

Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

SOM Kohonen's Self- Organising Maps

Victor Lobo

Sumário

- Para que serve ?
- O que é um SOM ?
 - Perspectiva histórica
 - Princípios básicos
- Formalização matemática
- Como ver os resultados ?
- Um exemplo
- Software e Aplicações de SOM

Mapas Auto-Organizados - SOM

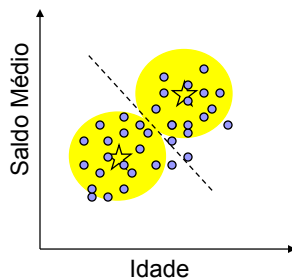
V 1.3 V.Lobo, EN 2009

Para que serve ?

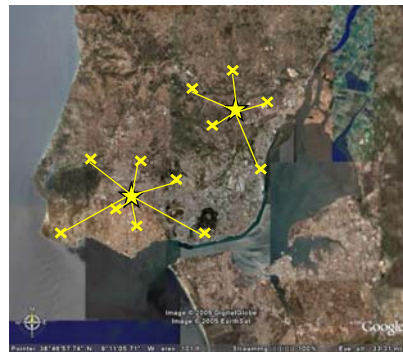
- Detectar agrupamentos (clusters)
- Visualizar dados multidimensionais
- Explorar os dados
- Outros...

Definir agrupamentos (clusters)

- Segmentação de mercado
- Localização



Clientes do banco ⇒ Gestores de conta



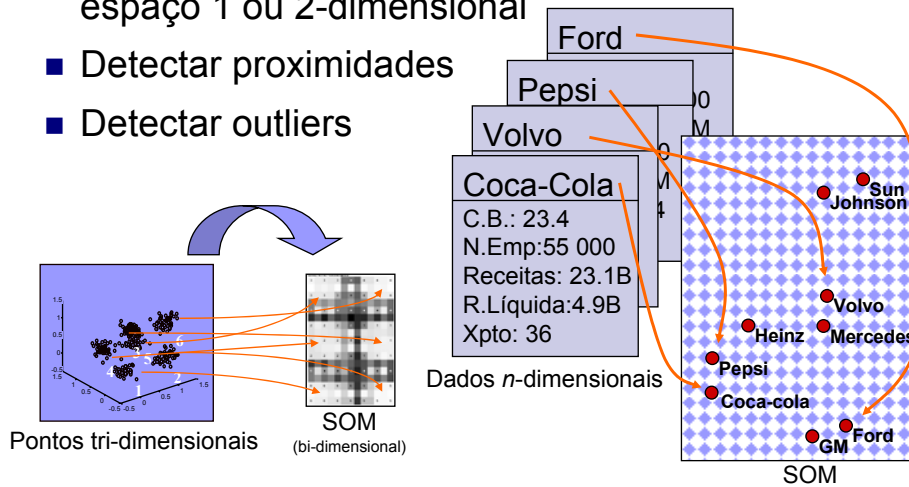
Locais de lojas ⇒ Locais de armazéns

Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

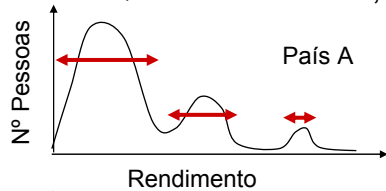
Visualizar dados multidimensionais

- Projectar de um espaço n -dimensional para um espaço 1 ou 2-dimensional
- Detectar proximidades
- Detectar outliers

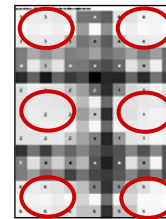


Detectar agrupamentos

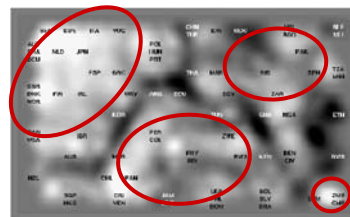
- Explorar os dados
- Identificar a estrutura dos dados
 - Quantos clusters ?, Quão importantes ?...



Distribuição de rendimento em 2 países



Pontos 3D em torno de 6 vértices de um cubo



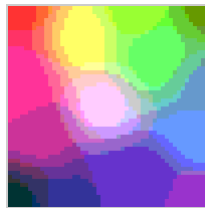
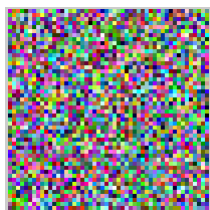
Dados sobre pobreza em diversos países

Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Outros tipos de problemas...

- Resolver o TSP, controlar Robots, ordenar dados, interpolar pontos, classificar dados, extrair características, amostrar dados, alarmes, etc,etc,etc



Ordenar cores



Problema do caixeiro-viajante

O que é um SOM ?

- Perspectiva histórica
- Visão geral
- A matemática

Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Perspectiva histórica

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 vectores
 - Memórias associativas
 - Preservar a topologia nos mapeamentos: padrões vizinhos devem ser mapeados para neurónios vizinhos

Mapas Auto-Organizados - SOM

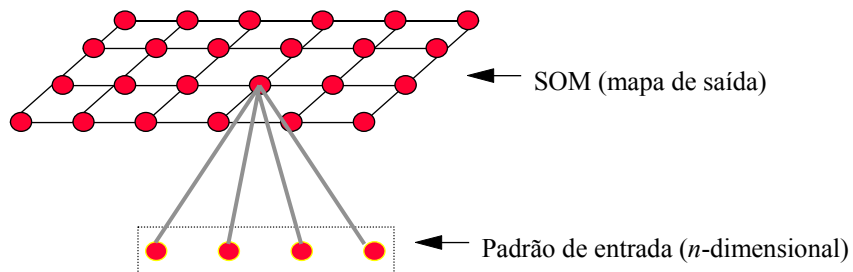
V 1.3 V.Lobo, EN 2009

Algumas ideias gerais

- Rede neuronal
 - Conjunto de neurónios, ou UNIDADES
 - Aprendizagem não supervisionada (contrário do usual)
- TREINO da rede
 - A rede é construída, ou seja os seus parâmetros são ajustados, com base nas características dos dados disponíveis
- UTILIZAÇÃO da rede
 - Com a rede já treinada, podemos usá-la para muitas coisas...

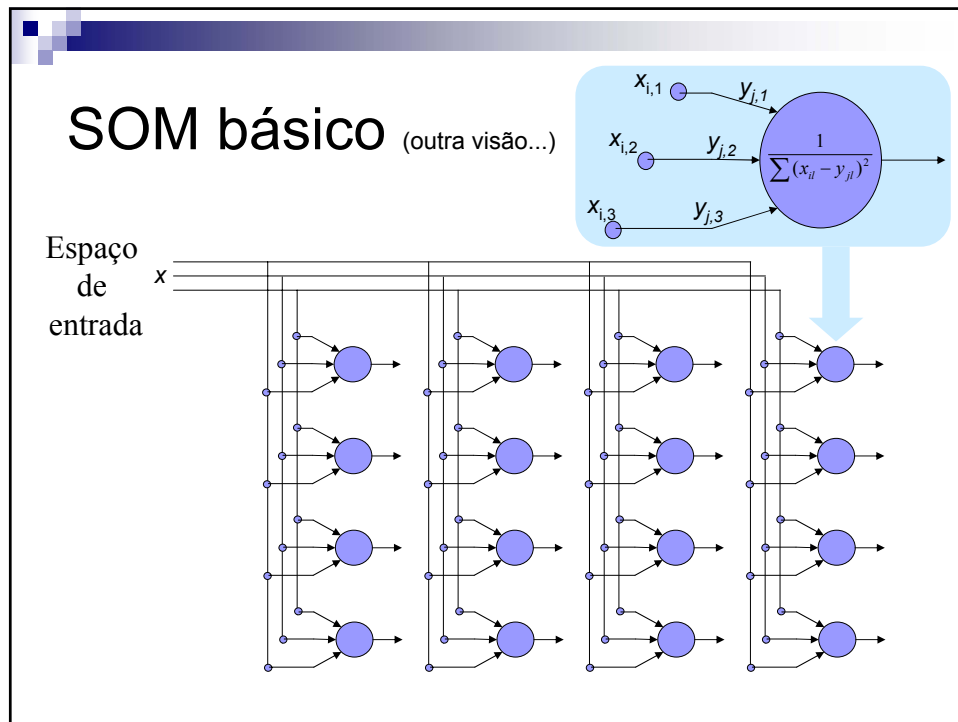
SOM básico

- Neurónios (**unidades**) dispostos numa grelha bi-dimensional
 - Pode ser uma grelha 1-dimensional (linha) ou m -dimensional ...
- Uma única camada
- Aprendizagem competitiva (quase "winner-take all")



Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009



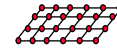
Espaço de entrada vs saída

- **Espaço de entrada** = espaço n-dimensional onde estão os dados
- **Espaço de saída** = espaço definido pela grelha das unidades
- Cada unidade é um ponto no espaço de saída e um **vector no espaço de entrada**, tal como os padrões de dados.

Mapas Auto-Organizados - SOM

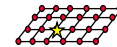
V 1.3 V.Lobo, EN 2009

Treino da rede



- 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)



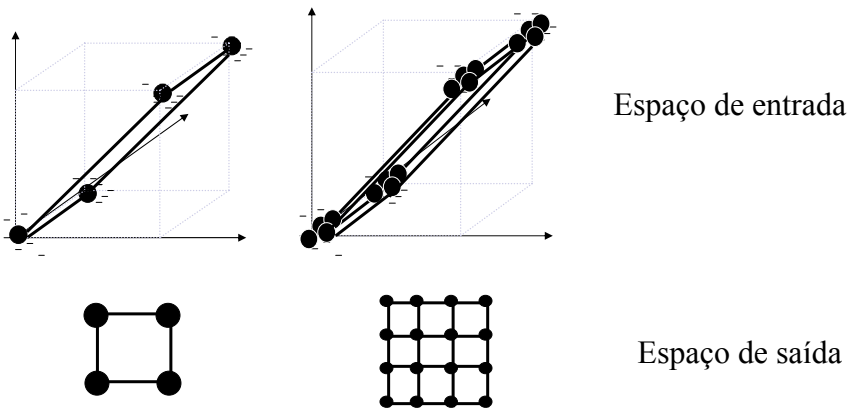
- Padrões de entrada são comparados com todas as unidades; a mais próxima é considerada a **BMU**.
- Consideramos que o padrão de entrada é **mapeado** para a BMU.
- **A BMU actualiza-se** (de modo a aproximar-se mais do padrão de dados que representa), e os seus vizinhos actualizam-se também um pouco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

Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Exemplo 1: mapeamento de 3D para 2D

- Pontos agrupados em 4 cantos de um cubo



Exemplo 2:

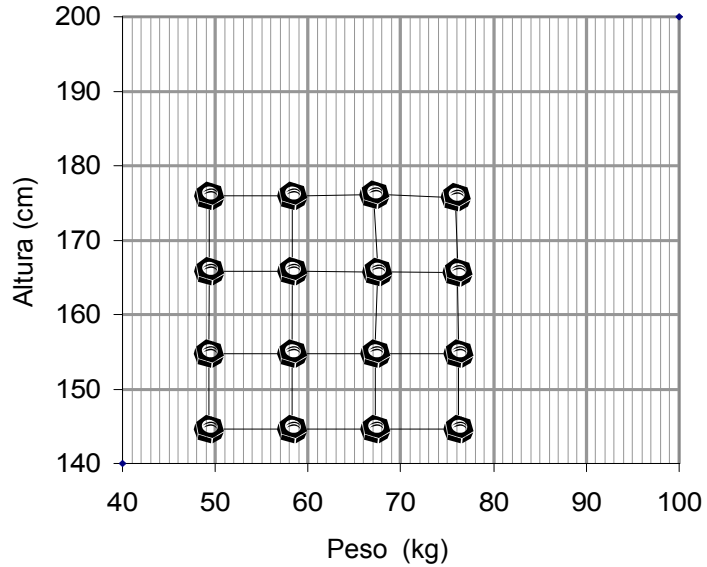
- Uma rede “manual” de porcas...
 - (ver para crer...)

Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

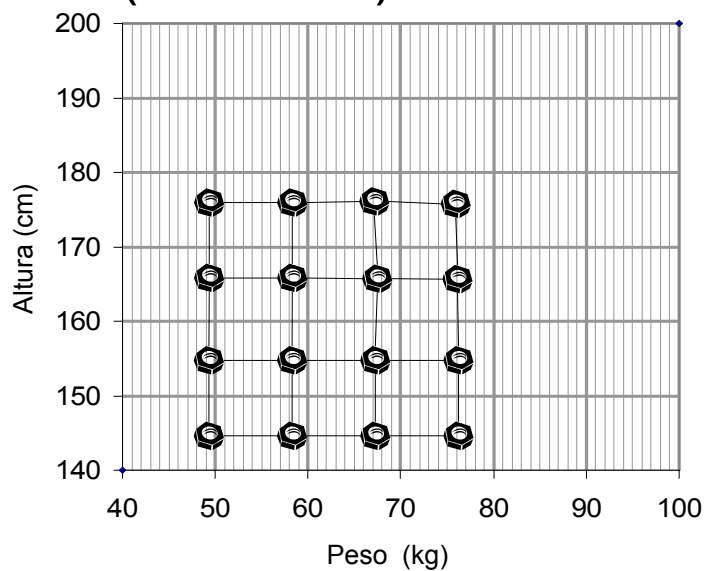
Exemplo 2 (simulado):

- Exemplo "físico"
 - Neurónios= porcas= unidades
 - Folha = espaço de input
- Problema:
- Analizar altura e peso das pessoas da turma



Exemplo 2 (simulado):

- Exemplo "físico"
 - Neurónios= porcas= unidades
 - Folha = espaço de input
- Problema:
- Analizar altura e peso das pessoas da turma

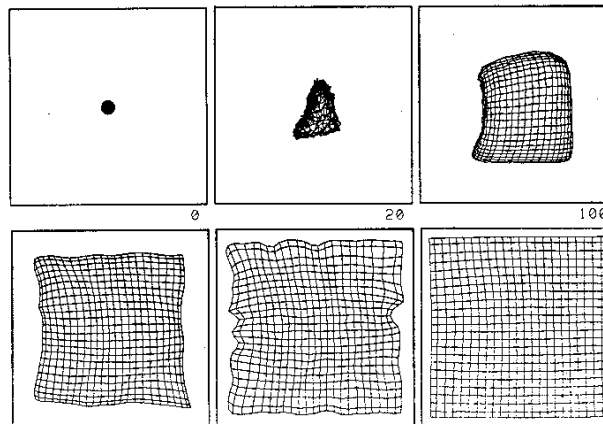


Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Exemplo 3: mapeamento de 2D para 2D

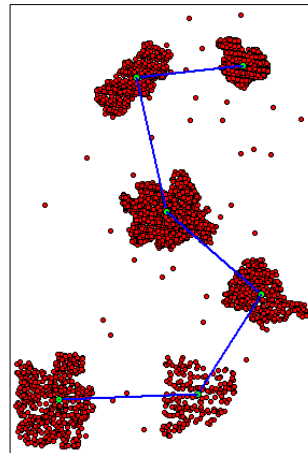
- Dados distribuídos uniformemente num quadrado
- Usado na demo do Matlab



[Kohonen 95]

Exemplo 4: mapeamento de 2D para 1D

- Animação

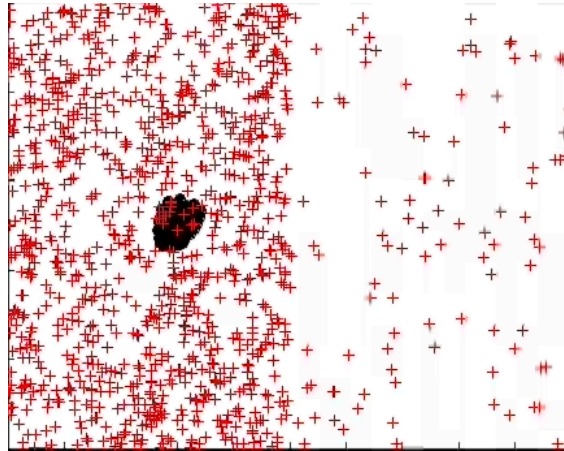


Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Exemplo 5: mapeamento de 2D para 2D

- Densidades diferentes



(animação)

Formalização
matemática

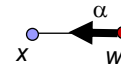
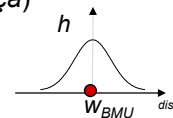
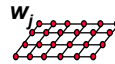
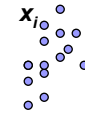
Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Os dados, a rede e a inicialização

Seja:

- $\mathbf{X} = \{ \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \}$ o conjunto de dados de treino, com dimensão m .
 - $\mathbf{x}_i = [x_i^1, x_i^2, \dots, x_i^m]^T$, onde x_{ij} são escalares de valor real.
- \mathbf{W} uma grelha de $p \times q$ unidades \mathbf{w}_j
 - $\mathbf{w}_j = [w_j^1, w_j^2, \dots, w_j^m]^T$
 - Valores de \mathbf{w}_j escolhidos aleatoriamente na zona dos dados
- $h(\mathbf{w}_i, \mathbf{w}_j, r)$ uma função real (denominada de vizinhança)
 - Quando $\| \mathbf{w}_i - \mathbf{w}_j \|_{(na\ grelha)} \rightarrow \infty$, $h(\mathbf{w}_i, \mathbf{w}_j, r) \rightarrow 0$
 - r determina o raio (zona de influência)
- α a taxa de aprendizagem
 - $0 \leq \alpha \leq 1$
 - A começa grande e vai diminuindo ao longo do treino



Algoritmo de treino

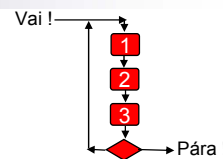
Para todos os $\mathbf{x}_i \in \mathbf{X}$:

- 1) **Calcular** a distância entre \mathbf{x}_i e todas as unidades \mathbf{w} ($d_{i,j} = \| \mathbf{x}_i - \mathbf{w}_j \|$)
- 2) **Escolher** a unidade vencedora (BMU)

$$w_{bmu} : d_{i,bmu} = \min(d_{i,j})$$
- 3) **Actualizar** cada unidade de acordo com a regra

$$w_j = w_j + \alpha \cdot h(w_{bmu}, w_j, r) \cdot \| \mathbf{x}_i - \mathbf{w}_j \|$$

Repetir o processo, diminuindo o valor de α e r , e percorrendo várias vezes todos os dados de treino, até atingir um critério de paragem.



Mapas Auto-Organizados - SOM

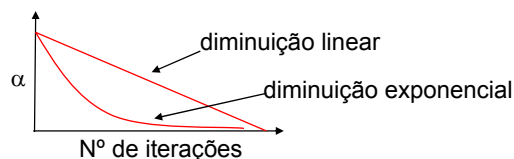
V 1.3 V.Lobo, EN 2009

Principais decisões a tomar:

- Quantas unidades ? Que tipo de grelha ?
Que dimensão ?
- Quantas iterações ?
- Que tipo de função de vizinhança ? Que
valor inicial para r ? Que valor final ?
- Que valor para α inicial ?

Taxa de aprendizagem α

- $0 \leq \alpha \leq 1$
- Regula a **plasticidade** da rede
 - Valores altos \Rightarrow A rede move-se muito, adapta-se “depressa”
 - Valores baixos \Rightarrow A rede move-se pouco e estabiliza
- Começa com valores altos e vai diminuindo até chegar a 0



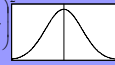
Mapas Auto-Organizados - SOM


V 1.3 V.Lobo, EN 2009

Função de vizinhança

■ A forma

- Gaussiana
- Rectangular (bolha)
- Rampa
- Outras

$$h_g(w_{pq}, w_{mn}, r) = e^{-\frac{1}{2} \left(\frac{\sqrt{(p-n)^2 + (q-m)^2}}{r} \right)^2}$$


$$h_r(w_{pq}, w_{mn}) = \begin{cases} 1 & \text{if } \sqrt{(p-n)^2 + (q-m)^2} \leq r \\ 0 & \text{if } \sqrt{(p-n)^2 + (q-m)^2} > r \end{cases}$$


■ Responsável pela ordenação topológica

- Faz com que haja “ligações laterais” entre unidades próximas na grelha

Raio r da função de vizinhança

■ Função to tempo (ou nº de iterações) $r(t)$

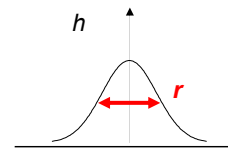
- Raio grande \Rightarrow Muitas unidades actualizados \Rightarrow Permite desdobragem
- Raio pequeno \Rightarrow Só vizinhos próximos são actualizados \Rightarrow Ajuste fino

■ Valores iniciais de r

- 1ª fase – semelhante ao tamanho da rede
- 2ª fase – semelhante ao raio dos clusters que esperamos obter

■ Valor final de r

- 0 \Rightarrow Ajuste bom (k -médias)
- 1 \Rightarrow Manter a ordenação, efeito de fronteira



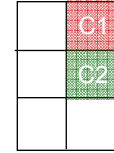
Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Tamanho do SOM

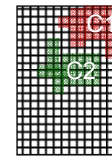
■ SOM de k-médias

- Tem poucas unidades
- 1 unidade para cada cluster esperado



■ SOM emergente

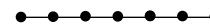
- Muitas unidades para cada cluster esperado
- Permite representar clusters “complicados”, e “variados”
- Permite entender a estrutura dos dados, detectar o número de clusters, etc...



Dimensão e tipo de grelha

■ Grelha unidimensional (linha)

- Substitui k -médias
- Permite estabelecer uma ordenação dos dados

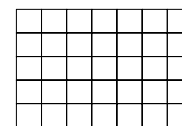


■ Grelha 3D ou n-dimensional

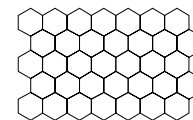
- Difícil de visualizar

■ Grelha bidimensional

- A mais usada
- Grelha quadrada
 - Fácil de trabalhar
- Grelha hexagonal ou triangular
 - Induz menos distorção



Grelha rectangular



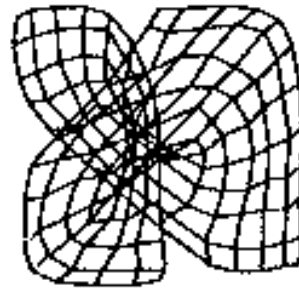
Grelha hexagonal

Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Nº Iterações e desdobraimento

- Nº Iterações
 - Épocas vs indivíduos
 - Na dúvida... escolher mais!
- Problemas de desdobraimento
 - Existem mínimos locais
 - Medidas de erro topológico
- Soluções
 - Várias inicializações
 - 2 fases (unfolding+fine tuning)
 - Observar erro topológico e erro de quantização



[Ritter 92]

Aspectos teóricos

- Função de energia minimizada: [Hertz 91]

$$V(w) = \frac{1}{2} \sum_x \sum_i \Lambda(i, i^*) |\bar{x} - \bar{w}_i|^2 = \frac{1}{2} \sum_x \sum_k M_{x,k} \sum_i \sum_j \Lambda(i, k) (x_j - w_{ij})^2$$

- Altamente não-linear, e não é uma função global devido ao conceito de unidade vencedora.
 - Resultados mais completos para dimensão 1 [Cottrell]
 - Algumas boas aproximações para 2D [Ritter]
 - Há regras de aprendizagem “bem comportadas” [Heskes]
 - Há outras alternativas “bem fundamentadas” [Bishop]
- } Pouco usadas
- Factor de magnificação
 - Densidade de neurónios \propto (densidade dos dados)^k, com $k < 1$
 - Existe um factor de “ampliação” das zonas com menor densidade

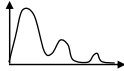
Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Como ver os resultados ?

- Matrizes U (U-MAT)
- Calibração (ou etiquetagem)
- Planos de componentes
- Outros

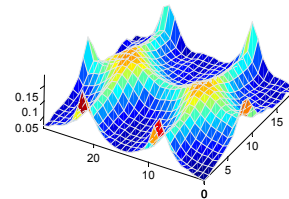
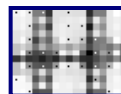
Matrizes U (U-MAT) [Ultsch 93]

- Permite-nos identificar clusters 
- Calcula a distância, no espaço de entrada, entre vizinhos no espaço de saída
 - Distâncias codificadas em cores
 - Valores baixos \Rightarrow Neurónios próximos \Rightarrow cluster
 - Valores altos \Rightarrow Neurónios longe \Rightarrow Espaço vazio

Ideal



U-mat real



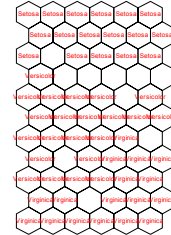
Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Calibração (ou etiquetagem)

■ Objectivo

- Identificar o *que são* os clusters
- Fazer classificação supervisionada
 - LVQ seria melhor...

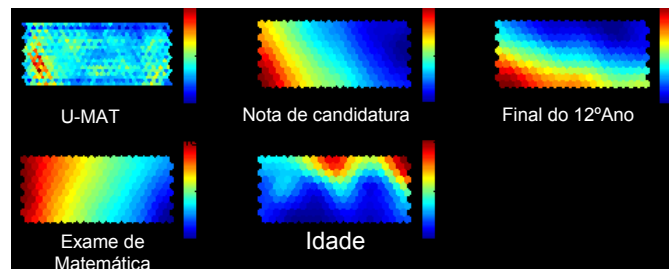


■ Como ?

- Se os dados de treino tiverem classes associadas...
- ...as suas BMU podem herdar essas classes

Planos de componentes

- Permite ver como é que uma dada variável (ou componente) varia ao longo do mapa
- Permite perceber o que define os clusters ou zonas, ou o que as forma



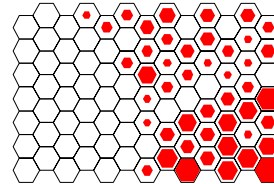
Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Outras visualizações

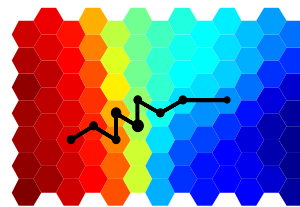
■ Hits

- Identificar quantos dados são mapeados para cada unidade



■ Trajectórias

- Observar como varia o BMU



Exemplo

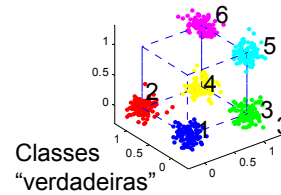
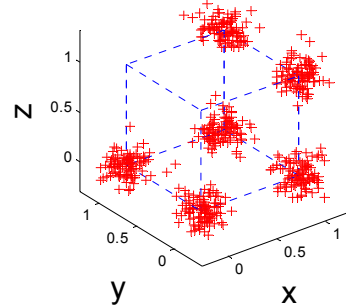
- Dados “artificiais”

Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

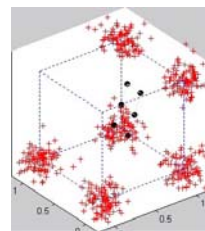
Dados de partida

- Pontos num espaço tri-dimensional
- Gerados com alguma dispersão em torno de 6 cantos de um cubo
- Código MATLAB
 - Com SOMTOOLBOX

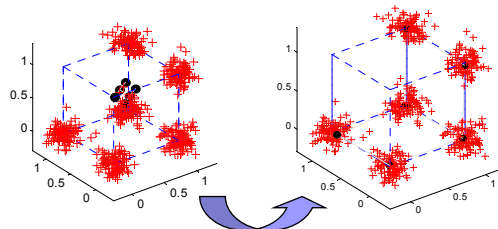


Treino de um mapa de 3x2

- Parâmetros
 - Grelha rectangular (3x2)
 - Inicialização linear
 - r inicial = 2
 - r final = 0
 - α inicial = 0.1
 - N^oIterac.=900 (1,5 épocas)



Animação

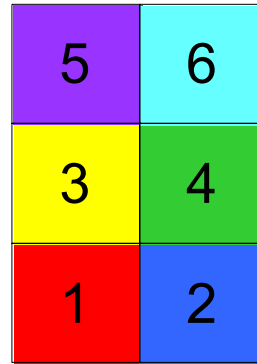
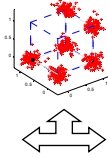


Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Visualização no espaço de output

0.79/0.44/0.61	0.80/0.66/0.62
0.67/0.49/0.43	0.73/0.71/0.41
0.56/0.52/0.29	0.67/0.74/0.25



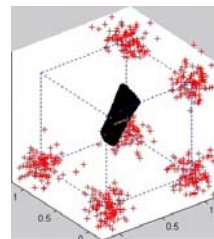
■ Durante o treino

■ Depois de calibrado com os dados originais

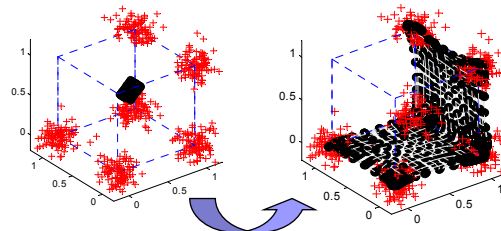
Treino de um mapa de 30x20

■ Parâmetros

- Grelha rectangular (30x20)
- Inicialização linear
- r inicial = 15
- r final = 0
- α inicial = 0.1
- N°Iterac.=900 (1,5 épocas)



Animação 900/900 training steps



Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Software e apoio

Software disponível

- SOM-PAK
 - (http://www.cis.hut.fi/research/som_lvq_pak.shtml)
 - Código C, compilável em UNIX ou MS-DOS
 - Rápido e fiável, fácil de utilizar
- Somtoolbox para MATLAB
 - (www.cis.hut.fi/projects/somtoolbox)
 - Boa visualização, facilmente alterável, “ideal” para I&D
- Muitos outros
 - SAS Enterprise Miner, SPSS-Clementine, IBM Intelligent Miner, Weka, etc...

© DSOM: Our very own software ! ☺

Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Bibliografia e apoio

- “Self-Organizing Maps”, Prof. Tuevo Kohonen
 - Springer-Verlag 2001
- Site da Universidade Técnica de Helsínquia
 - (www.cis.hut.fi/projects/somtoolbox/links)
 - **Public-domain software**
 - SOM-PAK para DOS, SOM Toolbox para MATLAB
 - **Manuais, guias, e documentação**
 - **Bibliografia extensa**
 - “www.cis.hut.fi/research/som-bibl”
 - 5384 referências em Abril de 2005

Como posso usar ?

Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

SOM-PAK 3.1

■ Colecção de programas que recebem parâmetros através do comando-linha:

- randinit- Inicializa um SOM
- vsom - Treina um SOM
- qerror - Erro de quantização
- visual - Calcular o neurónio vencedor para cada dado
- vcal - Calibrar um SOM (dar nomes aos neurónios)
- umat - Calcular a U-Matrix de um SOM
- plane - Visualizar uma das dimensões de um SOM

Formato dos dados

■ Texto simples

- Primeira linha tem informação de controlo
 - Nº de atributos de cada padrão
 - Outros: tamanho da rede, função de vizinhança, etc
- Linhas com dados
 - Valores dos atributos, opcionalmente com nomes
- Comentários
 - #

```
3
#Isto é um comentário
10 10 5 label_1
3 10 4
7 8 2 label_2
```

Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Sessão típica

1) Inicialização do SOM

- randinit : Inicializa com valores aleatórios

2) Treino do SOM

- vsom : Implementa o algoritmo de treino
- Usado 2 vezes:
 - Primeiro com vizinhanças e ritmos de aprendizagem grandes, para permitir o **desdobramento**
 - Depois, com pequenas vizinhanças e ritmos de aprendizagem para ajustes finos

Sessão típica

3) Medição do erro de quantização (opcional)

- qerror : O mapa está a representar bem os dados ?

4) Visualização

- vcal : Põe nomes (ou classes, ou labels) nos neurónios (só para problemas supervisionados)
- visual: Encontra o vencedor para cada padrão
 - Usar outro programa para visualizar de facto...
- umat : desenha a U-Matrix em ps
- plane : visualiza um dos planos

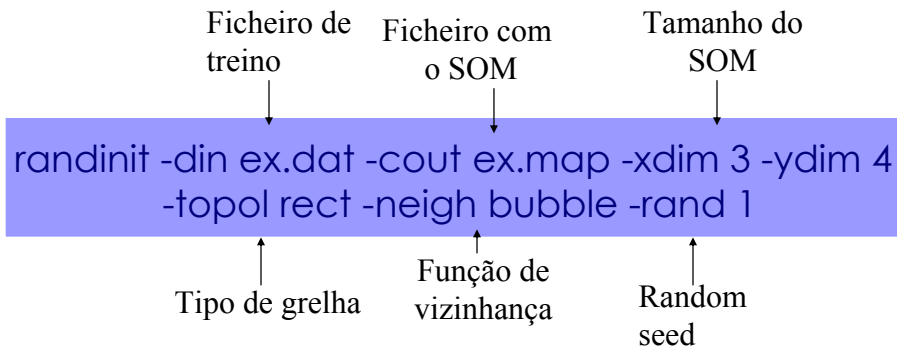
Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Inicialização do mapa

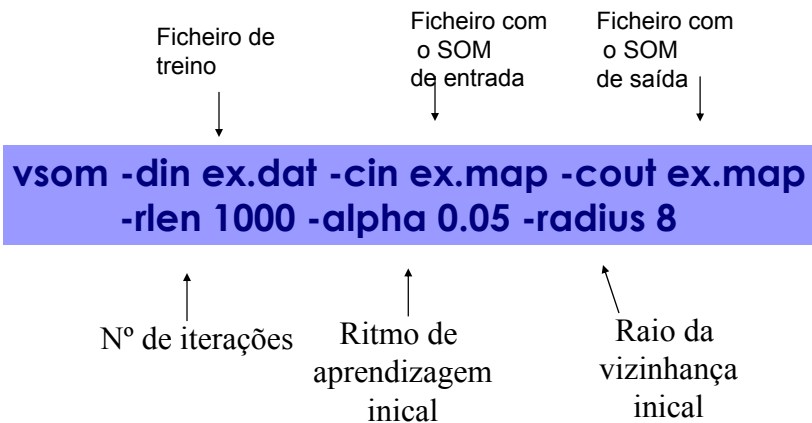
- randinit

- Usa um ficheiro de dados como protótipo



Treino do SOM

- Vsom – Correr duas vezes !



Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Visualização dos resultados

■ U-Mat

Input SOM
file

Output
Postscript file

```
umat -cin ex.map -ps 1 > output.ps
```

■ visual

Training
data file

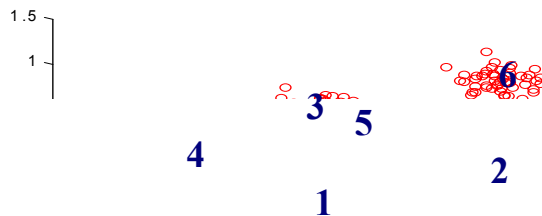
Input SOM
file

Output
file

```
visual -din ex.dat -cin ex.map -dout out.txt
```

Exemplo prático

- Vamos mapear um conjunto de pontos de um espaço 3-dimensional, que estão em vértices de um cubo

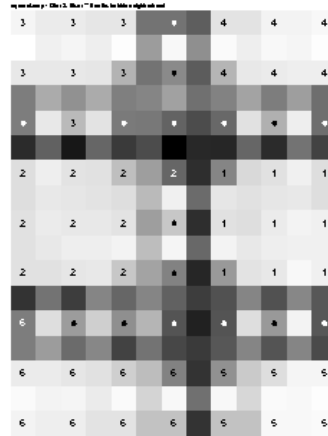


Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Exemplo prático

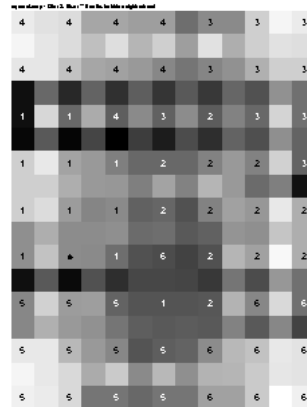
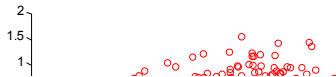
- 306 pontos centrados em 6 vértices com $\sigma=0.1$
 - Áreas escuras indicam fronteiras entre clusters



```
randinit -xdim 7 -ydim 9 -din square.dat -cout square.map -topol rect -neigh bubble  
vsom -din square.dat -cin square.map -cout square.map -rlen 1000 -alpha 0.1 -radius 7  
vsom -din square.dat -cin square.map -cout square.map -rlen 10000 -alpha 0.02 -radius 3  
vcal -din square.dat -cin square.map -cout squarel.map  
umat -cin squarel.map -ps 1 > squarel.ps
```

Outros testes

- $\sigma=0,3$

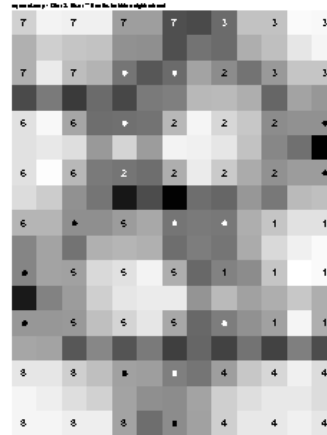
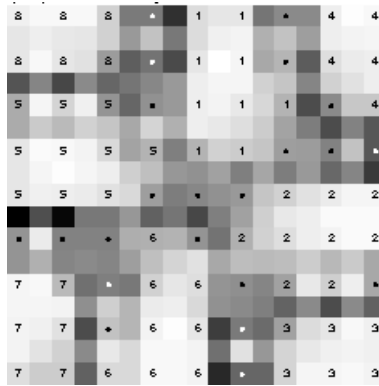


Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Outros testes

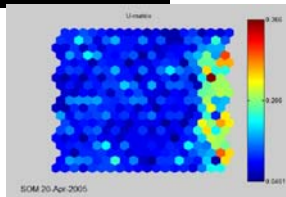
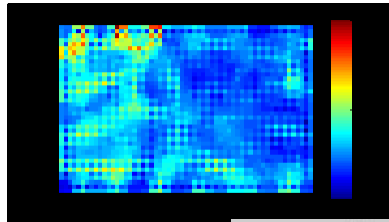
■ 8 cantos



Testes com o SOMToolbox

■ Excelentes gráficos

■ Fácil de alterar



```
load concelhos2
sD=som_data_struct(data);

sM = som_randinit_geo( sD, 'msize',[5 10], 'rect', 'sheet');
sM.neigh = 'bubble';

iterations_1 = 1000; iterations_2 = 2000;
radius_ini_1 = 5; radius_ini_2 = 2;
alpha_ini_1 = 0.7; alpha_ini_2 = 0.1;

% Standard SOM
sM1 = som_seqtrain(sM,sD,...
    'radius_ini',radius_ini_1,...
    'radius_fin',0,...
    'alpha_ini',alpha_ini_1,...
    'trainlen', iterations_1, 'epochs');
sM1 = som_seqtrain(sM1,sD,...
    'radius_ini',radius_ini_2,...
    'radius_fin',0,...
    'alpha_ini',alpha_ini_2,...
    'trainlen', iterations_2, 'epochs');
save concelhos_som sM1
[qe1,te1]=som_quality(sM1,sD);
som_show(sM1,'umat','all','footnote','Standard SOM')
sU=som_umat(sM1);
```


Mapas Auto-Organizados - SOM

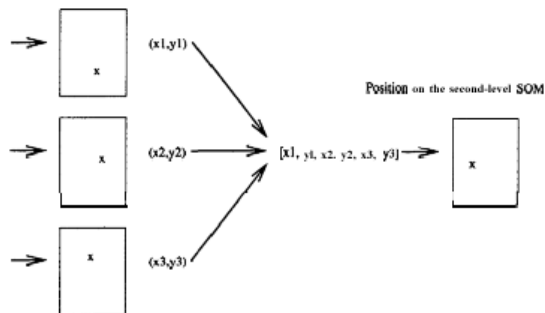
V 1.3 V.Lobo, EN 2009

Outros trabalhos com SOM

SOM multicamada e Redução da dimensionalidade

- Análise de relatórios Finaceiros [Kiviluoto 98]
 - Coordenadas do neurónio vencedor são os dados para a camada seguinte

- Por vezes todos os dados do nerónio vencedor são usados

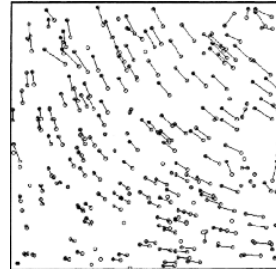


Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

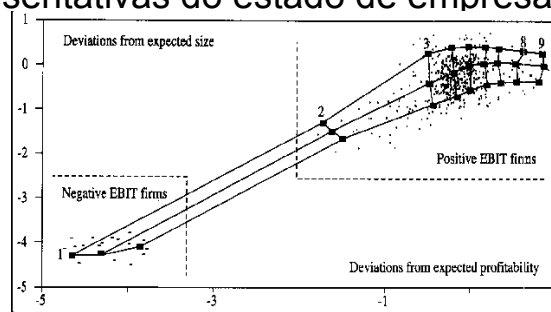
SOM para seguimento

- Análise do movimento de fluidos [Labonté 98]
 - Segue a posição de partículas num fluido
 - Analisa a trajectória dos neurónios durante a aprendizagem



SOM para amostragem

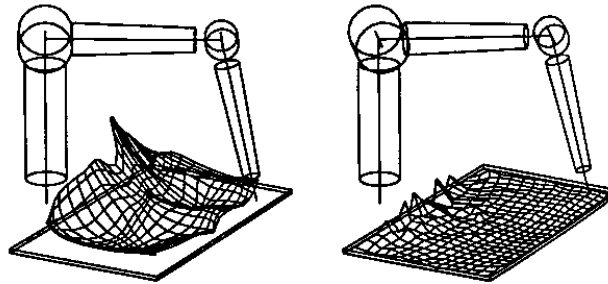
- Avaliação de empresas (contabilidade) [Trigueiros 94]
 - Mapeamento de 2D para 2D
 - Usado para seleccionar situações representativas do estado de empresas



[Trigueiros 94]

SOM em robótica

- Controlo de um braço robot [Ritter 91]
 - SOM contém os sinais de controlo necessários para atingir uma dada localização



[Ritter 91]

SOM para monitorização

- Trajectórias em SOM
 - Shu-Ching Kuo 2004

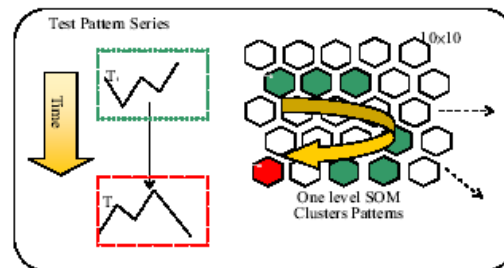


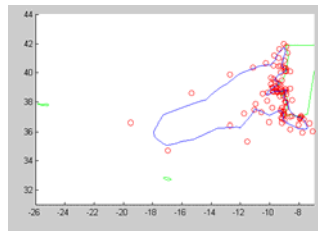
Figure 3. Trajectory analysis concept

Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

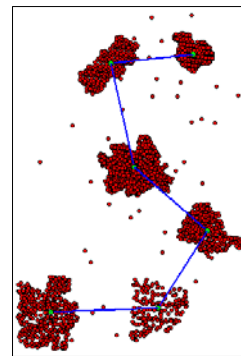
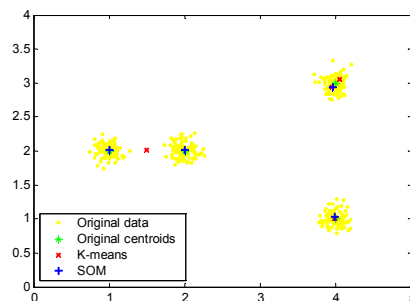
SOM para o problema do caixeiro viajante

- Problema exacto
 - Passar exactamente por n locais, fazer o planeamento de C.I. [Hueter 88][Choy 95]
- Problema relaxado
 - Determinação de rotas para patrulhas [Lobo 2005]



SOM para k-médias

- SOM 1-dimensional, com k-neurónios
- Serve de inicialização robusta [Bação 05]

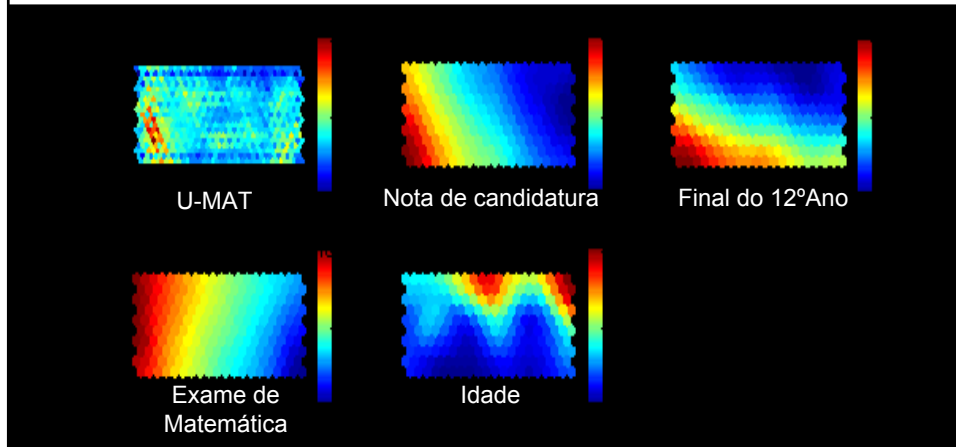


Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

SOMs e planos de componentes

- Permitem ver as diferentes variáveis



Outras aplicações de SOM

- Previsão
 - Consumos energéticos [Osowski 98]
- CIM
 - Agrupamento de ferramentas [Guerrero 98]
- Monitorização de processos (análise de trajectórias no espaço de saída), do estado de condição de máquinas, reconhecimento de voz, análise de imagem, estudos musicais, desengo de circuitos, análise geopolítica, linguística, pesquisa na Web, Economia, biologia, química, ...
etc, etc, etc, etc, etc, etc, etc...

Mapas Auto-Organizados - SOM

V 1.3 V.Lobo, EN 2009

Problema para casa: Iris

- A flor Iris tem várias variantes, 3 das quais são:

- 1 -Iris Setosa
- 2 -Iris Versicolour
- 3 -Iris Virginica

- Para 50 flores de cada uma das variantes foram medidas 4 características (medidas em cm)

- Largura da pétala
- Comprimento da pétala
- Largura da Sépala
- Comprimento da Sépala

- Questões:

- É possível determinar a variante a partir desses 4 parâmetros usando um SOM ?
- Quão parecidas são as 3 espécies ?



Iris Setosa

