

# ***U-array: Extensão da U-matrix para Mapas Auto-Organizáveis com Espaço de Saída 3-D***

José Alfredo Ferreira Costa<sup>1</sup> e Márcio Luiz de Andrade Netto<sup>2</sup>

<sup>1</sup>Departamento de Engenharia Elétrica – Centro de Tecnologia  
Universidade Federal do Rio Grande do Norte - 59.072-970 - Natal - RN

<sup>2</sup>Dept. de Engenharia de Computação e Automação - Faculdade de Engenharia Elétrica e de Computação  
Universidade Estadual de Campinas - 13083-970 - Campinas - SP  
E-mails: [alfredo@dee.ufrn.br](mailto:alfredo@dee.ufrn.br), [marcio@dca.fee.unicamp.br](mailto:marcio@dca.fee.unicamp.br)

**Abstract** – *The self-organizing map (SOM) has been widely used as a software tool for visualization of high-dimensional data. Important SOM features include information compression while trying to preserve topological and metric relationship of the primary data items. As the SOM networks are used extensively for data display, usually the output grid is 2-D. The assumption of topological preservation in SOM is not true for many data mappings involving dimension reduction. With the automation of cluster detection in SOM, higher output dimensions can be used in problems involving discovery of classes in multidimensional data. This paper presents the U-array as an extension of the U-matrix for 3-D SOM output grids. The algorithm uses the watershed transform as well as a heuristic approach for determining the image markers to perform volume segmentation. Examples of automatic class discovery using U-arrays are also presented.*

## **1. Introdução**

Análise de agrupamentos (ou classificação automática) tem como objetivo descobrir a estrutura (classes de um determinado conjunto de dados não rotulados,  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ , onde cada objeto (ou amostra)  $\mathbf{x}_i, i = 1, \dots, n$ , é descrito por  $p$  variáveis (atributos ou características). Apesar do processo ser autônomo, i.e., não supervisionado, na maioria das abordagens o usuário tem grande influência sobre o resultado final do processo através da escolha de parâmetros iniciais. Além disso, a maioria dos métodos, por exemplo estatísticos como o Expectation-Maximization (EM) [2] impõem geometrias pré-estabelecidas aos agrupamentos, além de possuírem custo computacional elevado.

O SOM – *self-organizing map* – é o modelo de redes neurais competitivas em maior uso na atualidade, e têm sido em uma gama de aplicações, incluindo agrupamento e visualização dados multidimensionais [3-5]. O método mais comum de visualização da estrutura de classes no SOM 2-D é através da *U-matrix*, que representa distâncias entre neurônios vizinhos na forma de uma superfície, ou imagem  $U(x, y)$ . As relações entre os neurônios vizinhos são vistas na superfície  $U(x, y)$  como vales e montanhas. Porém, a segmentação automática dos agrupamentos de neurônios,

representando as classes de padrões em um SOM treinado normalmente é uma tarefa complexa.

Costa e Netto [4] apresentaram um método de segmentação do SOM usando técnicas de morfologia matemática sobre a *U-matrix*. O resultado do algoritmo são regiões conectadas de neurônios rotuladas, e que definem no espaço de entrada geometrias complexas e não paramétricas. O método foi aplicado a diversos conjuntos de dados com formatos e dimensões variadas [1,4-5]. Recentemente Vesanto e Alhoniemi [6] descreveram o uso de técnicas de agrupamentos hierárquicos para aglomerar neurônios, porém há muita iteração do usuário com o resultado final obtido. Em [7], Costa e Netto apresentaram um método de segmentação do SOM utilizando a idéia de particionamento de grafos, através da eliminação de conexões (arestas) inconsistentes entre neurônios vizinhos. Tanto o número de agrupamentos quanto os membros das classes são determinados automaticamente pelo algoritmo. Aplicações com sucesso do algoritmo foram também obtidas por Silva *et al* [8] na área de detecção de fragmentos urbanos utilizando dados geoespaciais.

Uma das premissas principais da qual os métodos citados dependem é da preservação topológica efetuada pelo SOM. Em casos onde a dimensionalidade do espaço de entrada é elevado, geralmente ocorrem distorções no mapa, e o resultado da detecção de clusters pelos algoritmos fica comprometida. Este artigo apresenta o *U-array* como uma extensão da *U-matrix* para mapas com espaço de saída 3-D. A vantagem de se trabalhar com mapas com maiores espaços de saída é exatamente diminuir a interferência da redução da dimensionalidade nas distorções obtidas no processo automático de extração de classes do SOM. Como dispomos de métodos automáticos de extração de classes [1,4-9] não há necessidade de visualização. *U-arrays* de dimensão qualquer poderiam ser aplicados. Este trabalho mostra o caso 3-D e ilustra com um conjunto de dados onde duas classes são não linearmente separáveis.

O restante do artigo é organizado da seguinte forma: a seção 2 descreve-se brevemente o SOM e a *U-matrix* enquanto que a seção 3 aborda o problema de mapeamento de dados e preservação topológica no SOM. O *U-array* é apresentado na seção 4 e resultados de uma segmentação automática utilizando os métodos descritos em [1,4-5] são apresentados na seção 5. A seção 6 conclui o artigo.

## 2. O SOM e a *U-matrix*

O SOM é usado para mapear um espaço de entrada  $p$ -dimensional, contendo  $n$  padrões, para uma grade de neurônios uni ou bidimensional, com os objetivos de quantizar o espaço de entrada e representar da melhor forma possível a topologia original em um espaço de menor dimensão.

Para efetuar análise de agrupamentos, este mapeamento topologicamente ordenado geralmente não é suficiente, pois a informação de distâncias entre os neurônios é perdida. A saída de um SOM para um dado padrão inserido é geralmente o índice do neurônio vencedor  $c$ , no caso bidimensional um par de valores  $(i, j)$ , e o nível de ativação diretamente relacionado à quantização (no nosso caso a distância do padrão ao neurônio  $c$ , computado no espaço de pesos).

A *U-matrix* proposta por A. Ultsch [10], tem como objetivo de permitir a detecção visual das relações topológicas dos neurônios. A idéia básica é usar a mesma métrica que foi utilizada durante o treinamento para calcular distâncias entre pesos sinápticos de neurônios adjacentes. 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.  $(1,1), (1, 2) \dots (X, Y) \rightarrow (1,1), (1, 2) \dots (2*X-1, 2*Y-1)$ , e a intensidade de cada *pixel* na imagem  $f(x, y)$  corresponde a uma distância calculada. Pode-se pensar uma imagem como uma função tridimensional em que o valor do *pixel* na coordenada  $(x, y)$  é representado por um ponto na coordenada  $z$ . Neste caso, teremos uma superfície em 3-D cuja topografia revela a configuração dos neurônios obtida pelo treinamento. Vales, neste relevo topográfico, correspondem a regiões de neurônios que são similares, enquanto que montanhas, i.e., valores relativamente elevados na *U-matrix*, refletem a dissimilaridade entre neurônios vizinhos e podem ser associadas a regiões ou neurônios em fronteiras de agrupamentos. Pelo fato de geralmente a *U-matrix* ser uma imagem relativamente complexa, principalmente em problemas de análise de dados reais, geralmente seu uso é restrito a visualização, sendo uma ferramenta de auxílio na separação manual dos agrupamentos de um SOM [3,5,10].

Considere um mapa cujos neurônios possuem vizinhança retangular de tamanho  $X \times Y$ . Seja  $[b_{x,y}]$  a matriz de neurônios e  $[w_{i_{x,y}}]$  a matriz de pesos. Para cada neurônio em  $b$  existem três distâncias  $d_x, d_y$  e  $d_{xy}$ , na *U-matrix*.

Considerando distâncias Euclidianas, no caso da topologia da grade de neurônios ser retangular, as distâncias  $d_x, d_y$  e  $d_{xy}$  podem ser definidas como

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

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

$$dxy(x, y) = \frac{1}{2} \left( \frac{\|b_{x,y} - b_{x+1,y+1}\|}{\sqrt{2}} + \frac{\|b_{x,y+1} - b_{x+1,y}\|}{\sqrt{2}} \right) \\ = \frac{1}{2\sqrt{2}} \left[ \sqrt{\sum_i (w_{i_{x,y}} - w_{i_{x+1,y+1}})^2} + \sqrt{\sum_i (w_{i_{x,y+1}} - w_{i_{x+1,y}})^2} \right] \quad (3)$$

## 3. Redução de dimensionalidade e preservação topológica no SOM

O SOM forma um mapeamento de um espaço de entrada  $p$ -dimensional em um arranjo de neurônios, geralmente bidimensional, via aprendizado não supervisionado. Como geralmente a dimensão dos dados  $p \gg 2$  este mapeamento realiza redução de dimensionalidade. Dois aspectos estão envolvidos no mapeamento do espaço de entrada ao espaço de saída. Primeiro, os neurônios tendem a agrupar em regiões de elevada densidade de pontos, i.e., efetuam a tarefa de quantização vetorial. Por outro lado, há uma diferença básica em relação a esquemas convencionais de agrupamentos, como o *K-means*: a noção de vizinhança dos neurônios e a preservação da topologia entre os espaços. Porém, atingir os dois objetivos de quantização vetorial e preservação da topologia não é uma tarefa trivial.

Pelo fato do SOM ter sido descrito de forma algorítmica, e não extremar explicitamente um funcional conhecido durante o aprendizado, não há, para o SOM, uma função de otimização similar a Sammon [12], i.e., não há conexão formal entre o SOM a outro algoritmo de escalonamento multidimensional (MDS) [11] quanto a redução de dimensionalidade. Outra diferença fundamental entre o SOM e outros algoritmos de MDS é que no SOM o número de neurônios é diferente do tamanho do conjunto de dados,  $n$ , ocorrendo a quantização dos dados no espaço de saída.

Arranjos de neurônios em redes do tipo SOM de dimensões elevadas raramente são usados na prática por que o objetivo principal do SOM, na atualidade, é a visualização dos dados. Porém, quando usamos o SOM com uma dimensão menor que a dimensão natural dos dados quase sempre haverá alguma perda ou distorção da topologia. Por preservação da topologia podemos pensar de forma simples que as relações métricas entre os padrões no espaço original de dados devem ser mantidas no espaço de saída, seja em uma rede do tipo SOM, ou em outro processo de redução de dimensionalidade.

A principal motivação para o uso do SOM com dimensão maior do que 2 é a manutenção da topologia que geralmente é perdida quando diminuimos a

dimensionalidade via mapeamento de um espaço  $p$ -dimensional para um espaço de menor dimensão. Com a automação da descoberta de classes e relacionamentos entre dados, descritas pelos métodos descritos em [1,4,5], pode-se usar um arranjo de dimensão igual ou menor que a dimensão do espaço de entrada e fazer com que apenas os resultados finais sejam mostrados, na forma de sub-grupos de dados, o relacionamento entre os subgrupos etc.

Kiviluoto [13] caracterizou mapeamento contínuo quando vetores próximos no espaço de entrada são mapeados próximos no espaço de saída. Em relação à uma boa resolução, nenhum par de vetores que estejam distantes do espaço de entrada deveriam ser mapeados em posições próximas do espaço de saída.

Em algumas aplicações, a preservação da topologia do mapeamento é mais importante do que boa resolução. Seguindo Kiviluoto [13], deveríamos usar o mapa de forma flexível a fim de encontrar, possivelmente, componentes principais não lineares do espaço de entrada, mas de forma relativamente rígida para que o mapa não se dobre, tentando representar também os componentes menos importantes. A rigidez do mapa pode ser controlada ajustando o tamanho da influência da função de vizinhança, como proposto por Speckmann *et al.* [15].

O compromisso entre continuidade e resolução não é trivial e alguns critérios foram propostos para quantificar estas propriedades. O erro de quantização é simples de ser avaliado, sendo computado como a distância média das amostras do conjunto de dados  $X$  aos neurônios mais próximos. Para quantificar a continuidade vários critérios foram propostos na literatura [5,13]. O mais simples, o Erro Topográfico,  $\xi_i$ , é obtido considerando a adjacência dos campos receptivos e a proporção dos objetos que indicam a descontinuidade local do mapeamento. Dado um padrão  $x \in X$ , seja  $m_i$  e  $m_j$  o primeiro e o segundo vetores de pesos mais próximos de  $x$ . Caso os neurônios correspondentes  $n_i$  e  $n_j$  sejam adjacentes o mapeamento preserva, localmente, a topologia [13]. Caso contrário, i.e., se  $n_i$  e  $n_j$  não forem adjacentes há um erro topográfico local. O erro topográfico para o mapeamento é obtido somando-se todos os erros topográficos locais para todos os padrões.

$$\xi_i = \frac{1}{n} \sum_{k=1}^n u(x_k), \text{ onde } u(x_k) = \begin{cases} 1, & \text{caso o primeiro e segundo neurônios} \\ & \text{vencedores sejam adjacentes;} \\ 0, & \text{caso contrário.} \end{cases}$$

onde  $n$  é o número total de padrões.  $\xi_i$  corresponde a uma proporção do número de padrões que foram mapeados corretamente. Porém, não é descrito o tipo de mapeamento incorreto. Por exemplo, não há informação caso dois padrões próximos sejam mapeados em neurônios que distem de uma unidade ou que estejam em cantos opostos do mapa.

### 3.1. Exemplo de mapeamento de um espaço 2-D em um SOM unidimensional

Apresenta-se um exemplo simples de um conjunto de dados no  $\mathcal{R}^2$  com um mapa com espaço de saída unidimensional.

A figura 1 mostra um mapa com 40 neurônios, após 1000 épocas de treinamento. A figura ilustra também o diagrama de Voronoï, o qual podemos identificar as células de influência dos neurônios, ou campos receptivos induzidos. Note a contorção do *grid* de neurônios no espaço de dimensão maior que a dimensão do mapa para atingir o objetivo de quantização espacial. Os dados foram gerados a partir de médias centradas nos vértices de um triângulo equilátero, com ruído Gaussiano com desvio padrão 0.15. Vemos que apesar das classes serem equidistantes no espaço de entrada, caso utilizássemos a informação do mapa, de forma automática, iríamos pensar que as classes 2 e 3 (parte superior e parte direita da figura) estão em porções distantes do espaço de atributos, pois foram mapeadas em cantos opostos do mapa.

Por exemplo, um dado padrão  $x' = (0.2, 0.2)$  apresenta erro local topográfico, pois possui como neurônios mais próximos 13 e o 32, respectivamente. A figura 2 ilustra a configuração de neurônios e as regiões no espaço dos atributos onde há erro topográfico local. Apenas os objetos que caírem nas regiões marcadas em preto contribuirão para o erro topográfico.

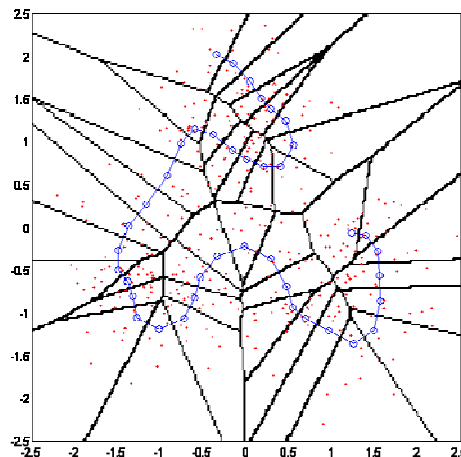


Figura 1: Grid de neurônios e diagrama de Voronoï.

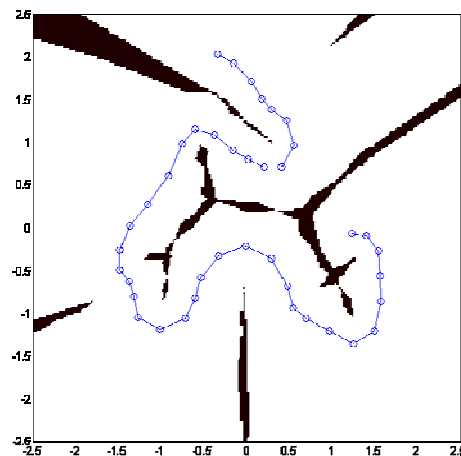


Figura 2 - Configuração de neurônios e as regiões no espaço que possuem erro topográfico local.

Tais problemas ocorrem devido ao fato do SOM estar efetuando quantização vetorial ao mesmo tempo em que

tenta reduzir a dimensionalidade, lembrando algoritmos de escalonamento multidimensional. Porém, aplicando-se (e estendendo-se) os métodos de segmentação de imagens 2-D, desenvolvidas para os mapas bidimensionais, para mapas com dimensão 3 ou maior, pode-se extrair, além do correto número das classes, suas informações de proximidades relativas, que é de importância em mineração de dados.

#### 4. *U-array*: Extendendo a análise para mapas com grid 3-D

Propõe-se o *U-array* como uma extensão da *U-matrix*. Para o caso de SOM com dimensão do espaço de saída 3 podemos ainda visualizar as relações entre os neurônios efetuando-se cortes no *array* tridimensional. Para dimensões maiores do espaço de saída, apesar de não podermos visualizar em uma única imagem as relações entre os neurônios, podemos escolher combinações de pares de componentes e visualizar imagens bidimensionais como projeções do espaço de maior dimensão.

A extensão para casos de dimensão acima de 3 pode ser efetuada. Idealmente deveríamos escolher a dimensão do espaço de saída do SOM através de informações provenientes diretamente dos dados, por exemplo, usando alguma informação derivada da dimensão fractal [14].

No caso do *U-array* onde o SOM tem espaço de saída tridimensional, temos, além das distâncias  $dx$ ,  $dy$  e  $dxy$ , as distâncias  $dz$ ,  $dxz$ ,  $dyz$  e  $dxyz$ . Considere um mapa retangular de tamanho  $X \times Y \times Z$ , no caso tridimensional. Da mesma forma que na *U-matrix*, o *U-array* terá tamanho  $(2X-1) \times (2Y-1) \times (2Z-1)$ . Por exemplo, para um mapa com tamanho  $3 \times 3 \times 3$  teremos um *U-array* de tamanho  $5 \times 5 \times 5$ , e poderíamos usar a informação das *U-matrizes* de cada mapa e interpolar nas posições interiores do *U-array*. As distâncias entre os neurônios podem ser vistas de forma mais simples para um mapa com tamanho  $2 \times 2 \times 2$ , como o apresentado na figura 3.

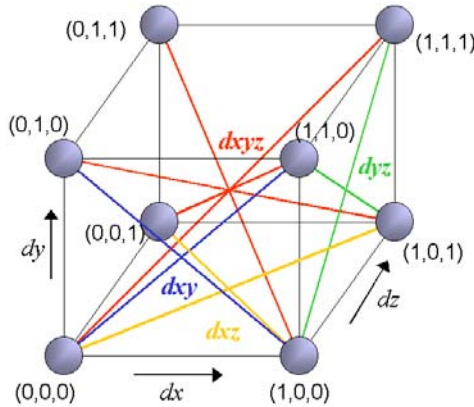


Figura 3 - Ilustração do mapa de tamanho  $2 \times 2 \times 2$

Apenas uma distância de cada tipo é apresentada na figura 6 por questões de visibilidade, onde aparece também o endereço ou índice do neurônio no espaço de saída da rede.

As distâncias  $dx$ ,  $dy$  e  $dz$  (equações 1-3) também são mostradas na figura 3. A notação usada nesta seção é a mesma da seção 2. Seja  $[b_{x,y,z}]$  a matriz de neurônios e  $[w_{i_{x,y,z}}]$  a matriz de pesos. As distâncias  $dxy$ ,  $dxz$  e  $dyz$  são semelhantes à distância usada no caso bidimensional,  $dxy$ , apresentada pela equação 3. As fórmulas para as outras distâncias são apresentadas nas equações 4-10.

$$dx(x, y, z) = \|b_{x,y,z} - b_{x+1,y,z}\| = \sqrt{\sum_i (w_{i_{x,y,z}} - w_{i_{x+1,y,z}})^2} \quad (4)$$

$$dy(x, y, z) = \|b_{x,y,z} - b_{x,y+1,z}\| = \sqrt{\sum_i (w_{i_{x,y,z}} - w_{i_{x,y+1,z}})^2} \quad (5)$$

$$dz(x, y, z) = \|b_{x,y,z} - b_{x,y,z+1}\| = \sqrt{\sum_i (w_{i_{x,y,z}} - w_{i_{x,y,z+1}})^2} \quad (6)$$

$$dxy(x, y, z) = \frac{1}{2} \left( \frac{\|b_{x,y,z} - b_{x+1,y+1,z}\|}{\sqrt{2}} + \frac{\|b_{x,y+1,z} - b_{x+1,y,z}\|}{\sqrt{2}} \right) \quad (7)$$

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

$$dxz(x, y, z) = \frac{1}{2} \left( \frac{\|b_{x,y,z} - b_{x+1,y,z+1}\|}{\sqrt{2}} + \frac{\|b_{x,y,z+1} - b_{x+1,y,z}\|}{\sqrt{2}} \right) \quad (8)$$

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

$$dyz(x, y, z) = \frac{1}{2} \left( \frac{\|b_{x,y,z} - b_{x,y+1,z+1}\|}{\sqrt{2}} + \frac{\|b_{x,y+1,z} - b_{x,y,z+1}\|}{\sqrt{2}} \right) \quad (9)$$

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

A distância  $dxyz$  é uma média das distâncias entre os vértices opostos do cubo:

$$dxyz(x, y, z) = \frac{1}{4\sqrt{3}} \left[ \sqrt{\sum_i (w_{i_{x,y,z}} - w_{i_{x+1,y+1,z+1}})^2} + \sqrt{\sum_i (w_{i_{x,y,z+1}} - w_{i_{x+1,y+1,z}})^2} + \sqrt{\sum_i (w_{i_{x,y+1,z+1}} - w_{i_{x+1,y,z}})^2} + \sqrt{\sum_i (w_{i_{x,y+1,z}} - w_{i_{x+1,y,z+1}})^2} \right] \quad (10)$$

Da mesma forma que na *U-matrix*, o cálculo dos valores  $du(x, y, z)$  pode ser feito tomando-se o valor médio ou a mediana dos elementos circunvizinhos.

#### 4.1 Adaptação do algoritmo *SL-SOM* para o *U-array*

Obviamente, a segmentação de um cubo, ou hipercubo, é mais complexa do que de uma imagem. No caso de um cubo, o gráfico de regiões planas conectadas *versus* limiar (nível de cinza) torna-se um gráfico de um volume conectado *versus* o limiar. A conectividade entre *pixels* torna-se conectividade entre *voxels*. O caso mais simples em 2-D, i.e., conectividade 4-adjacente, pode ser estendida e computada de forma relativamente simples em uma

imagem  $p$ -dimensional: para qualquer coordenada caso a distância entre dois *pixels* seja 1 eles estarão conectados.

A segmentação de uma imagem 3-D pode também ser vista como a segmentação de uma seqüência de imagens 2-D. O algoritmo *watershed* foi adaptado para efetuar tal operação (em 3-D). Um dos principais problemas é a detecção dos marcadores, suavização da imagem e eliminação de *voxels* isolados. A operação rotulação de regiões conectadas passa a ser uma operação de rotulação de volumes conectados. Os tipos de conectividades utilizados neste trabalho foram os padrões 6 e 26-adjacente. Maiores detalhes em [5].

## 5. Exemplo de segmentação de mapas com dimensão superior a 2.

Um exemplo para comparações de métodos de agrupamentos em dados multidimensionais foi proposto por Utsch [16], o *chainlink*, que consiste de 1000 pontos no espaço  $\mathcal{R}^3$  tal que eles possuem a forma de dois anéis tridimensionais entrelaçados. Um dos anéis se estende na direção  $x$ - $y$  enquanto o outro se estende na direção de  $x$ - $z$ . Os dois anéis podem ser pensados como elementos de uma corrente, cada um consistindo de 500 objetos de dados.

Este problema ilustra a capacidades do SOM em descobrir a estrutura dos dados mesmo para conjuntos de dados com formas complexas e não-esféricas, e não separáveis linearmente (figura 4).

O resultado da aplicação deste conjunto de dados em um mapa 2-D foi apresentada em Costa & Netto [17]. Apesar do resultado em 2-D ser suficiente para este problema, onde os dados estão no  $\mathcal{R}^3$ , a motivação para o uso desta base com um SOM com espaço de saída 3-D é ilustrativa, no sentido de tentar entender a adequação do mapa a problemas em que não há redução de dimensionalidade, e a topologia pode efetivamente ser preservada.

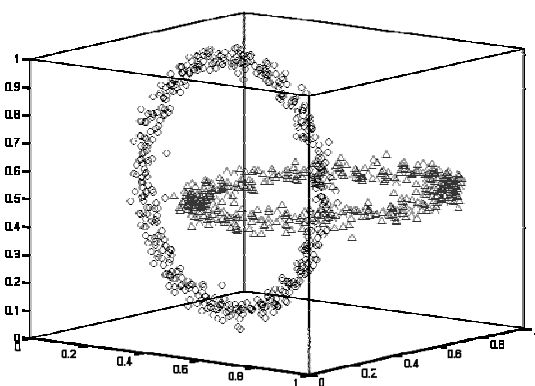


Figura 4 - Ilustração do conjunto de dados *chainlink*

O SOM utilizado no experimento tem  $8 \times 8 \times 8$  neurônios e o número de épocas (algoritmo batch) foi 200. A figura 5 ilustra a configuração da rede após o treinamento enquanto que o *U-array* correspondente é mostrado na figura 6, na forma de um cubo (3-D), onde níveis de cinza escuros simbolizam dissimilaridade mais acentuada.

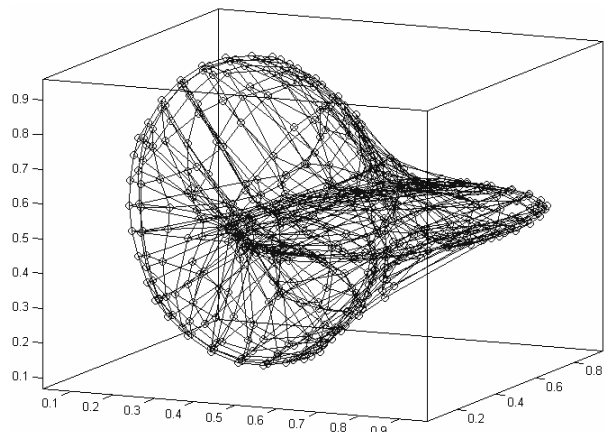


Figura 5: SOM após 200 épocas de treinamento (alg. batch).

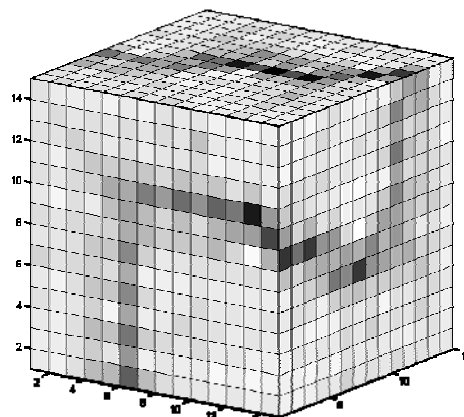


Figura 6: *U-array* relativo à rede mostrada na fig.9 (forma 3-D)

Um dos maiores problemas neste tipo de análise é a segmentação correta de imagens no  $\mathcal{R}^p$ ,  $p > 2$ . O algoritmo *watershed* foi estendido para suportar matrizes multidimensionais. A escolha correta dos marcadores, de forma automática, é sem dúvida, um dos elementos chaves no processo. Os marcadores utilizados foram obtidos pela binarização do *U-array* utilizando o valor inicial (40) do platô de estabilidade no gráfico número de volumes conectados *versus* o limiar, cuja faixa foi de 40 a 132, estável para a solução dois agrupamentos, solução obtida automaticamente pelo método SL-SOM [5].

Após segmentação, os rótulos obtidos do *U-array* são copiados para as posições respectivas do mapa de Kohonen, figura 8. A figura 9 ilustra a configuração dos neurônios eliminando o efeito dos neurônios inativos [5],  $H(i) = 0$ . A influência espacial dos agrupamentos detectados é ilustrado, limitando o raio das esferas em 0.05. Relações entre neurônios vizinhos também são mostradas. Vê-se que a solução obtida pelo método é coerente para este conjunto de dados não linearmente separável. Note que ambos os agrupamentos de neurônios fecham os círculos relacionados aos agrupamentos originais dos dados. No caso bidimensional, a deformação da rede 2-D apenas simula tal fechamento. Vale ressaltar que métodos como o *K-means* ou o baseado em misturas de densidades Gaussianas foram incapazes de resolver este problema, particionando o conjunto de dados por hiperplanos (no caso

do *K-means*) e superfícies quadráticas (maiores detalhes em [1, 4,5,17]).

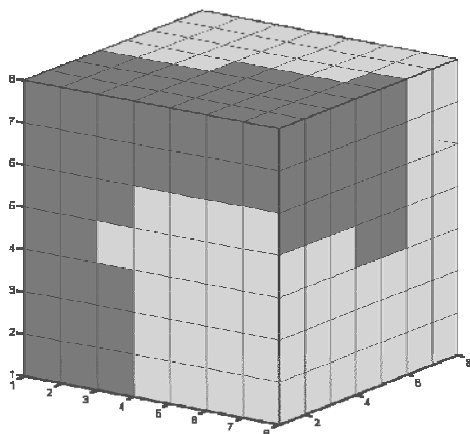


Figura 7: Espaço de saída da rede SOM rotulada (dois volumes segmentados) - representação 3-D.

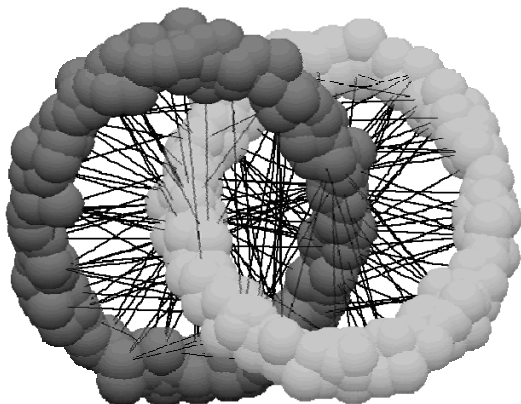


Figura 8: Configuração dos neurônios eliminando o efeito dos neurônios inativos (raio de influência do neurônio = 0.05).

## 6. Conclusões

A principal contribuição deste artigo é mostrar que os métodos de análise de dados utilizando o SOM [1,4,5-7,9,17] podem ser estendidos para redes com espaços de saída com dimensão maior que 2. Atualmente grande parte das aplicações e do uso dos mapas de Kohonen usam espaços de saída bidimensionais porque a principal função é a visualização do mapeamento do espaço original (de elevada dimensão) no display ou conjunto de neurônios da rede. Porém, geralmente há perda de relações topológicas entre mapeamentos de espaços com dimensão mais elevada para espaços com menor dimensão. Em aplicações de mineração de dados e descoberta de conhecimento em bases de dados, pode ser importante saber não só a existência das classes e seus elementos componentes, mas também que classes são semelhantes. Trabalhos futuros podem incluir tais medidas de erro topológico como indicadores para ajuste da dimensão do espaço de saída do SOM, de forma que tenhamos algoritmos que possam adaptar automaticamente a dimensão da rede para um dado problema, como sugerido em [14].

## Referências

- [1] Costa, J.A.F., & Netto, M. L. A., "Clustering of complex shaped data sets via Kohonen maps and mathematical morphology". In: *Proc. of the SPIE*, B. Dasarathy (Ed.), Vol. 4384, pp. 16-27, 2001.
- [2] Bishop, C.M., *Neural Networks for Pattern Recognition*, Oxford: Oxford University Press, 1995.
- [3] Kohonen, T., *Self-Organizing Maps*, 2<sup>nd</sup> Ed., Berlin: Springer, 1997.
- [4] Costa, J.A.F. and Netto, M.L.A., Estimating the Number of Clusters in Multivariate Data by Self-Organizing Maps. *Intl. Journal of Neural Systems*, vol. 9, pp. 195-202, 1999.
- [5] Costa, J. A. F., *Classificação Automática e Análise de Dados por Redes Neurais Auto-Organizáveis*. Tese de Doutorado, Unicamp, SP, 1999.
- [6] Vesanto, J. and Alhoniemi, E. (2000). Clustering of the Self-Organizing Map. *IEEE Trans. Neural Networks*, 11, (3), pp. 586-602.
- [7] Costa, J.A.F., e Netto, M.L.A. (2003) Segmentação do SOM Baseada em Particionamento de Grafos. *Anais do VI Congresso Brasileiro de Redes Neurais*, São Paulo, Junho de 2003, pp. 451-456.
- [8] Silva, M.A.S., Monteiro, A.M.V., and Medeiros, J.S. (2004). Semi-Automatic Geospatial Data Clustering by Self-Organizing Maps. In: *Anais do Simpósio Brasileiro de Redes Neurais*, São Luiz, MA, 2004.
- [9] Costa, J.A.F. and Netto, M.L.A., A new tree-structured self-organizing map for data analysis. In: *Proc. of the Intl. Joint Conf. on Neural Networks*, Washington, DC, July 2001, pp. 1931-1936.
- [10] Ultsch, A., Self-Organizing Neural Networks for Visualization and Classification. In: O. Opitz et al. (Eds). *Information and Classification*. Springer, Berlin, 307-313, 1993.
- [11] Torgerson, W. S. (1952). Multidimensional Scaling I. Theory and Method. *Psychometrika*, vol. 17, pp. 401-419.
- [12] Sammon, J. W. (1969). A Non-Linear Mapping for Data Structure Analysis. *IEEE Trans. in Computers*, vol. 18, pp. 401-409.
- [13] Kiviluoto, K. (1996). Topology preservation in Self-Organizing Maps. In: *Proc. IEEE Intl. Conf. Neural Networks*, v. 1, pp. 249-254.
- [14] Bauer, H.-U., Herrmann, M., & Villmann, T. (1999). Neural maps and topographic vector quantization. *Neural Networks*, June.
- [15] Speckmann, H., Raddatz, G. & Rosenstiel, W. (1994). Considerations of geometrical and fractal dimension of SOM to get better learning results. In: *Proc. ICANN'94*, vol. I, pp. 342-345.
- [16] Ultsch, A. (1995). Self-Organizing Neural Networks perform different from statistical k-means clustering. *Gesellschaft für Klassifikation*, Basel.
- [17] Costa, J.A.F., & Netto, M. L. A., "Cluster Analysis Using Self-Organizing Maps and Image Processing Techniques". In: *Proc. of the IEEE Conf. on SMC*, v. 5, pp. 367-372, Tokyo, Japan.