Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Programa de Pós-Graduação em Engenharia Eletrônica

Vivian de Oliveira Araujo

Clusterização através de Árvores de Padrões Fuzzy e Programação Genética Cartesiana

Rio de Janeiro

2017

Vivian de Oliveira Araujo

Clusterização através de Árvores de Padrões Fuzzy e Programação Genética Cartesiana

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao programa de Pós-Graduação em Engenharia Eletrônica da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Orientador: Prof. Dr. Jorge Luís Machado do Amaral

Rio de Janeiro

2017

CATALOGAÇÃO NA FONTE

UERJ / REDE SIRIUS / BIBLIOTECA CTC / B

B272 Araujo, Vivian de Oliveira.
Clusterização através de Árvores de Padrões Fuzzy e Programação Genética Cartesiana / Vivian de Oliveira Araujo. – 2017.
97f.

Orientador: Jorge Luís Machado do Amaral.
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Aprendizado de máquinas. 2. Árvores Fuzzy de Padrões– Dissertação. 3. Programação Genética Cartesiana– Dissertação. I. Amaral, Jorge Luís Machado do. II. Universidade do Estado do Rio de Janeiro. III. Título.

CDU 621.38

_____ _____

Assinatura Data

Vivian de Oliveira Araujo

Clusterização através de Árvores de Padrões Fuzzy e Programação Genética Cartesiana

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao programa de Pós-Graduação em Engenharia Eletrônica da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Aprovado em:

Banca Examinadora:

_____

Prof. Dr. Jorge Luís Machado do Amaral, D.Sc. (Orientador)
Faculdade de Engenharia – UERJ

_____

Prof. Dr. Alexandre Gonçalves Evsukoff, Ph.D.
COPPE UFRJ

_____

Prof. Dr. Douglas Mota Dias, D.Sc.
Faculdade de Engenharia – UERJ

_____

Prof. Dr. Marco Aurélio Botelho da Silva, D.Sc.
Faculdade de Engenharia – UERJ

Rio de Janeiro

2017

DEDICATÓRIA

A minha família, Lucia, Vitor e Leonardo por todo o apoio nesta jornada.

AGRADECIMENTOS

Agradeço:

Ao orientador desta dissertação, Professor Jorge Amaral pela perseverança, melhoria contínua e parceria para a realização deste trabalho.

À UERJ e ao Programa de Pós-Graduação em Engenharia Eletrônica pela oportunidade de realização deste curso.

Aos meus pais, que sempre priorizaram a educação e investiram seu tempo para me ajudar e nunca perderam o foco.

Ao Vitor, meu irmão por sempre estar disposto a me ouvir explicar um assunto desde pequeno.

Ao Leonardo, parceiro de muitos anos, por todo o apoio e por acreditar no meu potencial, mesmo quando eu não acreditava.

RESUMO

ARAUJO, VivianO. *Clusterização através de Árvores de Padrões Fuzzy e Programação Genética Cartesiana.* 2017. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2017.

Esta dissertação apresenta um modelo de clusterização fuzzy. Ao invés de utilizar a abordagem tradicional de sistemas fuzzy baseados em regras, foi utilizado o modelo de Árvore de Padrões Fuzzy*(APF),* que é um modelo hierárquico, com uma estrutura baseada em árvores que possuem como nós internos operadores lógicos *fuzzy* e as folhas são compostas pela associação de termos *fuzzy* com os atributos de entrada. O modelo sintetiza uma árvore para cada grupo, que será uma descrição "lógica" do grupo o que permite analisar e interpretar como é feita a clusterização. O método de aprendizado concebido utiliza Programação Genética Cartesiana onde a função de aptidão reflete a qualidade da clusterização obtida através de diferentes índices.O modelo proposto foi comparado com diferentes técnicas de clusterização tais como: k-means, k-medoids, hierárquico, Fuzzy C-means, Mapas de Kohonen e DBSCAN em bases de dados artificiais e do UCI *Machine Learning Repository*, tendo apresentado resultados competitivos. Ele também foi aplicado para resolver um problema de segmentação de mercado em uma operadora de telefonia com resultados promissores.

Palavras-Chave: Aprendizado de máquina, Árvores Fuzzy de Padrões, Programação Genética Cartesiana, Clusterização, Interpretabilidade.

ABSTRACT

This work presents o method for fuzzy clustering. Instead of the traditional fuzzy based rules, it was used a model called Fuzzy Pattern Trees (FPT), which is ahierarchical tree-based model, having as internal nodes, fuzzy logical operators and the leaves are composed of a combination of fuzzy terms with the input attributes. The method was obtained by creating a tree for each cluster, this tree will be a “logic class” description which allows the interpretation of the results. The learning method originally designed for FPT was replaced by Cartesian Genetic Programming where the fitness function reflects the quality of the clustering obtained through different indices. The FPT method was compared against other clustering techniques, such as: k-means, k-medoids, Agglomerative, Fuzzy C-means, Kohonen and DBSCAN on several datasets from artificial bases and the UCI Machine Learning Repository and it presented competitive results. It was also used to solve a segmentation problem from a mobile operator with promising results.

Keywords: Machine Learning, Clustering, Cartesian Genetic Programming, Clustering, Interpretability.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIAÇÕES

AG	*Algoritmos Genéticos*
APF	*Árvore de Padrões Fuzzy*
AuC	*Authentication Center*
BSC	*Base station subsystem*
CRM	*CustomerRelationship Management*
DB	*Database*
DBSCAN	*Density Based Spatial Clustering Applications with Noise*
EIR	*Equipment Identity Register*
FP	*Função de Pertinência*
GGSN	*Gateway GPRS Support Node*
GMSC	*Gateway Mobile Switching Center*
HLR	*Home Location Register*
IMS	*IP Multimedia Subsystem*
MDS	*Multidimensional Scaling*
MSC	*Mobile Switching Center*
PG	*Programação Genética*
PGC	*Programação Genética Cartesiana*
PSTN	*Public switched telephone network*
RNC	*Radio Network Controller*
SVM	*Support Vector Machine*
SGSN	*Serving GPRS Support Node*
SMSC	*Short Message Service Center*
VLR	*Visitor Location Register*

SUMÁRIO

INTRODUÇÃO

Segundo Tsiptsis(TSIPTSIS, 2011), o consumidor é o mais importante bem para uma empresa, não pode haver uma prospecção de negócio sem consumidores satisfeitos que permaneçam leais e que desenvolvam um relacionamento com a organização. Este é o motivo para que a organização planeje e implemente uma estratégia para tratar consumidores. O CRM (do inglês, *CustomerRelationship Management)* é uma estratégia para construir, gerenciar e fortalecer lealdade e uma relação de longa duração com o cliente.

O CRM deve possuir uma abordagem centralizada no cliente, seu escopo deve ser personalizado na identificação e compreensão de consumidores que possuem necessidades distintas, preferenciais e escolhas. Possui dois principais objetivos:

- Retenção do cliente por satisfação;
- Desenvolvimento por compreensão do cliente.

Para compreender melhor o cliente, uma das técnicas que pode ser utilizada é a segmentação de mercado. Ela consiste em obter um grupo de clientes que compartilham um conjunto semelhante de necessidades e desejos e pode ser realizada para campanhas das seguintes formas (KOTLER, 2003):

- Segmentação geográfica pressupõe que se criem programas de marketing sob medida para necessidades e desejos de grupos de clientes locais em áreas comerciais, bairros e até lojas individuais;
- Segmentação demográfica propõe que o mercado seja dividido por variáveis como idade, tamanho da família, ciclo de vida da família, sexo, renda, ocupação, grau de instrução, religião, raça, geração, nacionalidade e classe social. Uma razão para sua utilização é a facilidade de mensuração;
- Segmentação psicográfica divide os consumidores em diversos grupos em traços psicológicos/de personalidade, estilos de vida ou valores;
- Segmentação baseada no comportamento divide em grupos segundo seu conhecimento, atitude, uso ou reação a um produto.

A segmentação do mercado é uma aplicação popular da mineração de dados para segmentação de consumidores, o propósito é adaptar produtos, serviços e mensagens de marketing para cada segmento. Segundo Kotler, tradicionalmente, o mercado é

segmentado com base em pesquisas e parâmetros demográficos. O desafio é aplicar este conceito em clientes que não foram incluídos na pesquisa, ou mesmo, pessoas de um mesmo grupo demográfico podem exibir diferentes perfis psicográficos (BERRY, 1997), tornando a melhor abordagem,a segmentação por comportamento.

Alguns exemplos de campanhas alvo:

- Estimar quais clientes tem mais propensão para deixar a empresa;
- Estimar quais clientes tem mais chance para trocar ou iniciar o uso ou utilizar mais um produto, etc.

Para ser possível identificar estes segmentos, as técnicas de clusterização são utilizadas, pois elaé um método que tem como objetivo determinar um número finito de grupos que descrevem conjuntos de dados que estão agrupados de acordo com similaridades entre seus objetos.

A maior vantagem das técnicas de clusterização é ser possível gerenciar um grande número de atributos e criar segmentos guiados pelos dados analisados. Estes segmentos criados não são baseados em conceitos pessoais, intuições e percepções e, sim, na similaridade entre os dados. O objetivo é que os grupos detectados tenham uma homogeneidade interna e heterogeneidade entre os grupos. Entretanto, a clusterização é considerada o mais difícil e desafiador problema do aprendizado de máquina, devido a sua natureza não supervisionada, além dos algoritmos terem bastante sensibilidade à inicialização, podendo resultar em soluções que não sejam as melhores (HRUSCHKA, 2009). Em uma perspectiva de otimização, a clusterização pode ser formalmente considerada como um caso particular de grupamento NP-Difícil. Esta questão estimulou a busca por algoritmos eficientes de aproximação. Particularmente, algoritmos evolucionários são metaheurísticas amplamente consideradas eficazes em problemas do tipo NP-Difícil, provendo soluções quase ótimas para este tipo de problema em tempo razoável. Partindo desta premissa, uma grande quantidade de algoritmos evolucionários foi proposta na literatura, estes algoritmos são baseados na otimização da função objetivo (também chamada função de aptidão) que orienta a pesquisa evolucionária.

Os Algoritmos Evolucionários são inspirados no princípio darwiniano da evolução das espécies e na genética. Do mesmo modo que a Evolução Natural produz indivíduos mais aptos a sobreviver em um meio-ambiente, os Algoritmos Evolucionários podem ser vistos como procedimentos de otimização que melhoram o

desempenho de uma população de soluções em potencial em relação a um problema específico. Segundo este princípio, uma população de indivíduos evolui ao longo de gerações ou ciclos, pela sobrevivência dos mais aptos. Os principais Algoritmos Evolucionários são: os Algoritmos Genéticos, a Programação Genética, as Estratégias Evolutivas e a Programação Evolutiva (HRUSCHKA, 2009), (SANTOS; DO AMARAL, 2014).

Os Algoritmos Evolucionários desenvolvem soluções de clusterização que tendem a promover uma solução computacional mais eficiente e com maior qualidade que algoritmos tradicionais, pois utiliza informações de soluções já avaliadas para gerar soluções potencialmente, além de possuir a capacidade de encontrar soluções para problemas complexos e que envolvam um grande espaço de pesquisa.

Ao longo do processo evolucionário, o conjunto de potenciais soluções vai sofrendo alterações provocadas por operadores, geralmente denominados operadores genéticos, que permitem criar novos indivíduos a partir dos indivíduos já existentes. Uma vez que a aplicação continuada destes operadores aumenta progressivamente o tamanho do conjunto de potenciais soluções, em cada iteração é também aplicado um operador de seleção que escolhe probabilisticamente os melhores elementos da iteração anterior. Para que se possa aplicar este operador é necessário utilizar uma função de avaliação que, quando aplicada a um determinado elemento, indique o valor desse elemento enquanto solução para o problema. Este processo permite concentrar a pesquisa em zonas mais promissoras do espaço, aumentando progressivamente as possibilidades, não só de se encontrar uma solução para o problema, mas também de encontrar melhores soluções (GRILO, 2003).

Uma partição de um conjunto de dados é uma coleção de *kclusters* não sobrepostos dos mesmos. Usualmente o *k* é definido a priori pelo usuário, porém há algoritmos evolucionários que buscam a melhor quantidade de clusters, pois este valor não é conhecido.

Alguns trabalhos já publicados consideram a solução de problemas de clusterização quando o número de *clusters* já é conhecido anteriormente, por exemplo, com a utilização de k-means. Este método pode encontrar soluções sub-ótimas, sendo necessário que o algoritmo seja processado repetidamente com diferentes protótipos de inicialização. Só é possível garantir que a melhor solução foi encontrada se todos os protótipos forem avaliados, neste caso dependendo da complexidade dos dados e

quantidade dos mesmos este algoritmo pode ser ineficiente ou computacionalmente inviável (MURTHY, 1996), (BEZDEK, 1994), (KIVIJARVI, 2003), (KRISHINA, 1999), (FRANTI, 1997), (SCHEUNDERS, 1997), (KROVI, 1992), (BANDYOPADHYAY, 2002), (ESTIVILL-CASTRO, 1997), (SHENG, 2004) e (LU, 2004).

Outros trabalhos utilizam algoritmos em que a quantidade de *clusters* não é conhecida a priori. Estes algoritmos têm por objetivo encontrar um número ideal de grupos e a sua solução correspondente (NALDI, 2007), (PAN, 2007), (MA, 2007), (HRUSCHKA, 2003), (HANDL, 2007), (COLE, 1998), (COWGILL, 1999), (CASILLAS, 2003) e (ALVES, 2006).

As soluções geradas são armazenadas em *clusters* e, cada nova solução deve ser incluída no *cluster* mais relacionado de acordo com uma métrica de distância. Cada *cluster* possui uma solução central que o representa, e vai sendo preenchido com soluções até que um limiar seja atingido. Nesse momento, acredita-se que esse *cluster* indica um espaço promissor de busca e, então, um procedimento de busca local é aplicado à solução central (DE OLIVEIRA, 2007).

A representação do genótipo é similar tanto quando a quantidade de *clusters* é conhecida inicialmente quanto quando o algoritmo o define. A representação pode ser: binária onde cada solução de *cluster* é representada como uma *string* binária de tamanho *N*, onde *N* é o número de objetos (cada posição corresponde a um objeto), também pode ser inteira representada de duas formas: como um vetor com *N* posições onde *N* é a quantidade de objetos onde cada solução de *cluster* ou vetorde *k* elementos, em que a representação é realizada por números reais associada ao centroide, neste caso o espaço em memória para representar o genótipo aumenta em relação aos outros métodos de representação (HRUSCHKA, 2009).

Já a função de aptidão, para o caso em que a quantidade de *clusters* é fixa,pode propor a maximizar a distância entre *k* medoids (ESTIVILL-CASTRO, 1997), (LUCASIUS, 1997) e (SHENG, 2004) ou mesmo, minimizar o somatório da distância euclidiana ao quadrado (BANDYOPADHYAY, 2002), (MAULIK, 2000), (MERZ, 2002) e (MURTHY,1996) ou avaliar a região de alta densidade de dados (correspondente a um *cluster*) separada por uma região de baixa densidade de dados (DE OLIVEIRA, 2007) e (ESTER, 1996).

No caso que a quantidade de *clusters* não é conhecida a priori, a função de aptidão pode ser baseada no critério de variância (CASILLAS, 2003), (COLE,2008) e (COWGILL, 1999), na distância intra *cluster* e inter *cluster* (TSENG, 2001), entre outros métodos.

Muitos algoritmos de clusterização requerem que o usuário insira certos parâmetros como o número de *clusters* e a dimensionalidade média do *cluster*, que são dados que não são difíceis de determinar, porém não muito práticos dependendo da quantidade de dados envolvida (AGGARWAL, 2000) e (ZHANG, 2004). Em algoritmos hierárquicos, o usuário avalia posteriormente a solução e, assim, a quantidade de *clusters*, com o resultado da partição dos dados. Consequentemente, a saída destes algoritmos é muito sensível ao conhecimento do usuário (BOCK, 1996) e (LEE, 2000).

Para superar estes inconvenientes, uma subárea dos Algoritmos Evolucionários usada é a Programação Genética (PG). A PG é uma técnica que permite que computadores resolvam problemas sem que precisem ser explicitamente programados para tal (KOZA, 1992). A PG parte de uma declaração de alto nível sobre “o que se necessita ser feito” e cria automaticamente um programa de computador para resolver o problema. Este mecanismo também é conhecido como “programação automática” (DIAS, 2010).

A Programação Genética Cartesiana (PGC) (MILLER, 2009) é uma forma de programação genética na qual os programas são representados por uma grade bidimensional de nós, ou seja, por grafos acíclicos direcionados.

O benefício da utilização de grafos é o fato de que grafos são mais gerais, flexíveis e compactos e podem ser aplicados em diversos domínios (DHARWADKER; PIRZADA, 2011), neste tipo de representação há a reutilização implícita dos nós pertencentes ao grafo direcionado.

Dentre as vantagens da PGC está a característica de neutralidade que é responsável porminimizaro*bloat*, que é comum em outros métodos de programação genética (BANZHAF, 1994), (MILLER; SMITH, 2006),(MILLER, 2001). Neste caso, é preferível árvores mais compactas e de melhores aptidões, que por consequência exigem um menor esforço computacional.

As vantagens adicionais da PGC são: foco em conceitos e interpretação de problemas como um programa de computador e capacidade de encontrar dependência e

independência de variáveis e estabelecer relações entre as mesmas (YUVARAJU, 2013).

Os campos de aplicação de algoritmos de clusterização evolucionários são essencialmente o mesmo que os de algoritmos não evolucionários, porém o uso de estratégias evolucionárias é aparentemente mais apropriado quando o número aproximado de *clusters* não é conhecido (HRUSCHKA, 2009).

Quando o conjunto de dados pode ser representado de forma eficiente em um grafo, pode-se abordá-lo como um problema de agrupamento de dados em grafos. Esse problema consiste em definir grupos de nós que apresentem alta conectividade intra-grupo e baixa conectividade inter-grupo (KAWAJI, 2004), (KRAUSE, 2005) e (HUTTENHOWER, 2007). Este modelo tem uma vantagem não encontrada em muitos algoritmos e modelos de agrupamento de dados: a possibilidade da avaliação da qualidade das partições geradas, sempre tendo um limitante superior (ótimo) para ser atingido, o que é uma questão simples e direta para modelos de agrupamento (NASCIMENTO, 2010).

Na Figura1 é demonstrado em (a) um possível genótipo traduzido em uma estrutura de grafo, já em (b) apenas para auxílio na compreensão de como se origina de o genótipo, finalmente em (c) cada componente ligado dentro deste gráfico é finalmente interpretado como um conjunto individual, tal como visualizado pelas elipses.

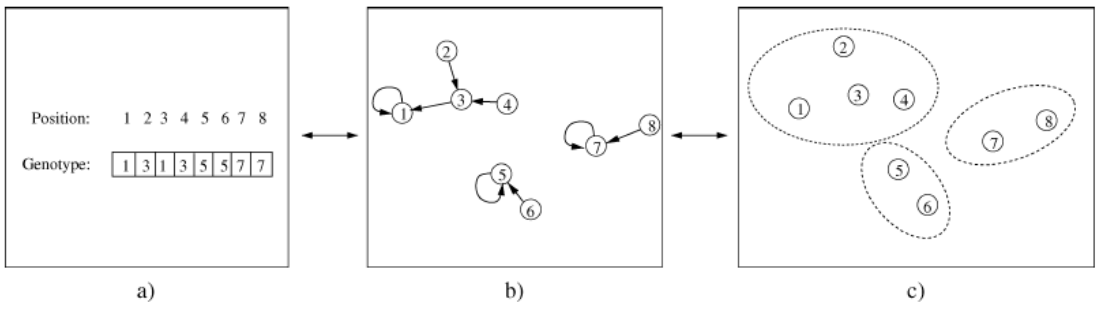


Figura1: Representação de grafo (HANDL, 2007).

Neste trabalho, ao invés de utilizar a codificação proposta por Handl, optou-se por representar Árvore de padrões *fuzzy* (APF), onde cada árvore representará um *cluster*. A APF é um modelo hierárquico, com uma estrutura similar a uma árvore, em que os nós são marcados com operadores lógicos *fuzzy* e operadores matemáticos e as folhas são compostas por termos *fuzzy* associadas ao atributo de entrada. Um nó assume os valores

de seus descendentes como entrada e realiza uma combinação usando o operador escolhido e envia a saída para o seu sucessor. A vantagem deste método se refere à interpretação do resultado, comumente cada árvore pode ser considerada como uma descrição lógica de um grupo (SENGE; HÜLLERMEIER, 2011).

O uso de APFs é interessante por possuir um mecanismo de seleção de atributos embutidos, além disso, elas são atrativas do ponto de vista de interpretabilidade (SENGE; HÜLLERMEIER, 2011).

Além de avaliar o modelo proposto em bases de dados artificiais e outras frequentemente usadas na literatura, este trabalho também propõe a aplicação deste método em um caso de segmentação de mercado. O propósito da segmentação é adaptar produtos, serviços e mensagens de marketing para cada segmento. A divisão de clientes em segmentos tem sido tradicionalmente baseada em pesquisa de mercado e dados demográficos. Pode haver um segmento "jovem e solteiro" ou um "segmento leal ao produto", o problema com segmentos com base em pesquisa de mercado é que é difícil saber como aplicá-los a todos os clientes que não faziam parte do inquérito. O problema com cliente segmentados com base em dados demográficos é que nem todos os "jovens e solteiros" ou "ninhos vazios" na verdade tem os gostos e afinidades de produtos atribuído ao seu segmento. A abordagem de mineração de dados é identificar o comportamento destes segmentos.

Os trabalhos relacionados na área de pesquisa de clusterização para segmentação são apresentados conforme abaixo:

- Em (JANSEN, 2007), os algoritmos K-means, K-medoids e Fuzzy C-means são utilizados para obter dois tipos de informações à segmentação dos clientes (hábitos, preferências, etc.) e agrupar por perfil dos clientes (idade, gênero, valores, etnia, etc.), o objetivo é definir quais ações de *marketing*serão adotadas para cada segmento. Para determinar a quantidade de *clusters* foramutilizados os métodos de validação tradicionais, como o índice de Dunn;
- Em (CHAN, 2005), é estudado o caso de leilão *online* utilizando Redes auto-organizáveis de Kohonen, o objetivo do modeloé auxiliar os compradores a submeter um lance vencedor a um preço razoável, considerando o comportamento do vendedor. Atualmente, *sites* de leilão fornecem, normalmente, a identificação do usuário. Os grupos foram divididos em três

tipos de ofertas: pacientes, impulsivas e analíticas, com base neste modelo, um concorrente pode tomar uma decisão mais acertada;

- Em (VENKATESAN, 2007) é tratada a questão de seguro para automóveis, o objetivo é segmentar os clientes de acordo com características valorizadas para a escolha do produto. O K-means foi o algoritmo utilizado epara a escolha da quantidade de *clusters* foi utilizada a análise do somatório das distâncias de cada ponto ao centro de *cluster*, esta avaliação mostra o número de *clusters* contra o somatório das distâncias dos pontos aos centros dos clusters, onde há uma depressão ou "joelho", representa o número estimado de *clusters* a ser utilizado;
- Em (HSIEH, 2004), é apresentado um modelo utilizando Redes auto-organizáveis de Kohonen para tratar dados bancários, o objetivo é prever o comportamento de um cliente existente em relação a um produto de crédito disponível. É utilizado o algoritmo Apriori após a clusterização para descobrir as relações possíveis entre os atributos, focando em características demográficas e geográficas para a construção e manutenção da base de clientes mais rentáveis.

Este trabalho apresenta um método para clusterização que sintetiza Árvores de Padrões Fuzzy (APF) de forma automática.O método de aprendizado de APF foi substituído pela Programação Genética Cartesiana (PGC). A PGC é um método de busca global capaz de explorar espaços de busca bastante grandes de forma eficiente e a representação dos programas na forma de grafos pode ser facilmente utilizada para representar APFs.Foram realizados diversos estudos de casos para obter uma melhor compreensão do funcionamento deste método, desenvolver estratégias eavaliar o desempenho em relação a métodos tradicionais de clusterização.

O restante desta dissertação está dividido da seguinte forma: oCapítulo 1 apresenta conceitos dos algoritmos tradicionais de clusterização, o Capítulo 2 introduz as Árvores de Padrões Fuzzy, oCapítulo 3 descreve a Programação Genética Cartesianaintroduz as Árvores de Padrões Fuzzy, o Capítulo 4 apresenta o modelo proposto, o Capítulo 5 discute sobre os resultados obtidos e o Capítulo 6 apresenta a conclusão e a sugestão de trabalhos futuros.

1 CLUSTERIZAÇÃO

A clusterização é um método que tem como objetivo determinar um número finito de grupos que descrevem conjuntos de dados que estão agrupados de acordo com similaridades entre seus objetos.

Para ilustrar o desafio da clusterização, o número total de diferentes formas de agrupamento de *n* elementos de um conjunto em *kclusters*, conforme exposto em (CHIOU, 2001) e (COLE, 1998) equivale à função *N*(*n*, *k*) apresentada em:

$$N(n,k) = \frac{1}{k!}\sum_{i=0}^{k}\binom{k}{i}(-1)^i(k-i)^n \qquad (1)$$

É importante observar que há um crescimento exponencial do número de soluções possíveis para um problema de k-clusterização, considerando a equação (1), para combinar 10 elementos em 2 *clusters*, 100 elementos em 2 *clusters*, 100 elementos em 5 *clusters* e 1000 elementos em 2 *clusters*, tem-se N(10, 2) = 511, N(100, 2) = $6{,}33825 \times 10^{29}$, N(100, 5) = $6{,}57384 \times 10^{67}$ e N(1000, 2) = $5{,}3575 \times 10^{300}$ formas diferentes.

Para o problema de clusterização automática o número total de combinações sofre um incremento significativo, sendo definido de acordo com a equação:

$$N(n) = \sum_{k=1}^{n}\frac{1}{k!}\sum_{i=0}^{k}\binom{k}{i}(-1)^i(k-i)^n \qquad (2)$$

Dessa forma, para um conjunto com 10 elementos, a clusterização automática tem que considerar 115.975 diferentes maneiras de combinar os elementos em um número de *clusters* que pode variar de 1 a 10.

Já a similaridade é comumente definida em termos de o quão "perto" entre si os dados estão no espaço, sendo assim, normalmente a similaridade é expressa através de medidas de distância, que, por exemplo, podem estar relacionadas ao coeficiente de correlação ou à distância euclidiana.

A "qualidade" de cada *cluster* pode ser representada em relação à máxima distância entre dois objetos em um *cluster*. A distância do centro é uma medida alternativa para medir a qualidade do *cluster* e é definida pela média da distância de cada dado para o centro do *cluster* (HAN, 2001). Diferentemente das funções de distância baseadas em correlação, a distância euclidiana leva em consideração a magnitude das diferenças dos valores dos dados. Dessa forma, ela preserva mais informação sobre os dados e pode ser preferível (LOPES, 2004).

A similaridade pode ser calculada utilizando a distância "*city-block*" (*Manhattan)* que corresponde à soma das diferenças entre todos os atributos de dois elementos (*x* e *y*), não sendo indicada para os casos em que existe uma correlação entre tais atributos (OCHI, 2004).

A similaridade também pode ser dada pela correlação de Pearson que é uma medida para o quão bem uma linha reta pode ser adequada para um gráfico de dispersão (*scatterplot*) de *x* e *y*. Se todos os pontos no gráfico de dispersão repousam sobre uma linha reta, o coeficiente de correlação de Pearson é ou +1 ou -1, dependendo se a inclinação da linha é positiva ou negativa. Se o coeficiente de correlação de Pearson é igual a zero, não existe correlação linear entre *x* e *y*. A correlação de Pearson automaticamente centraliza os dados pela subtração da média, e os normaliza pela divisão pelo desvio padrão. Essa normalização é útil em várias situações, porém existem casos em que a magnitude dos atributos precisa ser preservada. (LOPES, 2004).

Para cada algoritmo avaliado, uma técnica de cálculo de similaridade pode ser empregada, a escolha da técnica de similaridade afeta a forma com que os grupos são formados, portanto a escolha correta da medida tem papel fundamental no sucesso da clusterização em encontrar os grupamentos presentes nos dados, como exemplo, para a análise de documentos é comumente utilizada a correlação de Pearson descentralizada ou cosseno do ângulo entre dois vetores de dados, a distância euclidiana não é utilizada, pois os valores dos atributos não são escalares.

Assim, para cada tipo de conjunto de dados será avaliada a medida que apresentará a melhor distribuição de *clusters* para o conjunto de dados, nesta avaliação serão utilizados os índices de avaliação que serão apresentados no final deste Capítulo.

A Figura 2 exemplifica uma representação de uma clusterização utilizando a distância do centro como medida de similaridade.

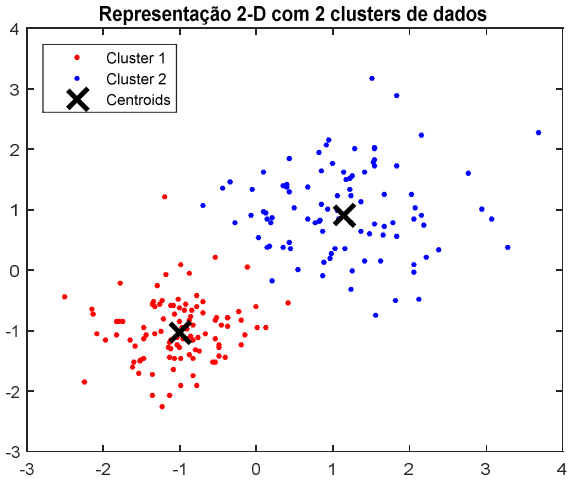


Figura 2: Representação 2-D com 2 *clusters* de dados. O centro de cada *cluster* está marcado com "X" (DO AUTOR).

O modelo de clusterização é incluído em métodos não supervisionados, e como os grupos não são conhecidos preliminarmente, o algoritmo analisa o padrão de dados de entrada e identifica um agrupamento, podendo analisar dados comportamentais, identificar grupos de consumidores ou mesmo, sugerir uma solução baseada nos padrões dos dados. Esta solução, se construída corretamente, pode revelar grupos com distintas características e conduzir a uma segmentação com maior significado e valor para o negócio (TSIPTSIS, 2011), (HAN, 2011). Pode ser utilizado também para diversas aplicações, como: processamento de imagens (SUMAN, 2008), (FRANEK, 2011), (ZHANG, 2008), bioinformática (CHENG, 2000), (EREN, 2013), (EISEN, 1998), categorização de documentos (YANG, 1997), (LEWIS, 1992).

As técnicas de clusterização podem ser divididas em três tipos principais: *overlapping*(cada dado pode pertencer a mais de um grupo), particional (normalmente, produzem *clusters* por meio da otimização de uma função) e hierárquico (constrói uma hierarquia de *clusters*, isto é, uma árvore de *clusters*). Os dois últimos são relacionados já que o modo hierárquico é formado por uma sequência de *clusters* particionais.

Na Figura 3 são descritos o exemplo de árvore de clusters utilização os dois tipos de clusterização hierárquica que será abordada em item posterior.

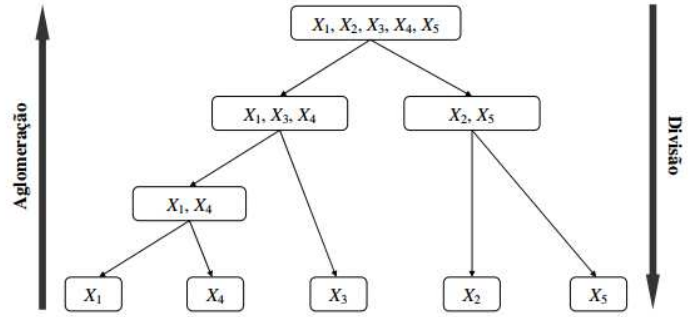


Figura3: Exemplo de árvore de *clusters* na clusterização hierárquica(TSIPTSIS, 2011).

Formalmente, este problema pode ser definido da seguinte maneira: dado um conjunto formado por $X= \{x_1,x_2, ..., x_N\}$, com cada objeto $x_i$possuindo p atributos (dimensões ou características), ou seja, $\overrightarrow{x_i} = \{ x_i^{\,1}, x_i^{\,2}, ..., x_i^{p}\}$, deve-se construir k grupos $C_j = \{1, ..., k\}$ a partir de $\vec{X}$, onde os objetos de cada grupo sejam homogêneos segundo alguma medida de similaridade.

$\bigcup_{i=1}^{k} Ci = X$ (3)

Ci ∩Cj= ∅ i,j=1, …, k e i≠j (4)

$Ci \neq \emptyset i = 1, \dots, k$ (5)

Estas restrições determinam, respectivamente, que: o conjunto X corresponde à união dos objetos dos grupos, cada objeto pertence a exatamente um grupo e todos os grupos possuem ao menos um objeto.

Se a condição 5 for flexibilizada, as partições de dados são do modo *overlapping*, como o algoritmo Fuzzy C-means que será detalhada em item posterior (onde cada dado pode pertencer a mais de um grupo) (SEMAAN, 2012).

Cada algoritmo de agrupamento possui suas próprias características, o que gera uma grande variedade de soluções possíveis para o agrupamento de um determinado conjunto de dados. Alguns algoritmos possuem parâmetros livres que influenciam na partição obtida, o que gera uma diversidade ainda maior de soluções. Dependendo da aplicação, se existir algum conhecimento prévio sobre o conjunto de dados, ele pode ser utilizado na análise. Também pode não ser possível agrupar toda a base de dados de uma única vez. Nesses casos, é importante que um agrupamento final seja gerado a partir de agrupamentos obtidos separadamente (NALDI et al, 2009).

Os algoritmos analisados serão apresentados nos itens que se seguem, todos os algoritmos foram executados no software Matlab versão R2011a.

1.1 Métodos Particionais

Os métodos particionais produzem *K* agrupamentos com de *n* objetos, geralmente otimizando uma função objetivo. Uma desvantagem do método é que a função objetivo usadatem como requisito o número de agrupamentos, *K*, como entrada.

Os métodos particionais mais conhecidos e utilizados são: K-means, K-medóides e suas variações (HAN, 2011).

1.1.1Redes auto-organizáveis de Kohonen

Os mapas auto-organizáveisde Kohonen, também denominados de SOM (*self-organizing map*) são considerados uma arquitetura de redes neurais artificiais de estrutura articulada, com aprendizado competitivo e não supervisionado (SILVA; FLAUZINI, 2010).

Em um mapa auto-organizável, os neurônios estão colocados em nós de uma grade que é normalmente uni ou bidimensional, os mapas de dimensionalidade mais alta são também possíveis, mas não são tão comuns (HAYKIN, 2011).

Os processos auto-organizados possuem algumas características básicas, como: coletividade, onde unidades que fazem parte deste coletivo competem, com chances de sucesso semelhantes, por recursos limitados. (COSTA, 1999).

Enquanto o treinamento da maior parte das redes neurais necessita que padrões de entrada e padrões de saída sejam conhecidos (treinamento supervisionado), os mapas auto-organizáveis de Kohonen utilizam apenas padrões de entrada para realizar seu treinamento (treinamento não supervisionado). Esse tipo de rede é útil em aplicações onde somente os padrões de entrada sejam conhecidos, não existindo padrões de saída para serem relacionadas à entrada (HONKELA, 1997).

A estrutura de uma rede neural SOM inclui a camada de entrada, camada de saída e pesos. A camada de entrada contém os nós de entrada , a camada de saída inclui os nós de saída , além disso, são definidos pesos entre a camada de entrada e a camada de saída (XIAO, 2012).

As redes SOM utilizam métodos de treinamento competitivo para detectar similaridade e correlacionar os padrões do conjunto de dados de entrada, agrupando esses dados em grupos (*clusters*). O processo de treinamento competitivo consiste em "premiar" o neurônio cujo vetor de pesos estiver mais próximo do vetor de entrada aplicado à rede. O prêmio do vencedor é o ajuste de seus pesos, fazendo com que o vetor de pesos fique mais próximo do vetor de entrada, de modo que quando uma entrada semelhante for apresentada, este neurônio tem mais chances de ser o vencedor. Uma das métricas de similaridade normalmente utilizada é a distância euclidiana entreesses dois vetores(SILVA; FLAUZINI, 2010).

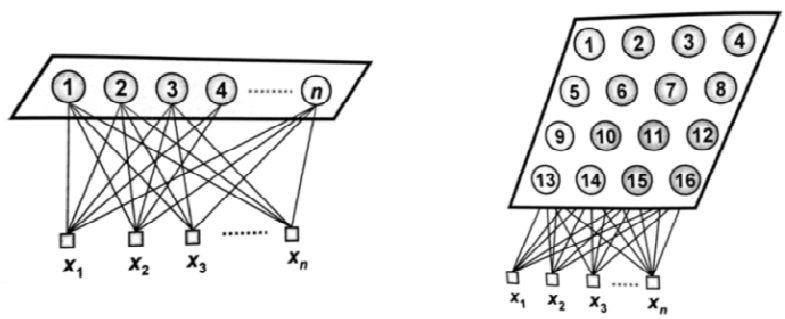
O neurônio que obtiver a menor distância em relação ao vetor de entrada será declarado vencedor e seus pesos serão ajustados de forma que ele se aproxime ainda mais do vetor de entrada (SILVA; FLAUZINI, 2010).

As conexões laterais são usadas para que um neurônio vencedor possa influenciar (colaborar) na resposta produzida pelos demais neurônios. A influência exercida pela conexão lateral entre dois neurônios vizinhos será proporcional a distância entre eles. As conexões laterais são fornecidas por mapas topológicos de vizinhança (SILVA; FLAUZINI, 2010).

A Figura 4 mostra duas topologias para a implementação do mapa auto-organizávelFigura4: Exemplos de topologias linear (a) e em duas dimensões (b) (SILVA; FLAUZINI, 2010).

(a) possui *n* neurônios organizados em uma dimensão, enquanto que na Figura4: Exemplos de topologias linear (a) e em duas dimensões (b) (SILVA; FLAUZINI, 2010).

(b) os neurônios estão organizados em um mapa de duas dimensões. Nas duas Figuras o vetor de $\vec{x}=x_1..\ x_n$, representa o dado de entrada que será apresentado a todos os neurônios da topologia (SILVA; FLAUZINI, 2010).



(A) (B)

Figura4: Exemplos de topologias linear (a) e em duas dimensões (b) (SILVA; FLAUZINI, 2010).

Adicionalmente à topologia, é necessário especificar o critério de vizinhança entre os neurônios. Essa informação irá indicar como os neurônios irão cooperar com seus vizinhos (SILVA; FLAUZINI, 2010).

Considerando que um determinado neurônio venceu a competição, para uma amostra de entrada, seu vetor de pesos e dos seus vizinhos serão ajustados. O maior ajuste será feito para o neurônio vencedor, ao passo que seus vizinhos serão ajustados com taxas menores (SILVA; FLAUZINI, 2010).

O mecanismo de Kohonen funciona da seguinte forma: os pesossinápticos iniciam contendo valores aleatoriamente baixos, e um sinal de entrada x (com valores que representam uma informação qualquer) é provido para a rede sem que se especifique a saída desejada (característica da rede não supervisionada). O sinal de entrada $\vec{x}$com dimensão *m*é descrito como (GONÇALVES, 2007):

$$\vec{x} = [x_1, x_2, \dots, x_m]^T \quad (6)$$

A rede consiste essencialmente de duas camadas: uma camada de entrada *I* e uma camada de saída *U* (camada de Kohonen). A entrada da rede corresponde a um vetor p-dimensional, $\vec{x}$, geralmente no espaço R^p. Todas as *p* componentes do vetor de entrada alimentam cada um dos neurônios do mapa. Cada neurônio *i* pode ser representado então por um vetor de pesos $\overrightarrow{w_i} = [w_{i1}, w_{i2}, \dots, w_{im}]^T$ também no espaço p-dimensional possui a mesma dimensão do vetor de entrada. Para cada padrão de entrada um neurônio é escolhido o vencedor, *c*, usando o critério de maior similaridade (COSTA, 1999):

$$\|\vec{x} - w_c\| = \min\{\|x_i - \overrightarrow{w_i}\|\} \quad (7)$$

A representação da distância Euclidiana é dada por$\|.\|$, já os pesos do neurônio vencedor, juntamente com os pesos dos seus neurônios vizinhos, são ajustados de acordo com a seguinte equação (COSTA, 1999):

$$\overrightarrow{w_i}(t+1) = \overrightarrow{w_i}(t) + h_{ci}(t)[x(t) - \overrightarrow{w_i}(t)] \quad (8)$$

Na equação 8, t indica a iteração do processo de treinamento, $x(t)$é o padrão de entrada e $h_{ci}(t)$é o núcleo de vizinhança em torno do neurônio vencedor.

Ao final do treinamento espera-se que o mapa esteja topologicamente ordenado, ou seja, n_i padrões que estejam próximos no espaço p-dimensional de atributos devem ser mapeados em neurônios que estejam próximos no espaço do *grid*, geralmente no mesmo neurônio ou em neurônios vizinhos, porém dois neurônios vizinhos no espaço do *grid* podem estar distantes no espaço de atributos (COSTA, 1999).

Na Figura 5, gerada a partir do software Matlab é a representação da matriz *U*, as linhas vermelhas conectam neurônios vizinhos. As cores nas regiões que contêm as linhas vermelhas indicam as distâncias entre os neurônios, as mais escuras representam distâncias maiores e as mais claras representam distâncias menores.

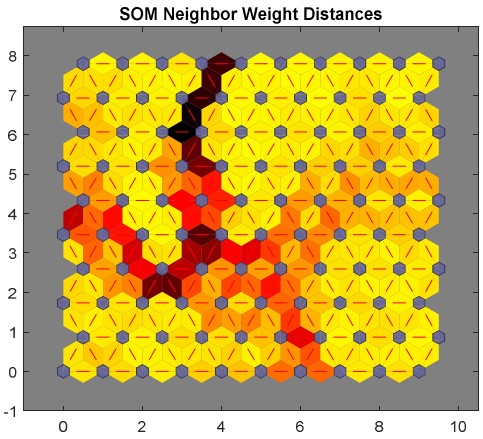


Figura 5: Representação das distância entre os neurônios (DO AUTOR).

Uma vez garantido que não sejam observadas mudanças significativas no mapa formado, a convergência do seu algoritmo de aprendizagem que pode ser realizada, o arranjo de neurônios do SOM reflete características estatísticas importantes do espaço de entrada. As principais propriedades da rede podem ser resumidas como seguem (HAYKIN, 2001):

- Aproximação do espaço de entrada: o SOM tem como objetivo básico armazenar um conjunto grande de vetores de entrada encontrando um conjunto menor de protótipos (vetores de pesos sinápticos w_i) de modo a fornecer uma boa aproximação para o espaço de entrada original;
- Ordenação Topológica: ao realizar o mapeamento não-linear dos vetores de entrada para o arranjo de neurônios da rede, o algoritmo do SOM tenta preservar ao máximo a topologia do espaço original, ou seja, procura fazer com que neurônios vizinhos no espaço de saída apresentem vetores de pesos que representem padrões vizinhos no espaço de entrada;
- Casamento de Densidade: o mapeamento efetuado pelo SOM reflete a distribuição de probabilidade dos dados no espaço de entrada original. Regiões do espaço de entrada de onde os vetores de amostra são retirados com uma alta probabilidade de ocorrência são mapeadas para domínios maiores no espaço de saída, e, portanto, com melhor resolução que regiões no espaço de entrada de onde vetores de amostra x são retirados com uma baixa probabilidade de ocorrência.

O algoritmo pode ser descrito simplificadamente conforme abaixo:

1. Início: Gerar os vetores de pesos iniciais $x_i, i = 1, \ldots, n$, onde n é o número de neurônios;
2. Um padrão de entrada $x_k = (\xi_1, \xi_2, .., \xi_p)$, $x_k \in R^p$ é selecionado aleatoriamente de todo conjunto de padrões;
3. Uma função de ativação é usada para calcular o estado de cada neurônio i em relação ao padrão x_k. Usando a distância euclidiana, tem-se:

$$d(m_i, x_k) = \sqrt{\sum_{j=1}^{p} [x_{kj}(t) - x_{ij}(t)]^2}$$

4. O neurônio vencedor, c, é escolhido de acordo com a equação:

$$\|\vec{x} - w_c\| = \min\{\|x_i - \overrightarrow{w_i}\|\}$$

5. Os pesos sinápticos do neurônio vencedor, c, como também os pesos dos neurônios que estão dentro da vizinhança de c são atualizados através da equação:

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

6. O termo h_c é uma função decrescente com o tempo e com a distância do neurônio i ao neurônio vencedor c, formado pela taxa de aprendizado $\alpha(t)$, função de vizinhança $h(d,t)$ e r_i é a posição do neurônio i na camada de Kohonen:

$$h_{ci}(t) = \alpha(t).h(\|r_c - r_i\|, t)$$

7. Repetir de 2 a 5 até o algoritmo de treinamento convergir.

Figura6: Algoritmo simplicado Kohonen (COSTA, 1999).

1.1.2K-means

O método K-means é muito popular para clusterização em geral (ZHA, 2001), seu objetivo é a minimização de uma medida de custo, que é o somatório das distâncias dos padrões de um grupo aos seus respectivos centros, ou seja, minimizar a variância. A minimização do custo garante encontrar um mínimo local da função objetivo, que dependerá do ponto inicial do algoritmo (DE CASTRO, 2002).

O método requer que se determine previamente o número de *clusters* que deve ser formado, por isso, é necessário que se faça muitos testes para validar o modelo, a solução ótima requer uma análise preliminar e a avaliação de diferentes soluções (HAN, 2011).

O algoritmo K-means utiliza como medida de similaridade mais comum a distância euclidiana e iterativamente calcula a distância entre todos os dados. O processo se inicia selecionando *k* dados iniciais como centros de *clusters*, cada dado é agrupado para o *cluster* mais próximo. À medida que novos dados são adicionados ao *cluster*, o centro é recalculado para refleti-los, este processo iterativo é repetido até a convergência e finalização da migração dos dados entre os *clusters*(HAN, 2011).

Normalmente, os *clusters* são obtidos pela otimização de uma função objetivo e a aproximação dos dados pode ser analisada a partir da formação de uma matriz de distância ou das similaridades de acordo com a métrica de distância estabelecida.

Embora o algoritmo convirja sempre, quando os grupos não são hiperesféricos e bem separados, ele pode não encontrar o agrupamento ideal, isto é, obtem-se um mínimo global da função objetivo. O maior problema com este algoritmo é que é significativamente sensível à seleção inicial dos centros dos grupos podendo convergir para um sub-ótimo local da função objetivo se os centróides não forem bem escolhidos inicialmente.Uma solução para tentar reduzir este efeito é executá-lo múltiplas vezes e escolher o agrupamento de dados que minimize o erro quadrático (DUARTE, 2008).

Este algoritmo só pode ser aplicado a dados que possibilitem definir a média de cada um dosgrupos, além de não ser apropriado para descobrirgrupos com formatos não convexos ou grupos de tamanhos muito diferentes.Além disso,ométodo k-means tem sensibilidade quando existirem *outliers* na distribuição de dados, (GANDHI, 2014)já que um pequeno número deste tipo de dados pode influenciar substancialmente o centroide do grupo, uma vez que este é representado pela média, que é uma medida bastante sensível a *outliers*(DUARTE, 2008).

Este método de clusterização é muito usado em análise exploratória de dados e mineração de dados em qualquer campo de pesquisa, especialmente com o crescimento da capacidade computacional alinhado ao aumento da ocorrência de grande base de dados. Por ser de fácil implementação, eficiente computacionalmente e com baixo consumo de memória, é um algoritmo muito utilizado para clusterização, podendo ser usado como etapa inicial para algoritmos com maior complexidade computacional(COSTA, 1999).

O algoritmo pode ser descrito simplificadamente conforme abaixo:

1. Início: Escolher o número de agrupamentos, K, e os valores iniciais dos K protótipos ou vetores de média, $\overline{x_k}(0),\ k = 1, 2, \ldots, K$. Outros parâmetros que podem fazer parte da inicializados do algoritmo: o número máximo de iterações, $t_{máx}$ e um valor para erro, ε;
2. Para $t = 1, \ldots, t_{máx}$, classifique os objetos $x_i, i = 1, 2, \ldots, n$, como pertencentes ao agrupamento C_k que satisfaça a equação:

$$u_{ik}(t) = 1 \Leftrightarrow \|x_i - \overline{x_k}(t-1)\|^2 < \left\|x_i - \overline{x_j}(t-1)\right\|^2, k \neq j, j = 1, 2, \ldots, K.$$

3. Determine o valor da função objetivo com a partição obtida, minimização do erro quadrático;
4. Recalcule os vetores de médias, $\overline{x_k}(0),\ k = 1, 2, \ldots, K$, baseado na informação $u_{ik}(t)$;
5. Repetir os passos 2 a 4 enquanto $t < t_{máx}$ ou $u_{ik}(t) - u_{ik}(t-1) \neq 0$ ou $|E_K^2(t) - E_K^2(t-1)| > \varepsilon$.

Figura7: Algoritmo simplicado K-means (COSTA, 1999).

Considerando um conjunto de dadosconforme Figura 8 (a). Seja *K*=3 e considerando o algoritmo da Figura7, escolhe-se arbitrariamente três objetos como os três centros iniciais de grupo (marcados com um "+"), cada objeto é incluído no grupo com centro mais próximo, tal distribuição forma silhuetas em círculo como se vê na Figura8 (a) (DUARTE, 2008).

Este novo agrupamento irá procurar uma atualização dos centros de massa dos grupos. Isto é, o valormédio de cada grupo é recalculado baseado nos objetos que ficam no grupo. Relativamente a estes novos centros, os objetos são redistribuídos pelos grupos com centro mais perto. Tal redistribuição forma novas silhuetas em forma de círculo com curvas tracejadas, como se mostra na Figura8(b) (DUARTE, 2008).

Este processo é iterativo, conduzindo à Figura8 (c), quando não se verifica nenhuma redistribuição dos objetos o processo termina (DUARTE, 2008).

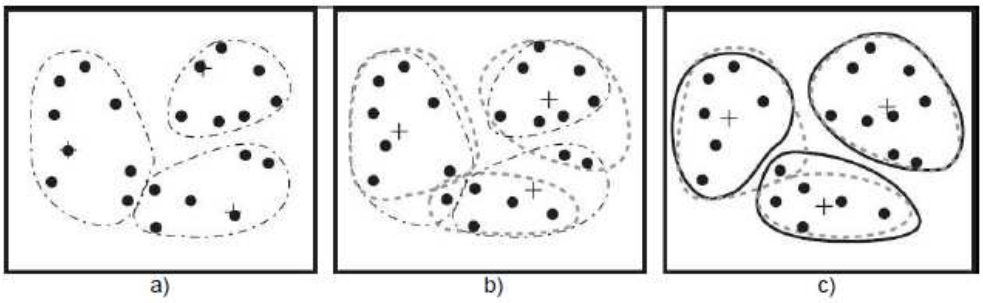


Figura8: Exemplo de clusterização com o algoritmo k-means (DUARTE, 2008).

1.1.3 Fuzzy C-means (FCM)

O algoritmo FCM tem como objetivo encontrar grupos *fuzzy* para um conjunto de dados. Para alcançar este objetivo, o algoritmo precisa minimizar uma função que diz respeito à minimização das distâncias entre os dados e os centros dos grupos aos quais tais dados pertencem com algum grau de pertinência (YONAMINE, 2002) (XU, 2005).

Um conjunto *fuzzyA* definido no universo de discurso *X* é caracterizado por uma função de pertinência μ_A, a qual mapeia os elementos de *X* para o intervalo [0,1].

No algoritmo,*k* agrupamentos são representados como um conjunto$C=\{C_1, .., C_k\}$ de vetores chamados “protótipos” e determinam os centros dos *clusters*. Cada vetor protótipo sempre está associado à representação de um grupo do conjunto de dados e, para isso, deve residir no mesmo espaço R^p que os dados do conjunto. O conjunto *C* é representado por uma matrizde dimensão*kxp*.

O objetivo do algoritmo é a busca de uma configuração ótima de parâmetros para minimizar $J_{CM}(U_h, C)$, que é dado por:

$$J_{CM}(U_h, C) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij} d(\overrightarrow{C_i}, \overrightarrow{x_j})^2 \qquad (9)$$

Este índice de desempenho mede, para todos os elementos, a soma das distâncias ponderadas de cada elemento a cada um dos centros de *cluster* da partição. Quanto menor o valor de $J_{CM}(U_h, C)$, melhor a partição *fuzzy*.

Na fórmula, $d(\overrightarrow{C_i}, \overrightarrow{x_j})$ é a distância euclidiana entre o vetor de dados $\overrightarrow{x_j}$ e o protótipo do grupo $\overrightarrow{C_i}$, c é o número de grupos a ser determinado pelo algoritmo, *n* é o número de dados no conjunto de dados e U_h é chamada “matriz de partição”, de dimensões *cxn*. Esta matriz é definida por:

$$U_h = \begin{bmatrix} u_{1,1} & \cdots & u_{1,n} \\ \vdots & \ddots & \vdots \\ u_{k,i} & \cdots & u_{k,n} \end{bmatrix}$$

Nesta matriz de partição, cada elementou_{ij} está entre 0 e 1 eindica a associação de um dado a um grupo. Um dado $\overrightarrow{x_j}$está associado ao grupo representado pelo protótipo $\overrightarrow{C_i}$ se u_{ij} para este *cluster* for maior que para os outros.

Com o processo de minimização, os dados são associados aos grupos de forma que, quanto menores forem as distâncias entre o dado $\vec{x}$ e o vetor protótipo $\vec{C}$ associado a ele,

menor é o valor da equação (10). Este processo deve obedecer à condição abaixo que garante que a soma das pertinências de um dado u_{ij} a todos os grupos de *C* seja igual a 1, onde cada coluna da matriz de partição deve possuir o valor 1 em uma e somente uma célula:

$$\sum_{t=1}^{k} u_{ij} = 1, \forall_j \in 1, \dots, n. \quad (10)$$

Adicionalmente, uma segunda restrição ao processo de otimização de J_{CM}visa garantir que todos os grupos possuam, no mínimo, um dado associado. Esta restrição é dada pela equação (11) de forma que cada linha da matriz de partição deve possuir o valor 1 em pelo menos uma célula.

$$\sum_{j=1}^{n} u_{ij} \geq 1, \forall_i \in 1, \dots, c. \quad (11)$$

As equações (12) e (13) são implementadas no processo de minimização de J_{CM}, onde a atualização de U_hé dada por:

$$u_{ij}^{t+1} = \begin{cases} \text{Maior valor de pertinência, se i} = \operatorname{argmin}_{t=1}^{k} d\left(\overrightarrow{C_i}, \overrightarrow{x_j}\right) \\ \text{Menor valor de pertinência, caso contrário.} \end{cases} \quad (12)$$

Em (12) *t* é o contador de iterações do processo de otimização e u_{ij}^{t+1} é o valor da pertinência do dado *j* ao grupo *i* na iteração *t*+1, faz com que cada dado seja associado ao grupo cujo protótipo que possui a menor distância dentro todos os protótipos.

Já, em (13) a atualização de *C* estabelece novos vetores protótipos para grupos de acordo com a média de todos os vetores de dados associados a eles.

$$\overrightarrow{C_i}^{t+1} = \frac{\sum_{j=1}^{n} u_{ij} \overrightarrow{x_j}}{\sum_{j=1}^{n} u_{ij}} \quad (13)$$

O resultado deste algoritmo é dependente da inicialização do parâmetro *c* e do conjunto de vetores protótipos *C* e a vantagem relacionada à ambiguidade mantém mais informação em relação aos dados (LU et al, 2013).

O algoritmo pode ser descrito simplificadamente conforme abaixo:

1. Início: Determinar a quantidade de partições k e um valor pequeno e positivo para um erro máximo, ε, permitido no processo;
2. Inicialize o conjunto de protótipos C aleatoriamente, escolhendo c dados do conjunto ou inicialize tais vetores escolhendo aleatoriamente c dados do conjunto de dados;
3. Inicialize o contador de iterações t como $t = 0$;
4. Calcular U_h segundo equação:

$$u_{ij}^{t+1} = \begin{cases} \text{Maior valor de pertinência, se } i = \text{argmin}_{t=1}^{k} d(\overrightarrow{C_i}, \overrightarrow{x_j}) \\ \text{Menor valor de pertinência, caso contrário.} \end{cases}$$

5. Calcular C segundo equação:

$$\overrightarrow{C_i}^{t+1} = \frac{\sum_{j=1}^{n} u_{ij}\, \overrightarrow{x_j}}{\sum_{j=1}^{n} u_{ij}}$$

6. Repetir os passos 4 e 5 enquanto $\|C^{(t)} - C^{(t-1)}\| < \varepsilon$

Figura9: Fuzzy C-means (PERES, 2012).

1.1.4K-medóides

Devido ao método k-means ter sensibilidade quando um *outlier* for inserido, na distribuição de dados, este método busca solucionar esta questão alterando o cálculo para o dado mais centralizado no *cluster*, medóide, ao invés de um centro médio(GANDHI, 2014).

O algoritmo*Partitioning Around Medoids* (PAM) é iniciado com a seleção de *k* dados aleatoriamente que são definidos como medóides representando *kclusters* e todos os dados restantes são agrupados de acordo com a menor distância das medóides. Após este processo, uma nova medóide é determinada de forma a representar mais precisamente o *cluster* e todo o processo é repetido (GANDHI, 2014).

O objetivo é que após analisar todas as medóides e dados que o algoritmo escolha o par que aperfeiçoa a qualidade global da clusterização e realize a troca (IBRAHIM, 2013) e o processo continua até nenhuma medóideser alterada (GANDHI, 2014).

Um exemplo simples para uma função objetivo pode ser dada pela equação (18):

$$E = \sum_{i=1}^{k} Di \qquad (18)$$

Na equação (18) Dié a soma das distâncias euclidianas entre cada membro de um cluster *i* e o medóidecorrespondente. Se os *clusters*são homogêneos e compactos, o que é desejável, os valores de Dipara cada cluster tendem a ser pequenos. A função

objetivo, neste caso, deve ser minimizada, para produzir um bom resultado dentro de um processo de otimização (RENNO;SOARES, 2000).

O algoritmo pode ser descrito simplificadamente conforme abaixo:

1. Início: Escolher o número de agrupamentos, *k* e a base de dados com n elementos.
2. Escolher, arbitrariamente, *k* elementos da base de dados como as medóides iniciais dos grupos;
3. Atribua cada elemento remanescente ao grupo com a medóide mais próximo;
4. Aleatoriamente, selecione um elemento que não esteja como medóide, *r*;
5. Calcule o custo total, *E*, de trocar a medóide *Oj* pelo elemento *r*;
6. Se *E*< 0 então troque Oj por *r* para formar o novo conjunto de k-medóides;
7. Repetir os passos 2 a 6 enquanto até que não haja mudança de objetos de um grupo para outro.

Figura10: Algoritmo k-medoides (GANDHI, 2014).

Para um grande conjunto de dados, o algoritmo PAM funciona ineficientemente, exemplificando esta afirmação, para cada medóide, serão investigados $(n-k)$ possibilidades de troca, no caso de um conjunto de dados de 1000 objetos e definindo 10 *clusters*, seriam avaliadas 9.900 trocas em cada iteração do algoritmo.

Devido à questão acima, Kaufman e Rousseeuw também propuseram o algoritmo *Clustering for Large Applications* (CLARA) que utiliza uma amostra contendo m dos k objetos e realiza o algoritmo PAM sobre ela, de forma a determinar os k-medóides que minimiza e utilizando apenas m objetos. Então, definidos os medóides, os $(k-m)$ objetos restantes são agrupados, em função da menor dissimilaridade em relação ao conjunto de medóides. Se a amostra é bem realizada, e consequentemente representativa, os medóides determinados sobre a amostra tendem a se aproximar no espaço de atributos dos medóides que seriam obtidos considerando todo o conjunto de k objetos (RENNO;SOARES, 2000).

O algoritmo CLARA é bem mais rápido que o PAM, pois examina apenas um subconjunto de k-medóides possíveis, mas a qualidade do dependente da qualidade da amostra e sua aplicabilidade cresce junto com o número de objetos a serem analisados (RENNO;SOARES, 2000).

1.2 Métodos Hierárquicos

Desenvolvidos inicialmente no campo da biologia, as técnicas hierárquicas ganharam popularidade devido a fatores como versatilidade, simplicidade e variedade de métodos disponíveis, como também ao aspecto intuitivo de que graus relativos de semelhança entre os objetos poderiam ser visualizados em uma representação hierárquica, como, por exemplo, em uma árvore (COSTA, 1999).

A base da clusterização hierárquica é que a solução produz uma sucessão de partições, cada qual correspondendo a um diferente número de agrupamentos, não requer que seja definido um número a priori de agrupamentos. Através da análise do dendrograma pode-se inferir o número adequado de agrupamentos(EVERITT, 1993).

Os métodos hierárquicos requerem uma matriz contendo as métricas de distância entre os agrupamentos em cada estágio do algoritmo; esta matriz é conhecida como matriz de dissimilaridades entre agrupamentos. Como ilustração, a matriz de dissimilaridades em um estágio do algoritmo comtrês agrupamentos (C_1, C_2 e C_3) pode ser exemplificada na Tabela1:

Tabela1: Exemplo de matriz de dissimilaridade (DO AUTOR, 2016).

	C_1	C_2	C_3
C_1	0	0.1	0.5
C_2	0.1	0	0.7
C_3	0.5	0.7	0

Pela Tabela1 é possível observar que C_1 e C_2 tem menordissimilaridade que C_2 e C_3.

Os métodos hierárquicos são subdivididos conforme abaixo exemplificados na Figura 11:

- Métodos aglomerativosconsideradosno próximo item, que consideram no início que os *n* objetos são *n* subgrupos e por meio de uniões sucessivas, uma

de cada vez, chega-se a um único agrupamento contendo todos os objetos no final do processo (COSTA, 1999);

- Métodos Divisivos são menos comuns entre os métodos hierárquicos devido a sua ineficiência e exigência computacional maior que os métodos aglomerativos (COSTA, 1999).

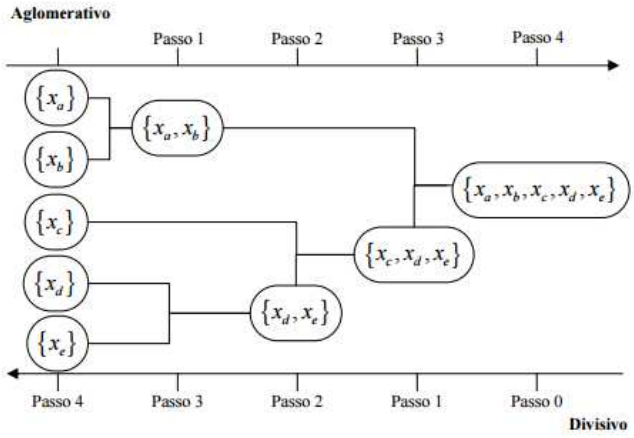


Figura11: Agrupamento hierárquico aglomerativo e divisivo (DUARTE, 2008)

1.2.1 *Algoritmo Hierárquico Aglomerativo*

As técnicasaglomerativas operam geralmente sobre uma matriz de similaridades ou dissimilaridades, Dij $= (\mathrm{i}, \mathrm{j} = 1, 2, \ldots, n)$produzindo uma sequência de partições dos dados, $P^n, P^{n-l}, \ldots, P^l$. A primeira partição consiste de n agrupamentos contendo um elemento apenas e a última, consiste de um agrupamento contendo todos os $n$objetos (EVERITT, 1993).

Em (WILLIAMS, LANCE, 1967) foi desenvolvida uma fórmula de recorrência generalizada que permite a determinação das novas distâncias entre o agrupamento formado (C_k) e todos os $l$agrupamentos, D_{kl}, onde $C_k = C_i \cup C_j \; e \; C_l$é outro agrupamento qualquer. A fórmula (19) apresenta a vantagem de necessitar, em cada estágio da análise, apenas das informações da matriz de similaridades (ou dissimilaridades) do estágio anterior e funciona para muitos métodos aglomerativos, onde os parâmetros $\alpha_i , \alpha_j , \beta \; e \; \gamma$ definem cada um dos métodos:

$$D_{kl} = \alpha_i . D_{ki} + \alpha_j . D_{jk} + \beta . D_{ij} + \gamma . \left| D_{ik} - D_{jk} \right| \quad (19)$$

Os parâmetros $\alpha_i, \alpha_j, \beta \, e \, \gamma$ definem as principais técnicas aglomerativas e são apresentadas a seguir, adicionalmente a Tabela2:

- Ligação simples (individual): também denominado de método dos vizinhos mais próximos, é caracterizada por considerar a dissimilaridade entre dois agrupamentos C_i e C_k como a menor dissimilaridade dentro de cada par de objetos (COSTA, 1999). É mais indicado quando os clusters não tem a forma hiperesférica e nem hiperelíptica (EVERITT, 1993);
- Ligação completa:o procedimento de ligação completa é semelhante ao da ligação simples ou individual, exceto em que o critério de agrupamento se baseia na distância máxima entre indivíduos em cada agregado e representa a menor esfera (diâmetro mínimo) que pode incluir todos os objetos em ambos os agrupamentos. Todos os objetos em um agrupamento são conectados um com o outro a alguma distância máxima ou similaridade mínima (MACIEL, 2008);
- Média das ligações: o agrupamento é caracterizado pela média de todas as dissimilaridades entre os seus membros, sendo menos sensível a *outlier* que o item anterior e gerando agrupamentos mais homogêneos do que o método de ligação simples(COSTA, 1999);
- Método de centroide: a distância entre dois agrupamentos é a distância entre seus centroides, geralmente euclidiana. Neste método, toda vez que os indivíduos são reunidos, um novo centroideé computado, também existe uma mudança no centroide do *cluster* toda vez que um novo indivíduo ou um grupo de indivíduos é acrescentado a um *cluster* existente (MACIEL, 2008);
- Método Ward: propõe que os agrupamentos sejam formados objetivando otimizar um critério, a fusão entre agrupamentos em geral prioriza a união dos grupos que minimiza a variância ou a soma dos quadrados dos desvios (ou distâncias) em relação à média dentro dos agrupamentos. Este método tem uma tendência a formas agrupamentos com o mesmo número de objetos (COSTA, 1999).

No caso do método Ward a proximidade entre dois grupos é definida como o aumento no erro quadrático que resulta da junção de dois grupos e o cálculo do aumento no erro quadrático na junção de dois grupos é calculado como disposto na equação (20):

$$\Delta(A+B) = \sum_{i \in AUB} \|\vec{x}_i - \vec{m}_{AUB}\|^2 - \sum_{i \in A} \|\vec{x}_i - \vec{m}_A\|^2 - \sum_{i \in B} \|\vec{x}_i - \vec{m}_B\|^2 = \frac{n_A n_B}{n_A + n_B} \|\vec{m}_A - \vec{m}_B\|^2 \quad (20)$$

Tabela2: Comparação entre os métodos hierárquicos (DO AUTOR, 2016).

Método	α_i	β	γ
Ligação simples	0.5	0	-0.5
Ligação completa	0.5	0	0.5
Média das ligações	$\frac{\|i\|}{\|i\| + \|j\|}$	0	0
Centroide	$\frac{\|i\|}{\|i\| + \|j\|}$	$-\frac{\|i\|.\|j\|}{(\|i\| + \|j\|)^2}$	0
Ward	$\frac{\|i\| + \|k\|}{\|i\| + \|j\| + \|k\|}$	$-\frac{\|k\|}{\|i\| + \|j\| + \|k\|}$	0

O dendrograma é a representação gráfica, em forma de árvore, da estrutura dos agrupamentos, onde as folhas representam os *clusters*formados por apenas um elemento.À medida que a altura da árvore cresce, os dados juntam-se para formar *clusters* maiores até que todos façam parte de um mesmo *cluster* (EVERITT, 1993).

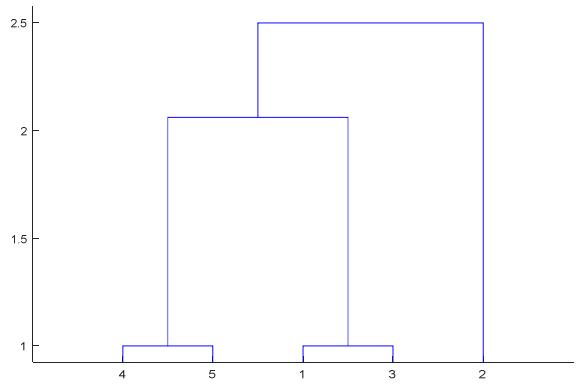


Figura12: Dendrograma (DO AUTOR).

O algoritmo pode ser descrito simplificadamente conforme abaixo:

1. Início: Conjunto de dados *X* com *n* objetos;
2. Calcular a matriz de similaridades *n*x*n* a partir do conjunto de *X*;
3. Enquanto não se verificar a condição de término, repetir de 4 a 7:
4. Atribuir cada objeto *X* a um grupo (*n* grupos);
5. Calcular a métrica de ligação (medida de similaridade) entre todos os pares de grupos de forma a encontrar o par de grupos mais semelhantes;
6. Fundir o par de grupos mais semelhantes com o objetivo de formar novo grupo;
7. Atualizar a matriz de similaridades: suprir as linhas e as colunas correspondentes aos grupos fundidos e adicionar uma linha e uma coluna correspondente às similaridades entre o novo grupo e os grupos já existentes anteriormente;
8. Construir dendrograma com base nas fusões efetuadas.

Figura13: Algoritmo hierárquico aglomerativo (DUARTE, 2008).

Uma das principais desvantagens de métodos hierárquicos é que a fusão de agrupamentos em um determinado estágio não poderá ser corrigida em estágios posteriores (MACIEL, 2008).

1.3Métodos Baseados na Densidade

Existem situações em que a distribuição de dadosapresenta diferenças de densidade e os métodos acima não apresentam resultados satisfatórios, sendo preferencial a utilização de métodos baseados na densidade, pois atingem melhores resultados (CAMILO, SILVA, 2009).

1.3.1 *Density Based Spatial Clustering of Applications with Noise* (DBSCAN)

Os algoritmos de agrupamento baseados em densidade têm como objetivo a determinação de regiões de alta densidade de objetos separados por regiões de baixa densidade (SEMAAN et al, 2012). O algoritmo DBSCAN pode identificar *clusters* em grandes conjuntos de dados, verificando a densidade local de elementos presentes nos mesmos. Além disso, o usuário recebe uma sugestão sobre qual o valor do parâmetro que seria adequado, requisitando a necessidade de um conhecimento mínimo do domínio, é capaz de descobrir grupos de formatos arbitrários em bases de dados espaciais e com ruídos (FERREIRA et al, 2015). Ele também pode determinar qual

informação deve ser classificada como ruído ou *outlier*. Apesar disso, o seu processamento é rápido e escalar com o tamanho da base de dados (ESTER, 1996).

O algoritmo utiliza-se de um conceito de densidade tradicional baseada em centro, ou seja, a densidade de um objeto x_ié a quantidade de objetos em um determinado raio de alcance de x_i, incluindo o próprio objeto (SEMAAN et al, 2012).

A abordagem da densidade baseada em centro realiza a classificação dos objetos em (SEMAAN et al, 2012) da seguinte forma:

- Interiores (ou centrais)que são objetos que pertencem ao interior de um grupo baseado em densidade;
- Limítrofes, que não é um objeto central, mas é alcançável por ao menos um objeto central, ou seja, está dentro do raio de vizinhança de algum objeto central;
- Ruídos que são os demais objetos que não são centrais e nem estão na vizinhança de um objeto central.

A Figura 14 ilustra a classificação dos objetos no algoritmo DBSCAN.

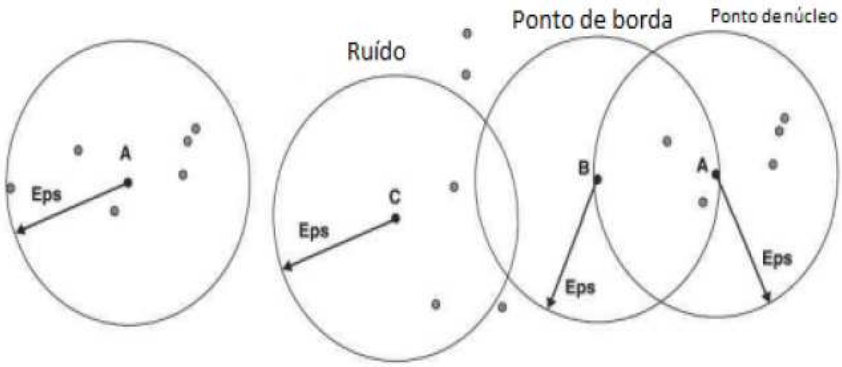


Figura14: Classificação dos objetos (adaptado de TAN et al, 2005)

É necessário especificar o parâmetro*Eps* que é valor que descreve a medida de proximidade, isto é, quantos pontos vizinhos próximos um par de pontos necessita ter em comum para serem considerados próximos, ou seja, raio máximo da vizinhança.Já o parâmetro *MinPts*é valor relativo da densidade mínima, ou seja, número de vizinhos próximos que um ponto precisa ter para ser considerado “ponto de núcleo”. Assim, as regiões com alta densidade são definidas como *clusters* separados por regiões sem ou com pouca densidade (NAGPAL, 2013).

Idealmente tem-se que saber os parâmetros *Eps* e *MinPts* adequados, pois pode-se recuperar todos os pontos que são alcançáveis por densidade a partir de um dado ponto usando os parâmetros corretos, tornando o algoritmo DBSCAN muito sensível aos parâmetros definidos pelo usuário (TAN et al, 2005)

O algoritmo pode ser descrito simplificadamente conforme abaixo:

1. Início: Definir os valores *Eps* e *MinPts*;
2. Arbitrariamente selecionar um ponto central;
3. Recuperar todos os pontos alcançáveis do centro, respeitando os parâmetros:*Eps* e*MinPts*.
4. Se o ponto é de interior, um *cluster* é formado;
5. Se o ponto é limítrofe ou não alcançável pelo ponto central deve ser analisado o próximo ponto da base de dados;
6. Continuar o processo até todos os pontos serem processados.

Figura15: Algoritmo DBSCAN (TAN et al, 2005).

GUHA (1998) afirma que o algoritmo DBSCAN também sofre do problema de falta de robustez que atinge os métodos hierárquicos de clusterização que utilizam todos os objetos, e como o método não desempenha qualquer etapa de pré-clusterização e trabalha diretamente sobre a base de dados inteira, ele pode ter alto custo computacional no caso de bases de dados grandes. O autor ainda diz que métodos baseados em densidade usando amostragem aleatória para reduzir o tamanho da entrada podem não ser possíveis.A razão para isto é que o tamanho das amostras deve ser grande, para que não existam variações substanciais na densidade dos objetos dentro de cada *cluster* na amostra utilizada em relação a população total (TAN et al, 2005).

1.4Comparação entre os métodos

Na Tabela3 é realizada a comparação das principais características dos métodos de clusterização tradicionais.

É possível tratar dados categóricos utilizando a técnica “1 para *N*”.Por exemplo, um atributo possui três opções (N=3) de valor categórico (“professor”,”técnico” e “aluno”), o mesmo será traduzido em três atributos e será codificado da seguinte forma:

“professor”: 100, “técnico: 010” e “aluno”: 001. Esta técnica aumenta a dimensão dos dados.

Tabela3: Comparação entre os métodos (DO AUTOR, 2016).

	K-means	FCM	K-medoids (PAM)	Aglomerativo	DBSCAN	SOM
Hiperparâmetros de entrada	Quantidade de *clusters*	Quantidade de *clusters*	Quantidade de *clusters*	Quantidade de *clusters*.	Raio e quantidade mínima de objetos na vizinhança	Vetores de pesos iniciais e forma do Mapa.
Outliers	Sensível	Sensível	Não sensível	Dependente da técnica escolhida	Não sensível	Sensível
Formato do *cluster*	Hiperesférico	Hiperesférico	Hiperesférico	Hiperesférico e arbitrário	Arbitrário	Esférico
Função Objetivo	E= $\sum \|\|xi - mi\|\|^2$	$J_{CM}(U_h, C) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij} d(\overrightarrow{C_i}, \overrightarrow{x_j})^2$	E = $\sum_{i=1}^{k} Di$	Não há	Não há	Não há.
Complexidade	O(Kn)	Variável	O(K(n-K)²)	O(n² log(n))	O(nlogn)	Variável

1.5Indicações de validade

Nas avaliações experimentais dos algoritmos de agrupamento são usados frequentemente conjuntos de dados com apenas duas ou três dimensões, para que se torne possível verificar visualmente a validade dos resultados obtidos, porém, no caso de conjuntos de dados multidimensionais (mais do que três dimensões), a visualização eficaz do conjunto de dados usando ferramentas de visualização é mais difícil para identificar grupos em espaços de elevada dimensionalidade. O escalonamento multidimensional (*MultiDimensional Scaling* – MDS) é constituído por um conjunto de técnicas estatísticas usadas frequentemente na área da visualização de informação, de forma a explorar, similaridades e dissimilaridades nos dados. A partir de uma matriz de

similaridades entre os objetos de dados, as técnicas de escalonamento multidimensional atribuem uma localização de cada objeto num espaço dimensional menor, apropriado para a visualização (DUARTE, 2008).

O objetivo das medidas de validade é aferir o quanto um determinado agrupamento de dados corresponde à estrutura natural dos dados. Nesta análise existem as seguintes propriedades principais que são seguidas na maior parte dos índices: a dispersão, que avalia o quanto os objetos de um grupo estão próximos e outros objetivos do mesmo grupo, como medida usual tem-se a variância e a separação que avalia o quanto os grupos estão afastados entre si (DUARTE, 2008).

1.5.1 Coeficiente de Silhouette

O valor da silhouette para cada ponto é a medida do quão similar este ponto é em relação os outros pontos do mesmo cluster. Esta medida é definida de acordo com as equações (20) e (21)(DUARTE, 2008):

$$s(i) = \frac{b(i)-w(i)}{\max\{b(i),w(i)\}} \quad (21)$$

$$b(i) = min_k\{B(i,k)\} \quad (22)$$

Na equação (21) $w(i)$ é a média da distância do ponto i-ésimo para os outros pontos do mesmo *cluster* e $B(i,k)$ é a média da distância desde o ponto i-ésimo para os pontos de um outro *cluster*(DUARTE, 2008).

Esta medida varia desde +1, indicando pontos que estão em uma distância muito próxima dos clusters vizinhos, “0”, que indica pontos que não possuem *clusters* definidos e -1 que indica pontos que provavelmente estão dispostos no *cluster* incorreto (DUARTE, 2008).

Por definição, este método não pode ser utilizado quando é definido somente um *cluster* para o conjunto de dados, já que $w(i) = 0$ para todo i e, por sua vez, $s(i) = 1$ para todo$i$ (DUARTE, 2008).

1.5.2 Índice de Davies e Bouldin

Considere-se $s_{C_iC_j}$ uma medida de similaridade entre dois grupos C_i e C_j, baseada numa medida de dispersão de um grupo $C_i\left(disp_{C_i}\right)$ e numa medida de dissimilaridade entre dois grupos C_i e C_j $\left(d_{C_iC_j}\right)$: $s_{C_iC_j} = \frac{disp_{C_i} + disp_{C_j}}{d_{C_iC_j}}$

O índice de Davies e Bouldin (*DB*) define-se como:

$$DB = \frac{1}{K}\sum_{i=1}^{K} s_{C_i} \quad (23)$$

Na equação (23), $s_{C_i} = max_{i,j=1,\ldots,K,\ i\neq j} s_{C_iC_j}$, valores baixos de *DB* indicam grupos dissimilares, já que *DB* representa a similaridade média entre cada um dos grupos do conjunto de dados com o grupo mais semelhante correspondente (DUARTE, 2008).

1.5.3 Índice de Calinski e Harabasz

O índice Calinski e Harabasz escolhe como melhor esquema de agrupamento aquele em que se obtém o maior valor do índice, o cálculo é dado por(DUARTE, 2008):

$$CH\ (K) = \frac{B(K)/(K-1)}{W(K)/(n-K)} \quad (24)$$

Na equação (24), $W(k) = \sum_{j=1}^{K}\sum_{i=1}^{n_j}\left\|x_i - c_j\right\|^2$ é a dispersão interna dos grupos e $B(k) = \sum_{j=1}^{K} n_j \left\|c_j - c\right\|^2$ é a dispersão entre os grupos e c o centro do conjunto de dados(DUARTE, 2008).

1.5.4 Índice de Dunn

Na equação (25), d_{C_i,C_j} é uma função de dissimilaridade entre dois grupos C_i e C_j definida como $d_{C_i,C_j} = min_{x_a\epsilon C_i, x_b\epsilon C_j} d_{x_a,x_b}$. O diâmetro de um grupo é definido por $diam(C_l) = max_{x_a,x_b\in C_l} d_{x_a,x_b}$ e pode ser considerado como uma medida de dispersão(DUARTE, 2008).

$$Dunn = min_{i=1,\dots,K}\left\{min_{j=i+1,\dots,K}\left(\frac{d_{C_i,C_j}}{max_{l=1,\dots,K}diam(C_l)}\right)\right\} \quad (25)$$

Valores elevados deste índice de validação indicam a presença de grupos compactos e bem separados no conjunto de dados (DUARTE, 2008).

Resumindo o entendimento em relação aos índices de validação avaliados, todos os índices determinam que quanto maior o mesmo, maior a qualidade referente à clusterização realizada.

2 ÁRVORES DE PADRÕES FUZZY

Ideias, conceitos e ferramentas da lógica *fuzzy* podem ser utilizados de variadas formas para aperfeiçoar os métodos de aprendizado de máquina. Neste caso, árvore de padrões *fuzzy* (APF) é um modelo hierárquico, com uma estrutura similar a uma árvore, em que os nós são os operadores lógicos *fuzzy* e operadores matemáticos, no Capítulo 4 mais detalhes em relação aos operadores que serão considerados no método proposto e as folhas são compostas por termos *fuzzy* associadas ao atributo de entrada. Um nó assume os valores de seus antecedentes como entrada e realiza uma combinação usando o operador escolhido e envia a saída para o seu sucessor, ou seja, a leitura da árvore ocorrerá no sentido de baixo para cima. A vantagem deste método se refere à interpretação do resultado, pois cada árvore descreve quais atributos tem o maior peso na definição do *cluster* formado. Comumente cada árvore pode ser considerada como uma descrição lógica de um grupo (SENGE; HÜLLERMEIER, 2011).

Um conjunto *fuzzyA* definido no universo de discurso *X* é caracterizado por uma função de pertinência μ_A, a qual mapeia os elementos de *X* para o intervalo [0,1].

Na lógica nebulosa, os termos *fuzzy* são os valores expressos linguisticamente, (por exemplo, alto, muito alto, não alto, etc.), onde cada termo linguístico é interpretado como um subconjunto *fuzzy* do intervalo unitário.

Os modificadores são termos ou operações que modificam a forma dos conjuntos *fuzzy* (ou seja, a intensidade dos valores *fuzzy*), podendo-se citar, por exemplo, os advérbios muito, pouco, extremamente, quase, mais ou menos, entre outros. Estes podem ser classificados em aumentadores, quando aumentam a área de pertinência de um conjunto *fuzzy*, ou, analogamente, diminuidores, quando diminuem a área de pertinência de um conjunto *fuzzy*.

Um classificador baseado em árvore de padrões é construídocriando uma árvore para cada*cluster*. São inseridos os valores dos atributos que sedeseja classificar nas entradas das árvores de cada *cluster* e a predição do *cluster* que esseconjunto de dados pertence é feita escolhendo a árvore que tem o maior valor de saída.

Árvores de padrões são interessantes especialmente para a interpretação dos resultados. A Figura17 mostra um exemplo de aprendizado preferencial (HÜLLERMEIER, 2008), onde cada *cluster*corresponde a uma possível escolha. Neste

caso a árvore pode ser vista como a mediçãodo grau de utilidade dessa alternativa em função da entrada.

Neste exemplo, é possível verificar que o atributo “idade” é particionado em jovem, meia idade e idoso conforme Figura 16.

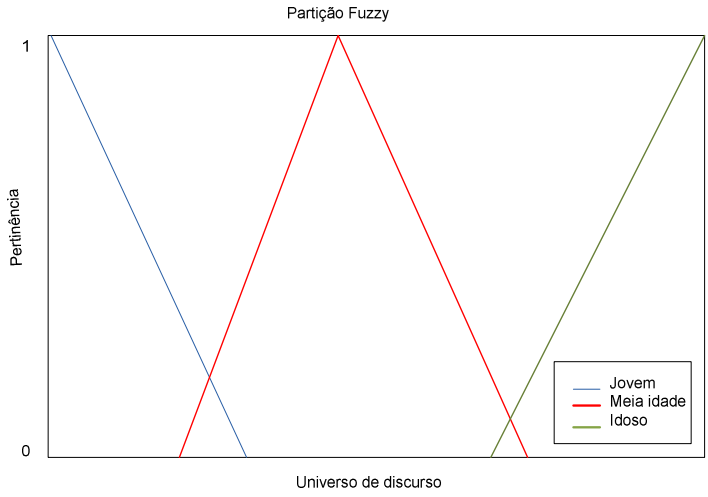


Figura 16: Particionamento do atributo idade (DO AUTOR, 2016)

Assim, assumindo que a árvore de padrão corresponde à escolha de uma alternativa específica, pode ser concluído a partir da Figura 17 que o produto A é útil para uma pessoa jovem e com alta renda ou para meia idade com alto grau de educação, com a mesma análise para os produtos B e C, é possível determinar quais grupos escolheriam quais tipos de produtos.

Assim, é possível observar que uma árvore pode informar que dois grupos distintos de consumidores podem consumir um mesmo produto, como ocorre com o produto A, trazendo mais informações sobre a característica de cada grupo de consumidor do que o retornado por uma clusterização tradicional, em que é necessário que o responsável pela análise seja mais intuitivo para observar as características apresentadas em cada grupo.

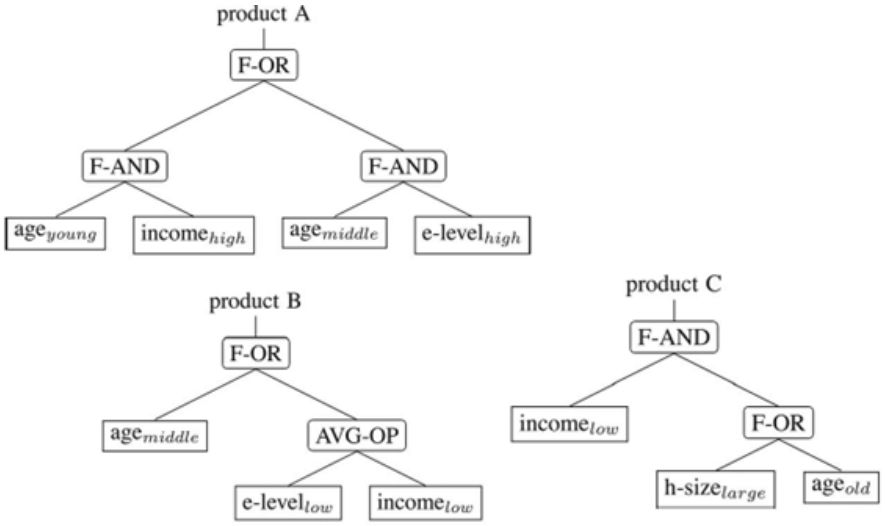


Figura 17: Exemplo de marketing de árvore de padrão (SENGE; HÜLLERMEIER, 2011)

Como descrito na Figura 17, nós internos representam agregação de valores de dois atributos, para esta operação podem ser utilizados três famílias de operadores: t-norms, t-conorms. Estes operadores são mostrados nas Tabelas 4 e 5, *a* e *b* denotam o grau de pertinência, a serem agregados conforme Figura 18 e dois operadores de média: *weighted average* (WA) e *ordered weighted average* (OWA) (SENGE; HÜLLERMEIER, 2011).

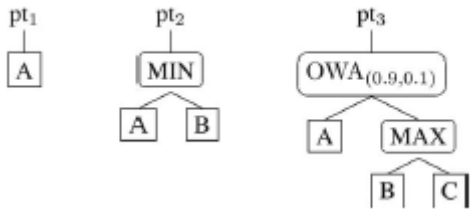


Figura 18: Exemplo de árvore de padrão (SENGE; HÜLLERMEIER, 2011)

O operador OWA, de média ponderada ordenada, é uma combinação de k números $v_1, v_2, \dots, v_k$ e é definido pela equação 26.

$$OWA_w\,(v_1, v_2, \dots, v_k) \stackrel{\text{def}}{=} \sum_{i=1}^{k} w_i \,.\, v_{\tau(i)} \tag{26}$$

Na equação 16,τ é a permutação entre o conjunto de números de $\{1, 2, \dots, k\}$ tal que $v_{\tau(1)} \leq v_{\tau(2)} \leq \dots \leq v_{\tau(k)}$, ou seja, de forma decrescente e $w = (w_1, w_2, \dots, w_k)$ é o vetor de pesos que satisfaz a condição $w_i \geq 0$ para $i = 1, 2, \dots, k$, com a condição que o somatório dos pesos de 1 a k deve ser igual a 1. O operador WA é calculado de forma similar, mas é operador de média ponderada.

Tabela 4: Operador Fuzzy, t-norm (SENGE; HÜLLERMEIER, 2011)

Nome do operador	Definição
Mínimo	$\min(a, b)$
Algébrico	ab
Lukasiewicz	$\max(a + b - 1,0)$
Einstein	$\frac{ab}{2 - (a + b - ab)}$

Tabela 5: Operador Fuzzy, t-conorm (SENGE; HÜLLERMEIER, 2011)

Nome do operador	Definição
Mínimo	$\max(a, b)$
Algébrico	$a + b - ab$
Lukasiewicz	$\min(a + b, 1)$
Einstein	$\frac{a + b}{1 + ab}$

O primeiro método de aprendizado proposto por Huang, Gedeon e Nikravesh (HUANG; GEDEON; NIKRAVESH, 2008), é chamado de *Bottom-up Induction*. Nele, a indução das árvores procura criar uma representação “lógica” para cada grupo de uma forma iterativa.

O processo ocorre do fundo para o topo, em cada iteração do processo,combinam-se as duas melhores árvores candidatas para criar uma nova árvore.

No modelo proposto, o método de aprendizado do APF foi substituído e em seu lugar foi utilizada a Programação Genética Cartesiana (PGC). A PGC é um método de busca global capaz de explorar espaços de busca bastante grandes de forma eficiente e a representação dos programas na forma de grafos pode ser facilmente utilizada para representar APFs.

3 PROGRAMAÇÃO GENÉTICA

Os Algoritmos Evolucionários são inspirados no princípio darwiniano da evolução das espécies e na genética. Do mesmo modo que a evolução natural produz indivíduos mais aptos a sobreviver em um meio-ambiente sujeito a constantes mudanças, os Algoritmos Evolucionários podem ser vistos como procedimentos de otimização que melhoram o desempenho de uma população de soluções em potencial em relação a um problema específico.

Os principais Algoritmos Evolucionários são: os Algoritmos Genéticos, aProgramação Genética, as Estratégias Evolutivas e a Programação Evolutiva(HRUSCHKA, 2009).

A Figura19 descreve o desenvolvimento básico de um algoritmo evolucionário.

```
Início
      1. t=0;
      2. Inicializa P(t); Primeira geração.
      3. Inicializa P(t); População inicial aleatório.
      4. Avalia P(t);     Calcula a função de aptidão para cada indivíduo.
      5. Enquanto o critério de parada não é atendido, realizar os passos
         abaixo:
      6. t=t+1; Próxima geração
      7. Seleciona P(t) de P(t-1)
      8. Altera P(t);Crossover e mutação.
      9. Avalia P(t);Calcula a função de aptidão para cada indivíduo.
      10. Retorna ao passo 5.
Fim
```

Figura19: Desenvolvimento de um algoritmo evolucionário

Na Programação Evolutiva cada indivíduo da população é representado por uma máquina de estados finitos e a reprodução é feita apenas por operadores de mutação, sendo que todos os indivíduos da população atual geram novos descendentes.O elitismo mais utilizado garante a sobrevivência apenas dos k-melhores indivíduos,$k<N$, onde N é o tamanho da população. O elitismo total, no entanto, pode diminuir significativamentea diversidade de indivíduos, podendo estagnar em ótimos locais e/ou aumentar o tempo de convergência do algoritmo (FOGEL, 1962).

As Estratégias Evolutivas foram desenvolvidas com o objetivo de solucionar problemas de otimização de parâmetros.Neste algoritmo um indivíduo da população gera um único descendente e ambos competem pela sobrevivência, o modelo original possui uma convergência lenta (GABRIEL, DELBEM, 2008) e (DE JONG, 2006).

Os Algoritmos Genéticos têm como seu principal diferencial a utilização do operador *crossover*, além do operador de mutação preservando soluções candidatas e provocando a troca de informação entre as soluções exploradas (GABRIEL, DELBEM, 2008).

Já a Programação Genética (PG) é uma técnica que permite que computadores resolvam problemas sem que precisem ser explicitamente programados para tal (KOZA,1992). Na PG, os indivíduos são codificados na forma de árvores, onde cada folha contém constantes, variáveis ou parâmetros para a execução de procedimentos e funções. Os nós internos contêm operações primárias (GABRIEL, DELBEM, 2008).

Na etapa de avaliação ocorre por meio da execução do programa representado pela árvore do indivíduo. Se este resolver o problema proposto ou se aproximar da resposta correta terá um valor de aptidão elevado; caso contrário, sua aptidão será baixa (GABRIEL, DELBEM, 2008).

A PG é útil nos seguintes cenários(SANTOS; DO AMARAL, 2014):

- A relação entre as variáveis é desconhecida ou há pouco conhecimento sobreesta relação e não se dispõe de métodos analíticos capazes de estabelecer esta relação de forma satisfatória;
- Situações em que os métodos matemáticos convencionais não podem criarsoluções analíticas. Em domínios nos quais as soluções analíticas não sejam possíveis, ouque estas soluções tenham um tempo de execução impraticável ou que precisem de umacaracterística indisponível nos dados, por exemplo, uma base de dados sem ruído.Nestes casos o uso da programação genética é adequado, pois garante uma boa soluçãoaproximada quando uma solução exata é impraticável;
- Problemas em que pequenos aumentos no desempenho são facilmente medidos ealtamente recompensados. Existem domínios que possuem soluções muito avançadas, sendodifícil melhorar as soluções já existentes, porém nestes domínios um pequeno aumentono desempenho pode ser muito

> valioso, a programação genética pode ser útil nodescobrimento de pequenas relações que podem criar estes pequenos e valiososaumentos no desempenho.

Porém, este tipo de programação não teve um bomdesempenho para grandes problemas. O maior fator para esta falha foi a ocorrência do*bloat* (SOULE, 1998).

O *bloat* é a tendência dos programas gerados com a PG ficarem muito maioressem que haja benefício para a aptidão, gerando árvores muito grandes, dificultando a interpretação das características do *cluster*, não apresentando qualquer efeito na avaliação do indivíduo. Análises realizadas sobre o *bloat* em PG indicam que representações detamanho variável tem maior incidência do mesmo (SOULE; HECKENDORN, 2002). O *bloat* pode causar o consumo total da memória disponível.

A Programação Genética Cartesiana (PGC) (MILLER, 2009) é uma forma de programação genética na qual os programas são representados por uma grade bidimensional de nós, ou seja, por grafos acíclicos direcionados.

Na representação de grafos, cada indivíduo g consiste em N genes $g_1, ..., g_N$ e cada gene $g_i$pode assumir valores de alelos j no intervalo $\{1,...,N\}$. Assim, um valor j é atribuído ao i-ésimo gene, que é interpretado como uma ligação entre os itens de dados i e j: na resolução de grupamento, os dois estarão no mesmo *cluster*. A decodificação desta representação requer a identificação de todos os componentes ligados. Todos os itens de dados pertencentes a um mesmo componente conectado são então atribuídos para um cluster (HANDL, 2007).

O benefício da utilização de grafos é o fato de que grafos são mais gerais, flexíveis e compactos e podem ser aplicados em diversos domínios (DHARWADKER; PIRZADA, 2011), neste tipo de representação há a reutilização implícita dos nós pertencentes ao grafo direcionado.

Dentre as vantagens da PGC está a característica de neutralidade que é responsável porminimizaro*bloat*, que é comum em outros métodos de programação genética (BANZHAF, 1994), (MILLER; SMITH, 2006),(MILLER, 2001). Neste caso, é preferível árvores mais compactas e de melhores aptidões, que por consequência exigem um menor esforço computacional.

As vantagens adicionais da PGC são: foco em conceitos e interpretação de problemas como um programa de computador e capacidade de encontrar dependência e

independência de variáveis e estabelecer relações entre as mesmas (YUVARAJU, 2013).

Na PGC, os programas são representados em uma sequência linear de números inteiros, que é denominada cromossomo ou genótipo conforme representado na Figura 20. Cada inteiro representa um gene, que pode ser de função, conexão ou de saída.

O programa gerado pela decodificação do genótipo é chamado de fenótipo.

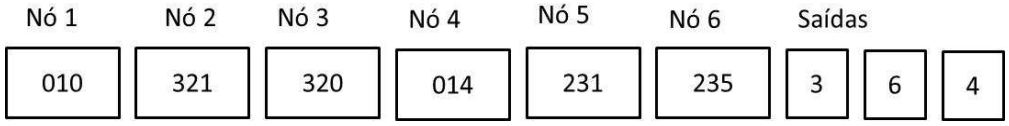


Figura20: Genótipo ou cromossomo (adaptado de MILLER, 2011)

O genótipo tem um comprimento fixo, no entanto, o tamanho do fenótipo pode ser desde zero nós até o número de nós definido anteriormente no genótipo.

Assim, esta técnica utiliza um mapeamento do tipo genótipo-fenótipo e não requer necessariamente que todos os nós devam estar ligados uns aos outros, resultando, assim, em um fenótipo com sua variação de tamanho limitado. Isto permite que alguns nós do genótipo fiquem inativos, não tendo qualquer influência sobre o fenótipo (PARIS, 2013).

A forma geral bidimensional da PGC, conforme Figura 21,é uma grade de nós cujas funções são escolhidas a partir de um conjunto de funções primitivas, cujo exemplo pode ser visto naTabela 4.Dois parâmetros *c* e *r*, respectivamente, definem o número de colunas e linhas da grade bidimensional, estes dois parâmetros são escolhidos pelo usuário. A partir do número de linhas e colunas, é possível determinar o número máximo de nós permitidos, $L_n = n_c n_r$(PARIS, 2013).

Tabela6: Funções (DO AUTOR, 2016).

Função	Referência da função
Soma	0
Subtração	1
Multiplicação	2
Divisão	3

A entrada de cada nó pode ser uma entrada do sistema ou a saída de um nó de uma coluna anterior(MILLER; HARDING, 2009; MILLER; THOMSON, 2000; MILLER, 2011). O *levelback*(*I*) determina a conectividade entre os nós, isto é, restringe as colunas que um nó pode obter os seus dados de entrada, se o valor for igual a um, por exemplo, um nó só pode obter suas entradas a partir de um nó pertencente à coluna imediatamente a sua esquerda ou da entrada primária, adicionalmente, se for igual a *c*, o nó poderá receber suas entradas de todas as colunas a sua esquerda e da entrada primária. Assim, a variação deste parâmetro pode resultar em várias topologias para o grafo conforme Figura 21 (PARIS, 2013).

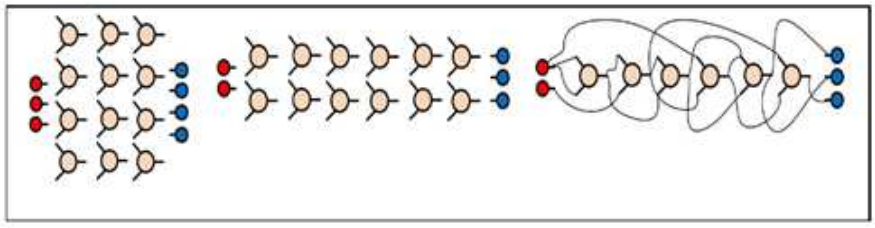


Figura21: Exemplo de topologias possíveis (MILLER, 2011)

A quantidade de entradas dos nós é chamada de aridade, e é determinada de acordo com a função que necessita do maior número de entradas entre as funções do conjunto. Tanto o *levelback*como a aridade, são parâmetros a serem definidos pelo programador (MILLER; HARDING, 2009; MILLER; THOMSON, 2000; MILLER, 2011).

Todos os dados de entrada e saída dos nós são identificados consecutivamente, iniciando em zero, o que garante um endereço único, que especifica de onde os dados de entrada ou o valor de saída do nó podem ser acessados (entradas (*n*), saídas (*m*)) (PARIS, 2013).

Um nó pode ter apenas as suas entradas ligadas a ambos os dados de entrada ou a saída de um nó em uma coluna anterior. Em geral, pode haver certo número de genes de saída (O_i) que especifica de onde as saídas do programa são retiradas, todos os nós de uma função F_i são endereços representados por números inteiros, todos os genes de conexão C_{ij}são endereços de dados absolutos tendo valores inteiros de zero até o endereço do nó na parte inferior da coluna anterior(PARIS, 2013).

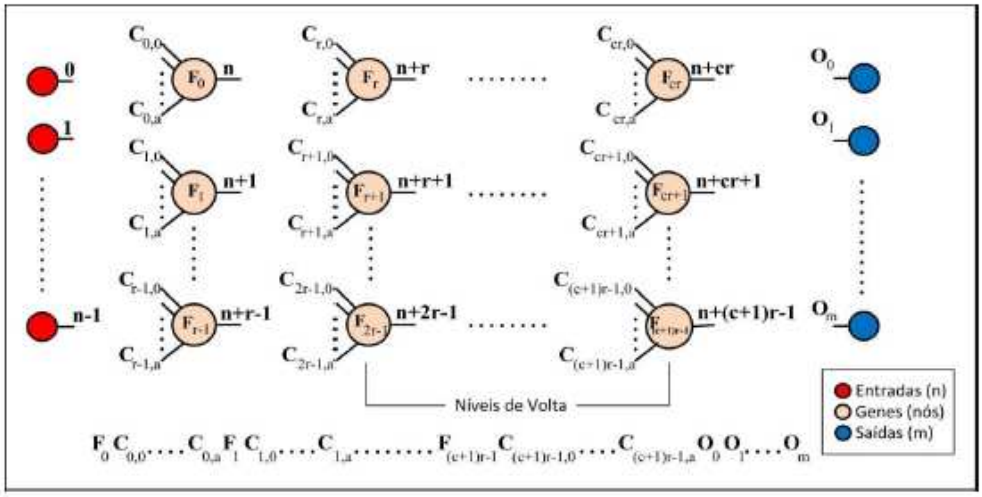


Figura22: Forma geral da PGC (adaptado de MILLER, 2011)

Na Figura 23 é demonstrado um exemplo de genótipo que é representado pelos seguintes parâmetros: n_c=3, n_r=2, I=n_c, n=2 e m=4, onde o primeiro número representa a função, o segundo representa a entrada superior e o terceiro, a entrada inferior de cada nó.

O nó que possui sua saída identificada com o número 6 não está conectado a nenhuma entrada de nós e a nenhuma das saídas do programa, isto significa que ele não será decodificado, e portanto, representará pouca sobrecarga computacional.

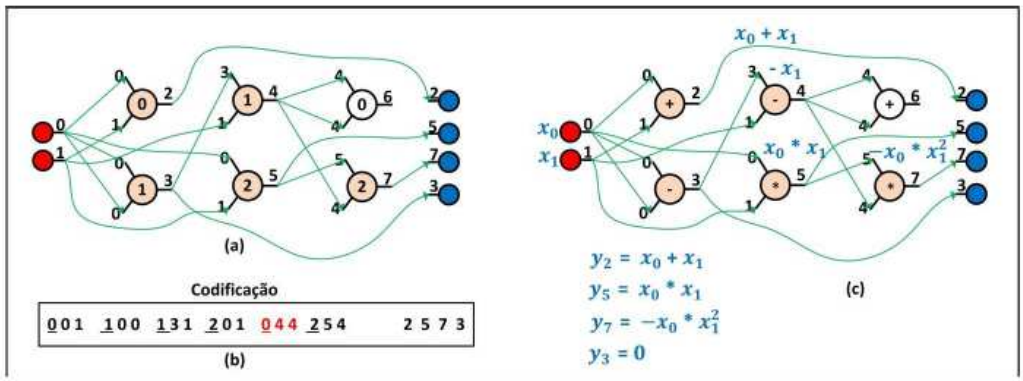


Figura23: Obtenção do grafo (adaptado de MILLER, 2011)

3.1Mutação

A mutação mais utilizada é determinada por um operador de ponto que é quando um alelo é escolhido aleatoriamente e em seguida é alterado para um valor também válido e aleatório (PARIS, 2013).

O exemplo da Figura 24destaca como uma pequena mudança no genótipo, na saída, pode produzir uma grande mudança no fenótipo. Os pontos em vermelho e pontilhado significam genes inativos e a saída em azul onde ocorre a mutação.

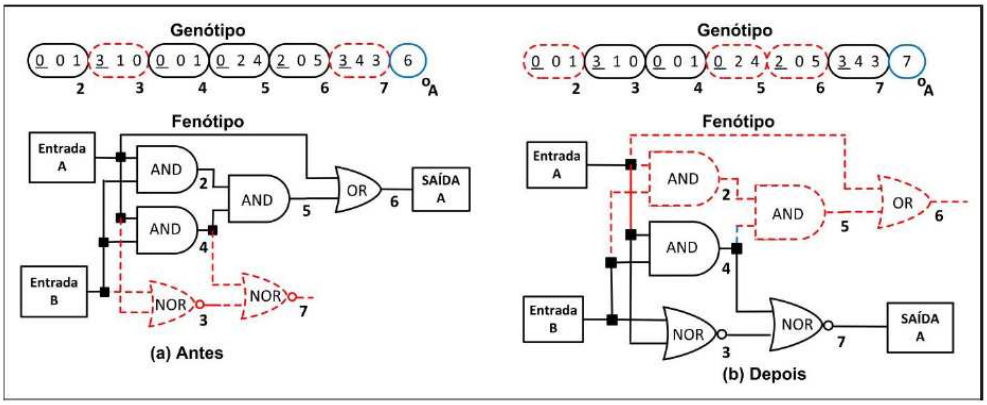


Figura24: Exemplo de mutação pontual (adaptado de MILLER, 2011))

Segundo (MILLER, 2011), quando a taxa de mutação ideal é utilizada, maiores genótipos necessitam de um menor número de avaliações da função de aptidão para alcançar o sucesso evolucionário que genótipos menores.

3.2Redundância

Uma forma de redundância é a inatividade de alguns genes que não exercem nenhuma influência sobre o fenótipo e por sua vez tem um efeito neutro sobre a aptidão.

O mapeamento genótipo-fenótipo (BANZHAF, 1994) consiste em separar oespaço de busca dos genótipos do espaço de soluções dos fenótipos. Muitos genótipospodem resultar em fenótipos com mesma aptidão, essa é uma vantagem deste mapeamento, pois estas variantes neutras são frequentes eimportantes para a variabilidade genética, esta característica é chamada de neutralidade.

A neutralidade fazcom que o processo evolucionário ultrapasse regiões onde as saídas teriam baixa “aptidão”,evitando a estagnação em pontos sub-ótimos. Mesmo que uma mutação passiva nãoaltere a aptidão imediatamente, ela pode ser utilizada em uma melhora futura.

As experiências feitas com a PGC (MILLER, 2011) concluem que é maisefetiva quando 95% dos genes estão inativos, resultado que acentua o benefício da neutralidade.

Em (MILLER; SMITH, 2006) foi investigado, dentre outras características e propriedades da PGC, o papel da redundância evolutiva onde foi possível determinar que o esforço computacional seja menor ao mesmo tempo em que a neutralidade é maior.

Em Programação Genética, em relação ao desempenho para grandes problemas, ocorre um fenômeno conhecido como*bloat*, que pode ser definido como o crescimento do programa sem significante retorno em termos da função objetivo (TUNER, 2015), criando um tempo limite para busca, deixando a criação e a avaliação doprograma mais demorada.

Mas, de acordo com (MILLER, 2001), o fato da pouca ocorrência de *bloat*na Programação Genética Cartesiana está ligada a presença de genes que podem ser ativados e desativados.

APGC vem sendo utilizado há mais de duas décadas em diversas aplicações e foi verificadoatravés das experiências que o *bloat* não ocorre de maneira significante na PGC. Osprogramas usando a PGC só aumentam de tamanho se for necessário para o aumento daaptidão (MILLER, 2001).

3.3Função de avaliação

A função de avaliação tem por objetivo qualificar a solução apresentada em termos do problema a ser resolvido. Neste trabalho, ela tem a função de avaliara a qualidade da clusterização realizada pelo algoritmo.O objetivo é minimizar ou maximizar o valor obtido pelo índice de validação escolhido como função de avaliação de forma a obter a melhor disposição de *clusters.*

O indivíduo com maior aptidão, com a melhor disposição de *clusters*, é promovido para a próxima geração até que um critério de parada seja atendido.

Na Figura 23, por exemplo, como o objetivo é obter funções matemáticas, a função de aptidão poderia ser considerada o cálculo do erro em um conjunto de pontos em que o valor da função é conhecido.

3.4Estratégia de evolução

Tipicamente a PGC utiliza uma estratégia evolucionária elitista $1 + \lambda$, onde usualmente os experimentos utilizamλé igual a4, a população inicial é gerada aleatoriamente pela definição de cada gene em cada cromossomo por um alelo aleatório válido (TURNER, 2015).

A Figura 25 apresenta a estratégia evolucionária.

1. Cinco indivíduos (genótipos)são criados aleatoriamente;
2. O indivíduo com maior valor na função de aptidão será promovido para a geração seguinte;
3. O operador mutação é aplicado quatro vezes no indivíduo mais apto da geração anterior;
4. Serão gerados quatro novos indivíduos, totalizando cinco como população atual;
5. O processo retorna para o passo2 enquanto o critério de parada não for atingido.

Figura25: Estratégia evolucionária (1+4) (adaptado de SANTOS; DO AMARAL; 2014)

No método proposto que será apresentado no próximo Capítulo, o método de aprendizado do APF foi substituído e em seu lugar foi utilizada a Programação Genética Cartesiana (PGC) para clusterização.

Os parâmetros obrigatórios necessários como entrada para o algoritmo serão detalhados no Capítulo 5.

4 MODELO PROPOSTO

Este capítulo apresenta um método para clusterização usando PGC com árvores de padrões *fuzzy* conforme Figuras 26 e 27.

O modelo é baseado em conceitos apresentados nos Capítulos 1, 2 e 3 e propõe que a clusterização seja tratada como um problema de otimização. Deseja-se obter um conjunto de árvores de padrões *fuzzy* que maximizem (ou minimizem) o índice que avalia a qualidade da clusterização.

Na Figura 26 é detalhado o processo utilizando os algoritmos de clusterização tradicionais. Nesta etapa serão avaliados quais parâmetros de entrada retornarão o melhor resultado para cada algoritmo.

Se oconjunto de dados puder ser representado de forma satisfatória através de*clusters* hiperesféricos, então o modelo proposto apresentará os *clusters* formados, sem que nenhum tratamento adicional seja necessário. Entretanto, se esta hipótese não for observada, então o modelo proposto fará a divisão do conjunto em diversos *clusters* hiperesféricos e sua agregação, para formar *clusters*de formato arbitrário. O algoritmo de máquinas de vetores de suporte será utilizado para auxiliar esta agregação e o método será detalhada em item posterior.

Os resultados obtidos pelos índices de validação serão utilizados para avaliar a disposição dos*clusters.*

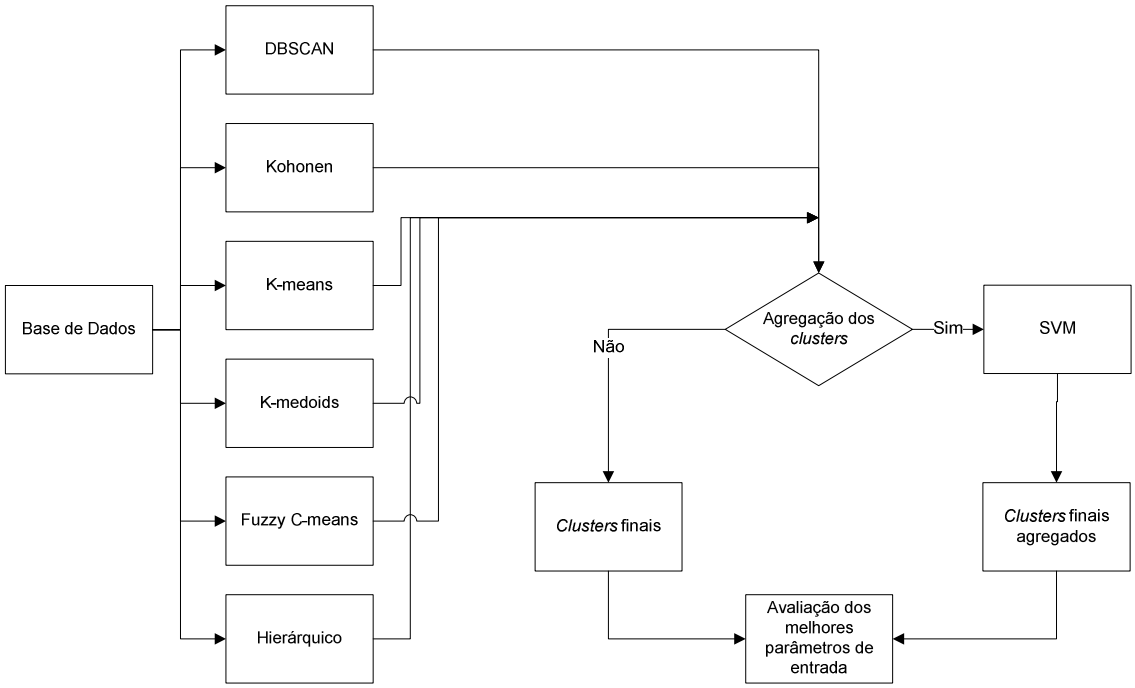


Figura 26: Apresentação geral do modelo (DO AUTOR, 2016)

Na Figura 27 é apresentado o método proposto utilizando como parâmetros de entrada a avaliação realizada anteriormente pelos métodos tradicionais. Estemodelo será detalhado nas próximas seções.

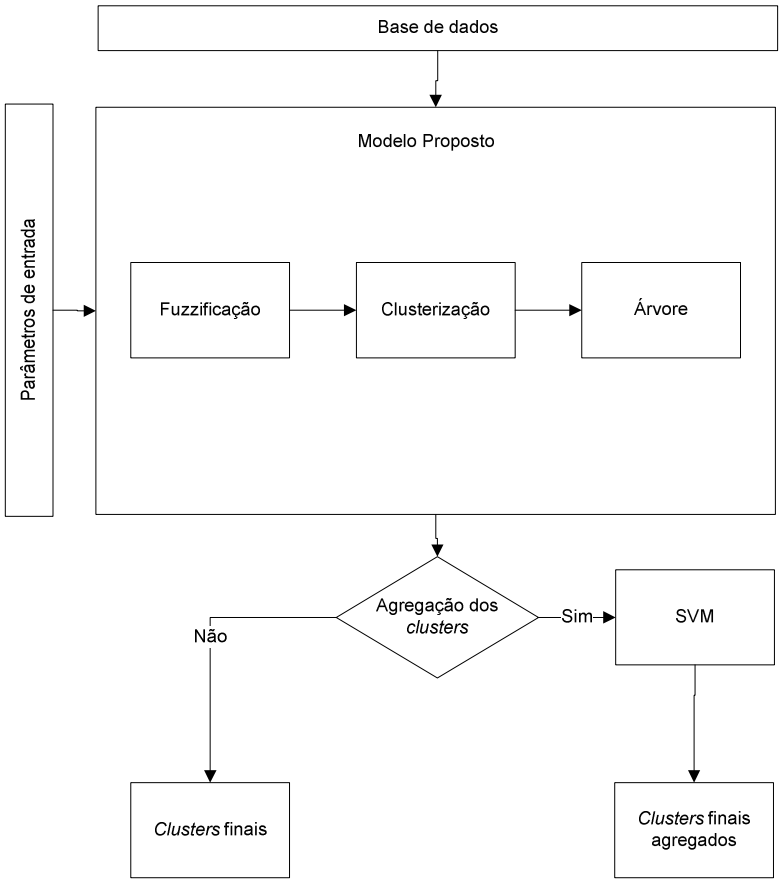


Figura27: Apresentação geral do modelo (DO AUTOR, 2016)

4.1Arcabouço proposto

Nesta fase, os métodos de clusterização clássica apresentados no Capítulo 1 serão utilizadospara ajustar o modelo proposto. Adicionalmente,será realizada a análise de resultado destes mesmos algoritmos em comparação com o modelo proposto.

Serão utilizados os algoritmos DBSCAN, Kohonen, K-means, K-medoids, Fuzzy C-means e Hierárquico(Figura26) para melhor avaliar os parâmetros de entrada necessários para o algoritmo PGC com árvores de padrões *fuzzy* que é apresentado na Figura27.

Quatro bases de dados artificiais eduas bases adicionais que foram obtidas a partir de problemas reais, como diagnóstico de doenças, serão utilizadas para comparar os resultados apresentados pelos algoritmos.

Será utilizada a informação da quantidade de *clusters* necessária para a entrada do algoritmo PGC e os resultados obtidos pelos índices de validade para avaliar a eficiência de cada método em relação ao método proposto.

Para avaliar se para aquela base de dados estudada será utilizado o método de agrupamento de *clusters*, serão avaliados os resultados obtidos pelos índices de validação que indicam que a formato dos *clusters* não é hiperesférico, sendo necessário outro método par agrupar os dados corretamente.

4.1.1 Particionamento Fuzzy

O particionamento será realizado em todo o conjunto de dados, que será utilizado no processo de obtenção da estrutura da APF.

O particionamento é feito escolhendo um atributo, por exemplo “idade” e selecionando subconjuntos deste atributo.

No método proposto, cada atributo teve seu domínio dividido em cinco termos linguísticos (termos *fuzzy*), denominados “Baixo”, ”Médio-Baixo”, “Médio”, “Médio-Alto”, “Alto”. A Figura28 exemplifica a partição de um atributo, cujo domínio é o intervalo [0,1]. O valor de cada atributo ativa a função de pertinência de cada termo *fuzzy* produzindo um valor de pertinência no intervalo [0,1].

Cada atributo presente no banco de dados terá um valor de pertinência associadoa cada um dos conjuntos*fuzzy*. Esta relação entre atributos e termos *fuzzy* serão as entradas da grade computacional da PGC, somando um total de entradas de cinco vezes a quantidade de atributos.

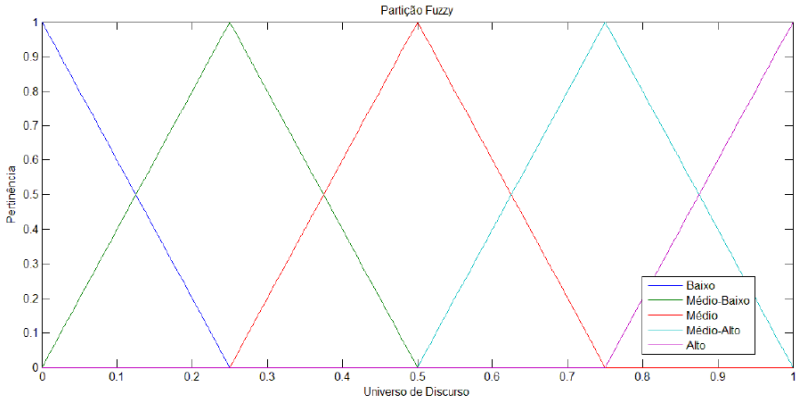


Figura28: Partição Fuzzy (DO AUTOR, 2016).

4.1.2 Operadores

Existem operadores utilizados naAPF, como mostrado no Capítulo 3.Porém com a intenção de aumentar a compreensão da solução (árvore), decidiu-se reduzir a quantidade de operadores, pois com base em experimentos preliminares a utilização de um conjunto reduzido de operadores não representou uma diminuição de acurácia significativa e contribuiu para facilitar a leitura da expressão fornecida pela árvore.

Os operadores utilizadosforam reduzidos para os seguintes:

$$Máximo = max(a, b) \quad (27)$$

$$Mínimo = min(a, b) \quad (28)$$

$$WA = xa + (1 - x)b \quad (29)$$

$$OWA = xmax(a, b) + (1 - x)min(a, b) \quad (30)$$

Os operadores *WA* e *OWA*são operadores parametrizados, onde *a* e *b* são os valores de entradas dos nós que serão operados e *x* será um valor aleatório dentro do intervalo [0,1], que poderá ser ajustado pelo operador de mutação. Os operadores *WA* e *OWA* são utilizados para preencher o espaço entre a maior combinação conjuntiva (t-norma) e a menor combinação disjuntiva (t-conorma). Os operadores foram codificados em números inteiros de acordo com a Tabela6.

Tabela7: Operadores utilizados (DO AUTOR, 2016).

Operador	Código
WA	0
OWA	1
Mínimo	2
Máximo	3

4.1.3Genótipo

Será utilizado o modelo em que é criado apenas um únicogenótipo com *n* saídas, onde cada saída representa um *cluste*r as alterações devido àmutação do genótipo podem influenciar diversos *clusters* (árvores) ao mesmo tempo.Apopulação neste caso será sempre igual a cinco, uma vez que a PGC usa a estratégia 1+ λ.

Assim, serão gerados cinco genótipos aleatórios na inicialização do algoritmo conforme Figura 25 que descreve a estratégia evolucionária.

Os genótipos são construídos poruma sequência de números inteiros, assim neste modelo foi fixado o número de linhas daPGC em 1.Neste caso o tamanho do genótipo irá mudar somente com a variação donúmero de colunas, onde cada coluna possui um nó e cada nó possui 3 genes.

Este modelo com o número de linhas limitado em 1 prevê que árvores mais simples sejam geradas, conforme definido por Miller, escolhendo um alto valor para o número de linhas serão geradas árvores mais altas (MILLER, 2011).

O primeirogene do nó se refere a um dos operadores de acordo com a Tabela 6, já o segundo eterceiro genes do nó representam os genes de conexão, ou seja, os pontos onde asentradas desse nó estão conectadas seja uma entrada ou a saída de um nó anterior.

Por fim, a quantidade total de genes em um genótipo será igual ao número de colunas vezes três,mais os genes de saída.

Na Figura 29 são representados três genótipos e suas respectivas árvores, no genótipo os nós que estão representados por “xxx” estão inativos e não estão conectados a saída.

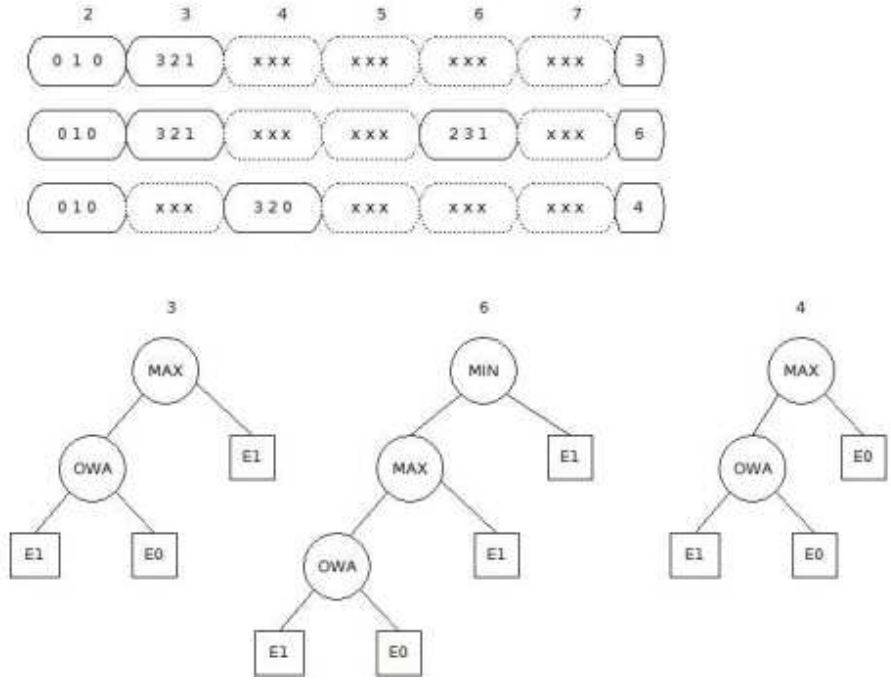


Figura29: Genétipo e suas árvores ((SANTOS; DO AMARAL, 2014).

4.1.4 Clusterização

Alguns parâmetros devem ser definidos previamente, como: a quantidade de linhas e colunas na grade computacional da PGC, o *levelback*, a quantidade máxima de gerações, o tamanho da população, a taxa de mutação do genótipo, a função utilizada para a avaliação do genótipo e quantidade de *clusters.*

A primeira etapa é a criação da população de genótipos inicial, que é criada de forma aleatória obedecendo aos limites impostos pelos parâmetros linhas, colunas, *levelback*e tamanho da população do modelo. A estratégia evolutiva utilizada é 1+λ com λ=4.

4.1.5Avaliação e critérios de parada

A avaliação dos genótipos proposta será realizada pelo critério de validação Calinski e Harabasz que escolhe como melhor aquele em que obtém o maior valor do índice, ou seja, o indivíduo com maior aptidão é promovido para a próxima geração.Como resultado desta avaliação, este índice gera *clusters* no formato hiperesférico.

Outros índices como o Silhouette, Davies Bouldin e Dunn também foram avaliados, porém a utilização do Calinski e Harabasz obteve modelos com um melhor desempenho em diferentes conjuntos de dados, sendo, portanto, o índice a ser utilizado em primeiro lugar.

Os índices de validação para avaliação dos genótipos Davies Bouldin e Dunn foram descartados a partir da análise do conjunto de dados ES2 que será apresentado no Capítulo 5.

Utilizando o índice Silhouette para avaliar a formação dos *clusters*gerados pelo algoritmo PGC-APF conforme Tabela 8, é possível observar que os índices Davies Bouldin e Dunn para avaliação do genótipo não apresentam resultados competitivos em relação ao Calinski e Harabasz, pois quanto mais próximo do valor um, melhor é o resultado obtido na clusterização.

Tabela 8: Comparação de função de aptidão para conjunto de dados ES2 (DO AUTOR, 2017)

	Davies Bouldin	Dunn	Calinski
ES2 - Silhouette	-0.2996	0.3005	0.8885

Já na Tabela 9, utilizando o índice de Dunn para avaliar a formação dos *clusters* conforme Tabela 9, é possível observar que o índice Silhouette para avaliação do genótipo não apresenta resultado competitivo em relação ao Calinski e Harabasz. Quanto mais alto a valor do índice de Dunn, melhor é o resultado obtido na clusterização.

A estratégia usada para gerar os dados da Tabela 9 é a mesma que será apresentada no item 5.1.5, utilizando a etapa de agregação de *clusters* quando o conjunto de dados forma *clusters* de formato arbitrário.

Tabela 9: Comparação de função de aptidão para conjunto de dados Banana 1 (DO AUTOR, 2017)

	Silhouette	Calinski
Banana 1 - Dunn	0.0297	0.2307

Já a quantidade de gerações é definida como critério de parada, este parâmetro é definido no algoritmo e determina a quantidade limite de gerações.Ele é útil para limitar o tempo total de execução caso os outros critérios não sejam acionados.

Além deste critério, é avaliado o resultado obtido pela função de avaliação com o índice de validação obtido com os resultados dos outros métodos de clusterização.

4.1.6 Árvore

Após a clusterização, é aplicado um método de decodificação do genótipo, que parte da saída e varre o genótipo para encontrar os genes que estão conectados a esta saída. Este método será repetido para todos os nós conectados até que se tenha a informação dos nós que estão conectados para formar a solução e também a ordem de conexão desses nós.

A quantidade de árvores geradas é igual à quantidade de *clusters* formada, assim, fazendo uma análise dos operadores e termos *fuzzy* presentes na árvore, é possível verificar a influência de cada atributo e de cada partição por atributo para a geração daquele *cluster*.

Fazendo a avaliação da árvore de um conjunto de dados bidimensional de teste ES1, que será apresentado com mais detalhes no Capítulo 5, que possui dois atributos

de entrada que estão associados a um termo *fuzzy* que representa um intervalo do universo de discurso do atributo.

Para simplificar a análise deste item, foram consideradas somente duas das quatro árvores geradas por este conjunto que possui quatro grupos.

É possível avaliar na Figura 30 que para este *cluster*, a partição "Médio-Baixo" do atributo 2 tem maior influência na definição deste agrupamento, pois está mais acima na árvore.

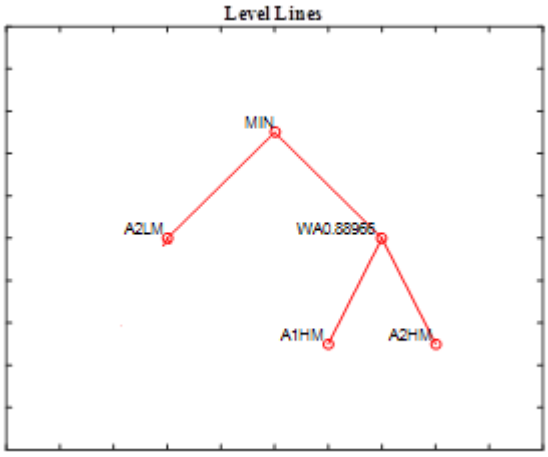


Figura30: Árvore do cluster "1" (DO AUTOR, 2016).

Já a Figura 31 apresenta a partição "Baixo" do atributo 2 e "Médio-Baixo do atributo 1, porém para avaliar qual delas apresenta maior influência, é necessário analisar o valor do operador *WA*.

O valor de pertinência obtido nos conjuntos *fuzzy* vai sendo agrupado através de operadores que mantêm os resultados parciais no intervalo [0,1], neste caso 0,41382, para o atributo 2, o peso é 0,41382 e para o atributo 1 é o complemento, resultando em 0,58618. Logo, o atributo 1 tem maior influência na definição deste *cluster*.

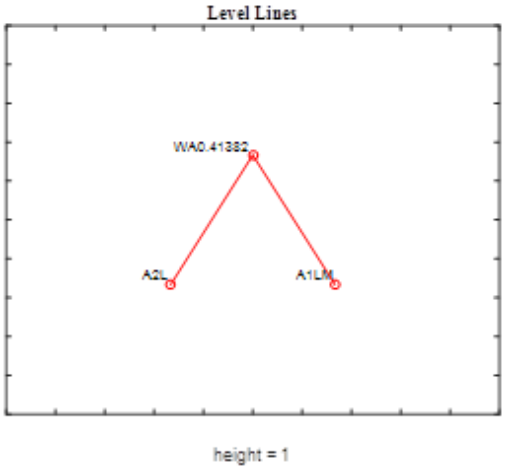


Figura31: Árvore do cluster "2" (DO AUTOR, 2016).

4.2Agregação dos *clusters*

O algoritmode máquinas de vetores de suporte ((*Support vector machines* - SVM) é usado aqui para determinar o quão próximos estão dois *clusters*. Isto é realizado calculando-se a fronteira de decisão entres os *clusters* dois a dois e calculando a mediana das margens dos vetores de suporte. Quanto menor esta mediana, mais próximos eles estão, o que significa que podem ser agregados em um novo *cluster*.

Conforme exemplificado na Tabela 3,*clusters* com formato arbitrário somente podem ser reconhecidos pelo o algoritmo DBSCAN e algumas técnicas do Hierárquico Aglomerativo. No caso do modelo proposto, a função de avaliação escolhida faz com que sejam gerados c*luster*shiperesféricos.

Para realizar a agregação dos *clusters*, é definida uma quantidade maior para o número de *clusters*como parâmetro de entrada no método de clusterização de forma que *clusters* hiperesféricos possam ser agregados em formatos não hiperesféricos.

O processo de agregação é iniciado com a utilização do algoritmo SVM que é um método supervisionado baseado na teoria do aprendizado estatístico, que auxilia na determinação de quais os *clusters*são mais indicados para serem agregados.

Inicialmente, o SVM calcula a fronteira de decisão entre os *clusters* dois a dois. A seguir, para cada par de *clusters* calcula-se a mediana das margens dos vetores. Quanto menor for o valor desta mediana, mais próximos estão estes *clusters*, indicando que são mais indicados para serem agregados.

O processo de agregação é iniciado com a posse dos índices de *clusters* definidos na etapa anterior através dos algoritmos de clusterização.

Os resultados obtidos com a agregação de *clusters* utilizando a SVM serão comparados com o apresentado pela clusterização realizada anteriormente.

Na Figura 32 é exemplificada a agregação de *clusters*, a decisão para que o algoritmo utilize esta etapa depende da interpretação dos resultados obtidos pelos índices de validação ou mesmo do conhecimento do formato dos *clusters* para o conjunto de dados escolhido.

Este processo de agregação pode ser utilizado para clusterização se o algoritmo utilizado somente gerar *clusters* de formatos hiperesféricos conforme Tabela 3 e se o índice de validação avaliado indicar que este conjunto de dados gera *clusters* de formato arbitrário ou mesmo se houver um conhecimento a priori sobre este conjunto, que resultado da clusterização através de grupos hiperesféricos não trará resultados satisfatórios.

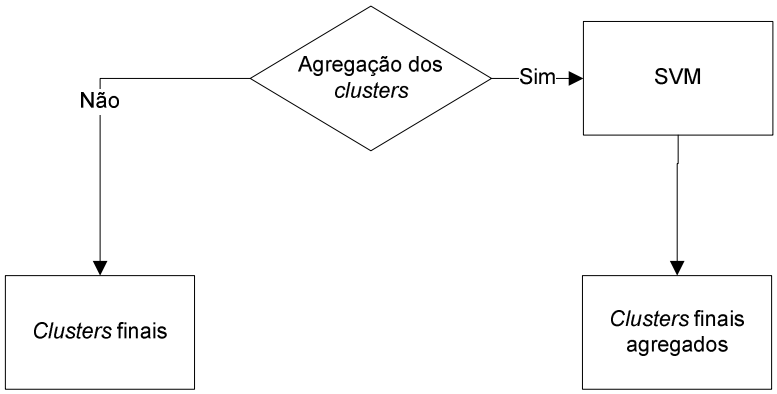


Figura32: Fluxo do SVM (DO AUTOR, 2016)

4.3Análise de agrupamento

Nesta fase existem duas ações principais conforme abaixo (HALKIDI; BATISTAKIS, 2001) e demonstradas na Figura 34:

- Validação dos resultados,os resultados obtidos pelos algoritmossão verificados utilizando critérios e técnicas apropriadas;
- • A interpretação dos resultados. Em muitos casos, os especialistas na área de aplicação têm que integrar os resultados de agrupamento com outra evidência experimental e análise, a fim de chegar à conclusão certa.

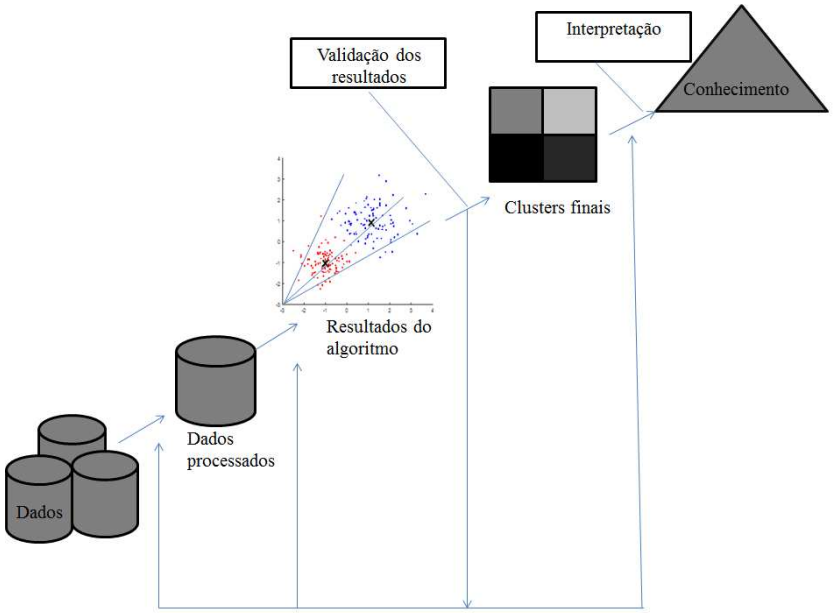


Figura33: Etapas do processo de clusterização (HALKIDI; BATISTAKIS, 2001)

A análise de agrupamento é uma ferramenta importante em uma série de aplicações em muitas áreas de negócios e Ciência. Uma das funções é a de geração de hipóteses em que a análise de *cluster* é usada aqui para inferir algumas hipótesessobre os dados(HALKIDI; BATISTAKIS, 2001). Por exemplo, é possível encontrar em um banco de dados de varejo que há dois grupos significativos de clientes com base em sua utilização de serviços. Então, podem-se inferir algumas hipóteses para os dados, que clientes em áreas que apresentam maior falha de rede tendem a utilizar menos serviços de mensagens de texto ou clientes em regiões com maior renda contratam mais serviços de recarga, por exemplo.

Esta é uma característica típica de aplicaçõesde agrupamento. Nos negócios, a clusterização pode ajudar os comerciantes a descobrir grupos significativos utilizando informações disponíveis no banco de dados de clientes e caracterizá-los com base em seus padrões de compra.(HALKIDI; BATISTAKIS, 2001)

5 EXPERIMENTOS

O objetivo do Capítulo de experimentos é primeiramente comparar os resultados obtidos pelos índices de validação definidos no Capítulo 1 para cada método de clusterização em cada conjunto de dados especificado na Tabela 10.

Após esta etapa, será avaliada a correlação entre a matriz de proximidade e incidência em que é possível verificar que em técnicas de clusterização que obtém formato de *clusters* hiperesféricos este resultado é mais próximo de um, porém este índice não consegue validar corretamente *clusters* de formato arbitrário.

O tratamento com o algoritmo SVM é realizado para melhorar os resultados verificados em algoritmos que tratam somente formato hiperesférico e neste passo foram considerados somente conjuntos identificados anteriormente com *clusters* de formatos arbitrários.

Após esta avaliação, conjuntos aleatórios são gerados com a mesma quantidade de dados dos conjuntos de teste considerados e aplicados nos métodos que retornaram o melhor resultado na etapa do algoritmo SVM. O objetivo desta etapa é verificar se os índices de validação retornados por conjunto aleatórios diferem aos obtidos por conjuntos reais e sustentar que o método proposto apresenta resultados coerentes.

Por fim, é avaliada a interpretabilidade das árvores obtidas pelos conjuntos *Breast Cancer*, Iris e o exemplo de segmentação de mercado, nesta etapa é verificado que as informações retornadas pelas árvores auxiliam na identificação de características do *cluster*, diminuindo a necessidade de possuir um especialista para esta análise, ou mesmo, auxiliando o mesmo a comprovar que há lógica na análise.

Todos os algoritmos foram executados no Matlab na versão R2011a.

5.1 Estudo de casos com bases de dados artificiais

As bases abaixo serão utilizadas serão utilizadas para a avaliação dos algoritmos de clusterização conforme Tabela10.

Tabela10: Descrição das bases artificiais (DO AUTOR, 2016).

Nome	Atributos	Pontos	Grupos
ES1	2	400	4
ES2	2	400	4
Banana1	2	200	2
Banana 2	2	2000	2
Iris	4	150	3
Breast cancer	9	682	2

Os conjuntos de dados ES1, ES2, Banana 1 e Banana 2 são bases de dados artificiais para avaliação do resultado do algoritmo e apresentam resultados esperados conhecidos.

Os conjuntos de dados Iris e *Breast cancer* são bases de dados utilizadas em arquivos científicos com dados reais e tem como objetivo avaliar se a correta interpretação dos resultados obtidos pelo algoritmo é possível. Estas bases foram retiradas do UCI *Machine Learning Repository*.

Os métodos utilizados foram descritos no Capítulo 1 e são resumidos na Tabela11 adicionando a principal característica para cada um.

Tabela11: Métodos de clusterização (DO AUTOR, 2016).

	K-means	FCM	K-medoids (PAM)	Aglomerativo	DBSCAN	SOM
Formato do *cluster*	Hiperesférico	Hiperesférico	Hiperesférico	Hiperesférico e arbitrário	Arbitrário	Esférico

5.1.1 Avaliação de quantidade de *clusters*

Para cada clusterização, é registrada a soma das distâncias euclidianas dos pontos aos centros dos *clusters*. A partir dos resultados obtidos nesse processo, é estimado o número de *clusters* que melhor representa os padrões existentes nos dados. Essa estimativa é feita pela análise do gráfico plotado a partir do número de *clusters*contra o somatório das distâncias euclidianas dos pontos aos centros dos *clusters* (KUMAR, 2006).

Para o conjunto de dados ES1 e ES2 são apresentados os gráficos para cada algoritmo em que a quantidade de *clusters* é um parâmetro de entrada. Na análise é possível verificar a melhor disposição de *clusters* para os métodos em que é necessário definir a quantidade de *clusters* como entrada, conforme apresentado nas Figuras 35 e 36para o conjunto de dados ES1e nas Figuras 37e 38 para o conjunto de dados ES2.

Para os dados ES1 e ES2, o valor encontrado para a quantidade de *clusters* como entrada do algoritmo é o mesmo que o esperado pelo número de grupos mapeado para estes conjuntos de dados.

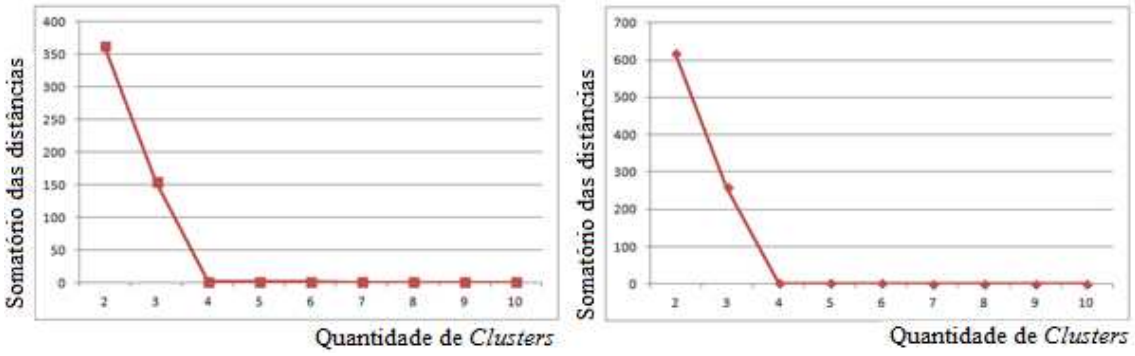


Figura 34: Gráfico da somas das distâncias ES1 para o algoritmo K-means (esquerda) e K-medoids (direita) (DO AUTOR, 2016).

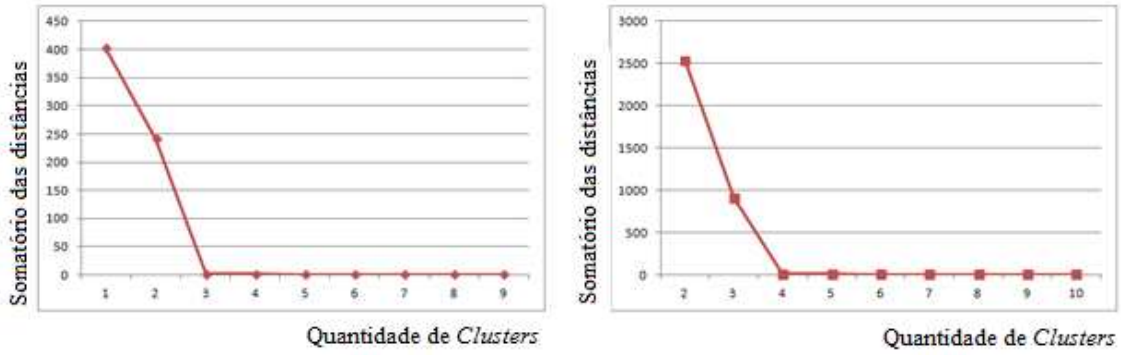


Figura 35: Gráfico da somas das distâncias ES1 (esquerda) para o algoritmo Fuzzy C-means e Aglomerativo (direita) (DO AUTOR, 2016).

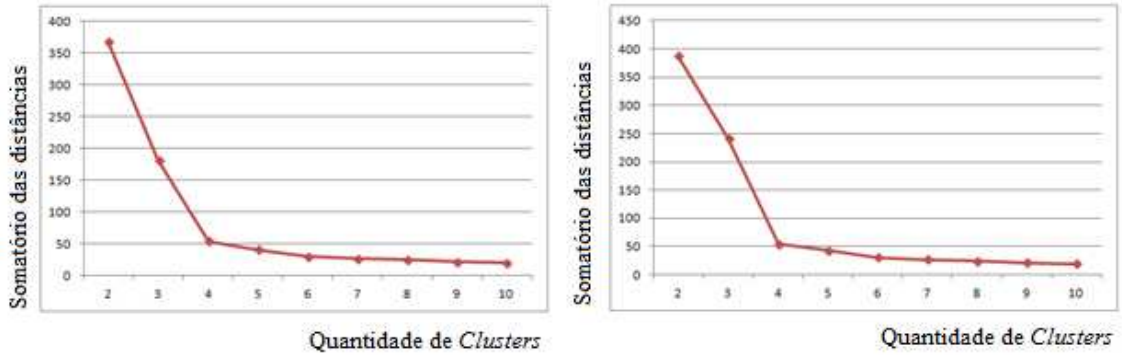


Figura 36: Gráfico da somas das distâncias ES2 para o algoritmo K-means (esquerda) e K-medoids (direita) (DO AUTOR, 2016).

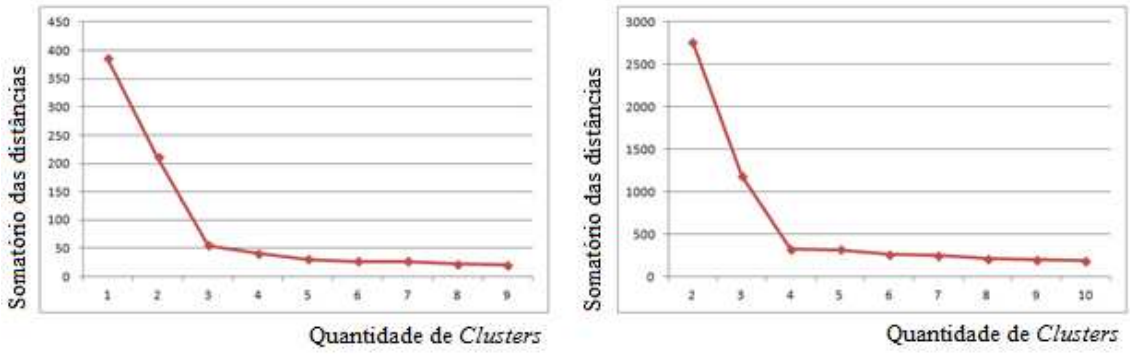


Figura 37: Gráfico da somas das distâncias ES2 para o algoritmo Fuzzy C-means (esquerda) e Aglomerativo (direita) (DO AUTOR, 2016).

Em relação aos dados Banana 1 e Banana 2, será utilizada a mesma análise, porém será considerado a melhor quantidade de *clusters*que foi obtida na maioria dos algoritmos, seis para efeito de comparação de resultados.

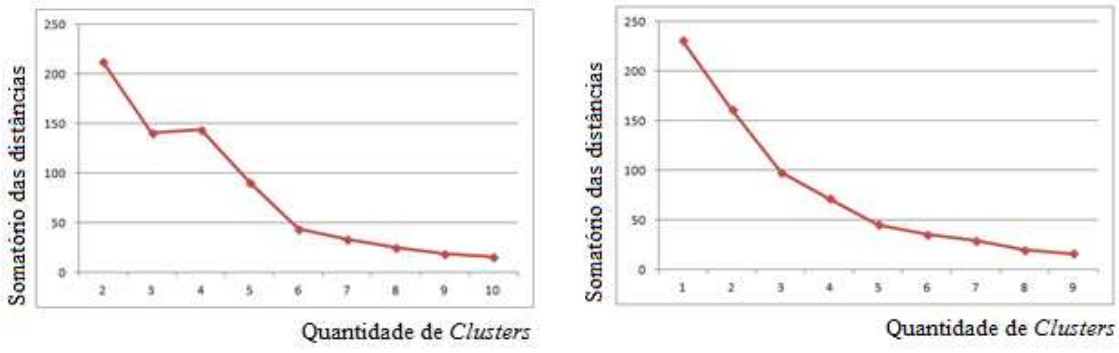


Figura 38: Gráfico da somas das distâncias Banana1 para o algoritmo K-means (esquerda) e K-medoids (direita) (DO AUTOR, 2016).

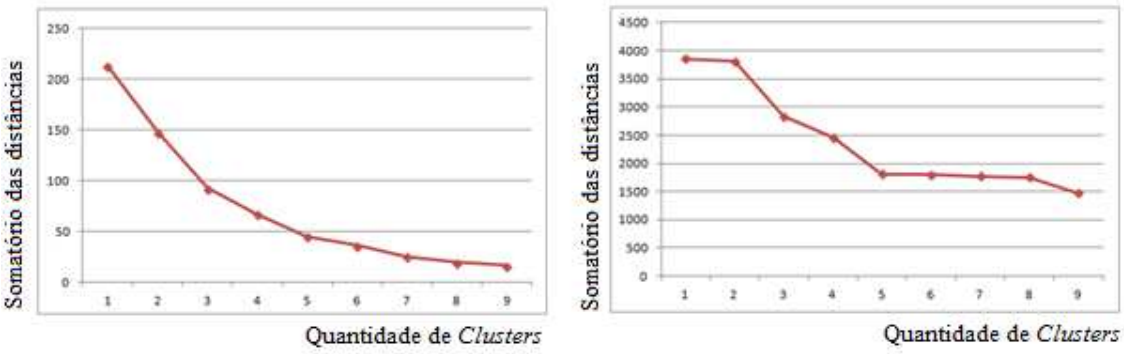


Figura 39:Gráfico da somas das distâncias Banana 1 para o algoritmo Fuzzy C-means (esquerda) e Aglomerativo (direita) (DO AUTOR, 2016).

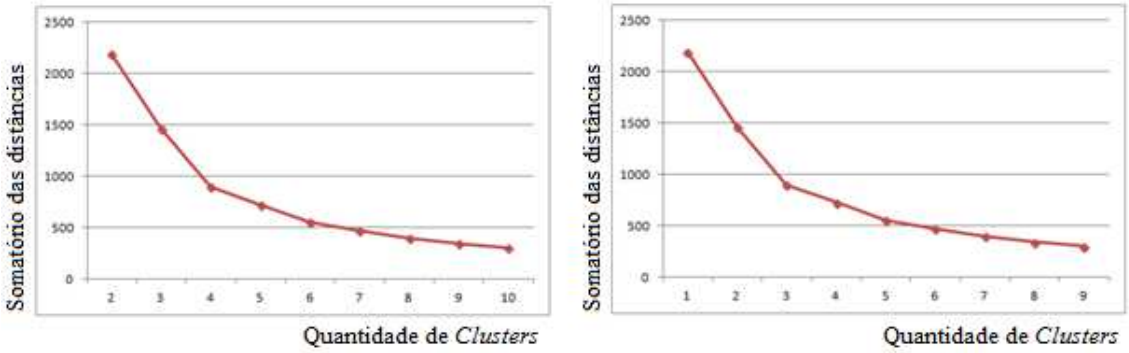


Figura 40: Gráfico da somas das distâncias Banana 2 para o algoritmo K-means (esquerda) e K-medoids (direita) (DO AUTOR, 2016).

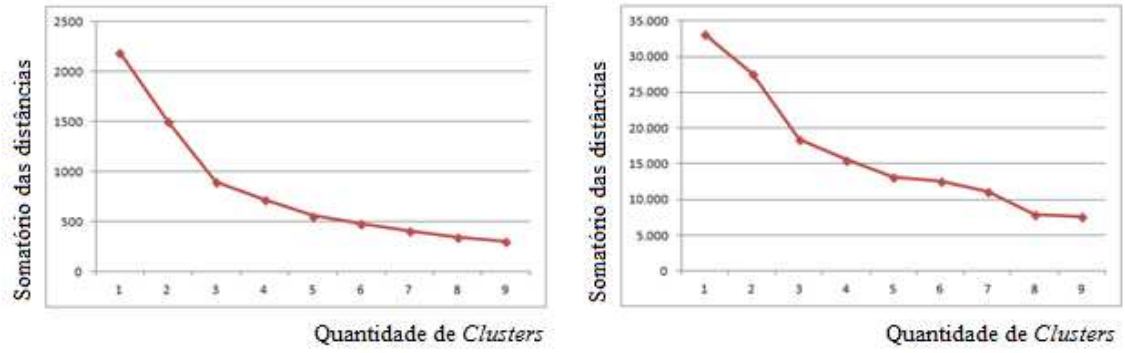


Figura 41: Gráfico da somas das distâncias Banana 1 para o algoritmo Fuzzy C-means (esquerda) e Aglomerativo (direita) (DO AUTOR, 2016).

5.1.2 Avaliação dos parâmetros de entrada

Para cada algoritmo,será avaliado o retorno dos índices de validação apresentados no Capítulo 1 de forma a avaliar a eficiência de cada método.

Este experimento foi executado trinta vezes para cada algoritmo e conjunto de dados e calculada a média e o desvio padrão para cada índice de validação.

Para o algoritmo K-meanse Aglomerativo Hierárquico,a distância euclidiana foi utilizada como medida de similaridade.

Utilizando o algoritmo Hierárquico Aglomerativo, as técnicas aglomerativas que apresentaram o melhor resultado para cada conjunto de teste são apresentadas na Tabela12.

Tabela12: Parâmetro linkage (DO AUTOR, 2016).

Nome	Técnica
ES1	*2.1.1 Average*
ES2	*2.1.2 Average*
Banana 1	*2.1.3 Single*
Banana 2	*2.1.4 Weighted*

Para o algoritmo DBSCAN, os parâmetros de entrada *Eps* e *MinPts* para cada conjunto de teste estão dispostos na Tabela 13.

Para este algoritmo foi utilizado o índice de Dunn como avaliação para a escolha dos parâmetros de entrada, pois é um dos índices de validação que melhor avalia a qualidade para *clusters* de formato arbitrário e conforme avaliação de Agrawal e Patel (AGRAWAL;PATEL, 2015), obtém melhores resultados para o algoritmo DBSCAN.

Tabela13: Parâmetros de entrada (DO AUTOR, 2016).

Nome	Eps	MinPts
ES1	0.8	7
ES2	0.8	10
Banana 1 com 2 *clusters*	2.1	3
Banana 1 com 6 *clusters*	0.9	7
Banana 2 com 2 *clusters*	0.9	30
Banana 2 com 6 *clusters*	0.8	30

Para o algoritmo PGC-APF, os melhores resultados foram obtidos com: linhas=1; colunas=20,*levelback*=19, tamanho da população=5, taxa de mutação=0.2 e função de aptidão= critério de Calinski Harabasz.Para o algoritmo PGC, a curva dedesempenho do melhor indivíduo para cada geração é apresentada nas Figuras 42 e 43.

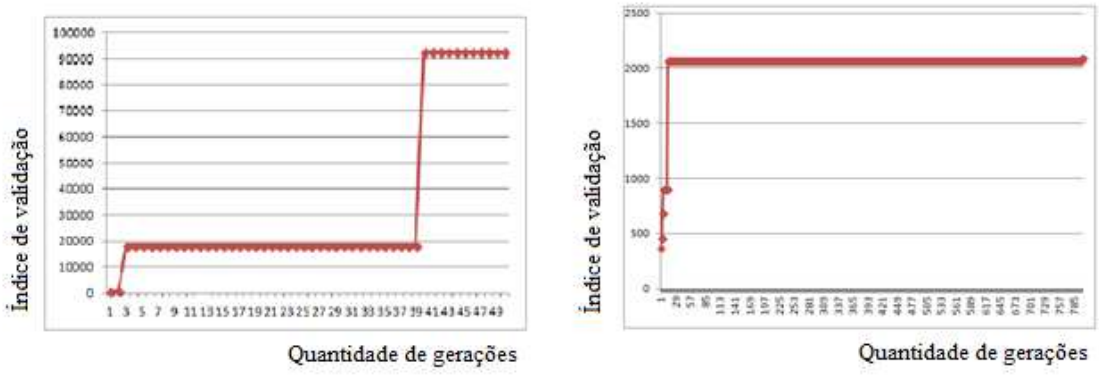


Figura 42: Curva de desempenho ES1 (esquerda) e ES2 (direita) (DO AUTOR, 2016).

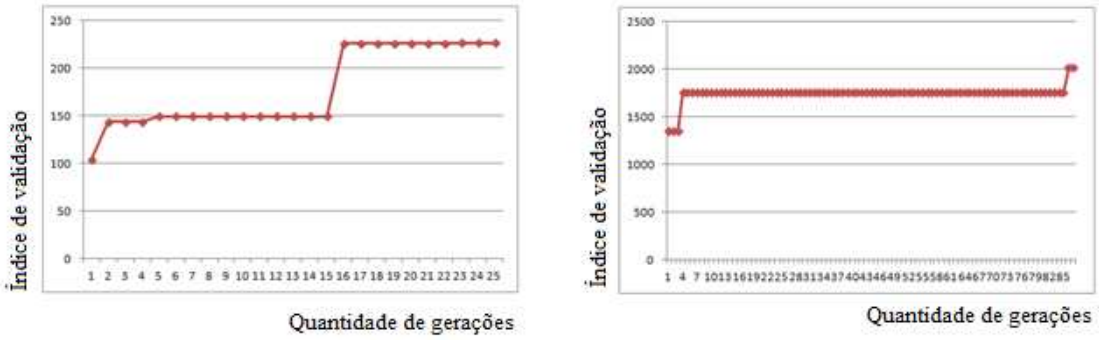


Figura 43: Curva de desempenho Banana 1 (esquerda) e Banana 2 (direita) (DO AUTOR, 2016).

5.1.3 Avaliação dos índices de validação

O primeiro conjunto de dados a ser avaliado é o ES1 como pode ser observado na Figura 44. Os grupos são visualmente separáveis, sem ruído.

Cada experimento foi executado trinta vezes para cada algoritmo e conjunto de dados e calculada a média e o desvio padrão para cada índice de validação, os resultados obtidos serão preenchidos nas tabelas no formato “média ± desvio padrão”.

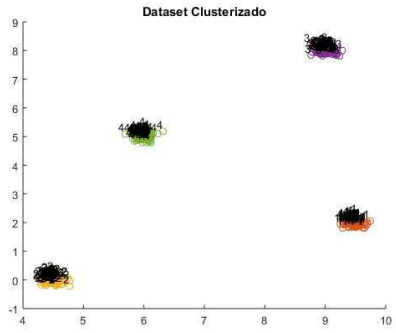


Figura 44: Conjunto de dados ES1 (DO AUTOR, 2016).

O resultado obtido pela clusterização para todos os algoritmos foi competitivo e nãoapresentou diferenças, como pode ser verificado na Tabela 14.

Tabela14: Valores dos índices para o ES1com 2 *clusters* (DO AUTOR, 2016).

ES1	K-means	K-medoids	C-means	Aglomerativo	DBSCAN	SOM	PGC/APF
Silhouette	**0.9981±0**	**0.9981±0**	**0.9981±0**	**0.9981±0**	**0.9981±0**	**0.9981±0**	**0.9981±0**
CH	**92613±0**	**92613±0**	**92613±0**	**92613±0**	**92613±0**	**92613±0**	**92613±0**
DB	**0.0540±0**	**0.0540±0**	**0.0540±0**	**0.0540±0**	**0.0540±0**	**0.0540±0**	**0.0540±0**
Dunn	**5.7062±0**	**5.7062±0**	**5.7062±0**	**5.7062±0**	**5.7062±0**	**5.7062±0**	**5.7062±0**

Para o conjunto de dados ES2, é possível verificar que o algoritmo DBSCAN apresenta resultado sensivelmente melhor que os outros algoritmos. Esta melhora no resultado ocorre devido à retiradados ruídos do conjunto de dados.

Na Figura 45, é possível avaliar o conjunto de dados e análise realizada pelo algoritmo DBSCAN, identificando os ruídos e os retirando da análise.

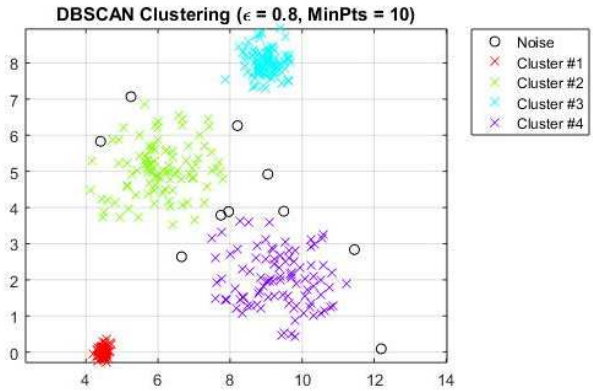


Figura 45: Conjunto de dados ES2 (DO AUTOR, 2016).

Na Tabela 15, é possível identificar os índices de validação retornados por todos os algoritmos avaliados para o conjunto de dados ES2.

Tabela15: Valores dos índices para o ES2 com 2 *clusters* (DO AUTOR, 2016).

ES2	K-means	K-medoids	C-means	Aglomerativo	DBSCAN	SOM	PGC
Silhouette	0.8893±0	0.8893±0	0.8897±0	0.8896±0	**0.9083±0**	0.8824±0.09	0.8885 ± 10^{-3}
CH	2128.4±0	2128.4±0	2131.8±0	2124.5±0	**2487.8±0**	2104.1±725	2120.0±33
DB	0.3889±0	0.3889±0	**0.3896±0**	0.3887±0	0.3670±0	0.3892±0.07	0.3895 ± 10^{-3}
Dunn	0.0412±0	0.0412±0	0.0550±0	0.0980±0	**0.3670±0**	0.0845±0.02	0.07165 ± 10^{-2}

Para o conjunto de dados Banana 1 e Banana 2 serão realizadas duas avaliações em relação à quantidade de *clusters* de entrada, pois para o algoritmo DBSCAN e para o PGC, o algoritmo apresenta o melhor índice de validação para a disposição de dois *clusters*.

Também será considerada a análise preliminar para os algoritmos em que a melhor disposição de *clusters* retornou seis *clusters* comparando com a disposição de dois *clusters*.

Para o algoritmo PGC-APF foram desativados os parâmetros: modificador e complemento para diminuir o espaço de busca e para o algoritmo retornar um resultado com mais de dois *clusters* para os conjuntos de dados Banana 1 e Banana 2.

Na Tabela 16 compara os índices de validação obtidos por todos os algoritmos avaliados para o conjunto de dados Banana 1 com a formação de 2 *clusters*.

Tabela16: Valores dos índices para o Banana 1 com 2 *clusters* (DO AUTOR, 2016).

Banana1	K-means	K-medoids	C-means	Aglomerativo	DBSCAN	SOM	PGC/APF
Silhouette	0.5525±0	0.5115±0.3	0.5525±0	0.3672±0	0.3672±0	0.5002±0.6	$\mathbf{0.572\pm10^{-3}}$
CH	**181,073±0**	155.40±21	**181,073±0**	110.94±0	110.94±0	131.16±42	172.65±0.9
DB	0.9979±0	1.0706±0.6	0.9979±0	**1.2856±0**	**1.2856±0**	0.9032±0.1	1.015 ± 10^{-3}
Dunn	0.0252±0	0.042 ± 10^{-3}	0.0252±0	**0.2307±0**	**0.2307±0**	0.047 ± 10^{-2}	0.028 ± 10^{-1}

O retorno do índice de Dunn relativamente maior para os algoritmos DBSCAN e Hierárquico Aglomerativo, para o conjunto de dados Banana 1 e Banana 2, indica que os dois conjuntos de dados formam clusters de formato arbitrário.

O algoritmo Aglomerativo, neste caso, é capaz de avaliar corretamente os grupos formadose é competitivo aos resultados obtidos pelo DBSCAN, pois o método ligação simples é capaz de identificar *clusters* de formato arbitrário.

Na Tabela 17 compara os índices de validação obtidos por todos os algoritmos avaliados para o conjunto de dados Banana 1 com a formação de 6 *clusters*.

Tabela17: Valores dos índices para o Banana 1 com 6 *clusters* (DO AUTOR, 2016).

Banana1	K-means	K-medoids	C-means	Aglomerativo	DBSCAN	SOM	PGC/APF
Silhouette	0.687 ± 10^{-2}	0.6874 ± 0	**0.6900 ± 0**	0.1532 ± 0	0.3202 ± 0	0.626 ± 10^{-1}	0.41 ± 10^{-1}
CH	329.21 ± 4.4	332.60 ± 0	**335.32 ± 0**	90.069 ± 0	83.852 ± 0	285.15 ± 41	151 ± 10^{-2}
DB	0.607610^{-2}	0.5996 ± 0	0.5899 ± 0	0.7861 ± 0	0.7220 ± 0	0.617 ± 10^{-1}	**1.04 ± 10^{-1}**
Dunn	0.046 ± 10^{-3}	0.0248 ± 0	0.0239 ± 0	**0.0700 ± 0**	0.0513 ± 0	0.051 ± 10^{-2}	0.01 ± 10^{-3}

Para o conjunto de dados Banana 2 foi realizada a mesma análise que o conjunto Banana 1.Na Tabela 18 compara os índices de validação obtidos por todos os algoritmos avaliados para o conjunto de dados Banana 2 com a formação de 2 *clusters*.

Tabela18: Valores dos índices para o Banana 2 com 2 *clusters* (DO AUTOR, 2016).

Banana2	K-means	K-medoids	C-means	Aglomerativo	DBSCAN	SOM	PGC/APF
Silhouette	0.555 ± 10^{-4}	0.541 ± 10^{-2}	0.559 ± 10^{-6}	**0.5952 ± 0**	0.4534 ± 0	0.453 ± 10^{-2}	0.522 ± 10^{-3}
CH	1872.6 ± 2	1777.2 ± 126	**1899.4 ± 10^{-1}**	2133.8 ± 0	1343.7 ± 0	1095.0 ± 450	1636 ± 10^{-2}
DB	0.9683 ± 10^{-4}	0.9958 ± 10^{-2}	0.9636 ± 10^{-4}	0.8782 ± 0	**1.1120 ± 0**	0.7889 ± 10^{-1}	1.030 ± 10^{-2}
Dunn	0.0042 ± 10^{-4}	0.0054 ± 10^{-3}	0.0088 ± 0	0.0127 ± 0	**0.0246 ± 0**	0.0084 ± 10^{-3}	0.004 ± 10^{-3}

Na Tabela 19 compara os índices de validação obtidos por todos os algoritmos avaliados para o conjunto de dados Banana 2 com a formação de 6 *clusters*.

Tabela19: Valores dos índices para o Banana 2 com 6 *clusters* (DO AUTOR, 2016).

Banana 2	K-means	K-medoids	C-means	Aglomerativo	DBSCAN	SOM	PGC/APF
Silhouette	0.541 ± 10^{-4}	**0.5458 ± 10^{-3}**	0.544 ± 10^{-5}	0.4012 ± 0	0.4299 ± 0	0.472 ± 10^{-2}	0.179 ± 10^{-1}
CH	2397 ± 10^{-1}	**2422.4 ± 52**	2408 ± 10^{-1}	1667.3 ± 0	1258.6 ± 0	1933.9 ± 247	1025 ± 10^{-2}
DB	0.877 ± 10^{-3}	0.8534 ± 10^{-2}	0.867 ± 10^{-3}	**1.089 ± 0**	0.6442 ± 0	0.820 ± 10^{-2}	1.456 ± 10^{-1}
Dunn	0.008 ± 10^{-4}	0.0049 ± 10^{-3}	0.0064 ± 0	**0.0166 ± 0**	0.0111 ± 0	0.011 ± 10^{-3}	0.004 ± 10^{-3}

O índice de validação que melhor trata oformato de *cluster* arbitrário é o de Dunn. Com a avaliação deste item, é possível verificar que os algoritmos DBSCAN e Hierárquico Aglomerativo apresentam o melhor resultado na formação de *clusters* de formato arbitrário.A Tabela 20 compara os valores retornados pelo índice de Dunn para todos os algoritmos.

Tabela20: Avaliação dos conjuntos Banana1 e Banana2 (DO AUTOR, 2016).

Índice de Dunn	K-means	K-medoids	FCM	Aglomerativo	DBSCAN	SOM	PGC/APF
Banana1 com *K*=2	0.0252	0.0420	0.0252	**0.2307**	**0.2307**	0.0477	0.0285
Banana 1 com *K*=6	0.0465	0.0248	0.0239	**0.0700**	0.0513	0.0514	0.0114
Banana 2 com *K*=2	0.0042	0.0054	0.0088	0.0127	**0.0246**	0.0084	0.0044
Banana 2 com *K*=6	0.0086	0.0049	0.0064	**0.0166**	0.0111	0.0111	0.0042

5.1.4 Avaliação dos resultados

A correlação entre a matriz de proximidade, que mede a estrutura do agrupamento (em termos de coesão e isolamento) e a matriz de incidência que em suas colunas e linhas apresenta cada elemento do conjunto de dados, cada cruzamento de linha com coluna em que os dois elementos façam parte do mesmo *cluster*, é atribuído o valor “1”, do contrário “0”.

Tabela21: Correlação entre matriz de proximidade e matriz de incidência (DO AUTOR, 2016).

Correlação	K-means	K-medoids	C-means	Aglomerativo	DBSCAN	SOM	PGC/APF
ES1	**0.8644**	**0.8644**	**0.8644**	**0.8644**	**0.8644**	**0.8644**	**0.8644**
ES2	0.7852	0.7852	0.7855	0.7851	**0.7947**	0.7827	0.7736
Banana1 com *K*=2	0.5388	0.5215	0.5388	0.4269	0.4269	0.5113	**0.5403**
Banana1 com *K*=6	**0.6127**	0.6120	0.6104	0.5419	0.4974	0.6035	0.5501
Banana2 com *K*=2	0.5441	0.5289	0.5473	**0.5933**	0.4771	0.4559	0.5223
Banana2 com *K*=6	0.5502	0.5515	**0.5516**	0.5125	0.4299	0.5432	0.4839

De acordo com a Tabela 21, é possível observar que os conjuntos de dados que formam *clusters* hiperesféricos, ES1 e ES2 apresentam melhor resultado para a correlação. Esta característica é resultado da definição do cálculo da matriz de similaridade, que utiliza a distância de cada dado ao centro do *cluster*.

5.1.5 Estratégia de agregação de *clusters*com SVM

O algoritmo SVM será utilizado paraagregar os *clusters* resultantesdos algoritmos que tratam somente dados hiperesféricos, como K-means, K-medoids, Fuzzy C-means, Kohonen e PGC-APF.

Para esta avaliação serão utilizados somente os conjuntos de dados Banana 1 e Banana2. Na análise anterior, é possível verificar que o algoritmo DBSCAN, que trata

formato arbitrário, apresenta um melhor retorno para o índice Dunn que avalia com maior qualidade este tipo de formato. O algoritmo Hierárquico Aglomerativo também apresenta resultado competitivo.

Os outros índices avaliam com mais qualidade *clusters* de formato hiperesférico, pois consideram no cálculo a distância para o centro do *cluster*.

Assim, de forma a solucionar a limitação de algoritmos que somente tratam dados hiperesféricos, será realizada neste item a etapa de agregação de *clusters* e o índice de Dunn será usado para realizar a comparação no resultado dos algoritmos.

O processo é iniciado com a definição de um valor alto para a quantidade de *clusters*do algoritmo de clusterização em que este é um parâmetro de entrada.Para o conjunto de dados Banana 1 foram definidos 10 *clusters* e para Banana 2 foram definidos 50 *clusters* como parâmetro de entrada do algoritmo de clusterização.

Será atribuição do algoritmo SVM realizar a agregação destes *clusters* para 2 com o objetivo de formar *clusters* de formato arbitrário.

O algoritmo Hierárquico Aglomerativo não será considerado nesta etapa, poisa técnica escolhida para tratar os dados trata *clusters* não hiperesféricos e retornou resultados competitivos ao algoritmo DBSCAN.

A Tabela22apresenta os dois melhores resultados obtidos pelos algoritmos de clusterização para oíndice de validação de Dunn retirados da Tabela 14.

Tabela22: Índice de Dunn para Banana (DO AUTOR, 2016).

Banana1	Aglomerativo	DBSCAN
Dunn	**0.2307**	**0.2307**

Na Tabela 23 é realizada a comparação dos algoritmos que não obtiveram um bom resultado para o índice de Dunn baseado na Tabela 14 com o valor do índice de Dunn obtido pela etapa de agregação utilizando o algoritmo SVM.

Tabela23: Comparação de índice de Dunn com SVM para Banana 1 (DO AUTOR, 2016).

Banana 1	K-means	K-medoids	FCM	SOM	PGC/APF
Dunn sem SVM	0.0252	0.0420	0.0252	**0.0477**	0.0285
Dunn com SVM	0.2307	0.2307	0.2307	0.2307	**0.2307**

Para o conjunto de dados Banana 1 é possível verificar que o processo de agregação realizado pelo algoritmo SVM (linha 3) obteve os mesmos resultados que os algoritmos que tratam *clusters* de formato arbitrário, concluindo que esta etapa sucedeu em gerar *clusters* de formato arbitrário quando o algoritmo de clusterização final só gera *clusters* hiperesféricos.

A Tabela 24 apresenta os dois melhores resultados obtidos pelos algoritmos de clusterização para o índice de validação de Dunn retirados da Tabela 16.

Tabela24: Índice de Dunn para Banana 2 (DO AUTOR, 2016).

Banana2	Aglomerativo	DBSCAN
Dunn	0.0127	**0.0246**

Na Tabela 25 é realizada a comparação dos algoritmos que não obtiveram um bom resultado para o índice de Dunn baseado na Tabela 16 com o valor do índice de Dunn obtido pela etapa de agregação utilizando o algoritmo SVM.

Tabela25: Comparação de índice de Dunn com SVM para Banana 2 (DO AUTOR, 2016).

Banana 2	K-means	K-medoids	FCM	SOM	PGC/APF
Dunn sem SVM	0.0042	**0.0088**	0.0054	0.0084	0.0044
Dunn com SVM	0.0067	0.0090	0.0068	**0.0104**	0.0052

Para o conjunto de dados Banana 2 é possível verificar que o processo de agregação realizado pelo algoritmo SVM melhorou os resultados obtidos anteriormente, porém não alcançou o resultado obtido pelo algoritmo DBSCAN.

Os dois conjuntos de dados estão mostrados no Apêndice A.

5.1.6 Avaliação estatística

Quando os dados de *k* amostras correspondentes se apresentam pelo menos em escala ordinal, o teste de Friedman é útil para comprovar a hipótese de nulidade, de que as *k* amostras tenham sido extraídas da mesma população (FRIEDMAN; RUBIN, 1967).

Foi considerado o índice de validação de Dunn para os conjuntos na Tabela 26 para o teste de Friedman. A comparação entre todos os algoritmos retornou o valor zero, que demonstra que não existe diferença estatisticamente significativa entre os resultados obtidos pelos métodos estudados.

Tabela26: Teste estatísticode Friedman (DO AUTOR, 2016).

Diff	K-means	K-medoids	C-means	Aglomerativo	DBSCAN	SOM	PGC/APF
ES1	0	0	0	0	0	0	0
ES2	0	0	0	0	0	0	0
Banana 1 - SVM	0	0	0	0	0	0	0
Banana 2 - SVM	0	0	0	0	0	0	0

5.1.7 Comparação com dados aleatórios

Para esta etapa serãocomparados os valores dos índices de validação obtidos pelo método proposto com os dois algoritmos que obtiveram o melhor resultado utilizando o processo de agregação de *clusters*, Hierárquico Aglomerativo e SOM com os obtidos com dados aleatórios.

Para a geração dos dados aleatórios foram utilizados os mesmos parâmetros de entrada dos conjuntos Banana 1 e Banana 2.

O DBSCAN obteve os melhores resultados para os dois conjuntos, porém não foi utilizado como comparação nesta etapa, pois este classifica todos os dados como ruído, não obtendo qualquer valor dos índices de validação.

Serãogeradas trinta vezes cada dado aleatórioe calculados a média e desvio padrão para cada algoritmo e avaliar se o resultado obtido com os algoritmosestudados é coerente e não um resultado do acaso.

Nas Tabelas 27 a 30, é possível avaliar que os resultados retornados por dados aleatórios não são comparáveis aos dados, para os conjuntos de Banana 1 e Banana 2, o índice de Dunn que melhor representa a comparação dos dados.

Tabela27: Comparação de dados aleatórios com conjunto de dados ES1 (DO AUTOR, 2016).

400 pontos x ES1	Aglomerativo Dados aleatórios	Aglomerativo ES1	SOM Dados aleatórios	SOM ES1	PGC/APF Dados aleatórios	PGC/APF ES1
Silhouette	0.4549 ± 10^{-2}	0.9981 ± 0	0.4533 ± 10^{-2}	0.9981 ± 0	-0.060 ± 10^{-2}	0.9981 ± 0
CH	319.42 ± 41	92613 ± 0	326.04 ± 31	92613 ± 0	0.8950 ± 10^{-1}	92613 ± 0
DB	0.9582 ± 10^{-2}	0.0540 ± 0	1.0516 ± 10^{-1}	0.0540 ± 0	25.396 ± 15	0.0540 ± 0
Dunn	0.0429 ± 10^{-3}	5.7062 ± 0	0.0190 ± 10^{-3}	5.7062 ± 0	0.0010 ± 10^{-4}	5.7062 ± 0

Tabela28: Comparação de dados aleatórios com conjunto de dados ES2 (DO AUTOR, 2016).

400 pontos x ES2	Aglomerativo Dados aleatórios	Aglomerativo ES2	SOM Dados aleatórios	SOM ES2	PGC/APF Dados aleatórios	PGC/APF ES2
Silhouette	0.4549 ± 10^{-2}	0.8896 ± 0	0.453 ± 10^{-2}	0.882 ± 10^{-1}	-0.060 ± 10^{-2}	0.888 ± 10^{-3}
CH	319.42 ± 41	2124.5 ± 0	326.04 ± 31	2104.1 ± 725	0.8950 ± 10^{-1}	2120.0 ± 33
Média DB	0.9582 ± 10^{-1}	0.3887 ± 0	1.051 ± 10^{-1}	0.389 ± 10^{-1}	25.396 ± 15	0.389 ± 10^{-3}
Dunn	0.0429 ± 10^{-2}	0.0980 ± 0	0.019 ± 10^{-3}	0.084 ± 10^{-2}	0.0010 ± 10^{-4}	0.071 ± 10^{-2}

Tabela29: Comparação de dados aleatórios com conjunto de dados Banana1 (DO AUTOR, 2016).

200 pontos x Banana1	Aglomerativo Dados aleatórios	Aglomerativo Banana1	SOM Dados aleatórios	SOM Banana 1 SVM	PGC/APF Dados aleatórios	PCG/APF Banana1
Silhouette	0.2563 ± 10^{-1}	0.3672 ± 0	0.45135 ± 10^{-2}	0.3672 ± 0	-0.0031 ± 10^{-2}	0.3672 ± 0
CH	3.6656 ± 12	110.94 ± 0	97.106 ± 11	110.94 ± 0	0.7279 ± 1	110.94 ± 0
DB	0.6463 ± 2	1.2856 ± 0	1.2221 ± 10^{-1}	1.2856 ± 0	16.094 ± 10	1.2856 ± 0
Dunn	0.0875 ± 10^{-1}	0.2307 ± 0	0.0327 ± 10^{-1}	0.2307 ± 0	0.0030 ± 10^{-3}	0.2307 ± 0

Tabela30: Comparação de dados aleatórios com conjunto de dados Banana2 (DO AUTOR, 2016).

2000 pontos x Banana1	Aglomerativo Dados aleatórios	Aglomerativo Banana2	SOM Dados aleatórios	SOM Banana 2 SVM	PGC/APF Dados aleatórios	PCG/APF Banana2
Silhouette	0.4503 ± 10^{-1}	0.5952 ± 0	0.4346 ± 10^{-2}	0.382 ± 10^{-1}	-0.0003 ± 10^{-1}	0.461 ± 10^{-2}
CH	981.22 ± 172	2133.8 ± 0	944.14 ± 143	1053.5 ± 561	0.6857 ± 1	$1122,0\pm98$
DB	1.2784 ± 10^{-1}	0.8782 ± 0	1.2380 ± 10^{-1}	1.272 ± 10^{-1}	50.869 ± 272	1.047 ± 10^{-3}
Dunn	0.0061 ± 10^{-2}	0.0127 ± 0	0.0044 ± 10^{-3}	0.010 ± 10^{-3}	0.0004 ± 10^{-1}	0.005 ± 10^{-3}

5.2 Interpretação de resultados

Serão utilizados os conjuntos Iris e *Breast cancer* para avaliar a interpretação de resultados. Estes conjuntos de dados foram retirados da base UCI *Machine Learning*e são utilizados para estudos de caso em artigos científicos.

Para o algoritmo Hierárquico Aglomerativo, os parâmetros de entrada na Tabela 31apresentaram o melhor resultado para cada conjunto de dados.

Tabela31: Parâmetros de entrada para Aglomerativo (DO AUTOR, 2016).

Nome	Técnica
Iris	*2.1.5* *Ward*
Breast cancer	***2.1.6*** *Ward*

Para o algoritmo DBSCAN, os parâmetros de entrada *Eps* e *MinPts* para cada conjunto de teste estão dispostos na Tabela 32.

Tabela32: Parâmetros de entrada para DBSCAN (DO AUTOR, 2016).

Nome	Eps	MinPts
Iris	0.6	3
Breast cancer	6	30

5.2.1 Avaliação da acurária

O cálculo da acurácia dos algoritmos de clusterização foi realizado com base na classificação dos conjuntos de dados que está disponibilizada no *UCI Machine LearningRepository*.

Para os dados *Breast cancer*, 119 dados foram considerados ruído para o algoritmo DBSCAN.

A Tabela 33 mede a acurácia, proximidade entre o valor obtido experimentalmente pelos métodos e o valor verdadeiro provido pelo responsável pelo banco de dados.

Na análise é possível avaliar que todos os algoritmos estão competitivos. Apesar do DBSCAN apresentar um melhor resultado, 17% dos dados não são avaliados por serem considerados ruídos.

Cada experimento foi executado trinta vezes para cada algoritmo e conjunto de dados e calculada a média e o desvio padrão para cada índice de validação, os resultados obtidos serão preenchidos nas tabelas no formato “média ± desvio padrão”.

Tabela33: Avaliação de acurácia para conjunto de dados *breast cancer* (DO AUTOR, 2016).

Algoritmo	Acurácia (%)
K-means	95.454 ±0.14
K-medoids	94.721±0.00
C-means	95.014±0.00
Aglomerativo	95.747±0.00
DBSCAN	**98.046**±0.00
SOM	96.627±0.82
PCG/APF	95.014±0.92

Para o conjunto de dados *iris*, 4dados foram considerados ruído. Na Tabela 34 é avaliada a acurácia média de cada método. O método DBSCAN não identificou corretamente um dos *clusters*.

Tabela34: Avaliação de acurácia para conjunto de dados *Iris* (DO AUTOR, 2016).

Algoritmo	Acurácia (%)
K-means	82.666±0.64
K-medoids	83.333±0.00
C-means	84.000±0.00
Aglomerativo	82.666±0.00
DBSCAN	70.2177±0.00
SOM	88.666±4.65
PCG/APF	**89.333**±0.00

5.2.2 Interpretação das árvores de padrões *fuzzy*

Para os dados *Breast Cancer* as árvores geradas são apresentadas conforme Figuras 46 e 47.

Para este conjunto, a classificação retorna se o diagnóstico do tumor se é maligno ou benigno. Os valores dos atributos são retirados a partir da análise de um exame de imagem realizadoatravés da coleta de material do paciente.

Na Figura 46, pode ser avaliada que a partição ou termobaixo do atributo compactação(A6L)e a partiçãomédio-alta da variação de comprimento(A5HM) influenciam de forma significativa a determinação do tipo de tumor maligno.

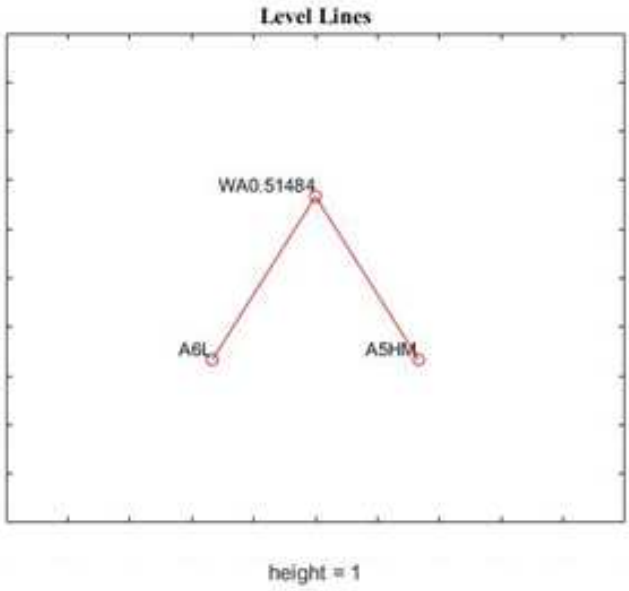


Figura 46: Árvore do *cluster* 1, paciente com tumor maligno (DO AUTOR, 2016).

Na Figura47, pode ser avaliado que valores baixos tanto do atributo textura (A2L) e perímetro (A3L) determinam que o tipo de tumor do paciente avaliado é benigno.

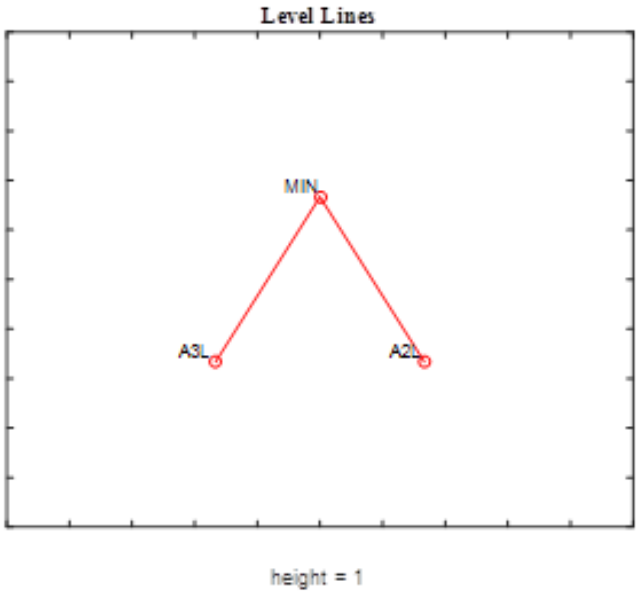


Figura 47: Árvore do *cluster* 2, paciente com tumor benigno (DO AUTOR, 2016).

Para os dados iris,cujas árvores geradas são apresentadas conforme Figuras 48 a 50, o objetivo é classificar corretamente a flor produzida por uma planta.

Para o primeiro *cluster*, é possível observar que valores na partição médio do atributo dois (A2M), largura da sépala e alto do atributo três (A3L), comprimento da pétala determinam o grupo Iris setosa.

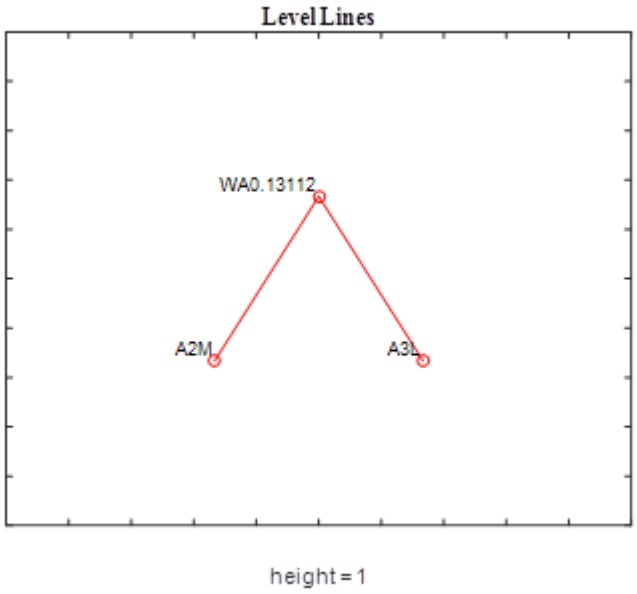


Figura 48:Árvore do *cluster* 1, Iris setosa (DO AUTOR, 2016).

Para o segundo *cluster*, é possível observar que valores médio-baixo do atributo um e dois, respectivamente comprimento e largura da sépala e médio do comprimento da pétala determinam o grupo Iris virginica.

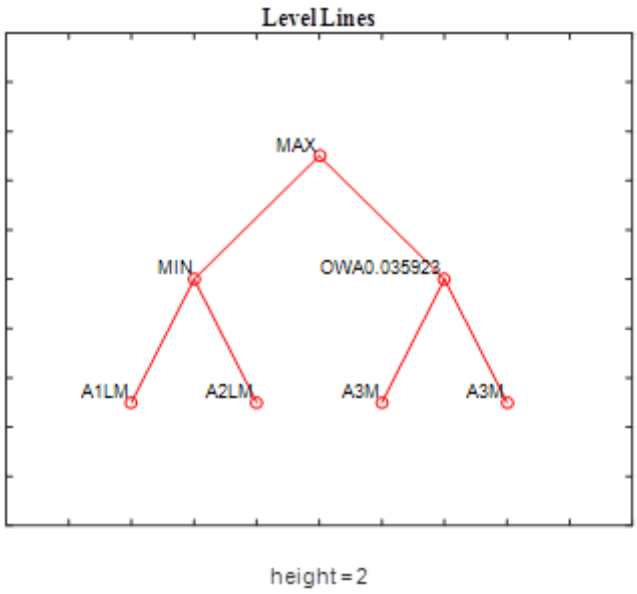


Figura 49: Árvore do *cluster* 2, Iris virginica (DO AUTOR, 2016).

Para o terceiro *cluster*, é possível observar que valores da partição médio-alto do atributo um, comprimento da sépala e da partição alto do atributo comprimento da pétala(A3H) determinam a grupo Iris versicolor.

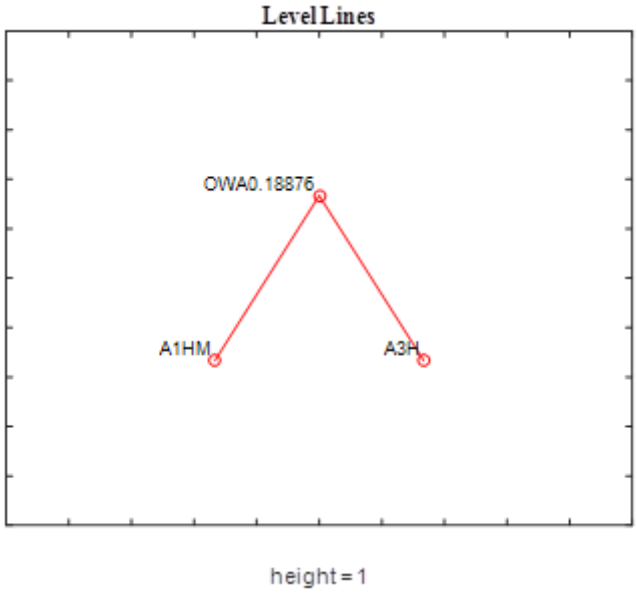


Figura 50: Árvore do *cluster* 3, Iris versicolor (DO AUTOR, 2016).

5.2.3 Avaliação dos termos

Nesta fase serão avaliados quais termos mais se apresentaram nas árvores obtidas em 30 repetições da clusterização realizada e qual o grau de influência na árvore (altura do termo em relação ao tamanho da árvore).

Para o conjunto de dados *Breast Cancer*, os resultados são apresentados na Tabela 35, em que é possível observar pelo exemplo das Figuras 46 e 47que os termos 3L, 2L, 6L e 5HM aparecem com frequência no processo de síntese das árvores sendo um indicativo de sua importância.

Tabela35: Avaliação de altura da árvore e importância dos termos do conjunto *Breast Cancer* (DO AUTOR, 2016).

Atributo	L		LM		M		HM		H	
Altura	0	1	0	1	0	1	0	1	0	1
1 - Raio	3	0	2	0	0	0	1	0	3	0
2 - Textura	**6**	1	2	0	1	0	3	0	3	0
3 - Perímetro	**19**	0	0	0	1	0	4	1	0	0
4 - Área	4	0	4	0	2	0	1	2	1	0
5 - Variação de comprimento	4	0	0	0	0	1	3	0	1	1
6 - Compactação	**9**	2	9	0	0	0	1	0	3	0
7 - Concavidade	3	0	0	0	3	0	0	0	1	0
8 - Número de porções de concavidade	**5**	1	0	0	1	0	**5**	0	0	0
9 - Simetria	2	0	0	0	1	0	**3**	0	1	0

Para o conjunto de dados iris, os resultados são apresentados na Tabela 36, em que é possível observar pelo exemplo das Figuras 48 a 50 que os termos 2M, 3L, 1LM, 2LM, 3M, 1HM e 3Haparecem com frequência no processo de síntese, o que é indicativo da sua importância.

Tabela36: Avaliação de altura da árvore e importância dos termos do conjunto Iris (DO AUTOR, 2016).

Atributo	L				LM				M				HM				H			
Altura	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
1 - Comprimento da sépala	**6**	2	0	0	**6**	1	1	0	**7**	0	0	1	9	1	0	0	**12**	0	1	0
2 - Largura da sépala	**6**	1	2	0	6	1	0	0	6	1	0	0	5	2	0	0	**7**	2	0	0
3 - Comprimento da pétala	**20**	**8**	3	1	**12**	4	0	0	**10**	2	0	0	**24**	7	0	0	**12**	1	0	0
4 - Largura da pétala	**12**	4	1	0	7	0	0	0	5	2	0	1	**14**	2	0	0	5	1	0	1

5.3Análise de segmentação de mercado

A base de utilização do serviço de mensagens curtas (do inglês, *Short Message Service*, SMS) será usada para aplicação do método proposto adicionalmente à comparação com as bases conhecidas. Este é um serviço provido por operadoras de telefonia móvel.

Atualmente, além da possibilidade de envio entre clientes de mesma operadora ou operadores diferentes (interconexão), é possível para o cliente contratar serviços de envio de notícias, entretenimento, alertas de transações bancárias, adquirir crédito para telefone para uso de serviços, contratar pacotes de serviços, acessar canais de atendimento da sua Operadora. No caso de um cliente corporativo, as empresas podem se comunicar com seus clientes para promoções, informar débitos, informações de *check-in* de voos, etc.

Um dos grandes benefícios da utilização de SMS por parte da comunicação de empresas é não necessitar que o cliente tenha pacote de dados contratado e ser suportado por quase a totalidade dos aparelhos no mercado.

A SMSC (*Short Message Service Center*)é a plataforma utilizada pelas Operadoras para prestar este serviço e funciona no modelo *store and forward* que consiste de uma fila que recebe os pedidos de envio de SMS. A SMSC realiza a primeira tentativa de entrega da mensagem para o telefone ou para a aplicação, se o destino estiver disponível, envia a mensagem. Do contrário, coloca na fila para tentar posteriormente. Possui as seguintes funcionalidades básicas:

- Um prazo de armazenamento dos SMS, com mecanismo inteligente de reenvio, ou seja, por tipo de destino pode ser definida quantas vezes a haverá a tentativa de reenviar a mensagem e o espaçamento de tempo entre cada tentativa. No Brasil, o órgão regulador (Agência Nacional de Telecomunicações, ANATEL) determina que o prazo limite para tarifar um cliente é de 24 horas;
- A possibilidade de priorizar as mensagens na fila de entrega, ou seja, serviços prioritários podem ser entregues antes para o destino quando estes estiverem na mesma fila de envio;
- A notificação de entrega tanto para o assinante do móvel, se este origina a mensagem quanto para a aplicação que enviou o conteúdo. A notificação de entrega informa se a mensagem foi enviada com sucesso ou não para o destino.

Na Figura 51é apresentada a arquitetura de rede. A SMSC é um elemento centralizador para as mensagens de texto de todos os assinantes da rede.

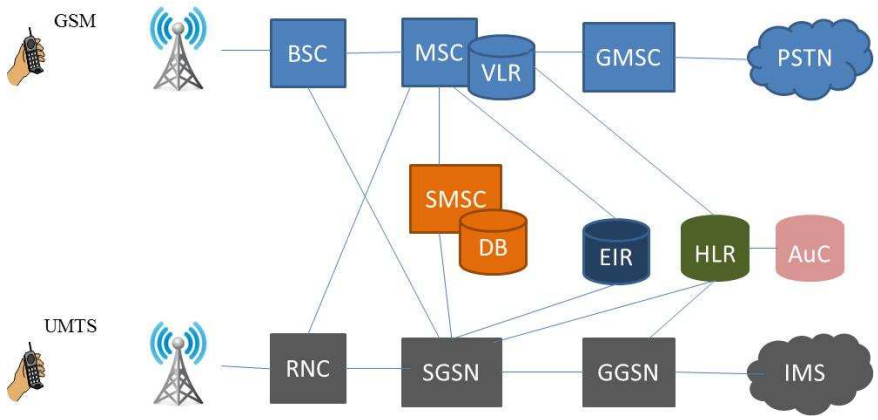


Figura51: Arquitetura geral de rede de telefonia móvel (DO AUTOR, 2017)

Para iniciar a análise, o primeiro passo é a compreensão de quais dados serão utilizados, no caso da segmentação por comportamento são considerados dados como, quantidades de mensagens, tipos de serviços utilizados, que serão explicados mais adiante neste item.

Já a determinação da população a ser segmentada será baseada em clientes ativos e que estatisticamente seja representativa, ou seja, a amostra da qual a análise será realizada oferece conclusões válidas sobre a população.

O nível de segmentação serágeográfico e por comportamento, ou seja, serão coletados dados de três regiões do Brasil para compor a massa a ser analisada e será avaliada a diferença de comportamento de utilização de serviços nestas regiões. Esta coleta será realizada no elemento SMSC,pois ele centralizada as informações de mensagens de texto originadas na rede de telefonia móvel e de aplicações.

Este conhecimento está contido na análise de registros dos assinantes (CDR, *call detail record),* que será utilizado na preparação dos dados.

Os CDRsgerados naSMSC tem como sua principal função registrar as informações necessárias para realizar a cobrança aos usuários pela utilização dos serviços oferecidos pela operadora e geração de estatísticas para o órgão regulador.

Estes arquivos de CDRs podem ser fechados por tempo ou por tamanho. Neste trabalho foram coletados arquivos e realizada a retirada de dados provenientes das regiões escolhidas.

As variáveis consideradas para o projeto abrangem os tipos de mensagens de texto que podem ser enviadas pelo cliente móvel para outro cliente ou para um serviço e as informações disponíveis nos CDRs são:

- Destinatário;
- Originador;
- Data e hora de submissão;
- Data e hora da entrega da mensagem;
- Tamanho do texto inserido pelo cliente em bytes;
- Número de tentativas de entrega (é maior que “uma”, se há falha na primeira tentativa);
- MSC (*Mobile Switching Center*), equipamento responsável por realizar as conexões dos assinantes móveis a outros assinantes, de origem e de destino. Se a origem ou o destino forem um assinante móvel esta informação é preenchida;
- Mensagem entregue com sucesso, insucesso ou expirada. A mensagem expirada é caracterizada pelo seguinte fluxo: é iniciada a política de reenvio, porém alcançada às 24 horas, ela não é entregue para o assinante;
- *International Mobile Number Subscriber Identity* (IMSI) tanto do originador quanto do destinatário, quando este forem um assinante móvel. Esta

informação é uma identidade única do *chip* e identifica adicionalmente o país e a operadora;

- Tipo de mensagem (originada ou terminada na rede e recibo de entrega).

É importante destacar que existem dois tipos de falhas, a primeira é a falha definitiva que não inicia a política de reenvio, um dos exemplos é quando uma mensagem de texto é encaminhada para a rede de uma operadora, porém aquele destino não é um telefone da mesma e sim de outra empresa e a segunda é falha temporária, em que o assinante pode estar indisponível, pois o telefone está desligado, por exemplo.

O fluxo simplificado da mensagem de texto originada por um assinante móvel é descrito na Figura 52, onde o assinante inicia o envio com o "*Message Transfer*" e a MSC encaminha esta mensagem para a SMSC utilizando protocolo especifico de rede, já a comunicação entre SMSC e a aplicação utiliza outro protocolo desenvolvido para comunicação entre o elemento e aplicações externas ou internas a operadora.

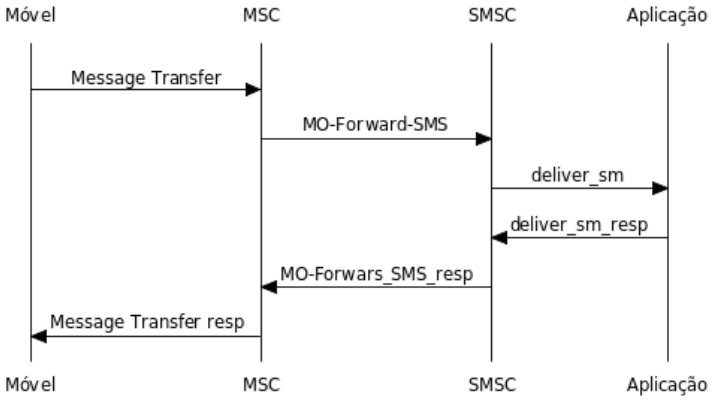


Figura52: Fluxo básico do SMS (DO AUTOR, 2016)

Os CDRs coletados a partir do servidor de uma operadora de telecomunicações, cuja identidade não será revelada nesse trabalho por questões de confidencialidade dos dados fornecidos, foram tratados por um especialista.

Os dados foram carregados em um banco de dados auxiliar (Access) para pré-processamento "manual" para limpeza na base para a retirada de dados que apresentavam campos vazios ou inconsistências, pois embora alguns algoritmos retirem o ruído, como o DBSCAN, alguns procedimentos para lidar com dados incompletos ou ruidosos que nem sempre são robustos, necessitando esta etapa manual.

Esta retirada foi baseada no originador, ou seja, se o cliente que originou a mensagem for da região 1, esta será considerada para análise desta região.

Para filtrar somente as mensagens originadas por assinantes da operadora foi utilizado o campo tipo de mensagem.

Os campos que se seguem foram escolhidos pela significância para análise da segmentação por comportamento:

- Tempo de entrega da mensagem, este atributo é a diferença entre data e hora da entrega da mensagem e data e hora de submissão;
- Destinatário;
- Originador;
- MSC (*Mobile Switching Center*) do originador;
- Tamanho do texto inserido pelo cliente em bytes.

O destino será consolidado por tipo de destinatário, conforme se segue para analisar que tipos de serviços são de maior interesse na região estudada:

- Mensagens destinadas para clientes da mesma operadora;
- Mensagens destinadas para clientes de outra operadora.

Será extraída a informação se a origem e o destino estão no mesmo Código de área ou não.

Se a MSC não pertencer ao mesmo código de área do originador, significa que a mensagem de texto foi originada por um assinante em *roaming*.

Neste trabalho, a avaliação dos resultados dos *clusters* dará a visão de como clientes de diferentes regiões possuem perfis específicos em relação à utilização de serviço. O objetivo é atingir a qualidade do serviço que é o grau em que um serviço atende ou supera as expectativas do cliente. Se o cliente recebe um serviço acima de sua expectativa, isto é, melhor do que o esperado, estará com um grau de satisfação elevado. Se o serviço recebido for percebido abaixo do esperado, o grau de satisfação será baixo. A qualidade é julgada de acordo com a satisfação percebida e é descobrir o que gera valor para o cliente uma ação fundamental para oferecer serviços mais indicados conforme o perfil do usuário.

Deve-se encontrar uma maneira de fornecer um serviço que atinja os objetivos do cliente (resolva seus problemas), de modo que seus valores fiquem evidenciados. Isto faz com que cada serviço prestado seja individualizado, isto é, tenha a personalidade do consumidor.

O interesse do receptor em receber a mensagem é um dos fatores cruciais para o sucesso da comunicação. Como a capacidade do cérebro para processar informações é

limitada, os consumidores são muito seletivos quanto ao que dedicar sua atenção. O processo de percepção seletiva significa que as pessoas atendem a somente uma pequena porção dos estímulos a que são expostas.

Isto faz com que cada serviço prestado possa ser diferente por região, trazendo melhor resultado para a operadora e maior satisfação para o cliente.

5.3.1 Avaliação dos dados com PGC-APF

A Tabela 37 informa a quantidade de pontos em cada conjunto de dados, cada conjunto contém informações de tráfego em cada região.

Tabela 37: Conjunto de dados de segmentação (DO AUTOR, 2016).

Região	Pontos
A	680
B	445

A Tabela 38 resume cada possível cenário de tráfego quando um cliente origina uma mensagem de texto.

Tabela 38: Avaliação de cenários (DO AUTOR, 2016).

Envio entre clientes da mesma região	Envio entre clientes da mesma operadora	Cliente de origem na mesma localização
Sim	Não	Não
Sim	Sim	Não
Sim	Não	Sim
Sim	Sim	Sim
Não	Não	Não
Não	Sim	Não
Não	Não	Sim
Não	Sim	Sim

Será avaliado para cada região o cenário de maior tráfego para cada conjunto de dados, no caso da região A. Este cenário é quando clientes de diferentes regiões trocam mensagens de texto e não são da mesma operadora. Adicionalmente, os clientes que originam esta mensagem não estão em *roaming*.

Para região B, o cenário ocorre quando os clientes são da mesma região e operadora, além do originador não estar em *roaming*.

5.3.2 Interpretação das árvores de padrões *fuzzy*

A proposta é gerar uma monitoração que avalie o comportamento do tráfego de assinante em uma região ao longo de cada hora por dia para que seja possível avaliar características do tráfego tanto para operação quanto para *marketing*.

Para a região A foram geradas cinco árvores da Figura 53 a 57, este valor foi o retornando pelo índice de validação Calinski-Harabasz para a melhor disposição de *clusters*, foi definido o número máxima de árvore de dez como entrada.

Na Figura 53, as mensagens são especificamente caracterizadas pelo atributo dois (tamanho da mensagem), nas partições alto e médio-alto (A2H e A2HM), devido a esta característica antes da mensagem ser entregue para o assinante de destino, essa é enviada para avaliação de características de *spam*, podendo ou não ser bloqueada.

Neste fluxo é avaliado o comportamento do originador, como quantidade de mensagens enviadas em uma faixa específica do tanto e se o texto da mensagem se repete com frequência, por exemplo.

Esta avaliação em relação ao tamanho da mensagem relacionada ao comportamento de *spam* é devido a um estudo de uma comunidade internacional que avalia este comportamento em todo o mundo e direciona as operadoras em relação a algumas características a serem monitoradas.

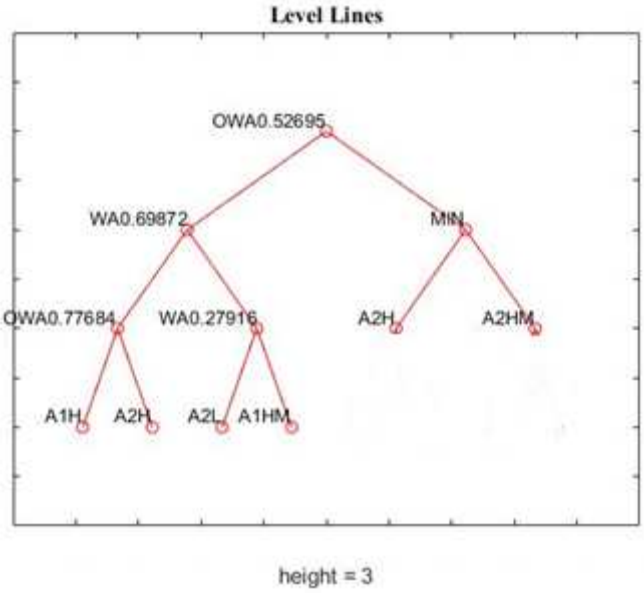


Figura 53:Árvore do *cluster* 1, será avaliado pelo anti-spam (DO AUTOR, 2016).

Na Figura 54, o *cluster* 2 é caracterizado por mensagens com alta quantidade de caracteres (A2HM) no campo texto, porém com quantidade de caracteres menor que o *cluster* 1.

Este *cluster* pode ser agregado pelo algoritmo SVM com o *cluster* 1, pois o tamanho do texto continua maior que o valor de controle em que a mensagem é considerada *spam*.

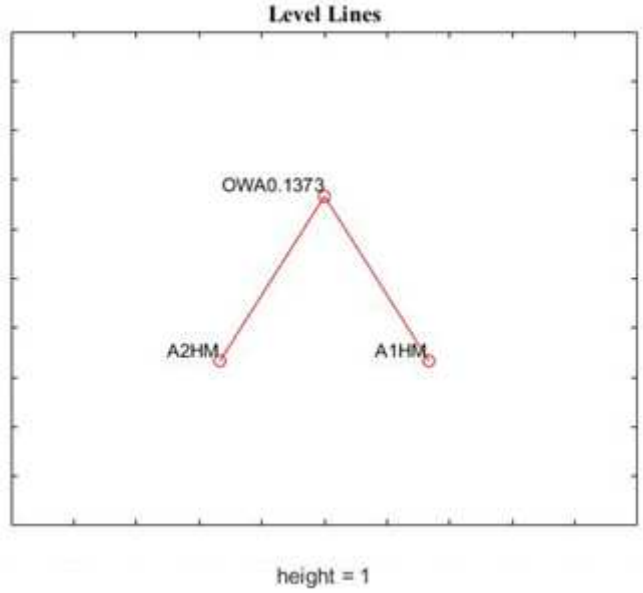


Figura 54: Árvore do *cluster* 2, será avaliado pelo anti-spam (DO AUTOR, 2016).

O *cluster* 3 na Figura 55 é caracterizado por mensagens com a quantidade de caracteres de mensagem menor (A2LM e A2M) que o limite de indicador de *spam*. Assim, estas mensagens serão consideradas válidas.

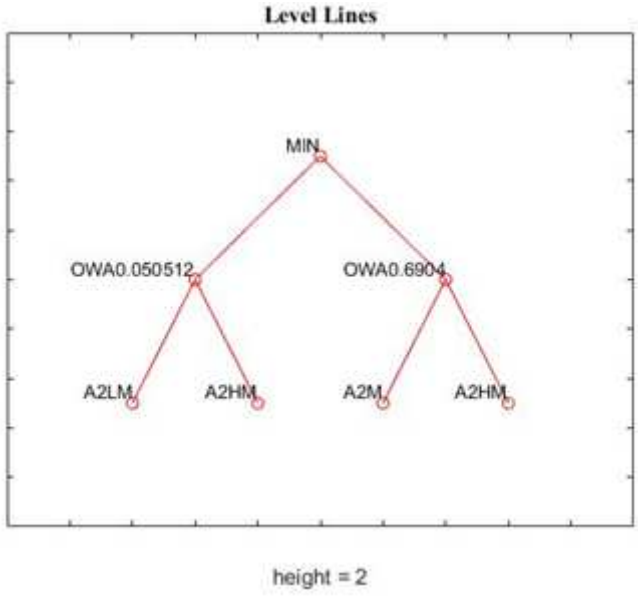


Figura 55: Árvore do *cluster*3, caracterizado por mensagens válidas (DO AUTOR, 2016).

O *cluster* 4 na Figura 56 é caracterizado por mensagens de texto com a variável de tempo de entrega com valores nas partições alto e médio (A1H e A1M). Os valores para o atributo um indicam que o assinante de destino não estava disponível durante todo o tempo que a plataforma de SMS tentou entregar a mensagem de texto.

Esta é uma indicação ruim tanto para operação, pois não atende ao tempo medido de entrega de 60 segundos da Anatel, causando impacto no indicador de desempenho de rede, quanto é mais custoso, pois gasta mais recursos de rede, pois são realizadas muitas tentativas de entrega para uma mesma mensagem.

Esta característica de mensagem com alta duração na tentativa de entrega e quantidade alta de caracteres é uma indicação que esta mensagem pode estar sendo originada por um *spammer*, que envia mensagens para toda a faixa numérica disponível da operadora, onde não necessariamente o cliente de destino está disponível.

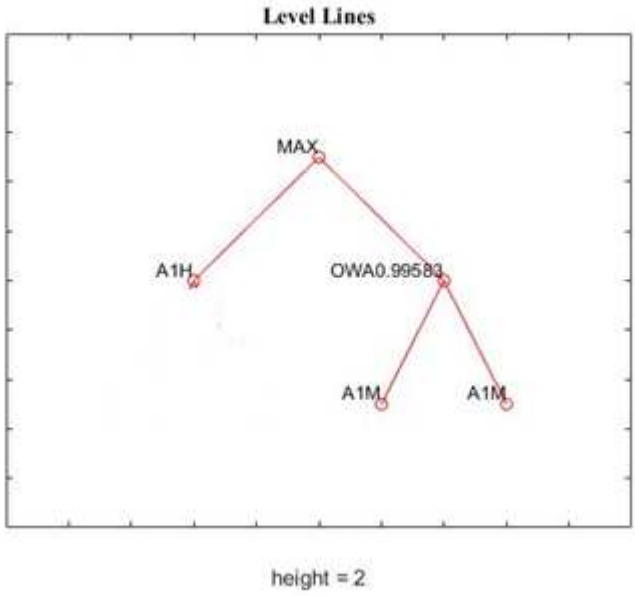


Figura 56: Árvore do *cluster* 4, caracterizado por *spam* (DO AUTOR, 2016).

O *cluster* 5 na Figura 57 é caracterizado por mensagens com a quantidade de caracteres de mensagem menor que o limite de indicador de *spam* (A2L). Assim, estas mensagens serão consideradas válidas.

A diferença deste *cluster* para o 3 é que o tamanho de mensagem indica que a mensagem trocada possa ser um diálogo curto, como: "Sim" ou "Já vou".

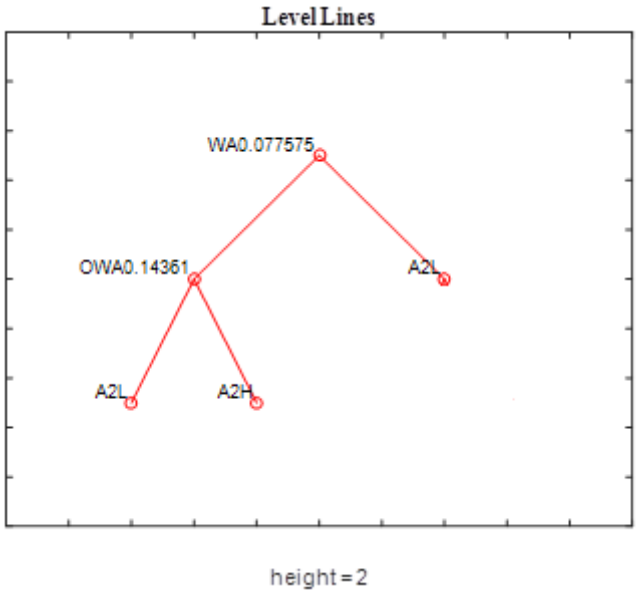


Figura 57:Árvore do *cluster* 5, caracterizado por mensagens válidas curtas (DO AUTOR, 2016).

Para a região B, as Figuras 58 a 60 apresentam as três árvores geradas, este valor foi o retornando pelo índice de validação Calinski-Harabasz para a melhor disposição de *clusters*, foi definido o número máxima de árvore de dez como entrada.

O*cluster* 1 na Figura 58 é caracterizado pela variável de tempo de entrega com valores nas partições alto e médio (A1H e A1M). Como comentado para o *cluster* 4 na região anterior, é uma indicação que esta mensagem pode estar sendo originada por um *spammer*, que envia mensagens para toda a faixa numérica disponível da operadora, onde não necessariamente o cliente de destino está disponível, tendo grande impacto no operacional e nos indicadores de desempenho.

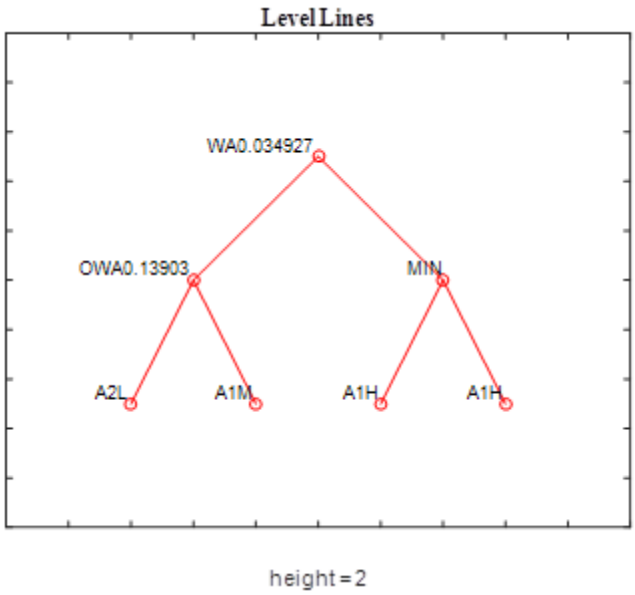


Figura 58:Árvore do *cluster* 1, caracterizado por *spam* (DO AUTOR, 2016).

O *cluster* 2 na Figura 59 é caracterizado com a quantidade de caracteres de mensagem menor que o limite de indicador de *spam*, atributo 2 baixo (A2L), com a maior parte das mensagens entregues em 60 segundos e poucas mensagens em até 5 minutos, este é um comportamento válido.

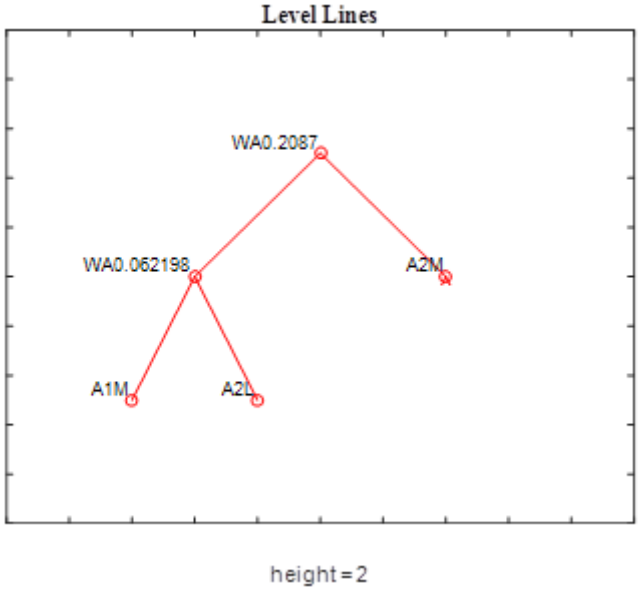


Figura 59: Árvore do *cluster* 2, caracterizado por mensagens válidas (DO AUTOR, 2016).

O *cluster* 3 na Figura 60 é comparável ao *cluster* 1 da região A, com os valores médio-alto e alto do atributo dois, quantidade de caracteres no campo texto, indicando que esta mensagen deve ser avaliada para processo de *spam* para identificar características adicionais para permitir ou não a entrega ao destinatário.

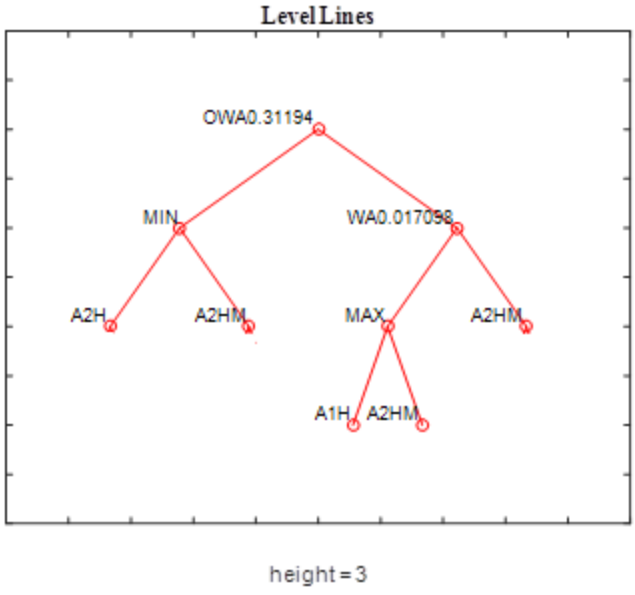


Figura 60: Árvore do *cluster* 3, será avaliado pelo anti-spam (DO AUTOR, 2016).

CONCLUSÃO

Este trabalho apresenta um modelo utilizando Árvores de Padrões Fuzzy e Programação Genética Cartesiana para solucionar um problema de clusterização.

Foram realizados estudos de casos e comparação com outros algoritmos tradicionais de clusterização para avaliar quais tipos de dados são melhores avaliados com este método e as vantagens em relação aos outros métodos.

Foi verificado que o método proposto apresenta desempenho similar em quase todos os tipos de dados e para solucionar a limitação de tratamento de clusters hiperesféricos foi utilizado o algoritmo SVM.

O algoritmo SVM sucedeu em agrupar conjunto de dados que formam *clusters* em formato arbitrário, especificamente para dados menos densos, como o conjunto Banana 1 atingiu o mesmo resultado que o algoritmo DBSCAN.

No caso de dados mais densos, como o conjunto de dados Banana 2, melhorou o resultado original obtido pelo algoritmo de clusterização.

É importante salientar a importância dos resultados obtidos pelas árvores para a interpretação dos dados contidos em um *cluster*, favorecendo a avaliação do especialista em relação ao grupo e tornando a análise menos subjetiva que nos algoritmos tradicionais.

As propostas para trabalhos futuros no desenvolvimento desta pesquisa estãolistadas a seguir:

- Avaliar a adoção de um índice de validação que trata conjuntos de dados que formam *clusters* de formato arbitrário. O índice de Dunn foi avaliado e não retornou bons resultados;
- O algoritmo DBSCAN poderá ser testado como tratamento inicial para os conjuntos de dados para a retirada de *outliers*antes da utilização de outras técnicas de clusterização;
- Avaliação do modelo multiobjetivo, que é a técnica onde diversos objetivos são levados em conta ao mesmo tempo para a obtenção da solução. Por exemplo, pode-se utilizar diferentes índices de validação ao mesmo tempo.

REFERÊNCIAS

AGGARWAL, Charu C.; YU, Philip S. **Finding generalized projected clusters in high dimensional spaces**. ACM, 2000.

AGRAWAL, K. P.; GARG, Sanjay; PATEL, Pinkal. Performance measures for densed and arbitrary shaped clusters. **Int. J. Comput. Sci. Commun**, v. 6, n. 2, p. 338-350, 2015.

ALVES, Vinícius S.; CAMPELLO, Ricardo JGB; HRUSCHKA, Eduardo R. Towards a fast evolutionary algorithm for clustering. In: **Evolutionary Computation, 2006. CEC 2006. IEEE Congress on**. IEEE, 2006. p. 1776-1783.

AMARAL, S. A. (2008). **Marketing da informação: entre a promoção e a comunicação integrada de marketing, Informação & Sociedade: Estudos 18(1).**

BABU, G. Phanendra; MURTY, M. Narasimha. Clustering with evolution strategies. **Pattern recognition**, v. 27, n. 2, p. 321-329, 1994.

ASSOCIATION, M. M. et al. (2006). **Mma annual mobile marketing guide: Recognizing leadership & innovation**.

BANDYOPADHYAY, Sanghamitra; MAULIK, Ujjwal. An evolutionary technique based on K-means algorithm for optimal clustering in RN.**Information Sciences**, v. 146, n. 1, p. 221-237, 2002.

BANZHAF, W. **Genotype-Phenotype-Mapping and Neutral Variation - A Case Study in Genetic Programming** Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature. Anais...: PPSN III.London, UK, UK:Springer-Verlag, 1994.

BATAINEH, K.; NAJI, M.; SAQER, M. A comparison study between various fuzzy clustering algorithms. **Editorial Board**, v. 5, n. 4, p. 335, 2011.

BERRY, M. J., & LINOFF, G. (1997). **Data mining techniques: for marketing, sales, and customer support**. John Wiley & Sons, Inc.

BEZDEK, James C. et al. Genetic algorithm guided clustering. In:**Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on**. IEEE, 1994. p. 34-39.

BILGIN, Gökhan; ERTÜRK, Sarp; YILDIRIM, Tülay. Segmentation of hyperspectral images via subtractive clustering and cluster validation using one-class support vector machines. **Geoscience and Remote Sensing, IEEE Transactions on**, v. 49, n. 8, p. 2936-2944, 2011.

BOCK, Hans H. Probabilistic models in cluster analysis. **Computational Statistics & Data Analysis**, v. 23, n. 1, p. 5-28, 1996

CAMILO, Cássio Oliveira; SILVA, João Carlos da. Mineração de dados: Conceitos, tarefas, métodos e ferramentas. **Universidade Federal de Goiás (UFC)**, p. 1-29, 2009.

CASILLAS, Arantza; DE LENA, MT González; MARTÍNEZ, R. Document clustering into an unknown number of clusters using a genetic algorithm. In:**Text, speech and dialogue**. Springer Berlin Heidelberg, 2003. p. 43-49.

CHAN, Chu-Chai Henry. Online auction customer segmentation using a neural network model. **International Journal of Applied Science and Engineering**, v. 3, n. 2, p. 101-109, 2005.

CHENG, Y.; CHURCH, G. M. Biclstering of Expression Data, in proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB). 2000

CHIU, Stephen L. Fuzzy model identification based on cluster estimation.**Journal of Intelligent & Fuzzy Systems**, v. 2, n. 3, p. 267-278, 1994.

CHRISTOPHER, L. and WRIGHT, L. (2001). **Serviços, marketing e gestão**, São Paulo: Saraiva .

CHIOU, Yu-Chiun; LAN, Lawrence W. Genetic clustering algorithms.**European journal of operational research**, v. 135, n. 2, p. 413-427, 2001.

CHRISTOPHER, Lovelock; WRIGHT, Louren. **Serviços, Marketing e Gestão**. São Paulo: Saraiva, 2001.

COLE, Rowena Marie. **Clustering with genetic algorithms**. University of Western Australia, 1998.

COSTA, José Alfredo Ferreira. Classificação automática e análise de dados por redes neurais auto-organizáveis. 1999.

COWGILL, Marcus Charles; HARVEY, Robert J.; WATSON, Layne T. A genetic algorithm approach to *cluster* analysis. **Computers & Mathematics with Applications**, v. 37, n. 7, p. 99-108, 1999.

CRISTOFOR, Dana; SIMOVICI, D. An information-theoretical approach to clustering categorical databases using genetic algorithms. In: **2nd SIAM ICDM, Workshop on clustering high dimensional data**. 2002.

DA SILVA, I. N., SPATTI, D. H. and FLAUZINO, R. A. (2010). **Redes neurais artificiais para engenharia e ciências aplicadas curso prático**, Artliber.

DE CASTRO, ARMANDO ANTONIO MONTEIRO; DO PRADO, PEDRO PAULO LEITE. Algoritmos para reconhecimento de padrões. **Revista Ciências Exatas**, v. 8, n. 2002, 2001.

DE FALCO, Ivanoe et al. An Innovative Approach to Genetic Programming—based Clustering. In: **Applied Soft Computing Technologies: The Challenge of Complexity**. Springer Berlin Heidelberg, 2006. p. 55-64.

DE JONG, Kenneth A. **Evolutionary computation: a unified approach**. MIT press, 2006.

DE OLIVEIRA, Rudinei Martins; LORENA, Antonio Nogueira; MAURI, Geraldo Regis. HEURÍSTICAS HÍBRIDAS PARA O PROBLEMA DE ALOCAÇÃO DE BERÇOS PARA NAVIOS E PARA UM PROBLEMA DE AGRUPAMENTOS.

DE OLIVEIRA, César S. et al. An evolutionary density and grid-based clustering algorithm. In: **Proc. XXIII Brazilian Symposium on Databases (SBBD-2007)**. 2007. p. 175-189

DHARWADKER, A.; PIRZADA, S. **Applications of Graph Theory**. [s.l.]CreateSpace Independent Publishing Platform, 2011.

DIAS, D. M. Programação Genética Linear com Inspiração Quântica. 2010.

DUARTE, Fernando Jorge Ferreira. **Optimização da Combinação de Agrupamentos baseado na Acumulação de Provas pesadas por Índices de Validação e com uso de Amostragem**. 2008. Tese de Doutorado. Universidade de Trás-os-Montes e Alto Douro.

EISEN, Michael B. et al. Cluster analysis and display of genome-wide expression patterns. **Proceedings of the National Academy of Sciences**, v. 95, n. 25, p. 14863-14868, 1998.

EREN, Kemal et al. A comparative analysis of biclustering algorithms for gene expression data. **Briefings in bioinformatics**, v. 14, n. 3, p. 279-292, 2013.

ESTER, Martin et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: **Kdd**. 1996. p. 226-231.

ESTIVILL-CASTRO, Vladimir; MURRAY, Alan T. **Spatial clustering for data mining with genetic algorithms**. Australia: Queensland University of Technology, 1997.

EVERITT, B. S. Cluster analysis. 1993. **Edward Arnold and Halsted Press,**, 1993.

FERREIRA, Fábio dos S.; DE CAMPOS, Gustavo AL; SILVA, Jackson S. de V. Algoritmo Genético para Clusterização Baseado nas Metodologias de Densidade e Grade.

FOGEL, Lawrence J. Autonomous automata. **Industrial research**, v. 4, n. 2, p. 14-19, 1962.

FRANEK, Lucas et al. Image segmentation fusion using general ensemble *cluster*ing methods. In: **Computer Vision–ACCV 2010**. Springer Berlin Heidelberg, 2011. p. 373-384.

FRÄNTI, Pasi et al. Genetic algorithms for large-scale clustering problems.**The Computer Journal**, v. 40, n. 9, p. 547-554, 1997.

FRIEDMAN, Herman P.; RUBIN, Jerrold. On some invariant criteria for grouping data. **Journal of the American Statistical Association**, v. 62, n. 320, p. 1159-1178, 1967.

GABRIEL, Paulo Henrique Ribeiro; DELBEM, Alexandre Cláudio Botazzo. Fundamentos de algoritmos evolutivos. 2008.

GANDHI, Gopi; SRIVASTAVA, Rohit. ANALYSIS AND IMPLEMENTATION OF MODIFIED K-MEDOIDS ALGORITHM TO INCREASE SCALABILITY AND EFFICIENCY FOR LARGE DATASET.

GONÇALVES, Márcio L.; DE ANDRADE NETTO, Márcio L.; COSTA, José Alfredo F. Explorando as Propriedades do Mapa Auto-organizável de Kohonen na Classificação de Imagens de Satélite.

GRILO, Carlos Fernando Almeida. Aplicação de algoritmos evolucionários à extracção de padrões musicais. 2003.

GUHA, Saikat; TANG, Kevin; FRANCIS, Paul. NOYB: privacy in online social networks. In: **Proceedings of the first workshop on Online social networks**. ACM, 2008. p. 49-54.

HALKIDI, Maria; BATISTAKIS, Yannis; VAZIRGIANNIS, Michalis. On clustering validation techniques. **Journal of intelligent information systems**, v. 17, n. 2-3, p. 107-145, 2001

HALL, Lawrence O.; ÖZYURT, Ibrahim Burak; BEZDEK, James C. Clustering with a genetically optimized approach. **Evolutionary Computation, IEEE Transactions on**, v. 3, n. 2, p. 103-112, 1999.

HAN, J., Kamber, M. and Pei, J. (2011). **Data mining: concepts and techniques: concepts and techniques**, Elsevier.

HANDL, Julia; KNOWLES, Joshua. An evolutionary approach to multiobjective clustering. **Evolutionary Computation, IEEE Transactions on**, v. 11, n. 1, p. 56-76, 2007.

HONKELA, Timo. **Self-organizing maps in natural language processing**. 1997. Tese de Doutorado. Helsinki University of Technology.

HRUSCHKA, Eduardo Raul, et al. "**A survey of evolutionary algorithms for clustering**." Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 39.2 (2009): 133-155.

HRUSCHKA, Eduardo R. et al. A genetic algorithm for *cluster* analysis.**Intelligent Data Analysis**, v. 7, n. 1, p. 15-25, 2003.

HSIEH, Nan-Chen. An integrated data mining and behavioral scoring model for analyzing bank customers. **Expert systems with applications**, v. 27, n. 4, p. 623-633, 2004.

HUANG, Zhiheng; GEDEON, Tamás D.; NIKRAVESH, Masoud. Pattern trees induction: A new machine learning method. **IEEE Transactions on Fuzzy Systems**, v. 16, n. 4, p. 958-970, 2008.

HÜLLERMEIER, Eyke et al. **Label ranking by learning pairwise preferences**. Artificial Intelligence, v. 172, n. 16, p. 1897-1916, 2008.

HUTTENHOWER, Curtis et al. Nearest Neighbor Networks: clustering expression data based on gene neighborhoods. **Bmc Bioinformatics**, v. 8, n. 1, p. 250, 2007.

IBRAHIM, Lamiaa Fattouh; HARBI, Manal Hamed Al. Using Modified Partitioning Around Medoids Clustering Technique in Mobile Network Planning. **arXiv preprint arXiv:1302.6602**, 2013.

JANSEN, S. M. H. Customer Segmentation and Customer Profiling for a Mobile Telecommunications Company Based on Usage Behavior. **A Vodafone Case Study July**, 2007.

KAWAJI, Hideya; TAKENAKA, Yoichi; MATSUDA, Hideo. Graph-based clustering for finding distant relationships in a large set of protein sequences.**Bioinformatics**, v. 20, n. 2, p. 243-252, 2004.

KIVIJÄRVI, Juha; FRÄNTI, Pasi; NEVALAINEN, Olli. Self-adaptive genetic algorithm for *cluster*ing. **Journal of Heuristics**, v. 9, n. 2, p. 113-129, 2003.

KRISHNA, K.; MURTY, M. Narasimha. Genetic K-means algorithm.**Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on**, v. 29, n. 3, p. 433-439, 1999.

KROVI, Ravindra. Genetic algorithms for *cluster*ing: a preliminary investigation. In: **System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on**. IEEE, 1992. p. 540-544.

KOTLER, P. (2003). **Kotler marketing de A a Z: 80 conceitos que todo profissional precisa saber, Gulf Professional Publishing**.

KOZA, J. **R. Genetic Programming: On the Programming of Computers by Means of Natural Selection**. Cambridge, MA, USA: MIT Press, 1992.

KRAUSE, Antje; STOYE, Jens; VINGRON, Martin. Large scale hierarchical clustering of protein sequences. **BMC bioinformatics**, v. 6, n. 1, p. 15, 2005.

TAN, Pang-Ning; STEINBACH, Micahel; KUMAR, Vipin. Introduction to Data Mining. Person Education. **Inc., New Delhi**, 2006.

LEE, C. Y.; ANTONSSON, E. K. Dynamic partitional clustering using evolution strategies. In: **IECON-PROCEEDINGS-**. 2000. p. 2716-2721.

LEWIS, David D. An evaluation of phrasal and *cluster*ed representations on a text categorization task. In: **Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval**. ACM, 1992. p. 37-50.

LINCK, Ricardo Ramos. SISTEMA DE CLUSTERIZAÇÃO PARA DESLOCAMENTO DE PESSOAS EM GRUPO.

LOPES, Maria Célia Santos. **Mineração de dados textuais utilizando técnicas de *cluster*ing para o idioma português**. 2004. Tese de Doutorado. UNIVERSIDADE FEDERAL DO RIO DE JANEIRO.

LU, Yinghua et al. Implementation of the fuzzy c-means clustering algorithm in meteorological data. **International Journal of Database Theory and Application**, v. 6, n. 6, p. 1-18, 2013

LU, Yi et al. FGKA: A fast genetic k-means clustering algorithm. In:**Proceedings of the 2004 ACM symposium on Applied computing**. ACM, 2004. p. 622-623.

LU, Yi et al. Incremental genetic K-means algorithm and its application in gene expression data analysis. **BMC bioinformatics**, v. 5, n. 1, p. 172, 2004.

LUCASIUS, Carlos B.; DANE, Adrie D.; KATEMAN, Gerrit. On k-medoidclustering of large data sets with the aid of a genetic algorithm: background, feasiblity and comparison. **Analytica Chimica Acta**, v. 282, n. 3, p. 647-669, 1993.

MA, Patrick CH et al. An evolutionary clustering algorithm for gene expression microarray data analysis. **Evolutionary Computation, IEEE Transactions on**, v. 10, n. 3, p. 296-314, 2006.

MACIEL, Andrilene Ferreira. Uma interpretação nebulosa dos mapas de Kohonen. **Modelagem Computacional de Conhecimento**, 2008.

MAULIK, Ujjwal; BANDYOPADHYAY, Sanghamitra. Genetic algorithm-based *cluster*ing technique. **Pattern recognition**, v. 33, n. 9, p. 1455-1465, 2000.

MERZ, Peter; ZELL, Andreas. Clustering gene expression profiles with memetic algorithms. In: **Parallel Problem Solving from Nature—PPSN VII**. Springer Berlin Heidelberg, 2002. p. 811-820.

MILLER, J. F.; HARDING, S. L. **Cartesian Genetic Programming Proceedings of the 11th Annual Confere**nce Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers. Anais...: GECCO '09.New York, NY, USA: ACM,2009Disponível em: <http://doi.acm.org/10.1145/1570256.1570428>.

MILLER, J. F.; SMITH, S. L. **Redundancy and computational efficiency in Cartesiangenetic programming**. IEEE Transactions on Evolutionary Computation, v. 10, n. 2,p. 167–174, abr. 2006.

MILLER, J. F.; THOMSON, P. **Cartesian Genetic Programming**. In: POLI, R. et al. (Eds.). Genetic Programming. Lecture Notes in Computer Science. [s.l.] Springer Berlin Heidelberg, 2000. p. 121–132.

MURTHY, Chivukula A.; CHOWDHURY, Nirmalya. In search of optimal clusters using genetic algorithms. **Pattern Recognition Letters**, v. 17, n. 8, p. 825-832, 1996.

NAGPAL, Arpita; JATAIN, Aman; GAUR, Deepti. Review based on data clustering algorithms. In: **Information & Communication Technologies (ICT), 2013 IEEE Conference on**. IEEE, 2013. p. 298-303.

NALDI, Murilo Coelho; ANDRÉ CARLOS PONCE LEON FERREIRA DE CARVALHO. Clustering using genetic algorithm combining validation criteria. In: **ESANN**. 2007. p. 139-144.

NALDI, Murilo Coelho; FACELI, Katti; CARVALHO, André. Uma Revisão Sobre Combinação de Agrupamentos. **Revista de Informática Teórica e Aplicada**, v. 16, n. 2, p. 25-52, 2009.

NASCIMENTO, Maria Cristina Vasconcelos. **Metaheurísticas para o problema de agrupamento de dados em grafo**. 2010. Tese de Doutorado. Tese de Doutorado, Universidade de São Paulo.[Links].

RYGIELSKI, Chris, JYUN-CHENG Wang, and David C. Yen. **"Data mining techniques for customer relationship management."** Technology in society24.4 (2002): 483-502.

SANTOS, Anderson R.; DO AMARAL, Jorge Luís M. Síntese de árvores de padrões fuzzy através de programação genética.

OCHI, Luiz Satoru; DIAS, Carlos Rodrigo; SOARES, Stênio S. Furtado. Clusterização em Mineração de Dados. **Instituto de Computação-Universidade Federal Fluminense-Niterói**, 2004.

OLIVEIRA, Alexandre CM; LORENA, Luiz AN. Hybrid evolutionary algorithms and clustering search. In: **Hybrid Evolutionary Algorithms**. Springer Berlin Heidelberg, 2007. p. 77-99.

PAN, Shih-Ming; CHENG, Kuo-Sheng. Evolution-based tabu search approach to automatic clustering. **Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on**, v. 37, n. 5, p. 827-838, 2007.

PARIS, Paulo Cesar Donizeti et al. Simulador de arquitetura para processamento de imagens usando programação genética cartesiana. 2013.

PERES, Sarajane Marques et al. Tutorial sobre Fuzzy-c-Means e Fuzzy Learning Vector Quantization: Abordagens Híbridas para Tarefas de Agrupamento e Classificação. **Revista de Informática Teórica e Aplicada**, v. 19, n. 1, p. 120-163, 2012.

PING, Kuik Sok; BT SALIM, Naomie. Optimized Subtractive Clustering for Cluster-Based Compound Selection. In: **Proceedings of the 1st International Conference on Natural Resources Engineering & Technology**. p. 492-499.

RENNO, C. D.; SOARES, J. V. Modelos hidrológicos para a gestão ambiental. Programa de ciência e tecnologia para gestão de ecossistemas–"Ação, métodos, modelos e geoinformação para a gestão ambiental". **Ministério da Ciência e Tecnologia. Instituto Nacional de Pesquisas Espaciais. Relatório técnico parcial**, 2000.

SANTANA, S. A. and FARIAS, S. d. A. (2003). Comunicação integrada de marketing e valor de marca: um estudo exploratório em empresas de tecnologia da informação, XXVI Congresso Anual em Ciência da Comunicação.

SCHEUNDERS, Paul. A genetic c-means *cluster*ing algorithm applied to color image quantization. **Pattern Recognition**, v. 30, n. 6, p. 859-866, 1997

SEMAAN, Gustavo Silva, et al. "**Proposta de um método de classificação baseado em densidade para a determinação do número ideal de grupos em problemas de clusterização**." Journal of the Brazilian Computational Intelligence Society 10.4 (2012): 242-262.

SENGE, Robin; HÜLLERMEIER, Eyke. Top-down induction of fuzzy pattern trees. **IEEE Transactions on Fuzzy Systems**, v. 19, n. 2, p. 241-252, 2011.

SHENG, Weiguo; LIU, Xiaohui. A hybrid algorithm for k-medoid *cluster*ing of large data sets. In: **Evolutionary Computation, 2004. CEC2004. Congress on**. IEEE, 2004. p. 77-82.

SILVA, IN da; SPATTI, Danilo Hernane; FLAUZINO, Rogério Andrade. Redes neurais artificiais para engenharia e ciências aplicadas. **São Paulo: Artliber**, p. 221-240, 2010.

SOLOMON, M. (n.d.). **Comportamento do consumidor–comprando, possuindo e sendo**, 5a edição, 2006.

SOULE, T. **Code Growth in Genetic Programming**. Moscow, ID, USA: University of Idaho, 1998.

SOULE, T.; HECKENDORN, R. B. An Analysis of the Causes of Code Growth in Genetic Programming. **Genetic Programming and Evolvable Machines**, v. 3, n. 3, p. 283–309, set. 2002.

TAN, Pang-Ning; STEINBACH, M.; KUMAR, V. Chapter 6. Association analysis: basic concepts and algorithms. Introduction to Data Mining. 2005.

TATIRAJU, Suman; MEHTA, Avi. Image Segmentation using k-means *cluster*ing, EM and Normalized Cuts. **University Of California Irvine**, 2008

TSIPTSIS, K. K., & CHORIANOPOULOS, A. A. (2011). **Data mining techniques in CRM: inside customer segmentation**. John Wiley & Sons.

TSENG, Lin Yu; YANG, Shiueng Bien. A genetic approach to the automatic clustering problem. **Pattern Recognition**, v. 34, n. 2, p. 415-424, 2001.

TURNER, Andrew. **Evolving Artificial Neural Networks using Cartesian Genetic Programming**. 2015. Tese de Doutorado. University of York.

VENKATESAN, R. (2007). **Cluster analysis for segmentation**, (publication number UVA-M-0748). Retrieved on February 14, 2014 from http://faculty.darden.virginia.edu/GBUS8630/doc/M-0748.pdf

XIAO, Qiang; QIAN, Xiao-dong; LIAO, Hui. Clustering algorithm analysis of web users with dissimilarity and SOM neural networks. **Journal of Software**, v. 7, n. 11, p. 2533-2537, 2012.

XU, Rui et al. Survey of clustering algorithms. **Neural Networks, IEEE Transactions on**, v. 16, n. 3, p. 645-678, 2005

YANG, Yiming; PEDERSEN, Jan O. A comparative study on feature selection in text categorization. In: **ICML**. 1997. p. 412-420.

YONAMINE, Frank Sussumu et al. Aprendizado não supervisionado em domínios fuzzy–algoritmo fuzzy c-means. **São Carlos: UFSCAR**, 2002.

YUVARAJU, M.; NIVEDITA, S. R. A Text Based Clustering Scheme With Genetic Programming To Eliminate Replicas. In: **International Journal of Engineering Research and Technology**. ESRSA Publications, 2013.

ZHA, Hongyuan et al. Spectral relaxation for k-means clustering. In:**Advances in neural information processing systems**. 2001. p. 1057-1064

ZHANG, Hui; FRITTS, Jason E.; GOLDMAN, Sally A. Image segmentation evaluation: A survey of unsupervised methods. **computer vision and image understanding**, v. 110, n. 2, p. 260-280, 2008.

ZHANG, Tian; RAMAKRISHNAN, Raghu; LIVNY, Miron. BIRCH: an efficient data clustering method for very large databases. In: **ACM SIGMOD Record**. ACM, 1996. p. 103-114.

WILLIAMS, W. T.; LANCE, G. N. A general theory of classification sorting strategies: 1. Hierarchical systems, 2. Clustering systems. **Computer Journal**, v. 9, n. 10.

APÊNDICE A

Este apêndice apresenta os testes realizados com os conjuntos de teste no item 5.1.3 e 5.1.5 que contem os resultados realizados pelos algoritmos de clusterização e matriz de proximidade do Capítulo 5.

A.1 Resultados das clusterizações realizadas

As Figuras 61 a 64comparam dois algoritmos para os resultados das clusterizações realizadas no Capítulo 4 para o conjunto de dados Banana 1 e 2, um que trata somente formatos hiperesféricos (PGC-AFP) de *cluster*se o outro que trata formato arbitrário.

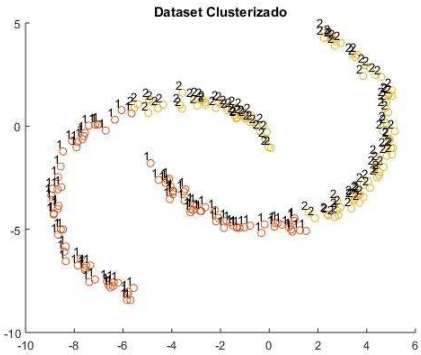


Figura61: Resultado obtido pelo algoritmo PGC-AFP para Banana 1 com *K*=2 (DO AUTOR, 2016)

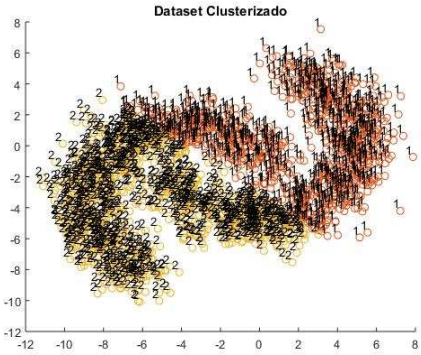


Figura62: Resultado obtido pelo algoritmo PGC-AFP para Banana 2 com *K*=2 (DO AUTOR, 2016)

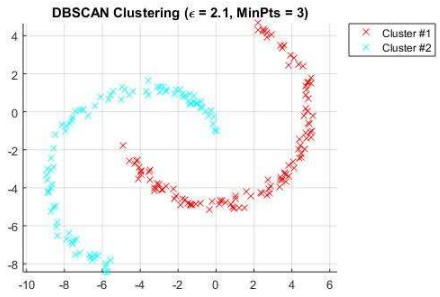


Figura63: Resultado obtido pelo algoritmo DBSCAN para Banana 1 (DO AUTOR, 2016)

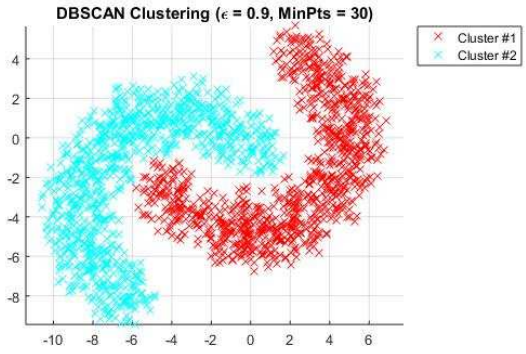


Figura64: Resultado obtido pelo algoritmo DBSCAN para Banana 2 (DO AUTOR, 2016)

A.2 Resultados obtidos com agrupamento de dados

Para o avaliação utilizando o SVM, somente os conjuntos de dados Banana 1 e Banana 2 foram avaliados, pois estes geram *clusters* não hiperesféricos.

Para comparação, serão destacados os resultados obtidos para o algoritmo PGC-AFP.

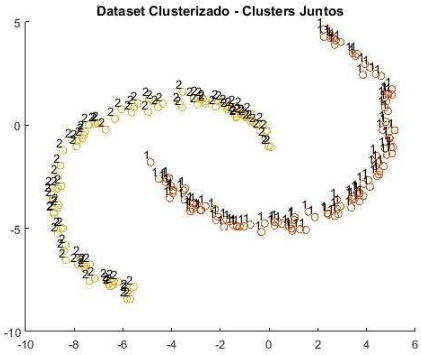


Figura65: Agrupamento 10 *clusters*para Banana 1 algoritmo PGC-AFP em 2 por SVM (DO AUTOR, 2016)

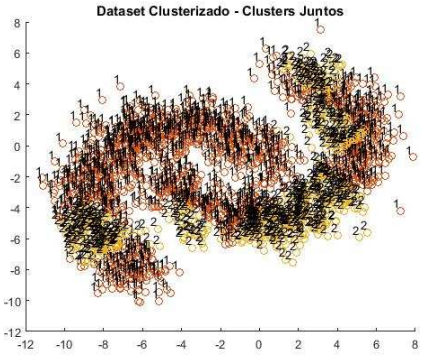


Figura 66: Agrupamento 50 *clusters* para Banana 2 algoritmo PGC-APF em 2 por SVM (DO AUTOR, 2016)