Fernando Luis Bordignon

Técnicas Inteligentes para Análise de Agrupamento de Dados

Florianópolis

Julho 2010

Fernando Luis Bordignon

Técnicas Inteligentes para Análise de Agrupamento de Dados

Monografia submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Bacharel em Ciências da Computação.

Orientador: Prof. Dr. Mauro Roisenberg

Universidade Federal de Santa Catarina
Centro Tecnológico
Departamento de Informática e Estatística

Florianópolis

Julho 2010

Monografia de graduação sob o título *"Técnicas Inteligentes para Análise de Agrupamento de Dados"*, defendida por Fernando Luis Bordignon e aprovada em 16 de julho de 2010, em Florianópolis, Santa Catarina, pela banca examinadora constituída pelos professores:

Prof. Dr. Mauro Roisenberg
Universidade Federal de Santa Catarina
Orientador

Prof. Dr. Pedro Alberto Barbetta
Universidade Federal de Santa Catarina
Membro da Banca

Prof. Dr. Ricardo Azambuja Silveira
Universidade Federal de Santa Catarina
Membro da Banca

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

Com o constante aumento da capacidade de armazenamento e obtenção em grande quantidade de dados brutos, faz-se necessário o estudo de métodos para extrair informações e classificar esses conjuntos de dados. Agrupamento de dados é um tema recente e amplamente discutido, com aplicações diversas em variadas áreas da Computação e outras ciências. Existem vários tipos de algoritmos para este propósito, dentre eles os que utilizam ferramentas de Inteligência Artificial se mostram interessantes pois conseguem extrair informações das relações entre os dados. O objetivo do presente estudo centra-se em efetuar uma pesquisa bibliográfica dos principais métodos para agrupamento de dados e de alguns que utilizam IA, além da implementação e comprovação do algoritmo U*C, o qual adiciona a utilização de informações de densidade dos dados na tentativa de obter resultados mais significativos. Testes com *Data-Sets* consagrados da área são apresentados, é também discutida uma forma alternativa de definir o raio utilizado para o cálculo da densidade.

Palavras-chave: SOM, Redes de Kohonen, mapas auto-organizáveis, agrupamento de dados, classificação, análise de agrupamentos

Abstract

With the constant growth of the storing capacity and the large scale acquiring of raw data, it is made necessary the study of methods to extract information and classify these data-sets. Data Clustering is a novel topic and broadly discussed, with various applications in several areas of Computer and other sciences. There are a lot of algorithms conceived to this end, among them the ones that uses Artificial Intelligence tools show themselves interesting because they are able to extract information from the relationship of the data. The goal of this work aims at a bibliographic research of the main methods for data clustering and some others that uses AI, also at an implementation and verification of the U*C algorithm, which adds the use of data density information attempting to reach more significative results. Tests with well-known data-sets from the area are presented, as well an alternative form to define the radius of the density calculation is discussed.

Keywords: SOM, Kohonen maps, self organizing maps, clustering, classification, cluster analysis

1 Introdução

Dentre as várias definições de Inteligência Artificial (IA) disponíveis na literatura como em Luger (2004, p. 48-49) e em Russel e Norvig (2004, p. 5), a que parece mais apropriada a este trabalho é aquela que diz que Inteligência artificial é o ramo da Ciência da Computação que estuda as estratégias para solução de problemas complexos utilizando conhecimento. Problemas esses que exigem uma capacidade computacional muito grande, ou inexistente hoje em dia, para serem resolvidos em um tempo razoável. Muitos deles são triviais para seres humanos (BISHOP, 1996, p. 1), mas quando formalizados utilizando um modelo computacional são caracterizados como intratáveis, ou até mesmo o modelo necessário para representar tal problema seria extremamente grande (RUSSEL; NORVIG, 2004, p. 10-11).

Análise de Agrupamentos de dados é um tema que desperta interesse em pesquisadores de IA pois além de ser um campo amplamente explorado e ser empregado em diversas aplicações práticas, possui várias das características mencionadas acima. A ideia principal do problema consiste em encontrar dentro de um aglomerado de dados, subconjuntos contendo características semelhantes entre seus elementos e ao mesmo tempo possuindo propriedades que os diferenciem de outros grupos de dados (KAUFMAN; ROUSSEEUW, 1990, p. 1).

Cluster analysis, como também é chamado este tema, vem ganhando popularidade pela atual disponibilidade de enormes conjuntos de dados obtidos em áreas como a bancária, hoteleira e comércio eletrônico, dados esses que contêm informações importantes que muitas vezes estão mascaradas na grande massa (COSTA, 1999). Para encontrar essas informações úteis, nas últimas décadas um novo tema de pesquisa vem sendo desenvolvido: a Descoberta de Conhecimento, que aborda justamente o processo de extração do conhecimento de grandes repositórios.

Parte do esforço desse processo é a utilização de algum algoritmo de análise de agrupamentos. A aplicação desses algoritmos é de grande importância pois possibilita encontrar

grupos menores e mais homogêneos de dados, uma vez que a dimensionalidade, complexidade ou quantidade dos registros de um banco de dados, por exemplo, pode ser proibitiva para simples observação humana. Por esses motivos, técnicas que eram conhecidas estritamente em ambiente acadêmico, agora estão sendo utilizadas também na indústria e no setor de serviços (VESANTO, 1999).

Para atacar o problema de análise de agrupamentos surgiram várias propostas interessantes. Algumas delas estão sumarizadas em Xu e Wunsch (2005), cada uma possui suas particularidades e são utilizadas em casos específicos, pois cada algoritmo faz suposições a respeito dos dados de entrada, necessitando assim a escolha da técnica correta para cada caso específico. A escolha errônea de um tipo de algoritmo pode levar a resultados incorretos ou sem sentido (PRASS, 2004).

O uso de ferramentas da área de Inteligência Artificial é uma abordagem amplamente utilizada para tentar solucionar o problema. Podemos verificar em Du (2010) os principais métodos baseados em aprendizado competitivo. Existem vários motivos para o uso de técnicas inteligentes, destacando-se para esse trabalho temos que somente os métodos competitivos conseguem criar categorias a partir de dados crus (*raw data*) automaticamente, ou seja, as informações obtidas vêm dos relacionamentos presentes nos dados e não de regras heurísticas ou programação lógica (KOHONEN, 2001, p. 82).

Uma das principais ferramentas de IA utilizadas para o fim em questão é a Rede Neural Artificial conhecida como SOM do inglês *self-organizing maps* ou mapas auto-organizáveis, que por si só não consiste em um algoritmo de agrupamento de dados, alguns autores preferem classificar a SOM como um meio de visualizar a estrutura dos dados (PAL; BEZDEK; TSAO, 1993). A aplicação de algum algoritmo de agrupamento é realizada sobre o resultado do treinamento de um mapa auto-organizável.

O mapa auto-organizável é caracterizado pela capacidade de mapear dados de alta dimensionalidade para dimensões reduzidas, tipicamente 2D. Através de seu aprendizado competitivo forma-se um mapa topológico dos vetores de entrada, onde os pesos dos neurônios são indicativos de propriedades estatísticas intrínsecas contidas nos padrões de entrada (HAYKIN, 2009, p. 454). A redução de dimensionalidade ocorre pois os neurônios, ou vetores de referência, estão organizados em uma grade geralmente bidimensional.

1.1 Escopo deste trabalho

Apesar de existirem variadas técnicas para agrupar dados, este trabalho apresenta uma pesquisa bibliográfica referente aos mais conhecidos tipos de algoritmos que utilizam técnicas de IA, principalmente por serem tema de pesquisa mais recente (DU, 2010). No escopo destes trabalhos, o algoritmo U*C (ULTSCH, 2005) mostrou-se interessante pois os resultados demonstrados pelo autor foram promissores conforme a Tabela 1, onde a taxa de acerto é calculada levando em consideração quantos exemplos de dados cada algoritmo classificou corretamente em sua respectiva classe, utilizando o próprio conjunto de treinamento.

Tabela 1: Resultados do algoritmo U*C segundo o autor do método

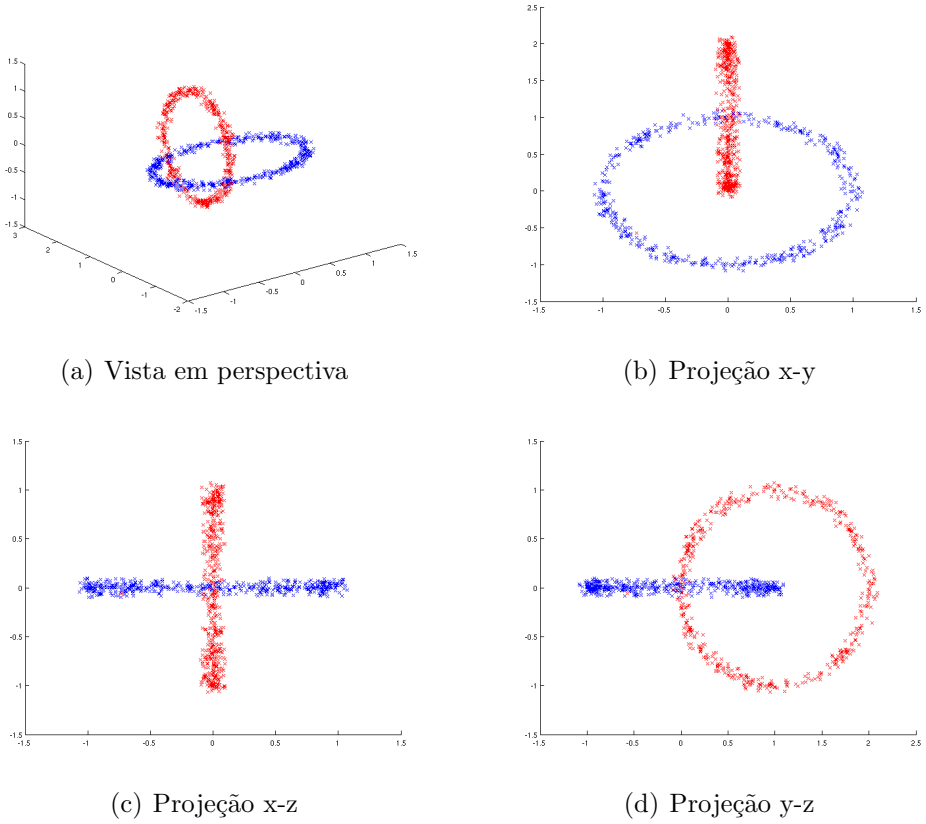
	Algoritmo			
Conjunto de Dados	**SingleLinkage**	**Ward**	**K-médias**	**U*C *Clustering***
Hepta	100%	100%	100%	100%
Lsun	100%	50%	50%	100%
Tetra	0.01%	90%	100%	100%
Chainlink	100%	50%	50%	100%
Atom	100%	50%	50%	100%
EngyTime	0%	90%	90%	90%
Target	100%	25%	25%	100%
TwoDiamonds	0%	100%	100%	100%
WingNut	0%	80%	80%	100%
GolfBall	100%	50%	*	100%

Fonte: Adaptada de Ultsch (2005)

Tendo em vista a subjetividade do problema de agrupamento de dados, encontrar uma métrica para analisar a qualidade dos resultados obtidos depende de cada caso específico. Ainda não se chegou a um resultado unânime com respeito ao melhor indicador de qualidade. Greene, Cunningham e Mayer (2008) apresenta algumas técnicas utilizadas para validar agrupamentos. Apesar de existirem essas métricas, é mais objetiva e didática a comparação com exemplos preparados para colocar a prova os algoritmos de agrupamento.

Deste modo, *Data-Sets* conhecidos e que possuem classes de dados definidas *a priori* são submetidos aos modelos a serem validados, esses conjuntos de dados têm dimensões reduzidas (até 3D) para facilitar a visualização dos parâmetros dos modelos, além disso fica mais natural a validação do treinamento e detecção de *bugs*. Dentre esses dados existe o conjunto clássico conhecido como *chainlink*, que pertence ao conjunto de *Data-Sets* FCPS (ULTSCH, 2005), apresentado na Figura 1 com seus dois agrupamentos representados por

cores diferentes. Essa coleção de conjuntos de dados, elaborados pelo mesmo autor do método U*C, foi utilizada para gerar a Tabela 1.



(a) Vista em perspectiva

(b) Projeção x-y

(c) Projeção x-z

(d) Projeção y-z

Figura 1: Conjunto de dados Chainlink.

1.2 Objetivos

1.2.1 Geral

O objetivo principal do presente trabalho é investigar através de pesquisa bibliográfica os métodos mais conhecidos para agrupamento de dados utilizando técnicas inteligentes, i.e. técnicas que fazem uso de métodos da área de IA. Além disso o algoritmo U*C foi escolhido para ser implementado e submetido a experimentos a fim de compreender o embasamento teórico, discutir a performance para cada caso e apontar possíveis melhoramentos em certos pontos do método.

1.2.2 Específicos

Para tanto foram traçados os seguintes objetivos específicos:

- Pesquisa bibliográfica dos métodos mais conhecidos apresentados em Du (2010).
- Apontamento das vantagens e desvantagens de cada um.
- Implementar utilizando linguagem C++ o treinamento simples da rede de Kohonen.
- Implementar o método de agrupamento U*C.
- Utilizar dados do *data-set* FCPS para realizar experimentos e comparações com a técnica clássica de K-Médias.
- Comparar e analisar os resultados obtidos com os resultados apresentados pelo autor.
- Analisar os procedimentos críticos a fim de sugerir melhoras.

1.3 Estrutura

Neste capítulo foi apresentada uma breve introdução ao tema, bem como a motivação, escopo e objetivos do presente trabalho. No capítulo 2 será apresentada uma revisão bibliográfica dos mais conhecidos métodos inteligentes de agrupamento de dados, mais atenção será dada ao método U*C e o uso de SOMs para agrupamento de dados.

Em seguida, no capítulo 3, é apresentada a metodologia utilizada para implementar e executar os experimentos com o método U*C. Continuando no capítulo 4 temos os resultados e comparação da técnica com os resultados do autor e com a técnica tradicional de K-médias, onde também será apresentada uma discussão dos resultados. Finalmente no capítulo 5 são apresentadas as conclusões do trabalho e perspectivas para trabalhos futuros.

2 Agrupamento de dados

Segundo Xu e Wunsch (2005), na literatura existem várias definições para os chamados grupo de dados (*clusters*) e como o conceito ainda é bastante intuitivo não há consenso, mas temos uma definição aceita por grande parde dos pesquisadores, que descrevem um grupo de dados levando em consideração uma alta similaridade, ou homogeneidade, entre os elementos de um grupo e ao mesmo tempo exibe um certo grau de separação em relação a outros grupos de dados.

Podemos também encontrar definições matemáticas para a estrutura dos tipos de grupos encontrados (HANSEN; JAUMARD, 1997):

(i) Subconjunto C de O onde O é um conjunto de dados com N amostras e p dimensões;

(ii) Partição $P_M = \{C_1, C_2, \ldots, C_M\}$ de O em M grupos;

  (a) $C_j \neq \emptyset \quad j = 1, 2, \ldots, M$;

  (b) $C_i \cap C_j = \emptyset \quad i, j = 1, 2, \ldots, M$ e $i \neq j$;

  (c) $\bigcup_{i=1}^{M} C_j = O$;

(iii) Empacotamento (*Packing*) $Pa_M = \{C_1, C_2, \ldots, C_M\}$ de O com M grupos:

  como em (ii) mas sem a condição (c);

(iv) Cobertura (*Covering*) $Co_M = \{C_1, C_2, \ldots, C_M\}$ de O por M grupos:

  como em (ii) mas sem a condição (b);

(v) Hierarquia $H = \{P_1, P_2, \ldots, P_q\}$ de $q \leq N$ partições de O.

  Conjunto de Partições $P_1, P_2, \ldots, P_q$ de O tal que $C_i \in P_k, C_j \in P_l$ e $k > l$ implica em $C_i \subset C_j$ ou $C_i \cap C_j = \emptyset$ para todos os $i, j \neq i, k, l = 1, 2, \ldots, q$.

Ainda segundo Hansen e Jaumard (1997), os tipos mais utilizados de grupos são a partição e a hierarquia. O processo de agrupamento de dados é então definido como os

passos executados para encontrar um conjunto de algum tipo de grupo, dado um conjunto de dados e nenhuma, ou pouca, informação sobre os grupos contidos nele. Outro conceito importante é o de classificação, pois a partir de um esquema de agrupamento existe a necessidade de atribuir os exemplos dados aos grupos encontrados, ou posteriormente quando surgem novos exemplos, executar essa mesma atribuição.

Existem várias técnicas e algoritmos para agrupamento de dados, Greene, Cunningham e Mayer (2008) coloca as técnicas por K-médias e Hierárquicas como as clássicas para este fim. Cada uma possui suas peculiaridades e são aplicados em casos diferentes, além de possuírem complexidades distintas como apresentado em Xu e Wunsch (2005). Nas próximas seções será apresentada a ideia geral das técnicas clássicas, além de outras técnicas importantes que fazem uso de ferramentas de Inteligência Artificial. Aprofundando-se no algoritmo U*C (ULTSCH, 2005) baseado em Mapas Auto-organizáveis.

2.1 Aprendizado Competitivo

O aprendizado competitivo serve de base para várias técnicas baseadas em aprendizado não supervisionado. A ideia principal do aprendizado supervisionado é a existência de um professor que guia o processo de aprendizado fornecendo rótulos para os exemplos de entrada, já quando não temos esse guia fica caracterizado o aprendizado não supervisionado. No segundo caso, geralmente o que se faz é agrupar ou organizar os dados de alguma maneira (GREENE; CUNNINGHAM; MAYER, 2008).

Podemos implementar o aprendizado competitivo utilizando uma Rede Neural Artificial de duas camadas completamente conectadas, onde a camada de saída é chamada de camada competitiva e as conexões laterais entre os neurônios são utilizadas para inibição (DU, 2010). A arquitetura geral de uma rede deste tipo é mostrada na Figura 2.

A definição original de uma rede competitiva pode ser encontrada em Haykin (1998, p. 80-82), mas na prática no contexto do problema de análise de agrupamentos o treinamento é realizado minimizando o erro médio quadrático (*MSE*) mostrado nas equações 2.1 e 2.2 adaptadas de Du (2010):

$$E = \frac{1}{N} \sum_{i=1}^{N} E_i \tag{2.1}$$

$$E_i = \sum_{k=1}^{K} \mu_{ki} \left\| x_i - w_k \right\|^2 \tag{2.2}$$

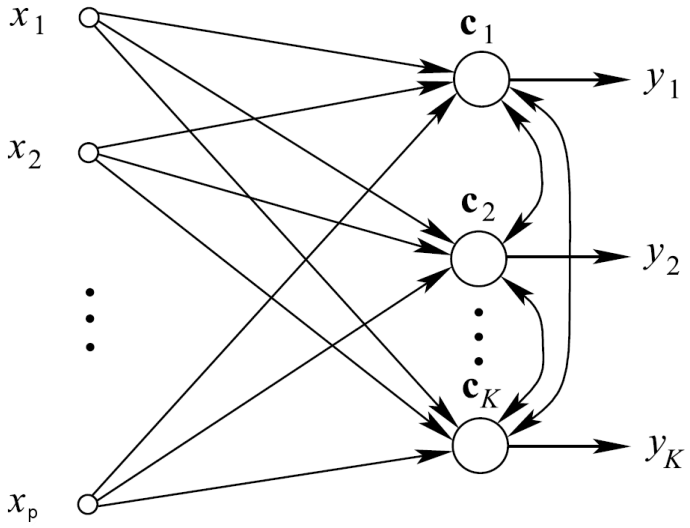


Figura 2: Arquitetura de uma Rede Neural Competitiva
Fonte: Adaptada de Du (2010)

Na Figura 2, w_{ki} representa o peso entre o neurônio c_k em relação a um exemplo x_i do conjunto de dados de entrada, ou o centroide do neurônio k, quando a unidade c_k é a mais próxima do exemplo x_i, $\mu_{ki} = 1$ senão $\mu_{ki} = 0$. Com isso define-se o neurônio vencedor calculando as minímas distâncias entre os dados de entrada e todas as unidades da camada de saída, representado por c_v e seu peso por w_v. Então o treinamento se resume a:

$$w_v(t+1) = w_v(t) + \eta(t)[x_t - w_v(t)] \tag{2.3}$$

$$w_i(t+1) = w_i(t),\ i \neq v \tag{2.4}$$

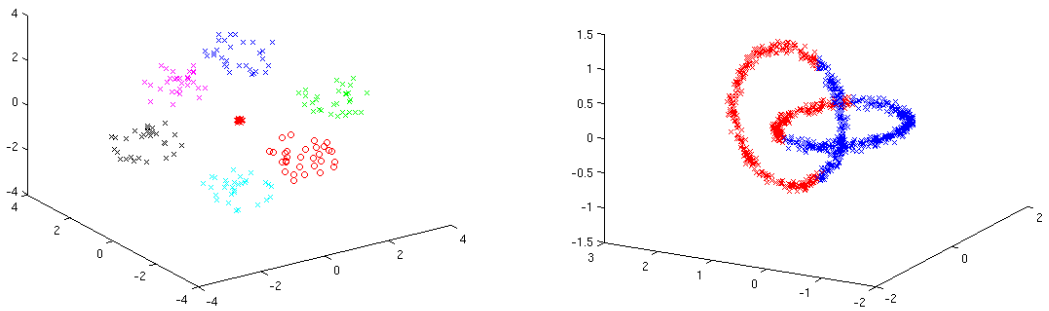
Onde η é a taxa de aprendizagem, que tem o efeito geral de mover o neurônio vencedor em direção ao padrão de entrada (HAYKIN, 1998, p. 81).

2.2 K-médias

Também chamado de *C-Means* (DU, 2010), esse método possui alta aceitação pois ele é um dos mais simples e rápidos (XU; WUNSCH, 2005). O algoritmo mais utilizado consiste em dividir o conjunto de dados em K partições, definindo de forma aleatória inicialmente K centroides dentro dos limites dos dados de entrada. Depois disso, dois passos principais são tomados a cada iteração, o primeiro é atribuir os exemplos de dados

aos seus respectivos centroides utilizando o critério de menor distância e, em seguida, cada centroide é recalculado para os objetos que pertencem ao seu agrupamento, ou seja, os que estão mais próximos dos centroides. Esses últimos procedimentos são repetidos até que um critério de parada seja atingido, o que geralmente é quando o algoritmo não provoca mais nenhuma mudança nos agrupamentos (GREENE; CUNNINGHAM; MAYER, 2008).

Para dados mais simples este algoritmo é uma boa escolha, mas quando a separação dos grupos não é realizável linearmente, é preciso escolher outra técnica que suprime essa deficiência, veja na figura 3 a aplicação do algoritmo K-médias, onde cada cor representa um grupo encontrado, a dois conjuntos de dados sintéticos elaborados por Ultsch (2005) demonstrando essa insuficiência da técnica.



(a) Grupos separáveis linearmente

(b) Grupos não separáveis linearmente

Figura 3: Aplicação do algoritmo K-médias

Um outro ponto negativo a se considerar é que o processo é aleatório e dependendo das partições iniciais selecionadas é possível obter resultados diferentes. Na figura 3, por exemplo, foram executadas 20 vezes o algoritmo para cada conjunto de dados, e foi considerado como correto o resultado que mais se repetiu.

2.3 Agrupamento Nebuloso

Na técnica K-médias os grupos resultantes são caracterizados por serem partições, já para o agrupamento *Fuzzy* (Nebuloso) temos que cada objeto de entrada possui um grau de pertinência a cada grupo ou, seguindo a teoria de lógica nebulosa, os conjuntos utilizados para formar os grupos são conjuntos nebulosos. Essa relação entre dados de entradas e conjuntos é formalizada definindo que um conjunto nebuloso $\tilde{A}$ é formado a partir de tuplas (ZIMMERMANN, 1992, p. 12):

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | \, x \in O\} \tag{2.5}$$

Onde O é o conjunto de dados, como definido no inicio do capítulo, e $\mu_{\tilde{A}}(x)$ é a função de pertinência (ou grau de verdade), é através dessa função que o mapeamento entre os dados de entrada e os conjuntos nebulosos é feito. Com isso o resultado acaba se tornando um tipo de Cobertura.

O uso de lógica nebulosa no problema de agrupamento é uma prática bastante utilizada pois ajuda a definir os limites naturais vagos entre os grupos de dados (DU, 2010), isso não significa falta de informação ou imprecisão nos exemplos de entradas, e sim que realmente não se pode escolher com exatidão somente um único grupo para atribuir certas amostras, o que ocorre frequentemente em dados de aplicações reais (ZIMMERMANN, 1992, p. 5-6).

Na prática um dos algoritmos mais utilizados é o FCM (*Fuzzy C-Means*), que utiliza os princípios definidos acima, otimizando a função de erro (BEZDEK†, 1973 apud DU, 2010):

$$E = \sum_{j=1}^{K} \sum_{i=1}^{N} (\mu_{ij})^m \left\| x_i - c_j \right\|^2 \tag{2.6}$$

Note que pela equação 2.6 o número de grupos (K) é definido *a priori*, outro parâmetro dessa função é o m, que representa o grau de nebulosidade dos conjuntos formados, onde $m \in (1, \infty)$. Quando $m \to 1+$ o modelo se aproxima de uma Partição, já quando $m \to \infty$ os conjuntos gerados se aproximam de conjuntos fuzzy máximos. Usualmente são utilizados valores entre 1.5 e 2 para esse parâmetro (DU, 2010). Na mesma equação, a função de pertinência é representada por uma matriz ($\mathbf{U}$) com a restrição de que $\sum_{j=1}^{K} \mu_{ij} = 1, \; i = 1, \ldots, N$.

O algoritmo FCM foi chamado assim por sua relação próxima ao K-médias, o último é uma especialização do FCM, quando μ_{ij} tem o valor 1 para somente uma coluna de cada linha e zero para as demais posições da matriz, além de $m = 1$. Por essa razão, a técnica com uso de lógica nebulosa também possui as deficiências de: depender da inicialização dos parâmetros, nesse caso da matriz $\mathbf{U}$ e dos centros $\mathbf{C}$, e poder ficar preso em mínimos locais (DU, 2010). A seguir é apresentado um resumo dos passos do algoritmo FCM adaptado de Xu e Wunsch (2005):

(1) Selecionar valores apropriados para m e K, atribuir um valor positivo e pequeno para

ε. Inicializar a matriz de centros $\mathbf{C}$ aleatoriamente.

Atribuir a variável de passo $t = 0$.

(2) Calcular $(t = 0)$ ou atualizar $(t > 0)$ a Matriz de pertinência $\mathbf{U}$ utilizando:

$$\mu_{ij}^{(t+1)} = 1 \Bigg/ \left(\sum_{l=1}^{K} \left(D_{lj}/D_{ij} \right)^{1/(1-m)} \right)$$
$$para \qquad i = 1, \ldots, K \; e \; j = 1, \ldots, N$$

(3) Atualizar a Matriz de centros $\mathbf{C}$ utilizando:

$$c_i^{(t+1)} = \left(\sum_{j=1}^{N} \left(\mu_{ij}^{(t+1)} \right)^m x_j \right) \Bigg/ \left(\sum_{j=1}^{N} \left(\mu_{ij}^{(t+1)} \right)^m \right)$$
$$para \qquad i = 1, \ldots, K$$

(4) Repetir passos 2 e 3 até que $\|\mathbf{C}^{t+1} - \mathbf{C}^t\| < \varepsilon$

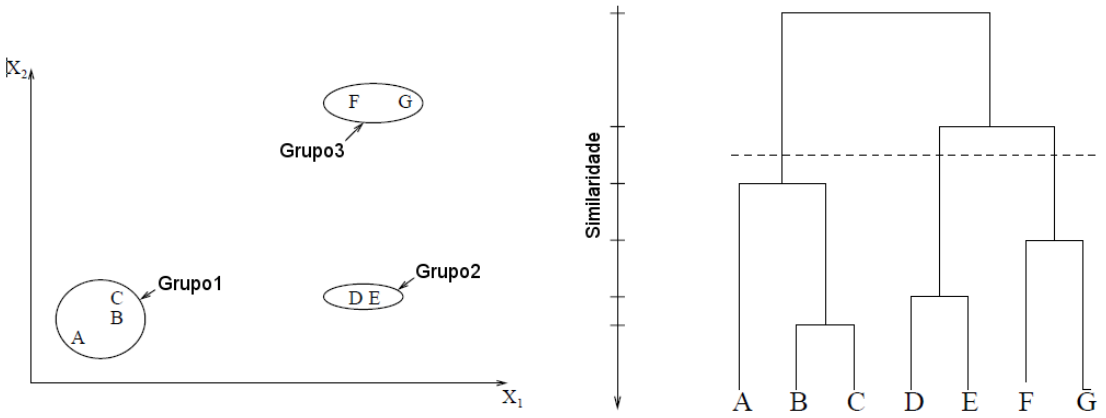
Onde $D_{ij} = D(x_j, c_i)$ é a função distância escolhida, que na equação 2.6 aparece como a distância euclidiana representada pela sua fórmula: $\|x_j - c_i\|^2$.

2.4 Agrupamento Hierárquico

Os algoritmos hierárquicos são caracterizados por construírem, como resultado, uma estrutura em forma de árvore denominada dendograma (JAIN; MURTY; FLYNN, 1999). Essa estrutura representa uma Hierarquia, como definida na introdução do capítulo, de forma gráfica, onde cada linha vertical representa um agrupamento e cada linha horizontal representa uma operação realizada para alterar a composição das Partições.

Existem duas maneiras principais de executar o procedimento hierárquico: construindo um dendograma de maneira *top-down* ou *bottom-up*. A primeira é denominada divisiva, pois inicialmente todo o conjunto de dados é considerado como se fosse um único grupo, e ações são tomadas a cada passo afim de dividir os grupos. Já no segundo modo, cada amostra de dado é considerada como um grupo e são executados procedimentos para aglomerar dois grupos distintos, por isso é chamada de aglomerativa. O segundo tipo é o mais utilizado na prática (XU; WUNSCH, 2005). Um exemplo de dendograma e os dados que o geraram são apresentados na Figura 4.

Dependendo do tipo de algoritmo escolhido temos um dendograma diferente, o resultado final do processo de agrupamento é um corte do dendograma definido por alguma



(a) Exemplos simples de dados (b) Dendograma para o exemplo ao lado

Figura 4: Exemplo de dendograma e seus dados de entrada
Fonte: Adaptado de Jain, Murty e Flynn (1999)

métrica específica do algoritmo utilizado, representado por uma linha pontilhada na Figura 4(b). Este é um dos pontos críticos dos algoritmos hierárquicos, pois definir esta linha consiste no problema de avaliação do agrupamento, i.e. verificar se um certo conjunto de grupos é o melhor dado um conjunto de dados de entrada. Jung et al. (2003) discute esse problema e fornece métricas para determinar o número ótimo de grupos. Outra abordagem é escolher anteriormente o número de grupos desejados.

Os passos gerais dos algoritmos de agrupamento hierárquico aglomerativo são apresentados a seguir (XU; WUNSCH, 2005):

1) Iniciar com N grupos de dados, um para cada amostra da entrada.

2) Calcular e encontrar a distância mínima entre dois grupos

$$D(C_i, C_j) = \min_{\substack{1 \leq m\,,\, l \leq N \\ m \neq l}} D(C_m, C_l)$$

Onde $D(*, *)$ é a função distância que será discutida posteriormente.

3) Combinar C_i e C_j formando um novo grupo.

4) Repetir os passos 2) e 3) até todos os objetos estarem num mesmo grupo ou algum critério de parada ser atingido.

Como o conceito de distância entre grupos de dados não tem uma definição muito precisa, foi proposta uma generalização da formula da função distância, parametrizando-

a para se adaptar aos vários algoritmos propostos (LANCE; WILLIAMS, 1967):

$$D(C_l, (C_i, C_j)) = \alpha_i D(C_l, C_i) + \alpha_j D(C_l, C_j) + \beta D(C_i, C_j) + \gamma |D(C_l, C_i) - D(C_l, C_j)|$$

Em Jain e Dubes (1988, p. 80) é apresentada uma tabela contendo os coeficientes para os algoritmos mais comuns.

Uma vantagem dos algoritmos hierárquicos é que eles podem encontrar grupos que não são separáveis linearmente (JAIN; MURTY; FLYNN, 1999), eles são capazes, por exemplo, de encontrar corretamente os grupos do conjunto *chainlink* apresentado na Figura 1. Como o agrupamento é feito de forma a não utilizar um centro para cada grupo, um problema que aparece é que os grupos encontrados não possuem um vetor representativo, ou seja, uma amostra dos dados de entrada que represente o seu grupo (XU; WUNSCH, 2005).

2.5 Quantização Vetorial por Aprendizagem

Embora a quantização vetorial por aprendizagem, ou LVQ (sigla do seu nome em inglês - *Learning Vector Quantization*), seja um tipo de aprendizado supervisionado, essa técnica utiliza-se de outros métodos para preprocessar os dados e obter centros dos grupos contidos no conjunto de entrada (MELIN; CASTILLO, 2005).

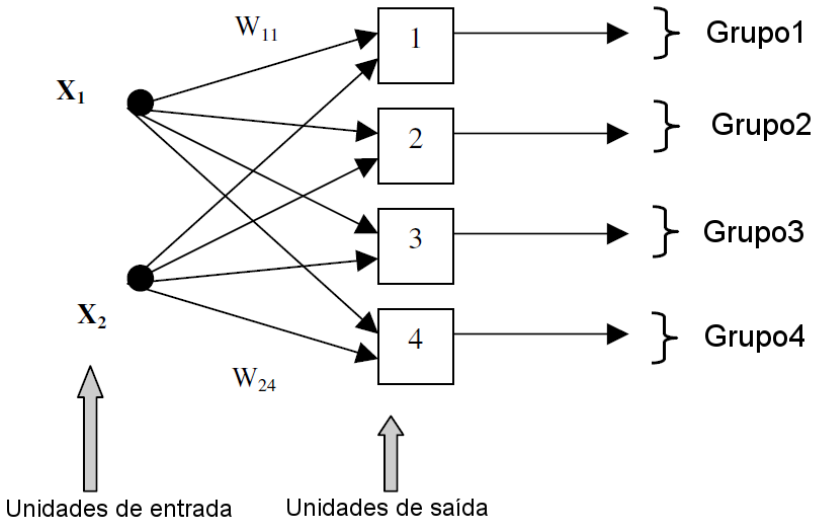


Figura 5: Arquitetura de uma rede LVQ
Fonte: Adaptada de Melin e Castillo (2005, p. 95)

A arquitetura da rede utilizada no LVQ é muito parecida com a do aprendizado competitivo, exceto pelo fato que, na saída da camada competitiva, cada unidade é associada

a um grupo de dados. Veja na figura 5 um exemplo de uma rede para LVQ com dados de entrada bidimensionais e com os neurônios mapeados para quatro grupos. O aprendizado é feito em duas etapas, a etapa não supervisionada que faz uso de algum método de agrupamento de dados, e.g. K-médias, é executada primeiramente para definição do número de neurônios a ser utilizado, além das coordenadas dos pesos de cada um.

Na segunda etapa, os exemplos dos dados de entrada são rotuladas para serem submetidos ao treinamento supervisionado. O objetivo desse treinamento é reduzir os erros de classificação gerados pelas técnicas tradicionais (MELIN; CASTILLO, 2005). Veja na figura 6 um típico erro de classificação gerado após a execução do algoritmo K-médias.

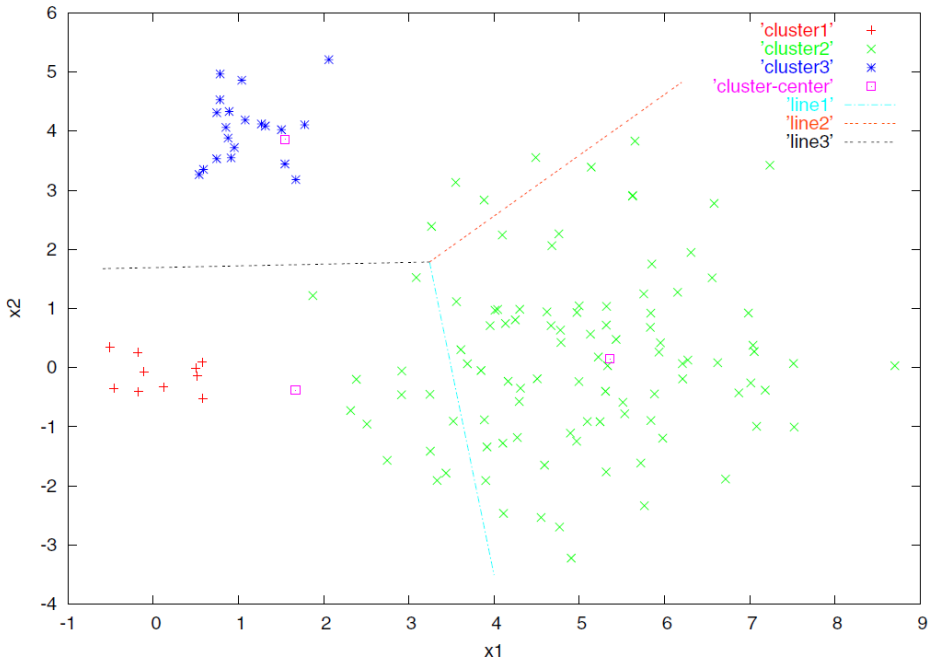


Figura 6: Erro na classificação após execução do K-médias
Fonte: Adaptada de Morii (2008)

O processo de rotulagem pode ser feito com o método de votação (*voting*), que executa várias vezes um algoritmo de agrupamento realizando uma espécie de votação, onde os resultados que mais se repetem são combinados para formar uma classificação melhor dos dados (DIMITRIADOU; WEINGESSEL; HORNIK, 2001).

Após ter os dados rotulados, um treinamento competitivo supervisionado é realizado, considerando os rótulos representados pelo vetor binário y_t, com base nas funções a seguir (DU, 2010):

$$c_w(t+1) = c_w(t) + \eta(k)[x_t - c_w(t)],\ y_{t,w} = 1 \tag{2.7}$$

$$c_w(t+1) = c_w(t) - \eta(k)[x_t - c_w(t)],\ y_{t,w} = 0 \tag{2.8}$$

$$c_i(t+1) = c_i(t), i \neq w \tag{2.9}$$

Onde c_w representa o neurônio vencedor. A principal diferença para um aprendizado competitivo simples é exibida na equação 2.8, onde um centro que não pertence a classe em questão é afastado dos dados daquele grupo. Essa técnica então tem a característica de aprimorar os resultados encontrados com outros métodos, melhorando a classificação em alguns casos, principalmente quando as densidades dos grupos são diferentes (MORII, 2008).

2.6 Agrupamento com Mapas Auto-Organizáveis

O mapa auto-organizável foi inventado por Teuvo Kohonen em 1982 (KOHONEN, 1982), por isso também são chamados de redes de Kohonen, ou ainda pelo seu nome em inglês: *self-organizing maps (SOM's)*. As redes de Kohonen não são algoritmos para agrupamento de dados, e sim um tipo específico de Rede Neural que adquire as propriedades topológicas dos dados apresentados (KOHONEN, 1982), assim, após treinado, o mapa auto-organizável pode ser submetido a técnicas de visualização de dados e de agrupamento de dados.

2.6.1 A estrutura do Mapa Auto-Organizável

Esse tipo de Rede Neural Artificial possui somente uma camada de neurônios geralmente organizados em uma topologia bidimensional. Cada unidade da camada é conectada aos dados de entrada com pesos de mesma dimensionalidade, possuindo assim um parâmetro para cada dimensão da entrada para cada neurônio, na figura 7 pode-se ver uma ilustração da arquitetura da rede.

Um ponto importante na estrutura de uma rede de Kohonen é a sua topologia, que é o modo como as unidades são organizadas umas em relação as outras. Existem dois principais modos de dispor os elementos e são chamados de topologia hexagonal e quadrada, ilustradas na figura 8. A organização dos neurônios no mapa é que define a função vizinhança $N(i)$ de um determinado neurônio pois na hexagonal cada neurônio pode se

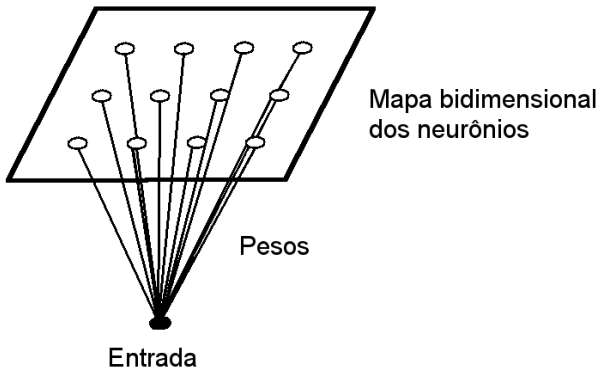
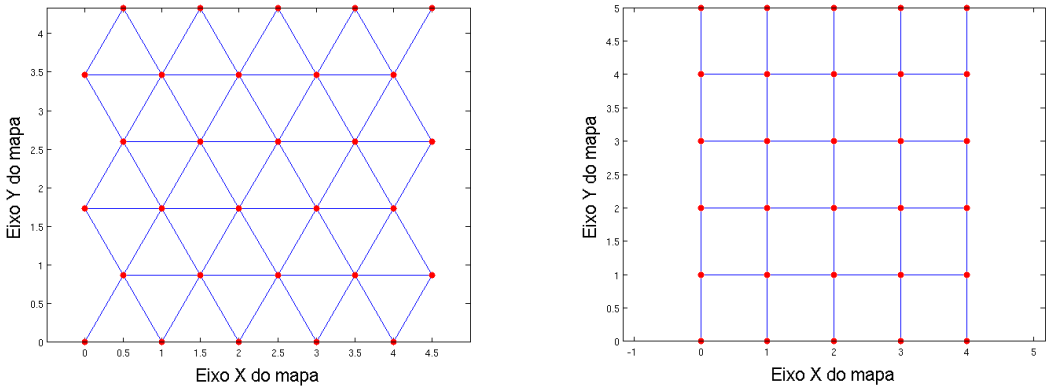


Figura 7: Mapa Auto-organizável
Fonte: Adaptada de Haykin (2009, p. 455)

conectar diretamente com até seis outras unidades e na quadrada com até quatro.



(a) Topologia Hexagonal (b) Topologia Quadrada

Figura 8: Topologias típicas de um *SOM*

2.6.2 Treinamento

O treinamento da rede de Kohonen é divido em duas fases principais, a fase de ordenação e a de ajuste fino. Elas diferem nos valores dos parâmetros de treinamento e número de épocas, mas ambas executam os seguintes passos:

- Processo competitivo
- Processo cooperativo
- Processo adaptativo

O primeiro passo consiste na competição entre os neurônios para descobrir qual está mais próximo do padrão de entrada x sorteado a cada iteração. Para tanto é tomada uma medida de distância qualquer. Em muitas aplicações práticas é utilizada a distância Euclidiana. A medida é calculada entre os pesos w de cada neurônio i e o exemplo em questão, o menor valor obtido elege o neurônio c como o neurônio vencedor, processo sumarizado na equação 2.10 adaptada de Kohonen (2001, p. 110).

$$c = \arg\min_i \{\|x - w_i\|\} \tag{2.10}$$

Essa informação será utilizada no próximo passo, o processo cooperativo. Nessa etapa o neurônio vencedor é considerado como o centro de uma vizinhança gaussiana de raio pré estabelecido, assim é calculada uma proporção de quanto cada unidade do mapa coopera para gerar o padrão de entrada x:

$$h_{ci} = \alpha(t) \cdot \exp\left(-\frac{||r_c - r_i||^2}{2\sigma^2(t)}\right) \tag{2.11}$$

Na equação 2.11 adaptada de Kohonen (2001, p. 111) o $\alpha(t)$ é o parâmetro chamado de *learning rate*, o qual é recomendado que seja em função da época atual de treinamento, e.g. $\alpha(t) = 0.9(1 - t/1000)$. O fator $\sigma(t)$ é um raio definido no início do treinamento que também decai conforme o passar das épocas (t). Estabelecidos esses fatores o próximo passo os utiliza para atualizar os parâmetros de cada neurônio conforme a equação 2.12 também adaptada de Kohonen (2001, p. 111).

$$w_i(t+1) = w_i(t) + h_{ci}(t)\,[x(t) - w_i(t)] \tag{2.12}$$

Ao fim de cada iteração as unidades do mapa mais próximas ao exemplo apresentado têm seus pesos alterados, desta forma o mapa tende a adquirir o formato da organização dos conjunto de dados apresentado. Na figura 9, é ilustrado o processo de atualização dos pesos do *SOM*, onde a unidade destacada em vermelho é a chamada *best matching unit*, ou seja, o neurônio vencedor da etapa competitiva e as unidades destacadas em cinza são o resultado da atualização dos pesos.

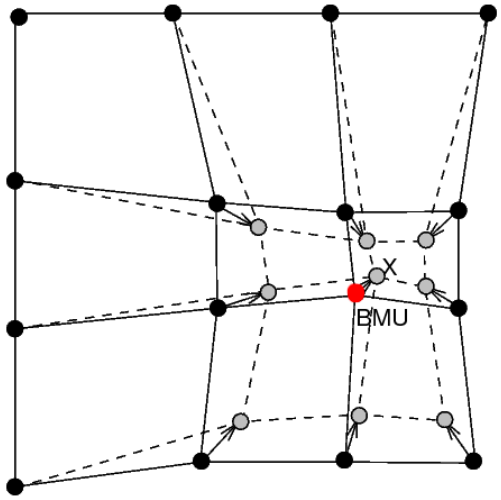


Figura 9: Atualização dos pesos do *SOM*
Fonte: Adaptado de Vesanto et al. (2000)

2.6.3 Matriz-U

Após um treinamento bem sucedido, é possível computar a matriz de distâncias entre os pesos de neurônios vizinhos, que é chamada de Matriz-U. Possibilitando a visualização de dados em dimensões elevadas, pois a topologia da rede é geralmente bidimensional. Algumas técnicas de visualização dos mapas são expostas em Vesanto (1999).

Os métodos desenvolvidos com o propósito de visualização de dados foram inicialmente utilizados para somente este fim, mais recentemente eles têm sido explorados para servir de entrada para algoritmos de agrupamento de dados. Um exemplo é o método exposto em Ultsch (2005) analisado nesse trabalho.

Existem duas maneiras de calcular a Matriz-U, uma delas é a apresentada em Costa (1999) e não será abordada pois não é utilizada para os fins deste trabalho. O outro tipo de cálculo proposto pelo mesmo autor do método a ser analisado é feito simplesmente calculando a média das distâncias do neurônio em questão n_i para seus vizinhos N(i), veja a equação 2.13 (ULTSCH, 2005):

$$uh(i) = \frac{1}{n}\sum_{j} d(w_i, w_j), j \in N(i), n = |N(i)| \tag{2.13}$$

Onde $d(w_1, w_2)$ é a função de distância escolhida para o treinamento da rede.

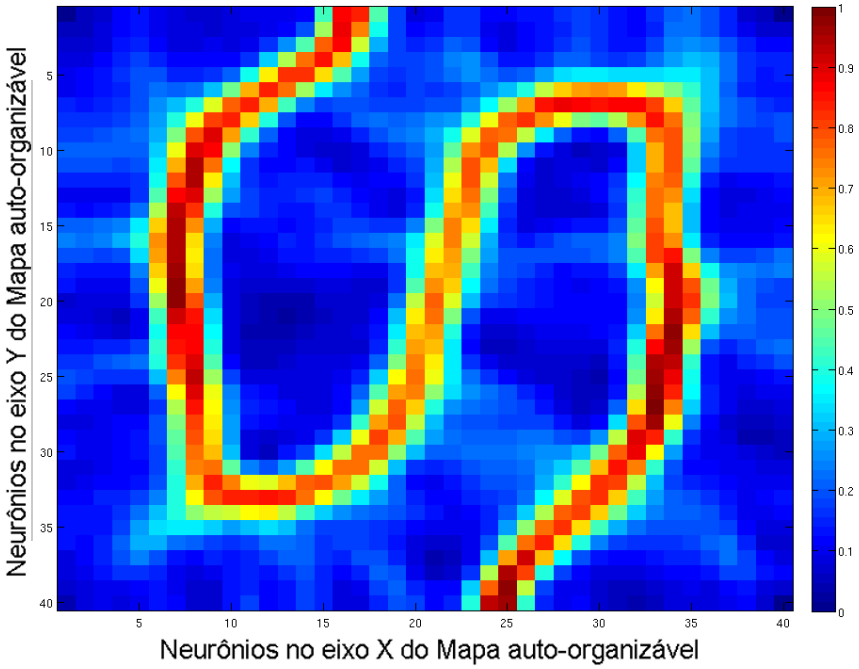


Figura 10: Matriz-U para uma grade de 40x40 neurônios

Na figura 10 é mostrada uma Matriz-U calculada a partir de um mapa de 40x40 neurônios treinados com o conjunto de dados *chainlink* (figura 1). Como o conjunto de treinamento é 3D, é possível a exibição dos pesos da rede, veja figura 11. Na Matriz-U os neurônios entre os agrupamentos de dados aparecem com um valor maior, representado pela cor vermelha. Esses neurônios indicam uma região de separação entre os grupos existentes. Já os neurônios que estão mais próximos uns dos outros, e que estão também mais próximos dos dados de treinamento, são denotados pela cor azul.

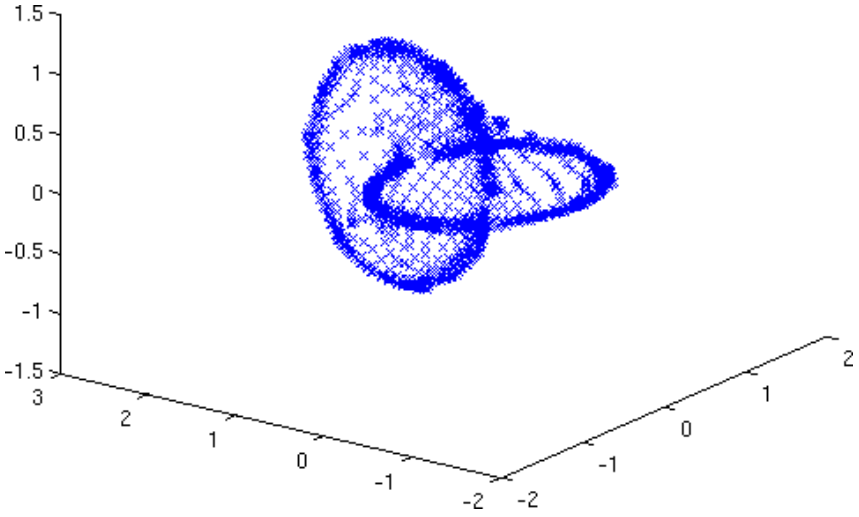


Figura 11: Pesos de um mapa auto-organizável treinado com dados de entrada 3D

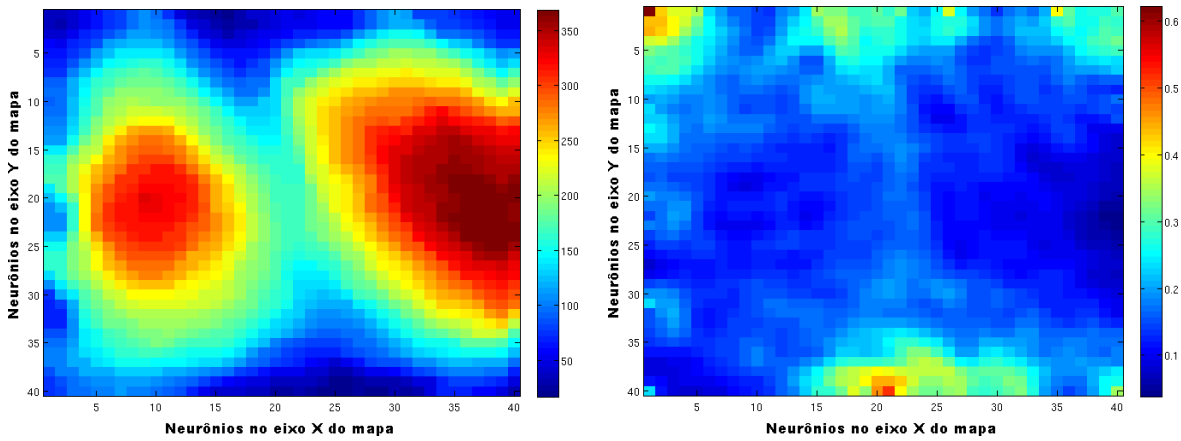
2.6.4 Matriz-P

Segundo Ultsch (2005) a informação obtida com a Matriz-U não é suficiente em alguns casos, a proposta do autor é realizar um cálculo de densidade de cada unidade da rede de Kohonen, ou seja, verificar qual a proporção de amostras dos dados de entrada estão localizadas próximas a uma unidade.

Esse procedimento é realizado definindo um raio em torno das unidades da rede formando hiperesferas centradas nos pesos dos neurônios, em seguida é calculada a distância de cada exemplo do conjunto de treinamento para cada neurônio. Tendo essa informação, o passo final é a contagem das distâncias menores do que o raio utilizado, que na teoria representa verificar quantos exemplos dos dados de entada estão localizados dentro das hiperesferas mencionadas. O cálculo é sumarizado pela equação 2.14 (ULTSCH, 2005):

$$p(i) = |\{x \in O | d(x, w_i) < r > 0,\ r \in R\}| \tag{2.14}$$

A exibição dos pesos $p(i)$ utilizando as coordenadas dos neurônios é chamada de Matriz-P. O problema principal no cálculo dessa matriz é a definição do raio, o autor apresenta uma opção para estipular o raio utilizando o chamado raio de Pareto, o algoritmo para obtê-lo é descrito em (ULTSCH, 2003a). Uma discussão mais detalhada da implementação e resultados obtidos com o raio de Pareto é apresentada no capítulo 3.



(a) Matriz-P para rede treinada com os dados *Engytime* do FCPS

(b) Matriz-U para rede treinada com os dados *Engytime* do FCPS

Figura 12: Comparação entre uma Matriz-U e uma Matriz-P

As informações extras obtidas pelo calculo da Matrix-P são visíveis na figura 12, na Matriz-U a divisão exata dos dois grupos de dados presentes no conjunto não pode ser

extraída tão facilmente como na Matrix-P, pois os valores são próximos na região de separação entre os grupos de dados, já na Matriz-P as regiões que delimitam os grupos tem seu valor aproximadamente igual a metade dos valores de pico que encontram-se no centro dos grupos.

Nesse novo tipo de matriz, as regiões com baixa densidade definem regiões em que os neurônios estão solitários, ou longe dos exemplos de dados, já onde forma-se um platô é caracterizado como centros de grupos, e nas áreas que formam um declive são apontadas as bordas dos grupos de dados.

2.6.5 Matriz-U*

Para utilizar as informações obtidas na Matriz-P, em Ultsch (2005) é proposto um método que combina as duas matrizes calculadas até agora, para posteriormente submeter o resultado ao algoritmo proposto pelo autor. Para tanto, é definido o cálculo de um fator baseado numa função de densidade empírica, que representa a probabiliade da densidade no neurônio n_i ser baixa, representado na equação 2.15 (ULTSCH, 2005):

$$plow(i) \cong \frac{|\{p \in \text{Matriz-P} | p > ph(i)\}|}{|p \in \text{Matriz-P}|} \tag{2.15}$$

Onde $ph(i)$ representa o valor na Matriz-P para o neurônio i. Os valores para preencher a Matriz-U* são calculados então da seguinte maneira:

$$u^*h(i) = uh(i)plow(i) \tag{2.16}$$

Nessa equação o $uh(i)$ é o valor da Matriz-U para o neurônio i. Na prática essa multiplicação representa que o valor do peso $u^*h(i)$ será aproximadamente igual ao próprio $uh(i)$ se a densidade for baixa, já se a densidade for alta o valor final se aproximará de 0.

Na figura 13 é mostrado um exemplo de Matriz-U*, é visível a separação e a melhora da percepção dos grupos de dados em relação a Matriz-U convencional, o cálculo foi feito utilizando as matrizes da figura 12. Uma questão que pode ser levantada é que na figura 12(a) (Matriz-P) a visualização dos grupos de dados pode também ser claramente identificada, pra alguns até mais facilmente. Segundo o autor a justificativa para o uso da Matriz-U* é que, em alguns casos, somente a densidade não consegue identificar os grupos de dados existentes no conjunto de entrada.

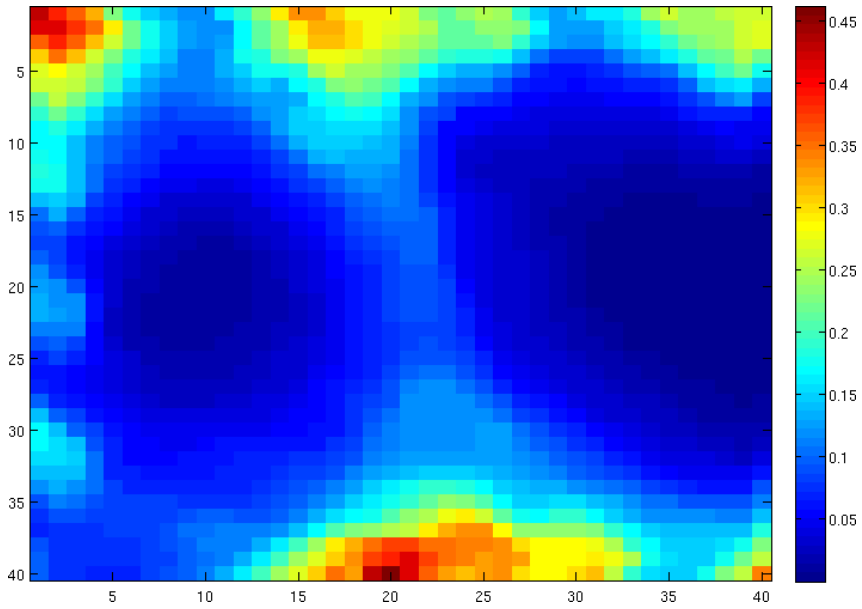


Figura 13: Matriz-U* calculada a partir das matrizes da figura 12

2.6.6 Algoritmo U*C

Para finalizar o método, nesta seção será apresentado o algoritmo, para isso Ultsch (2005) define o conceito de imersão, que consiste em: a partir do neurônio n mais próximo de uma amostra de dado x, seguir um gradiente descendente na Matriz-U encontrando um mínimo nu, e a partir daí seguir um gradiente ascendente na Matriz-P parando em uma unidade np, esse processo foi definido como a imersão do neurônio n, i.e. $I(n) = np$.

O propósito dessa definição é a partir de um exemplo de dado, encontrar o neurônio que está mais próximo do centro de um grupo de dados, já que descer um gradiente na Matriz-U, na maioria dos casos, representa mover-se em direção ao centro de um grupo, o mesmo sentido é dado para a subida de um gradiente na Matriz-P. Tudo isso para melhorar a classificação de exemplos de dados que encontram-se nas bordas dos grupos.

Depois de executar esse procedimento para todos os neurônios do mapa, é executado um algoritmo de processamento de imagens denominado *Watershed* utilizando a Matriz-U* como entrada. Esse passo tem como finalidade definir os limites de separação entre os grupos, os quais são chamadas de barreiras, ou diques no contexto desse algoritmo. Considerando os dados de entrada como um relevo topográfico, a ideia intuitiva é "inundar" a imagem a partir de cada mínimo local, chamados de bacias de captação ou represas, e conforme as bacias vão aumentando de nível elas podem se encontrar, nesses pontos de encontro são construídos diques (VINCENT; SOILLE, 1991). Veja na figura 14 a ilustração do problema.

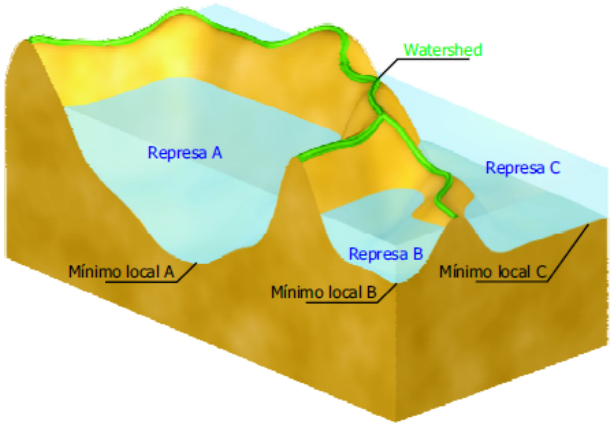


Figura 14: Ilustração do algoritmo *Watershed*
Fonte: Klava (2006)

Finalmente, o resumo do algoritmo proposto segue:

1) Calcular a Imersão de todos os neurônios do mapa.

2) Executar o *Watershed* da Matriz-U*.

3) Realizar uma Partição do resultado do passo 1 utilizando as regiões demarcadas pelo *Watershed*.

4) Retirar um exemplo x dos dados de entrada e calcular o neurônio mais próximo $bm(x)$.

5) Atribuir o exemplo x a um grupo j se $I(bm(x))$ estiver na região C_j.

3 Metodologia

A parte prática da proposta do presente trabalho, consiste na implementação e verificação do método exposto na seção 2.6, utilizando a linguagem C++ afim de obter absoluto controle sobre cada detalhe envolvido. Para atingir esse objetivo primeiramente foi implementado um treinamento simples do mapa auto-organizável segundo Haykin (2009, p. 464). Após se verificar por comparação com a implementação presente no software MATLAB® que o treinamento estava ocorrendo como devido, ou seja, os pesos do mapa estavam adquirindo a topologia dos dados de entrada, foram realizados os cálculos das matrizes U, P e U*.

A primeira matriz a ser calculada não precisa de nenhum método auxiliar, o resultado é obtido somente pelas médias das distâncias entre os neurônios vizinhos, como apresentado na equação 2.13. Já para o cálculo da Matriz-P, enfrentamos o problema de estimação da densidade e a saída apontada pelo autor é utilizar hiperesferas ao redor dos pesos da rede. O ponto crítico desse método é a escolha do raio, que foi contornado, como sugerido pelo autor, utilizando o método do raio de Pareto (ULTSCH, 2003a). Esse método assume que existe um certo número de grupos nos dados, para amenizar o impacto dessa hipótese, em Ultsch (2003b) é introduzido um fator de correção obtido através de experimentos práticos executados pelo autor. O método para cálculo do raio foi implementado utilizando busca binária para reduzir o tempo de execução.

Após o cálculo da Matriz-P o autor sugere a aplicação de um filtro por mediana (PHILLIPS, 2000, p. 80) para reduzir o ruído, mantendo os gradientes presentes na matriz. Em seguida, a ultima matriz a ser calculada é mais simples, pois somente envolve as fórmulas apresentadas na seção 2.6.5. Finalmente foi executada a implementação do processo de Imersão e do algoritmo *Watershed* sugerido pelo autor, descrito em Vincent e Soille (1991). Pelos motivos já expostos anteriormente, os testes foram executados utilizando o *data-set* FCPS, como já é conhecida a classificação desses dados, a verificação dos resultados se resume a contagem de quantos exemplos de dados foram atribuídos aos grupos corretos.

4 Resultados

Neste capítulo serão apresentados os resultados dos testes realizados com o método U*C de agrupamento de dados. O ponto crítico dessa técnica foi identificado como a definição do raio para o cálculo da estimativa de densidade, por isso foram feitos testes utilizando maneiras diferentes para estimar o raio. Primeiramente foi utilizado o raio de Pareto sugerido pelo autor. Outro modo de estipular o raio foi uma tentativa de automatizar e aperfeiçoar o processo, a ideia consiste em aproveitar informações contidas na Matriz-U utilizando o seu valor máximo, justificado pelo fato que os neurônios que encontram-se em regiões sem dados seriam os que forneceriam, ou seriam próximos, do valor máximo da Matriz-U.

Desta forma, o raio teria valores pequenos o suficiente para evidenciar as unidades com baixa densidade, pois poucos exemplos de dados estariam dentro da esfera centrada nesses neurônios. Ao mesmo tempo, seria grande suficiente para, nas regiões com densidade alta, representar a densidade real. Finalmente, um raio ótimo foi estipulado da seguinte maneira: escolher o melhor dos dois casos anteriores levando em consideração a representatividade obtida no cálculo da densidade, se nenhum dos dois raios obteve bons resultados, um valor foi estipulado manualmente comparando os valores obtidos na Matriz-P com a topologia dos dados de entrada.

Tabela 2: Valores dos raios utilizados no cálculo da Matriz-P

Data-Set	**Pareto**	**Ótimo**	**max(Matriz-U)**
Hepta	**0,96**	0,96	0,89
Lsun	0,34	0,30	**0,30**
Tetra	**0,40**	0,40	0,36
Chainlink	0,33	0,28	**0,28**
Atom	4,61	11,49	**11,49**
EngyTime	0,45	0,55	**0,55**
Target	**0,17**	0,17	0,74
TwoDiamonds	0,21	0,47	**0,47**
WingNut	0,29	0,35	0,17
GolfBall	0,29	0,12	**0,12**

Os três valores de raio para cada caso estão demonstrados na tabela 2, onde os valores destacados em negrito mostram qual raio foi escolhido como raio ótimo, se este foi o caso.

Tabela 3: Resultados do algoritmo U*C

	Raio utilizado para Matriz-P			
Data-Set	**Pareto**	**Ótimo**	**max(Matriz-U)**	**Segundo o autor**
Hepta	15,09%	15,09%	15,09%	100%
Lsun	75%	75%	75%	100%
Tetra	95,75%	95,75%	93,75%	100%
Chainlink	60,40%	53%	53%	100%
Atom	80,50%	81%	81%	100%
EngyTime	95,87%	95,85%	95,85%	90%
Target	51,30%	51,30%	51,30%	100%
TwoDiamonds	42,63%	99,62%	99,62%	100%
WingNut	77,95%	99,21%	50,59%	100%
GolfBall	30%	32%	32%	100%

A tabela 3 fornece os resultados obtidos com a implementação deste trabalho para os raios expostos anteriormente, onde o percentual exibido consiste em quantos exemplos de dados foram classificados corretamente em seus respectivos grupos. Abaixo é apresentada uma tabela contendo o valor dos raios para comparação:

4.1 Discussão dos resultados

Somente a exibição dos resultados em forma de percentual de acerto não é definitiva para a análise do método, pois no caso do conjunto *Chainlink* foi obtido uma taxa de acerto de 60,40% com o raio de Pareto, um pouco maior que nos outros resultados, mas se observarmos a segmentação resultante veremos que os grupos formados exibem estruturas incorretas não muito diferentes. Por esse motivo, os resultados intermediários do algoritmo, i.e. as matrizes U,P e U*, e os dados de entrada estão disponíveis no anexo A

Observando os resultados podemos ver que em alguns casos não se chegou no percentual de acerto obtido pelo autor do método, pois a coluna referente a execução com o raio de Pareto na tabela 3 deveria se equiparar aos valores apresentados em Ultsch (2005), replicados na última coluna da tabela. Um motivo claro e que pode ser notado é que o algoritmo *Watershed* geralmente sobre-segmenta a imagem apresentada(VINCENT; SOILLE, 1991), na tentativa de contornar esse problema, o autor da técnica U*C propôs o uso de um algoritmo que utiliza mínimos locais já demarcados por um preprocessamento (VINCENT; SOILLE, 1991).

Ultsch e Herrmann (2006) propõe a utilização de informações da Matriz-U e Matriz-P para encontrar marcadores, definindo centros de grupos, fornecendo esse ponto de partida para o algoritmo de segmentação, esse adicional não foi citado no artigo em que foi baseada a implementação (ULTSCH, 2005), por esse motivo não foi implementado. Um problema perceptível é que em casos como o do conjunto *GolfBall* e *ChainLink*, não existe um centro do grupo, mas no caso do conjunto *Hepta* o resultado poderia ter sido melhor.

Outro ponto que pode ser levado em consideração é a utilização de topologias alternativas da rede de Kohonen. Segundo o autor, existem casos em que uma configuração toroidal consegue adquirir melhor a topologia dos dados de entrada. O modelo toroidal define as coordenadas dos neurônios em uma estrutura em forma de um toroide, de modo que quando representado em duas dimensões, os neurônios da extremidade superior são considerados vizinhos da extremidade inferior, o mesmo ocorre para as unidades da extrema esquerda e direita (ULTSCH; HERRMANN, 2006). Veja na figura 15 a representação gráfica desse tipo de topologia.

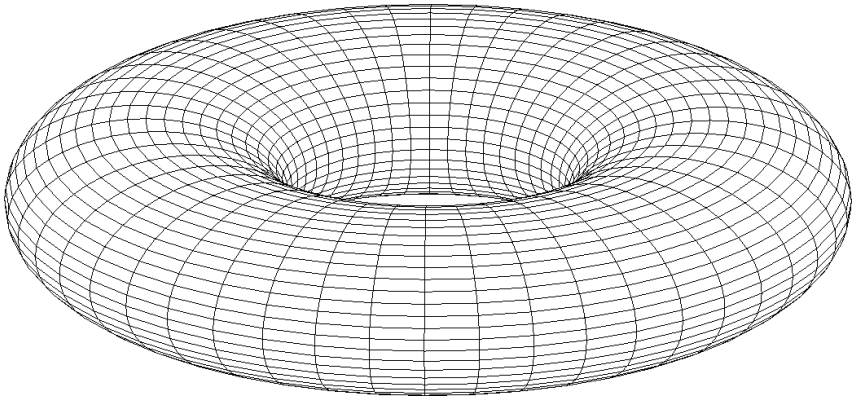


Figura 15: Topologia toroidal

Cada caso apresentou sua particularidade em seus resultados, para o conjunto *GolfBall* que apresenta somente um grupo, a matriz U* foi contaminada pois as matrizes U e P são praticamente uniformes. Pode-se notar que o intervalo que seus valores encontram-se é menor, então uma variação de uma unidade na densidade representa uma variação grande relativamente. O raio de Pareto calculado obteve maior percentual de acerto pois ele sobre-estimou a densidade em algumas regiões, criando um grupo de dados que abrange uma parte maior das amostras.

A utilização do valor máximo da Matriz-U mostrou-se interessante em alguns casos, pois somente nos casos dos conjuntos *Tetra* e *WingNut*, se obteve resultados melhores utilizando o raio ótimo. No primeiro conjunto, a performance caiu de forma a não prejudicar

fortemente os resultados, já no *WingNut*, pode-se notar que foi obtido o pior resultado entre os três tipos de raio experimentados, isso aconteceu pois existem regiões dentro dos grupos que apresentam distâncias entre neurônios maiores do que os espaços entre grupos, elevando assim os valores da Matriz-U.

5 Conclusão

O método implementado mostrou-se interessante pois propõe uma nova abordagem para agrupamento de dados utilizando Mapas Auto-Organizáveis, a Matriz-P exibe informações importantes sobre um conjunto de dados, tornando evidentes suas estruturas e grupos presentes na maioria dos casos experimentados. O uso desta técnica para grandes bases de dados pode ser feito explorando as propostas de treinamento rápido da SOM, um exemplo pode ser visto em Nöcker, Mörchen e Ultsch (2006).

O uso de informações da Matriz-U indicou uma possibilidade a ser investigada, pois o custo de encontrar seu valor máximo é linear, além de ser um método simples e que em seis dos dez casos se mostrou a melhor escolha, além disso, em dois casos que o raio de Pareto se saiu melhor, os valores de raio obtidos foram bem próximos. Conclusões definitivas não podem ser feitas pois este uso não foi embasado cientificamente, foi puramente intuitivo.

5.1 Trabalhos futuros

Como trabalho futuro pode ser feita a implementação do complemento para a técnica U*C descrito em (ULTSCH; HERRMANN, 2006), o qual indica como utilizar as informações das matrizes U e P para encontrar núcleos de agrupamentos, melhorando o resultado final do algoritmo. O trabalho citado também descreve um fluxo automático para o uso do método implementado, o que pode ser de grande interesse. Outra sugestão seria a execução de testes com *data-sets* reais, ou de grandes proporções, colocando a prova as capacidades de visualização e acurácia na formação de grupos nessas condições.

Uma investigação mais a fundo das razões pelas quais os valores máximos da Matriz-U são raios bons para o cálculo da Matriz-P faz-se necessária. Existe uma explicação intuitiva, mas é preciso formalizá-la e justificá-la, se esse for o caso. A implementação da topologia toroidal da SOM também é uma adição que pode vir a diminuir os erros de representação da rede neural, assim pode ser considerada uma continuação deste trabalho.

Referências

BEZDEK†, J. C. Cluster validity with fuzzy sets. *Cybernetics and Systems*, Taylor & Francis, v. 3, n. 3, p. 58–73, 1973.

BISHOP, C. M. *Neural networks for pattern recognition*. Oxford, UK: Oxford University Press, 1996. ISBN 0-19-853849-9.

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.

DIMITRIADOU, E.; WEINGESSEL, A.; HORNIK, K. Voting-merging: An ensemble method for clustering. In: *ICANN '01: Proceedings of the International Conference on Artificial Neural Networks*. London, UK: Springer-Verlag, 2001. p. 217–224. ISBN 3-540-42486-5.

DU, K.-L. Clustering: A neural network approach. *Neural Networks*, v. 23, n. 1, p. 89–107, jan. 2010. ISSN 08936080. Disponível em: <http://dx.doi.org/10.1016/j.neunet-.2009.08.007>.

GREENE, D.; CUNNINGHAM, P.; MAYER, R. Unsupervised learning and clustering. In: CORD, M.; CUNNINGHAM, P. (Ed.). *Machine Learning Techniques for Multimedia*. [S.l.]: Springer Berlin Heidelberg, 2008, (Cognitive Technologies). p. 51–90.

HANSEN, P.; JAUMARD, B. Cluster analysis and mathematical programming. *Math. Program.*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 79, n. 1-3, p. 191–215, 1997. ISSN 0025-5610.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation (2nd Edition)*. 2. ed. [S.l.]: Prentice Hall, 1998. Hardcover. ISBN 0132733501.

HAYKIN, S. *Neural Networks and Learning Machines*. 3. ed. Upper Saddle River, NJ, USA: Pearson Education, 2009. 936 p. Softcover. ISBN 0-13-129376-1.

JAIN, A. K.; DUBES, R. C. *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988. ISBN 0-13-022278-X. Disponível em: <http://www.cse-.msu.edu/~jain/Clustering_Jain_Dubes.pdf>.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 31, n. 3, p. 264–323, 1999. ISSN 0360-0300.

JUNG, Y. et al. A decision criterion for the optimal number of clusters in hierarchical clustering. *Journal of Global Optimization*, Kluwer Academic Publishers, Hingham, MA, USA, v. 25, n. 1, p. 91–111, 2003. ISSN 0925-5001.

KAUFMAN, L.; ROUSSEEUW, P. *Finding Groups in Data An Introduction to Cluster Analysis*. New York: Wiley Interscience, 1990.

KLAVA, B. Ferramenta interativa para segmentação de imagens digitais. Trabalho de Formatura Supervisionado - Instituto de Matemática e Estatística, Universidade de São Paulo. dez. 2006.

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

KOHONEN, T. *Self-Organizing Maps*. 3. ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001. 501 p.

LANCE, G. N.; WILLIAMS, W. T. A general theory of classificatory sorting strategies: 1. hierarchical systems. *The Computer Journal*, v. 9, n. 4, p. 373–380, February 1967.

LUGER, G. F. *Inteligência Artificial: Estruturas e Estratégias para a Solução de Problemas Complexos*. 4. ed. Porto Alegre: Bookman, 2004. 774 p. Hardcover. ISBN 8536303964.

MELIN, P.; CASTILLO, O. *Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing*. Heidelberg, Alemanha: Springer Berlin, 2005.

MORII, F. Clustering based on lvq and a split and merge procedure. Springer-Verlag, Berlin, Heidelberg, p. 57–66, 2008.

NÖCKER, M.; MÖRCHEN, F.; ULTSCH, A. An algorithm for fast and reliable esom learning. In: *ESANN*. [S.l.: s.n.], 2006. p. 131–136.

PAL, N.; BEZDEK, J.; TSAO, E.-K. Generalized clustering networks and kohonen's self-organizing scheme. *Neural Networks, IEEE Transactions on*, v. 4, n. 4, p. 549 –557, jul 1993. ISSN 1045-9227.

PHILLIPS, D. *Image Processing in C*. 2. ed. Lawrence, Kansas: Dwayne Phillips, 2000. 794 p.

PRASS, F. S. Estudo comparativo entre algoritmos de análise de agrupamentos em data mining. Santa Catarina. Dissertação (Mestrado) – Instituto de Informática e Estatística, UFSC. nov. 2004.

RUSSEL, S.; NORVIG, P. *Inteligência artificial: tradução da segunda edição*. Rio de Janeiro: Elsevier, 2004. 1021 p. ISBN 8535211772.

ULTSCH, A. Maps for the visualization of high-dimensional data spaces. In: *Proc. Workshop on Self-Organizing Maps*. Kyushu, Japão: [s.n.], 2003. p. 225–230.

ULTSCH, A. Optimal density estimation in data containing clusters of unknown structure. *Technical Report No. 34, Dept. of Mathematics and Computer Science, University of Marburg, Germany*, 2003.

ULTSCH, A. Clustering with SOM: U*C. In: *Proc. Workshop on Self-Organizing Maps*. Paris, France: [s.n.], 2005. p. 75–82.

ULTSCH, A.; HERRMANN, L. *Automatic Clustering with U*C*. 2006.

VESANTO, J. Som-based data visualization methods. *Intelligent Data Analysis*, v. 3, p. 111–126, 1999.

VESANTO, J. et al. *SOM Toolbox for Matlab*. [S.l.], 2000. Disponível em: <citeseer.ist-.psu.edu/vesanto00som.html>.

VINCENT, L.; SOILLE, P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 13, n. 6, p. 583–598, 1991.

XU, R.; WUNSCH, D. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, v. 16, n. 3, p. 645–678, 2005. Disponível em: <http://dx.doi.org/10.1109-/TNN.2005.845141>.

ZIMMERMANN, H.-J. *Fuzzy Set Theory and its Applications*. 2. ed. [S.l.]: Springer, 1992. Hardcover. ISBN 079239075X.

*ANEXO A – Matrizes intermediárias do método U*C*

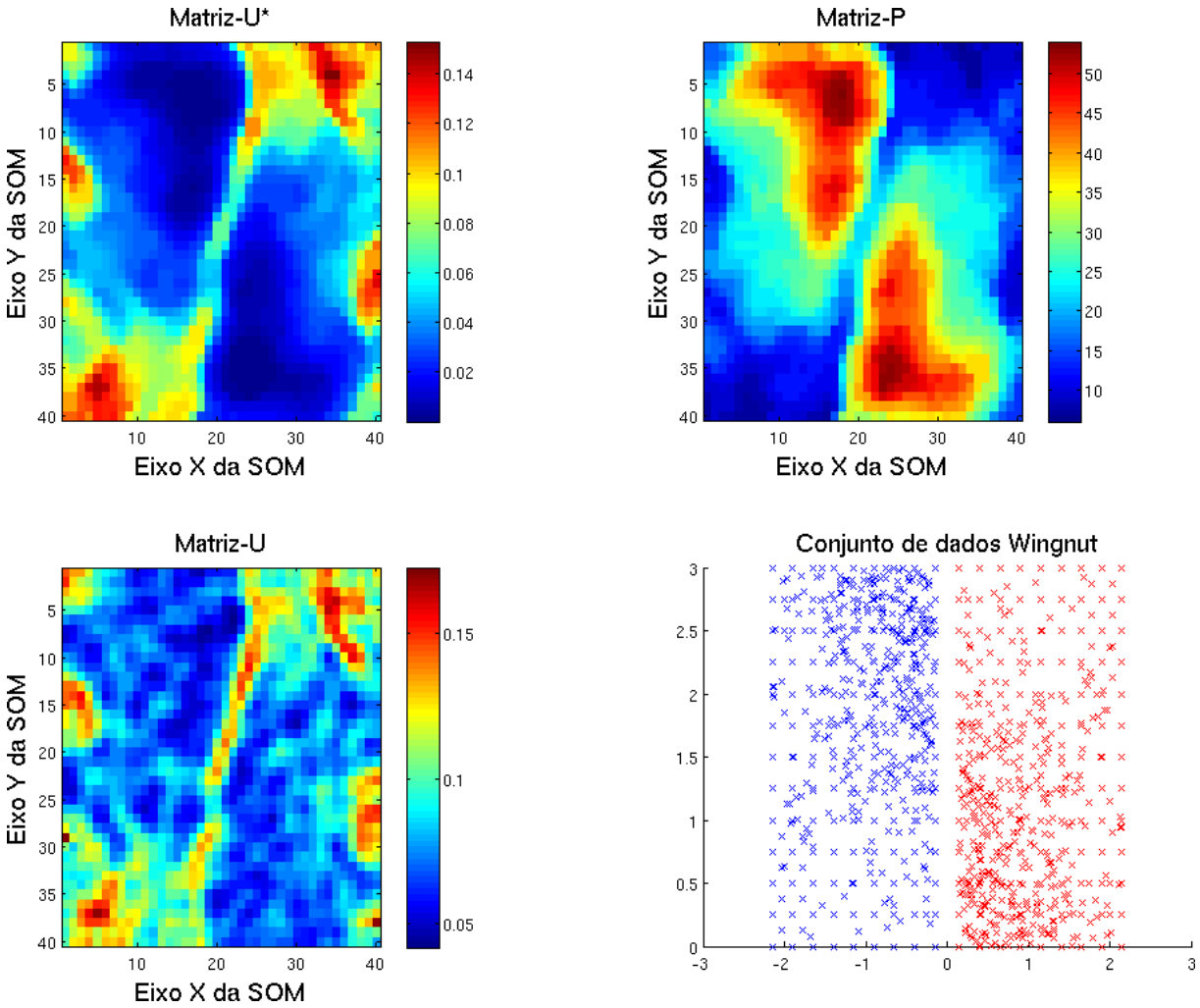


Figura 16: Matrizes intermediárias com raio de Pareto para o conjunto Wingnut

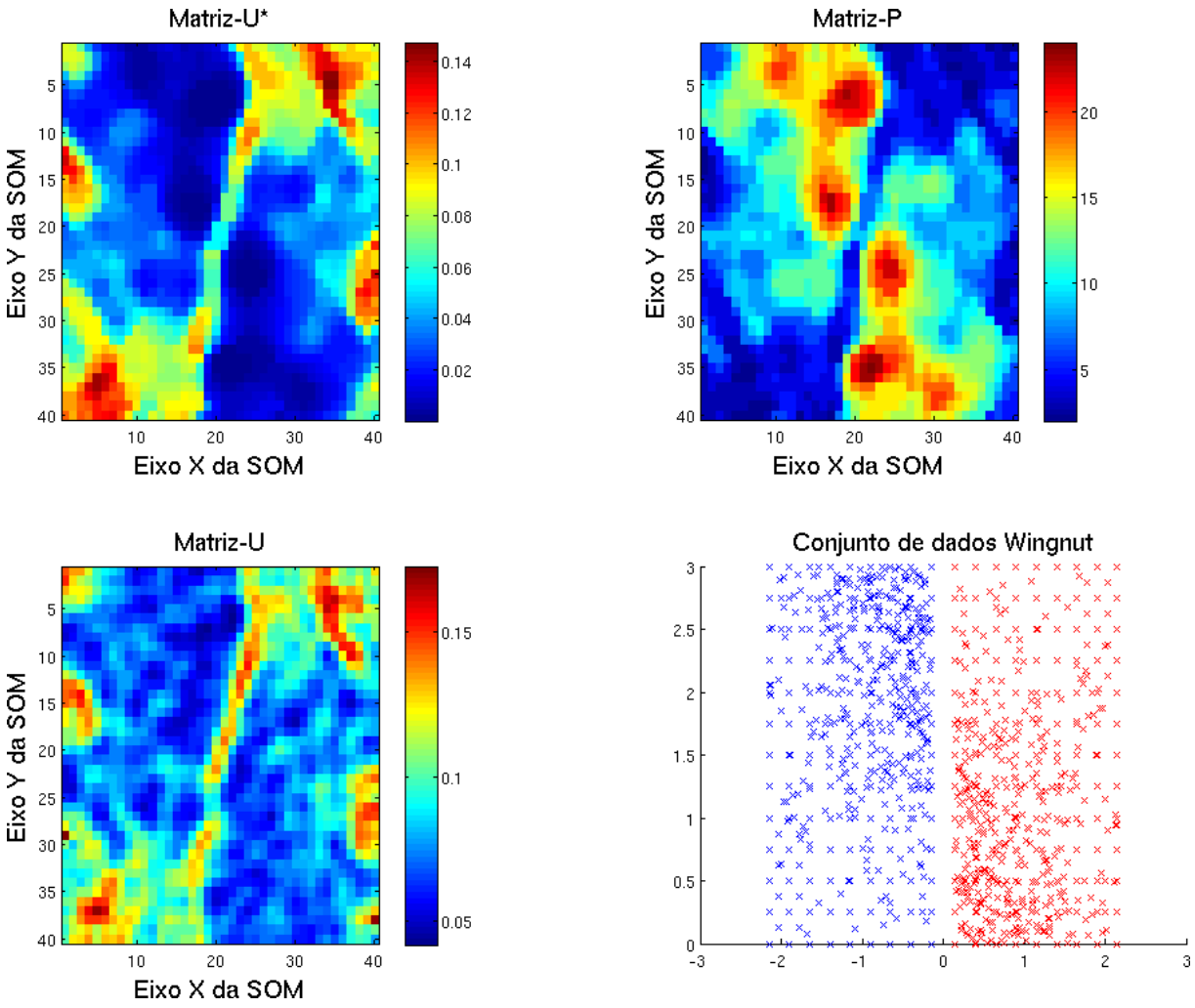


Figura 17: Matrizes intermediárias com raio max(Matriz-U) para o conjunto Wingnut

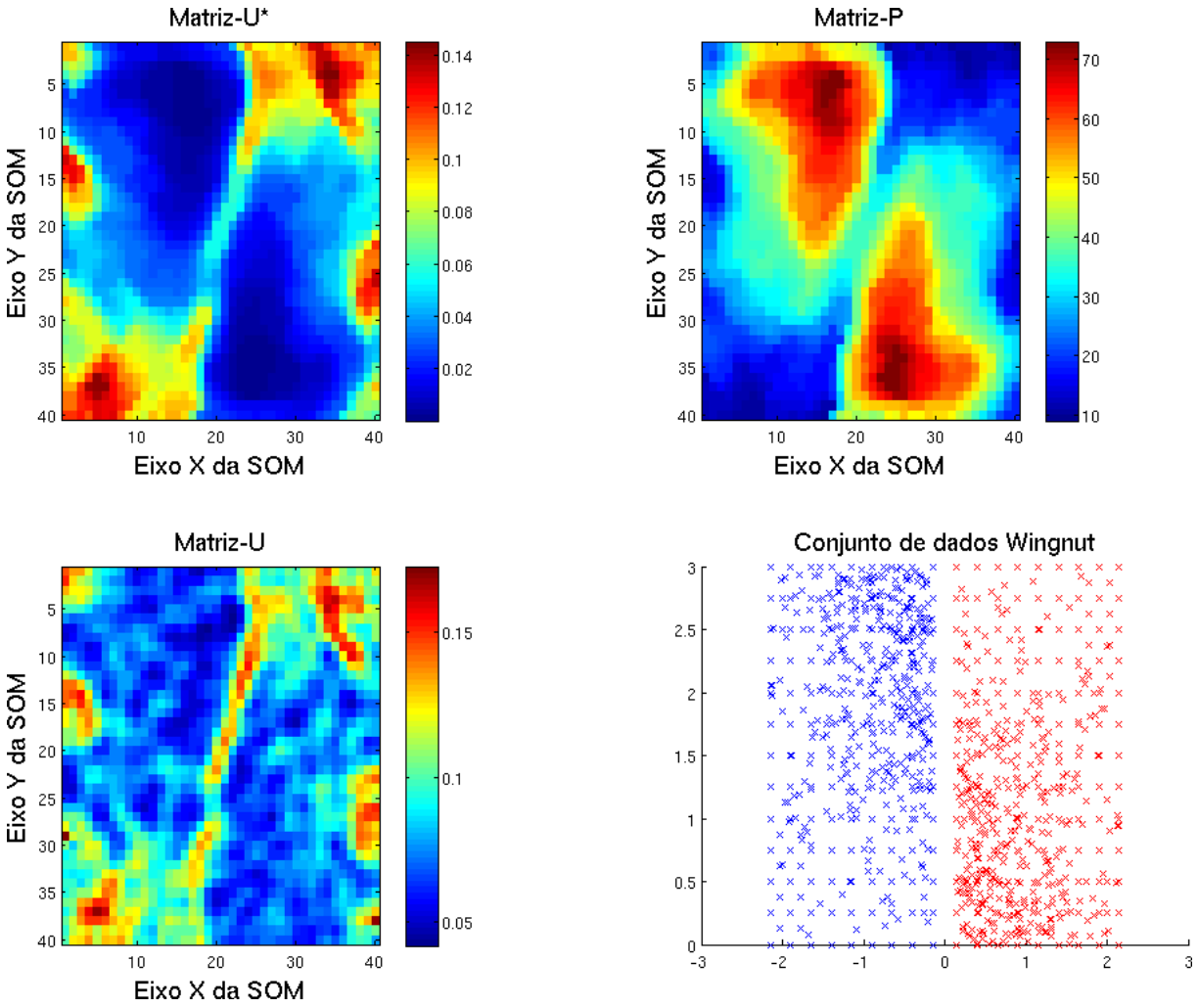


Figura 18: Matrizes intermediárias com raio ótimo para o conjunto Wingnut

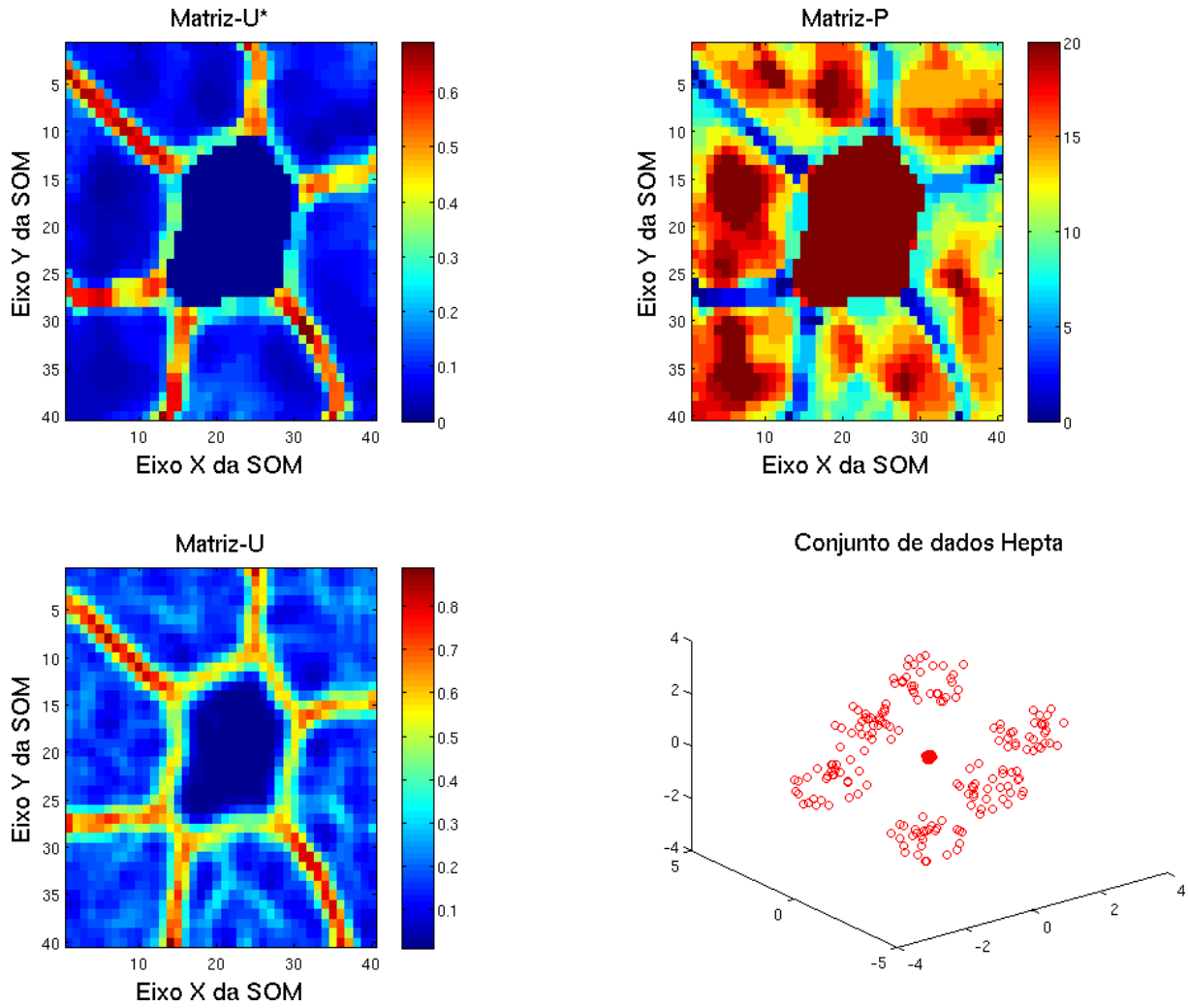


Figura 19: Matrizes intermediárias com raio de Pareto para o conjunto Hepta

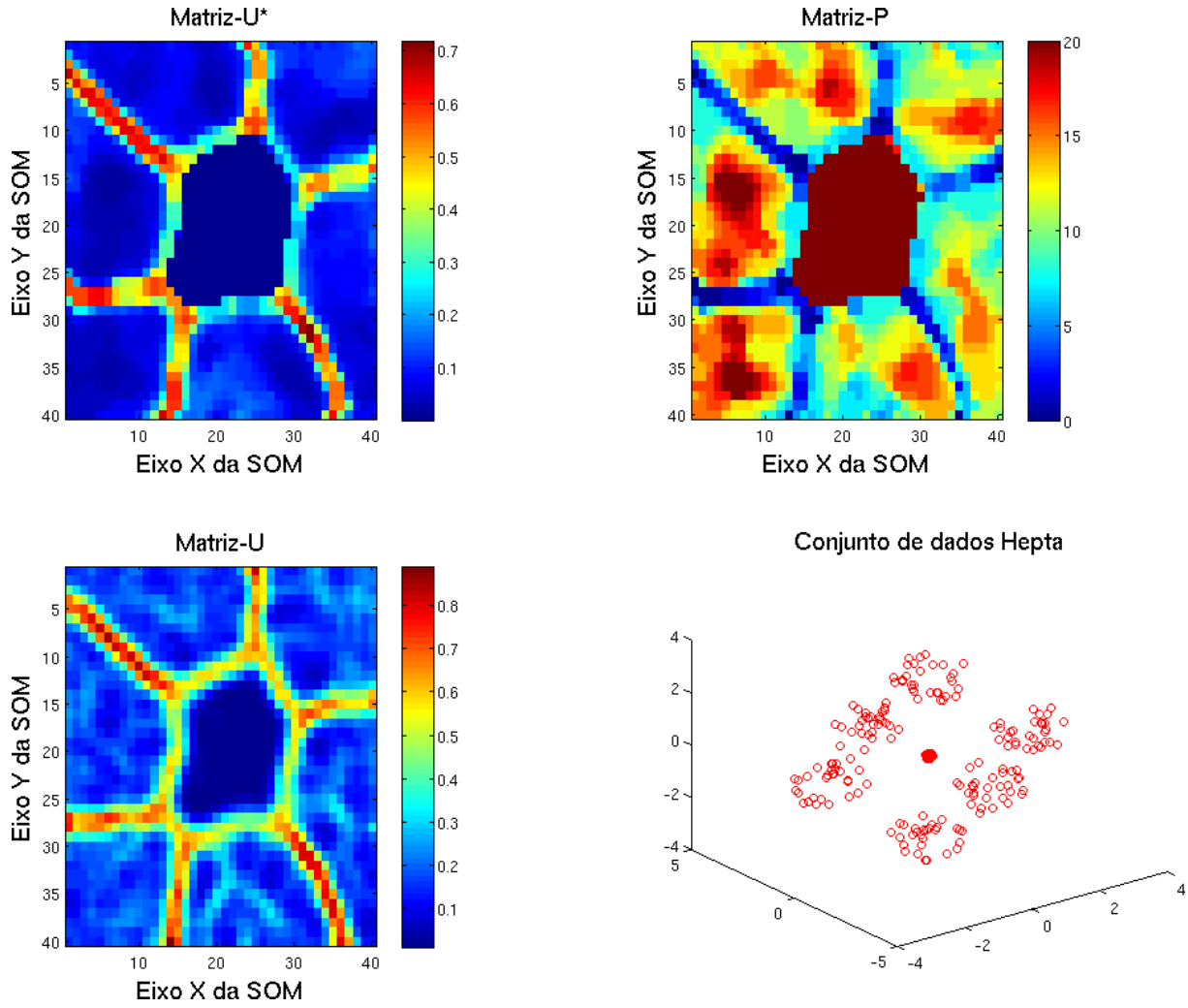


Figura 20: Matrizes intermediárias com raio max(Matriz-U) para o conjunto Hepta

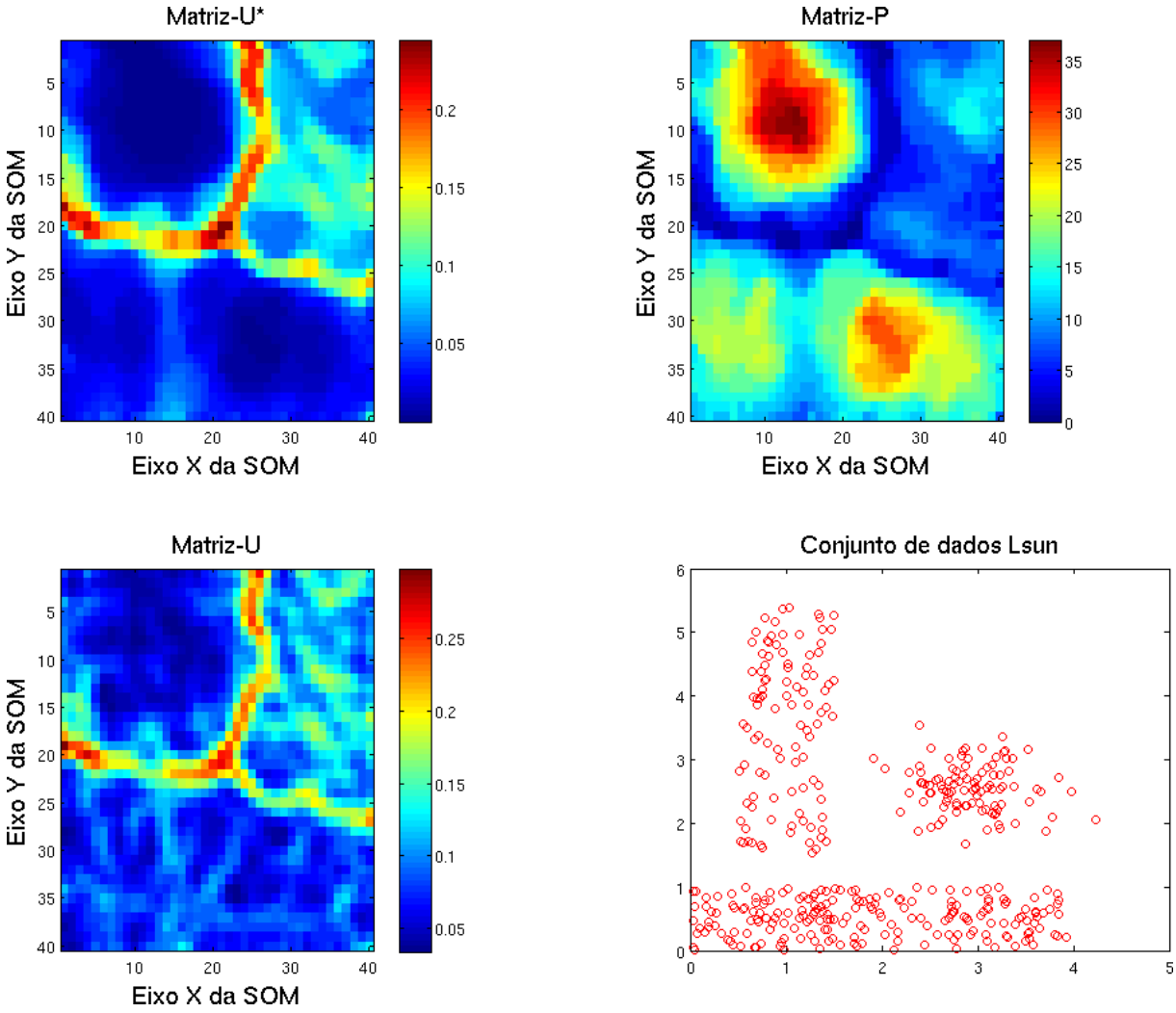


Figura 21: Matrizes intermediárias com raio de Pareto para o conjunto Lsun

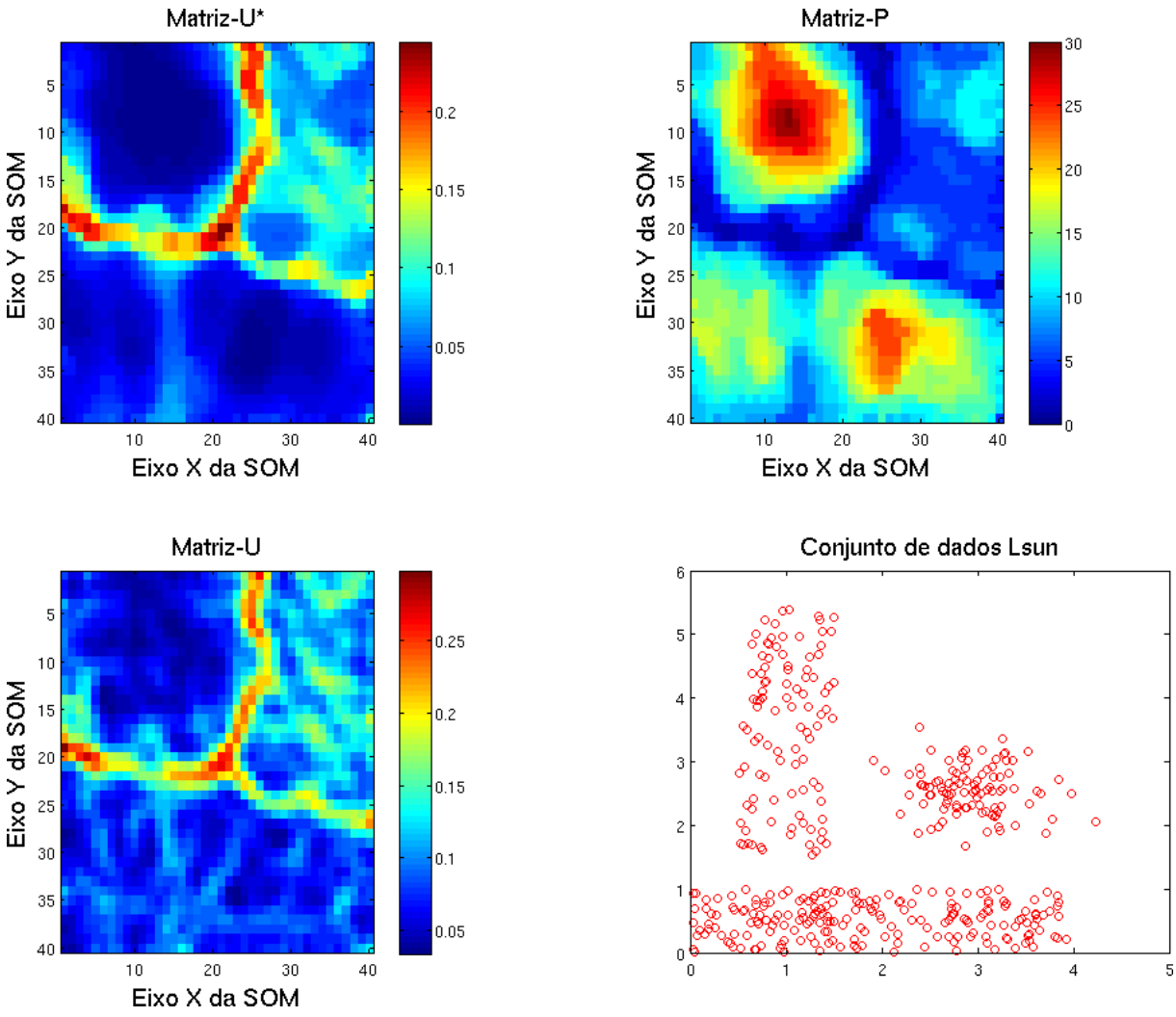


Figura 22: Matrizes intermediárias com raio max(Matriz-U) para o conjunto Lsun

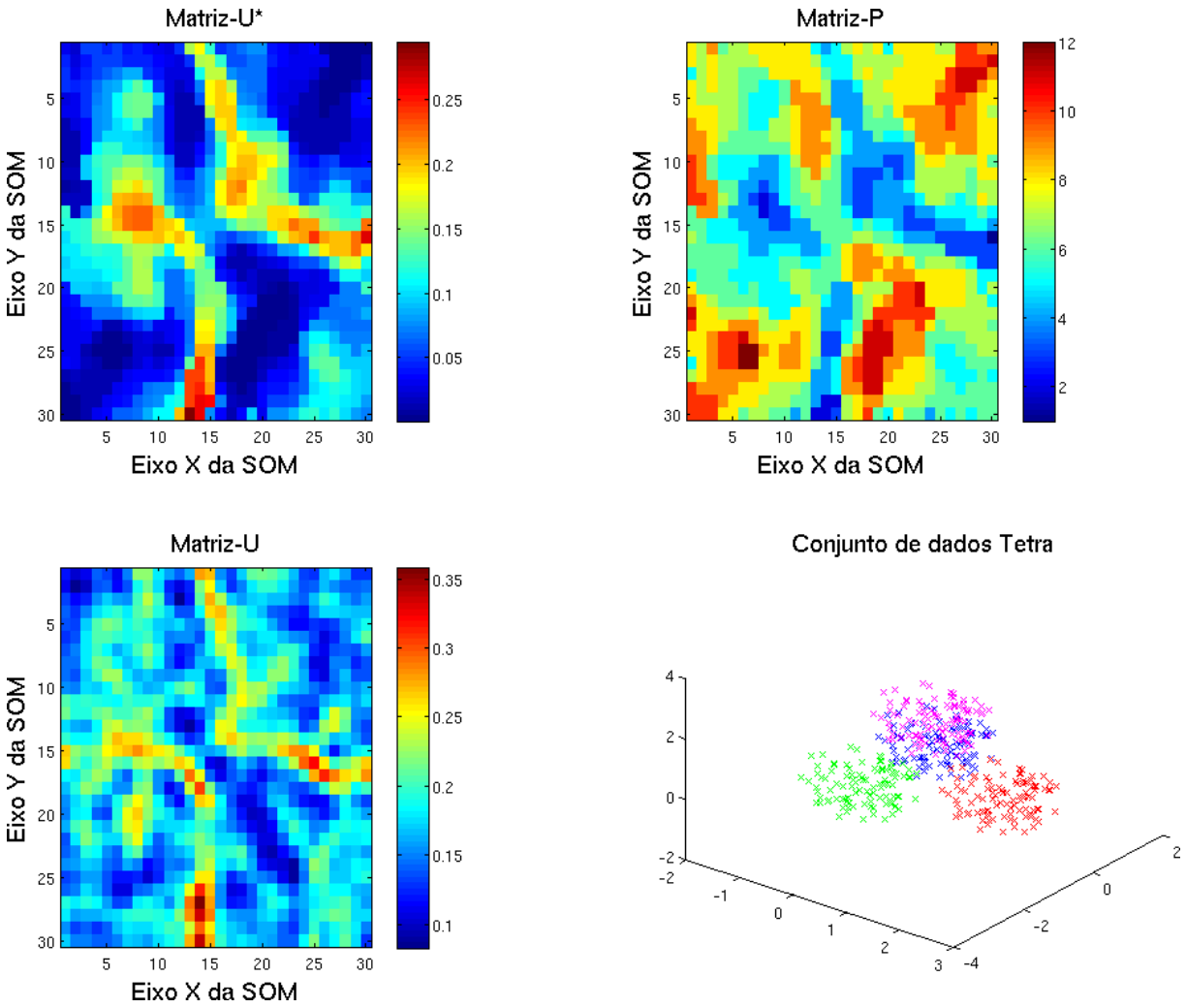


Figura 23: Matrizes intermediárias com raio de Pareto para o conjunto Tetra

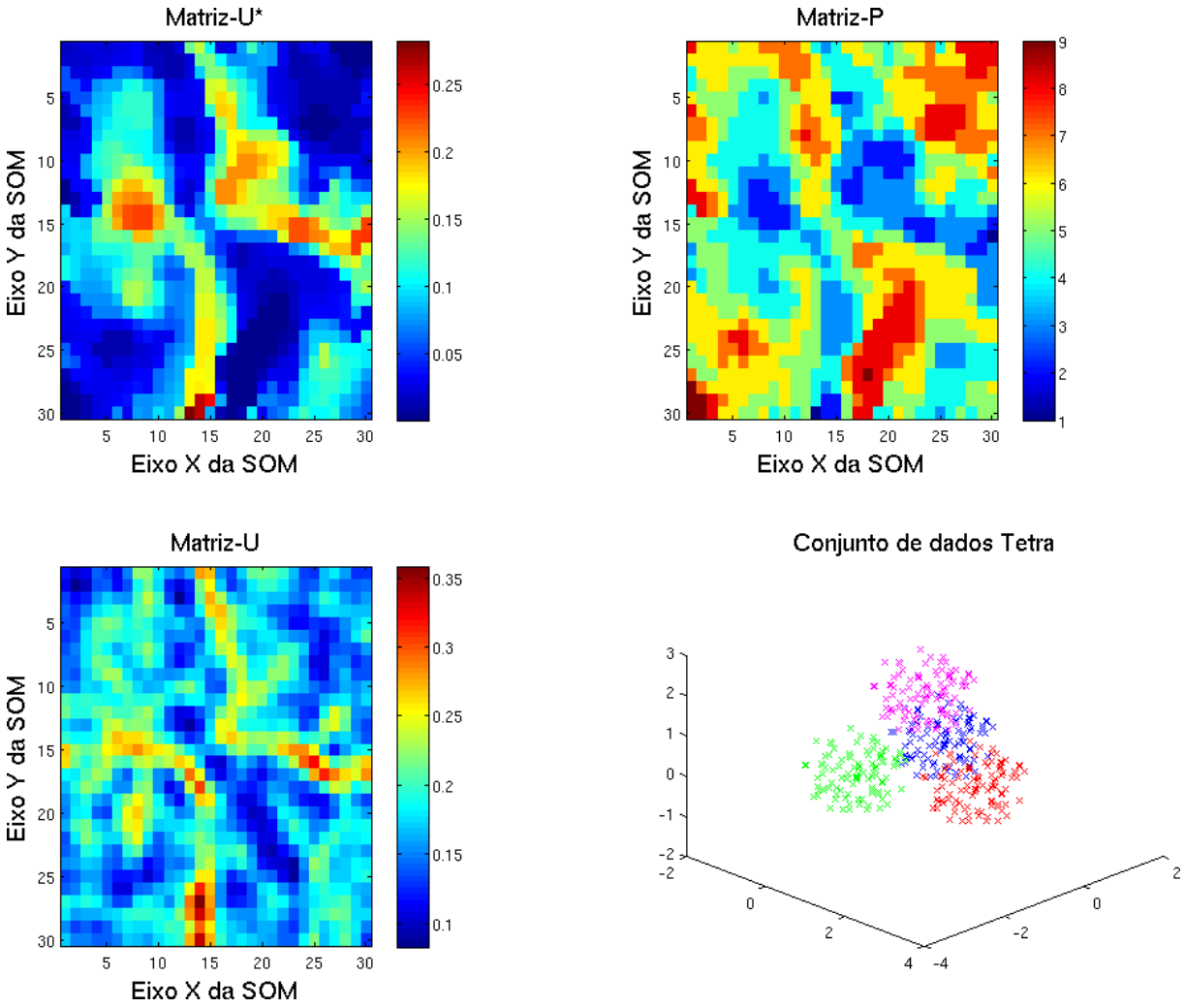


Figura 24: Matrizes intermediárias com raio max(Matriz-U) para o conjunto Tetra

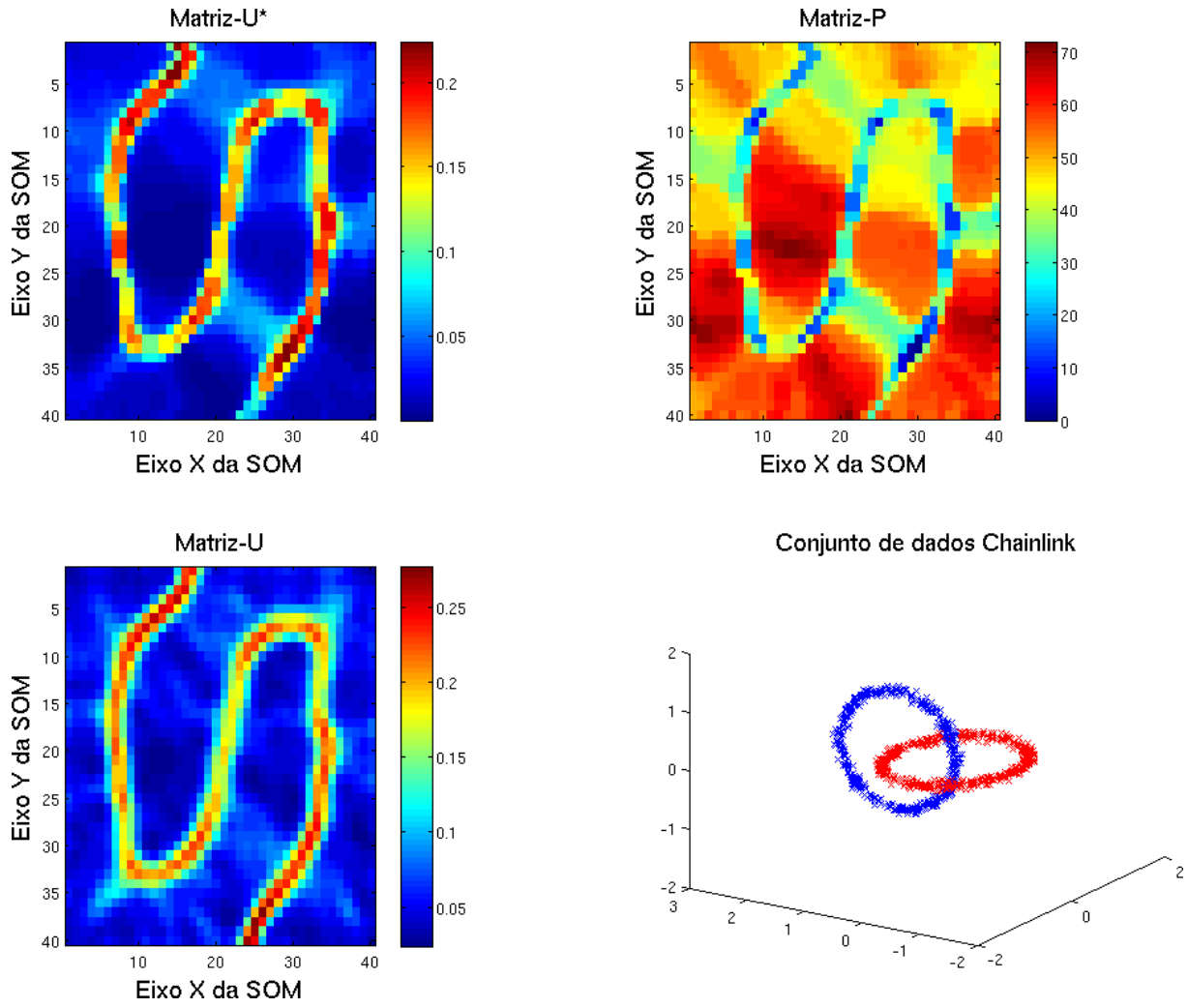


Figura 25: Matrizes intermediárias com raio de Pareto para o conjunto Chainlink

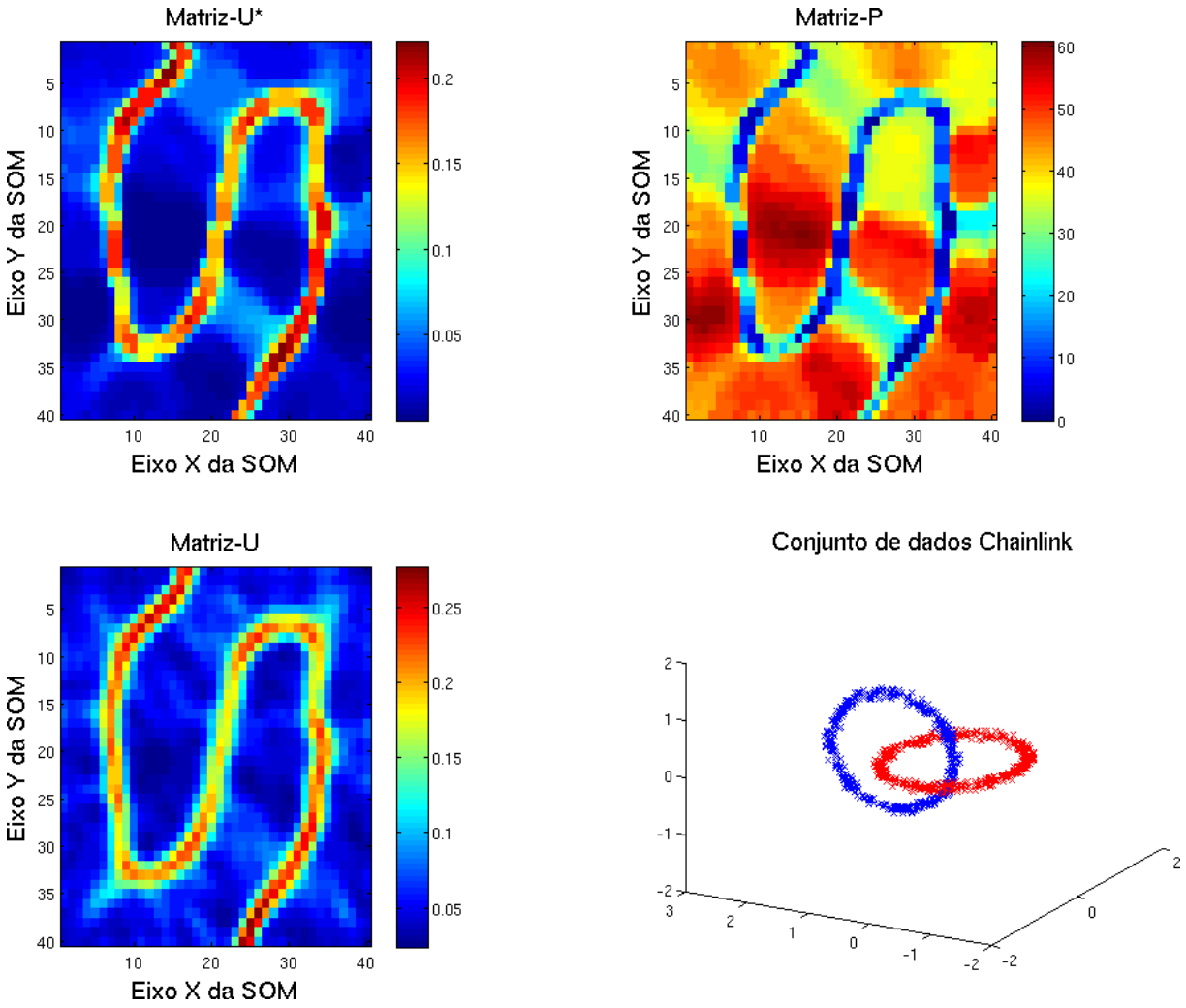


Figura 26: Matrizes intermediárias com raio max(Matriz-U) para o conjunto Chainlink

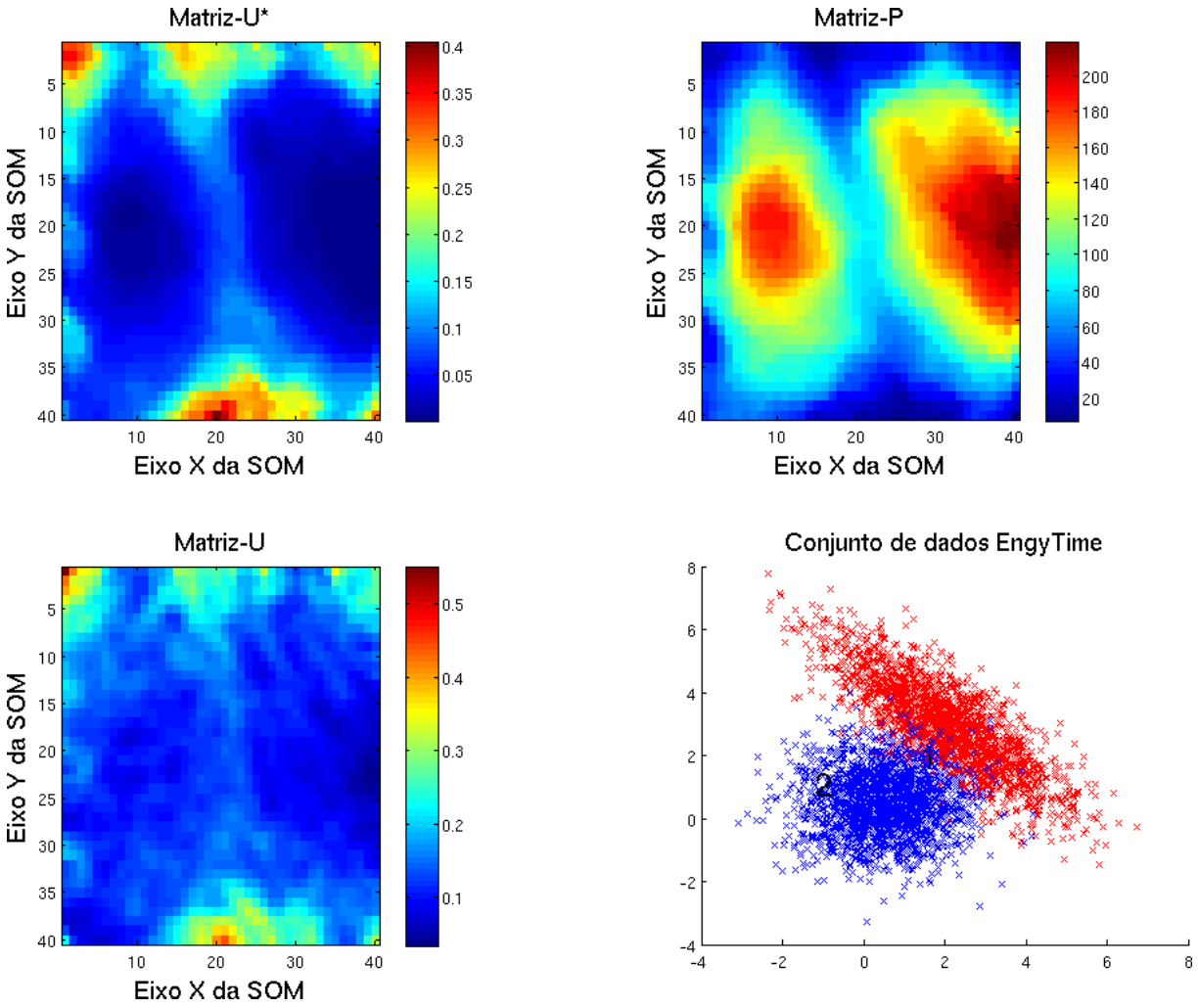


Figura 27: Matrizes intermediárias com raio de Pareto para o conjunto EngyTime

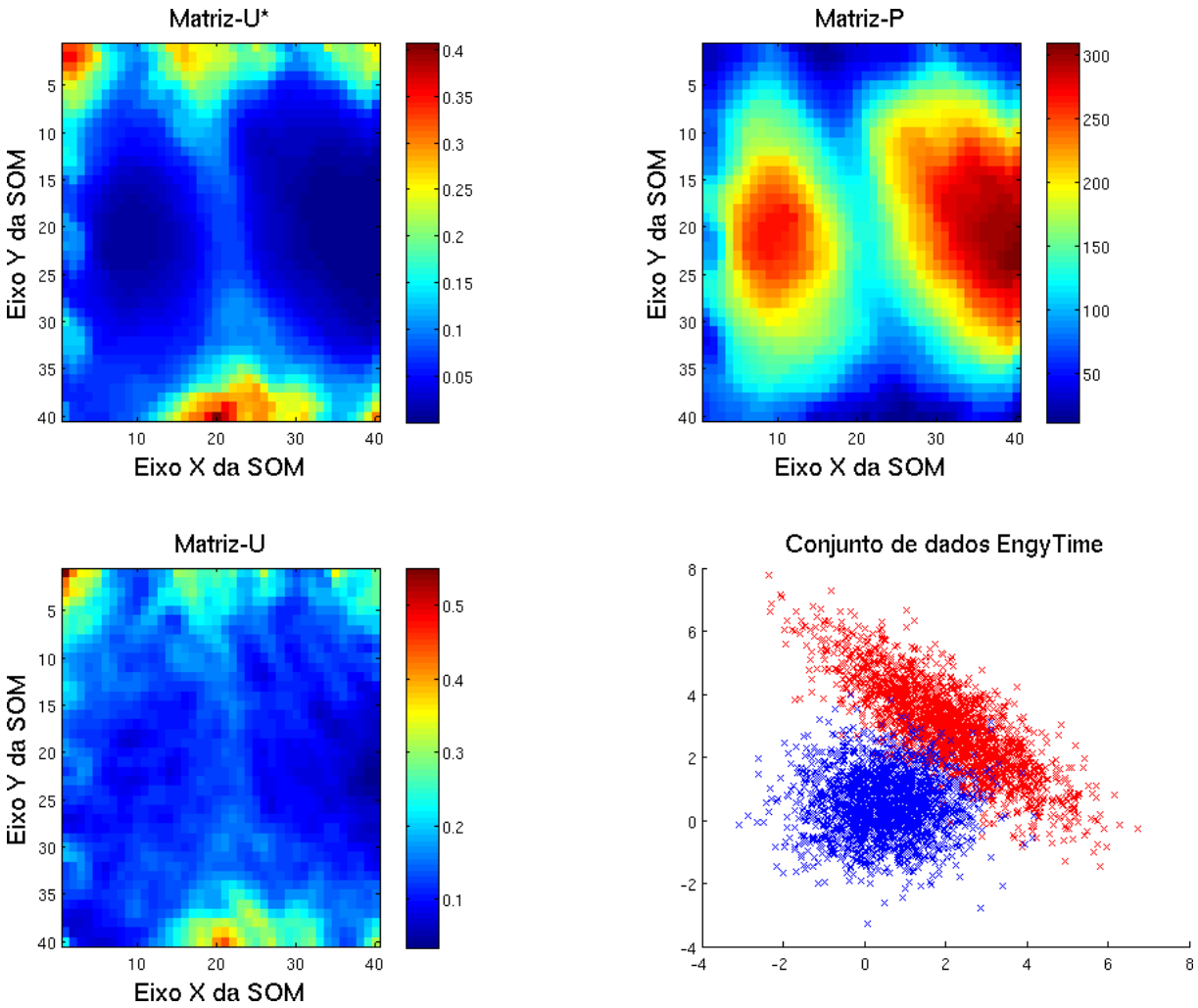


Figura 28: Matrizes intermediárias com raio max(Matriz-U) para o conjunto EngyTime

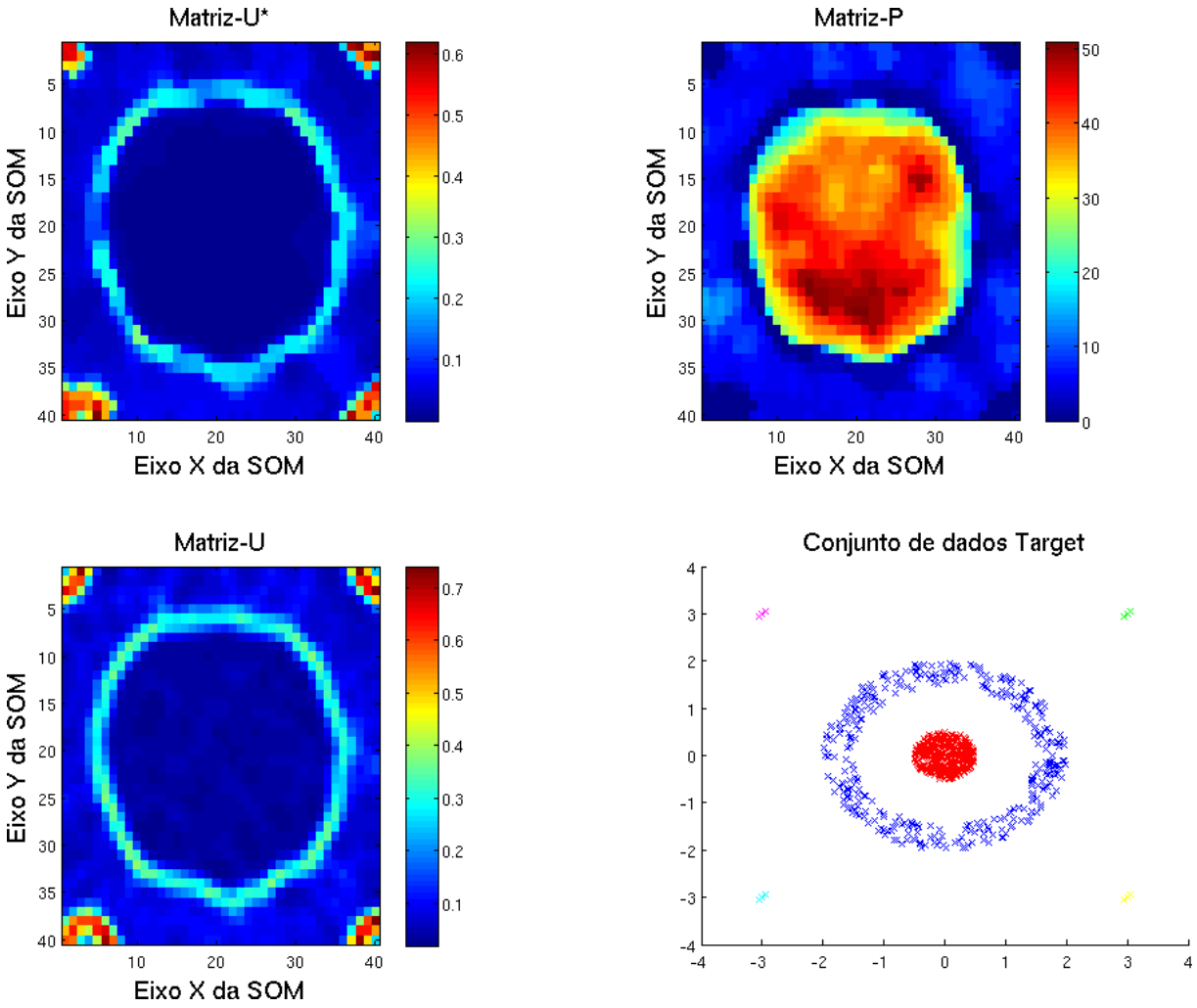


Figura 29: Matrizes intermediárias com raio de Pareto para o conjunto Target

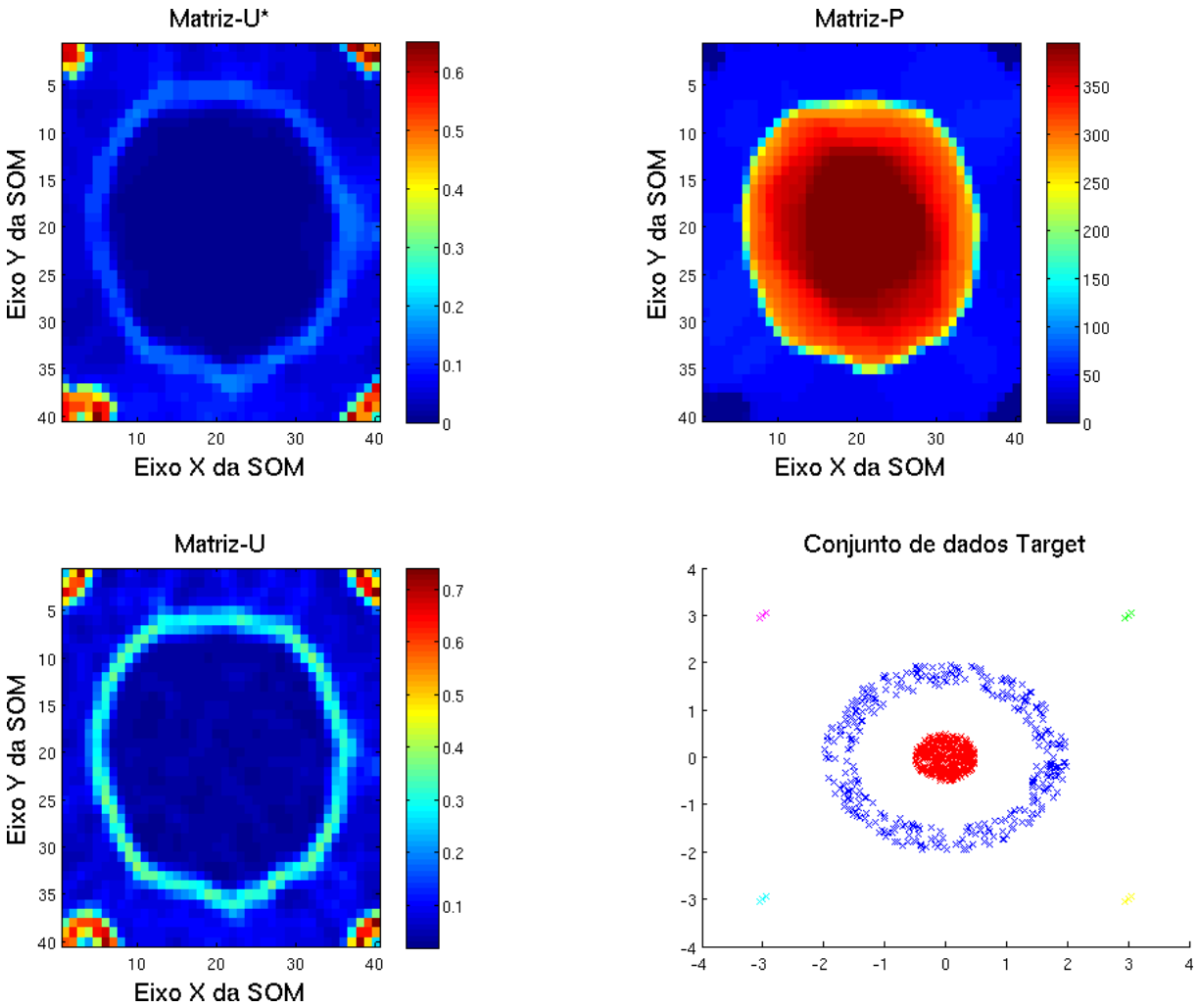


Figura 30: Matrizes intermediárias com raio max(Matriz-U) para o conjunto Target

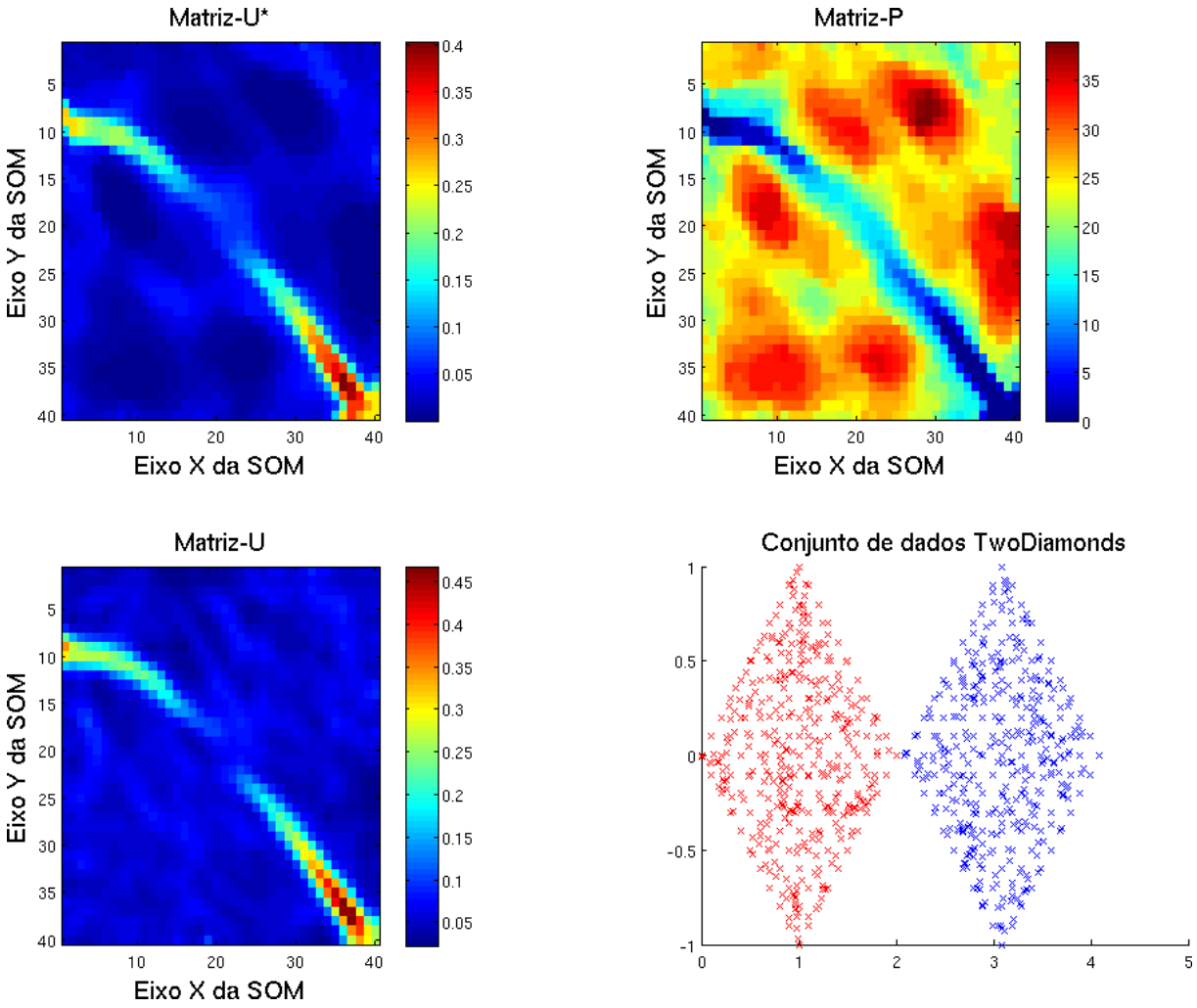


Figura 31: Matrizes intermediárias com raio de Pareto para o conjunto TwoDiamonds

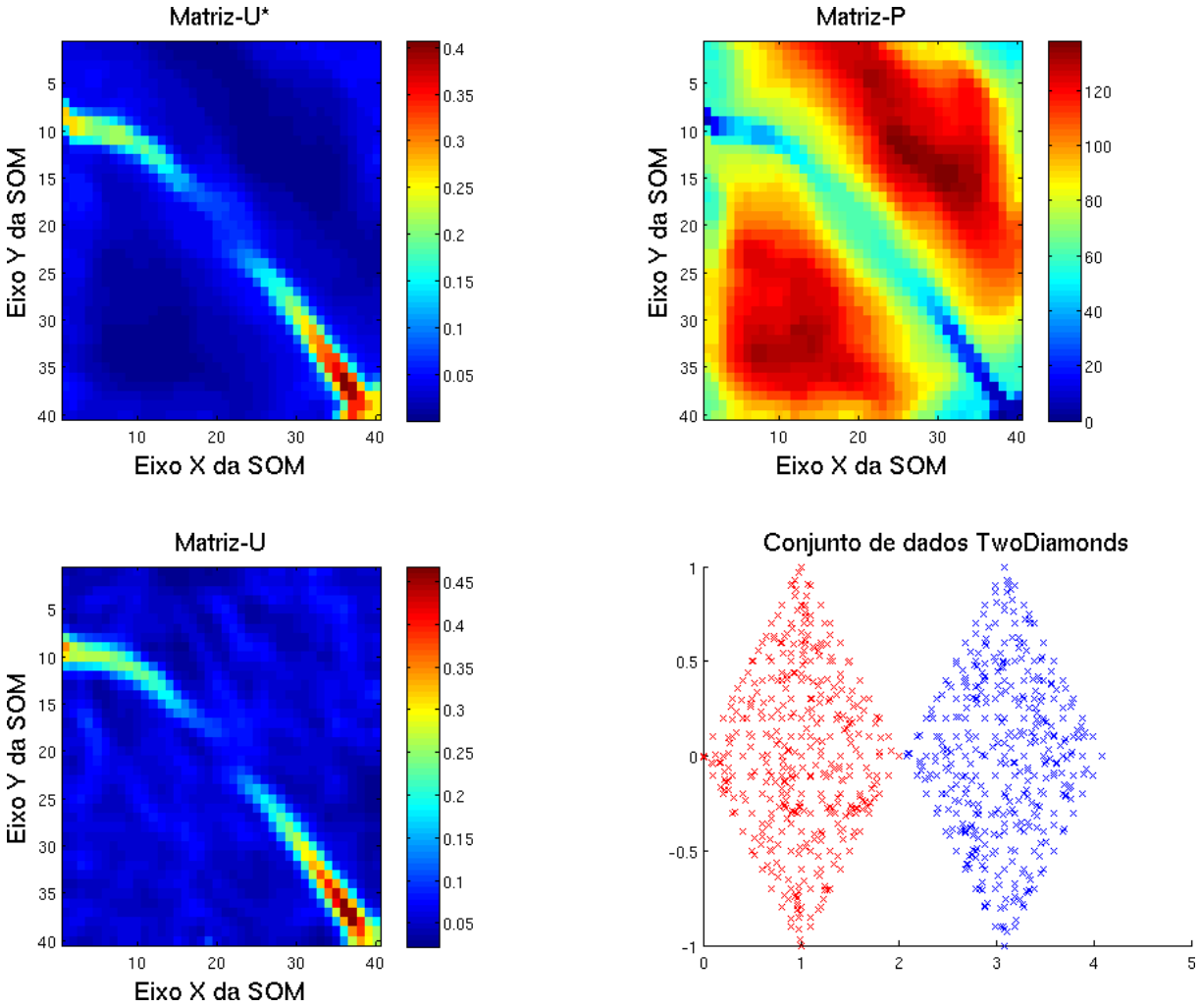


Figura 32: Matrizes intermediárias com raio max(Matriz-U) para o conjunto TwoDiamonds

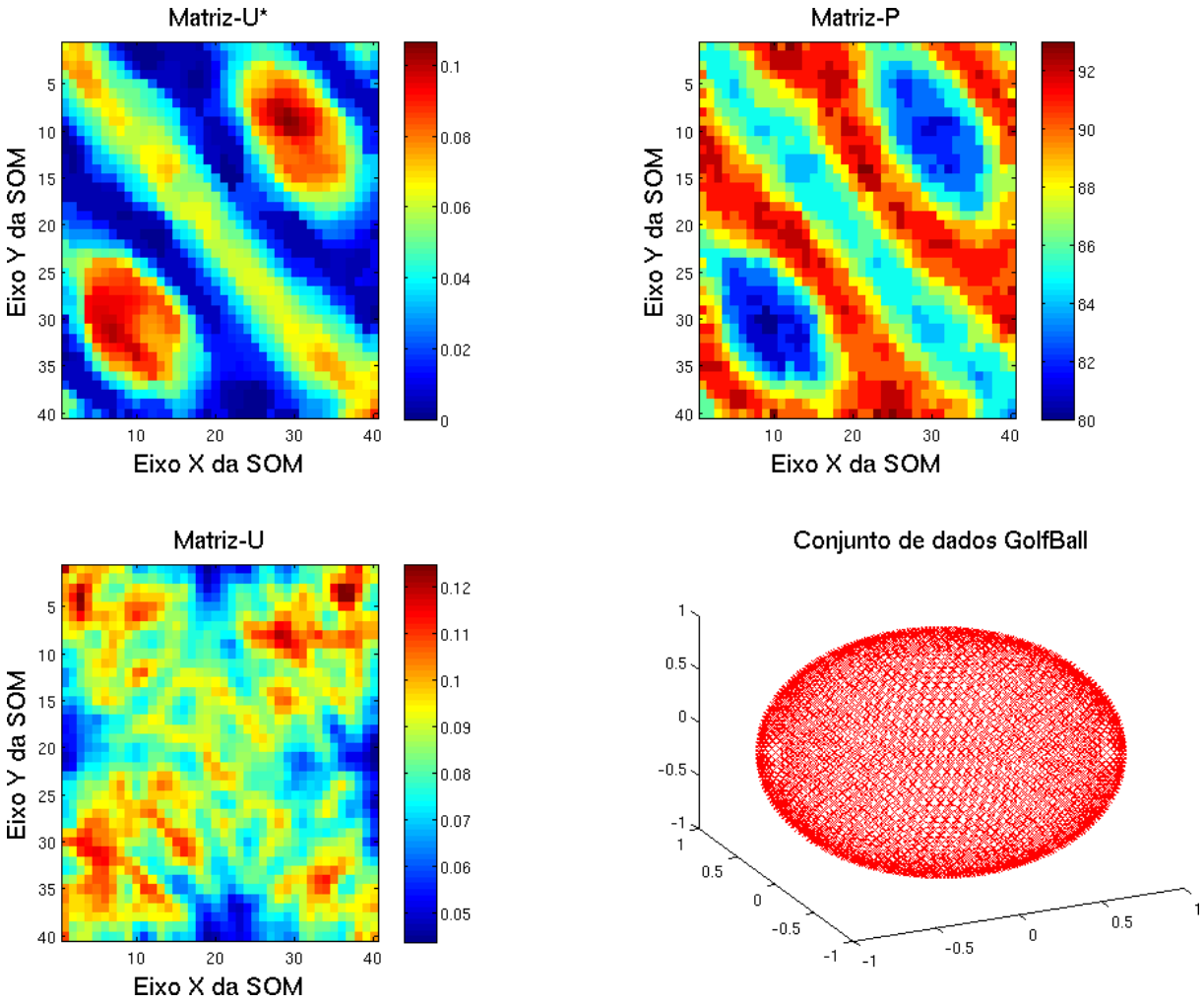


Figura 33: Matrizes intermediárias com raio de Pareto para o conjunto GolfBall

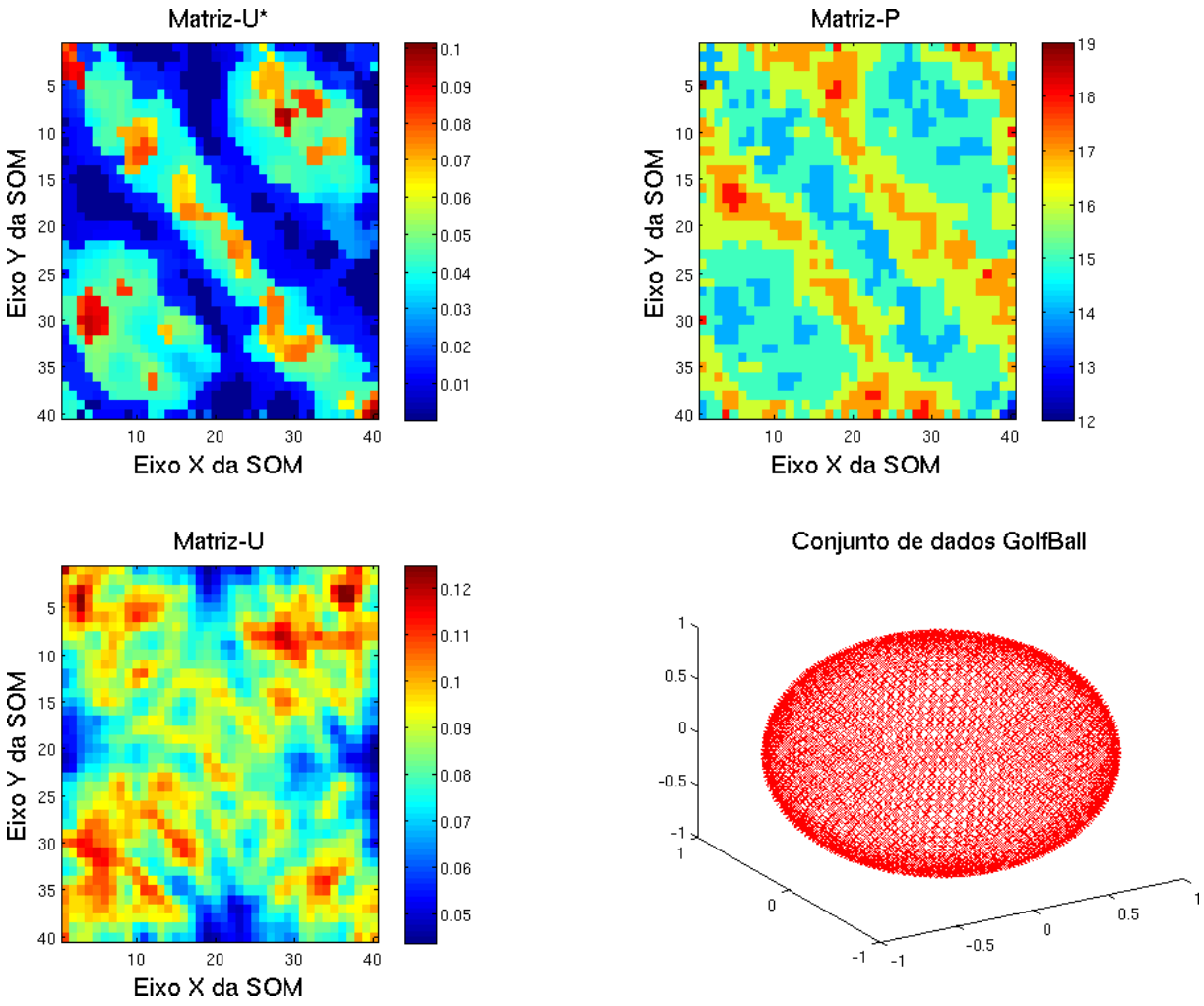


Figura 34: Matrizes intermediárias com raio max(Matriz-U) para o conjunto GolfBall