Universidade Federal de Pernambuco
Centro de Informática

Bacharelado em Engenharia da Computação

Uma abordagem multivariada para redes de agrupamento Fuzzy Kohonen

Rodrigo Bruno de Carvalho Cavalcanti

Trabalho de Graduação

Recife
Abril de 2018

Universidade Federal de Pernambuco
Centro de Informática

Rodrigo Bruno de Carvalho Cavalcanti

Uma abordagem multivariada para redes de agrupamento Fuzzy Kohonen

Trabalho apresentado ao Programa de Bacharelado em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

Orientadora: *Profa. Dra. Renata Maria Cardoso Rodrigues de Souza*

Recife
Abril de 2018

Dedico este trabalho à minha mãe Iva, ao meu pai Graciliano, e especialmente à minha esposa Raphaelle que me apoiaram em todos os momentos difíceis dessa caminhada.

Agradecimentos

Gostaria de agradecer primeiramente a Deus, por ter me guiado e por nunca ter me abandonado nos momentos que pensei em desistir. Agradeço à Ele todas as vezes que sua presença mantinha minha fé e me encorajava a manter o foco e superar os obstáculos que me impediam de seguir. Aos meus pais, Iva e Graciliano, agradeço pela educação e valores os quais foram base para minhas escolhas e também aos conselhos que recebi, sem os quais não teria chegado até aqui. Agradeço também por todo amor recebido e sem o qual não seria quem sou. À minha esposa, Raphaelle, que me acompanhou durante todos esses anos de graduação e que presenciou todas as dificuldades que passei dando apoio e amor incondicional. À professora Dr. Renata Maria Cardoso Rodrigues de Souza pelo apoio, paciência e mentoria durante do Trabalho de Graduação. Aos doutores Dr. Bruno Pimentel e Dr. Carlos Wilson pelo apoio, orientações e material cedido que foram essenciais para a conclusão deste trabalho. Obrigado Doutores! Agradeço também a todos os meus professores, tanto da universidade, como do colégio, os quais me guiaram durante minha trajetória até a graduação. Obrigado, professores! Agradeço, finalmente, a todos os meus amigos que ao longo do curso compartilharam dúvidas, projetos, alegrias e tristezas mas sempre acreditando que o melhor está por vir.

Não é preciso entrar para a história para fazer um mundo melhor.

—MAHATMA GANDHI

Resumo

Em um cenário onde as tarefas cotidianas estão se tornando automatizadas e a qualidade de vida remete à máquina como coadjuvante, os algoritmos de aprendizagem de máquina tem papel decisivo. Além de automatizar funções, as máquinas precisam de rapidez e corretude para realizar as funções de forma autônoma e ubíqua. Este trabalho visa apresentar uma abordagem multivariada para o método Fuzzy Kohonen Clustering Networks (FKCN) com o propósito de otimizar a classificação não supervisionada, analisando separadamente a influência de cada variável. O FKCN é um algoritmo de agrupamento em lotes que combina as ideias de grau de pertinência difuso para taxas de aprendizado, o paralelismo do *fuzzy c-means* e a estrutura e regras auto-organizáveis de atualização da rede de agrupamento Kohonen. O método proposto incorpora a abordagem multivariada do Multivariate Fuzzy C-Means (MFCM) que minimiza uma função objetivo na qual a contribuição de cada variável na entropia é calculada de forma independente. Para avaliar o desempenho do método 3 conjuntos de dados sintéticos são gerados e através do experimento Monte Carlo, o índice difuso de Rand e a taxa global de erro são calculadas. O método também é avaliado para 4 conjuntos de dados reais extraídos da base da UCI. Os resultados mostraram que o método proposto obteve um desempenho superior ao método original na grande maioria dos casos, levando em conta o índice difuso de Rand. Tais resultados mostram que a aplicação da abordagem multivariada em classificadores não-supervisionados aumenta a percepção do modelo em conjuntos onde as características não apresentam nível igual de relevância.

Palavras-chave: Agrupamento, Fuzzy, C-means, Kohonen, multivariado, FCM, FKCN

Abstract

In a scenario where everyday tasks are becoming automated and the quality of life refers to the machine as a coadjuvant, machine learning algorithms play a decisive role. In addition to automating functions, machines need speed and accuracy to perform functions autonomously and ubiquitously. This work aims to present a multivariate approach to the Fuzzy Kohonen Clustering Networks (FKCN) method with the purpose of optimizing the unsupervised classification, analyzing separately the influence of each variable. The FKCN is a batch-grouping algorithm that combines the ideas of diffuse membership values for learning rates, the parallelism of fuzzy c-means, and the self-organizing structure and rules of Kohonen clustering. The proposed method incorporates the Multivariate Fuzzy C-Means (MFCM) multivariate approach that minimizes an objective function in which the contribution of each variable to the entropy is calculated independently. To evaluate the performance of the method 3 synthetic datasets are generated and through the Monte Carlo experiment, the Rand diffuse index and the overall error rate tikare calculated. The method is also evaluated for 4 real data sets extracted from the UCI database. The results showed that the proposed method performed better than the original method in the majority of cases, taking into account the diffuse Rand index. These results show that the application of the multivariate approach in unsupervised classifiers increases the perception of the model in data sets where the characteristics do not present equal level of relevance.

Keywords: Clustering, Fuzzy, C-means, Kohonen, multivariate, FCM, FKCN

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

Em um cenário onde as tarefas cotidianas estão se tornando automatizadas e a qualidade de vida remete à máquina como coadjuvante, os algoritmos de aprendizagem de máquina tem papel decisivo. Além de automatizar funções, as máquinas precisam de rapidez e corretude para realizar as funções de forma autônoma e ubíqua.

A Aprendizagem de Máquina tem como principal objetivo estudar e propor métodos capazes de aprender a partir de dados [2], sendo esta um ramo importante da Inteligência Artificial. Os algoritmos de aprendizagem, em relação aos dados usados como entrada na etapa de treinamento, se dividem em: supervisionados e não-supervisionados [3, 4]. Nos algoritmos supervisionados, a etapa de treinamento é feita usando dados rotulados, enquanto no outro os dados usados não são rotulados. Para grande parte dos problemas, não é possível obter os rótulos ou o custo computacional é alto [5], portanto, é mais adequado utilizar abordagens de agrupamento não-supervisionadas para resolver esses problemas.

O principal objetivo dos algoritmos de agrupamento é separar objetos em conjuntos em que os objetos de cada grupo são mais semelhantes entre si do que em outros grupos. Dado a forma como os algoritmos encontram os grupos, eles podem ser classificados em: hierárquicos e de particionamento [6, 7]. No primeiro, os algoritmos produzem várias partições encadeadas com base em um critério aglomerativo ou divisivo. O aglomerativo começa com vários grupos individuais e junta-os com base na semelhança. Já o divisivo, começa com um único grupo e divide-o até que todos objetos estejam separados. Em contrapartida, o algoritmo de particionamento obtêm uma única partição de dados. A escolha do número de grupos desejados pode ser um problema, pois nem sempre existe informação a priori suficiente para definir esse número [8]. Existem vários trabalhos sobre como definir automaticamente o número de partições [9]. Entretanto, os métodos de particionamento tem superioridade em muitos conjuntos de dados, uma vez que a complexidade de tempo é menor do que a dos métodos hierárquicos [10].

Métodos de particionamento podem ser classificados em duas principais categorias: hard e soft [7]. No particionamento hard, os objetos pertencem totalmente a um grupo ou não pertencem-no, dentro do conjunto $\{0,1\}$. Um dos métodos mais populares é o K-Means [11]. Na segunda categoria, os objetos tem grau de pertinência para todos os grupos, no intervalo

[0,1]. Métodos soft podem ser difusos, probabilísticos ou possibilísticos [12]. A principal vantagem dos métodos soft é sua capacidade de encontrar grupos sobrepostos, isto é, onde existem objetos de diferentes grupos com alta similaridade.

O método de agrupamento Fuzzy C-Means (FCM) [13] é amplamente conhecido e poderoso na abordagem difusa [14]. A sua principal desvantagem é o seu baixo desempenho em conjuntos com presença de dados ruidosos. O método Fuzzy Kohonen Clustering Networks (FKCN) [15] integra o modelo FCM ao uso de taxa de aprendizado e estratégias de atualização das redes auto organizáveis de Kohonen (SOM) [16].

As redes auto organizáveis de Kohonen são também conhecidas como Self-Organizing Maps (SOM) [16]. O algoritmo SOM é uma rede neural pertencente à categoria de redes de aprendizado competitivo. A rede consiste em duas camadas, uma camada de entrada e uma de saída. Dado um conjunto de entrada, os nós da camada de saída competem entre si e aquele conjunto que possuir a menor distância aos dados de entrada atualiza seus pesos.

A rede de Kohonen é baseada em aprendizado não supervisionado, não necessitando de intervenção humana durante o processo de aprendizagem, além disso pouco precisa ser conhecido à respeito dos dados de entrada. Apesar disso, a rede sofre de alguns problemas [16]: (1) o critério de parada não é baseado em nenhum modelo de otimização, (2), os vetores de pesos finais normalmente dependem da sequência de entrada, (3), diferentes condições iniciais geralmente produzem resultados diferentes, (4), para que seja obtido resultados desejáveis, vários parâmetros do algoritmo devem ser variados durante a aprendizagem de um conjunto de dados para outro. Entre os parâmetros que podem ser alterados está a taxa de aprendizagem, a distância usada no cálculo da vizinhança e a estratégia para alterar esses parâmetros.

Com o propósito de resolver esses problemas, o método FKCN foi criado, integrando os métodos do Fuzzy C-Means e as redes de Kohonen. No método apresentado por Kohonen [16], a taxa de aprendizado é ajustada automaticamente e executa treinamento em lote. O critério de parada das SOM é alcançado quando o limite de iterações é atingido ou quando uma estratégia artificial força a taxa de aprendizagem seja nula. Já a parada do FKCN ocorre quando o erro associado a otimização da função objetivo atinge um limiar definido a priori.

O algoritmo FKCN transforma o processo de aprendizagem heurística das SOM em um processo de otimização de uma função objetivo, com o propósito de viabilizar a independência da sequência de entrada de amostras e melhorar a velocidade de convergência [13, 15].

1.1 Objetivos

Este trabalho visa apresentar uma abordagem multivariada para o método Fuzzy Kohonen Clustering Networks (FKCN) com o propósito de otimizar a classificação não supervisionada, analisando separadamente a influência de cada variável.

Será apresentado um método de agrupamento difuso onde os valores do grau de pertinência de cada objeto para um dado grupo é dado por um vetor p-dimensional, onde p é o número de variáveis. Ou seja, será possível calcular quão similar um objeto é de um protótipo de acordo com cada variável separadamente. O principal objetivo é avaliar a melhoria proporcionada pelo cálculo dos valores do grau de pertinência por cada variável separadamente em uma rede difusa de agrupamento de Kohonen.

O método proposto toma como base o que foi apresentado nos métodos multivariado FCM [17], o qual adicionou a abordagem de múltiplos graus de pertinências e do método Fuzzy Kohonen Clustering Networks (FKCN) para dados intervalares [18].

Com o objetivo de avaliar o desempenho do método proposto e os presentes na literatura, um estudo comparativo destes métodos em relação ao agrupamento difuso usando o experimento Monte Carlo será realizado. Serão planejados experimentos com dados sintéticos e reais e o índice difuso de Rand (FR) [19] e a taxa de erro global de classificação (OERC) serão usados para avaliar os métodos.

1.2 Estrutura do trabalho

Este trabalho é dividido seguindo a descrição seguinte: No Capítulo 2 será apresentado um referencial teórico acerca dos conceitos abordados ao longo deste trabalho, apresentando suas características e o estado atual da arte; o Capítulo 3 apresenta uma descrição da metodologia utilizada no desenvolvimento deste trabalho; o Capítulo 4 apresenta os experimentos realizados e os resultados obtidos, bem como as análises obtidas através destes. Por fim, o Capítulo 5 apresenta as conclusões extraídas a partir dos resultados encontrados e possíveis melhorias e desafios futuros.

CAPÍTULO 2

Revisão bibliográfica

Este Capítulo tem como objetivo apresentar, de forma geral, conceitos da área de Clustering, de modo a facilitar a compreensão do trabalho elaborado.

A capacidade de classificação do ser humano parece uma tarefa fácil, mas é adquirida ao longo do tempo com o reconhecimento de novos padrões. Esse reconhecimento é facilitado quando aliado a percepção aguçada do ambiente devido aos sentidos. A facilidade com que o ser humano consegue classificar um objeto nunca antes visto como uma cadeira não é simples para a máquina dadas as inúmeras variáveis que participam do processo de classificação pelo ser humano. A classificação realizada pelo ser humano pode ser considerada supervisionada, pois usa como entrada dados anteriormente classificados, as outras cadeiras conhecidas.

Existem outros métodos de classificação que não dispõe de informações a respeito da classe dos dados. Estes métodos são classificados como não-supervisionados e suas partições são chamadas de agrupamentos. Métodos não-supervisionados são mais adequados em situações onde é impossível ou demasiadamente custoso de se obter os rótulos dos dados. Esse tipo de aprendizado não supervisionado pode ser dividido em duas grandes categorias: Métodos Hierárquicos e Métodos de Particionamento. Nas sessões que se seguem serão descritos em detalhes as características de cada um desses conjuntos de métodos, com maior grau de importância aqueles mais usados ou que tenham relevância para este trabalho.

2.1 Métodos Hierárquicos

Os métodos hierárquicos são técnicas simples onde os dados são particionados sucessivamente, produzindo uma representação hierárquica dos agrupamentos [20]. Essa representação facilita a visualização a respeito da construção dos grupos em cada fase onde a partição ocorreu e com que grau de semelhança entre os grupos a partição foi feita.

Os métodos hierárquicos não necessitam que seja escolhido um número de grupos a *priori*. Através da análise do diagrama que mostra a hierarquia e a relação dos agrupamentos, dendrograma, é possível inferir o número de agrupamentos adequado. O dendrograma é uma árvore

construída usando o nível de similaridade entre os elementos. A Figura 2.1 apresenta um dendrograma onde é possível identificar o nível de similaridade entre 2 elementos através da altura da linha horizontal que os liga. Note que quanto mais baixo o nível da linha horizontal maior o grau de similaridade entre os elementos.

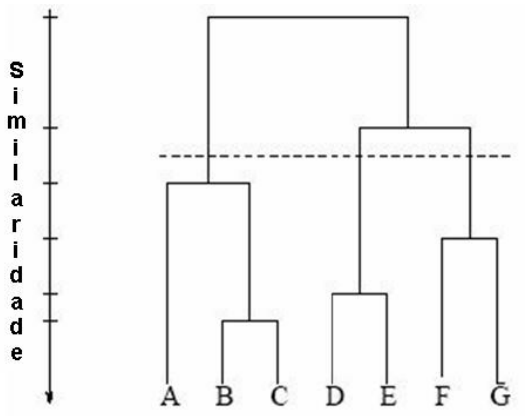

Figura 2.1 Exemplo de Dendrograma
Fonte:
https://www.researchgate.net/figure/237699601_fig10_FIG-33-Exemplo-de-um-dendrograma

Os métodos hierárquicos utilizam uma matriz contendo as métricas de distância entre os agrupamentos em cada estágio do algoritmo. Essa matriz é conhecida como matriz de similaridades entre agrupamentos. Existem vários métodos usados para calcular o grau de similaridade entre os grupos. Aqueles considerados mais importantes [21] são mostrados abaixo:

- O Método *Single Linkage* é baseado na menor distância entre quaisquer par de elementos disjuntos. O algoritmo é conduzido a encontrar poucos grupos cuja dissimilaridade entre os elementos é alta. Portanto, grupos mais alongados são produzidos devido a procura ser sempre por elementos mais similares possíveis. Sua expressão é dada por:

$$d(A,B) = min\{d(a,b)|a \in A, b \in B\} \tag{2.1}$$

- O Método *Complete Linkage*, de forma semelhante ao primeiro, procura pela maior distância entre este par de elementos. Essa medida produz grupos compactos ou bem encadeados e tem a propriedade de deixar elementos relativamente similares em diferentes grupos. Sua expressão é dada por:

$$d(A,B) = max\{d(a,b)|a \in A, b \in B\} \tag{2.2}$$

- No Método *Average Linkage* (Eq. (2.3)) a dissimilaridade é obtida a partir da média de distância entre todos os pares de objetos, onde cada um pertence a um grupo diferente. Esse tipo de ligação procura equilibrar as vantagens e desvantagens das ligações simples e completa produzindo grupos relativamente compactos e também afastados.

$$d(A,B) = \frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a,b) \tag{2.3}$$

- No Método *Median Linkage* (ou Mediana) cada grupo é representado por um vetor de distâncias entre cada elemento pertencente ao grupo e a distância entre grupos é medida entre os objetos médios de cada grupo. Esse método é bastante robusto à presença de *outliers* devido ao uso do valor médio da mediana no grupo. Sua expressão é dada por:

$$d(A,B) = \frac{2|A||B|}{|A|+|B|} d(\mathbf{m}_A, \mathbf{m}_B),\ \mathbf{m}_A, \mathbf{m}_B \text{ são } \textit{medoids} \tag{2.4}$$

- O Método *Centroid Linkage* representa cada grupo por um centroide. O centroide é calculado a partir da média dos objetos no grupo. E, por consequência, a distância entre os grupos é medida entre os centroides de cada grupo usando a expressão (2.5).

$$d(A,B) = d(\mathbf{c}_A, \mathbf{c}_B),\ \mathbf{c}_A, \mathbf{c}_B \text{ são } \textit{centroides} \tag{2.5}$$

- O Método de ligação de *Ward* (2.6) procura por partições que minimizem a perda associada a cada grupo. Essa perda é quantificada pela diferença entre a soma dos erros quadráticos de cada objeto e a média da partição a que pertence.

$$d(A,B) = \frac{2|A||B|}{|A|+|B|} d(\mathbf{c}_A, \mathbf{c}_B),\ \mathbf{c}_A, \mathbf{c}_B \text{ são } \textit{centroides} \tag{2.6}$$

Os métodos hierárquicos são subdivididos em Métodos Aglomerativos e Métodos Divisivos [21]. Na primeira abordagem, os dados são inicialmente distribuídos de modo que cada exemplo represente um grupo e, então, esses grupos são recursivamente agrupados usando alguma medida de similaridade, até que todos os exemplos pertençam a apenas um grupo. Na abordagem divisiva, o processo inicia-se com apenas um agrupamento contendo todos os exemplos e segue dividindo-o recursivamente seguindo alguma métrica até que alcance algum critério de parada, frequentemente o número de grupos desejados [22].

Os métodos de similaridade descritos anteriormente nessa sessão requerem uma forma de calcular a distância. A distância $d(A,B)$ entre dois conjuntos A e B pode ser calculada de

várias maneiras dependendo da aplicação do problema. Ela é fundamental para definir qual conjunto deve ser unido ou separado, na abordagem aglomerativa ou divisiva, respectivamente. De acordo com o tipo de distância usada os grupos irão obter diferentes formatos. Considere que cada elemento $a \in A$ é descrito tal que $a = (a_1, ..., a_i, ..., a_m)^T$. Na tabela a seguir estão algumas das medidas de distâncias mais usadas.

Tabela 2.1 Medidas de distância.

Nome da distância	Expressão
Manhattan-norm	$d(a,b) = \sum_i \left\lvert a_i - b_i \right\rvert$
Euclidean-norm	$d(a,b) = \sum_i (a_i - b_i)^2$
Minkowsky	$d(a,b) = \sqrt[p]{\sum_i \left\lvert a_i - b_i \right\rvert^p}$
Mahalanobis	$d(a,b) = \sqrt{(a-b)^T \Sigma^{-1} (a-b)}$, Σ^{-1} é a matriz de covariância

A Soma da Diferença Absoluta (SAD, em inglês) ou Manhattan-norm é equivalente a norma L_1, também conhecida como Taxicab-norm. A Soma da Diferença Quadrada (SSD, em inglês), também conhecida como Euclidean-norm, corresponde ao espaço L_2 e Minkowski ao espaço L_p. Esta última pode ser considerada uma generalização das duas anteriores. A distância de Mahalanobis é baseada nas correlações entre variáveis com as quais distintos padrões podem ser identificados e analisados.

2.1.1 Algoritmos Aglomerativos

Os métodos aglomerativos são os mais comuns entre os métodos hierárquicos. O algoritmo inicia com cada padrão formando seu próprio grupo e gradualmente os grupos são unidos até que um único grupo contendo todos os dados seja gerado [10]. No início do processo, cada elemento é um grupo e estes possuem um alto grau de similaridade entre si. Ao final do processo, têm-se poucos agrupamentos com vários elementos e os grupos possuem pouca similaridade.

Inicialmente uma matriz de similaridade é criada entre os pares de grupos. Como no início todos os grupos são unitários, a matriz de similaridade é calculada usando a métrica par a par. A cada iteração do algoritmo um novo grupo é formado pela união dos dois grupos mais similares.

A cada iteração a matriz de similaridades é calculada e os grupos são unidos de acordo com a menor dissimilaridade encontrada. Esse processo é repetido até que apenas um grupo exista.

Uma importante propriedade dos algoritmos aglomerativos é a sua monotonicidade, ou seja, a dissimilaridade entre dois grupos que são agrupados é monotonicamente crescente de acordo

com o nível em a junção ocorre. Isso pode ser observado a partir da ordem de união dos grupos, pois estas são feitas a partir dos grupos mais similares entre si. Essa propriedade implica que, para dado um grupo, a dissimilaridade de seus filhos é maior que as dissimilaridades entre os sub-grupos de cada um dos dois filhos. Isso permite que o agrupamento seja representado usando um dendograma, por exemplo.

A representação gráfica por meio do dendograma dos grupos e sub-grupos permite que seja feita uma análise estatística e comparações entre estes grupos. Embora, o agrupamento hierárquico não possa fornecer somente uma, mas sim um conjunto de partições, é possível encontrar uma partição dos dados usando o dendrograma.

As principais desvantagens dos métodos hierárquicos aglomerativos são: os agrupamentos não podem ser corrigidos, isto é, os padrão de um determinado grupo irão permanecer no grupo até o fim da execução do método; modelos como este requerem muito espaço de memória e poder de computacional devido às matrizes de similaridades crescerem rapidamente com o tamanho do conjunto de entrada.

2.1.2 Algoritmos Divisivos

Os métodos divisivos são os menos comuns entre os métodos hierárquicos devido a sua ineficiência, além disso eles exigem uma capacidade computacional maior em relação aos métodos hierárquicos aglomerativos [10]. Uma potencial vantagem dos métodos divisivos sobre os métodos aglomerativos pode ocorrer quando o objetivo é encontrar uma partição dos dados em um relativamente pequeno número de grupos [22].

Esse método começa com um único grupo contendo todos os elementos e gradualmente vai dividindo os agrupamentos até que todos os grupos sejam formados apenas por um elemento. O objetivo é encontrar a partição que maximiza a matriz de similaridades.

O primeiro passo do algoritmo envolve todas as divisões possíveis dos dados em dois grupos, o que o tornaria impraticável para um grande número de elementos, implicando, por tanto, em um número elevado de combinações [20].

A propriedade de encontrar partições usando dendogramas é uma característica diferencial dessa abordagem hierárquica quando comparado com os algoritmos de particionamento ou sequencias, por exemplo. Entretanto, algoritmos hierárquicos tem um custo de espaço e tempo computacional mais elevados. Todavia, quando se conhece o número de agrupamentos no conjunto de dados de entrada, é preferível usar métodos de particionamento. Estes métodos estão descritos na sessão a seguir.

2.2 Métodos de Particionamento

Os métodos não hierárquicos, ou de particionamento, foram desenvolvidos para agrupar elementos em K grupos, onde K é a quantidade de grupos definida previamente. Estes métodos são baseados na minimização de uma função de custo. Cada elemento é agrupado na classe em que essa função de custo é minimizada.

Uma das principais vantagens dos métodos particionais em relação aos métodos hierárquicos é a possibilidade de um elemento poder mudar de grupo com a evolução do algoritmo, como também a possibilidade de se trabalhar com bases de dados maiores. Os métodos particionais são extremamente mais rápidos que os métodos hierárquicos [23]. Essa vantagem se deve ao fato de que os métodos de particionamento não necessitam calcular e armazenar a matriz de similaridades durante o processamento.

Uma das desvantagens dos métodos particionais é o fato do número de agrupamentos ter que ser escolhido *a priori*. Isso pode levar à interpretações erradas sobre a estrutura dos dados cujo o número de agrupamentos não seja o ideal. Outra desvantagem está no fato do algoritmo ser em geral sensível as condições inicias. Isso implica que método gera resultados diferentes a cada rodada [23].

Os métodos de particionamento possuem dois tipos principais: os rígidos (ou *hard*), onde cada elemento pertence à uma única classe e os difusos (ou *fuzzy*), onde cada elemento tem um grau de pertinência a cada classe. Entre os métodos rígidos, dentre os métodos mais conhecidos estão o K-Means e o K-Medoids. Já entre os que utilizam a abordagem difusa o Fuzzy C-Means é aquele que está na lista dos mais usados.

2.2.1 Algoritmos Rígidos

Os algoritmos rígidos consistem na obtenção de uma partição a partir de um determinado conjunto de x elementos em um número pré-estabelecido de c classes, dado que $c \leq x$. Isso implica que cada classe deve possuir ao menos um elemento e cada elemento deve pertencer a uma única classe, portanto, não é admitido uma classe vazia e classes onde não existem elementos em comum. A seguir é tratado, em resumo, as características mais relevantes dos principais algoritmos de particionamento rígidos: K-Means e K-Medoids.

K-Means

O método *k-means* é um dos métodos mais populares das técnicas particionais [23]. Este

método particiona os dados em k agrupamentos mutualmente exclusivos. O *k-means* não cria uma estrutura em árvore para descrever o grupo, diferentemente dos métodos hierárquicos. O *k-means* é mais adequado para uma grande quantidade de dados, dado a maneira com qual faz os particionamentos.

O algoritmo busca, dentro do limite, a partição em que os padrões de cada grupo são mais parecidos entre si e mais distantes de padrões de outros agrupamentos. Este algoritmo é do tipo iterativo, onde a soma das distâncias de cada padrão ao centroide de cada grupo é minimizada, sobre todos os grupos. A cada iteração do algoritmo, dois principais passos são executados: representação e alocação. No primeiro, os padrões são movidos entre os grupos até que a função objetivo não se altere ou altere muito pouco, ou até que o valor máximo de iterações seja atingido. Enquanto no segundo passo, objetos são atribuídos às classes cuja distância entre o objeto e o representante da classe seja mínima dentre todas as distâncias para as demais classes da partição. O resultado é um conjunto dos grupos compactos e separados, na medida do possível.

O algoritmo tem alta sensibilidade à ruídos, dado que um elemento com um valor alto pode distorcer a distribuição dos dados. Além disso, devido ao uso do centroide para comparação entre grupos, o algoritmo tem uma tendência a formar grupos esféricos. Devido a isso, ele não é indicado para descobrir grupos com formas não convexas ou de diferentes tamanhos. Outra característica é que a quantidade de grupos é fixa durante todo o processo.

Na figura 2.2, lado esquerdo, é possível observar um conjunto de dados gerados de maneira aleatória. No lado direito, o resultado da classificação usando o algoritmo K-Means, bem como os centroides finais que identificam cada grupo.

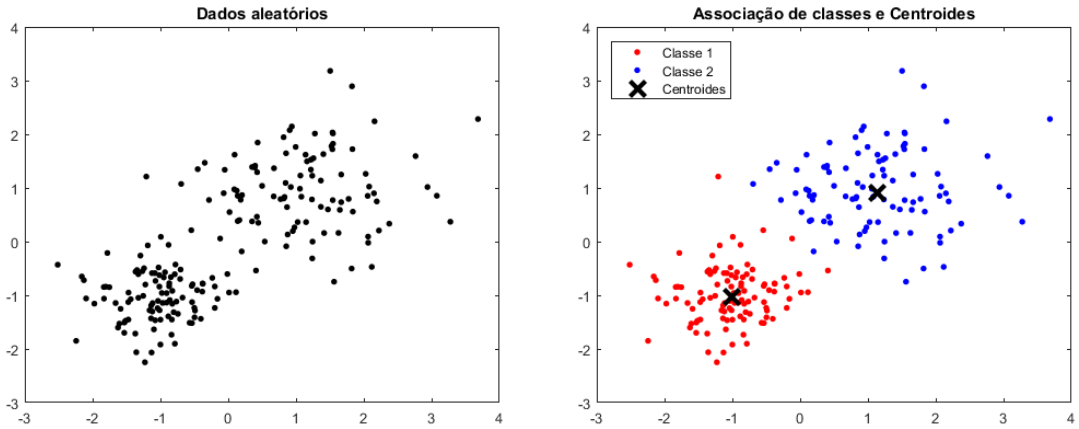

Figura 2.2 Exemplo de uma classificação usando K-Means.

K-Medoids

Outra técnica popular de particionamento, o *k-medoid*, utiliza o valor médio dos elementos de um grupo como um ponto de referência, esse valor médio é chamado de *medoid*. O *medoid* é o elemento localizado mais centralmente em um grupo.

A estratégia básica inicia por definir uma partição com base numa seleção aleatória de objetos, os quais serão os protótipos iniciais. Em seguida, o algoritmo executa duas fases principais: escolha dos protótipos, onde o elemento mais representativo de cada grupo é selecionado, e o passo da atribuição, onde os objetos são trocados e passam a ocupar o grupo tal que a distância para o *medoid* é menor. O processo finaliza quando não houver alteração no *medoid* ou quando a diferença entre o critério de partição anterior e atual seja inferior ao limiar pré-definido.

Dentre as características do método *K-Medoids* estão a independência da ordem do conjunto de entrada, produzindo sempre os mesmos resultados, maior robustes em relação ao *K-Means*, pois o *medoid* é menos influenciado pelo ruído do que as médias. Em contrapartida, o *K-Medoid* é mais custoso computacionalmente, prejudicando seu uso em grandes bases de dados. Além disso, mantém característica similar ao *K-Means* de construção de grupos esféricos devido ao uso dos elementos centrais como protótipos.

2.2.2 Algoritmos Difusos

A segmentação dos dados forma a base de muitos algoritmos de classificação. O seu objetivo é identificar agrupamentos naturais de dados para produzir uma representação concisa do comportamento do sistema. É comum em um processo de classificação que cada objeto pertença a um único agrupamento [20]. Os algoritmos difusos, também chamados de não-exclusivos ou *fuzzy*, permitem que os dados sejam classificados em mais de um grupo ao mesmo tempo. Essa ambiguidade é alcançada através da representação da classificação do elemento usando um grau de pertinência a cada grupo. Essa representação permite uma análise mais rica da distribuição dos dados nos diversos grupos.

Segundo Everitt [20], a principal vantagem dos agrupamentos *fuzzy* em relação aos demais métodos particionais está no fato deste representar, com muito mais detalhes, a informação sobre a estrutura de dados. Entretanto, isso pode ser considerado uma desvantagem, pois a quantidade de informação gerada aumenta muito rapidamente com o número de objetos e o número de grupos. Outra desvantagem é a ausência de objetos representativos dos agrupamentos formados, como o centroide no *k-means*. Além disso, geralmente os algoritmos difusos são mais complicados e consomem mais tempo computacional. Em contra partida, trata-se de uma

técnica válida, pois ela associa graus de incerteza aos elementos nos grupos, se aproximando de maneira geral das características reais dos dados [21].

Um algoritmo difuso bastante utilizado é o *fuzzy c-means*. Na sessão abaixo ele será apresentado de maneira detalhada pois seu entendimento é considerando importante para compreensão deste trabalho.

2.3 Fuzzy C-Means

O método de agrupamento *fuzzy c-means* foi proposto por Bezdek [13]. Este é um algoritmo de agrupamento bastante conhecido da abordagem difusa. O algoritmo compartilha características do *k-means* como a determinação do número de grupos *a priori* e o uso da distância Euclidiana para o cálculo da dissimilaridade entre objetos. Contudo, a similaridade de um objeto com cada grupo é fornecida através de uma função cujos valores estão entre zero e um. Essa característica faz com que a função objetivo do método e o cálculo dos protótipos sejam diferentes do *k-means*. Além disso, o algoritmo possui uma nova etapa de processamento importante, o cálculo do grau de pertinência.

Em resumo, trata-se de um algoritmo iterativo que inicia com c valores arbitrários. Com base nesses valores, associa cada elemento ao valor ao qual possui menor distância, formando c grupos. Em sequência, é calculado o centro de cada grupo e os elementos são reassociados ao ao grupo que pertence o centro mais próximo a ele. Dessa forma, o algoritmo é executado até que as diferenças entre os centros de cada passo sejam mínimas.

2.3.1 Função Objetivo

Dentre as várias abordagens usadas no cálculo da função objetivo, o *fuzzy c-means* usa a abordagem interna. Essa abordagem usa a relação entre objetos do mesmo grupo no cálculo do sua função. No caso do FCM, a função objetivo é baseada na soma de erros quadrados (SSE, em inglês). Dessa forma, o problema resume-se em minimizar a soma das distâncias entre os elementos e seu respectivo representante. Quanto menor for o valor encontrado, mais próximos os elementos estarão dos seus protótipos. O cálculo das distâncias é feito para todos os grupos da partição e é definido pela equação (2.7) abaixo.

$$J = \sum_{i=1}^{c} \sum_{j=1}^{n} (u_{ij})^m d(\mathbf{x}_j, \mathbf{y}_i) \tag{2.7}$$

onde $d(\mathbf{x}_j, \mathbf{y}_i)$ é a distância Euclidiana, encontrada na tabela 2.1.

O cálculo da função objetivo J depende do protótipo y_i, do grau de pertinência u_{ij} e do valor do parâmetro difuso m. A cada iteração, os dois primeiros são recalculados buscando minimizar a função objetivo até que se atinja um mínimo local.

2.3.2 Protótipo

O *fuzzy c-means* faz parte de uma categoria de métodos que se baseiam em protótipos [22]. Os protótipos são importantes para descrever os grupos de uma forma concisa, permitindo uma interpretação mais detalhada da partição. Estes também influenciam no cálculo do grau de pertinência dos objetos.

Os protótipos são calculados através de informações a respeito dos grupos, como a dispersão ou posição dos elementos aos quais pertencem. Desta forma, para cada forma de avaliar as estatística da partição existirá uma categoria de representante de grupo.

No método *k-means*, os protótipos são elementos centrais dos grupos definidos através do cálculo de médias; no *k-medoids*, medianas são utilizadas; e no FCM, o parâmetro difuso m e os graus de pertinência são importantes para determinar os representantes. A expressão (2.8) mostra como os protótipos na abordagem difusa são calculados.

$$\mathbf{y}_i = \frac{\sum_{j=1}^{n} (u_{ij})^m \mathbf{x}_j}{\sum_{j=1}^{n} (u_{ij})^m} \tag{2.8}$$

2.3.3 Grau de Pertinência

Segundo Zadeh [24] a teoria de Conjuntos Difusos (Fuzzy Set, em inglês) fornece uma rigorosa base matemática na qual fenômenos conceituais vagos podem ser estudados rigorosamente. Baseado nessa teoria o grau de pertinência é associado a uma função de pertinência.

O conceito de grau de pertinência pode ser entendido como a probabilidade de um elemento x_j pertencer a um grupo C_i. Esse valor é expresso por u_{ij} e geralmente são armazenados em uma matriz difusa de forma $U = [u_{ij}]_{c \times n}$, c é o número de grupos e n o número de elementos a serem particionados. O cálculo do grau de pertinência u_{ij} é feito de acordo com a equação (2.9) abaixo.

$$u_{ij} = \left[\sum_{h=1}^{c} \left(\frac{d(\mathbf{x}_j, \mathbf{y}_i)}{d(\mathbf{x}_j, \mathbf{y}_h)} \right)^{(1/m-1)} \right]^{-1} \tag{2.9}$$

A equação acima deve cumprir algumas propriedades para garantir certas características do grau de pertinência. Estas são listadas abaixo.

1. $u_{ij} \in [0,1] \quad \forall i,j$
2. $0 < \sum_{j=1}^{n} u_{ij} < n \quad \forall i$
3. $\sum_{i=1}^{c} u_{ij} = 1 \quad \forall j$

Para que o grau de pertinência seja garantido entre 0 e 1, qualquer que seja o objeto ou o grupo do agrupamento, o primeiro item foi definido. A segunda propriedade assegura a primeira, na qual o somatório será n se todos os n objetos tiverem uma chance de 100% de pertencerem a um mesmo grupo, e consequentemente 0% para outros grupos. O último item confirma que a soma dos graus para todos os grupos fixando um objeto é obrigatoriamente 1, qualquer que seja o objeto.

2.3.4 Algoritmo

O algoritmo é semelhante ao *k-means*, portanto, compartilha a mesma organização em quatro etapas principais: inicialização, representação, alocação e critério de parada. A primeira define os valores dos graus de pertinência para todos os objetos do conjunto e todos os grupos de acordo com as restrições, assim como inicializa alguns parâmetros usados durante a execução do algoritmo. Na segunda etapa, os protótipos são recalculados. Na etapa de alocação a matriz difusa é atualizada. Por fim, no critério de parada, é verificado se o algoritmo deve parar sua execução ou continuar.

Geralmente, existem três opções de verificar o critério de parada: o algoritmo para se a diferença entre o valor calculado da função objetivo na iteração atual e o da iteração anterior é menor que um limiar definido *a priori*; se não existir uma diferença significativa entre a matriz difusa atual e a matriz anterior; ou se o algoritmo atingiu um número de iterações máximo definido. Similarmente ao *k-means*, o *fuzzy c-means* também precisa conhecer *a priori* o número de agrupamentos c que o método irá usar para encontrar as c partições. Os passos do algoritmo

[25] são mostrados a seguir.

Algoritmo 1: Fuzzy C-Means (Ω,c,m)

Entrada: Um conjunto de dados Ω, o número de grupos c e o parâmetro fuzzificador $m \in]1, +\infty[$;

Saída : Um agrupamento difuso P ;

Faça $\varepsilon > 0$. Inicialize aleatoriamente os valores da partição $U = [u_{ij}]$, para $i = 1, ..., c$ e $j = 1, ..., n$ tal que $u_{ij} \in [0, 1]$ e $\sum_{i=1}^{c} u_{ij} = 1$;

$t \leftarrow 0$;

$E_t \leftarrow 0$;

$E_{t+1} \leftarrow \sum_{i=1}^{c} \sum_{j=1}^{n} (u_{ij})^m d(\mathbf{x}_j, \mathbf{y}_i)$;

while $\left|E_{t+1} - E_t\right| > \varepsilon$ **do**

  Atualize os representantes dos grupos usando a equação (2.8), $i = 1, ..., c$;

  Atualize a matriz dos graus de pertinência U usando a equação (2.9) ;

  $E_t \leftarrow E_{t+1}$;

  $E_{t+1} \leftarrow \sum_{i=1}^{c} \sum_{j=1}^{n} (u_{ij})^m d(\mathbf{x}_j, \mathbf{y}_i)$;

  $t \leftarrow t + 1$;

end

return Agrupamento Difuso P.

2.4 Redes Neurais Artificiais

Uma rede neural artificial tem duas características elementares: a arquitetura e o algoritmo de aprendizagem. Essa divisão se deve ao fato de como a rede é treinada. São usados exemplos de treino para possibilitar o armazenamento do conhecimento. O algoritmo de aprendizagem generaliza esses dados e memoriza o conhecimento dentro dos pesos, os parâmetros adaptáveis da rede. Sendo assim, a construção de um sistema baseado em redes neurais tem duas atividades a serem observadas, a definição à respeito do tipo de rede para resolver o problema considerado e o algoritmo de treinamento da rede [26].

As redes neurais artificiais são compostas por neurônios. Esses neurônios artificiais são modelos que buscam simular as realidades biológicas que ocorrem dentro de uma célula do sistema nervoso. A informação fornecida por neurônios adjacentes entra em S entradas x_k, ou sinapses, no neurônio processador. O processamento consiste de uma combinação linear das entradas, dada pela equação (2.10) abaixo.

$$net = \sum_{k=1}^{S} w_k x_k \tag{2.10}$$

Cada entrada está conectada a um peso w_k que reflete a importância da entrada x_k. O resultado dessa combinação linear é o valor *net*. Quando esse valor ultrapassa um limiar μ, o neurônio ativa e dispara o valor 1 na saída binária y. Se não ultrapassar o limiar a saída fica desativada, portanto, valor 0. A comparação de *net* com o limiar é realizada pela função de Heaveside (2.11) que tem como saída 1 se o valor na entrada for positivo e saída 0 caso contrário. Na figura 2.3 é possível observar todo funcionamento de um neurônio de McCulloch e Pitts.

$$u(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{2.11}$$

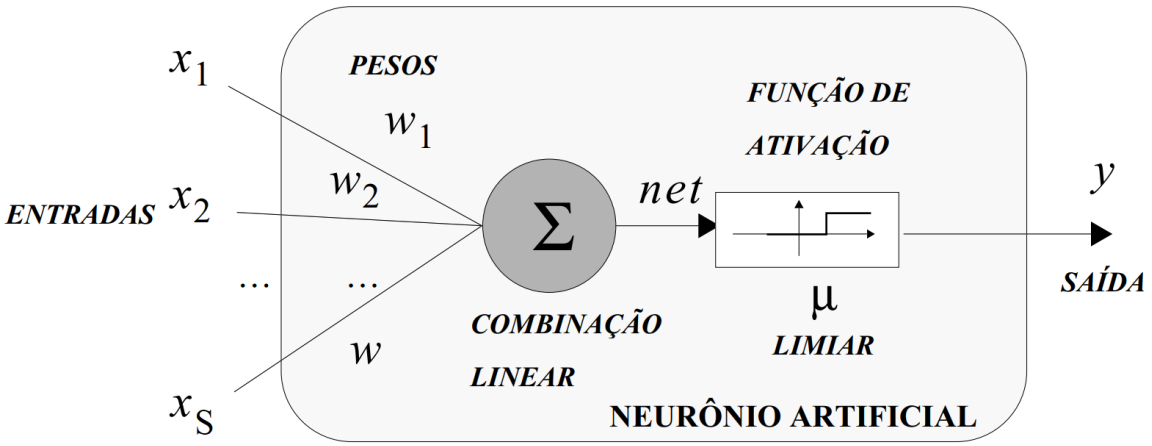

Figura 2.3 Modelo de um neurônio de McCulloch e Pitts. Adaptado de [1].

Existem outras funções que podem ser usadas para produzir o valor de saída do neurônio. Dentre elas estão: a função linear, que produz uma saída contínua; a função escada, que produz uma saída binária (não linear discreta) e a função sigmoidal, cuja saída é não-linear contínua.

A função sigmoidal é bastante usada pois possui um conjunto de propriedades muito úteis nos cálculos relacionados à aprendizagem dos pesos e ao mapeamento realizado pela rede. Em termos do domínio dos valores de saída distingue-se basicamente entre saídas binárias com $y_i \in \{0,1\}$ ou $y_i \in \{-1,1\}$ e saídas contínuas com $y_i \in \mathbb{R}$.

2.4.1 Topologia

O potencial e flexibilidade do método baseado em redes neurais é proporcionado através do conjunto de ligações entre os neurônios. O processamento das redes se baseiam em um elemento da rede receber um estímulo nas duas entradas, processar o sinal e emitir novo sinal de saída. Este sinal de saída é propagado aos demais neurônios da rede.

Dentre as topologias das redes neurais pode-se distinguir entre redes de propagação para frente ou *feedforward* e redes realimentadas ou *recurrent*. No primeira, o fluxo de informação é unidirecional. Neurônios que recebem a informação simultaneamente são agrupados em camadas. Aquelas camadas que não estão ligadas às entradas ou às saídas da rede são chamadas de camadas escondidas. As redes mais conhecidas que seguem esse paradigma são: o perceptron [27], o perceptron multicamada [28] e o *Adaline* [29]. Uma rede que adicionalmente tem uma relação topológica de vizinhança entre os neurônios é o mapa auto-organizável de Kohonen [16]. A rede de Kohonen será discutida em mais detalhes na próxima sessão.

Redes realimentadas tem ligações sem restrições entre os neurônios. Ao contrário das redes não alimentadas, o comportamento dinâmico deste paradigma desempenha papel importante neste modelo. Os valores de ativação da rede podem passar por um processo de ajuste até chegar a um nível estável em alguns casos. Um modelo bastante conhecido desse paradigma é a rede auto associativa de Hopfield [30].

2.4.2 Paradigmas de Aprendizagem

Após definida a estrutura da rede é necessário iniciar o processo de treinamento. Para solucionar o problema em questão, os graus de liberdade cuja rede dispõe precisam ser adaptados de uma maneira ótima. Geralmente, o processo de aprendizagem envolve ajustar os pesos w_{ij} entre o neurônio i e j, de acordo com um algoritmo. Em seguida, a fase de treinamento da rede usa um conjunto finito T de n exemplos de treino para o executar o algoritmo.

Em relação ao paradigma de aprendizagem, todo tipo de sistema com capacidade de adaptação se distingue entre aprendizagem supervisionada e aprendizagem não-supervisionada.

Na *aprendizagem supervisionada* cada exemplo de treino está acompanhado por um valor desejado. De outro modo, cada padrão do conjunto de treino está classificado previamente. Um exemplo de uma tarefa de aprendizagem supervisionada é a regressão linear. A regressão linear é uma equação para se estimar a condicional (valor esperado) de uma variável y, dados os valores de algumas outras variáveis x. Em geral, tem-se como objetivo encontrar um valor que não se consegue estimar inicialmente.

Quando a única informação disponível são os valores do conjunto de entrada, a tarefa de aprendizagem é inferir correlações entre os exemplos. O número de grupos ou classes não está definido *a priori*. Portanto, a rede precisa encontrar atributos estatísticos relevantes. Isso pode ser feito desenvolvendo uma representação própria dos estímulos que entram na rede. Um sinônimo para *aprendizagem não-supervisionada* é aglomeração ou *clustering*. Um exemplo da área de medicina é a detecção de doenças a partir de imagens, em imagens de raio-X, por exemplo. Existem várias regiões similares na imagens que são do mesmo material, como o osso. O desafio é descobrir o número de materiais diferentes e simultaneamente categorizar cada ponto da imagem para o respectivo material.

2.4.3 Regras de Atualização dos Pesos

Durante o processo de aprendizagem os pesos normalmente passam por uma modificação iterativa. A cada iteração o peso influencia a a função calculada pela rede, a qual gera uma qualidade do peso. Essa qualidade determina se o peso deve sofrer uma modificação no seu valor de uma diferença na iteração seguinte.

Normalmente a inicialização dos pesos de dá de maneira aleatória. O algoritmo de aprendizagem executa por um número fixo de iterações, até que uma condição de parada seja atingida ou ambas. Em uma aprendizagem supervisionada, essa condição de parada pode ser calculada através da discrepância entre o valor calculado e o valor desejado. Seja w_{ij} o peso entre o neurônio i e j e k a iteração atual, a regra básica de atualização dos pesos é definida pela equação abaixo (2.12).

$$w_{ij}^{k+1} = w_{ij}^{k} + \Delta w_{ij}^{k} \tag{2.12}$$

Os algoritmos de aprendizagem podem ser divididos em três principais modelos [1, 31]: Regra de Hebb, Regra de Delta e aprendizagem competitiva.

A aprendizagem pela *Regra de Hebb* foi um dos estudos pioneiros sobre sistemas com capacidade de aprendizado. Hebb [31] desenvolveu uma hipótese de que o peso da ligação entre dois neurônios que estão ativos ao mesmo tempo deve ser reforçado. A regra de Hebb pode ser definida pela equação abaixo.

$$\Delta w_{ij}^{k} = \eta y_i y_j \tag{2.13}$$

onde a taxa de aprendizagem η é um fator de escala positivo que determina a velocidade da aprendizagem. A definição acima se baseia em estudos biológicos do cérebro, mas a correspondência do modelo com a realidade biológica é apenas uma idealização aproximada. A regra de

Hebb define um algoritmo de adaptação, porém sem definir um objetivo, como a minimização de uma função objetivo, por exemplo.

Em contrapartida, a *Regra de Delta* desenvolvida por Windrow-Hoff [31] tem uma regra de adaptação dos pesos com um objetivo definido. A rede calcula na saída de cada nerônio uma função $y_i^{'}$. O resultado dessa função é usado na aprendizagem supervisionada para calcular o erro $e_i = y_i - y_i^{'}$ entre o valor desejado e calculado. A regra então define que o peso entre o neurônio i e j que são responsáveis pelo erro deve ser atualizado proporcionalmente ao valor da ativação e ao erro. O valor ainda é escalonado novamente pela taxa de aprendizagem η como definido na equação abaixo.

$$\Delta w_{ij} = \eta e_i y_j = \eta (y_i - y_i^{'}) y_j \tag{2.14}$$

Outro método de aprendizagem é conhecido como *Aprendizagem Competitiva*. Nesse modelo, as redes funcionam de maneira que apenas um único neurônio pode ser ativo ao mesmo tempo. Isso implica que todos os outros neurônios tem um estado de ativação igual a zero e somente o vencedor emite um sinal de ativação.

$$\Delta w_{ij} = \eta y_i (x_j - w_{ij}) \tag{2.15}$$

O efeito dessa regra é que os pesos w_i se deslocam em direção ao estímulo, entrada da rede. Um exemplo bastante usado desse modelo são as Redes de Kohonen, as quais usam a aprendizagem competitiva para adaptar os pesos. Esta rede será detalhada na sessão a seguir.

2.4.4 Redes de Kohonen

Um algoritmo muito popular para agrupamento de dados, os Mapas Auto-Organizáveis (SOM, em inglês), constrói uma rede de camada dupla. Essas camadas, de entrada e saída, são conectadas através de conexões sinápticas.

As redes de Kohonen, como são também chamadas, utilizam regras de aprendizado competitivo, onde os neurônios de uma camada competem entre si pelo privilégio de permanecerem ativos. Esse processo permite que o neurônio com maior atividade seja o principal participante do processo de aprendizado. Esse processo é conhecido como *winner-take-all*.

A rede se auto-organiza de maneira que vetores de entrada muito similares ativam o mesmo neurônio, definindo conjuntos de vetores de entrada associados a cada neurônio. De forma semelhante, conjuntos de vetores que apresentem similaridades devem ativar neurônios próximos no mapa de saída. Dessa forma, comparando a localização de cada saída no mapa tem-se uma

estimativa de quão similares os vetores de entrada estão entre si. Cada região do mapa destaca determinadas similaridades e funções relacionadas ao problema em questão, isto caracteriza a denominação de mapa de funções-auto-organizáveis ou *self-organizing feature map*.

A rede de Kohonen se caracteriza por utilizar uma topologia na qual os neurônios da camada de entrada estão ligados a todos os neurônios da camada de saída. Estes, por sua vez, estão ligados a alguns neurônios na camada de saída, os quais constituem sua vizinhança. A partir da figura 2.4 abaixo é possível observar os vetores na camada de entrada $x_1, \ldots, x_i$ ligados a todos os neurônios na camada de saída. No lado direito da figura é observado o mapa de características formado a partir da classificação obtida a partir dos valores da camada de saída.

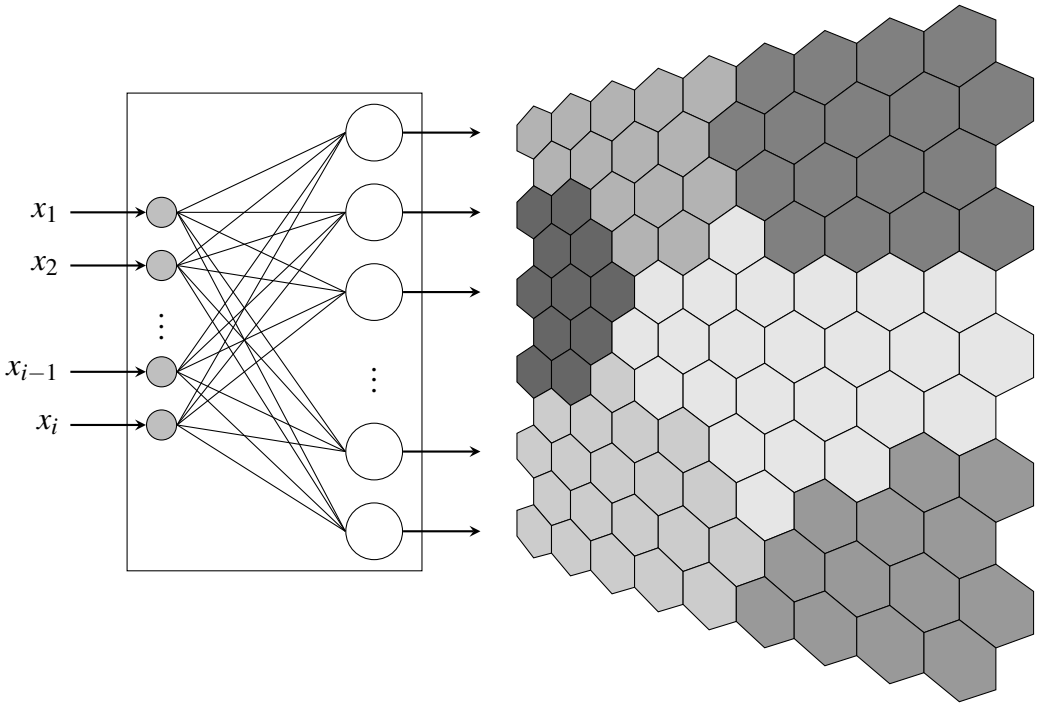

Figura 2.4 Topologia e mapa de características de uma rede de Kohonen.

2.5 Redes de Agrupamento Fuzzy Kohonen

Os modelos das redes de agrupamento Fuzzy Kohonen surgiram a partir do interesse de unificar os dois modelos afim de se obter um melhor desempenho na classificação de base de dados. As redes neurais são ferramentas poderosas no processamento de dados numéricos devido a sua capacidade computacional [26]. Já os modelos difusos agregam valor aos resultados pois fornecem uma saída mais representativa da classificação dos dados, através dos graus de

pertinência.

Pode-se separar as maneiras de integrar esses dois modelos em duas categorias. A primeira funciona a partir da incorporação da lógica difusa na arquitetura de redes neurais. A outra refere-se ao uso das redes neurais no modelo difuso. Geralmente, a taxa de aprendizagem em redes neurais não considera o grau em que um padrão de entrada pertence a um grupo ou categoria. Assim, a substituição da taxa de aprendizagem arbitrária por uma função de pertinência difusa pode produzir melhores resultados de agrupamento [32].

Vários autores abordaram a incorporação da lógica difusa na rede neural. Lee e Lee transformaram o modelo do neurônio McCulloch-Pitts em um modelo difuso e o chamaram de neurônio difuso [33]. Keller e Hunt incorporaram funções de grau de pertinência no algoritmo do *perceptron* [34]. Huntsberger e Ajjimarangsee propuseram a fuzzificação do mapa de recursos auto-organizados de Kohonen [35]. Eles usaram um valor de grau de pertinência difuso em vez da taxa de aprendizado arbitrária. Bezdek propôs um modelo de rede de agrupamento Kohonen (FKCN) [15, 32] que integra o modelo *fuzzy c-means* (FCM) [13] na taxa de aprendizagem e estratégias de atualização da rede Kohonen (KCN) [13].

2.5.1 O algoritmo

A rede de agrupamento difusa de Kohonen (FKCN, em inglês) é um algoritmo de agrupamento em lotes que combina as ideias de grau de pertinência difuso para taxas de aprendizado, o paralelismo do *fuzzy c-means* e a estrutura e regras auto-organizáveis de atualização da rede de agrupamento Kohonen.

A estrutura de uma rede de Kohonen consiste em duas camadas, uma camada de espalhamento de entrada e uma camada competitiva de saída. Dado um vetor de entrada, os nós na camada de saída competem entre si e o vencedor (cujo peso no grupo tem a distância mínima da entrada) atualiza seus pesos do grupo em algum conjunto de vizinhos predefinidos. Além disso, a função de vizinhança de atualização deve ser definida e reduzida com o tempo [16].

Uma escolha apropriada para a taxa de aprendizagem com a quantidade de etapas de aprendizagem é importante para obter bons resultados e convergência rápida. Se a redução da taxa de aprendizagem for feita de maneira muito rápida, a capacidade das células propagarem os resultados diminuem drasticamente antes que o mapa atinja um estado de equilíbrio. Se a redução for muito lenta, o processo demora mais do que o necessário [36].

O FKCN integra o modelo do *fuzzy c-means* com a taxa de aprendizagem e as estratégias de atualização da rede de Kohonen. Uma vez que os algoritmos FCM são métodos ótimos, a integração de métodos FCM e redes de Kohonen resolve de uma só vez vários problemas dos

métodos de Kohonen. Na rede de agrupamento difusa de Kohonen, a distribuição das taxas de aprendizagem e o tamanho das vizinhanças são controladas alterando o valor dos expoentes de peso com o tempo. Em vez de diminuir as taxas de aprendizado e o tamanho da vizinhança para zero, ele diminui o valor do expoente de peso de uma determinada constante positiva maior que um. Quando o valor do expoente de peso atinge o valor mínimo, apenas o vencedor é atualizado com o grau de pertinência similarmente ao algoritmo *c-means*.

A FKCN garante que os pesos do grupo convergem minimizando uma função objetivo como no modelo do *fuzzy c-means*. Devido à sua característica de paralelismo, o FKCN é independente da ordem dos padrões de entrada. Assim como, o algoritmo sempre para no mesmo conjunto de pesos de nós [32].

Seja $\Omega = \{1, ..., n\}$ um conjunto de n objetos indexados por k onde cada objeto é representado por um vetor de p valores contínuos $\mathbf{x}_k = (x_k^1, ..., x_k^p)$. Considere também um conjunto de c pesos de grupos indexados por i onde cada objeto é descrito por um vetor de valores contínuos

$\mathbf{v}_i = (v_i^1, ..., v_i^k)$. Uma versão resumida do algoritmo é descrita abaixo [32].

Algoritmo 2: Algoritmo FKCN

Ajuste o número de grupos c. **Defina** o valor do critério de parada ε.

Defina o número máximo de iterações t_{max} e $m_0 > 1$. **Inicialize** o vetor de pesos $\mathbf{v}_1(0), \mathbf{v}_2(0), ..., \mathbf{v}_c(0)$.

for $t = 1, ..., t_{max}$ **do**

  Calcule a taxa de aprendizagem:

$$m_t = m_0 - t * \frac{m_0 - 1}{t_{max}}$$

  for *todo o conjunto de dados* **do**

    Calcule o grau de pertinência difuso:

$$u_{ik,t} = \left[\sum_{h=1}^{c} \left(\frac{\sum_{j=1}^{p} (x_k^j - v_{i,t-1}^j)^2}{\sum_{j=1}^{p} (x_k^j - v_{h,t-1}^j)^2} \right)^{(1/m_t - 1)} \right]^{-1}$$

    Calcule a taxa de aprendizagem difusa:

$$\theta_{ik,t} = (u_{ik,t})^{m^t}$$

  end

  Atualize os pesos dos grupos:

$$\mathbf{v}_{i,t} = \frac{\sum_{k=1}^{n} \theta_{ik,t} \mathbf{x}_k}{\sum_{k=1}^{n} \theta_{ik,t}}$$

  Calcule o critério de parada:

$$E_t = \|\mathbf{v}_t - \mathbf{v}_{t-1}\|^2$$

  Se $E_t \leqslant \varepsilon$ **Pare**. Caso contrário, vá para o próxima iteração t.

end

2.6 Trabalhos relacionados

Nesta sessão, serão apresentados algumas abordagens em que métodos univariados foram usados como base para o desenvolvimento dos seus respectivos métodos multivariados. Todos

esses métodos foram desenvolvidos com o propósito de otimizar os resultados obtidos por suas versões não multivariadas. O desenvolvimento da grande maioria dessas abordagens foi proporcionado através da popularização do poder computacional observado a partir da década de 2010, onde o preço do gigaFLOP ultrapassou a barreira de menos de 1 dólar americano [37].

O primeiro desses métodos, o qual deu origem a este trabalho, foi proposto por Pimentel e Souza [17] em 2013. O método desenvolvido por Pimentel é baseado no *fuzzy c-means*. O método do FCM apresenta bom desempenho na detecção de agrupamentos, porém não fornece uma saída detalhada devido aos graus de pertinência dos objetos não serem fornecidos por variável dentro dos grupos. Em contra partida, o método *Multivariate Fuzzy C-Means* (ou MFCM) produz uma matriz de graus de pertinência para cada grupo e por cada variável. Devido a maneira como o algoritmo apresenta sua saída é possível identificar qual ou quais características influenciaram de forma decisiva para tal classificação.

Outro método proposto por Pimentel e Souza [38] na área de Análise de Dados Simbólicos (SDA, Symbolic Data Analysis em inglês) também implementa a abordagem multivariada na classificação dos dados. A área de estudo da SDA foi desenvolvida como uma extensão da Análise de Dados tradicional devido a complexidade encontrada em um grupo de dados no qual estão presentes, além dos valores ou categorias, intervalos internos e estruturas [39]. O método propôs um método de classificação baseado em uma abordagem difusa que produz graus de pertinência multivariados ajustados por pesos para dados intervalares. O método apresentado se mostrou robusto na atribuição ambígua de graus de pertinência dos grupos devido aos pesos representarem as diferentes importâncias das quais as variáveis tem com os grupos.

Posteriormente, Pimentel e Souza [40] apresentaram um modelo do MFCM, desenvolvido em 2014, porém com a consideração da influência de desbalanceada das variáveis presentes no conjunto de dados. Essa variação do algoritmo utiliza pesos para representar a importância que cada variável tem para cada grupo e otimizar a qualidade da classificação.

Influenciados pelos trabalhos de Pimentel e Souza, Himmelspach[41] apresentou uma versão multivariada do algoritmo proposto por Pal *et al.* [12]. O método desenvolvido por Pal *et al.* difere do FCM clássico por produzir, além dos graus de pertinências, valores possibilísticos usados no método do Possibilistic C-Means (PCM). Esses valores possibilísticos proporcionam ao PFCM uma otimização no desempenho de classificação em relação ao FCM devido a tendência do segundo em formar grupos coincidentes. Portanto, o PFCM pode ser considerado como uma versão híbrida do FCM com o PCM. Por conseguinte, Himmelspach propôs um algoritmo que produz uma partição multivariada do conjunto de dados que detecta grupos com dados distribuídos desigualmente nas variáveis. Este também reduz a influência do ruído e *outliers* no cálculo dos centroides.

Outra contribuição baseada no trabalho de Pimentel e Souza foi proposta por Maciel *et al.* [42]. O algoritmo proposto implementa a abordagem multivariada do MFCM para o *fuzzy k-modes*. O *fuzzy k-modes*[43] foi apresentado como uma abordagem baseada no paradigma do FCM, mas dedicada a classificação eficiente de grandes conjuntos de dados categóricos. O FkM (fuzzy k-modes) é caracterizado por usar uma medida de dissimilaridade de correspondência simples para objetos categóricos, bem como o uso da moda no lugar das médias usada no FCM. Assim sendo, o MFkM, como chamado pelo autor, faz uso de matrizes de graus de pertinência múltiplos para tratar de uma forma melhor a ambiguidade de dados que compartilhem propriedades em diferentes grupos.

Com o intuito de apresentar uma interpretação dos métodos multivariados propostos, Pimentel *et al.* propuseram uma estratégia de interpretação dos graus de pertinências multivariados. A estratégia proposta se baseia nos seguintes fatos: (1) o grau de pertinência de uma dada variável para um determinado objeto deve ser maior no grupo onde essa variável é mais discriminativa e (2) quanto maior for o grau de pertinência de uma variável para um grupo, mais relevante essa variável é para caracterizá-lo. Assim sendo, a estratégia se resume a calcular a média dos graus de pertinência multivariados considerando todos os objetos. Os resultados mostraram que os graus de pertinência multivariados podem apontar as variáveis mais importantes para descrever cada grupo.

CAPÍTULO 3

Método proposto e Metodologia

O método proposto é um modelo difuso de classificação não-supervisionado que busca resolver o problema de otimização que minimiza, de forma interativa, a entropia entre os elementos. Ele é baseado no FKCN [13] que integra o modelo do FCM [13, 7] com as taxas de aprendizagem e estratégias de atualização do KCN [16]. O FKCN busca minimizar a equação 3.1, levando em conta a contribuição do grupo de variáveis de maneira global, garantindo a convergência dos pesos dos grupos.

O processo de integração do FCM com o KCN foi feito por Bezdek definindo a equação de atualização da taxa de aprendizagem da rede de Kohonen por 3.2, onde m_0 é o valor inicial do peso exponencial maior do que 1, t_{max} é o valor máximo de iterações. A taxa de aprendizagem difusa foi definida por 3.3, onde $u_{ik,t}$ é o valor difuso do grau de pertinência do padrão de entrada $\mathbf{x}_k$ no i-ésimo grupo. O método do FKCN converte um processo heurístico de aprendizagem do Kohonen Clustering Network (KCN) em um processo de otimização de uma função objetivo, então o resultado da iteração é independente da ordem de entrada dos dados e a velocidade de convergência é também otimizada [15, 32].

$$J(mt) = \sum_i \sum_k u_{ik}^m (\|\mathbf{x}_k - \mathbf{v}_i\|)^2 \tag{3.1}$$

$$m_t = m_0 - t * \frac{m_0 - 1}{t_{max}} \tag{3.2}$$

$$\theta_{ik,t} = (u_{ik,t})^{m^t} \tag{3.3}$$

O método proposto também incorpora a abordagem multivariada do Multivariate Fuzzy C-Means (MFCM) [17] que busca minimizar a equação 3.4 na qual a contribuição de cada variável na entropia é calculada de forma independente. No algoritmo original do FCM [13, 7] o objetivo é encontrar um conjunto de protótipos L e uma partição difusa $\mathbf{U} = [ik](i = 1,...,c)(k = 1,...,n)$ do conjunto de dados, minimizando a função objetivo dada por 3.1. A partição difusa U representa uma matriz de graus de pertinência onde u_{ik} é o grau de pertinência de um dado ponto k que pertence ao cluster i observadas a seguintes restrições: 3.5, 3.6, 3.7.

No algoritmo MFCM, os graus de pertinência são diferente de uma variável para outra e de um grupo para outro. Seja λ um conjunto de n padrões indexados por k e formado por p características indexadas por j. Cada padrão j é representado por um vetor de características $\mathbf{x}_k = (x_{1k}, ..., x_{pk})^t$. Seja $L = \{\mathbf{y}_1, ..., y_c\}$ um conjunto de c protótipos associados a uma partição difusa de c agrupamentos. Cada protótipo de um grupo também é representado como um vetor de características $\mathbf{y}_i = (y_{i1}, ..., x_{ip})^t$. Essa mudanças na equação da função objetivo do algoritmo para que a representação apropriada seja utilizada aliada à forma de calcular a distância entre os grupos e vetor de protótipos estão representados na Equação 3.4.

$$J^M = \sum_i \sum_k \sum_j u_{ikj}^{m_t} (\|x_{jk} - y_{ij}\|)^2 \tag{3.4}$$

$$u_{ijk} \in [0,1] \ \ \forall i, j, k \tag{3.5}$$

$$0 < \sum_{j=1}^{p} \sum_{k=1}^{n} u_{ijk} < n \ \ \forall i, j, k \tag{3.6}$$

$$\sum_{i=1}^{c} \sum_{j=1}^{p} u_{ijk} = 1 \ \ \forall k. \tag{3.7}$$

O algoritmo para a abordagem multivariada apresentada neste trabalho será descrita abaixo. Essa abordagem pode ser entendida como a atualização da função objetivo do algoritmo FKCN, a qual usa o algoritmo FCM, para o algoritmo multivariado MFCM apresentado por [17]. Essa atualização implica que a matriz de graus de pertinências será representada por mais uma dimensão. Essa dimensão representa a partição do grau de pertinência de cada exemplo em p variáveis. Uma maneira de aglutinar esses graus de pertinência de forma a obter o grau de pertinência do objeto k ao grupo C_i foi proposta por [17]. Então, o valor do grau de pertinência δ_{ik} para todos os fatores p é dado pela equação 3.8.

$$\delta_{ik} = \sum_{j=1}^{p} u_{ijk}. \tag{3.8}$$

Algoritmo 3: FKCN multivariado

1 **Ajuste** o número de grupos c. **Defina** o valor do critério de parada ε.

2 **Defina** o número máximo de iterações t_{max} e $m_0 > 1$. **Inicialize** o vetor de pesos $\mathbf{v}_1(0), \mathbf{v}_2(0), ..., \mathbf{v}_c(0)$.

3 Para $t = 1, ..., t_{max}$, **faça**:

1. **Calcule** a taxa de aprendizagem:

$$m_t = m_0 - t * \frac{m_0 - 1}{t_{max}}$$

2. Para todos o conjunto de dados:

   (a) **Calcule** o grau de pertinência difuso:

$$u_{ijk,t} = \left[\sum_{h=1}^{c} \sum_{l=1}^{p} \left(\frac{d_{ijk}}{d_{hlk}} \right)^{(1/m_t - 1)} \right]^{-1}$$

   (b) **Unifique** os graus de pertinência:

$$\delta_{ik,t} = \sum_{j=1}^{p} (u_{ijk,t})$$

   (c) **Calcule** a taxa de aprendizagem difusa:

$$\theta_{ik,t} = (\delta_{ik,t})^{m^t}$$

3. **Atualize** os pesos dos grupos:

$$\mathbf{v}_{i,t} = \frac{\sum_{k=1}^{n} \theta_{ik,t} \mathbf{x}_k}{\sum_{k=1}^{n} \theta_{ik,t}}$$

4. **Calcule** o critério de parada:

$$E_t = \|\mathbf{v}_t - \mathbf{v}_{t-1}\|^2$$

   Se $E_t \leqslant \varepsilon$ **Pare**. Caso contrário, vá para o próxima iteração t.

CAPÍTULO 4

Experimentos e Resultados

Com o objetivo de avaliar o desempenho do métodos proposto neste trabalho, dois conjuntos de dados sintéticos e quatro configurações de dados reais foram considerados. Nos dados sintéticos, foram produzidos com o intuito de avaliar o comportamento do método proposto em cenários mais comportados. Enquanto os dados reais foram escolhidos para avaliar o desempenho do método proposto em situações reais e compará-lo com outros modelos. A comparação é feita com o método FKCN original proposto por Bezdek [15].

A avaliação dos agrupamentos resultantes da execução dos métodos é baseado nas seguintes métricas: índice de Rand Difuso[19] (FR index, em inglês), na taxa de erro global de classificação (OERC), na quantidade de iterações necessárias para atingir a convergência e na taxa final do valor de fuzzificação. O índice difuso de Rand, foi criado a partir de uma extensão do índice de Rand ajustado [44]. Este índice (CR a partir daqui) tem valores no intervalo [-1,1] onde o valor 1 indica uma perfeita correspondência entre a partição original e a obtida pelo método de agrupamento, valores perto de 0 correspondem a uma correspondência aleatória e valores negativos indicam uma insuficiência do método em reconhecer as propriedades da configuração de dados avaliada.

Apesar das características citadas acima, o índice de Rand ajustado não considera as informações das partições e não é apropriado para verificação de agrupamentos difusos. Entretanto, a extensão dele resolve esse problema, o índice FR considera partições difusas de um conjunto de dados. A extensão difusa tem valores no intervalo [0,1] onde o valor 1 indica uma perfeita correspondência entre as partições difusas, o contrário do valor 0 que indica falta de correspondência entre as partições. O FR para 2 partições difusas P e Q é dado por:

$$FR(P,Q) = 1 - \frac{\sum_{k=1}^{n}\sum_{k'=1}^{n}|E_P(\mathbf{x}_k,\mathbf{x}_{k'}) - E_Q(\mathbf{x}_k,\mathbf{x}_{k'})|}{n(n-1)/2} \tag{4.1}$$

onde $E_P(\mathbf{x}_k,\mathbf{x}_{k'}) = 1 - ||\delta_k - \delta'_k||$ e $\delta_k = (\delta_{1k},...,\delta_{ik},...,\delta_{ck})$ é um vetor de graus de pertinência por grupo do objeto $\mathbf{x}_k$, similarmente no E_Q. Neste trabalho, a métrica $||.||$ usada no intervalo $[0,1]^c$ que geram valores no mesmo intervalo $[0,1]$ é dado por $d(\delta_k,\delta_{k'}) = \frac{1}{c}\sum_{i=1}^{c}(\delta_{ik} - \delta_{ik'})^2$.

O índice OERC [45] tem seus valores no intervalo [0,1] onde o valor 0 indica uma perfeita correspondência entre a partição original e a obtida pelo método de agrupamento, por outro

lado valor 1 representa que a partição original e a obtida são totalmente diferentes.

Para os conjuntos de dados sintéticos, em todas as métricas acima descritas, seus valores foram calculados usando o experimento Monte Carlo com 100 repetições. Para cada conjunto de dados real, o método de agrupamento foi executado, até que a convergência fosse atingida, por 50 vezes e o melhor resultado de acordo com o critério da menor função objetivo foi selecionado. Após as 100 repetições a média e o desvio padrão dos índices foram calculados. O mesmo processo foi usado para os conjuntos de dados sintéticos em cada repetição Monte Carlo. Para todos os experimentos, o número máximo de iterações foi fixado em 1000 iterações e o valor inicial da taxa de aprendizado usado foi de $m_0 = 2.0$.

4.1 Dados Sintéticos

Dois conjuntos de dados sintéticos no R^2 foram observados nesse trabalho. Cada conjunto apresenta três classes com tamanhos e forma diferentes. Cada classe foi gerada com base em uma distribuição normal bivariada com os valores de médias, matriz de covariância e cardinalidade dados pelas tabelas abaixo.

Tabela 4.1 Parâmetros usados na geração do conjunto 1 de dados sintéticos.

Parâmetros	Classe 1	Classe 2	Classe 3
μ_1	60	70	90
μ_2	50	60	70
σ_1^2	49	36	49
σ_2^2	36	49	36
Cardinalidade	100	400	100

Tabela 4.2 Parâmetros usados na geração do conjunto 2 de dados sintéticos.

Parâmetros	Classe 1	Classe 2	Classe 3
μ_1	55	70	95
μ_2	20	40	55
σ_1^2	100	25	64
σ_2^2	9	81	16
Cardinalidade	100	400	100

É possível observar, a partir das configurações apresentadas, que a configuração 1 gera conjuntos onde as matrizes de covariâncias entre as classes e as variâncias dentro da classe são similares. Em contrapartida, a configuração 2 apresenta matrizes de covariância diferente entre as classes e variância diferentes dentro das classes. Essas configurações foram escolhidas para testar o desempenho do método proposto em cenários onde o uso da abordagem multivariada pode trazer melhorias.

Nas imagens que se seguem os conjuntos de dados sintéticos 1 e 2 são apresentados de maneira gráfica. Em sequência, as tabelas 4.3 a 4.6 apresentam os resultados obtidos e seus respectivos desvios-padrões a partir de cada método e para cada classe.

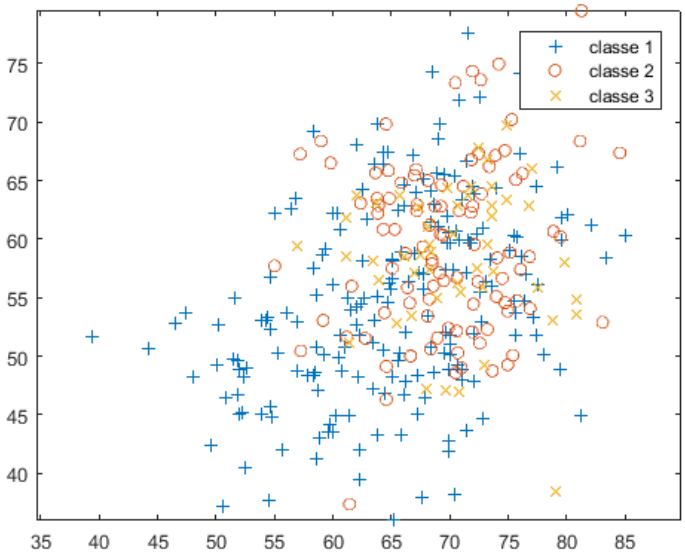

Figura 4.1 Imagem gerada a partir de uma execução aleatória do conjunto de dados 1.

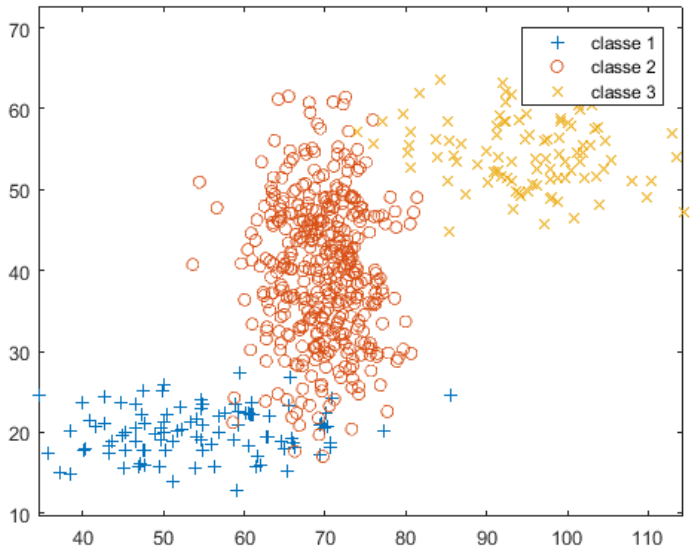

Figura 4.2 Imagem gerada a partir de uma execução aleatória do conjunto de dados 2.

Tabela 4.3 Médias e desvios-padrões do índice FR para os métodos de agrupamento e conjunto de dados 1 e 2.

Índice difuso de Rand				
Conjunto	FKCN		MFKCN	
	Média	Desvio Padrão	Média	Desvio Padrão
1	0,3322	3×10^{-6}	0,4265	0,0195
2	0,3550	0,0836	0,6750	0,0365

Tabela 4.4 Médias e desvios-padrões do índice OERC para os métodos de agrupamento e conjunto de dados 1 e 2.

Taxa de erro global de classificação				
Conjunto	FKCN		MFKCN	
	Média	Desvio Padrão	Média	Desvio Padrão
1	0,4225	0,0091	0,4112	0,0157
2	0,4216	0,0791	0,1950	0,0341

Tabela 4.5 Médias e desvios-padrões valor da taxa de aprendizagem *m* para os métodos de agrupamento e conjunto de dados 1 e 2.

Taxa de aprendizagem (m)				
Conjunto	FKCN		MFKCN	
	Média	Desvio Padrão	Média	Desvio Padrão
1	1,9980	2×10^{-15}	1,9773	0,0045
2	1,9969	0,0041	1,9753	0,0058

Tabela 4.6 Médias e desvios-padrões valor da distância de Hausdorff para os métodos de agrupamento e conjunto de dados 1 e 2.

Distância de Hausdorff				
Conjunto	FKCN		MFKCN	
	Média	Desvio Padrão	Média	Desvio Padrão
1	18,3867	0,6497	6,5727	1,2115
2	22,0689	5,0782	5,3822	1,0439

A Tabela 4.3 mostra que o MFKCN obteve vantagem nos cenários considerados, para o índice difuso de Rand. O mesmo comportamento foi observado no índice do OERC no qual o método proposto obteve grande vantagem na classificação do segundo cenário, apresentado na Tabela 4.4. Na Tabela 4.5 é apresentado os valores da taxa de apredizagem *m* para as execuções dos métodos. Por fim, na Tabela 4.6 é apresentado o cálculo da distância de Hausdorff [46] entre os protótipos originais do conjunto e os protótipos finais a partir da execução do algoritmo. Essa métrica é interessante pois demonstra o quão próximos ou não estão os protótipos finais daqueles vetores que representam o conjunto de dados.

4.2 Dados Reais

Com o propósito de avaliar o método proposto em dados reais, o método foi executado para cinco conjuntos de dados, os quais podem ser encontrados no repositório de aprendizagem de máquina da Universidade da Califórnia Irvine (UCI Machine Learning Repository) [47]. Os conjuntos usados foram: Abalone, Balance Scale, Haberman, Pima Indians e Iris.

O Abalone ou Haliotis, é um gênero de moluscos gastrópodes marinhos da família Haliotidae. O conjunto de dados Abalone contêm informações sobre medidas físicas do Abalone realizadas com o propósito de estimar a idade. Sendo assim, serão consideradas as seguintes características contínuas: comprimento da medida mais longa da concha (comprimento), diâmetro medido perpendicularmente ao comprimento (diâmetro), altura, peso total, peso ajustado, peso das vísceras, peso da concha e anéis. O conjunto é formado por 4177 padrões dispersos em 3 classes.

O conjunto do Balance Scale foi gerado para modelar resultados experimentais psicológicos. Cada exemplo é classificado como tendo a escala de equilíbrio inclinada para a direita, inclinada para a esquerda ou equilibrada. Os atributos são o peso à esquerda, a distância à esquerda, o peso correto e a distância correta. A maneira correta de encontrar a classe é o maior produto entre o peso de cada lado vezes a distância do respectivo lado. Se os valores são iguais, a balança está equilibrada. Esse conjunto tem 625 exemplos divididos entre 49 exemplos balanceados e 288 inclinados para cada lado.

O conjunto de dados do Haberman's Survival contém casos de um estudo realizado entre 1958 e 1970, no Hospital Billings, da Universidade de Chicago, sobre a sobrevida de pacientes submetidos à cirurgia para câncer de mama. Os dados são compostos por 3 atributos: idade do paciente na operação, ano de operação do paciente e o número de nós axilares positivos detectados. A formação do conjunto é de 306 exemplos divididos em 2 classes que indicam que o paciente sobreviveu por 5 anos ou mais, ou não.

O conjunto Pima Indians Diabetes [48] é composto por 768 observações de detalhes médicos para as patentes dos índios Pima. Os registros descrevem medidas instantâneas do paciente, como idade, número de gestantes e exames de sangue. Todos os pacientes são mulheres com 21 anos ou mais. Todos os atributos são numéricos e suas unidades variam de atributo para atributo. Cada registro tem um valor de classe que indica se o paciente sofreu um ataque de diabetes dentro de 5 anos a partir do momento em que as medidas foram tomadas (1) ou não (0).

Por fim, o conjunto Iris contém 3 classes de 50 instâncias cada, onde cada classe se refere a um tipo de planta da íris. Uma classe é linearmente separável das outras duas, as últimos não são linearmente separáveis um do outro. Um dos bancos de dados mais conhecidos encontrado na literatura sobre reconhecimento de padrões.

Os índices encontrados para a execuções dos métodos para o Abalone, Balanced Scale, Haberman's Survival, Pima Indians Diabetes e Iris são apresentados nas Tabelas abaixo.

Tabela 4.7 Valores dos índices obtidos para a melhor execução em 200 inicializações para o conjunto Abalone.

Índices	FKCN	MFKCN
Fuzzy Rand	0,3315	0,3951
Função objetivo (J)	5725,9294	0,2964
Distância de Hausdorff	0,7970	0,6851
OERC	0,4120	0,3886
Número de iterações	17	8
Taxa de aprendizagem	1,9830	1,9920

Tabela 4.8 Médias e desvio padrão das características por classe para o conjunto de dados Abalone.

Características	Classe 1 (1307 exemplos)		Classe 2 (1342 exemplos)		Classe 3 (1528 exemplos)	
	Média	Desvio	Média	Desvio	Média	Desvio
Comprimento	0,5791	0,0862	0,4277	0,1089	0,5614	0,1027
Diâmetro	0,4547	0,0710	0,3265	0,0881	0,4393	0,0844
Altura	1,0465	0,0400	0,1080	0,0320	0,1514	0,0348
Peso total	0,4462	0,4303	0,4314	0,2863	0,9915	0,4706
Peso ajustado	0,2307	0,1987	0,1910	0,1284	0,4329	0,2230
Peso das víceras	0,3020	0,0976	0,0920	0,0625	0,2151	0,1049
Peso da concha	0,0158	0,1256	0,1282	0,0849	0,2820	0,1308
Anéis	11,1293	3,1043	7,8905	2,5116	10,7055	3,0263

Tabela 4.9 Valores dos índices obtidos para a melhor execução em 200 inicializações para o conjunto Balance Scale.

Índices	FKCN	MFKCN
Fuzzy Rand	0,2401	0,3217
Função objetivo (J)	1005,5576	21,4359
Distância de Hausdorff	0,6179	0,6111
OERC	0,3960	0,4672
Número de iterações	2	8
Taxa de aprendizagem	1,9980	1,992

Tabela 4.10 Médias e desvio padrão das características por classe para o conjunto de dados Balance Scale.

Características	Classe 1 (49 exemplos)		Classe 2 Esq. (288 exemplos)		Classe 3 Dir. (288 exemplos)	
	Média	Desvio	Média	Desvio	Média	Desvio
Peso lado esquerdo	2,9388	1,4202	3,6111	1,2275	2,3993	1,3318
Distância lado esquerdo	2,9388	1,4202	3,6111	1,2275	2,3993	1,3318
Peso lado direito	2,9388	1,4202	2,3993	1,3318	3,6111	1,2275
Distância lado direito	2,9388	1,4202	2,3993	1,3318	3,6111	1,2275

Tabela 4.11 Valores dos índices obtidos para a melhor execução em 200 inicializações para o conjunto Haberman's Survival.

Índices	FKCN	MFKCN
Fuzzy Rand	0,2189	0,1409
Função objetivo (J)	2519,2858	463,2720
Distância de Hausdorff	3,4301	2,9138
OERC	0,5013	0,4990
Número de iterações	2	22
Taxa de aprendizagem	1,9980	1,9780

Tabela 4.12 Médias e desvio padrão das características por classe para o conjunto de dados Haberman's Survival.

Características	Classe 1 (225 exemplos)		Classe 2 (81 exemplos)	
	Média	Desvio	Média	Desvio
Idade do paciente	52,0178	11,0122	53,6790	10,1671
Ano da operação	62,8622	3,2229	62,8272	3,3421
Número de nódulos positivos	2,7911	5,8703	7,4568	9,1857

Tabela 4.13 Valores dos índices obtidos para a melhor execução em 200 inicializações para o conjunto Pima Indians Diabetes.

Índices	FKCN	MFKCN
Fuzzy Rand	0,0901	0,0463
Função objetivo (J)	5938,4399	22,1160
Distância de Hausdorff	20,3226	17,4885
OERC	0,4826	0,4859
Número de iterações	2	15
Taxa de aprendizagem	1,9980	1,9850

Tabela 4.14 Médias e desvio padrão das características por classe para o conjunto de dados Pima Indians Diabetes.

Características	Classe 1 (500 exemplos)		Classe 2 (268 exemplos)	
	Média	Desvio	Média	Desvio
Quantidade de gravidezes	3,2980	3,0172	4,8657	3,7412
Nível de glucose no sangue	109,9800	26,1412	141,2575	31,9396
Pressão arterial diastólica	68,1840	18,0631	70,8246	21,4918
Espessura do tríceps	19,6640	14,8899	22,1642	17,6797
Nível de insulina no sangue	68,7920	98,8653	100,3358	138,6891
IMC	30,3042	7,6899	35,1425	7,2630
Pedigree diabético	0,4297	0,2991	0,5505	0,3724
Idade	31,1900	11,6677	37,0672	10,9683

Tabela 4.15 Valores dos índices obtidos para a melhor execução em 200 inicializações para o conjunto Iris.

Índices	FKCN	MFKCN
Fuzzy Rand	0,5099	0,4579
Função objetivo (J)	114,1258	0,9702
Distância de Hausdorff	0,4075	0,4336
OERC	0,0955	0,2274
Número de iterações	5	5
Taxa de aprendizagem	1,9950	1,9950

Tabela 4.16 Médias e desvio padrão das características por classe para o conjunto de dados Iris.

Características	Classe 1 (50 exemplos)		Classe 2 (50 exemplos)		Classe 3 (50 exemplos)	
	Média	Desvio	Média	Desvio	Média	Desvio
Comprimento de sépala	5,0060	0,3525	5,9360	0,5162	6,5880	0,6359
Largura de sépala	3,4180	0,3810	2,7700	0,3138	2,9740	0,3225
Comprimento de pétala	1,4640	0,1735	4,2600	0,4699	5,5520	0,5519
Largura de pétala	0,2440	0,1072	1,3260	0,1978	2,0260	0,2747

A partir da Tabela 4.7 pode-se observar que o MFKCN obteve um melhor desempenho em relação ao FKCN no Fuzzy Rand, OERC, número de iterações e taxa de aprendizagem para o conjunto Abalone. Na Tabela 4.9, referente ao conjunto Balance Scale, pode-se observar um melhor desempenho do MFKCN apenas no índice Fuzzy Rand, enquanto que os outros índices obtiveram valores próximos mas a favor do FKCN. Para o conjunto Haberman's Survival, os dois métodos obtiveram desempenho ruins o qual pode ser justificado pela alta variância encontrada nas suas variáveis. Os valores das variâncias para o Haberman's Survival pode ser observado na Tabela 4.11. Na Tabela 4.13, onde pode ser observado o conjunto Pima Indians Diabetes, o FKCN obteve vantagem apesar do baixo desempenho de ambos os métodos. Novamente os valores das variância das variáveis do conjunto foram bastante elevadas. Por fim, para o conjunto Iris, observado na Tabela 4.15, o método proposto obteve vantagem apenas na otimização da função objetivo, sendo semelhante ou obtendo desvantagem nos outros critérios.

As Tabelas 4.8, 4.10, 4.12, 4.14 e 4.16 apresentam os valores das médias e desvios-padrões para cada conjunto e por cada classe. A partir desses dados e dos resultados dos índices para cada conjunto pode-se perceber que o método proposto apresenta vantagem em relação do método original em cenários onde as variáveis tem variâncias diferentes entre as classes.

Apesar do desempenho no índice difuso de Rand não atingir valores desejados para os conjuntos, em todos eles o valor da função objetivo (J) obtido foi menor no método proposto. O resultado desse índice demonstra que o método proposto alcança partições onde a função objetivo é menor em relação ao método original.

A partir dos dados apresentados pode-se perceber que o método proposto obteve melhor desempenho nos dois primeiros conjuntos de dados para a métrica difusa de Rand. Esse índice indica o quão próximo os valores dos graus de pertinência ficaram do conjunto hard esperado para os grupos. Em relação aos valores da função objetivo (J) pode-se observar que o MFKCN apresenta, em todos os casos, valores de J menores do que os valores do FKCN. Comparando os valores da distância de Hausdorff para os dois métodos pode-se perceber que o MFKCN em todos os casos entrega protótipos mais próximos aos originais. A distância de Hausdorff indica o quão próximo os protótipos finais gerados pelo método estão dos valores originais esperados calculados a partir das médias dos exemplos em cada conjunto.

CAPÍTULO 5

Conclusão e trabalhos futuros

Este trabalho apresentou uma nova abordagem para o algoritmo do FKCN. O MFKCN, modelo apresentado, se diferencia de outros modelos propostos na literatura pois utiliza no cálculo dos graus de pertinência a contribuição de cada variável de forma independente. O MFKCN tem como referência a abordagem do Multivariate Fuzzy C-Means, o qual implementou o cálculo dos graus multivariados para o Fuzzy C-Means. Neste trabalho foi demonstrado através do uso dos índices difuso que o métodos proposto para o cálculo dos graus multivariados é pertinente e permite alcançar bons resultados.

Foram realizados experimentos com dados sintéticos e reais em comparação com o FKCN tradicional. Para os conjuntos de dados sintéticos, os índices foram calculados usando o método Monte Carlo com 100 replicações de cada conjunto. Nos experimentos com dados reais, foram usados conjuntos de dados do repositório da UCI e os índices foram obtidos da melhor execução de 200 repetições.

Os resultados baseados na métrica do índice difuso de Rand, valor final da função objetivo (J) e distância de Hausdorff mostraram que, apesar das características apresentarem dispersões diversas, o método proposto foi superior ao FKCN na maioria dos cenários executados. A superioridade do método proposto em comparação com o FKCN neste trabalho pode ser explicada pelo cálculo dos graus de membros obtidos levando em conta as informações das características. Neste caso, o método proposto supõe que as características não são consideradas igualmente importantes e possuem dispersões diversas.

Como trabalhos futuros pretende-se definir uma estratégia para aplicar o método proposto em dados intervalares, bem como comparar seus resultados com outros métodos de agrupamento de dados conhecidos e mais utilizados na literatura. Pretende-se ainda realizar testes com outros cenários e testar outras configurações de inicialização dos parâmetros e avaliar seu comportamento em comparação com o método proposto.

Referências Bibliográficas

[1] T. W. Rauber, "Redes neurais artificiais," *Universidade Federal do Espírito Santo*, 2005. (document), 2.3, 2.4.3

[2] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, third ed., Jan. 2011. 1

[3] D. J. C. MacKay, *Information theory, inference and learning algorithms*. Cambridge University Press, Oct. 2003. 1

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research 12*, pp. 2825–2830, Oct. 2011. 1

[5] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," *In Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 19–26, 2001. 1

[6] A. K. Jain and R. C. Dubes, "Algorithms for clustering data.," *Prentice-Hall, Inc., Upper Saddle River, NJ, USA.*, 1988. 1

[7] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review.," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999. 1, 3, 3

[8] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set.," *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985. 1

[9] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis.," *The computer journal*, vol. 41, no. 8, pp. 578–588, 1998. 1

[10] R. Xu and D. Wunsch, "Survey of clustering algorithms.," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005. 1, 2.1.1, 2.1.2

[11] A. K. Jain, "Data clustering: 50 years beyond k-means.," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010. 1

[12] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A possibilistic fuzzy c-means clustering algorithm.," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 4, pp. 517–530, 2005. 1, 2.6

[13] J. C. Bezdek, "Pattern recognition with fuzzy objective function algorithms.," *Plenum Press, New York*, 1981. 1, 2.3, 2.5, 3, 3

[14] M. S. Yang, "A survey of fuzzy clustering," *Mathematical and Computer modelling*, vol. 18, no. 11, pp. 1–16, 1993. 1

[15] E. C. K. Tsao, J. C. Bezdek, and N. R. Pal, "Fuzzy kohonen clustering networks," *Pattern Recognition*, vol. 27, no. 5, pp. 757–764, 1994. 1, 2.5, 3, 4

[16] T. Kohonen, "Essentials of the self-organizing map," *Neural networks : the official journal of the International Neural Network Society*, vol. 37, Oct. 2012. 1, 2.4.1, 2.5.1, 3

[17] B. A. Pimentel and R. M. C. R. de Souza, "A multivariate fuzzy c-means method," *Applied Soft Computing 13*, p. 1592–1607, 2013. 1.1, 2.6, 3, 3

[18] C. W. D. de Almeida, R. M. C. R. de Souza, and L. B. Candeias Ana, "Fuzzy kohonen clustering networks for interval data.," *Neurocomputing 99*, p. 65–75, 2013. 1.1

[19] E. Hullermeier, M. Rifqi, S. Henzgen, and R. Senge, "Comparing fuzzy partitions: A generalization of the rand index and related measures," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 3, pp. 546–556, 2012. 1.1, 4

[20] B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis*. Wiley, 2001. 2.1, 2.1.2, 2.2.2

[21] M. V. Doni, "Análise de cluster: Métodos hierárquicos e de particionamento," Master's thesis, Universidade Presbiteriana Mackenzie, 2004. 2.1, 2.1, 2.2.2

[22] P. Berkhin, "A survey of clustering data mining techniques," *Grouping Multidimensional Data. Berlin Heidelberg: Springer*, pp. 25–71, 2006. 2.1, 2.1.2, 2.3.2

[23] G. Fung, "A comprehensive overview of basic clustering algorithms," *IOSR Journal of Computer Engineering (IOSRJCE)*, June 2001. 2.2, 2.2.1

[24] L. A. Zadeh, "Information and control," *Fuzzy sets*, vol. 8, no. 3, pp. 338–353, 1965. 2.3.3

[25] B. A. Pimentel, "Agrupamento de dados simbólicos usando abordagem possibilistic," *UFPE*, 2013. 2.3.4

[26] S. Haykin, *Redes neurais: princípios e prática*. Bookman Editora, 2007. 2.4, 2.5

[27] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958. 2.4.1

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, p. 533, 1986. 2.4.1

[29] B. Widrow and M. E. Hoff, "Adaptive switching circuits," tech. rep., STANFORD UNIV CA STANFORD ELECTRONICS LABS, 1960. 2.4.1

[30] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*, pp. 411–415, World Scientific, 1987. 2.4.1

[31] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural network design*. Martin Hagan, 2014. 2.4.3, 2.4.3

[32] J. C. Bezdek, E.-K. Tsao, and N. Pal, "Fuzzy kohonen clustering networks.," *Proceedings of the First IEEE Conference on Fuzzy Systems*, 1992. 2.5, 2.5.1, 3

[33] S. C. Lee and E. T. Lee, "Fuzzy neural networks," *Mathematical Biosciences*, vol. 23, no. 1-2, pp. 151–177, 1975. 2.5

[34] J. M. Keller and D. J. Hunt, "Incorporating fuzzy membership functions into the perceptron algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 693–699, 1985. 2.5

[35] T. L. HUNTSBERGER and P. AJJIMARANGSEE, "Parallel self-organizing feature maps for unsupervised pattern recognition," *International Journal Of General System*, vol. 16, no. 4, pp. 357–372, 1990. 2.5

[36] H. Ritter, T. Martinetz, K. Schulten, D. Barsky, M. Tesch, and R. Kates, *Neurprial computation and self-organizing maps: an introduction*. Addison-Wesley Reading, MA, 1992. 2.5.1

[37] W. contributors, "Flops — wikipedia, the free encyclopedia," 2018. [Online; accessed 16-January-2018]. 2.6

[38] B. A. Pimentel and R. M. de Souza, "A weighted multivariate fuzzy c-means method in interval-valued scientific production data," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3223–3236, 2014. 2.6

[39] E. Diday and F. Esposito, "An introduction to symbolic data analysis and the sodas software," *Intelligent Data Analysis*, vol. 7, no. 6, pp. 583–601, 2003. 2.6

[40] B. A. Pimentel and R. M. de Souza, "Multivariate fuzzy c-means algorithms with weighting," *Neurocomputing*, vol. 174, pp. 946–965, 2016. 2.6

[41] L. Himmelspach and S. Conrad, "A possibilistic multivariate fuzzy c-means clustering algorithm," in *International Conference on Scalable Uncertainty Management*, pp. 338–344, Springer, 2016. 2.6

[42] D. B. Maciel, G. J. Amaral, R. M. de Souza, and B. A. Pimentel, "Multivariate fuzzy k-modes algorithm," *Pattern Analysis and Applications*, vol. 20, no. 1, pp. 59–71, 2017. 2.6

[43] Z. Huang and M. K. Ng, "A fuzzy k-modes algorithm for clustering categorical data," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 4, pp. 446–452, 1999. 2.6

[44] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985. 4

[45] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984. 4

[46] W.-L. Hung and M.-S. Yang, "Similarity measures of intuitionistic fuzzy sets based on hausdorff distance," *Pattern Recognition Letters*, vol. 25, no. 14, pp. 1603–1611, 2004. 4.1

[47] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017. 4.2

[48] K. Inc, "Pima indians diabetes database," 2018. [Online; accessed 24-March-2018]. 4.2

Este volume foi tipografado em LaTeX na classe UFPEThesis (www.cin.ufpe.br/~paguso/ufpethesis).
Para detalhes sobre este documento, clique aqui.