algoritmo **k-medóides**: onde cada cluster é representado por um dos objetos localizado **próximo ao centro do cluster**.

**K-means e K-medóides** são os mais comuns dentro da categoria de particionamento.

Trabalham bem em bases de dados pequenas e médias e encontram clusters de formas esféricas.



# Clusterização

---

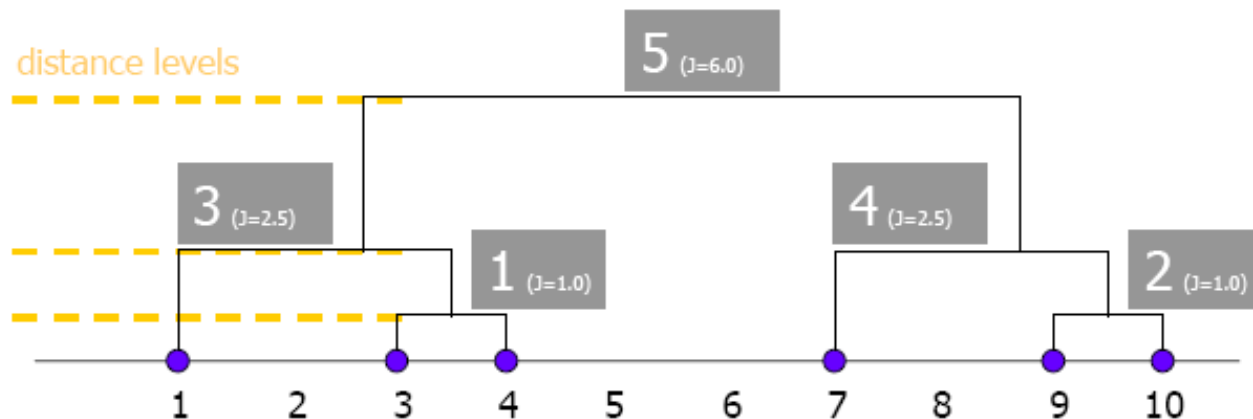
## ▶ Hierárquicos

- ▶ Criam uma decomposição hierárquica da base de dados, representada por um **dendograma** (árvore que iterativamente divide a base de dados em subconjuntos menores até que cada subconjunto consista de somente um objeto).
- ▶ Em tais hierarquias, cada nó da árvore representa um cluster da base de dados. O dendograma pode ser criado de duas formas:
  - ▶ Abordagem aglomerativa (button-up)
  - ▶ Abordagem divisa (top-down)

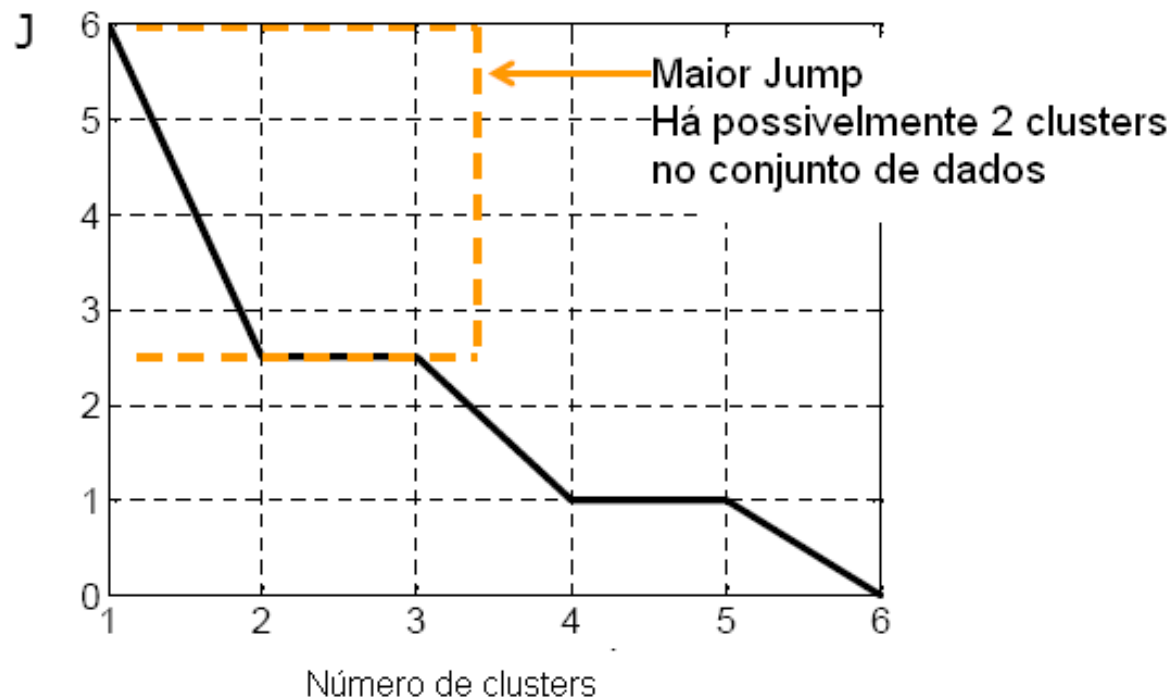
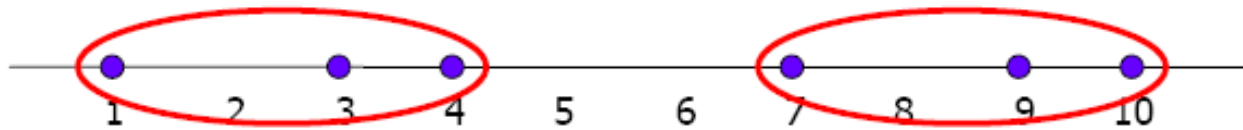


# Dendograma

- É um diagrama representando o processo de agrupamento de clusters aninhado
- Exemplo: Clusterização aglomerativa

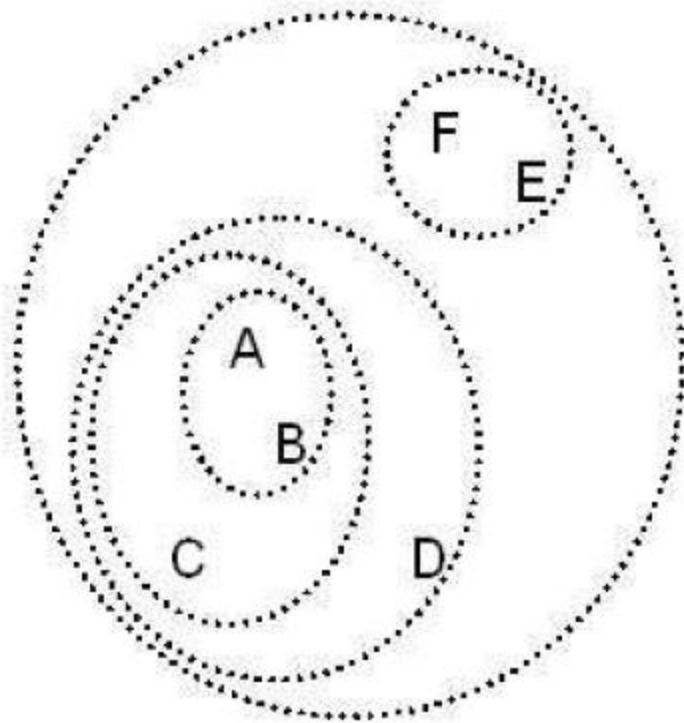


# Dendrograma

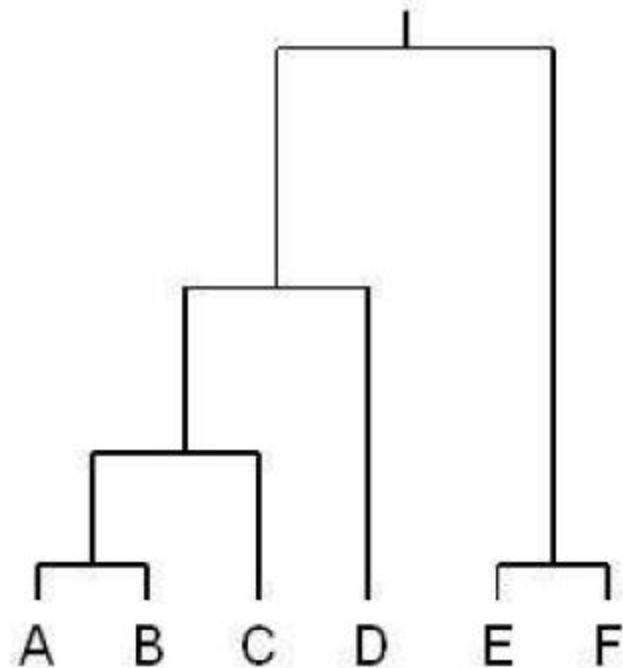


# Clusterização Hierárquica

---



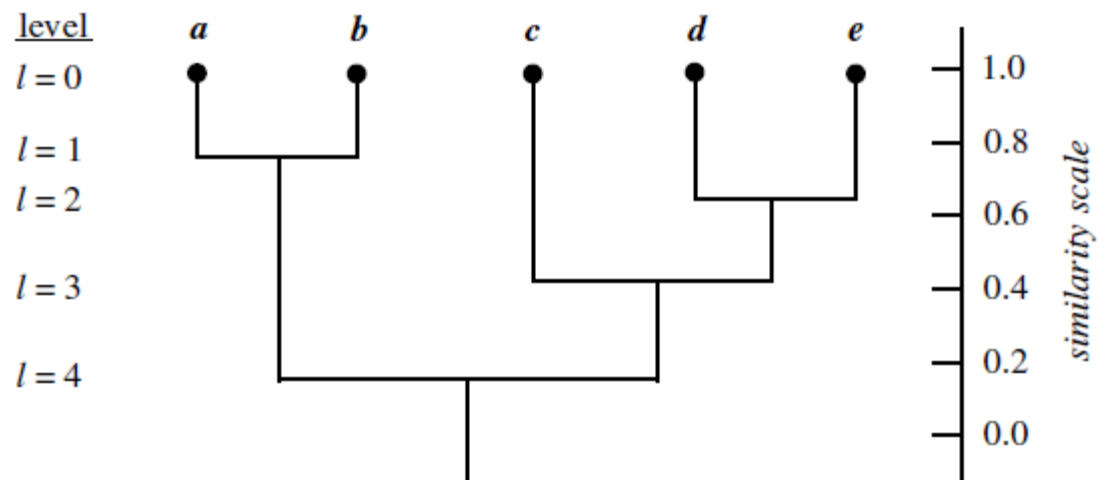
(a) Diagrama aninhado de grupos



(b) Dendrograma

# Clusterização

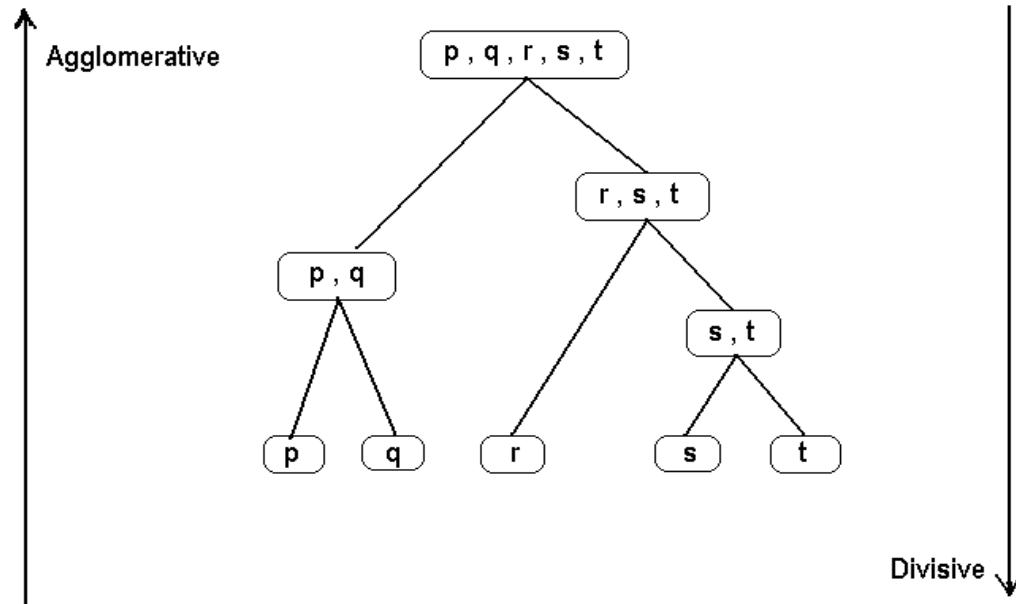
---



Exemplo de um dendograma!!!!



# Clusterização Hierárquica



- Clusterização aglomerativa trata cada ponto como um cluster e então sucessivamente agrupar os clusters até que todos os pontos tem sido agrupados em único cluster restante.
- Clusterização divisiva inicializa todos os objetos em um único cluster. Divide os clusters sucessivamente até objetos tornam-se um cluster de se próprio

# Clusterização

---

- ▶ Abordagem aglomerativa ou bottom-up.
  - ▶ Parte-se das folhas para a raiz.
  - ▶ Coloca-se, inicialmente, cada objeto em seu próprio cluster (ou seja, todos os objetos estão separados), totalizando  $n$  clusters.
  - ▶ Em cada etapa, calcula-se a distância entre cada par de clusters.
  - ▶ Essas distâncias são, geralmente, armazenadas em uma matriz de similaridades.
  - ▶ São escolhidos 2 clusters com distância mínima, juntando-os em seguida.
  - ▶ Atualiza-se a matriz de similaridade.
  - ▶ Esse processo continua até que todos os objetos estejam em um único clusters (o nível mais alto da hierarquia), ou até que uma condição de término ocorra (por exemplo, o número de clusters desejado tenha sido alcançado).



# Clusterização Hierárquica

---

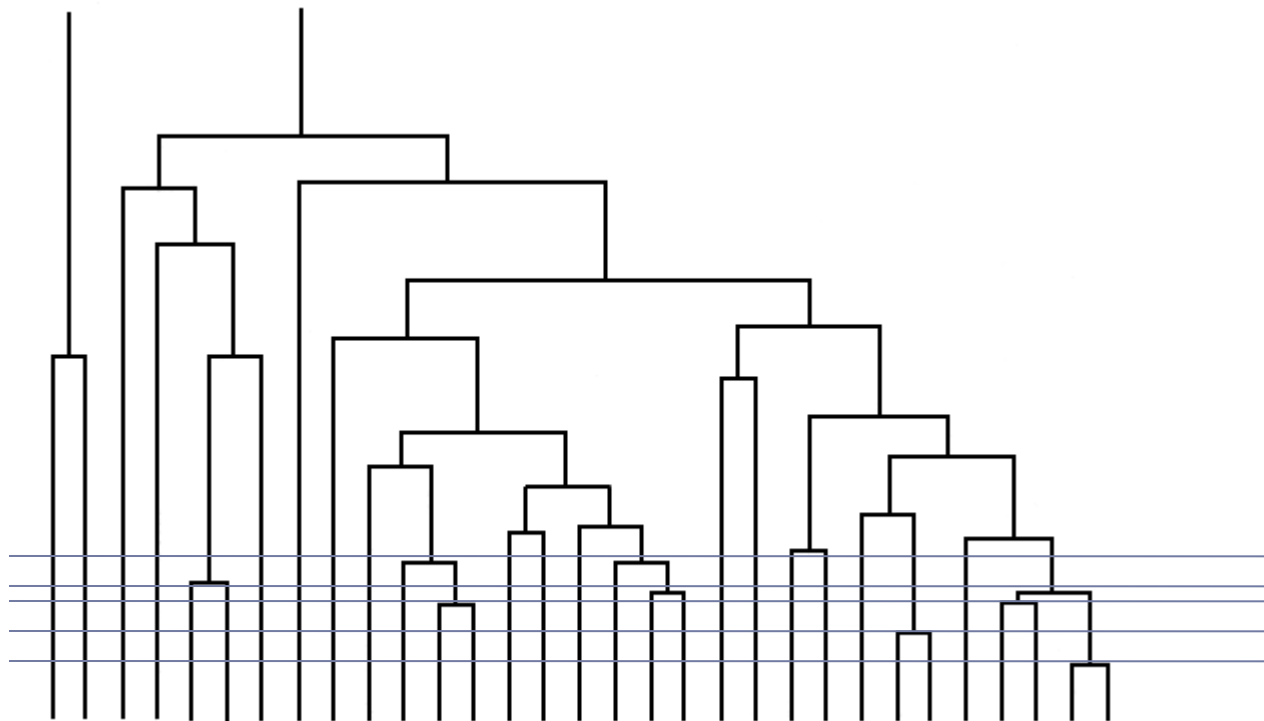
- Clusterização Aglomerativa
  - Passo 1) Inicialize com cluster  $C_1, \dots, C_N$  cada contendo um único indivíduo  $z_j$
  - Passo 2) Encontre os pares mais próximos de clusters distintos, isto é,  $C_i$  e  $C_j$ , agrupe os, delete  $C_j$  e decremente o numero de cluster em 1
  - Passo 3) Se o numero de cluster é 1, então PARE e retorne os rótulos dos cluster, caso contrario vá para o passo 2





# Clusterização

---



Exemplo de um dendograma!!!!



# Medida de Similaridade/Distância entre clusters

- A distância de vizinhança mais próxima entre os clusters A e B é a distância entre dois indivíduos mais próximos, um de cada cluster
- $d(A,B) = \min(d_{ij}), i \in A, j \in B$
- A distância entre os elementos pode ser a distância Euclidiana ou qualquer outra distância
- Algoritmo clusterização *Single Link* (Single Linkage, Nearest Neighbor)
  - Passo 1) O numero de cluster desejado é c.
  - Passo 2) Comece com clusters  $C_1, \dots, C_N$ , cada contendo um único individuo  $z_j$
  - Passo 3) Encontre os pares de clusters com menor  $d(C_i, C_j)$ , agrupe-os, delete  $C_j$  e decremente o numero de cluster por 1
  - Passo 4) Se o número de cluster é c, então PARE e retorne os labels dos cluster, caso contrário vá para o Passo 2

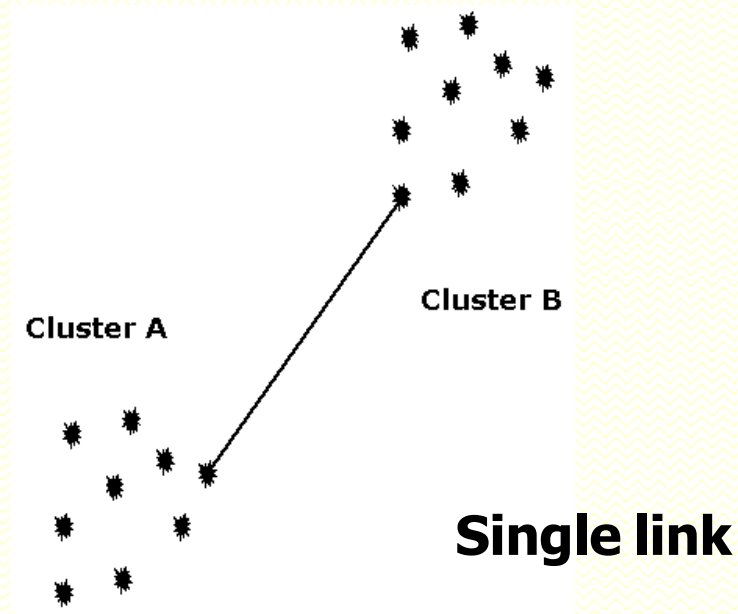
# Clusterização Hierárquica

---

- ▶ **In complete-link** (or complete linkage) hierarchical clustering, we merge in each step the two clusters whose merger has the smallest diameter (or: the two clusters with the smallest **maximum** pairwise distance).
- ▶ **In single-link** (or single linkage) hierarchical clustering, we merge in each step the two clusters whose two closest members have the smallest distance (or: the two clusters with the smallest **minimum** pairwise distance).
- ▶ **Average-link** (or group average) clustering is a compromise between the sensitivity of complete-link clustering to outliers and the tendency of single-link clustering to form long chains that do not correspond to the intuitive notion of clusters as compact, spherical objects

# Clusterização Hierárquica

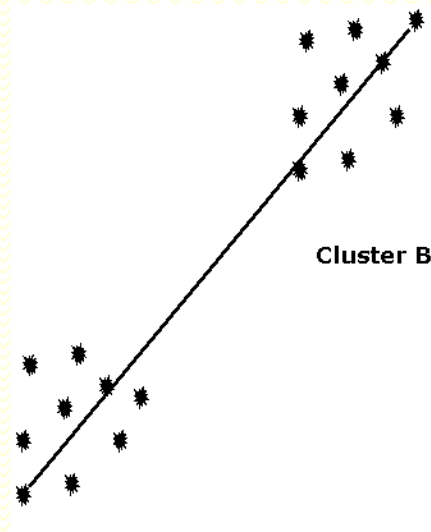
---



- Na clusterização hierárquica single-link, nós agrupamos em cada passo os dois clusters cujos **dois membros mais próximos tem a menor distância..**

# Clusterização Hierárquica

---

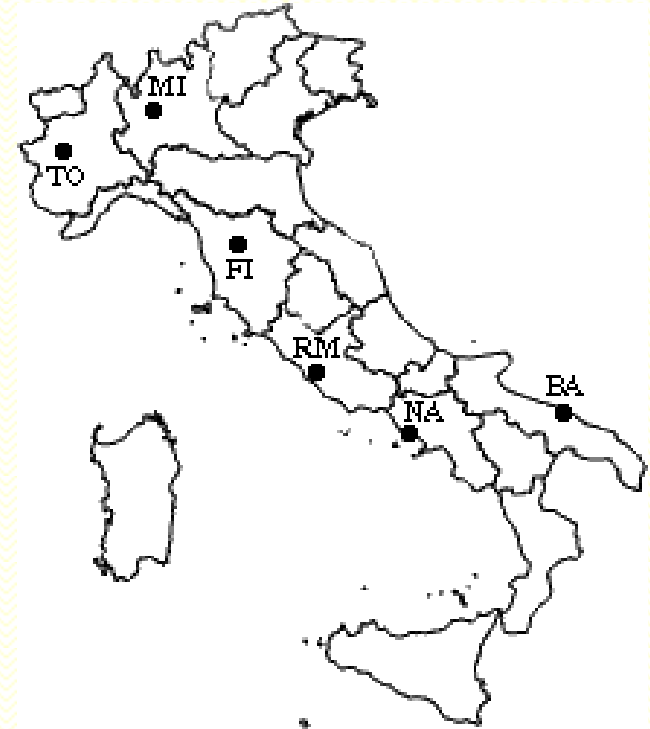


**Complete link**

- Na clusterização hierarquica complete-link, nós agrupamos em cada passo os dois clusters cujos grupos tem a menor distância.
- Average linkage é um compromisso entre a sensibilidade da clusterização complete-link para outliers e a tendência da clusterização single-link para formar cadeias longas que não correspondem para a noção intuitiva de clusters como compacto, objetos esféricos.

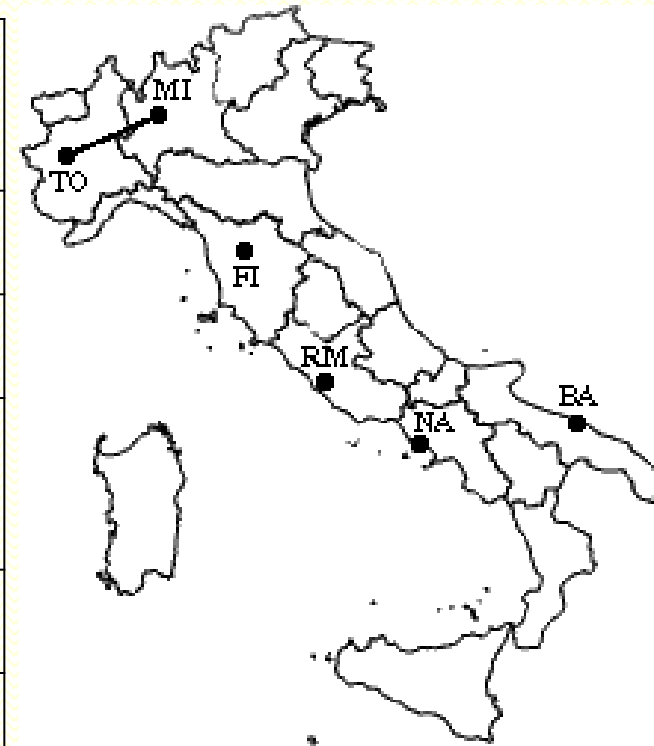
# Exemplo – Single Link AC

	BA	FI	MI	NA	RM	TO
BA	0	662	877	255	412	996
FI	662	0	295	468	268	400
MI	877	295	0	754	564	138
NA	255	468	754	0	219	869
RM	412	268	564	219	0	669
TO	996	400	138	869	669	0



# Exemplo – Single Link AC

	BA	FI	MI/TO	NA	RM
BA	0	662	877	255	412
FI	662	0	295	468	268
MI/TO	877	295	0	754	564
NA	255	468	754	0	219
RM	412	268	564	219	0



# Exemplo – Single Link AC

	BA	FI	MI/TO	NA/RM
BA	0	662	877	255
FI	662	0	295	268
MI/TO	877	295	0	564
NA/RM	255	268	564	0





# Exemplo – Single Link AC

	BA/NA/ RM	FI	MI/TO
BA/NA/ RM	0	268	564
FI	268	0	295
MI/TO	564	295	0

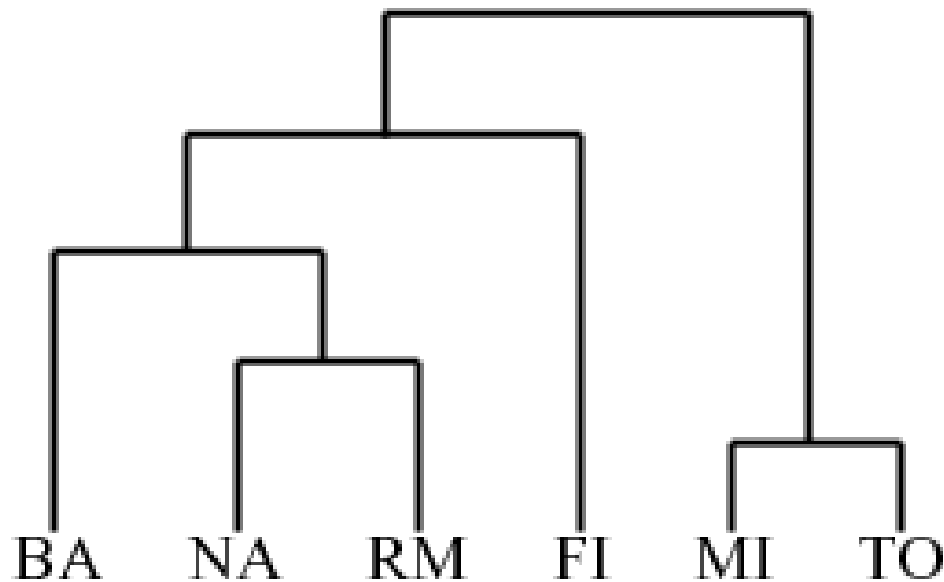


# Exemplo – Single Link AC

	BA/FI/NA/RM	MI/TO
BA/FI/NA/RM	0	295
MI/TO	295	0



# Exemplo – Single Link AC



# Clusterização - Sumarização

---

- ▶ Muito comum em KDD. Encadeia as tarefas primárias de Clusterização e Sumarização.
- ▶ Aplicável em situações nas quais o conjunto completo de registros do banco de dados disponha **de pouca similaridade** entre seus elementos, esta tarefa consiste de:
  - ▶ Agrupar os dados em função de sua similaridade.
    - ▶ Cada cluster obtido passa a ser considerado isoladamente como um novo conjunto de dados.
  - ▶ Cada **novo conjunto de dados** é apresentado a um algoritmo de sumarização que descreve as principais características dos registros presentes naquele conjunto.



# Clusterização

---

- ▶ Abordagem divisiva ou top-down.
  - ▶ Parte-se da raiz para as folhas.
  - ▶ Inverte-se o processo por começar com todos os objetos em um único cluster.
  - ▶ Em cada etapa, um cluster é escolhido e dividido em dois clusters menores.
  - ▶ Esse processo continua até que se tenha  $n$  clusters ou até que uma condição de término aconteça.



---

Medidas de dissimilaridade

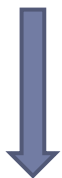
---



# Tipos de Atributos

---

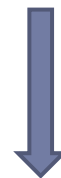
Altura	Sexo	Estado Civil	Grau de Instrução	Temperatura



Escala Razão



Escala Nominal



Escala Ordinal



Escala Intervalar



# O que é similaridade?

A qualidade, caráter ou condição das coisas similares.

*(Dicionário Houaiss)*



- ▶ Similaridade é difícil de definir, mas...  
*Reconhecemos quando a vemos!*



# Dissimilaridade entre objetos

✉ Distâncias são normalmente usadas como medida de dissimilaridade entre objetos

✉ Entre as mais populares: distancia de *Minkowski*

$$d(i, j) = \sqrt[q]{(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \dots + |x_{ip} - x_{jp}|^q)}$$

onde  $i = (x_{i1}, x_{i2}, \dots, x_{ip})$  e  $j = (x_{j1}, x_{j2}, \dots, x_{jp})$  são dois vetores  $p$ -dimensionais, e  $q$  é um inteiro positivo

✉ Se  $q = 1$ ,  $d$  é a distância de Manhattan

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$



# Dissimilaridade entre objetos

✉ Se  $q = 2$ ,  $d$  é a distância:

$$d(i, j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \dots + |x_{i_p} - x_{j_p}|^2)}$$

- Properties

- ♦  $d(i, j) \geq 0$ ,  $d(i, i) = 0$ ,  $d(i, j) = d(j, i)$
- ♦  $d(i, j) \leq d(i, k) + d(k, j)$

✉ Outras alternativas: distância ponderada, correlação (similaridade), etc.



# Variáveis binárias

- ✉ Tabela de contingencia para dados binários

		Objeto $j$		
		1	0	soma
Objeto $i$	1	a	b	a + b
	0	c	d	c + d
	soma	a + c	b + d	p

- ✉ Simple matching (invariante, se a variável binaria é simetrica):

$$d(i, j) = \frac{b+c}{a+b+c+d}$$

- ✉ Jaccard (não invariante se a variável binaria é assimetrica)

$$d(i, j) = \frac{b+c}{a+b+c}$$

# Variáveis Simétricas x Assimétricas

## ✉ Variáveis binárias simétricas

- ✉ Aquelas que não possuem preferência na codificação (caso da variável sexo), o resultado não sofre alterações quando os códigos são modificados, assim a e d tem a mesma função.

## ✉ Variáveis binárias assimétricas

- ✉ Aquelas cuja codificação usa o número 1 para indicar a presença do atributo e 0 para a ausência (na área de saúde 1 indica a presença do agravo e 0 a ausência)
- ✉ A modificação desta codificação altera os resultados. Por esta razão deve-se utilizar coeficientes específicos para esta mensuração; indivíduos com códigos 1-1 indicam semelhança, mas indivíduos 0-0 não indicam necessariamente semelhança



# Dissimilaridade entre variáveis binárias

## Exemplo

Nome	Genero	Febre	Tosse	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

- ▶ genero é um atributo simétrico
- ▶ os atributos restantes são binários assimétricos
- ▶ considere que os valores de Y e P sejam fixados para 1, e o valor de N para 0

$$d(jack, mary) = \frac{0 + 1}{2 + 0 + 1} = 0.33$$

$$d(jack, jim) = \frac{1 + 1}{1 + 1 + 1} = 0.67$$

$$d(jim, mary) = \frac{1 + 2}{1 + 1 + 2} = 0.75$$

Jack\Mary	Y ou P	N
Y ou P	2	0
N	1	3

# Variáveis Nominais

✉ Variável de escala nominal que pode assumir **mais de 2 categorias**, e.x., vermelho, amarelo, azul, verde

✉ **Metodo 1:** Concordancias simples

- $m$ : # das concordancias,  $p$ : numero de variáveis

$$d(i, j) = \frac{p - m}{p}$$

✉ **Metodo 2:** usa um grande numero de variáveis binárias

- Criação de uma nova variável binária para cada uma das  $M$  categorias



# Variáveis intervalares

- ✉ Permitem não apenas ordenar em postos os itens que estão sendo medidos, mas também quantificar e comparar o tamanho das diferenças entre eles.
- ✉ Exemplo: temperatura medida em graus Celsius constitui uma variável intervalar.
- ✉ Pode-se dizer que a temperatura de  $40^{\circ}\text{C}$  é maior do que  $30^{\circ}\text{C}$  e que um aumento de  $20^{\circ}\text{C}$  para  $40^{\circ}\text{C}$  é duas vezes maior do que um aumento de  $30^{\circ}\text{C}$  para  $40^{\circ}\text{C}$ .



# Variáveis ordinais

- ✉ Uma variável ordinal pode ser qualitativa ou quantitativa (escolaridade)
- ✉ A ordem é importante, e.x., rank
- ✉ Pode ser tratada como uma variável de escala intervalar
  - Trocando  $x_{if}$  pelo seu rank
  - Mapear a amplitude de cada variável em  $[0, 1]$  trocando  $r_{if}$  por

$$r_{if} \in \{1, \dots, M_f\}$$

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

- Calcular a dissimilaridade usando os métodos das variáveis de escala intervalar





# Variáveis de razão

- ✉ variável de de razão: uma medida positiva sobre uma escala não linear. Aproxima-se da escala exponencial, como por exemplo:  $Ae^{Bt}$  ou  $Ae^{-Bt}$
- ✉ Exemplos de variáveis (escalas) de razão são: idade, salário, preço, volume de vendas, distâncias.
- ✉ Metodos:
  - Trata-las como variáveis de escala intervalar — *não é uma boa escolha! (porque?)*
    - *A escala do intervalo pode ser distorcido*
  - Aplicar uma transformação logaritmica
$$y_{if} = \log(x_{if})$$
  - Trata-las como os dados ordinais quantitativos
    - Tratando os seus ranks como escala intervalar.



# Variaveis de vários tipos

- ✉ Uma base de dados pode conter todos os 6 tipos:
  - simetrica binaria, assimetrica binária, nominal, ordinal, intervalar e proporcional.
- ✉ Pode-se usar uma expressão ponderada para combina-las.

$$d(i, j) = \frac{\sum_{f=1}^P \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^P \delta_{ij}^{(f)}}$$

- $f$  é binária ou nominal:  
 $d_{ij}^{(f)} = 0$  se  $x_{if} = x_{jf}$ , ou  $d_{ij}^{(f)} = 1$  senão
- $f$  é intervalar: use a distancia normalizada
- $f$  é ordinal ou de escala proporcional
  - ♦ Calcule ranks  $r_{if}$  e
  - ♦ E trate  $z_{if}$  como intervalar

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

---

C-means (ou K-means)

---



# Introdução

---

- ▶ Exemplo de um algoritmo de partição de um conjunto de dados, cujo processo é utilizado no algoritmo C-means.

A partir da escolha, **geralmente randômica**, de  $C$  pontos dentro do espaço dos dados a serem “clusterizados”, é iniciado um trabalho iterativo de partição dos dados do conjunto. Cada um destes  $C$  pontos define uma diretriz para a realização deste trabalho.

**Cada dado do conjunto é colocado em uma partição definida por um dos  $C$  pontos inicialmente gerados.** A escolha da partição é feita de acordo com uma métrica de distância/similaridade entre os pontos do conjunto e os pontos representantes das partições.

Então, **um novo representante para cada partição é escolhido a partir de uma relação entre os pontos presente na partição e o processo reinicia.**

O processo termina quando a variação de cada representante de partição é mínima.

# C-means

---

- ▶ No C-means o ponto representante da partição é dito “centróide”.
- ▶ Na atualização dos pontos de representação de partição é utilizada a média dos pontos clusterizados dentro da respectiva partição.
- ▶ Assim, o algoritmo C-means
  - ▶ Toma o parâmetro de entrada  $C$ , e divide um conjunto de  $n$  objetos em  $c$  clusters (ou partições) tal que a similaridade intracusters resultante seja alta, mas a similaridade intercluster seja baixa.
- ▶ A similaridade em um cluster é medida em respeito ao valor médio dos objetos neste clusters.



# C-means

---

- ▶ **Processo de execução:**
  - ▶ Selecionar randomicamente  $c$  objetos, que inicialmente representam cada um a média de um cluster.
  - ▶ Para cada um dos objetos remanescentes, é feita a atribuição ao cluster ao qual o objeto é mais similar, baseado na distância entre o objeto e a média do cluster.
  - ▶ Então, o algoritmo computa as novas médias para cada clusters.
  - ▶ Esse processo se repete até que uma condição de parada seja atingida.



# C-means

(Han & Kamber, 2006)

- ▶ Técnica baseada em centróide.

Objetivo: maximizar a similaridade intracluster e minimizar a similaridade intercluster.

- ▶ A similaridade do cluster é medida em relação ao valor médio dos objetos no cluster (centróide ou centro de gravidade).

- ▶ Critério do erro quadrado:
$$E = \sum_{i=1}^k \sum_{p \in C_i} |p - m_i|^2$$

onde **E** é a soma do erro quadrado (distância) para todos os objetos no conjunto de dados; **p** é o ponto em um espaço de representação do objeto; **m<sub>i</sub>** é a média do clusters **C<sub>i</sub>** (tanto **p** quanto **m** são multidimensionais).

# K-means - algoritmo

---

(Han & Kamber, 2006)

**Input:**

k: o número de clusters;  
D: o conjunto de dados com **n** objetos;

**Output:**

um conjunto de k clusters (os vetores protótipos de cada clusters);

**Method:**

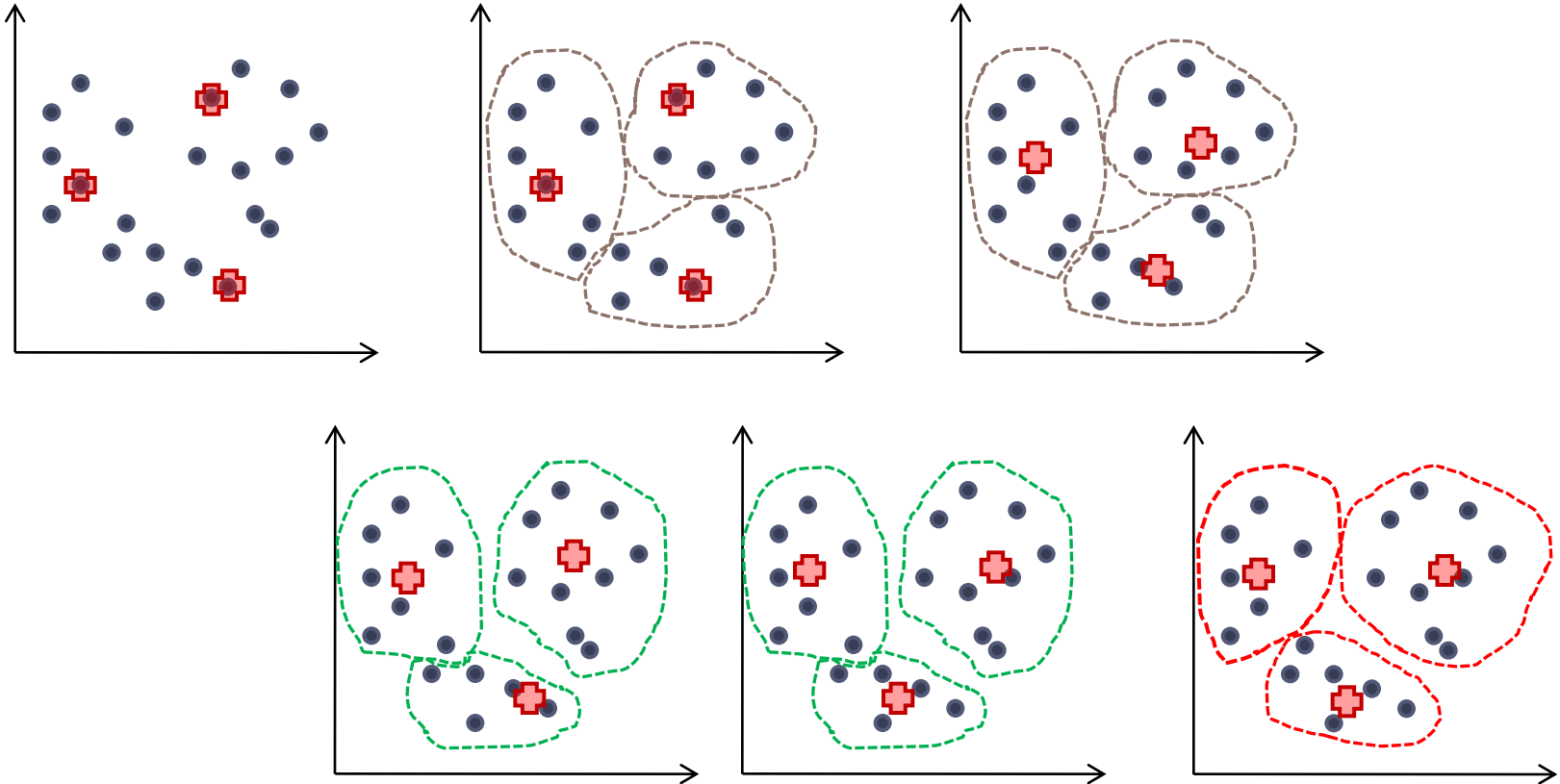
- (1) escolha k objetos de D, arbitrariamente, para representar os centros dos clusters (partições) iniciais;
- (2) **repeat**
- (3) (re)associe cada objeto para o cluster que tem seu centro mais similar ao objeto;
- (4) atualize as médias dos clusters, i.e., calcule o valor médio dos objetos para cada cluster;
- (5) **until** "nenhuma mudança ocorrer" (ou outro critério de parada);





# K-means - ilustrando

(Han & Kamber, 2006)



# C-means

---

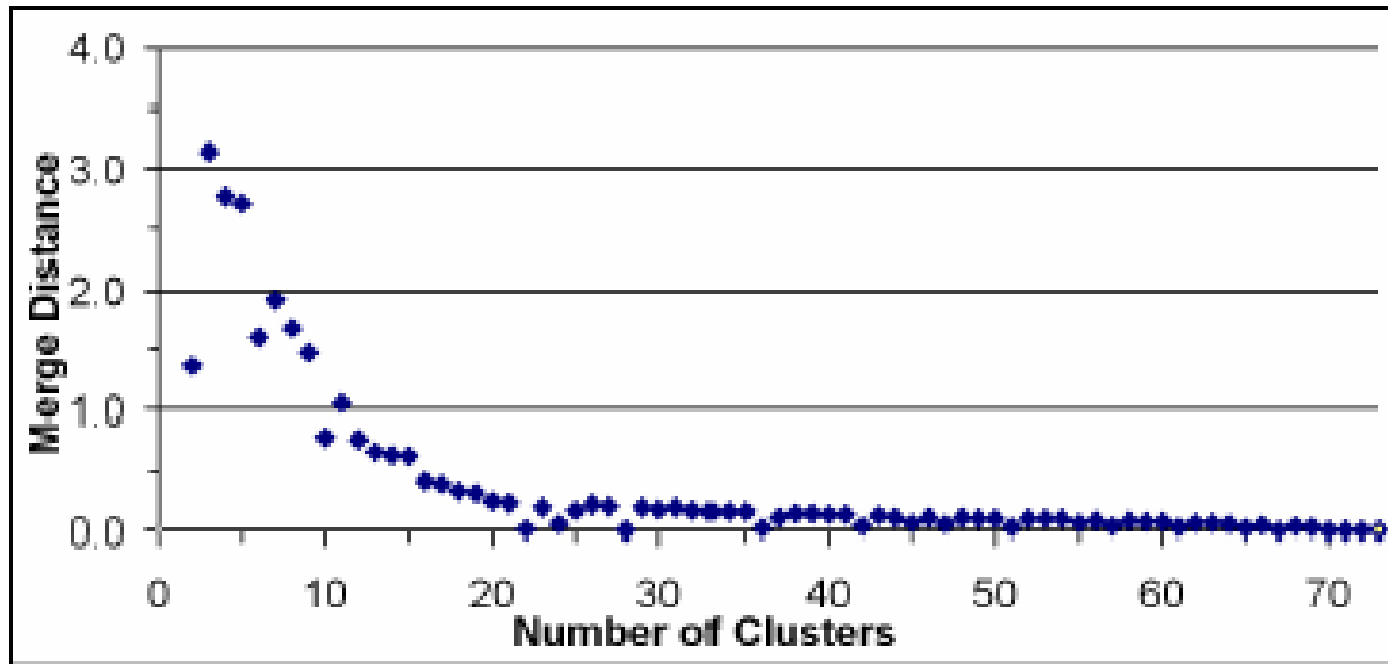
## ▶ Alguns detalhes:

- ▶ Apresenta bons resultados para conjuntos de dados com clusters **densos e compactos** e bem separados um dos outros.
- ▶ A necessidade de especificar o parâmetro  $c$  (o número de clusters) é visto como uma desvantagem em relação a outros algoritmos de clusterização.
- ▶ O parâmetro  $c$  **precisa ser inspecionado** com cuidado.
- ▶ Não é adequado para descobrir clusters com formas não convexas ou clusters de tamanhos muito diferentes.
- ▶ É sensível a ruído



# K-Means – How many K's ?

---



- ▶ O joelho de uma curva é definido como o ponto de máxima curvatura

# Comentários sobre o método *K-Means*

## ✉ Pontos fortes

- *Relativamente eficiente:  $O(tkn)$ , onde  $n$  é # objetos,  $k$  é # grupos, e  $t$  é # iterações. Normalmente,  $k, t \ll n$ .*
- *Frequentemente termina em um ótimo local. O ótimo global pode ser encontrado usando técnicas tais como: deterministic annealing e algoritmos genéticos*

## ✉ Pontos fracos

- *Aplicavel apenas quando a média é definida, o que fazer com dados categóricos?*
- *É necessário especificar a priori  $k$ , o número de grupos*
- *É sensível a ruídos e valores aberrantes*
- *Não é apropriado para a descoberta de grupos não esféricos*



# C-means

---

## ▶ Variações

- ▶ Seleção das  $c$  médias iniciais;
- ▶ Medidas de similaridade;
- ▶ Estratégias de cálculo de atualização dos C-pontos.

## ▶ C-modes

- ▶ Utilizado para clusterização de **dados categóricos** (nominais).
- ▶ Em geral, no lugar do cálculo da média, calcula-se a moda (valor que aparece com maior frequência) dos objetos.
- ▶ A medida de similaridade precisa ser apropriada para dados categóricos.



# C-means

---

## ▶ C-Prototypes

- ▶ Integração do métodos C-means com o método C-modes.
  - ▶ Aplicado a bases de dados que contenham tanto atributos **numéricos** quanto **atributos categóricos**.

## ▶ C-Medoids

- ▶ Medoid: objeto mais centralmente localizado em um cluster.
  - ▶ Utilizado como representante da partição.
- ▶ Existe a troca iterativa de um *medoid* por um não *medoid*, visando a melhoria da clusterização.
- ▶ A qualidade é estimada usando uma função que mede a similaridade média entre os objetos e o *medoid* de seu cluster.



---

C-modes (ou K-modes)

---



**Moda** corresponde ao número que ocorre mais frequentemente num grupo de números. Por exemplo, a moda de 2, 3, 3, 5, 7 e 10 é 3.

# Introdução

- ▶ Seja  $X$  e  $Y$  dois objetos descrito por  $F$  atributos categóricos
- ▶ A medida de dissimilaridade  $d(X, Y)$  entre  $X$  e  $Y$  pode ser definida pelo número total de incompatibilidade dos atributos categóricos correspondente dos dois objetos
- ▶ Menor o numero de incompatibilidade, maior similaridade dos dois objetos.

- ▶ Matematicamente, nós podemos dizer

$$d(X, Y) = \sum_{j=1}^F \delta(x_i, x_j)$$

- ▶ onde 
$$\delta(x_i, x_j) = \begin{cases} 0, & (x_i = y_j) \\ 1, & (x_i \neq y_j) \end{cases}$$

- ▶  $d(X, Y)$  da uma importância igual para cada categoria de um atributo



# Introdução

---

- ▶ Seja  $Z$  um conjunto de objetos descritos por atributos categóricos  $A_1, A_2, \dots, A_F$  uma moda de  $Z = \{Z_1, Z_2, \dots, Z_n\}$  é um vetor  $Q = [q_1, q_2, \dots, q_F]$  que minimiza

$$d(Z, Q) = \sum_{j=1}^F d(Z_j, Q)$$

- ▶  $Q$  não é necessariamente um elemento de  $Z$ .
- ▶ Empregando a medida de dissimilaridade definida anteriormente, a função custo torna-se

$$C(Q) = \sum_{p=1}^k \sum_{i=1}^n \sum_{j=1}^F \delta(z_{ij}, q_{pij})$$

- 
- ▶ Onde  $Q_p = [q_{p1}, q_{p2}, \dots, q_{pm}] \in Q$

# Algoritmo

---

- ▶ O algoritmo K-modes minimiza a função custo definida abaixo

$$C(Q) = \sum_{p=1}^k \sum_{i=1}^n \sum_{j=1}^F \delta(z_{ij}, q_{pij})$$

- ▶ O algoritmo consiste dos seguintes passos

- ▶ 1) Selecione K modas iniciais, um para cada cluster
- ▶ 2) Atribua cada objeto para o cluster cuja moda é mais próxima de acordo com a equação abaixo

$$d(X, Q) = \sum_{j=1}^F \delta(x_i, q_j)$$

- ▶ 3) Calcule as novas modas de todos os clusters

$$C(Q) = \sum_{p=1}^k \sum_{i=1}^n \sum_{j=1}^F \delta(z_{ij}, q_{pij})$$

- ▶ 4) Repita passo 2 e 3 até nenhum objeto tem alterado a pertinência ao cluster

---

K-medóides



# O Método K-Medoids

- ✉ Recebe como entrada um banco de dados  $X$  e um número  $k$ 
    - ✉  $K$  representa o número de clusters que se deseja formar
    - ✉ Os dados devem ser transformados em uma matriz de dissimilaridade
  - ✉ Visar neutralizar a desvantagem do Kmeans no que diz respeito a sua sensibilidade a ruídos
  - ✉ Encontre objetos *representativos*, chamados medoids, nos grupos
  - ✉ PAM (Partitioning Around Medoids, 1987)
    - Inicie com um conjunto inicial de medoids e iterativamente troque um deles por um não medoide se a distancia total do agrupamento melhora
    - PAM funciona para conjuntos de dados pequenos, mas não possui escalabilidade suficiente para os grandes
  - ✉ CLARA (Kaufmann & Rousseeuw, 1990)
  - ✉ CLARANS (Ng & Han, 1994): Amostragem aleatória
- 



# PAM (Partitioning Around Medoids) (1987)

- ✉ O objetivo é determinar qual a melhor escolha de  $k$  objetos do banco de dados que serão os centros de  $k$  clusters.
- ✉ Estes são chamados medóides
- ✉ O processo para encontrar estes  $k$  objetos é iterativo e utiliza a matriz de dissimilaridade dos dados
- ✉ Após determinados este  $k$  objetos ideiais, os clusters são criados em torno deles



# PAM (Partitioning Around Medoids) (1987)

---

- ▶ 1) Seleciona-se os objetos do banco de dados de forma arbitrária. Estes vão ser os medóides iniciais
- ▶ 2) Para todos os pares de objetos  $O_i$  e  $O_h$ , onde  $O_i$  é um medóide e  $O_h$  é um não medóide, calcula-se

$$CT_{ih} = \sum_j C_{jih}$$

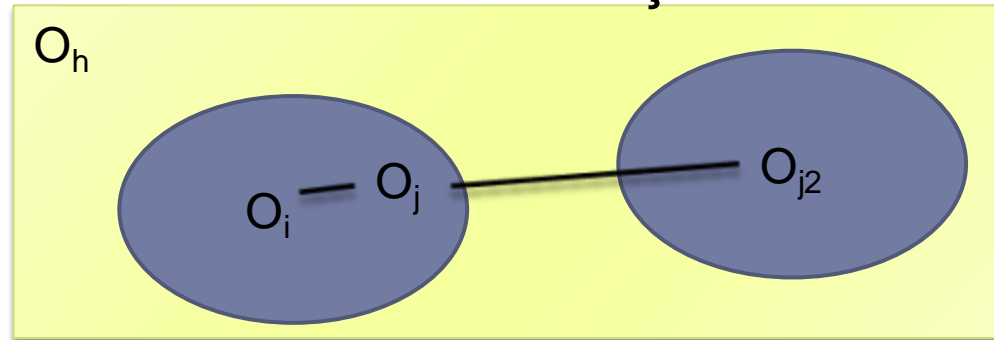
- ▶ onde  $CT_{ih}$  é o custo de substituir  $O_i$  por  $O_h$ ,
- ▶ a somatória  $\sum_j$  é feita para todos os objetos  $O_j$  que não são medóides
- ▶ e  $C_{jih}$  o termo é um número que mede o erro obtido por esta substituição, sobre o objeto não medóide. Este número é calculado da seguinte maneira



# PAM (Partitioning Around Medoids) (1987)

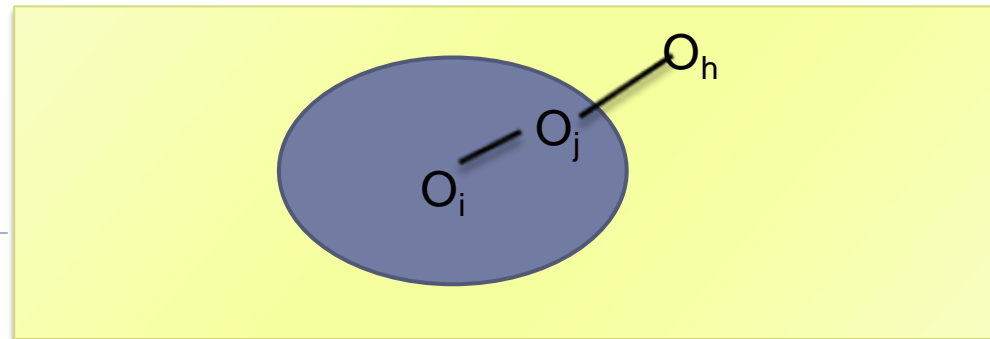
- Se  $O_j$  está no cluster  $O_i$  e com a mudança de  $O_i$  para  $O_h$ ,  $O_j$  fica mais próximo de outro medóide  $O_{j2}$ . Neste caso  $O_j$  iria para o cluster  $O_{j2}$  e o custo desta mudança seria:

$$C_{jih} = d(O_j, O_{j2}) - d(O_j, O_i)$$



- Se  $O_j$  está no cluster  $O_i$  e com a mudança de  $O_i$  para  $O_h$ ,  $O_j$  fica mais próximo de  $O_h$ . Neste caso  $O_j$  iria para o cluster  $O_h$  e o custo desta mudança seria:

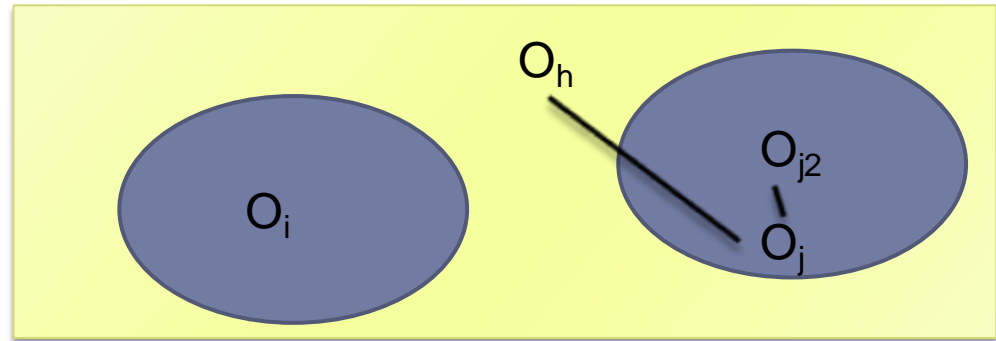
$$C_{jih} = d(O_j, O_h) - d(O_j, O_i)$$



# PAM (Partitioning Around Medoids) (1987)

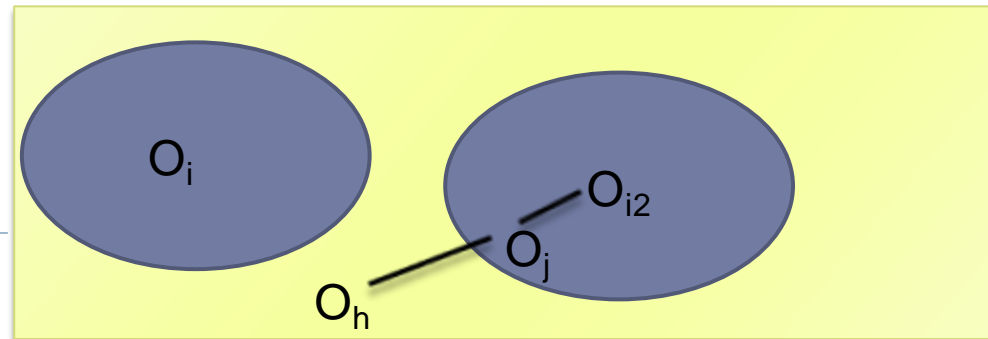
- Se  $O_j$  não está no cluster  $O_i$ , por exemplo está no cluster de  $O_{j2}$ , e com a mudança de  $O_i$  para  $O_h$ ,  $O_j$  fica mais próximo  $O_{j2}$  do que de  $O_h$ . Neste caso  $O_j$  permaneceria em seu próprio, nenhuma mudança ocorreria portanto cluster :

$$C_{jih} = 0$$



- Se  $O_j$  não está no cluster  $O_i$ , por exemplo está no cluster de  $O_{j2}$ , e com a mudança de  $O_i$  para  $O_h$ ,  $O_j$  fica mais próximo  $O_h$  do que de  $O_{j2}$ . Neste caso  $O_j$  iria para o cluster de  $O_h$  e portanto :

$$C_{jih} = d(O_j, O_h) - d(O_j, O_{j2})$$





# PAM (Partitioning Around Medoids) (1987)

---

- ▶ 3. Seleciona-se o para  $O_i$ ,  $O_h$  que corresponde ao  $\min_{O_i O_h} TC_{ih}$
- ▶ Se este mínimo é negativo então substitui  $O_i$   $O_h$  e volta ao passo 2. Repare que se este mínimo é negativo então existe uma possibilidade de substituir um medóide atual  $O_i$  por um outro objeto  $O_h$  de modo que a distribuição de dos outros objetos do banco de dados em torno dos novos cluster seja mais adequada, isto é, os objetos em cada cluster estão mais próximos de seus centros do que antes.
- ▶ 4. Caso o mínimo seja positivo, isto significa que não há maneira de substituir os atuais medóides de modo a baixar o custo total do atual aglutinamento. Neste caso, os medóides convergiram.
- ▶ 5. Agora varre-se o banco de dados para alocar os seus elementos nos clusters adequados, isto é, cada elemento  $O$  é alocado ao cluster cujo representante medóide  $m$  está mais próximo de  $O$ .



---

K-prototypes

---



# K-prototypes Algorithm

---

- ▶ Integra o algoritmo K-means e K-modes para agrupar objetos de tipos numéricos e categóricos
- ▶  $A_1^r, A_2^r, \dots, A_p^r, A_{p+1}^c, \dots, A_m^c$  é o número de atributos, os  $p$  primeiros são dados numéricos, o restante são dados categóricos
- ▶ Função custo

$$d_2(X, Y) = \sum_{j=1}^p (x_j - y_j)^2 + \gamma \sum_{j=p+1}^m \delta(x_j, y_j)$$

- ▶ O primeiro termo é a distância Euclidiana medida sobre os atributos numéricos e o segundo termo é uma medida de dissimilaridade medida sobre atributos categóricos
  - ▶ O peso  $\gamma$  é usado para evitar favorecer algum tipo de atributo
- 





---

# Uma Rede Neural Artificial Não Supervisionada

**Self Organizing Maps**  
(by Teuvo Kohonen)



# Exemplos de auto-organização na natureza

---

## ► Conchas



# Exemplos de auto-organização na natureza

---

- ▶ Dunas
- ▶ Reagentes químicos
- ▶ Rachaduras na lama
- ▶ Células de convecção de Bénard
- ▶ Rachadura em tinta
- ▶ Rugas em verniz





# Exemplos de auto-organização na natureza

---

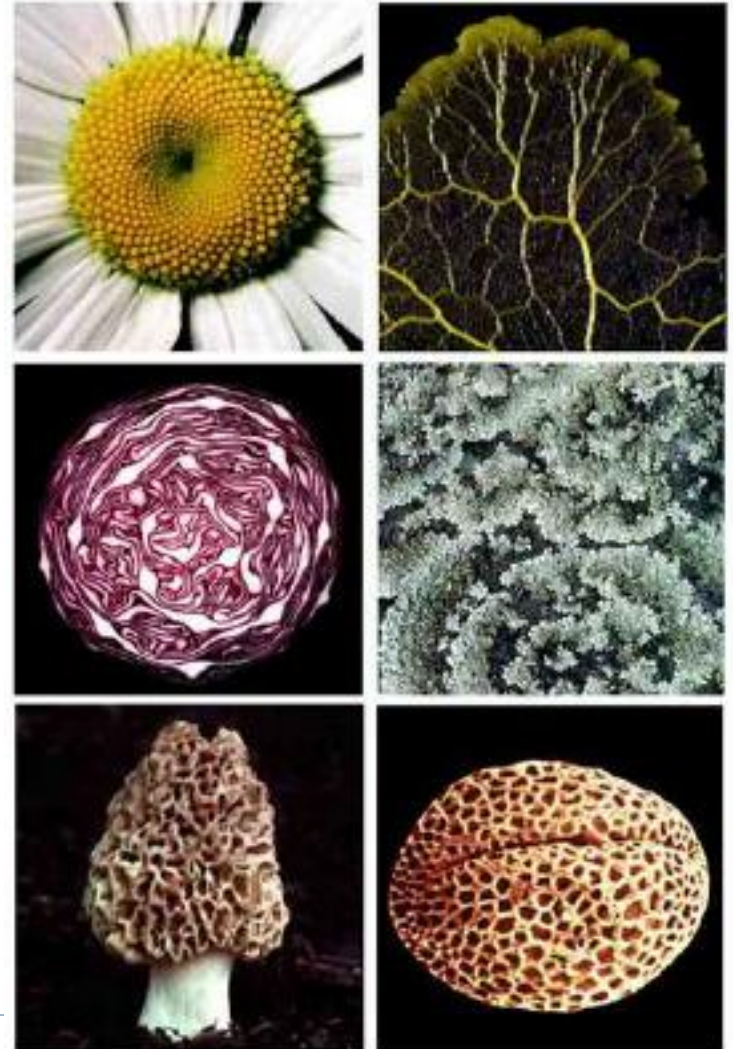
- ▶ Em animais
  - ▶ Listras da zebra
  - ▶ Manchas da girafa
  - ▶ Listras do tigre
  - ▶ Anfibios
  - ▶ Largatos



# Exemplos de auto-organização na natureza

---

- ▶ Em plantas
  - ▶ Margarida
  - ▶ Slime mold
  - ▶ Repolho vermelho
  - ▶ Líquen
  - ▶ Cogumelo
  - ▶ Grão de pólen





# Exemplos de auto-organização na natureza

---

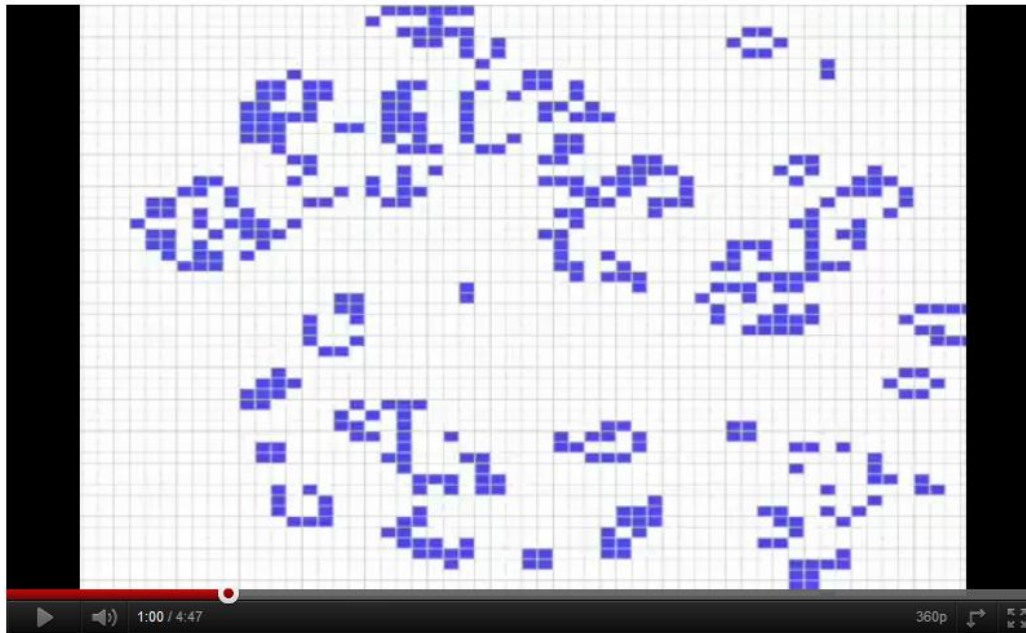
- ▶ Em insetos
  - ▶ Asas de borboletas
  - ▶ Exoesqueletos de besouros



# Jogo da Vida

---

- ▶ Concebido por John Horton Conway, matemático britânico, em 1970.
- ▶ Procure assistir ao vídeo: <http://www.youtube.com/watch?v=XcuBvj0pw-E&feature=related%E2%8C%AA>



- ▶ Célula morta e com três vizinho vivos ressuscita;
- ▶ Célula viva e com dois ou três vizinhos vivos permanece viva;
- ▶ Em todos os outros casos, a célula morre ou permanece morta

## Inspiração biológica (Apenas interessante...)

---

- ▶ Sistemas biológicos têm auto-organização
- ▶ Evidência de:
  - ▶ Estrutura de camadas no cérebro
  - ▶ Cérebro organiza espacialmente a informação
  - ▶ “Conceitos” similares são mapeados para áreas adjacentes
  - ▶ Trabalho experimental com visão em animais sugere uma organização similar ao SOM no córtex



# Perspectiva Histórica

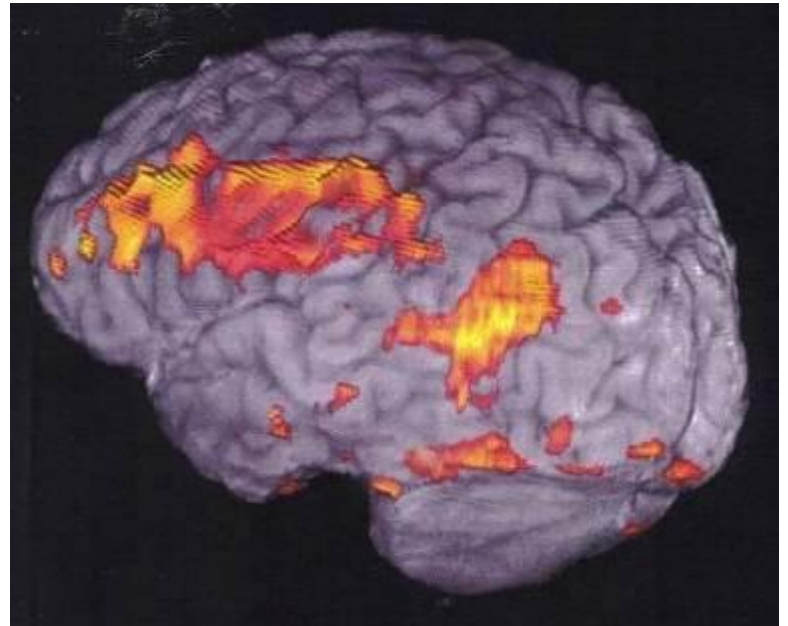
---

- ▶ **Prof. Tuevo Kohonen** (Universidade Técnica de Helsinquia)
  - ▶ 1970s - Memórias associativas
  - ▶ 1982 - Primeiros artigos sobre SOM
  - ▶ 1988 - Livro sobre SOM, artigos sobre SOM no IEEE
  - ▶ 1990s - Grande divulgação
  - ▶ 1995,1997,2001 – Livro “Self Organizing Maps”
- ▶ **Inspiração**
  - ▶ Códigos para quantização de vetores
  - ▶ Memórias associativas
  - ▶ Preservar a topologia nos mapeamentos: padrões vizinhos devem ser mapeados para neurônios vizinhos

# Inspiração Neurofisiológica

---

- ▶ Observação de imagens
  - ▶ Ressonância magnética (MRI)
  - ▶ Tomografia Computadorizada (CT)
- ▶ Diferentes estímulos geram
  - ▶ Regiões de excitação
  - ▶ Organização topográfica

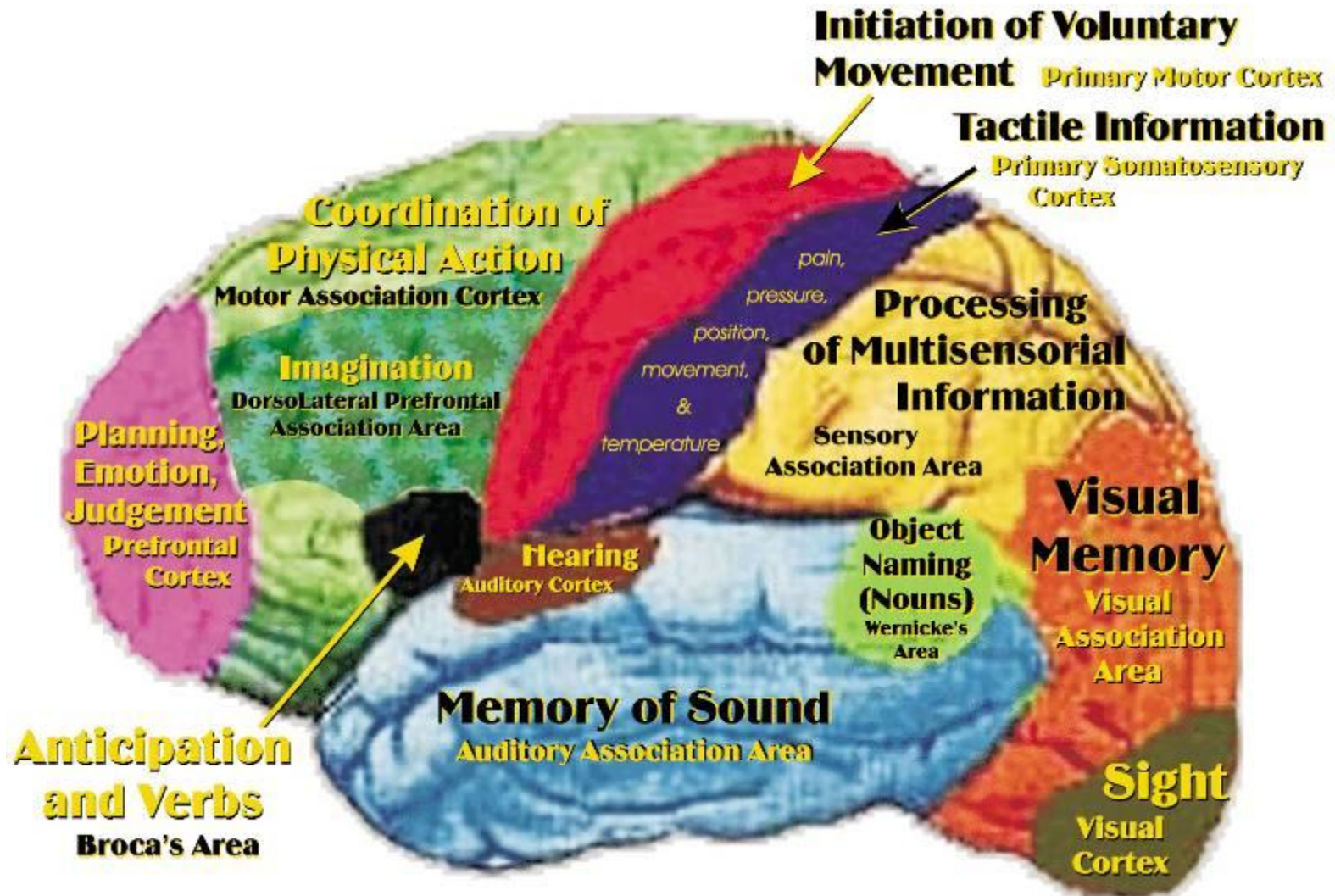


*Figura: Regiões ativas durante aplicação de acupuntura no ponto LI-5*

*Fonte: Neuro-imaging of Acupuncture Project*

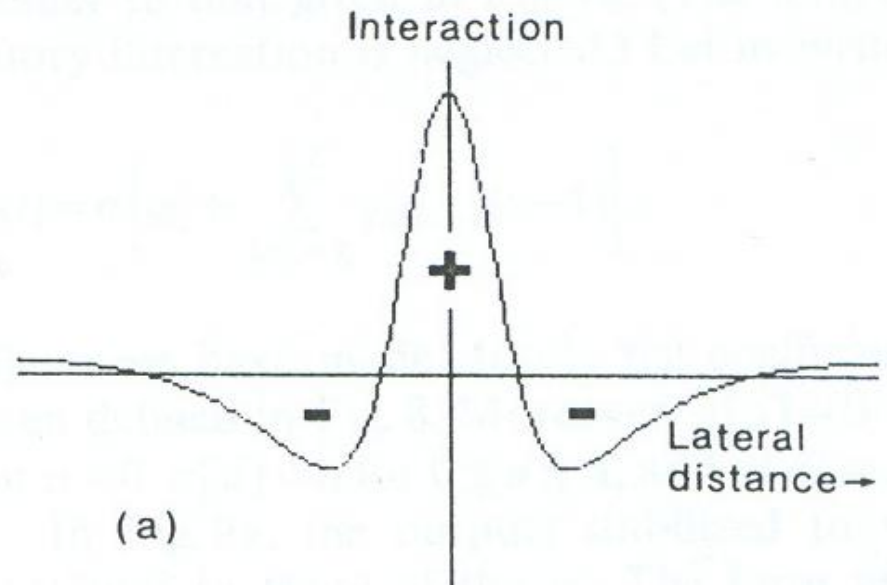


# Inspiração Neurofisiológica



# Inspiração Neurofisiológica

- ▶ Quando um neurônio é excitado, ao redor uma área entre 50 e 100  $\mu m$  também sofre excitação
- ▶ Ao redor, uma área sofre inibição para impedir a propagação do sinal a áreas não relacionadas
- A figura ilustra a interação lateral entre os neurônios



# Aprendizagem Competitiva

---

- ▶ Neurônios de saída da RNA competem entre si para se tornar ativos
- ▶ Apenas um neurônio de saída está ativo em um determinado instante
- ▶ Três elementos básicos:
  1. Neurônios com mesma estrutura, diferente pelos *pesos*, de forma que tenham *respostas diferentes* a uma entrada
  2. Um *limite* imposto sobre a força de cada neurônio
  3. Mecanismo de competição entre neurônios, de forma que um neurônio é vencedor em um dado instante.





# Aprendizagem Competitiva

---

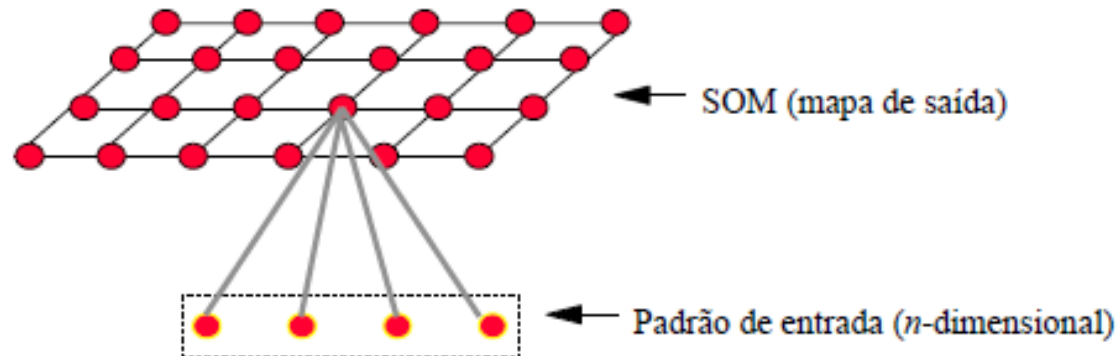
- ▶ Em cada momento o neurônio vencedor:
  - ▶ aprende a se especializar em agrupamentos de padrões similares
  - ▶ tornam-se *detectores de características* para classes diferentes de padrões de entrada
- ▶ O número de unidades de entrada define a dimensionalidade dos dados



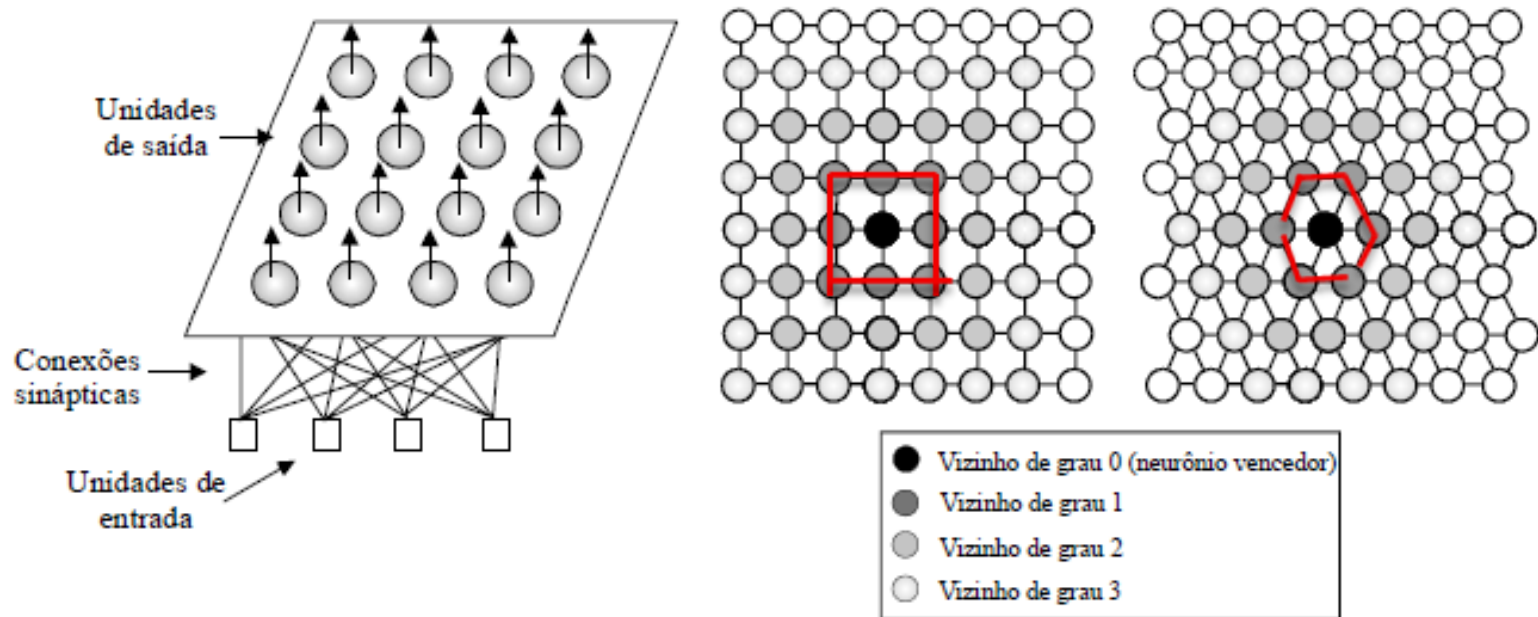
# SOM básico

---

- ▶ Neurônios (**unidades**) dispostos num grid uni ou bi-dimensional
  - ▶ Pode ser uma grid 1-dimensional (linha) ou *m-dimensional* ...
- ▶ Uma única camada
- ▶ Aprendizagem competitiva (“winner-take all”)



# Arranjo bidimensional de um mapa de Kohonen



Rede de Kohonen em arranjo bidimensional

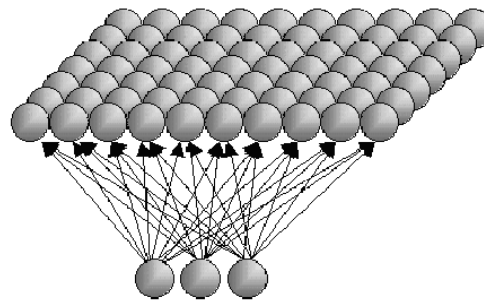
# Self Organizing Maps (um método baseado em modelo)

## Motivação

0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,1.17585,0.67585,0.64226,0.14226,0.31  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.29039,-0.20961,0.  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,1.06266,0.56266,0.  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.79816,0.29816,0.1  
0.0000,1.36521,0.86521,-0.71231,-1.23231,-1.29992,-1.79992,0.0000,-0.5000,-1.22400,-1.72400,  
0.0000,0.0000,-0.5000,-1.17435,-1.67435,-1.79825,-1.79825,-1.41995,-1.41995,0.87436,0.37436,  
0.0000,0.0000,-0.5000,-1.06272,-1.56272,-1.18937,0.68937,0.09955,0.59955,0.61335,0.11335,-0.  
0.0000,0.0000,-0.5000,1.41561,0.91561,1.43247,0.93247,1.15896,0.63896,0.32466,-0.17536,0.201  
0.0000,0.0000,-0.5000,-1.29859,-1.79859,1.14296,0.64296,1.30617,0.80617,0.62812,0.12812,-0.2  
0.0000,-1.40481,-1.90481,0.0000,-0.5000,0.0000,-0.5000,1.48325,0.98325,1.17493,0.67493,0.7  
0.0000,0.0000,-0.5000,-1.06211,-1.56211,1.36531,0.86531,1.20781,0.70781,1.17506,0.67506,0.7  
0.0000,-0.67369,-1.17369,-1.01123,-1.51123,-0.69758,-1.19758,-0.55988,-1.05988,-0.32949,-0.825  
0.0000,-1.01123,-1.51123,0.87457,0.37457,-0.44633,-0.94633,-0.06130,-0.56130,0.15969,-0.34031,  
0.0000,0.53814,0.03814,0.87478,0.37478,0.87478,0.37478,-0.87500,-1.37500,-0.19169,-0.69169,-0.  
0.0000,0.0000,-0.5000,0.0000,-0.5000,-0.44634,-0.94634,-0.06130,-0.56130,0.15969,-0.34031,  
0.0000,0.87436,0.37436,-0.87691,-1.37691,-0.87460,-1.37460,-0.51297,-1.01297,-0.34075,-0.84075,  
0.0000,0.53814,0.03814,0.0000,-0.5000,-1.17493,-1.67493,-0.26988,-0.76988,-0.20653,-0.70653,  
0.0000,0.37932,-0.12088,0.58694,0.08694,0.49800,-0.00200,0.55668,0.05668,0.34793,-0.15207,0.02  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.87457,0.37457,0.  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.87457,0.37457,0.67369,0.17369,0.29072,-0.20928,0.23  
0.0000,-0.87436,-1.37436,-0.87478,-1.37478,0.79816,0.29816,0.49534,-0.00446,0.25996,-0.24004,0.  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.84053,0.34053,0.34053,0.0000,-1.  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,1.24827,0.74827,0.81601,0.31601,0.97  
0.0000,-0.29039,-0.79039,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,1.14823,0.64823,0.35177,  
0.0000,0.0000,-0.5000,-0.87224,-1.37224,-0.87691,-1.37691,0.0000,-0.5000,0.0000,-0.5000,  
0.0000,1.17585,0.67585,0.64226,0.14226,0.0000,-0.5000,-0.87224,-1.37224,0.87478,0.37478,0.31  
0.0000,0.0000,-0.5000,0.14823,-0.35177,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,-1.  
0.0000,-0.53909,-1.03909,0.0000,-0.5000,-0.37991,-0.87991,0.87691,0.37691,0.08528,-0.41472,0.  
0.0000,-0.87478,-1.37478,-0.87478,-1.37478,0.0000,-0.5000,1.29907,0.79907,0.79887,0.29887,  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,-1.34752,-2.04752,0.0000,-0.5000,0.  
0.0000,-0.87436,-1.37436,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,-0.27975,-0.77975,  
0.0000,-1.17401,-1.67401,-0.54004,-1.04004,0.0000,-0.5000,-0.67369,-1.17369,-0.54004,-1.0400  
0.0000,0.0000,-0.5000,0.87691,0.37691,0.0000,-0.5000,0.23449,-0.26551,-1.31093,-1.81093,0.  
0.0000,-0.54004,-1.04004,0.0000,-0.5000,0.38031,-0.11849,0.41519,-0.08481,0.24073,-0.21927,  
0.0000,-0.87691,-1.37691,0.0000,-0.5000,0.64216,0.14216,0.36573,-0.13427,0.05561,-0.44439,0.  
0.0000,0.0000,-0.5000,0.09441,-0.00519,0.31531,-0.14443,0.02912,-0.47083,0.02776,-0.50776,  
0.0000,0.0000,-0.5000,-0.87436,-1.37436,-0.87478,-1.37478,0.0000,-0.5000,1.17401,0.67401,0.  
0.0000,-0.14876,-0.64876,0.47392,-0.02607,0.20521,-0.29479,0.12596,-0.36404,0.09728,-0.41272,0.  
0.0000,0.37843,-0.12157,-0.37976,-0.87976,0.0000,-0.5000,0.65085,0.15085,0.13120,-0.34880,-0.  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.16898,-0.33102,-0.14813,-0.64813,0.  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,-0.  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.11191,-0.38809,-0.  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.32949,-1.17051,0.11888,-0.38117,0.  
0.0000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.0000,-0.5000,0.

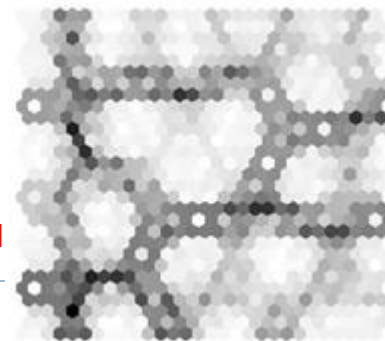
Conjunto de dados  
(n-dimensional)

Som bi-dimensional



Pós-  
processamento

Visualização dos dados  
em um espaço bi-dimensional



Matriz U

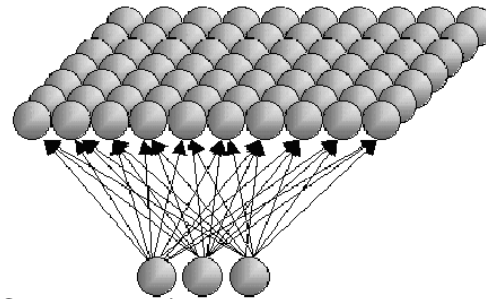
# Self Organizing Maps (um método baseado em modelo)

## Motivação

0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,1.17585,0.67585,0.64226,0.14226,0.31  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.29039,-0.20961,0.  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,1.06266,0.56266,0.  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.79816,0.29816,0.  
0.00000,1.36521,0.86521,-0.71231,-1.23231,-1.29992,-1.79992,0.00000,-0.50000,-1.22400,-1.72400,  
0.00000,0.00000,-0.50000,-1.17435,-1.67435,-1.79825,-1.79825,-1.41991,-1.41991,0.87436,0.37436,  
0.00000,0.00000,-0.50000,-1.06272,-1.56272,-1.18937,0.68937,0.59955,0.59955,0.61335,0.11335,-0.  
0.00000,0.00000,-0.50000,1.43161,0.93161,1.43247,0.93247,1.15896,0.63896,0.23466,-0.17534,0.201  
0.00000,0.00000,-0.50000,-1.29859,-1.79859,1.14296,0.64296,1.30617,0.80617,0.62812,0.12812,-0.2  
0.00000,-1.40481,-1.90481,0.00000,-0.50000,0.00000,-0.50000,1.48325,0.98325,1.17493,0.67493,0.7  
0.00000,0.00000,-0.50000,-1.06211,-1.56211,1.36531,0.86531,1.20781,0.70781,1.17506,0.67506,0.7  
0.00000,-0.67369,-1.17369,-1.01123,-1.51123,-0.69758,-1.19758,-0.55988,-1.05988,-0.32949,-0.825  
0.00000,-1.01123,-1.51123,0.87457,0.37457,-0.44633,-0.94633,-0.06130,-0.56130,0.15969,-0.34031,  
0.00000,0.53814,0.03814,0.87478,0.37478,0.87478,0.37478,-0.87500,-1.37500,-0.19169,-0.69169,-0.  
0.00000,0.00000,-0.50000,0.00000,-0.50000,-0.44634,-0.94634,-0.47403,-0.97403,-0.24855,-0.74855,  
0.00000,0.87436,0.37436,-0.87691,-1.37691,-0.87460,-1.37460,-0.51297,-1.01297,-0.34075,-0.84075,  
0.00000,0.53814,0.03814,0.00000,-0.50000,-1.17493,-1.67493,-0.26948,-0.76948,-0.20653,-0.70653,  
0.00000,0.37932,-0.12088,0.58694,0.08694,0.49800,-0.00200,0.55668,0.05668,0.34793,-0.15207,0.02  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.87457,0.37457,0.  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.87457,0.37457,0.67341,0.17341,0.29072,-0.29072,0.23  
0.00000,-0.87436,-1.37436,-0.87478,-1.37478,0.79816,0.29816,0.49534,-0.00446,0.25996,-0.24004,0.  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.84053,0.34053,0.34053,0.1  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,1.24827,0.74827,0.81601,0.31601,0.97  
0.00000,-0.29039,-0.79039,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,1.14823,0.15177,0.  
0.00000,0.00000,-0.50000,-0.87224,-1.37224,-0.87691,-1.37691,0.00000,-0.50000,0.00000,-0.50000,  
0.00000,1.17585,0.67585,0.87478,0.37478,0.00000,-0.50000,-0.87224,-1.37224,0.87478,0.37478,0.31  
0.00000,0.00000,-0.50000,0.14823,-0.35177,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,-1.  
0.00000,-0.53909,-1.03909,0.00000,-0.50000,-0.37991,-0.87991,0.87691,0.37691,0.08528,-0.41472,0.  
0.00000,-0.87478,-1.37478,-0.87478,-1.37478,0.00000,-0.50000,1.29907,0.79907,0.79907,0.87687,0.  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,-1.34752,-2.04752,0.00000,-0.50000,0.  
0.00000,-0.87436,-1.37436,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,-0.27975,-0.77975,0.  
0.00000,-1.17401,-1.67401,-0.54004,-1.04004,0.00000,-0.50000,-0.67369,-1.17369,-0.54004,-1.04004,  
0.00000,0.00000,-0.50000,0.87691,0.37691,0.00000,-0.50000,0.23449,-0.26551,-1.31093,-1.81093,0.  
0.00000,-0.54004,-1.04004,0.00000,-0.50000,0.87031,-0.11949,0.41519,-0.08481,0.24073,-0.21927,  
0.00000,-0.87691,-1.37691,0.00000,-0.50000,0.64216,0.14216,0.36573,-0.13427,0.05561,-0.44439,0.  
0.00000,0.00000,-0.50000,0.09441,-0.00519,0.31531,-0.14443,0.02912,-0.47083,0.02776,-0.50776,0.  
0.00000,0.00000,-0.50000,-0.87436,-1.37436,-0.87478,-1.37478,0.00000,-0.50000,1.17401,0.67401,0.  
0.00000,-0.14876,-0.64876,0.47392,-0.02607,0.20521,-0.29479,0.15596,-0.36404,0.09729,-0.41272,0.  
0.00000,0.37843,-0.12157,-0.37976,-0.87976,0.00000,-0.50000,0.65095,0.15095,0.15102,-0.34880,-0.  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.16898,-0.33102,-0.14813,-0.64813,-0.64813,0.  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,-0.  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.11191,-0.38809,-0.38809,-0.  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.32949,-1.17511,0.11888,-0.38117,0.  
0.00000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.00000,-0.50000,0.

Conjunto de dados  
(n-dimensional)

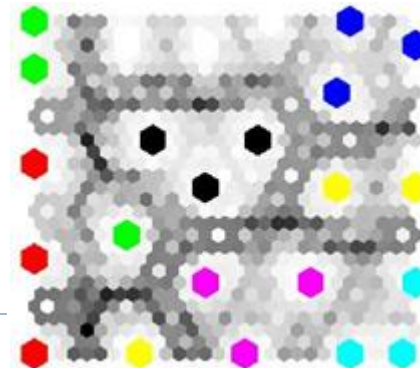
Som bi-dimensional



(Costa, 1999)

Pós-  
processamento

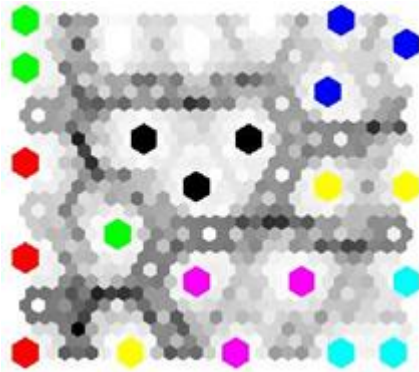
Quantização do espaço  
dos dados.



Matriz U (rotulada)

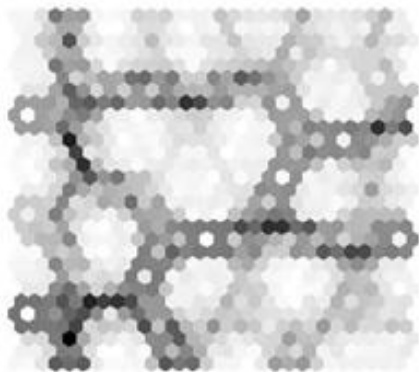
# Self Organizing Maps - Objetivos

---



- ▶ **Aproximação do espaço de entrada:**

Um dos objetivos de um SOM é representar um conjunto grande de vetores de entrada, por meio de um conjunto menor de vetores localizados em um espaço de dimensão mais baixa. Ou seja, realizar a quantização do espaço e a redução de dimensão.



- ▶ **Visualização do conjunto de dados:**

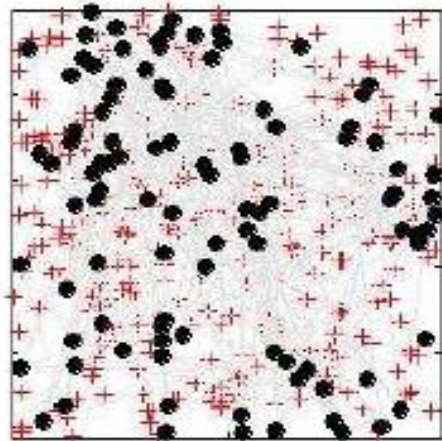
Quando é este o objetivo pode-se desconsiderar a necessidade de quantização do espaço.



# Self Organizing Maps - espaços

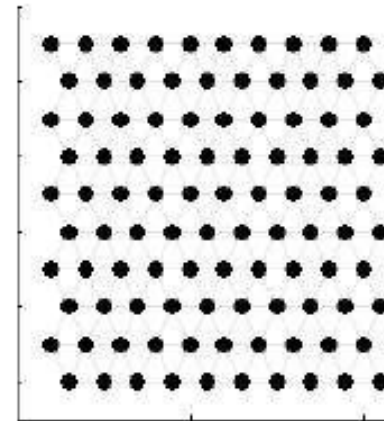
**Espaço de entrada:**  
as distâncias são medidas  
vetorialmente.

Organização vetorial  
com relações de  
vizinhança (no espaço  
vetorial).



(SomToolbox – Matlab)

**Espaço de saída:**  
as distâncias são medidas  
matricialmente.

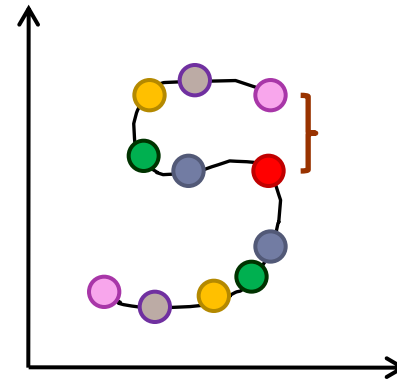
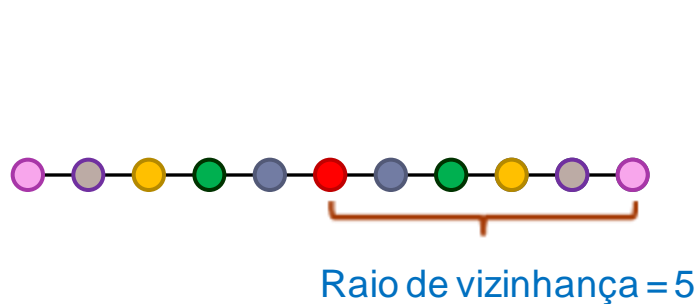


(SomToolbox – Matlab)

Organização matricial  
com relações de  
vizinhança (matricial).

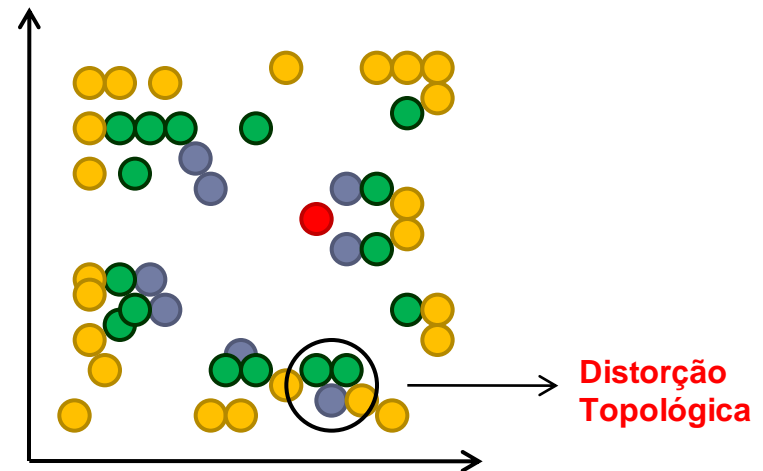
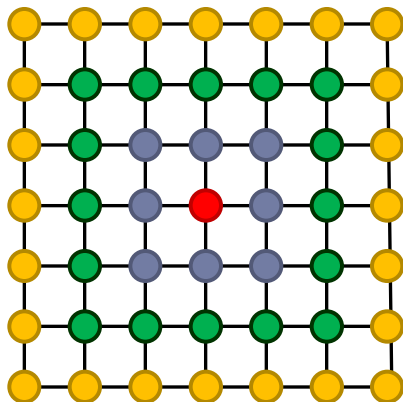
- O **espaço de entrada** é determinado pelo conjunto de dados a ser explorado pelo SOM.
- O **espaço de saída** é determinado pelo projetista da rede neural.

# Self Organizing Maps - vizinhanças



Vizinhança matricial unidimensional X vizinhança vetorial bidimensional.

Existem outras topologias de vizinhança matricial!

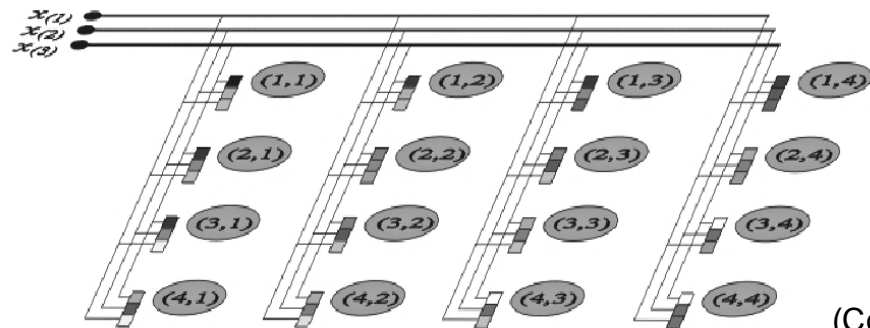


Vizinhança matricial bidimensional X vizinhança vetorial bidimensional



# Self Organizing Maps - arquitetura

- ▶ Número de neurônio na camada de entrada.
- ▶ Número de neurônios na camada de saída (tamanho mapa)
- ▶ Tipo de vizinhança topológica (dimensão e lattice) (forma do mapa)
- ▶ Função de vizinhança (...)



(Costa, 1999)

- **3** neurônios na camada de entrada (espaço vetorial tridimensional)
- **16** neurônios na camada de saída
- mapa **bidimensional** (espaço matricial bidimensional)
- lattice **retangular**

# Self Organizing Maps - Propriedades

---

- ▶ Os mapas auto-organizáveis possuem duas propriedades principais:
  - ▶ 1. Quantização vetorial (redução do espaço de entrada)
  - ▶ 2. Agrupamento de padrões similares em regiões geograficamente próximas



# Os dados, a rede e a inicialização

## ■ Seja:

- $X = \{x_1, x_2, \dots, x_n\}$  o conjunto de dados de treino, com dimensão  $m$ .

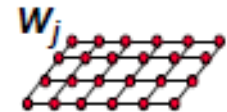
- $x_i = [x_i^1, x_i^2, \dots, x_i^m]^T$ , onde  $x_{ij}$  são escalares de valor real.



- $W$  um grid de  $p \times q$  unidades  $w_j$

- $w_j = [w_j^1, w_j^2, \dots, w_j^m]^T$

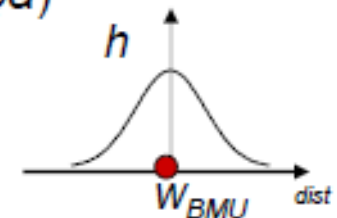
- Valores de  $w_j$  escolhidos aleatoriamente na zona dos dados



- $h(w_i, w_j, r)$  uma função real (denominada de vizinhança)

- Quando  $\|w_i - w_j\|_{(\text{grid})} \rightarrow \infty$ ,  $h(w_i, w_j, r) \rightarrow 0$

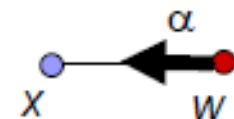
- $r$  determina o raio (zona de influência)



- $\alpha$  a taxa de aprendizagem

- $0 \leq \alpha \leq 1$

- A começa grande e vai diminuindo ao longo do treino



# Self Organizing Maps – algoritmo de aprendizado (sequencial)

---

- ▶ **Passo 0:**
  - ▶ Determine a arquitetura da rede neural
  - ▶ Inicialize os pesos  $w_{ij}$ . – **posicionar os neurônio no espaço vetorial**
  - ▶ Determine os parâmetros da taxa de aprendizado (valor inicial e função de atualização).
  - ▶ Definir função de vizinhança e raio
- ▶ **Passo 1:** Enquanto condição de parada é falsa, execute os passos 2-8.
  - ▶ **Passo 2:** Para cada vetor de entrada  $\mathbf{x}$ , execute os passos 3-5;
    - ▶ Passo 3: Para cada  $\mathbf{j}$ , compute:  **$D(\mathbf{j}) = \sum_i (w_{ij} - x_i)$** .
    - ▶ Passo 4: Encontre o  $\mathbf{J}$  tal que  $D(\mathbf{J})$  seja mínimo.
    - ▶ Passo 5: Para todas as unidades  $\mathbf{j}$  dentro de uma vizinhança específica de  $\mathbf{J}$ , e para todo  $i$ :  **$w_i(\text{new}) = w_i(\text{old}) + \alpha h(w_j, w_i, r)[x_i - w_i(\text{old})]$** .
  - ▶ **Passo 6:** Altere a taxa de aprendizado (se for o caso)
  - ▶ **Passo 7:** Reduza o raio de vizinhança (se for o caso)
  - ▶ **Passo 8:** Teste a condição de parada.



# Inicialização de pesos

---

## ▶ Aleatório

- ▶ posiciona os neurônios do mapa de forma aleatória, dentro do espaço dos dados.

## ▶ Linear

- ▶ usa os componentes principais da matriz de autocorrelação do conjunto de dados  $X$ . As posições dos neurônios são determinadas de forma a distribuir-se na direção dos espaços dos autovetores correspondentes aos maiores autovalores encontrados.
- ▶ os neurônios se distribuem nas direções de maior variância dos dados.

## ▶ Usando conhecimento *a priori*

- ▶ usa algum conhecimento sobre os dados para posicionar os neurônio em locais adequados dentro do espaço dos dados.



# Principais decisões a tomar:

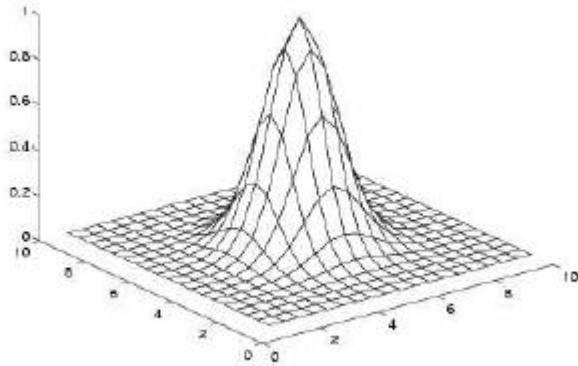
---

- ▶ Quantas unidades ? Que tipo de grid ?
  - ▶ Que dimensão ?
  - ▶ Quantas iterações ?
  - ▶ Que tipo de função de vizinhança ?
  - ▶ Que valor inicial para  $r$  ? *Que valor final ?*
  - ▶ Que valor para  $\alpha$  inicial ?
- 

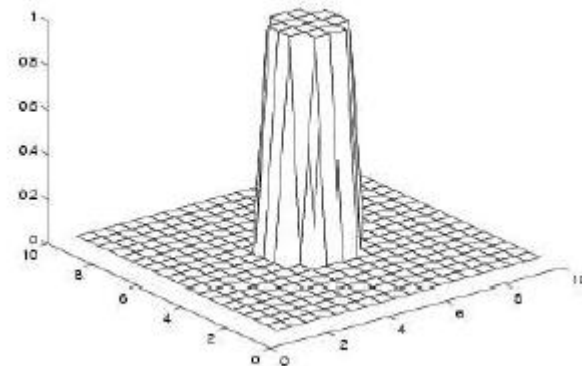


# Analizando funções de vizinhança

---



Gaussiana ( $g(j)$ )



Bubble ( $b(j)$ )

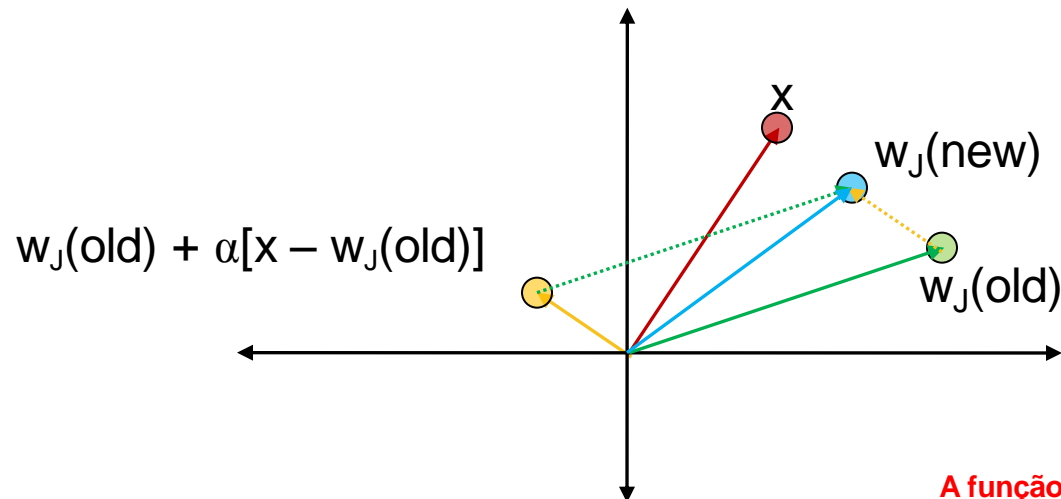
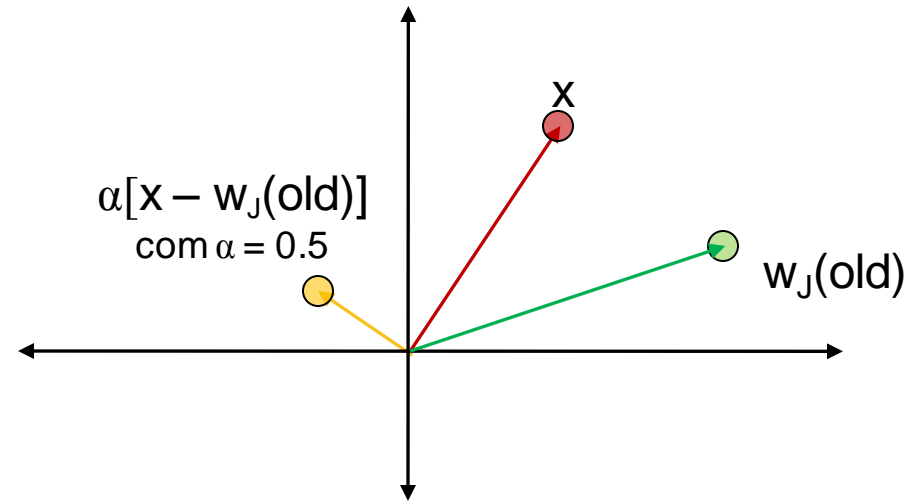
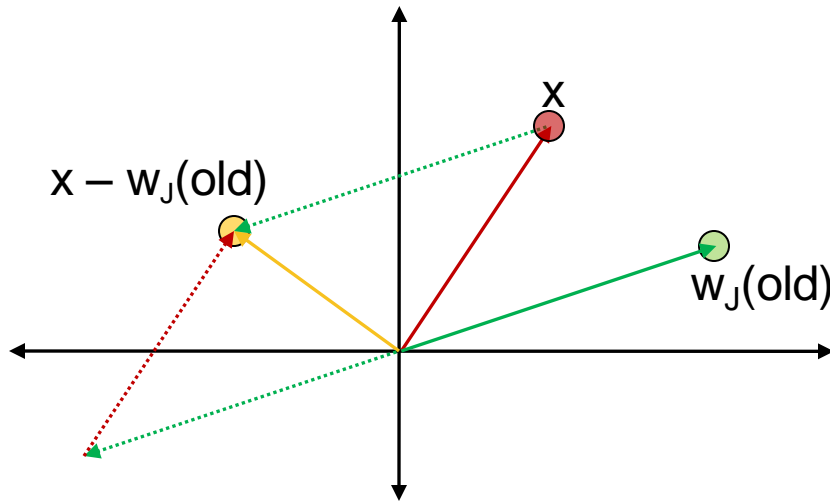
$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + g(j)\alpha[x_i - w_{ij}(\text{old})].$$

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + b(j)\alpha[x_i - w_{ij}(\text{old})].$$

(Fausett, 1994)



# Estudando a taxa de aprendizado

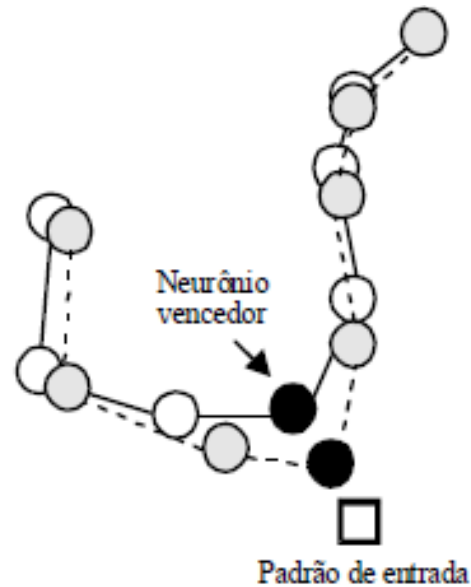


A função de vizinhança é mais um fator que opera da mesma maneira (vetorial) que a taxa de aprendizado.



# Um passo de ajuste no arranjo unidimensional

---



—	Conformação do mapa anterior ao processo competitivo
- - -	Conformação do mapa posterior ao processo competitivo

Ajuste do neurônio vencedor e de seus vizinhos mais próximos



# Aprendizagem

---

- ▶ As unidades são **puxadas para as posições dos dados, arrastando consigo as suas vizinhas no espaço de saída**
- ▶ SOM  $\approx$  **superfície de borracha, esticada** e torcida de modo a passar pelos padrões de dados (ou pelo menos a ficar perto)



# BMU- Best Matching Unit (Unidade vencedora)

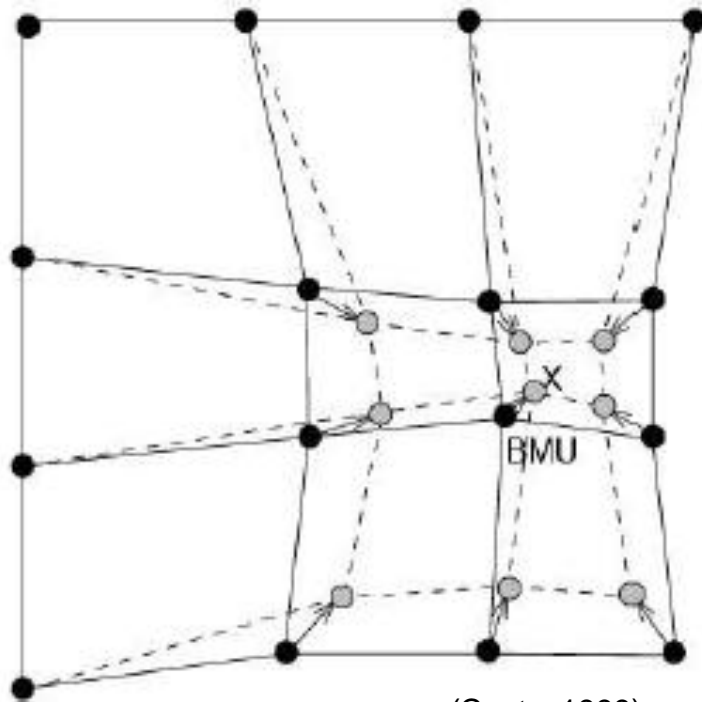
---

- ▶ Padrões de entrada são comparados com todas as unidades; a mais próxima é considerada a **BMU**.
- ▶ Consideramos que o padrão de entrada é **mapeado** para a BMU.
- ▶ **A BMU atualiza-se (de modo a aproximar-se mais do padrão de dados que representa), e os seus vizinhos atualizam-se também um pouco**
- ▶ Há sempre uma ligeira diferença entre os dados e as BMUs que as representam. Essa diferença é o **erro de quantização**.



# Analizando o ajuste de pesos

---

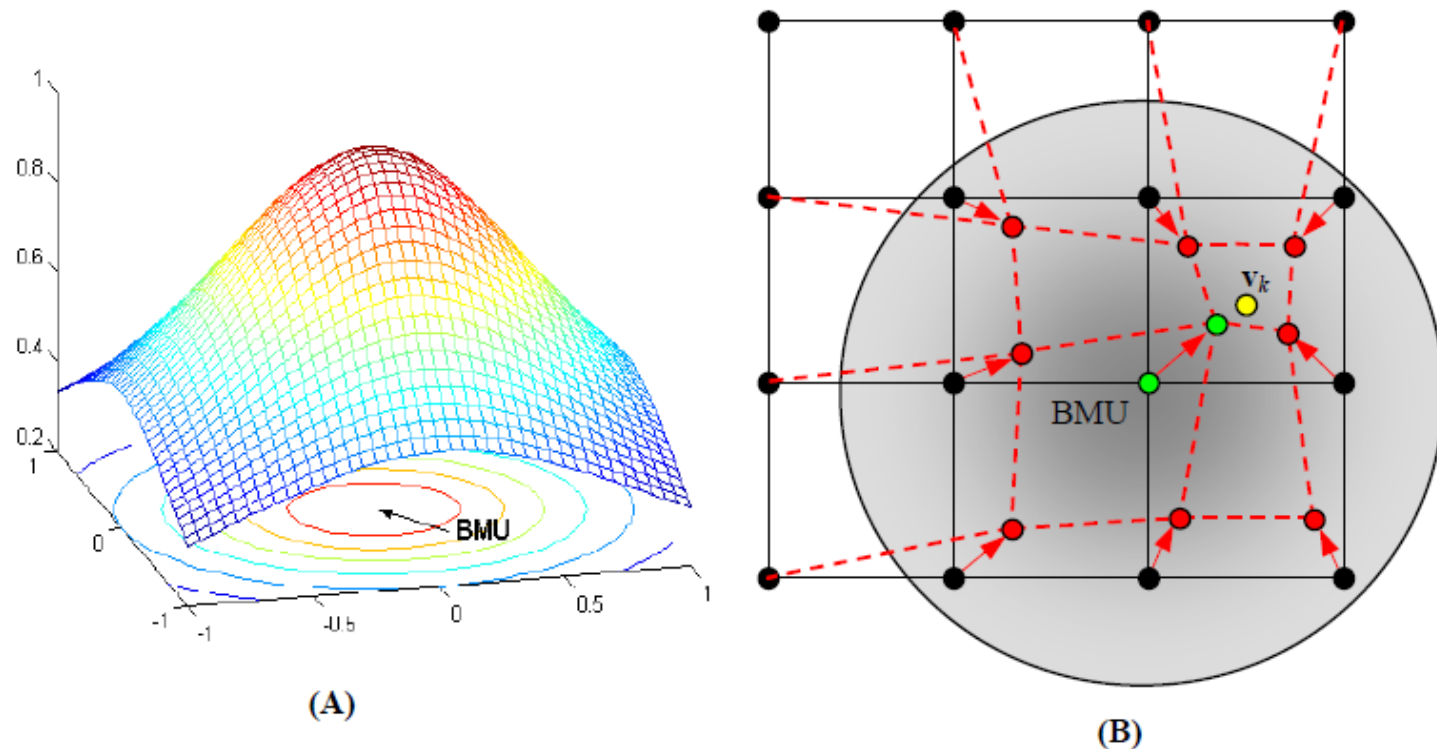


(Costa, 1999)

$$w_{ij}(\text{new}) = w_{ij}(\text{old}) + \alpha[x_i - w_{ij}(\text{old})]$$

j varia na vizinhança do neurônio BMU

# Adaptação dos pesos de um SOM



**Figura 3-5 – Ilustração da adaptação dos pesos de um SOM para a apresentação de um único padrão em  $\mathbb{R}^2$ . Em (A) uma representação da função  $h_{ci}$  sobre um mapa bidimensional cuja projeção pode ser vista em (B). Quanto mais próximo um neurônio encontra-se do BMU, isto é, quanto menor a distância  $\|r_c - r_i\|$ , maior é a adaptação aplicada ao neurônio. O neurônio com maior adaptação é, obviamente, o BMU.**

# Outras decisões e conceitos

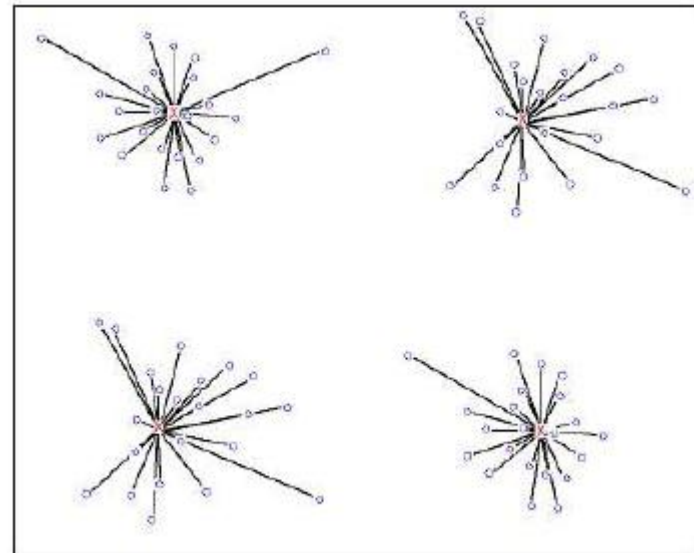
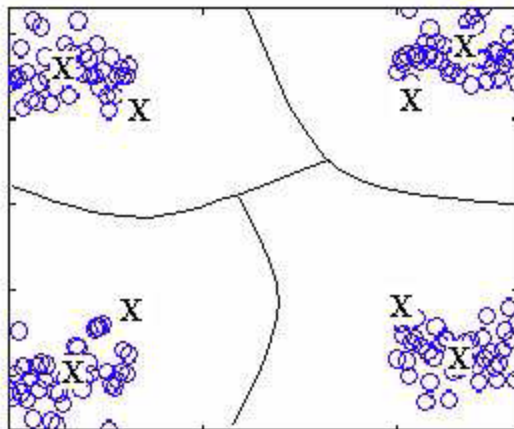
---

- ▶ Número de épocas
  - ▶ fase de ordenação
  - ▶ fase de sintonização (ajuste fino)
- ▶ Função de atualização da taxa de aprendizado
- ▶ Função de atualização do raio de vizinhança
- ▶ Algoritmo de treinamento em lote (*batch*)
- BMU – Best Matching Unit



# Self Organizing Maps – qualidade

O número de neurônios no mapa deve ser menor (bem menor) que o número de dados no conjunto de dados estudados.



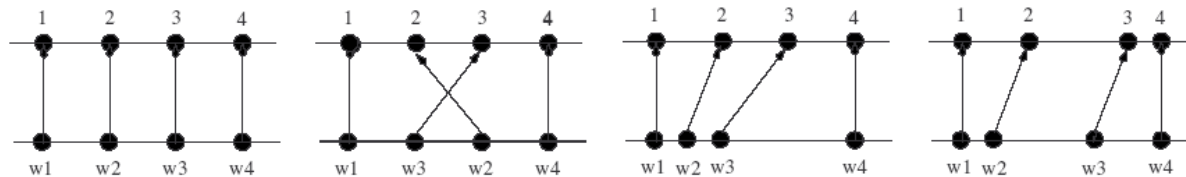
Medida de Qualidade:  
Erro de Quantização.

$$E_q = \frac{1}{N} \sum_{i=1}^N \|x_i - w_{bi}\|$$

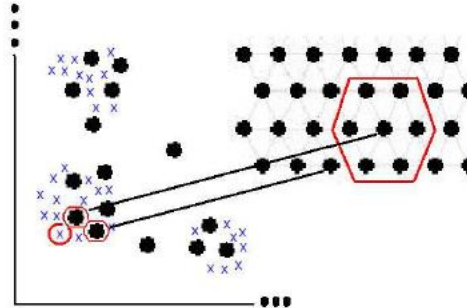
# Self Organizing Maps – qualidade

## ► Distorções topológicas

Existem outras formas de  
analisada a qualidade da  
preservação topológica!



## ► Erro topológico



$$\xi = \frac{1}{n} \sum_{k=1}^n u(x_k), \text{ em que } u(x_k) = \begin{cases} 1, & \text{se o primeiro e o segundo neurônios} \\ & \text{vencedores não forem adjacentes} \\ 0, & \text{caso contrário} \end{cases}$$



# Discriminação dos Agrupamento

---

- ▶ dada a conformação final de 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.



# Discriminação dos Agrupamento

---

- ▶ recebeu a denominação de matriz por ter sido proposta no caso de **mapas bidimensionais**, sendo que o grau de similaridade é plotado na terceira dimensão gerando uma superfície em relevo em 3D.
- ▶ para o caso de **mapas unidimensionais**, tem-se o vetor-U.
- ▶ 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**.
- ▶ como utilizar o mapa de Kohonen, após a fase de treinamento não-supervisionado e depois de ter as classes devidamente discriminadas, para classificação de padrões?
  - ▶ VEREMOS MAIS ADIANTE

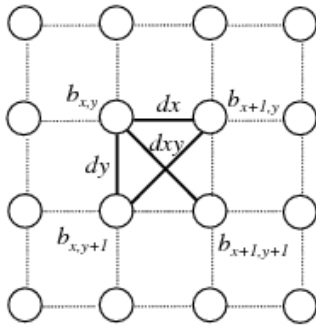


# Pós-processamento – Matriz U

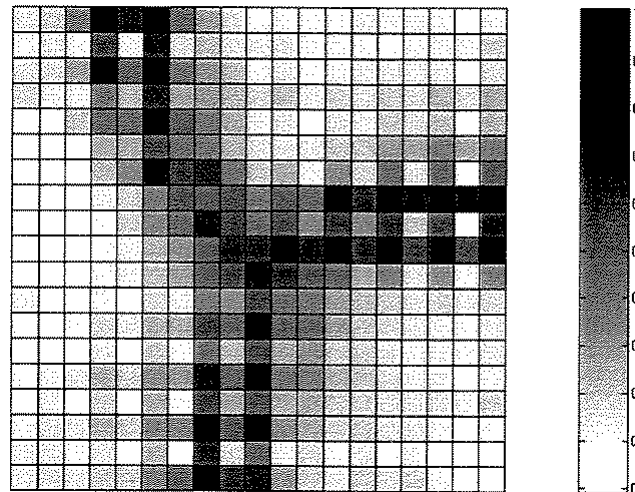
---

- ▶ Um método de visualização de um SOM treinando, denominado a matriz de distâncias unificadas, ou U-matrix, foi desenvolvido por A. Ultsch (1993) com o objetivo de permitir a **detecção visual das relações** topológicas dos neurônios.
- ▶ O resultado é uma imagem  $f(x,y)$ , na qual as coordenadas de cada pixel  $(x,y)$  são derivadas das coordenadas dos neurônios no grid do mapa, ex.
  - ▶  $(1,1), (1,2), \dots, (X,Y) \rightarrow (1,1), (1,2), \dots, (2*X-1, 2*Y-1)$
- ▶ A intensidade de cada pixel na imagem  $f(x,y)$  corresponde
- ▶ a uma distância calculada

# Pós-processamento – Matriz U



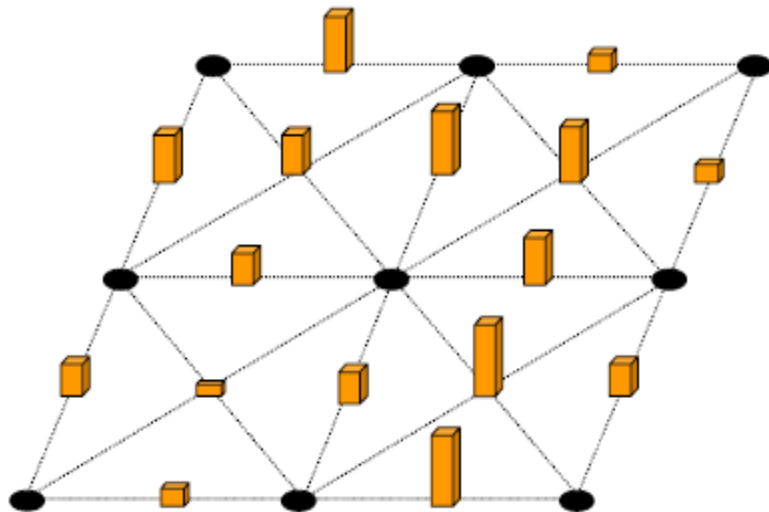
$$\begin{bmatrix} du(0,0) & dx(0,0) & du(1,0) & \dots & du(X-1,0) \\ dy(0,0) & dxy(0,0) & dy(1,0) & \dots & dy(X-1,0) \\ du(0,1) & dx(0,1) & du(1,1) & \dots & du(X-1,1) \\ dy(0,1) & dxy(0,1) & dy(1,1) & \dots & dy(X-1,1) \\ \dots & \dots & \dots & \dots & \dots \\ du(0,Y-1) & du(0,Y-1) & du(1,Y-1) & \dots & du(X-1,Y-1) \end{bmatrix}$$



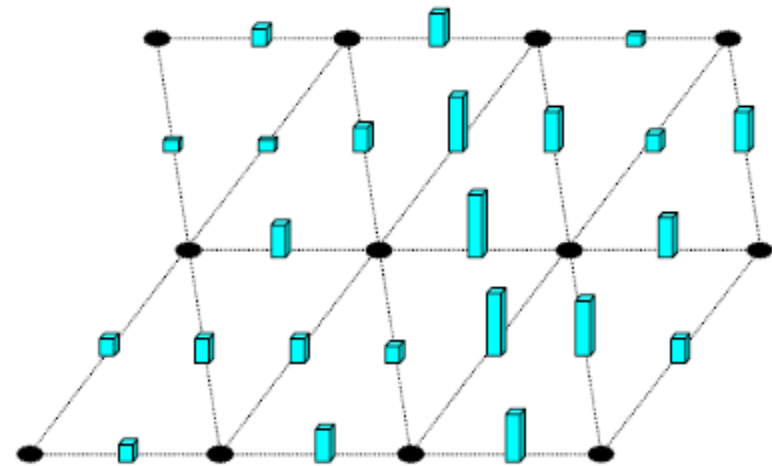
(Costa, 1999)

# Pós-processamento – Matriz U

---



(A)



(B)

Figura 3-12 – Exemplo da matriz-U num arranjo retangular (A) e hexagonal (B). No caso (A), a composição da distância nas diagonais é obtida pela média aritmética das 2 diagonais envolvidas.

# Pós-processamento – Matriz U

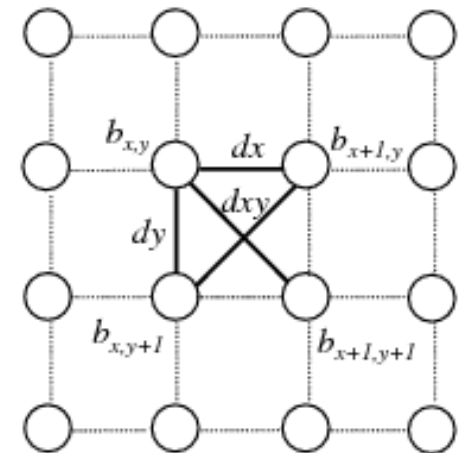
$$dx(x, y) = \|b_{x,y} - b_{x+1,y}\| = \sqrt{\sum_i (w_{i_{x,y}} - w_{i_{x+1,y}})^2}$$

$$\begin{bmatrix} du(0,0) & dx(0,0) & du(1,0) & \dots & du(X-1,0) \\ dy(0,0) & dxy(0,0) & dy(1,0) & \dots & dy(X-1,0) \\ du(0,1) & dx(0,1) & du(1,1) & \dots & du(X-1,1) \\ dy(0,1) & dxy(0,1) & dy(1,1) & \dots & dy(X-1,1) \\ \dots & \dots & \dots & \dots & \dots \\ du(0,Y-1) & dx(0,Y-1) & du(1,Y-1) & \dots & du(X-1,Y-1) \end{bmatrix}$$

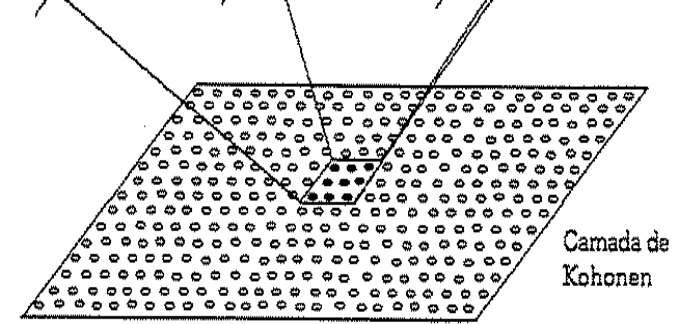
$$dy(x, y) = \|b_{x,y} - b_{x,y+1}\| = \sqrt{\sum_i (w_{i_{x,y}} - w_{i_{x,y+1}})^2}$$

$$dxy(x, y) = \frac{1}{2} \left( \frac{\|b_{x,y} - b_{x+1,y+1}\|}{\sqrt{2}} + \frac{\|b_{x,y+1} - b_{x+1,y}\|}{\sqrt{2}} \right)$$

$$= \frac{1}{2\sqrt{2}} \left[ \sqrt{\sum_i (w_{i_{x,y}} - w_{i_{x+1,y+1}})^2} + \sqrt{\sum_i (w_{i_{x,y+1}} - w_{i_{x+1,y}})^2} \right]$$



# Cálculo de $du(x,y)$



- ▶ Existem pelo menos duas possibilidades para o cálculo de  $du(x,y)$ , pode-se usar o valor **médio** ou a **mediana** dos elementos vizinhos
- ▶ Seja  $C = (c_1, c_2, \dots, c_k)$  os valores dos elementos circunvizinhos de  $U_{2 \times 2} y$  aparecendo na forma de um vetor ordenado com cardinalidade  $k$  ( $k = |C|$ ). Considerando uma topologia retangular,  $k = 8$ .

$$du(x, y) = \begin{cases} c[(k+1)/2] & \text{se } k \text{ for ímpar} \\ \frac{c(k/2) + c[(k+1)/2]}{2} & \text{se } k \text{ for par} \end{cases}$$

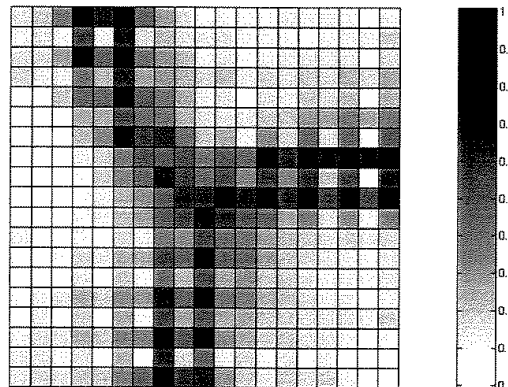
- ▶ Onde  $c(k), k=1, 2, \dots, K, K \leq 8$ , denotam os elementos circunvizinhos ordenados de forma crescente de magnitude

# Cálculo de $du(x,y)$

- ▶ No caso de  $du(x,y)$  ser o valor médio de  $C$ , teremos

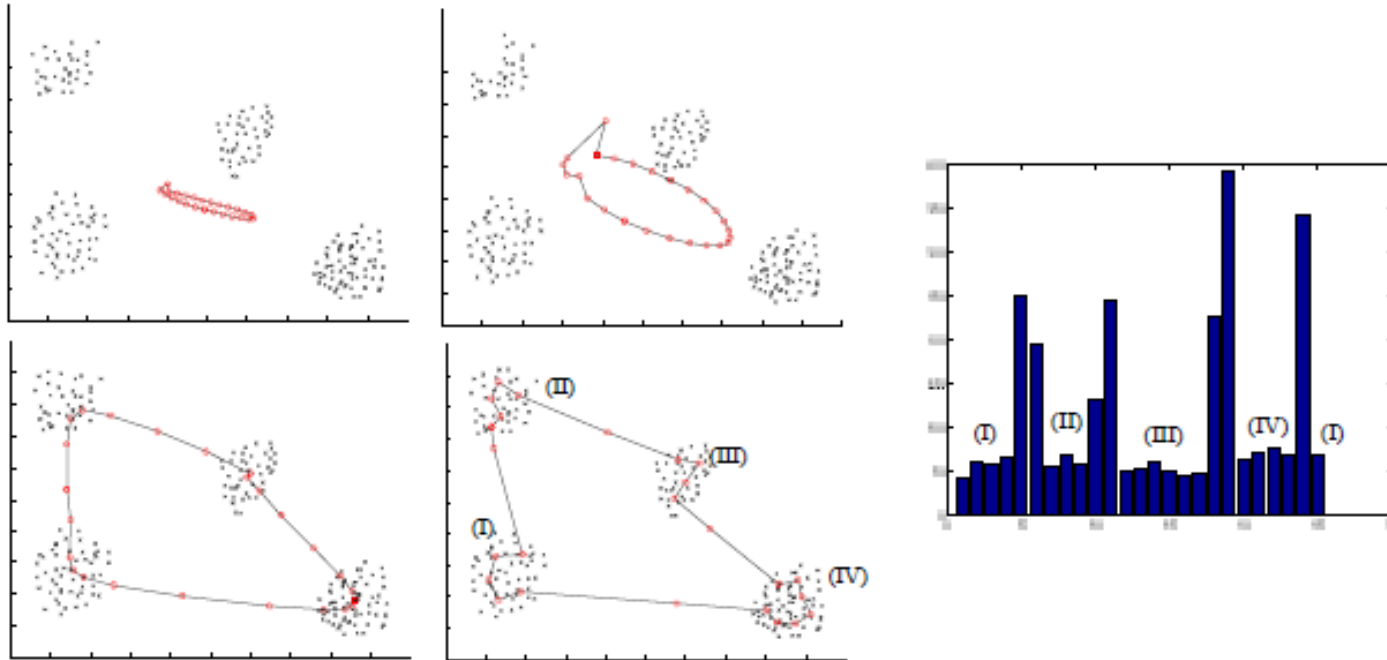
$$du(x, y) = \tilde{c} = \frac{1}{k} \sum_{i=1}^k c_i.$$

- ▶ Resultado obtido para o SOM de tamanho  $10 \times 10$ , treinado com 1000 iterações do algoritmo em lote, e inicialização de forma linear. Note que as distâncias foram escalonadas linearmente, de forma que a distância máxima seja  $1$  e a mínima  $0$ .
- ▶ Analisando a figura percebe-se a existência de três grandes regiões.





# Pós-processamento – Matriz U

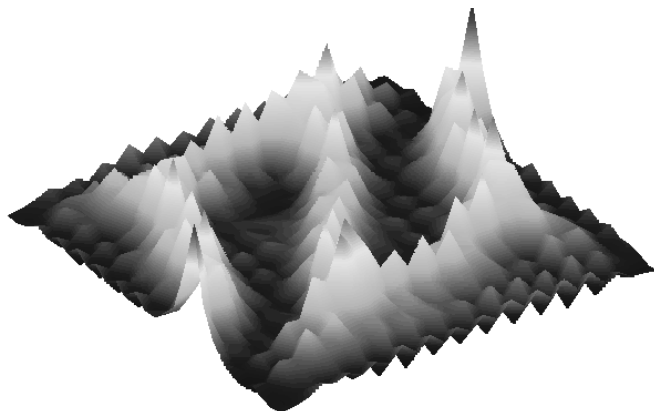
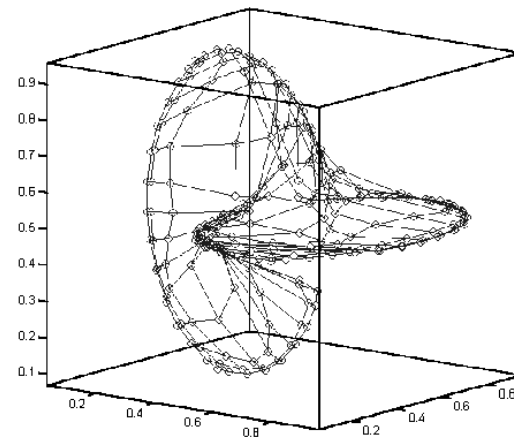
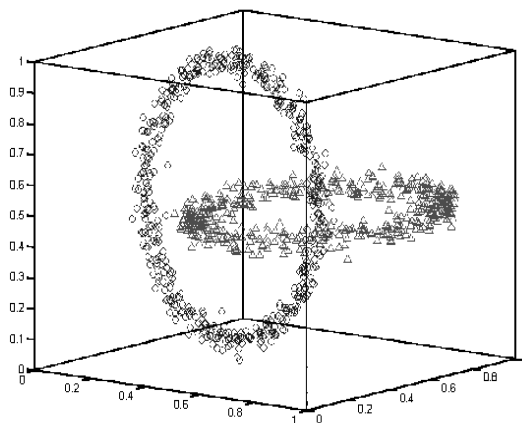


Determinação automática do número de agrupamentos com base no vetor-U para arranjo unidimensional

# Pós-processamento – Matriz U

---

## Conjunto de dados Chain Link.



---

(Costa, 1999)

# Exemplo

---

- ▶ Usa o SOM para agrupar 4 vetores
  - ▶  $(1,1,0,0)$ ,  $(0,0,0,1)$ ,  $(1,0,0,0)$  e  $(0,0,1,1)$
- ▶ Suponhamos que o número máximo de clusters podem ser formados são  $m = 2$  e a taxa de aprendizagem é  $\alpha(0) = 0.6$
- ▶  $\alpha(t+1) = 0.5 \alpha(t)$



# Exemplo

---

- ▶ Passo 0) Raio inicial  $R = 0$ , taxa de aprendizagem inicial  $\alpha(0) = 0.6$
- ▶ Inicialize a matriz de pesos

$w_1$	$w_2$
0.2	0.8
0.6	0.4
0.5	0.7
0.9	0.3



# Exemplo

## ► Passo 1) Começa o treinamento

### ► Passo 2) Para o primeiro vetor (1,1,0,0), faça Passos 3-5

#### □ Passo 3)

□  $D(1) = (0.2-1)^2 + (0.6-1)^2 + (0.5-0)^2 + (0.9-0)^2 = 1.86$

□  $D(2) = (0.8-1)^2 + (0.4-1)^2 + (0.7-0)^2 + (0.3-0)^2 = 0.98$

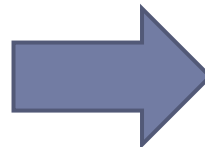
#### □ Passo 4) O vetor de entrada é mais próximo ao nó 2

□ Então,  $J = 2$

#### □ Passo 5) Os pesos da unidade vencedora é atualizado

□  $w_{i2}(\text{new}) = w_{i2}(\text{old}) + 0.6 * [x_i - w_{i2}(\text{old})]$

$w_1$	$w_2$
0.2	0.8
0.6	0.4
0.5	0.7
0.9	0.3



$w_1$	$w_2$
0.2	0.92
0.6	0.76
0.5	0.28
0.9	0.12

# Exemplo

## ► Passo 1) Começa o treinamento

### ► Passo 2) Para o primeiro vetor (0,0,0,1), faça Passos 3-5

#### □ Passo 3)

□  $D(1) = (0.2-0)^2 + (0.6-0)^2 + (0.5-0)^2 + (0.9-1)^2 = 0.66$

□  $D(2) = (0.92-0)^2 + (0.76-0)^2 + (0.28-0)^2 + (0.12-1)^2 = 2.2768$

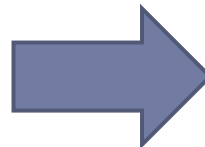
#### □ Passo 4) O vetor de entrada é mais próximo ao nó 1

□ Então,  $J = 1$

#### □ Passo 5) Os pesos da unidade vencedora é atualizado

□  $w_{i1}(\text{new}) = w_{i1}(\text{old}) + 0.6 * [x_i - w_{i1}(\text{old})]$

$w_1$	$w_2$
0.08	0.92
0.24	0.76
0.2	0.28
0.96	0.12



$w_1$	$w_2$
0.2	0.92
0.6	0.76
0.5	0.28
0.9	0.12

# Exemplo

## ► Passo 1) Começa o treinamento

### ► Passo 2) Para o primeiro vetor (1,0,0,0), faça Passos 3-5

#### □ Passo 3)

□  $D(1) = (0.08-1)^2 + (0.24-0)^2 + (0.20-0)^2 + (0.96-0)^2 = 1.8656$

□  $D(2) = (0.92-1)^2 + (0.76-0)^2 + (0.28-0)^2 + (0.12-0)^2 = 0.6768$

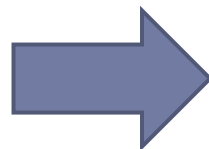
#### □ Passo 4) O vetor de entrada é mais próximo ao nó 2

□ Então,  $J = 2$

#### □ Passo 5) Os pesos da unidade vencedora é atualizado

□  $w_{i2}(\text{new}) = w_{i2}(\text{old}) + 0.6 * [x_i - w_{i2}(\text{old})]$

$w_1$	$w_2$
0.08	0.92
0.24	0.76
0.2	0.28
0.96	0.12



$w_1$	$w_2$
0.08	0.968
0.24	0.304
0.5	0.112
0.9	0.048

# Exemplo

## ► Passo 1) Começa o treinamento

### ► Passo 2) Para o primeiro vetor (0,0,1,1), faça Passos 3-5

#### □ Passo 3)

□  $D(1) = (0.08-0)^2 + (0.24-0)^2 + (0.20-1)^2 + (0.96-1)^2 = 0.7056$

□  $D(2) = (0.968-0)^2 + (0.304-0)^2 + (0.112-1)^2 + (0.048-1)^2 = 2.7243$

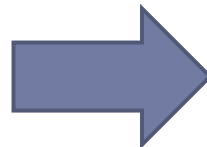
#### □ Passo 4) O vetor de entrada é mais próximo do nó 1

□ Então,  $J = 1$

#### □ Passo 5) Os pesos da unidade vencedora é atualizado

□  $w_{i1}(\text{new}) = w_{i1}(\text{old}) + 0.6 * [x_i - w_{i1}(\text{old})]$

$w_1$	$w_2$
0.08	0.968
0.24	0.304
0.2	0.112
0.96	0.048



$w_1$	$w_2$
0.032	0.968
0.096	0.304
0.680	0.112
0.984	0.048



# Exemplo

## ► Passo 6) Reduz a taxa de aprendizagem $\alpha = 0.5 * (0.6) = 0.3$

- Agora a equação para atualizar os pesos é
- $w_{ij}(\text{new}) = w_{ij}(\text{old}) + 0.6 * [x_i - w_{ij}(\text{old})]$
- A matriz de pesos após a segunda época de treinamento é

$w_1$	$w_2$
0.032	0.968
0.096	0.304
0.680	0.112
0.984	0.048



# Exemplo

- ▶ O resultado de 100 iterações (épocas) são (a taxa de aprendizagem diminui de 0.6 para 0.01)

Iteração 0

$w_1$	$w_2$
0.032	0.968
0.096	0.304
0.680	0.112
0.984	0.048

Iteração 1

$w_1$	$w_2$
0.032	0.968
0.096	0.304
0.680	0.112
0.984	0.048

Iteração 2

$w_1$	$w_2$
0.016	0.980
0.047	0.360
0.630	0.055
0.999	0.024

# Exemplo

---

Iteração 10

$w_1$	$w_2$
1.5e-7	1.0000
4.6e-7	0.3700
0.630	5.4e-7
1.000	2.3e-7

Iteração 50

$w_1$	$w_2$
1.9e-19	1.000
5.7e-15	0.470
0.5300	6.6e-15
1.0000	2.8e-15

Iteração 100

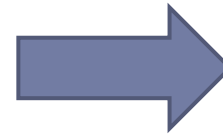
$w_1$	$w_2$
6.7e-17	1.000
2.0e-16	0.4900
0.510	2.3e-16
1.000	1.0e-16



# Exemplo

Iteração 100

$w_1$	$w_2$
6.7e-17	1.000
2.0e-16	0.4900
0.510	2.3e-16
1.000	1.0e-16



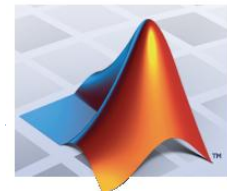
$w_1$	$w_2$
0	1.000
0	0.5
0.5	0
1.000	0

$x_1$	1	1	0	0	Cluster 2
$x_2$	0	0	0	1	Cluster 1
$x_3$	1	0	0	0	Cluster 2
$x_4$	0	0	1	1	Cluster 1

- ▶ A primeira coluna representa a media de dois vetores são colocados no cluster 1 e a segunda coluna representa a media dos dois vetores que são colocados no cluster 2

---

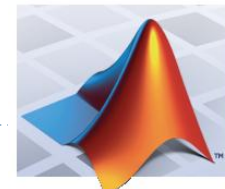
# SOM TOOLBOX (Matlab)



# Sobre a SOM TOOLBOX

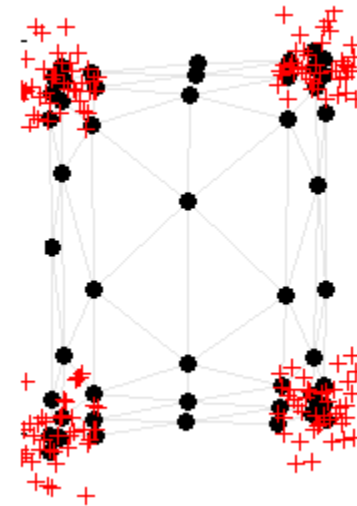
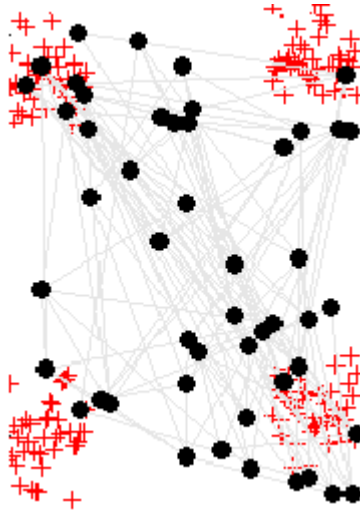
---

- ▶ Copyrighth ...
- ▶ SOM Toolbox 2.0, a software library for Matlab 5 implementing the Self-Organizing Map algorithm is Copyright (C) 1999 by **Esa Alhoniemi, Johan Himberg, Jukka Parviainen and Juha Vesanto**.
- ▶ This package is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version.
- ▶ Note: only part of the files distributed in the package belong to the SOM Toolbox. The package also contains contributed files, which may have their own copyright notices. If not, the GNU General Public License holds for them, too, but so that the author(s) of the file have the Copyright.
- ▶ ...
- ▶ If you wish to obtain such permission, you can reach us by paper mail: SOM Toolbox team Laboratory of Computer and Information Science P.O.Box 5400 FIN-02015 HUT Finland Europe and by email: **[somtlbx@mail.cis.hut.fi](mailto:somtlbx@mail.cis.hut.fi)**



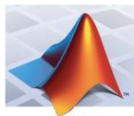
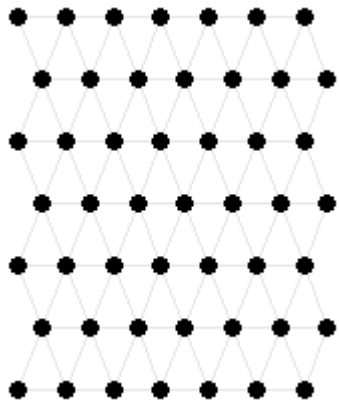
# som\_demo1

**Espaço Vetorial:  
inicialização randômica**

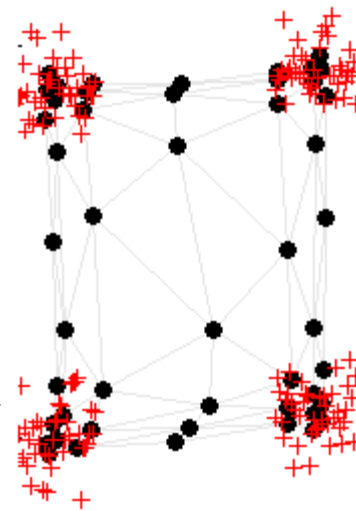
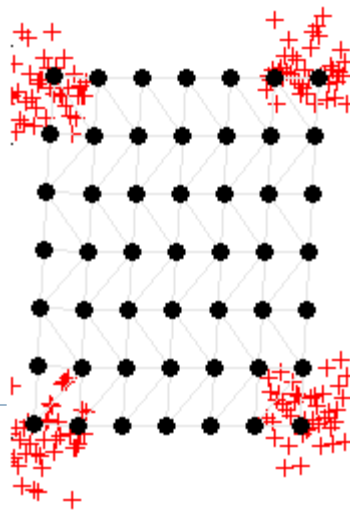


**Espaço Vetorial:  
após treinamento**

**Espaço Matricial**

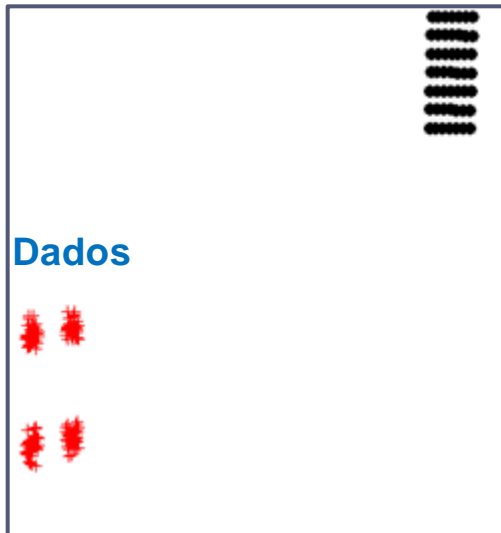


**Espaço Vetorial:  
inicialização linear**

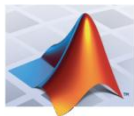
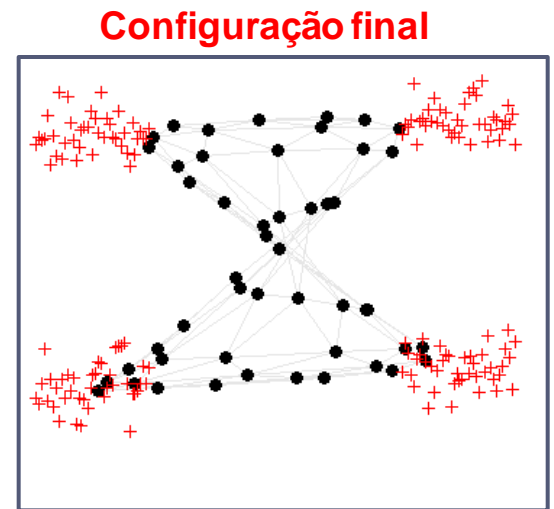


# som\_demo1

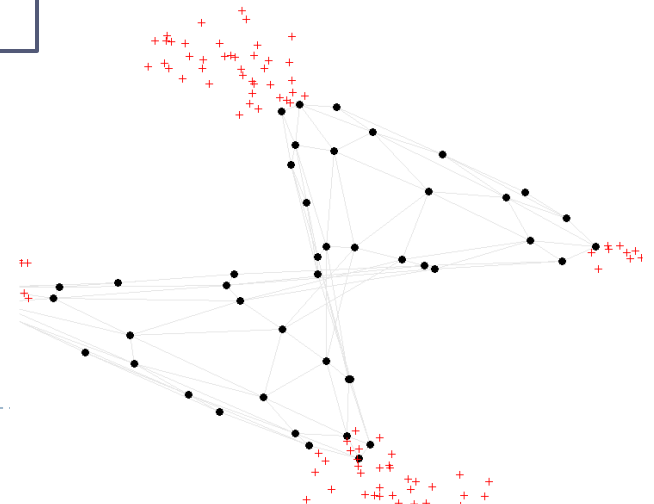
## Rede (SOM)



Espaço vetorial:  
inicialização



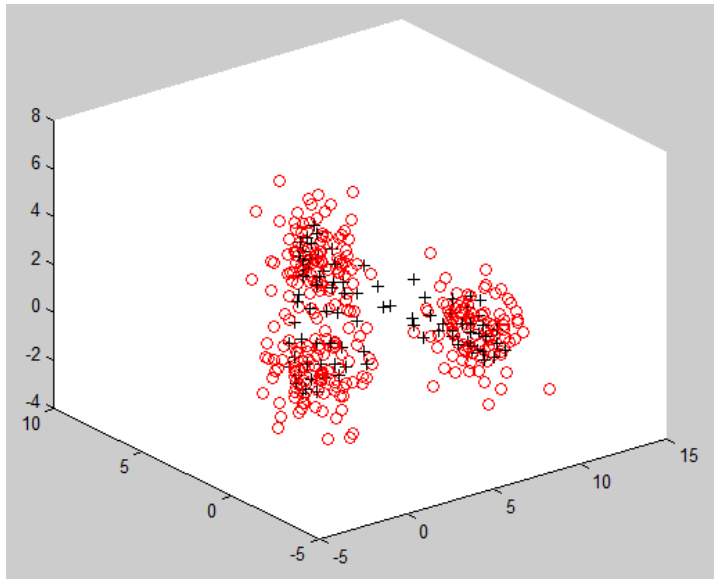
Observe a distorção  
topológica!!!



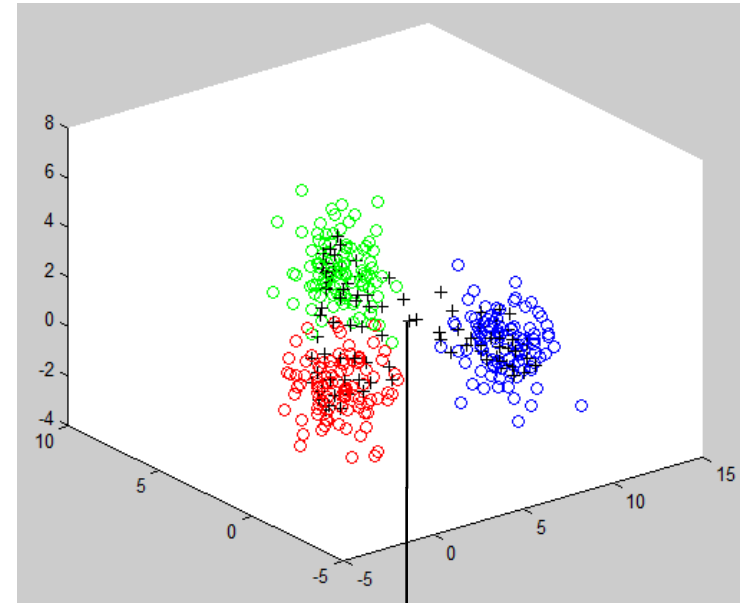


# som\_demo3

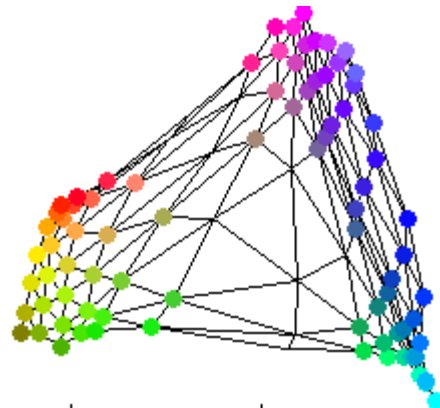
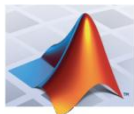
Dados em três dimensões



Classes - coloridas



Neurônios da SOM

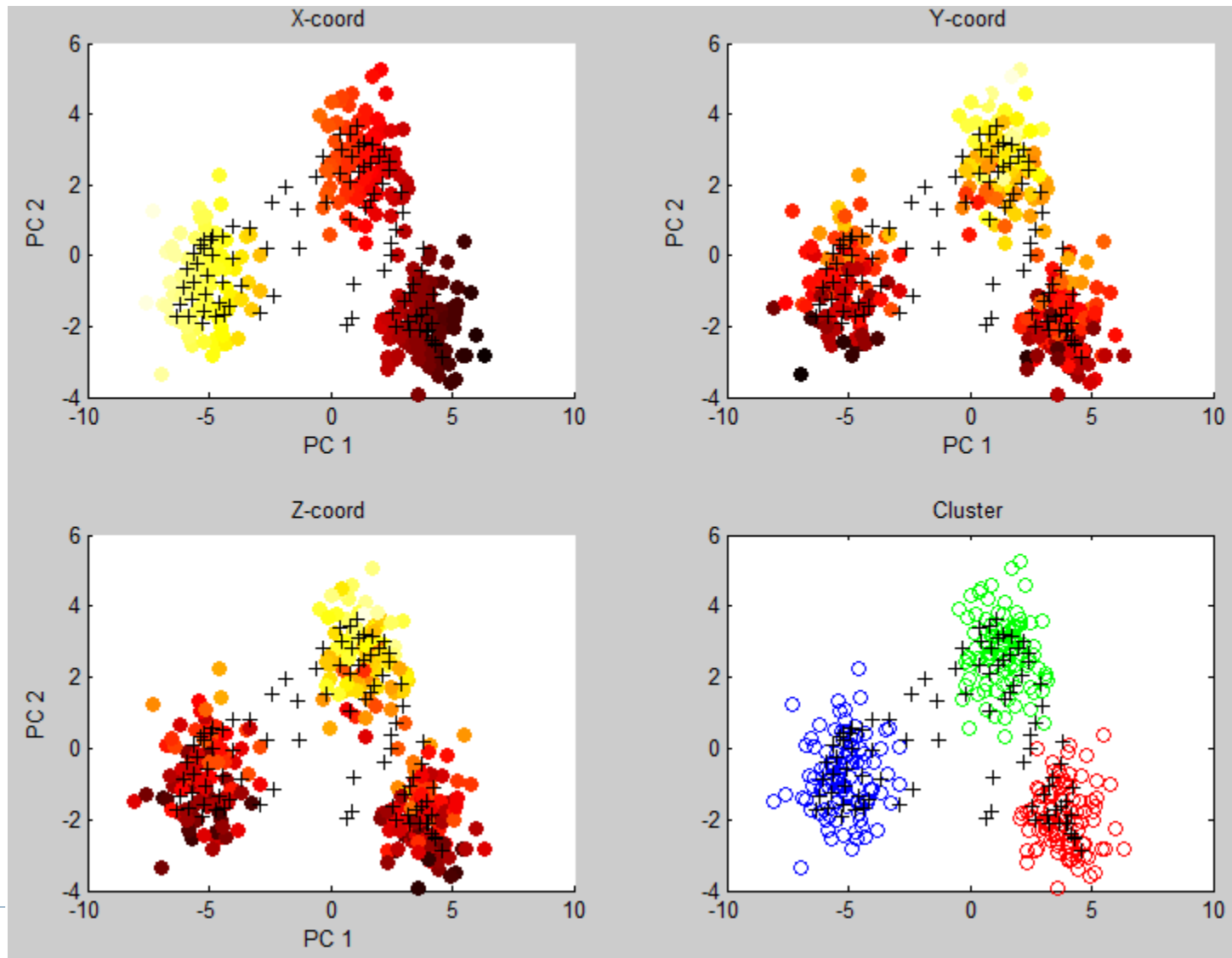


Visão tri-dimensional da disposição dos neurônios no espaço vetorial (dos dados).

Observe a estrutura matricial presente no gráfico.

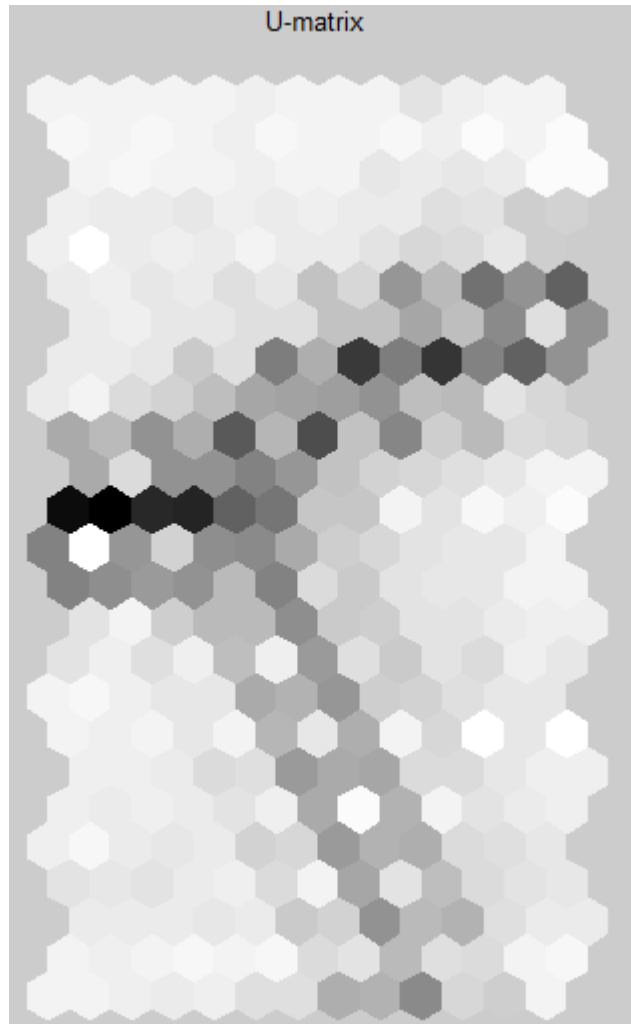
# som\_demo3

Cores similares representam a similaridade dos dados em relação a um dos atributos que os descrevem.

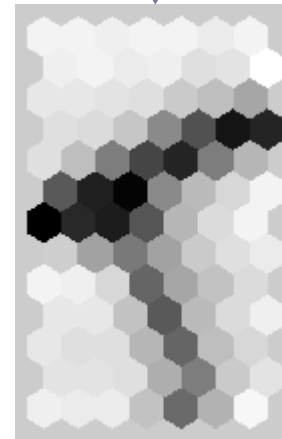


# som\_demo3

Matriz U



Tons de cinza!



Matriz de distâncias  
entre os neurônios

Com cores!

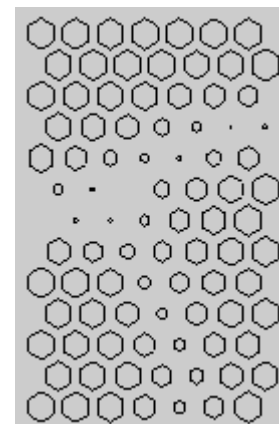
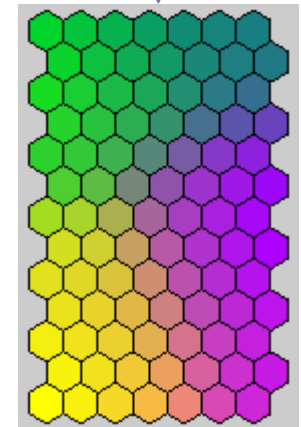
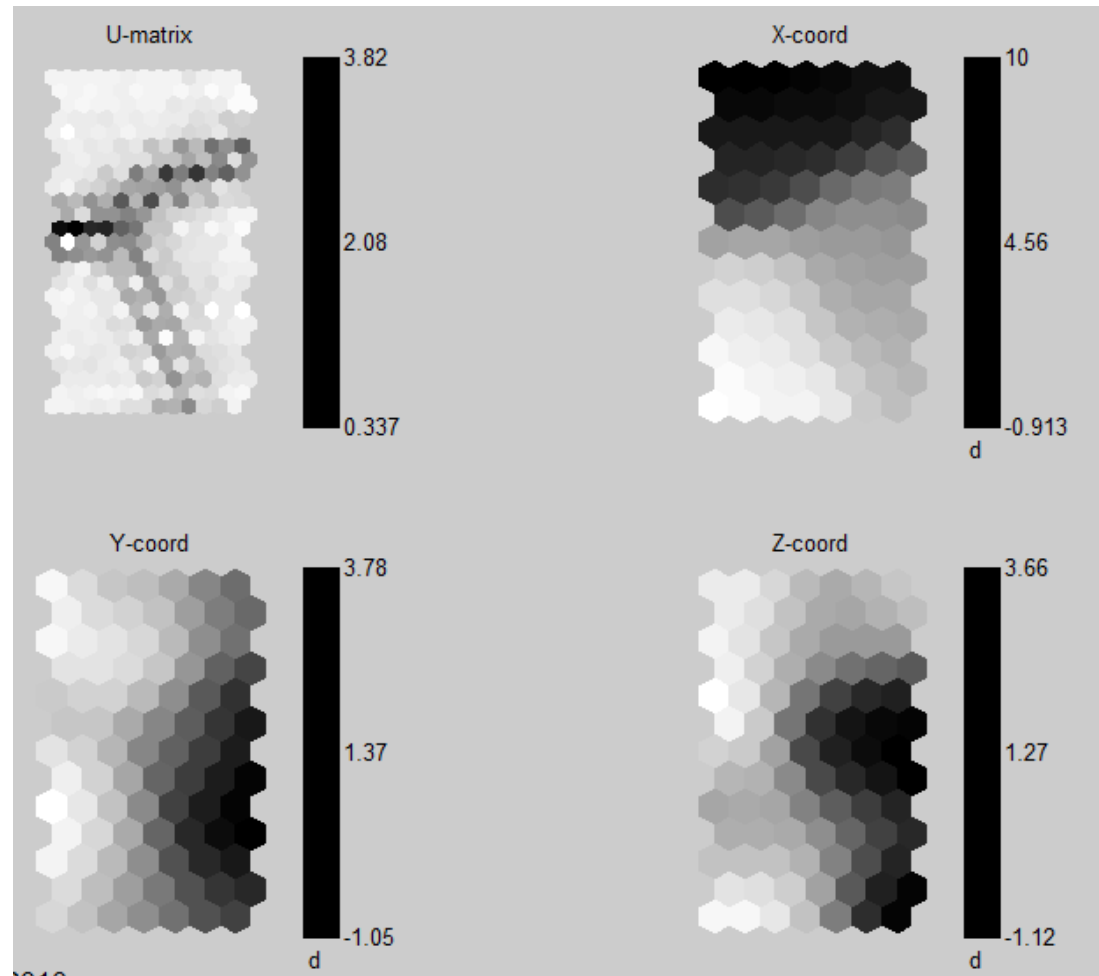
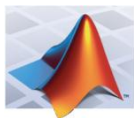


Gráfico demonstrando a  
representatividade do  
neurônio em relação aos  
dados.

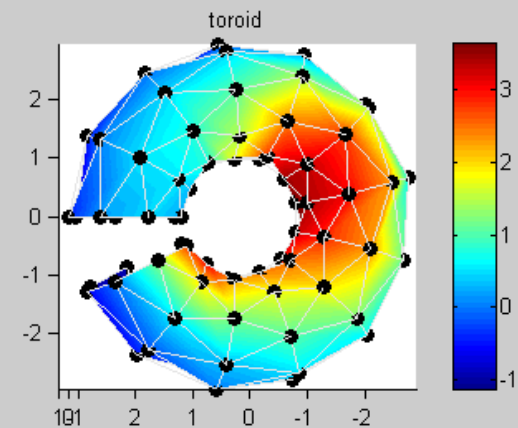
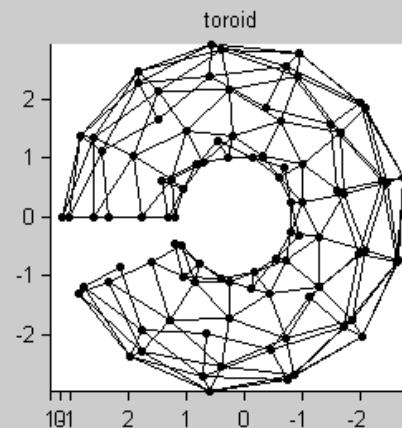
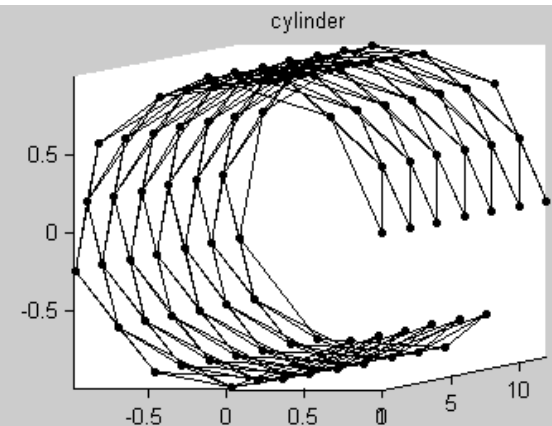
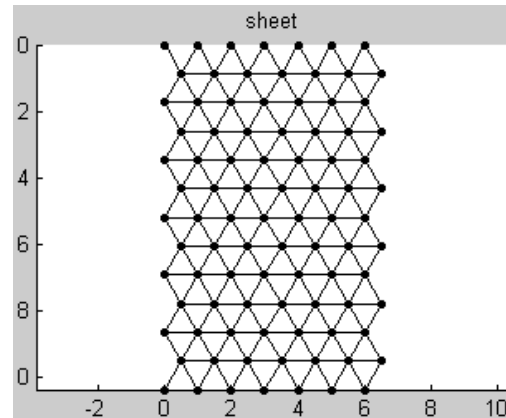
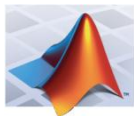
# som\_demo3

Matrizes de  
distâncias  
considerando  
cada dimensão  
do espaço  
vetorial.



# som\_demo3

Exemplos de organização matricial para o mapa de saída do SOM.

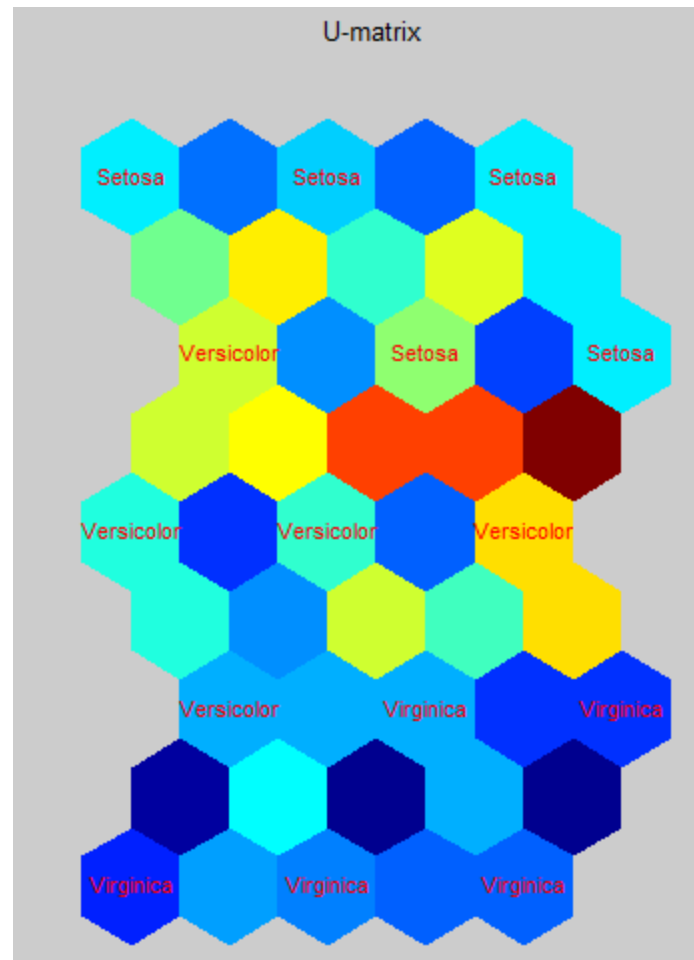
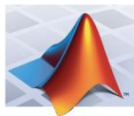


# som\_demo4

---

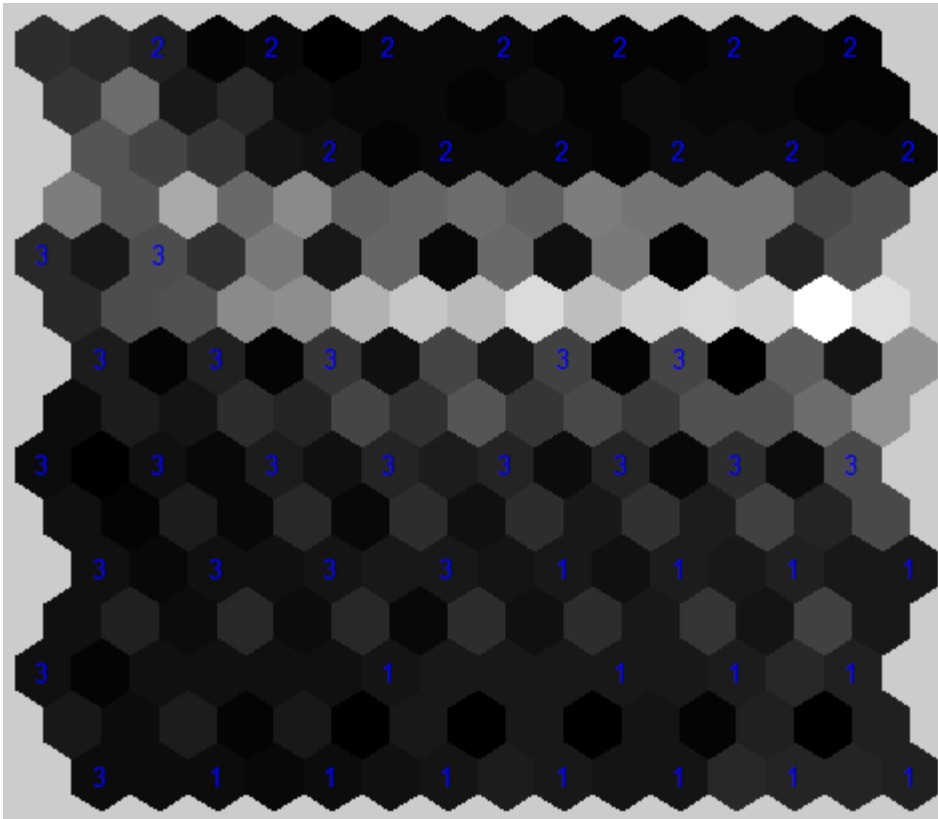
Matriz U com rótulos.

Os neurônios com rótulos são BMUS para os dados do conjunto de dados IRIS.



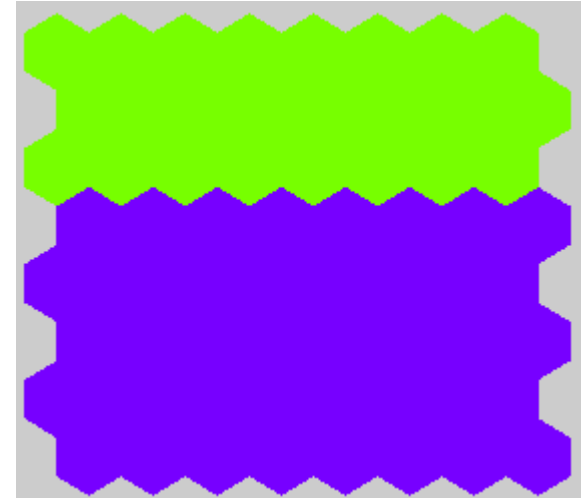
# Mais testes...

- SOM [8,8] no Conjunto de Dados **Iris**



Matriz U rotulada

2-means sobre o mapa resultante

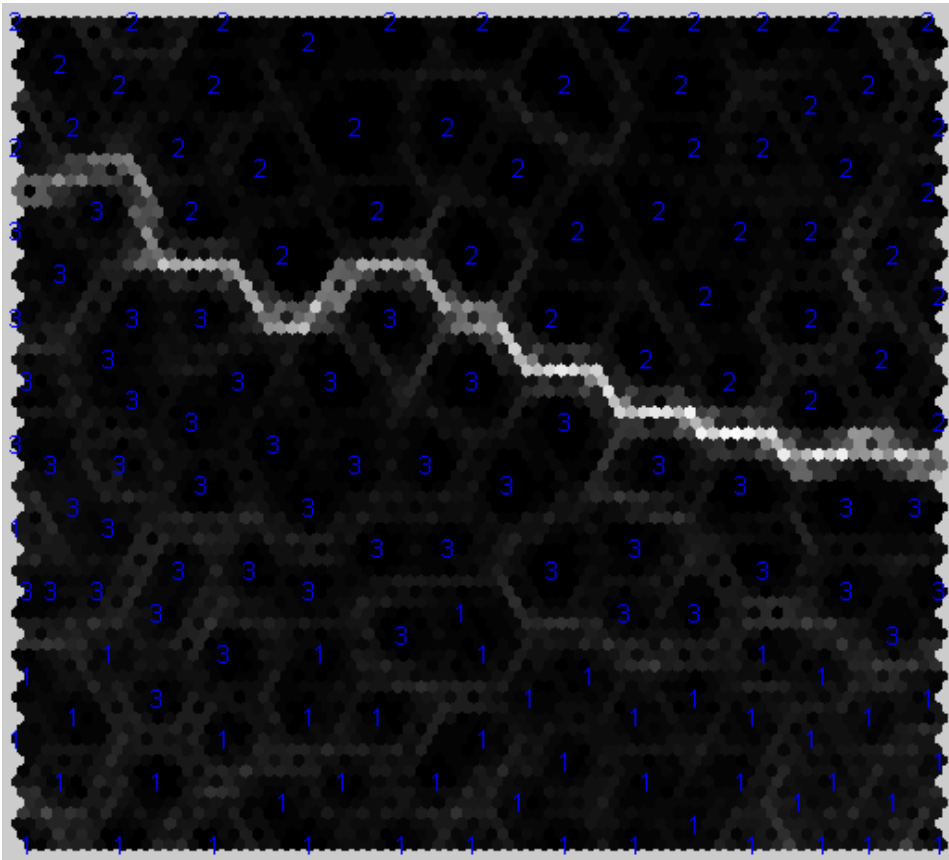


3-means sobre o mapa resultante



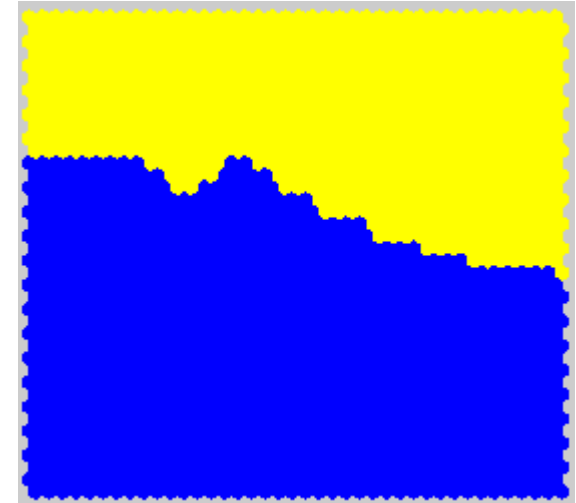
# Mais testes ...

- SOM [40,40] no Conjunto de Dados **Iris**

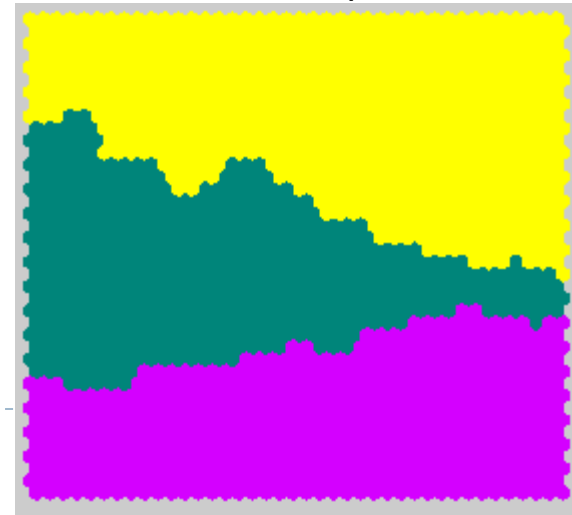


Matriz U rotulada

2-means sobre o mapa resultante



3-means sobre o mapa resultante





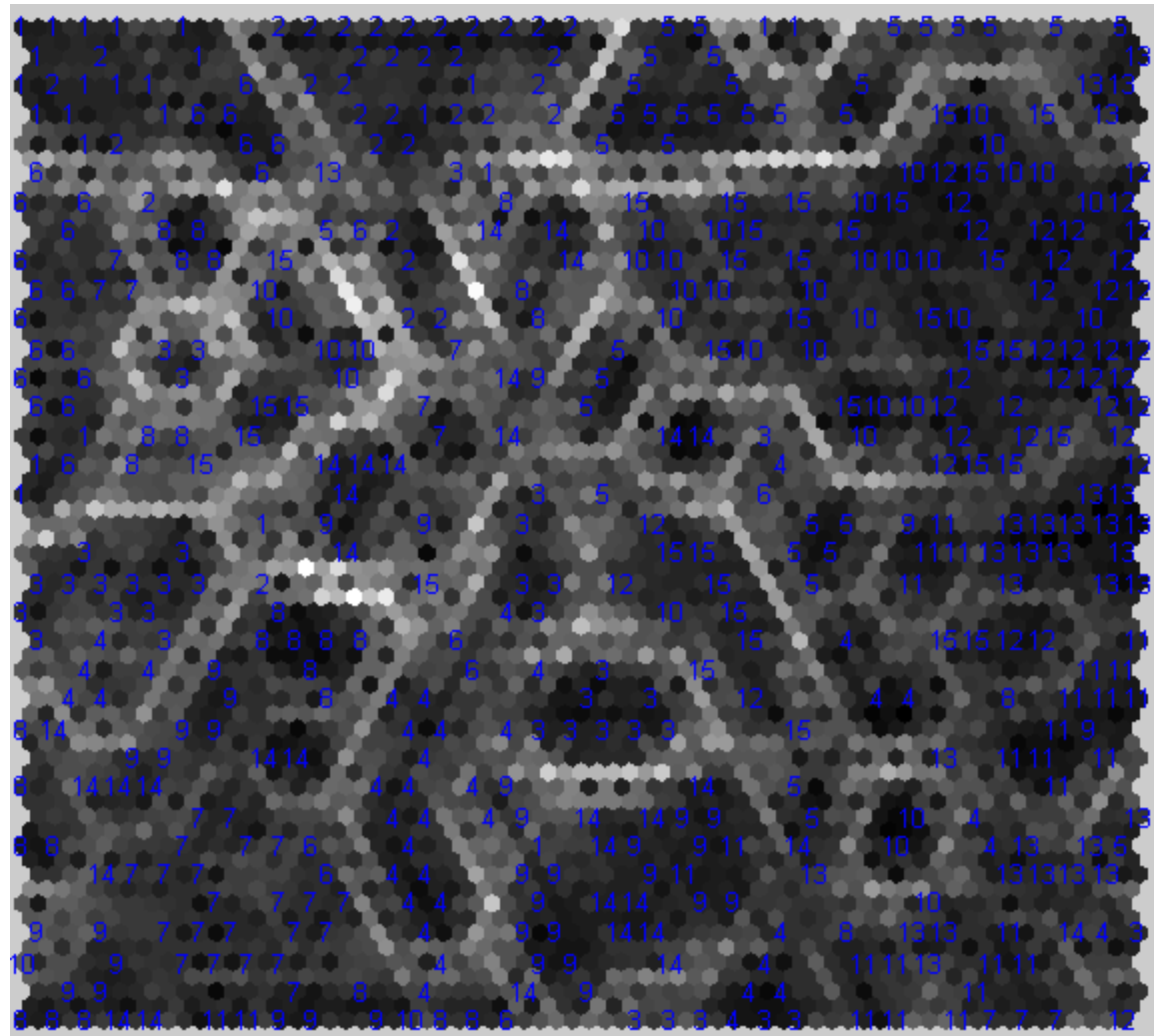
# 15 movimentos - LIBRAS

---

- 01 - arcos\_antihorario
- 02 - arcos\_horario
- 03 - balancar\_curva
- 04 - balancar\_horizontal
- 05 - balancar\_vertical
- 06 - circulos
- 07 - curvas\_inferior
- 08 - curvas\_superior
- 09 - ondulatorio\_horizontal
- 10 - ondulatorio\_vertical
- 11 - reta\_horizontal
- 12 - reta\_vertical
- 13 - tremular
- 14 - ziguezague\_horizontal
- 15 - ziguezague\_vertical



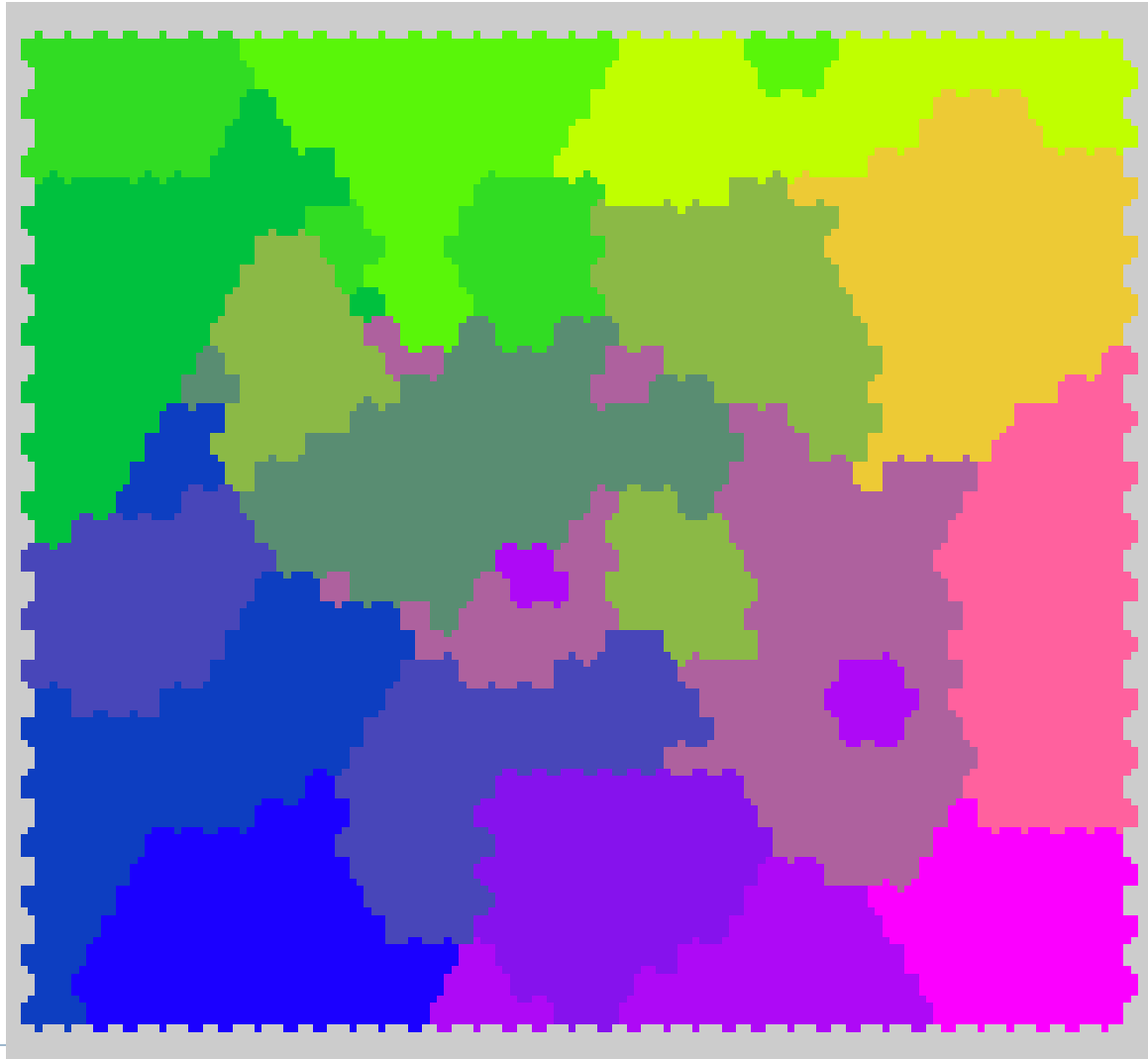
# 15 movimentos - LIBRAS



Representação: Coordenadas (X,Y) do centróide do objeto em movimento!

# 15 movimentos - LIBRAS

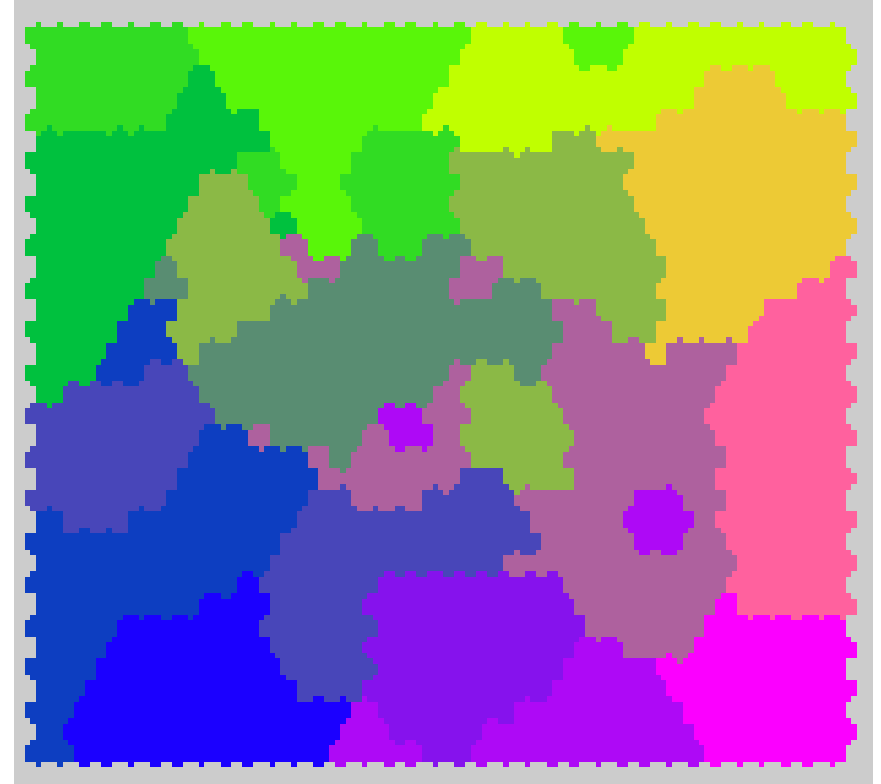
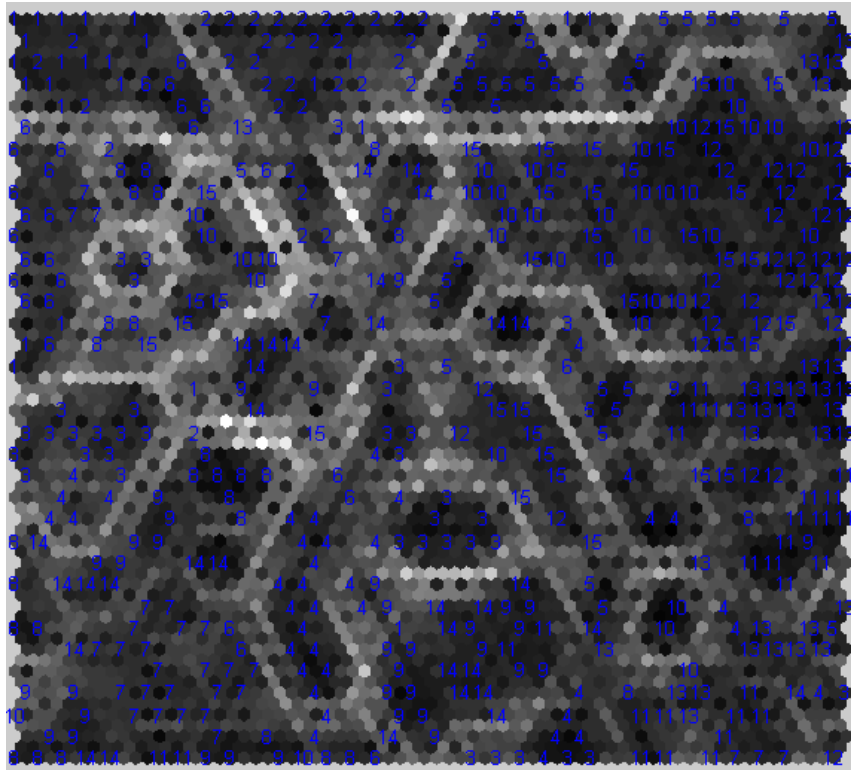
---



► Representação: Coordenadas (X,Y) do centróide do objeto em movimento!

# 15 movimentos - LIBRAS

---



**Representação: Coordenadas (X,Y) do centróide do objeto em movimento!**

---



# 15 movimentos - LIBRAS

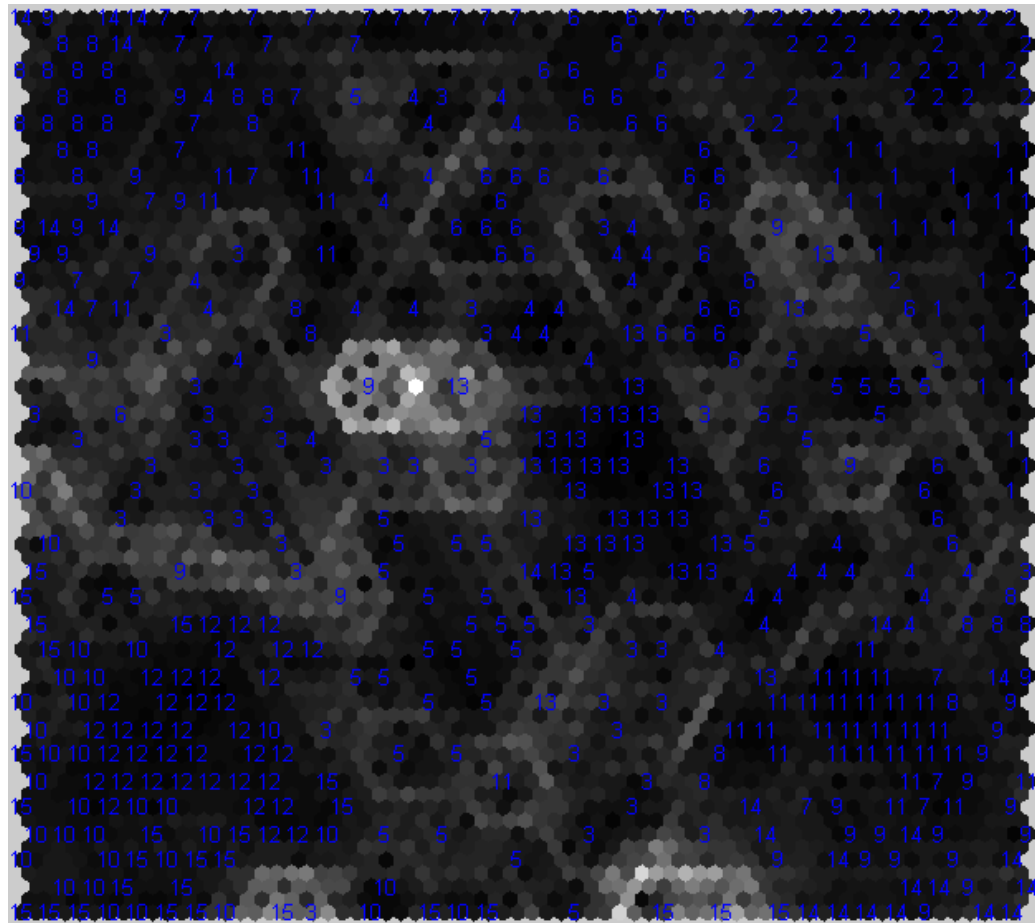
---

- 01 - arcos\_antihorario
- 02 - arcos\_horario
- 03 - balancar\_curva
- 04 - balancar\_horizontal
- 05 - balancar\_vertical
- 06 - circulos
- 07 - curvas\_inferior
- 08 - curvas\_superior
- 09 - ondulatorio\_horizontal
- 10 - ondulatorio\_vertical
- 11 - reta\_horizontal
- 12 - reta\_vertical
- 13 - tremular
- 14 - ziguezague\_horizontal
- 15 - ziguezague\_vertical



# 15 movimentos - LIBRAS

---



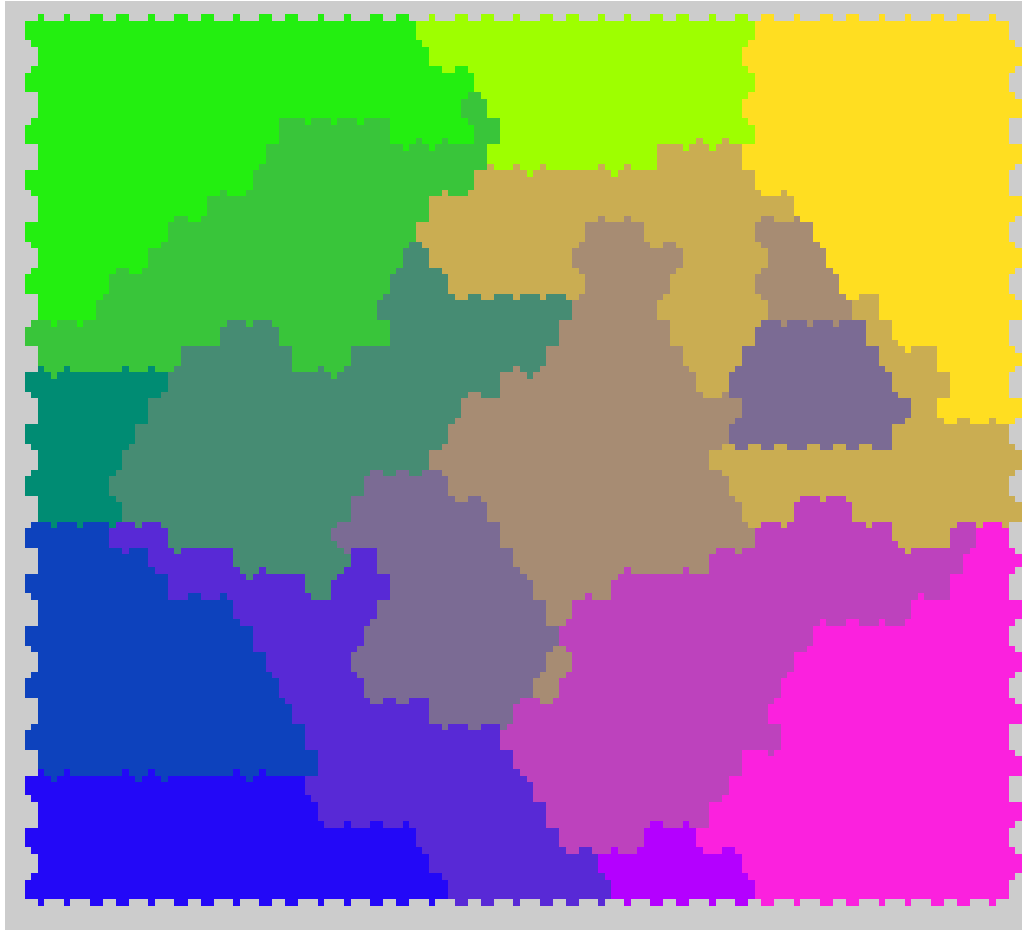
**Representação: Coordenadas (X,Y) do centróide do objeto em movimento, no espaço de frequências (Fourier)**

---



# 15 movimentos - LIBRAS

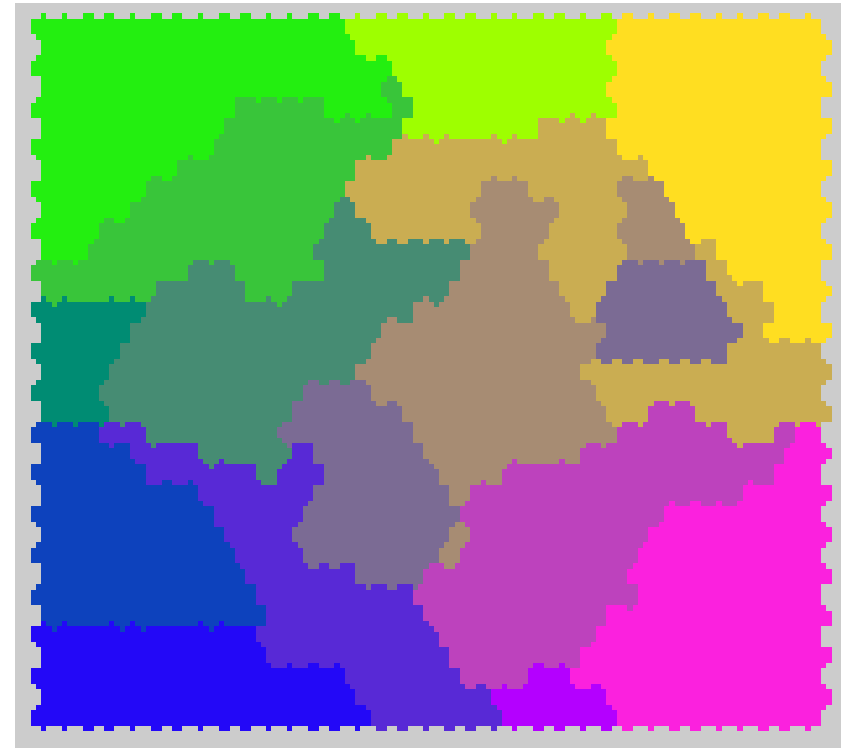
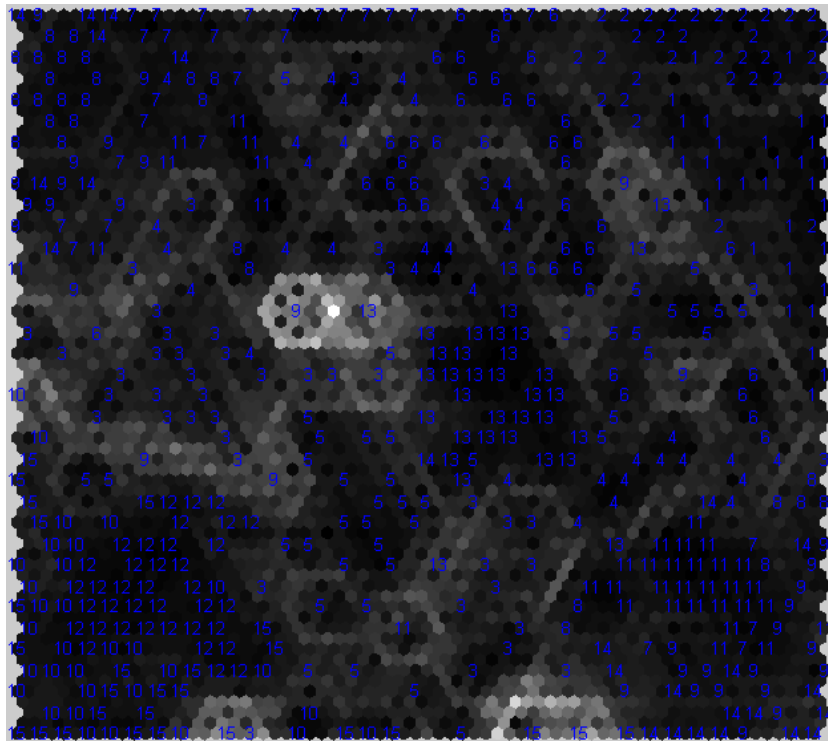
---



Representação: Coordenadas (X,Y) do centróide do objeto em movimento, no espaço de frequências (Fourier)

# 15 movimentos - LIBRAS

---



**Representação: Coordenadas (X,Y) do centróide do objeto em movimento, no espaço de frequências (Fourier)**

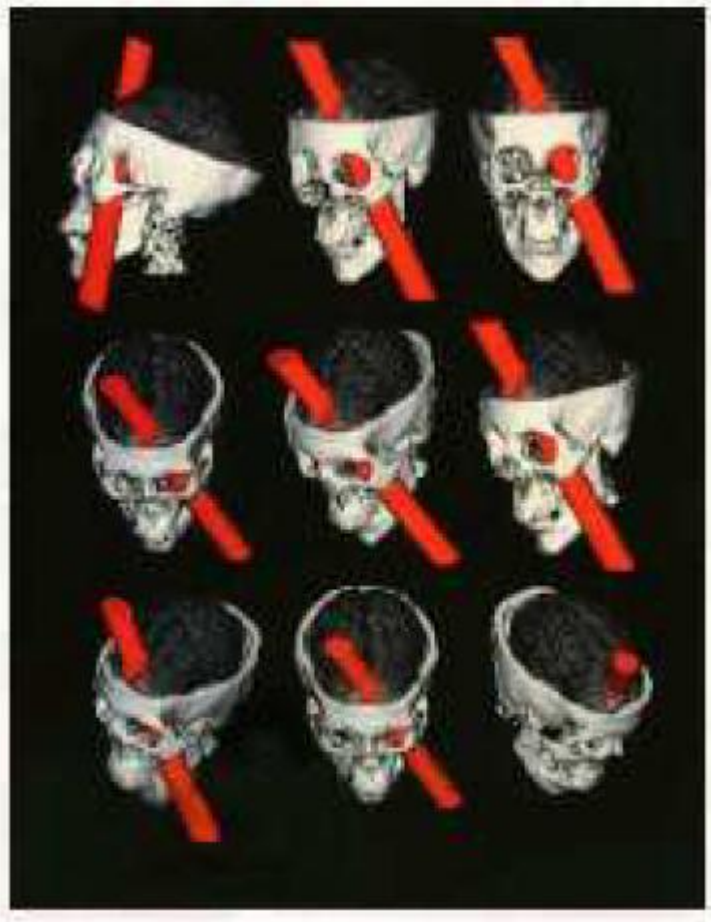
---





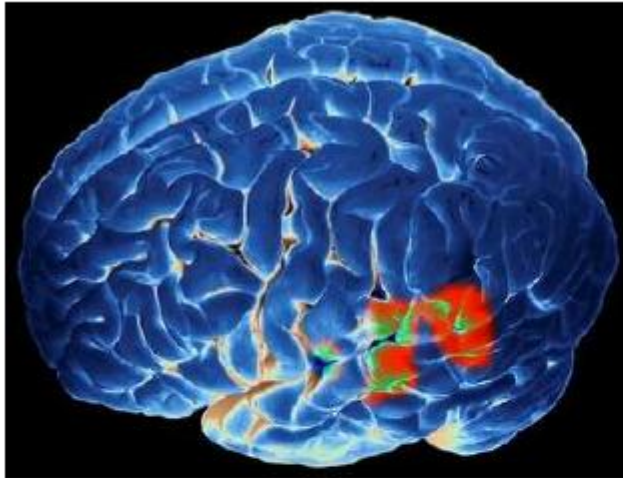
# O caso Phineas Cage

---

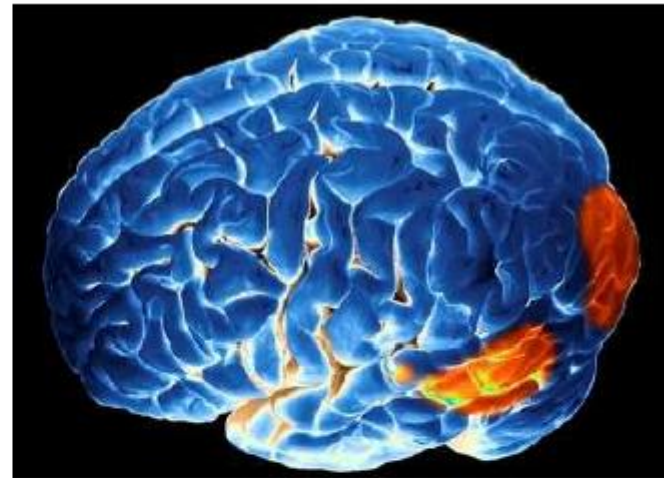


# Monitorando a atividade cerebral

---



Ouvindo palavras



Lendo palavras



Falando palavras



Pensando em palavras



# Bibliografia

---

## ▶ Esses slides correspondem ao conteúdo apresentado em:

- ▶ Data Mining: um guia prático. Ronaldo Goldshmidt e Emmanuel Passos. Editora Campus, 2005.
- ▶ Inteligência Artificial. Stuart Russel e Peter Norvig. Editora Campus, 2ª ed, 2004.
- ▶ (Han & Kamber, 2006) Han, Jiawei & Kamber, Micheline. Data Mining: Concepts and Techniques. 2ed., Morgan Kaufmann, 2006.
- ▶ (Peres, 2006) Peres, S. M. Dimensão Topológica e Mapas Auto-Organizáveis de Kohonen. Tese de Doutorado defendida Faculdade de Engenharia Elétrica e de Computação – Unicamp, 2006.
- ▶ (Costa, 1999) Costas, J. A. F. Classificação Automática e Análise de Dados por Redes Neurais Auto-Organizáveis. Tese de Doutorado defendida Faculdade de Engenharia Elétrica e de Computação – Unicamp, 1999.
- ▶ SOM Toolbox <http://www.cis.hut.fi/projects/somtoolbox/>
- ▶ (Fausett, 1994) Fausett, L. Fundamentals of Neural Networks, 1994.

