

UNIVERSIDADE FEDERAL DE UBERLÂNDIA - CAMPUS
MONTE CARMELO

Lusmar Mendes Freitas

**Uma Análise Comparativa entre Técnicas de
Detecção de Comunidades com Aplicação para
o Problema de Agrupamento de Objetos
Invariantes**

Monte Carmelo, Brasil

2018

UNIVERSIDADE FEDERAL DE UBERLÂNDIA - CAMPUS
MONTE CARMELO

Lusmar Mendes Freitas

**Uma Análise Comparativa entre Técnicas de Detecção de
Comunidades com Aplicação para o Problema de
Agrupamento de Objetos Invariantes**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia - Campus Monte Car-
melo, Minas Gerais, como requisito exigido
parcial à obtenção do grau de Bacharel em
Sistemas de Informação.

Orientador: Dr. Murillo Guimarães Carneiro

Universidade Federal de Uberlândia - Campus Monte Carmelo

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Monte Carmelo, Brasil

2018

Lusmar Mendes Freitas

Uma Análise Comparativa entre Técnicas de Detecção de Comunidades com Aplicação para o Problema de Agrupamento de Objetos Invariantes

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia - Campus Monte Carmelo, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Trabalho aprovado. Monte Carmelo, Brasil, 19 de Dezembro de 2018:

Dr. Murillo Guimarães Carneiro
Orientador

Dra. Fernanda Maria Cunha Santos
Professor Convidado

Dr. Rafael Dias Araújo
Professor Convidado

Monte Carmelo, Brasil
2018

*A todos aqueles que estiveram presentes direta ou indiretamente na minha formação e
aos que contribuíram de alguma maneira para a realização deste projeto.*

Agradecimentos

Agradeço a Deus, porque na Tua palavra eu encontrei a força necessária para superar todos os obstáculos e continuar caminhando. Por ser meu abrigo e proteção em todos os momentos e por me conceder a realização de tantos sonhos.

A meu pai, Osmar, por compreender a minha ausência nos dias tumultuados, divididos entre trabalho e graduação. Por me incentivar tanto e por se alegrar com cada pequena conquista minha.

Agradeço minha irmã, Kathlin e às minhas sobrinhas, Karolayne e Suzane por serem alegria e tolerância nos meus dias de mau humor.

Ao Professor Murillo Carneiro, minha eterna gratidão por sua disponibilidade em me orientar, por sua paciência, por seus ensinamentos e incentivo que foram fundamentais para a realização deste trabalho.

Aos professores Fernanda Maria Cunha e Rafael Dias Araújo por aceitarem participar da banca examinadora, neste momento tão importante para mim.

Por fim, gostaria de expressar minha gratidão a todos os meus professores, vocês foram peças fundamentais na minha formação.

*“Ele é o meu Deus, o meu refúgio, a minha fortaleza, e nele confiarei.”
(Salmos 91:2)*

Resumo

O agrupamento de dados consiste na identificação de grupos de objetos de acordo com algum critério de similaridade. Normalmente, tal critério está associado apenas aos atributos físicos dos dados utilizando, por exemplo, medidas de distância ou centróides. Uma abordagem mais recente é o uso de redes complexas, também conhecido por Detecção de Comunidades, o qual permite examinar, além dos atributos físicos, a estrutura topológica dos dados. Neste trabalho, propõe-se o uso de algoritmos de Detecção de Comunidades para o problema não supervisionado de reconhecimento de padrões invariantes. Dado um conjunto de imagens de objetos em diferentes posições, ângulos e rotações, o problema consiste em detectar e agrupar imagens relacionadas a um mesmo objeto. Para a realização deste trabalho, foram aplicados três algoritmos de Detecção de Comunidades em bases de dados reais disponíveis na literatura. As bases de dados foram divididas em dois conjuntos, bases de dados no formato de grafo e bases de dados para o agrupamento de objetos invariantes. Experimentos apontam bons resultados por parte destes algoritmos de acordo com um conjunto de métricas de desempenho.

Palavras-chave: redes complexas, detecção de comunidades, agrupamento, reconhecimento de objetos invariantes.

Lista de ilustrações

Figura 1 – Ilustração da (a) classificação de dados e do (b) agrupamento de dados. Retirado e adaptado de (CARNEIRO, 2016).	15
Figura 2 – Exemplos de problemas reais estudados em Redes Complexas. Retiradas de (CARNEIRO, 2016)	16
Figura 3 – Ilustração de um dendrograma com o corte que representa o valor máximo da modularidade. (RAGHAVAN; ALBERT; KUMARA, 2007) . . .	19
Figura 4 – Exemplo das categorias da base dados ETH-80. Cada categoria possui 10 objetos distintos.	25
Figura 5 – Exemplo contendo dois objetos da categoria maçã da base ETH-80. (a) Categoria Maçã - Objeto 2 (b) Categoria Maçã - Objeto 7	26
Figura 6 – Número de Comunidades para as bases de dados (a) Apple, (b) Car, (c) Cow, (d) Cup, (e) Dog, (f) Horse, (g) Pear e (h) Tomato.	34
Figura 7 – Modularidade para as bases de dados (a) Apple, (b) Car, (c) Cow, (d) Cup, (e) Dog, (f) Horse, (g) Pear e (h) Tomato.	34
Figura 8 – Informação Mútua Normalizada para as bases de dados (a) Apple, (b) Car, (c) Cow, (d) Cup, (e) Dog, (f) Horse, (g) Pear e (h) Tomato.	35
Figura 9 – Pureza para as bases de dados (a) Apple, (b) Car, (c) Cow, (d) Cup, (e) Dog, (f) Horse, (g) Pear e (h) Tomato.	36
Figura 10 – Colocação para as bases de dados (a) Apple, (b) Car, (c) Cow, (d) Cup, (e) Dog, (f) Horse, (g) Pear e (h) Tomato.	36
Figura 11 – Média Harmônica para as bases de dados (a) Apple, (b) Car, (c) Cow, (d) Cup, (e) Dog, (f) Horse, (g) Pear e (h) Tomato.	37

Lista de tabelas

Tabela 1	–	Número de Comunidades para Bases de Dados do Tipo Grafo	29
Tabela 2	–	Modularidade para Bases de Dados do Tipo Grafo	30
Tabela 3	–	Pureza para Bases de Dados do Tipo Grafo	30
Tabela 4	–	Colocação para Bases de Dados do Tipo Grafo	30
Tabela 5	–	Média Harmônica para Bases de Dados do Tipo Grafo	31
Tabela 6	–	Comparativo entre os melhores resultados obtidos pelos algoritmos de Detecção de Comunidades utilizando a Média Harmônica das medidas de Pureza e Colocação.	33
Tabela 7	–	Comparativo entre os melhores resultados obtidos utilizando a Média Harmônica	38

Sumário

1	INTRODUÇÃO	11
2	REVISÃO BIBLIOGRÁFICA	14
2.1	Agrupamento de Dados	14
2.1.1	Algoritmo de Agrupamento k-means	14
2.2	Deteccção de Comunidades em Redes Complexas	15
2.2.1	Modularidade em Redes Complexas	17
2.2.2	Algoritmos de Deteccção de Comunidades	18
2.2.2.1	Fastgreedy	18
2.2.2.2	Labelpropagation	18
2.2.2.3	Walktrap	19
2.3	Medidas de Avaliação de Agrupamento	21
2.3.1	Informação Mútua Normalizada	21
2.3.2	Pureza	22
2.3.3	Colocação	22
2.3.4	Média Harmônica entre Pureza e Colocação	23
2.4	Reconhecimento de Objetos Invariantes	23
3	DESENVOLVIMENTO	24
3.1	Bases de Dados	24
3.1.1	Bases de Dados do Tipo Grafo	24
3.1.2	Bases de Dados para Agrupamento de Objetos Invariantes	25
3.2	Pré-processamento das Bases de Dados	26
3.3	Algoritmo para a Geração da Rede	26
3.4	Aplicação dos Algoritmos de Deteccção de Comunidades	27
3.5	Implementação das Medidas de Avaliação	28
4	RESULTADO PARA BASES DE DADOS DO TIPO GRAFO	29
4.1	Número de Comunidades	29
4.2	Modularidade	29
4.3	Pureza	30
4.4	Colocação	30
4.5	Média Harmônica	31
5	RESULTADOS PARA O AGRUPAMENTO DE OBJETOS INVARIANTES	32

5.1	Escolha da Medida de Proximidade para Construção dos Grafos . .	32
5.2	Número de Comunidades	33
5.3	Modularidade	34
5.4	Informação Mútua Normalizada - NMI	35
5.5	Pureza	35
5.6	Colocação	36
5.7	Média Harmônica	36
5.8	Comparação com o k-means	37
6	CONCLUSÕES	39
	REFERÊNCIAS	40

1 Introdução

Com o avanço computacional e tecnológico, temos grande acesso à informação. Ao mesmo tempo, mais informação é gerada e armazenada em bases de dados. A crescente evolução dos dispositivos de armazenamento de dados traz consigo um grande desafio, extrair informações relevantes desse grande amontoado de dados coletados. Dentre as principais áreas de pesquisa que lidam com esse desafio estão o Aprendizado de Máquina e, Mineração de Dados.

A Mineração de Dados tem como objetivo encontrar tendências e padrões consistentes em bases de dados para extrair conhecimentos específicos com a utilização de métodos e técnicas derivadas principalmente do Aprendizado de Máquina. O Aprendizado de Máquina é uma área de pesquisa relacionada ao desenvolvimento de técnicas computacionais que visa a criação de sistemas capazes de extrair conhecimento dos dados aprendendo determinados padrões ou comportamentos através de experiências adquiridas, gerando assim sistemas cada vez mais inteligentes e autônomos ([CARNEIRO, 2016](#); [BISHOP, 2006](#)).

O agrupamento é uma tarefa comum em Mineração de Dados e tem por objetivo encontrar grupos de dados caracterizados por algum critério de similaridade. Tais grupos podem trazer informações sobre padrões ocultos nos dados, auxiliando no processo de descoberta do conhecimento ([BERKHIN et al., 2006](#)). Apesar dos dados serem considerados na forma de vetor de atributos pela maioria dos algoritmos de agrupamento, recentemente o estudo da formação de grupos em redes se tornou um tópico de intensa pesquisa na área, especialmente quando utilizando redes complexas ([FORTUNATO, 2010](#)). Redes complexas são uma forma de representação de sistemas reais por meio de grafos, utilizada em diversas áreas do conhecimento como Sociologia, Biologia e Computação.

Em redes complexas, comunidade é uma estrutura bastante investigada. Comunidades representam a organização da rede em grupos, onde elementos de um mesmo grupo possuem comportamentos ou desempenham funções semelhantes. O processo de determinar as comunidades em uma rede é chamado de Detecção de Comunidades. Como vantagem em comparação com as abordagens mais usadas de agrupamento, os métodos de detecção de comunidades examinam também as conexões da rede e não apenas os atributos físicos dos dados (por exemplo, distância ou similaridade). Além disso, estes métodos normalmente formam as comunidades sem estabelecer um número fixo de grupos a priori ([FORTUNATO, 2010](#)).

A tarefa de agrupamento (ou Detecção de Comunidades) elimina os custos no processo de rotulação das bases de dados, além de facilitar a aplicação em problemas reais.

Por isso, ela tem sido utilizada em várias aplicações como exploração de dados científicos, mineração de textos, recuperação de informações, diagnósticos médicos, análises Web, entre outros (FORTUNATO, 2010). Uma das possíveis aplicações da tarefa de agrupamento é no reconhecimento de padrões invariantes explicado com mais detalhes na seção 2.4. O problema proposto neste trabalho consiste em dado um conjunto de imagens de objetos fotografados em ângulos ou posições diferentes, ter algoritmos capazes de detectar e agrupar as imagens relacionadas a cada objeto.

O objetivo geral deste trabalho é aplicar técnicas de detecção de comunidades no reconhecimento de objetos invariantes, a fim de obter resultados relevantes para a solução do problema. Assim, a nossa hipótese é que pelas suas próprias características, tais técnicas podem prover melhores resultados do que algoritmos convencionais, como o k-means, no problema de agrupamento de objetos invariantes. Os objetivos específicos deste trabalho são:

- investigar os métodos de detecção de comunidades presentes na literatura;
- desenvolver um ambiente computacional para avaliar o desempenho dos diferentes métodos de detecção de comunidades selecionados;
- comparar os resultados dos algoritmos de detecção de comunidades com o algoritmo de agrupamento tradicional k-means;
- demonstrar empiricamente que os algoritmos de detecção de comunidades podem ser eficientes para encontrar grupos mesmo em bases de dados do tipo vetor de atributos.

Como resultado, espera-se que a investigação de métodos de detecção de comunidades aplicados ao problema de reconhecimento de objetos invariantes possa oferecer novos conceitos para a divisão de grupos e que tais resultados sejam significativos em comparação com as abordagens mais usadas na literatura.

O restante desta dissertação é dividido nos seguintes capítulos:

- Capítulo 2 com a revisão dos principais temas: agrupamento de dados, detecção de comunidades em redes complexas, medidas de avaliação de agrupamentos e reconhecimento de objetos invariantes.
- Capítulo 3 descreve todo o desenvolvimento deste trabalho, coleta e tratamento das bases de dados, aplicação dos algoritmos de detecção de comunidades e medidas de avaliação.
- Capítulo 4 discute os resultados obtidos para as bases de dados do tipo grafo.

-
- Capítulo 5 discute os resultados obtidos para o agrupamento de objetos invariantes.
 - Capítulo 6 apresenta as conclusões do trabalho e sugestões para trabalhos futuros.

2 Revisão Bibliográfica

Neste capítulo é apresentado uma revisão da literatura abordando os principais assuntos envolvidos neste trabalho. Na seção 2.1 é falado sobre a tarefa de agrupamento de dados e apresentado o algoritmo k-means. Na seção 2.2 é discutido sobre Detecção de Comunidades em Redes complexas, Modularidade e Algoritmos de Detecção de Comunidades. A seção 2.3 apresenta as medidas de avaliação de agrupamento. Por fim, na seção 2.4 falamos sobre o problema de reconhecimento de objetos invariantes.

2.1 Agrupamento de Dados

A Mineração de Dados lida com o desafio da descoberta de conhecimento em grandes bases de dados e tem como objetivo encontrar tendências e padrões consistentes para extrair conhecimentos específicos com a utilização de métodos e técnicas derivados principalmente do Aprendizado de Máquina. Entre as tarefas mais comuns realizadas para extrair tal conhecimento estão a classificação e o agrupamento.

Na *classificação de dados* é fornecido para o algoritmo de aprendizado um conjunto de amostras contendo um vetor com os dados de entrada, o algoritmo deve fornecer um vetor com os dados de saída correspondentes. A principal característica é a presença de um supervisor, o qual possui os valores de rótulos de saída corretos para cada valor de entrada. No *agrupamento de dados* as amostras não possuem rótulos pré-definidos, então o algoritmo tenta agrupá-los de alguma maneira, utilizando algum critério de similaridade, formando grupos ou *clusters* (CARNEIRO, 2016; BISHOP, 2006). A Figura 1 apresenta uma ilustração das principais diferenças entre as duas tarefas de aprendizado. Observa-se que na classificação de dados todas as amostras possuem um rótulo pré-definido enquanto que no agrupamento de dados não.

Métodos tradicionais de agrupamento de dados levam em consideração as características físicas dos dados para calcular o grau de similaridade entre os objetos de um conjunto de dados. Algumas das medidas utilizadas para calcular a (dis)similaridade são a Distância Euclidiana e Distância de Manhattan.

2.1.1 Algoritmo de Agrupamento k-means

O algoritmo de agrupamento mais conhecido é o *k-means* (WU et al., 2008) onde os elementos de um conjunto são divididos em k grupos seguindo estes passos:

1. Seleciona-se aleatoriamente k elementos do conjunto para ser os pontos centrais de

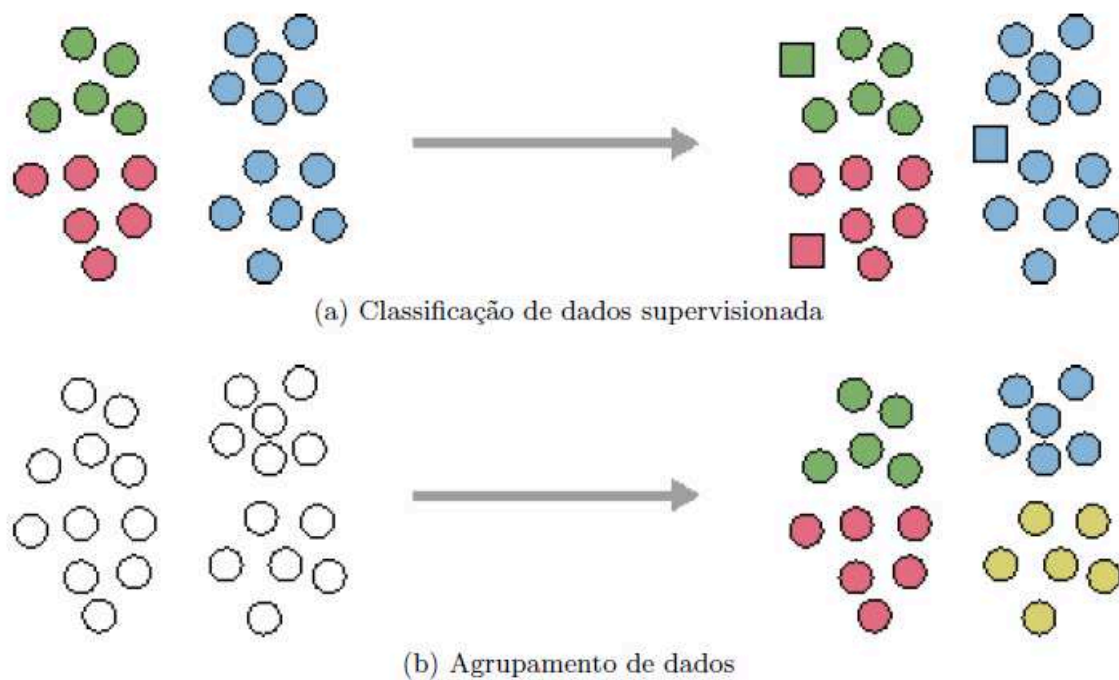


Figura 1 – Ilustração da (a) classificação de dados e do (b) agrupamento de dados. Retirado e adaptado de (CARNEIRO, 2016).

cada grupo, denominados centróides.

2. Calcula-se a distância entre cada objeto do conjunto de dados para os centróides definidos no passo anterior.
3. Atribui-se cada objeto ao grupo do centróide mais próximo.
4. Recalcula-se os centróides realizando a média das coordenadas dos objetos do grupo.
5. Repete-se os passos 2, 3 e 4 até que não haja mais mudança nos centróides.

O *k-means* é um método tradicional de fácil compreensão e implementação porém é limitado ao fato de ter que definir um número de grupos a priori, ser sensível a *outliers* e permitir a geração de grupos vazios. (WU et al., 2008)

2.2 Detecção de Comunidades em Redes Complexas

Qualquer situação onde temos objetos e relações entre eles, podem ser representados por meio de uma rede. Uma rede é um grafo onde os vértices representam os objetos de dados e as arestas interligando os vértices representam as relações existentes entre eles. Sistemas reais representados por grafos recebem a denominação de Redes complexas e são utilizadas em diversas áreas do conhecimento como Sociologia, Biologia e Computação.

Sua denominação se dá por possuir características não triviais, nem completamente regular, nem completamente aleatória (ALBERT; BARABÁSI, 2002; FORTUNATO, 2010; CARNEIRO, 2016).

Exemplos de sistemas reais presentes no nosso cotidiano que podem ser representados por Redes Complexas são as Redes Sociais onde indivíduos estabelecem relações de amizade entre si, podemos considerar cada indivíduo sendo um vértice do grafo e as relações de amizade entre eles sendo arestas entre os vértices; e a Internet onde temos roteadores e canais de comunicação que transmitem informações de um roteador a outro, consideramos roteadores como vértices e os canais de comunicação como arestas interligando os vértices. As redes de Distribuição de Energia Elétrica também podem ser representadas por uma Rede Complexa (ALBERT; BARABÁSI, 2002; FORTUNATO, 2010). A Figura 2 ilustra dois exemplos de problemas reais representados por Redes Complexas.

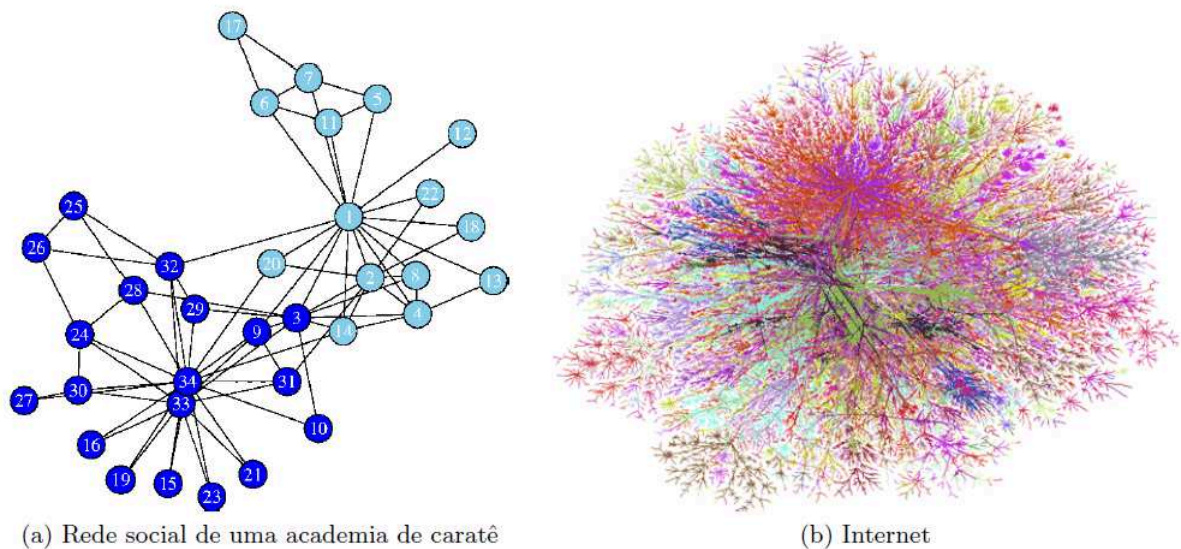


Figura 2 – Exemplos de problemas reais estudados em Redes Complexas. Retiradas de (CARNEIRO, 2016) .

Em redes complexas, a estrutura de comunidade é uma das características mais investigadas. Comunidades representam a organização de vértices em grupos, os quais podem ser identificados pela existência de muitas arestas ligando vértices de um mesmo grupo e poucas arestas ligando vértices de grupos diferentes. O processo de determinar as comunidades de uma rede complexa é chamado de Detecção de Comunidades (FORTUNATO, 2010; ARRUDA; COSTA; RODRIGUES, 2012).

Comunidades são grupos de vértices que provavelmente possuem propriedades comuns ou desempenham funções similares na rede. Um exemplo real de aplicação de Detecção de Comunidades pode ser o agrupamento de clientes na WEB que possuem interesses em comum e estão geograficamente próximos uns dos outros, onde cada grupo pode ser atendido por um servidor específico. Outro exemplo são sistemas de compras

online, onde agrupar os clientes por semelhança de interesses pode proporcionar a criação de sistemas de recomendações mais eficientes (FORTUNATO, 2010; ARRUDA; COSTA; RODRIGUES, 2012).

Os métodos de Detecção de Comunidades se diferenciam dos métodos de agrupamentos tradicionais por levar em consideração não apenas um critério de similaridade entre os dados, mas também a sua topologia, estrutura ou dinâmica. Tal característica é importante pois permite capturar vários padrões pela análise de funcionalidades e processos operando sobre a rede.

Nos métodos de agrupamento tradicionais, o algoritmo recebe um conjunto com os atributos dos dados e calcula a similaridade dos dados baseado em algumas medidas de distância presentes na literatura. Em Redes Complexas, utilizamos esse conjunto de dados para a formação da rede e aplicamos os métodos de Detecção de Comunidades sobre a rede formada.

2.2.1 Modularidade em Redes Complexas

No estudo de Redes Complexas existem algumas medidas que ajudam a caracterizar a estrutura de redes complexas. Em termos de Detecção de Comunidades, a medida mais conhecida é a *Modularidade*. Tal medida, proposta por Newman e Girvan (NEWMAN; GIRVAN, 2004), está relacionada a qualidade de uma determinada divisão da rede. Considerando que não é esperado uma estrutura de comunidades em um grafo aleatório, a modularidade realiza então uma comparação entre a densidade de arestas dentro das comunidades e a densidade esperada em uma rede aleatória. Assume-se que valores altos de modularidade representam uma boa divisão da rede (FORTUNATO, 2010).

A modularidade é definida em (NEWMAN; GIRVAN, 2004) da seguinte maneira. Dado uma rede com K comunidades, definimos uma matriz simétrica $k \times k$ onde cada elemento e_{ij} é a fração das arestas que ligam elementos da comunidade i a elementos da comunidade j dentro da rede. É definido o traço da matriz $Tr e = \sum_i e_{ii}$ sendo a fração das arestas da rede que ligam vértices da mesma comunidade i . Boas divisões de rede esperam um valor alto para o traço, porém o valor alto do traço não significa necessariamente uma boa divisão da rede, por exemplo, considerando um grafo que possui duas comunidades, em uma divisão com apenas uma comunidade o traço apresentaria valor máximo. Então é definida a soma das linhas ou colunas $a_i = \sum_j e_{ij}$ que representa a fração de arestas que conectam vértices na comunidade i . Ou seja, em uma rede onde as arestas são inseridas sem levar em consideração a qual comunidade elas pertencem teríamos $e_{ij} = a_i a_j$. Assim, podemos definir a equação de modularidade como (NEWMAN; GIRVAN, 2004):

$$Q = \sum_i (e_{ii} - a_i^2) = Tr e - \|e^2\| , \quad (2.1)$$

onde $\|e^2\|$ é o somatório dos elementos da matriz e^2 . A Modularidade calcula a fração de arestas que conectam vértices de uma mesma comunidade menos o valor esperado em uma rede com o mesmo número de vértices, porém com conexões aleatórias (NEWMAN; GIRVAN, 2004).

2.2.2 Algoritmos de Detecção de Comunidades

Nesta seção serão descritos alguns algoritmos existentes na literatura que lidam com a tarefa de Detecção de Comunidades em redes complexas.

2.2.2.1 Fastgreedy

O Algoritmo Fastgreedy (NEWMAN, 2004) (CLAUSET; NEWMAN; MOORE, 2004) é baseado na ideia de modularidade. A Modularidade calcula um valor para uma determinada divisão da rede, e quanto maior esse valor, mais bem divididos são os grupos na rede. Se altos valores de modularidade correspondem a uma melhor divisão da rede em comunidades, o algoritmo então gera todas as possíveis divisões da rede, calcula a modularidade para cada uma e encontra a divisão que resulta em um maior valor para a modularidade.

Considerando que as possibilidades de divisão da rede crescem exponencialmente ao número de vértices, ter que encontrar todas as divisões possíveis da rede não seria uma tarefa simples, portanto deve-se utilizar uma heurística para minimizar as divisões geradas. O algoritmo Fastgreedy utiliza a heurística gulosa que é sempre tentar descartar as possibilidades que não geram o melhor resultado.

O algoritmo inicia com cada vértice do grafo sendo uma comunidade e a cada iteração pares de comunidades são unidas. A escolha dos pares é feita de forma a maximizar o valor da modularidade. O processo de união é repetido até que todos os vértices façam parte de uma única comunidade. As uniões obtidas a cada iteração são armazenadas em uma estrutura hierárquica denominada dendrograma. Após todas as execuções, o dendrograma gerado é analisado e se escolhe a divisão que produz o melhor valor para a modularidade. A Figura 3 é uma ilustração de um dendrograma com o corte que representa o valor máximo da modularidade.

2.2.2.2 Labelpropagation

O Algoritmo Labelpropagation (RAGHAVAN; ALBERT; KUMARA, 2007) se baseia na ideia de que cada vértice v possui vizinhos $v1, v2, \dots, vn$ e que cada vizinho possui um rótulo que diz a qual comunidade ele pertence. Então v determina a sua comunidade baseado nos rótulos dos seus vizinhos. Define-se a comunidade de um vértice aquela na qual a maioria dos seus vértices vizinhos pertence.

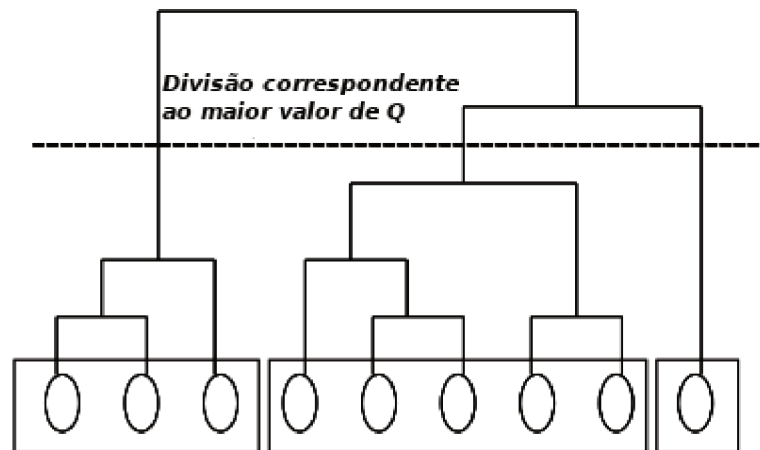


Figura 3 – Ilustração de um dendrograma com o corte que representa o valor máximo da modularidade. (RAGHAVAN; ALBERT; KUMARA, 2007)

Cada vértice v possui um rótulo, que diz a qual comunidade ele pertence, inicialmente, cada vértice faz parte de uma única comunidade, a medida que o algoritmo é executado os rótulos se propagam pela rede. Grupos de vértices densamente conectados alcançam rapidamente um consenso sobre um único rótulo. A cada iteração os vértices são ordenados de forma aleatória, para cada vértice é analisado os rótulos dos seus vizinhos para definir o seu rótulo. O processo de propagação é repetido até que todos os vértices possuam um número de vizinhos pertencentes ao mesmo rótulo maior ou igual ao número de vizinhos pertencentes a rótulos diferentes.

Como o grafo é ordenado de forma aleatória a cada iteração, mesmo com condições iniciais idênticas, obtém-se resultados diferentes. Uma outra complicação é na escolha dos rótulos, pois pode acontecer empates e ter duas opções de rótulos. Para lidar com tais problemas, o algoritmo é normalmente executado várias vezes e considera-se a média de seu desempenho.

2.2.2.3 Walktrap

O Algoritmo Walktrap (PONS; LATAPY, 2005) é baseado na intuição de que a caminhada aleatória em um grafo tende a ficar presa em partes densamente conectadas correspondente às comunidades. O Walktrap usa a ideia de métodos estatísticos que estimam a distância entre dois clusters, mas define sua própria medida de distância baseada em caminhada aleatória e probabilidade.

Dado uma matriz de adjacência $A : A_{ij} = 1$ se os vértices i e j são conectados e $A_{ij} = 0$ caso contrário. O grau $d(i) = \sum_j A_{ij}$ do vértice i é o número dos seu vizinhos (incluindo ele mesmo).

Vamos considerar um processo de caminhada aleatória discreta ou processo de difusão no grafo G . Em cada passo um caminhante, isto é, um vértice se move para o outro vértice escolhido aleatoriamente dentre os seus vizinhos. A sequência de vértices visitados é uma cadeia de Markov, cujos estados são os vértices do grafo. Em cada passo a probabilidade de transição do vértice i para o vértice j é $P_{ij} = \frac{A_{ij}}{d(i)}$. Isso define a matriz de transição P do processo de caminhada aleatória. Também pode-se escrever $P = D^{-1}A$ onde D é a matriz diagonal dos graus ($\forall_i, D_{ii} = d(i)$ and $D_{ij} = 0$ for $i \neq j$).

A probabilidade de ir do vértice i para o vértice j através de uma caminhada aleatória de tamanho t é P_{ij}^t .

Se dois vértices i e j estão na mesma comunidade, então a probabilidade $P^t(i, j)$ certamente será alta. Mas não podemos afirmar o contrário, o fato de $P^t(i, j)$ ser alto não necessariamente implica que i e j estão na mesma comunidade.

Temos i e j sendo dois vértices de um grafo. (PONS; LATAPY, 2005) definem a distância entre eles por:

$$r_t(i, j) = \sqrt{\sum_{k=1}^n \frac{(P_{ik}^t - P_{jk}^t)^2}{d(k)}} \quad (2.2)$$

A probabilidade de uma caminhada aleatória de um vértice na comunidade C alcançar um vértice j em t passos pode ser definida por:

$$P^t C_j = \frac{1}{|C|} \sum_{i \in C} P_{ij}^t \quad (2.3)$$

Temos C_1 e C_2 sendo duas comunidades, (PONS; LATAPY, 2005) definem a distância entre essas duas comunidades por:

$$r_{C_1 C_2} = \sqrt{\sum_{k=1}^n \frac{(P_{C_1 k}^t - P_{C_2 k}^t)^2}{d(k)}} \quad (2.4)$$

Inicialmente temos uma partição $P_1 = v, v \in V$ do grafo onde cada vértice pertence a uma comunidade. Então é computado a distância entre todos os vértices adjacentes e repetido os seguintes passos:

- Escolher duas comunidades C_1 e C_2 em P_k baseado no critério de distância entre comunidades;
- Unir essas duas comunidades em uma nova comunidade $C_3 = C_1 \cup C_2$ e criar a nova partição $P_{k+1} = (P_k \setminus \{C_1, C_2\}) \cup \{C_3\}$;
- Alterar a distância entre comunidades.

A escolha das comunidades para se unirem desempenha um papel fundamental na qualidade das comunidades. A cada passo, une-se as duas comunidades que minimizam a média das distâncias ao quadrado σ_k entre elas:

$$\sigma_k = \frac{1}{n} \sum_{C \in P_k} \sum_{i \in C} r_{iC}^2 \quad (2.5)$$

2.3 Medidas de Avaliação de Agrupamento

Como mencionado anteriormente, a medida de modularidade pode ser utilizada para caracterizar a divisão dos grupos encontrados para dados representados em rede. Outras medidas podem ser utilizadas para analisar a qualidade dos grupos encontrados, independente do tipo de representação. Essas medidas desempenham papel importante no estudo, pois permitem comparar os resultados obtidos pelos algoritmos de Detecção de Comunidades com algoritmos convencionais, como o k-means por exemplo.

2.3.1 Informação Mútua Normalizada

A Informação Mútua Normalizada ou NMI é uma medida da teoria da informação ([LANCICHINETTI; FORTUNATO, 2009](#)) e pode ser definida por:

$$NMI(\Omega, C) = \frac{I(\Omega, C)}{[H(\Omega) + H(C)]/2} \quad (2.6)$$

onde $\Omega = w_1, w_2, \dots, w_k$ é o conjunto de grupos e $C = c_1, c_2, \dots, c_k$ é o conjunto de categorias. A informação mútua $I(\Omega, C)$ nos diz o quanto aprendemos sobre Ω se conhecemos C e vice-versa, pode ser definida por:

$$I(\Omega, C) = \sum_k \sum_j P(w_k \cap c_j) \log \frac{P(w_k \cap c_j)}{P(w_k)P(c_j)} \quad (2.7)$$

$$= \sum_k \sum_j \frac{|w_k \cap c_j|}{N} \log \frac{N|w_k| \cap |c_j|}{|w_k||c_j|} \quad (2.8)$$

onde $P(w_k)$, $P(c_j)$ e $P(w_k \cap c_j)$ são respectivamente a probabilidade do item pertencer ao cluster k , a categoria c e a intersecção de w_k e c_j , respectivamente.

A entropia $H(\Omega)$ é definida por:

$$H(\Omega) = - \sum_k P(w_k) \log P(w_k) \quad (2.9)$$

$$= - \sum_k \frac{w_k}{N} \log \frac{w_k}{N} \quad (2.10)$$

$I(\Omega, C)$ terá valor 0 se o agrupamento for aleatório em relação às categorias. Neste caso, saber que um item está em determinado cluster não nos fornece nenhuma nova informação sobre sua categoria. E será máximo se recriar perfeitamente as categorias ou se $K = N$, ou seja, cada item está em um cluster. A normalização pelo denominador $[H(\Omega) + H(C)]/2$ na equação (2.6) corrige este problema pois a entropia tende a aumentar com o número de clusters. Por exemplo, $H(\Omega)$ alcança seu log máximo para $K = N$ o que garante que o NMI seja baixo para $K = N$. Como o NMI é normalizado podemos usá-lo para comparar os agrupamentos com diferentes números de clusters.

2.3.2 Pureza

A Pureza concentra-se na frequência da categoria mais comum em cada cluster, ou seja, para cada cluster verificamos a categoria mais frequente. Dado $\Omega = \{w_1, w_2, \dots, w_k\}$ sendo o conjunto de clusters a serem avaliados, $C = \{c_1, c_2, \dots, c_j\}$ o conjunto de categorias dos itens e N o número de itens agrupados (AMIGÓ et al., 2009), a pureza pode ser escrita como:

$$PU(\Omega, C) = \frac{1}{N} \sum_k \max_j (w_k \cap c_j) \quad (2.11)$$

A pureza penaliza itens de diferentes categorias em um mesmo cluster, mas não recompensa os itens de uma mesma categoria agrupadas em um mesmo cluster, portanto pode ser máxima quando cada elemento pertencer a um único grupo (AMIGÓ et al., 2009), o que é uma situação indesejada.

2.3.3 Colocação

A colocação se concentra no cluster com mais elementos de uma categoria, ou seja, para cada categoria, verificamos o cluster que possui o maior número de itens (AMIGÓ et al., 2009). A colocação pode ser definida como:

$$CO(\Omega, C) = \frac{1}{N} \sum_j \max_k (w_k \cap c_j) \quad (2.12)$$

A colocação pode ser considerada o inverso da pureza, pois recompensa quando itens de uma mesma categoria são agrupados juntos mas não penaliza quando duas ou mais categorias se misturam. Portanto a colocação pode ser máxima quando todos os itens estiverem agrupados em um mesmo cluster (AMIGÓ et al., 2009), o que é uma situação indesejada.

2.3.4 Média Harmônica entre Pureza e Colocação

Neste trabalho, também é utilizada a média harmônica entre Pureza e Colocação. Tal medida tende a priorizar a maximização conjunta da Pureza e da Colocação, de modo a produzir resultados ruins para agrupamentos que priorizam apenas uma das medidas (AMIGÓ et al., 2009). A média harmônica pode ser definida por:

$$F = \frac{2 \cdot CO \cdot PU}{CO + PU} \quad (2.13)$$

2.4 Reconhecimento de Objetos Invariantes

O reconhecimento de objetos é uma atividade que realizamos constantemente. No trânsito, reconhecemos os automóveis, pedestres, placas de sinalização, animais. Para nós seres humanos trata-se de uma tarefa realizada de forma inconsciente através de todas as experiências que adquirimos ao longo dos anos. Na computação trata-se de uma tarefa um pouco mais complexa, um exemplo onde precisamos de algoritmos capazes de reconhecer objetos pode ser o uso de radares para fiscalização de trânsito. Quando você passa por um radar, é tirado uma foto do veículo, as informações do veículo podem ser acessadas através da sua placa. Como buscamos sempre a automatização dos processos, não desejamos precisar de uma pessoa pra ficar analisando as imagens capturadas pelo radar e acessando as informações dos veículos. Portanto esse é um exemplo onde precisamos de algoritmos capazes de realizar o reconhecimento de objetos e extrair as informações referentes a placa do veículo (LEIBE; SCHIELE, 2003).

A tarefa de Reconhecimento de Objetos Invariantes consiste em dado um conjunto de imagens de objetos fotografados em rotações, translações e escalas diferentes, ter algoritmos capazes de detectar e agrupar as imagens relacionadas a cada objeto. Métodos tradicionais de reconhecimento de objetos invariantes normalmente analisam características físicas dos dados (similaridades ou distância) extraídas a partir da cor, textura, contorno e forma (LEIBE; SCHIELE, 2003). Neste trabalho propomos uma nova abordagem para a solução do problema utilizando redes complexas.

3 Desenvolvimento

Este capítulo descreve as principais etapas para o desenvolvimento deste trabalho, desde a seleção, coleta e tratamento das bases de dados utilizadas, passando pelo processo de construção da rede e de aplicação dos algoritmos de Detecção de Comunidades até as medidas de avaliação usadas para análise e interpretação dos resultados.

3.1 Bases de Dados

Para análise dos algoritmos de Detecção de Comunidades, selecionamos dois conjuntos de bases de dados. O primeiro conjunto compreende vários problemas de domínios distintos cujos dados já estão representados na forma de grafo, enquanto o segundo conjunto corresponde ao problema de agrupamento de objetos invariantes estando os dados representados na forma de imagens.

3.1.1 Bases de Dados do Tipo Grafo

Para o conjunto de bases de dados do tipo grafo, foram coletadas as bases Dolphins, Football, Karate e Polbooks, as quais podem ser encontradas em ([NEWMAN, 2016](#)) e são brevemente descritas a seguir:

- A base Dolphins é uma rede social de uma população de 62 golfinhos-nariz-de-garrafa que residem no fiorde Doubtful Sound, Nova Zelândia. A rede é não direcionada e existem 159 ligações entre os 62 golfinhos. As ligações foram feitas de acordo com os golfinhos que foram vistos juntos com mais frequência, analisados durante um período de 7 anos ([LUSSEAU et al., 2003](#)).
- A base Football é a representação do calendário de Jogos de Futebol Americano da Divisão 1 para a temporada do ano 2000. Os 115 nós são as equipes identificadas pelo nome das faculdades e as 615 arestas representam os jogos de temporada regular que acontece entre as duas equipes. O objetivo da rede é separar os times em conferências, onde os jogos são mais frequentes entre times de uma mesma conferência ([GIRVAN; NEWMAN, 2002](#)).
- A base Karate é uma rede social dos integrantes de um clube de karate em uma Universidade dos Estados Unidos. São 34 nós representando os integrantes e 78 arestas representando as relações de amizade entre eles em atividades fora do clube. O objetivo da rede Karate é encontrar os grupos em que essa rede ficou dividida

após uma briga entre o instrutor e o presidente do clube, onde o instrutor abriu um novo clube e levou com ele alguns membros do clube original (ZACHARY, 1977).

- A base Polbooks é uma representação dos livros sobre política dos estados unidos vendidos pela Amazon.com durante a corrida presidencial americana. Possui 105 nós que representam os livros e 441 arestas, uma aresta entre dois livros a e b indica que clientes que compraram o livro a também compraram o livro b . O objetivo da base de dados é agrupar os clientes em liberais, neutros e conservadores baseado nos livros que eles compram (KREBS, 2008).

3.1.2 Bases de Dados para Agrupamento de Objetos Invariantes

Para o conjunto de bases de dados relacionadas ao agrupamento de objetos invariantes, foram coletadas as imagens da base de dados ETH-80 proposta por (LEIBE; SCHIELE, 2003), a qual contém 3280 imagens divididas em 8 categorias: Apple, Pear, Tomato, Cow, Dog, Horse, Cup e Car. Cada categoria contém 10 objetos com grandes variações entre si, mesmo que claramente pertençam a mesma categoria. Cada objeto conta com 41 imagens para representá-lo em condições de rotação, translação e escala diferentes. A Figura 4 apresenta as 8 categorias e os 10 objetos de cada uma delas. A Figura 5 apresenta as imagens que representam dois objetos da categoria maçã da base de dados (CARNEIRO, 2016).



Figura 4 – Exemplo das categorias da base dados ETH-80. Cada categoria possui 10 objetos distintos.

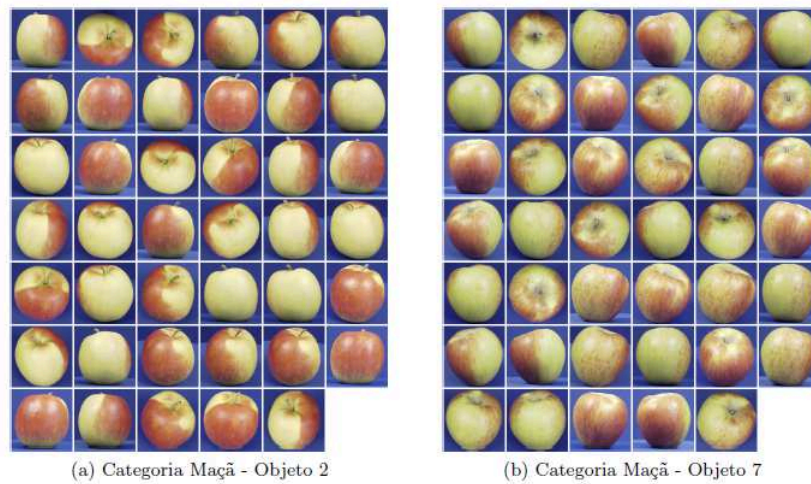


Figura 5 – Exemplo contendo dois objetos da categoria maçã da base ETH-80. (a) Categoria Maçã - Objeto 2 (b) Categoria Maçã - Objeto 7

3.2 Pré-processamento das Bases de Dados

Diferente das bases de dados do tipo grafo, as bases de dados da coleção ETH-80 estão representadas na forma de imagens, ou seja, os algoritmos de Detecção de Comunidades não podem ser aplicados diretamente sobre elas. Nesse sentido, duas etapas são realizadas para tornar viável a descoberta de comunidades em tais bases: a extração dos atributos das imagens e a geração da rede a partir deles.

Para a extração dos atributos, as imagens são redimensionadas de 128x128 pixels para 32x32 a fim de melhorar o desempenho computacional e então é calculado o histograma RGB de cada imagem resultando na representação das imagens em vetores de 512 atributos numéricos. A etapa de transformação dos dados para o tipo grafo é apresentada a seguir.

3.3 Algoritmo para a Geração da Rede

O método utilizado para a formação da rede é o algoritmo *kNN* (*k-nearest neighbors algorithm*) (FORTUNATO, 2010), um algoritmo bem popular na literatura e bem simples de ser implementado. O algoritmo para criação da rede executa os seguintes passos (WU et al., 2008):

1. Define-se um número k de vizinhos.
2. Cada objeto do conjunto de dados é construído como um nó da rede.
3. Calcula a similaridade entre todos os pares de objetos do conjunto de dados.

4. Para cada objeto do conjunto de dados seleciona os k objetos mais próximos a ele e cria uma ligação entre eles na rede.

Em relação aos dados da coleção ETH-80, após a transformação das imagens em vetores de atributos é aplicado o método kNN para a formação da rede para cada categoria da base dados. Assim, cada imagem é um vértice no grafo. O kNN calcula a similaridade dos dados utilizando alguma medida de proximidade e conecta os k vértices mais próximos segundo o resultado de tal cálculo. Duas medidas de proximidade são consideradas neste estudo: a Distância Euclidiana e a Distância de Bhattacharyya.

3.4 Aplicação dos Algoritmos de Detecção de Comunidades

Para a aplicação dos algoritmos utilizou-se a biblioteca *iGraph* disponível para *Python* e outras linguagens. Os algoritmos foram selecionados baseado nas pesquisas realizadas, por serem algoritmos muito utilizados e apresentarem bons resultados.

1. **Fastgreedy** : O Fastgreedy define as comunidades baseado na otimização gulosa da modularidade. Une os nós de uma maneira que maximize a modularidade do grafo. Diz-se que o algoritmo funciona quase em tempo linear para grafos esparsos. O Fastgreedy possui o parâmetro *weights*, nome do atributo ou uma lista com o peso das arestas. O algoritmo retorna então um dendrograma cortado com o agrupamento que gera o maior valor de modularidade.
2. **Walktrap** : O Walktrap é um algoritmo baseado em caminhadas aleatórias, a ideia básica é que objetos que se encontram com mais frequência (adjacentes) em uma caminhada aleatória curta tende a permanecer na mesma comunidade. O Walktrap possui os parâmetros *weights*, nome do atributo ou uma lista com o peso das arestas e *steps* tamanho da caminhada aleatória. O algoritmo retorna um dendrograma inicialmente cortado com o maior valor da modularidade.
3. **Labelpropagation** : Inicialmente, cada nó é atribuído a um rótulo, em cada iteração o vértice escolhe o rótulo dominante em sua vizinhança, os casos de empate são resolvidos aleatoriamente. A ordem em que os vértices são atualizados é randomizada em cada iteração. O algoritmo para quando todos os vértices possuem número de vizinhos pertencentes ao mesmo rótulo igual ou maior ao número de vizinhos pertencentes a rótulos diferentes. Como os empates são resolvidos aleatoriamente, cada execução do algoritmo retorna um resultado diferente. Por isso, ele foi executado 10 vezes e calculado a média dos valores obtidos. O Labelpropagation tem como parâmetros: *weights* nome do atributo ou uma lista com o peso das arestas, *initial* nome do atributo ou uma lista contendo os rótulos iniciais dos vértices e

fixed uma lista de booleanos para cada vértice. *True* caso o rótulo do vértice não deva ser mudado durante a execução. Neste caso só faz sentido se os rótulos iniciais forem fornecidos.

3.5 Implementação das Medidas de Avaliação

Foram implementadas em python as seguintes medidas de avaliação dos algoritmos de Detecção de Comunidades: Informação Mútua Normalizada, Pureza, Colocação e Média Harmônica entre Pureza e Colocação.

Após testes preliminares foi analisado que a modularidade e informação mútua normalizada priorizavam agrupamentos com um elevado número de grupos. A fim de priorizar um número reduzido de grupos que facilite a análise dos dados e a extração de informações e que estes sejam puros, ou seja, que as categorias não se misturem entre os grupos, optamos por utilizar a média harmônica entre pureza e colocação. As medidas são discutidas com mais detalhes na seção 2.3.

4 Resultado para Bases de Dados do Tipo Grafo

Neste capítulo apresentamos os resultados obtidos pelos algoritmos de Detecção de Comunidades aplicados em bases de dados reais já representadas através de redes, as quais estão disponíveis na literatura. Para análise do desempenho dos algoritmos de Detecção de Comunidades em tais bases, adotou-se as seguintes medidas de avaliação: número de comunidades, modularidade, pureza, colocação e a média harmônica entre pureza e colocação.

4.1 Número de Comunidades

Na Tabela 1, os números de comunidades obtidos pelos algoritmos para cada base de dados analisada. O Walktrap consegue definir o número de grupos exato conforme o ground-truth para a base Football. O Labelpropagation é o que mais se aproxima do *ground-truth* para todas as bases de dados.

Tabela 1 – Número de Comunidades para Bases de Dados do Tipo Grafo

Categoria	Fastgreedy	Walktrap	Labelpropagation	Ground-truth
Dolphins	4,0	4,0	3,7	3,0
Football	6,0	10,0	10,5	10,0
Karate	3,0	5,0	2,4	2,0
Polbooks	4,0	4,0	3,4	3,0

4.2 Modularidade

A medida modularidade é apresentada na Tabela 2, o algoritmo fastgreedy apresenta o maior resultado para as bases Dolphins e Karate enquanto o walktrap para as bases football e polbooks. O maior valor de modularidade nem sempre representa o melhor agrupamento, mas valores aproximados a 0,4 costumam apresentar bons agrupamentos.

Tabela 2 – Modularidade para Bases de Dados do Tipo Grafo

Categoria	Fastgreedy	Walktrap	Labelpropagation
Dolphins	0,495	0,489	0,465
Football	0,550	0,603	0,592
Karate	0,381	0,353	0,308
Polbooks	0,502	0,507	0,494

4.3 Pureza

A pureza dos grupos, na Tabela 3 apresenta altos valores o que representa agrupamentos bem definidos e separados de acordo com o ground-truth. Para a base Karate o algoritmo Fastgreedy obteve o maior resultado de pureza, enquanto para as bases Football e Polbooks foi o Walktrap e para a base Dolphins o Labelpropagation .

Tabela 3 – Pureza para Bases de Dados do Tipo Grafo

Categoria	Fastgreedy	Walktrap	Labelpropagation
Dolphins	96,8	95,2	98,2
Football	57,4	87,0	84,4
Karate	97,1	94,1	88,8
Polbooks	83,8	84,8	84,5

4.4 Colocação

Os valores obtidos para colocação mostrados na Tabela 4 nos diz que o agrupamento mantém elementos de uma mesma categoria em um mesmo grupo e possui um número de grupos reduzido. O algoritmo Walktrap apresenta o melhor resultado para a base de dados Football enquanto o Labelpropagation para as demais bases.

Tabela 4 – Colocação para Bases de Dados do Tipo Grafo

Categoria	Fastgreedy	Walktrap	Labelpropagation
Dolphins	66,4	66,1	73,7
Football	87,8	92,2	90,8
Karate	73,5	52,9	90,0
Polbooks	82,9	81,9	83,2

4.5 Média Harmônica

Na Tabela 5 é apresentado os valores obtidos pela média harmônica. Como pode ser visto, os valores são altos porque obtemos altos valores tanto para pureza quanto para colocação. Isso nos diz que elementos de uma mesma categoria tendem a estar em um mesmo grupo e em um grupo não há elementos de mais de uma categoria. Para a base Dolphins, o Labelpropagation apresenta melhor pureza e colocação, e se aproxima mais do número ideal de grupos. Para a base Football o Walktrap teve os melhores resultados para pureza e colocação além de melhor modularidade e estar mais próximo do número ideal de grupos. Para a base Karate, o Labelpropagation obteve a melhor colocação e o número de grupos mais próximo do ideal, enquanto o Fastgreedy obteve a melhor pureza. Por fim, para a base Polbooks o Labelpropagation apresentou melhor colocação e número de grupos mais próximo do número ideal, enquanto o Walktrap obteve o melhor valor de pureza.

Tabela 5 – Média Harmônica para Bases de Dados do Tipo Grafo

Categoria	Fastgreedy	Walktrap	Labelpropagation
Dolphins	80,8	78,0	83,3
Football	69,4	89,5	87,4
Karate	83,7	67,8	87,6
Polbooks	83,3	83,3	83,7

5 Resultados para o Agrupamento de Objetos Invariantes

Neste capítulo é apresentado os resultados obtidos através da aplicação dos algoritmos de Detecção de Comunidades aplicados ao problema de reconhecimento de objetos invariantes.

Foram realizadas simulações com algoritmos de Detecção de Comunidades aplicados as bases de dados disponíveis em (LEIBE; SCHIELE, 2003). Por estarem em forma de vetor de atributos foi utilizado a Distância de Bhattacharyya para calcular a similaridade entre os atributos e formar os grafos utilizando o método kNN . Alguns algoritmos de Detecção de Comunidades são métodos estocásticos, portanto foram executados 10 vezes e calculado a média dos resultados obtidos. Para a geração dos grafos variou-se o parâmetro k entre 1 e 30 a fim de obter redes com diferentes características topológicas.

As análises dos resultados obtidos serão realizadas por medidas de desempenho. Na seção 5.1 é apresentado um comparativo entre os resultados obtidos utilizando a Distância Euclidiana e a Distância de Bhattacharyya na geração da rede. As seções 5.2, 5.3, 5.4, 5.5, 5.6 e 5.7 representa a análise de cada uma das medidas obtidas para o agrupamento de objetos invariantes. Na seção 5.8 é realizado um comparativo entre os algoritmos de Detecção de Comunidades e o algoritmo de agrupamento tradicional k-means.

5.1 Escolha da Medida de Proximidade para Construção dos Grafos

A Tabela 6 apresenta um comparativo entre os resultados obtidos utilizando a Distância Euclidiana e a Distância de Bhattacharyya na geração dos grafos. Para o comparativo, utilizamos a média harmônica por priorizar um agrupamento com poucos grupos e a pureza entre eles. De modo geral com o uso da Distância de Bhattacharyya houve uma melhora nos resultados comparado ao uso da Distância Euclidiana. Apenas para a categoria *Tomato* os grafos gerados utilizando a Distância Euclidiana apresentaram melhor resultado, o que pode ser explicado devido ao aspecto da cor dos objetos, que são bastante parecidas e não contribuem para discriminá-los no espaço.

Tabela 6 – Comparativo entre os melhores resultados obtidos pelos algoritmos de Detecção de Comunidades utilizando a Média Harmônica das medidas de Pureza e Colocação.

Base de Dados	Fastgreedy		Walktrap		Labelpropagation	
	Eucl.	Bhat.	Eucl.	Bhat.	Eucl.	Bhat.
Apple	58,0	63,4	55,6	61,6	55,4	62,3
Car	41,7	59,1	37,3	56,2	36,7	49,6
Cow	32,4	43,5	31,1	41,9	29,3	39,6
Cup	46,7	49,8	47,3	53,0	46,9	48,2
Dog	41,5	46,0	39,5	47,2	39,7	42,8
Horse	42,1	43,1	40,2	41,6	37,6	40,9
Pear	33,4	42,7	32,3	41,0	31,4	38,5
Tomato	46,3	43,5	47,8	41,7	44,3	39,2

Devido a superioridade da medida de Bhattacharya, todas as simulações e resultados apresentados daqui em diante consideraram tal medida para a construção do grafo.

5.2 Número de Comunidades

Na Figura 6 são apresentados os gráficos referentes ao número de comunidades obtido para cada base de dados analisada. Na figura, a cor vermelha representa os valores obtidos pelo método Fastgreedy, a cor azul pelo método Walktrap e a cor preta pelo método Labelpropagation. Ainda na figura, o eixo x é a variação do parâmetro k na geração da rede e o eixo y são os valores obtidos pela medida avaliada. Como pode ser observado, os algoritmos apresentam número de comunidades relativamente próximos, especialmente com o aumento de k . De modo geral, o Fastgreedy normalmente apresenta número de comunidades um pouco menor que o Walktrap, que, por sua vez, apresenta número de comunidades menor que o Labelpropagation.

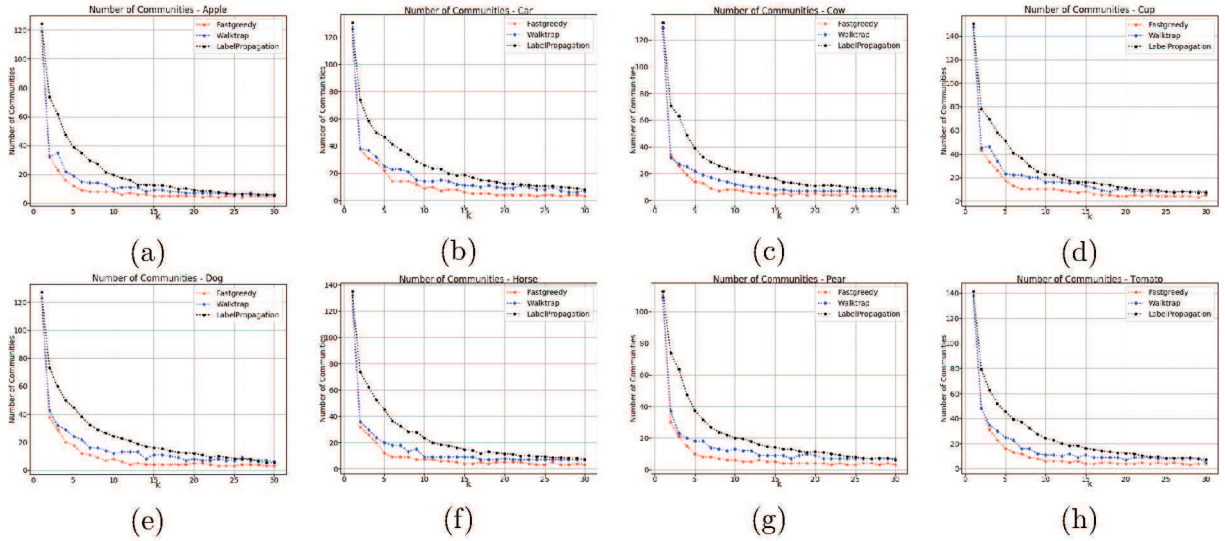


Figura 6 – Número de Comunidades para as bases de dados (a) Apple, (b) Car, (c) Cow, (d) Cup, (e) Dog, (f) Horse, (g) Pear e (h) Tomato.

5.3 Modularidade

A Figura 7 apresenta os valores obtidos para a Modularidade. Com o aumento de k a modularidade diminui e o algoritmo Fastgreedy tende a se distanciar um pouco dos outros. No geral, os algoritmos apresentam valores acima de 0.5 o que sinaliza para a existência de grupos bem definidos, de acordo com várias análises encontradas na literatura.

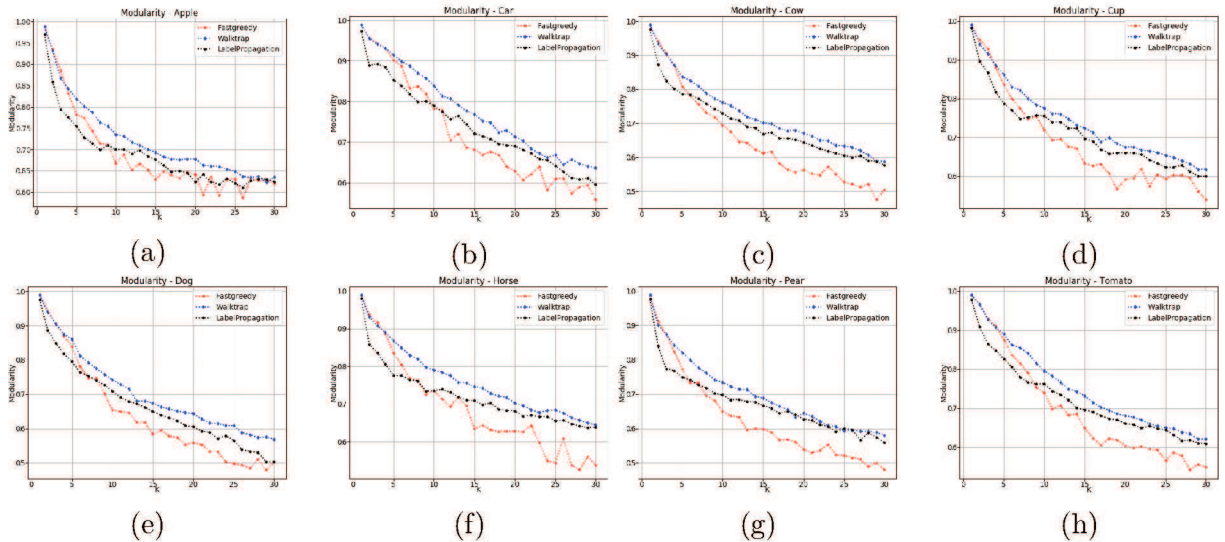


Figura 7 – Modularidade para as bases de dados (a) Apple, (b) Car, (c) Cow, (d) Cup, (e) Dog, (f) Horse, (g) Pear e (h) Tomato.

5.4 Informação Mútua Normalizada - NMI

A Figura 8 mostra os resultados para a medida NMI. Aqui podemos observar que os algoritmos são bem sensíveis a alteração de k , em especial o algoritmo Fastgreedy. A medida que o k aumenta, os valores de NMI diminuem. Isso se dá porque o NMI penaliza grupos que não são puros e a medida que o k aumenta, os números de grupos diminuem e possivelmente existem elementos de mais de uma categoria em um único grupo.

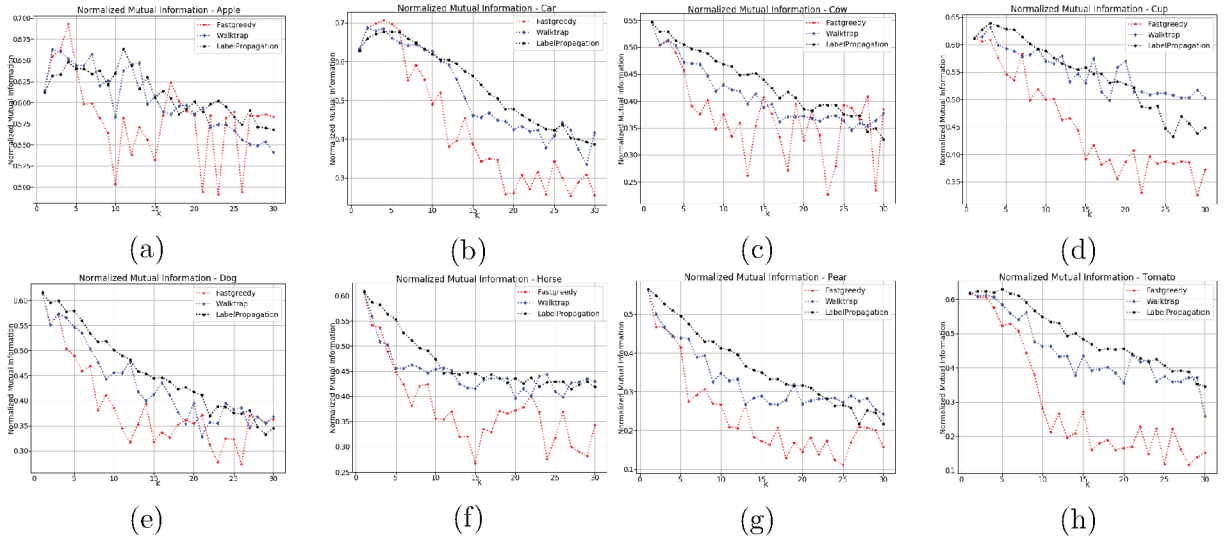


Figura 8 – Informação Mútua Normalizada para as bases de dados (a) Apple, (b) Car, (c) Cow, (d) Cup, (e) Dog, (f) Horse, (g) Pear e (h) Tomato.

5.5 Pureza

Como pode ser observado na Figura 9, com o aumento de k os valores da pureza diminuem, isso se dá pelo fato de que com o aumento de k o número de grupos aumentam e tende-se a ter elementos de mais de uma categoria em um mesmo grupo. Quando o k atinge o valor 15 a queda no valor da pureza se torna mais acentuada.

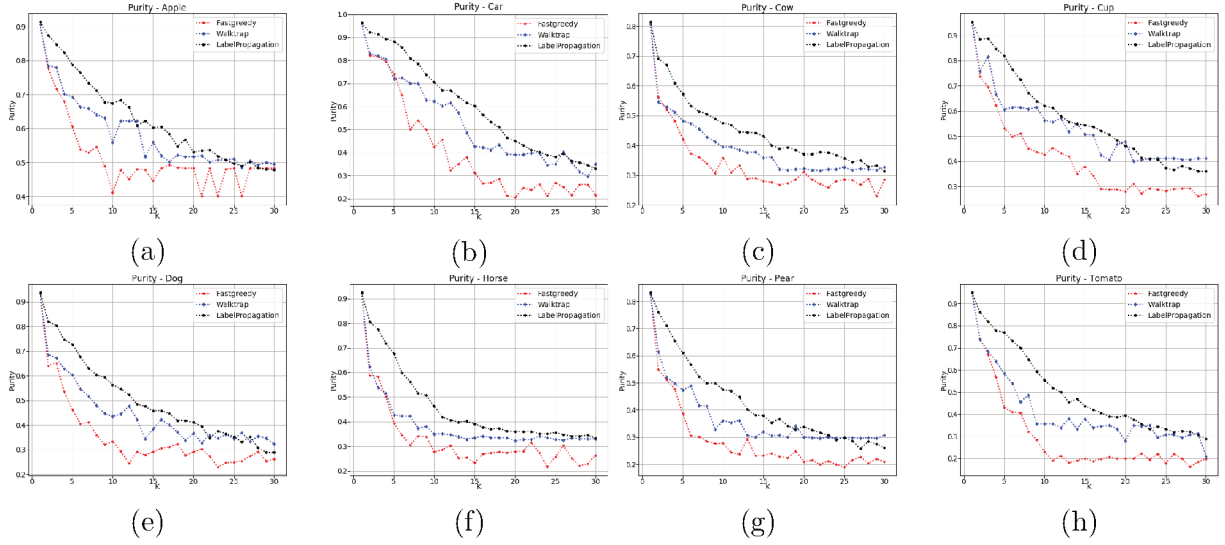


Figura 9 – Pureza para as bases de dados (a) Apple, (b) Car, (c) Cow, (d) Cup, (e) Dog, (f) Horse, (g) Pear e (h) Tomato.

5.6 Colocação

Na Figura 10 observamos os resultados para a colocação que ao contrário da pureza, aumenta conforme o aumento de k . Isso devido ao fato de que a colocação busca agrupamentos com um número reduzido de grupos.

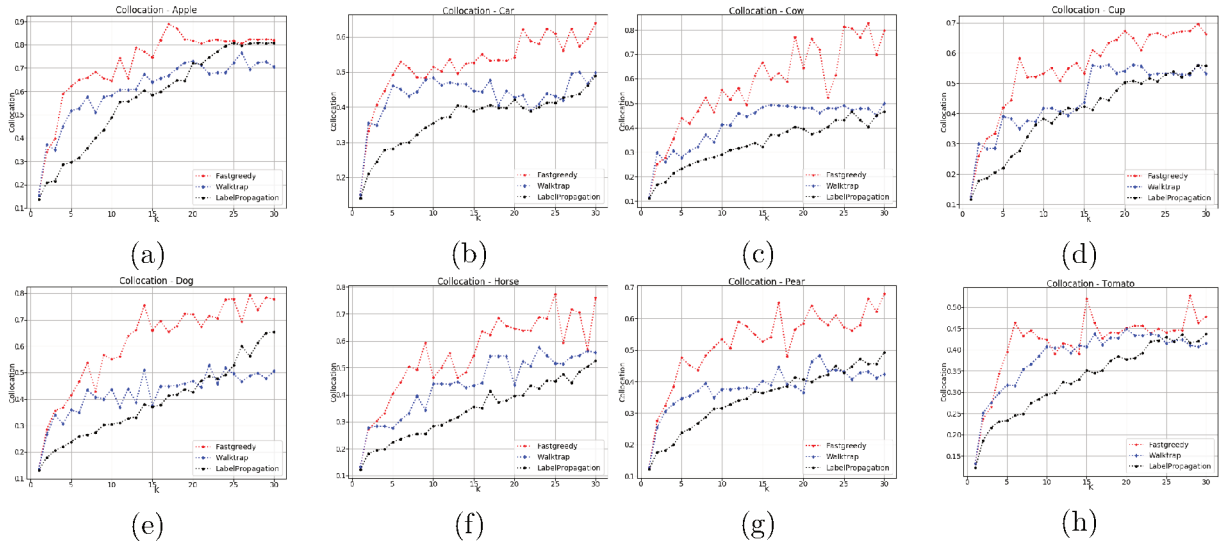


Figura 10 – Colocação para as bases de dados (a) Apple, (b) Car, (c) Cow, (d) Cup, (e) Dog, (f) Horse, (g) Pear e (h) Tomato.

5.7 Média Harmônica

Na Figura 11 é apresentado os resultados da média harmônica. Um bom agrupamento é quando temos um número reduzido de grupos (colocação) e tais grupos são puros

(pureza), portanto não queremos nem um número muito grande de grupos que dificulte a análise dos dados e nem um número muito pequeno onde temos elementos de várias categorias em um mesmo grupo. Utilizamos a média harmônica para balancear os valores de pureza e colocação, a média harmônica tende a ser elevada quando os valores de pureza e colocação são aproximados.

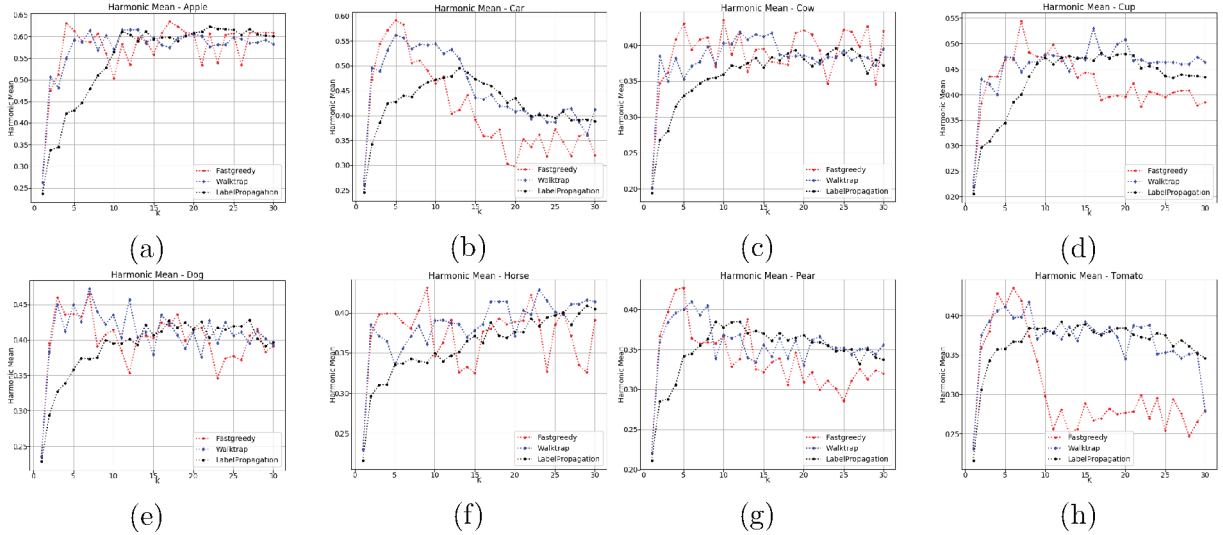


Figura 11 – Média Harmônica para as bases de dados (a) Apple, (b) Car, (c) Cow, (d) Cup, (e) Dog, (f) Horse, (g) Pear e (h) Tomato.

5.8 Comparação com o k-means

É importante fazer a comparação entre Detecção de Comunidades e algoritmos convencionais de agrupamento que são bastante utilizados, como o k-means, como parâmetro. Foi realizado simulações com o k-means sobre as bases de dados variando o parâmetro k de 2 a 100. Devido a sua natureza estocástica o algoritmo foi executado 10 vezes e calculado a média dos resultados obtidos.

Foi escolhido como medida de desempenho a média harmônica entre pureza e colocação, pois tenta-se priorizar configurações de agrupamento normalmente desejáveis: grupos mais puros e em quantidade não exagerada.

Na Tabela 1, é apresentado um comparativo entre os melhores resultados obtidos entre os métodos de Detecção de Comunidades e o método de agrupamento tradicional k-means. O algoritmo de Detecção de Comunidades Fastgreedy apresentou o melhor resultado para as categorias Apple, Car, Cow, Horse, Pear e Tomato. O Walktrap apresentou melhor resultado para as categorias Cup e Dog. Vale ressaltar que os algoritmos de Detecção de Comunidades não exigem um número de grupos previamente definido, embora seja dependente do grafo formado.

Tabela 7 – Comparativo entre os melhores resultados obtidos utilizando a Média Harmônica

Categoria	Fastgreedy	Walktrap	Labelpropagation	k-means
Apple	63,4	61,6	62,3	50,4
Car	59,1	56,2	49,6	38,0
Cow	43,5	41,9	39,6	29,0
Cup	49,8	53,0	48,2	47,5
Dog	46,0	47,2	42,8	40,3
Horse	43,1	41,6	40,9	38,3
Pear	42,7	41,0	38,5	32,8
Tomato	43,5	41,7	39,2	41,9

6 Conclusões

Este trabalho compara os métodos de Detecção de Comunidades aplicados ao problema de agrupamento de objetos invariantes e em bases de dados reais disponíveis na literatura. De modo geral, as medidas de avaliação de grupos utilizadas neste trabalho demonstram que os algoritmos de Detecção de Comunidades conseguem atingir boas divisões da rede.

Os gráficos apresentados nos resultados mostram um bom desempenho por parte dos métodos de Detecção de Comunidades, se saindo melhor que o método tradicional de agrupamento k -means para todas as categorias testadas. Portanto, conclui-se que a utilização de redes complexas se mostra promissora enquanto nova abordagem para o problema de agrupamento de objetos invariantes.

Observa-se que a utilização da Distância de Bhattacharya, muito utilizado no processamento de imagens, gera melhores resultados quando comparado aos resultados obtidos pela Distância Euclidiana.

Para as bases de dados reais disponíveis da literatura e testadas neste trabalho, os algoritmos de Detecção de Comunidades apresentam resultados satisfatórios validando assim o seu bom desempenho em diversos problemas abordados, sejam eles do tipo grafo ou não.

Para trabalhos futuros sugere-se testar outras alternativas para a construção do grafo além do método kNN ; analisar um método automático para seleção de um valor de k apropriado; analisar outras medidas de similaridade além da Distância Euclidiana e Distância de Bhattacharya; realizar a coleta de outras bases de dados relevantes ao problema de reconhecimento de objetos invariantes; comparar os resultados dos algoritmos de Detecção de Comunidades com outros métodos de agrupamento tradicionais além do k -means.

Referências

- ALBERT, R.; BARABÁSI, A.-L. Statistical mechanics of complex networks. *Reviews of modern physics*, APS, v. 74, n. 1, p. 47, 2002. Citado na página 16.
- AMIGÓ, E. et al. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, Springer, v. 12, n. 4, p. 461–486, 2009. Citado 2 vezes nas páginas 22 e 23.
- ARRUDA, G. F. de; COSTA, L. da F.; RODRIGUES, F. A. A complex networks approach for data clustering. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 391, n. 23, p. 6174–6183, 2012. Citado 2 vezes nas páginas 16 e 17.
- BERKHIN, P. et al. A survey of clustering data mining techniques. *Grouping multidimensional data*, Springer, v. 25, p. 71, 2006. Citado na página 11.
- BISHOP, C. M. *Pattern recognition and machine learning*. [S.l.]: springer, 2006. Citado 2 vezes nas páginas 11 e 14.
- CARNEIRO, M. G. *Redes complexas para classificação de dados via conformidade de padrão, caracterização de importância e otimização estrutural*. Tese (Doutorado) — Universidade de São Paulo, 2016. Citado 6 vezes nas páginas 7, 11, 14, 15, 16 e 25.
- CLAUSET, A.; NEWMAN, M. E.; MOORE, C. Finding community structure in very large networks. *Physical review E*, APS, v. 70, n. 6, p. 066111, 2004. Citado na página 18.
- FORTUNATO, S. Community detection in graphs. *Physics reports*, Elsevier, v. 486, n. 3, p. 75–174, 2010. Citado 5 vezes nas páginas 11, 12, 16, 17 e 26.
- GIRVAN, M.; NEWMAN, M. E. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, National Acad Sciences, v. 99, n. 12, p. 7821–7826, 2002. Citado na página 24.
- KREBS, V. Political books network. 2008. Citado na página 25.
- LANCICHINETTI, A.; FORTUNATO, S. Community detection algorithms: a comparative analysis. *Physical review E*, APS, v. 80, n. 5, p. 056117, 2009. Citado na página 21.
- LEIBE, B.; SCHIELE, B. Analyzing appearance and contour based methods for object categorization. In: IEEE. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. [S.l.], 2003. v. 2, p. II–409. Citado 3 vezes nas páginas 23, 25 e 32.
- LUSSEAU, D. et al. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, Springer, v. 54, n. 4, p. 396–405, 2003. Citado na página 24.
- NEWMAN, M. *Network data*. 2016. Disponível em: <<http://www-personal.umich.edu/~mejn/netdata/>>. Acesso em: 14 dec. 2018. Citado na página 24.

NEWMAN, M. E. Fast algorithm for detecting community structure in networks. *Physical review E*, APS, v. 69, n. 6, p. 066133, 2004. Citado na página 18.

NEWMAN, M. E.; GIRVAN, M. Finding and evaluating community structure in networks. *Physical review E*, APS, v. 69, n. 2, p. 026113, 2004. Citado 2 vezes nas páginas 17 e 18.

PONS, P.; LATAPY, M. Computing communities in large networks using random walks. In: SPRINGER. *International symposium on computer and information sciences*. [S.l.], 2005. p. 284–293. Citado 2 vezes nas páginas 19 e 20.

RAGHAVAN, U. N.; ALBERT, R.; KUMARA, S. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, APS, v. 76, n. 3, p. 036106, 2007. Citado 3 vezes nas páginas 7, 18 e 19.

WU, X. et al. Top 10 algorithms in data mining. *Knowledge and information systems*, Springer, v. 14, n. 1, p. 1–37, 2008. Citado 3 vezes nas páginas 14, 15 e 26.

ZACHARY, W. W. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, University of New Mexico, v. 33, n. 4, p. 452–473, 1977. Citado na página 25.