

Faculdade de Engenharia da Universidade do Porto



FEUP

Determinação de perfis de consumo baseada em mapas de Kohonen Modificados

Rui Manuel Proença Bidarra

VERSÃO PROVISÓRIA

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Eletrotécnica e de Computadores
Major Energia

Orientador: Prof. Dr. José Nuno Moura Marques Fidalgo

setembro 2012

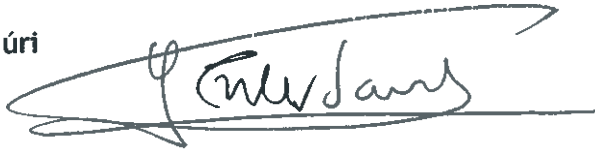
© Rui Bidarra, 2012

A Dissertação intitulada

“Determinação de Perfis de Consumo Baseada em Mapas de Kohonen Modificados”

foi aprovada em provas realizadas em 19-10-2012

o júri



Presidente **Professor Doutor José Eduardo Roque Neves dos Santos**
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto



Professora Doutora Fernanda de Oliveira Resende
Professora Auxiliar da Universidade Lusófona



Professor Doutor José Nuno Moura Marques Fidalgo
Professor Associado do Departamento de Engenharia Eletrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.



Autor - Rui Manuel Proença Bidarra

Faculdade de Engenharia da Universidade do Porto

Resumo

A determinação de perfis de consumo é um processo indispensável quer para planeamento e operação do sistema elétrico, quer no funcionamento do mercado de energia elétrica.

A obtenção dos perfis de consumo baseia-se habitualmente em algoritmos de *clustering* - agrupamento de consumidores de acordo com a similaridade de diagramas ou, alternativamente, na divisão apriorística em função dos dados de faturação (consumo, potência contratada, etc.)

Na primeira hipótese, as classes obtidas são coesas mas a classificação com base nas características de faturação é problemática. No segundo caso, a classificação é simples, mas as classes obtidas são demasiado heterogéneas.

Neste trabalho pretendeu-se implementar um algoritmo de *clustering* que permita obter classes coesas e que permita, simultaneamente, uma classificação fácil a partir dos dados de faturação. O processo implementado é baseado em Mapas de *Kohonen*, com a sua análise efetuada através do algoritmo *K-means*, modificados de modo a satisfazer os requisitos já referidos.

Mediante os resultados dos testes realizados, concluiu-se que a abordagem adotada permitiu obter classes características e facilmente identificáveis.

Palavras-chave: Perfis de consumo, Algoritmo de *clustering*, Mapas de Kohonen, SOM, *K-means*.

Abstract

The determination of consumption patterns is an essential process for both planning and operation of the electric system or the functioning of the electricity market.

The attainment of flow profiles is usually based on clustering algorithms - according to similar consumer group similarity diagrams or, alternatively, the division on the basis of a priori billing data (consumption, contracted power, etc.).

In the first case, the obtained classes are cohesive but the classification based on the characteristics of billing is problematic. In the second case, the classification is simple but the classes obtained are too heterogeneous.

In this work I do intended to implement a clustering algorithm that allows to obtain cohesive classes and that allows, simultaneously, an easy classification from billing data. The process is implemented based on Kohonen maps also known as a SOM (self-organizing map), with your analysis performed using the K-means algorithm, modified to meet the requirements mentioned above.

With the results obtained of tests performed, it was concluded that the approach adopted allowed us to obtain characteristics classes and easily identifiable.

Keywords: *Consumption patterns, Clustering algorithm, Kohonen Maps, SOM (self-organizing maps), K-means.*

Agradecimentos

Ao meu orientador, Professor Doutor José Nuno Moura Marques Fidalgo, por todo o seu apoio, disponibilidade e conselhos. A sua orientação foi essencial para a elaboração deste trabalho.

Ao meu pai Manuel Bidarra e mãe Maria da Graça pelo apoio incondicional que me deram ao longo da vida.

Ao meu irmão Eng.º Joaquim Bidarra pelo apoio e conselhos ao longo da vida e em particular ao longo da minha atividade académica.

Aos meus sobrinhos Diogo e Leonor, por serem uns miúdos fantásticos e com uma alegria contagiante.

Aos meus amigos e amigas pela paciência e apoio ao longo deste trabalho.

Índice

Resumo	iii
Abstract.....	v
Agradecimentos	vii
Índice.....	ix
Lista de figuras	xi
Lista de tabelas	xiv
Abreviaturas e Símbolos	xv
Capítulo 1	1
Introdução.....	1
1.1 - Objetivos da Dissertação	1
1.2 - Estrutura da Dissertação.....	2
1.3 - <i>Software</i> utilizado	2
Capítulo 2	3
Revisão Bibliográfica	3
2.1 - Perfis de consumo	3
2.2 - Algoritmos de <i>clustering</i>	4
2.3 - Estratégia de dois níveis de algoritmos de <i>clustering</i>	4
Capítulo 3	5
Técnicas e algoritmos utilizados	5
3.1 - Mapas de Kohonen	5
3.2 - K-means	7
3.3 - Índice de Davies-Bouldin	7
3.4 - SOM Toolbox	8
Capítulo 4	9
Metodologia adotada	9
4.1 - Introdução.....	9
4.2 - Tratamento de dados	9
4.3 - Uso da <i>SOM Toolbox</i>	10

Capítulo 5	18
Testes e análise de resultados	18
5.1 - Teste t1.....	18
5.2 - Teste t1n.....	24
5.3 - Teste t2.....	28
5.4 - Teste t2n.....	34
5.5 - Teste parte2.....	39
5.6 - Teste parte2n.....	45
Capítulo 6	53
Conclusão e trabalhos futuros.....	53
Referências	54
Anexos.....	56
Código fonte do teste t1:	56
Código fonte do teste t1n:.....	60
Código fonte do teste t2:	64
Código fonte do teste t2n:.....	68
Código fonte do teste parte2:	72
Código fonte do teste parte2n:	76
Código fonte da função normalizar:	80
Código fonte da função M_p2:	80
Código fonte da função M_p2_n:	85
Código fonte da função quad:.....	89

Lista de figuras

Figura 3-1 - Arranjo de neurónios do mapa e definição de variáveis [26].	6
Figura 3-2 - Dois exemplos de vizinhança topológica, a) retangular e b) hexagonal [27].	7
Figura 4-1 - Grelha de células de consumo versus potência contratada	12
Figura 4-2 - Plano <i>u-matrix</i> exemplo	14
Figura 4-3 - Código de cores (a) e projeção de componentes principais (b)	15
Figura 4-4 - Gráfico dos índices de Davies-Bouldin (a) e respetivos agrupamentos com menor índice (b)	16
Figura 4-5 - Matriz das distâncias (a) e protótipos (b)	17
Figura 5-1 - Matriz unificada das distâncias	19
Figura 5-2 - Código de cores (a) e projeção de componentes principais (b)	20
Figura 5-3 - Gráfico dos índices de Davies-Bouldin (a) e respetivo agrupamentos com menor índice (b)	21
Figura 5-4 - Matriz de distâncias (a) e Protótipos (b)	22
Figura 5-5 - Diagramas dos clientes colocados pela SOM na unidade 50 e o seu diagrama médio	23
Figura 5-6 - Gráfico com os perfis de consumo dos agrupamentos criados com <i>k-means</i> ($k=3$)	23
Figura 5-7 - Gráfico com o protótipo e diagrama médio da unidade 50 do SOM	24
Figura 5-8 - Matriz unificada das distâncias	25
Figura 5-9 - Código de cores (a) e projeção de componentes principais (b)	25
Figura 5-10 - Gráfico dos índices de <i>Davies-Bouldin</i> (a) e respetivo agrupamentos com menor índice (b)	26
Figura 5-11 - Matriz das distâncias (a) e Protótipos (b)	27
Figura 5-12 - Diagramas dos clientes colocados no agrupamento 50 e o seu diagrama médio	27

Figura 5-13 - Gráfico com os perfis de consumo dos agrupamentos criados com <i>k-means</i> ($k=9$)	28
Figura 5-14 - Gráfico com o protótipo e diagrama médio da unidade 50 do SOM.....	28
Figura 5-15 - Matriz unificada das distâncias	29
Figura 5-16 - Código de cores (a) e projeção de componentes principais (b)	30
Figura 5-17 - Gráfico dos índices de <i>Davies-Bouldin</i> (a) e respectivos agrupamentos com menor índice (b).....	31
Figura 5-18 - Matriz de distâncias (a) e Protótipos (b).....	32
Figura 5-19 - Diagramas dos clientes colocados na unidade 19 do SOM e o seu diagrama médio.....	33
Figura 5-20 - Gráfico com os perfis de consumo dos agrupamentos criados com <i>k-means</i> ($k=10$).....	33
Figura 5-21 - Gráfico com o protótipo e diagrama médio da unidade 19 do SOM.....	34
Figura 5-22 - Matriz unificada das distâncias	35
Figura 5-23 - Código de cores (a) e projeção de componentes principais (b)	35
Figura 5-24 - Gráfico dos índices de <i>Davies-Bouldin</i> (a) e respectivos agrupamentos com menor índice (b).....	36
Figura 5-25 - Matriz de distâncias (a) e Protótipos (b).....	37
Figura 5-26 - Diagramas dos clientes colocados na unidade 12 do SOM e o seu diagrama médio.....	37
Figura 5-27 - Gráfico com os perfis de consumo dos agrupamentos criados com <i>k-means</i> ($k=5$)	38
Figura 5-28 - Gráfico com o protótipo e diagrama médio da unidade 12 do SOM.....	38
Figura 5-29 - Matriz unificada das distâncias	40
Figura 5-30 - Código de cores (a) e projeção de componentes principais (b)	41
Figura 5-31 - Gráfico dos índices de <i>Davies-Bouldin</i> (a) e respetivo agrupamentos com menor índice (b).....	41
Figura 5-32 - Matriz de distâncias (a) e Protótipos (b).....	42
Figura 5-33 - Agrupamentos formados na grelha com os níveis de consumo e potência contratada, e quantidade de elementos em cada posição	43
Figura 5-34 - Diagramas dos estratos de consumo colocadas na unidade 1 do SOM e o seu diagrama médio	43
Figura 5-35 - Gráfico com os perfis de consumo criados com <i>k-means</i> ($k=2$)	44
Figura 5-36 - Gráfico com o protótipo e diagrama médio da unidade 1 do SOM	44
Figura 5-37 - Matriz unificada das distâncias	45

Figura 5-38 - Código de cores (a) e projeção de componentes principais (b).....	46
Figura 5-39 - Gráfico dos índices de Davies-Bouldin (a) e respetivo agrupamentos com menor índice (b).....	47
Figura 5-40 - Agrupamentos com $k=3$ (3 grupos)	48
Figura 5-41 - Matriz de distâncias (a) e Protótipos (b)	49
Figura 5-42 - Agrupamentos formados na grelha com os níveis de consumo e potência contratada, e quantidade de elementos em cada posição para $k=5$	50
Figura 5-43 - Agrupamentos formados na grelha com os níveis de consumo e potência contratada, e quantidade de elementos em cada posição para $k=3$	50
Figura 5-44 - Diagramas das divisões colocadas na unidade 3 do SOM e o seu diagrama médio.....	51
Figura 5-45 - Gráfico com os perfis de consumo criados com $k-means(k=5)$	51
Figura 5-46 - Gráfico com os perfis de consumo criados com $k-means(k=3)$	52
Figura 5-47 - Gráfico com o protótipo e diagrama médio da unidade 3 do SOM	52

Lista de tabelas

Tabela 4.1 - Valores da <i>u-matrix</i> exemplo	13
Tabela 5.2 - Níveis de consumo/Potência contratada	39

Abreviaturas e Símbolos

Lista de abreviaturas

BTN	Baixa Tensão Normal
DEEC	Departamento de Engenharia Eletrotécnica e de Computadores
FEUP	Faculdade de Engenharia da Universidade do Porto
SOM	<i>Self-Organizing Map</i>

Capítulo 1

Introdução

O planeamento e operação do sistema elétrico devem ser racionais e eficazes, apresentar uma boa qualidade, fiabilidade e rentabilidade, pois o mercado de energia elétrica é cada vez mais dinâmico e competitivo.

Nesta perspetiva, são necessários cada vez mais mecanismos para efetuar o planeamento e operar o sistema elétrico de energia. A determinação de perfis de consumo é um deles.

Neste processo é habitual usar-se algoritmos de *clustering* para obter os perfis de consumo-agrupamento de consumidores segundo a similaridade de diagramas. Em alternativa pode ser considerada a divisão apriorística em função das características de faturação, sejam eles o consumo, a potência contratada, o tipo de consumidor (doméstico, comercial, industrial, etc.) ou tipo de tarifa (simples, bi-horário, etc.). O tipo de utilizador ou tipo de tarifa nesta divisão apriorística, não é frequentemente utilizado, porque nem sempre existem diferenças significativas entre os vários tipos.

1.1 - Objetivos da Dissertação

Nesta dissertação pretende-se desenvolver uma metodologia, a utilizar para a determinação de perfis de consumo com base nos mapas de Kohonen¹, conjugando-a com outras técnicas de *clustering* para retirar alguma ambiguidade e ao mesmo tempo validar os resultados.

Numa segunda fase pretende-se evoluir da configuração inicial, modificando o uso deste algoritmo através de uma divisão apriorística dos consumidores em função dos dados de faturação. Desta forma pretende-se classificar os consumidores de uma forma mais conveniente, e mediante os dados que são conhecidos para todos os consumidores (dados de faturação).

¹ Primeiramente desenvolvido por Teuvo Kohonen. O algoritmo também é conhecido como um SOM (*self-organizing map*).

1.2 - Estrutura da Dissertação

Este primeiro capítulo apresenta o tema da Dissertação, bem como o seu enquadramento, motivação, objetivo e estrutura do trabalho.

No capítulo 2 é apresentada uma revisão bibliográfica.

No capítulo 3 é efetuado um resumo das bases teóricas do trabalho.

No capítulo 4 é descrita a metodologia seguida ao longo do trabalho.

No capítulo 5 apresenta-se os testes realizados e a análise dos resultados obtidos.

Na conclusão além de comentar os resultados, vai-se sugerir possíveis trabalhos futuros sobre o tema.

1.3 - *Software* utilizado

Este trabalho é todo ele de índole computacional pelo que foi necessário selecionar as ferramentas a utilizar para a sua elaboração.

Para correr o algoritmo de *clustering* foi selecionado o *Mathworks Matlab*. Trata se de um bom ambiente computacional embora as *toolboxes* de redes neurais baseadas em *SOM* incluídas neste *software* não sejam as mais atuais nem as mais documentadas, pelo que preferiu-se usar a *SOM Toolbox*².

Para o pré-processamento, filtragem e tratamento dos dados das amostras foi utilizado o *Microsoft Excel* bem como para realizar parte da análise dos dados resultantes dos Mapas de *Kohonen*.

² SOM Toolbox 2.0 © 2000-2005 por Esa Alhoniemi, Johan Himberg, Juha Parhankangas e Juha Vesanto.

Capítulo 2

Revisão Bibliográfica

Neste capítulo faz-se uma abordagem a trabalhos de outros investigadores sobre determinação de perfis de consumo, os diferentes métodos de lidar com este tema, as várias abordagens utilizadas e as conclusões obtidas nesses estudos.

2.1 - Perfis de consumo

Os perfis de consumo, sempre foram uma importante matéria de estudo para melhor gerir as redes de distribuição. Como o setor elétrico está a evoluir para uma maior liberalização e desregulamentação, verifica-se uma crescente necessidade de efetuar estudos nesta área, quer os operadores de sistema e quer os comercializadores necessitam de conhecer melhor o comportamento dos consumidores. No caso dos operadores de distribuição para uma melhor exploração da rede, e mesmo para gerir melhor o balanço procura-oferta em sistemas com crescentes fontes de produção intermitentes. No caso dos comercializadores, para conseguirem desenvolver contratos de comercialização de energia mais à medida das características dos consumidores que fazem parte da sua carteira de clientes. [1]-[5][6]

A obtenção dos perfis de consumo está normalmente dividida em três fases: Recolha de dados, definição de classes e testes de avaliação de performance [1]. A primeira fase compreende a recolha de amostras de dados na forma de um diagrama diário, que representam a evolução do consumo ao longo do dia. A atual estratégia energética da União Europeia promove a utilização de infraestruturas avançadas de medição para obter esta recolha de amostras de dados [7]. No entanto, a infraestruturas a implementar exige avultados investimentos, uma vez que é necessário instalar, entre outros dispositivos, contadores horários em todos os pontos de consumo.

Na segunda fase a forma mais comum de definir classes, consiste em defini-las antecipadamente com base em dados de faturação como potência contratada, consumo de energia, ou tipo de consumidor (doméstico, industrial, comercial, etc). Mas desta forma podemos obter uma enorme quantidade de classes porque a definição de classes não

considera a forma dos diagramas [8][9]. Daí a existência de várias tentativas em definir essas classes, usando algoritmos de *clustering*, estas tentativas são efetuadas com base em amostras de diagramas de perfis de consumo [1][2][4][5][10][11][12]. Este tipo de tentativas conduz a boas classes protótipo, mas até agora ainda não é conhecido nenhum método padrão que as consiga obter a partir dos dados de faturação [1].

2.2 - Algoritmos de *clustering*

Quando se tenta retirar informação de um conjunto de dados, um dos grandes desafios é identificar grupos semelhantes, ou seja *clustering*. Existem vários métodos que podem ser usados para identificar e analisar grupos semelhantes, desde os métodos clássicos de *clustering e técnicas estatísticas* [13]-[18], *data mining* [19][20], mapas de Kohonen [13][14][16][21], redes neuronais [22][23], e lógica difusa [6][16][17][22]-[24].

Assim uma decisão deve ser feita para escolher o método de *clustering* com melhor performance. Considerando os algoritmos *k-means*, SOM, e a estratégia de dois níveis de algoritmos de *clustering* verifica-se que o *k-means* tem uma melhor performance que o SOM em algumas aplicações, mas usando uma estratégia de dois níveis de algoritmos de *clustering* apresenta ainda uma melhor performance, o que indica que a combinação de *k-means* com o SOM seja uma boa prática para melhorar a performance do SOM [2].

2.3 - Estratégia de dois níveis de algoritmos de *clustering*

Quando o SOM é usado como algoritmo de *clustering* cada unidade do mapa define um pequeno grupo de amostras do seu diagrama de Voronoi [25]. O vetor protótipo de cada unidade representa o centroide do grupo. Por causa das relações de vizinhança, os protótipos vizinhos são puxados na mesma direção, e portanto os vetores protótipo das unidades vizinhas assemelham-se. Mas a situação ótima do SOM não é quando o número de protótipos é igual ao número de agrupamentos, mas sim quando o número de unidades na grelha de saída é muito maior do que o número de agrupamentos esperados. Através da definição da função de vizinhança é possível extrair unidades vizinhas em conjunto, e assim as unidades vizinhas terem as mesmas propriedades de um só agrupamento. A transição de um agrupamento para outro ocorre ao longo de várias unidades do mapa. Se for pretendido um número pequeno de agrupamentos, para evitar a redução da grelha a uma pequena grelha de unidades do mapa, as unidades do SOM devem ser agrupadas [26].

Assim se tivermos uma estratégia de dois níveis de *clustering*, o SOM é usado no primeiro nível, implementando uma redução ordenada dimensional do mapa, e no segundo nível o *k-means* partindo da redução do SOM, cria um mapa com um *k* número de agrupamentos de menor dimensão que as unidades do SOM [2][11][12].

Capítulo 3

Técnicas e algoritmos utilizados

Neste capítulo é apresentada a informação recolhida sobre as várias técnicas, métodos, e ferramentas utilizados ao longo do trabalho, começando pelos mapas de Kohonen, de seguida o método de *clustering K-means*, o índice Davies-Bouldin e finalmente a *SOM Toolbox*.

3.1 - Mapas de Kohonen

Os mapas de Kohonen também conhecidos por *SOM* ou ainda como Redes de Kohonen, são um método computacional muito útil para visualização de dados de grande dimensão. Na sua forma mais básica produz um gráfico de semelhança dos dados de entrada. Converte uma relação estatística não linear entre dados de elevada dimensão numa simples relação geográfica de pontos exibidos numa baixa dimensão, geralmente numa grelha bidimensional de nós. Resumindo, compacta a informação ao mesmo tempo que preserva as importantes relações topológicas e métricas dos dados de entrada na exibição, isto de uma forma, que se pode dizer, abstrata. Por esta razão este método permite uma vasta utilização, desde análise de processos, controlo, perceção de maquinaria, e comunicações [27].

A *SOM* pode ainda ser descrita como uma rede neuronal artificial não supervisionada que produz uma representação de baixa dimensão, normalmente a duas dimensões, mas que preserva as propriedades topológicas dos dados de entrada, no que resulta o que se chama mapa ou rede.

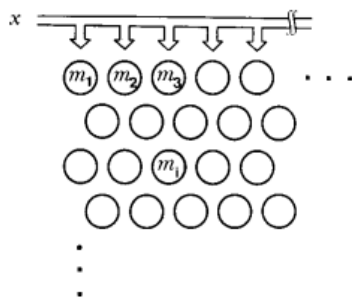


Figura 3-1 - Arranjo de neurónios do mapa e definição de variáveis [27].

Segue-se uma demonstração do modo como o algoritmo escolhe o neurónio com melhor correspondência, na sua forma mais simples, com inicialização aleatória. Considerando uma rede bidimensional de neurónios como a representada na Figura 3-1, a sua topologia pode assumir diversas formas tais como as da Figura 3-2. Para cada nó i , existe um vetor paramétrico modelo, também chamado de vetor referência $\mu_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T \in \mathfrak{R}^n$, que tem de ser inicializado antes do processo recursivo. Considerando também $x = [x_1, x_2, \dots, x_n]^T \in \mathfrak{R}^n$ como um vetor de entrada estocástico, que está ligado em paralelo com todos os neurónios da rede, via um peso escalar variável μ_{ij} , que normalmente é diferente para cada neurónio. Numa forma algo abstrata pode-se dizer que a entrada x é comparada com todos os μ_i e a localização da melhor correspondência é definida como a resposta. Esta comparação é feita pela distância euclidiana $\|x - \mu_i\|$ e a menor é definida como o nó com melhor correspondência, que se designará por $c = \operatorname{argmin}\{\|x - \mu_i\|\}$. Durante a aprendizagem os nós que estão topograficamente perto até uma determinada distância geométrica vão ativar os outros à sua volta para aprenderem algo com a entrada x . Isto vai provocar um efeito de aproximação dos seus vetores modelo, que numa aprendizagem contínua resulta na ordenação global. Considerando um eventual limite de convergência do processo de aprendizagem como $\mu_i(t+1) = \mu_i(t) + h_{ci}(t)[x(t) - \mu_i(t)]$, onde $\mu_i(0)$ é um valor aleatório inicial e t é o índice de iteração com valor inteiro crescente, e $h_{ci}(t)$ é a função vizinhança que tende para zero quando t tende para infinito, para que haja convergência. Consideram-se habitualmente duas formas simples para esta função. A mais simples é considerar a vizinhança como um conjunto de pontos de forma regular à volta do nó c , como na Figura 3-2. Neste caso representando o seu índice como N_c .

Pelo que $h_{ci}(t) = \alpha(t)$ se $i \in N_c$ e $h_{ci}(t) = 0$ se $i \notin N_c$. O valor de $\alpha(t)$ é o fator taxa de aprendizagem ($0 < \alpha(t) < 1$). Ambas $\alpha(t)$ e o raio de $N_c(t)$ decrescem monotonamente no tempo.

Outra forma extensamente aplicada é escrever esta função em termos de função gaussiana,

$$h_{ci} = \alpha(t) \cdot \exp\left(-\frac{\|r_c - r_i\|}{2\sigma^2(t)}\right) \quad (2.1)$$

onde $\alpha(t)$ é um valor escalar do fator taxa de aprendizagem, e $\sigma(t)$ define a largura da vizinhança que vai mais tarde corresponder a $N_c(t)$, com ambas $\alpha(t)$ e $\sigma(t)$ são decrescentes monotonamente no tempo [27].

Esta é uma das muitas formas possíveis para este algoritmo, pretendendo-se demonstrar aqui a mecânica do algoritmo.

Há que referir ainda que neste algoritmo não existe uma forma formal de definir o erro, isto principalmente pelo fato de que o treino é não supervisionado.

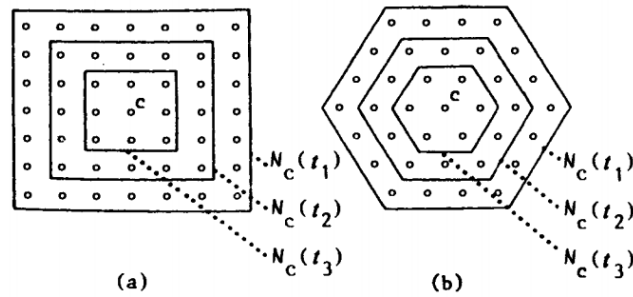


Figura 3-2 - Dois exemplos de vizinhança topológica, a) retangular e b) hexagonal [27].

3.2 - K-means

Este método *K-means* é um dos mais simples algoritmos de aprendizagem não supervisionada. Este algoritmo funciona da seguinte forma, inicialmente o utilizador escolhe um número K de unidades desejadas, isto é, o número de agrupamentos pretendido. Cada valor de entrada é então atribuído à unidade mais próxima e cada conjunto de valores de entrada é um agrupamento. A unidade de cada agrupamento é atualizada com base no conjunto de valores de entrada a ele atribuída, repetindo-se a atribuição e os passos de atualização até que nenhum valor de entrada mude de agrupamento [28].

Na *SOM Toolbox* existem funções que aplicam este método.

3.3 - Índice de Davies-Bouldin

O índice de Davies-Bouldin foi introduzido pela primeira vez por David L. Davies e Donald W. Bouldin em 1979. É uma medida que se baseia na ideia de identificar numa partição dos elementos analisados, conjuntos de agrupamentos que sejam compactos no seu interior e ao mesmo tempo tenham uma boa separação entre os vários agrupamentos.[29]

Com isto em mente é necessário definir uma medição geral da separação dos agrupamentos, que permita o cálculo da semelhança média de cada grupo com o seu agrupamento mais semelhante.[30]

Passando agora à sua formulação matemática:

$$DB_m = \frac{1}{m} \sum_{i=1}^m \max_{j=1, \dots, m, j \neq i} \frac{S_i + S_j}{d_{ij}} \quad (2.2)$$

onde DB_i é o índice de Davies-Bouldin, S_i é a medição da dispersão do agrupamento C_i , S_j a medição da dispersão do agrupamento C_j , e d_{ij} a separação entre os dois agrupamentos $d_{ij}=d(C_i, C_j)$, calculada por uma medida de distância, que na sua versão mais simples será Euclidiana.

Considerando $R_{ij} = \frac{S_i + S_j}{d_{ij}}$, R_{ij} deve ser definido de forma a seguir as seguintes condições:

$$R_{ij} > 0. \quad (2.3)$$

$$R_{ij} = R_{ji}. \quad (2.4)$$

$$\text{Se } s_i = 0 \text{ e } s_j = 0 \text{ então } R_{ij} = 0. \quad (2.5)$$

Técnicas e algoritmos utilizados

$$\text{Se } s_j > s_k \text{ e } d_{ij} = d_{ik} \text{ então } R_{ij} > R_{ik}. \quad (2.6)$$

$$\text{Se } s_j = s_k \text{ e } d_{ij} < d_{ik} \text{ então } R_{ij} > R_{ik}. \quad (2.7)$$

Ou seja R_{ij} é não negativo e simétrico.[31]

Este índice está implementado em diversas funções da *SOM Toolbox*.

3.4 - SOM Toolbox

A *SOM Toolbox* nasceu da necessidade de uma boa implementação e de fácil uso em Matlab para fins de investigação.[32]

Esta *Toolbox*³ pode ser usada para inicializar, treinar, visualizar e analisar o *SOM*.

Ela foi desenvolvida para ter uma grande flexibilidade, como pode ser evidenciado, por exemplo, com a função *som_make*, a qual pode ser utilizada com uma enorme variedade de argumentos, desde tipo de inicialização, tipo e duração do treino, tamanho e forma do mapa/rede, forma de entrelaçar, quantidade de informação a reportar, etc.

O mesmo pode ser dito em relação às funções de visualização e análise de dados. Mais pormenores sobre este aspeto serão abordados mais adiante na descrição da metodologia usada no trabalho.

³ Daqui em diante vou passar a referir-me assim em relação à *SOM Toolbox*.

Capítulo 4

Metodologia adotada

4.1 - Introdução

A metodologia adotada foi a de seguir uma sequência de passos passando por dois níveis de algoritmos de *clustering*, até à fase final em que se obtenha uma classificação baseada em dados de faturação.

Como foi referido anteriormente a *Toolbox* é muito flexível. Para entender o seu funcionamento e flexibilidade, além de ler o manual e tutorial começou-se por utilizar as demonstrações das suas funções que estão disponíveis na própria *Toolbox*, de nome *som_demo1*, *som_demo2*, *som_demo3*, *som_demo4*. Elas constituem um resumo bastante alargado das funções da *Toolbox* quer em termos de construção dos mapas de Kohonen, quer em termos de visualização e análise.

No seguimento disto, e depois de entender o funcionamento da *Toolbox* passou-se ao trabalho efetivamente pretendido.

Para tal foram necessários dados reais, pois sem eles este estudo perdia a credibilidade exigida. Os dados disponibilizados são medições de um conjunto de consumidores BTN ao longo de vários meses, com informação suficiente para desenhar os seus perfis de consumo, e também as características de consumo e potência contratada de cada consumidor.

Antes de serem utilizados estes dados tiveram de ser tratados e convertidos num formato compatível com a *Toolbox*, só depois passando para testes de diversa ordem.

4.2 - Tratamento de dados

No tratamento de dados começou-se por selecionar só dias úteis, porque são a maioria dos dias do ano, e como tal têm uma boa representatividade. Também foi escolhido só o mês de novembro para limitar o número de amostras a usar. Estas filtragens permitem avaliar o processo numa forma geral com um conjunto simplificado. Depois de selecionados, verificou-

se a existência de possíveis erros de medição ou registo. Como por exemplo o caso do consumo e potência contratada não compatíveis (consumo superior ao possível com a respetiva potência contratada), foram também retiradas as medições com valores nulos (zero).

Este tratamento foi efetuado no *Microsoft Excel*.

4.3 - Uso da *SOM Toolbox*

Para iniciar o uso da *Toolbox* foi necessário converter os dados previamente tratados, num formato compatível.

Nesta fase do trabalho, foram realizados diversos testes com os dados de amostra, variando vários parâmetros das funções da *Toolbox*, quer nas funções de construção do mapa de *Kohonen*, quer nas funções de visualização e análise. Houve também necessidade de programar de raiz várias funções.

De seguida são descritas as funções utilizadas da *Toolbox*, funções implementadas para o trabalho e os testes com maior relevância que foram realizados.

4.3.1. Descrição das principais funções da *Toolbox*

Para se explicar o funcionamento da *Toolbox*, vai-se passar à descrição das principais funções da *Toolbox* usadas neste trabalho, mas focando esta descrição apenas nos aspetos mais pertinentes.

As principais funções da *Toolbox* dividem-se em vários grupos.

Do grupo de funções de conversão de estruturas e leitura/escrita de ficheiros, utilizou-se a função *som_read_data* para ler os dados previamente guardados num ficheiro de dados *ASCII*, que são os dados recolhidos e tratados previamente, e ao mesmo tempo criar uma estrutura de dados necessária para a construção do mapa de *Kohonen*.

No grupo de inicialização e treino foi utilizada a função *som_make* para a partir da estrutura de dados resultante da *som_read_data* inicializar, treinar e criar o mapa de *Kohonen*.

Já no grupo de visualização foi necessário utilizar as funções seguintes, *som_show*, *som_grid* e *som_cplane* e *som_plotplane*.

A *som_show* é usada para produzir as mais básicas visualizações do mapa de *Kohonen*, tais como a *u-matrix*, gráficos de cada componente do mapa, representações de agrupamentos ou componentes do mapa através de cores, ou ainda como base para outras visualizações.

Através da função *som_grid* visualiza-se os conjuntos de unidades do mapa e as relações topológicas entre elas.

A *som_cplane* é uma função que permite acrescentar mais informação às figuras obtidas por outras funções de visualização.

A função *som_plotplane* produz gráficos de linhas referentes aos valores das unidades do mapa.

Com estas funções consegue-se produzir a maioria dos gráficos mais tarde utilizados para analisar os resultados obtidos.

Do grupo de algoritmos de agrupamentos foi usada a função *kmeans_clusters* para a partir do mapa de *Kohonen* já obtido pela função *som_make*, criar agrupamentos com o algoritmo *k-means*. Esta função funciona de forma a correr o algoritmo *k-means* várias vezes para

diferentes valores de k , com k a representar o número de agrupamentos, e selecionar o melhor k com base no índice de Davies-Bouldin calculado para cada k .

No grupo de funções auxiliares de visualização foram usadas as funções *som_hits*, *som_umat*, *pcaproj* e *som_colorcode*.

A função *som_hits* calcula o número de ocorrências. A função *som_umat* computa a matriz unificada das distâncias da SOM. A função *pcaproj* projeta vetores de dados usando a análise de componentes principais. Por último a função *som_colorcode* calcula uma heurística de codificação de cores para a função *som_grid*.

No grupo de funções de algoritmos de agrupamentos utilizou-se a função *kmeans_clusters*. Esta função usa o algoritmo *k-means* para a partir de um conjunto de dados, testar e a avaliar vários k através do índice de Davies-Bouldin, com k a representar o número de agrupamentos.

4.3.2. Descrição das funções implementadas

Depois de analisar as funções da *Toolbox* chegou-se à conclusão que era necessário implementar funções complementares para realizar o trabalho da forma pretendida.

Embora a *Toolbox* tenha uma função que normaliza os valores de entrada, esta normaliza os valores em relação ao máximo da matriz de valores de entrada, mas o que era pretendido para este trabalho era uma normalização que separe os diagramas pela forma, ou seja, de modo a que todos os diagramas de entrada resultem com escalas semelhantes. Posto isto a primeira função implementada foi uma função que normaliza-se os valores de cada cliente de forma individual, a esta função foi dado o nome de **normalizar**.

Para analisar melhor os resultados achou-se por bem extrair alguns valores intermédios e finais dos testes, por isso foi também necessário implementar várias funções para tornar acessíveis os valores dos diagramas dos protótipos usados pela SOM, os diagramas médios dos agrupamentos criados pela SOM e também os diagramas médios dos agrupamentos criados pelo algoritmo *k-means*. Estas funções guardam os resultados em folhas de cálculo, e a partir delas consegue-se comparar graficamente os diversos valores de forma mais fácil.

Na segunda parte do trabalho foi efetuada uma divisão apriorística dos dados de entrada do algoritmo em função dos dados de faturação, para fazer esta divisão foi implementada uma função que separa os dados dos clientes em vários níveis de consumo e potência contratada, a que vou chamar de células, depois de separados os clientes, esta função calcula o valor médio do diagrama de cada célula, estes diagramas médios são os dados de entrada na SOM desta parte. Esta função tem duas variantes a que foram dados os nomes **M_p2** e **M_p2_n**, esta ultima utiliza os dados normalizados de cada cliente, para que os diagramas de cada cliente resultem com escalas semelhantes, tal como na função **normalizar**. A grelha de células resultante é como a que está representada na figura 4-1, em que e_1 , e_2 , e_3 e e_4 representam os níveis de energia consumida e pc_1 , pc_2 , pc_3 e pc_4 representam os níveis de potência contratada, os valores de cada nível estão representados no capítulo 5 na tabela 5.2.

e4				
e3				
e2				
e1				
	PC1	PC2	PC3	PC4

Figura 4-1 - Grelha de células de consumo versus potência contratada

Para mais fácil interpretação dos agrupamentos resultantes na parte 2, foi implementada uma função que produz duas grelhas semelhante à da figura 4-1, uma preenchida com cores, a que cada cor corresponde um agrupamento, e outra que contabiliza o número de clientes atribuídos a cada célula, a essa função foi atribuído o nome **quad**.

4.3.3. Descrição dos testes efetuados

Os testes efetuados, que têm maior importância para o presente trabalho, dividem-se em duas partes. Na primeira são testes realizados só com dados do perfil de consumo. Na segunda os dados foram reorganizados de modo a proporcionar uma separação evidente com base nos dados de faturação, consumo e potência contratada.

O nome dado aos testes da primeira parte são **t1**, **t1n**, **t2** e **t2n**.

Na segunda parte os nomes são, **parte2** e **parte2n**.

Todos estes testes seguem uma sequência semelhante, embora com algumas diferenças, que se apresentam a seguir.

Todos os testes usam a função *som_make*, mas no caso do **t1**, **t1n**, **parte2** e **parte2n**, os parâmetros usados deixam que a função calcule automaticamente o número de neurónios e consequentemente o número de unidades do mapa. Nos restantes testes **t2** e **t2n**, esta função utiliza um parâmetro que possibilita escolher o número máximo de neurónios.

Os testes, **t1n**, **t2n**, e **parte2n**, são testes em que os valores de entrada são normalizados, para tal foi utilizado a função *normalizar* para **t1n** e **t2n** ou *M_p2_n* para **parte2n**.

Todos os testes usam a função *kmeans_clusters* que tem neste caso como dados de entrada o valores de saída da função *som_make*.

Quanto à visualização os todos testes têm a maioria das funções em comum. A única diferença é nos testes da parte 2, em que foi necessário acrescentar outras formas de visualização.

Em termos de resultado do SOM, a principal figura é o da *u-matrix*, pois é em torno da *u-matrix* que se produz a maioria das restantes figuras.

A *u-matrix* é uma matriz unificada de distâncias, através dela consegue-se visualizar a distância entre vizinhanças das unidades do mapa, ou seja, com ela consegue-se distinguir os agrupamentos no mapa. Os valores baixos indicam os agrupamentos em si e os valores altos as fronteiras dos agrupamentos. Esta matriz é de maior dimensão do que o mapa. Considerando o mapa de dimensão $[n \times m]$ a *u-matrix* é de dimensão $[(n*2-1) \times (m*2-1)]$. Num mapa com a dimensão 5×1 teríamos as unidades do mapa $m(1)$, $m(2)$, $m(3)$, $m(4)$, $m(5)$. Por seu lado uma *u-matrix* de dimensão 9×1 seria $u(1)$, $u(1,2)$, $u(2)$, $u(2,3)$, $u(3)$, $u(3,4)$, $u(4)$, $u(4,5)$, $u(5)$, em que $u(i,j)$ é a distância entre unidades do mapa $m(i)$ e $m(j)$ e o $u(k)$ é o valor médio dos valores circundantes. Como por exemplo o $u(4) = (u(3,4) + u(4,5))/2 = (0,420219 + 0,334717)/2 = 0,377468$, como pode se visto na tabela 4.1, em que os valores apresentados são adimensionais. O plano *u-matrix* deste exemplo é representado na figura 4-2, nesta figura os valores apresentados são adimensionais.

Tabela 4.1 - Valores da *u-matrix* exemplo

$u(1)$	0,278844
$u(1,2)$	0,278844
$u(2)$	0,338026
$u(2,3)$	0,397207
$u(3)$	0,408713
$u(3,4)$	0,420219
$u(4)$	0,377468
$u(4,5)$	0,334717
$u(5)$	0,334717

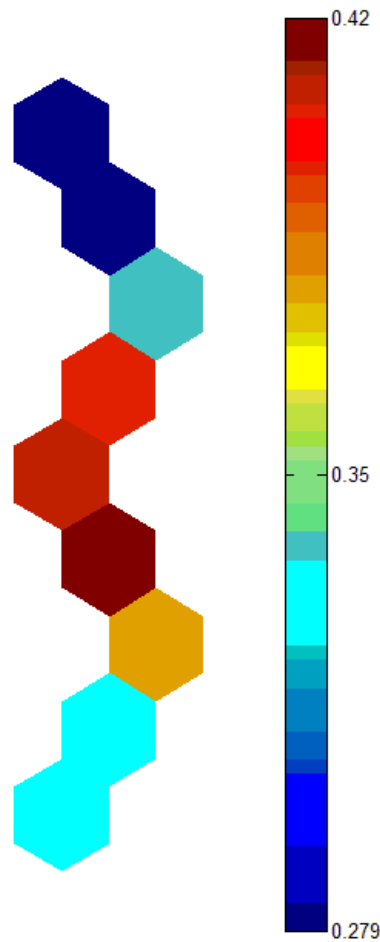


Figura 4-2 - Plano *u-matrix* exemplo

Na figura 4-3 (a) pode ser vista a gelha com o nome código de cores onde é efetuada a contagem de quantidade de clientes em cada unidade do mapa, obtidos pela função *som_hits*, mas também o código de cores que é a conjugação da matriz de cores obtida pela função *som_colorcode* com a matriz das distâncias calculada com valores retirados da *u-matrix*, precisamente os valores $u(i,j)$, mas transformados em percentagem.

Na figura 4-3 (b) com o nome projeção de componentes principais pode-se observar a representação gráfica da matriz das componentes principais obtidas pela função *pcaproj*, em que as cruces representam as projeções dos dados iniciais e os pontos coloridos com o código de cores usado anteriormente na grelha código de cores, representam as projeções dos dados do mapa de Kohonen, estas projeções funcionam como uma correlação entre variáveis. O gráfico desta figura é adimensional, com o eixo vertical a representar os valores próprios e o eixo horizontal a representar os componentes principais.

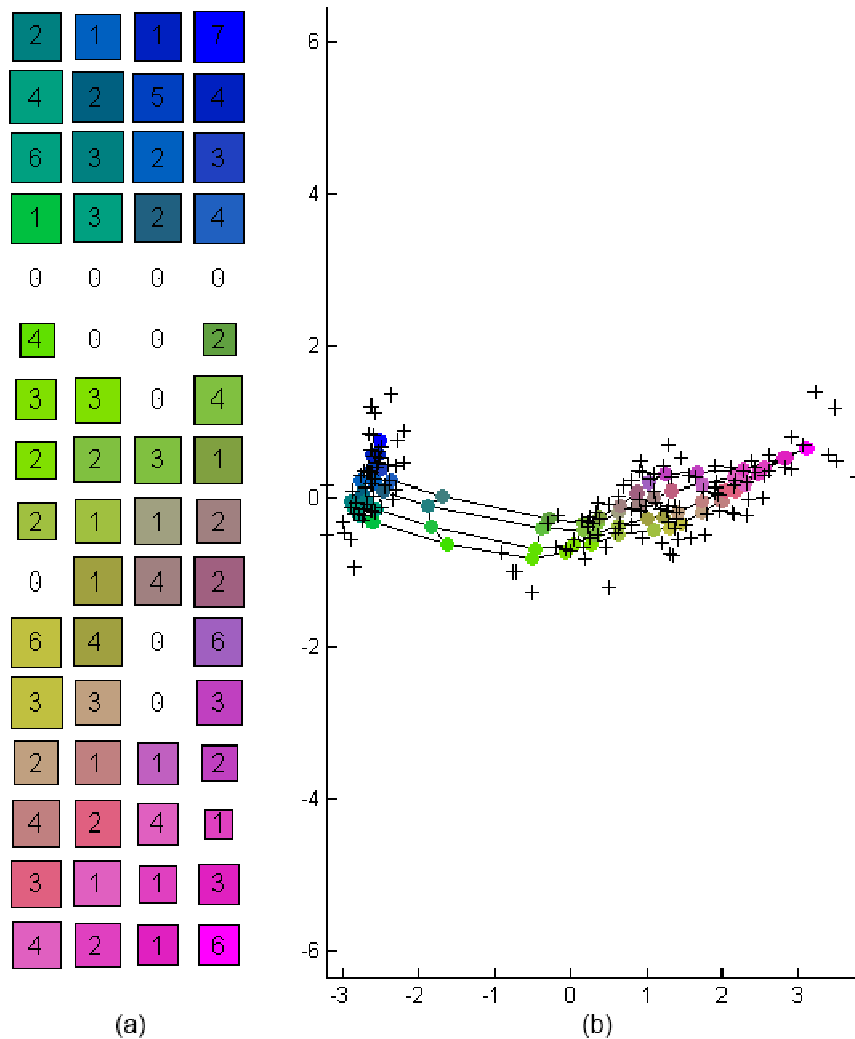


Figura 4-3 - Código de cores (a) e projeção de componentes principais (b)

Na figura 4-4 (a) pode ser visto o gráfico com o nome índice de Davies-Bouldin que é produzido com os valores da avaliação efetuada pela função *kmeans_clusters*, estes valores são adimensionais, em que o eixo vertical representa o valor do índice e o eixo horizontal representa o número de agrupamentos *k*. Na figura 4-4 (b) com o nome agrupamentos com o menor índice de *Davies-Bouldin*, mostra-se o número de agrupamentos resultantes com o menor índice de *Davies-Bouldin*, em que cada cor é um agrupamento.

Nos testes da parte 2 é acrescentada a identificação da célula correspondente a cada unidade do mapa, à figura agrupamentos com menor índice, na forma (i,j), em que *i* é o nível de consumo e *j* o nível de potência contratada.

Metodologia adotada

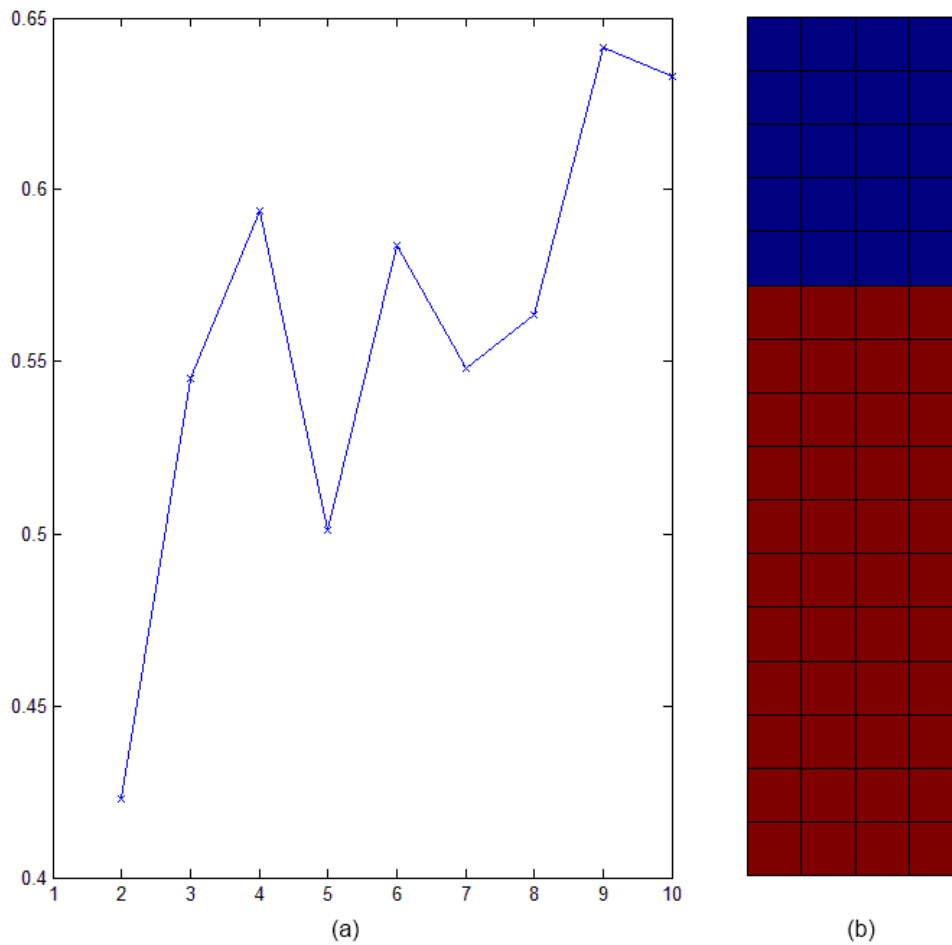


Figura 4-4 - Gráfico dos índices de Davies-Bouldin (a) e respectivos agrupamentos com menor índice (b)

A figura 4-5 (a) com nome matriz das distâncias é a visualização da matriz de distâncias na sua forma mais simples, que representa a distância entre unidades do mapa, útil para perceber a coesão dos agrupamentos.

Quanto à figura 4-5 (b) com o nome de protótipos, representa o desenho numa grelha dos vetores paramétricos modelo, usados pelo mapa de Kohonen, também chamados de vetores referência.

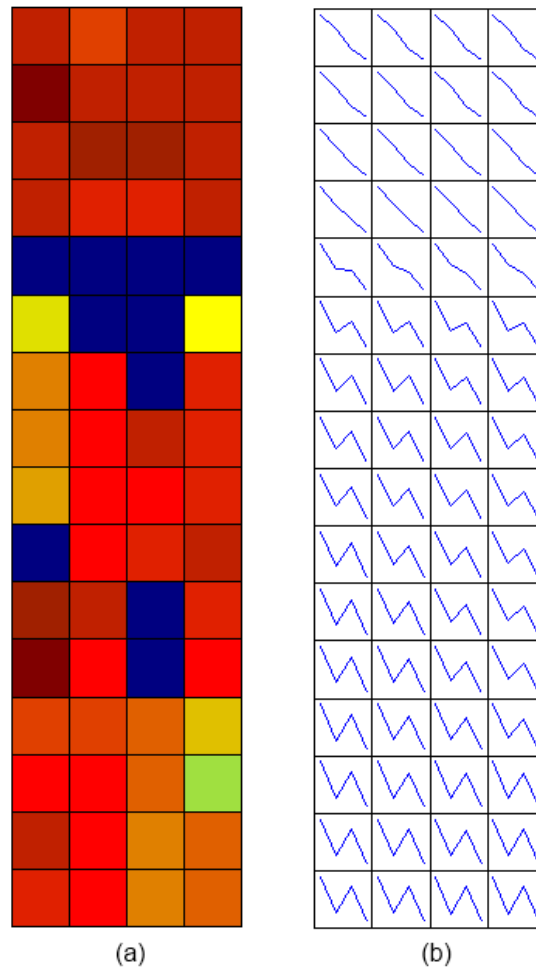


Figura 4-5 - Matriz das distâncias (a) e protótipos (b)

As figuras 4-3, 4-4 e 4-5 foram produzidas com um mapa de Kohonen obtido com os dados do ficheiro *iris.data*⁴.

Na parte 2 para ajudar na análise do resultado utilizou-se a função **quad** para produzir duas grelhas, uma com o número de agrupamentos e outra com o número de clientes em cada célula.

⁴ Ficheiro exemplo do conhecido conjunto de dados de flores iris, em que se tenta classificar as flores com base no comprimento das pétalas e das sépalas

Capítulo 5

Testes e análise de resultados

A principal função do algoritmo SOM é criar visualizações simples de dados de grande dimensão, como tal neste capítulo apresenta-se as figuras obtidas em cada um dos testes efetuados, seguida de uma análise dessas figuras.

Também são introduzidos e analisados os gráficos obtidos com valores dos diagramas dos protótipos usados pela SOM, os diagramas médios dos agrupamentos criados pela SOM e também os diagramas médios dos agrupamentos criados pelo algoritmo *k-means*, retirados do algoritmo SOM para folhas de cálculo pelas funções implementadas para esse efeito.

5.1 - Teste t1

Seguindo a metodologia já descrita, este teste produziu as figuras 5-1 até 5-7.

Resumo dos parâmetros e condições considerados neste teste:

- Diagramas não normalizados.
- Função *som_make* com o parâmetro tamanho do mapa não definido.

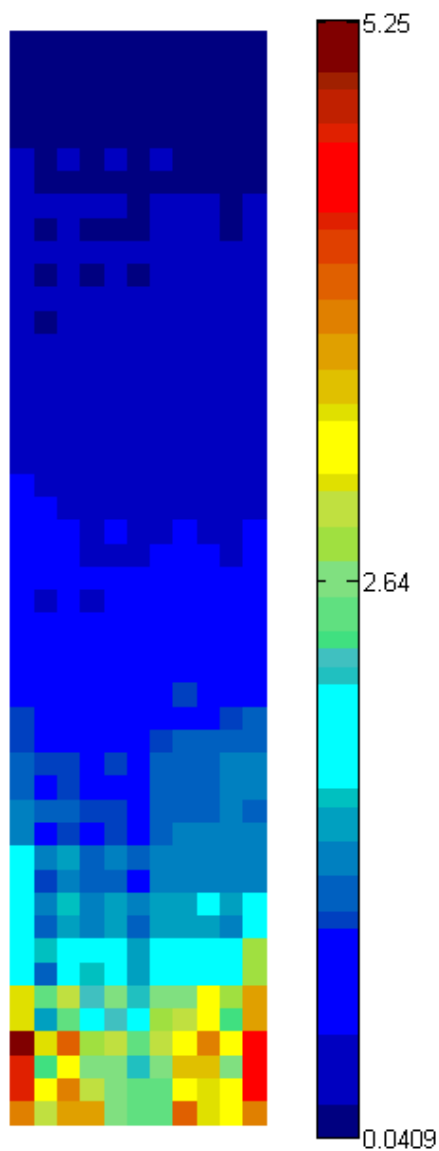


Figura 5-1 - Matriz unificada das distâncias

Na figura 5-1, verificamos desde logo grandes blocos de unidades do mapa que estão muito próximos uns dos outros. Onde está a cor azul escura, valores baixos, as unidades estão muito coesas. Só no fundo da figura quando aparecem as cores muito misturadas é que estamos na presença de unidades pouco coesas. Mas nas restantes figuras obtemos mais informação.

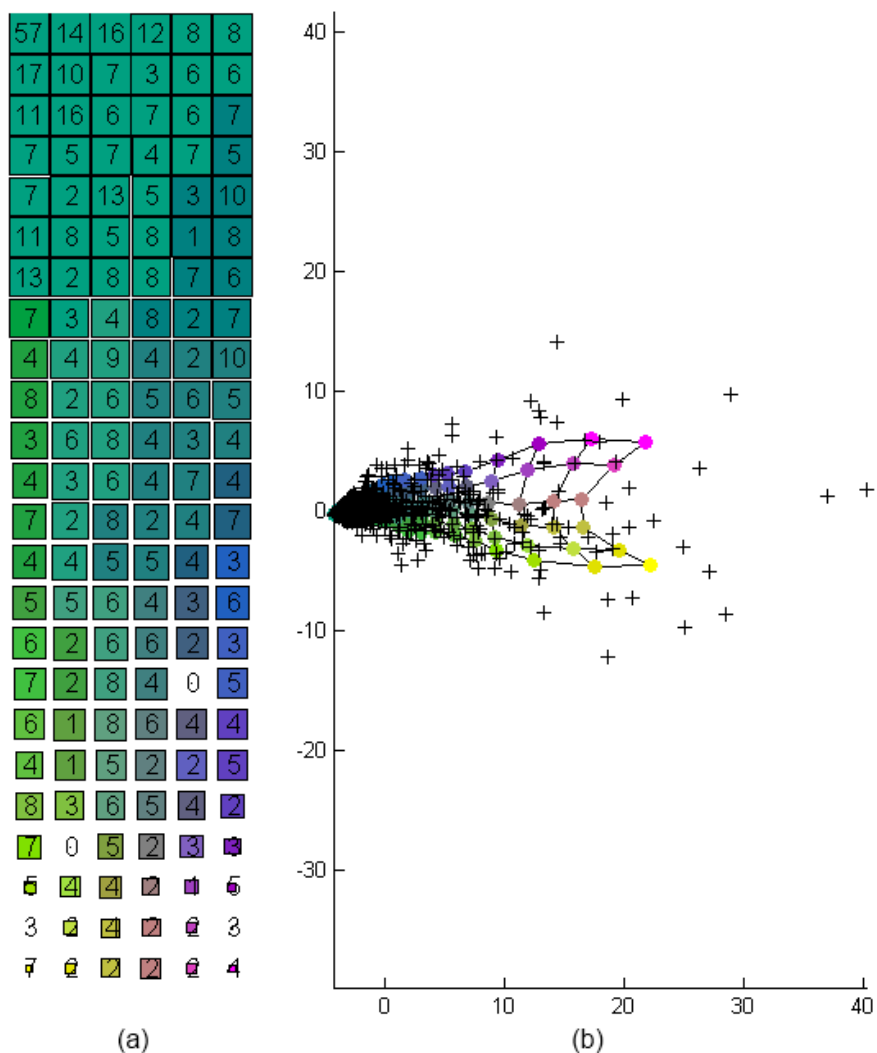


Figura 5-2 - Código de cores (a) e projeção de componentes principais (b)

Na figura 5-2 (a) o código de cores, a dimensão dos quadrados das unidades do mapa representa a sua coesão, mas nessas mesmas unidades podemos ver a quantidade de amostras nas unidades do mapa. A maioria dos valores de entrada está na parte superior, é também isso que se pode observar na figura 5-2 (b) projeção de componentes principais, onde estão os pontos de cor verde e azul que representam as projeções dos dados do mapa de Kohonen é também onde está a maior concentração de cruces que representam as projeções dos dados iniciais. Mais uma vez indicando a coesão dessa zona do mapa.

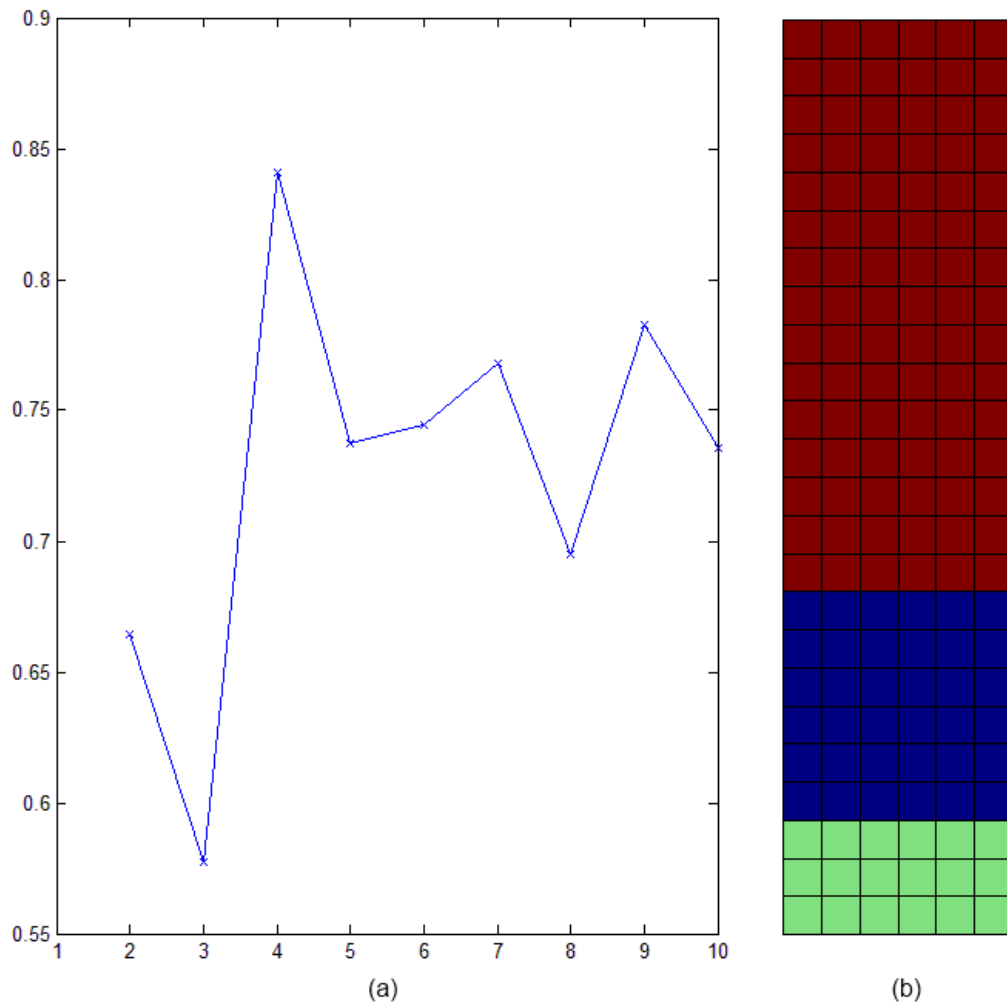


Figura 5-3 - Gráfico dos índices de Davies-Bouldin (a) e respetivo agrupamentos com menor índice (b)

Na figura 5-3 (a), está desenhado o gráfico do índice de Davies-Bouldin, e na figura 5-3 (b) o respetivo mapa com os agrupamentos criados pelo algoritmo *k-means*, neste caso com $k=3$, precisamente quando o índice é menor. Tal como o esperado o *k-means* retira a ambiguidade do mapa de Kohonen, embora que comparando a figura 5-1 e o código de cores da figura 5-2, seja visível perfeitamente a separação feita por *k-means*, na figura 5-1 por si só era mais difícil criar uma separação.

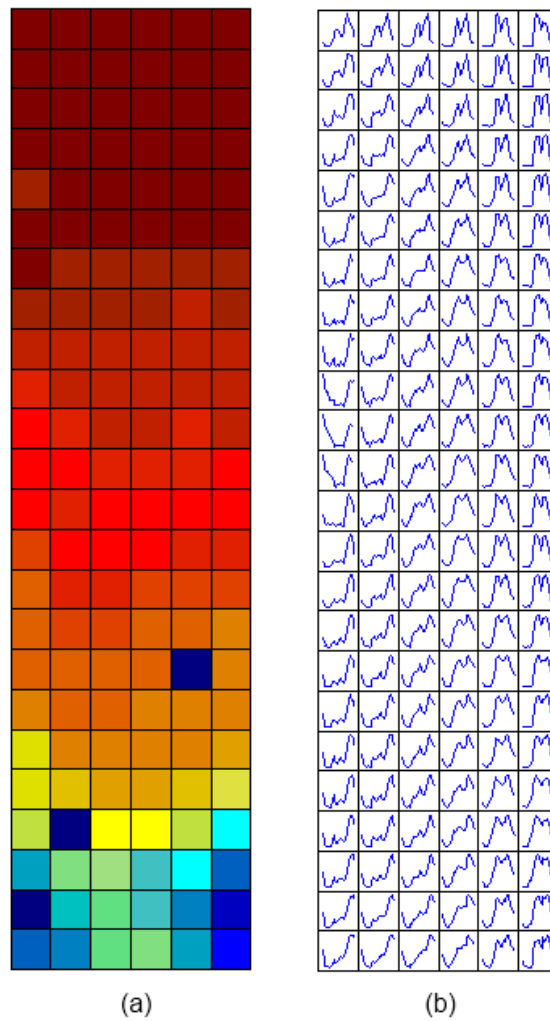


Figura 5-4 - Matriz de distâncias (a) e Protótipos (b)

Na figura 5-4 (a), podemos observar a representação da matriz de distâncias num código de cores, cores vermelhas escura, representam distâncias médias menores de cada unidade do mapa em relação ao vetor paramétrico também chamado de protótipo, o que mais uma vez representa unidades mais coesas, a cor azul escura representa distâncias maiores. Na figura 5-4 (b) protótipos são representados os vetores paramétricos modelo de cada unidade.

Partindo dos protótipos e da matriz de distâncias pode-se observar em que regiões do mapa existem maiores *erros* em relação os protótipos.

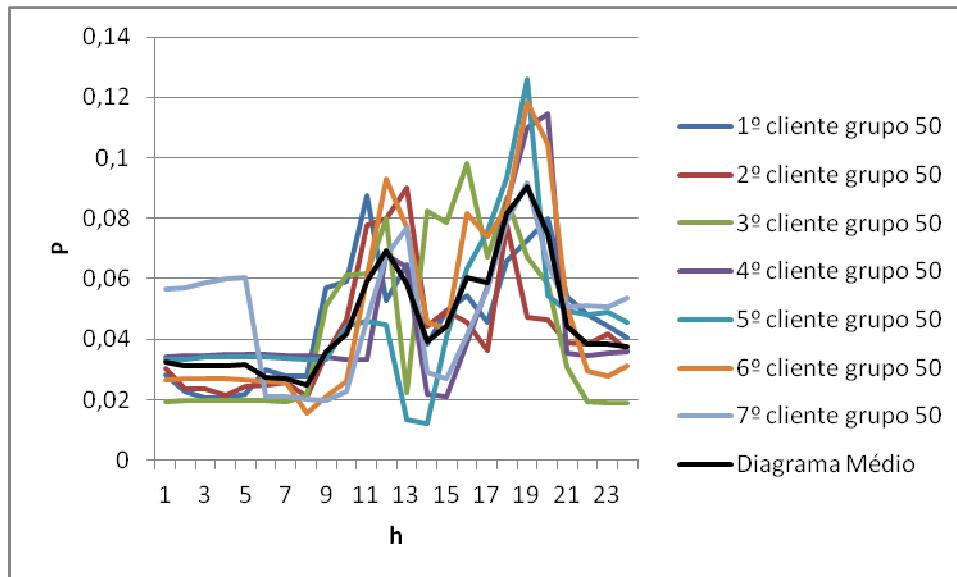


Figura 5-5 - Diagramas dos clientes colocados pela SOM na unidade 50 e o seu diagrama médio

Na figura 5-5, está representado um gráfico obtido com a informação exportada para folhas de cálculo com as funções implementadas para separar os elementos de cada unidade da SOM e também calcular o diagrama médio de cada unidade. Neste caso escolheu-se a unidade 50 da SOM, porque combina uma boa representatividade e ao mesmo tempo não tem excesso de informação.

Como pode ser observado existe uma boa semelhança entre os elementos desta unidade, o que demonstra a eficácia deste método.

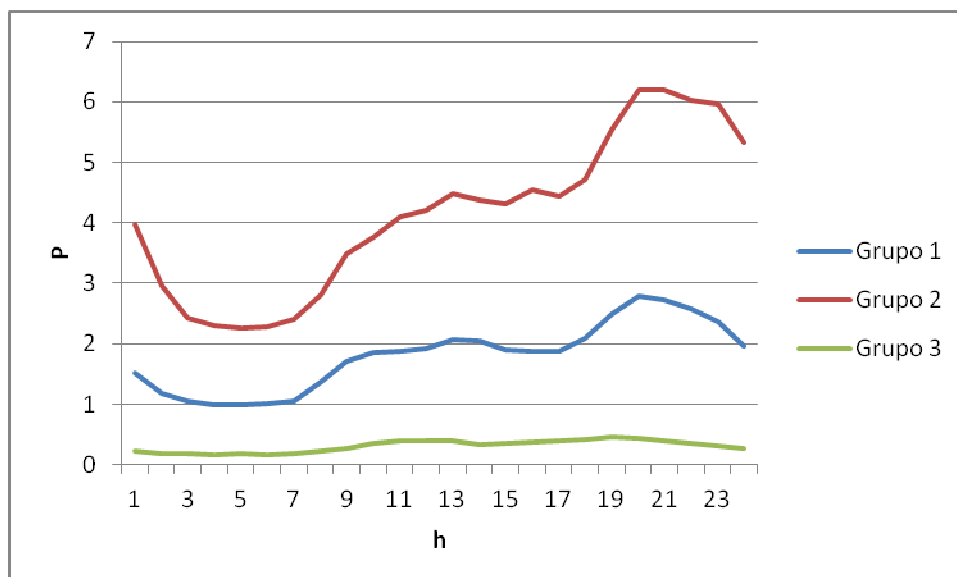


Figura 5-6 - Gráfico com os perfis de consumo dos agrupamentos criados com *k-means*($k=3$)

Na figura 5-6, estão representados os diagramas médios dos 3 agrupamentos criados pelo *k-means*, neste caso percebe-se perfeitamente que a separação efetuada teve em conta a amplitude dos diagramas e não só a forma. Trata-se de um resultado esperado porque neste teste os valores de entrada não foram normalizados.

Na figura 5-7, compara-se o diagrama médio da unidade 50 do mapa com o respectivo protótipo, como se vê estão muito próximos.

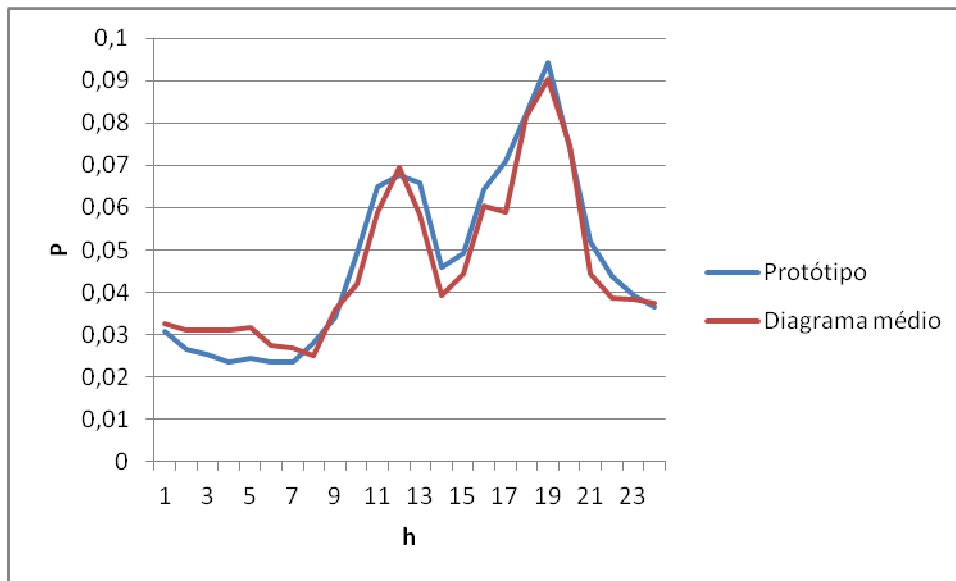


Figura 5-7 - Gráfico com o protótipo e diagrama médio da unidade 50 do SOM

5.2 - Teste t1n

Neste teste a diferença para o teste t1 é que os valores de entrada são normalizados. Vai-se analisar então que efeito que isso provoca no mapa de *Kohonen*.

Resumo dos parâmetros e condições considerados neste teste:

- Diagramas normalizados.
- Função *som_make* com o parâmetro tamanho do mapa não definido.

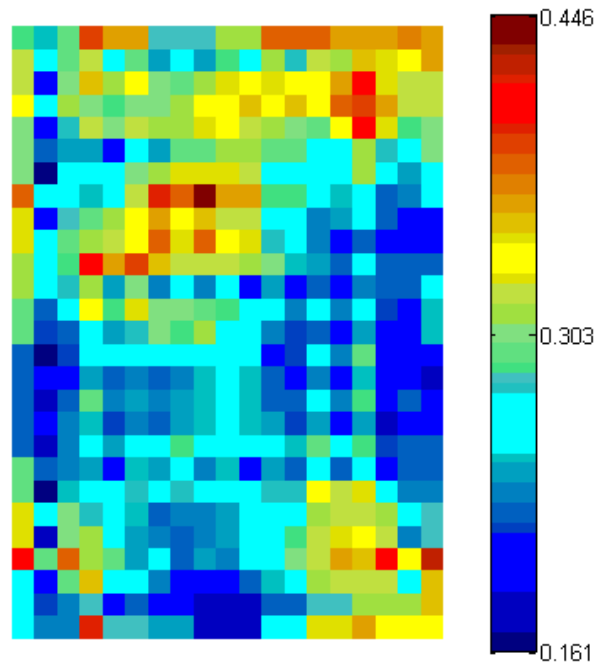


Figura 5-8 - Matriz unificada das distâncias

Na figura 5-8, ao contrário da figura 5-1, tem uma coesão pouco nítida. É precisamente um dos reflexos da normalização. O algoritmo fez uma separação em que a amplitude perde a importância, passando a fazê-la mais pela forma.

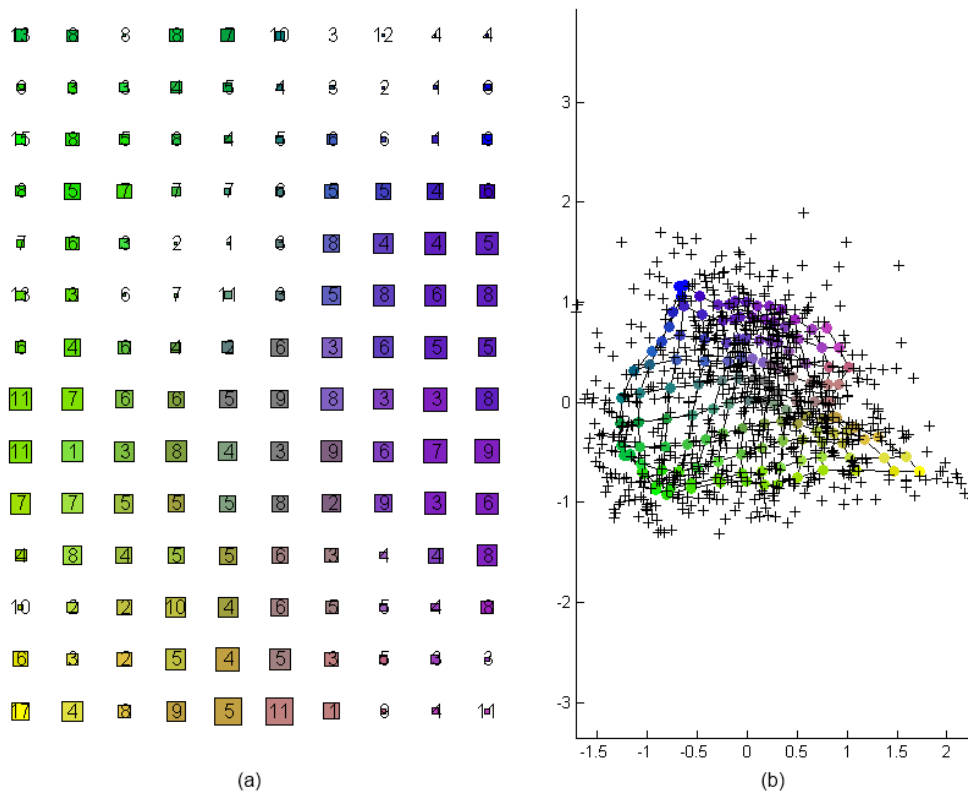


Figura 5-9 - Código de cores (a) e projeção de componentes principais (b)

Na figura 5-9 (a), tal como já se observou na figura 5-2 (a), vê-se uma menor coesão das unidades do mapa, e também na figura 5-9 (b) em relação à figura 5-2 (b) se verifica uma dispersão maiores dos componentes, o que subscreve a ideia de uma separação pela forma e não tanto pela amplitude.

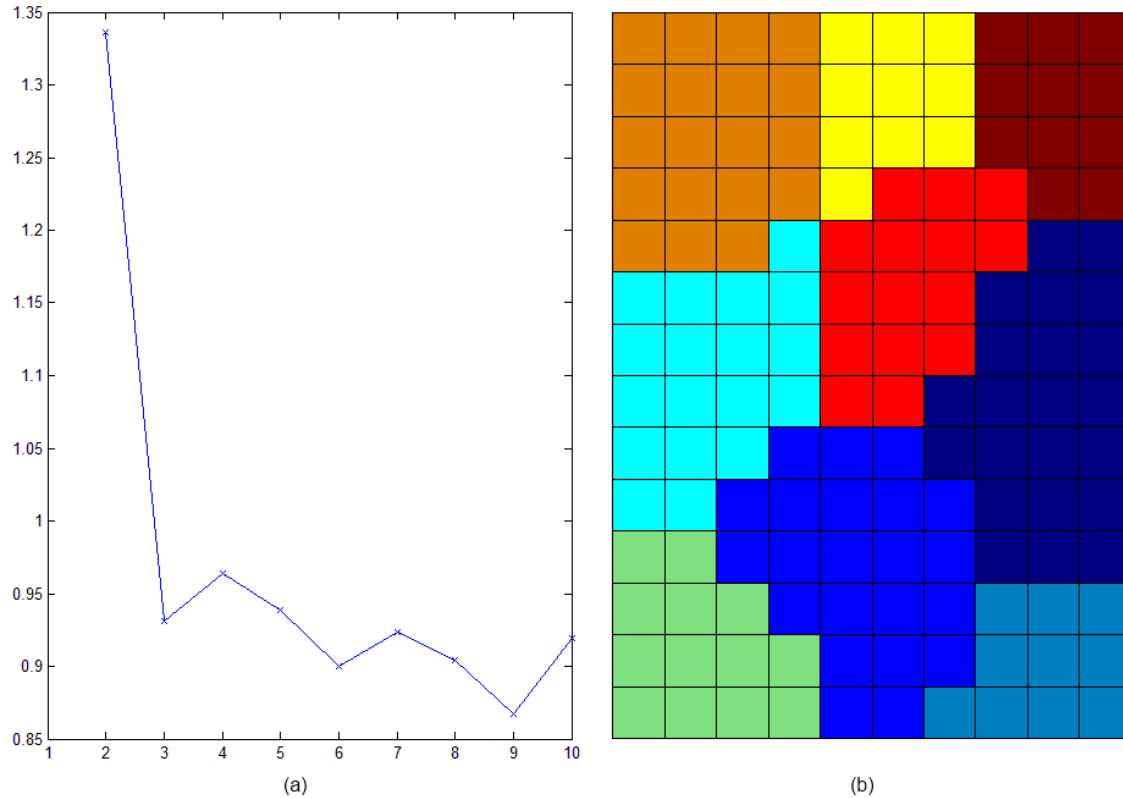


Figura 5-10 - Gráfico dos índices de *Davies-Bouldin* (a) e respetivo agrupamentos com menor índice (b)

Na figura 5-10, aqui também temos o que já se esperava, pois existe uma maior dispersão das unidades da SOM, o *k-means* encontrou uma maior quantidade de agrupamentos. Consegue-se perceber como no teste t1, que a separação feita por *k-means* retira a ambiguidade à matriz unificada de distâncias, que neste caso é maior.

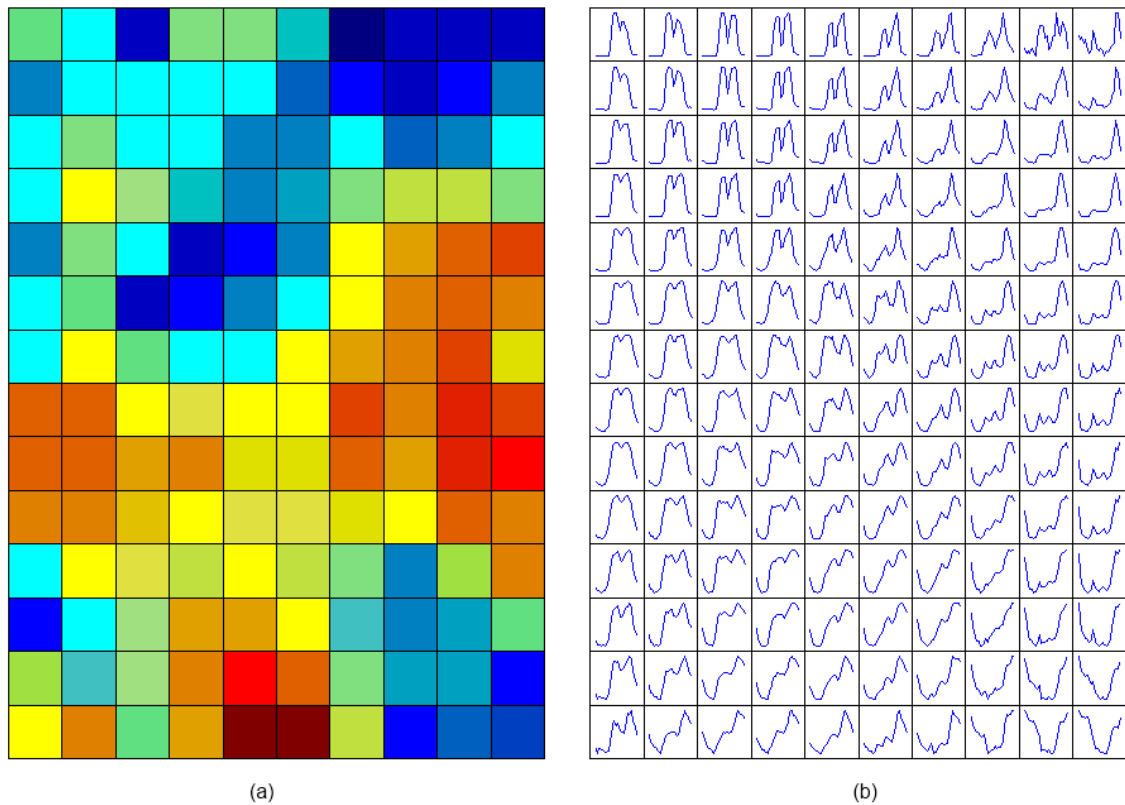


Figura 5-11 - Matriz das distâncias (a) e Protótipos (b)

Na figura 5-11, mais uma vez podemos observar a diferença de distância ao protótipo, e neste caso com uma maior distância aos protótipos.

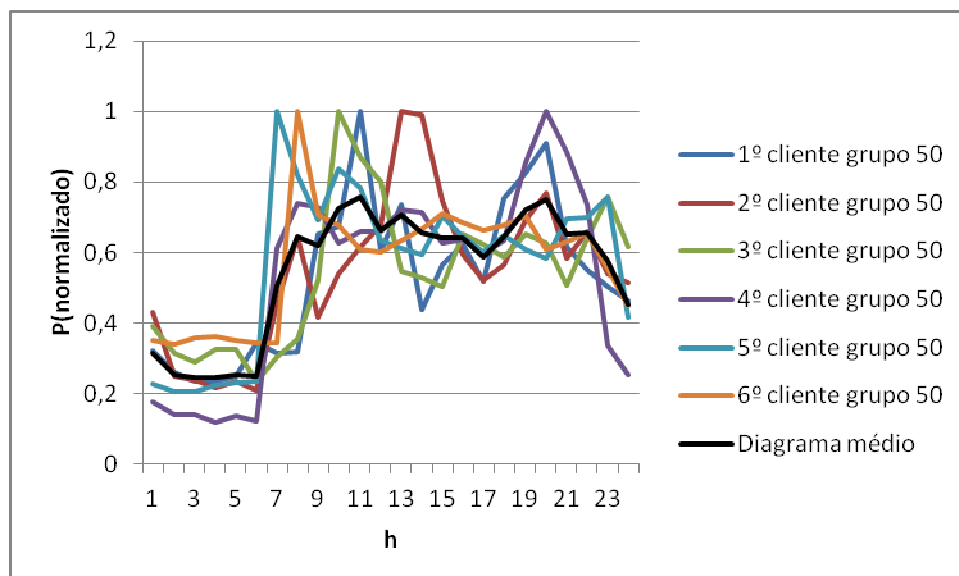


Figura 5-12 - Diagramas dos clientes colocados no agrupamento 50 e o seu diagrama médio

No gráfico da figura 5-12, temos mais um exemplo de uma unidade do mapa, os seus elementos e o diagrama médio.

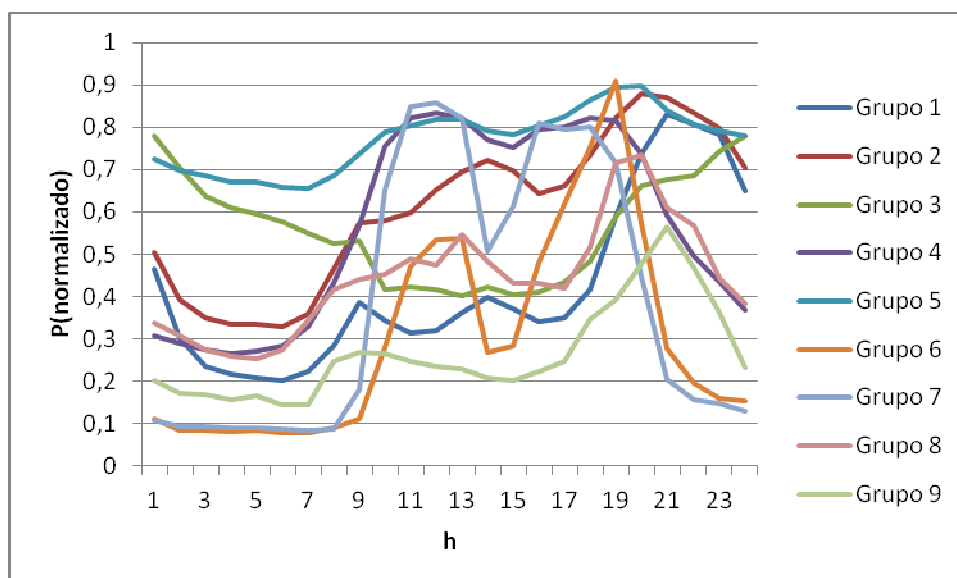


Figura 5-13 - Gráfico com os perfis de consumo dos agrupamentos criados com *k-means*($k=9$)

Na figura 5-13, como nesta situação temos uma maior quantidade de agrupamentos já conseguimos distinguir vários perfis de consumo, desde o típico consumo mais noturno, ao doméstico e ao comercial e industrial, etc.

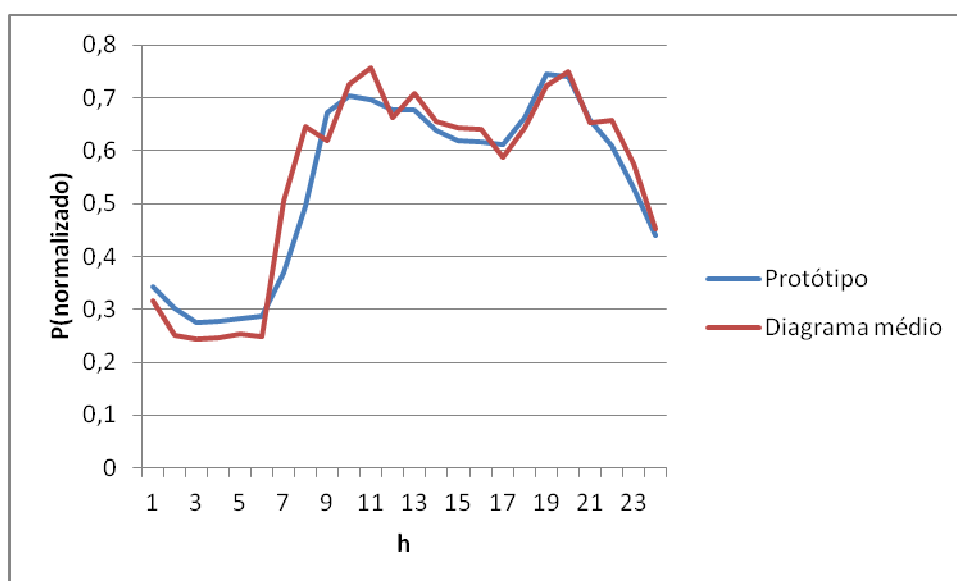


Figura 5-14 - Gráfico com o protótipo e diagrama médio da unidade 50 do SOM

Na figura 5-14, mais uma vez se demonstra a semelhança entre valor médio e o protótipo.

5.3 - Teste t_2

Neste teste, limitou-se o número de unidades do mapa através do parâmetro *msize* da função *som_make*, para ver o efeito que essa situação provoca no algoritmo.

Resumo dos parâmetros e condições considerados neste teste:

- Diagramas não normalizados.
- Função *som_make* com o parâmetro tamanho do mapa não definido.

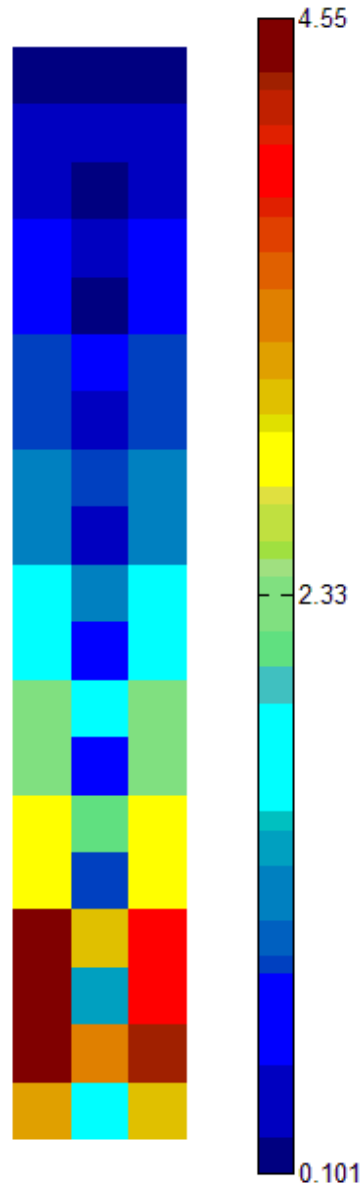


Figura 5-15 - Matriz unificada das distâncias

Na figura 5-15, tal como na figura 5-1 consegue-se ver onde estão as unidades mais coesas mas com uma resolução menor. O que dificulta a separação na parte inferior do mapa.

Na figura 5-16, observa-se o mesmo, a parte superior do mapa é mais coeso e não apresenta dificuldade para efetuar separação de agrupamentos nessa zona, mas na parte inferior isso já não acontece.

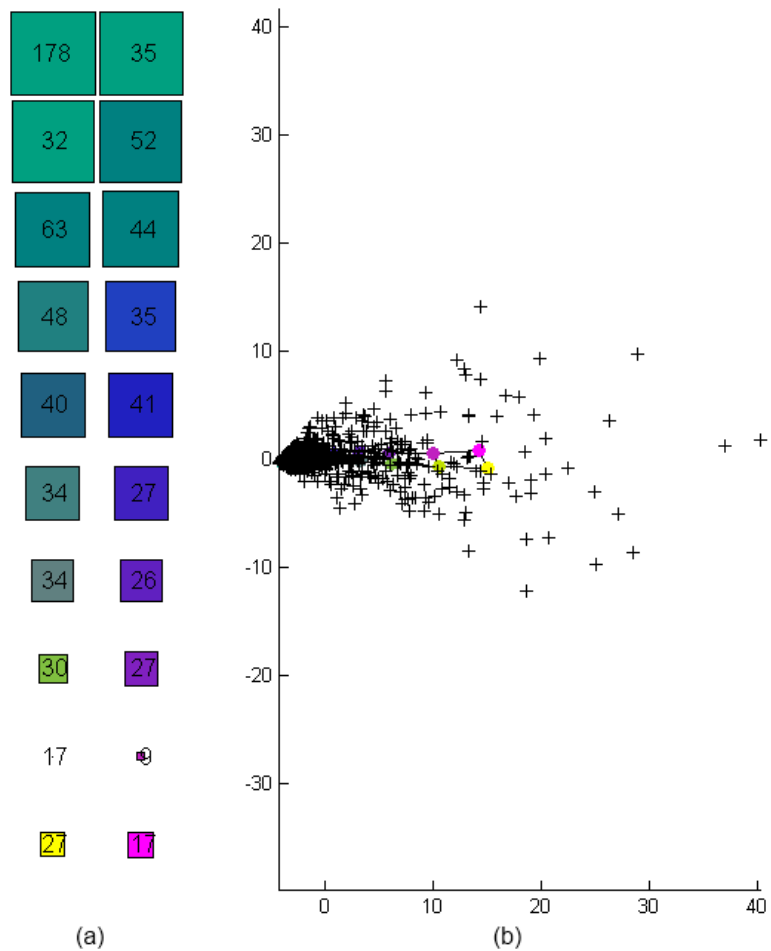


Figura 5-16 - Código de cores (a) e projeção de componentes principais (b)

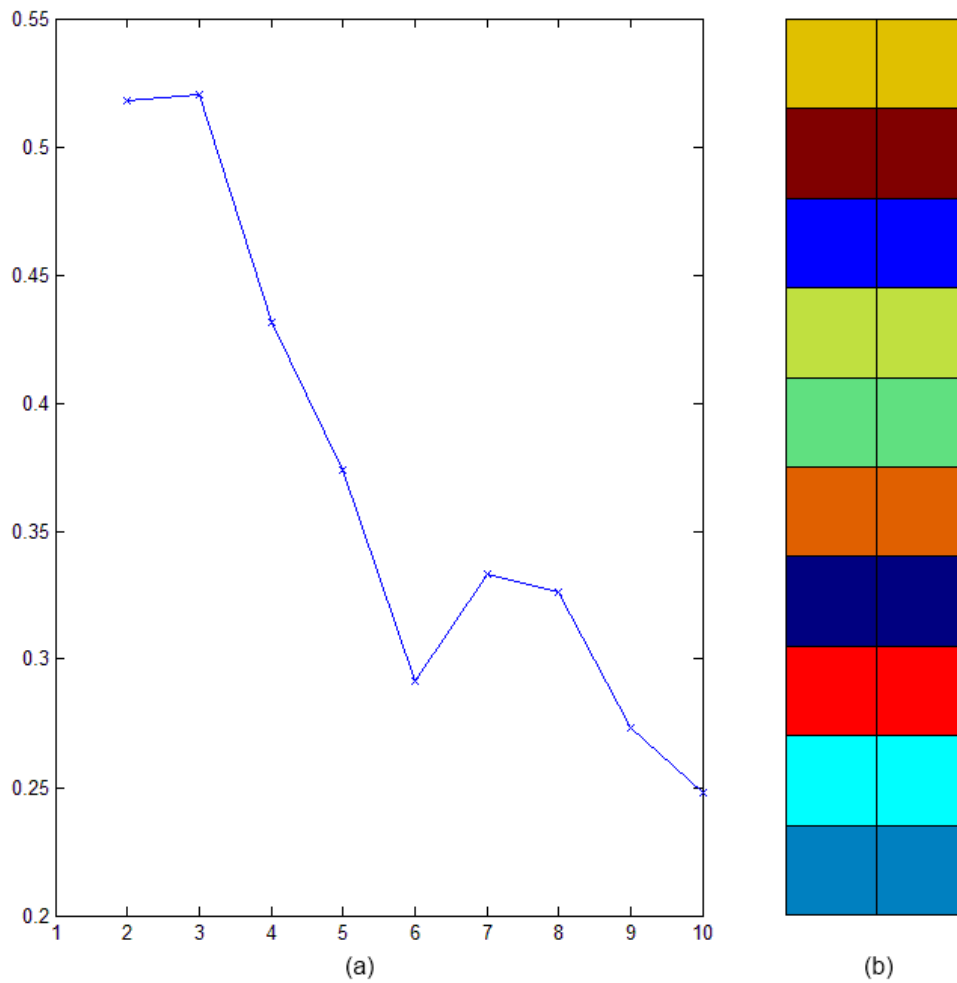


Figura 5-17 - Gráfico dos índices de *Davies-Bouldin* (a) e respetivos agrupamentos com menor índice (b)

Nesta situação o algoritmo *k-means* com a perda de resolução não consegue retirar tanta ambiguidade ao SOM como nos casos anteriores, pois separou o mapa em 10 agrupamentos, tal como se pode observar na figura 5-17.

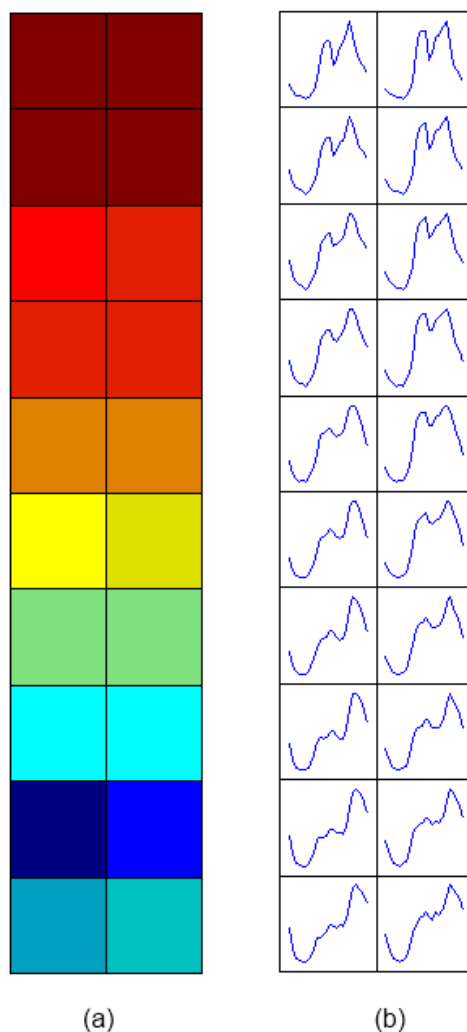


Figura 5-18 - Matriz de distâncias (a) e Protótipos (b)

A figura 5-18 também confirma isso, pois da perda de resolução resulta uma matriz de distâncias com valores mais elevados.

Na figura 5-19, apesar da perda de resolução verificada nas figuras anteriores observamos que os elementos deste grupo têm diagramas com muita semelhança.

Na figura 5-20, acontece a mesma situação observada na figura 5-6, os agrupamentos consideram a amplitude como uma diferença e não só a forma, mas neste caso com maior quantidade e agrupamentos e conseqüentemente mais níveis de amplitude.

Na figura 5-21, o protótipo e o diagrama médio estão próximos mas com maior erro, mais uma vez a comprovar a falta de resolução.

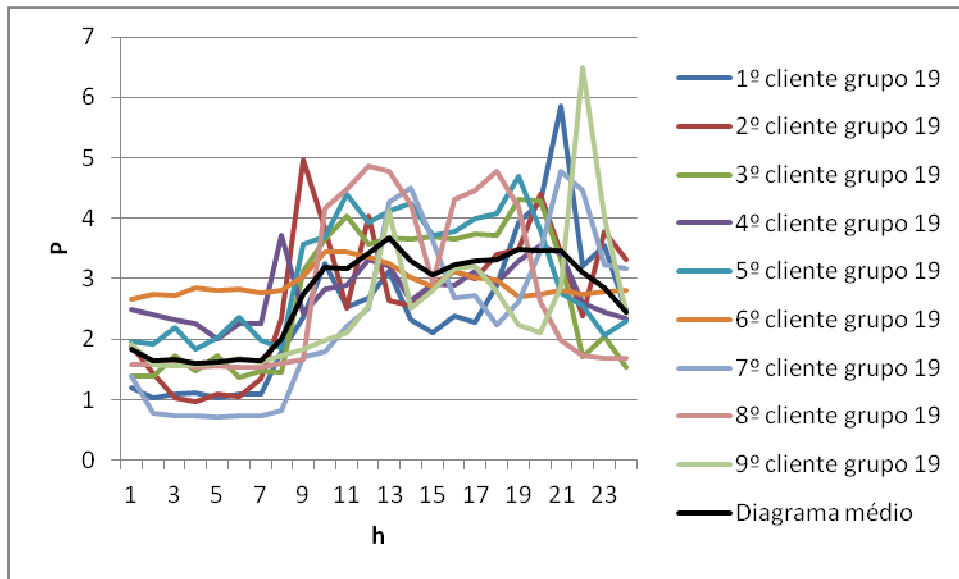


Figura 5-19 - Diagramas dos clientes colocados na unidade 19 do SOM e o seu diagrama médio

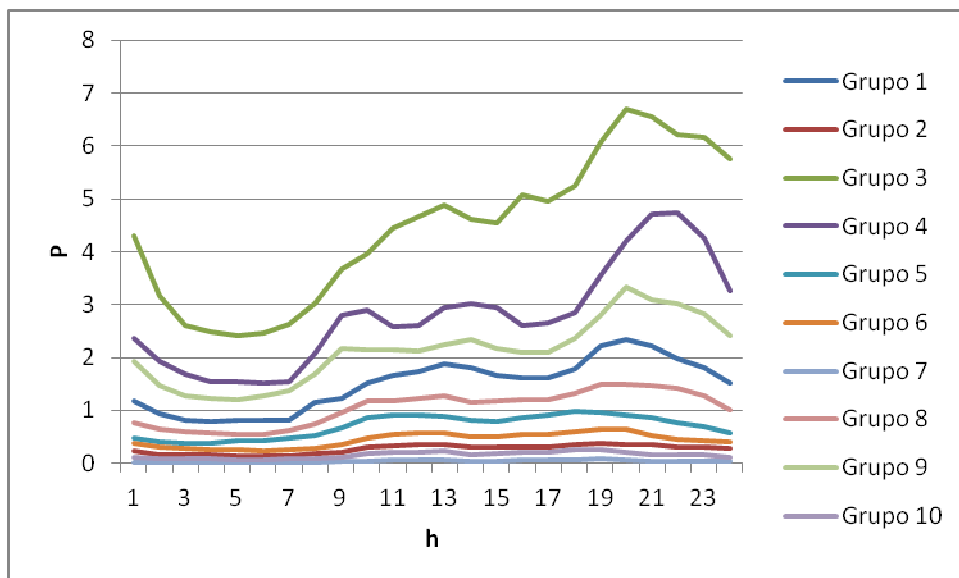


Figura 5-20 - Gráfico com os perfis de consumo dos agrupamentos criados com *k-means*($k=10$)

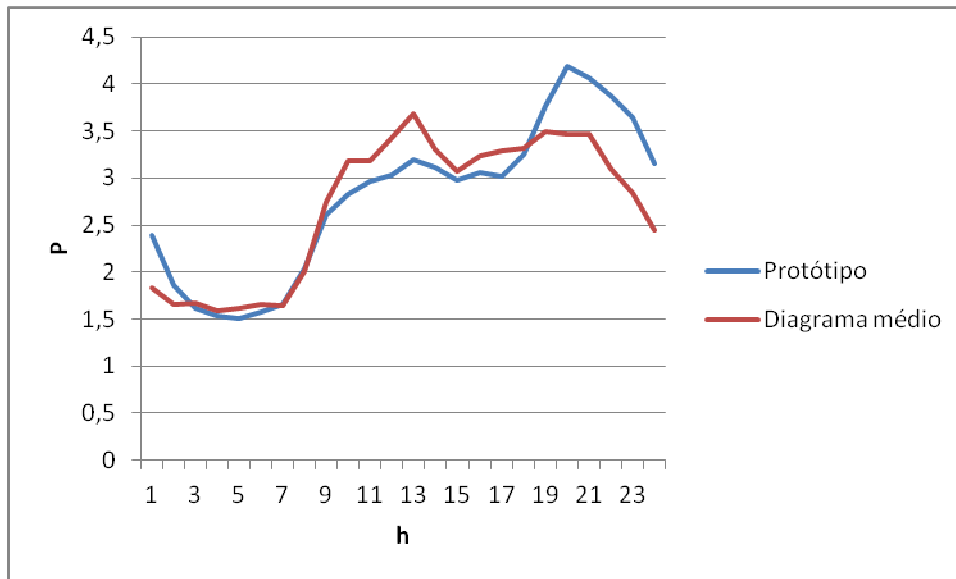


Figura 5-21 - Gráfico com o protótipo e diagrama médio da unidade 19 do SOM

5.4 - Teste t2n

Neste teste a diferença para o teste t2 é ser normalizado.

Resumo dos parâmetros e condições considerados neste teste:

- Diagramas normalizados.
- Função *som_make* com o parâmetro tamanho do mapa não definido.

Analisando este teste a partir da figura 5-22 fica-se com uma ideia de melhoria em relação ao anterior. Mas se continuar a análise das figuras seguintes, encontra-se já uma grande dispersão na projeção de componentes da figura 5-23.

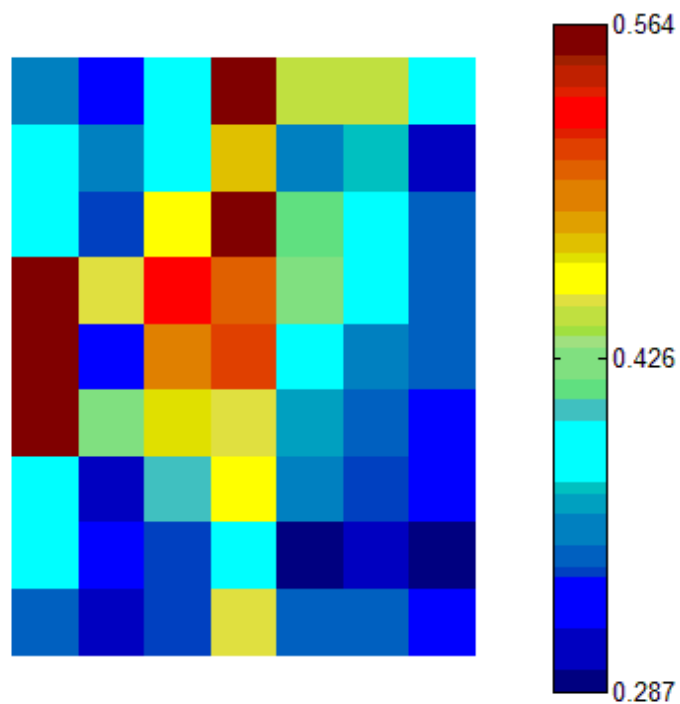


Figura 5-22 - Matriz unificada das distâncias

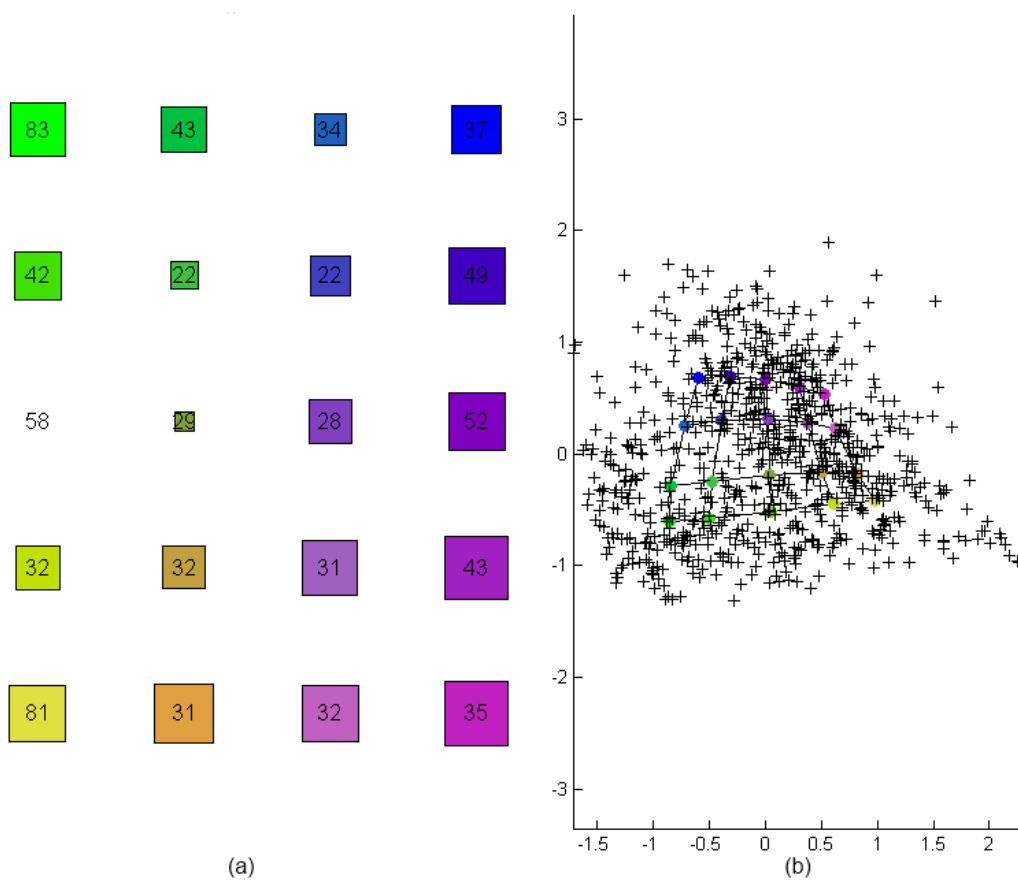


Figura 5-23 - Código de cores (a) e projeção de componentes principais (b)

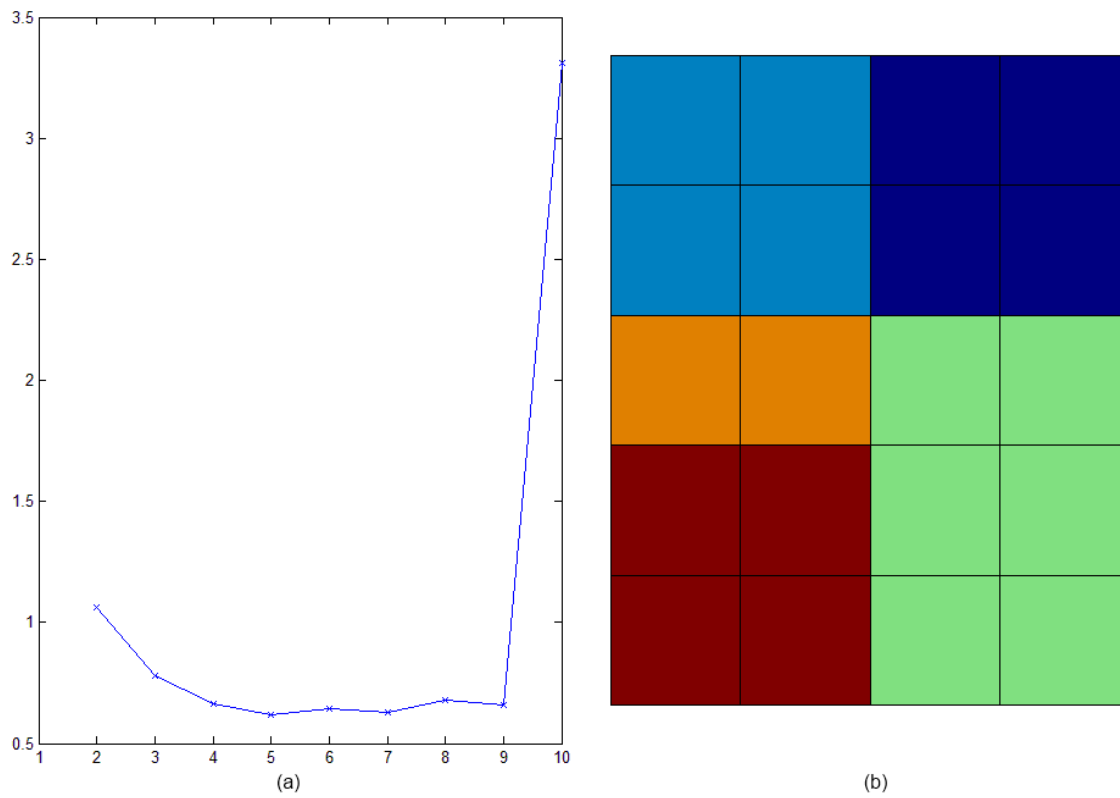


Figura 5-24 - Gráfico dos índices de *Davies-Bouldin* (a) e respectivos agrupamentos com menor índice (b)

Na figura 5-24, realmente o *k-means* já produz uma separação de menos agrupamentos, mas como se vê na figura 5-25 o erro ainda é considerável.

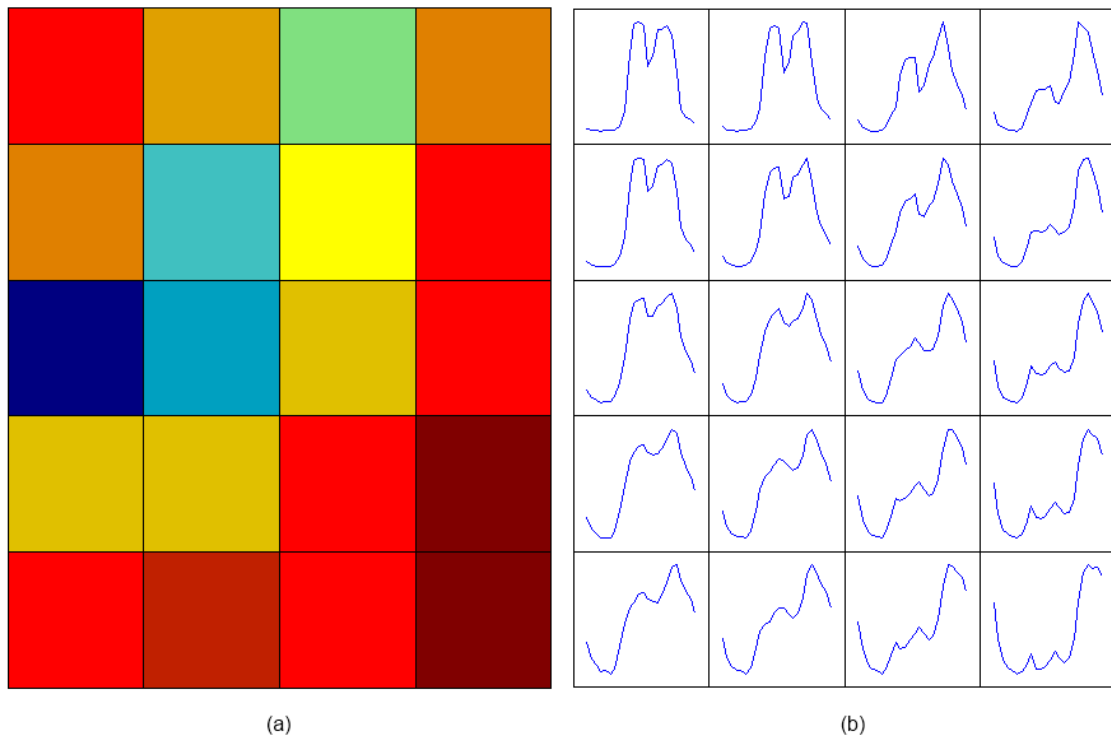


Figura 5-25 - Matriz de distâncias (a) e Protótipos (b)

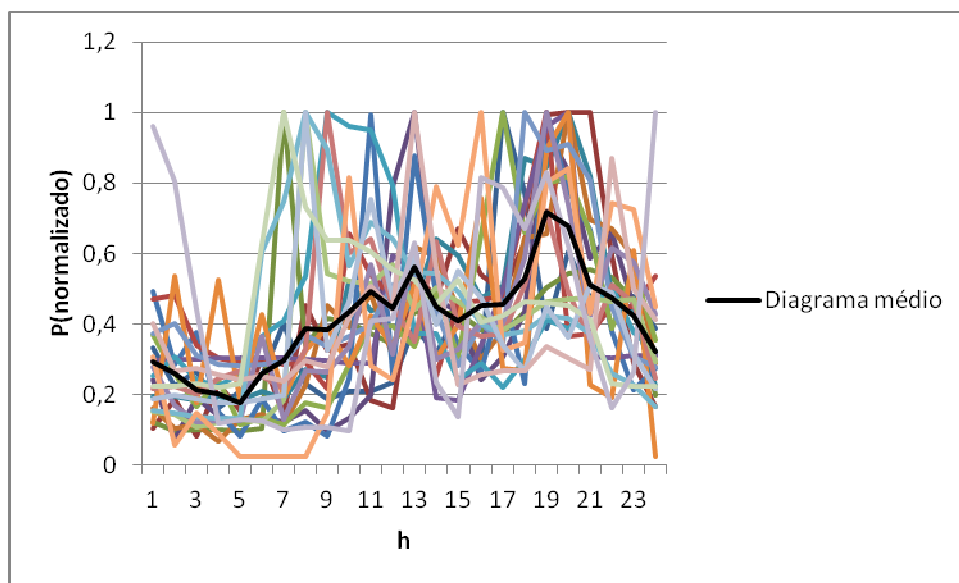


Figura 5-26 - Diagramas dos clientes colocados na unidade 12 do SOM e o seu diagrama médio

Na figura 5-26, demonstra-se o que já foi comentado anteriormente, isto é, o erro é maior. Isto porque o número de elementos dos agrupamentos é muito grande e com grandes diferenças entre eles, embora esta figura seja só demonstrativa de um agrupamento.

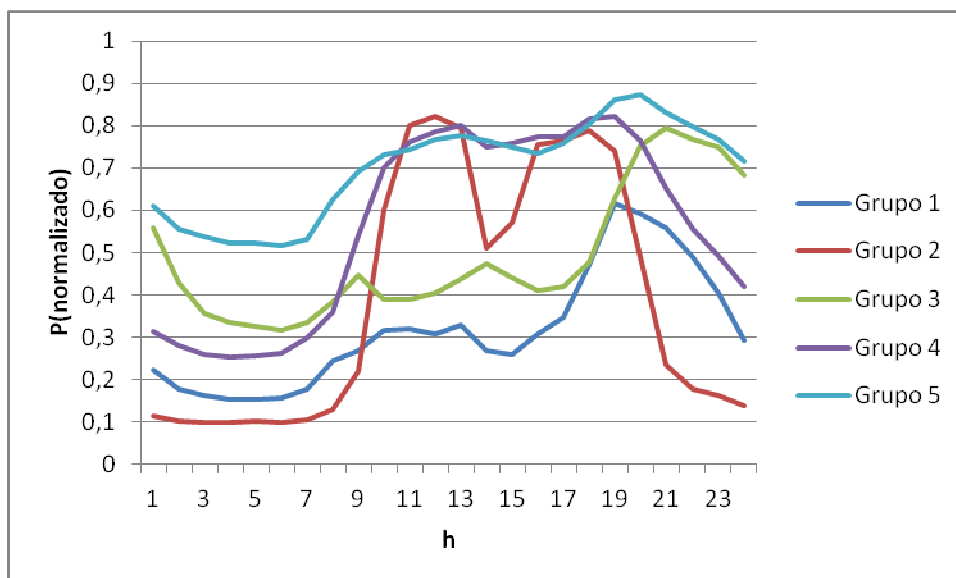


Figura 5-27 - Gráfico com os perfis de consumo dos agrupamentos criados com *k-means*(k=5)

Na figura 5-27, observa-se uns perfis de consumo típicos tal como aconteceu na figura 5-13 e como seria de esperar, pois ambas são obtidas com valores de entrada normalizados, mas neste caso mostram perfis bastante distintos entre classes.

Na figura 5-28, apresenta uma boa semelhança entre o protótipo e o diagrama médio desta unidade do SOM.

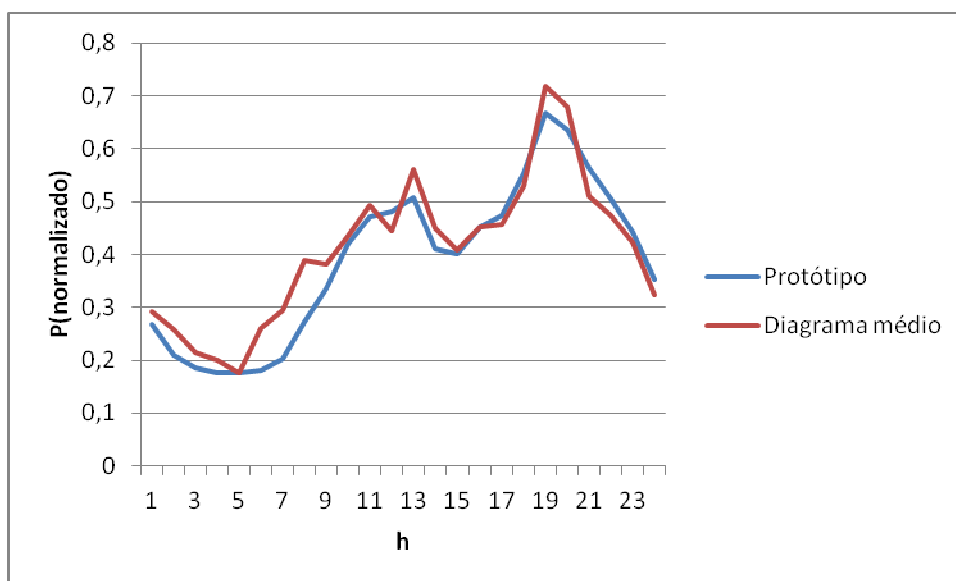


Figura 5-28 - Gráfico com o protótipo e diagrama médio da unidade 12 do SOM

Nesta primeira parte do trabalho verificou-se que o método baseado em mapas de Kohonen é eficaz para obter perfis de consumo, mas para obter estes perfis de consumo é necessário ter acesso a medições do consumo de cada cliente com maior frequência tipicamente horariamente. Os consumidores BTN, que representam a maior parte dos clientes têm normalmente só medições mensais, uma solução para esse problema seria instalar

medidores com capacidade de medição horária, mas é uma solução considerada cara. Logo esta solução é de difícil aplicação no mercado energético de eletricidade.

Uma possível solução para obter classes de perfis de consumo dos consumidores BTN, passaria então por uma metodologia que use um algoritmo de *clustering* combinado com uma divisão apriorística com os dados de faturação, potência contratada e consumo mensal.

Nos testes seguintes a que se vai designar parte 2 implementou-se uma divisão apriorística que relaciona os dados de faturação com classes de perfis de consumo, resolvendo o problema atrás descrito.

5.5 - Teste parte2

Nesta parte do trabalho, inicia-se uma nova abordagem do trabalho. Porque os dados de entrada no mapa de *Kohonen* já são em menor número, uma vez que os dados utilizados na primeira parte do trabalho, já foram separados automaticamente por níveis de consumo e potência contratada conforme a tabela 5.2, para então calcular o diagrama médio de cada um dos níveis, este diagrama médio de cada nível vão passar a ser os valores de entrada do algoritmo. Os níveis de consumo e de potência contratada foram obtidos com o auxílio de histogramas na folha de cálculo para tentar efetuar uma separação suficientemente uniforme dos clientes por cada célula.

Foram considerados 4 níveis de consumo por 4 de potência contratada, o que dá 16 vetores de entrada, que corresponde a 16 células⁵.

Resumo dos parâmetros e condições considerados neste teste:

- Diagramas não normalizados.
- Função *som_make* com o parâmetro *msize=20*.

Tabela 5.2 - Níveis de consumo/Potência contratada

Níveis de Consumo (kWh)	Níveis de Potência contratada (kVA)
Nível 1 < 690	Nível 1: 1,15; 2,3; 3,45
690 < Nível 2 < 2960	Nível 2: 4,6; 5,75; 6,9
2960 < Nível 3 < 9800	Nível 3: 10,35; 13,8; 17,25
9800 < Nível 4	Nível 4 : 20,70; 27,6; 34,5; 41,4

A figura 4-1 do capítulo anterior ajuda a visualizar melhor esta tabela.

⁵ Podiam ter sido considerados outros números para os escalões de consumo e de potência contratada. Neste caso, optou-se por um valor relativamente baixo apenas por questões de simplicidade.

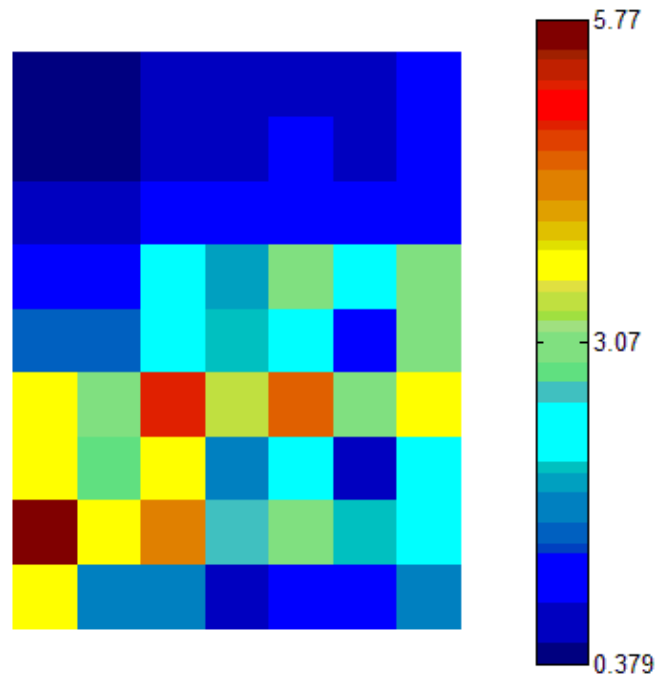


Figura 5-29 - Matriz unificada das distâncias

Na figura 5-29 observa-se uma matriz coesa com agrupamentos facilmente identificáveis, mas também uma fronteira entre classes a meio onde se observa cores mais quentes.

A figura 5-30 (a) vem comprovar a posição desses agrupamentos, dispostos nitidamente em dois blocos de unidades do mapa. Na figura 5-3 (b) mostra-se que a zona das projeções das unidades do mapa com cores em torno do azul, está mais perto dos valores de entrada, e a zona com cores em torno do rosa e amarelo está mais longe.

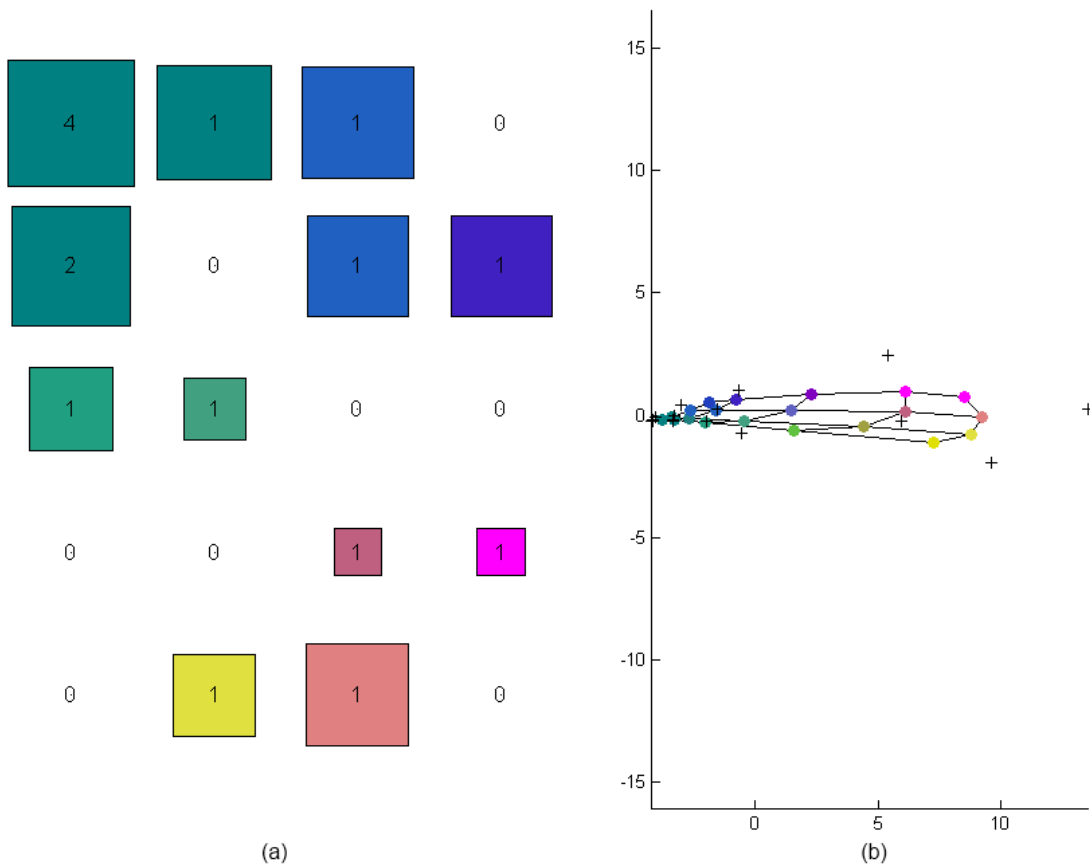


Figura 5-30 - Código de cores (a) e projeção de componentes principais (b)

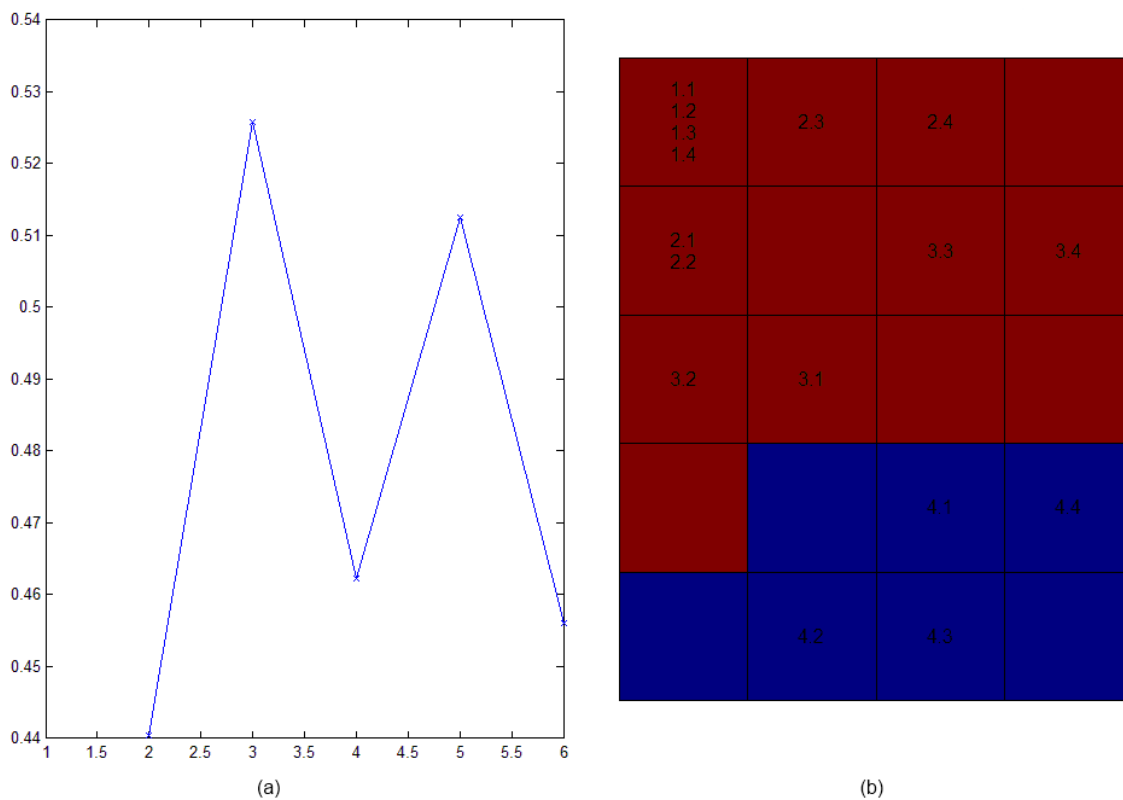


Figura 5-31 - Gráfico dos índices de *Davies-Bouldin* (a) e respetivo agrupamentos com menor índice (b)

Agora na figura 5-31, comprova-se com o *k-means*, aquilo que já se tinha observado nas figuras 5-29 e 5-30, isto é, uma separação por dois agrupamentos.

Na figura 5-32 pode-se observar se quais os protótipos que estão mais *próximos* dos diagramas médios das unidades do mapa. As unidades com cor azul estão mais afastadas e as de cor mais vermelho escuro mais perto do protótipo. As cores entre médias representam distâncias também entre médias. Neste caso as unidades com cor azul-escuro estão vazias daí estarem mais longe do protótipo.

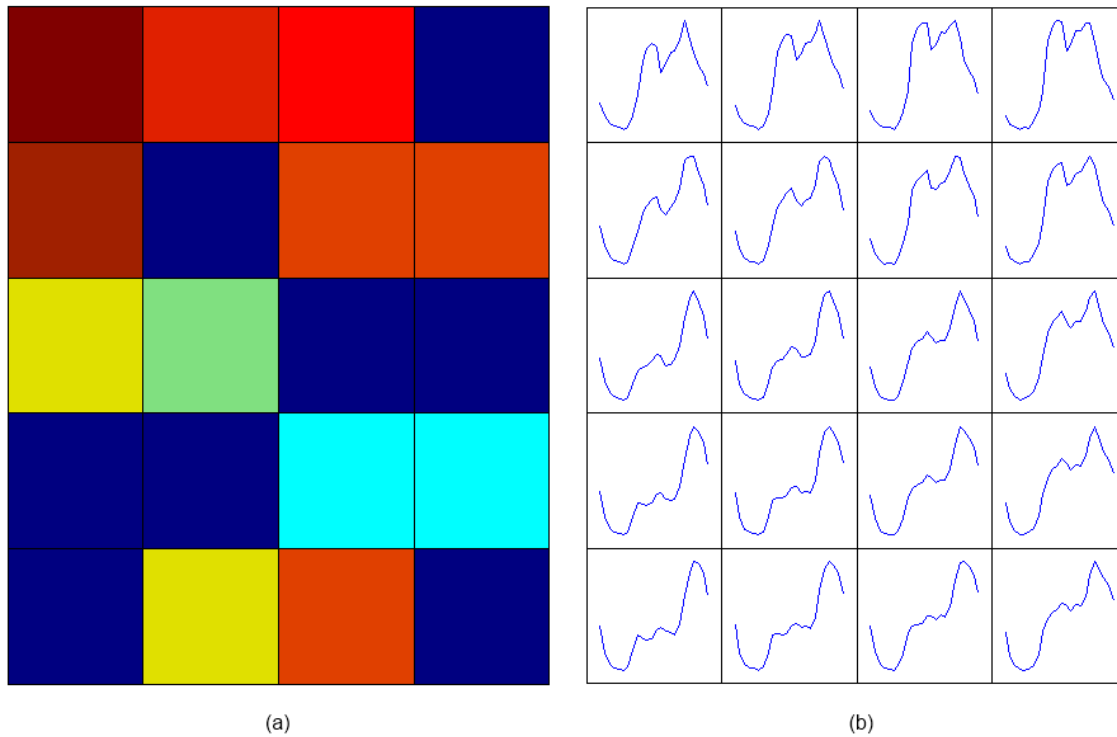


Figura 5-32 - Matriz de distâncias (a) e Protótipos (b)

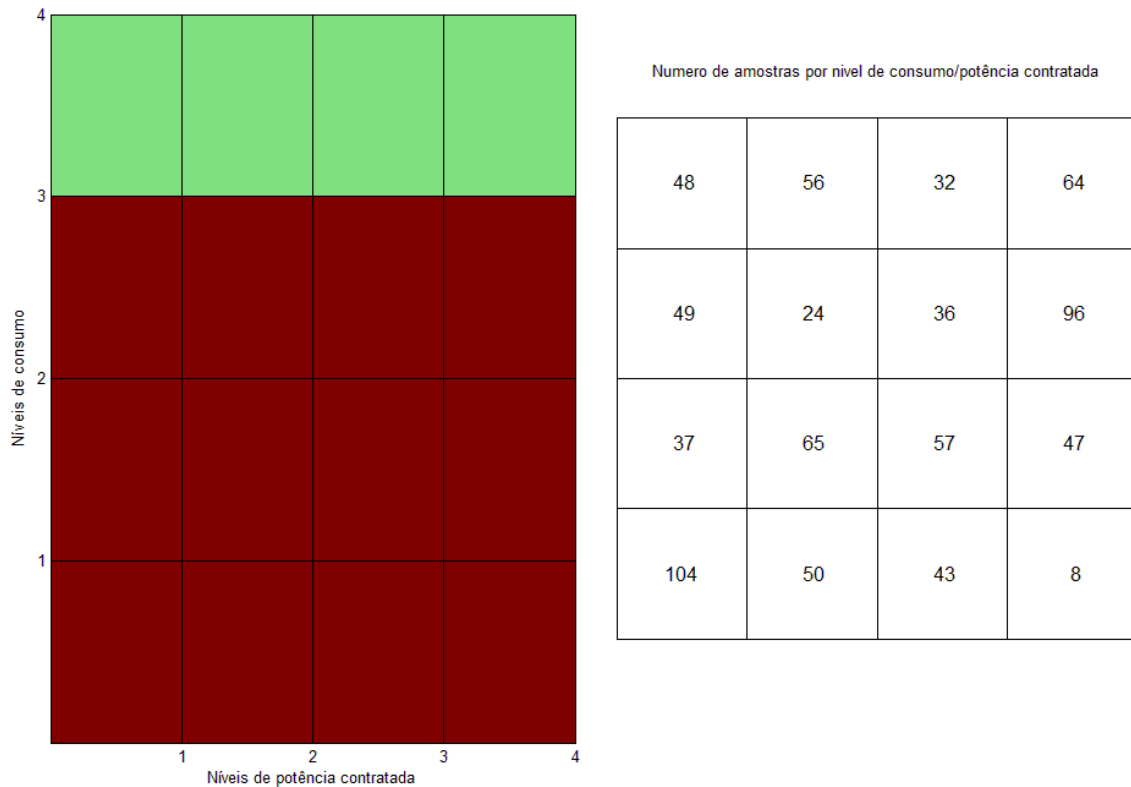


Figura 5-33 - Agrupamentos formados na grelha com os níveis de consumo e potência contratada, e quantidade de elementos em cada posição

A figura 5-33 é o resultado de um programa implementado pelo autor da dissertação para demonstrar mais facilmente a posição dos agrupamentos em relação aos níveis de consumo e potência contratada. Estes níveis correspondem aos valores da tabela 5.2.

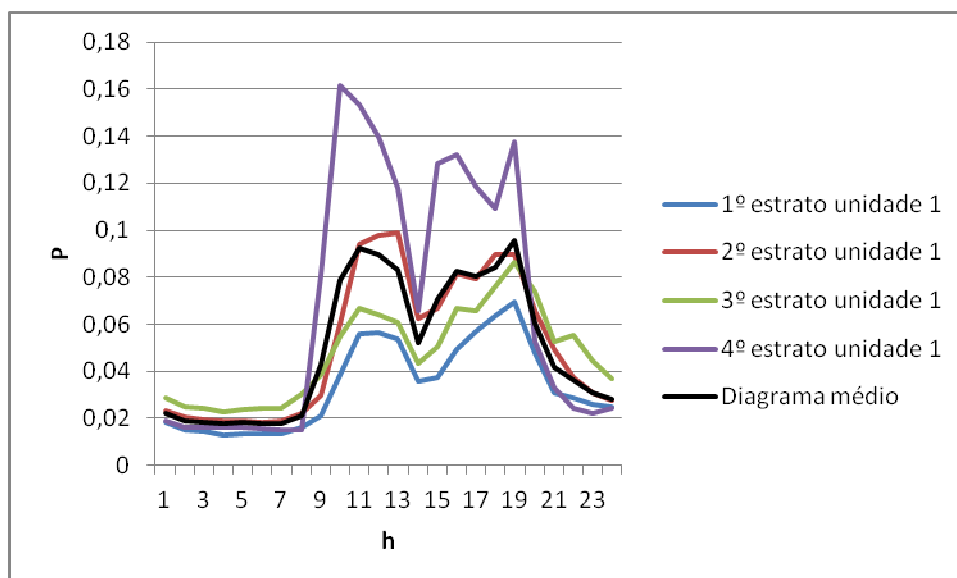


Figura 5-34 - Diagramas dos estratos de consumo colocadas na unidade 1 do SOM e o seu diagrama médio

Na figura 5-34, estão desenhadas os diagramas das divisões por níveis colocadas na unidade 1 do SOM e o respetivo diagrama médio.

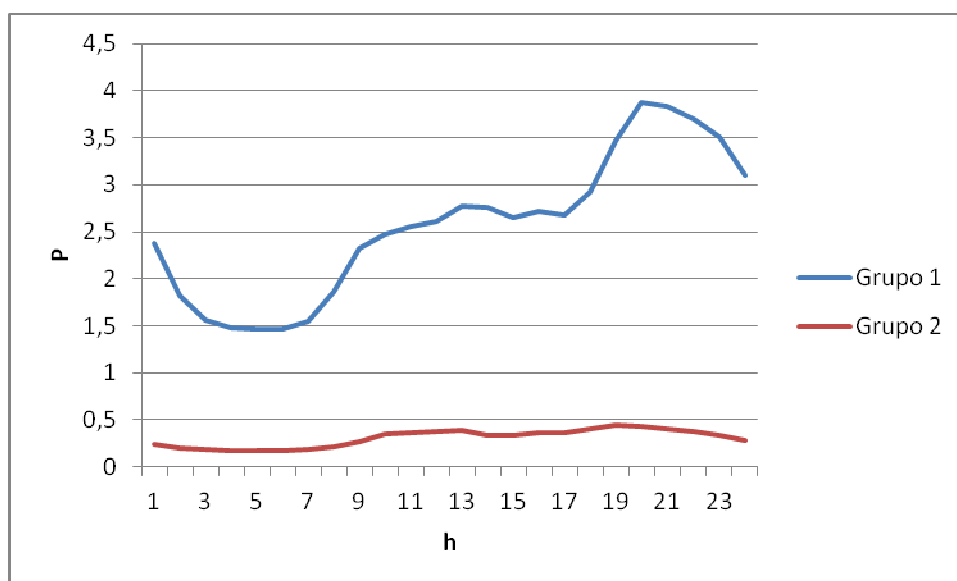


Figura 5-35 - Gráfico com os perfis de consumo criados com *k-means*(k=2)

Na figura 5-35, mostra o resultado que se obtém após o *k-means*, neste caso temos um resultado lógico, pois já foi visto na figura 5-33 que a separação resultante origina dois agrupamentos, um com amplitude mais elevada, a que correspondem o níveis de consumos mais elevado da tabela 5.2 e os restantes num agrupamento com valores de amplitude bem mais baixa, tal como aconteceu no teste t1, em que a separação de classes é feita sobretudo pela amplitude do diagrama.

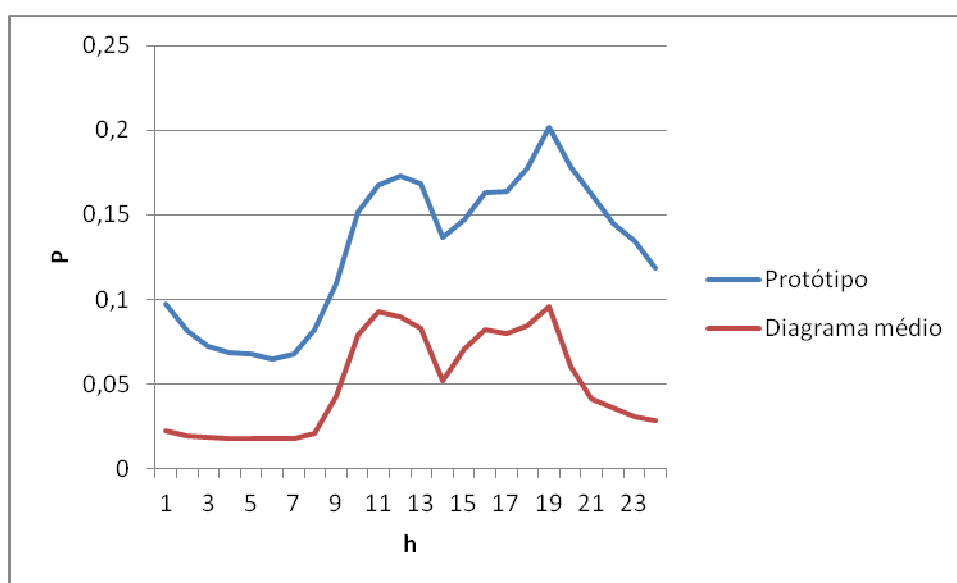


Figura 5-36 - Gráfico com o protótipo e diagrama médio da unidade 1 do SOM

Na figura 5-36, o protótipo e o diagrama tem a forma semelhante mas um valor de amplitude completamente diferente. Ou seja à primeira vista quererá indicar que esta separação envolve erros elevados.

5.6 - Teste parte2n

Nesta última parte introduziu-se mais uma vez valores normalizados, mas neste caso os valores normalizados depois de separados por níveis de consumo e de potência contratada.

Resumo dos parâmetros e condições considerados neste teste:

- Diagramas normalizados.
- Função *som_make* com o parâmetro *msize=20*.

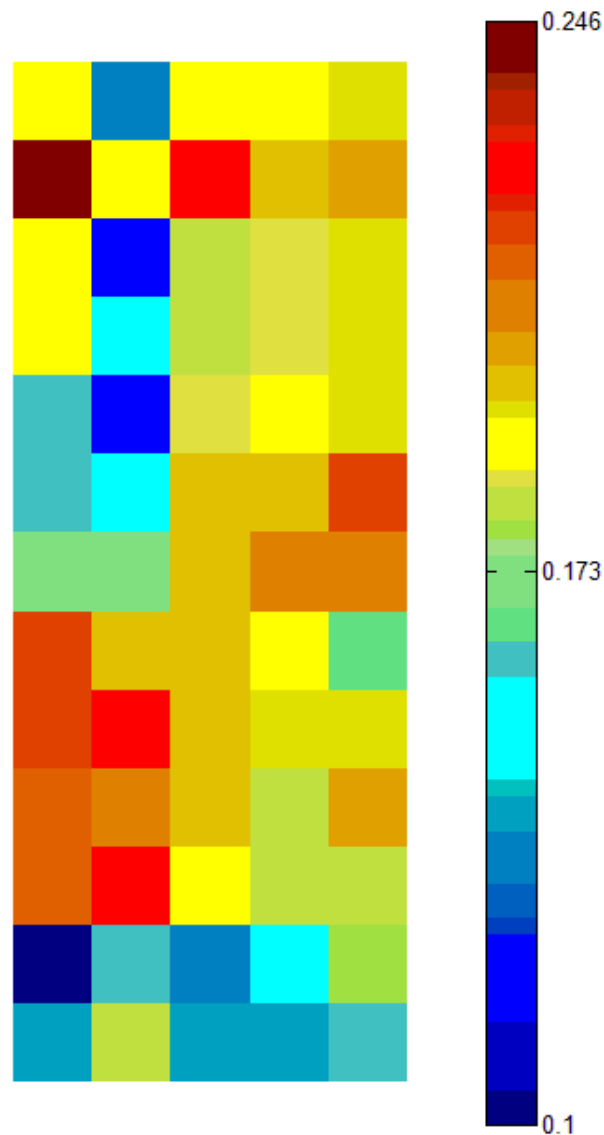


Figura 5-37 - Matriz unificada das distâncias

Na figura 5-37, volta a acontecer o mesmo que nos outros testes que envolvem valores normalizados. Denotando pouca coesão mas continuando a análise, a dispersão também é elevada como se pode observar na figura 5-38 (b), neste caso as projeções dos valores de entrada e dos elementos do mapa de Kohonen estão ambos espalhados mas dando uma ideia de maior proximidade entre elas.

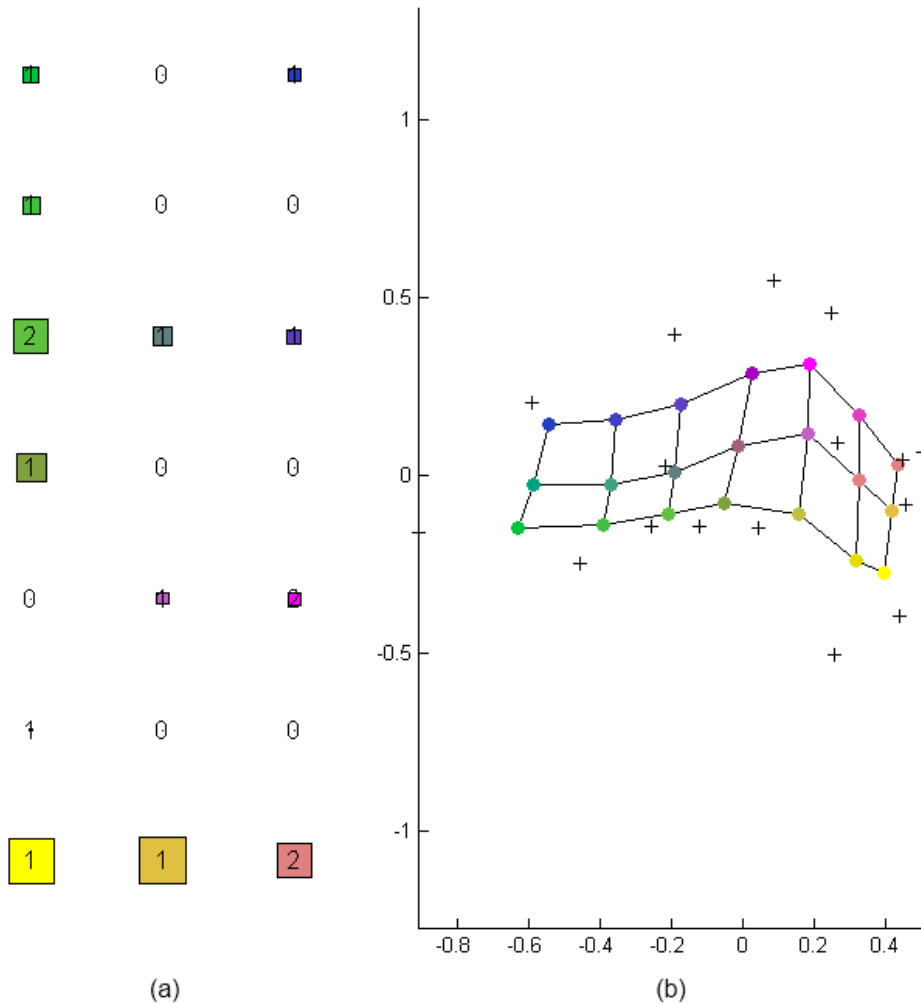


Figura 5-38 - Código de cores (a) e projeção de componentes principais (b)

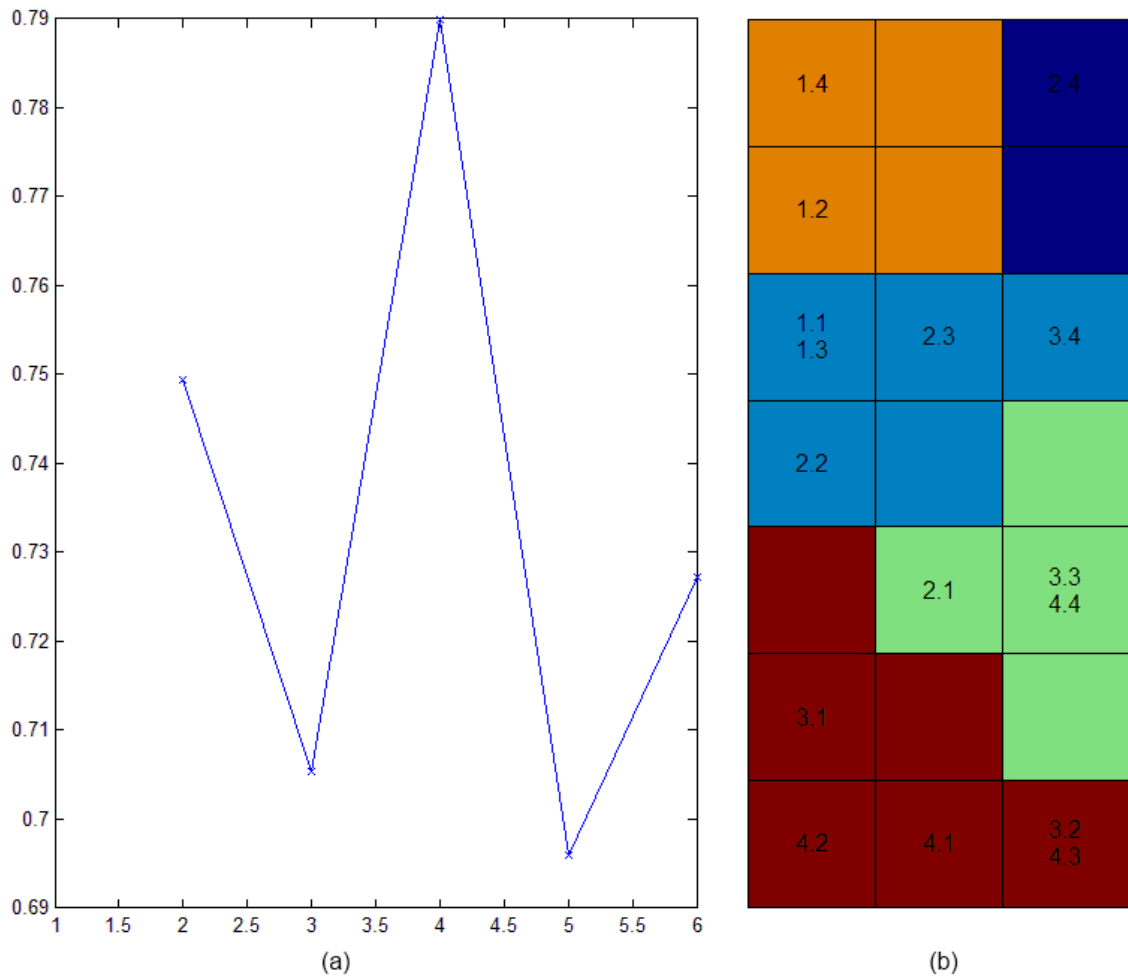


Figura 5-39 - Gráfico dos índices de Davies-Bouldin (a) e respetivo agrupamentos com menor índice (b)

Na figura 5-39, vemos que o número de agrupamentos subiu, desta vez pela falta de coesão demonstrada na figura 5-37, mas se repararmos o índice de Davies-Bouldin para $k=3$ que também é baixo, o que indica que a solução para 3 agrupamento também é viável e até faz mais sentido em termos de classes de perfis de consumo ter um número mais baixo de agrupamentos. Esta solução pode ser vista na figura 5-40.

1.4		2.4
1.2		
1.1 1.3	2.3	3.4
2.2		
	2.1	3.3 4.4
3.1		
4.2	4.1	3.2 4.3

Figura 5-40 - Agrupamentos com k=3(3 grupos)

Na figura 5-41, vê-se que as duas unidades do mapa do canto inferior esquerdo com melhor distância ao protótipo correspondem aos níveis (4.1) e (4.2).

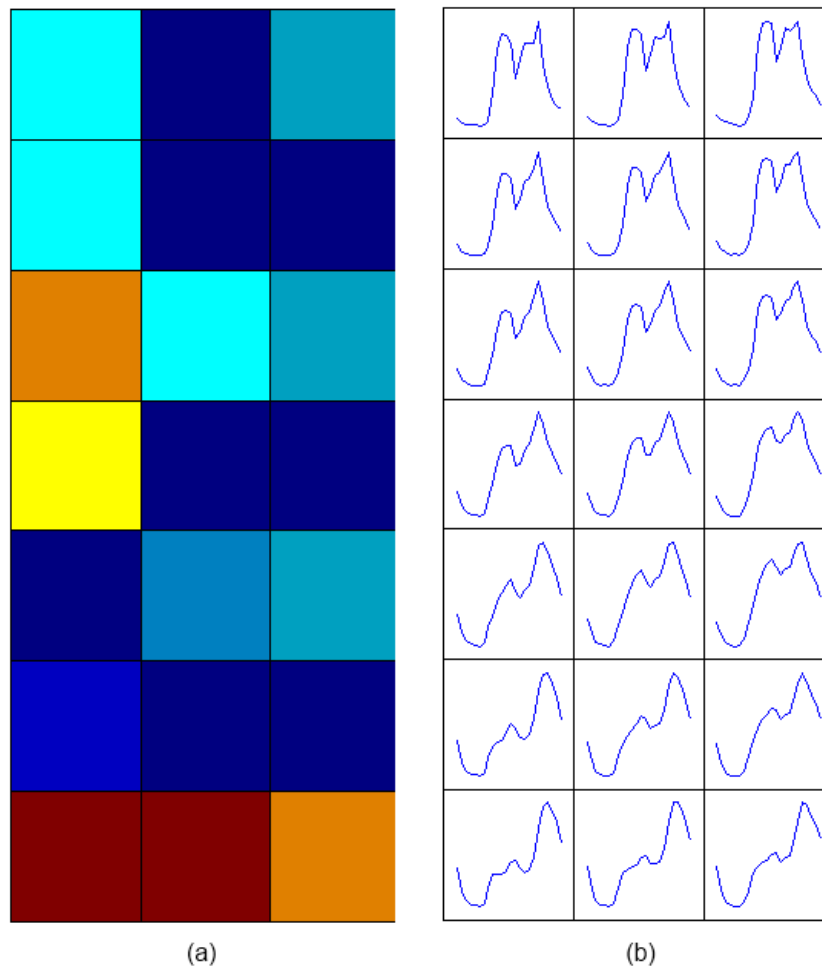


Figura 5-41 - Matriz de distâncias (a) e Protótipos (b)

Ao analisar a figura 5-42, percebe-se que a quantidade agrupamentos encontrados pelo algoritmo *k-means* é demasiado elevado pois passamos a ter elementos de agrupamentos não contíguos, e analisando o que já se tinha visto na figura 5-39 (a) seria melhor fazer uma separação com o *k-means* mas com um índice de *Davies-Bouldin* menor, neste caso o índice imediatamente abaixo é quando $k=3$, resultando numa situação expressa na figura 5-43.

Testes e análise de resultados

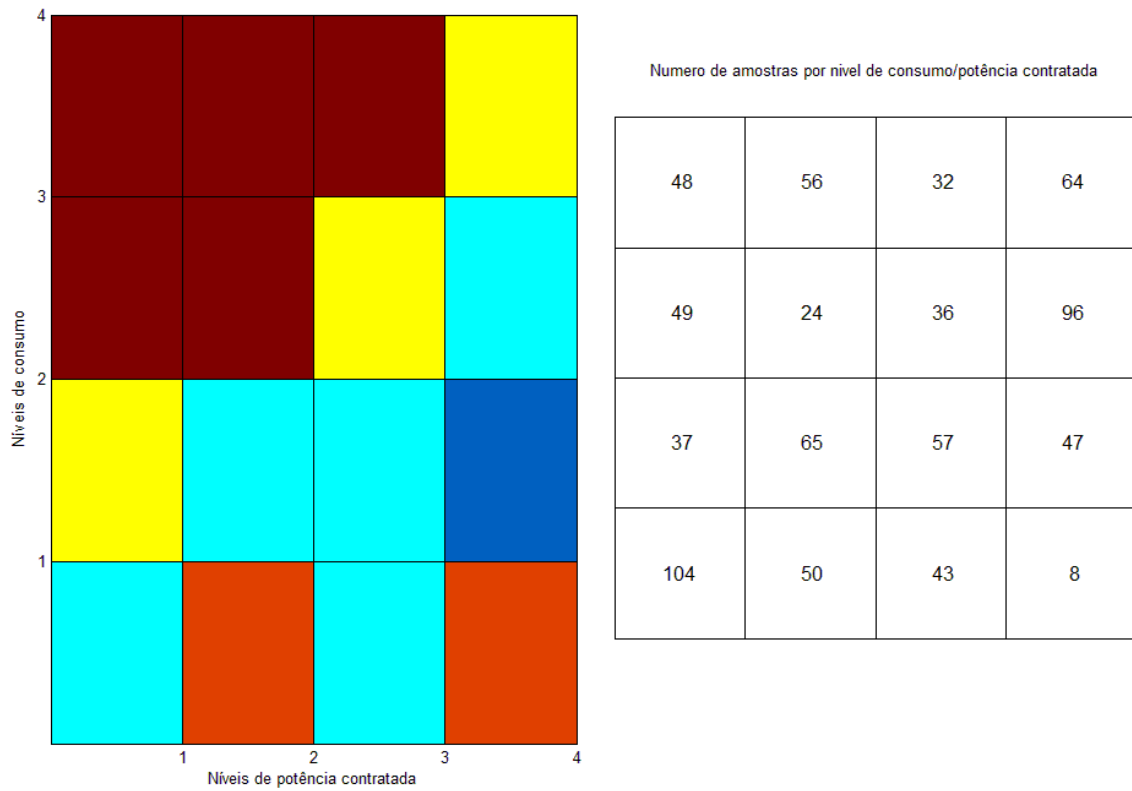


Figura 5-42 - Agrupamentos formados na grelha com os níveis de consumo e potência contratada, e quantidade de elementos em cada posição para k=5

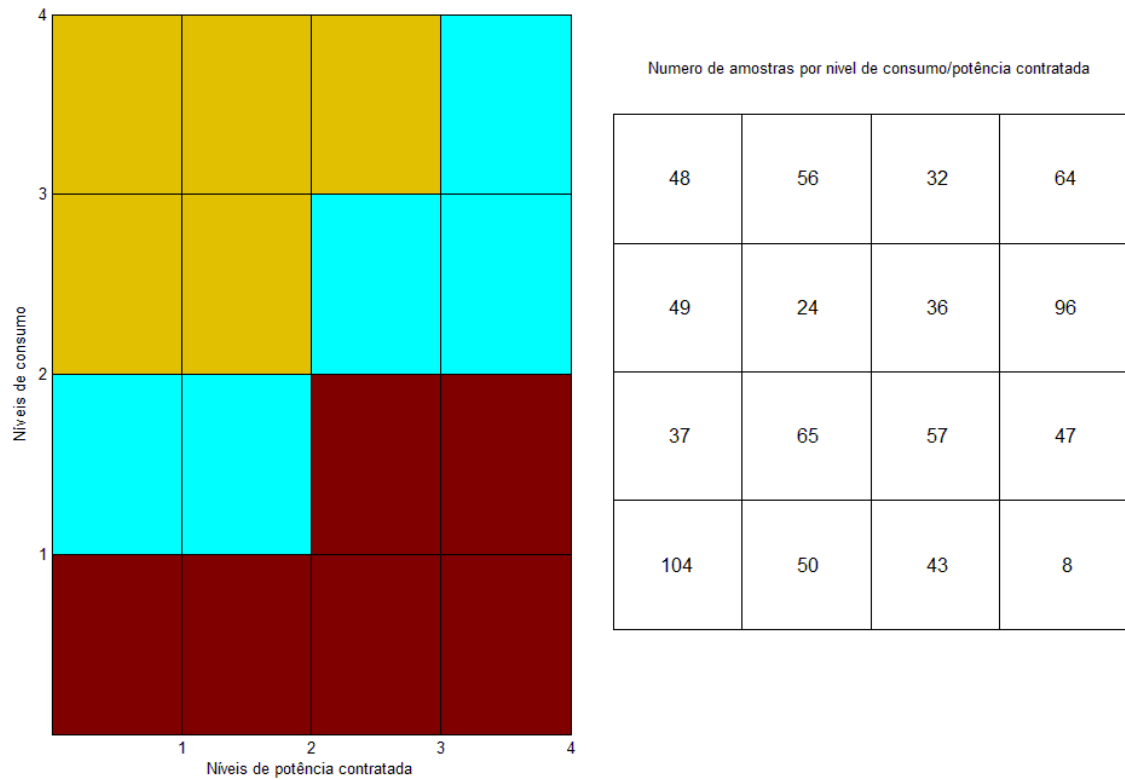


Figura 5-43 - Agrupamentos formados na grelha com os níveis de consumo e potência contratada, e quantidade de elementos em cada posição para k=3

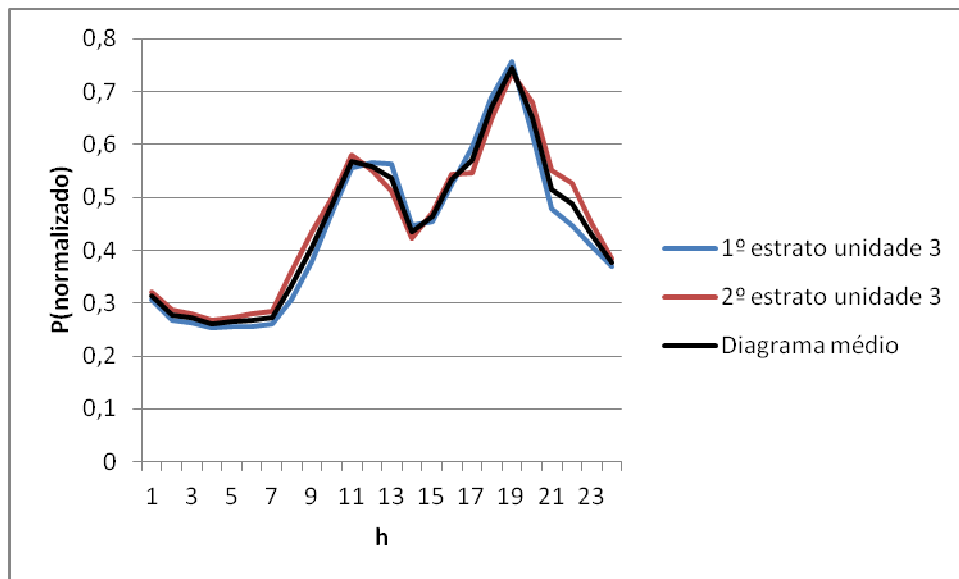


Figura 5-44 - Diagramas das divisões colocadas na unidade 3 do SOM e o seu diagrama médio

Nos gráficos das figuras 5-44 e 5-45, confirma-se a separação efetuada aproximam-se de perfis de consumo típicos.

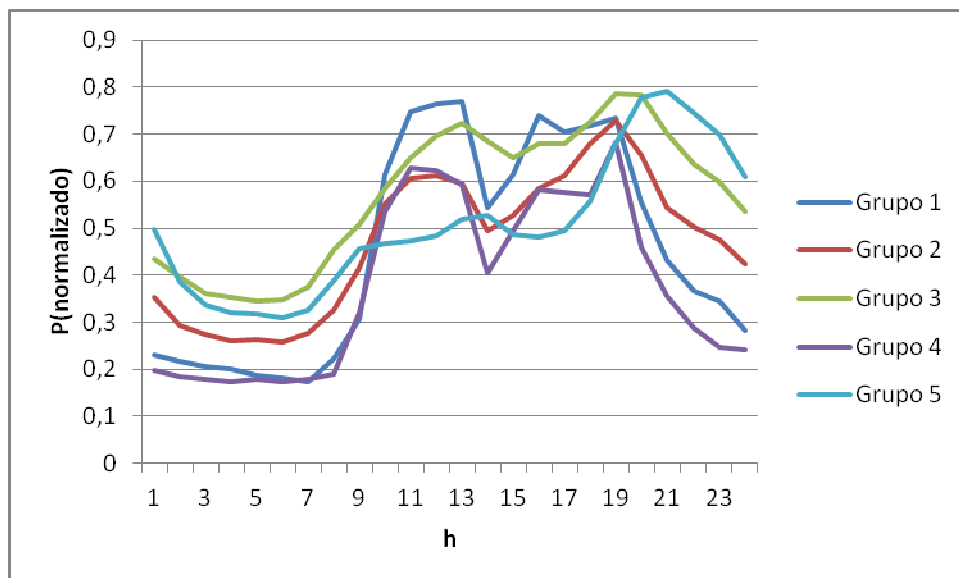


Figura 5-45 - Gráfico com os perfis de consumo criados com *k-means*(k=5)

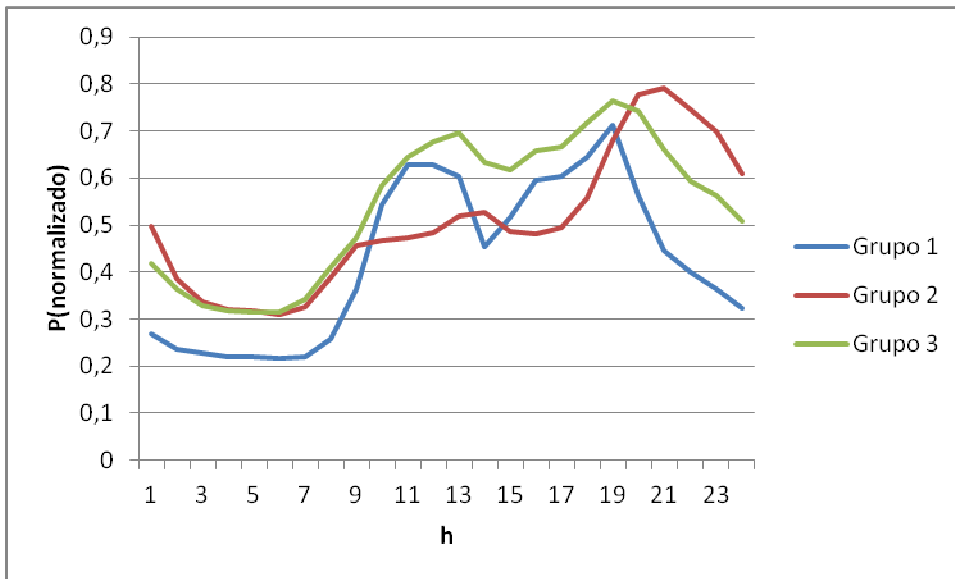


Figura 5-46 - Gráfico com os perfis de consumo criados com *k-means*(k=3)

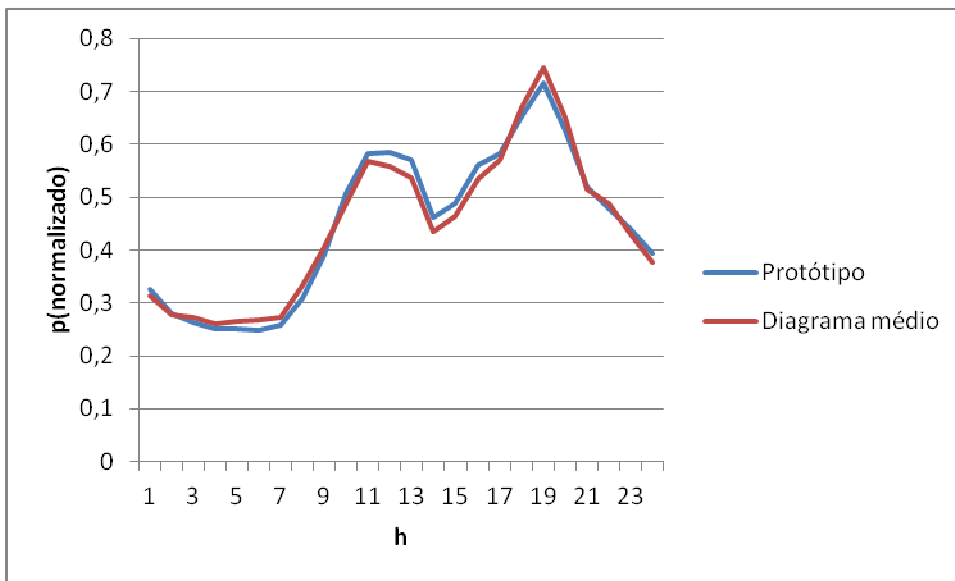


Figura 5-47 - Gráfico com o protótipo e diagrama médio da unidade 3 do SOM

Capítulo 6

Conclusão e trabalhos futuros

Os resultados obtidos parecem demonstrar a possibilidade de usar o algoritmo mapa de *Kohonen* para determinar perfis de consumo, principalmente combinado com o algoritmo *k-means*, pois este consegue-se retirar alguma ambiguidade à criação de agrupamentos.

Ficou também comprovado que com a normalização dos valores de entrada conjugada com outros fatores pode ser benéfica para o estudo, embora também apresente desvantagens pelo que é necessário ter cuidado com a sua utilização, pois em alguns casos a separação de classes de perfis de consumo resulta numa forma menos direta.

Quanto à segunda parte do trabalho, teoricamente também fica comprovado os seus pressupostos. A divisão apriorística efetuada foi simplista mas mesmo assim conseguiu-se resultados bastante interessantes.

Os objetivos propostos para este trabalho, era validar de uma metodologia. Os resultados obtidos na segunda parte mostraram que é possível obter perfis de consumo a partir das características dos diagramas mas classificados com base em dados de faturação.

Este estudo abre perspetivas para trabalhos futuros. Por exemplo num novo estudo com um conjunto de dados maiores e/ou com uma divisão apriorística mais alargada, isto é, considerando diferentes números de escalões de consumo e de potência contratada, e considerar um ano inteiro de dados. Por outro lado também se pode aplicar outros algoritmos de *clustering* e comparar resultados entre diferentes algoritmos.

Referências

- [1] José Nuno Fidalgo, Manuel António Matos, Luís Ribeiro, “A new clustering algorithm for load profiling based on billing data”, Elsevier, 2011
- [2] Fátima Rodrigues, Jorge Duarte, Vera Figueiredo, Zita Vale, M. Cordeiro, “A Comparative Analysis of Clustering Algorithms Applied to Load Profiling”, Springer-Verlag, 2003
- [3] J. Llamas, D.Sáez, R. Palma-Behnke, A. Núñez, G. Jimenez-Estévez, “Load Profile Generator and Load Forecasting for a Renewable Based Microgrid Using Self Organizing Maps and Neural Networks”, IEEE The 2012 International Joint Conference on Neural Networks (IJCNN), Junho 2012
- [4] Antti Mutanen, Maija Ruska, Sami Repo, Pertti Jarventausta, “Customer Classification and Load Profiling Method for Distribution Systems”, IEEE, 2011
- [5] I. P. Panapakidis, M. C. Alexiadis, G. K. Papagiannis, “Load Profiling in the Deregulated Electricity Markets: A Review of the Applications”, IEEE 2012 9th International Conference on the European Energy Market (EEM), 2012
- [6] Z. Zakaria, K. L. Lo, “Two-stage Fuzzy Clustering Approach for Load Profiling”, IEEE 2009
- [7] Diretiva 2009/72/EC do Conselho e Parlamento Europeu de 13 de julho de 2009
- [8] Maine Public Service Company, 2Maine Public Service Company Settlement and Load profiling Methodology”, Capítulo 321 2º relatório, 2000
- [9] R. Yao, K. Steemers, “A method of formulating energy load profile for domestic buildings in UK”, International Journal of Energy and Buildings, 2000
- [10] Mihai Gavrilas, Ovidiu Ivanov, “Load Profiling by Self Organization with Affinity Control Strategy”, Rev. Roum. Sci. Techn. - Électrotechn. et Énerg., 53, 4, p. 413-421, Bucarest, 2008
- [11] Sergio Carmini Cerchiari, “Determinação de Curvas de demanda de Consumidores de Baixa Tensão Utilizando Mapas Auto-Organizáveis (SOM) Para Agrupamentos e Conjuntos Aproximados Para Classificação de Consumidores”, Universidade Federal de Mato Grosso do Sul, Brasil, 2006
- [12] Khadir M. Tarek, Khairia Sofiane, Benabbas Farouk, “Kohonen Maps Combined to K-means in a Two Level Strategy for time Series Clustering Application to Meteorological and Electricity Load data”, Intech Austria, 2010
- [13] G. Chicco, R. Napoli, F. Piglione, “Comparison among clustering techniques for electricity customer classification”, IEEE Trans. Power Syst., vol. 21, nº2, maio 2006

- [14]G. Chicco, R. Napoli, F. Piglione, “Application of clustering algorithms and self organizing maps to classify electricity customers”, conferência IEEE PowerTech, Bolonha, Italia, junho 2003
- [15]A. Seppala, “Load research and load estimation in electricity distribution”, Dissertação de Doutorado, Helsinki Univ. Technol., Espoo, Finlandia, 1996
- [16]G. J. Tsekouras, N. D. Hatziaargyriou, E. N. Dialynas, “Two-stage pattern recognition of load curves for classification of electricity customers”, IEEE Trans. Power Syst., vol. 22, no. 3, agosto 2007
- [17]Z. Zakaria, M. N. Othman, M.H. Sohod, “Consumer load profiling using fuzzy clustering and statistical approach”, Proc. 4th Student Conf. Res. Develop., Selangor, Malasia, 2006
- [18]I. H. Yu, J. K. Lee, J. M. Ko, S. I. Kim, “A method for classification of electricity demands usng load profile data”, Proc. 4th Annu. ACIS Int. Conf. Comput. Inf. Sci., ilha Jeju, Coreia do Sul, 2005
- [19]B. D. Pitt, D. S. Kirschen, “Application of data mining techniques to load profiling”, Proc. 21st IEEE Int. Conf. Power Ind. Comput. Appl., Santa Clara, California, 1999
- [20]S. Ramos, Z. Vale, “Data mining techniques application in power distribution utilities”, IEEE Power Energy Soc. Transm. Distrib. Conf. Expo., Chicago, Illinóis, 2008
- [21]S. V. Verdú, M. O. García, F. J. G. Franco, N. Encinas, A. G. Marín, A. Molina, E. G. Lázaro, “Characterization and identification of electrical customers through the use of self-organizing maps anda daily load parameters” Proc. IEEE Power Eng. Soc. Power System Conf. Expo., Atlanta, Georgia, 2004
- [22]K. L. Lo, Z. Zakaria, “Electricity consumer classification using artificial intelligence”, Proc. 39th Int. Univ. Power Eng. Conf., Bristol, Reino Unido, 2004
- [23]D. Gerbec, S. Gasperic, I. Smon, F. Gubina, “Determining the load profiles of consumers based on fuzzy logic and probability neural networks”, Proc. Inst. Elect. Eng., Gen., Transm. Distrib., vol. 151, maio 2004
- [24]N. Mahmoudi-Kohan, M. P. Moghaddam, S. M. Bidaki, “Evaluating performance of WFA K-means and modified follow the leader methods for clustering load curves” IEEE/Power Energy Soc. Power Syst. Conf. Expo., Seattle, Washington, 2009
- [25]Franz Aurenhammer. “*Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure*”, ACM Computing Surveys, 1991
- [26]Zhang, T. Ramakrishnon, R., Livny M., “BIRCH: An Efficient Data Clustering Method for Very Large Databases”, Proc. ACM SIGMOD Conference on Management of Data, Montreal, Canada, 1996
- [27]Teuvo Kohonen, “Self-Organizing Maps”, 3rd edition, Springer, 2001
- [28]Pang-Ning Tan, Michael Steinbach, Vipin Kumar, “Introduction to Data Mining”, Addison-Wesley, 2006
- [29]Machaon CVE, “Cluster validity algorithms”. Disponível em http://machaon.karanagai.com/validation_algorithms.html . Acedido em 20/junho/2012
- [30]Davies, David L., Bouldin, “IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE”, VOL. PAMI-1, NO. 2, abril 1979
- [31]Sergios Theodoridis, Konstantinos Koutroumbas, “Pattern Recognition”, Third Edition, Academic Press, 2006
- [32]Manual e Tutorial da SOM Toolbox (Abril 2000), Disponível em <http://www.cis.hut.fi/projects/somtoolbox/download/>. Acedido em 23/fevereiro/2012.

Anexos

Código fonte do teste t1:

```
clc;
clear;
close all;
% Lê dados
sD = som_read_data('tudo.data');
% Cria o Mapa
sM = som_make(sD,'long','rect');
sM = som_autolabel(sM,sD,'add1');
% Visualizar o mapa...
f1=figure;
som_show(sM, 'umat','all');
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

f2=figure;
[Pd,V,me,l] = pcaproj(sD,2); Pm = pcaproj(sM,V,me); % PC-projection
Code = som_colorcode(Pm); % color coding
hits = som_hits(sM,sD); % hits
U = som_umat(sM); % U-matrix
Dm = U(1:2:size(U,1),1:2:size(U,2)); % distance matrix
Dm = 1-Dm(:)/max(Dm(:)); Dm(find(hits==0)) = 0; % clustering info
subplot(1,2,1)
som_cplane(sM,Code,Dm);
hold on
som_grid(sM,'Label',cellstr(int2str(hits)),...
        'Line','none','Marker','none','Labelcolor','k');
hold off
title('Código de cores')
subplot(1,2,2)
som_grid(sM,'Coord',Pm,'MarkerColor',Code,'Linecolor','k');
hold on, plot(Pd(:,1),Pd(:,2),'k+'), hold off, axis tight, axis equal
title('Projeção de componentes principais')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

[c,p,err,ind] = kmeans_clusters(sM,10,[100]); % find at most 10 clusters

f3=figure;
subplot(1,2,1)
plot(1:length(ind),ind,'x-')
```

```

[dummy,k] = min(ind);
cl = p{k};
title('Índices de Davies-Bouldin')
subplot(1,2,2)
som_cplane(sM,cl)
title('Agrupamentos com menor índice Davies-Bouldin');
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

f4=figure;
subplot(1,2,1)
som_cplane(sM,Dm(:));
title('Matriz de distâncias')
subplot(1,2,2)
som_plotplane(sM, sM.codebook,'b');
hold on
r=sM.topol.msize(1,1);
s=sM.topol.msize(1,2);
som_cplane('rect',[r s],'none')
hold off
title('Protótipos')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

% extrair diagramas
[Vec,lnx]=som_divide(sM, sD);
[a b]=size(Vec);
for k=1:50;
    if isempty(Vec{k,1})==1;
        else
            xlswrite('C:\Users\Rui\Desktop\analise\livro1_t1.xlsx',Vec{k,1},k);
        end
    end
end
for k=51:100;
    if isempty(Vec{k,1})==1;
        else
            xlswrite('C:\Users\Rui\Desktop\analise\livro2_t1.xlsx',Vec{k,1},k-50);
        end
    end
end
for k=101:a;
    if isempty(Vec{k,1})==1;
        else
            xlswrite('C:\Users\Rui\Desktop\analise\livro3_t1.xlsx',Vec{k,1},k-100);
        end
    end
end

%Diagramas médios dos agrupamentos criados pela som_make
[Vec,lnx]=som_divide(sM, sD);

```

```

[a b]=size(Vec);
Msm=zeros(a,24);
for e=1:a;
    if isempty(Vec{e,1})==0
        Msm(e,:)=mean(Vec{e,1},1);
    elseif isempty(Vec{e,1})==1
        Msm(e,:)=zeros(1,24);
    end
end

% Diagrams médios dos agrupamentos criados após kmeans_clusters
% -----Máximo de k=10-----
m1=zeros(a,24);
n1=0;
m2=zeros(a,24);
n2=0;
m3=zeros(a,24);
n3=0;
m4=zeros(a,24);
n4=0;
m5=zeros(a,24);
n5=0;
m6=zeros(a,24);
n6=0;
m7=zeros(a,24);
n7=0;
m8=zeros(a,24);
n8=0;
m9=zeros(a,24);
n9=0;
m10=zeros(a,24);
n10=0;

for z=1:a
    if cl(z)==1
        m1(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n1=n1+size(Vec{z,1},1);
    elseif cl(z)==2
        m2(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n2=n2+size(Vec{z,1},1);
    elseif cl(z)==3
        m3(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n3=n3+size(Vec{z,1},1);
    elseif cl(z)==4
        m4(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n4=n4+size(Vec{z,1},1);
    end
end

```

```

elseif cl(z)==5
    m5(z,:)=Msm(z,)*size(Vec{z,1},1);
    n5=n5+size(Vec{z,1},1);
elseif cl(z)==6
    m6(z,:)=Msm(z,)*size(Vec{z,1},1);
    n6=n6+size(Vec{z,1},1);
elseif cl(z)==7
    m7(z,:)=Msm(z,)*size(Vec{z,1},1);
    n7=n7+size(Vec{z,1},1);
elseif cl(z)==8
    m8(z,:)=Msm(z,)*size(Vec{z,1},1);
    n8=n8+size(Vec{z,1},1);
elseif cl(z)==9
    m9(z,:)=Msm(z,)*size(Vec{z,1},1);
    n9=n9+size(Vec{z,1},1);
elseif cl(z)==10
    m10(z,:)=Msm(z,)*size(Vec{z,1},1);
    n10=n10+size(Vec{z,1},1);
end
end
M1=sum(m1);
M1=M1(1,+)/n1;
M2=sum(m2);
M2=M2(1,+)/n2;
M3=sum(m3);
M3=M3(1,+)/n3;
M4=sum(m4);
M4=M4(1,+)/n4;
M5=sum(m5);
M5=M5(1,+)/n5;
M6=sum(m6);
M6=M6(1,+)/n6;
M7=sum(m7);
M7=M7(1,+)/n7;
M8=sum(m8);
M8=M8(1,+)/n8;
M9=sum(m9);
M9=M9(1,+)/n9;
M10=sum(m10);
M10=M10(1,+)/n10;

xlswrite('C:\Users\Rui\Desktop\analyse\livro4_t1.xlsx',Msm);
xlswrite('C:\Users\Rui\Desktop\analyse\livro5_t1.xlsx',M1,'A1:X1');
xlswrite('C:\Users\Rui\Desktop\analyse\livro5_t1.xlsx',M2,'A2:X2');
xlswrite('C:\Users\Rui\Desktop\analyse\livro5_t1.xlsx',M3,'A3:X3');
xlswrite('C:\Users\Rui\Desktop\analyse\livro5_t1.xlsx',M4,'A4:X4');

```

```

xlswrite('C:\Users\Rui\Desktop\analise\livro5_t1.xlsx',M5,'A5:X5');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t1.xlsx',M6,'A6:X6');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t1.xlsx',M7,'A7:X7');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t1.xlsx',M8,'A8:X8');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t1.xlsx',M9,'A9:X9');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t1.xlsx',M10,'A10:X10');
xlswrite('C:\Users\Rui\Desktop\analise\livro6_t1.xlsx',sM.codebook);

saveas(f1,'C:\Users\Rui\Desktop\analise\f1_t1.bmp')
saveas(f2,'C:\Users\Rui\Desktop\analise\f2_t1.bmp')
saveas(f3,'C:\Users\Rui\Desktop\analise\f3_t1.bmp')
saveas(f4,'C:\Users\Rui\Desktop\analise\f4_t1.bmp')

save 'C:\Users\Rui\Desktop\analise\t1.mat'

```

Código fonte do teste t1n:

```

clc;
clear;
close all;
% Lê dados
sD = som_read_data('tudo.data');
%%normaliza valores
normalizar;
% Cria o Mapa
sM = som_make(sD,'long','rect');
sM = som_autolabel(sM,sD,'add1');
% Visualizar o mapa...
f1=figure;
som_show(sM, 'umat','all');
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

f2=figure;
[Pd,V,me,l] = pcaproj(sD,2); Pm = pcaproj(sM,V,me); % PC-projection
Code = som_colorcode(Pm); % color coding
hits = som_hits(sM,sD); % hits
U = som_umat(sM); % U-matrix
Dm = U(1:2:size(U,1),1:2:size(U,2)); % distance matrix
Dm = 1-Dm(:)/max(Dm(:)); Dm(find(hits==0)) = 0; % clustering info
subplot(1,2,1)
som_cplane(sM,Code,Dm);
hold on
som_grid(sM,'Label',cellstr(int2str(hits)),...
        'Line','none','Marker','none','Labelcolor','k');
hold off

```

```

title('Código de cores')
subplot(1,2,2)
som_grid(sM,'Coord',Pm,'MarkerColor',Code,'Linecolor','k');
hold on, plot(Pd(:,1),Pd(:,2),'k+'), hold off, axis tight, axis equal
title('Projeção de componentes principais')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

[c,p,err,ind] = kmeans_clusters(sM,10,[100]); % find at most 10 clusters

f3=figure;
subplot(1,2,1)
plot(1:length(ind),ind,'x-')
[dummy,k] = min(ind);
cl = p{k};
title('Índices de Davies-Bouldin')
subplot(1,2,2)
som_cplane(sM,cl)
title('Agrupamentos com menor índice Davies-Bouldin');
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

f4=figure;
subplot(1,2,1)
som_cplane(sM,Dm(:));
title('Matriz de distâncias')
subplot(1,2,2)
som_plotplane(sM, sM.codebook,'b');
hold on
r=sM.topol.msize(1,1);
s=sM.topol.msize(1,2);
som_cplane('rect',[r s],'none')
hold off
title('Protótipos')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

% extrair diagramas
[Vec,lnx]=som_divide(sM, sD);
[a b]=size(Vec);
for k=1:50;
    if isempty(Vec{k,1})==1;
        else
            xlswrite('C:\Users\Rui\Desktop\analise\livro1_tn1.xlsx',Vec{k,1},k);
        end
    end
end
for k=51:100;
    if isempty(Vec{k,1})==1;
        else

```

```

        xlswrite('C:\Users\Rui\Desktop\analise\livro2_tn1.xlsx',Vec{k,1},k-50);
    end
end
for k=101:a;
    if isempty(Vec{k,1})==1;
        else
            xlswrite('C:\Users\Rui\Desktop\analise\livro3_tn1.xlsx',Vec{k,1},k-100);
        end
    end
end

```

%Diagramas médios dos agrupamentos criados pela som_make

```

[Vec,Inx]=som_divide(sM, sD);
[a b]=size(Vec);
Msm=zeros(a,24);
for e=1:a;
    if isempty(Vec{e,1})==0
        Msm(e,:)=mean(Vec{e,1},1);
    elseif isempty(Vec{e,1})==1
        Msm(e,:)=zeros(1,24);
    end
end
end

```

% Diagrams médios dos agrupamentos criados após kmeans_clusters

% -----Máximo de k=10-----

```

m1=zeros(a,24);
n1=0;
m2=zeros(a,24);
n2=0;
m3=zeros(a,24);
n3=0;
m4=zeros(a,24);
n4=0;
m5=zeros(a,24);
n5=0;
m6=zeros(a,24);
n6=0;
m7=zeros(a,24);
n7=0;
m8=zeros(a,24);
n8=0;
m9=zeros(a,24);
n9=0;
m10=zeros(a,24);
n10=0;

```

```

for z=1:a

```

```

if cl(z)==1
    m1(z,:)=Msm(z,)*size(Vec{z,1},1);
    n1=n1+size(Vec{z,1},1);
elseif cl(z)==2
    m2(z,:)=Msm(z,)*size(Vec{z,1},1);
    n2=n2+size(Vec{z,1},1);
elseif cl(z)==3
    m3(z,:)=Msm(z,)*size(Vec{z,1},1);
    n3=n3+size(Vec{z,1},1);
elseif cl(z)==4
    m4(z,:)=Msm(z,)*size(Vec{z,1},1);
    n4=n4+size(Vec{z,1},1);
elseif cl(z)==5
    m5(z,:)=Msm(z,)*size(Vec{z,1},1);
    n5=n5+size(Vec{z,1},1);
elseif cl(z)==6
    m6(z,:)=Msm(z,)*size(Vec{z,1},1);
    n6=n6+size(Vec{z,1},1);
elseif cl(z)==7
    m7(z,:)=Msm(z,)*size(Vec{z,1},1);
    n7=n7+size(Vec{z,1},1);
elseif cl(z)==8
    m8(z,:)=Msm(z,)*size(Vec{z,1},1);
    n8=n8+size(Vec{z,1},1);
elseif cl(z)==9
    m9(z,:)=Msm(z,)*size(Vec{z,1},1);
    n9=n9+size(Vec{z,1},1);
elseif cl(z)==10
    m10(z,:)=Msm(z,)*size(Vec{z,1},1);
    n10=n10+size(Vec{z,1},1);
end
end
M1=sum(m1);
M1=M1(1,)/n1;
M2=sum(m2);
M2=M2(1,)/n2;
M3=sum(m3);
M3=M3(1,)/n3;
M4=sum(m4);
M4=M4(1,)/n4;
M5=sum(m5);
M5=M5(1,)/n5;
M6=sum(m6);
M6=M6(1,)/n6;
M7=sum(m7);
M7=M7(1,)/n7;

```



```

M8=sum(m8);
M8=M8(1,:)/n8;
M9=sum(m9);
M9=M9(1,:)/n9;
M10=sum(m10);
M10=M10(1,:)/n10;

```

```

xlswrite('C:\Users\Rui\Desktop\analise\livro4_tn1.xlsx',Msm);
xlswrite('C:\Users\Rui\Desktop\analise\livro5_tn1.xlsx',M1,'A1:X1');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_tn1.xlsx',M2,'A2:X2');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_tn1.xlsx',M3,'A3:X3');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_tn1.xlsx',M4,'A4:X4');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_tn1.xlsx',M5,'A5:X5');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_tn1.xlsx',M6,'A6:X6');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_tn1.xlsx',M7,'A7:X7');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_tn1.xlsx',M8,'A8:X8');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_tn1.xlsx',M9,'A9:X9');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_tn1.xlsx',M10,'A10:X10');
xlswrite('C:\Users\Rui\Desktop\analise\livro6_tn1.xlsx',sM.codebook);

```

```

saveas(f1,'C:\Users\Rui\Desktop\analise\f1_tn1.bmp')
saveas(f2,'C:\Users\Rui\Desktop\analise\f2_tn1.bmp')
saveas(f3,'C:\Users\Rui\Desktop\analise\f3_tn1.bmp')
saveas(f4,'C:\Users\Rui\Desktop\analise\f4_tn1.bmp')

```

```

extrai_t1n_d;

```

```

save 'C:\Users\Rui\Desktop\analise\t1n.mat'

```

Código fonte do teste t2:

```

clc;
clear;
close all;

% Lê dados
sD = som_read_data('todo.data');
x = input('Numero de agrupamentos= ');
%criar o Mapa
sM = som_make(sD,'long','rect','munits',x);
sM = som_autolabel(sM,sD,'add1');
% Visualizar o mapa...
f1=figure;
som_show(sM, 'umat','all');
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

```

```

f2=figure;
[Pd,V,me,l] = pcaproj(sD,2); Pm = pcaproj(sM,V,me); % PC-projection
Code = som_colorcode(Pm); % color coding
hits = som_hits(sM,sD); % hits
U = som_umat(sM); % U-matrix
Dm = U(1:2:size(U,1),1:2:size(U,2)); % distance matrix
Dm = 1-Dm(:)/max(Dm(:)); Dm(find(hits==0)) = 0; % clustering info
subplot(1,2,1)
som_cplane(sM,Code,Dm);
hold on
som_grid(sM,'Label',cellstr(int2str(hits)),...
         'Line','none','Marker','none','Labelcolor','k');
hold off
title('Código de cores')
subplot(1,2,2)
som_grid(sM,'Coord',Pm,'MarkerColor',Code,'Linecolor','k');
hold on, plot(Pd(:,1),Pd(:,2),'k+'), hold off, axis tight, axis equal
title('Projeção de componentes principais')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

[c,p,err,ind] = kmeans_clusters(sM,10,[100]); % find at most 10 clusters

f3=figure;
subplot(1,2,1)
plot(1:length(ind),ind,'x-')
[dummy,k] = min(ind);
cl = p{k};
title('Índices de Davies-Bouldin')
subplot(1,2,2)
som_cplane(sM,cl)
title('Agrupamentos com menor índice Davies-Bouldin');
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

f4=figure;
subplot(1,2,1)
som_cplane(sM,Dm(:));
title('Matriz de distâncias')
subplot(1,2,2)
som_plotplane(sM, sM.codebook,'b');
hold on
r=sM.topol.msize(1,1);
s=sM.topol.msize(1,2);
som_cplane('rect',[r s],'none')
hold off
title('Protótipos')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

```

```

% extrair diagramas
[Vec,lnx]=som_divide(sM, sD);
[a b]=size(Vec);
for k=1:a;
    if isempty(Vec{k,1})==1;
        else
            xlswrite('C:\Users\Rui\Desktop\analise\livro1_t2.xlsx',Vec{k,1},k);
        end
    end
end
%Diagramas médios dos agrupamentos criados pela som_make
Msm=zeros(a,24);
for e=1:a;
    if isempty(Vec{e,1})==0
        Msm(e,:)=mean(Vec{e,1},1);
    elseif isempty(Vec{e,1})==1
        Msm(e,:)=zeros(1,24);
    end
end
end

% Diagrams médios dos agrupamentos criados após kmeans_clusters
% -----Máximo de k=10-----
m1=zeros(a,24);
n1=0;
m2=zeros(a,24);
n2=0;
m3=zeros(a,24);
n3=0;
m4=zeros(a,24);
n4=0;
m5=zeros(a,24);
n5=0;
m6=zeros(a,24);
n6=0;
m7=zeros(a,24);
n7=0;
m8=zeros(a,24);
n8=0;
m9=zeros(a,24);
n9=0;
m10=zeros(a,24);
n10=0;

for z=1:a
    if cl(z)==1
        m1(z,:)=Msm(z,:)*size(Vec{z,1},1);
    end
end

```

```

    n1=n1+size(Vec{z,1},1);
elseif cl(z)==2
    m2(z,:)=Msm(z,)*size(Vec{z,1},1);
    n2=n2+size(Vec{z,1},1);
elseif cl(z)==3
    m3(z,:)=Msm(z,)*size(Vec{z,1},1);
    n3=n3+size(Vec{z,1},1);
elseif cl(z)==4
    m4(z,:)=Msm(z,)*size(Vec{z,1},1);
    n4=n4+size(Vec{z,1},1);
elseif cl(z)==5
    m5(z,:)=Msm(z,)*size(Vec{z,1},1);
    n5=n5+size(Vec{z,1},1);
elseif cl(z)==6
    m6(z,:)=Msm(z,)*size(Vec{z,1},1);
    n6=n6+size(Vec{z,1},1);
elseif cl(z)==7
    m7(z,:)=Msm(z,)*size(Vec{z,1},1);
    n7=n7+size(Vec{z,1},1);
elseif cl(z)==8
    m8(z,:)=Msm(z,)*size(Vec{z,1},1);
    n8=n8+size(Vec{z,1},1);
elseif cl(z)==9
    m9(z,:)=Msm(z,)*size(Vec{z,1},1);
    n9=n9+size(Vec{z,1},1);
elseif cl(z)==10
    m10(z,:)=Msm(z,)*size(Vec{z,1},1);
    n10=n10+size(Vec{z,1},1);
end
end
M1=sum(m1);
M1=M1(1,)/n1;
M2=sum(m2);
M2=M2(1,)/n2;
M3=sum(m3);
M3=M3(1,)/n3;
M4=sum(m4);
M4=M4(1,)/n4;
M5=sum(m5);
M5=M5(1,)/n5;
M6=sum(m6);
M6=M6(1,)/n6;
M7=sum(m7);
M7=M7(1,)/n7;
M8=sum(m8);
M8=M8(1,)/n8;

```

```

M9=sum(m9);
M9=M9(1,+)/n9;
M10=sum(m10);
M10=M10(1,+)/n10;

```

```

xlswrite('C:\Users\Rui\Desktop\analise\livro4_t2.xlsx',Msm);
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2.xlsx',M1,'A1:X1');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2.xlsx',M2,'A2:X2');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2.xlsx',M3,'A3:X3');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2.xlsx',M4,'A4:X4');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2.xlsx',M5,'A5:X5');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2.xlsx',M6,'A6:X6');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2.xlsx',M7,'A7:X7');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2.xlsx',M8,'A8:X8');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2.xlsx',M9,'A9:X9');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2.xlsx',M10,'A10:X10');
xlswrite('C:\Users\Rui\Desktop\analise\livro6_t2.xlsx',sM.codebook);

```

```

saveas(f1,'C:\Users\Rui\Desktop\analise\f1_t2.bmp')
saveas(f2,'C:\Users\Rui\Desktop\analise\f2_t2.bmp')
saveas(f3,'C:\Users\Rui\Desktop\analise\f3_t2.bmp')
saveas(f4,'C:\Users\Rui\Desktop\analise\f4_t2.bmp')

```

```

save 'C:\Users\Rui\Desktop\analise\t2.mat'

```

Código fonte do teste t2n:

```

clc;
clear;
close all;

% Lê dados
sD = som_read_data('tudo.data');
%%normaliza valores
normalizar;
%criar o Mapa
x = input('Numero de agrupamentos= ');
sM = som_make(sD,'long','rect','munits',x);
sM = som_autolabel(sM,sD,'add1');
% Visualizar o mapa...
f1=figure;
som_show(sM, 'umat','all');
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

f2=figure;
[Pd,V,me,l] = pcaproj(sD,2); Pm = pcaproj(sM,V,me); % PC-projection

```

```

Code = som_colorcode(Pm); % color coding
hits = som_hits(sM,sD); % hits
U = som_umat(sM); % U-matrix
Dm = U(1:2:size(U,1),1:2:size(U,2)); % distance matrix
Dm = 1-Dm(:)/max(Dm(:)); Dm(find(hits==0)) = 0; % clustering info
subplot(1,2,1)
som_cplane(sM,Code,Dm);
hold on
som_grid(sM,'Label',cellstr(int2str(hits)),...
        'Line','none','Marker','none','Labelcolor','k');
hold off
title('Código de cores')
subplot(1,2,2)
som_grid(sM,'Coord',Pm,'MarkerColor',Code,'Linecolor','k');
hold on, plot(Pd(:,1),Pd(:,2),'k+'), hold off, axis tight, axis equal
title('Projeção de componentes principais')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

[c,p,err,ind] = kmeans_clusters(sM,10,[100]); % find at most 10 clusters

f3=figure;
subplot(1,2,1)
plot(1:length(ind),ind,'x-')
[dummy,k] = min(ind);
cl = p{k};
title('Índices de Davies-Bouldin')
subplot(1,2,2)
som_cplane(sM,cl)
title('Agrupamentos com menor índice Davies-Bouldin');
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

f4=figure;
subplot(1,2,1)
som_cplane(sM,Dm(:));
title('Matriz de distâncias')
subplot(1,2,2)
som_plotplane(sM, sM.codebook,'b');
hold on
r=sM.topol.msize(1,1);
s=sM.topol.msize(1,2);
som_cplane('rect',[r s],'none')
hold off
title('Protótipos')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

% extrair diagramas

```

```

[Vec,lnx]=som_divide(sM, sD);
[a b]=size(Vec);
for k=1:a;
    if isempty(Vec{k,1})==1;
        else
            xlsxwrite('C:\Users\Rui\Desktop\analise\livro1_t2n.xlsx',Vec{k,1},k);
        end
    end
end
%Diagramas médios dos agrupamentos criados pela som_make
Msm=zeros(a,24);
for e=1:a;
    if isempty(Vec{e,1})==0
        Msm(e,:)=mean(Vec{e,1},1);
    elseif isempty(Vec{e,1})==1
        Msm(e,:)=zeros(1,24);
    end
end
end

% Diagrams médios dos agrupamentos criados após kmeans_clusters
% -----Máximo de k=10-----
m1=zeros(a,24);
n1=0;
m2=zeros(a,24);
n2=0;
m3=zeros(a,24);
n3=0;
m4=zeros(a,24);
n4=0;
m5=zeros(a,24);
n5=0;
m6=zeros(a,24);
n6=0;
m7=zeros(a,24);
n7=0;
m8=zeros(a,24);
n8=0;
m9=zeros(a,24);
n9=0;
m10=zeros(a,24);
n10=0;

for z=1:a
    if cl(z)==1
        m1(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n1=n1+size(Vec{z,1},1);
    elseif cl(z)==2

```

```

        m2(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n2=n2+size(Vec{z,1},1);
elseif cl(z)==3
        m3(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n3=n3+size(Vec{z,1},1);
elseif cl(z)==4
        m4(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n4=n4+size(Vec{z,1},1);
elseif cl(z)==5
        m5(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n5=n5+size(Vec{z,1},1);
elseif cl(z)==6
        m6(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n6=n6+size(Vec{z,1},1);
elseif cl(z)==7
        m7(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n7=n7+size(Vec{z,1},1);
elseif cl(z)==8
        m8(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n8=n8+size(Vec{z,1},1);
elseif cl(z)==9
        m9(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n9=n9+size(Vec{z,1},1);
elseif cl(z)==10
        m10(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n10=n10+size(Vec{z,1},1);
end
end
M1=sum(m1);
M1=M1(1,+)/n1;
M2=sum(m2);
M2=M2(1,+)/n2;
M3=sum(m3);
M3=M3(1,+)/n3;
M4=sum(m4);
M4=M4(1,+)/n4;
M5=sum(m5);
M5=M5(1,+)/n5;
M6=sum(m6);
M6=M6(1,+)/n6;
M7=sum(m7);
M7=M7(1,+)/n7;
M8=sum(m8);
M8=M8(1,+)/n8;
M9=sum(m9);
M9=M9(1,+)/n9;

```



```
M10=sum(m10);
M10=M10(1,:)/n10;
```

```
xlswrite('C:\Users\Rui\Desktop\analise\livro4_t2n.xlsx',Msm);
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2n.xlsx',M1,'A1:X1');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2n.xlsx',M2,'A2:X2');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2n.xlsx',M3,'A3:X3');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2n.xlsx',M4,'A4:X4');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2n.xlsx',M5,'A5:X5');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2n.xlsx',M6,'A6:X6');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2n.xlsx',M7,'A7:X7');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2n.xlsx',M8,'A8:X8');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2n.xlsx',M9,'A9:X9');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_t2n.xlsx',M10,'A10:X10');
xlswrite('C:\Users\Rui\Desktop\analise\livro6_t2n.xlsx',sM.codebook);
```

```
saveas(f1,'C:\Users\Rui\Desktop\analise\f1_t2n.bmp')
saveas(f2,'C:\Users\Rui\Desktop\analise\f2_t2n.bmp')
saveas(f3,'C:\Users\Rui\Desktop\analise\f3_t2n.bmp')
saveas(f4,'C:\Users\Rui\Desktop\analise\f4_t2n.bmp')
```

```
extrai_t2n_d;
```

```
save 'C:\Users\Rui\Desktop\analise\t2n.mat'
```

Código fonte do teste parte2:

```
% Parte 2
clc;
clear;
close all;
%carrega os dados separados numa forma de grelha
M_p2;
% Cria o Mapa
sM = som_make(sD,'long','rect');
sM = som_autolabel(sM,sD,'add1');
% Visualizar o mapa...
f1=figure;
som_show(sM, 'umat','all');
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

f2=figure;
[Pd,V,me,l] = pcaproj(sD,2); Pm = pcaproj(sM,V,me); % PC-projection
Code = som_colorcode(Pm); % color coding
hits = som_hits(sM,sD); % hits
U = som_umat(sM); % U-matrix
```

```

Dm = U(1:2:size(U,1),1:2:size(U,2)); % distance matrix
Dm = 1-Dm(:)/max(Dm(:)); Dm(find(hits==0)) = 0; % clustering info
subplot(1,2,1)
som_cplane(sM,Code,Dm);
hold on
som_grid(sM,'Label',cellstr(int2str(hits)),...
         'Line','none','Marker','none','Labelcolor','k');
hold off
title('Código de cores')
subplot(1,2,2)
som_grid(sM,'Coord',Pm,'MarkerColor',Code,'Linecolor','k');
hold on, plot(Pd(:,1),Pd(:,2),'k+'), hold off, axis tight, axis equal
title('Projeção de componentes principais')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

[c,p,err,ind] = kmeans_clusters(sM,10,[100]); % find at most 10 clusters

f3=figure;
subplot(1,2,1)
plot(1:length(ind),ind,'x-')
[dummy,k] = min(ind);
cl = p{k};
title('Índices de Davies-Bouldin')
subplot(1,2,2)
som_cplane(sM,cl)
title('Agrupamentos com menor índice Davies-Bouldin');
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

f4=figure;
subplot(1,2,1)
som_cplane(sM,Dm(:));
title('Matriz de distâncias')
subplot(1,2,2)
som_plotplane(sM, sM.codebook,'b');
hold on
r=sM.topol.msize(1,1);
s=sM.topol.msize(1,2);
som_cplane('rect',[r s],'none')
hold off
title('Protótipos')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

% extrair diagramas
[Vec,lnx]=som_divide(sM, sD);
[a b]=size(Vec);
for k=1:a;

```

```

    if isempty(Vec{k,1})==1;
    else
        xlswrite('C:\Users\Rui\Desktop\analise\livro1_parte2.xlsx',Vec{k,1},k);
    end
end
%Diagramas médios dos agrupamentos criados pela som_make
Msm=zeros(a,24);
for e=1:a;
    if isempty(Vec{e,1})==0
        Msm(e,:)=mean(Vec{e,1},1);
    elseif isempty(Vec{e,1})==1
        Msm(e,:)=zeros(1,24);
    end
end
end

% Diagramas médios dos agrupamentos criados após kmeans_clusters
% -----Máximo de k=10-----
m1=zeros(a,24);
n1=0;
m2=zeros(a,24);
n2=0;
m3=zeros(a,24);
n3=0;
m4=zeros(a,24);
n4=0;
m5=zeros(a,24);
n5=0;
m6=zeros(a,24);
n6=0;
m7=zeros(a,24);
n7=0;
m8=zeros(a,24);
n8=0;
m9=zeros(a,24);
n9=0;
m10=zeros(a,24);
n10=0;

for z=1:a
    if cl(z)==1
        m1(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n1=n1+size(Vec{z,1},1);
    elseif cl(z)==2
        m2(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n2=n2+size(Vec{z,1},1);
    elseif cl(z)==3

```

```

        m3(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n3=n3+size(Vec{z,1},1);
elseif cl(z)==4
        m4(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n4=n4+size(Vec{z,1},1);
elseif cl(z)==5
        m5(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n5=n5+size(Vec{z,1},1);
elseif cl(z)==6
        m6(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n6=n6+size(Vec{z,1},1);
elseif cl(z)==7
        m7(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n7=n7+size(Vec{z,1},1);
elseif cl(z)==8
        m8(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n8=n8+size(Vec{z,1},1);
elseif cl(z)==9
        m9(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n9=n9+size(Vec{z,1},1);
elseif cl(z)==10
        m10(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n10=n10+size(Vec{z,1},1);
    end
end
M1=sum(m1);
M1=M1(1,:)/n1;
M2=sum(m2);
M2=M2(1,:)/n2;
M3=sum(m3);
M3=M3(1,:)/n3;
M4=sum(m4);
M4=M4(1,:)/n4;
M5=sum(m5);
M5=M5(1,:)/n5;
M6=sum(m6);
M6=M6(1,:)/n6;
M7=sum(m7);
M7=M7(1,:)/n7;
M8=sum(m8);
M8=M8(1,:)/n8;
M9=sum(m9);
M9=M9(1,:)/n9;
M10=sum(m10);
M10=M10(1,:)/n10;

```

```

xlswrite('C:\Users\Rui\Desktop\analise\livro4_parte2.xlsx',Msm);
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2.xlsx',M1,'A1:X1');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2.xlsx',M2,'A2:X2');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2.xlsx',M3,'A3:X3');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2.xlsx',M4,'A4:X4');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2.xlsx',M5,'A5:X5');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2.xlsx',M6,'A6:X6');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2.xlsx',M7,'A7:X7');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2.xlsx',M8,'A8:X8');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2.xlsx',M9,'A9:X9');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2.xlsx',M10,'A10:X10');
xlswrite('C:\Users\Rui\Desktop\analise\livro6_parte2.xlsx',sM.codebook);

```

```

saveas(f1,'C:\Users\Rui\Desktop\analise\f1_parte2.bmp')
saveas(f2,'C:\Users\Rui\Desktop\analise\f2_parte2.bmp')
saveas(f3,'C:\Users\Rui\Desktop\analise\f3_parte2.bmp')
saveas(f4,'C:\Users\Rui\Desktop\analise\f4_parte2.bmp')

```

```

save 'C:\Users\Rui\Desktop\analise\parte2.mat'

```

Código fonte do teste parte2n:

```

% Parte 2 normalizados
clc;
clear;
close all;
%carrega os dados separados numa forma de grelha
M_p2_n;
% Cria o Mapa
sM = som_make(sD,'long','rect');
sM = som_autolabel(sM,sD,'add1');
% Visualizar o mapa...
f1=figure;
som_show(sM, 'umat','all');
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

f2=figure;
[Pd,V,me,l] = pcaproj(sD,2); Pm = pcaproj(sM,V,me); % PC-projection
Code = som_colorcode(Pm); % color coding
hits = som_hits(sM,sD); % hits
U = som_umat(sM); % U-matrix
Dm = U(1:2:size(U,1),1:2:size(U,2)); % distance matrix
Dm = 1-Dm(:)/max(Dm(:)); Dm(find(hits==0)) = 0; % clustering info
subplot(1,2,1)
som_cplane(sM,Code,Dm);
hold on

```

```

som_grid(sM,'Label',cellstr(int2str(hits)),...
        'Line','none','Marker','none','Labelcolor','k');
hold off
title('Código de cores')
subplot(1,2,2)
som_grid(sM,'Coord',Pm,'MarkerColor',Code,'Linecolor','k');
hold on, plot(Pd(:,1),Pd(:,2),'k+'), hold off, axis tight, axis equal
title('Projeção de componentes principais')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

[c,p,err,ind] = kmeans_clusters(sM,10,[100]); % find at most 10 clusters

f3=figure;
subplot(1,2,1)
plot(1:length(ind),ind,'x-')
[dummy,k] = min(ind);
cl = p{k};
title('Índices de Davies-Bouldin')
subplot(1,2,2)
som_cplane(sM,cl)
title('Agrupamentos com menor índice Davies-Bouldin');
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

f4=figure;
subplot(1,2,1)
som_cplane(sM,Dm(:));
title('Matriz de distâncias')
subplot(1,2,2)
som_plotplane(sM, sM.codebook,'b');
hold on
r=sM.topol.msize(1,1);
s=sM.topol.msize(1,2);
som_cplane('rect',[r s],'none')
hold off
title('Protótipos')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.

% extrair diagramas
[Vec,Inx]=som_divide(sM, sD);
[a b]=size(Vec);
for k=1:a;
    if isempty(Vec{k,1})==1;
    else
        xlswrite('C:\Users\Rui\Desktop\analise\livro1_parte2n.xlsx',Vec{k,1},k);
    end
end
end

```

```
%Diagramas médios dos agrupamentos criados pela som_make
```

```
[Vec,Inx]=som_divide(sM, sD);
```

```
[a b]=size(Vec);
```

```
Msm=zeros(a,24);
```

```
for e=1:a;
```

```
    if isempty(Vec{e,1})==0
```

```
        Msm(e,:)=mean(Vec{e,1},1);
```

```
    elseif isempty(Vec{e,1})==1
```

```
        Msm(e,:)=zeros(1,24);
```

```
    end
```

```
end
```

```
% Diagramas médios dos agrupamentos criados após kmeans_clusters
```

```
% -----Máximo de k=10-----
```

```
m1=zeros(a,24);
```

```
n1=0;
```

```
m2=zeros(a,24);
```

```
n2=0;
```

```
m3=zeros(a,24);
```

```
n3=0;
```

```
m4=zeros(a,24);
```

```
n4=0;
```

```
m5=zeros(a,24);
```

```
n5=0;
```

```
m6=zeros(a,24);
```

```
n6=0;
```

```
m7=zeros(a,24);
```

```
n7=0;
```

```
m8=zeros(a,24);
```

```
n8=0;
```

```
m9=zeros(a,24);
```

```
n9=0;
```

```
m10=zeros(a,24);
```

```
n10=0;
```

```
for z=1:a
```

```
    if cl(z)==1
```

```
        m1(z,:)=Msm(z,:)*size(Vec{z,1},1);
```

```
        n1=n1+size(Vec{z,1},1);
```

```
    elseif cl(z)==2
```

```
        m2(z,:)=Msm(z,:)*size(Vec{z,1},1);
```

```
        n2=n2+size(Vec{z,1},1);
```

```
    elseif cl(z)==3
```

```
        m3(z,:)=Msm(z,:)*size(Vec{z,1},1);
```

```
        n3=n3+size(Vec{z,1},1);
```

```
    elseif cl(z)==4
```

```

        m4(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n4=n4+size(Vec{z,1},1);
    elseif cl(z)==5
        m5(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n5=n5+size(Vec{z,1},1);
    elseif cl(z)==6
        m6(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n6=n6+size(Vec{z,1},1);
    elseif cl(z)==7
        m7(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n7=n7+size(Vec{z,1},1);
    elseif cl(z)==8
        m8(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n8=n8+size(Vec{z,1},1);
    elseif cl(z)==9
        m9(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n9=n9+size(Vec{z,1},1);
    elseif cl(z)==10
        m10(z,:)=Msm(z,:)*size(Vec{z,1},1);
        n10=n10+size(Vec{z,1},1);
    end
end
M1=sum(m1);
M1=M1(1,:)/n1;
M2=sum(m2);
M2=M2(1,:)/n2;
M3=sum(m3);
M3=M3(1,:)/n3;
M4=sum(m4);
M4=M4(1,:)/n4;
M5=sum(m5);
M5=M5(1,:)/n5;
M6=sum(m6);
M6=M6(1,:)/n6;
M7=sum(m7);
M7=M7(1,:)/n7;
M8=sum(m8);
M8=M8(1,:)/n8;
M9=sum(m9);
M9=M9(1,:)/n9;
M10=sum(m10);
M10=M10(1,:)/n10;

xlswrite('C:\Users\Rui\Desktop\analise\livro4_parte2n.xlsx',Msm);
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2n.xlsx',M1,'A1:X1');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2n.xlsx',M2,'A2:X2');

```



```

xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2n.xlsx',M3,'A3:X3');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2n.xlsx',M4,'A4:X4');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2n.xlsx',M5,'A5:X5');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2n.xlsx',M6,'A6:X6');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2n.xlsx',M7,'A7:X7');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2n.xlsx',M8,'A8:X8');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2n.xlsx',M9,'A9:X9');
xlswrite('C:\Users\Rui\Desktop\analise\livro5_parte2n.xlsx',M10,'A10:X10');
xlswrite('C:\Users\Rui\Desktop\analise\livro6_parte2n.xlsx',sM.codebook);

```

```

saveas(f1,'C:\Users\Rui\Desktop\analise\f1_parte2n.bmp')
saveas(f2,'C:\Users\Rui\Desktop\analise\f2_parte2n.bmp')
saveas(f3,'C:\Users\Rui\Desktop\analise\f3_parte2n.bmp')
saveas(f4,'C:\Users\Rui\Desktop\analise\f4_parte2n.bmp')

```

```

extrai_p2n_d

```

```

save 'C:\Users\Rui\Desktop\analise\parte2.mat'

```

Código fonte da função normalizar:

```

%%Normalizar entre 0 a 1

M=max(sD.data, [], 2);
[i, j]=size(sD.data);

for x=1:i;
    for y=1:j;
        sD.data(x,y)=sD.data(x,y)/M(x);
    end;
end;

clear x y

```

Código fonte da função M_p2:

```

% Criar Grelha com os dados
dados=importdata('p2.mat'); % importa lista de consumo e potência contratada
[m, n] = size(dados);
grelha=zeros(m,n);
% Separa a Potência contratada Kva
for x=1:m;
    if dados(x,2)==1.15 || dados(x,2)==2.3 || dados(x,2)==3.45
        grelha(x,2)=1;
    elseif dados(x,2)==4.6 || dados(x,2)==5.75 || dados(x,2)==6.9

```

```

        grelha(x,2)=2;
    elseif dados(x,2)==10.35 || dados(x,2)==13.8 || dados(x,2)==17.25
        grelha(x,2)=3;
    elseif dados(x,2)==27.6 || dados(x,2)==34.5 || dados(x,2)==41.4 ||
dados(x,2)==20.7
        grelha(x,2)=4;
    end
end
% Separa o Consumo kWh
for x=1:m;
    if dados(x,1)<=690;
        grelha(x,1)=1;
    elseif dados(x,1)<=2960 && dados(x,1)>690
        grelha(x,1)=2;
    elseif dados(x,1)<=9800 && dados(x,1)>2960
        grelha(x,1)=3;
    elseif dados(x,1)>9800
        grelha(x,1)=4;
    end
end
% Diagramas de cada quadrícula da grelha
diags=importdata('p1.mat'); % importa o diagrama de consumo de todos os clientes nº da
linha corresponde ao nº de cliente

C11=zeros(1,24);
n11=0;
C12=zeros(1,24);
n12=0;
C13=zeros(1,24);
n13=0;
C14=zeros(1,24);
n14=0;
C21=zeros(1,24);
n21=0;
C22=zeros(1,24);
n22=0;
C23=zeros(1,24);
n23=0;
C24=zeros(1,24);
n24=0;
C31=zeros(1,24);
n31=0;
C32=zeros(1,24);
n32=0;
C33=zeros(1,24);
n33=0;

```

```

C34=zeros(1,24);
n34=0;
C41=zeros(1,24);
n41=0;
C42=zeros(1,24);
n42=0;
C43=zeros(1,24);
n43=0;
C44=zeros(1,24);
n44=0;

```

% Faz a soma dos diagramas dos clientes em cada quadrícula da grelha e contabiliza o seu número.

```

for r=1:m;
    if grelha(r,1)==1&&grelha(r,2)==1;
        ca1=diags(r,:);
        C11=C11+ca11;           % vector soma dos diagramas de cada agrupamento
        n11=n11+1;           % numero de clentes em cada agrupamento
        l11(1,n11)=r;       % lista de clientes nesta quadricula
    elseif grelha(r,1)==1&&grelha(r,2)==2;
        ca12=diags(r,:);
        C12=C12+ca12;
        n12=n12+1;
        l12(1,n12)=r;
    elseif grelha(r,1)==1&&grelha(r,2)==3;
        ca13=diags(r,:);
        C13=C13+ca13;
        n13=n13+1;
        l13(1,n13)=r;
    elseif grelha(r,1)==1&&grelha(r,2)==4;
        ca14=diags(r,:);
        C14=C14+ca14;
        n14=n14+1;
        l14(1,n14)=r;
    elseif grelha(r,1)==2&&grelha(r,2)==1;
        ca21=diags(r,:);
        C21=C21+ca21;
        n21=n21+1;
        l21(1,n21)=r;
    elseif grelha(r,1)==2&&grelha(r,2)==2;
        ca22=diags(r,:);
        C22=C22+ca22;
        n22=n22+1;
        l22(1,n22)=r;
    elseif grelha(r,1)==2&&grelha(r,2)==3;
        ca23=diags(r,:);

```

```

C23=C23+ca23;
n23=n23+1;
l23(1,n23)=r;
elseif grelha(r,1)==2&&grelha(r,2)==4;
ca24=diags(r,:);
C24=C24+ca24;
n24=n24+1;
l24(1,n24)=r;
elseif grelha(r,1)==3&&grelha(r,2)==1;
ca31=diags(r,:);
C31=C31+ca31;
n31=n31+1;
l31(1,n31)=r;
elseif grelha(r,1)==3&&grelha(r,2)==2;
ca32=diags(r,:);
C32=C32+ca32;
n32=n32+1;
l32(1,n32)=r;
elseif grelha(r,1)==3&&grelha(r,2)==3;
ca33=diags(r,:);
C33=C33+ca33;
n33=n33+1;
l33(1,n33)=r;
elseif grelha(r,1)==3&&grelha(r,2)==4;
ca34=diags(r,:);
C34=C34+ca34;
n34=n34+1;
l34(1,n34)=r;
elseif grelha(r,1)==4&&grelha(r,2)==1;
ca41=diags(r,:);
C41=C41+ca41;
n41=n41+1;
l41(1,n41)=r;
elseif grelha(r,1)==4&&grelha(r,2)==2;
ca42=diags(r,:);
C42=C42+ca42;
n42=n42+1;
l42(1,n42)=r;
elseif grelha(r,1)==4&&grelha(r,2)==3;
ca43=diags(r,:);
C43=C43+ca43;
n43=n43+1;
l43(1,n43)=r;
elseif grelha(r,1)==4&&grelha(r,2)==4;
ca44=diags(r,:);
C44=C44+ca44;

```

```

        n44=n44+1;
        l44(1,n44)=r;
    end
end
% Diagrama médio de cada quadrícula da grelha
C11=C11/n11;
C12=C12/n12;
C13=C13/n13;
C14=C14/n14;
C21=C21/n21;
C22=C22/n22;
C23=C23/n23;
C24=C24/n24;
C31=C31/n31;
C32=C32/n32;
C33=C33/n33;
C34=C34/n34;
C41=C41/n41;
C42=C42/n42;
C43=C43/n43;
C44=C44/n44;
% Passa os diagramas médios para uma matriz
DIAG=zeros(16,24);
DIAG(1,:)=C11;
DIAG(2,:)=C12;
DIAG(3,:)=C13;
DIAG(4,:)=C14;
DIAG(5,:)=C21;
DIAG(6,:)=C22;
DIAG(7,:)=C23;
DIAG(8,:)=C24;
DIAG(9,:)=C31;
DIAG(10,:)=C32;
DIAG(11,:)=C33;
DIAG(12,:)=C34;
DIAG(13,:)=C41;
DIAG(14,:)=C42;
DIAG(15,:)=C43;
DIAG(16,:)=C44;

% Carrega um 'struct' tipo e adapta-o com o valor do diagrama médio de cada
% Quadrícula da grelha
sD=importdata('sD.mat');
sD.data=DIAG;

% Limpa variáveis (liberta memória)

```

```
clear m n i j r x C11 C12 C13 C14 C21 C22 C23 C24 C31 C32 C33 C34 C41 C42 C43 C44
clear ca11 ca12 ca13 ca14 ca21 ca22 ca23 ca24 ca31 ca32 ca33 ca34 ca41 ca42 ca43 ca44
clear dados diags grelha
```

Código fonte da função M_p2_n:

```
% Criar Grelha com os dados normalizados
dados=importdata('p2.mat'); % importa lista de consumo e potência contratada
[m, n] = size(dados);
grelha=zeros(m,n);
% Separa a Potência contratada Kva
for x=1:m;
    if dados(x,2)==1.15 || dados(x,2)==2.3 || dados(x,2)==3.45
        grelha(x,2)=1;
    elseif dados(x,2)==4.6 || dados(x,2)==5.75 || dados(x,2)==6.9
        grelha(x,2)=2;
    elseif dados(x,2)==10.35 || dados(x,2)==13.8 || dados(x,2)==17.25
        grelha(x,2)=3;
    elseif dados(x,2)==27.6 || dados(x,2)==34.5 || dados(x,2)==41.4 ||
dados(x,2)==20.7
        grelha(x,2)=4;
    end
end
% Separa o Consumo kWh
for x=1:m;
    if dados(x,1)<=690;
        grelha(x,1)=1;
    elseif dados(x,1)<=2960 && dados(x,1)>690
        grelha(x,1)=2;
    elseif dados(x,1)<=9800 && dados(x,1)>2960
        grelha(x,1)=3;
    elseif dados(x,1)>9800
        grelha(x,1)=4;
    end
end
% Diagramas de cada quadrícula da grelha
diags=importdata('p1.mat'); % importa o diagrama de consumo de todos o clientes
ordenada de forma que o numero da linha corresponde ao nº de cliente

% Limpa variáveis
clear x

% Normalização
M=max(diags, [], 2);
[i, j]=size(diags);
```

```

for x=1:i;
    for y=1:j;
        diags(x,y)=diags(x,y)/M(x);
    end;
end;

```

```

C11=zeros(1,24); % vector soma dos diagramas de cada agrupamento
n11=0;          % numero de clientes em cada agrupamento
C12=zeros(1,24);
n12=0;
C13=zeros(1,24);
n13=0;
C14=zeros(1,24);
n14=0;
C21=zeros(1,24);
n21=0;
C22=zeros(1,24);
n22=0;
C23=zeros(1,24);
n23=0;
C24=zeros(1,24);
n24=0;
C31=zeros(1,24);
n31=0;
C32=zeros(1,24);
n32=0;
C33=zeros(1,24);
n33=0;
C34=zeros(1,24);
n34=0;
C41=zeros(1,24);
n41=0;
C42=zeros(1,24);
n42=0;
C43=zeros(1,24);
n43=0;
C44=zeros(1,24);
n44=0;

```

% Faz a soma dos diagramas dos clientes em cada quadricula da grelha e contabiliza o seu numero.

```

for r=1:m;
    if grelha(r,1)==1&&grelha(r,2)==1;
        ca11=diags(r,:);
        C11=C11+ca11;
        n11=n11+1;
    end;
end;

```

```

    l11(1,n11)=r;           % lista de clientes nesta quadricula
elseif grelha(r,1)==1&&grelha(r,2)==2;
    ca12=diags(r,:);
    C12=C12+ca12;
    n12=n12+1;
    l12(1,n12)=r;
elseif grelha(r,1)==1&&grelha(r,2)==3;
    ca13=diags(r,:);
    C13=C13+ca13;
    n13=n13+1;
    l13(1,n13)=r;
elseif grelha(r,1)==1&&grelha(r,2)==4;
    ca14=diags(r,:);
    C14=C14+ca14;
    n14=n14+1;
    l14(1,n14)=r;
elseif grelha(r,1)==2&&grelha(r,2)==1;
    ca21=diags(r,:);
    C21=C21+ca21;
    n21=n21+1;
    l21(1,n21)=r;
elseif grelha(r,1)==2&&grelha(r,2)==2;
    ca22=diags(r,:);
    C22=C22+ca22;
    n22=n22+1;
    l22(1,n22)=r;
elseif grelha(r,1)==2&&grelha(r,2)==3;
    ca23=diags(r,:);
    C23=C23+ca23;
    n23=n23+1;
    l23(1,n23)=r;
elseif grelha(r,1)==2&&grelha(r,2)==4;
    ca24=diags(r,:);
    C24=C24+ca24;
    n24=n24+1;
    l24(1,n24)=r;
elseif grelha(r,1)==3&&grelha(r,2)==1;
    ca31=diags(r,:);
    C31=C31+ca31;
    n31=n31+1;
    l31(1,n31)=r;
elseif grelha(r,1)==3&&grelha(r,2)==2;
    ca32=diags(r,:);
    C32=C32+ca32;
    n32=n32+1;
    l32(1,n32)=r;

```



```

elseif grelha(r,1)==3&&grelha(r,2)==3;
    ca33=diags(r,:);
    C33=C33+ca33;
    n33=n33+1;
    l33(1,n33)=r;
elseif grelha(r,1)==3&&grelha(r,2)==4;
    ca34=diags(r,:);
    C34=C34+ca34;
    n34=n34+1;
    l34(1,n34)=r;
elseif grelha(r,1)==4&&grelha(r,2)==1;
    ca41=diags(r,:);
    C41=C41+ca41;
    n41=n41+1;
    l41(1,n41)=r;
elseif grelha(r,1)==4&&grelha(r,2)==2;
    ca42=diags(r,:);
    C42=C42+ca42;
    n42=n42+1;
    l42(1,n42)=r;
elseif grelha(r,1)==4&&grelha(r,2)==3;
    ca43=diags(r,:);
    C43=C43+ca43;
    n43=n43+1;
    l43(1,n43)=r;
elseif grelha(r,1)==4&&grelha(r,2)==4;
    ca44=diags(r,:);
    C44=C44+ca44;
    n44=n44+1;
    l44(1,n44)=r;
end
end
% Diagrama médio de cada quadrícula da grelha
C11=C11/n11;
C12=C12/n12;
C13=C13/n13;
C14=C14/n14;
C21=C21/n21;
C22=C22/n22;
C23=C23/n23;
C24=C24/n24;
C31=C31/n31;
C32=C32/n32;
C33=C33/n33;
C34=C34/n34;
C41=C41/n41;

```

```

C42=C42/n42;
C43=C43/n43;
C44=C44/n44;
% Passa os diagramas médios para uma matriz
DIAG=zeros(16,24);
DIAG(1,:)=C11;
DIAG(2,:)=C12;
DIAG(3,:)=C13;
DIAG(4,:)=C14;
DIAG(5,:)=C21;
DIAG(6,:)=C22;
DIAG(7,:)=C23;
DIAG(8,:)=C24;
DIAG(9,:)=C31;
DIAG(10,:)=C32;
DIAG(11,:)=C33;
DIAG(12,:)=C34;
DIAG(13,:)=C41;
DIAG(14,:)=C42;
DIAG(15,:)=C43;
DIAG(16,:)=C44;

% Carrega um 'struct' tipo e adapta-o com o valor do diagrama médio de cada
% quadricula da grelha
sD=importdata('sD.mat');
sD.data=DIAG;

% Limpa variáveis
clear m n i j r x y

```

Código fonte da função quad:

```

% extrai a posição da amostra no mapa

N=zeros(4,4);

N(1,1)=n41;
N(1,2)=n42;
N(1,3)=n43;
N(1,4)=n44;
N(2,1)=n31;
N(2,2)=n32;
N(2,3)=n33;
N(2,4)=n34;
N(3,1)=n21;
N(3,2)=n22;

```

```

N(3,3)=n23;
N(3,4)=n24;
N(4,1)=n11;
N(4,2)=n12;
N(4,3)=n13;
N(4,4)=n14;

N_h=N(:);

clear z;
close all;
clear f1 f2 f3 f4 f5 f6 f7 f8 f9 f10;
[V,l]=som_divide(sM, sD);
[a b]=size(l);
z=zeros(5, 5);
for ff=2:6;
for i=1:a;
[c d]=size(l{i,1});
if isempty(l{i,1})==0
for f=1:c;
if l{i,1}(f,1)==1
z(1,1)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==2
z(1,2)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==3
z(1,3)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==4
z(1,4)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==5
z(2,1)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==6
z(2,2)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==7
z(2,3)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==8
z(2,4)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==9
z(3,1)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==10
z(3,2)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==11
z(3,3)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==12
z(3,4)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==13
z(4,1)=p{ff,1}(i,1);

```

```

elseif l{i,1}(f,1)==14
    z(4,2)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==15
    z(4,3)=p{ff,1}(i,1);
elseif l{i,1}(f,1)==16
    z(4,4)=p{ff,1}(i,1);
end
end
else
    % ignorar pq está vazio
end
end
figure;
subplot(1,2,1)
x = [1:5];
y = [1:5];
[X,Y] = meshgrid(x,y);
pcolor(X,Y,z);
set(gca,'XTick',[1 2 3 4 5],'YTick',[1 2 3 4 5],'XTicklabel',{'1';'2'; '3'; '4'},'YTicklabel',{'1';'2'; '3'; '4'});
xlabel('Níveis de potência contratada')
ylabel('Níveis de consumo')
subplot(1,2,2)
som_cplane('rect',[4 4],'none')
hold on
som_grid('rect',[4 4],'Label',cellstr(int2str(N_h)),...
'Line','none','Marker','none','Labelcolor','k');
hold off
title('Numero de amostras por nivel de consumo/potência contratada')
set(gcf, 'Position', get(0,'Screensize')); % Maximize figure.
end

```