UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

MÉTODOS DE SEGMENTAÇÃO DE MAPAS AUTO-ORGANIZÁVEIS PARA ANÁLISE DE AGRUPAMENTO

LEONARDO FREITAS NOLETO

FLORIANÓPOLIS – SANTA CATARINA
2007

LEONARDO FREITAS NOLETO

MÉTODOS DE SEGMENTAÇÃO DE MAPAS AUTO-ORGANIZÁVEIS PARA ANÁLISE DE AGRUPAMENTO

Trabalho de conclusão de Curso apresentado como exigência para a obtenção do titulo de bacharel em Ciências da Computação da Universidade Federal de Santa Catarina - UFSC.

Orientador: **Prof. Dr. Mauro Roisenberg**

Banca examinadora

_____

Paulo José de Freitas Filho

_____

Paulo José Ogliari

FLORIANÓPOLIS – SANTA CATARINA

2007

Sumário

Lista de Abreviaturas, Siglas e Símbolos

RNA – Redes Neurais Artificiais

SOM – Self Organizing Map (Mapa Auto-Organizável)

AA – Análise de Agrupamentos

Lista de figuras

Lista de tabelas

Resumo

Este trabalho investiga duas técnicas de segmentação do espaço de saída do mapa auto-organizável (rede de Kohonen) como métodos de análise de agrupamento. A segmentação da U-matriz por segmentação morfológica e o particionamento de grafos. Na primeira técnica, o algoritmo de *watershed* é aplicado a U-matriz. Após a segmentação, o número de regiões conectadas reflete o número de agrupamentos presente nos dados. As duas técnicas são explicadas e confrontadas com o algoritmo estatístico k-means. Um conjunto de dados formado por mistura de gaussianas e um conjunto de dados conhecido como *chainlink* são usados para testes. Os resultados são comparados através da estatística lambda de Wilk's.

Palavras chaves: análise de agrupamentos, mapas auto-organizáveis, SOM, k-médias, U-matriz e particionamento por grafo.

Abstract

In this work, we investigate two segmentation techniques in the output space of the Self-Organizing Maps (Kohonen Neural Network) for clusters analysis. The segmentation of the U-matrix for morphologic segmentation and segmentation of graphs. In the first technique, the algorithm of watershed is applied to the U-matrix. After the segmentation, the number of conected regions reflects the present number of clusters in data set. These techniques are explained and collated with the statistical algorithm k-means. A mixture gaussian data set and chainlink data set are used for tests. The results are compared through the lambda Wilk's statistics.

Keywords: clusters analysis, Self-Organizing Maps, k-means, U-matrix and graph segmentation.

1. Introdução

Diariamente uma quantidade imensa de dados são coletadas e armazenadas nos bancos de dados, como por exemplo, nos bancos, hospitais, indústrias, economia, geologia e etc. Com esta crescente disponibilidade de grandes massas de dados, cresceu também a busca pela informação que estes dados "escondem". Analisar e visualizar esse grande volume de dados na forma de registros, descritos por *p* atributos, suas inter-relações, similaridades inerentes, etc, torna-se um problema bastante difícil, principalmente pelo fato de que frequentemente a dimensionalide dos dados é superior a 3. Desta forma, a necessidade de métodos que possam analisá-los de forma automática torna-se cada vez maior.

A análise de agrupamentos (*cluster analysis*) é uma sub-área de análise multivariada, que por sua vez é uma sub-área da estatística, dedicada a análise de problemas onde amostras são descritas por variáveis (atributos) *p*-dimensionais. Basicamente, segundo (BUSSAB *et. al*., 1990), a análise de agrupamentos, ou simplesmente AA, é um processo de classificação não supervisionado que engloba uma variedade de técnicas e algoritmos parar agrupar objetos (dados) em grupos similares.

Recentemente a análise de agrupamentos tem recebido bastante interesse da comunidade científica e nas empresas, sendo usada como ferramenta básica nas áreas de mineração de dados (*datamining*) e descoberta de conhecimento (*knowledge discovery*) (CARVALHO, 2001).

As aplicações de interesse podem ter diferentes objetivos, como por exemplo, a determinação de objetos que sejam semelhantes ou o enfoque em determinada classe de objetos. Pode-se assim fazer uma síntese do banco de dados observando os objetos representantes de cada subgrupo, que vai confirmar ou não, hipóteses a respeito da massa de dados em questão. Pode–se também formular hipóteses sobre a estrutura dos dados e determinar esquemas de classificação.

A análise de agrupamentos traz consigo uma série de benefícios, tais como (Prass, 2004):

- possibilita ao usuário encontrar grupos úteis;
- auxilia no entendimento das características do conjunto de dados;
- pode ser usada na geração de hipóteses;
- permite predição com base nos grupos formados.

- permite o desenvolvimento de um esquema de classificação para novos dados.

Existem diversos algoritmos de AA e as ferramentas de mineração de dados trazem implementados tipos específicos destes algoritmos. As particularidades de cada algoritmo e das medidas de similaridade influenciam decisivamente na geometria dos grupos encontrados. Isso porque os métodos de AA fazem suposições implícitas sobre o tipo de geometria presente nos dados (CARVALHO, 2001). Como exemplo disto, pode-se citar o fato de alguns algoritmos encontrarem facilmente grupos esféricos, mas possuírem dificuldades para encontrar grupos com formato cilíndrico ou com formato aleatório A escolha do algoritmo errado trará como conseqüência resultados equivocados e inúteis (PRASS, 2004).

Os métodos tradicionais de AA necessitam um conhecimento avançado do domínio por parte do usuário e também e, o que talvez seja o maior inconveniente destas técnicas, deixa ao encargo do usuário definir a priori o número de agrupamentos (BUSSAB *et. al*., 1990).

Os mapas auto-organizáveis (ou *Self-organizing Maps* – SOM), também conhecidos como redes de Kohonen (KOHONEN, 2001), têm sido usados largamente como uma ferramenta de visualização e análise de dados apresentados em dimensões elevadas. O mapa auto-organizável define, via treinamento não supervisionado, um mapeamento de um espaço *p*-dimensional contínuo para um conjunto discreto de vetores de referência, ou neurônios, geralmente dispostos na forma de uma matriz. Cada neurônio tem a mesma dimensão do espaço de entrada, *p*, e o objetivo principal do treinamento é reduzir dimensionalidade ao mesmo tempo em que tenta-se preservar, ao máximo, a topologia do espaço de entrada (COSTA, 1999 e HAYKIN, 2001).

O uso do mapa auto-organizável em análise de agrupamentos requer ferramentas adicionais. Em um mapa tradicional, a única informação de saída quando apresentamos um padrão, *x*, são os índices do neurônio vencedor, *c*, e o erro de quantização, que pode ser dado pela distância entre o padrão de entrada *x* e o neurônio *c* (HAYKIN, 2001). Geralmente usam-se informações da classe dos padrões mais freqüentes para rotular neurônios em um mapa organizado.

O mapa funciona como uma rede elástica ocupando o espaço *p*-dimensional de forma a representar de melhor maneira, dada uma topologia de vizinhança entre os neurônios, as regiões do espaço com maior densidade de pontos (HAYKIN, 2001). Existem basicamente quatro formas distintas de visualizar as relações entre os neurônios

do espaço de saída do mapa: representação dos pesos no espaço $\Re^p$, histograma, mapas contextuais e U-matriz.

Este trabalho busca por soluções automáticas de descoberta de agrupamentos nos dados usando a propriedade de aproximação de densidade de probabilidade do espaço de entrada pelo mapa, assumindo que a única informação disponível são os dados, descritos por *p* atributos, buscamos descobrir a estrutura inerente dos dados e agrupá-los. A análise de agrupamento via mapa auto-organizável é feita pela segmentação do espaço de saída de um mapa treinado por duas técnicas: a segmentação da U-matriz com morfologia matemática e o particionamento de grafos, técnicas essas tema-objeto do presente trabalho.

Três conjuntos de dados foram submetidos as duas técnicas de segmentação apresentadas e a técnica estatística das k-médias. Os resultados são apresentados e avaliados segundo a tabela de distâncias Mahalanobis entre grupos e o fator lambda de Wilk's.

Tema

Existem algumas abordagens conhecidas na literatura para a investigação do espaço de saída do mapa auto-organizável. Geralmente as ferramentas fornecem apenas um instrumento de visualização para indicar tendências de agrupamentos e há necessidade de intervenção do usuário que guia manualmente a escolha dos parâmetros e segmentação do espaço de saída do mapa (SILVA, 2004).

Este trabalho explora duas técnicas de segmentação automática do espaço de saída do mapa treinado capaz de determinar e agrupar os agrupamentos de neurônios formados pela organização topológica após a execução do algoritmo de treinamento. Agrupamentos de neurônios podem ser usados para a constituição de grupos no conjunto de dados desejado.

Delimitação do Tema

Este trabalho aborda duas técnicas de segmentação do espaço de saída do mapa auto-organizável treinado: segmentação morfológica da U-matriz e particionamento por grafos. É analisado as características, o conceito matemático e estatístico no qual elas se fundamentam sua implementação e os domínios de problemas em que cada técnica pode

ser melhor aplicada. Desta forma, a proposta do trabalho concerne à análise das técnicas como são encontradas na bibliografia relacionada. O trabalho não sugere nenhuma outra técnica e também não se concentra nos aspectos de otimização ou eficiência dos algoritmos implementados.

O método de segmentação por U-Matriz utiliza vários conceitos de processamento de imagens. Este trabalho introduz alguns conceitos referentes à morfologia matemática e algoritmos de segmentação de imagem na medida em que forem necessários para o entendimento da técnica explicada. As técnicas de processamento de imagens é abordada na maioria dos textos de computação gráfica e está além do alcance deste trabalho.

A técnica de particionamento por grafo utiliza os conceitos da teoria de grafos para sua implementação. Da mesma forma, uma bibliografia sobre teoria dos grafos pode ser encontrada em (RABUSKE, 1992).

Objetivo Geral

Estudar as duas técnicas de análise de agrupamento utilizando mapas auto-organizáveis existente na bibliografia relacionada. Na abordagem de cada técnica são exploradas as características, as aplicabilidades e suas vantagens e desvantagens. O método de particionamento pelas *k*-médias, pertencente às técnicas tradicionais, também será relacionada com os técnicas analisadas neste trabalho, visto que ele é um dos métodos mais conhecidos para análise de agrupamentos.

Objetivo Específico

- investigar as técnicas de segmentação da U-matriz com morfologia matemática e particionamento de grafos para análise de agrupamentos;
- Implementar e realizar experimentos no ambiente MATLAB utilizando bibliotecas de redes neurais de cada método estudado;
- Comparar alguns resultados obtidos pelo métodos tradicional (k-médias) com as técnicas baseadas em mapa auto-organizáveis.

Motivações

A descoberta de agrupamentos de dados, ou seja, a tarefa de dividir um conjunto de objetos em grupos menores e mais homogêneos é uma operação fundamental para mineração de dados. (HUANG, 1997 *apud*. PRASS, 2004).

As técnicas estatísticas tradicionais exigem muita interferência do usuário na definição dos parâmetros. Se o processo é repetido varias vezes, como geralmente é feito para a descoberta de um bom particionamento, este processo se torna cansativo e enfadonho. Buscar técnicas automáticas de detecção de agrupamentos e que ainda possam garantir um bom grau de acurácia em relação aos agrupamentos formados torna a análise de agrupamento uma tarefa mais fácil de ser realizada nos processos de mineração de dados. As características de ordenação de topológica dos dados, a compactação dimensional e a manutenção da densidade dos dados, tornam os mapas auto-organizáveis o modelo neural mais apropriado para a primeira etapa do processo de análise de agrupamento automático e com bom grau de qualidade.

Ferramentas utilizadas

A implementação das técnicas e os experimentos foram desenvolvidos no ambiente MATLAB® 7. Foi utilizada a ferramenta "SOM toolbox *for MATLAB*" para a utilização de mapas auto-organizáveis e a para morfologia matemática "*SDC Morphology Toolbox for MATLAB*".

Estrutura do trabalho

Esta monografia é apresentada em 6 capítulos, onde busca-se inicialmente apresentar o problema da análise de agrupamentos e motivações para o uso de técnicas neurais para resolvê-lo.

O capítulo 2 apresenta uma breve introdução aos métodos de análise de agrupamentos estatísticos. São descritos os métodos hierárquicos, particionais, baseados em densidade, baseados em grade e baseados em modelo.

O capítulo 3 descreve o modelo de rede neural do mapa auto-organizável que é a base para a análise dos dois métodos de segmentação. É descrito sua arquitetura, seu

processo de aprendizagem, o algoritmo de Kohonen e os métodos de visualização do mapa treinado.

O capítulo 4 apresenta a técnica de segmentação da U-matriz com o algoritmo de watershed e a técnica de particionamento de grafos para a descoberta automática de agrupamentos em conjunto de dados.

O capítulo 5 apresenta o resultado da aplicação das duas técnicas explicadas no capítulo anterior a três conjuntos de dados. É apresentado também o resultado da análise de agrupamento pelo algoritmo das k-médias. Finalmente, os resultados de cada técnica são comparados utilizando o teste lambda de Wilk's e a matriz de distância Mahalanobis entre grupos.

Finalmente, o capítulo 6 aponta as conclusões deste trabalho e comentários sobre as técnicas tradicionais e a baseada em mapas auto-organizáveis.

Os códigos fontes em formato Matlab® dos algoritmos implementados podem ser encontrados nos anexos.

2. Análise de Agrupamentos

Segundo (PRASS, 2004), Análise de Agrupamentos (AA) é uma técnica para reunir objetos em grupos, de tal forma que os objetos pertencentes ao mesmo grupo são mais similares entre si do que objetos que estão definidos em outro, segundo uma medida de proximidade pré-estabelecida. Desse modo, o problema que a análise de agrupamentos pretende resolver é dado uma amostra de ***n*** objetos (ou indivíduos, entidades), cada um deles medido segundo ***p*** variáveis (ou atributos), procurar um esquema de classificação que agrupe os objetos em ***k*** grupos mutuamente exclusivos baseado nas similaridades entre os objetos. Devem ser determinados também o número e as características desses grupos.

Os grupos de objetos resultantes devem exibir alta homogeneidade interna (dentro do grupo) e alta heterogeneidade externa (entre grupos), (BUSSAB *et. al*, 1990). Logo, se houver sucesso na classificação, os objetos dentro dos grupos estarão todos juntos quando representados geometricamente, e diferentes grupos estarão todos separados. A partição de objetos em grupos homogêneos é uma operação fundamental para a fase de descoberta de novas relações em mineração de dados, pois a técnica de AA ajuda a visualização de relações não identificáveis a olho nu. (CARVALHO, 2001)

(HRUSCHKA; EBECKEN, 2003 apud. PRASS, 2004) define formalmente Análise de Agrupamento como: um conjunto de ***n*** objetos $X = \{X_1, X_2, ..., X_n\}$, onde $X_i \in \Re^p$ é um vetor de dimensão *p* que pode ser agrupado em ***k*** agrupamentos disjuntos $C = \{C_1, C_2, ..., C_k\}$, respeitando as seguintes condições:

- $C1 \cup C2 \cup ... C_k = X$
- $C_i \neq \{\}, \forall\ i, 1 \leq i \leq k;$
- $C_i \cap C_j = \{\}, \forall\ i \neq j, 1 \leq i \leq k \text{ e } 1 \leq j \leq k;$

Ou seja, a união dos subgrupos forma o conjunto original, um objeto não pode pertencer a mais de um agrupamento e cada agrupamento deve possuir ao menos um objeto.

A análise de agrupamentos tem sido referida como Q-análise, tipologia, análise de classificação, análise de conglomerados e taxonomia numérica. Esta variedade de nomes deve-se em parte, ao uso de métodos de agrupamentos em diversas áreas como a

psicologia, biologia, sociologia, economia, engenharia e negócios. Embora os diferentes nomes associados a cada domínio de conhecimento, todos partilham da mesma dimensão: a classificação de dados de acordo com relacionamentos “naturais” entre eles. (CARVALHO, 2001)

Medidas de similaridade e dissimilaridade (Parecença)

Um conceito fundamental na utilização das técnicas de análise de agrupamentos é a escolha de um critério que meça a distância entre dois objetos, ou seja, um critério que quantifique o quanto eles são parecidos. Essa medida será chamada de coeficiente de semelhança. Cabe observar que tecnicamente pode-se dividir em duas categorias: medidas de similaridade e de dissimilaridade. Na primeira, quanto maior o valor observado, mais parecidos são os objetos. Já para a segunda, quanto maior o valor observado, menos parecidos (mais dissimilares) serão os objetos. A maioria dos algoritmos de análise de agrupamentos está programada para operar com o conceito de distância (dissimilaridade). (BUSSAB *et. al*, 1990)

A semelhança entre objetos pode ser medida de várias formas, mas três métodos dominam as aplicações: medidas de correlação, medidas de distância e medidas de associação. Cada método representa uma perspectiva de similaridade dependente dos objetivos e do tipo de dados (PRASS, 2004). As medidas de correlação e de distância requerem dados métricos, enquanto medidas de associação são para dados não métricos.

Medidas de correlação representam similaridade pela correspondência dos padrões por meio das suas variáveis (atributos). Uma medida de correlação de similaridade não olha a magnitude dos valores dos dados, mas sim os padrões desses valores. Devido a isso, medidas de correlação são raramente usadas na maioria das aplicações em análise de agrupamentos. (CARVALHO, 2001)

Medidas de distância representam a similaridade como a proximidade entre as observações por meio das variáveis. São as medidas de similaridades mais usadas. Muitas medidas de distância estão disponíveis. A mais comumente usada é a distância euclidiana. Vários tipos de distâncias euclidianas são usadas para calcular medidas específicas. Algumas medidas usam a distância euclidiana simples, enquanto outras usam a distância euclidiana quadrada, ou absoluta, sendo que o valor da distância é a

soma da diferença dos quadrados sem tomar a raiz quadrada. A distância euclidiana quadrada tem a vantagem de não tomar a raiz quadrada, o que aumenta sensivelmente a velocidade computacional.

Várias medidas não baseadas na distância euclidiana também estão disponíveis. Uma das alternativas mais usadas envolve a troca das diferenças quadradas pela soma das diferenças absolutas das variáveis, e é chamada de função de Hamming. A utilização desta função pode ser apropriada sob certas circunstâncias, mas ele causa vários problemas. Um deles é a suposição de que as variáveis não estão correlacionadas entre si; se elas estiverem correlacionadas, os grupos não são válidos.

Um problema enfrentado por todas as medidas de distância que usam dados não padronizados é a inconsistência entre soluções de grupos quando a escala de variáveis é mudada. Uma medida comumente usada de distância euclidiana que incorpora diretamente um procedimento de normalização é a distância de Mahalanobis.

Na abordagem de Mahalanobis, os dados são normalizados, escalando respostas em termos de desvios padrão, e ajustes são feitos para a intercorrelação das variáveis conforme mostrado na equação (1), onde X é a média e P é a matriz de covariância.

$$D_M(x) = \sqrt{(x - X)^T P^{-1} (x - X)} \qquad (1)$$

Na maioria das situações, diferentes medidas de distância podem levar a diferentes soluções de grupos. Logo, é aconselhável usar várias medidas e comparar os resultados com padrões teóricos ou conhecidos. (CARVALHO, 2001)

Medidas de associação de similaridade são usadas para comparar objetos cujas características são medias apenas em termos não métricos (medidas nominais ou ordinais). Por exemplo, entrevistados poderiam responder sim ou não para um número de perguntas. Uma medida de associação poderia avaliar o grau de concordância entre cada par de entrevistados. A forma mais simples de medida de associação poderia ser a porcentagem de vezes que houve a concordância (ambos os entrevistados disseram sim ou ambos disseram não para a questão) no conjunto de questões. Extensões deste simples coeficiente de comparação têm sido desenvolvidas para acomodar variáveis nominais multicategóricas e até medidas ordinais. (CARVALHO, 2001)

A descrição das funções de similaridade e a forma como são calculadas para os diversos tipos de variáveis pode ser encontrada em (BUSSAB *et. al*, 1990, cap. 2)

Os algoritmos básicos

Os algoritmos para formação de agrupamentos baseiam-se em duas idéias: coesão interna dos objetos e isolamento externo entre os grupos. Todos os algoritmos de agrupamento tentam maximizar as diferenças entre grupos relativas à variação dentro dos grupos, como mostrado na Figura 1. (CORMACK, 1971 *apud*. BUSSAB *et. al*, 1990).

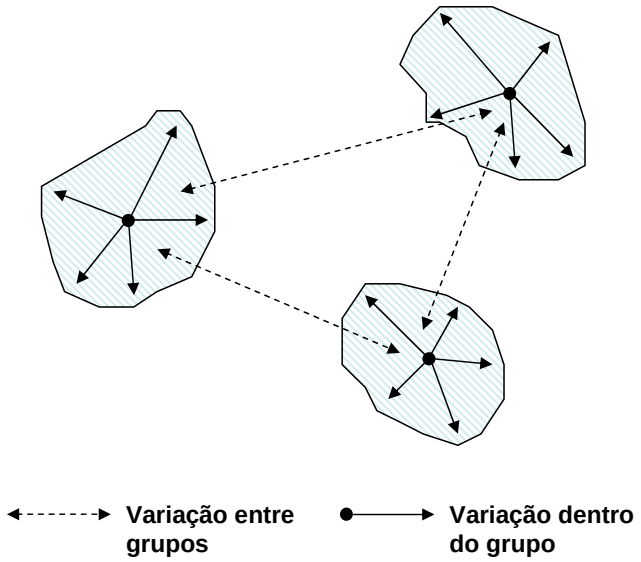


Figura 1 - Príncipio dos algoritmos básicos de análise de agrupamentos: coesão interna dos objetos e isolamento externo entre os grupos (CARVALHO, 2001) adptado.

Características desejadas de um algoritmo de agrupamento

Segundo (HAN e KAMBER, 2001), para que um algoritmo de análise de agrupamento possa ser satisfatoriamente usado em grandes massas de dados é preciso que ele atenda uma série de requisitos:

- Possua escalabilidade: capacidade de aumentar o número de dados à particionar sem que acarrete um aumento no tempo de execução na mesma proporção;
- Possua alta dimensionalidade: suporte de massa de dados que possuem milhares de registros e/ou atributos;
- Possa ser aplicado em diferentes tipos de atributos (variáveis);
- Identifique grupos com diferentes formatos;

- Tenha a habilidade de trabalhar com dados incorretos (ruídos);
- Não seja sensível a ordem com que os dados são apresentados, ou seja, deve encontrar sempre o mesmo resultado para um mesmo conjunto de dados;
- Gere resultado de fácil interpretação.

A normalização dos dados

Em qualquer aplicação, os objetivos da análise de agrupamentos não podem ser separados da seleção de variáveis usadas para caracterizar os objetos a serem agrupados. Os possíveis resultados estão diretamente ligados pela seleção das variáveis usadas.

O resultado de uma análise de agrupamentos deve ser um conjunto de grupos que podem ser consistentemente descritos por meio de suas características (CARVALHO, 2001). Conjuntamente, esses descritores são as variáveis do problema. Assim, um dos fatores que mais influencia o resultado de uma análise de agrupamentos é a escolha de variáveis.

Variáveis que assumem praticamente o mesmo valor para todos os objetos são pouco discriminatórias e sua inclusão pouco contribuiria para a determinação da estrutura do agrupamento. Por outro lado, a inclusão de variáveis com grande poder de discriminação, porém, irrelevantes ao problema, podem mascarar os grupos e levar a resultado equivocados. Além disso, é desejável que os objetos sejam comparáveis segundo o significado de cada uma delas.

(BUSSAB *et. al*, 1990) define dois tipos básicos de dados: qualitativos (ou não métricos) e quantitativos (métricos). Dados qualitativos identificam ou descrevem um objetivo. Eles descrevem diferenças nos tipos ou qualidades, indicando a presença ou ausência de uma característica ou propriedade. Muitas propriedades são discretas, tendo uma característica particular, enquanto todas as outras características são excluídas.

Ao contrário, mensurações de dados quantitativos são feitas de tal modo que os dados possam a ser identificados como diferenças em quantidade ou grau. Variáveis medidas quantitativamente refletem quantidade relativa ou distância. Medidas quantitativas são apropriadas quando é possível fazer declarações como a quantidade ou magnitude, tal como o nível de satisfação ou compromisso para um emprego.

Um aspecto importante a ser considerado é a homogeneidade entre variáveis. Ao agrupar observações, é necessário combinar todas as variáveis em um único índice de

similaridade, de forma que a contribuição de cada variável dependa tanto de sua escala de mensuração como daquelas das demais variáveis. (BUSSAB *et. al*, 1990)

Há casos em que a variação de uma unidade em uma variável expressa em toneladas é menos significativa que a variação de uma unidade medida em quilogramas em outra variável. Visando reduzir o efeito de escalas diferentes, surgiram várias propostas de relativização das variáveis. (PRASS, 2004)

Considerando as observações originais $x_1, x_2, ..., x_n$, a transformação mais comum apresentada em (BUSSAB *et. al*, 1990) é definida por:

$$Z_i = \frac{x_i - X}{s}, i = 1,...,n \quad (2)$$

Em que $\boldsymbol{X}$ e $\boldsymbol{s}$ denotam, respectivamente, a média e o desvio padrão das observações. Esta transformação, conhecida com *norma-Z*, faz com que os dados transformados tenham média zero e variância unitária. As desvantagens desta padronização é reduzir todas as variáveis ao mesmo grau de agrupabilidade.

Outra forma de transformar variáveis é mostrada na equação (3). Neste caso, tomamos os desvios em relação ao menor valor e normalizá-lo pela amplitude, ou seja, esta transformação reduz os dados para o intervalo [0,1],

$$Z_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\max}}, i = 1,...,n \quad (3)$$

Tomando a média como fator normalizador, pode-se definir ainda

$$Z_i = \frac{x_i}{X}, i = 1,...,n \quad (4)$$

A despeito da variedade de propostas, recomenda-se que a escala das variáveis seja definida por meio de transformações sugeridas pelo bom senso e pela área de conhecimento da aplicação. (CARVALHO, 2001)

Métodos de formação de agrupamentos

Os algoritmos mais comumente usados para formar agrupamentos podem ser classificados em cinco categorias (PRASS, 2004): hierárquicos, de partição, baseados em densidade, baseados em grades e baseados em modelos.

Métodos Hierárquicos

O que define os métodos hierárquicos é a reunião de dois grupos num certa etapa produz um dos grupos da etapa superior, caracterizando o processo hierárquico e permitindo a construção de um dendograma (gráfico em forma de árvore, conforme mostrado na Figura 2). As técnicas hierárquicas podem ainda ser subdivididas em dois tipos: aglomerativas e divisivas. Os métodos aglomerativos são mais populares do que os divisivos para a aplicação de análise de agrupamento (BUSSAB *et. al*, 1990).

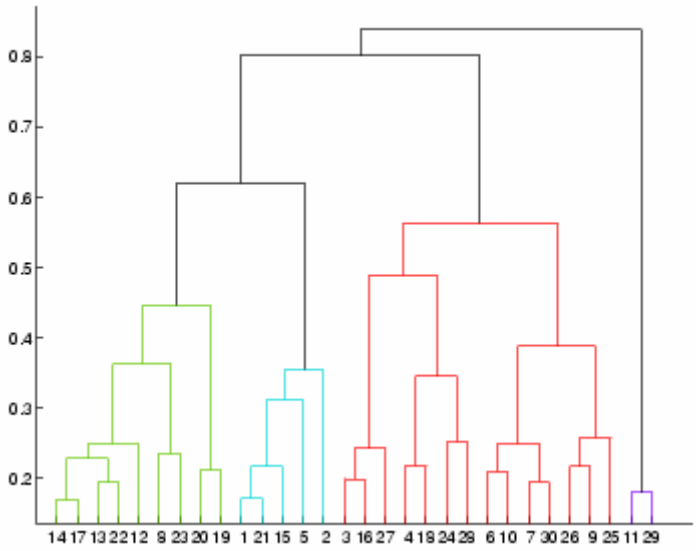


Figura 2 - Dendograma: cada linha representa um grupo criado em algum momento do processo.

Métodos hierárquicos aglomerativos

Neste método, inicialmente cada um dos ***n*** objetos é considerado como sendo um agrupamento. A cada passo, através de sucessivas fusões, vão se obtendo ***n – 1***, ***n - 2***, ..., agrupamentos, até que todos os objetos estejam em um único grupo.

No início do processo os agrupamentos são muitos pequenos e "puros", pois os membros são fortemente relacionados. Em direção ao fim do processo, os agrupamentos são grandes e pouco definidos. (BERRY e LINOFF, 1997 apud. PRASS, 2004)

O processo de execução ocorre da seguinte forma: dado um conjunto contendo ***n*** objetos, agrupando-se os dois mais similares, o que faz com que o conjunto passe a ter ***n – 1*** agrupamentos. A partir daí, calcula-se o centróide do agrupamento recém formado.

O centróide, ou semente, é o ponto cujo valor é a referência dos demais objetos do agrupamento onde ele se encontra. Normalmente, a referência utilizada é a média dos valores. Por exemplo, o centróide do conjunto formado pelos valores {21, 16, 22} é 23.

O processo de fusão dos objetos que ainda não foram agrupados é feito calculando-se a distância entre os centróides dos grupos e estes objetos, unindo-os de acordo com suas semelhanças. Este passo se repete até que todos os objetos estejam juntos, formando um único agrupamento. A Figura 3 ilustra o fluxo de execução deste método aplicado ao conjunto {0, 2, 4, 5, 8}.

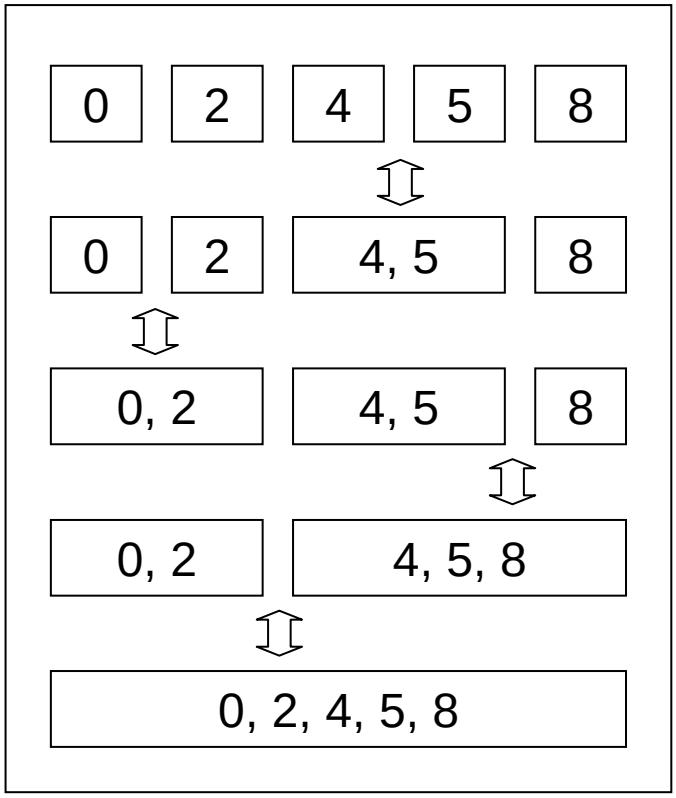


Figura 3 - Fluxo de execução para o conjunto {0, 2, 4, 5, 8} utilizando o método hierárquico aglomerativo.

Métodos hierárquicos divisivos

Quando o processo de agrupamento ocorre na direção oposta do método aglomerativo, ele é referido com um método divisivo. Nos métodos divisivos, o processo começa com um grande grupo contendo todos os objetos. Nas etapas seguintes, as observações que são mais dissimilares são divididas e grupos menores são formados. Este processo continua até que cada objeto é um grupo em si mesmo.

(MICHAUD, 1997 apud. PRASS, 2004) comenta que os métodos divisivos tendem a ser menos utilizados que os métodos aglomerativos, pois não conseguem recuperar facilmente uma partição feita por uma má escolha.

Algoritmos hierárquicos

Alguns dos algoritmos hierárquicos mais conhecidos (PRASS, 2004):

- Aglomerativos: Método da Ligação Simples (ou do Vizinho mais próximo, *Single linkage*) e Método da Ligação Completa (ou do Vizinho mais Longe, *Complete average*).
- Decisivos: DIANA (*DIvisive ANAlysis*)

Métodos baseados em partição

Nos métodos de partição procura-se diretamente uma partição dos ***n*** objetos, de modo que satisfaçam as duas premissas básicas de coesão interna e isolamento dos grupos. (BUSSAB *et. al*, 1990)

O uso dos métodos de partição pressupõe também o conhecimento do número ***k*** de partições desejadas.

De uma forma geral, o processo funciona da seguinte maneira: o primeiro passo é selecionar um registro (um semente) como o centro inicial do grupo, e todos os objetos dentro de uma pré-determinada distância são incluídos no grupo resultante. O critério mais usado é a soma de quadrados residual. Então, outra semente de grupo é escolhida e as atribuições continuam até que todos os objetos sejam atribuídos a algum grupo. (BUSSAB *et. al*, 1990)

Então, os objetos podem ser realocados se eles estiverem mais próximos de outro agrupamento do que daquele que originalmente lhes foi atribuído. Existem várias abordagens diferentes para selecionar sementes de grupos e atribuir objetos. Tipicamente usa-se uma das três abordagens seguintes para atribuir objetos individuais a um grupo (PRASS, 2004) :

- Selecionando os ***k*** primeiros objetos;
- Selecionando ***k*** objetos aleatoriamente;
- Escolhendo ***k*** objetos de modo que seus valores sejam bastante diferentes.

Algoritmo das k-médias (*k-means*)

Este algoritmo, com pequenas variações, talvez seja um dos mais usados em análise de agrupamentos. (BUSSAB *et. al*, 1990)

O algoritmo inicia com a escolha dos ***k*** objetos que formarão as sementes iniciais dos ***k*** grupos. Escolhida as sementes iniciais, é calculada a distância de cada elemento em relação às sementes, agrupando o objeto ao grupo que possuir a menor distância (mais similar) e recalculando o centróide do mesmo. O processo é repetido até que todos os objetos façam parte de um dos ***k*** grupos.

Após agrupar todos os elementos, procura-se encontrar uma partição melhor do que a gerada no passo anterior. O critério de avaliação de uma melhor partição é dado pelo grau de homogeneidade interna dos grupos. O grau de homogeneidade é calculado através da Soma de Quadrados Residual, *SQRes*. A *SQRes* é calculada pela equação (5):

$$SQ\,\mathrm{Re}\,s(j) = \sum_{i=1}^{n_j} d^2 \left(o_i(j); \bar{o}(j) \right) \tag{5}$$

onde $o_i(j), \bar{o}(j)$ e $\boldsymbol{n_j}$ são respectivamente, os valores do *i-ésimo* objeto, o centróide do grupo ***j*** e o número de objetos no mesmo.

Após o cálculo, move-se o primeiro objeto para os demais grupos e verifica-se se existe ganho na Soma de Quadrados Residual, ou seja, se ocorre uma diminuição no valor de *SQRes*. Existindo, o objeto é movido para o grupo que produz o maior ganho, a *SQRes* dos grupos é recalculada e passa-se ao objeto seguinte. Após certo número de iterações ou não havendo mais mudanças, o processo é interropido. A Figura 4 mostra o fluxo de execução do algoritmo das k-médias para formar dois agrupamentos, k = 2. Os objetos para agrupar são: {2, 6, 9, 1, 5, 4, 8}. A semente escolhida foi os dois primeiros objetos e o centróide de cada grupo é definido como a média. Na Figura 4, C1 e C2 apresentam os valores dos centróides em cada passo da execução do algoritmo.

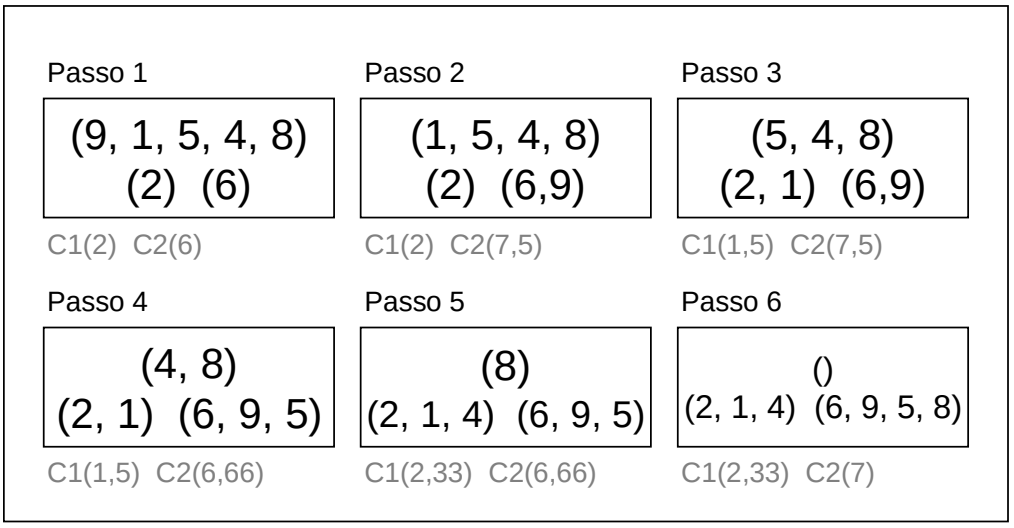


Figura 4 - Exemplo de execução do algoritmo *k-means*.

O algoritmo das k-médias e bastante escalar e confiável, porém apresenta alguns problemas (PRASS, 2004) :

- Exige que as variáveis sejam numéricas ou binárias;
- É sensível a valores discrepantes, um único objeto com valor muito extremo pode modificar substancialmente a distribuição dos dados nos grupos.

Algoritmo baseado no objeto representativo (*k-medoid*)

O funcionamento do algoritmo *k-medoid* é semelhante ao das k-médias, com exceção de que ao invés de utilizar o valor médio dos objetos do agrupamento como centróide, é utilizado o objeto mais centralmente localizado. Com isto, a sensibilidade a valores discrepantes diminui (ANDRITSOS, 2002 *apud*. PRASS, 2004).

Métodos baseados em densidade

Nos métodos baseados em densidade, um agrupamento é uma região que tem uma densidade maior de objetos do que outra região vizinha. As regiões de baixa densidade, geralmente formadas por valores discrepantes, separam um agrupamento de outro.

A idéia chave destes métodos é que, para cada objeto de um grupo, deve existir um número mínimo de outros objetos em dado raio, r, ou seja, o número de objetos que circulam um determinado grupo tem que exceder a algum limite.

O método inicia sua execução por um objeto arbitrário e, se sua vizinhança satisfaz o mínimo de densidade, inclui o objeto e os que estão em sua vizinhança no mesmo agrupamento. O processo é repetido para os novos objetos adicionados.

A principal vantagem dos métodos baseados em densidade é que eles podem descobrir grupos com formas arbitrárias. (ZAIANE *et al*., 2002 *apud*. PRASS, 2004)

Algoritmos baseados densidade

Uma leitura detalhada sobre os algoritmos baseados em densidade pode ser encontrada em (HAN e KAMBER, 2001). Os algoritmos mais conhecidos desta técnica são: DBSCAN (*Density-Based Spatial Clustering of Applications with Noise)*, OPTICS (*Ordering Points To Identify the Clustering Structures*) e DENCLUE (*DENsity-Based CLUstEring*).

Métodos baseados em grade

Os métodos baseados em grade dividem o espaço de objetos em certo número de células. Estas por sua vez são dividas em outras e assim sucessivamente, formando diferentes níveis de resolução. É através destas células que os objetos são agrupados. Maiores detalhes desta técnica e seus algoritmos, STING (Statistical *Information Grid*), WaveCluster e CLIQUE (*Clustering In QUEst*) podem ser encontrados em (HAN e KAMBER, 2001).

Métodos baseados em modelos

Os métodos baseados em modelos tentam ajustar algum modelo matemático aos dados. Os métodos são freqüentemente baseados na suposição de que os dados são gerados a partir de uma mistura de distribuições de probabilidades e seguem uma das duas principais abordagens: estatística ou por rede neural. (HAN e KAMBER, 2001)

Abordagem estatística

A abordagem estatística utiliza uma forma de agrupamento via aprendizado de máquina onde, dado um conjunto de objetos não agrupados, é construído um esquema de classificação sobre os objetos, este processo é chamado de *agrupamento conceitual*.

Ao contrário das formas de agrupamento convencionais estudadas até o momento, que antes de tudo identificavam os grupos de objetos, o agrupamento conceitual realiza uma etapa adicional para encontrar descrições das características de cada grupo que representa um conceito ou classe. (HAN e KAMBER, 2001)

Abordagem por rede neural

A abordagem de análise de agrupamento por redes neurais artificiais (RNA) possui variações com dois conceitos diferentes de RNA: redes competitivas simples e os mapas auto-organizáveis.

Na abordagem por redes competitivas simples é comum definir o numero de neurônios como o numero de agrupamentos possíveis. Após o treinamento da rede, cada neurônio estará associado a um grupo de vetores de entrada. Embora seja um método válido, já que se trata de uma análise exploratória de dados, este procedimento impõe uma restrição sobre a estrutura dos agrupamentos, pois assume-se uma estrutura hiperesferica para cada grupo de dados. Este método é aplicado para o caso de redes pequenas, pois a separação manual de padrões nessas redes é mais fácil e menos trabalhosa (SILVA, 2004).

Na abordagem por mapas auto-organizáveis é definido uma grade de neurônios como saída da rede, onde após treinamento não supervisionado surge um mapeamento do conjunto de dados de entrada para um conjunto discreto de vetores de referências, vetores esses associados a cada neurônio. O mapa auto-organizável funciona como uma rede elástica ocupando o espaço p-dimensional de forma a representar da melhor maneira, dada uma topologia de vizinhança entre os neurônios, as regiões do espaço com maior densidade de pontos. A visualização das relações entre os neurônios no espaço de saída de mapa treinado permite sugerir agrupamentos de neurônios como representantes de um determinado espaço do conjunto de entrada. Isto permite a descoberta de geometrias variadas de agrupamentos, diferentemente dos métodos estatísticos, que geralmente assumem agrupamentos nas formas hiper-esféricas ou hiper-elipsoidais (COSTA, 1999).

Os mapas auto-organizáveis têm sido usados largamente como uma ferramenta de visualização de dados apresentados em dimensões elevadas (COSTA, 1999) e é tema do próximo capítulo, onde estudaremos suas características, a preservação da topologia dos dados de entrada e formas de visualização das relações detectadas na camada de saída do mapa.

3. Mapas Auto-Organizáveis

O mapa auto-organizável, ou mapa de Kohonen, é um tipo de rede neural artificial com duas camadas (KOHONEN, 2001): a camada de entrada *I* e a de saída *U*. A entrada da rede corresponde a um vetor no espaço p-dimensional em $\Re^p$, representado por $\boldsymbol{x}_k = [\xi_1, ..., \xi_p]^{\boldsymbol{T}}$, $k = 1, ..., n,$ sendo n o número de vetores do espaço de entrada. A camada de saída é defina dispondo-se um conjunto de neurônio como nós computacionais de uma grade. Cada neurônio j desta grade, possui um vetor peso sináptico $\boldsymbol{w}$, também no espaço p-dimensional em $\Re^p$, associado ao vetor de entrada x_k, $\boldsymbol{w}_j = [w_{j1}, ...,w_{jd}]^T$. O conjunto de pesos sinápticos representa um espaço de saída discreto (HAYKIN, 2001).

Os neurônios da grade estão totalmente conectados com todos os neurônios da camada de entrada e podem estar dispostos de acordo com uma determinada topologia. A topologia da grade dita como os neurônios estão interconectados por uma relação de vizinhança. A grade pode ser unidimensional, bidimensional ou n-dimensional, sendo a grade bidimensional geralmente a mais utilizada para a maioria das aplicações (COSTA, 1999). Por exemplo, na Figura 5, tem-se um mapa com grade bidimensional e topologia retangular de dimensões 8x10.

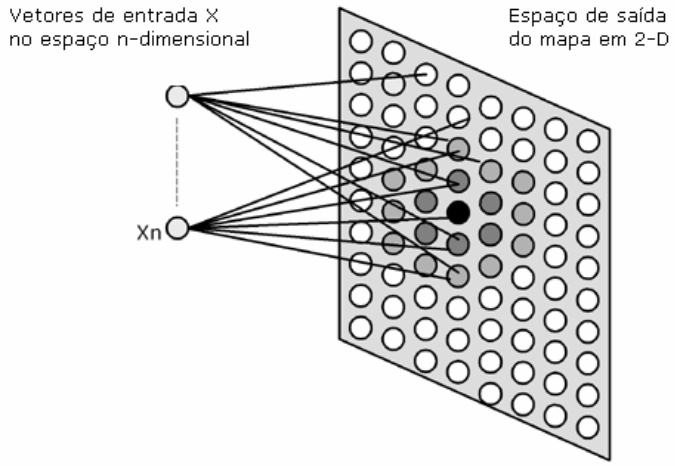


Figura 5 – Arquitetura do mapa auto-organizável com grade 2-D.

O mapa auto-organizável foi idealizado a partir da analogia com a região do córtex cerebral humano. Descobriu-se que esta parte do cérebro aloca regiões

específicas para estímulos parecidos e que para uma determinada ativação cerebral, o grau de ativação dos neurônios contribui para estimular neurônios próximos e diminui à medida que se aumenta a distância da região da ativação inicial (KOHONEN, 2001).

O processo de aprendizado do mapa auto-organizável é, segundo (HAYKIN, 2001), competitivo e não-supervisionado e pode ser dividido em três processos essenciais: processo competitivo, processo cooperativo e processo adaptativo.

O processo competitivo

É o processo que define o neurônio vencedor, isto é, aquele que apresentou o melhor casamento do padrão (vetor) de entrada com os vetores de pesos sinápticos $\boldsymbol{w_j}$ após o cálculo de uma função discriminante. Esta função discriminante fornece a base para a competição entre os neurônios. O neurônio particular com o maior valor da função discriminante é chamado o neurônio melhor casado, (BMU - *Best match unit*) ou simplesmente neurônio vencedor para o vetor de entrada $\boldsymbol{x}$ (HAYKIN, 2001). A Figura 6 mostra a grade de um mapa de dimensão 4x4 no momento do casamento de um padrão de entrada $\boldsymbol{x}$ e os pesos sinápticos $\boldsymbol{w}_j$.

A distância euclidiana é usualmente usada como função discriminante e neste caso o neurônio vencedor é aquele que apresenta a menor distância entre o vetor de entrada e os pesos $\boldsymbol{w}_j$. Haykin (HAYKIN, 2001) observa que a minimização da distância euclidiana leva a seguinte afirmação: *"um espaço contínuo de entrada de padrões de ativação é mapeado para um espaço discreto de saída de neurônios por um processo de competição entre os neurônios da grade"*.

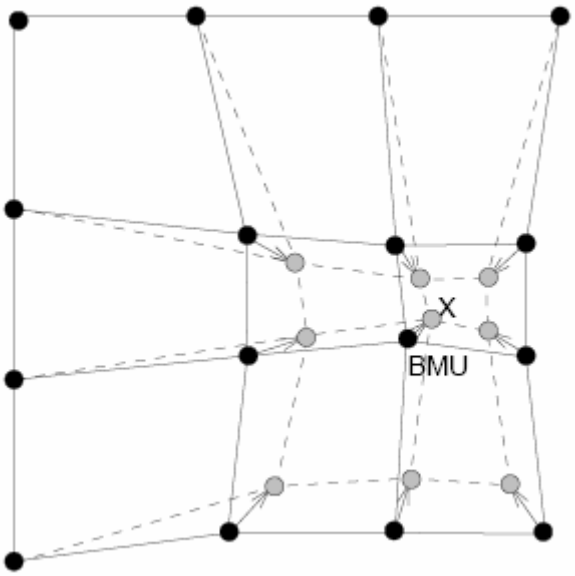


Figura 6 – Grade do mapa auto-organizável com o neurônio melhor casado (BMU) para o padrão de entrada *x*. (VESANTO, 2000)

Dependendo da aplicação de interesse, a resposta da grade pode ser tanto o índice do neurônio vencedor (i.e., sua posição na grade), como o vetor de peso sináptico que está mais próximo do vetor de entrada em um sentido euclidiano (HAYKIN, 2001).

O processo cooperativo

Neste processo são definidos quais os neurônios, além do vencedor, que terão seus pesos sinápticos ajustados. A grande dificuldade é como definir uma vizinhança topológica que seja correta do ponto de vista neurobiológico. Segundo (HAYKIN, 2001), neurobiologicamente, existe uma *interação lateral* entre os neurônios de forma que o neurônio que dispara tende a excitar mais fortemente os neurônios na sua vizinhança imediata que aqueles distantes dele. Uma função de vizinhança topológica, $h_{j,\mathrm{i}}$, deve satisfazer duas exigências distintas, considerando $d_{i,j}$ a distância lateral entre o neurônio vencedor i e o neurônio excitado j (HAYKIN, 2001):

- Deve ser simétrica em relação ao ponto máximo definido por $d_{i,j} = 0$; em outras palavras, ela alcança o seu valor máximo no neurônio vencedor i para o qual a distância $d_{i,j}$ é zero.
- Sua amplitude deve decrescer monotonamente com o aumento da distância lateral $d_{i,j}$, decaindo a zero para $d_{i,j} \to \infty$; esta é uma condição necessária para a convergência.

Duas funções de vizinhança comumente usadas são a Chapéu-mexicano e a Gaussiana. Na função do tipo Chapéu-mexicano o neurônio vencedor estimula lateralmente uma pequena vizinhança ao seu redor e à medida que a distância aumenta a estimulação torna-se inibição. Já na função do tipo Gaussiana a amplitude da vizinhança tende a zero à medida que a distância lateral aumenta. A Figura 7 mostra a configuração da curva para a função Chapéu mexicano (a) e função do tipo Gaussiana (b).

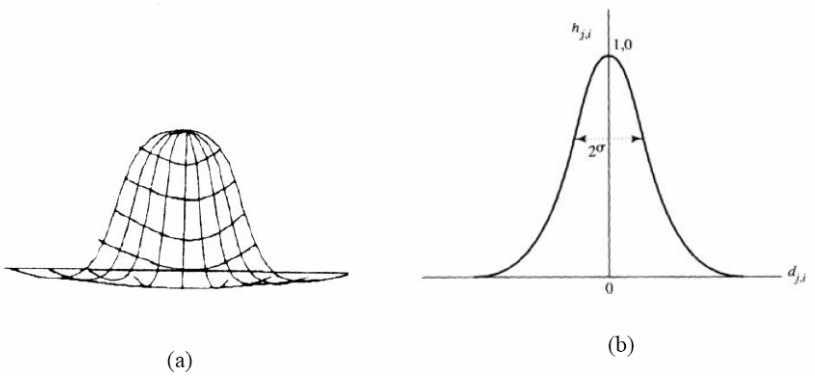


Figura 7 - Função de vizinhança Chapeu Mexicano (a) e Gaussiana (b)

A topologia da grade de um mapa pode assumir diferentes formas, sendo a topologia quadrada e a hexagonal as duas mais comuns. (VESANTO, 2000). Na Figura 8 vê-se um mapa auto-organizável com a grade de saída disposta nas duas principais topologias.

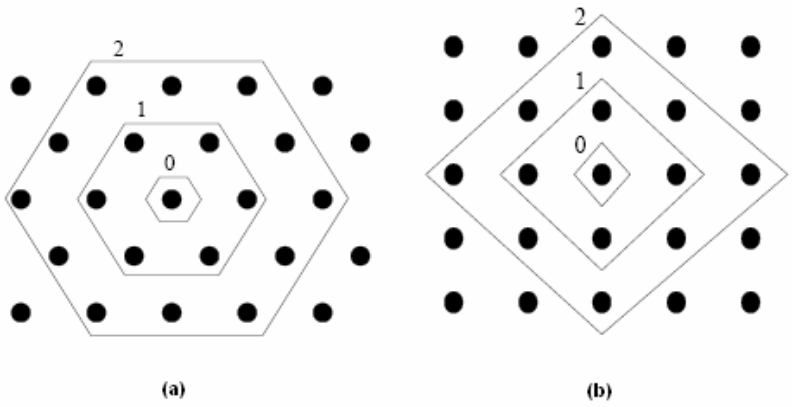


Figura 8 – Topologias: (a) grade com disposição quadrada e (b) grade com disposição hexagonal. (VESANTO, 2000)

O processo adaptativo

O processo adaptativo sináptico é último passo na formação auto-organizada de um mapa de características. Para que esse mapa seja auto-organizável é necessário que o vetor de peso sináptico $\boldsymbol{w_j}$ do neurônio j da grade se modifique em relação ao vetor de entrada $\boldsymbol{x}$. Os pesos dos neurônios (vencedor e seus vizinhos) serão atualizados a partir da equação (6) (HAYKIN, 2001).

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i}(n)(x(n) - w_j(n)) \tag{6}$$

Onde $\boldsymbol{w_j(n)}$ é o vetor de pesos no tempo n, $\boldsymbol{w_j(n+1)}$ é o vetor de pesos atualizado, $\boldsymbol{x}$ é um padrão de entrada e $\eta(n)$ é taxa de aprendizado no instante n. A taxa de aprendizado segue as mesmas regras de decaimento do raio da vizinhança, isto é, pode ser calculada por um decaimento exponencial ou ainda variar de acordo com um valor fixo pré-determinado a cada iteração. Portanto, o processo adaptativo é divido, segundo (HAYKIN, 2001), em duas fases: fase de ordenação e fase de convergência.

Na fase de ordenação os vetores de pesos, iniciados linearmente ou aleatoriamente, são ordenados topologicamente. Esta fase exige em torno de 1000 ciclos ou iterações da rede e tem como objetivo organizar os neurônios evidenciando a distribuição dos padrões do espaço de entrada. Deve-se ter cuidado na escolha dos parâmetros da taxa de aprendizagem e raio de vizinhança. Durante esta fase, a taxa de aprendizagem inicialmente é alta em torno de 1 e reduzida a um valor próximo de 0,1. Quanto à vizinhança, deve envolver inicialmente todos ou quase todos os neurônios da rede, sendo reduzida até atingir um raio por de um ou nenhum neurônio (HAYKIN, 2001).

A fase de convergência faz um ajuste fino no mapa e tem como objetivo produzir uma quantização estatística precisa do espaço de entrada. Esta fase necessita, segundo (HAYKIN, 2001), de no mínimo 500 vezes o número de neurônios na grade. A taxa de aprendizado, nessa fase, é baixa em torno de 0,01 ou menos, porém deve-se evitar que diminua a zero (HAYKIN, 2001), pois caso ocorra, é possível que a grade fique presa em um estado metaestável. Para o raio de vizinhança tem-se apenas um ou nenhum vizinho.

Propriedades do Mapa Auto-Organizável

Uma vez concluído o processo de aprendizagem do mapa auto-organizável, o mapa de códigos gerado, representado pelos vetores $\boldsymbol{w}_j$, mostrará características importantes do espaço de entrada. Segundo (HAYKIN, 2001; SILVA, 2004), algumas propriedades são:

***Propriedade 1. Aproximação do Espaço de Entrada**. O mapa auto-organizável, representado pelo conjunto de vetores de pesos sinápticos {$\mathbf{w}_j$} no espaço de saída, fornece uma boa aproximação para o espaço de entrada.*

***Propriedade 2. Ordenação topológica**. O mapa auto-organizável calculado pelo algoritmo de Kohonen é ordenado de modo topológico, no sentido de que a localização espacial de um neurônio na grade corresponde a um domínio particular ou características dos padrões de entrada.*

***Propriedade 3. Casamento de densidade**. O mapa auto-organizável reflete variações na estatística da distribuição da entrada: regiões no espaço de entrada de onde vetores de amostra **x** são retirados com uma alta probabilidade de ocorrências são mapeadas para domínios maiores do espaço de saída, e, portanto com melhor resolução que regiões do espaço de entrada das quais vetores de amostra **x** são retiradas com uma baixa probabilidade de ocorrência.*

***Propriedade 4. Seleção de características.** A partir de dados do espaço de entrada com uma distribuição não-linear; o mapa auto-organizável é capaz de selecionar um conjunto das melhores características para aproximar a distribuição subjacente.*

Desta forma, pode-se afirmar que os mapas auto-organizáveis fornecem uma aproximação discreta das assim chamadas curvas principais, e podem, portanto, ser vistos como uma generalização não-linear da análise de componentes principais (HAYKIN, 2001).

Treinamento do Mapa Auto-Organizável

A seguir é apresentado o algoritmo de Kohonen, passo a passo (HAYKIN, 2001):

- *Inicialização*. Escolha valores aleatórios para os vetores de peso iniciais $\boldsymbol{w}_j(0)$. A única restrição aqui é que os $\boldsymbol{w}_j(0)$ sejam diferentes para j = 1, 2, ...,*l*, onde *l* é o número de neurônios na grade. Pode ser desejável manter a magnitude dos pesos pequena. Outro modo de inicializar o algoritmo é selecionar os vetores de peso $\{w_j(0)\}_{j=1}^{l}$ a partir do conjunto disponível de vetores $\{x_i\}_{i=1}^{N}$ de uma maneira aleatória;
- *Amostragem*. Retire uma amostra $\boldsymbol{x}$ do espaço de entrada com uma certa probabilidade; o vetor $\boldsymbol{x}$ representa o padrão de ativação que é aplicado à grade. A dimensão do vetor $\boldsymbol{x}$ é igual a *m*.
- *Casamento por similaridade*. Encontre o neurônio com o melhor casamento (vencedor) i(**x**) no passo de tempo *n* usando o critério da mínima distância euclidiana:

$$\mathrm{i}(\boldsymbol{x}) = \arg\min_{\mathrm{j}} \| \boldsymbol{x}(\mathrm{n}) - \boldsymbol{w}_j \|, \ \mathrm{j} = 1, 2, ..., l$$

- *Atualização*. Ajuste os vetores de peso sináptico de todos os neurônios usando a fórmula de atualização:

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i}(n)(x(n) - w_j(n))$$

onde $\eta(n)$ é o parâmetro da taxa de aprendizagem e $h_{\mathrm{j,i(x)}}(n)$ é a função de vizinhança centrada em torno do neurônio vencedor i($\boldsymbol{x}$); ambos $\eta(n)$ e $h_{\mathrm{j,i(x)}}(n)$ são variados dinamicamente durante a aprendizagem para obter melhores resultados.

- *Continuação*. Continue com o passo 2 até que não sejam observadas modificações significativas no mapa auto-organizável.

Exemplo de treinamento do Mapa Auto-Organizável

Esta seção apresenta um exemplo simples do uso do mapa auto-organizável para o caso de entrada bidimensional. Um conjunto de dados de 375 exemplares foi gerado por uma mistura de três gaussianas a partir dos vetores de média $\mu_1 = (0,0)$, $\mu_2 = (5,5)$ e $\mu_3 = (9,0)$. Este conjunto é apresentado na Figura 9 - Conjunto artificial para testes.Figura 9 sendo divido em três classes.

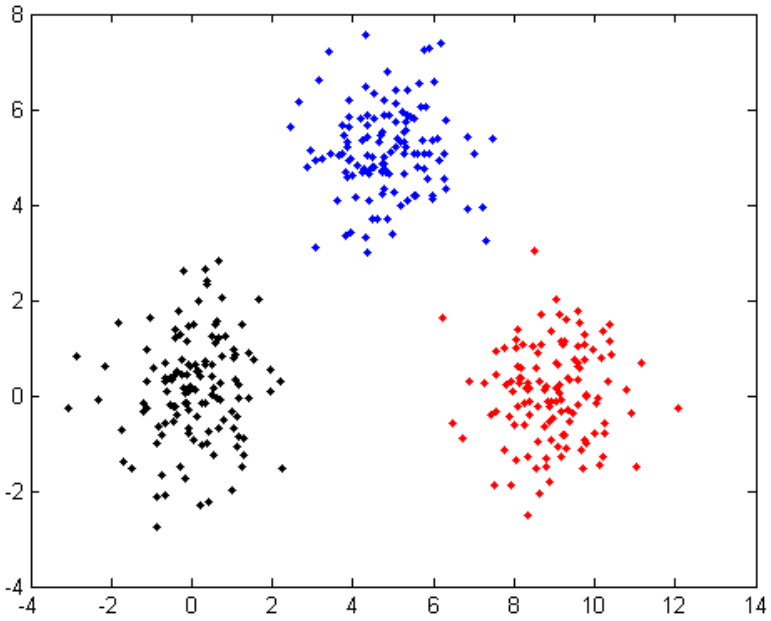


Figura 9 - Conjunto artificial para testes.

O mapa utilizado para treinamento possui uma grade bidimensional 10x10 com topologia retangular. A inicialização dos pesos foi linear. A função de vizinhança usada foi gaussiana e o raio inicial foi 8 caindo para 1 no final da primeira fase do treinamento (fase de ordenação). A Figura 10 ilustra a configuração dos neurônios no espaço 2-D após o final do treinamento. Observe que podemos identificar 3 regiões de grande densidade de neurônios.

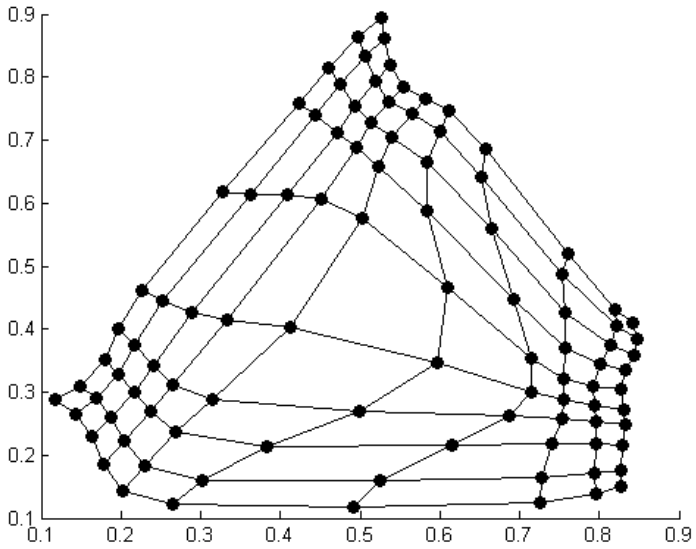


Figura 10 - Grade do mapa 10x10 após 300 épocas de treinamento.

Visualização do espaço de saída do Mapa Auto-Organizável

Após o processo de treinamento do mapa auto-organizável é interessante que possamos verificar visualmente o resultado da ordenação topológica. No entanto, devido a dimensionalide do conjunto de dados nem sempre é possível obter uma visualização de identificação rápida dos agrupamentos. A seguir são descritos quatro modos fundamentalmente diferentes de se visualizar o espaço de saída de um mapa auto-organizável.

Representação dos pesos sinápticos no espaço $\Re^d$

Quando os vetores de código possuem dimensão menor ou igual a 3, podemos projetar seus valores como coordenadas no espaço $\Re^d$ para visualização da organização dos neurônios. Neste método de visualização a grade de saída do mapa auto-organizável é vista como uma rede elástica com os vetores de peso sináptico tratados como ponteiros para os respectivos neurônios que estão direcionados para o espaço de entrada. Este método de visualização é particularmente útil para mostrar a propriedade de ordenação topológica do algoritmo de treinamento. (HAYKIN, 2001)

A Figura 10 é um exemplo deste modo de visualização aplicado ao mapa treinado da seção 0.

Histograma dos neurônios vencedores

Após o processo de treinamento pode ser determinado o nível de atividade dos neurônios da grade de saída, *H(i)*. O nível de atividade de um neurônio representa a quantidade de dados do espaço de entrada que são mapeados para este neurônio. Desta forma, a projeção da atividade associada a cada neurônio gera um histograma que refletirá os neurônios mais ativos. A Figura 11 mostra a aplicação deste método para o mapa treinado da seção 0. A Figura 12 - Histograma da atividade dos neurônios do mapa 10x10 com neurônios inativos, *H(i) = 0*, apagados da grade.Figura 12 mostra o histograma para neurônios com atividade maior que zero.

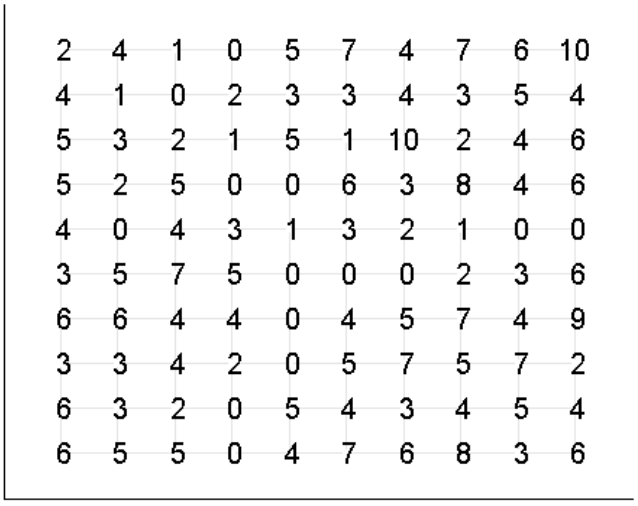


Figura 11 - Histograma da atividade dos neurônios do mapa 10x10.

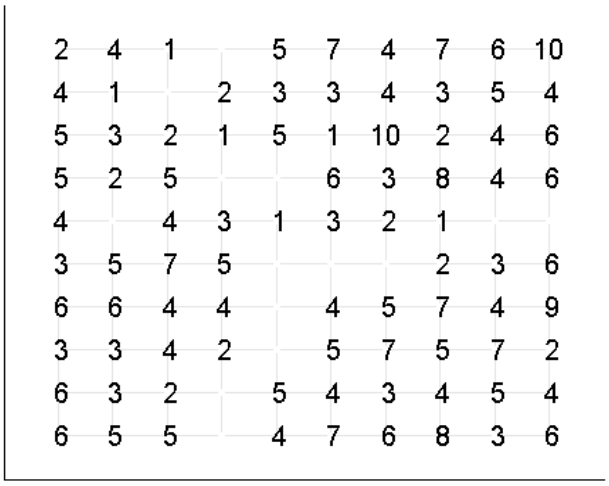


Figura 12 - Histograma da atividade dos neurônios do mapa 10x10 com neurônios inativos, *H(i) = 0*, apagados da grade.

Mapas contextuais

Assumindo que além da informação dos padrões também disponhamos de suas classes podemos atribuir rótulos a neurônios vencedores na grade de saída da rede as classes cujos dados eles representam. Dados do espaço de entrada são apresentados ao mapa e o neurônio vencedor será o mais similar, ou o mais próximo, de acordo com o critério de similaridade escolhido. A este neurônio atribui-se um rótulo (a classe do

dado de entrada). Como resultado, os neurônios na grade de saída do mapa são particionados em um numero de regiões coerentes (RITTER e KOHONEN, 1989 *apud*. HAKIN, 2001).

Para o exemplo do mapa treinado da seção 0, assumimos que os dados de entrada foram rotulados de acordo com cada classe pertencente, originando os rótulos classe1, classe2 e classe3. A Figura 13 mostra o resultado da aplicação do mapa contextual. Para cada neurônio melhor casado com um dado de entrada, associamos o rótulo de classe do dado ao neurônio representante. Regiões vazias no mapa indicam não atividade do neurônio.

Figura 13 - Mapa contextual para o mapa treinado da seção 0.

Uma iniciativa interessante desta abordagem é o projeto WEBSOM. Este projeto emprega mapas auto-organizáveis para a organização automática de vários tipos de documentos, incluindo textos na *web*. Uma imagem de saída da rede é gerada usando a técnica contextual, permitindo uma fácil visualização e exploração dos conceitos retidos dos textos submetidos ao mapa. (LAGUS, 1998)

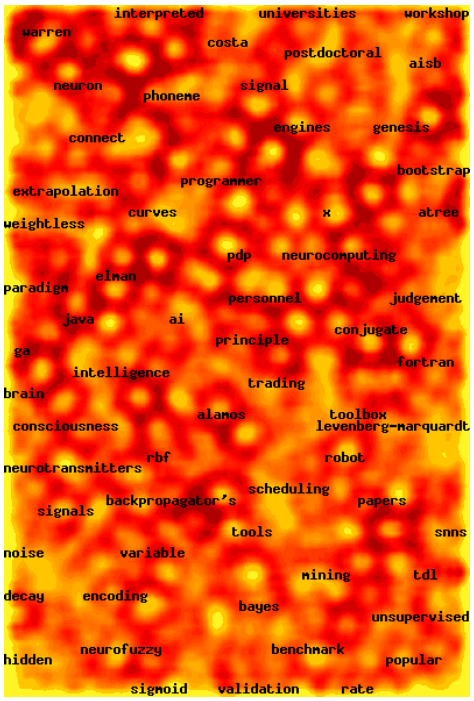


Figura 14 - Mapa conceitual obtido dos textos da lista de discussão comp.ai.neural-nets (WEBSOM, 1999)

Matriz de distância unificada U-Matriz

O método denominado matriz de distâncias unificadas, ou *U-matriz*, foi desenvolvido por Alfred Ultsch com o objetivo de permitir a detecção visual das relações topológicas dos neurônios. Usa-se a mesma forma de cálculo utilizada durante o treinamento para determinar a distância entre os vetores de peso 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 da grade de saída do mapa, e a intensidade de cada pixel na imagem *f(x,y)* corresponde a uma distância calculada. Um mapa bidimensional *N x M* gera uma imagem *f(x, y)* de *(2N – 1) x (2M – 1)* pixels(COSTA, 1999).

A imagem gerada pode ser vista como uma função tridimensional em que o valor do pixel na coordenada (x, y) é representado por um ponto no eixo z. Neste caso, teremos uma superfície em 3-D cuja topografia revela a configuração dos neurônios obtida pelo treinamento. A Figura 15 apresenta a visualização da imagem *f(x, y)* para o mapa treinado da seção 0 como uma imagem de relevo topográfico. A Figura 16 apresenta *f(x, y)* como uma grade 2-D.

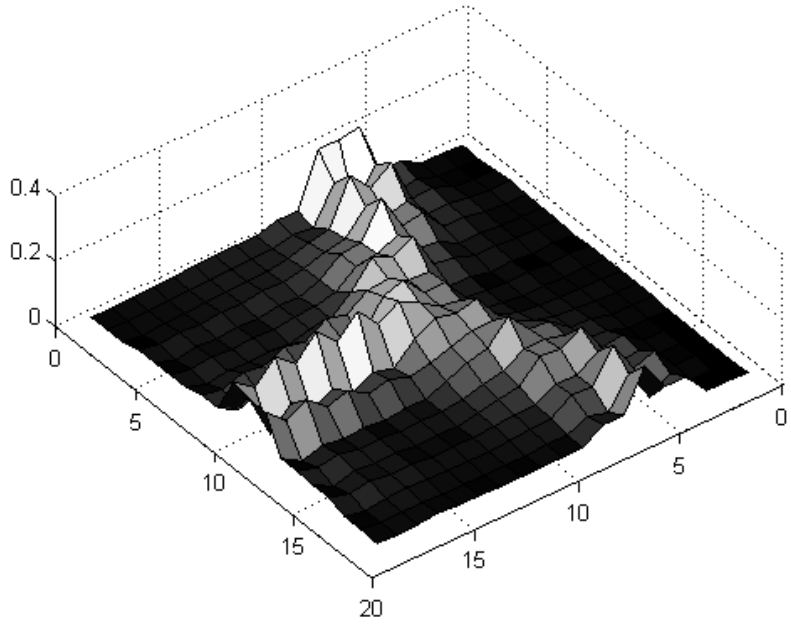


Figura 15 - Visualização da U-matriz como um relevo topográfico para o mapa 10x10 da seção 0.

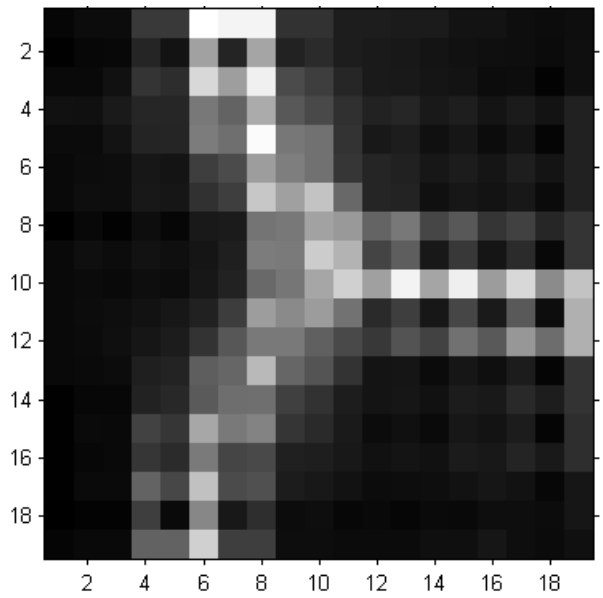


Figura 16 - Representação 2-D da U-matriz.

Vales, no relevo topográfico, correspondem a regiões de neurônios que são similares, enquanto que montanhas, distâncias grandes, refletem a dissimilaridade entre neurônios vizinhos e podem ser associadas a regiões de fronteiras entre neurônios. (COSTA, 1999)

Este método de visualização do espaço de saída do mapa é extremamente útil quando os vetores de entrada possuem dimensão maior que 3, pois para estes casos não se pode obter uma representação gráfica da disposição final dos neurônios.

Cálculo da U-matriz

Considere um mapa com grade retangular de tamanho *N x M*. Seja [$\mathbf{b_{x,y}}$] a matriz de neurônios e $\boldsymbol{w_{jx,y}}$ a matriz de pesos sinápticos. Para cada neurônio **b** existem três distâncias d_x, d_y, d_{xy}, na U-matriz, a seus vizinhos. A Figura 17 mostra essas distâncias para um neurônio $\mathbf{b_{x,y}}$.

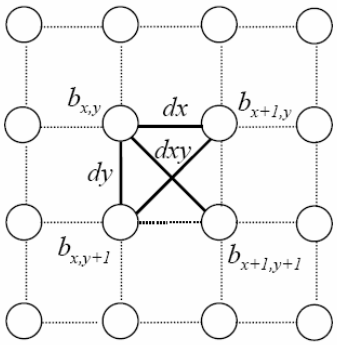


Figura 17 - Distâncias d_x, d_y, d_{xy} para o neurônio $b_{x,y}$.

No caso da topologia da grade de neurônio ser retangular e considerando a distância euclidiana, os valores para d_x, d_y, d_{xy} podem ser definidos como:

$$dx(x,y) = \sqrt{\sum_i \left(w_{i_{x,y}} - w_{i_{x+1,y}}\right)^2} \tag{7}$$

$$dy(x,y) = \sqrt{\sum_i \left(w_{i_{x,y}} - w_{i_{x,y+1}}\right)^2} \tag{8}$$

$$dxy(x,y) = \frac{1}{2\sqrt{2}}\left[\sqrt{\sum_i \left(w_{i_{x,y}} - w_{i_{x+1,y+1}}\right)^2} + \sqrt{\sum_i \left(w_{i_{x,y+1}} - w_{i_{x+1,y}}\right)^2}\right] \tag{9}$$

Estas distâncias, calculadas no espaço dos pesos, são plotadas em uma matriz U de tamanho *(N-1) x (M-1)*. A U-matriz combina as três distâncias considerando a

posição de todos os neurônios do mapa. Para cada neurônio de b, as distâncias para os vizinhos (se estas existirem) tornam-se d_x, d_y, d_x, e a U-matriz é preenchida de acordo com a tabela abaixo.

Tabela 1 - Esquema para preenchimento dos elementos da U-matriz (COSTA, 1999)

i	*j*	*(i,j)*	$U_{i,j}$
I	*P*	*(2x + 1, 2y)*	*dx(x,y)*
P	*I*	*(2x, 2y + 1)*	*dy(x,y)*
I	*I*	*(2x + 1, 2y + 1)*	*dxy(x,y)*
P	*P*	*(2x, 2y)*	*du(x,y)*

As abreviações I e P referem-se ao índice ou posição do neurônio, sendo ímpar e par, respectivamente. O cálculo de *du(x,y)* pode ser a média, mediana, valor máximo ou mínimo dos elementos circunvizinhos. Seja C = (c_1, c_2,..., c_k) os valores dos elementos circunvizinhos de $U_{2x,2y}$ aparecendo na forma de um vetor ordenado com cardinalidade k. No caso de uma topologia retangular, k = 4. No caso de *du(x,y)* ser o valor mediano, temos

$$du(x,y) = \begin{cases} c[(k+1)/2], & se\ k\ for\ ímpar \\ \dfrac{c(k/2) + c[(k+1)/2)}{2}, & se\ k\ for\ par \end{cases} \qquad (10)$$

Pelo fato de geralmente a U-matriz gerar uma imagem relativamente complexa, principalmente em problemas de dados reais, geralmente seu uso é restrito a visualização, sendo uma técnica de auxílio na separação manual dos agrupamentos de um mapa, ou seja, o usuário realiza a separação dos grupos baseados nos critérios que lhe pareçam mais adequados. (COSTA, 1999)

No próximo capítulo explica-se como determinar de forma automática os agrupamentos do conjunto de dados utilizando a imagem gerada pelo cálculo da U-matriz.

4. Descoberta automática de agrupamentos pela segmentação do espaço de saída do Mapa Auto-Organizável

Como descrito anteriormente, a proposta deste trabalho é investigar a detecção automática de agrupamentos baseado em técnicas de segmentação do espaço de saída de um mapa auto-organizável. Vimos no capítulo anterior quatros formas que nos permitem analisar visualmente o resultado do treinamento do mapa.

As próximas seções apresentam duas técnicas que particionam a grade do mapa em grupos de neurônios. Essas duas técnicas se baseiam somente nas relações topológicas dos neurônios e sua vizinhança. A partição da grade em conjuntos de neurônios também determina uma partição dos dados no conjunto de dados de entradas, bastando para isso saber para cada dado de entrada seu neurônio de referência na grade.

Segmentação da U-matriz

Como vimos no capítulo anterior, a matriz de distância unificada, U-matriz, pode ser interpretada como uma imagem através da coloração dos *pixels* de acordo com a intensidade de cada componente da matriz. O nível de intensidade de cada *pixel* corresponde a uma distância calculada entre um neurônio e sua vizinhança. Valores altos correspondem a neurônios vizinhos dissimilares e valores baixos correspondem a neurônios vizinhos similares (COSTA, 1999). Podemos interpretar esta imagem como uma superfície em 3-D cuja topografia revela a configuração dos neurônios obtida pelo treinamento como mostrado na Figura 15. Regiões com baixos valores do gradiente correspondem a vales que agrupam neurônios especializados em padrões similares e são candidatas para representar agrupamentos de neurônios. Regiões com valores altos correspondem a montanhas e podem ser associadas às fronteiras entre agrupamentos de neurônios.

O algoritmo SL-SOM (Self *Labeled SOM*) apresentado em (COSTA, 1999) utiliza a abordagem vale/montanha para segmentar a U-matriz determinando automaticamente os possíveis agrupamentos presentes em conjunto de dados. O método SL-SOM efetua, automaticamente, segmentação e rotulação do mapa auto-organizável

apenas usando os padrões (COSTA, 1999). O funcionamento do algoritmo é tema das próximas seções.

O Algoritmo SL-SOM

Imagens e segmentação watershed

A representação convencional de uma imagem é considerá-la uma função *f(x,y)*, onde amplitude *f* nas coordenadas *x* e *y* é proporcional à intensidade ou brilho, denominado nível de cinza no caso de imagens monocromáticas. O número de níveis diferentes de intensidade depende da discretização obtida na imagem (FILHO e NETO, 1999). No caso deste trabalho, imagens possuem 256 escalas de cinzas de cinzas, também denotada da forma $[f_{min}, f_{max}] = [0,255]$.

Cada ponto da imagem, ou pixel, $f(x_i, y_i)$, possui um conjunto de pixels vizinhos. A conectividade entre pixels é um conceito extremamente importante principalmente quando se trata de segmentação de regiões em uma imagem (FILHO e NETO, 1999). Os casos mais comuns consideram que os pixels estejam conectados à quatro vizinhos, padrão de conectividade 4, ou à oito, padrão de conectividade 8. A Figura 18 mostra a configuração da vizinhança de um *pixel p* para esses dois padrões.

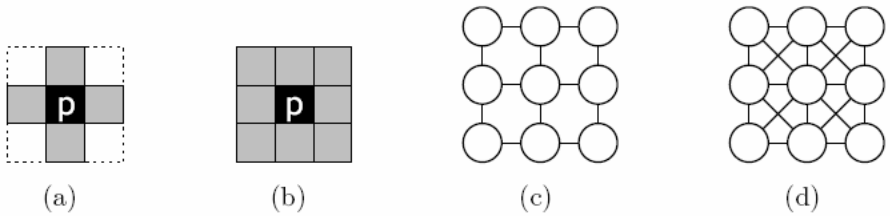


Figura 18 - Configuração da vizinhança 4-conectividade (a) e (c), e 8-conectividade (b) e (d).

A segmentação de uma imagem consiste em subdividir a imagem em suas partes ou objetos constituintes. Atualmente existem diversas técnicas, geralmente aplicadas, cada uma, a um subconjunto de aplicações, não havendo um método universal para o processo de segmentação, principalmente por haver diversos tipos de imagens de sensores, e suas várias formas de representação (FILHO e NETO, 1999).

Em geral, a segmentação de imagens pode ser categorizada em duas classes de técnicas: técnicas baseadas em extração de contornos e técnicas baseadas em crescimento de regiões. No primeiro caso, são usadas as alterações bruscas nos níveis de

cinza da imagem para tentar traçar contornos entre regiões. Por outro lado, técnicas de segmentação baseadas em crescimento de regiões usam um critério de similaridade para agrupar pixels ou regiões a partir de marcadores.

O método mais simples de segmentar um imagem *f(x,y)* é através da limiarização, onde um valor escolhido entre [f_{min}, f_{max}] é usado para binarizar a imagem (FILHO e NETO, 1999). A Figura ilustra o processo de segmentação por limiarização com fator de limiar, $k = 85$.

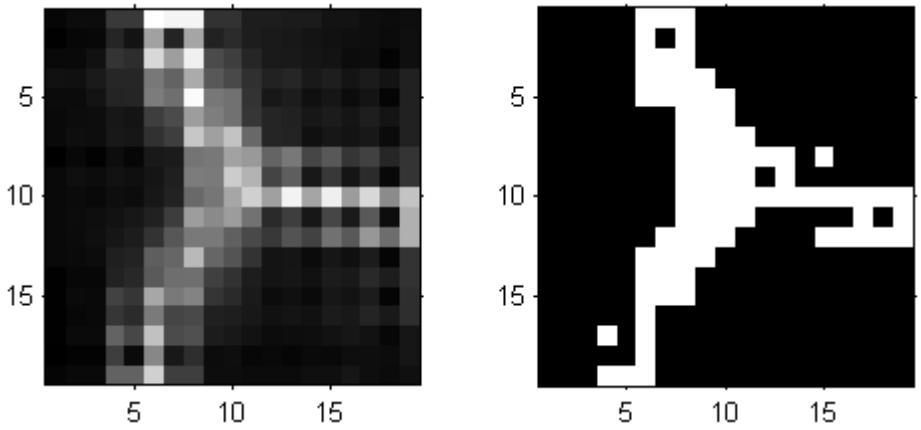


Figura 19 - Segmentação de imagem por limiarização com fator de limiar k=85.

Em relação a U-matriz, o uso de técnicas simples como a limiarização em geral conduz à resultados insatisfatórios, pois geralmente este tipo de imagem possui um histograma complexo e ruidoso, não havendo, em geral, um método simples de encontrar um limiar adequado (COSTA, 1999).

Uma forma considerada mais eficiente para segmentação de imagens é o algoritmo (ou transformada) *watershed*. Ele pode ser considerado um algoritmo híbrido, combinando tanto a abordagem por crescimentos de regiões quanto na extração de contornos. Uma maneira simples de idealizar o funcionamento do watershed é associar a imagem *f* a um relevo topográfico, por exemplo, a Figura 15, onde considera-se o nível de cinza como altitude. Definimos $B_f(m)$, uma bacia de retenção associada a um mínimo *m* da superfície topográfica de *f*, como a região na qual, caso uma gota de água caísse em qualquer ponto desta bacia, iria percorrer um caminho até atingir este ponto de mínimo. A segmentação por *watershed* vai constituir na determinação das bacias de retenção a partir do tipo das primitivas da região de contorno, a partir dos pontos de mínimos. Imaginando gotas de água caindo em todas as coordenadas da imagem, as *k* bacias de retenção captam a água, partindo de cada mínimo m_k, e à medida que o nível

das bacias aumenta é possível que ocorra transbordamento da água de uma bacia para outra. Porém, isto é evitado com a construção de diques, separando as bacias retentoras e impedindo que águas de diferentes bacias sejam compartilhadas. Quando a inundação atinge o nível máximo da altura da superfície, no caso f_{max}, os diques construídos separando as bacias retentoras são as linhas da *watershed* da imagem, geralmente com espessura de 1 *pixel*, formando os contornos das regiões segmentadas da imagem(COSTA, 1999; KLAVA, 2006).

Na Figura 20 podemos visualizar o princípio da formação de bacias e construção dos diques. As bacias surgem de mínimos locais e são separadas pela construção de diques. O encontro dos diques, suas bordas, determina as linhas da *watershed*. Na Figura 21 é mostrado o resultado do algoritmo de *watershed*, Figura 21(b), aplicado a U-matriz, Figura 21(a), gerada à partir do mapa da seção 0.

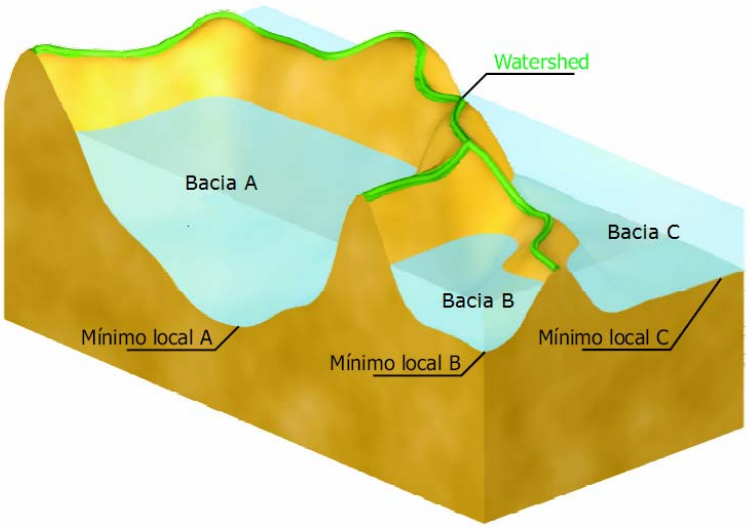


Figura 20 - Idéia básica do funcionamento do algoritmo *watershed* (KLAVA, 2006) adaptado.

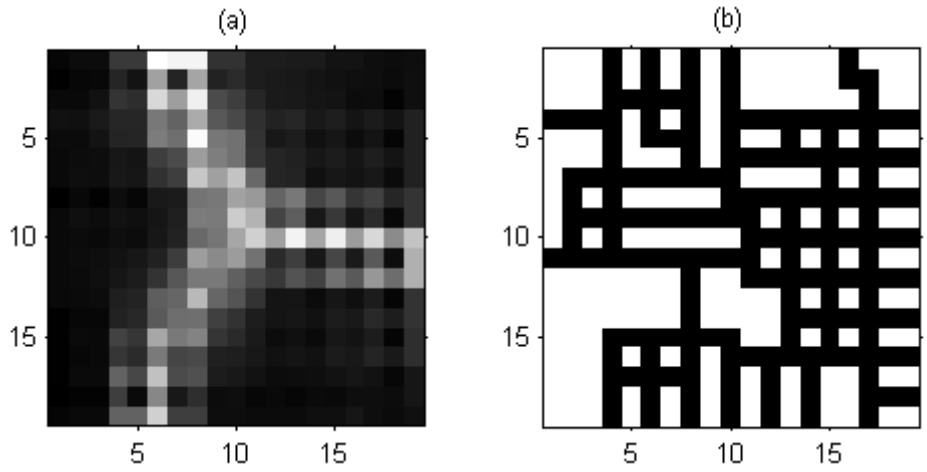


Figura 21 - Aplicação do *watershed* na U-matriz gerada a partir do mapa treinado da seção 0

Entretanto apenas nos casos mais simples o algoritmo de *watershed* pode ser aplicado diretamente à imagem da U-matriz (KLAVA, 2006). Este impedimento vem do fato que esta imagem possui muitas zonas de depressões regionais, imagem rugosa, o que pode acarretar em dois tipos de problemas: a sobre-segmentação ou a sub-segmentação. A primeira leva a imagem final a um grande número de partições, ou seja, um número de regiões acima do desejado, caso da Figura 21(b). Por outro lado, a sub-segmentação pode levar a fusão de algumas "linhas de partições", ocasionando a perda da bacia, o que implica em uma fusão de duas ou mais regiões.

(COSTA, 1999) sugere a regularização da imagem da U-matriz com o uso de marcadores específicos, escolhidos de forma a indicar quais as bacias retentoras que são importantes e que devem ser levadas em consideração na execução do algoritmo de *watershed*.

Escolha de marcadores

A importância de determinar marcadores para a imagem que será processada pela *watershed* é a eliminação das "bacias" indesejáveis muitas vezes formadas por ruídos ou excesso de mínimos locais. Apenas as águas de bacias retentoras associadas aos mínimos descritos pelos marcadores terão rotulo, o que implica que o número de regiões finais da imagem segmentada será igual ao número de marcadores escolhidos. (COSTA, 1999)

A escolha de marcadores para a U-matriz proposta em (COSTA, 1999) é: Seja a U-matriz de um mapa treinado dada pela imagem f, de tamanho *(2N-1)x(2M-1)*, onde

NxM é o tamanho do mapa. Considere que [f_{min}, f_{max}] = [0, 255], ou seja, há 256 níveis de cinza na imagem f. Os seguintes passos são efetuados:

1. Filtragem: a imagem f_1 é gerada removendo-se pequenos buracos na imagem *f*. Pequenas depressões com área inferior a τ pixels são eliminados.
2. Para $k = 1, ..., f_{max}$, onde f_{max} é o nível de cinza máximo na imagem f_1, crie as imagens binárias f_2^k correspondendo a conversão de f_1 usando *k* como valor de limiar.
3. Calcule o número de regiões conectadas de f_2^k, para cada valor de *k*, N_{rc}^k.
4. Procure no gráfico $k \times N_{rc}^k$ a maior seqüência contígua e constante de número de regiões conectadas N_{rc}^k, denotado por S_{max}.
5. A imagem de marcadores será a imagem f_2^j, onde *j* é o valor inicial da seqüência S_{max}.

O passo 1 do algoritmo serve para suavizar a imagem original da U-matriz, resultando em uma imagem melhor para processamento.

O passo 2 consiste em a aplicar o processo de limiarização em lote utilizando como fator de limiar a gama de valores disponíveis entre fmin e fmax da imagem. Como no nosso caso as imagens são em níveis de cinza, k (fator de limiar) assumira todos os valores de 0 a 255. Para cada imagem binária derivada de uma operação de limiarização, aplica-se um processo de contagem dos componentes conectados, que é o número de objetos na imagem para cada valor de limiar *k* aplicado, N_{rc}^k.

A Figura 22 esquematiza o processo de aplicação do algoritmo de watershed com a escolha de marcadores. Inicialmente a U-matriz é segmentada por limiarização em todos os seus níveis de tons de cinza, *k* variando de 0 a 255. O gráfico $k \times N_{rc}^k$ é analisado e detecta-se a maior zona de estabilidade de *k* = 18 até 125. O primeiro valor de da seqüência contígua é tomado, *k* = 18, e gera-se a imagem de marcadores. Em seguida, aplicamos o algoritmo de *watersheld* que além da imagem à segmentar, recebe como parâmetro a imagem de marcadores. O resultado final pode ser visto na Figura

22 e como podemos observar, o algoritmo revelou três regiões separadas pelas linhas de *watershed*, o que corresponde ao numero de classes esperado.

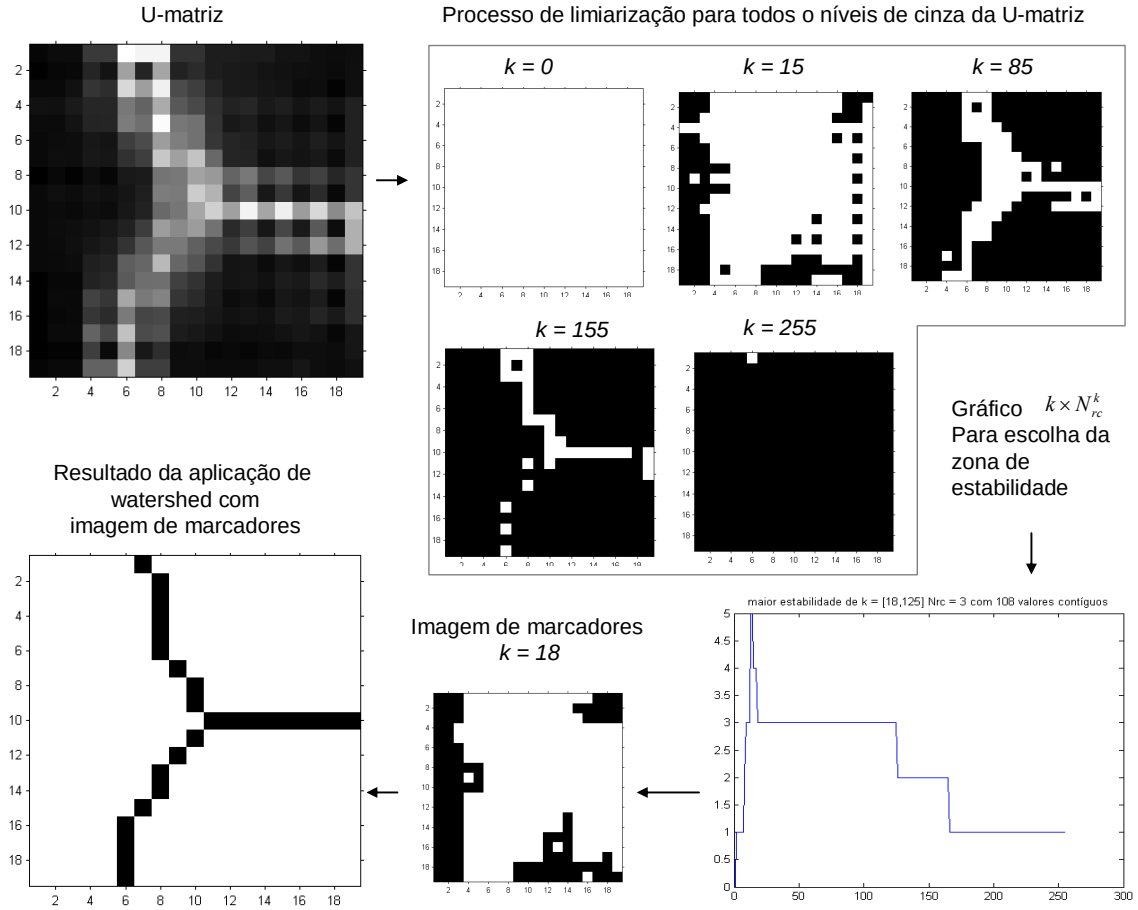


Figura 22 - Aplicação do algoritmo de watershed com escolha de marcadores.

Rotulagem de regiões conectadas

A imagem resultante da aplicação do algoritmo de *watershed*, ver Figura 22, deve ser rotulada de forma que possamos tratar as várias regiões descobertas como objetos de forma independente. Considere que a imagem segmentada possua os valores zeros para as linhas de watershed e valores maiores que zero para cada bacia retentora. Regiões diferentes possuem códigos diferentes. A determinação dos componentes conectados e disjuntos consiste em agrupar todos os *pixels* de uma mesma bacia retentora sob o mesmo rótulo. Os *pixels* que formam a linha da watershed não representam nenhuma região, mas representam neurônios de separação de classes na U-matriz. Devemos então, rotulá-los de acordo com um critério de distância dos *k*-vizinho mais próximos, por exemplo, para que possamos realizar uma total conversão das coordenadas de cada *pixel* em de neurônio da grade do mapa treinado. A Figura 23

apresenta a U-matriz segmentada da Figura 22 após o processo de rotulação. Os *pixels* pertencentes às linhas de watershed rotulados segunda o critério de distância 4-vizinhos.

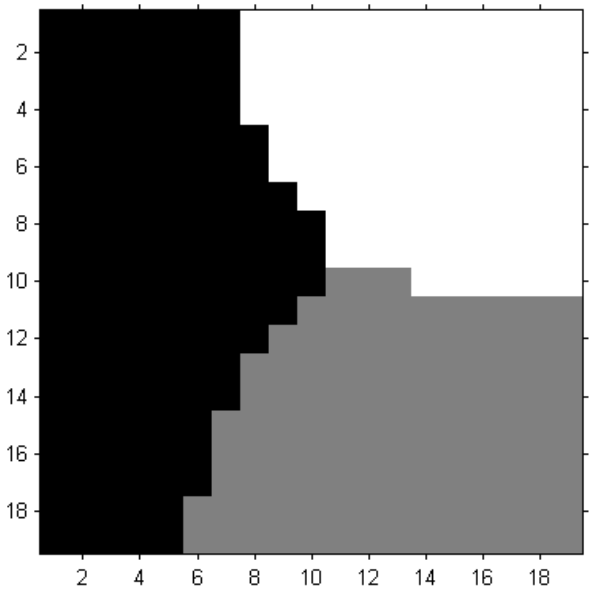


Figura 23 - Regiões da watershed após rotulação.

Uma rotina auxiliar que converta as coordenadas dos *pixels* da U-matriz de volta em neurônios determina os agrupamentos de neurônios, que por sua vez, determinam também os grupos formados no conjunto de dados de entrada.

Resumo do algoritmo SL-SOM

Toda análise e segmentação de um mapa de tamanho *NxM* treinado é feita inicialmente sobre a U-matriz, e posteriormente as informações associadas aos pixels são associadas os neurônios (COSTA, 1999).

Os passos do algoritmo SL-SOM são descritos a seguir:

1. Obtenção da U-matriz
2. Encontrar os marcadores para a U-matriz
3. Aplicar o *watershed* sobre a U-matriz usando os marcadores obtidos no passo 2.
4. Rotulagem das regiões conectadas da imagem segmentada no passo 3 e formação dos grupos de neurônios.

Segmentação por particionamento de grafos

Outra forma de segmentar a saída de um mapa auto-organizável treinado é a técnica baseada no particionamento de grafo proposta em (COSTA e NETTO, 2003). Esta técnica é independente da U-matriz e da dimensão da grade do mapa.

Grade do mapa auto-organizável com um grafo

Na técnica por particionamento de grafos, a grade do mapa é vista com um grafo não orientado composto por todos os neurônios do mapa com conexões definidas pela topologia da grade.

Um grafo *G(V, A)* é um conjunto finito não vazio *V* e um conjunto E de pares não ordenados de elementos distintos de *V*. Os elementos de *V* são os vértices e os de *A* são as arestas de *G*, respectivamente. As arestas podem ser valoradas. Esses valores determinam um custo ou peso associado à aresta. Duas arestas que possuem um vértice em comum são chamadas de adjacentes (RABUSKE, 1992).

A matriz de adjacências é uma matriz quadrada de natureza *booleana*, *A*, de ordem *n*, onde *n* é o número de vértices do grafo *G*, onde cada elemento *A(i,j)* indica a existência de uma relação entre os vértices *i* e *j* (RABUSKE, 1992).

A técnica de particionamento de grafos opera sobre um grafo *G'(V', A')*, onde os conjunto de vértices, *V'*, é formado por todos os neurônios da grade e o conjunto de arestas, *A'*, é determinado pela topologia da grade, retangular ou hexagonal, e os valores das arestas é a distância entre os pesos sinápticos do neurônio *i* e *j*, *d(wi, wj).* A Figura 24 mostra um par de neurônios *i* e *j* conectados por uma aresta valorada pela distância entre os pesos sinápticos de *i* e *j*.

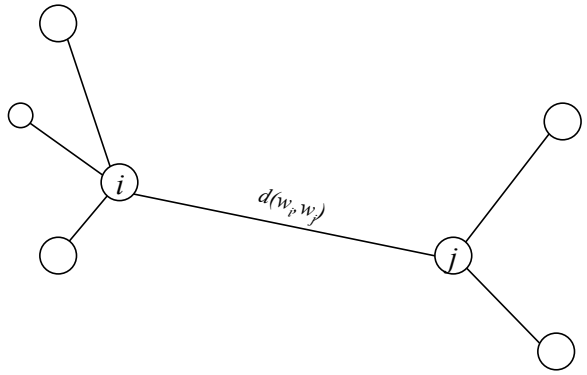


Figura 24 - Vértices e arestas do grafo extraído da grade do mapa auto-organizável (COSTA e NETTO, 2003).

A Figura 25 mostra a grade do mapa da seção 0 visto como o grafo *G'*, onde vértices correspondem aos neurônios da grade e as arestas são representadas pelas relações de vizinhança definida pela topologia da grade. As arestas são valoradas pelas distâncias entre os pesos sinápticos dos neurônios participantes da aresta.

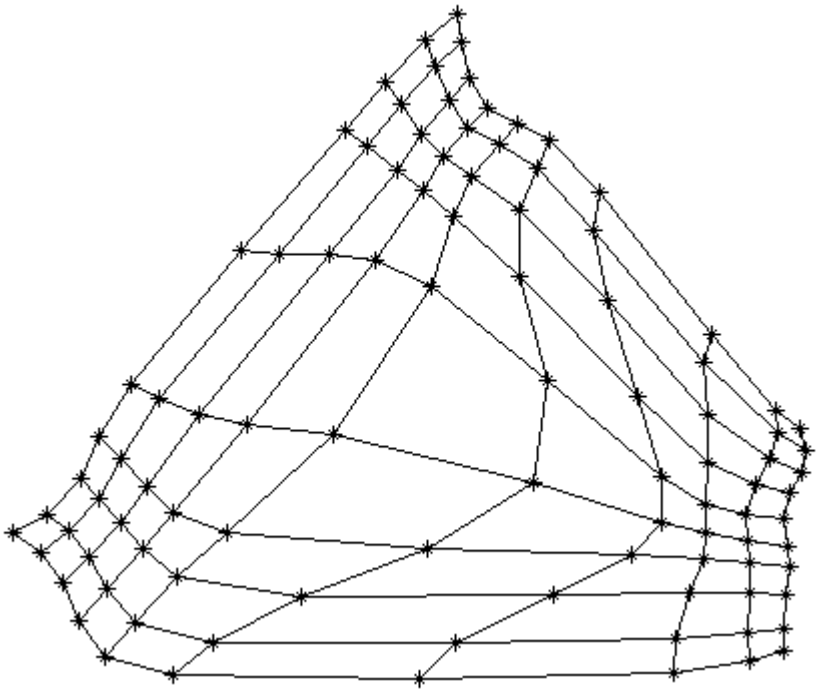


Figura 25 - Grade do mapa treinado da seção visto como um grafo G'.

Eliminação de arestas inconsistentes

O algoritmo proposto em (COSTA e NETTO, 2003) baseia-se em informações geométricas de distância entre os neurônios, no erro de quantização e no nível de atividade do neurônio. A estratégia é unir essas informações e através de heurísticas eliminar conexões do grafo *G'* consideradas inconsistentes, ou seja, eliminar conexões entre neurônios vizinhos que não satisfaçam aos pré-requisitos de um agrupamento de neurônios.

O algoritmo é realizado em duas etapas distintas. A primeira é a detecção de centros de elevada ativação. Em seguida os componentes conectados resultantes são rotulados e há um processo opcional de expansão dos rótulos no grafo (COSTA e NETTO, 2003).

Como resultado do processamento, temos conjuntos de neurônios rotulados que representam os agrupamentos de dados. Tanto o número de agrupamento quanto os membros das classes são determinadas automaticamente pelo algoritmo.

Descrição do algoritmo utilizado para segmentação: Elimina Arestas Inconsistentes

Os parâmetros do algoritmo são:

- Ativação média:

H_{media} = tamanho do conjunto de dados / número total de neurônios

- Ativação mínima:

$H_{min} = \sigma * H_{media}$, onde σ é um valor entre 0,1 e 0,6.

1) Dado um mapa treinado, obtenha as distâncias entre os pesos dos neurônios adjacentes ***i*** e ***j***, $\boldsymbol{d(w_i, w_j)}$ e o número de padrões associados a cada neurônio ***i***, *H(i)*.

2) Para cada par de neurônios adjacentes ***i*** e ***j***, a aresta ***(i,j)*** é considerada inconsistente caso ocorra as condições:

   i) Se a distância entre os pesos excede em 2 a distância média dos outros neurônios adjacentes a ***i*** ou a ***j***;

   ii) Se os dois neurônios adjacentes ***i*** e ***j*** possuem atividade ***H*** abaixo da mínima permitida, $\boldsymbol{H_{min}}$, ou um dos neurônios for inativo, *H(i) = 0*;

3) Remoção dos ramos (arestas) inconsistentes. Para cada aresta ***(i,j)*** considerada inconsistente resultará em conexão nula no endereço ***(i,j)*** da matriz de adjacência.

4) Executar um algoritmo de detecção do número de componentes conexas no grafo podado.

5) Remover componentes conexas com menos de 3 neurônios. O número de componentes conexas restantes representa o número de agrupamentos em que foi divido o conjunto de dados. Os neurônios que compõem cada componente conexa são representantes dos dados do espaço de entrada. Desta forma, obtemos os registros que compõem cada agrupamento.

O algoritmo faz uso de alguns limiares empíricos definidos por meio de experimentações, porém, consegue particionar os dados usando somente as informações inerentes ao mapa treinado, como a distância entre os neurônios, o erro de quantização e o nível de atividade (COSTA e NETTO, 2003).

A Figura 26 mostra o resultado da eliminação das arestas inconsistentes pelo algoritmo proposto sobre o grafo *G'* extraído do mapa treinado da seção 0. Como podemos observar, o grafo tem três componentes conexas o que, como esperado, nos comprava a detecção de três classes no conjunto de dados utilizado para treinamento.

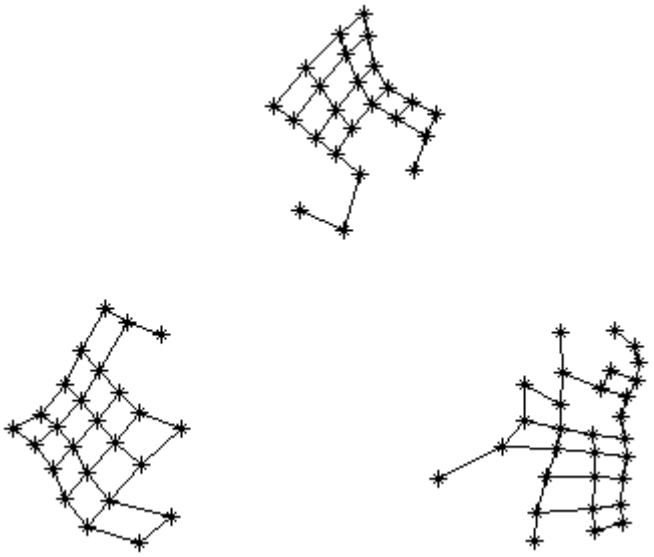


Figura 26 - Grafo G' particionado pelo algoritmo de eliminação de arestas inconsistentes.

5. Exemplos de aplicação e análise das técnicas de segmentação em alguns conjuntos de dados

Este capítulo compara a aplicação de três técnicas de análise de agrupamento sobre alguns conjuntos de dados. Primeiramente, foi aplicado o algoritmo estatístico das k-médias, com o valor de *k* sendo o número de classes esperadas. Em seguida, aplicou-se as duas técnica baseadas no mapa auto-organizável, a segmentação da U-matriz e particionamento de grafos.

A comparação dos diferentes grupos formados foi realizada através da análise de discriminantes para determinar as diferenças significativas entre os grupos.

A estatística de Wilk´s Lambda oferece informação sobre as diferenças entre os grupos, para cada variável individualmente. Obtém-se, pela razão da variação dentro dos grupos (variação não explicada) sobre a variação total. Varia de 0 e 1, onde os pequenos valores indicam grandes diferenças entre os grupos, enquanto que os valores elevados indicam não haver diferenças entre os mesmos. Este teste não considera as correlações entre as variáveis explicativas.

A estatística F é utilizada para descrever os grupos mais parecidos e testar a igualdade das médias (centróides) dos grupos. Pode ser entendida como uma medida de distância entre cada par de grupos, onde valores grandes indicam que os grupos estão longe um dos outros e valores menores indica o inverso.

Conjunto de dados formado por misturas gaussianas

Este conjunto é composto por 375 amostras oriundas de três classes, cada classe tem 125 amostras, geradas à partir de distribuições gaussianas com vetores de média $\mu_1 = (0,0)$, $\mu_2 = (5,5)$ e $\mu_3 = (9,0)$ e foi utilizado na seção 0 para o exemplo do treinamento do mapa. Os dados gerados são apresentados na Figura 27. Podemos identificar visualmente a separação das três classes e como mostrado no decorrer do trabalho, as técnicas de análise de agrupamentos baseadas em mapa auto-organizável conseguiram revelar e determinar a separação do conjunto de dados corretamente.

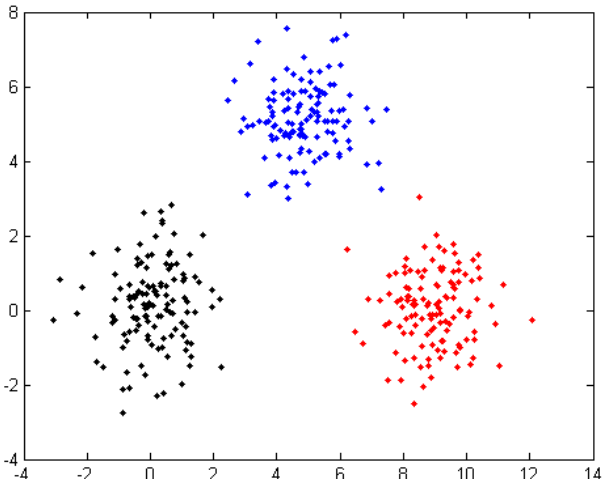


Figura 27- Conjunto de dados formado à partir de três classes com distribuições gaussianas.

O mapa utilizado para este conjunto de dados possui grade bidimensional com topologia retangular de dimensões 10x10. A inicialização dos pesos foi linear. A função de vizinhança usada foi gaussiana e o raio inicial foi 8 caindo para 1 no final da primeira fase do treinamento (fase de ordenação). Foram utilizadas 300 épocas para o treinamento do mapa.

A seguir, comparamos os resultados obtidos pela técnica de agrupamento por k-médias com as duas técnicas de agrupamentos baseadas na segmentação do espaço de saída do mapa treinado.

Análise de agrupamento por k-médias

O algoritmo das k-médias foi executado com $k = 3$, que é o número esperado de classes. A separação das classes pode ser vista na Figura 28 e como podemos observar, o algoritmo teve êxito na separação do conjunto de dados em três classes. As classes são: classe 1 (preto), classe 2(vermelho) e classe 3 (azul).

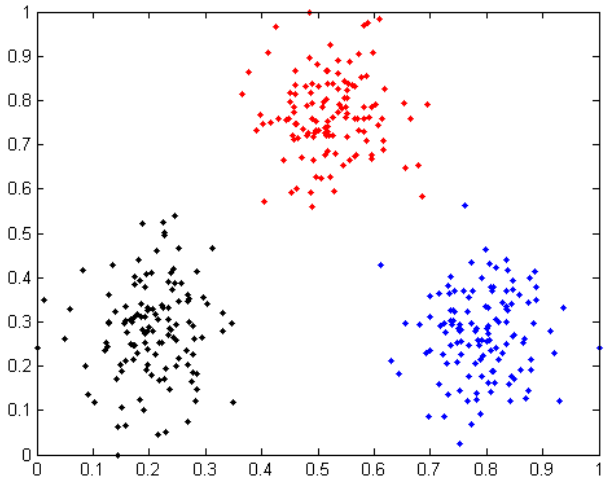


Figura 28 - Separação do conjunto de dados gaussianos pela técnica das k-médias.

A Tabela 2 mostra o resultado da análise de discriminante para os agrupamentos formados. A estatística lambda de Wilk's para esta formação é 0,01012 o que indica que os grupos estão bem separados.

Tabela 2 - Análise de discriminantes dos agrupamentos do conjunto de dados gaussianos obtidos pelo método das k-médias.

Discriminant Function Analysis Summary (dados-exemplo-kmedias_k_3)
No. of vars in model: 2; Grouping: Var3 (3 grps)
Wilks' Lambda: ,01012 approx. F (4,742)=1658,4 p<0,0000

	Wilks' Lambda	Partial Lambda	F-remove (2,371)	p-level	Toler.	1-Toler. (R-Sqr.)
Var1	0,151016	0,067021	2582,277	0,00	0,998449	0,001551
Var2	0,067060	0,150929	1043,555	0,00	0,998449	0,001551

A análise da estatística F para os grupos formados indica quais grupos estão mais longes um dos outros. A Tabela 3 mostra o maior valor de F (F = 2581,4) para o par de agrupamentos 1 (classe 1, em preto na Figura 28) e 3 (classe 3, em azul). Os grupos mais próximos com F = 1395,5 são o 2 (classe 2, em vermelho) e 3.

Tabela 3 - Estatística F para os agrupamentos do conjunto de dados gaussianos pelo método das k-médias.

Estatística F; df = 2,371 (dados-exemplo-kmedias_k_3)

Grupos	1	2	3
1	-	1460,717	2581,483
2	1460,717	-	1395,593
3	2581,483	1395,593	-

Análise de agrupamento por U-matriz

A imagem de marcadores foi obtida à partir da análise do gráfico $k \times N_{rc}^{k}$, onde detectou-se uma zona de estabilidade iniciando em k = 18, Figura 29 (b) . O resultado do processo da aplicação do algoritmo de watershed sobre a imagem da U-matriz é mostrado na Figura 29. Como podemos observar, a U-matriz foi divida em três regiões distintas, Figura 29(d), o que corresponde ao número de classes esperada. A conversão destas regiões para neurônios permite formar três agrupamentos que, por sua vez, podem ser utilizados para determinar os grupos nos dados do conjunto de entrada.

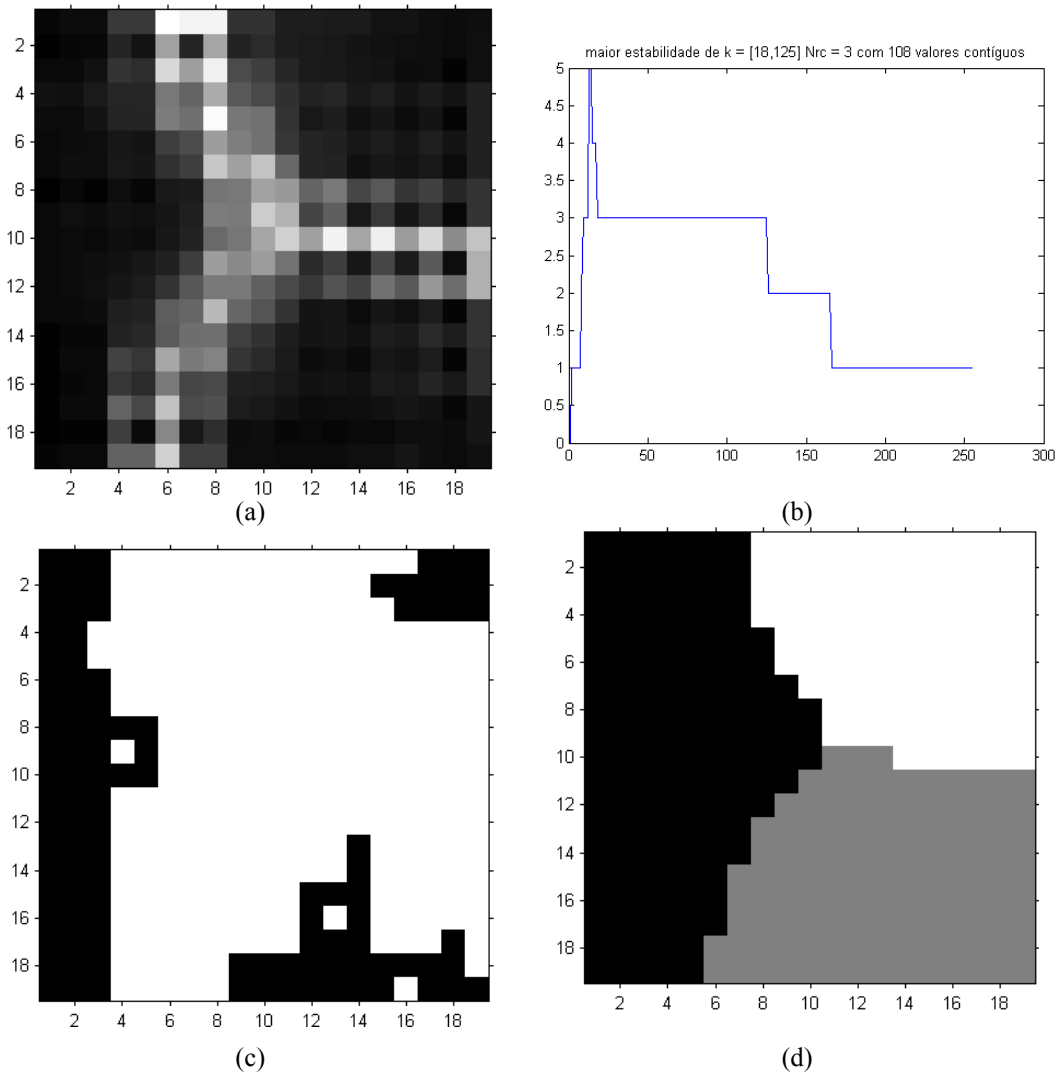


Figura 29 - Segmentação da U-matriz para o conjunto de dados gaussianos.

A separação das classes pode ser vista na Figura 30 - Separação do conjunto de dados gaussianos pela técnica de segmentação da U-matriz.Figura 30 e como podemos

observar, o algoritmo obteve êxito na separação do conjunto de dados em três classes. As classes são: classe 1 (preto), classe 2(vermelho) e classe 3 (azul).

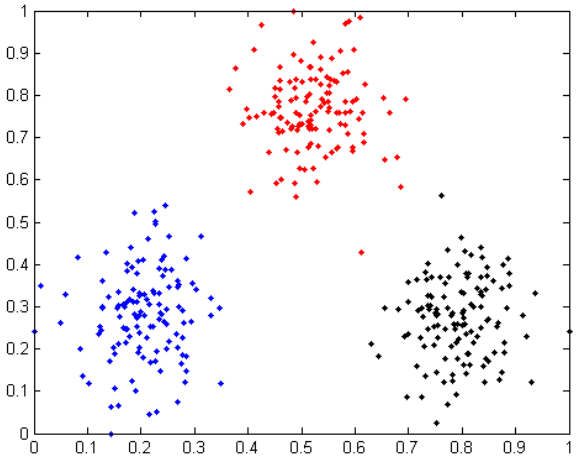


Figura 30 - Separação do conjunto de dados gaussianos pela técnica de segmentação da U-matriz.

A Tabela 4Tabela 2 mostra o resultado da análise de discriminante para os agrupamentos formados por esta técnica. A estatística lambda de Wilk's para esta formação é 0,01021, o que é ligeiramente maior que a obtida por k-médias. No entanto, esta técnica determinou o número de classes na qual se divide o conjunto automaticamente.

Tabela 4 - Análise de discriminantes dos agrupamentos do conjunto de dados gaussianos obtidos pelo método da segmentação da U-matriz.

Discriminant Function Analysis Summary (dados-exemplo-umatrix)
No. of vars in model: 2; Grouping: Var3 (3 grps)
Wilks' Lambda: ,01021 approx. F (4,742)=1650,1 p<0,0000

	Wilks' Lambda	Partial Lambda	F-remove (2,371)	p-level	Toler.	1-Toler. (R-Sqr.)
Var1	0,154858	0,065945	2627,464	0,00	0,998528	0,001472
Var2	0,065977	0,154781	1012,966	0,00	0,998528	0,001472

A análise da estatística F através da Tabela 5 mostra que o maior valor de F (F = 2626,3) é para o par de agrupamentos 1 e 3, o que é também ligeiramente maior que o valor para a técnica das k-médias. Lembrando que quanto maior o valor de F para um par de grupos, mais distantes eles estão um do outro. Desta forma, esta técnica conseguiu uma ligeira melhora na separação dos grupos 1 e 3.

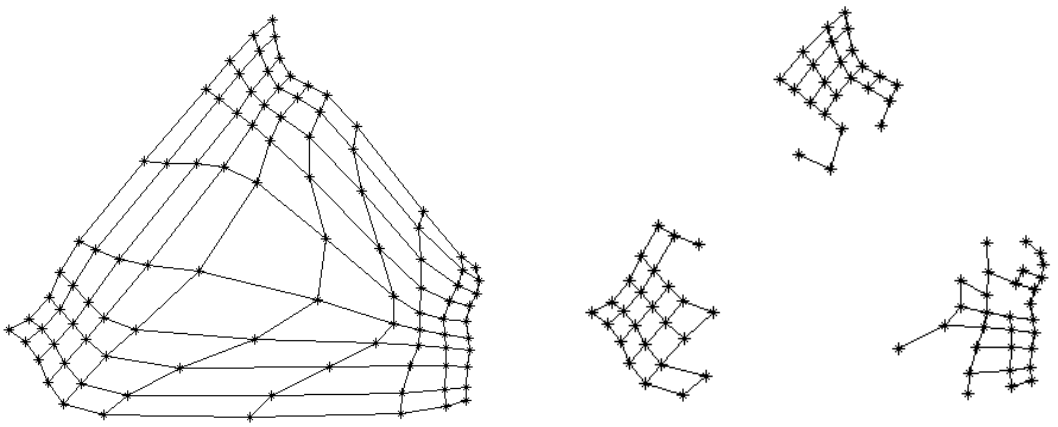
Os grupos mais próximos com F = 1381,5 são o 1 e 2. Novamente, esta técnica conseguiu uma ligeira melhora na separação dos grupos mais próximos que a técnica das k-médias.

Tabela 5 - Estatística F para os agrupamentos do conjunto de dados gaussianos pelo método da segmentação da U-matriz.

Estatística F; df = 2,371 (dados-exemplo-umatrix)			
Grupos	1	2	3
1	-	1381,537	2626,335
2	1381,537	-	1456,724
3	2626,335	1456,724	-

Análise de agrupamento por particionamento de grafos

A Figura 31 (b) mostra o resultado da segmentação do grafo extraído da grade do mapa treinado para o conjunto de dados gaussianos. O algoritmo de eliminação de arestas inconsistentes foi aplicado com o valor de $\sigma = 0,3$.



(a) (b)

Figura 31 - Grafo extraído da grade do mapa (a) e grafo particionado após execução do algoritmo de eliminação de arestas inconsistentes.

Após o particionamento do grafo, busca-se determina todas as componentes conexas ainda presente. Como podemos observar na Figura 31 (b) existem 3 componentes conexas, o que corresponde ao número de classes presentes no conjunto de dados gaussianos. As componentes conexas do grafo são formadas por 3 ou mais neurônios e podemos rotular todos os neurônios pertencentes à uma mesma componente conexa sob um mesmo código. O resultado prático desta rotulação é mostrado na Figura 32. Neurônios marcados em preto na Figura 32 não pertencem a nenhuma componente conexa, pois não satisfazem aos pré-requisitos de um agrupamento de neurônio especificados no algoritmo de eliminação de arestas inconsistentes. No entanto, para obter uma cobertura total do conjunto de dados de entrada, aplicamos uma expansão dos

rótulos aos neurônios sem rótulos. A expansão dos rótulos é feito segundo o critério dos *k*-vizinho mais próximos.

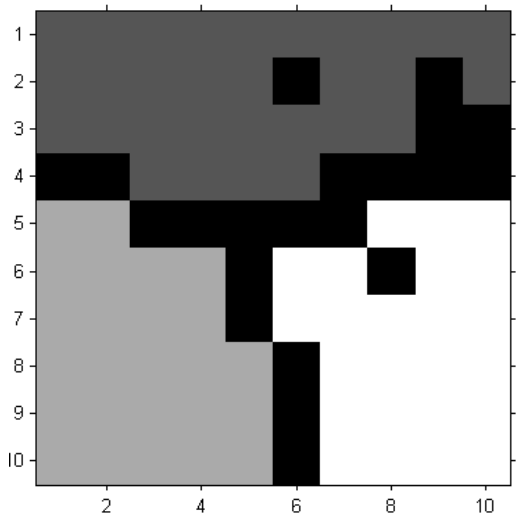


Figura 32 - Rotulação dos neurônios presente em uma mesma componente conexa.

A Figura 33 mostra a separação das classes obtida pela técnica de particionamento de grafos. Como podemos observar, o algoritmo teve êxito na separação do conjunto de dados em três classes. As classes são: classe 1 (preto), classe 2(vermelho) e classe 3 (azul).

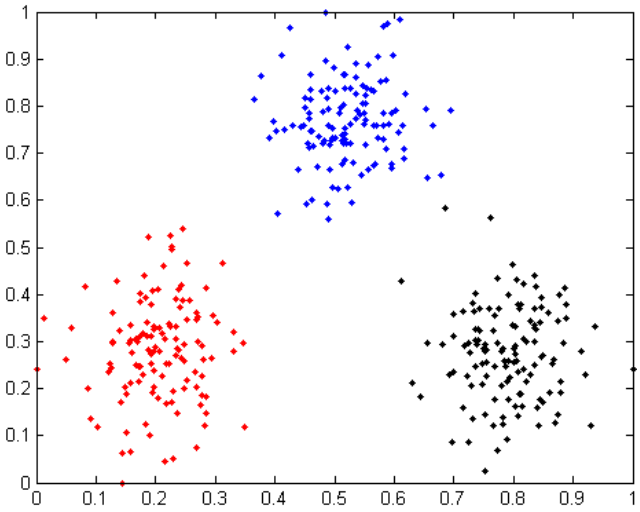


Figura 33 - Separação do conjunto de dados gaussianos pela técnica de particionamento de grafos.

A Tabela 6Tabela 2 mostra o resultado da análise de discriminante para os agrupamentos formados por esta técnica. A estatística lambda de Wilk's para esta formação é 0,01020. Este valor é praticamente igual ao da técnica por segmentação da

U-matriz e com a mesma vantagem de ter descoberto automaticamente o número de classes e ainda apresenta uma boa separação entre os agrupamentos.

Tabela 6 - Análise de discriminantes dos agrupamentos do conjunto de dados gaussianos obtidos pelo método de particionamento de grafos.

Discriminant Function Analysis Summary (dados-exemplo-partGrafo)
No. of vars in model: 2; Grouping: Var3 (3 grps)
Wilks' Lambda: ,01020 approx. F (4,742)=1651,4 p<0,0000

	Wilks' Lambda	Partial Lambda	F-remove (2,371)	p-level	Toler.	1-Toler. (R-Sqr.)
Var1	0,153645	0,066373	2609,313	0,00	0,998563	0,001437
Var2	0,066404	0,153575	1022,383	0,00	0,998563	0,001437

A análise da estatística F através da Tabela 5 mostra que o maior valor de F (F = 2608,7) é para o par de agrupamentos 1 e 2 (preto e vermelho, respectivamente).

Os grupos mais próximos com F = 1383,7 são o 1 e 3 (classe preta e azul, respectivamente).

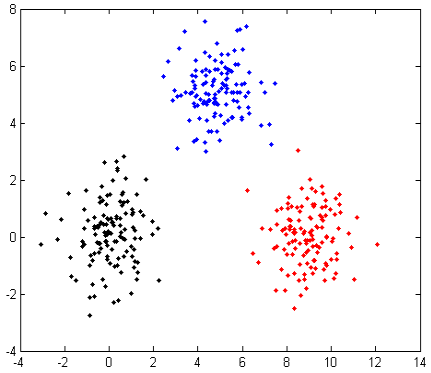
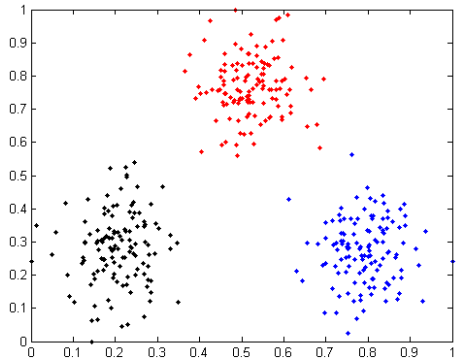
Tabela 7- Estatística F para os agrupamentos do conjunto de dados gaussianos pelo método de particionamento de grafos.

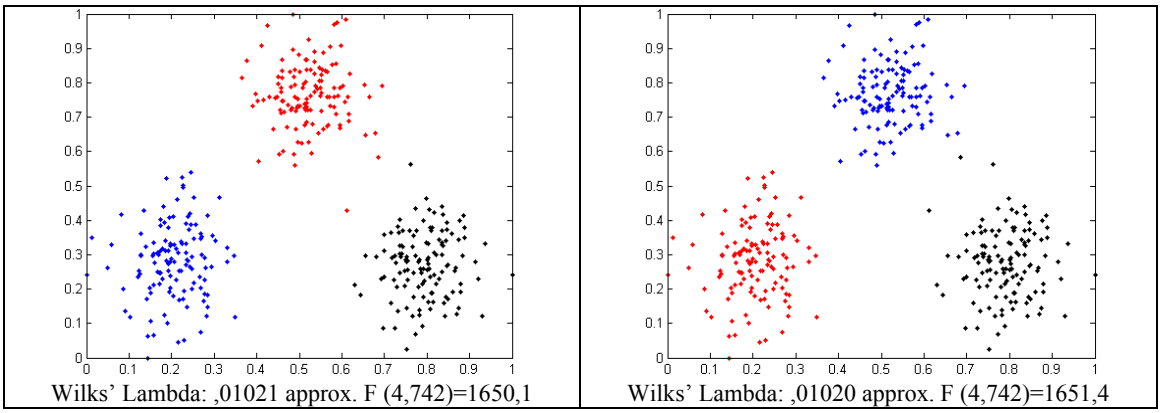
Estatística F; df = 2,371 (dados-exemplo-partGrafo)

Grupos	1	2	3
1	-	2608,717	1383,714
2	2608,717	-	1449,812
3	1383,714	1449,812	-

Sumário

Tabela 8 - Comparativo das técnicas k-médias, U-matriz e particionamento de grafos para o conjunto de dados gaussiano.

Dados gaussianos gerados artificialmente com vetores de média μ1 = (0,0), μ2 = (5,5) e μ3 = (9,0)	Grupos formados pelo algoritmo das k-médias com k = 3
	 Wilks' Lambda: ,01012 approx. F (4,742)=1658,4
Grupos formados pela segmentação da U-matriz. τ = 3 e conectividade 4-pixels.	Grupos formados pelo particionamento de grafos. σ = 0.3



Conjunto de dados Chainlink

Ultcsh (FCPS, 2007) propôs uma variedade de conjuntos de dados para problemas de análise de agrupamento. O **FCPS,** *Fundamental Clustering Problems Suite,* serve como *benchmark* para avaliar as técnicas de análise de agrupamento.

Um conjunto de dados não trivial para comparações de métodos de agrupamentos disponível no FCPS é o *chainlink*. Este conjunto consiste de 1000 pontos no espaço $\Re^3$ 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 *x*-*z*. Os anéis podem ser pensados como elementos de uma corrente, cada um consistindo de 500 objetos de dados. A Figura 34 ilustra o conjunto de dados usado.

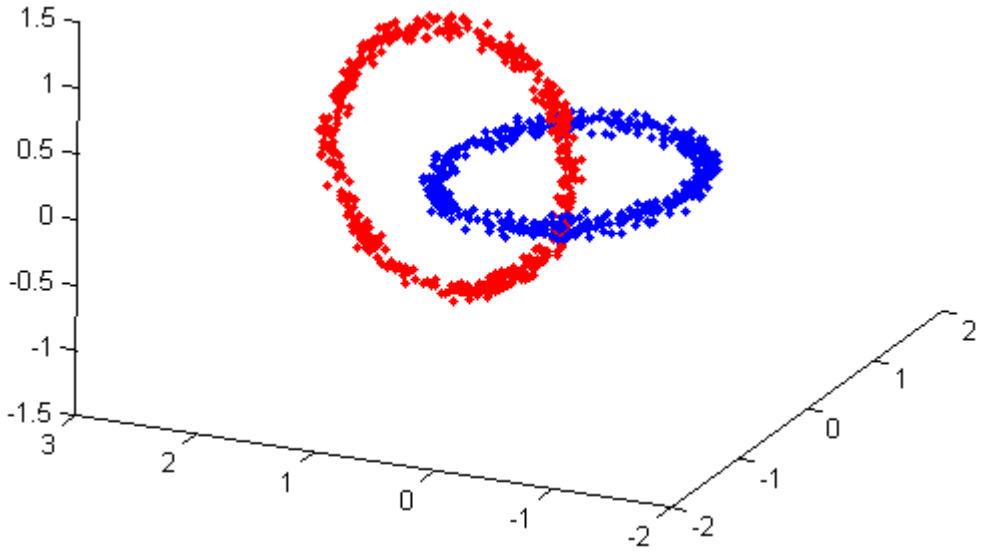


Figura 34 - Conjunto de dados *chainlink*.

O mapa utilizado para este conjunto de dados possui grade bidimensional com topologia retangular de dimensões 15x15. A inicialização dos pesos foi linear. A função de vizinhança usada foi gaussiana e o raio inicial foi 12 caindo para 1 no final da primeira fase do treinamento (fase de ordenação). Foram utilizadas 500 épocas para o treinamento do mapa. O resultado da ordenação topológica dos neurônios após o treinamento pode ser vista na

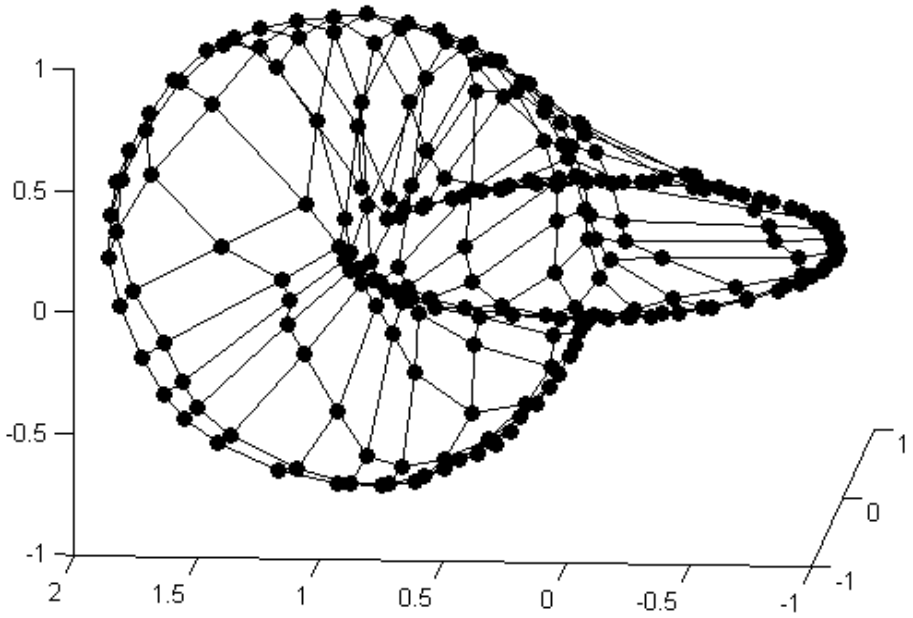


Figura 35 - Grade do mapa 15x15 após treinamento por 500 épocas sobre o conjunto de dados chainlink.

Análise de agrupamento por k-médias

O algoritmo das k-médias foi executado com $k = 2$, que é o número esperado de classes. A separação das classes pode ser vista na Figura 36,Figura 28 e como podemos observar, o algoritmo das k-médias não conseguiu separa o conjunto de dados *chainlink* de acordo com a geometria dos dados. As classes encontradas são: classe 1 (azul) e classe 2 (vermelha).

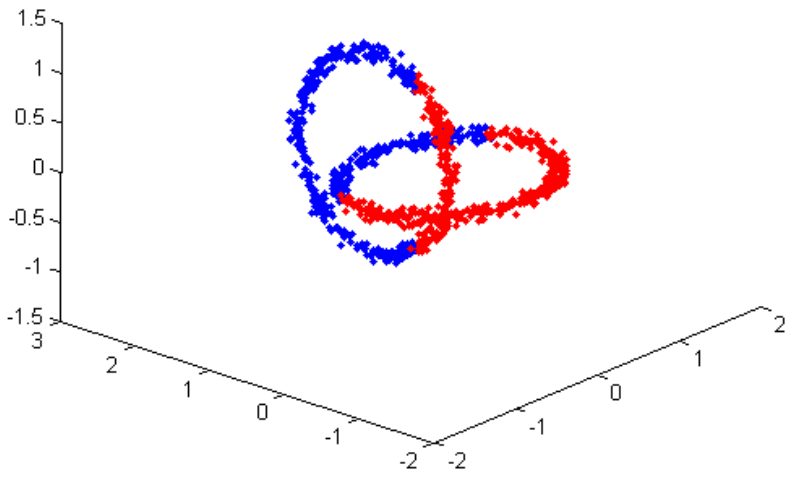


Figura 36 - Separação do conjunto de dados chainlink pela técnica das k-médias.

Tabela 9 - Análise de discriminantes dos agrupamentos do conjunto de dados chainlink obtidos pelo método das k-médias.

Discriminant Function Analysis Summary (chainlink-k-médias)
No. of vars in model: 3; Grouping: Var4 (2 grps)
Wilks' Lambda: ,31183 approx. F (3,996)=732,69 p<0,0000

	Wilks' Lambda	Partial Lambda	F-remove (1,996)	p-level	Toler.	1-Toler. (R-Sqr.)
Var1	0,311837	0,999973	0,027	0,869914	0,999931	0,000069
Var2	0,998551	0,312281	2193,431	0,000000	0,998816	0,001184
Var3	0,312515	0,997805	2,191	0,139160	0,998820	0,001180

Tabela 10 - Estatística F para os agrupamentos do conjunto de dados *chainlink* pelo método das k-médias.

Estatística F; df = 3,996 (chainlink-kmedias)		
Grupos	1	2
1	-	732,6866
2	732,6866	-

Análise de agrupamento por U-matriz

A U-matriz em forma de relevo topográfico para o conjunto de dados chainlink pode ser vista na Figura 37. A imagem de marcadores foi obtida à partir da análise do gráfico $k \times N_{rc}^{k}$, onde detectou-se uma zona de estabilidade iniciando em k = 61, Figura 38 (b) . O resultado do processo da aplicação do algoritmo de watershed sobre a imagem da U-matriz é mostrado na Figura 38.

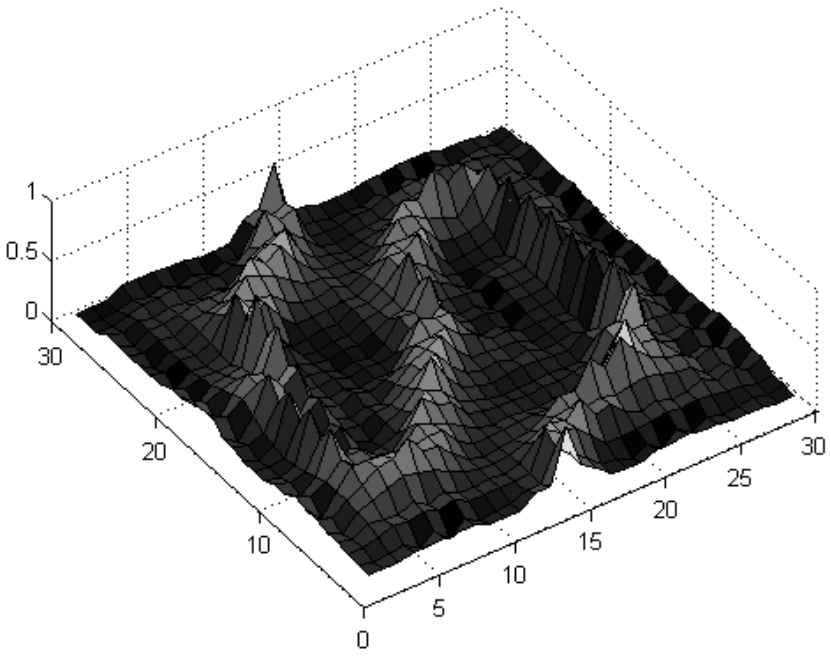
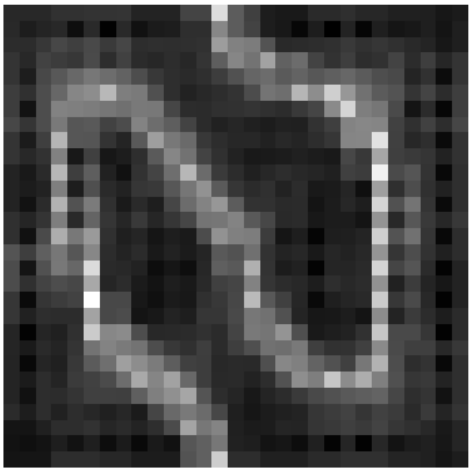
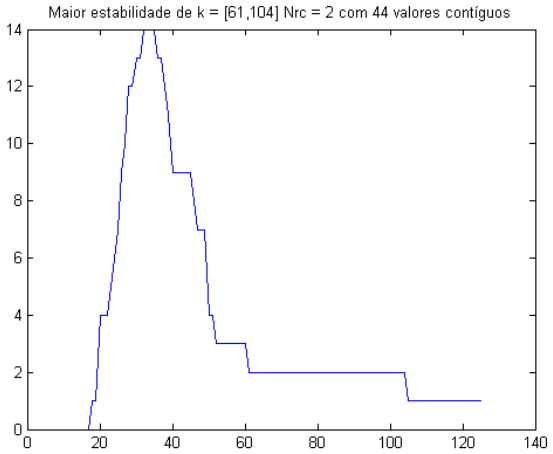


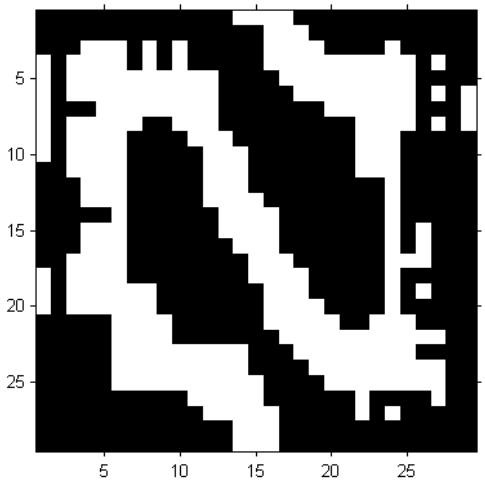
Figura 37 - U-matriz como relevo topográfico para mapa 15x15 com topologia retangular.



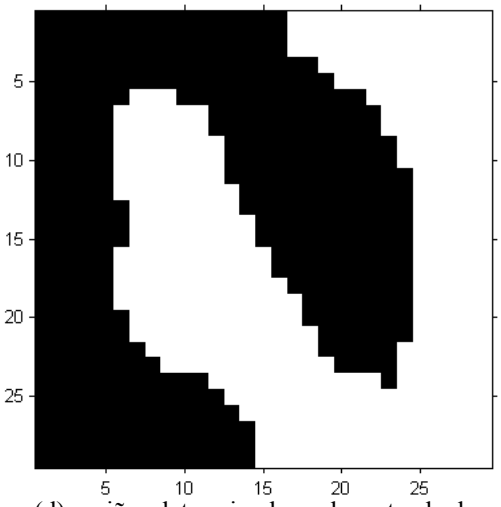
(a) U-matriz



(b) $k \times N_{rc}^{k}$



(c) imagem de marcadores



(d) regiões determinadas pela watershed

Figura 38 - Segmentação da U-matriz para o conjunto de dados *chainlink*.

Como podemos observar, a U-matriz foi divida em duas regiões distintas, Figura 38 (d) e a Figura 39 mostra as classes encontradas: classe 1 (azul) e classe 2 (vermelha).

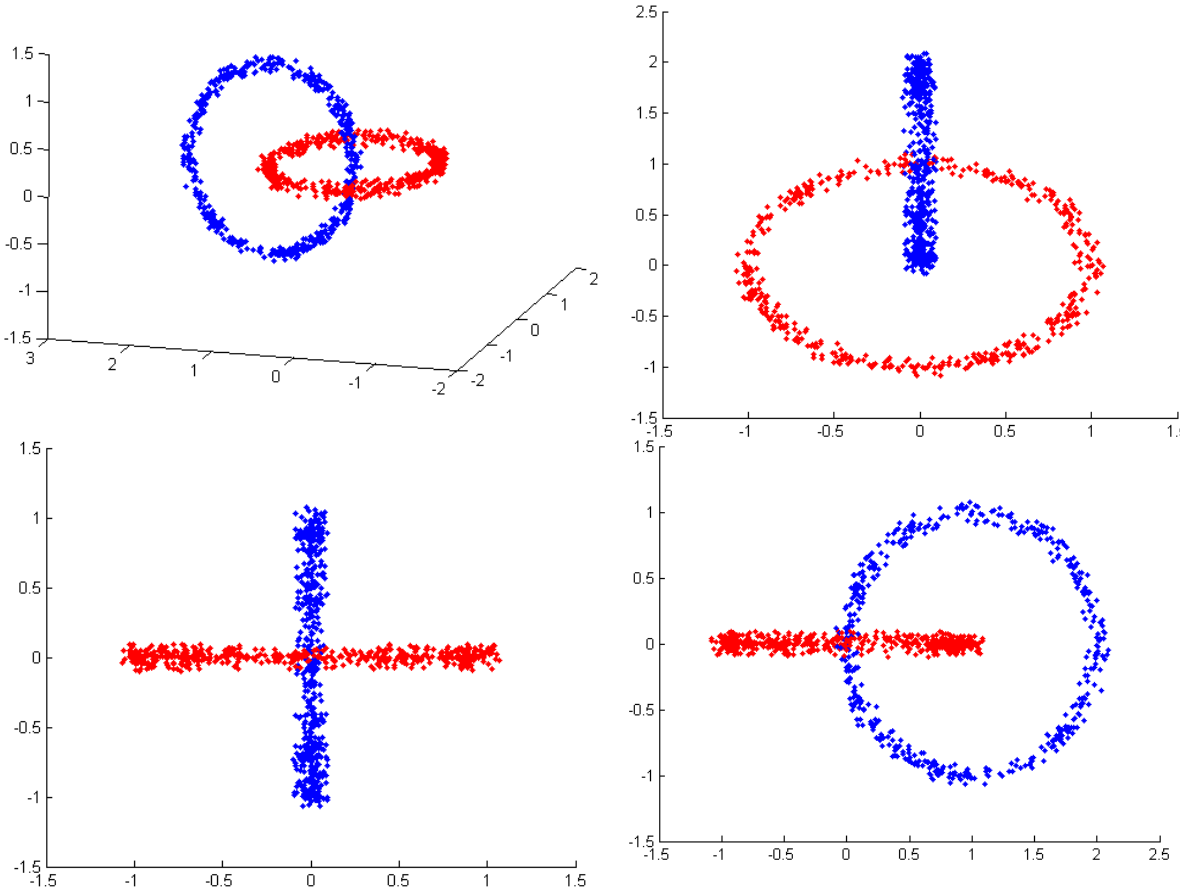


Figura 39 - Separação do conjunto de dados chainlink pela técnica de segmentação da U-matriz.

Tabela 11 - Análise de discriminantes dos agrupamentos do conjunto de dados chainlink obtidos pelo de segmentação da U-matriz.

Discriminant Function Analysis Summary (chainlink-umatrix-grade-rect)
No. of vars in model: 3; Grouping: Var4 (2 grps)
Wilks' Lambda: ,67289 approx. F (3,996)=161,39 p<0,0000

	Wilks' Lambda	Partial Lambda	F-remove (1,996)	p-level	Toler.	1-Toler. (R-Sqr.)
Var1	0,673073	0,999734	0,2651	0,606732	0,999934	0,000066
Var2	0,998869	0,673656	482,5007	0,000000	0,999935	0,000065
Var3	0,673245	0,999479	0,5193	0,471294	0,999934	0,000066

Tabela 12- Estatística F para os agrupamentos do conjunto de dados chainlink pelo método de segmentação da U-matriz.

Estatística F; df = 3,996 (chainlink-umatrix-grade-rect)

Grupos	1	2
1	-	161,3914
2	161,3914	-

Análise de agrupamento por particionamento de grafos

A Figura 40Figura 31 mostra o resultado da segmentação do grafo extraído da grade do mapa treinado, Figura 40(a), para o conjunto de dados *chainlink*. O algoritmo de eliminação de arestas inconsistentes foi aplicado com o valor de σ = 0,3.

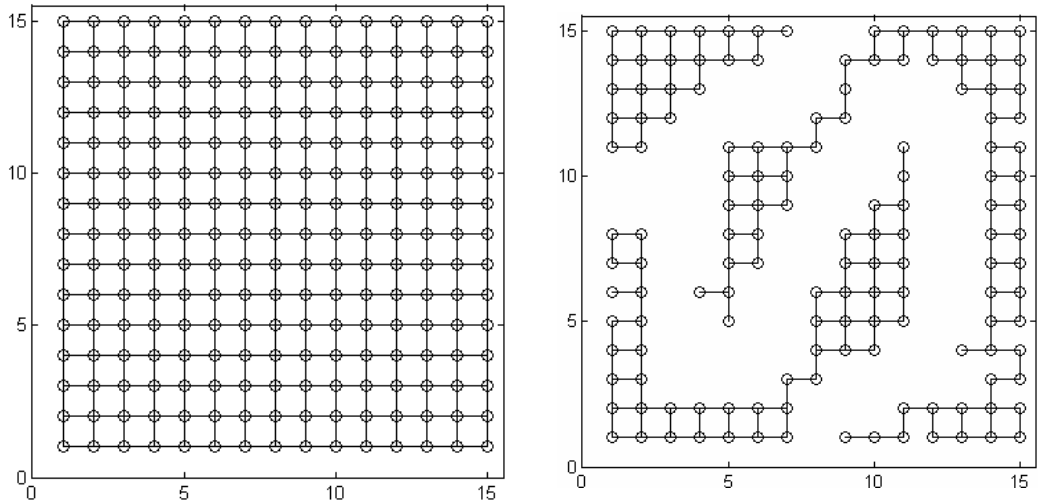
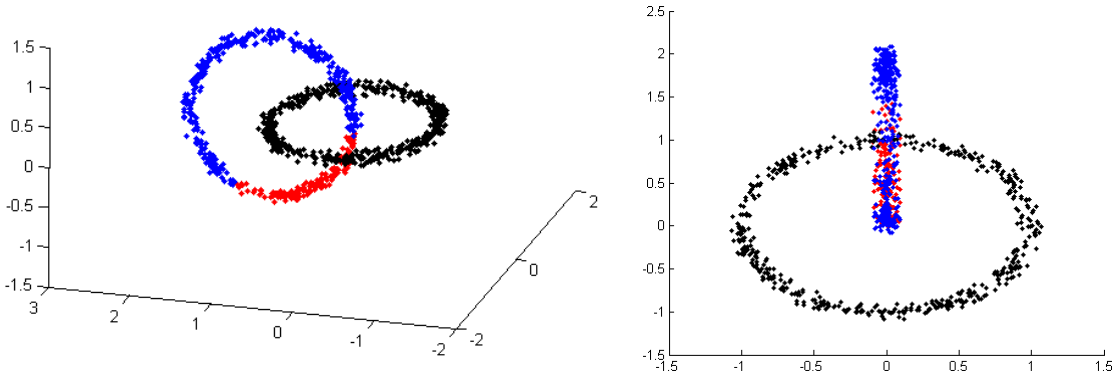


Figura 40 - Grafo extraído da grade do mapa 15x15 (a) e grafo particionado após execução do algoritmo de eliminação de arestas inconsistentes.

Como podemos observar, o algoritmo determinou três componentes conexas, o que faz com que o método tenha determinado três classes. As classes são mostradas na Figura 41: classe 1 (azul), classe 2 (vermelha) e classe 3 (preta).



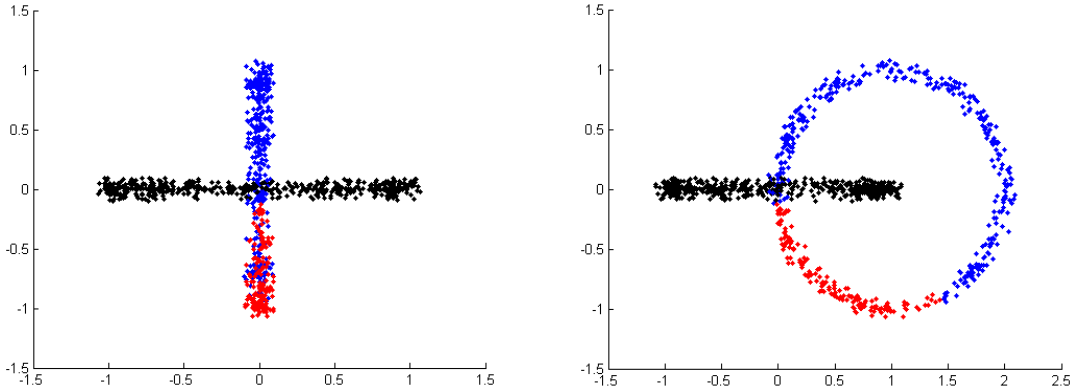


Figura 41 – Separação das classes pelo método de particionamento de grafos sobre uma grade 15x15 retangular.

No entanto, como estamos utilizando alguns valores empíricos para o critério de aresta inconsistente, pode ser que ocorra uma eliminação indesejada, pois os testes realizados utilizam valores absolutos nas comparações. Uma alternativa para se conseguir melhores resultados com esta técnica é utilizar uma topologia que tenha mais vizinhança para os neurônios, i.e., mais arestas serão avaliadas no momento da avaliação de inconsistência. Para a mesma rede com grade bidimensional utilizada anteriormente, mudamos a topologia de retangular para hexagonal e o resultado da organização dos neurônios após os mesmos parâmetros de treinamento é mostrado na Figura 42. Podemos observar que as conexões entre neurônios vizinhos são bem mais densas do que para o caso da grade com topologia retangular.

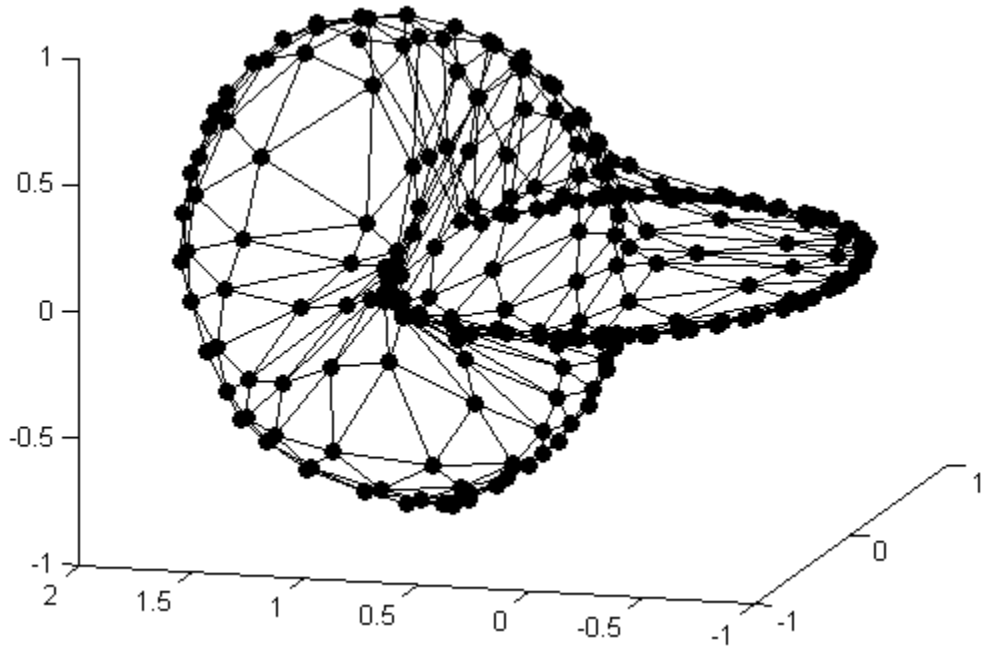


Figura 42 - Grade 15x15 treinada com topologia hexagonal para o conjunto de dados *chainlink*.

A Figura 43 mostra o resultado da segmentação do grafo extraído da grade do mapa utilizando a topologia hexagonal. O algoritmo de eliminação de arestas inconsistentes foi aplicado com o valor de σ = 0,3. Desta vez, podemos observar a detecção de duas componentes conexas de neurônios o que determina as duas classes esperadas. A Figura 44 mostra a separação das classes obtidas pela técnica de particionamento de grafos quando aplicada a um mapa com topologia hexagonal.

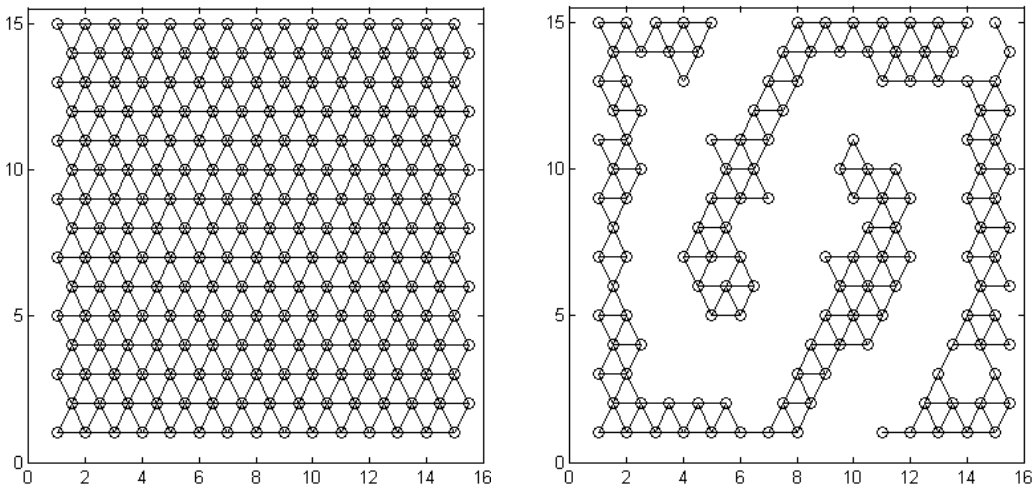
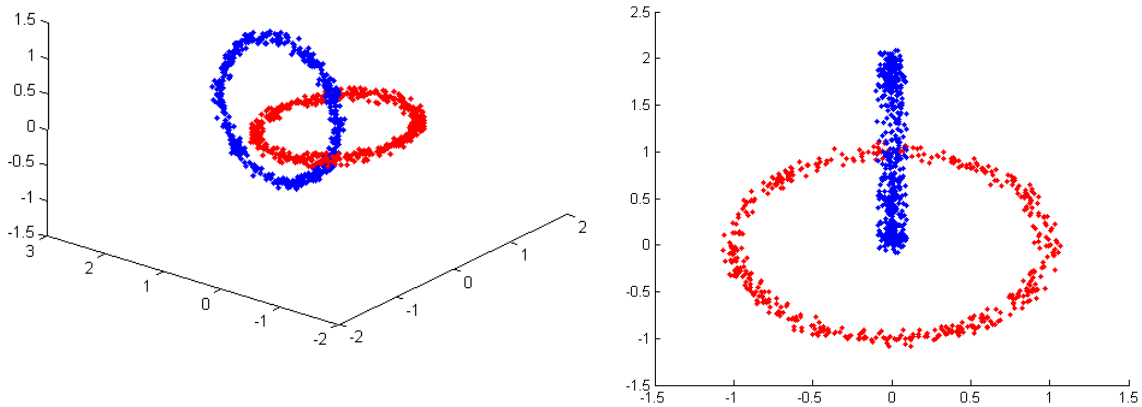


Figura 43 - Grafo extraído da grade do mapa 15x15 com topologia hexagonal (a) e grafo particionado após execução do algoritmo de eliminação de arestas inconsistentes.



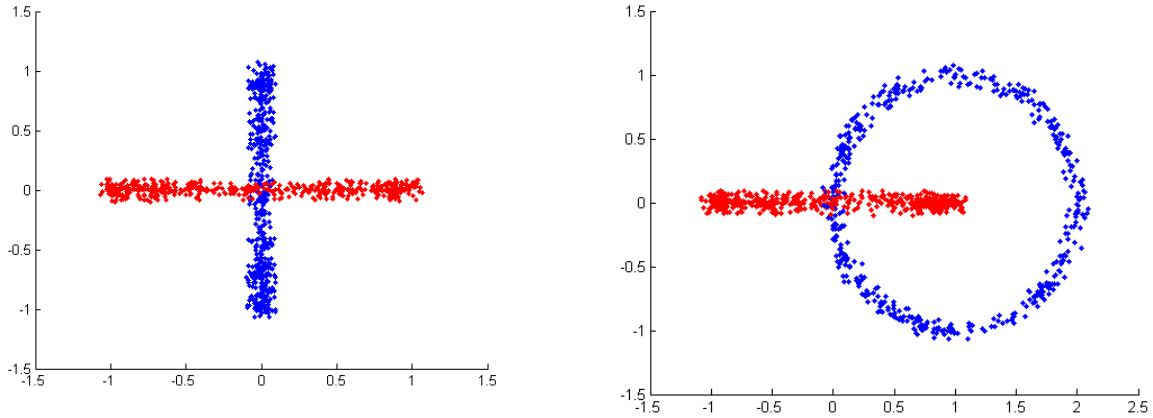


Figura 44 - Separação das classes pelo método de particionamento de grafos sobre uma grade 15x15 hexagonal.

Tabela 13 - Análise de discriminantes dos agrupamentos do conjunto de dados chainlink obtidos pelo particionamento de grafos.

Discriminant Function Analysis Summary (chainlink-partGrafo-grade-hexa)
No. of vars in model: 3; Grouping: Var4 (2 grps)
Wilks' Lambda: ,67289 approx. F (3,996)=161,39 p<0,0000

	Wilks' Lambda	Partial Lambda	F-remove (1,996)	p-level	Toler.	1-Toler. (R-Sqr.)
Var1	0,673073	0,999734	0,2651	0,606732	0,999934	0,000066
Var2	0,998869	0,673656	482,5007	0,000000	0,999935	0,000065
Var3	0,673245	0,999479	0,5193	0,471294	0,999934	0,000066

Tabela 14 - Estatística F para os agrupamentos do conjunto de dados chainlink pelo método de particionamento de grafos.

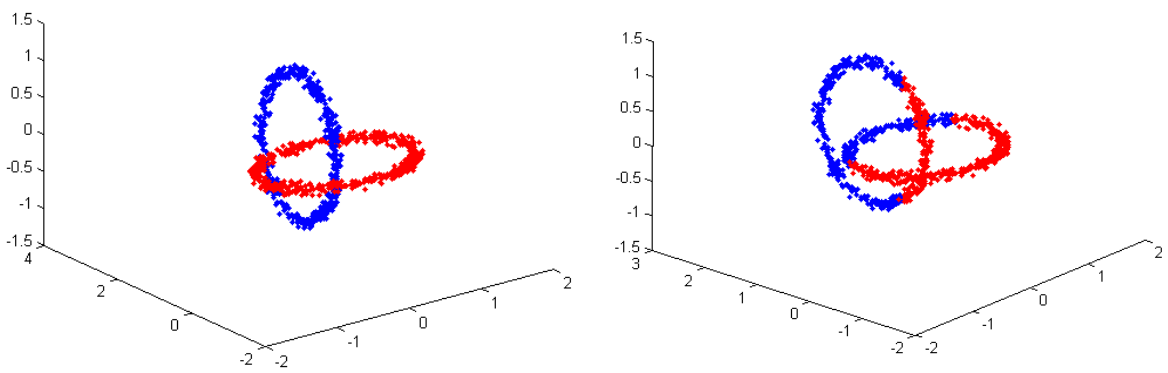
Estatística F; df = 3,996 (chainlink-partGrafo-grade-hexa)		
Grupos	1	2
1	-	161,3914
2	161,3914	-

Sumário

Tabela 15 - Comparativo das técnicas k-médias, U-matriz e particionamento de grafos para o conjunto de dados *chainlink*.

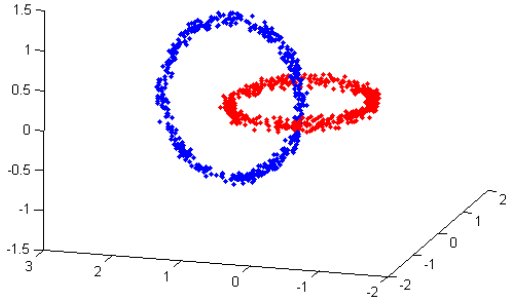
Conjunto de dados chainlink

Grupos formados pelo algoritmo das k-médias com k = 3



Três classes não separáveis linearmente.

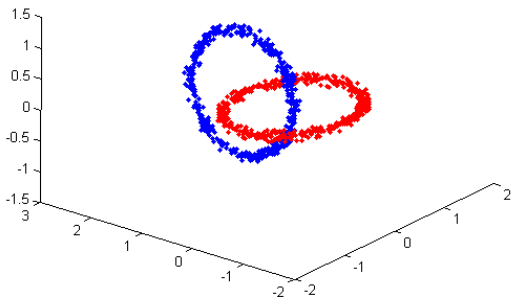
Grupos formados pela segmentação da U-matriz.
$\tau = 3$ e conectividade 4-pixels.



Wilks' Lambda: ,67289 approx. F (3,996)=161,39 p<0,0000

Wilks' Lambda: ,31183 approx. F (3,996)=732,69 p<0,0000

Grupos formados pela partição do grafo.
Mapa com topologia hexagonal de dimensão 10x10.
$\sigma = 0.3$



Wilks' Lambda: ,67289 approx. F (3,996)=161,39 p<0,0000

Este problema ilustra a capacidade do mapa auto-organizável em descobrir a estrutura dos dados mesmo para um conjunto de dados com forma complexa, não-esféricas e não separáveis linearmente.

Conclusões

As duas técnicas apresentadas baseadas no espaço de saída do mapa usam um conjunto de neurônios para representar agrupamentos no conjunto de entrada, o que resulta em uma capacidade das técnicas de se adaptar a diferentes geometrias no espaço dos dados ao invés da imposição de uma estrutura aos dados, o que ocorre nos principais métodos estatísticos de análise de agrupamento.

Em aplicações tais como os de mineração de dados e descoberta de conhecimento em bases de dados, os algoritmos apresentados podem ser bastante úteis, apresentando não só os grupos existentes, mas também seus elementos e relações. Porém não há na atualidade, métodos gerais que sejam adaptáveis a uma vasta gama de tipos de dados e geometrias dos agrupamentos.

A análise dos mapas descrita no trabalho supõe que o treinamento tenha sido efetuado com sucesso. Seja qual for a ferramenta utilizada, a interpretação dos agrupamentos é essencial para a certeza do que se está realizando. Apenas um especialista no assunto tratado pode interpretar corretamente os grupos formados (CARVALHO, 2001).

Anexos

```
function [idxsCluster] = clusterPorKmedias(sD, k)

IDX = kmeans(sD.data, k, 'distance', 'sqEuclidean', 'start',
'uniform');

idxsCluster=[];

for i=1:k
    idxs = find(IDX == i);
    idxsCluster = [idxsCluster, {idxs}];
end
```

```
function [idxsCluster, nc] = clusterPorUmatriz(sM, sD)
%sM - mapa treinado
%sD - conjunto de dados matriz m x n, onde m e o numero de exemplos e
n a
%dimensionalidade

%obtem a U-matriz
U = som_umat(sM);

%obtem a imagem de marcadores
[im, nUmatriz, conectividade] = encontraMarcadores(U);

%executa watershed com imagem de marcadores
L = mmcwatershed(nUmatriz, ~im, conectividade, 'REGIONS');
figure, imshow(im, [], 'notruesize');
figure, imshow(L, [], 'notruesize');


%numero de regioes encotradas (clusters)
nc = mmstats(L,'max');

%converte a u-matriz para o tamanho da grade do mapa
dMatriz = L(1:2:size(L,1),1:2:size(L,2));
%dimensao do mapa
grade = sM.topol.msize;

clutersNeuronios = [];
for i=1:nc
    [pixelx, pixely] = find(dMatriz == i);
    neuronios = pos2neuronio([pixelx pixely]', grade(1), grade(2));
    clutersNeuronios = [clutersNeuronios, {neuronios}];
end

%numero do neuronio vencedor para cada dado do conj. de entrada
bmu = som_bmus(sM, sD);

idxsCluster = [];
for i=1:nc
    idxs = neuronio2indiceRegistro(bmu, clutersNeuronios{i});
    idxsCluster = [idxsCluster, {idxs}];
end
```

```
function [idxsCluster, nc, clutersNeuronios] =
clusterPorSegmentacaoGrafo(sM, sD)
%segmenta o grafo gerado a partir da grade de saída do mapa treinado
```

```
sigma = 0.3;
A = segmentaPartGrafo(sM, sD, sigma);

limiarComponente = 3;
[nc, clutersNeuronios] = componentesconexas(A, limiarComponente);

%numero do neuronio vencedor para cada dado do conj. de entrada
bmu = som_bmus(sM, sD);

%analisa se todos os neuronios fazem parte de algum agrupamento
mgrade = sM.topol.msize;
 classes = repmat(-1, 1, mgrade(1) * mgrade(2));
 for i=1:nc
     neuronios = clutersNeuronios{i};
     classes(neuronios) = i;
 end;
 neuroniosSemClasse = find(classes == -1);

 %mapeamento dos neuronios sem classe por k-vizinhos mais proximos
 if ~isempty(neuroniosSemClasse)
     %matriz de vizinhanca dos neuronios
     ne1 = som_unit_neighs(mgrade, sM.topol.lattice, sM.topol.shape);

     %para cada neuronio sem classe
     for k=1:length(neuroniosSemClasse)

         nSemClasse = neuroniosSemClasse(k);
         vizinhos = find(ne1(nSemClasse,:) == 1);

         %rank das classes dos vizinhos
         rank = zeros(1, nc);
         for j=1:length(vizinhos)
             %numero do neuronio vizinho
             v = vizinhos(j);
             %ve sua classe
             cl = classes(v);
             if cl ~= -1 %tem uma classe
                 rank(cl) = rank(cl) + 1;
             end
         end
         %ve qual classe ganhou
         [maxRank, cl] = max(rank);
         if maxRank == 0
             disp(['Problemas']);
         end

         %atribui neuronio sem classe ao grupo esta mais proximo
         clutersNeuronios{cl} = [clutersNeuronios{cl}, nSemClasse];
     end
 end

idxsCluster = [];
for i=1:nc
    idxs = neuronio2indiceRegistro(bmu, clutersNeuronios{i});
    idxsCluster = [idxsCluster, {idxs}];
end
```

```
function [im, filtradaU, conectividade] = encontraMarcadores(Umatrix)

%normaliza a u-matrix para representar cores
intensidadeU = mat2gray(Umatrix);
```

```
%converte para 256 tons de cinza
indexadaU = uint8(gray2ind(intensidadeU,256));

% conectiviade para contagem de componentes conexas da imagem BW
quatroConectividade = mmsecross;
oitoConectividade = mmsebox;

%escolha da conectividade
conectividade = oitoConectividade;

%Filtragem
minArea = 3;
filtradaU = mmareaclose(indexadaU, minArea, mmsecross);
%mmareaopen
%filtradaU = mmgradm(indexadaU, mmsecross, mmsecross(2));
fmax = uint16(max(filtradaU(:)));
fmin = uint16(min(filtradaU(:)));;

%figure, imshow(filtradaU, [], 'notruesize')
fk = zeros(1, fmax - fmin + 1);
for k=fmin:fmax
    ipb = mmbinary(filtradaU, k);
    %os objetos representam as linhas de separacoes
    %para saber os numeros de regios ~bw
    rotulos = mmlabel(~ipb, conectividade);
    %numero de regioes conectadas
    nrc = mmstats(rotulos,'max');
    fk(k - fmin + 1) = nrc;
end

[iniK, fimK, qtd] = maxContiguaRC(fk);

% ajusta indices para [0,255]
fk_inicio = iniK + fmin - 1;
fk_fim = fimK + fmin - 1;
NrcEstavel = fk(iniK);
idxs = fmin:fmax;
plot(idxs, fk);
title(['Maior estabilidade de k = [', num2str(fk_inicio), ',',
num2str(fk_fim), '] Nrc = ', num2str(NrcEstavel), ' com ',
num2str(qtd),' valores contíguos'])

%imagem de marcadores
im = mmbinary(filtradaU, fk_inicio);

%figure, imshow(im, [], 'notruesize')
```

```
function [iniK, fimK, qtdElementos] = maxContiguaRC(aVetor)
%identifica a maxima sequencia contigua de elementos em um vetor
idxInicio = 1;
numElementos = 1;

qtdElementos = 1;
iniK = 1;
fimK = 1;

oElemento = aVetor(idxInicio);

tamanho = length(aVetor);
for i=2:tamanho
```

```
    if oElemento == aVetor(i)
        numElementos = numElementos + 1;
    else
        if numElementos >= qtdElementos
            iniK = idxInicio;
            fimK = i - 1;
            qtdElementos = numElementos;
        end
        oElemento = aVetor(i);
        idxInicio = i;
        numElementos = 1;
    end
end
if numElementos >= qtdElementos
    iniK = idxInicio;
    fimK = tamanho;
    qtdElementos = numElementos;
end
```

```
function [A] = segmentaPartGrafo(sM, sD, sigma)

%% Algoritmo proposto pelo artigo "Segmentção do SOM baseada em
%% particionamneto de grafos" de José Alfredo Costa e Márcio Luiz
Netto.

if nargin < 3, sigma = 0.5; end; %% range 0.1 e 0.6

%% obtem as dimensoes do mapa
dims = sM.topol.msize;

%% obtem o numero de neuronios
nTotalNeuronios = dims(1) * dims(2);

% %% obtem o tamanho do conjunto de dados
nTamanhoDados = length(sD.data);

%% obtem a topologia da som
topologia = sM.topol.lattice;


%% vetor de atividade
H = som_hits(sM,sD);;

%% obtem a distancias entre neuronios
W = dist(sM.codebook, sM.codebook');

%% Hmin e Hmed
Hmed = nTamanhoDados / nTotalNeuronios;
Hmin = sigma * Hmed;

%matriz de adjacencia inicializada sem nenhuma incidencia.
Adj = repmat(0, nTotalNeuronios, nTotalNeuronios);

%matriz de vizinhanca dos neuronios
ne1 = som_unit_neighs(dims,topologia, sM.topol.shape);

for i=1:nTotalNeuronios

    %% obtem os 1-vizinhos para neurônio i
    vizinhosI = find(ne1(i,:) == 1);
```

```
    %% seta 1 no end. (i,j) da matriz de adjacencia
    Adj(i, vizinhosI) = 1;
    Adj(vizinhosI, i) = 1;

    %% para cada neurônio adjacente ao neuronio i, verificar
    %% se a aresta (i,j) é inconsitente e podar a aresta.
    for idx=1:length(vizinhosI)
         %% o neurônio j
         j = vizinhosI(idx);

         %% corta a componente (i,j)
         idxs = find( vizinhosI ~= j );
         vizinhosIsemJ = vizinhosI(idxs);
         %% obtem a distancia média dos outros pesos dos neuronios
adjacentes
         %% diferente de (i,j)
         Wimedio = mean( W(i, vizinhosIsemJ) );

        %% obtem os 1-vizinhos para neurônio j
        vizinhosJ = find(ne1(j,:) == 1);

        %% corta a componente (j,i)
        idxs = find( vizinhosJ ~= i );
        vizinhosJsemI = vizinhosJ(idxs);

        %% obtem a distancia média dos outros pesos dos neuronios
adjacentes
        %% diferente de (j,i)
        Wjmedio = mean( W(j, vizinhosJsemI) );

        if ehInconsistente(i, j, W, Wimedio, Wjmedio, H, Hmin)
            Adj(i,j) = 0;
            Adj(j,i) = 0;
        end;
    end;
end;

A = Adj;
```

```
function bool = ehInconsistente(i, j, D, Dimean, Djmean, H, Hmin)

bool = 0;
limiar = 2;
if (D(i, j) > limiar * Dimean) || (D(i, j) > limiar * Djmean)
    bool = 1;
   % disp(['Eliminado por distancia (', num2str(i), ',',
num2str(j),')']);
    return;
end;

if (((H(i) < Hmin) || (H(j) < Hmin))  )
    bool = 1;
    %disp(['Eliminado por ativação. (', num2str(i), ',',
num2str(j),')']);
    return;
end;
if (H(i) == 0 || H(j) == 0)
    bool = 1;
    %disp(['Eliminado por ativação zero. (', num2str(i), ',',
num2str(j),')']);
    return;
```

```
end;
```

```
function [c, theComponets] = componentesconexas(A, threshold)

% [c, theComponets] = componentesconexas(A, threshold) - retorna o
numero
% de componentes conexas em um grafo.
%   c - o numero de componentes conexas
%   theComponets - as arestas que compõe cada componente conexa.

if nargin < 2, threshold = 1; end;

size = length(A);

 %% seta p/ branco
 flag = repmat(0, 1, size);

 %% Conta quantidade de componentes
 c = 0;

 %%Os componentes da componente :)
 theComponets = [];

 for s=1:size %% s é o nodo
     if flag(s) == 0 %% branco
         Q = [s];
         pos = 1;
         flag(s) = 1 ;
         while  pos <= length(Q)
             u = Q(pos); %% retira da frente
             %% pegas os vizinhos
             V = find( A(u,:) );
             size = length(V);
             %% para cada vertice adjacente
             for i=1:size
                 if flag( V(i) ) == 0 %% se jah não passou
                     Q = [Q, V(i)];
                     flag( V(i) ) = 1; %% marca
                 end;
             end;
             pos = pos + 1;
         end;
         if (length(Q) >= threshold) && (length(V) > 0)
           c = c + 1;
           theComponets = [theComponets, {Q}];
         end;
     end;
 end;
```

```
function indicesRegistros = neuronio2indiceRegistro(bmus, neuronios)
%bmus - numero do neuronio vencedor para cada dado do espaco de
%entrada [nx1], n é o numero de exemplares do espaco de entrada.
%neuronios - grupos de neuronios. [1xg], g é tamanho do grupo de
neurônios

indicesRegistros = [];

for i=1:length(neuronios)
    idxs = find(bmus == neuronios(i));
    indicesRegistros = [indicesRegistros; idxs];
```

```
end;

indicesRegistros =  sort(indicesRegistros);
```

```
function im = neuronios2imagem(sM, neuronios)

%% obtem as dimensoes do mapa
dims = sM.topol.msize;

im = zeros(dims(1), dims(2));

for i=1:length(neuronios)
    n = neuronios{i};
    pos = neuronio2pos(n, dims(1), dims(2));

    for k=1:length(pos)
        x = pos(1,k);
        y = pos(2,k);
        % linhas(y) x colunas (x)
        im(y, x) = i;
    end
end
```

Referências bibliográficas

BUSSAB, Wilton de Oliveira; MIAZAKI, Édina Shizue; ANDRADE, Dalton Francisco de. In-**Introdução a análise de agrupamento**. In: SIMPÓSIO NACIONAL DE PROBABILIDADE E ESTATÍSTICA, 9., São Paulo, 1990. Anais. São Paulo: IME/USP, 1990. 105p.

CARVALHO, Luís Alfredo Vidal de Carvalho. **Datamining: a mineração de dados no marketing, medicina, economia, engenharia e administração**. 8. ed. São Paulo: Érica, 2001. 234 p.

COSTA, J. A. F. **Classificação automática e análise de dados por redes neurais auto-organizáveis**. São Paulo. Tese (Doutorado) – Faculdade de Engenharia Elétrica e de Computação, UNICAMP, 1999.

COSTA, J.A.F., e NETTO, M.L.A., "**Segmentação do SOM Baseada em Particionamento de Grafos**". In: *Proc.VI Brazilian Conf. on neural networks*, São Paulo, pp. 451-456, 2003.

FILHO, Ogê Marques; NETO, Hugo Vieira. **Processamento digital de imagens**. Rio de Janeiro: Brasport, 1999. 409 p.

FCPS**, The Fundamental Clustering Problems Suite.** Disponível em http://www.uni-marburg.de/fb12/datenbionik/data?set_language=en. Acesso em: 15 de outubro de 2007.

GABRIEL, Marta Cristina Arouck Ferreira. **Análise da utilização de redes de Kohonen no auxílio ao diagnóstico de doenças reumatológicas**. Belém. Dissertação (Mestrado) – Universidade Federal de Santa Catarina, Belém, 2002.

HAN, Jiawei; KAMBER, Micheline. **Data mining: Concepts and Techniques**. San Diego: Acadelic Press, 2001.

HAYKIN, **Simon. Redes Neurais: princípios e prática**. 2. ed. Porto Alegre: Bookman, 2001. 893 p.

KLAVA, Bruno. **Ferramenta interativa para segmentação de imagens digitais**. 2006. 36 f. Monografia (Trabalho de Formatura Supervisionado) – Instituto de Matemática e Estatística, Universidade de São Paulo, 2006.

KOHONEN, Teuvo. **Self-organizing maps**. 3.ed. Berlim: Springer, 2001. 501 p.

LAGUS, Krista, **Generalizability of the WEBSOM method to document collections of various types.** In *Proc. of 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98)*, Verlag Mainz, Aachen, Germany, volume 1, pp 210-214, 1998.

PRASS, Fernando Sarturi. **Estudo comparativo entre algoritmos de análise de agrupamentos em data mining**. 2004. 71 f. Dissertação (Mestrado) - Ciências da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2004.

RABUSKE, Marcia Aguiar. **Introdução a teoria dos grafos**. Florianópolis: Ed. Da UFSC, 1992. 173p.

SILVA, Marcos Aurélio Santos da. **Mapas Auto-Organizáveis na análise exploratória de dados Geoespaciais multivariados**. 2004. 120 f. Dissertação (Mestrado) - Inpe, São José dos Campos, 2004.

VESANTO, Juha *et al*. **SOM Toolbox for Matlab 5**. A57 Finland: Libella Oy Espoo, 2000. 60 p.

WEBSOM, **A novel SOM-based approach to free-text mining**. Disponível em http://websom.hut.fi/websom/. Acesso em: 09 de outubro de 2007.