UM ESTUDO COMPUTACIONAL COMPARATIVO ENTRE ALGORITMOS DE AGRUPAMENTO E DE DETECÇÃO DE COMUNIDADES

Daiana Medeiros da Silva
Centro Federal de Educação Tecnológica Celso Suchow da Fonseca – CEFET/RJ
Av. Maracanã, 229 - Maracanã – Rio de Janeiro/RJ
daiana.medeirosdasilva@gmail.com

José André de Moura Brito
Escola Nacional de Ciências Estatística – ENCE/IBGE
Rua André Cavalcanti, 106, sala 403, Centro – Rio de Janeiro – RJ.
jose.m.brito@ibge.gov.br

Carla Silva Oliveira
Escola Nacional de Ciências Estatística – ENCE/IBGE
Rua André Cavalcanti, 106, Centro – Rio de Janeiro – RJ.
carla.oliveira@ibge.gov.br

RESUMO

O presente trabalho tem por objetivo comparar o desempenho de dois algoritmos de Agrupamento não-hierárquicos, quando aplicados a um conjunto de redes, frente a dois algoritmos de Detecção de Comunidades (Fast Greedy e Walktrap). A comparação entre os resultados foi feita a partir da aplicação do Índice de Silhueta. Esses algoritmos foram aplicados em 30 bases de dados artificiais, obtidas através do pacote clustergeneration do software R. Foi possível perceber que os algoritmos de agrupamento (k-means e PAM) apresentam melhores soluções quanto às silhuetas, frente aos algoritmos de Detecção de Comunidades em redes (Fast Greedy e Walktrap).

Palavra-chave: Análise de Agrupamentos; Detecção de Comunidades; k-means; PAM; Fast Greedy.

ABSTRACT

This paper aims to compare the performance of two non-hierarchical clustering algorithms when a network set is applied against a Fast Greedy and Walktrap. A comparison between the results was made by applying the Silhouette Index. The algorithms were applied to 30 artificial databases, through the clustergeneration of software R package. It was possible to understand which clustering algorithms (k-means and PAM) presented the best solutions for silhouettes, facing the Community Detection algorithms in networks (Fast Greedy and Walktrap).

Keywords: Cluster Analysis; Community Detection; k-means; PAM; Fast Greedy.

Como Citar:
SILVA, D. M.; BRITO, J. A. M.; OLIVEIRA, C. S. Um Estudo Computacional Comparativo entre Algoritmos de Agrupamento e de Detecção de Comunidades. *In*: SIMPÓSIO DE PESQUISA OPERACIONAL E LOGÍSTICA DA MARINHA, 19., 2019, Rio de Janeiro, RJ. **Anais** […]. Rio de Janeiro: Centro de Análises de Sistemas Navais, 2019.

1. INTRODUÇÃO

A prática de agrupar os objetos de acordo com as suas semelhanças é a base de grande parte da ciência, ao passo que, organizar dados em agrupamentos é um dos modos mais fundamentais que configura a compreensão e aprendizado [1].

Entretanto, esta tarefa não é trivial. Com o volume expressivo de informações que é recebido todos os dias através da internet, por exemplo, faz-se necessário uma maneira automática de organizar, separar e classificar tais informações.

Uma alternativa para tratar esse problema diz respeito à aplicação da Análise de Agrupamentos que corresponde a uma técnica de análise multivariada que agrega um conjunto de algoritmos que possibilitam alocar os objetos de uma base de dados em grupos sem a necessidade, a priori, do conhecimento de suas classes ou suas categorias [1].

Atualmente, existem diversas aplicações reais as quais podem ser abordadas e resolvidas através da Análise de Agrupamentos, em diversas áreas do conhecimento, a saber: psicologia, ciências sociais, biologia, medicina, marketing, estatística, mineração de dados [[1], [2], [3]].

Outro tipo de análise, além da Análise de Agrupamentos, diz respeito à aplicação de algoritmos de Detecção de Comunidades. Mais especificamente, redes são definidas como "sistemas físicos, biológicos ou sociais caracterizados por um conjunto de entidades bem definidas que se integram e interagem dinamicamente entre si" [4]. Considerando essa definição, podemos representar através de redes, diversas estruturas: redes de coautoria, redes de esgoto, redes de telefonia, redes de computadores, entre outras. As redes podem ser modeladas a partir da utilização de uma estrutura matemática denominada grafo $G = (V, E)$. Através desta representação, pode-se obter variadas informações no que concerne à topologia dessas redes como, por exemplo, o fluxo de informação que passa pelos nós e clusters de nós [5]. Por sua vez, estudar e analisar redes com substancial número de vértices e arestas, como a exemplo a rede de contato do twitter, com mais de 50 milhões de indivíduos, ou mesmo uma rede de neurônios, demanda, com frequência, a utilização de métodos associados à Teoria dos Grafos. Neste sentido, um dos problemas utilizados na análise de redes via Teoria dos Grafos é o de detecção de comunidades, ou seja, o particionamento do grafo que modela a rede em clusters [6].

Por esta razão, os algoritmos de Análise de Agrupamentos e de Detecção de Comunidades podem ser utilizados como uma importante ferramenta auxiliar de análise, em diferentes tipos de redes tais como: Qualidade, como instrumento de apoio à melhoria da qualidade dos serviços aos clientes [7]; Educação, como instrumento de apoio na avaliação da formação técnica profissionalizante para o setor industrial [8]; World Wide Web, como instrumento de apoio para melhorar os resultados disponibilizados por sites de busca [9].

Neste contexto, este trabalho tem por objetivo comparar o desempenho de dois algoritmos de Agrupamento não-hierárquicos, quando aplicados a um conjunto de redes, frente a dois algoritmos de Detecção de Comunidades (*Fast greedy* e *Walktrap*).

O presente trabalho está dividido da seguinte forma: na Seção 2 são apresentados os

conceitos básicos sobre a Teoria dos Grafos, Análise de Agrupamentos e os algoritmos utilizados para o agrupamento de dados e detecção de Comunidades. Na Seção 3 é apresentada a metodologia deste trabalho. E por fim, na Seção 4, são apresentados os resultados computacionais e as considerações finais.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. UMA BREVE REVISÃO SOBRE GRAFOS

Um grafo é uma estrutura *G= (V,E)*, sendo *V* correspondente a um conjunto discreto cujos objetos são denominados de vértices ou nós e, *E* um conjunto de subconjuntos de objetos de *V*, cujos objetos são denominados de ligações ou arestas. O grafo *G* é de ordem *n* e tamanho *m* quando $|V|=n$ e $|E|=m$ e este pode ser direcionado ou orientado quando existe uma orientação nas ligações, as quais são denominadas de arcos.

Quando não existe orientação nas arestas dizemos que o grafo é não orientado ou não direcionado. Neste trabalho, foram considerados grafos não orientados. Um vértice *v* é denominado *adjacente* ou *vizinho* a outro vértice *w* quando existe uma ligação entre eles. O grau de um vértice $v_i \in V$, denotado por $d(v_i)$, é o número de vértices adjacentes ao vértice v_i, ou ainda é o número de vizinhos a v_i.

Um *grafo* é denominado *vazio* quando $|E|=0$ e denominado *trivial* ou *singleton* se possui apenas um único vértice. Um *caminho* é uma sequência de vértices $v_1, \dots, v_i, \dots, v_k$ de modo que $(v_i, v_{i+1}) \in E, \forall 1 \leq i \leq k-1$. O comprimento de um caminho entre dois vértices v_i e v_j é o número de arestas existentes nele e denominamos de distância *geodésica* o comprimento do menor caminho entre dois vértices v_i e v_j, a qual denotamos por $d_{v_i v_j}$. Dado dois grafos $G=(V,E)$ e $H=(V',E')$, H é denominado *subgrafo* de *G* se $v' \subseteq v$ e $E' \subseteq E$.

Existem várias maneiras de representar um grafo, sendo uma delas por matrizes, como**,** por exemplo, através de: matriz de adjacência, Laplaciana, Laplaciana sem sinal, Laplaciana Normalizada e de Incidência. Essas maneiras de representar grafos através de matrizes é extremamente útil e, ao mesmo tempo, facilita o seu armazenamento. Neste trabalho, utilizamos *matriz de adjacência*, denotada por $A(G) = \left[a_{v_i v_j}\right]$, a qual é uma matriz de ordem *n*, onde o elemento $a_{v_i v_j} = 1$ se $(v_i, v_j) \in E$, $a_{v_i v_j} = 0$, caso contrário.

2.2. ANÁLISE DE AGRUPAMENTOS

A Análise de Agrupamentos é o estudo formal de algoritmos para agrupar objetos [1]. Dado um conjunto de dados formado por *n* objetos e com *p* atributos (características), o objetivo da Análise de Agrupamentos será particionar o conjunto de objetos em *k* grupos, de forma que, os objetos que pertencem a um mesmo grupo sejam homogêneos entre si, e os objetos pertencentes a grupos distintos sejam heterogêneos. A fim de, verificar a homogeneidade dentro de um grupo, faz-se necessário utilizar alguma métrica, como, por exemplo: distância Euclidiana e distância de Manhattan. É importante observar que, ao utilizar métricas diferentes para um mesmo conjunto de dados, a homogeneidade poderá não ser a mesma, dado que, métricas distintas geram agrupamentos distintos.

De acordo [10], o problema de agrupamento é definido, formalmente, da seguinte maneira: dado um conjunto X, formado por n objetos, sendo

$X = \{x_1, x_2, \cdots, x_n\}$ no qual, para cada objeto x_i estão associados p atributos, $x_i = \{x_i^1, x_i^2, \cdots, x_i^p\}$, deve-se construir, a partir *X*, k grupos/clusters (C_i), tal que $C = \{C_1, C_2, \cdots, C_k\}$.

Na literatura, o problema de agrupamento pode ser definido ainda de duas formas, quais sejam: problema de agrupamento clássico e problema de agrupamento automático (ou ainda clusterização automática). No problema de agrupamento clássico é necessário que o parâmetro *k* (número de grupos), seja definido a priori. Entretanto, quando este parâmetro não é definido a priori, o problema é dito de agrupamento automático [11]. Neste sentido, o número de soluções possíveis (para o problema de agrupamento clássico), ou seja, o número de possiblidades de agrupar *n* objetos pertencentes a um conjunto *X* e k grupos/*clusters* (C_i), que sejam diferentes entre si, é dado através do número de *Stirling* de segundo tipo, o qual é definido pela equação (I) [12]:

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

De modo a ilustrar a complexidade do problema, a Tabela 1 traz os valores de *Ns*, considerando valores relativamente pequenos de n e k.

Tabela 1: Número de soluções do problema de agrupamento usando o número de *Stirling*

n	k	$Ns(n,k)$
10	2	511
10	3	9.330
10	4	34.105
20	2	524.287
20	3	580.606.446
20	4	45.232.115.901
40	2	549.755.813.887
40	3	2.026.277.026.753.674.246
40	4	50.369.882.873.307.917.364.901

Fonte: Os Autores

Na Tabela 1, pode-se observar que, à medida que o número n (de objetos) aumenta, o número de soluções possíveis cresce exponencialmente, ou seja, o número de maneiras em que os *n* objetos podem ser agrupados em *k* grupos cresce substancialmente. Tal fato torna proibitiva a aplicação de um método de enumeração exaustiva para encontrar a melhor

solução associada ao *k* grupos. Assim sendo, abre-se mão do ótimo global, e são considerados os algoritmos de agrupamento que sejam capazes de produzir soluções viáveis de qualidade razoável, consumindo um tempo computacional factível [13].

Quando é considerada a dissimilaridade entre os objetos x_i e x_j, denotada por $d(x_i,x_j)$, quanto mais próximo $d(x_i,x_j)$ for de zero, mais semelhante em relação aos p atributos são os objetos x_i e x_j são [14]. Ainda neste sentido, definimos a matriz de dissimilaridades como sendo uma matriz quadrada, denotada por *D*, sendo cada entrada correspondente à dissimilaridade dos objetos x_i e x_j. Para atributos do tipo quantitativos, as medidas de dissimilaridade mais utilizadas são: distância Euclidiana, distância de Manhattan e distância de Mahalanobis.

A distância Euclidiana entre dois objetos, calculada a partir de seus *p* atributos, é definida pela equação (II), [3]:

$$d(x_i,x_j) = \sqrt{(x_{i1}-x_{j1})^2+(x_{i2}-x_{j2})^2+\cdots+(x_{ip}-x_{jp})^2} \quad \text{(II)}$$

A distância de Manhattan entre os objetos, calculada a partir de *p* atributos, é definida pela equação (III), [3]:

$$d(x_i,x_j) = |x_{i1}-x_{j1}|+|x_{i2}-x_{j2}|+\cdots+|x_{ip}-x_{jp}| \quad \text{(III)}$$

De modo geral, quando os objetos possuem apenas atributos quantitativos e esses atributos têm magnitudes diferentes, faz-se necessário realizar uma padronização dos mesmos [14]. Para padronizar cada variável, x_i, deve-se utilizar a seguinte expressão:

$Z_{ij} = \frac{x_{ij}-\mu_j}{\sigma_i}$, sendo μ_i e σ_i correspondentes, respectivamente, à média e ao desvio padrão da *i*-ésima variável x_{ij} em relação aos n objetos. Após realizar a padronização das variáveis, e gerar a matriz de distâncias, é necessário escolher um algoritmo que possibilitará o agrupamento dos dados.

Os algoritmos de agrupamento podem ser classificados, basicamente, em hierárquicos e não hierárquicos. Os algoritmos do tipo hierárquico caracterizam-se por não ter como dado de entrada o número de grupos. E ainda são divididos em: aglomerativos e divisivos. Nos algoritmos não hierárquicos, o número de grupos é definido a priori. Em particular, neste trabalho, foram utilizados os algoritmos não hierárquicos *k-means* e *PAM*. Esses algoritmos foram escolhidos em virtude de sua grande utilização em diversos trabalhos da literatura.

O algoritmo *k-means* é um dos mais conhecidos da literatura e mais utilizado em problemas práticos [15]. Uma vez definido o valor de *k*, inicialmente, cada objeto é alocado a um grupo de forma aleatória, ou seja, é gerado um vetor com n posições, sendo atribuído a cada posição desse vetor um valor entre 1 e k que indica que a qual grupo inicialmente o i-ésimo objeto será alocado. Posteriormente, deve-se calcular as distâncias de cada objeto a cada centroide (médias dos atributos), dos objetos de cada um dos grupos. Os objetos são realocados ao grupo associado ao centroide que estiver a menor distância calculada e os

centroides são novamente atualizados. Este processo continua até que não haja mais mudanças significativas em relação aos valores dos centroides entre duas iterações seguidas.

Já no que se refere às principais características deste algoritmo, pode-se destacar que: é suscetível a valores atípicos e trabalha com variáveis quantitativas, sendo de fácil implementação, [15].

O algoritmo *Partitioning Aroud Medoids* (*PAM)* tem como objetivo "selecionar *k medoids* do conjunto *X*, que serão alocados a cada um dos grupos, de forma que a soma das distâncias dos outros pontos do grupo a este *medoid* é mínima" [16].

Formalmente, é definido da seguinte maneira: seja o conjunto *X* com *n* objetos, $X = \{x_1, x_2, \ldots, x_n\}$. A partir deste conjunto, inicialmente, deve-se escolher *k* objetos, a fim de formar o conjunto de *medoid*s, $M = \{\text{medoid}_1, \text{medoid}_2, \ldots, \text{medoid}_k\}$.

Posteriormente, cada um dos $(n-k)$ objetos restantes são alocados ao grupo cuja distância do objeto ao *medoid* seja mínima. Este processo deve ser repetido até que não haja mudanças significativas no valor da função (IV) abaixo.

$$\sum_{i=1}^{k} \sum_{\forall x_j \in med_i} d(x_i, x_j) \qquad \text{(IV)}$$

Em síntese, o algoritmo é dividido em: (1) selecionar, dentre os *n* objetos disponíveis de *X*, os *k* objetos que definem um conjunto *M* de *medoid*s; (2) associar os $(n-k)$ objetos restantes ao seu *medoid* mais próximo; (3) substituir os *k medoid*s de forma a minimizar a soma das distâncias dos $(n-k)$ objetos ao seu *medoid* mais próximo; (4) repetir os passos 2 e 3 até que não haja nenhuma mudança significativa nas posições dos *medoids*.

Ao aplicarmos os algoritmos de agrupamento há alguns fatores que podem influenciar na qualidade dos agrupamentos gerados, como a escolha da medida de dissimilaridade ou, até mesmo, os parâmetros utilizados pelo próprio algoritmo como, por exemplo, o número de grupos, as variáveis utilizadas ou número de iterações do algoritmo.

Por esta razão é importante avaliar, a posteriori, a qualidade dos agrupamentos produzidos pelos algoritmos. De modo geral, esta qualidade pode ser verificada com base em índices estatísticos. Estes índices estão associados ao que a literatura chama de critérios de validação [17]. Em particular, neste trabalho foi considerado o índice de Silhueta, proposto por Rousseeuw em 1987, o qual é capaz de avaliar os agrupamentos encontrados, no que tange à qualidade da alocação dos objetos aos grupos [18].

Mais especificamente, esse índice evidência o posicionamento dos objetos nos grupos, o que permite identificar se cada um dos destes objetos estão bem posicionados, ou seja, se foram alocados ao grupo ideal. Para cada objeto x_i o valor da Silhueta, $s(x_i)$, é dado através da seguinte expressão (V) abaixo:

$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max\{a(x_i), b(x_i)\}}, \qquad \text{(V)}$$

sendo:

- $a(x_i) = \frac{1}{|C_z|} \sum d(x_i, x_j), \quad \forall x_i \neq x_j,\ x_i\ e\ x_j \in C_z$;

- $d_{externa}(x_i, C_t) = \frac{1}{|C_t|} \sum d(x_i, x_j), \quad x_i \notin C_t \, , \, \forall x_j \in C_t$;
- $b(x_i) = \min d_{externa}(x_i, C_t) \, , \; C_t \neq C_z \, , x_i \in C_t$;

Temos que $b(x_i)$ é o mínimo da distância externa entre o objeto x_i em relação a todos os objetos do grupo vizinho, mais próximo a ele. Já o termo $a(x_i)$ representa o cálculo da distância média em relação aos demais objetos do mesmo grupo; $|c_z|$ representa a quantidade de objetos do grupo c_z e analogamente, $|c_t|$ representa a quantidade de objetos do grupo c_t.

O valor da silhueta $s(x_i)$, de cada objeto varia em [-1,1]. Se observarmos os possíveis valores de $s(x_i)$, verificamos que, se $a(x_i) > b(x_i)$, $s(x_i)$ assume valores negativos, isto indica que o objeto não está bem posicionado no seu grupo. Entretanto, se verificarmos o caso contrário, $a(x_i) < b(x_i)$, o valor do coeficiente de silhueta fica próximo a 1 e isto indica que o objeto está bem posicionado no seu grupo e que os objetos do grupo C_z estão bem próximos dos objetos do seu próprio grupo. Além disso, existe a possiblidade de que $a(x_i) = b(x_i)$, ou seja, $s(x_i) = 0$. Em síntese, valores que são mais próximos de 1 indicam um melhor agrupamento [19], [20].

A partir do valor da silhueta de cada objeto, define-se a silhueta média, que é a medida efetivamente utilizada para avaliar os grupos formados. Quando a silhueta média tem um valor acima de 0.7, implica uma boa estrutura do agrupamento. Valores na faixa de 0.51 e 0.7 indicam uma estrutura razoável de agrupamento e valores entre 0.26 e 0.5 indicam estrutura fraca [14].

2.3. DETECÇÃO DE COMUNIDADES

Em muitas redes reais uma propriedade comum é a existência de comunidades (grupos) com alta interação entre os elementos de um mesmo grupo, ou seja, a conexão dentro de cada grupo ser densa. Em contrapartida, existe uma baixa densidade nas relações entre os grupos. Esta propriedade é conhecida na literatura como “estrutura de comunidade”.

A Figura 1 ilustra uma rede associada a um grafo onde há uma alta densidade de arestas entre si, dentro de cada comunidade, e uma baixa densidade de arestas entre as comunidades [21].

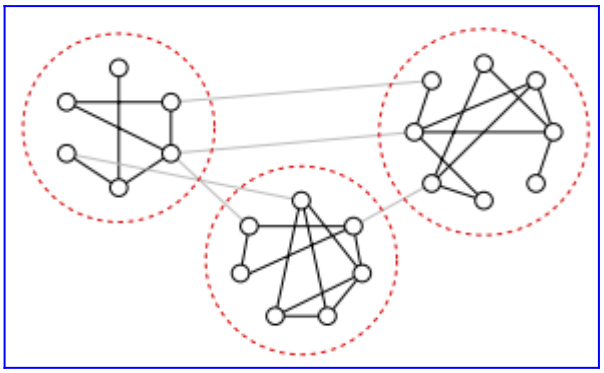

Figura 1: Exemplo de três comunidades de uma rede
Fonte: [21]

O Problema de Detecção de Comunidades é similar ao problema de corte em grafos ou particionamento, porém com objetivos diferentes [[22], [23], [24], [25]]. De modo geral, o problema de particionamento em grafos é definido como o problema de particionamento em k grupos cujos vértices sejam muito parecidos ou até mesmo igual, minimizando o número de arestas entre os grupos. Já a Detecção de Comunidades tem por objetivo procurar comunidades que possuam alguma similaridade entre seus vértices [25].

Vale ressaltar que, bem como o número de vértices e o número de comunidades de uma rede modelada por um grafo são desconhecidos a priori. A fim de determinar essas informações, faz-se necessário aplicar algum algoritmo de Detecção de Comunidades [15]. Admitindo que C_1, C_2,$\cdots$, C_k sejam comunidades encontradas de uma rede, essas comunidades devem satisfazer as seguintes propriedades [24]: (i) $C_i \cap C_j = \emptyset$ e (ii) $\cup C_i = V$. A propriedade (i) indica que nenhum elemento pode pertencer a mais de uma comunidade e a (ii) indica que a união das comunidades é igual ao conjunto de vértices da rede.

De acordo com [25], um dos principais problemas relacionados à Detecção de Comunidades diz respeito: como pode ser definido a melhor divisão da rede em comunidades, visto que, em redes reais, geralmente não há um conhecimento prévio de informações importantes tais como, o número e o tamanho das comunidades existentes.

Desta maneira, não existe nenhuma regra que permita definir a melhor divisão de uma rede real em comunidades. A fim de resolver esse problema, Newman & Girven em 2004 [21], propuseram uma medida, denominada Modularidade. De acordo com [26], o conceito de modularidade foi um dos primeiros a aparecer, quando os pesquisadores começaram a avaliar formas de definir o problema de agrupamentos em grafos, no que tange ao conceito de comunidades. Por sua vez, nos últimos anos, as inúmeras aplicações reais que remetem ao problema de Detecção de Comunidades têm motivado o estudo e a proposta de vários algoritmos que resolvem o problema de determinar a modularidade máxima em uma rede. Outra motivação para o estudo desses métodos diz respeito à complexidade intrínseca a esse problema, que pertence à classe de problemas NP-difíceis [26].

Dentre os vários algoritmos propostos na literatura, apresentamos neste trabalho, os algoritmos *Fast Greedy* e o *Walktrap*. Esses algoritmos foram escolhidos em virtude da boa aceitação e grande utilização pela literatura.

O *Fast Greedy*, proposto por [26] maximiza a medida de modularidade definida em (VI), fazendo uso de uma busca gulosa; como pode ser visto em [27]. Este algoritmo foi baseado no algoritmo proposto por Newman também em 2004.

A modularidade possibilita mensurar a qualidade de uma possível divisão da rede em comunidades. Em outras palavras, se é significativa ou não essa divisão. Essa medida, de acordo com [26], é definida da seguinte maneira:

$$Q = \frac{1}{2m} \sum_{v_i v_j} \left[a_{v_i v_j} - \frac{d(v_i) d(v_j)}{2m} \right] \delta\left(c_{v_i}, c_{v_j}\right) \quad \text{(VI)}$$

sendo cada $a_{v_i v_j}$ correspondente a uma entrada da matriz de adjacência associada aos vértices v_i e v_j do grafo que modela a rede, *m* o número de arestas do grafo, $d(v_i)$ e $d(v_j)$ representam o grau do vértice v_i e do vértice v_j e, $\delta\left(c_{v_i}, c_{v_j}\right) = 1$ quando v_i e v_j pertencem à mesma comunidade, caso contrário $\delta\left(c_{v_i}, c_{v_j}\right) = 0$.

Assim, podemos observar que quando o valor de Q encontra-se muito próximo de zero, isto é um indicativo de que há uma baixa probabilidade de a rede estar dividida em comunidades reais [27]. Para valores próximos de 1, é um indicativo de uma estrutura de comunidade forte. Porém, em muitas das aplicações reais, os valores de Q, encontrados para as redes associadas, não excedem a 0,7 e valores mais altos não ocorrem de maneira frequente, [15].

Neste contexto, o *Fast Greedy* inicialmente considera que cada vértice da rede/grafo como o único membro de uma comunidade e, iterativamente, novos vértices são adicionados à comunidade. A associação de mais um vértice a uma comunidade produz um maior incremento no valor da *modularidade*.

Para que duas comunidades sejam unidas será necessário encontrar o maior valor da variação de modularidade (ΔQ_{ij}), a qual é definida da seguinte maneira:

$$\Delta Q_{ij} = \begin{cases} \frac{1}{2m} - \frac{d(v_i) d(v_j)}{(2m)^2} & \text{se } v_i \text{ e } v_j \text{estiverem } ligados \\ 0 \quad \text{caso contrário} \end{cases} \quad \text{(VII)}$$

A equação (VII) expressa o cálculo da mudança de modularidade. De acordo com [26], o *Fast Greedy* consiste nos seguintes passos:

1 – Primeiramente, deve ser calculados os valores iniciais para a variação de modularidade com relação a cada par *i*, *j* (ΔQ_{ij}), e os valores de $\frac{d(v_i)}{2m}$. Feito isso, armazene no *max-heap* o maior elemento de cada linha da matriz ΔQ;

2 – posteriormente, é identificado o maior valor de ΔQ_{ij} do *max-heap H*, e então deve se juntar as comunidades correspondentes, concomitantemente deve se atualizar a ma-

triz ΔQ, o *max-heap H* e o vetor contendo os valores de $\frac{d(v_i)}{2m}$, e então incrementar *Q* através de ΔQ_{ij};

3 – Por fim, deve se repetir o passo anterior até que reste apenas uma única comunidade.

No *Fast Greedy* a entrada do algoritmo é a rede a ser estuda, e como saída o algoritmo retorna o valor da modularidade *Q* e o número de comunidades. O dendograma também pode ser umas das saídas do algoritmo. A árvore hieráquica ou dendograma é uma representação hierárquica da rede em clusters.

Já o *Walktrap* é um algoritmo de Detecção de Comunidades baseado em caminhada aleatória desenvolvido por [28]. A ideia principal do *Walktrap* é realizar caminhos aleatórios na rede com o intuito que estes caminhos aleatórios permaneçam dentro da mesma estrutura de comunidade. O algoritmo funciona da seguinte maneira: é atribuída uma medida de similaridade entre as comunidades, distância *r*.

Esta medida possibilita o cálculo da distância entre os vértices e/ ou comunidades baseado no caminho percorrido pelo *random walk* entre dois vértices e/ou comunidades [29].

A equação (VIII) representa a distância entre duas comunidades, [28]:

$$r_{C_1 C_2} = \sqrt{\sum_{j=1}^{n} \frac{\left(P^t_{C_1 j} - P^t_{C_2 j}\right)}{d(v_j)}} \qquad \text{(VIII)}$$

onde $d(v_i)$ é o grau do vértice v_i e $P^t_{C,j}$ é a probabilidade da comunidade C_1 ir para o vértice v_i em *t* passos. Com esta medida de distância definida, o algoritmo segue a mesma metodologia do *Fast Greedy*, que funciona como um algoritmo de agrupamento hierárquico aglomerativa. Entretanto, o *Fast Greedy* é baseado na medida de modularidade, já o algoritmo *Walktrap* usa a medida de distância, descrita anteriormente, para identificar as comunidades mais próximas.

Os passos do algoritmo *Walktrap* são apresentados em [28]. Em um primeiro momento, são calculadas as distâncias entre todos os vértices adjacentes e inicia-se com uma partição $P_1 = \{v\}$ onde v $\in$V, e esta partição varia até P_i. Em seguida, o algoritmo segue os seguintes passos:

• escolhe duas comunidades C_1 e C_2 em P_i;

• faz a junção dessas duas comunidades em uma nova comunidade $C_3 = C_1 \cup C_2$, e assim é criada uma nova partição: $P_i + 1 = \{P_i \setminus \{C_1, C_2\}\} \cup \{C_3\}$;

• atualiza as distâncias entre as comunidades. Depois de *n - 1* passos, o algoritmo termina e é obtido P_n = V.

3. METODOLOGIA

Esta seção tem por objetivo descrever o procedimento metodológico utilizado para subsidiar este trabalho.

Para este trabalho foram utilizadas bases de dados artificiais obtidas através do pacote *clustergeneration* do *software R*. Desse pacote foi utilizada a função *genrandomclust*, que permite gerar base de dados, a partir de critérios especificados como pode ser visto em [30].

A fim de gerar as bases de dados artificiais foi especificado, a priori, como parâmetro de entrada, a quantidade de objetos de cada grupo por meio de um vetor. Os objetos que compõem as bases de dados artificiais possuem duas variáveis, as quais são utilizadas para a realização dos cálculos das dissimilaridades, o que permite, posteriormente, realizar a aplicação dos algoritmos de agrupamento. A Figura 2 ilustra a visualização de uma destas bases de dados geradas a partir da função *genrandomclust.*

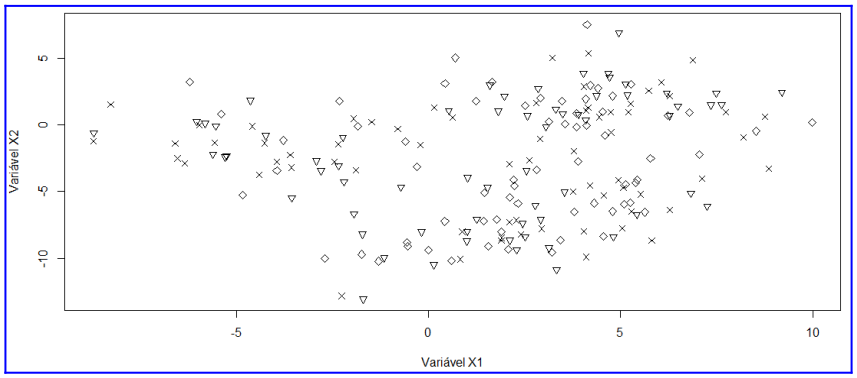

Figura 2: Exemplo de visualização de uma base de dados com duas variáveis

Foram utilizadas 30 bases de dados artificiais. Vale ressaltar que todos os experimentos para este trabalho foram realizados em um computador dotado de processador Intel I5 de 2.4 GHz, com 4 GB de memória RAM e sistema operacional *Windows* 8. Os resultados foram obtidos utilizando os pacotes *igraph*, *cluster*, disponíveis no *software R* versão 3.5.2. Após a obtenção dessas bases de dados o presente estudo baseou-se no esquema ilustrado na Figura *3*.

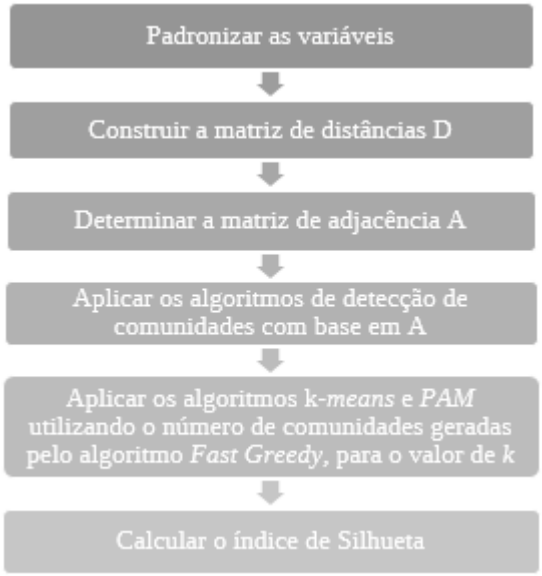

Figura 3: Etapas do Processo Metodológico da Pesquisa

As etapas expostas na Figura 3, são: (i) padronização dos dados; (ii) construção de uma matriz de distância, considerando as distâncias dos objetos tomados dois a dois; (iii) determinação de uma matriz de adjacência dos dados. A construção desta matriz foi obtida da seguinte maneira: cada objeto da base correspondeu a um vértice do grafo e dois objetos são conectados por uma aresta se a distância entre eles for menor que o 3º quartil das distâncias entre todos os objetos, vide [15]. Considerando esse critério, foram removidos do grafo associado à rede os vértices que não tinham nenhuma aresta; (iv) aplicação dos algoritmos de Detecção de Comunidades *Walktrap e Fast Greedy*; (v) aplicação dos algoritmos de Agrupamento *k-means* e *PAM*, utilizando como valor de *k*, o mesmo número de comunidades detectadas pelo algoritmo *Fast Greedy;* e por fim (vi) foi calculada a silhueta média como medida de validação dos resultados.

4. RESULTADOS COMPUTACIONAIS E CONSIDERAÇÕES FINAIS

De modo a comparar os algoritmos: *Fast Greedy*, *Walktrap*, *k*-means e *PAM*, em um primeiro momento, foram aplicados às bases de dados os algoritmos *Fast Greedy* e *Walktrap*. Como entrada para esses dois algoritmos, foi dado o grafo construído e as saídas geradas foram o valor da modularidade *(Q)* e o número de comunidades.

Em um segundo momento, foram aplicados às bases de dados os algoritmos de agrupamento, *k-means* e *PAM*. Para aplicar esses algoritmos, faz-se necessário o conhecimento, a priori, do número de agrupamentos, *k*. Sendo assim, para realizar a análise desses quatro algoritmos foi utilizado o mesmo número de comunidades geradas pelo algoritmo *Fast Greedy*, para o valor de *k*. A justificativa para utilização do resultado do algoritmo *Fast Greedy* se dá pelo fato de ser um dos mais utilizados e conhecidos na literatura, no que tange aos algoritmos de Detecção de Comunidades em redes.

Em síntese, os dados de entrada para os algoritmos *k-means* e *PAM* foram a matriz de dados, para cada base de dados, e o valor de *k* correspondente ao mesmo número das comunidades geradas pelo algoritmo *Fast Greedy*. Após a aplicação de todos os algoritmos, foi utilizado o índice de silhueta como critério de validação para a comparação dos algoritmos, o que pode ser visto na Figura 4:

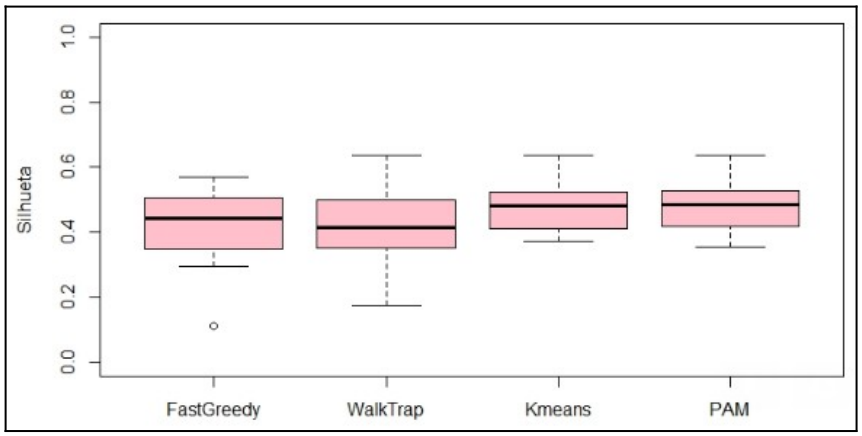

Figura 4: Comparação dos Algoritmos
Fonte: Os Autores

A Figura 4 apresenta o boxplot com as silhuetas médias, construído da seguinte maneira: após aplicar os algoritmos *Fast Greedy*, *Walktrap*, *k-means* e *PAM* nas 30 bases de dados, como resultado, foram produzidas 120 soluções (30 soluções do *Fast Greedy*, 30 soluções do *Walktrap*, 30 soluções do *k-means* e 30 soluções do PAM). Os dados de entrada

para o cálculo da silhueta das 30 soluções do *Fast Greedy* foi a matriz de distâncias de cada uma das bases de dados e o vetor de comunidades (alocação dos objetos às comunidades). Feito isto, foi obtido a distribuição de valores correspondentes às 30 silhuetas médias do algoritmo *Fast Greedy*. O cálculo da silhueta média para os demais algoritmos seguiu o mesmo procedimento. Com essas informações foi possível construir os quatro boxplots, contendo as silhuetas médias associadas a cada algoritmo, apresentado na Figura 4.

Com base na Figura 4, é possível perceber que os algoritmos de agrupamento (*k-means* e *PAM*) apresentam melhores soluções quanto às silhuetas, frente aos algoritmos de Detecção de Comunidades em redes (*Fast Greedy* e *Walktrap*). Ao observar o valor da mediana, representada pela linha interna do boxplot, observa-se que os maiores valores da silhueta média foram apresentados pelos algoritmos *k-means* e *PAM* frente aos valores das medianas dos algoritmos *Fast Greedy* e *Walktrap*.

No que se refere, aos algoritmos de Detecção de Comunidades é possível perceber que o algoritmo *Fast Greedy* apresentou melhores soluções comparado ao algoritmo *Walktrap*. Em contrapartida, observando os dois boxplots dos algoritmos de agrupamento (*k-means* e *PAM*) apresentaram soluções semelhantes.

Para trabalhos futuros, pretende-se utilizar um maior número de bases de dados e outros índices para avaliação da qualidade dos grupos.

5. REFERÊNCIAS BIBLIOGRÁFICAS

[1] JAIN, A.; DUBES, R. Algorithms for Clustering Data. Prentice Hall. 1988.

[2] KUMAR, V.; STEINBACH, M.; TAN, P. N. Introdução ao Data Mining – Mineração de Dados. Ciência Moderna. 2009.

[3] HAN, J.; KAMBER, M.; PEI, J. Data Mining: Concepts and Techniques: Concepts and Techniques. The Morgan Kaufmann Series in Data Management Systems. Elsevier Science, 2012.

[4] TEIXEIRA, L. S.; LIMA, L. S.; ABREU, N. M. M. Grafos que modelam redes confiáveis. In: XL SBPO: A Pesquisa Operacional e o uso racional de recursos hídricos. João Pessoa, PB, Brasil, 2008.

[5] COSTA, J. M. S.; TEIXEIRA, P. M.; BARBASTEFANO, R. G.; SOUZA, C. G.; LIMA, L. S. Aplicação da análise de redes sociais em uma rede de publicações sobre gestão da cadeia de suprimentos. In: XXXIII Encontro Nacional de Engenharia de Produção, Bahia, 2013.

[6] MENTZ, J.; CALVO, R.; SENO, E. R. M.; ROMERO, R. A. F.; LIANG, Z. Redes Complexas: Conceitos e aplicações. Instituto de Ciências Matemáticas e de Computação. Relatório Técnico, n. 290, 2007.

[7] MOORI, R. G.; MARCONDES, R. C.; AVILA, R. T. A análise de agrupamentos como instrumento de apoio à melhoria da qualidade dos serviços aos clientes. Rev. adm. contemp. [online]. vol.6, n.1, pp. 63-84, ISSN 1982-7849, 2002.

[8] SANTOS, M.; GUIZZO, C. S. P.; SAMPAIO, R. R.; CORDEIRO, A. J. A. Aplicação da análise de agrupamento na avaliação da formação técnica profissionalizante para o setor industrial. XVLI Simpósio brasileiro de pesquisa operacional, Bahia, 2014.

[9] HAMMOUDA, K. M. “Web Mining: Identifying Document Structure for Web Document Clustering”, Tese de Mestrado, Department of Systems Design Engineering, University of Waterloo, Canada, 2002.

[10] HANSEN, P.; JAUMARD B. Cluster Analysis and Mathematical Programming, Mathematical Programming, 1997.

[11] NALDI, M. C. Técnicas de combinação para agrupamento centralizado e distribuído de dados. Tese de doutorado, USP, 2011.

[12] LIU, C. L. Introduction to Combinatirial Mathematics (Computer Science Sereies). Mcgraw-Hill College. 1968.

[13] BRITO, J.A.M.; SEMAAN, G.S.; BRITO, L. R. Resolução do Problema dos k-medoids Via Algoritmo Genético de Chaves Aleatórias Viciadas. In: XVIII Simpósio de Pesquisa Operacional e Logística da Marinha (SPOLM), Rio de Janeiro, 2015.

[14] KAUFMAN, L.; ROUSSEEUW, P. J. An introduction to cluster analysis. John Wiley and Sons. 1990.

[15] ALVES, I.; OLIVEIRA, C. S.; BRITO, J. A. M. Um estudo do problema de detecção de comunidades em redes. Revista Eletrônica Sistemas & Gestão, v. 9 n. 4, pp. 566-576, 2014.

[16] CRUZ, M. D. O problema de Clusterização Automática. Tese de Doutorado, COPPE/ UFRJ, Rio de Janeiro, RJ, Brasil, 2010.

[17] OLIVEIRA, T. B. S. Cluserização de dados utilizando técnicas de redes complexas e computação bioinspirada Dissertação de mestrado -Instituo de Ciências Matemática e de Computação - USP, São Carlos -SP, 2008.

[18] ROUSSEUW, P. J. Silhouetts: A grafical aid to the interpretation and validation of cluster analysis. Journal of Computatuinal and Applied Mathematics, 20(1):53-65. 1987.

[19] SILVA, L. F. Uma Análise Híbrida Para Detecção De Anomalias Da Mama Usando Séries Temporais De Temperatura. Tese de Doutorado em Computação. Universidade Federal Fluminense, Niterói, 2015.

[20] FILHO, J. A. A. Definição Automática da Quantidade de Atributos Selecionados em Tarefas de Agrupamento de Dados. Tese de Doutorado em Ciências. Instituto de Ciências Matemáticas e de Computação-ICMC-USP, São Carlos, 2013.

[21] NEWMAN, M. E. J.; GIRVAN, M. Finding and evaluating community structure in networks. Physical Review E, 69(026113), 2004.

[22] KARGER, D. Minimum cuts in near-linear time. Journal of the ACM (JACM), v. 47, n. 1, p. 46-76, 2000.

[23] KERNIGHAN, B.; LIN, S. An efficient heuristic procedure for partitiong graphs. Bell System Technical Journal, v. 49, n. 2, p.291-307, 1970.

[24] FIDUCCIA, C.; MATTHEYSES, R. M. “A linear-time heuristic for improving network partitions”,In: 19th IEE Conference on Design Automation, 1982.

[25] LINARES, O. A. C. Segmentação de Imagens de altadimensão por meio de algortimos de detecção de comunidades e super pixels. Dissertação de Mestrado, Instituto de Ciência Matemáticas e de Computação – ICMC - USP, São Carlos, 2013.

[26] CLAUSET, A.; NEWMAN, M. E., MOORE, C. “Finding community structure in very large networks”, Physical Review E., Vol 70, N º 6, 2004.

[27] BOTELHO, G. M.; SOUSA, F. B. Comparação de Algoritmos de Detecção de Comunidades em Redes Complexas. Instituto de Ciências Matemáticas e de Computação São Carlos - SP – Brasil. Universidade de São Paulo (USP). Disponível em:< http://wiki.icmc.usp.br/images/8/81/Trabalho2GlendaFabiano.pdf>. Acessado em: 25 ago. 2015.

[28] PONS, P.; LATAPY, M. Computing communities in large networks using random walks. J. Graph Algorithms Appl., v. 10, n. 2, p. 191–218, 2006.

[29] PORTO, S. M. Metodologia para a Evolução de Comunidades em Redes Complexas Dinâmicas. Dissertação de M.Sc., Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2014.

[30] QIU, W.; JOE, H. Generation of random groups with specified degree of separation. Journal of Classification, v. 23, n. 2, p. 315-334, 2006.