



**UNIVERSIDADE ESTADUAL DE SANTA CRUZ
PRO-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO**

**PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL
EM CIÊNCIA E TECNOLOGIA**

THAISE SANTOS SILVA

**AVALIAÇÃO DA INFLUÊNCIA DE UMA MEDIDA DE SIMILARIDADE FUZZY NA
CLUSTERIZAÇÃO HIERÁRQUICA BASEADA NA CONEXIDADE DE GRAFOS FUZZY**

**ILHÉUS-BA
2016**

THAISE SANTOS SILVA

**AVALIAÇÃO DA INFLUÊNCIA DE UMA MEDIDA DE
SIMILARIDADE FUZZY NA CLUSTERIZAÇÃO
HIERÁRQUICA BASEADA NA CONEXIDADE DE GRAFOS
FUZZY**

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Computacional em Ciência e Tecnologia da Universidade Estadual de Santa Cruz, como parte das exigências para obtenção do título de Mestre em Modelagem Computacional em Ciência e Tecnologia.

Orientador: Prof. Dr. Eduardo Silva Palmeira

ILHÉUS-BA
2016

S586

Silva, Thaise Santos.

Avaliação da influência de uma medida de similaridade fuzzy na clusterização hierárquica baseada na conexidade de grafos fuzzy / Thaise Santos Silva. – Ilhéus, BA: UESC, 2016.

69 f. : il.

Orientador: Eduardo Silva Palmeira.

Dissertação (mestrado) – Universidade Estadual de Santa Cruz. Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia.

Inclui referências.

1. Teoria dos grafos. 2. Sistemas difusos. 3. Análise por agrupamento – Processamento de dados. 4. Topologia. I. Título.

CDD 511.5

THAISE SANTOS SILVA

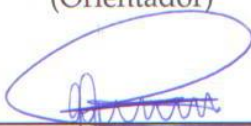
**AVALIAÇÃO DA INFLUÊNCIA DE UMA MEDIDA DE
SIMILARIDADE FUZZY NA CLUSTERIZAÇÃO
HIERÁRQUICA BASEADA NA CONEXIDADE DE GRAFOS
FUZZY**

Ilhéus-BA, 04/03/2016

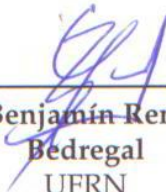
Comissão Examinadora



Prof. Dr. Eduardo Silva Palmeira
UESC
(Orientador)



Prof. Dr. Gildson Queiroz de Jesus
UESC



Prof. Dr. Benjamín René Callejas
Bedregal
UFRN

Aos meus pais e aos meus irmãos.

Agradecimentos

- Agradeço primeiramente à Deus, porque Dele, por Ele e para Ele são todas as coisas.
- Aos meus pais, meus irmãos e meu noivo pela cumplicidade e apoio.
- Ao meu orientador Eduardo Silva Palmeira pela paciência, confiança, conselhos e sugestões para este projeto e minha vida acadêmica.
- Aos professores, colegas e funcionários que, de certa forma, contribuíram para a realização desta pesquisa.
- À UESC pelo espaço, ao PPGMC pelo mestrado e à FAPESB pela concessão da bolsa.

*"Nada se obtém sem esforço; tudo se pode
conseguir com ele."*

(Ralph Waldo Emerson)

Avaliação da Influência de uma Medida de Similaridade Fuzzy na Clusterização Hierárquica Baseada na Conexidade de Grafos Fuzzy

Resumo

Clusterização é o processo no qual se deseja agrupar conjuntos de objetos em classes ou clusters considerando determinadas características dos objetos a partir de uma comparação de similaridade e/ou dissimilaridade. Dentre várias técnicas existentes na literatura, uma em particular é aquela que leva em consideração a conexidade de grafos fuzzy. Entretanto, apesar dos resultados satisfatórios, a noção de conexidade ainda é vista no sentido clássico, o que pode influenciar na eficácia desse processo. Na tentativa de melhorar os resultados, desenvolvemos um estudo para avaliar a influência de uma medida de similaridade fuzzy na clusterização baseada na conexidade de grafos fuzzy. Para isso, realizamos um estudo dos principais artigos, livros e outras literaturas que tratam sobre o tema e um estudo teórico sobre clusterização, grafos fuzzy e topologia em grafos fuzzy a fim de aprofundar os nossos conhecimentos nessas teorias que serviram como base para a realização da pesquisa. Em seguida, realizamos uma análise do algoritmo de clusterização hierárquica baseada na conexidade de grafo fuzzy (FHC) visando compreender sua dinâmica ou funcionamento. Posteriormente, avaliamos a influência de uma medida de similaridade fuzzy neste algoritmo e definimos um ajuste de centro fuzzy diante do processo de clusterização. Depois, aplicamos o algoritmo modificado para realizar agrupamento de cidades de acordo com o Índice de Desenvolvimento Humano Municipal (IDHM) e no processo de segmentação de imagens com o intuito de verificarmos se os resultados da clusterização utilizando uma medida de similaridade fuzzy são mais satisfatórios em comparação ao algoritmo FHC. Os resultados mostraram que o algoritmo FHC, apesar de ser um algoritmo de clusterização baseada na conexidade de grafos fuzzy, ele é aplicável a uma nuvem de dados e agrupa os objetos respeitando uma distância entre eles preestabelecida. Dessa forma, o agrupamento não leva em consideração o grau de similaridades entre os dados. Quando incluímos uma medida de similaridade fuzzy, os resultados foram mais satisfatórios, pois obtemos clusters em que os dados tinham um maior grau de similaridade.

Palavras-chave: Clusterização. Grafos fuzzy. Topologia em Grafos fuzzy.

Evaluation of Influence of a Fuzzy Similarity Measure on Hierarchical Clustering Based on Fuzzy Graphs Connectivity

Abstract

Clustering is a process in which desire to group a set of objects in grades or clusters considering certain characteristics of the objects from a comparative of similarity and/or dissimilarity. Among various existents techniques in literature, an special is one that takes in consideration the fuzzy graph connectivity. However, in spite of the satisfactory results, the connectivity idea is still seen in the classical sense, which can influence on the efficiency of this process. Trying to improve the results, we developed a study to evaluate the influence of a fuzzy similarity measure on clustering based on fuzzy graphs connectivity. For this, we made a study of the main paper, books and other literatures that talk about the subject and a theoretical study involving clustering, fuzzy graphs and topology on fuzzy graphs, in order to deepen our knowledge in this theory, being important as base for research execution. After that, we did an analysis of the hierarchical clustering algorithm based on fuzzy graph connectivity (FHC) aiming to understand its dynamic or operation. Posteriorly, we evaluated the influence of a fuzzy similarity measure on the same algorithm and we defined an adjustment of fuzzy center upon clustering process. Then, we applied the modified algorithm so that achieve city grouping according Municipal Human Development Index (MHDI) and on the images segmentation process, purposing verify if clustering results using a fuzzy similarity measure are more satisfactory than FHC algorithm. The results showed that FHC, despite being a clustering algorithm based on fuzzy graph connectivity, such is applicable to data cloud and group the objects respecting a preestablished distance between them. In this way, the grouping doesn't take in consideration the similarity degree among data. When we included a fuzzy similarity measure, the results were the most satisfactory because we obtained clusters whereupon the data had a grater similarity.

Keywords: Clustering. Fuzzy Graphs. Topology on Fuzzy Graphs.

Lista de figuras

Figura 1 – Exemplo de uma função característica	5
Figura 2 – Exemplo de uma função de pertinência	6
Figura 3 – Exemplo de grafo clássico	10
Figura 4 – Exemplos de subgrafos clássicos	10
Figura 5 – Exemplo de grafo fuzzy formal	11
Figura 6 – Exemplo de grafo fuzzy simplificado	12
Figura 7 – λ -cortes de um grafo fuzzy	13
Figura 8 – Exemplo de uma vizinhança	15
Figura 9 – Exemplo de vértices de conexão	16
Figura 10 – Exemplo de clusterização	18
Figura 11 – Etapas da clusterização	20
Figura 12 – Grafo fuzzy \tilde{G}	26
Figura 13 – Grafo fuzzy \tilde{G} no plano cartesiano	27
Figura 14 – Clusters iniciais do grafo \tilde{G}	27
Figura 15 – Subcluster SC_1 do grafo \tilde{G}' usando distância clássica	28
Figura 16 – Subcluster SC_2 do grafo \tilde{G}' usando distância clássica	28
Figura 17 – Subcluster SC_3 do grafo \tilde{G}' usando distância clássica	29
Figura 18 – Grafo fuzzy de subclusters \tilde{G}' usando distância clássica	29
Figura 19 – λ -corte do grafo \tilde{G}' usando distância clássica	30
Figura 20 – Recortes da implementação do FHC	31
Figura 21 – Método updateCenter	31
Figura 22 – Função calculatelinks	32
Figura 23 – Subcluster SC_1 do grafo \tilde{G}' usando distância fuzzy	36
Figura 24 – Subcluster SC_2 do grafo \tilde{G}' usando distância fuzzy	39
Figura 25 – Grafo fuzzy de subclusters \tilde{G}' usando distância fuzzy	40
Figura 26 – Cálculo do IDHM atualmente	43
Figura 27 – Mesorregiões da Bahia	44
Figura 28 – Resultados para $\lambda = 0.2$	47
Figura 29 – Resultados para $\lambda = 0.5$	47
Figura 30 – Representação de uma imagem no plano cartesiano	49
Figura 31 – Representação espacial dos pixels da imagem	50
Figura 32 – Representação dos pixels da imagem no intervalo $[0, 1]$	51
Figura 33 – Resultado da segmentação da imagem com o FHC	51
Figura 34 – Resultado da segmentação da imagem com o FHC-M	51
Figura 35 – Resultados da clusterização	53

Lista de tabelas

Tabela 1 – Operações entre conjuntos fuzzy	8
Tabela 2 – Pesos das arestas do grafo da Figura 12	34
Tabela 3 – Municípios da microrregião de Porto Seguro	45
Tabela 4 – Municípios da microrregião de Valença	45
Tabela 5 – Municípios da microrregião de Itabuna-Ilhéus	46
Tabela 6 – Municípios da mesorregião sul da Bahia	56

Lista de abreviaturas e siglas

DCET	Departamento de Ciências Exatas e Tecnológicas
EUA	Estados Unidos da América
IDH	Índice de Desenvolvimento Humano
IDHM	Índice de Desenvolvimento Humano Municipal
Ipea	Instituto de Pesquisa Econômica Aplicada
FAPESB	Fundação de Amparo à Pesquisa do Estado da Bahia
FHC	Algoritmo de Clusterização Hierárquica Baseada na Conexidade de Grafos <i>fuzzy</i>
FJP	Fundação João Pinheiro
PNUD	Programa das Nações Unidas para o Desenvolvimento
PPGMC	Programa de Pós-Graduação em Modelagem Computacional em Ciência e Tecnologia
UESC	Universidade Estadual de Santa Cruz

Sumário

1 – INTRODUÇÃO	1
1.1 Organização da Dissertação	2
2 – FUNDAMENTAÇÃO TEÓRICA	3
2.1 Teoria de Conjunto fuzzy e Aplicações	3
2.1.1 Histórico	3
2.1.2 Conjuntos fuzzy	5
2.2 Grafos	8
2.2.1 Grafos Clássicos	9
2.2.2 Grafos fuzzy	11
2.3 Clusterização	17
2.3.1 Métodos de Clusterização	20
2.3.1.1 Métodos Hierárquicos	21
2.3.1.2 Métodos Baseados em Lógica fuzzy	22
2.3.1.3 Métodos Baseados em Grafos	23
2.3.2 Medidas de Avaliação de Clusters	24
2.3.2.1 Índices Internos	24
2.3.2.2 Índices Externos	24
3 – MATERIAIS E MÉTODOS	25
3.1 Análise do Algoritmo FHC	25
3.1.1 O Algoritmo FHC	25
3.1.2 Implementação do Algoritmo FHC	30
3.1.2.1 Explicação do Código	30
3.2 FHC e Distância fuzzy	32
3.2.1 Ajuste de N_{centro} fuzzy	33
3.3 Avaliação Experimental	41
3.3.1 Agrupamento de Cidades de Acordo com IDHM	42
3.3.2 Segmentação de Imagens	49
4 – DISCUSSÃO DOS RESULTADOS	52
4.1 Resultados Iniciais	52
4.2 Resultados da Avaliação Experimental	55
5 – CONCLUSÃO	60
5.1 Trabalhos Futuros	61

6 – REFERÊNCIAS	62
---------------------------	----

1 INTRODUÇÃO

A Matemática é uma ciência construída ao longo de muitos anos em que diversos conceitos, noções e resultados se mantêm válidos e presentes no cotidiano. Com efeito, as teorias matemáticas que trazem, em suas essências, as organizações que revelam tais conceitos ou noções, encontram aplicações em várias áreas de conhecimentos como a: Física, Engenharia, Biologia, Química e Ciência Computacional. Dentre estas teorias, estão no centro deste trabalho a de conjuntos fuzzy, visto que, a mesma permite a modelagem dos fenômenos que ansiamos estudar, a saber, clusterização de grafos.

A clusterização é o processo no qual se deseja agrupar dados em classes e os grupos resultantes são chamados de clusters. O agrupamento é geralmente baseado em alguma medida de similaridade definida sobre os dados. Os algoritmos utilizados na clusterização de dados podem ser classificados de acordo com a abordagem utilizada na definição dos clusters. Uma dessas abordagens é a que se baseia em grafos, que são estruturas formadas por um conjunto de vértices e um conjunto de arestas. A clusterização em grafos é a tarefa de agrupar os vértices do grafo em clusters levando em consideração a estrutura das arestas, que são conexões entre os pares de vértices dos grafos, de uma maneira que os vértices dentro de um determinado cluster tenham um maior grau de similaridade.

Em abordagens tradicionais de clusterização cada objeto pertence, ao final, a um e somente um cluster. Entretanto, os problemas reais geralmente são associados a uma série de parâmetros que não são naturalmente precisos. Além disso, podem existir casos em que nem a estrutura do grafo é precisa. Isso nos faz recorrer a um modelagem sob o ponto de vista da teoria fuzzy. Dentro da vasta aplicabilidade desta teoria, destacamos a sua importância no processo de clusterização hierárquica baseada na conexidade de grafos fuzzy. Essa classe especial de clusterização mostrou melhores resultados em relação às determinadas classes até então exibidas pelos estudiosos nessa área (DONG, et al., 2006). Entretanto, apesar dos resultados satisfatórios apresentados, a noção de conexidade ainda é vista do aspecto clássico, o que pode influenciar na eficácia nos resultados desse processo.

Na tentativa de gerar melhores resultados, propomos, na dissertação, avaliar a influência de uma medida de similaridade fuzzy no processo de clusterização. Dessa forma, o objetivo geral da pesquisa é desenvolver um estudo da influência de uma medida de similaridade fuzzy no algoritmo de clusterização hierárquica baseada na conexidade de grafos fuzzy (FHC) proposto por Dong et. al (2006). E como específicos:

1. Realizar uma análise a fim de compreender o comportamento do algoritmo FHC;

2. Implementar o algoritmo FHC;
3. Avaliar a influência de uma medida de similaridade fuzzy neste algoritmo;
4. Testar computacionalmente e analiticamente o algoritmo FHC usando a medida de similaridade fuzzy afim de verificar a sua eficiência sob esse ponto de vista.

Com efeito, nos questionamos: até que ponto uma medida de similaridade fuzzy influencia na classificação dos clusters no processo de clusterização hierárquica de grafos fuzzy? Provisoriamente, acreditamos que esta visão topológica fuzzy no conceito de conexidade pode emanar na diminuição dos ruídos, que são os objetos isolados, ou gerar clusters em que os vértices possuam um maior grau de similaridades. A seguir, apresentamos como estará organizada a dissertação.

1.1 Organização da Dissertação

No capítulo 2 abordamos a fundamentação teórica que serviu como base para a realização da pesquisa. Fazemos uma breve explanação sobre a teoria de conjuntos fuzzy e suas aplicações, teoria de grafos clássicos e fuzzy, alguns conceitos topológicos de grafos fuzzy, clusterização de grafos, alguns algoritmos de clusterização encontrados na literatura e algumas aplicações.

No capítulo 3 exibimos os procedimentos metodológicos adotados na pesquisa. Primeiramente fizemos uma análise do algoritmo FHC, depois apresentamos uma proposta de modificação desse algoritmo utilizando uma medida de similaridade fuzzy seguida de uma definição de ajuste de centro fuzzy. Posteriormente, aplicamos o algoritmo FHC e o algoritmo modificado para realizar agrupamento de cidades e no processo de segmentação de imagens.

No capítulo 4 apresentamos uma discussão dos resultados obtidos durante o desenvolvimento da pesquisa e na avaliação experimental. E, no capítulo 5, apresentamos uma visão geral da pesquisa, as contribuições e as conclusões decorrentes da realização da mesma. Além disso, destacamos algumas sugestões de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção apresentaremos o referencial teórico utilizado como base para a realização dessa pesquisa.

2.1 Teoria de Conjunto fuzzy e Aplicações

Para descrever alguns fenômenos recorreremos a graus que representam categorias, intensidades ou forças. Esse é o caso, por exemplo, dos conceitos de alto, baixo, perto, longe, cheio, vazio. Nesse sentido, tomamos, por exemplo, o conjunto das pessoas altas. Na abordagem clássica (crisp), predeterminamos um valor para que a partir dele uma pessoa possa ser considerada alta ou não. Na abordagem fuzzy, é dada de maneira que a pessoa seja considerada alta com mais ou menos intensidade. Isso significa que, quanto menor for a medida da altura do indivíduo, menor será seu grau de pertinência a esta classe. Dessa forma, podemos afirmar que os indivíduos pertencem à classe das pessoas altas, com mais ou menos intensidade. Foi a partir de desafios como esse, que surgiu a teoria de conjuntos fuzzy. A palavra “fuzzy” é de origem inglesa e significa incerto, vago, impreciso, subjetivo, nebuloso ou difuso.

2.1.1 Histórico

Os princípios de conjuntos fuzzy foram introduzidos primeiramente por Jan Lukasiewicz em 1920, que desenvolveu conjuntos com grau de pertinência que combinados aos conceitos da lógica clássica, desenvolvida por Aristóteles, deu embasamento suficiente para que Lofti Asker Zadeh em 1965 chegasse a ser o primeiro autor de uma publicação sobre lógica fuzzy (DUBOIS e PRADES, 2000).

Nascido em 04 de fevereiro de 1921 na República Soviética do Azerbaijão, Zadeh foi graduado e recebeu o título de Bacharel de Ciência em Engenharia Eletrotécnica. Em 1944 mudou-se para os EUA e simplificou seu nome de Lotfi Aliaskerzadeh para Lotfi Asker Zadeh. Neste país, deu continuidade aos estudos de pós-graduação e em 1946 recebeu o título de mestre em Engenharia Eletrotécnica pelo Instituto de Tecnologia de Massachusetts. Posteriormente, em 1949, ingressou no corpo docente da Universidade de Columbia, em Nova York, como instrutor em Engenharia Elétrica, onde concentrou sua atenção em sistemas lineares e teoria dos autômatos e obteve o Ph.D.

Em 1959, Zadeh deixou a Universidade de Columbia para integrar o corpo docente do Departamento de Engenharia Elétrica da Universidade da Califórnia em Berkeley. Foi nomeado presidente do departamento em 1963 e durante seu mandato de

cinco anos, o nome do departamento foi alterado para Engenharia Elétrica e Ciência da Computação. Em 1965, publicou seu trabalho sobre a lógica fuzzy, em que detalhou a matemática da teoria dos conjuntos difusos e em 1973, propôs sua teoria da lógica fuzzy.

Tudo que é novo causa medo, receito, resistência. Não foi diferente com a lógica fuzzy. Essa teoria enfrentou forte oposição por parte da comunidade científica, em particular, dos estatísticos norte americanos. Contudo, apesar de toda a resistência, muitos cientistas e pesquisadores enxergaram as inúmeras possibilidades de aplicações que essa teoria oferecia e muitos trabalhos surgiram em todo o mundo, principalmente no Japão onde foi formado o primeiro grupo de pesquisa em sistemas fuzzy (TAMIR, RISHE e KANDEL, 2015).

Nas décadas de 60 e 70 novos conceitos foram introduzidos nessa teoria como relações fuzzy, variáveis linguísticas, álgebra com números fuzzy, sistemas topológicos, clusterização fuzzy, entre outros. Em 1974, no Reino Unido, Ebrahim Mamdani criou o primeiro controlador fuzzy, dando um passo para que outros pesquisadores aplicasse a lógica fuzzy para sistemas em engenharia.

Na Dinamarca, em 1976, ocorreu a primeira aplicação da lógica fuzzy no setor industrial. A modelagem desenvolvida pelo Circle Cement e SIRA consistia em incorporar os conhecimentos e experiências dos operários para controlar os fornos das fábricas. O primeiro sistema especialista fuzzy foi desenvolvido por Hans Zimmermann em 1977 e Didie Dubois aplicou conjuntos fuzzy no estudo de condições de tráfego (YEN e LANGARI, 1999).

Segundo Reznik (1997), no laboratório Bell, Estados Unidos, em 1985 foi criado o primeiro chip fuzzy e em 1987 inaugurado o primeiro trem controlado por lógica fuzzy no Japão. Além disso, em 1987 a Yamaha desenvolveu seu helicóptero não tripulado supervisionado totalmente por um controlador fuzzy. Em 1988 o primeiro sistema de comércio financeiro fuzzy começou a ser operado e em 1990 a teoria tornou-se popular com a primeira máquina de lavar roupas fuzzy da marca Matsushita Electric Industrial. Atualmente é possível encontrar no Japão vários eletrodomésticos que utilizam sistemas baseados em controles fuzzy e existem hoje várias empresas que possuem laboratórios de pesquisa em logica fuzzy para o desenvolvimento de seus produtos.

Nesse contexto, notamos que a teoria de conjuntos fuzzy é muito importante para modelar problemas reais que envolvam principalmente as imprecisões e não-linearidades. Esse campo tem resultados nas mais variadas aplicações, sendo capaz de capturar informações imprecisas, descritas em linguagem natural, e transformá-las para um formato numérico, afim de efetuar um raciocínio aproximado, com proposições imprecisas, através de conjuntos fuzzy. A lógica fundamentada em conjuntos fuzzy direciona o desenvolvimento de modelos semelhantes às decisões humanas,

onde a representação de conceitos vagos e imprecisos pode ser incorporada a modelos sofisticados.

Diante disso, nos últimos anos ela vem sendo aplicada em diversas áreas que lidam com imprecisões, como Engenharia (YAN e MA, 2013; KAUR e KAUR, 2012; LI, 2011; WU e ADOKO, 2011; FREITAG, GRAF e KALISKE, 2011; LAI, 2010; PENG, 2010; CHANG, 2010), Química (MEZEY, 2014; BRUYLANTS, BARTIK, REISSE, 2011; BRUYLANTS, BARTIK e REISSE, 2010; GULER et al., 2012; FONSECA e KNAPP, 2000), Medicina (CHEN et al., 2013; PAL et al., 2012; KIM e KIM, 2012; MUTHUKARUPPANA e ER, 2012; ANOOJ, 2012; WAGHOLIKAR et al., 2012; ORTEGA et al., 2001; MASSAD et al., 2001; ORTEGA et al., 2000), Biologia (LIU et al., 2011; KANNANA et al., 2010; HAMED, AHSON e PARVEEN, 2010; LIN e CHEN, 2010; WU et al., 2010; LIBICH et al., 2010; BARROS et al., 2000), Economia (XIN, 2012; TARRAZO e GUTIERREZ, 2000) e Educação (XAVIER, 2014; LEA, 2011; KHAN, AMIN e REHMAN, 2011; GOKMEN et al., 2010; SRIPAN e SUKSAWAT, 2010).

2.1.2 Conjuntos fuzzy

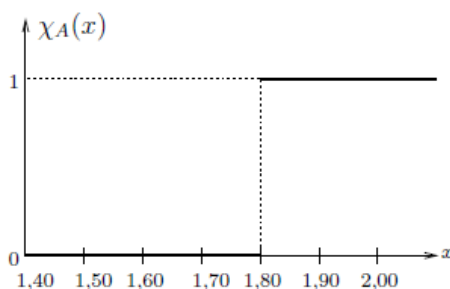
Sabemos que existe uma forte ligação entre a lógica aristotélica e a teoria de conjuntos clássicos. Neste sentido, trabalhamos com a possibilidade de um objeto pertencer ou não a um determinado conjunto. Dessa forma, qualquer conjunto clássico pode ser expresso por uma função característica, cuja definição é dada a seguir.

Definição 1 *Seja U um universo de discurso e A um subconjunto de U . A função característica de A é dada por*

$$\chi_A(x) = \begin{cases} 1 & \text{se } x \in A; \\ 0 & \text{se } x \notin A. \end{cases}$$

Nesse contexto, notamos que χ_A é uma função com domínio U e imagem contida no conjunto $\{0, 1\}$. Além disso, $\chi_A(x) = 1$ indica que o elemento x está em A , enquanto $\chi_A(x) = 0$ indica que x não é elemento de A .

Figura 1 – Exemplo de uma função característica



Fonte: Takahashi (2004) adaptada pelos autores

Na Figura 1 temos um exemplo de um gráfico de uma função característica de A . Definimos A como o conjunto das pessoas altas da forma $A = \{x \in R/x \geq 1,80\}$. Assim, por exemplo, uma pessoa com $1,79m$ não é considerada alta, enquanto uma pessoa com $1,81m$ é seguramente alta.

Como na teoria de conjuntos clássicos, existe também uma forte ligação entre a lógica *fuzzy* e a teoria de conjuntos *fuzzy*. A extensão sugerida por Zadeh está na possibilidade de um determinado elemento poder pertencer a um conjunto com um valor chamado grau de pertinência. Zadeh (1965) apud Bede (2013), apresenta a formalização matemática de um conjunto fuzzy como a definição a seguir.

Definição 2 *Seja U um universo de discurso. Um subconjunto fuzzy A de U é caracterizado por uma função*

$$\mu_A : U \longrightarrow [0, 1],$$

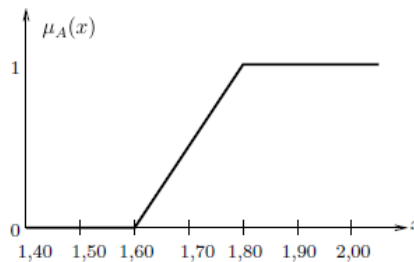
chamada função de pertinência do subconjunto fuzzy A .

O valor $\mu_A(x)$ é o grau de pertinência do elemento x de U no conjunto fuzzy A . Nesse sentido, um elemento poderá pertencer a um conjunto com grau de pertinência que varia no intervalo $[0, 1]$, onde o valor 0 indica uma completa exclusão, o valor 1 representa completa pertinência e os valores intermediários deste intervalo representam graus intermediários de pertinência do objeto em relação ao conjunto. Da mesma forma, se considerarmos o conjunto A das pessoas altas, um tipo de função de pertinência que caracteriza esse conjunto poderia ser expresso por:

$$\mu_A(x) = \begin{cases} 0 & \text{se } x < 1.6; \\ 5x - 8 & \text{se } 1.6 \leq x \leq 1.8 \\ 1 & \text{se } x > 1.8. \end{cases}$$

Nesse sentido, uma pessoa com $1.65m$ é 0.25 alta, enquanto uma pessoa com 1.79 é 0.95 alta, pois $\mu_A(1.65) = 0.25$ e $\mu_A(1.79) = 0.95$. A representação gráfica dessa função de pertinência pode ser observada na Figura 2.

Figura 2 – Exemplo de uma função de pertinência



Como na teoria de conjuntos clássicos, na teoria dos conjuntos fuzzy existem operações típicas como contingência, união, intersecção e complementar. Antes de defini-las, apresentaremos os conceitos de α -corte (ou α -nível), suporte e núcleo de um subconjunto A de U . Essas definições têm papéis importantes na inter-relação entre as teorias clássica e fuzzy, pois oferecem uma forma de relacionar conjuntos fuzzy com conjuntos clássicos. Tais definições, assim como as operações típicas, foram baseadas em Bede (2013) e Syropoulos (2013).

Definição 3 (α -corte de A) *Dado um conjunto fuzzy A de U , ao conjunto $A_\alpha = \{x \in U, \mu_A(x) \geq \alpha\}$, $\alpha \in [0, 1]$, damos o nome de α -corte de A .*

Definição 4 (Suporte de A) *Dado um conjunto fuzzy A de U , definimos seu suporte por $\text{supp}(A) = \{x \in U; \mu_A(x) > 0\}$.*

Definição 5 (Núcleo de A) *Dado um conjunto fuzzy A de U , definimos seu núcleo como $\text{nuc}(A) = \{x \in U; \mu_A(x) = 1\}$.*

Se o conjunto $\text{nuc}(A) \neq \emptyset$, dizemos que A é normalizado. Dessa forma, se considerarmos o conjunto A das pessoas altas novamente então o $\text{nuc}(A) = \{x \in R; x \geq 1.80\}$ e A é normalizado. Além disso, $\text{supp}(A) = \{x \in R; x > 1.60\}$ e se $\alpha = 0.25$ então $A_\alpha = \{x \in R; x \geq 1.65\}$. Assim, notamos que os conjuntos α -corte de A , $\text{supp}(A)$ e $\text{nuc}(A)$ são conjuntos clássicos.

Definição 6 (Contingência) *Seja U o universo de discurso. Dizemos que um conjunto fuzzy A está contido ou é igual ao conjunto fuzzy B (representação $A \subseteq B$) se para todo $x \in U$, tem-se $\mu_A(x) \leq \mu_B(x)$.*

Definição 7 (União) *Dados dois conjuntos fuzzy A e B de U , a união entre eles é uma função de pertinência dada por*

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}.$$

Definição 8 (Intersecção) *Dados dois conjuntos fuzzy A e B de U , a intersecção entre eles é uma função de pertinência dada por*

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}.$$

Definição 9 (Complementar) *Dado o conjunto fuzzy A do universo de discurso U . O seu complementar A' é uma função de pertinência dada por*

$$\mu_{A'}(x) = 1 - \mu_A(x).$$

Para compreendermos melhor as definições apresentadas acima, vamos supor que o conjunto universo U seja composto pelas cidades identificadas pelos números 1,2,3,4 e 5. Sejam A , B e C os subconjuntos fuzzy, em que $\mu_A(x)$, $\mu_B(x)$ e $\mu_C(x)$ é o grau de pertinência de uma cidade $x \in U$ nos conjuntos A , B e C , respectivamente, como apresentados na Tabela 1.

Tabela 1 – Operações entre conjuntos fuzzy

Cidade x	$\mu_A(x)$	$\mu_B(x)$	$\mu_C(x)$	$\mu_{A \cup B}(x)$	$\mu_{A \cap B}(x)$	$\mu_{C'}(x)$
1	1.0	0.8	1.0	1.0	0.8	0.0
2	0.3	0.3	0.3	0.3	0.3	0.7
3	0.5	0.2	0.5	0.5	0.2	0.5
4	0.8	0.7	0.8	0.8	0.7	0.2
5	0.9	0.5	0.9	0.9	0.5	0.1

Fonte: Elaborada pelos autores

Essa Tabela ilustra as operações de união, interseção e complementar. Além disso, podemos notar que o conjunto B está contido no conjunto A e também no conjunto C , pois para todo $x \in U$, tem-se $\mu_B(x) \leq \mu_A(x)$ e $\mu_B(x) \leq \mu_C(x)$.

Um conceito muito importante dentro da teoria de conjuntos fuzzy é o de relação fuzzy. Na teoria de conjuntos clássicos uma relação indica se há ou não alguma associação entre dois objetos, enquanto que uma relação fuzzy, além de indicar se há ou não tal associação, indica também o grau dessa relação.

Definição 10 (Relação fuzzy) *Dados A e B dois conjuntos fuzzy. Uma relação fuzzy R entre os conjuntos A e B é uma função de pertinência*

$$\mu_R : A \times B \longrightarrow [0, 1].$$

Um exemplo evidente de relação fuzzy pode ser observado na representação de um grafo fuzzy. O conjunto dos vértices pode ser visto como um conjunto fuzzy e o conjunto das arestas como uma relação fuzzy. Mais adiante apresentamos a definição formal de grafo fuzzy, antes faremos uma breve explanação sobre a teoria de grafos clássicos, conforme a seguir.

2.2 Grafos

De acordo com Ziniani (2004) muitos problemas práticos podem ser resolvidos por meio de uma modelagem em grafos. Por exemplo, um mapa pode ser visto como grafo: os vértices representariam as cidades e as estradas, as aresta. Dessa forma, poderíamos determinar o caminho mais curto entre duas cidades por meio dos pesos das arestas.

Outro exemplo da utilização da teoria de grafos é a estrutura de links de um site de enciclopédia: os vértices são os artigos do site e existe uma aresta do artigo A para o artigo B se e somente se A contém um link para B. As redes de computadores também podem ser vistas como uma aplicação de grafos: cada terminal seria representado por um vértice, o cabo de rede pelas arestas e o custo seria associado ao número de máquinas que a comunicação atravessa entre os vértices.

Nesse sentido, notamos que a teoria de grafo é um ramo da matemática que estuda as relações entre os objetos de um determinado conjunto e é muito útil na representação de problemas da vida real, em vários campos profissionais, pois permitem determinar caminhos, distâncias e conexões entre objetos. Nesta seção, introduziremos alguns conceitos sobre a teoria de grafos clássicos e a teoria de grafos fuzzy, um ramo da teoria de conjuntos fuzzy.

2.2.1 Grafos Clássicos

Como citamos acima, muitos problemas da vida real podem ser resolvidos por meio de uma modelagem em grafos. Diversos autores como Mendoza et al. (2015), Anderson et al. (2014), Shao, Wang e Li (2013), Elhadi, Maarof e Osman (2012), Petersen et al. (2011), Zhi et al. (2011), Hong, Oguntebi e Olukotun (2011), Couprie (2010), Bunke e Riesen (2011), Ellis-Monaghan e Merino (2010) realizaram estudos mostrando essas aplicações.

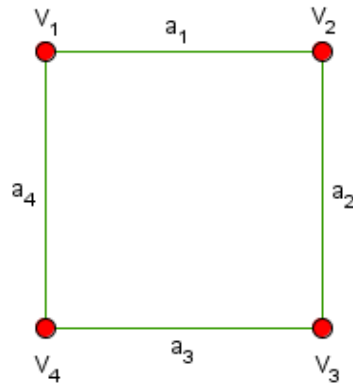
Neste seção, apresentamos os conceitos básicos da teoria de grafos clássicos que foram fundamentados em Ruohonen (2013), Chartrand e Zhang (2012) e Balakrishnan e Ranganathan (2012) e são necessários para a compreensão da próxima seção.

Definição 11 (Grafo clássico) *Um grafo clássico (ou crisp) $G(V, A)$ é definido pelo par de conjuntos V e A , onde:*

- V - representa os vértices do grafo;
- A - representa as arestas do grafo e $A \subseteq V \times V$.

Na Figura 3, temos um exemplo de grafo clássico $G(V, A)$, onde o conjunto V é formado pelos vértices $V_i, i = 1, \dots, 4$, e o conjunto A é formado pelas arestas $a_i, i = 1, \dots, 4$.

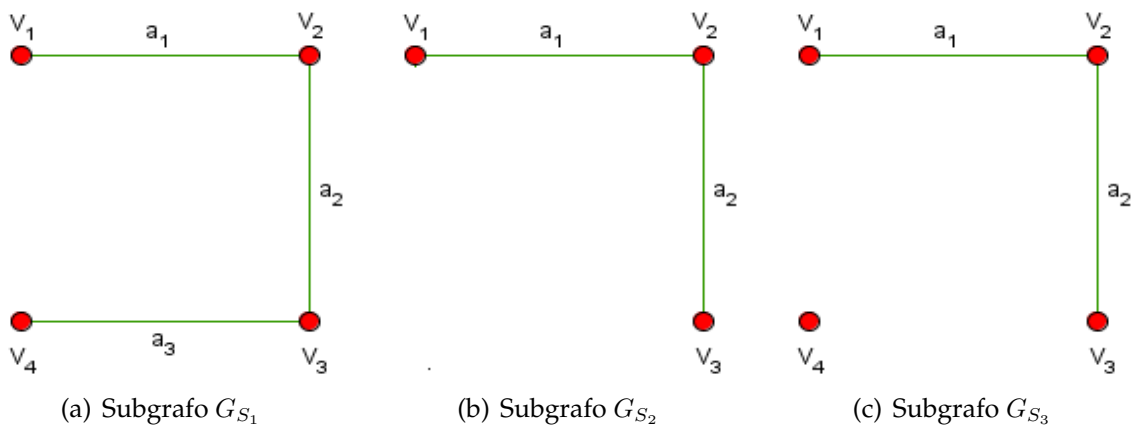
Figura 3 – Exemplo de grafo clássico



Fonte: Elaborada pelos autores

Definição 12 (Subgrafo clássico) Um grafo $G_S(V_S, A_S)$ é dito subgrafo do grafo $G(V, A)$ quando $V_S \subseteq V$ e $A_S \subseteq A$.

Figura 4 – Exemplos de subgrafos clássicos



Fonte: Elaboradas pelos autores

Na Figura acima, G_{S_1} , G_{S_2} e G_{S_3} são subgrafos do grafo G da Figura 3.

Definição 13 (Caminho em um grafo clássico) Um caminho de um vértice V_1 a um vértice V_n em um grafo $G(V, A)$ é uma sequência de vértices distintos V_1, V_2, \dots, V_n tal que $(V_i, V_{i+1}) \in A$, para todo $i = 1, \dots, n$.

Definição 14 (Grafo conexo) Seja G um grafo. Dois vértices V_1 e V_2 de G estão conectados se, e somente se, existe um caminho que liga V_1 a V_2 . Um grafo G é conexo se, e somente se, qualquer dois vértices em G estão conectados.

Definição 15 (Componente conexa) Um grafo G_C é uma componente conexa de um grafo G se e somente se ele é um subgrafo maximal conexo de G .

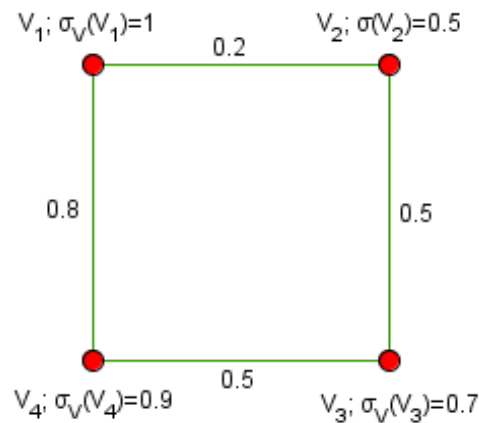
Na Figura 4, os grafos G_{S_1} e G_{S_2} são conexos, enquanto G_{S_3} não é conexo, pois o vértice V_4 não está conectado aos outros vértices. Além disso, o grafo G_{S_1} tem apenas uma componente conexa que é o próprio grafo. O mesmo acontece com G_{S_2} . Entretanto, o grafo G_{S_3} tem duas componentes conexas: a primeira componente contém os vértices V_1, V_2 e V_3 e a segunda componente contém apenas o vértice V_4 .

2.2.2 Grafos fuzzy

De acordo com Tom e Sunitha (2015), a teoria de grafos fuzzy foi desenvolvida por Rosenfeld no ano de 1975. Durante o mesmo tempo, Yeh e Bang também introduziram vários conceitos de conexidade em grafos fuzzy. Rosenfeld (1975), apresenta uma definição formal de grafo fuzzy, conforme a enunciamos abaixo.

Definição 16 (Grafo fuzzy formal) Um grafo fuzzy $\tilde{G} < V, R >$ é relação $R \subseteq V \times V$ sobre um conjunto V , onde V é o conjunto dos vértices e R o conjunto das arestas. O vértice $V_i \in V$ tem peso $\sigma_V(V_i) \in [0, 1]$, a aresta $(V_i, V_j) \in V \times V$ tem peso $\mu_R(V_i, V_j) \in [0; 1]$ e $\mu_R(V_i, V_j) \leq \min\{\sigma_V(V_i), \sigma_V(V_j)\}$.

Figura 5 – Exemplo de grafo fuzzy formal



Fonte: Elaborada pelos autores

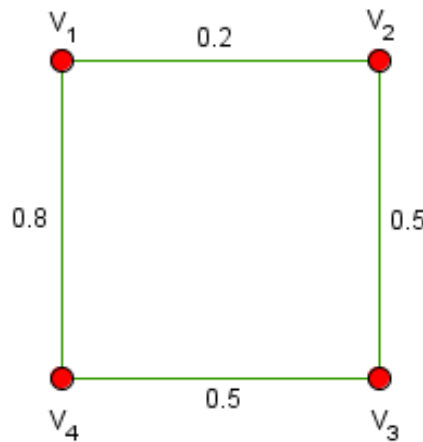
Na Figura acima, temos um exemplo de grafo fuzzy formal. Os vértices V_1, V_2, V_3 e V_4 têm grau 1, 0.5, 0.7 e 0.9, respectivamente. As arestas (V_1, V_2) , (V_2, V_3) , (V_3, V_4) e (V_4, V_1) tem peso $\mu_R(V_1, V_2) = 0.2$, $\mu_R(V_2, V_3) = 0.5$, $\mu_R(V_3, V_4) = 0.5$ e $\mu_R(V_4, V_1) = 0.8$, respectivamente.

Dessa forma, diferentemente de grafos crisps, para um grafo ser considerado fuzzy, cada vértice tem um grau de pertinência e cada aresta tem um peso que será um valor do intervalo $[0, 1]$. Esse diferencial fez com que muitos autores como Lara et al. (2015), Sahoo e Pal (2015), Wangab, Wanga e Penga (2015), González-Arangüenaa, Manuela e Pozob (2014), Samanta e Pal (2014), Gross, Nagi e Sambhoos (2014), Yina et al. (2014) usassem em seus estudos esse tipo de abordagem para modelar problemas no cotidiano.

Por questão de simplicidade, consideraremos no trabalho apenas os grafos não direcionados, isto é, a relação fuzzy é simétrica, ou seja, $\mu_R(V_i, V_j) = \mu_R(V_j, V_i)$. Não consideraremos as arestas da forma (V_i, V_i) , consequentemente, tomaremos o conjunto dos vértices como um conjunto clássico. Nesse contexto, apresentamos a seguir, uma definição de grafo fuzzy simplificado.

Definição 17 (Grafo fuzzy simplificado) *Um grafo $\tilde{G} < V, R >$ é fuzzy se V é o conjunto dos vértices, R é uma relação fuzzy simétrica e μ_R é uma função que associa cada aresta a um valor no intervalo $[0, 1]$.*

Figura 6 – Exemplo de grafo fuzzy simplificado



Fonte: Tom e Sunitha (2015) adaptada pelos autores

As definições de grafo conexo e componente conexa apresentadas na seção de grafos clássicos também valem para grafos fuzzy. Outras definições importantes para grafos fuzzy apresentaremos a seguir e foram fundamentadas em Rosenfeld (1975) e Dong, et al. (2006).

Definição 18 (Caminho em um grafo fuzzy) *Um caminho L em um grafo fuzzy $\tilde{G} < V, R >$ é uma sequência de vértices distintos V_1, V_2, \dots, V_n tal que $\mu_R(V_i, V_{i+1}) > 0, i = 1, \dots, n$.*

Definição 19 (Intensidade de caminho) *se L é um caminho de vértices V_1, V_2, \dots, V_n então a intensidade do caminho, $S(L)$, é o $\min\{\mu_R(V_i, V_{i+1})\}, i = 1, \dots, n$.*

Na definição acima a intensidade de um caminho é definida como o menor peso das arestas do caminho que liga dois vértices distintos. Se existem n caminhos que ligam dois vértices distintos V_i e V_j então a intensidade conexa $S(V_i, V_j)$ é o máximo das intensidades dos n caminhos que ligam os vértices V_i e V_j .

Para compreendermos melhor as noções de intensidade de um caminho e de intensidade conexa, vamos considerar o grafo fuzzy da Figura 6. Tomamos, por exemplo, os vértices V_1 e V_2 . Existem dois caminhos que ligam esses dois vértices, a saber: a própria aresta que conecta esses dois vértice e que tem peso 0.2 ($V_1 \rightarrow V_2$) e o caminho que passa pelos pontos V_4 e V_3 ($V_1 \rightarrow V_4 \rightarrow V_3 \rightarrow V_2$). Assim, temos:

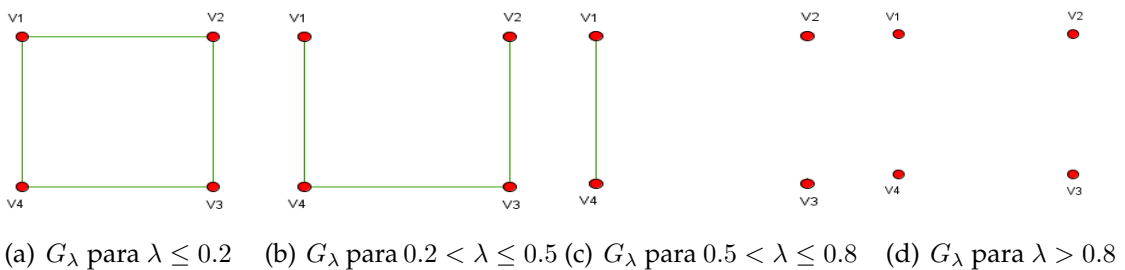
- $S(L_1) = \min\{0.2\} = 0.2$;
- $S(L_2) = \min\{0.8; 0.5; 0.5\} = 0.5$;
- $S(V_1, V_2) = \max\{S(L_1), S(L_2)\} = \{0.2; 0.5\} = 0.5$.

Portanto, a intensidade do caminho L_1 é 0.2, a do caminhos L_2 é 0.5 e a intensidade conexa de V_1 e V_2 é 0.5.

Definição 20 (λ -corte de um grafo fuzzy) *Seja $\tilde{G} = \langle V, R \rangle$ um grafo fuzzy e μ_R uma função que associa cada aresta a um valor no intervalo $[0, 1]$. $E_\lambda = \{(V_i, V_j) / \mu_R(V_i, V_j) > \lambda, V_i, V_j \in V\}$, com $0 \leq \lambda \leq 1$. $G_\lambda = \langle V, E_\lambda \rangle$ é chamado de λ -corte de \tilde{G} .*

No grafo da Figura 6, para diferentes λ teremos diferentes G_λ . Por exemplo, para qualquer $\lambda \leq 0.2$, o G_λ será conforme a Figura 7(a). Para $0.2 < \lambda \leq 0.5$, $0.5 < \lambda \leq 0.8$ e $\lambda > 0.8$ o grafo G_λ será conforme as Figuras 7(b), 7(c) e 7(d), respectivamente.

Figura 7 – λ -cortes de um grafo fuzzy



Fonte: Elaboradas pelos autores

Nas próximas definições, apresentaremos alguns conceitos topológicos de grafos fuzzy fundamentados em Dong, et al. (2006) e Rosenfeld (1975), Rosenfeld (1975) apud Tom e Sunitha (2015).

Definição 21 (Distância euclidiana entre dois vértices) *Sejam V_i e V_j quaisquer dois vértices de um grafo fuzzy $\tilde{G} < V, R >$. A distância euclidiana (ou distância clássica) entre esses dois vértices é dada por:*

$$d(V_i, V_j) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + \dots + (z_1 - z_2)^2} \quad (1)$$

onde x_1, y_1, \dots, z_1 são as coordenadas do vértice V_i e x_2, y_2, \dots, z_2 são as coordenadas do vértice V_j .

Definição 22 (Distância fuzzy entre dois vértices) *Se L é um caminho constituído pelos vértices V_1, V_2, \dots, V_n em um grafo fuzzy $\tilde{G} < V, R >$, então o comprimento de L é definido por:*

$$\rho(L) = \sum_{k=1}^n \mu_R(V_i, V_{i+1})^{-1} = \sum_{k=1}^n \frac{1}{\mu_R(V_i, V_{i+1})}. \quad (2)$$

A distância $\delta(V_i, V_j)$ (ou distancia fuzzy) de dois vértices distintos V_i e V_j é definida como o mínimo dos comprimentos de todos os caminhos que unem esses dois vértices. Se $V_i = V_j$ então $\delta(V_i, V_j) = 0$.

Caso o grafo seja um grafo fuzzy simplificado, é notável que $\delta(V_i, V_j)$ é uma métrica, pois:

1. $\delta(V_i, V_j) = 0$ se e somente se $V_i = V_j$, por definição;
2. $\delta(V_i, V_j) = \delta(V_j, V_i)$ já que o reverso de um caminho é um caminho e μ_R é simétrico;
3. $\delta(V_i, V_j) \leq \delta(V_i, V_k) + \delta(V_k, V_j)$, já que a união de um caminho de V_i a V_k com um caminho de V_k a V_j é um caminho de V_i a V_j .

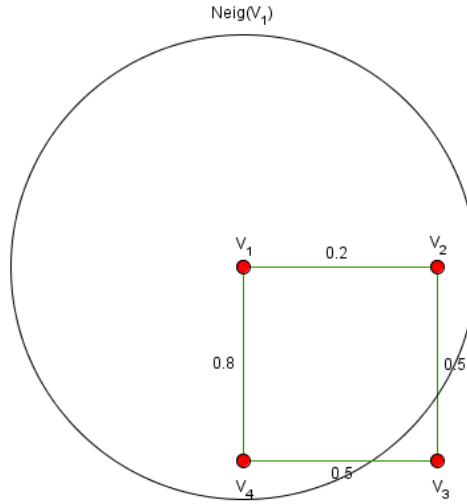
Para compreendermos melhor as duas definições de distância enunciadas acima, vamos considerar o grafo fuzzy da Figura 6 e associar os vértices V_1, V_2, V_3 e V_4 as coordenadas $(2, 4), (4, 4), (4, 2), (2, 2)$, respectivamente.

Se tomarmos os vértices V_1 e V_2 , a distância euclidiana entre eles é 2, pois: $d(V_1, V_2) = \sqrt{(2-4)^2 + (4-4)^2} = \sqrt{4} = 2$. Em contrapartida, a distância fuzzy entre eles é 5, pois existem dois caminhos que ligam V_1 a V_2 . O primeiro caminho (L_1) é a própria aresta que conecta os dois vértices ($V_1 \rightarrow V_2$) e tem comprimento $\rho(L_1) = \frac{1}{\mu_R(V_1, V_2)} = \frac{1}{0.2} = 5$, já o segundo caminho (L_2) passa pelos vértices V_4 e V_3 ($V_1 \rightarrow V_4 \rightarrow V_3 \rightarrow V_2$) e tem comprimento $\rho(L_2) = \frac{1}{\mu_R(V_1, V_4)} + \frac{1}{\mu_R(V_4, V_3)} + \frac{1}{\mu_R(V_3, V_2)} = \frac{1}{0.8} + \frac{1}{0.5} + \frac{1}{0.5} = 5.25$.

Definição 23 (Vizinhança de um vértice) *Seja $\tilde{G} < V, R >$ um grafo fuzzy. A vizinhança de um vértice V_i em V é uma área em que V_i é o centro e r é o raio. Denotaremos como $Neig(V_i)$, onde $Neig(V_i) = \{V_j \in V / d(V_j, V_i) < r\}$.*

O atributo $d(V_j, V_i)$ revela o grau de similaridade entre dois vértice que pode ser obtido por meio da distância euclidiana entre esses dois vértices ou pela distância fuzzy. Na Figura a seguir, temos um exemplo de vizinhança, onde V_1 é o centro.

Figura 8 – Exemplo de uma vizinhança



Fonte: Elaborada pelos autores

Definição 24 (Vértice de conexão) O vértice V_k é um vértice de conexão entre duas vizinhanças $Neig(V_i)$ e $Neig(V_j)$, se e somente se, V_k pertence a vizinhança $Neig(V_i)$ e a vizinhança $Neig(V_j)$.

Definição 25 (Grau de conexidade fuzzy entre duas vizinhanças) O grau de conexidade fuzzy (GCF) entre duas vizinhanças $Neig(V_i)$ e $Neig(V_j)$ é dado por:

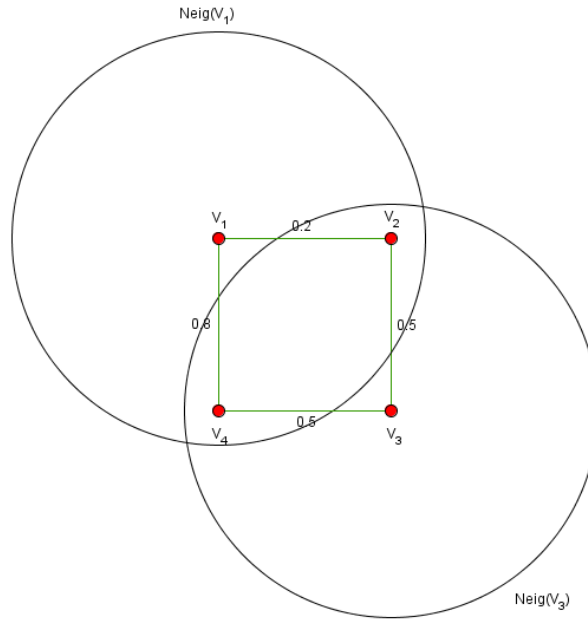
$$GCF(Neig(V_i), Neig(V_j)) = \frac{N_{(Neig(V_i) \cap Neig(V_j))}}{N_{Neig(V_i)} + N_{Neig(V_j)} - N_{(Neig(V_i) \cap Neig(V_j))}}, \quad (3)$$

onde:

- $N_{(Neig(V_i) \cap Neig(V_j))}$ é o número de vértices comum entre as duas vizinhanças $Neig(V_i)$ e $Neig(V_j)$;
- $N_{Neig(V_i)}$ é o número de vértices da vizinhança $Neig(V_i)$;
- $N_{Neig(V_j)}$ é o número de vértices da vizinhança $Neig(V_j)$.

Na figura 9, os vértices V_2 e V_4 são vértices de conexão das vizinhanças $Neig(V_1)$ e $Neig(V_3)$.

Figura 9 – Exemplo de vértices de conexão



Fonte: Elaborada pelos autores

Além disso, o grau de conexidade fuzzy entre as vizinhanças $Neig(V_1)$ e $Neig(V_3)$ é 0.25, pois

$$GCF(Neig(V_1), Neig(V_3)) = \frac{N_{(Neig(V_1) \cap Neig(V_3))}}{N_{Neig(V_1)} + N_{Neig(V_3)} - N_{(Neig(V_1) \cap Neig(V_3))}} = \frac{1}{3 + 3 - 2} = 0.25.$$

Dessa forma, notamos que o grau de conexidade fuzzy é um valor no intervalo $[0, 1]$. Quanto maior o $GCF(Neig(V_i), Neig(V_j))$, mais conectadas estão as vizinhanças. Quando $GCF(Neig(V_i), Neig(V_j)) = 1$ as vizinhanças estão sobrepostas e quando $GFC(Neig(V_i), Neig(V_j)) = 0$ as duas vizinhanças não estão conectadas.

Definição 26 (Ncentro de uma vizinhança) Seja $\tilde{G} < V, R >$ um grafo fuzzy onde $V = \{V_1, V_2, \dots, V_n\}$ é o conjunto dos vértices. Sejam V_i e V_j vértices de V com coordenadas $(x_{V_i}, y_{V_i}, \dots, z_{V_i})$ e (x_1, y_1, \dots, z_1) , respectivamente. O Ncentro da vizinhança $Neig(V_i)$ é o ponto V_j , $i \neq j$, mais próximo de V_i , ou seja, $Ncentro(Neig(V_i)) = V_j$; $d(V_i, V_j) = \min\{d(V_i, V_1), \dots, d(V_i, V_n)\}$.

Definição 27 (Ajuste de um Ncentro) Se $V_j = O'_i$ é o Ncentro de $Neig(V_i)$, então o Ncentro ajustado O_i será determinado por:

$$O_i = \frac{(x_{V_i}, y_{V_i}, \dots, z_{V_i}) + (x_1, y_1, \dots, z_1)}{2}. \quad (4)$$

Definição 28 (Ajuste de p Ncentros) Se existirem p Ncentros; $(x_1, y_1, \dots, z_1), (x_2, y_2, \dots, z_2), \dots, (x_p, y_p, \dots, z_p)$; em uma vizinhança $Neig(V_i)$; $V_i(x_{V_i}, y_{V_i}, \dots, z_{V_i})$, o Ncentro ajustado será:

$$O_i = \frac{(x_{V_i}, y_{V_i}, \dots, z_{V_i}) + (x_1, y_1, \dots, z_1) + (x_2, y_2, \dots, z_2) + \dots + (x_p, y_p, \dots, z_p)}{p + 1}. \quad (5)$$

Na Figura 9, para a vizinhança $Neig(V_1)$ encontramos dois Ncentros, a saber, V_2 e V_4 , pois $d(V_1, V_2) = d(V_1, V_4)$ e estes pontos são os mais próximos de V_1 . Assim, o Ncentro ajustado será

$$O_i = \frac{(2, 4) + (4, 4) + (4, 2)}{3} = \frac{(10, 10)}{3} = \left(\frac{10}{3}, \frac{10}{3}\right).$$

Definição 29 (Vizinhança de um Ncentro ajustado) Seja $\tilde{G} < V, R >$ um grafo fuzzy, $V = \{V_1, V_2, \dots, V_n\}$ e O_k um Ncentro ajustado em \tilde{G} . A vizinhança de um Ncentro ajustado é uma área em que O_k é o centro e r é o raio. Denotaremos como $Neig(O_k)$, onde $Neig(O_k) = \{V_i \in V / d(V_i, O_k) < r\}$.

Definição 30 (Grafo fuzzy de vizinhanças) Um grafo fuzzy $\tilde{G}' < V', R' >$ é um grafo fuzzy de vizinhanças se $V' = \{V'_1, V'_2, \dots, V'_n\}$ é o conjunto dos vértices e cada vértice representa uma vizinhança. O grau de conexidade fuzzy entre duas vizinhanças $Neig(V'_i)$ e $Neig(V'_j)$ representa o peso da aresta que liga essas duas vizinhanças, ou seja, $GCF(Neig(V'_i), Neig(V'_j)) = \mu_{R'}(Neig(V'_i), Neig(V'_j))$.

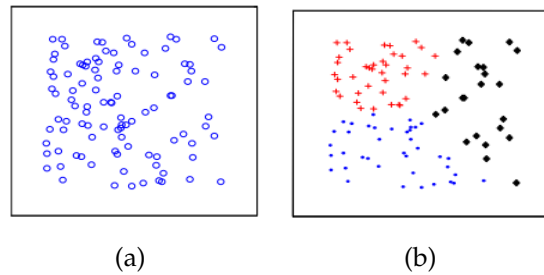
Dentro do estudo da teoria de grafos encontramos várias ramificações, entre elas, a clusterização de grafos que é um processo de agrupamento de vértices que leva em consideração a estrutura das arestas. Esse campo de estudo tem se tornado popular e o número de algoritmos propostos é bem alto. A seguir, faremos uma explanação sobre clusterização, destacando, por exemplo, alguns tipos de clusterização e alguns algoritmos usualmente encontrados na literatura.

2.3 Clusterização

Clusterização é uma técnica computacional cujo propósito consiste em agrupar objetos de acordo com as suas características. A ideia básica consiste em colocar em um mesmo grupo objetos que sejam similares de acordo com um ou mais critérios predeterminados. Esses grupos chamamos de "clusters".

Um exemplo de clusterização é mostrado na Figura 10. Os padrões originais são mostrados na Figura 10(a) e os agrupamentos (clusters) desejados são mostrados na Figura 10(b). Aos pontos pertencentes ao mesmo cluster é atribuída a mesma identificação.

Figura 10 – Exemplo de clusterização



Fonte: Jain (2010, p.658)

De acordo com Jain (2010), a clusterização tem sido amplamente estudada na mineração de dados por causa de suas inúmeras aplicações. Na ausência de informações específicas, a clusterização pode ser considerada como um modelo conciso para análise e interpretação de dados. O problema básico desse tipo de abordagem é conseguir particionar um conjunto de dados em grupos de forma que os dados pertencentes a cada grupo sejam tão semelhantes quanto possível.

Aggarwal e Chandan (2013) destacam alguns domínios de aplicações comuns que podem ser modeladas por meio da clusterização:

- Passo intermediário para outros problemas fundamentais de mineração de dados: Um cluster pode ser considerado uma forma de compactação de dados e muitas vezes serve como um passo intermediário importante para muitos problemas de mineração de dados.
- Filtragem Colaborativa: Nos métodos de filtragem colaborativas a clusterização fornece grupo de usuários de mesma opinião. As classificações fornecidas pelos diferentes usuários são utilizadas a fim de executar a filtragem colaborativa.
- Segmentação de Clientes: Esta aplicação é bastante semelhante ao de filtragem colaborativa, uma vez que a clusterização dos dados cria-se grupos de clientes semelhantes. A principal diferença entre filtragem colaborativa e segmentação de clientes é que esta pode utilizar atributos arbitrários sobre os dados para realizar o agrupamento.
- Resumo de dados: Muitos métodos de clusterização estão intimamente relacionados com métodos de redução de conjuntos de dados de alta dimensão. Tais métodos podem ser considerados como uma forma de compactação de dados. O resumo de dados pode ser útil na criação de representações de dados compactos, que, por sua vez, são mais fáceis de processar e interpretar.

- **Detecção de Tendência:** Muitas formas de algoritmos dinâmicos podem ser usados para realizar detecção de tendência em uma variedade de aplicações em redes sociais. Em tais aplicações, os dados agrupados podem ser utilizados a fim de determinar os importantes padrões de mudanças.
- **Análise de Dados de Multimídia:** Uma variedade de diferentes tipos de documentos, tais como imagens, áudio ou vídeo, caem na categoria geral de dados multimídia. A determinação de segmentos similares tem inúmeras aplicações, tais como a determinação de trechos semelhantes de músicas ou fotografias semelhantes.
- **Análise de Dados Biológicos:** Os dados biológicos são geralmente estruturados como sequências ou como redes. Algoritmos de clusterização proporcionam boas ideias das principais tendências nos dados, bem como as sequências incomuns.
- **Análise de redes sociais:** Nestas aplicações, a estrutura de uma rede social é usado para determinar as comunidades importantes na rede subjacente. A detecção de comunidades tem importantes aplicações em análise de redes sociais, porque fornece uma importante compreensão da estrutura da comunidade na rede. A clusterização também tem aplicações para compactação de rede social, que é útil em outras aplicações.

Dessa forma, notamos que a clusterização pode ser empregada para reduzir, minerar e analisar dados, reconhecer padrões, processar e segmentar imagens, para realizar pesquisa de mercado, padrão de compra, análise de sintomas de doenças, características de seres vivos, funcionalidades de genes, verificar a composição de solos, agrupar documentos, agrupar cidades, estudar dados de genoma na biologia, dentre muitas outras (CASSIANO, 2014).

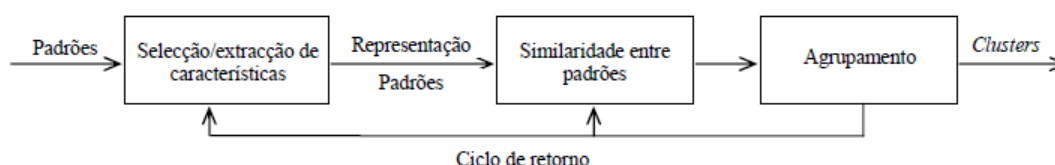
Nesse contexto, diversos autores tem utilizado a clusterizações para realizar estudo e/ou aplicações em diversas áreas como: na Física (ANDERSON et al., 2014; HATAMLOU, 2013); na Geografia (FITRIANAH, 2016); na Web (CHRISTODOULOU, PATON e FERNANDES, 2015; CÔME e LATOUCHE, 2015; ESMIN, TOLEDO e OLIVEIRA, 2015; TOMASEV et al., 2013); na Engenharia (ALVES e MELLO, 2015; LOPES, 2014; CERVO, 2013); na Biologia (ESMIN, CALEGARI e PEREIRA, 2015; SOUSA e NASCIMENTO, 2015); na Medicina (BUENO et al., 2014) e na Educação (BAKER e INVENTADO, 2014; SILVA e SANTOS, 2007).

Em geral, a atividade de clusterização de dados envolve os seguintes passos:

1. Representação de padrões (opcionalmente inclui extração de características e/ou seleção);

2. Definição de uma medida de proximidade de padrões apropriada ao contexto dos dados;
3. Clusterização;
4. Abstração de dados (se necessário);
5. Avaliação dos clusters.

Figura 11 – Etapas da clusterização



Fonte: Castro (2003)

A figura acima ilustra os três primeiros passos da clusterização. Como destacado, o segundo passo consiste na definição de uma medida de proximidade entre os dados. As medidas de similaridade geralmente utilizadas nos algoritmos de clusterização são as medidas de distância. Dentre elas, destacam-se as medidas de Minkowski, Manhattan, Euclidiana e Chebyshev.

Outra forma de medir a similaridade dos dados é usando medidas relacionais como o coeficiente de Pearson e medidas de associação como o índice de Jaccard. Tanto as medidas de distância quanto as correlacionais requerem dados métricos, enquanto as medidas de associação é para dados não métricos.

2.3.1 Métodos de Clusterização

De acordo com Bailey (1975) apud Cassiano (2014) os estudos de clusterização surgiram na Antropologia a partir do trabalho de Driver e Kroeber em 1932 e na Psicologia a partir dos trabalhos de Zubin em 1938 e Tryon em 1939. Contudo, as diferentes técnicas de clusterização ficaram restritas durante muito tempo a um grupo reduzido de pesquisadores devido a sua complexidade matemática. Por causa do desenvolvimento da tecnologia computacional foi possível que ocorresse a propagação da técnica entre os diferentes ramos do conhecimento e o desenvolvimento de novos métodos.

Os particionais e os hierárquicos são os métodos mais tradicionais de clusterização. Alguns algoritmos de clusterização integram as ideias de vários outros, dessa

forma, é difícil classificar um dado algoritmo como unicamente pertencendo a somente uma categoria de método de clusterização.

Cassiano (2014) sublinha que uma classificação geral dos algoritmos de clusterização divide os algoritmos em dez tipos principais. Entre eles estão os hierárquicos, os baseados em lógica fuzzy e os que se baseiam em grafos.

2.3.1.1 Métodos Hierárquicos

Para Jain (2010) os algoritmos de clusterização hierárquica, em geral, encontram os clusters de modo aglomerativo ou divisivo. No modo aglomerativo cada ponto do conjunto de dados é um cluster e nos passos do algoritmo ocorrem agrupamentos sucessivos de pares similares de clusters a fim de formar uma hierarquia de clusters. No modo divisivo o processo começa com todos os pontos do conjunto de dados em um mesmo cluster e de forma recursiva, ocorre divisões de clusters. A entrada de um algoritmo hierárquico, geralmente é uma matriz de similaridade $n \times n$ em que n é o número de objetos a serem agrupados.

O primeiro registro publicado de um método de clusterização ocorreu em 1948 com Sorensen, em que o autor definiu o Método Hierárquico Aglomerativo de Ligação Completa. Nove anos depois, em 1957, Sneath lança o método Hierárquico de Ligação Simples e em 1963 Ward apresenta o método de clusterização WARD que minimiza o total de variância intra-clusters (CASSIANO, 2014).

Em 1987 o popular método hierárquico baseado em modelos que organiza os dados em uma árvore de forma incremental, o COBWEB, foi apresentado por Fisher. Dois anos depois, em 1989, Fisher e outros autores apresentaram o CLASSIT que é a versão do COBWEB para atributos numéricos ou contínuos. No mesmo ano, Kauffman e Rousseeuw desenvolveram três algoritmos hierárquicos: o DIANA (Divise Analysis), o MONA (Monotetic Analysis) e o AGNES (Aglomerative Nesting) (CASSIANO, 2014).

Em 1994, García e outros autores desenvolveram o EJCluster, um algoritmo hierárquico de formação divisiva na área de processamento de sinais. Dois anos depois, Schikuta apresenta o Gridfile que é um algoritmo baseado em grades para base de dados muito grande e Zhuang et al. apresentam o BIRCH, um algoritmo hierárquico de formação aglomerativa otimizada. Wang et al. (1997) e Guha et al (1998) também apresentaram algoritmos de formação aglomerativa o STING e o CURE, respectivamente (CASSIANO, 2014).

Em 1999, Hrasna et al. apresentam o MAFIA, um algoritmo adaptativo de intervalos finitos. Entre os anos de 2000 e 2010, diversos autores não mediram esforços na busca de maneiras alternativas de tornar o processo de clusterização mais eficiente tanto do ponto de vista dos resultados quanto do ponto de vista de custo computacional,

como por exemplo: o K-Means Divisivo por Steinbach et. al em 2000; o ROCK por GUHA et al. em 2000; o GRIN por Chien-Yu et al. em 2002; o EBABS por Yin et al. em 2006 e o SCAHIPAT por Li e Wang em 2010 (CASSIANO, 2014).

Em geral, o resultado de uma clusterização tradicional gera clusters disjuntos, ou seja, cada objeto pertence a um e somente um cluster. Entretanto, com o surgimento da lógica fuzzy, diversos autores passaram a aplicar a teoria dos conjuntos fuzzy para fazer clusterização, como veremos a seguir.

2.3.1.2 Métodos Baseados em Lógica fuzzy

Uma das características da clusterização baseada na lógica fuzzy é a possibilidade de um objeto pertencer a vários clusters e cada cluster é um conjunto fuzzy de todos os indivíduos. O conceito de conjuntos fuzzy oferece a vantagem de expressar o tipo de situação em que um indivíduo compartilha similaridade com vários grupos, o algoritmo assim associa cada indivíduo parcialmente a todos os grupos.

De acordo com Nunes (2014), diversos algoritmos utilizando lógica fuzzy foram apresentados por vários estudiosos. Dentre eles, destacamos o Fuzzy C-Means que é uma extensão do tão popular algoritmo K-Means desenvolvido por Steinhaus em (1956). O Fuzzy C-Means ou FCM é um algoritmo de partição que se baseia na lógica fuzzy e foi desenvolvido por Dunn em 1973.

Outras abordagens de clusterização baseadas em lógica fuzzy foram os métodos de BACKER proposto em 1978 por Backer e o de Bezdek em 1981 que desenvolveu o método Fuzzy C-Means Melhorado. Depois disso, em 1993, surgem na literatura outros métodos variantes do C-Means como o Fuzzy C-Sheels (FCS) que foi um método desenvolvido por Yang, o PCM- Probabilístico C-Means que foi apresentado por Krishnapuram e Keller e é uma abordagem fuzzy para lidar com valores atípicos (NUNES, 2014).

Yager e Filev em 1994 propuseram o método de Clusterização de Montanha, um algoritmo fuzzy capaz de estimar os centros de clusters. Já nove anos depois o método MSVC–Multi-Sphere Support Vector Clustering foi proposto por Chiang e Hao. Eles combinaram o conceito de pertinência fuzzy com clusterização de suporte vetor (CASSIANO, 2014).

Dois anos depois, em 2005, o método C-Means com GA foi desenvolvido por Mota e Gomide. Esse algoritmo é uma aplicação do método Fuzzy C-Means na fase de avaliação do cromossomo em algoritmo genético. E, em 2010, He e Pan apresentam o DENCLUE NEURO-Fuzzy que é um algoritmo que usa uma abordagem fuzzy para o DENCLUE, algoritmo de densidade apresentado por Hinneburg e Keim em 1998 (CASSIANO, 2014).

Alguns algoritmos hierárquicos que se baseiam na lógica fuzzy são relacionados a algoritmos de clusterização baseados em grafos, como veremos na próxima subseção.

2.3.1.3 Métodos Baseados em Grafos

Nos algoritmos de clusterização baseados em grafos cada objeto do conjunto de dados é representado por um vértice e a similaridade existente entre dois vértices é expressada por meio da existência de uma aresta entre eles.

Uma forma de estabelecer conexões entre os vértices é ligar cada vértice aos vértices restantes, onde o peso indicará a similaridade entre os dois dados. Para isso, adota-se uma medida de similaridade no processo de agrupamento e, quando se trata de grafos fuzzy, vimos que o peso das aresta é sempre um valor no intervalo $[0, 1]$. Nesse sentido, quanto mais próximo de 1 for o peso de uma aresta que liga dois vértices, maior é o grau de similaridade entre eles.

A primeira abordagem de clusterização baseada em grafos ocorreu em 1992 quando Hagen e Kahng desenvolveram o Ratio Cut. Sete anos depois, em 1999, Karypis et al. lançam o inovador CHAMALLEON, algoritmo hierárquico baseado em grafos. O HCS (High Connected Subgraph - Subgrafos altamente conectados) é outro método de clusterização baseado na teoria dos grafos. Ele foi desenvolvido por Hartuv et al. em 2000 (CASSIANO, 2014).

Em 2002, Belkin e Niyogi, apresentaram o Laplacian Eigenmap, um método de clusterização espectral. E o ORCLUS (Arbitrarily Oriented Projected Cluster Generation) foi introduzido por Aggarwal e Yu em 2002 e também se baseia na teoria de grafos (CASSIANO, 2014).

Já em 2006, Dong et al. apresentaram o FHC. Além disso, em 2007, Banerjee et al. apresentaram o método de clusterização baseada em grafos para dados relacionais, onde diferentes tipos de objetos são simultaneamente agrupados com base em seus valores de atributos intrínsecos (CASSIANO, 2014).

O algoritmo FCH, em particular, é um algoritmo aglomerativo hierárquico que se baseia na conexidade de grafos fuzzy. Nos passos do algoritmo são criadas vizinhanças e entre cada duas vizinhanças é calculado o grau de conexidade fuzzy e criado um grafo fuzzy de vizinhança. Diante disso, Dong, et al. (2006) apresentam algumas definições referentes ao processo de clusterização de grafos.

Definição 31 (Cluster) Se $G'_\lambda < V', E'_\lambda >$ é um grafo λ -corte de um grafo fuzzy de vizinhanças $\tilde{G}' < V', R' >$ então cada componente conexa de G'_λ representa um cluster (C).

Definição 32 (Subcluster) Um subcluster (SC) é uma vizinhança de um grafo fuzzy de vizinhanças.

Definição 33 (Ruído) *Seja $\tilde{G} < V, R >$ um grafo fuzzy. Um ruído é um vértice de V que pertence a nenhum cluster.*

Notamos que um grafo fuzzy de vizinhanças também pode ser chamado de grafo fuzzy de subclusters já que um subcluster é uma vizinhança. Para melhor ilustrar as ideias de grafo fuzzy de subclusters e cluster propostas por Dong, et al. (2006), no próximo capítulo apresentamos uma análise mais detalhada do algoritmo FHC. Antes disso, faremos uma breve explicação sobre métodos de análise de clusters.

2.3.2 Medidas de Avaliação de Clusters

O processo de avaliação do resultado obtido a partir de um algoritmo de agrupamento é comumente chamado de validação. Para análise de agrupamentos, a questão análoga é como avaliar o quão satisfatório são os grupos resultantes. Esse processo se torna importante, pois evita encontrar padrões nos ruídos, comparar algoritmos de agrupamento ou dois conjuntos de grupos ou duas partições ou dois clusters.

Existem dois principais tipos de índices de validação de agrupamento: os índices externos e os índices internos. Os índices externos comparam a estrutura de grupos descoberta com uma estrutura de grupos previamente conhecida, enquanto os índices internos analisam a estrutura de grupos descoberta em relação a algum critério, como por exemplo, compacidade e/ou separabilidade.

2.3.2.1 Índices Internos

Na literatura os índices internos de validação de agrupamento fuzzy são divididos nos que se baseiam ou não em medidas de similaridade. Dentro dos que não se baseiam em medidas de similaridade destacam-se: coeficiente da partição; entropia da partição; critério de Xie-Beni; critério de Fukuyama-Sugeno; critério de Zahid; hipervolume fuzzy; densidade média da partição; distância média dentro do grupo e silhueta fuzzy. Nos que se baseiam em medidas de similaridade destaca-se o critério de avaliação baseado em medidas de similaridade (GARCÍA, PONS e SHULCLOPER, 2012).

2.3.2.2 Índices Externos

Os índices externos de validação de agrupamento fuzzy podem ser divididos em dois critérios: o índice de Rand fuzzy e o novo índice de Rand fuzzy. Existe uma grande variedade de índices que utilizam os mesmo termos que o índice de Rand. Estes índices externos encontrados são: o índice de Rand Ajustado, o coeficiente de Jaccard, o índice Fowlkes-Mallows, entre outros (GARCÍA, PONS e SHULCLOPER, 2012).

3 MATERIAIS E MÉTODOS

Para conduzirmos a nossa pesquisa, inicialmente, realizamos os estudos bibliográficos focalizados na compreensão dos conceitos e definições que acabamos de resumir, definindo assim a nossa fundamentação teórica. Em seguida, realizamos uma análise do algoritmo de clusterização hierárquica baseada na conexidade de grafo fuzzy (FHC) visando compreender sua dinâmica ou seu funcionamento. Posteriormente, avaliamos a influência de uma medida de similaridade fuzzy neste algoritmo e definimos um ajuste de N_{centro} fuzzy diante do processo de clusterização. Depois, aplicamos o algoritmo modificado para realizar agrupamento de cidades de acordo com o Índice de Desenvolvimento Humano Municipal (IDHM) e no processo de segmentação de imagens a fim de verificarmos se os resultados da clusterização utilizando uma medida de similaridade fuzzy são mais satisfatórios em comparação ao algoritmo FHC.

3.1 Análise do Algoritmo FHC

Como discorreremos na fundamentação teórica, o FHC é um algoritmo de clusterização hierárquica baseada na conexidade de grafos fuzzy. Esse algoritmo foi desenvolvido por um grupo de pesquisadores chineses em 2006 e se aplica a teoria de conjuntos fuzzy em um método de agrupamento hierárquico, de modo a descobrir clusters com formas arbitrárias.

O algoritmo pode ser aplicado em conjuntos de dados de alta dimensão, encontrando clusters de formatos como o esférico, linear, alongado ou côncavo. Nesse sentido, o FCH gera clusters de melhor qualidade do que algoritmos tradicionais para conjunto de dados muito grandes (DONG, et al., 2006). Apresentamos, na próxima seção, a descrição desse algoritmo.

3.1.1 O Algoritmo FHC

Adiante segue uma ideia adaptada do FHC. O algoritmo recebe como entrada o conjunto de dados, que são as coordenadas dos vértices, um raio dos subclusters (ou vizinhanças) que serão gerados nos passos do algoritmo e um parâmetro λ que é um valor dentro do intervalo $[0, 1]$.

Nas etapas 1-8 são formados os subclusters do grafo fuzzy de subclusters. Inicialmente cada vértice é um cluster. Para cada vértice encontramos o N_{centro} do subcluster que é o ponto mais próximo do vértice considerado. Se a distância entre o vértice considerado e o N_{centro} é menor do que o raio dado, o vértice considerado pertence ao

subcluster de centro no N_{centro} ajustado. Caso contrário, se a distância é maior do que o raio, um novo subcluster é criado e contém apenas o vértice considerado.

Nas etapas 9-11 marcamos os vértices que pertencem a mais de um subcluster e nas etapas 12-14 construímos o grafo fuzzy de subclusters após calcular o grau de conexidade fuzzy entre as vizinhanças. Com o parâmetro λ dado, geramos o λ -corte do grafo e as componentes conexas desse grafo será o resultado da clusterização.

Algoritmo FHC

Entrada: As coordenadas dos vértices do grafo $\tilde{G} = \langle V, R \rangle$, o raio r dos subclusters e o parâmetro λ

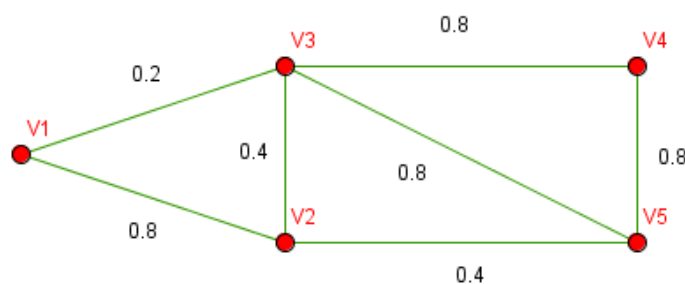
Saída: Componentes conexas do grafo λ -corte do grafo fuzzy de subclusters

Algoritmo:

1. Cada ponto é um cluster
 2. Para cada ponto $V_i \in V$
 3. { Encontrar o N_{centro} O'_i
 4. Se ($dist(V_i, O'_i) < r$)
 5. { V_i pertence ao subcluster SC_j : $SC_j = SC_j \cup V_i$ em que o centro é o N_{centro} ajustado O_i }
 6. Se não
 7. {Criar um novo subcluter contendo apenas V_i e V_i é o centro do subcluster}
 8. }
 9. Para cada ponto $V_i \in V$
 10. {Para cada subcluster $SC_j \in SC$
 11. Se (V_i está no subcluster SC_j) fazer uma marcação neste subcluster SC_j
 12. // Construir um grafo fuzzy dos subclusters e então calcular o grau de conexidade fuzzy entre as vizinhanças
 13. Calcular o grau de conexidade fuzzy entre as vizinhanças
 14. Gerar o $\lambda_Grafo(G')$
-

Para compreendermos melhor como funciona a dinâmica do algoritmo, a título de ilustração, consideraremos o grafo fuzzy abaixo:

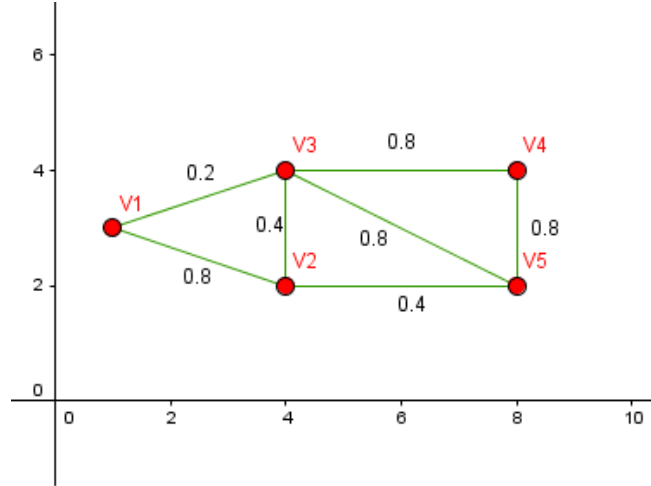
Figura 12 – Grafo fuzzy \tilde{G}



Fonte: Yeh e Bang (1975) adaptada pelos autores

Como podemos notar na Figura 12, o grafo fuzzy é composto por cinco vértices que compõem o conjunto dados, a saber, $V = \{V_1, V_2, V_3, V_4, V_5\}$. Cada um desses vértices está associado a um ponto no plano cartesiano, conforme a Figura 13, onde os pontos V_1, V_2, V_3, V_4 e V_5 tem coordenadas $(1, 3), (4, 2), (4, 4), (8, 4), (8, 2)$, respectivamente.

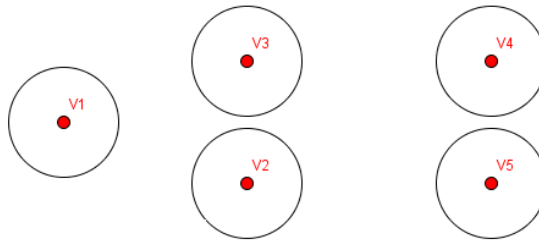
Figura 13 – Grafo fuzzy \tilde{G} no plano cartesiano



Fonte: Yeh e Bang (1975) adaptada pelos autores

Como dados de entrada, além das coordenadas $(1, 3), (4, 2), (4, 4), (8, 4), (8, 2)$, consideraremos o raio como 3.1 e o parâmetro λ como 0.5. Como vimos no algoritmo, inicialmente cada vértice é um cluster.

Figura 14 – Clusters iniciais do grafo \tilde{G}



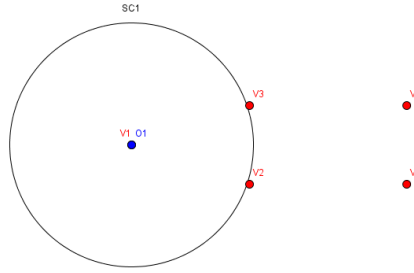
Fonte: Elaborada pelos autores

Cada um dos clusters acima pode ser visto como uma vizinhança. Encontraremos o N_{centro} de cada uma dessas vizinhanças. Assim, calculando primeiramente a distância euclidiana de V_1 em relação a todos os outros vértices, temos:

$$\begin{aligned} d(V_1, V_2) &= \sqrt{(1-4)^2 + (3-2)^2} = \sqrt{(-3)^2 + (1)^2} = \sqrt{9+1} = \sqrt{10} \approx 3.16; \\ d(V_1, V_3) &= \sqrt{(1-4)^2 + (3-4)^2} = \sqrt{(-3)^2 + (-1)^2} = \sqrt{9+1} = \sqrt{10} \approx 3.16; \\ d(V_1, V_4) &= \sqrt{(1-8)^2 + (3-4)^2} = \sqrt{(-7)^2 + (-1)^2} = \sqrt{49+1} = \sqrt{50} \approx 7.07; \\ d(V_1, V_5) &= \sqrt{(1-8)^2 + (3-2)^2} = \sqrt{(-7)^2 + (1)^2} = \sqrt{49+1} = \sqrt{50} \approx 7.07. \end{aligned}$$

Logo, os vértices mais próximos de V_1 são V_2 e V_3 . Assim, os $N_{centros}$ de $Neig(V_1)$ são V_2 e V_3 . Entretanto, como $d(V_1, V_2) = d(V_1, V_3) = 3.16 \geq r$ criaremos um subcluster SC_1 contendo apenas o ponto V_1 e o centro será esse próprio vértice, conforme destacamos na Figura 15.

Figura 15 – Subcluster SC_1 do grafo $\widetilde{G'}$ usando distância clássica



Fonte: Elaborada pelos autores

Analogamente, calculando a distância euclidiana do vértice V_2 em relação aos outros vértices, obtemos: $d(V_2, V_1) \approx 3.16$; $d(V_2, V_3) = 2.00$; $d(V_2, V_4) \approx 4.47$ e $d(V_2, V_5) = 4.00$. Logo, o vértice mais próximo de V_2 é V_3 e V_3 é o N_{centro} de $Neig(V_2)$. Como $d(V_2, V_3) = 2.00 < r$ o vértice V_2 pertence ao subcluster SC_2 de centro no N_{centro} ajustado O_2 com coordenadas $(4, 3)$. Calculando a distância de todos os pontos ao centro O_2 , encontramos:

$$d(V_1, O_2) = 3.00 < r;$$

$$d(V_2, O_2) = 1.00 < r;$$

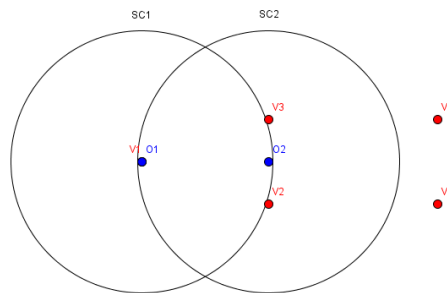
$$d(V_3, O_2) = 1.00 < r;$$

$$d(V_4, O_2) \approx 4.12 \geq r;$$

$$d(V_5, O_2) \approx 4.12 \geq r.$$

Assim, o subcluster SC_2 é composto pelos vértices V_1 , V_2 e V_3 , como podemos observar na Figura 16.

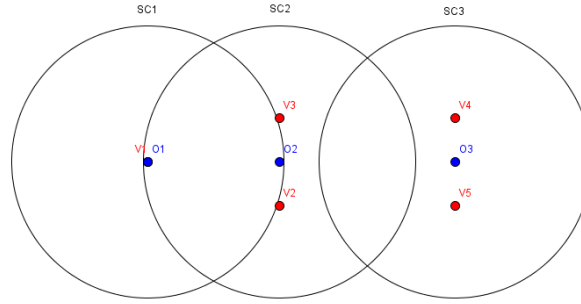
Figura 16 – Subcluster SC_2 do grafo $\widetilde{G'}$ usando distância clássica



Fonte: Elaborada pelos autores

Realizando o mesmo procedimento para o vértice V_3 , encontramos um subcluster igual ao subcluster SC_2 , pois o vértice mais próximo de V_3 é V_2 e o N_{centro} ajustado será o mesmo ponto O_2 com coordenadas $(4, 3)$. Para os vértices V_4 e V_5 encontramos um subcluster SC_3 , conforme a Figura 17.

Figura 17 – Subcluster SC_3 do grafo $\widetilde{G'}$ usando distância clássica



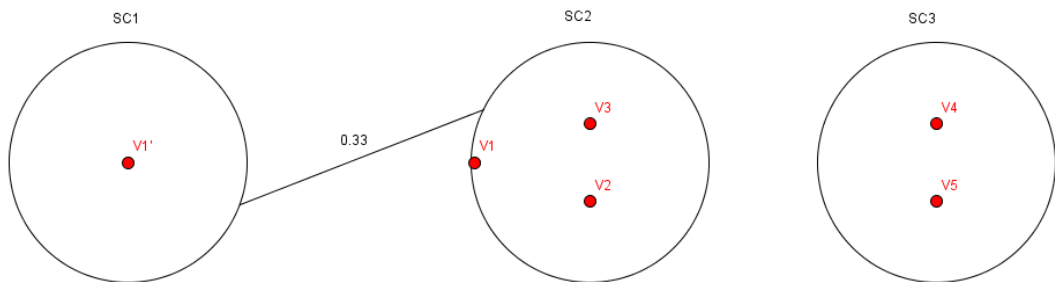
Fonte: Elaborada pelos autores

De acordo com a Figura 17, notamos que o vértice V_1 é um vértice de conexão entre as vizinhanças SC_1 e SC_2 . Os vértices V_2 e V_3 pertencem somente a vizinhança SC_2 e os vértices V_4 e V_5 pertencem somente a vizinhança SC_3 . Assim, calculando o grau de conexidade fuzzy entre as vizinhanças, temos:

$$\begin{aligned}\mu_{R'}(SC_1, SC_2) &= \frac{\text{cont}(SC_1, SC_2)}{N_{SC_1} + N_{SC_2} - \text{cont}(SC_1, SC_2)} = \frac{1}{1+3-1} \approx 0.33; \\ \mu_{R'}(SC_1, SC_3) &= \frac{\text{cont}(SC_1, SC_3)}{N_{SC_1} + N_{SC_3} - \text{cont}(SC_1, SC_3)} = \frac{0}{1+2-0} = 0; \\ \mu_{R'}(SC_2, SC_3) &= \frac{\text{cont}(SC_2, SC_3)}{N_{SC_2} + N_{SC_3} - \text{cont}(SC_2, SC_3)} = \frac{0}{3+2-0} = 0.\end{aligned}$$

A partir das informações acima, geramos o grafo fuzzy de subclusters em que SC_1 , SC_2 e SC_3 são os vértices do grafo e $\mu_{R'}(SC_1, SC_2) \approx 0.33$ é o peso da aresta que conecta o vértice SC_1 ao vértice SC_2 .

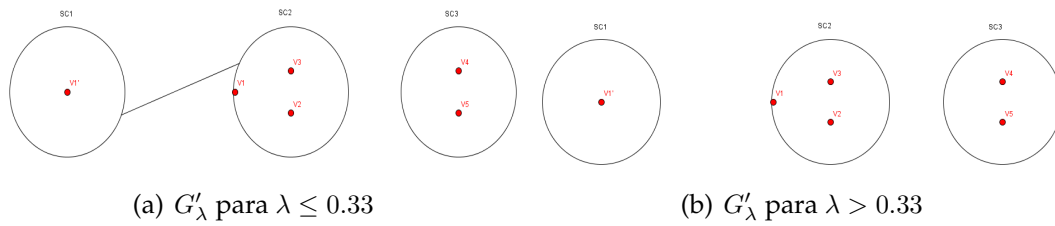
Figura 18 – Grafo fuzzy de subclusters $\widetilde{G'}$ usando distância clássica



Fonte: Elaborada pelos autores

O parâmetro λ define o resultado da clusterização. Para um $\lambda \leq 0.33$, λ -corte do grafo (G'_λ) será conforme a Figura 19(a) e o resultado gera dois clusters (já que o grafo é desconexo e tem duas componentes conexas), a saber, $C_1 = \{V_1, V_2, V_3\}$ e $C_2 = \{V_4, V_5\}$. Para qualquer $\lambda > 0.33$, a aresta que conecta SC_1 ao SC_2 é eliminada e o G'_λ será o grafo conforme a Figura 19(b) com três componentes conexas, gerando assim, três clusters: $C_1 = \{V_1\}$, $C_2 = \{V_1, V_2, V_3\}$ e $C_3 = \{V_4, V_5\}$.

Figura 19 – λ -corte do grafo $\widetilde{G'}$ usando distância clássica



Fonte: Elaboradas pelos autores

Para testarmos o algoritmo FHC em um conjunto de dados maior, foi necessário implementarmos computacionalmente. A implementação foi feita na linguagem de programação C++, conforme discutiremos na próxima seção.

3.1.2 Implementação do Algoritmo FHC

Entre todas as linguagem de programação e softwares disponíveis, optamos em implementar o algoritmo FHC na linguagem C++. Tal escolha foi feita porque o C++ é uma das linguagens de programação mais completa que existe, é altamente expressiva e lógica, flexível, de multiparadigma e pode ser utilizada em qualquer plataforma para qualquer finalidade.

Além disso, o C++ é uma extensão da linguagem C, dessa forma, ela possui todas as características apresentadas por esta linguagem e, como incremento, o C++ permite o uso de orientação a objetos e possuem algumas bibliotecas facilitadoras que permitem que o código fique menos extenso. A seguir, apresentamos uma síntese de como feita a implementação do algoritmo.

3.1.2.1 Explicação do Código

O código recebe como entrada um arquivo de texto contendo as coordenadas dos pontos, o valor do raio e do parâmetro λ . Criamos uma função "distance" que calcula a distância de cada ponto em relação a todos os outros e um vetor de vetores do tipo float para armazenar essas distâncias como uma matriz. Criamos um vetor "smallerDistance" que armazena a menor distancia maior que zero de cada linha da

matriz com as distâncias, permitindo assim, encontrar o ponto (ou pontos) mais próximo de cada vértice (N_{centro}).

Figura 20 – Recortes da implementação do FHC

```

float distance(vector<float> point1, vector<float> point2) {
    if (point1.size() != point2.size()) {
        throw "Incompatible size";
    }
    float d = 0;
    for (int i = 0; i < point1.size(); i++) {
        d += pow(point1[i] - point2[i], 2);
    }
    return sqrt(d);
}

vector<float> smallerDistance;
for (int i = 0; i < howManyPoints; i++) {
    float smaller;
    if (i == 0) smaller = distances[i][1];
    else smaller = distances[i][0];
    for (int j = 0; j < howManyPoints; j++) {
        if (j == i) continue;
        if (distances[i][j] < smaller) smaller = distances[i][j];
    }
    smallerDistance.push_back(smaller);
}

```

(a) Função "distance"

(b) Vetor "smallerDistance"

Fonte: Elaboradas pelos autores

Criamos uma classe "cluster" em que o método "updateCenter" permite realizar o ajuste do N_{centro} . Dessa forma, o algoritmo verifica se a distância do ponto considerado até o ponto (ou pontos) mais próximo a ele é menor que o raio dado e caso afirmativo, chamamos esse método para ajustar o N_{centro} e usamos novamente a função "distance" para calcular a distância do N_{centro} ajustado em relação a todos os pontos e caso as distâncias sejam menores que o raio, armazenaremos os pontos no cluster usando um dos métodos na classe "cluster". Criamos um vetor "clusters" cujos elementos são os objetos criados pela classe "cluster", armazenando, dessa forma, os subclusters.

Figura 21 – Método updateCenter

```

void updateCenter() {
    if (points.size() == 0) return;
    for (int i = 0; i < points[0].point.size(); i++) {
        center[i] = 0;
        for (int j = 0; j < points.size(); j++) {
            center[i] += points[j].point[i];
        }
        center[i] /= points.size();
    }
}

```

Fonte: Elaborada pelos autores

Criamos a função "ClearRepeatedClusters" para limpar os subclusters iguais armazenados no vetor "clusters" e a função "CalculateLinks" para calcular o grau de conexão fuzzy entre os subclusters. Em seguida, criamos a função "MescleClusters" que mescla os subclusters a partir do valor de λ sobre os graus de conexões fuzzy encontrados e a função "printClustersWithPointNumber" que permite exibir os clusters como resultado final.

Figura 22 – Função calculatelinks

```

vector <vector<float>> calculateLinks(vector <Cluster> clusters) {
    vector <vector<float>> links;
    for (int i = 0; i < clusters.size(); i++) {
        links.emplace_back();
        for (int j = 0; j < clusters.size(); j++) {
            if (i == j) {
                links[i].push_back(1);
                continue;
            }
            float Np = clusters[i].points.size();
            float Nq = clusters[j].points.size();
            float contPq = 0;
            for (int k = 0; k < clusters[i].points.size(); k++) {
                for (int l = 0; l < clusters[j].points.size(); l++) {
                    if (clusters[i].points[k] == clusters[j].points[l]) {
                        contPq += 1;
                        break;
                    }
                }
            }
            links[i].push_back(contPq / (Np + Nq - contPq));
        }
    }
    return links;
}

```

Fonte: Elaborada pelos autores

Após implementarmos o algoritmo, pudemos testar para conjuntos com maior número de dados. Entretanto, não conseguimos aplicar no mesmo conjunto de dados experimental utilizados pelos autores do FCH, pois eles não destacaram o conjunto de dados no artigo e além disso, os dados foram obtidos a partir de "Ningbo On-line"(www.nbol.net), um site cujo idioma está em chinês.

Após a análise do algoritmo, notamos que os dados de entrada são vistos como uma nuvem de dados e não como um grafo fuzzy, já que em nenhum momento do algoritmo são considerados os pesos das arestas do grafo. Como proposta de melhoria do algoritmo, na próxima seção adotaremos uma medida de similaridade fuzzy e definiremos um ajuste de N_{centro} fuzzy afim de gerar clusters em que os vértices possuam um maior grau de similaridade.

3.2 FHC e Distância fuzzy

Dentre as vertentes atuais de clusterização, se destacam aquelas que buscam maneiras alternativas de tornar o processo mais eficiente. Quando se refere a clusterização de grafos, o principal interesse é gerar clusters em que os vértices possuam um maior grau de similaridade. Entretanto, vimos que a proposta do FHC é gerar um grafo fuzzy de subclusters a partir de uma nuvem de dados e o agrupamento desses vértices ocorre obedecendo uma distância clássica preestabelecida. Isso pode contribuir para que os

resultados não sejam geralmente satisfatórios, pois proximidade geométrica de dados pode não significar similaridade.

A utilização de uma medida de similaridade que considera os pesos das arestas que conecta os vértices, se torna fundamental nesse processo. Nesse sentido, recorreremos a medida de similaridade fuzzy proposta por Rosenfeld (1975). Essa definição de distância ou métrica é definida como o mínimo do comprimento de todos os caminhos L_i que ligam dois vértices distintos. O comprimento $\rho(L)$ de cada caminho é obtido pela Equação (2). É importante ressaltar que, caso não exista um caminho ligando dois vértices distintos, isso significa que a distância fuzzy entre eles não existe e esses vértices nunca estarão no mesmo cluster, seja qual for valor do raio.

O diferencial desse tipo de métrica é que se dois vértices estão mais próximos significa que eles são mais similares. Por outro lado, se não existe um caminho, isso significa que não há similaridade entre esses dois vértices. Entretanto, para substituir a definição de distância fuzzy nas duas etapas do algoritmo que usam a distância euclidiana, precisaremos definir um ajuste de N_{centro} fuzzy, como veremos a seguir.

3.2.1 Ajuste de N_{centro} fuzzy

Vimos na análise do algoritmo FHC que após encontrarmos o ponto (ou pontos) mais próximo de cada vértice e caso a distância de cada vértice considerado até o ponto (ou pontos) mais próximo seja menor que o raio dado, é necessário que o N_{centro} seja ajustado. O ajuste de N_{centro} definido no algoritmo FHC permite encontrar a coordenada do N_{centro} ajustado e, dessa forma, calcular a distância euclidiana de cada N_{centro} ajustado a todos os pontos do conjunto de dados e assim gerar os subclusters.

Uma vez que utilizamos a distância fuzzy para encontrar o ponto (ou pontos) mais próximo de cada vértice, não poderemos utilizar a definição de ajuste proposta no algoritmo FCH, pois geralmente essa definição encontra um ponto ou N_{centro} ajustado que não pertence ao conjunto de dados. Sendo assim, caso o ponto não pertença ao conjunto de dados, não será possível aplicar a definição de distância fuzzy, pois não existe uma aresta ponderada que liga o N_{centro} ajustado aos demais vértices.

Visando resolver esse entrave apresentamos uma definição de ajuste de N_{centro} que nomeamos como ajuste de N_{centro} fuzzy.

Definição 34 (Ajuste de N_{centro} fuzzy) *Seja $\tilde{G} < V, R >$ um grafo fuzzy em que $V = \{V_1, V_2, \dots, V_n\}$ é o conjunto dos vértices. Sejam $O'_{i1}, O'_{i2}, \dots, O'_{im}$ os $N_{centros}$ do vértice $V_i \in V, i = 1, \dots, n$; e $d(V_i, O'_{i1}) = d(V_i, O'_{i2}) = \dots = d(V_i, O'_{im}) = \phi$. O ajuste do N_{centro} fuzzy, será um ponto O_i em que:*

$$\mu_R(V_i, O_i) = \mu_R(O'_{i1}, O_i) = \mu_R(O'_{i2}, O_i) = \dots = \mu_R(O'_{im}, O_i) = \sqrt{\frac{1}{\phi}}.$$

A justificativa da escolha da função raiz quadrada é porque acreditamos que o N_{centro} ajustado precisa está fortemente conectado com os pontos $V_i, O'_{i1}, \dots, O'_{im}$. Assim, caso $\mu_R(V_i, O'_{i1}) = \mu_R(V_i, O'_{i2}) = \dots = \mu_R(V_i, O'_{im}) = 1$ então é conveniente que $\mu_R(V_1, O_i) = \mu_R(O'_{i1}, O_i) = \mu_R(O'_{i2}, O_i) = \dots = \mu_R(O'_{im}, O_i) = 1$. Por outro lado, caso $\mu_R(V_1, O'_{i1}) = \mu_R(V_1, O'_{i2}) = \dots = \mu_R(V_1, O'_{im}) = \eta$, com $\eta < 1$ então é conveniente que $\mu_R(V_1, O_i) = \mu_R(O'_{i1}, O_i) = \mu_R(O'_{i2}, O_i) = \dots = \mu_R(O'_{im}, O_i) = \omega > \eta$.

Da mesma forma que procedemos com o algoritmo FHC, para compreendermos melhor como funciona a dinâmica do algoritmo FHC usando uma distância fuzzy e o ajuste de N_{centro} fuzzy, a título de ilustração, consideraremos novamente o grafo fuzzy da Figura 12.

Notamos que nesse grafo, $\mu_R(V_1, V_2) = \mu_R(V_2, V_1) = 0.8$, $\mu_R(V_1, V_3) = \mu_R(V_3, V_1) = 0.2$, $\mu_R(V_2, V_3) = \mu_R(V_3, V_2) = 0.4$, $\mu_R(V_2, V_5) = \mu_R(V_5, V_2) = 0.4$, $\mu_R(V_3, V_4) = \mu_R(V_4, V_3) = 0.8$, $\mu_R(V_3, V_5) = \mu_R(V_5, V_3) = 0.8$, $\mu_R(V_4, V_5) = \mu_R(V_5, V_4) = 0.8$. A partir desses dados, geramos uma tabela que chamaremos de M_p destacando em cada entrada o peso entre cada par de vértices.

Tabela 2 – Pesos das arestas do Grafo da Figura 12

M_p	V_1	V_2	V_3	V_4	V_5
V_1	0.0	0.8	0.2	0.0	0.0
V_2	0.8	0.0	0.4	0.0	0.4
V_3	0.2	0.4	0.0	0.8	0.8
V_4	0.0	0.0	0.8	0.0	0.8
V_5	0.0	0.4	0.8	0.8	0.0

Fonte: Elaborada pelos autores

Conforme vimos no algoritmo e evidenciado na Figura 14, cada vértice é considerado um cluster. Para cada vinhança $Neig(V_i)$ encontraremos o N_{centro} . Dessa forma, calcularemos a distância fuzzy de V_1 em relação a todos os outros vértices. Para isso, encontraremos primeiramente o comprimento de todos os caminhos que ligam V_1 a V_2 , V_1 a V_3 , V_1 a V_4 e V_1 a V_5 . Assim, temos:

1. Quatro caminhos que conectam V_1 a V_2 . O primeiro caminho (L_1) é a própria aresta que liga V_1 a V_2 ($V_1 \rightarrow V_2$). O segundo caminho (L_2) passa pelo vértice V_3 ($V_1 \rightarrow V_3 \rightarrow V_2$). O terceiro caminho (L_3) passa pelos vértice V_3, V_4 e V_5 ($V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5 \rightarrow V_2$) e o quarto e ultimo caminho (L_4) que liga V_1 a V_2 é o caminho que passa pelos pontos V_3 a V_5 ($V_1 \rightarrow V_3 \rightarrow V_5 \rightarrow V_2$). Logo:

- $L_1 : V_1 \rightarrow V_2; \rho(L_1) = \frac{1}{0.8} = 1.25$
- $L_2 : V_1 \rightarrow V_3 \rightarrow V_2; \rho(L_2) = \frac{1}{0.8} + \frac{1}{0.4} = 7.50$
- $L_3 : V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5 \rightarrow V_2; \rho(L_3) = \frac{1}{0.2} + \frac{1}{0.8} + \frac{1}{0.8} + \frac{1}{0.4} = 10.00$

- $L_4 : V_1 \rightarrow V_3 \rightarrow V_5 \rightarrow V_2; \rho(L_4) = \frac{1}{0.2} + \frac{1}{0.8} + \frac{1}{0.4} = 8.75$

$$\delta(V_1, V_2) = \min\{1.25, 7.50, 10.00, 8.75\} = 1.25.$$

2. Quatro caminhos que conectam V_1 a V_3 . O primeiro caminho passa pelo vértice V_2 ($V_1 \rightarrow V_2 \rightarrow V_3$). O segundo caminho passa pelos vértices V_2 e V_5 ($V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_3$). O terceiro caminho passa pelos vértices V_2 , V_5 e V_4 ($V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_4 \rightarrow V_3$) e o quarto caminho é a própria aresta que liga V_1 a V_3 ($V_1 \rightarrow V_3$). Assim:

- $L_1 : V_1 \rightarrow V_2 \rightarrow V_3; \rho(L_1) = \frac{1}{0.8} + \frac{1}{0.4} = 3.75$
- $L_2 : V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_3; \rho(L_2) = \frac{1}{0.8} + \frac{1}{0.4} + \frac{1}{0.8} = 5.00$
- $L_3 : V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_4 \rightarrow V_3; \rho(L_3) = \frac{1}{0.8} + \frac{1}{0.4} + \frac{1}{0.8} + \frac{1}{0.8} = 6.25$
- $L_4 : V_1 \rightarrow V_3; \rho(L_4) = \frac{1}{0.2} = 5.00$

$$\delta(V_1, V_3) = \min\{3.75, 5.00, 6.25, 5.00\} = 3.75.$$

3. Sete caminhos que conectam V_1 a V_4 . O primeiro caminho passa pelos vértices V_2 e V_3 ($V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4$). O segundo passa pelos vértices V_2 , V_3 e V_5 ($V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_5 \rightarrow V_4$). O terceiro passa pelos vértices V_2 , V_5 e V_3 ($V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_3 \rightarrow V_4$). O quarto caminho passa pelos vértices V_2 e V_5 ($V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_4$), o quinto caminho passa pelos vértices V_3 , V_2 e V_5 ($V_1 \rightarrow V_3 \rightarrow V_2 \rightarrow V_5 \rightarrow V_4$), o sexto caminho passa pelo vértice V_3 ($V_1 \rightarrow V_3 \rightarrow V_4$) e o sétimo e último caminho passa pelos vértices V_3 e V_5 ($V_1 \rightarrow V_3 \rightarrow V_5 \rightarrow V_4$). Logo,

- $L_1 : V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4; \rho(L_1) = \frac{1}{0.8} + \frac{1}{0.4} + \frac{1}{0.8} = 5.00$
- $L_2 : V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_5 \rightarrow V_4; \rho(L_2) = \frac{1}{0.8} + \frac{1}{0.4} + \frac{1}{0.8} + \frac{1}{0.8} = 6.25$
- $L_3 : V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_3 \rightarrow V_4; \rho(L_3) = \frac{1}{0.8} + \frac{1}{0.4} + \frac{1}{0.8} + \frac{1}{0.8} = 6.25$
- $L_4 : V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_4; \rho(L_4) = \frac{1}{0.8} + \frac{1}{0.4} + \frac{1}{0.8} = 5.00$
- $L_5 : V_1 \rightarrow V_3 \rightarrow V_2 \rightarrow V_5 \rightarrow V_4; \rho(L_5) = \frac{1}{0.2} + \frac{1}{0.4} + \frac{1}{0.4} + \frac{1}{0.8} = 11.25$
- $L_6 : V_1 \rightarrow V_3 \rightarrow V_4; \rho(L_6) = \frac{1}{0.2} + \frac{1}{0.8} = 6.25$
- $L_7 : V_1 \rightarrow V_3 \rightarrow V_5 \rightarrow V_4; \rho(L_7) = \frac{1}{0.2} + \frac{1}{0.8} + \frac{1}{0.8} = 7.50$

$$\delta(V_1, V_4) = \min\{5.00, 6.25, 6.25, 5.00, 11.25, 6.25, 7.50\} = 5.00.$$

4. Seis caminhos conectam V_1 a V_5 . O primeiro caminho passa pelos pontos V_2 , V_3 e V_4 ($V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5$). O segundo caminho passa pelos pontos V_2 e V_3 ($V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_5$), o terceiro passa pelo ponto V_2 ($V_1 \rightarrow V_2 \rightarrow V_5$), o quarto passa pelos vértices V_3 e V_2 ($V_1 \rightarrow V_3 \rightarrow V_2 \rightarrow V_5$), o quinto passa pelos vértices V_3 e V_4 ($V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5$) e o sexto caminho passa pelo vértice V_3 ($V_1 \rightarrow V_3 \rightarrow V_5$). Dessa forma:

- $L_1 : V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5; \rho(L_1) = \frac{1}{0.8} + \frac{1}{0.4} + \frac{1}{0.8} + \frac{1}{0.8} = 6.25$
- $L_2 : V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_5; \rho(L_2) = \frac{1}{0.8} + \frac{1}{0.4} + \frac{1}{0.8} = 5.00$
- $L_3 : V_1 \rightarrow V_2 \rightarrow V_5; \rho(L_3) = \frac{1}{0.8} + \frac{1}{0.4} = 3.75$
- $L_4 : V_1 \rightarrow V_3 \rightarrow V_2 \rightarrow V_5; \rho(L_4) = \frac{1}{0.2} + \frac{1}{0.4} + \frac{1}{0.4} = 10.00$
- $L_5 : V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5; \rho(L_5) = \frac{1}{0.2} + \frac{1}{0.8} + \frac{1}{0.8} = 7.50$
- $L_6 : V_1 \rightarrow V_3 \rightarrow V_5; \rho(L_6) = \frac{1}{0.2} + \frac{1}{0.8} = 6.25$

$$\delta(V_1, V_5) = \min\{6.25, 5.00, 3.75, 10.00, 7.50, 6.25\} = 3.75.$$

Nesse sentido, como $\delta(V_1, V_2) = 1.25$, $\delta(V_1, V_3) = 3.75$, $\delta(V_1, V_4) = 5.00$ e $\delta(V_1, V_5) = 3.75$, o vértice mais próximo de V_1 é V_2 , logo $N_{centro}(Neig(V_1)) = V_2$. Como $\delta(V_1, V_2) = 1.25 < 3.1 = r$, o vértice V_1 pertence ao subcluster SC_1 e o N_{centro} ajustado será um ponto O_1 em que o grau da aresta que liga O_1 a V_1 e O_1 a V_2 será $\sqrt{\frac{1}{1.25}} \approx 0.89$. Assim, calculando a distância fuzzy do centro ajustado a todos os pontos, encontramos:

$$\delta(O_1, V_1) \approx 1.12 < r;$$

$$\delta(O_1, V_2) \approx 1.12 < r;$$

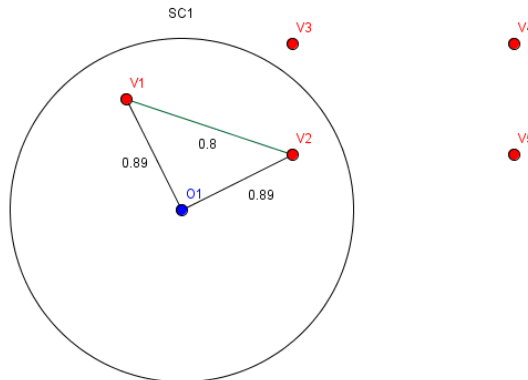
$$\delta(O_1, V_3) \approx 3.62 \geq r;$$

$$\delta(O_1, V_4) \approx 4.87 \geq r;$$

$$\delta(O_1, V_5) \approx 3.62 \geq r.$$

Dessa forma, o subcluster SC_1 é composto pelos vértices V_1 e V_2 como podemos observar na Figura abaixo.

Figura 23 – Subcluster SC_1 do grafo $\widetilde{G'}$ usando distância fuzzy



Fonte: Elaborada pelos autores

Analogamente, para encontrarmos o N_{centro} de $Neig(V_2)$, devemos determinar $\delta(V_2, V_1)$, $\delta(V_2, V_3)$, $\delta(V_2, V_4)$ e $\delta(V_2, V_5)$. Dessa forma, encontraremos todos os caminhos que ligam V_2 a V_1 , V_2 a V_3 , V_2 a V_4 e V_2 a V_5 . Assim, temos:

1. Quatro caminhos que conectam V_2 a V_1 . Esses caminhos são equivalentes aos que destacamos acima quando encontramos os caminhos que conectam V_1 a V_2 . Como a distância fuzzy é uma métrica, segue que $\delta(V_2, V_1) = \delta(V_1, V_2) = 1.25$.
2. Quatro caminhos que conectam V_2 a V_3 . O primeiro caminho passa pelo vértice V_1 ($V_2 \rightarrow V_1 \rightarrow V_3$). O segundo caminho é a própria aresta que conecta V_2 a V_3 ($V_2 \rightarrow V_3$). O terceiro caminho passa pelo vértice V_5 ($V_2 \rightarrow V_5 \rightarrow V_3$) e o quarto caminho passa pelos vértices V_5 e V_4 ($V_2 \rightarrow V_5 \rightarrow V_4 \rightarrow V_3$). Assim:

- $L_1 : V_2 \rightarrow V_1 \rightarrow V_3; \rho(L_1) = \frac{1}{0.8} + \frac{1}{0.2} = 6.25$
- $L_2 : V_2 \rightarrow V_3; \rho(L_2) = \frac{1}{0.4} = 2.50$
- $L_3 : V_2 \rightarrow V_5 \rightarrow V_3; \rho(L_3) = \frac{1}{0.4} + \frac{1}{0.8} = 3.75$
- $L_4 : V_2 \rightarrow V_5 \rightarrow V_4 \rightarrow V_3; \rho(L_4) = \frac{1}{0.4} + \frac{1}{0.8} + \frac{1}{0.8} = 5.00$

$$\delta(V_2, V_3) = \min\{6.25, 2.50, 3.75, 5.00\} = 2.50.$$

3. Seis caminhos que conectam V_2 a V_4 . O primeiro caminho passa pelos vértices V_1 e V_3 ($V_2 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4$). O segundo passa pelos vértices V_1 , V_3 e V_5 ($V_2 \rightarrow V_1 \rightarrow V_3 \rightarrow V_5 \rightarrow V_4$). O terceiro passa pelo vértice V_3 ($V_2 \rightarrow V_3 \rightarrow V_4$). O quarto caminho passa pelos vértices V_3 e V_5 ($V_2 \rightarrow V_3 \rightarrow V_5 \rightarrow V_4$), o quinto caminho passa pelos vértices V_5 e V_3 ($V_2 \rightarrow V_5 \rightarrow V_3 \rightarrow V_4$) e o sexto caminho passa pelo vértice V_5 ($V_2 \rightarrow V_5 \rightarrow V_4$). Logo,

- $L_1 : V_2 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4; \rho(L_1) = \frac{1}{0.8} + \frac{1}{0.2} + \frac{1}{0.8} = 7.50$
- $L_2 : V_2 \rightarrow V_1 \rightarrow V_3 \rightarrow V_5 \rightarrow V_4; \rho(L_2) = \frac{1}{0.8} + \frac{1}{0.2} + \frac{1}{0.8} + \frac{1}{0.8} = 8.75$
- $L_3 : V_2 \rightarrow V_3 \rightarrow V_4; \rho(L_3) = \frac{1}{0.4} + \frac{1}{0.8} = 3.75$
- $L_4 : V_2 \rightarrow V_3 \rightarrow V_5 \rightarrow V_4; \rho(L_4) = \frac{1}{0.4} + \frac{1}{0.8} + \frac{1}{0.8} = 5.00$
- $L_5 : V_2 \rightarrow V_5 \rightarrow V_3 \rightarrow V_4; \rho(L_5) = \frac{1}{0.4} + \frac{1}{0.8} + \frac{1}{0.8} = 5.00$
- $L_6 : V_2 \rightarrow V_5 \rightarrow V_4; \rho(L_6) = \frac{1}{0.4} + \frac{1}{0.8} = 3.75$

$$\delta(V_2, V_4) = \min\{7.50, 8.75, 3.75, 5.00, 5.00, 3.75\} = 3.75.$$

4. Cinco caminhos conectam V_2 a V_5 . O primeiro caminho passa pelos pontos V_1 , V_3 e V_4 ($V_2 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5$). O segundo caminho passa pelos pontos V_1 e V_3 ($V_2 \rightarrow V_1 \rightarrow V_3 \rightarrow V_5$), o terceiro passa pelos pontos V_3 e V_4 ($V_2 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5$), o quarto passa pelo vértice V_3 ($V_2 \rightarrow V_3 \rightarrow V_5$) e o quinto caminho é a própria aresta que conecta V_2 a V_5 ($V_2 \rightarrow V_5$). Dessa forma:

- $L_1 : V_2 \rightarrow V_1 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5; \rho(L_1) = \frac{1}{0.8} + \frac{1}{0.2} + \frac{1}{0.8} + \frac{1}{0.8} = 8.75$
- $L_2 : V_2 \rightarrow V_1 \rightarrow V_3 \rightarrow V_5; \rho(L_2) = \frac{1}{0.8} + \frac{1}{0.2} + \frac{1}{0.8} = 7.50$

- $L_3 : V_2 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5; \rho(L_3) = \frac{1}{0.4} + \frac{1}{0.8} + \frac{1}{0.8} = 5.00$
- $L_4 : V_2 \rightarrow V_3 \rightarrow V_5; \rho(L_4) = \frac{1}{0.4} + \frac{1}{0.8} = 3.75$
- $L_5 : V_2 \rightarrow V_5; \rho(L_5) = \frac{1}{0.4} = 2.50$

$$\delta(V_2, V_5) = \min\{8.75, 7.50, 5.00, 3.75, 2.50\} = 2.50.$$

Como $\delta(V_2, V_1) = 1.25$, $\delta(V_2, V_3) = 2.50$, $\delta(V_2, V_4) = 3.75$ e $\delta(V_2, V_5) = 2.50$, o vértice mais próximo de V_2 é V_1 , logo $N_{centro}(Neig(V_2)) = V_1$. Como $\delta(V_2, V_1) = 1.25 < 3.1 = r$, o N_{centro} ajustado é um ponto $O_2=O_1$ em que o grau da aresta que liga O_2 a V_2 e O_2 a V_1 será $\sqrt{\frac{1}{1.25}} \approx 0.89$. Dessa forma, calculando a distância fuzzy do N_{centro} ajustado a todos os pontos, encontramos:

$$\begin{aligned} \delta(O_2, V_1) &= \delta(O_1, V_1) \approx 1.12 < r; \\ \delta(O_2, V_2) &= \delta(O_1, V_2) \approx 1.12 < r; \\ \delta(O_2, V_3) &= \delta(O_1, V_3) \approx 3.62 \geq r; \\ \delta(O_2, V_4) &= \delta(O_1, V_4) \approx 4.87 \geq r; \\ \delta(O_2, V_5) &= \delta(O_1, V_5) \approx 3.62 \geq r. \end{aligned}$$

Assim, o subcluster encontrado é o mesmo SC_1 destacado na Figura 23. Agora, para encontrarmos o N_{centro} de $Neig(V_3)$ devemos determinar $\delta(V_3, V_1)$, $\delta(V_3, V_2)$, $\delta(V_3, V_4)$ e $\delta(V_3, V_5)$. Dessa forma, encontraremos todos os caminhos que ligam V_3 a V_1 , V_3 a V_2 , V_3 a V_4 e V_3 a V_5 . Nesse contexto, temos que:

1. Os caminhos que conectam V_3 a V_1 são equivalentes aos que conectam V_1 a V_3 . Assim: $\delta(V_3, V_1)=\delta(V_1, V_3) = 3.75$.
2. Os caminhos que conectam V_3 a V_2 são equivalentes aos que conectam V_2 a V_3 . Logo, $\delta(V_3, V_2)=\delta(V_2, V_3) = 2.50$.
3. Existem quatro caminhos que conectam V_3 a V_4 . O primeiro caminho passa pelos vértices V_1, V_2 e V_5 ($V_3 \rightarrow V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_4$). O segundo passa pelos vértices V_2 e V_5 ($V_3 \rightarrow V_2 \rightarrow V_5 \rightarrow V_4$). O terceiro caminho é a aresta que liga V_3 a V_4 ($V_3 \rightarrow V_4$) e o quarto caminho passa pelo vértice V_5 ($V_3 \rightarrow V_5 \rightarrow V_4$). Dessa forma:

- $L_1 : V_3 \rightarrow V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_4; \rho(L_1) = \frac{1}{0.2} + \frac{1}{0.8} + \frac{1}{0.4} + \frac{1}{0.8} = 10.00$
- $L_2 : V_3 \rightarrow V_2 \rightarrow V_5 \rightarrow V_4; \rho(L_2) = \frac{1}{0.4} + \frac{1}{0.4} + \frac{1}{0.8} = 6.25$
- $L_3 : V_3 \rightarrow V_4; \rho(L_3) = \frac{1}{0.8} = 1.25$
- $L_4 : V_3 \rightarrow V_5 \rightarrow V_4; \rho(L_4) = \frac{1}{0.8} + \frac{1}{0.8} = 2.50$

$$\delta(V_3, V_4) = \min\{10.00, 6.25, 1.25, 2.50\} = 1.25.$$

4. Existem quatro caminhos que conectam V_3 a V_5 . O primeiro caminho passa pelos pontos V_1 e V_2 ($V_3 \rightarrow V_1 \rightarrow V_2 \rightarrow V_5$). O segundo caminho passa pelo ponto V_2 ($V_3 \rightarrow V_2 \rightarrow V_5$), o terceiro passa pelo ponto v_4 ($V_3 \rightarrow V_4 \rightarrow V_5$) e o quarto caminho é a aresta que conecta V_3 a V_5 ($V_3 \rightarrow V_5$). Dessa forma:

- $L_1 : V_3 \rightarrow V_1 \rightarrow V_2 \rightarrow V_5; \rho(L_1) = \frac{1}{0.2} + \frac{1}{0.8} + \frac{1}{0.4} = 8.75$
- $L_2 : V_3 \rightarrow V_2 \rightarrow V_5; \rho(L_2) = \frac{1}{0.4} + \frac{1}{0.4} = 5.00$
- $L_3 : V_3 \rightarrow V_4 \rightarrow V_5; \rho(L_3) = \frac{1}{0.8} + \frac{1}{0.8} = 2.50$
- $L_4 : V_3 \rightarrow V_5; \rho(L_4) = \frac{1}{0.8} = 1.25$

$$\delta(V_3, V_5) = \min\{8.75, 5.00, 2.50, 1.25\} = 1.25.$$

Nesse sentido, como $\delta(V_3, V_1) = 3.75$, $\delta(V_3, V_2) = 2.50$, $\delta(V_3, V_4) = 1.25$ e $\delta(V_3, V_5) = 1.25$, então os vértices mais próximos de V_3 são V_4 e V_5 . Como $\delta(V_3, V_4) = \delta(V_3, V_5) = 1.25 < 3.1 = r$, o vértice V_3 pertence ao subcluster SC_2 de N_{centro} ajustado no ponto O_3 em que o grau da aresta que liga O_3 a V_3 , O_3 e V_4 e O_3 e V_5 será $\sqrt{\frac{1}{1.25}} \approx 0.89$. Assim, calculando a distância fuzzy do N_{centro} ajustado a todos os pontos, encontramos:

$$\delta(O_3, V_1) \approx 4.87 \geq r;$$

$$\delta(O_3, V_2) \approx 3.62 \geq r;$$

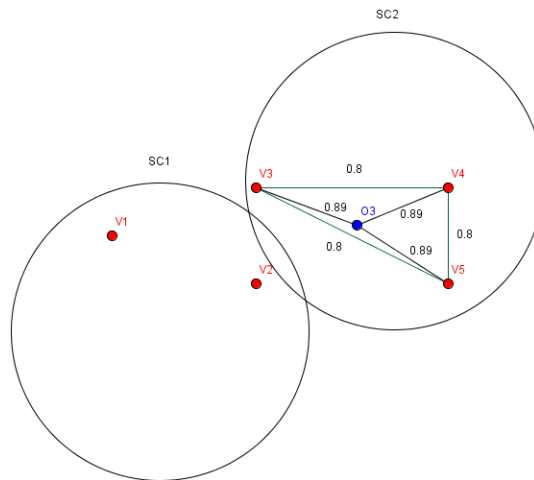
$$\delta(O_3, V_3) \approx 1.12 < r;$$

$$\delta(O_3, V_4) \approx 1.12 < r;$$

$$\delta(O_3, V_5) \approx 1.12 < r.$$

Dessa forma, o subcluster SC_2 é composto pelos vértices V_3 , V_4 e V_5 como podemos observar na Figura abaixo.

Figura 24 – Subcluster SC_2 do grafo $\widetilde{G'}$ usando distância fuzzy



Fonte: Elaborada pelos autores

Procedemos da mesma forma para encontrarmos os $Ncentros$ de $Neig(V_4)$ e $Neig(V_5)$. Para descobrir o $Ncentro$ de V_4 , temos que determinar $\delta(V_4, V_1)$, $\delta(V_4, V_2)$, $\delta(V_4, V_3)$ e $\delta(V_4, V_5)$. Ora, $\delta(V_4, V_1) = \delta(V_1, V_4) = 5,00$, $\delta(V_4, V_2) = \delta(V_2, V_4) = 3.75$, $\delta(V_4, V_3) = \delta(V_3, V_4) = 1.25$ e $\delta(V_4, V_5) = 1.25$, pois existem quatro caminhos que ligam V_4 a V_5 e o menor comprimento vale 1.25, a saber:

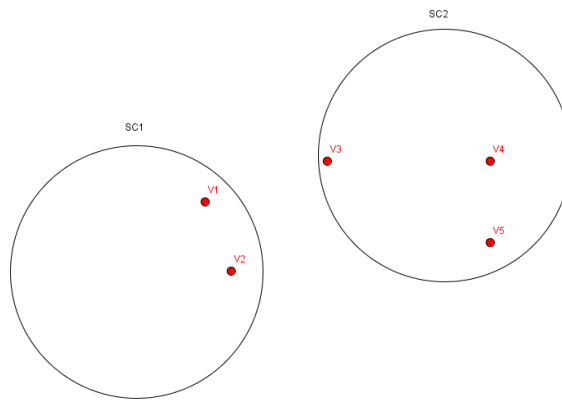
- $L_1 : V_4 \rightarrow V_3 \rightarrow V_1 \rightarrow V_2 \rightarrow V_5; \rho(L_1) = \frac{1}{0.8} + \frac{1}{0.2} + \frac{1}{0.8} + \frac{1}{0.4} = 10.00$
- $L_2 : V_4 \rightarrow V_3 \rightarrow V_2 \rightarrow V_5; \rho(L_2) = \frac{1}{0.8} + \frac{1}{0.4} + \frac{1}{0.4} = 6.25$
- $L_3 : V_4 \rightarrow V_3 \rightarrow V_5; \rho(L_3) = \frac{1}{0.8} + \frac{1}{0.8} = 2.50$
- $L_4 : V_4 \rightarrow V_5; \rho(L_4) = \frac{1}{0.8} = 1.25$.

Além disso, para descobrir o $Ncentro$ de $Neig(V_5)$, temos que determinar $\delta(V_5, V_1)$, $\delta(V_5, V_2)$, $\delta(V_5, V_3)$ e $\delta(V_5, V_4)$. Entretanto, estas já foram descobertos nas etapas anteriores, pois $\delta(V_5, V_1) = \delta(V_1, V_5) = 3.75$, $\delta(V_5, V_2) = \delta(V_2, V_5) = 2.50$, $\delta(V_5, V_3) = \delta(V_3, V_5) = 1.25$ e $\delta(V_5, V_4) = \delta(V_4, V_5) = 1.25$.

Nesse âmbito, os pontos mais próximos de V_4 são V_3 e V_5 e os pontos mais próximos de V_5 são V_3 e V_4 . Assim, os $Ncentros$ ajustados O_4 e O_5 serão equivalentes a O_3 , ou seja, $O_4=O_5=O_3$. Dessa forma, o subcluster encontrado é o mesmo SC_2 destacado na Figura 24 para ambos os casos.

De acordo com a Figura 24, notamos que não há vértice de conexão entre as vizinhanças SC_1 e SC_2 . Assim, o grau de conexidade fuzzy entre essas vizinhanças é zero. Diante dessas informações, geramos o grafo fuzzy de subclusters em que SC_1 e SC_2 são os vértices do grafo, conforme Figura 25.

Figura 25 – Grafo fuzzy de subclusters $\widetilde{G'}$ usando distância fuzzy



Fonte: Elaborada pelos autores

O parâmetro λ não influencia, nesse caso, no resultado da clusterização. Para qualquer λ , o G'_λ será conforme a Figura 25 e o resultado final sempre será dois clusters:

$C_1 = \{V_1, V_2\}$ e $C_2 = \{V_3, V_4, V_5\}$. Dessa forma, notamos que nesse exemplo a distância fuzzy colocou no mesmo cluster os dados que tinham um grau de similaridade maior, pois $\mu(V_1, V_2) = 0.8$, assim como $\mu(V_3, V_4) = \mu(V_3, V_5) = \mu(V_4, V_5) = 0.8$.

Diante disso, apresentamos a seguir, uma proposta de um algoritmo com a essência do FCH, entretanto utilizando a medida de similaridade fuzzy e o ajuste de N_{centro} fuzzy. Chamaremos esse algoritmo de FHC-Modificado ou simplesmente FHC-M.

Algoritmo FHC-M

Entrada: M_p , o raio r dos subclusters e o parâmetro λ

Saída: Componentes conexas do grafo λ -corte

Algoritmo:

1. Cada ponto é um cluster
 2. Para cada ponto $V_i \in V$
 3. { Encontrar o N_{centro} O'_i fuzzy
 4. Se $(\delta(V_i, O'_i) < r)$
 5. $\{V_i$ pertence ao subcluster $SC_j : SC_j = SC_j \cup V_i$ em que o centro é o N_{centro} ajustado fuzzy O_i
 6. Se não
 7. {Criar um novo subcluter contendo apenas V_i e V_i é o centro do subcluster}
 8. }
 9. Para cada ponto $V_i \in V$
 10. {Para cada subcluster $SC_j \in SC$
 11. Se $(V_i$ no subcluster $SC_j)$ fazer uma marcação neste subcluster SC_j
 12. // Construir um grafo fuzzy dos subclusters e então calcular o grau de conexidade fuzzy entre as vizinhanças
 13. Calcular o grau de conexidade fuzzy entre as vizinhanças
 14. Gerar o λ _Grafo(G')
-

Para validar a eficiência dessa modificação realizaremos uma avaliação experimental, na qual aplicaremos os algoritmos FHC e FHC-M para avaliar e comparar os resultados. O hardware utilizado foi um notebook com processador Intel Core i5 4210U, 2.4 GHz, e 4GB de memória ram. O sistema operacional utilizado foi o Windows 8.1 64bits.

3.3 Avaliação Experimental

Vimos na fundamentação teórica que a clusterização pode ser empregada para reduzir, minerar e analisar dados, reconhecer padrões, processar e segmentar imagens, realizar pesquisa de mercado, agrupar documentos, agrupar cidades, estudar dados de genoma na Biologia, dentre muitas outras. Nesse âmbito, selecionamos dois desses fenômenos, a saber, agrupamento de cidades de acordo com o Índice de Desenvolvimento

Humano Municipal (IDHM) e segmentação de imagens, para aplicar os algoritmos e analisar o seus comportamentos.

3.3.1 Agrupamento de Cidades de Acordo com IDHM

O Índice de Desenvolvimento Humano (IDH) foi apresentado em 1990 no primeiro relatório de desenvolvimento humano do Programa das Nações Unidas para o Desenvolvimento (PNUD). Idealizado pelo economista paquistanês Mahbub ul Haq com a colaboração do economista Amartya Sen, o IDH reúne três dos requisitos fundamentais para a expansão das liberdades das pessoas: saúde, educação e renda (PNUD, 2012).

Em 1998, o Brasil foi um dos primeiros países a adaptar e calcular um IDH para todos os seus municípios, criando o Índice de Desenvolvimento Humano Municipal (IDHM). Em 2012 o Programa das Nações Unidas para o Desenvolvimento (PNUD), o Instituto de Pesquisa Econômica Aplicada (Ipea) e a Fundação João Pinheiro (FJP) comprometeram-se a adaptar a metodologia do IDH Global para calcular o IDH Municipal de todos os municípios brasileiros. Esse cálculo foi realizado a partir das informações dos 3 últimos Censos Demográficos do IBGE: 1991, 2000 e 2010. Em razão desta adaptação metodológica, os valores do IDHM, subíndices e indicadores para 2000 e 1991 foram recalculados (PNUD, 2013).

O IDHM brasileiro utiliza as mesmas três dimensões do IDH Global. Além disso, ajusta a metodologia global ao contexto brasileiro e os indicadores levados em conta no IDHM são mais adequados para avaliar o desenvolvimento dos municípios brasileiros. O cálculo do IDHM pode ser observado na Figura 26.

O IDHM longevidade é medido pela expectativa de vida ao nascer, calculada por método indireto e mostra o número médio de anos que uma pessoa nascida em determinado lugar viveria a partir do nascimento, mantidos os mesmos padrões de mortalidade (PNUD, 2013).

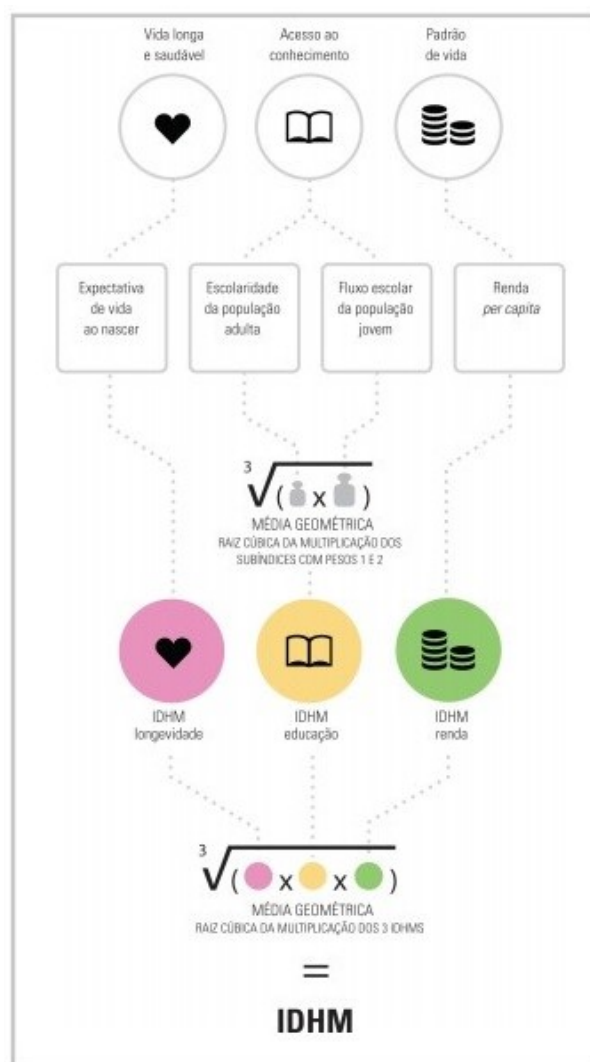
O IDHM educação considera escolaridade da população adulta, medida pelo percentual de pessoas de 18 anos ou mais de idade com ensino fundamental completo com peso 1 e a adequação do fluxo escolar da população jovem, medida pela média aritmética do percentual de crianças de 5 a 6 anos frequentando a escola, do percentual de jovens de 11 a 13 anos frequentando os anos finais do ensino fundamental, do percentual de jovens de 15 a 17 anos com ensino fundamental completo e do percentual de jovens de 18 a 20 anos com ensino médio completo com peso 2. A média geométrica desses dois componentes resulta no IDHM Educação (PNUD, 2013).

O IDHM renda é a soma da renda de todos os residentes, dividida pelo número de pessoas que moram no município, inclusive crianças e pessoas sem registro de renda

(PNUD, 2013). Os dados para o cálculo de cada componente acima são dos Censos Demográficos do IBGE.

Após encontrar o IDH longevidade, IDH educação e IDH renda, é realizado uma média geométrica desses valores, resultando no IDHM que é um número que varia entre 0 e 1. Quanto mais próximo de 1, maior o desenvolvimento do município.

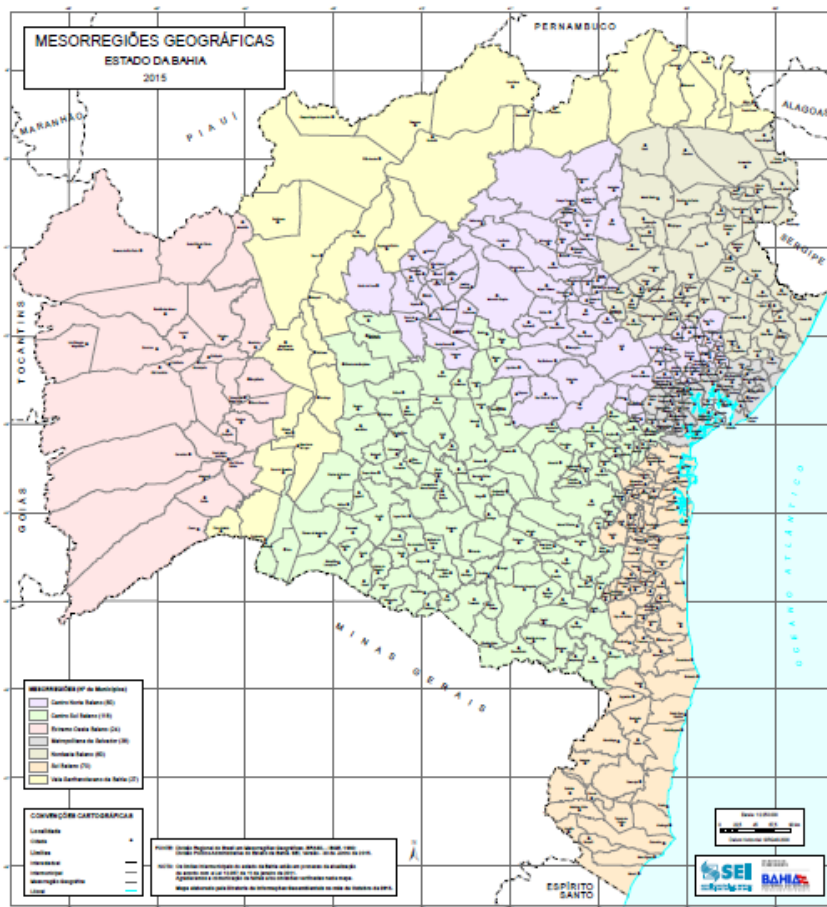
Figura 26 – Cálculo do IDHM atualmente



Fonte: PNUD (2013)

De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE), no Brasil existe um total de 5.570 municípios distribuídos nas 27 unidades da federação. Desse total, 1794 municípios estão espalhados pelos 9 estados da região nordeste e 417 estão no estado da Bahia. Esses 417 municípios estão compreendidos em 7 mesorregiões (ver Figura 27), a saber: Centro Norte Baiano com 80 municípios, Centro Sul Baiano com 118, Extremo Oeste Baiano com 24, Metropolitana de Salvador com 38, Nordeste Baiano com 60, Vale Sanfranciscano da Bahia com 27 e Sul Baiano com 70 municípios.

Figura 27 – Mesorregiões da Bahia



Fonte: Diretoria de Informações Geoambientais (2015)

Dentre as 7 mesorregiões destacadas no mapa, escolhemos os municípios da mesorregião sul baiano (em laranja) para realizar o agrupamento conforme o IDHM. Esta mesorregião está dividida em 3 microrregiões: Ilhéus-Itabuna com 41 municípios, Porto Seguro com 19 municípios e Valença com 10 municípios.

Com base nessas 70 municípios, criamos um grafo fuzzy. Para cada um dos municípios associamos um vértice $V_i, i = 1, \dots, 70$. Cada vértice está conectado, por meio de uma aresta, com todos os outros vértices e o grau de cada aresta foi obtido por meio da função:

$$P(V_i, V_j) = 1 - |IDHM_{V_i} - IDHM_{V_j}|, V_i \neq V_j, \quad (6)$$

onde $IDHM_{V_i}$ e $IDHM_{V_j}$ é o valor correspondente ao IDHM do vértice V_i e V_j , respectivamente. Essa função sempre retornará um valor no intervalo $[0, 1]$. E quanto menor $|IDHM_{V_i} - IDHM_{V_j}|$ maior será o peso da aresta. Dessa forma, o grafo tem total de 70 vértice e 2415 arestas.

Para aplicarmos o algoritmo FHC, convertemos as coordenadas geográficas de cada municípios (latitude, longitude) em coordenadas cartesianas aproximadas

(x, y). Essas coordenadas geográficas foram obtidas no site "www.geografos.com.br". Esses dados podem ser observados nas tabelas seguintes, onde destacamos na primeira coluna as cidades em ordem alfabética, na segunda, o vértice associado a cada cidade, na terceira, o valor aproximado da coordenada no eixo-x e na quarta coluna o valor aproximado da coordenada no eixo-y. Devido à magnitude, organizamos as cidades em três tabelas de acordo com cada microrregião.

Tabela 3 – Municípios da microrregião de Porto Seguro

Município	Vértice	IDHM	Eixo-x	Eixo-y
Alcobaça	V_1	0.608	-4354.626222	-1946.410278
Caravelas	V_{13}	0.616	-4362.434083	-1970.019028
Eunápolis	V_{15}	0.677	-4397.368861	-1819.540250
Guaratinga	V_{20}	0.558	-4419.866611	-1842.470056
Ibirapuã	V_{23}	0.614	-4456.128417	-1965.112111
Itabela	V_{28}	0.599	-4394.375333	-1841.482500
Itagimirim	V_{32}	0.634	-4401.164778	-1787.228667
Itamaraju	V_{35}	0.627	-4391.906444	-1893.051417
Itanhém	V_{37}	0.637	-4480.663000	-1907.185806
Jucuruçu	V_{42}	0.541	-4461.652556	-1871.232611
Lajedão	V_{44}	0.632	-4482.298639	-1956.872194
Medeiros Neto	V_{47}	0.625	-4468.503722	-1930.239056
Mucuri	V_{48}	0.665	-4394.097583	-2009.397806
Nova Viçosa	V_{51}	0.654	-4374.223028	-1987.795028
Porto Seguro	V_{54}	0.676	-4340.090639	-1827.564139
Prado	V_{55}	0.621	-4357.434583	-1926.597444
Santa Cruz Cabrália	V_{57}	0.654	-4335.646639	-1808.491972
Teixeira de Freitas	V_{62}	0.685	-4415.330028	-1948.138500
Vereda	V_{69}	0.577	-4453.443500	-1913.419750

Fonte: Elaborada pelos autores

Tabela 4 – Municípios da microrregião de Valença

Município	Vértice	IDHM	Eixo-x	Eixo-y
Cairu	V_9	0.627	-4337.776056	-1498.399528
Camamu	V_{11}	0.565	-4344.442056	-1549.258639
Igrapiúna	V_{25}	0.574	-4348.700889	-1536.111806
Ituberá	V_{41}	0.606	-4349.472417	-1525.649889
Maraú	V_{45}	0.593	-4334.535639	-1566.849472
Nilo Peçanha	V_{49}	0.547	-4344.781528	-1510.898278
Piraí do Norte	V_{53}	0.533	-4374.994556	-1528.952028
Presidente Tancredo Neves	V_{56}	0.559	-4379.716306	-1494.140694
Taperoá	V_{61}	0.566	-4343.855694	-1504.077972
Valença	V_{68}	0.623	-4341.016472	-1485.437861

Fonte: Elaborada pelos autores

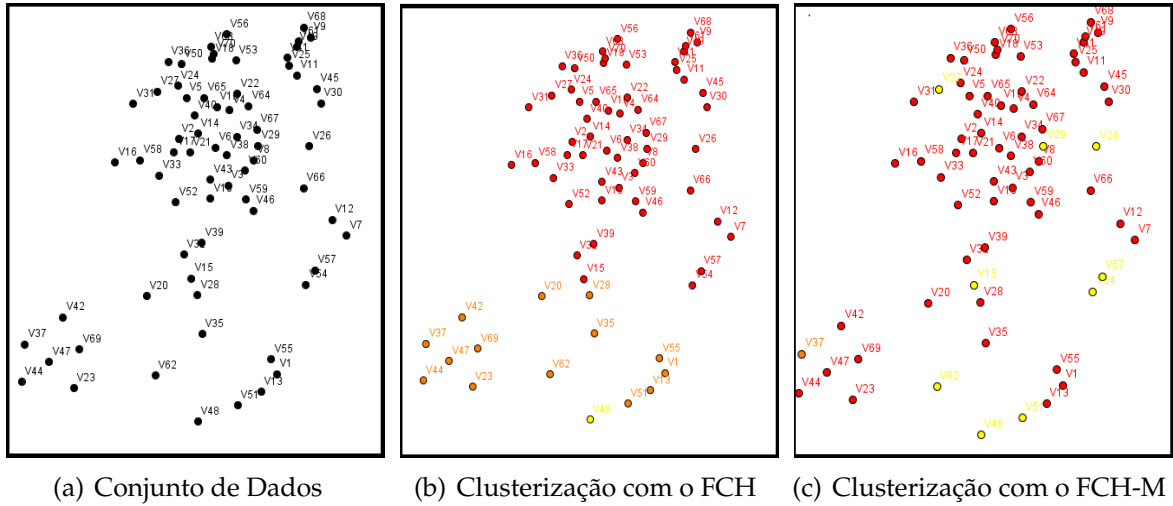
Tabela 5 – Municípios da microrregião de Itabuna-Ilhéus

Município	Vértice	IDHM	Eixo-x	Eixo-y
Almadina	V_2	0.563	-4403.695389	-1633.756361
Arataca	V_3	0.559	-4378.944778	-1695.756333
Aurelino Leal	V_4	0.568	-4378.296694	-1595.488583
Barra do Rocha	V_5	0.577	-4399.776028	-1578.792722
Barro Preto	V_6	0.602	-4385.271306	-1645.360139
Belmonte	V_7	0.598	-4319.876611	-1762.385472
Buerarema	V_8	0.613	-4366.199139	-1661.994278
Camacan	V_{10}	0.581	-4387.987083	-1713.069417
Canavieiras	V_{12}	0.590	-4327.036389	-1741.492500
Coaraci	V_{14}	0.613	-4394.128444	-1626.596583
Firmino Alves	V_{16}	0.578	-4435.513194	-1664.864361
Floresta Azul	V_{17}	0.557	-4406.195139	-1650.946000
Gandu	V_{18}	0.632	-4386.968667	-1526.946056
Gongogi	V_{19}	0.576	-4384.530639	-1591.198889
Ibicaraí	V_{21}	0.625	-4398.171250	-1651.501500
Ibirapitanga	V_{22}	0.558	-4374.408194	-1573.638917
Ibirataia	V_{24}	0.576	-4404.065722	-1562.837528
Ilhéus	V_{26}	0.690	-4338.393278	-1643.045556
Ipiaú	V_{27}	0.670	-4414.435056	-1570.645389
Itabuna	V_{29}	0.712	-4364.038861	-1642.675222
Itacaré	V_{30}	0.583	-4332.529667	-1586.230250
Itagibá	V_{31}	0.589	-4426.532611	-1586.909194
Itaju do Colônia	V_{33}	0.592	-4413.385778	-1682.331750
Itajuípe	V_{34}	0.599	-4374.562500	-1630.731972
Itamari	V_{36}	0.578	-4408.849194	-1530.741972
Itapé	V_{38}	0.599	-4379.685444	-1655.204833
Itapebi	V_{39}	0.572	-4392.215056	-1772.137583
Itapitanga	V_{40}	0.571	-4395.671500	-1602.370611
Jussari	V_{43}	0.567	-4387.925361	-1687.763306
Mascote	V_{46}	0.581	-4366.507750	-1729.055472
Nova Ibiá	V_{50}	0.570	-4402.399222	-1534.291000
Pau Brasil	V_{52}	0.583	-4405.238444	-1718.068917
Santa Cruz da Vitória	V_{58}	0.610	-4423.076167	-1662.179444
Santa Luzia	V_{59}	0.556	-4370.025917	-1714.211278
São José da Vitória	V_{60}	0.546	-4370.581417	-1675.881778
Teolândia	V_{63}	0.555	-4387.493306	-1511.145167
Ubaitaba	V_{64}	0.611	-4368.822333	-1590.118750
Ubatã	V_{65}	0.593	-4390.980611	-1579.163056
Una	V_{66}	0.560	-4341.263361	-1699.089333
Uruçuca	V_{67}	0.616	-4364.501778	-1621.288472
Wenceslau Guimarães	V_{70}	0.544	-4386.166278	-1520.619528

Fonte: Elaborada pelos autores

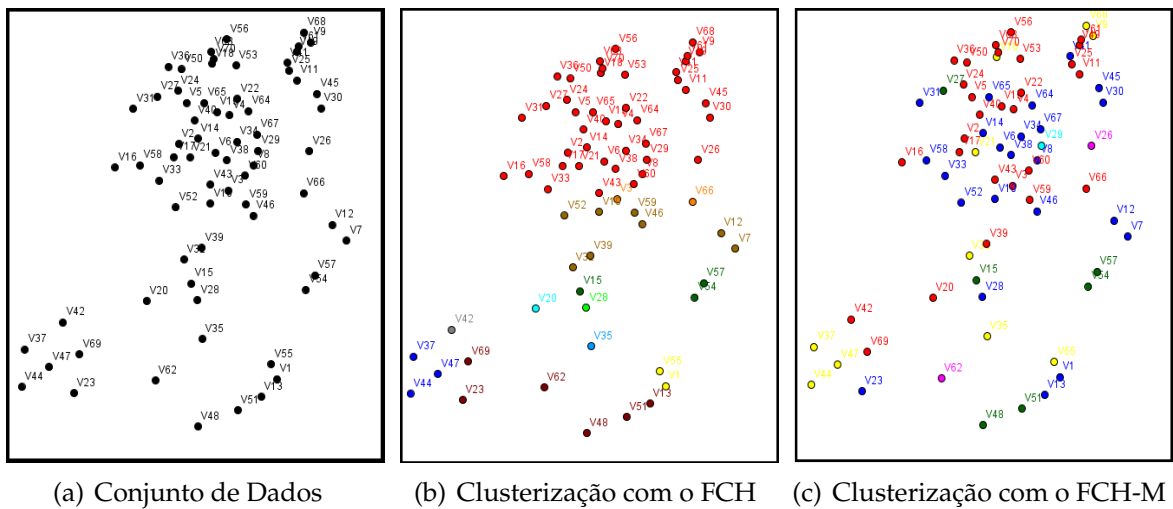
A partir das informações acima citadas, aplicamos o algoritmo FHC e FHC-M. Escolhemos dois parâmetros λ aleatórios, a saber, $\lambda_1 = 0.2$ e $\lambda_2 = 0.5$ e obtemos os seguintes resultados:

Figura 28 – Resultados para $\lambda = 0.2$



Fonte: Elaboradas pelos autores

Figura 29 – Resultados para $\lambda = 0.5$



Fonte: Elaboradas pelos autores

Na Figura 28 temos em (a) o conjunto de dados, em (b) o resultado da clusterização usando o FHC e em (c) usando o FHC-M. O resultado da clusterização considerando um parâmetro $\lambda = 0.2$ gerou dois clusters para ambos os casos. Os objetos em vermelho pertencem somente ao cluster C_1 , os objetos em amarelo pertencem somente ao cluster C_2 e os objetos em laranja pertencem aos dois clusters.

Na Figura 29, da mesma forma, temos em (a) o conjunto de dados, em (b) o resultado da clusterização usando o FHC e em (c) o resultado usando o FHC-M. Para $\lambda = 0.5$, o algoritmo FHC nos forneceu cinco clusters, enquanto o FCH-M quatro clusters.

Com relação ao resultado da clusterização da Figura 29 usando o algoritmo FCH, os objetos em vermelho pertencem somente ao cluster C_1 , os objetos em amarelo pertencem somente ao cluster C_2 e os que estão em azul pertencem somente ao cluster C_5 . Alguns objetos pertencem a mais de um cluster. Os que estão de:

- Marrom escuro pertencem aos clusters C_1 e C_5 ;
- Laranja pertencem aos clusters C_2 e C_3 ;
- Verde escuro pertencem aos clusters C_3 e C_4 ;
- Cinza pertencem aos clusters C_4 e C_5 ;
- Verde claro pertencem aos clusters C_1 , C_3 e C_4 ;
- Azul claro pertencem aos clusters C_1 , C_4 e C_5 ;
- Marrom claro pertencem aos clusters C_2 , C_3 e C_4 ;
- Ciano pertencem aos clusters C_3 , C_4 e C_5 .

Referindo ao resultado da clusterização da Figura 29 usando o algoritmo FCH-M, os objetos em vermelho pertencem somente ao cluster C_1 , os objetos em amarelo pertencem somente ao cluster C_2 , os de verde escuro somente ao cluster C_3 e o que está em ciano pertence somente ao cluster C_4 . Os que estão de:

- Azul pertencem aos clusters C_1 e C_2 ;
- Magenta pertencem aos clusters C_3 e C_4 .

Uma avaliação dos resultados que obtemos nessa etapa, será realizada no próximo capítulo na análise e discussão dos resultados. Antes disso, apresentaremos a aplicação dos algoritmos no processo de segmentação de imagens, como veremos a seguir.

3.3.2 Segmentação de Imagens

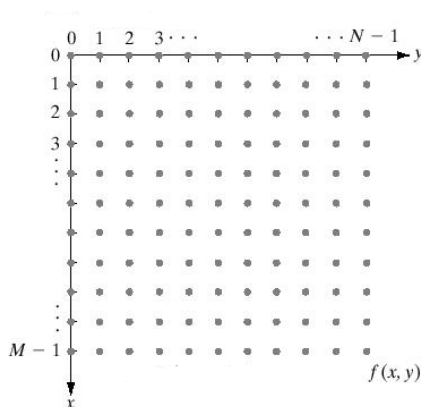
Um dos grandes desafios na área de processamento de imagens digitais é a extração rápida de informações a partir das imagens para a realização de análise. O grande avanço tecnológico dos últimos tempos possibilitou a utilização cada vez maior do computador nas tarefas de extração de informações de imagens. Uma dessas tarefas é a segmentação. A segmentação de imagens tem sido uma área de pesquisa ativa nas duas últimas décadas e como resultado, várias técnicas de segmentação de imagens têm sido propostas e descritas na literatura (BLOCH, 2015).

Utilizar uma imagem para solucionar problemas reais só é possível devido a digitalização. Após esse processo podem ser feitas diversas transformações na imagem até conseguir a solução do problema para o qual ela está sendo usada. Essas transformações podem ser agrupadas nos seguintes conjuntos: filtragem, análise, compressão e síntese de imagens. Referindo-se a análise, o primeiro passo é, geralmente, segmentar a imagem.

Segmentar uma imagem significa subdividir uma imagem em suas partes ou objetos constituintes, ou seja, encontrar e individualizar as regiões na imagem que tenham feições similares. Os pixels pertencentes a um grupo devem ser mais similares entre si do que em relação aos pixels pertencentes a outros grupos. Dessa forma, as técnicas de clusterização encontradas na literatura possuem objetivos similares e podem ser aplicadas para a segmentação de imagens (MEENA e RAJA, 2013).

Uma imagem pode ser definida como uma função bidimensional, $f(x, y)$, onde x e y são coordenadas espaciais, e a amplitude de f em qualquer par de coordenadas (x, y) é chamada de intensidade da imagem naquele ponto.

Figura 30 – Representação de uma imagem no plano cartesiano



Fonte: http://www.vision.ime.usp.br/ronaldo/mac0417-03/aula_03.html

Outra forma de representar uma imagem digital é por meio de uma matriz, onde os elementos p_{ij} da matriz representam os pixels. Para as imagens em preto e branco,

os valores dos pixels representam os tons de cinza da imagem. Cada pixel possui um valor que varia de 0 a 255, onde 255 representa a cor branca e o 0 a cor preta. Qualquer valor do intervalo $[0, 255]$ pode ser convertido a um valor no intervalo $[0, 1]$ por meio da função:

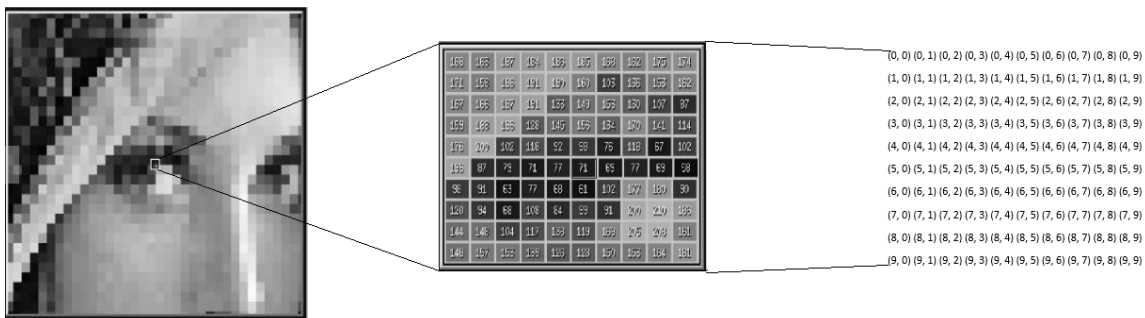
$$G(p_{ij}) = \frac{p_{ij}}{255}. \quad (7)$$

Assim, um pixel com tom 255 de cinza tem grau de intensidade 1, enquanto o pixel 0 tem grau de intensidade 0. Nesse âmbito, podemos também olhar uma imagem como um grafo fuzzy. Cada pixel representa um vértice com grau de intensidade no intervalo $[0, 1]$ e o grau de similaridade de um pixel a outro pode ser representado por uma aresta.

O peso da aresta que revela o grau de similaridade entre dois pixels distintos pode ser obtido por meio da Equação 6. Dessa forma, uma matriz M_p é gerada, onde cada entrada da matriz revela o grau de similaridade (ou o peso da aresta) entre dois pixels (ou vértices).

Para aplicarmos o FHC no processo de segmentação de imagem, selecionamos o recorte de uma imagem e a representamos no plano cartesiano, relacionando cada pixel a sua posição espacial (ver Figura 31), pois uma vez que a representação dos pixels é por meio das suas coordenadas, podemos perfeitamente aplicar o algoritmo FHC.

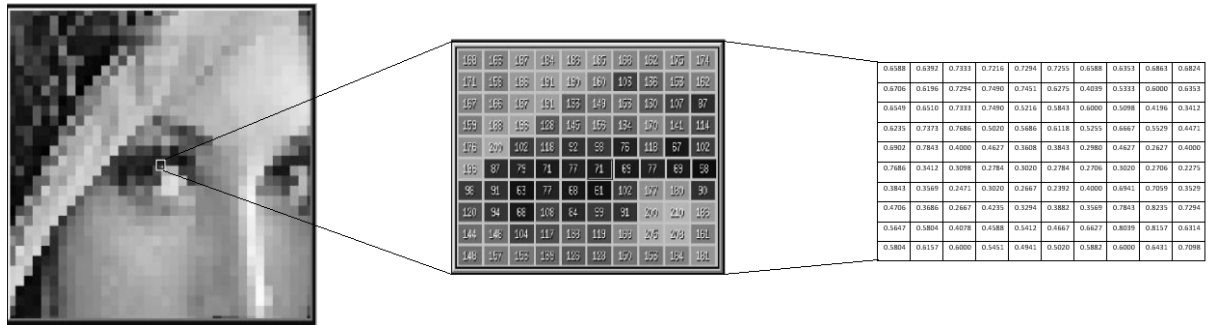
Figura 31 – Representação espacial dos pixels da imagem



Fonte: Elaborada pelos autores

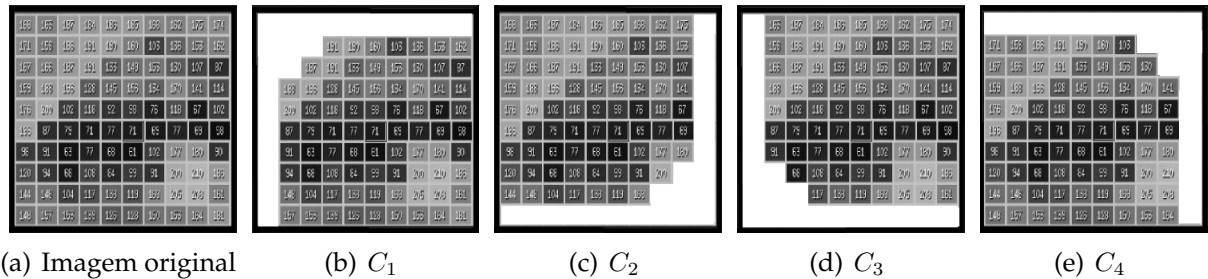
Por outro lado, para aplicarmos o algoritmo FHC-M, selecionamos o recorte da imagem e representamos no formato de uma matriz, onde cada entrada representa a intensidade do pixel no intervalo $[0, 1]$ (ver Figura 32), usando a Equação (7).

Em seguida, encontramos o peso de cada aresta que liga dois pixels distintos, gerando assim a matriz M_p . A partir dessas informações, aplicamos os algoritmos FHC e FHC-M para um λ aleatoriamente escolhido igual a 0.8 e os resultados podem ser observados nas Figuras 33 e 34.

Figura 32 – Representação dos pixels da imagem no intervalo $[0, 1]$ 

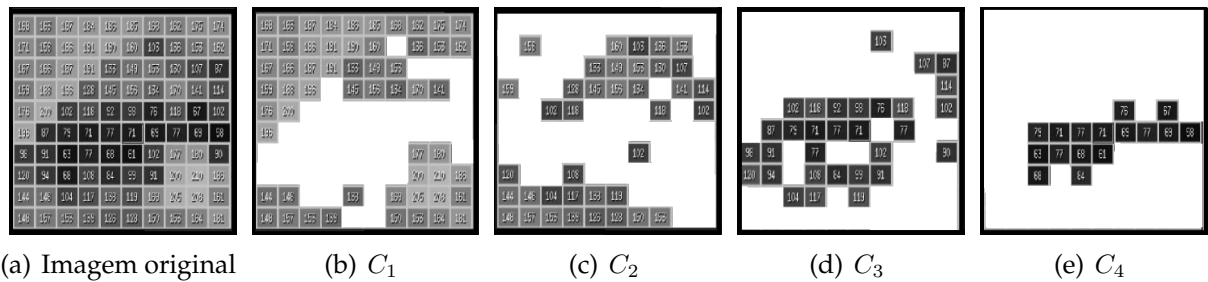
Fonte: Elaborada pelos autores

Figura 33 – Resultado da segmentação da imagem com o FHC



Fonte: Elaboradas pelos autores

Figura 34 – Resultado da segmentação da imagem com o FHC-M



Fonte: Elaboradas pelos autores

Nas duas últimas figuras acima, temos em (a) a imagem original e em (b), (c), (d) e (e), os clusters C_1 , C_2 , C_3 e C_4 , respectivamente, gerados com a aplicação dos algoritmos FHC e FHC-M. Notamos que para ambos os casos, o $\lambda = 0.8$ forneceu 4 clusters. Uma análise desses resultados juntamente com a da aplicação anterior apresentaremos a seguir na discussão e análises dos resultados que obtemos durante a realização da pesquisa.

4 DISCUSSÃO DOS RESULTADOS

Vimos que o desenvolvimento da nossa pesquisa foi compreendida em três etapas. Primeiramente pela análise do algoritmo FHC. Em seguida, pela avaliação da influência de uma medida de similaridade fuzzy nesse algoritmo e, por ultimo, por uma avaliação experimental a fim de verificarmos como se comportam os algoritmos diante de problemas que podem ser convertidos em grafos.

4.1 Resultados Iniciais

A análise do FHC nos permitiu compreender como funciona a dinâmica desse algoritmo. Vimos que ele é um algoritmo de clusterização hierárquica aglomerativo, pois inicialmente cada ponto (ou vértice) do conjunto de dados é visto como um cluster. Em seguida, para cada cluster inicial é encontrado o N_{centro} . Notamos que o N_{centro} sempre será um vértice do conjunto de dados, entretanto o N_{centro} ajustado geralmente não é um objeto do conjunto. Ele é um ponto que servirá como centro do subcluster gerado para cada ponto considerado, sendo utilizado para posicionar o subcluster de forma a encontrar os pontos que a ele pertencerão.

A partir dos subclusters, um grafo fuzzy é gerado. Cada subcluster é visto como um vértice e o peso da aresta que liga dois subclusters é obtido por meio da Equação (3). Essa fórmula revela o grau de conexidade fuzzy entre dois subclusters. Quanto mais pontos comuns entre eles, maior será o grau de conexidade. Após gerar o grafo fuzzy de subclusters, o parâmetro λ define o resultado da clusterização. Ele gera o λ -corte do grafo fuzzy de subclusters e cada componente conexa desse grafo corresponderá a um cluster.

A partir da compreensão do algoritmo, notamos que o FHC considera o conjunto de entrada como uma nuvem de dados, pois em nenhum passo do algoritmo é considerado o peso das arestas do grafo fuzzy ao qual se deseja realizar o agrupamento.

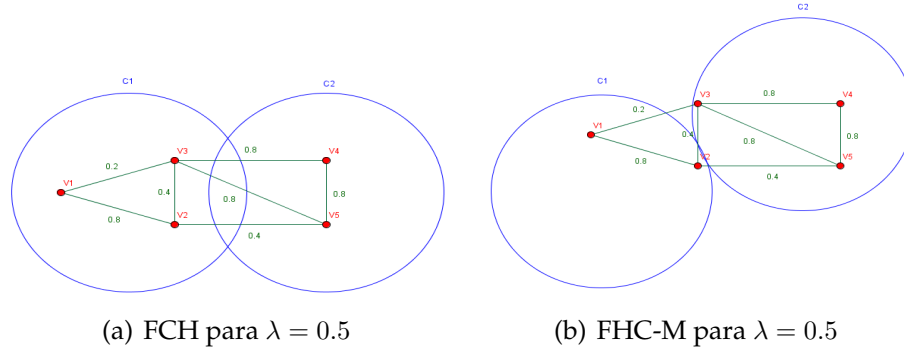
Além disso, o fato de considerar a distância euclidiana como uma medida de similaridade, o resultado do agrupamento gera clusters em que os dados são mais próximos dentro do espaço euclidiano. O que nos fez recorrer a uma medida de similaridade que considerasse os pesos das arestas, pois quando se trata de um grafo fuzzy, a proximidade no espaço euclidiano pode não significar similaridade.

A medida de similaridade fuzzy que utilizamos foi a definição de distância proposta por Rosenfeld em 1975. Esse tipo de métrica permite encontrar a distância entre dois vértices distintos, sendo que quanto menor a distância, maior o grau de

similaridade. Nesse sentido, a partir de um raio preestabelecido, os subclusters gerados conterão pontos mais similares entre si.

Avaliamos o comportamento do algoritmo FCH e do FHC-M no exemplo da Figura 12 e, para o mesmo valor de r e de λ , a saber, $r = 3.1$ e $\lambda = 0.5$, obtemos dois clusters para ambos os casos (ver figura 35).

Figura 35 – Resultados da clusterização



Fonte: Elaboradas pelos autores

Os cluters gerados pelo algoritmo FCH, por não considerar o peso das aresta do grafo, agrupou os dados que estavam próximos de acordo com a distância euclidiana. Por exemplo, a distância euclidiana de V_1 a V_2 é igual a distância euclidiana de V_1 a V_3 , então o FHC colocou no mesmo cluster esses três vértices. Entretanto, $\mu_R(V_1, V_3) = 0.2$, ou seja, a similaridade entre os dois vértices é muito pequena, então não é conveniente que eles estejam no mesmo cluster.

Em contrapartida, notamos a eficiência do algoritmo FHC-M nesse simples exemplo de grafo fuzzy. Ele gerou dois clusters, a saber, $C_1 = \{V_1, V_2\}$ e $C_2 = \{V_3, V_4, V_5\}$, em que os dados em cada cluster estão conectados fortemente, pois a aresta que conecta dois vértices distintos tem peso 0.8, que é um valor bem próximo de 1. A seguir, apresentamos alguns proposições que obtemos a partir da análise do algoritmo FHC.

Seja $V = \{V_1, V_2, V_3, \dots, V_n\}$ um conjunto de vértices (ou pontos) com coordenadas cartesianas. Seja M_D o conjunto de todas as distâncias euclidianas entre dois vértices distintos do conjunto V , a saber: $M_D = \{d(V_1, V_2), d(V_1, V_3), \dots, d(V_2, V_3), d(V_2, V_4), \dots, d(V_{n-1}, V_n)\}$. E seja $M_{CA} = \{d(V_1, O_1), \dots, d(V_n, O_1), d(V_2, O_2), \dots, d(V_n, O_2), \dots, d(V_n, O_n)\}$, onde: O_i é o N_{centro} ajustado do subcluster SC_i , $i = 1, \dots, n$. Além disso, as notações SC_i e C_i significam o i -ésimo subcluster e o i -ésimo cluster, respectivamente. As notações SC e C significa um único subcluster e um único cluster, respectivamente.

Proposição 1 Se $r \leq \min\{M_D\}$ então $SC_i = V_i$.

Demonstração: Se O_i é o N_{centro} ajustado de V_i , a $d(V_i, O_i) \geq r$, já que $\min\{M_D\} \geq r$. Assim, um subcluster é criado contendo apenas V_i . Portanto, $SC_i = V_i$.

Proposição 2 Se $SC_1 = \{V_1\}, SC_2 = \{V_2\}, SC_3 = \{V_3\}, \dots, SC_n = \{V_n\}$ então $\mu_{R'}(SC_i, SC_j) = 0$, para todo $i \neq j$.

Demonstração: Como $SC_1 = \{V_1\}, SC_2 = \{V_2\}, SC_3 = \{V_3\}, \dots, SC_n = \{V_n\}$, então:

$$SC_1 \cap SC_2 = \emptyset \Leftrightarrow N_{SC_1 \cap SC_2} = 0$$

$$SC_1 \cap SC_3 = \emptyset \Leftrightarrow N_{SC_1 \cap SC_3} = 0$$

:

$$SC_2 \cap SC_3 = \emptyset \Leftrightarrow N_{SC_2 \cap SC_3} = 0$$

$$SC_2 \cap SC_4 = \emptyset \Leftrightarrow N_{SC_2 \cap SC_4} = 0$$

:

$$SC_i \cap SC_j = \emptyset \Leftrightarrow N_{SC_i \cap SC_j} = 0$$

Assim, $N_{SC_i \cap SC_j} = 0$, para todo $i \neq j$, então $\mu_{R'}(SC_i, SC_j) = \frac{N_{SC_i \cap SC_j}}{N_{SC_i} + N_{SC_j} - N_{SC_i \cap SC_j}} = 0$.

Proposição 3 Se $r \leq \min\{M_D\}$ então: $C_i = V_i$.

Demonstração: Para todo $V_i \in V$, se $r \leq \min\{M_D\}$ então a $d(V_i, O_i) \geq r, i = 1, \dots, n$. Logo, $SC_i = V_i$ (Proposição 1) e $\mu_{R'}(SC_i, SC_j) = 0, i \neq j$ (Proposição 2). Assim, qualquer que seja $\lambda \in [0, 1]$, $G'_\lambda = \langle V', E'_\lambda \rangle$ tem $V' = \{SC_1, \dots, SC_n\} = \{V_1, \dots, V_n\}$. Portanto, $C_i = V_i$.

Proposição 4 Se $r > \max\{M_{CA}\}$ então $SC_i = V$.

Demonstração: Como $r > \max\{M_{CA}\}$, então $d(V_j, O_i) < r, i \neq j$. Logo, o subcluster SC_i de centro no N_{centro} ajustado O_i tem V_1, \dots, V_n como elementos. Portanto, $SC_i = V$.

Proposição 5 Se $SC_1 = \{V_1, V_2, \dots, V_n\}, SC_2 = \{V_1, V_2, \dots, V_n\}, \dots, SC_n = \{V_1, V_2, \dots, V_n\}$ então $\mu_{R'}(SC_i, SC_j) = 1$, para todo $i \neq j$.

Demonstração: Como $SC_1 = \{V_1, V_2, \dots, V_n\}, SC_2 = \{V_1, V_2, \dots, V_n\}, \dots, SC_n = \{V_1, V_2, \dots, V_n\}$, então

$$SC_1 \cap SC_2 = \{V_1, V_2, \dots, V_n\} \Rightarrow N_{SC_1 \cap SC_2} = N_{SC_1} = N_{SC_2}$$

$$SC_1 \cap SC_3 = \{V_1, V_2, \dots, V_n\} \Rightarrow N_{SC_1 \cap SC_3} = N_{SC_1} = N_{SC_3}$$

:

$$SC_2 \cap SC_3 = \{V_1, V_2, \dots, V_n\} \Rightarrow N_{SC_2 \cap SC_3} = N_{SC_2} = N_{SC_3}$$

$$SC_2 \cap SC_4 = \{V_1, V_2, \dots, V_n\} \Rightarrow N_{SC_2 \cap SC_4} = N_{SC_2} = N_{SC_4}$$

:

$$SC_{n-1} \cap SC_n = \{V_1, V_2, \dots, V_n\} \Rightarrow N_{SC_{n-1} \cap SC_n} = N_{SC_{n-1}} = N_{SC_n}$$

Portanto, para todo $i \neq j$ $\mu_{R'}(SC_i, SC_j) = \frac{N_{SC_i \cap SC_j}}{N_{SC_i} + N_{SC_j} - N_{SC_i \cap SC_j}} = 1$.

Proposição 6 Se $r > \max\{M_{CA}\}$ então $C = V$.

Demonstração: Como $r > \max\{M_{CA}\}$ então $d(V_i, O_j) < r$, $i \neq j$. Logo, $SC_i = V$ (Proposição 4) e $\mu_{R'}(SC_i, SC_j) = 1$ (Proposição 5). Assim, para qualquer $\lambda \in [0, 1]$, $G'_\lambda = \langle V', E'_\lambda \rangle$ tem $V' = \{SC\}$, onde $SC = \{V_1, \dots, V_n\}$. Portanto, $C = \{V\}$.

As proposições acima nos permitiram concluir que o raio precisa ser convenientemente escolhido. Se o raio for muito pequeno, o resultado da clusterização gera apenas ruídos. Por outro lado, se o raio for muito grande o resultado gera um único cluster contendo todos os pontos. Com relação ao parâmetro λ notamos que quanto mais próximo de 1, o resultado fornecerá clusters em que os vértices possuem um maior grau de similaridade. Além disso, o número de clusters também aumenta à medida que o parâmetro λ aumenta.

4.2 Resultados da Avaliação Experimental

Na avaliação experimental buscamos, em uma das duas etapas, aplicar o algoritmo FHC-M para realizar agrupamento de cidades de acordo com o IDHM. Para isso, selecionamos os municípios da mesorregião sul da Bahia e geramos um grafo fuzzy, onde os vértices correspondem às cidades e o peso da aresta que liga duas cidades distintas foi obtida a partir da Equação 6.

Diante disso, escolhemos dois parâmetros λ aleatórios e aplicamos o FHC e o FHC-M. Para um $\lambda = 0.2$ os algoritmos forneceram dois clusters e para $\lambda = 0.5$, o FCH forneceu cinco clusters e o FHC-M quatro. Visualmente notamos a diferença entre os algoritmos. Devido ao FHC usar a distância euclidiana para medir a similaridade dos dados, a tendência foi que os pontos próximos dentro do espaço euclidiano ficassem no mesmo cluster, já o FCH-M gerou clusters em que os dados estão extremamente distantes dentro do espaço euclidiano.

Para avaliarmos os resultados demonstrados nas Figuras 28 e 29, colocamos os IDHM dos municípios em ordem decrescente conforme a Tabela a seguir. O maior IDHM dessa mesorregião corresponde ao município de Itabuna e o menor ao município de Pirai do Norte.

Tabela 6 – Municípios da mesorregião sul da Bahia

1. Itabuna	0.712	36. Itaju do Colônia	0.592
2. Ilhéus	0.690	37. Canavieiras	0.590
3. Teixeira de Freitas	0.685	38. Itagibá	0.589
4. Eunápolis	0.677	39. Itacaré	0.583
5. Porto Seguro	0.676	40. Pau Brasil	0.583
6. Ipiaú	0.670	41. Camacan	0.581
7. Mucuri	0.665	42. Mascote	0.581
8. Nova Viçosa	0.654	43. Firmino Alves	0.578
9. Santa Cruz Cabrália	0.654	44. Itamari	0.578
10. Itanhém	0.637	45. Barra do Rocha	0.577
11. Itagimirim	0.634	46. Vereda	0.577
12. Gandu	0.632	47. Gongogi	0.576
13. Lajedão	0.632	48. Ibirataia	0.576
14. Cairu	0.627	49. Igrapiúna	0.574
15. Itamaraju	0.627	50. Itapebi	0.572
16. Ibicaraí	0.625	51. Itapitanga	0.571
17. Medeiros Neto	0.625	52. Nova Ibiá	0.570
18. Valença	0.623	53. Aurelino Leal	0.568
19. Prado	0.621	54. Jussari	0.567
20. Caravelas	0.616	55. Taperoá	0.566
21. Uruçuca	0.616	56. Camamu	0.565
22. Ibirapuã	0.614	57. Almadina	0.563
23. Buerarema	0.613	58. Una	0.560
24. Coaraci	0.613	59. Arataca	0.559
25. Ubaitaba	0.611	60. Presidente Tancredo Neves	0.559
26. Santa Cruz da Vitória	0.610	61. Guaratinga	0.558
27. Alcobaça	0.608	62. Ibirapitanga	0.558
28. Ituberá	0.606	63. Floresta Azul	0.557
29. Barro Preto	0.602	64. Santa Luzia	0.556
30. Itabela	0.599	65. Teolândia	0.555
31. Itajuípe	0.599	66. Nilo Peçanha	0.547
32. Itapé	0.599	67. São José da Vitória	0.546
33. Belmonte	0.598	68. Wenceslau Guimarães	0.544
34. Maraú	0.593	69. Jucuruçu	0.541
35. Ubatã	0.593	70. Piraí do Norte	0.533

Fonte: Elaborada pelos autores

Nos Quadros 1 e 2, destacamos os objetos de cada cluster resultante de forma simbólica a fim de visualizarmos claramente a disposição. O quadro 1 mostra os resultados destacados na Figura 28. O FCH colocou no primeiro cluster quase todas as cidades dispostas na Tabela 6, exceto a cidade de Mucuri. Dessa forma, ele agrupou no mesmo cluster cidades como Itabuna e o município de Piraí do Norte que têm o maior e o menor IDHM, respectivamente.

Quadro 1 – Resultado da Clusterização das Cidades baseada no IDHM

FHC para $\lambda = 0.2$
$C_1 = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}, V_{11}, V_{12}, V_{13}, V_{14}, V_{15}, V_{16}, V_{17}, V_{18}, V_{19}, V_{20}, V_{21}, V_{22}, V_{23}, V_{24}, V_{25}, V_{26}, V_{27}, V_{28}, V_{29}, V_{30}, V_{31}, V_{32}, V_{33}, V_{34}, V_{35}, V_{36}, V_{37}, V_{38}, V_{39}, V_{40}, V_{41}, V_{42}, V_{43}, V_{44}, V_{45}, V_{46}, V_{47}, V_{49}, V_{50}, V_{51}, V_{52}, V_{53}, V_{54}, V_{55}, V_{56}, V_{57}, V_{58}, V_{59}, V_{60}, V_{61}, V_{62}, V_{63}, V_{64}, V_{65}, V_{66}, V_{67}, V_{68}, V_{69}, V_{70}\}$
$C_2 = \{V_1, V_{13}, V_{20}, V_{23}, V_{28}, V_{35}, V_{37}, V_{42}, V_{44}, V_{47}, V_{48}, V_{51}, V_{55}, V_{62}, V_{69}\}$
FHCM para $\lambda = 0.2$
$C_1 = \{V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}, V_{11}, V_{12}, V_{13}, V_{14}, V_{16}, V_{17}, V_{18}, V_{19}, V_{20}, V_{21}, V_{22}, V_{23}, V_{24}, V_{25}, V_{28}, V_{30}, V_{31}, V_{32}, V_{33}, V_{34}, V_{35}, V_{36}, V_{37}, V_{38}, V_{39}, V_{40}, V_{41}, V_{42}, V_{43}, V_{44}, V_{45}, V_{46}, V_{47}, V_{49}, V_{50}, V_{52}, V_{53}, V_{55}, V_{56}, V_{58}, V_{59}, V_{60}, V_{61}, V_{63}, V_{64}, V_{65}, V_{66}, V_{67}, V_{68}, V_{69}, V_{70}\}$
$C_2 = \{V_{15}, V_{26}, V_{27}, V_{29}, V_{37}, V_{48}, V_{51}, V_{54}, V_{57}, V_{62}\}$

Fonte: Elaborado pelos autores

No segundo cluster, o FHC agrupou as cidades de Alcobaça (0.608), Caravelas (0.616), Guaratinga (0.558), Ibirapuã (0.614), Itabela (0.599), Itamaraju (0.627), Itanhém (0.637), Jucuruçu (0.541), Lajedão (0.632), Medeiros Neto (0.625), Mucuri (0.665), Nova Viçosa (0.654), Prado (0.621), Teixeira de Freitas (0.685) e Vereda (0.577). Esse agrupamento também não gerou um resultado satisfatório, pois no mesmo cluster estão as cidades de Teixeira de Freitas (0.685) e Jucuruçu (0.541) que têm o IDHM com valores bem distintos.

Por outro lado, o algoritmo FHC-M agrupou 61 cidades no primeiro cluster. As nove cidades que não entraram nesse cluster foram: Eunápolis, Ilhéus, Ipiaú, Itabuna, Mucuri, Nova Viçosa, Porto Seguro, Santa Cruz Cabrália e Teixeira de Freitas. Estas foram agrupadas no segundo cluster juntamente com Itanhém. Notamos que estas são as dez primeiras cidades que aparecem na Tabela 6, ou seja, são as dez cidades que têm o maior IDHM da mesorregião sul da Bahia.

Da mesma forma, como podemos notar no Quadro 2, que mostra os resultados destacados na Figura 29, o resultado da clusterização não foi eficiente usando o algoritmo FHC. No cluster C_1 encontramos novamente a cidade que possui o maior IDHM agrupada com a cidade que tem o menor IDHM: Itabuna e Piraí do Norte, respectivamente. Os demais clusters também não apresentaram um agrupamento satisfatório, pois encontramos cidades com IDHM bem distintos no mesmo cluster.

Em contrapartida, no primeiro cluster o FHC-M agrupou 51 cidades que foram todas as cidades compreendidas de Caravelas até Piraí do Norte da Tabela 6. No cluster C_2 ele agrupou 33 cidades que foram todas as que estão de Itanhém até Mascote. No cluster C_3 agrupou 9 cidades que foram Ilhéus, Teixeira de Freitas, Eunápolis, Porto Seguro, Ipiaú, Mucuri, Nova Viçosa, Santa Cruz Cabrália e Itanhém. E no cluster C_4 ele agrupou as três cidades da mesorregião sul da Bahia que têm os maiores IDHM.

Quadro 2 – Resultado da Clusterização das Cidades baseada no IDHM

FHC para $\lambda = 0.5$	
$C_1 =$	$\{ V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_9, V_{10}, V_{11}, V_{12}, V_{14}, V_{16}, V_{17}, V_{18}, V_{19}, V_{21}, V_{22}, V_{24}, V_{25}, V_{26}, V_{27}, V_{29}, V_{30}, V_{31}, V_{32}, V_{33}, V_{34}, V_{36}, V_{38}, V_{39}, V_{40}, V_{41}, V_{43}, V_{45}, V_{46}, V_{49}, V_{50}, V_{52}, V_{53}, V_{56}, V_{58}, V_{59}, V_{60}, V_{61}, V_{63}, V_{64}, V_{65}, V_{66}, V_{67}, V_{68}, V_{70} \}$
$C_2 =$	$\{ V_1, V_{13}, V_{23}, V_{28}, V_{35}, V_{48}, V_{51}, V_{55}, V_{62}, V_{69} \}$
$C_3 =$	$\{ V_3, V_7, V_{10}, V_{12}, V_{15}, V_{20}, V_{28}, V_{32}, V_{39}, V_{46}, V_{52}, V_{54}, V_{57}, V_{59}, V_{66} \}$
$C_4 =$	$\{ V_7, V_{10}, V_{12}, V_{15}, V_{20}, V_{28}, V_{32}, V_{35}, V_{39}, V_{42}, V_{46}, V_{52}, V_{54}, V_{57}, V_{59} \}$
$C_5 =$	$\{ V_{13}, V_{20}, V_{23}, V_{35}, V_{37}, V_{42}, V_{44}, V_{47}, V_{48}, V_{51}, V_{62}, V_{69} \}$
FHCM para $\lambda = 0.5$	
$C_1 =$	$\{ V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8, V_{10}, V_{11}, V_{12}, V_{13}, V_{14}, V_{16}, V_{17}, V_{19}, V_{20}, V_{22}, V_{23}, V_{24}, V_{25}, V_{28}, V_{30}, V_{31}, V_{33}, V_{34}, V_{36}, V_{38}, V_{39}, V_{40}, V_{41}, V_{42}, V_{43}, V_{45}, V_{46}, V_{49}, V_{50}, V_{52}, V_{53}, V_{56}, V_{58}, V_{59}, V_{60}, V_{61}, V_{63}, V_{64}, V_{65}, V_{66}, V_{67}, V_{69}, V_{70} \}$
$C_2 =$	$\{ V_1, V_6, V_7, V_8, V_9, V_{10}, V_{12}, V_{13}, V_{14}, V_{18}, V_{21}, V_{23}, V_{28}, V_{30}, V_{31}, V_{32}, V_{33}, V_{34}, V_{35}, V_{37}, V_{38}, V_{41}, V_{44}, V_{45}, V_{46}, V_{47}, V_{52}, V_{55}, V_{58}, V_{64}, V_{65}, V_{67}, V_{68} \}$
$C_3 =$	$\{ V_{15}, V_{26}, V_{27}, V_{37}, V_{48}, V_{51}, V_{54}, V_{57}, V_{62} \}$
$C_4 =$	$\{ V_{26}, V_{29}, V_{62} \}$

Fonte: Elaborado pelos autores

Nesse sentido, fica evidente que o FHC-M gerou clusters com cidades que o IDHM são mais similares, enquanto o FHC colocou no mesmo cluster as cidades que estão geometricamente mais próximas entre si. É possível até observar nos resultados da aplicação do FHC que muitas cidades com IDHM bem próximos estavam no mesmo cluster. Isso aconteceu porque as cidades próximas tendem a ter os Índices de Desenvolvimento Humano parecidos. Entretanto, isso não eliminou o entrave que o FHC carregou ao colocar no mesmo cluster cidades com IDHM bem distintos.

Notamos que o FHC-M se tornou um algoritmo conveniente para realizar agrupamento de cidades, pois ele permitiu, por meio dos clusters, uma visão geral das cidades que possuem IDHM mais similares. Assim, ficou mais evidente que as cidades pertencentes ao mesmo cluster possuem uma situação econômica, cultural e social mais parecidas. Nesse sentido, tendo um conhecimento das cidades com IDHM baixo é possível que estudos sejam realizados para identificar qual ou quais das dimensões que são utilizadas no cálculo do IDHM estão influenciando para que isso ocorra e, dessa forma, medidas poderão ser tomadas para elevar o IDHM dessas cidades.

Essa discrepância do FCH ficou evidente também nos resultados apresentados na etapa da avaliação experimental referente a segmentação de imagens. Para avaliar o comportamento desses algoritmos nesse processo, convertamos a imagem a um grafo fuzzy, onde cada pixel representou um vértice e cada aresta representou o grau de similaridade entre dois vértices distintos.

Para aplicar o algoritmo FHC foi necessário ter como informação as coordenadas

cartesianas de cada vértice. Dessa forma, usamos a noção de que uma imagem pode ser vista como uma função bidimensional, onde o par de coordenadas (x, y) pode representar um pixel. Diante disso, para um $\lambda = 0.8$, o FHC forneceu quatro clusters, onde reuniu em cada um, os pixels geometricamente mais próximos.

É notável na Figura 33 que a imagem não foi bem segmentada. O algoritmo não respeitou o critério de tonalidade dos pixels para formar os clusters, pois nos quatro clusters gerados é possível encontrar pixels com tons de cinzas de maiores e menores intensidades. Em outras palavras, bastou que os pixels estivessem próximos para que o FHC os colocasse no mesmo cluster.

Por outro lado, para o mesmo valor do parâmetro λ , o FHC-M segmentou a imagem de forma coerente, pois agrupou os pixels que tinham tons de cinza similares. É possível observar na Figura 34 que o cluster C_1 é formado pelos pixels próximos ao nível de cinza 255, os clusters C_2 e C_3 são formados pelos pixels com níveis de cinza intermediários e o cluster C_4 pelos pixels próximos ao nível de cinza 0.

O principal interesse dos algoritmos de clusterização de grafos é gerar clusters em que os vértices possuam um maior grau de similaridade e esse grau é expresso por meio dos pesos das arestas. O fato do FHC não considerar os pesos das arestas do grafo nos passos do algoritmo, faz com que vértices que não tenham conexão com nenhum outro sejam colocados em algum cluster. Basta que a distância desse vértice até aos demais respeite a distância que é preestabelecida como dado de entrada no algoritmo.

Dessa forma, ficou evidente na avaliação experimental que, apesar do FHC ser um algoritmo de clusterização hierárquica baseada na conexidade de grafos fuzzy, ele não é eficiente quando aplicado a grafos fuzzy ou problemas que podem ser convertidos a grafos fuzzy. Nesse sentido, notamos o quanto é importante conhecer as dinâmicas dos algoritmos de clusterização antes de aplicá-los a um conjunto de dados, pois a escolha de um algoritmo não conveniente pode gerar uma análise incorreta dos dados após o processo de clusterização. A seguir, apresentamos as considerações finais do nosso trabalho.

5 CONCLUSÃO

Neste trabalho de dissertação propomos responder questões em torno do estudo de clusterização de grafos fuzzy. Com base nas teorias que fundamentaram a nossa pesquisa, realizamos uma análise do algoritmo de clusterização hierárquica baseada na conexidade de grafo fuzzy, o FHC. Essa análise nos possibilitou compreender sua dinâmica e o seu funcionamento. Posteriormente, avaliamos a influência de uma medida de similaridade fuzzy neste algoritmo e definimos um ajuste de centro fuzzy diante do processo de clusterização. Depois, aplicamos os algoritmos para realizar agrupamento de cidades de acordo com o IDHM e também no processo de segmentação de imagens.

Diante disso, concluímos que a proposta do FHC é gerar um grafo fuzzy de subclusters a partir de uma nuvem de dados e o agrupamento desses vértices ocorre obedecendo uma distância clássica preestabelecida. Isso contribuiu para que os resultados não fossem satisfatórios, pois proximidade geométrica de dados pode não significar similaridade. Assim, notamos que o algoritmo FHC não é de fato um algoritmo para grafos fuzzy. O fato dele considerar os dados de entrada como uma nuvem de dados faz com que em nenhum momento do algoritmo os pesos das arestas do grafo sejam levados em consideração. Isso faz com que o FHC se limite a certas aplicações, não sendo conveniente, por exemplo, para ser aplicado a problemas que podem ser convertidos em grafos.

Na tentativa de eliminar esse entrave do FHC, criamos o algoritmo FHC-M. Esse algoritmo criado foi baseado essencialmente no FHC, mas com uma nova abordagem: modificamos a distância e o ajuste de centro clássicos utilizados por uma distância e ajuste de centro fuzzy. Essa modificação no algoritmo gerou melhores resultados na aplicação de problemas que podem ser convertidos em grafos, pois para calcular a distância fuzzy entres dois vértices foi necessário considerar o peso das arestas. Quanto menor a distância fuzzy, maior a similaridade dos dados e, conseqüentemente, os dados mais similares ficaram no mesmo cluster.

Devido ao número de operações que ocorre nas etapas do algoritmo FHC-M, o custo computacional se torna maior, entretanto este mostrou melhores resultados em comparação ao FHC. Dessa forma, notamos que o FHC, que é um algoritmo baseado na teoria de conjuntos fuzzy, foi melhorado quando utilizamos a medida de similaridade fuzzy. Essa nova visão pode ampliar as possibilidades de trabalhos na área, servindo de inspiração para trabalhos futuros, pois o algoritmo modificado se mostrou funcional e os resultados se mostraram promissores.

5.1 Trabalhos Futuros

A seguir, enumeramos alguns trabalhos futuros que podem ser feitos como continuação ou extensão da pesquisa que foi realizada nessa dissertação:

- Vimos que λ e raio utilizados como dados de entradas influenciam diretamente no resultado da clusterização tanto no algoritmo FHC quanto no FHC-M. Uma proposta de trabalho futuro é automatizar a escolha desses parâmetros.
- Um dos entraves da implementação em linguagens de programação como C ou C++ é falta de uma interface gráfica. Dessa forma, um próximo passo interessante seria implementar os algoritmos em softwares mais interativos como o MATLAB.
- Como forma de avaliar o comportamento do algoritmo FHC-M diante de problemas que podem ser convertidos em grafos, aplicamos no processo de agrupamento de cidades e na segmentação de imagens. O objetivo da avaliação experimental foi principalmente analisar o comportamento do FHC-M, por isso, não aplicamos em um conjunto de dados demasiadamente grande. Um trabalho futuro bastante relevante seria aplicar o algoritmo em conjunto de dados de alta dimensão.
- Vimos na revisão de literatura que é possível avaliar o quão satisfatório foi o resultado da clusterização por meio de índices internos ou externos. Propomos como trabalho futuro utilizar um desses critérios para avaliar os clusters resultantes.
- Vimos que a essência do algoritmo FHC é gerar um grafo fuzzy de subclusters e o λ -corte desse grafo é o resultado da clusterização. Ou seja, os clusters gerados não são conjuntos fuzzy. Uma proposta de trabalho futuro bem significativa é incluir uma graduação nesses conjuntos de forma a permitir associar um objeto a todos os clusters usando uma função de pertinência.

6 REFERÊNCIAS

- AGGARWAL, C.C.; REDDY, C.K. Data Clustering: Algorithms and Applications. Chapman e Hall/CRC Data Mining and Knowledge Discovery Series, 2013.
- ALVES, L. A; MELLO, J.C.C. Weights based clustering in Data Envelopment Analysis using Kohonen Neural Network: An Application in Brazilian Electrical Sector. Latin America Transactions, IEEE (Revista IEEE America Latina), v. 13, 2015.
- ANDERSON, P.; FAROOQ, B.; EFTHYMIOU, D.; BIERLAIRE, M. Association Generation in Synthetic Population for Transportation Applications: Graph-Theoretic Solution. Transportation Research Record, Journal of the Transportation Research Board, v. 2429, p. 38-50, 2014.
- ANDERSON, L. et al. The clustering of galaxies in the SDSS-III Baryon Oscillation Spectroscopic Survey: baryon acoustic oscillations in the data releases 10 and 11 Galaxy samples. Oxford Journals Science and Mathematics MNRAS, v. 441, p. 24-62, 2014.
- ANOOJ, P.K. Clinical decision support system: Risk level prediction of heart disease using weighted fuzzy rules. Journal of King Saud University - Computer and Information Sciences, V. 24, p. 27-40, 2012.
- BAKER, R. S.; INVENTADO, P. S. Educational Data Mining and Learning Analytics. Springer New York, p. 61-75, 2014.
- BALAKRISHNAN, R.; RANGANATHAN, K. A Textbook of Graph Theory. Springer New York. Book Universitext, 2012.
- BARROS, L.C.; BASSANEZI, R.C.; TONELLI, P.A. Fuzzy modeling in population dynamics. Ecological Modelling 128, p. 27-33, 2000.
- BEDE, B. Mathematics of Fuzzy Sets and Fuzzy Logic. Springer Berlin Heidelberg, v. 295, 2013.
- BLOCH, I. Fuzzy sets for image processing and understanding. Fuzzy Sets and Systems 281, p. 280-291, 2015.
- BRUYLANTS, G.; BARTIK, K.; REISSE, J. Prebiotic chemistry: A fuzzy field. Comptes Rendus Chimie, v. 14, p. 388-391, 2011.
- BRUYLANTS, G.; BARTIK, K.; REISSE, J. Is it Useful to Have a Clear-cut Definition of Life? On the Use of Fuzzy Logic in Prebiotic Chemistry. Origins of Life and Evolution of Biospheres, v. 40, p. 137-143, 2010.
- BUENO, Lourenço, et al. Agrupamento Evolutivo aplicado ao reconhecimento de padrões em dados médicos. In: Anais do XX Congresso Brasileiro de Automática. Minas

Gerais, 2014.

BUNKE, H.; RIESEN, K. Recent advances in graph-based pattern recognition with applications in document analysis. *Pattern Recognition*, v. 44, p. 1057–1067, 2011.

CASSIANO, K. M.; SOUZA, R. C. *Análise de Séries Temporais Usando Análise Espectral Singular (SSA) e Clusterização de Suas Componentes Baseada em Densidade*. Rio de Janeiro. 172p. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, 2014.

CASTRO, M. A. T. A. *Agrupamento – "Clustering"*. Engenharia Informática: Projeto, 2003.

CERVO, V. L. *Seleção de variáveis para a clusterização através de índices de importâncias das variáveis e análise de componentes*. Dissertação de Mestrado - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal do Rio Grande do Sul, 2013.

CHANG, M. An adaptive self-organizing fuzzy sliding mode controller for a 2-DOF rehabilitation robot actuated by pneumatic muscle actuators. *Control Engineering Practice*, v. 18, p. 13–22, 2010.

CHARTRAND, G.; ZHANG, P. *A First Course in Graph Theory*, 2012.

CHEN, H. et al. An efficient diagnosis system for detection of Parkinson's disease using fuzzy k-nearest neighbor approach. *Expert Systems with Applications*, v. 40, p. 263–271, 2013.

CHRISTODOULOU, K.; PATON, N. W.; FERNANDES, A. A.; A. Structure Inference for Linked Data Sources Using Clustering. *Transactions on Large-Scale Data- and Knowledge-Centered Systems XIX*, v. 8990 of the series *Lecture Notes in Computer Science*, p. 1-25, 2015.

CÔME, E.; LATOUCHE, P. Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood. *Statistical Modeling*, 2015.

COUPRIE, C. et al. Power Watershed: A Unifying Graph-Based Optimization Framework. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 33, 2010.

Diretoria de Informações Geoambientais (2015). Acesso em 01 de novembro de 2015, http://www.sei.ba.gov.br/site/geoambientais/mapas/pdf/mesorregioe_2015.pdf

DONG, Y. et al.. A hierarchical clustering algorithm based on fuzzy graph connectedness. *Fuzzy Sets and Systems*, v. 157, p. 1760-1774, 2006.

DUBOIS, D.; PRADE, H. *Fundamentals of Fuzzy Sets (The Handbooks of Fuzzy Sets)*, Springer Verlag, 2000.

DUBOIS, D.; PRADE, H. *Fuzzy Sets and Systems, Theory and Applications*, Academic

Press, New York, 1980.

ELHADO, A. A. E.; MAAROF, M. A.; OSMAN, A. H.; Malware Detection Based on Hybrid Signature Behaviour Application Programming Interface Call Graph. *American Journal of Applied Sciences* 9 (3), p. 283-288, 2012.

ELLIS-MONAGHAN, J.A.; MERINO, C. Graph Polynomials and Their Applications I: The Tutte Polynomial. Birkhäuser Boston. *Structural Analysis of Complex Networks*, p. 219-255, 2010.

ESMIN, A. A. A; CALEGARI, C. L. L. e PEREIRA, M. R. Aplicação de clusterização de dados na base de dados do zoneamento ecológico-econômico de minas gerais, 2015.

ESMIN, A. A. A.; TOLEDO, C. F. M.; OLIVEIRA, H. C. B. de. Estudo e adaptação da técnica de otimização por exame de partículas (pso) aplicada ao agrupamento e classificação em bases de dados textuais, 2015.

FITRIANAH, D. et al. A Data Mining Based Approach for Determining the Potential Fishing Zones. *International Journal of Information and Education Technology*, v. 6, nº.3, 2016.

FONSECA, D.J.; KNAPP, G. M. An expert system for reliability centered maintenance in the chemical industry". *Expert Systems with Applications* 19, p. 45-57, 2000.

FREITAG, S.; GRAF, W.; KALISKE, M. Recurrent neural networks for fuzzy data. *Journal Integrated Computer-Aided Engineering - Data Mining in Engineering*, v. 18, p. 265-280, 2011.

GARCÍA, J. S.; PONS, S. V.; SHULCLOPER, J. R. Algoritmos de agrupamiento difuso, índices de validación: un estado del arte. Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV). 2012.

GOKMEN, G. et al. Evaluation of student performance in laboratory applications using fuzzy logic. *Procedia - Social and Behavioral Sciences. Innovation and Creativity in Education*, v. 2, p. 902-909, 2010.

GONZALEZ-ARANGUENAA, E.; MANUELA, C. M.; POZOB, M.; Values of games with weighted graphs. *European Journal of Operational Research*, p. 248-257, 2014.

GROSS, G.; NAGI, R.; SAMBHOOS, K. A fuzzy graph matching approach in intelligence analysis and maintenance of continuous situational awareness. *Information Fusion*, v. 18, p. 43-61, 2014.

GULER, C. et al. Assessment of the impact of anthropogenic activities on the groundwater hydrology and chemistry in Tarsus coastal plain (Mersin, SE Turkey) using fuzzy clustering, multivariate statistics and GIS techniques. *Journal of Hydrology*, v. 414-415, p. 435-451, 2012.

- HAMED, R. I.; AHSON, S. I. PARVEEN, Rafat. Designing genetic regulatory networks using fuzzy Petri nets approach. *International Journal of Automation and Computing*, v. 7, p. 403-412, 2010.
- HATAMLOU, A. Black hole: A new heuristic optimization approach for data clustering. *Information Sciences*, v. 222, p. 175–184, 2013.
- HONG, S.; OGUNTEBI, T.; OLUKOTUN, K. Efficient Parallel Graph Exploration on Multi-Core CPU and GPU. *Parallel Architectures and Compilation Techniques (PACT)*, 2011 International Conference on, p. 78-88, 2011.
- IGBE - Instituto Brasileiro de Geografia e Estatística. Censo Demográfico. Rio de Janeiro, 2013.
- JAIN, A. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, p.651–666, 2010.
- KANNANA, S.R. et al. Effective fuzzy c-means based kernel function in segmenting medical images. *Computers in Biology and Medicine*. v. 40, p. 572–579, 2010.
- KAUR, A.; KAUR, A. Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Air Conditioning System. *International Journal of Soft Computing and Engineering (IJSCE)*, v. 2, 2012.
- KHAN, A. R.; AMIN, H. U.; REHMAN, Z. U. Application of Expert System with Fuzzy Logic in Teachers' Performance Evaluation. (IJACSA) *International Journal of Advanced Computer Science and Applications*, v. 2, nº 2, 2011.
- KIM, K.; KIM, J. Self Health Diagnosis System with Korean Traditional Medicine Using Fuzzy ART and Fuzzy Inference Rules. *Intelligent Information and Database Systems*, v. 7198, p. 326-335, 2012.
- LAI, L. et al. Fuzzy Knowledge Management through Knowledge Engineering and Fuzzy Logic. *Journal of Convergence Information Technology*, v. 5, Nº 3, 2010.
- LARA, R. B. et al. Modelo para el análisis estático en grafos difusos basado en indicadores compuestos de centralidad. *Revista Cubana de Ciencias Informáticas*, v. 9, nº 2, p. 50-63, 2015.
- LEA, T. et al. Policy fuzz and fuzzy logic: researching contemporary Indigenous education and parent-school engagement in north Australia. *Journal of Education Policy*, v. 26, p. 321-339, 2011.
- LI, S.G. et al. Energy and Battery Management of a Plug-In Series Hybrid Electric Vehicle Using Fuzzy Logic. *IEEE*, v. 60, p. 3571-3585, 2011.
- LIBICH, D. S. et al. Fuzzy complexes of myelin basic protein: NMR spectroscopic investigations of a polymorphic organizational linker of the central nervous system.

Biochemistry and Cell Biology, p. 143-155. 2010.

LIN, M. CHEN, C. Application of fuzzy models for the monitoring of ecologically sensitive ecosystems in a dynamic semiarid landscape from satellite imagery. *Engineering Computations*, v. 27, p.5-19, 2010.

LIU, G. et al. Combination of Neuro-Fuzzy Network Models with Biological Knowledge for Reconstructing Gene Regulatory Networks. *Journal of Bionic Engineering* 8, p.98–106, 2011.

LOPES, R. V. Desenvolvimento de novas metodologias baseadas em clustering para identificação de sistemas dinâmicos híbridos no espaço de estados. 2014. xx, 145 f. Tese (Doutorado em Engenharia de Sistemas Eletrônicos e Automação)—Universidade de Brasília, Brasília, 2014.

MEENA, A.; RAJA, K. Spatial Fuzzy C-Means PET Image Segmentation of Neurodegenerative Disorder. *Indian Journal of Computer Science and Engineering (IJCSE)*, v. 4, 2013.

MENDOZA, N. A. et al. A Nectar of Frequent Approximate Subgraph Mining for Image Classification. *Revista Cubana de Ciencias Informáticas, La Habana*, v.9, nº 1, 2015.

MEZEY, P. G. Fuzzy Electron Density Fragments in Macromolecular Quantum Chemistry, Combinatorial Quantum Chemistry, Functional Group Analysis, and Shape–Activity Relations. *American Chemical Society*, p. 2821–2827, 2014.

MUTHUKARUPPAN, S. A; ER, M.J. hybrid particle swarm optimization based fuzzy expert system for the diagnosis of coronary artery disease. *Expert Systems with Applications*, v. 39, p. 11657–11665, 2012.

NUNES, C. M.; COELHO, C. J. Análise e Classificação de de pixel em imagens de satélite empregando agrupamento difuso. Dissertação de Mestrado. Mestrado em Ciências Ambientais, Centro Universitário de Anápolis. Goiás, 2014.

ORTEGA, N.R.S.; BARROS, L.C.; MASSAD, E. An Application of the Extension Principle to Fuzzy Epidemic Models. *Fuzzy Sets and Systems*, 2001.

ORTEGA, N.R.S.; SALLUM, P.C.; MASSAD, E. Fuzzy Dynamical System in Epidemic Modelling". *Kybernetes* 29 (1-2), p.201-218, 2000.

PAL, D. et al. Fuzzy expert system approach for coronary artery disease screening using clinical parameters. *Knowledge-Based Systems*, v. 36, p.162–174, 2012.

PEDRYCZ, W.; GOMIDE, F. *An Introduction to Fuzzy Sets: Analysis and Design*, MIT Press, 1998.

PENG, G. Variable Pitch Control of Wind Turbine Generator Combined With Fuzzy Feed Forward and Fuzzy PID Controller. *Proceedings of the CSEE*, 2010.

PETERSEN, J. et al. Optimal Graph Based Segmentation Using Flow Lines with Application to Airway Wall Segmentation. Information Processing in Medical Imaging Volume 6801 of the series Lecture Notes in Computer Science, p. 49-60, 2011.

Programa das Nações Unidas para o Desenvolvimento – PNUD. (2013). Atlas do Desenvolvimento Humano no Brasil 2013. Acesso em 01 de novembro 2015, em <http://www.pnud.org.br/arquivos/idhm-brasileiro-atlas-2013.pdf>

Programa das Nações Unidas para o Desenvolvimento – PNUD. (2012). Desenvolvimento Humano e IDH. Acesso em 01 de novembro de 2015, em <http://www.pnud.org.br/IDH/DH.aspx>

REZANIK, L. Fuzzy Controllers. Newnes, Reino Unido, 1997.

ROSENFELD, A. Fuzzy Graphs. In: Zadeh LA, Fu KS, Shimura M (eds). Fuzzy Sets and their Application to Cognitive and Decision Processes. Academic Press, New York, p. 77-95, 1975.

RUOHONEN, K. Graph Theory, 2013.

SAHOO, S.; PAL, M. Intuitionistic fuzzy competition graphs. Korean Society for Computational and Applied Mathematics, 2015.

SHAO, B.; WANG, H.; LI, Y. Trinity: a distributed graph engine on a memory cloud .Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, p. 505-516, 2013.

SAMANTA, S.; PAL, M. A New Approach to Social Networks Based on Fuzzy Graphs. An Official Journal of Turkish Fuzzy Systems Association, v.5, n° 2, p. 078-099, 2014.

SILVA, D. T. A. da; SANTOS, V. F. ClusteringTools: Uma Ferramenta de Auxílio ao Ensino de Técnicas de Clusterização. UFRN, 2007.

SOUSA, A.; NASCIMENTO, A. Visualização e Algoritmos de Clustering para Análise Filogenética. Instituto Superior de Engenharia de Lisboa. Relatório final realizado no âmbito de Projeto e Seminário do curso de licenciatura em Engenharia Informática e de Computadores Semestre de Verão 2014/2015, 2015.

SRIPAN, R.; SUKSAWAT, B. Propose of fuzzy logic-based students' learning assessment. Control Automation and Systems (ICCAS), 2010 International Conference on, p. 414-417, 2010.

SYROPOULOS, A. Theory of Fuzzy Computation. Springer Publishing Company, 2013.

TAKAHASHI, M. T. Contribuições ao Estudo de Grafos Fuzzy: Teoria e Algoritmos. Tese de Doutorado. UNICAMP, 2004.

TAMIR, D. E.; RISHE, N. D.; KANDEL, A. Fifty Years of Fuzzy Logic and its Applications. Studies in Fuzziness and Soft Computing. Springer, 2015

- TARRAZO, M.; GUTIERREZ, L. Economic expectations, fuzzy sets and financial planning". *European Journal of Operational Research* 126, p.89-105, 2000.
- TOM, M.; SUNITHA, M. S. Strong sum distance in fuzzy graphs. Springer, 2015.
- TOMASEV, N. et al. The Role of Hubness in Clustering High-Dimensional Data. *Knowledge and Data Engineering, IEEE Transactions on*, v. 26, 2013.
- WAGHOLIKAR, K. et al. Evaluation of Fuzzy Relation Method for Medical Decision Support. *Journal of Medical Systems*. v. 36, p. 233-239, 2012.
- WANGAB, W.; WANGA, D.; PENGGA, Z.H. Cooperative fuzzy adaptive output feedback control for synchronisation of nonlinear multi-agent systems under directed graphs. *International Journal of Systems Science*, v. 46, 2015.
- WU, L. W.; ADOKO, A. C. Fuzzy Inference Systems-based Approaches in Geotechnical Engineering- a Review. *EJGE*, v. 16, 2011.
- WU, Y. et al. Arbuscular mycorrhiza fungi as an ecology indicator for evaluating desert soil conditions. *Frontiers of Agriculture in China*, v. 4, p. 24-30, 2010.
- XAVIER, C. S. O. Avaliação da Aprendizagem em Matemática : uma abordagem utilizando à lógica fuzzy. IN: XVIII EBRAPEM, 2014.
- XIN, L. The Evaluation on Finance Ecological Environment using Unascertained Measure and Fuzzy Analytic Hierarchy Process. *International Journal of Digital Content Technology and its Applications (JDCTA)*, v. 6, 2012.
- YAN, L.; MA, Z. M. Conceptual design of object-oriented databases for fuzzy engineering information modeling. *Integrated Computer-Aided Engineering*, v. 20, p.183-197, 2013.
- YEH, R.; BANG, S. Fuzzy graphs, fuzzy relations, and their applications to cluster analysis. In: LA Zadeh, KS Fu and M. Shimura, Eds., *Fuzzy Sets and Their Applications*, Academic Press, New York, p. 125-149, 1975.
- YEN, J.; LANGARI, R. *Fuzzy Logic: Intelligence, Control, and Information*. Prentice Hall, EUA, 1999.
- YINA, S. et al. Efficient multilevel image segmentation through fuzzy entropy maximization and graph cut optimization. *Pattern Recognition*, v. 47, p. 2894–2907, 2014.
- ZADEH, L. A. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on systems, man and cybernetics*. S1C-3 (1), p. 28-44, 1973.
- ZADEH, L. A., *Fuzzy Sets*, *Inf. Control* 8, p. 338-353, 1965.
- ZHI, R. et al. Graph-Preserving Sparse Nonnegative Matrix Factorization With Appli-

cation to Facial Expression Recognition. IEEE Systems, Man, and Cybernetics Society. IEEE, v.41, 2011.

ZINIANI, N. Projeto de Algoritmos: com implementação em pascal e C. Cengage Learning, 2004.