UNIVERSIDADE FEDERAL DE SÃO CARLOS
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

Paulo Henrique Dal Bello

Avaliação de métodos de construção de redes e detecção de comunidades no agrupamento de textos

São Carlos - SP
2022

Paulo Henrique Dal Bello

Avaliação de métodos de construção de redes e detecção de comunidades no agrupamento de textos

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Engenharia de Computação da Universidade Federal de São Carlos, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientação Prof. Dr. Alan Demétrius Baria Valejo

São Carlos - SP
2022

Dedico este trabalho a todos que acompanharam minha caminhada e a fizeram mais leve.

Agradecimentos

Primeiramente agradeço aos meus pais, Emília e Antônio, pelo amor e confiança em mim depositados.

Agradeço a minha irmã, Érika, e meu cunhado, Roberto, pelas longas conversas sobre a área acadêmica.

Agradeço também a meu orientador, prof. Alan, pela sugestão de tema, oportunidade cedida e ao apoio dado à construção deste trabalho.

Agradeço aos colegas que fiz por tornarem a jornada acadêmica muito mais divertida.

E por fim, agradeço a minha namorada, Luma, pelo companheirismo, paciência e todo carinho.

Quando se espera que a máquina seja infalível, ela não poderá ser também inteligente.
(Alan Turing)

Resumo

Devido à grande quantidade de dados produzidos diariamente no formato de texto, seja publicamente em redes sociais ou de forma privada dentro de empresas, há a necessidade de analisá-los e extrair deles informação. O objetivo é transformá-los em ferramentas úteis, como sistemas de tradução e assistentes virtuais. A área de Processamento de Linguagem Natural, em conjunto com o Aprendizado de Máquina, fornece as tecnologias necessárias para tal objetivo. Uma tarefa muito explorada nesse contexto é o agrupamento de documentos por meio de classificação não supervisionada. Grupos de documentos podem fornecer uma descrição dos assuntos abordados por uma coleção de documentos, representando, em geral, categorias ou temas. Considerando essa tarefa, além dos algoritmos tradicionais de agrupamento, como o k-Means, as abordagens baseadas em redes vem ganhando notoriedade na literatura, as quais constroem uma rede a partir da coleção de documento e utilizam detecção de comunidades para encontrar grupos de documentos que representem temas similares. Essas abordagens necessitam, inicialmente, da construção de uma rede a partir dos documentos analisados, sendo que diversos algoritmos podem ser utilizados para esse propósito, os quais produzem redes com características topológicas distintas, interferindo diretamente na qualidade do agrupamento. Nesse contexto, o objetivo deste estudo é analisar a influência dos algoritmos de construção de redes no agrupamento de textos. Busca-se avaliar se as diferentes formas de se construir redes podem influenciar na geração de estruturas de comunidades que sejam representativas considerando as classes dos documentos de textos.

Palavras-chave: Aprendizado de Máquina; Aprendizado Não Supervisionado; Agrupamento; Detecção de Comunidades; Redes.

Abstract

Due to the large amount of data produced daily in text format, whether publicly on social networks or privately within companies, there is a need to analyze and extract information from them. The goal is to turn them into useful tools, such as translation systems and virtual assistants. The area of Natural Language Processing, in conjunction with Machine Learning, provides the necessary technologies for such an objective. One of the most explored tasks in this context is the clustering of documents through unsupervised classification. Document clusters can provide a description of the subjects covered by a collection of documents, representing, in general, categories or themes. Considering this task, in addition to the traditional clustering algorithms, such as k-Means, approaches based on networks have been gaining notoriety in the literature, which build a network from the document collection and use community detection to find groups of documents representing similar themes. These approaches initially need the construction of a network from the documents analyzed, and several algorithms can be used for this purpose, which produce networks with distinct topological characteristics, directly interfering with the quality of the cluster. In this context, the aim of this study is to analyze the influence of network construction algorithms in the clustering of texts. It seeks to assess whether the different ways of building networks can influence the generation of community structures that are representative considering the classes of text documents.

Keywords: Machine Learning; Unsupervised Learning; Clustering; Community Detection; Networks.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Os avanços computacionais ocorridos nas últimas décadas e a popularização da internet democratizaram o acesso à informação, com isso, a quantidade de informação disponível tornou-se imensa e crescente a cada dia. Grande parcela desses dados é disponibilizada na forma de textos, o que gerou o desafio de extrair e processar toda essa informação.

O Processamento de Linguagem Natural (PLN), subárea da ciência da computação, surgiu com o objetivo de lidar com esse desafio, buscando criar ferramentas que proporcionem a devida extração de informação a partir de línguas humanas naturais. Dentre as aplicações de PLN, é possível citar os *chatbots*, programas capazes de manter conversas com usuários humanos em linguagem natural, os assistentes virtuais, *softwares* que recebem comandos em linguagem natural e realizam tarefas (como a Siri da Apple, Google Assistente, Alexa da Amazon, entre outros) e a classificação automática de textos, que promove a identificação de similaridades com o objetivo de gerar grupos de documentos, que podem representar temas ou categorias.

A classificação automática de textos tem sido vastamente explorada dentro da área de PLN, visto que a quantidade de dados textuais cresce de maneira exponencial com o passar do tempo (postagens em redes sociais, artigos científicos, textos publicados em portais de notícia e blogs pessoais, informação em abundância e constantemente gerada). Para cumprir tal tarefa, são utilizados algoritmos de Aprendizado de Máquina (AM), subárea da inteligência artificial, que, com pouca interação humana, são capazes de moldar soluções baseadas em padrões identificados nas bases de dados. Existem três categorias para os algoritmos de aprendizado de máquina, sendo elas: não supervisionados, supervisionados e semi-supervisionados. No contexto de dados textuais, os não supervisionados, conhecidos como algoritmos de agrupamento, inferem a divisão de documentos em diferentes grupos com base em padrões e similaridades identificadas em dados não rotulados. Os supervisionados utilizam-se de dados rotulados a fim de treinar um modelo que seja capaz de classificar novos documentos. Por fim, os semi-supervisionados consideram dados rotulados e não rotulados para a classificação de novos documentos.

Recentemente, técnicas de agrupamento baseadas em redes têm ganhado notoriedade, apresentando resultados interessantes ao comparadas a outros algoritmos mais tradicionais da literatura, como o k-Means (ARTHUR; VASSILVITSKII, 2007) e o DBScan (ESTER et al., 1996). Redes podem ser definidas como estruturas relacionais cujos objetos são dadas por vértices e as relações entre eles por arestas. Sendo assim, diversas estruturas possuem representações em forma de redes, como redes de computadores, de telefonia, de contatos, dentre outras.

Para a utilização de algoritmos de agrupamento baseados em redes, é preciso primeiramente que a base de dados seja transformada em uma rede, que, por sua vez, terá seus vértices particionados em comunidades através de um algoritmo de Detecção de Comunidades. Em seguida, as comunidades serão projetadas em forma de grupos no conjunto de dados original. É possível citar como vantagem dos algoritmos de detecção de comunidades a possibilidade de se estudar a estrutura topológica gerada a partir da transformação dos dados em uma rede, além de permitir a utilização de todo conhecimento teórico presente na Teoria das Redes Complexas.

Tendo em vista que, para a utilização de algoritmos de agrupamento baseados em redes, é necessária a construção de uma rede a partir da base de dados, diferentes algoritmos de construção de redes podem ser usados, cada qual com uma estratégia própria e que resultam em redes com características topológicas diferentes. Portanto, há a necessidade da existência de pesquisas que busquem analisar os diferentes algoritmos de construção de redes e verificar sua influência sobre os métodos de detecção de comunidades, além de avaliar a performance destas técnicas de agrupamento ao comparadas a um algoritmo tradicional da literatura, como o k-Means.

1.1 Objetivos

1.1.1 Objetivo Geral

O presente trabalho tem como objetivo analisar a influência dos algoritmos de construção de redes no agrupamento de textos. Busca-se avaliar se as diferentes formas de se construir redes podem influenciar na geração de estruturas de comunidades que sejam representativas considerando as classes dos documentos de textos e como essas técnicas se compariariam com algoritmos de agrupamento mais usuais, utilizando, para tanto, o k-Means como linha de base comparativa.

1.1.2 Objetivos Específicos

Os objetivos específicos podem ser desmembrados nos seguintes:

- Estudar o estado da arte para algoritmos de construção de redes e detecção de comunidades no âmbito de agrupamento de documentos textuais;
- Organizar uma base de dados em forma de documentos de texto sobre diferentes temas, processada com técnicas de pré-processamento de textos;
- Gerar redes a partir dos dados através de algoritmos de construção de redes diversos;

- Avaliar os algoritmos de detecção de comunidades executados sobre as redes previamente geradas;
- Analisar os resultados obtidos, comparando-os com os resultados obtidos nas mesmas bases utilizando um algoritmo de agrupamento tradicional.

1.2 Organização do Trabalho

O trabalho em questão está organizado da seguinte forma:

- No Capítulo 1 é feita a descrição do problema e dos objetivos gerais e específicos da pesquisa;
- No Capítulo 2 ocorre a fundamentação teórica do estado arte sobre aprendizado de máquina, algoritmos de agrupamento, processamento de linguagem natural, algoritmos de construção de redes e de detecção de comunidades;
- No Capítulo 3 define-se as metodologias empregadas na pesquisa, informando detalhes sobre os conjuntos de dados e algoritmos escolhidos, bem como da modelagem e ferramentas de análise empregadas;
- No Capítulo 4 há a discussão dos experimentos e seus resultados;
- Finalmente, no Capítulo 5, são apresentadas as conclusões obtidas no trabalho e as sugestões para pesquisas futuras.

2 Fundamentação Teórica

Neste Capítulo serão apresentados os conceitos de inteligência artificial e aprendizado de máquina com foco nos algoritmos de agrupamento, detecção de comunidades e nas medidas de avaliação escolhidas para a análise dos resultados da presente pesquisa.

2.1 Inteligência Artificial

O termo Inteligência Artificial foi cunhado em 1956 por John McCarthy, que define que o objetivo da área é criar máquinas capazes de emular comportamentos humanos e raciocinar de forma lógica.

Em (ERTEL, 2018), Ertel define que, entre as principais qualidades dos seres humanos, que os definem como seres inteligentes, está a capacidade de adaptação e habilidade em ajustar-se a diferentes ambientes, aprendendo com o meio e, assim, mudando seu comportamento de acordo com a situação.

Como discutido em (BREWKA, 1996), em parte dos estudos feitos na área de IA o objetivo é entender mais sobre a espécie humana, compreendo o funcionamento de seu cérebro, para que seja possível replicar em máquinas as ações tidas como inteligentes.

Espera-se que técnicas de IA sejam capazes de solucionar problemas específicos de maneira mais eficaz, utilizando-se dessa sugerida capacidade de adaptação. Nesta visão, o objetivo é alcançar a solução ótima (ou a melhor possível) para um dado problema, sem que haja a definição prévia de um método específico. É válido notar que, portanto, as soluções baseadas em IA não são universais, e acabam por apresentar especificidades em cada problema aplicado.

2.2 Aprendizado de Máquina

O Aprendizado de Máquina (AM) é uma subárea da IA que estuda o desenvolvimento de ferramentas capazes de extrair informação a partir da experiência com dados, seguindo, essencialmente, o conceito humano de “aprendizagem”. Essa aprendizagem ocorre através de algoritmos estatísticos que identificam padrões em conjuntos de dados. Atualmente, como categorias de técnicas de AM pode-se citar o aprendizado indutivo, o aprendizado transdutivo e o aprendizado por reforço.

As técnicas de AM aplicadas neste trabalho são baseadas na ideia de aprendizado indutivo, que trata-se da obtenção de conclusões genéricas a partir de conjuntos de exemplos ou casos anteriormente observados. Em indução, ocorre a aprendizagem de um conceito através da

inferência indutiva feita sobre os exemplos apresentados, sendo assim, as hipóteses geradas através desta inferência podem representar ou não a verdade (MONARD; BARANAUSKAS, 2003).

Existem duas principais categorias dentro do aprendizado indutivo, que são: Aprendizado Supervisionado e Aprendizado Não Supervisionado. A Figura 1 apresenta a hierarquia do aprendizado indutivo.

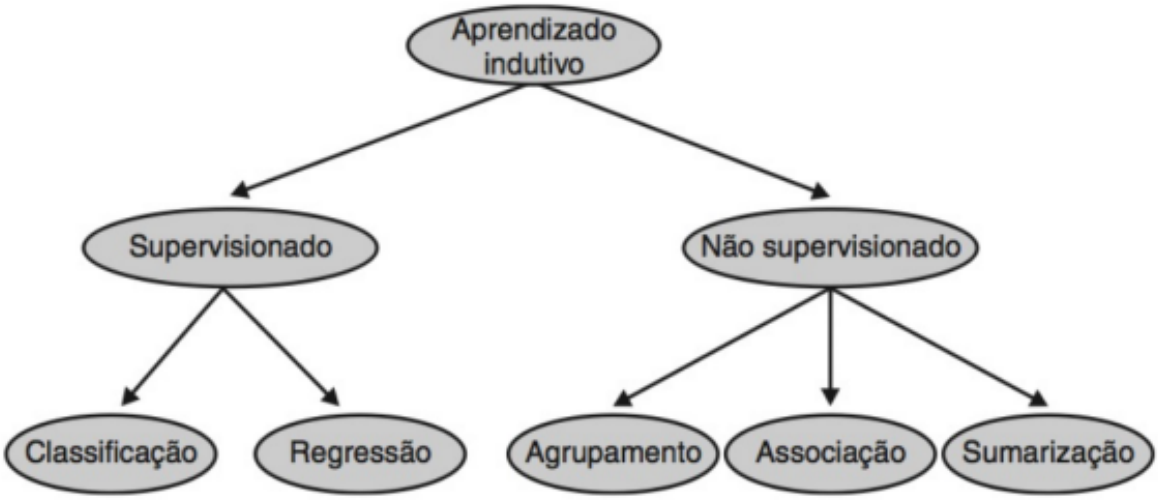


Figura 1 – Hierarquia do aprendizado indutivo (FACELI et al., 2011).

2.2.1 Aprendizado de Máquina Supervisionado

O Aprendizado de Máquina Supervisionado consiste na utilização de dados rotulados, ou seja, com suas classes ou valores conhecidos, para o treinamento de um modelo, que deverá em seguida ser capaz de generalizar esse rótulo para novas amostras (KOTSIANTIS et al., 2007). Nesta abordagem de AM, há a presença de um "professor externo", ou seja, um fator que sugere ao algoritmo o conhecimento do ambiente através de exemplos: uma entrada e a saída esperada para aquela dada entrada (HAYKIN; NETWORK, 2004).

Os principais tipos de algoritmos de Aprendizado Supervisionado são os de classificação e de regressão. Enquanto na classificação o resultado do algoritmo é uma classe, ou seja, uma categoria, na regressão o resultado esperado é um valor contínuo. Portanto, a escolha do método está diretamente relacionada ao escopo do problema. Pensando, por exemplo, em um modelo de previsão de precipitação: um classificador pode prever se haverá ou não chuva dadas determinadas característica do clima, enquanto um regressor pode prever o volume de chuva dadas essas mesmas características.

2.2.2 Aprendizado de Máquina Não Supervisionado

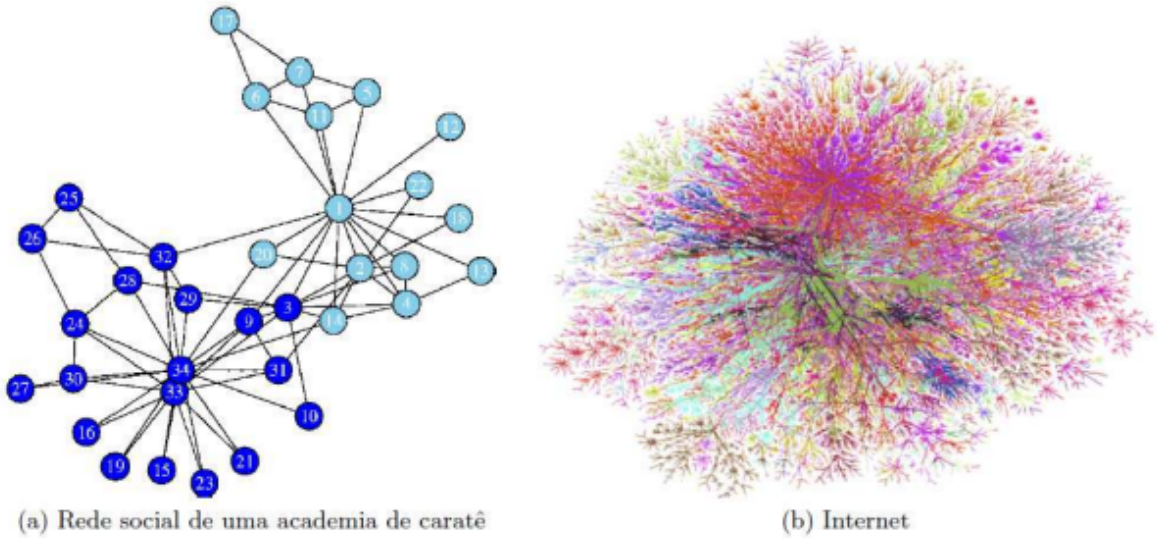
Existem casos em que as bases de dados escolhidas para análise não estão rotuladas, sendo assim, outra abordagem precisa ser considerada. Neste contexto a utilização de métodos de Aprendizado de Máquina Não Supervisionado pode ser a melhor solução.

Os algoritmos de Aprendizado de Máquina Não Supervisionado não trabalham com exemplos de saídas, não há um conjunto predefinido de soluções para o problema. Portanto, os algoritmos visam representar ou agrupar as entradas baseados em padrões ou semelhanças presentes nos dados.

Neste trabalho, o método de aprendizado abordado é o não-supervisionado, com destaque em algoritmos de agrupamento e detecção de comunidades, que serão futuramente abordados neste Capítulo.

2.3 Redes

Sempre que um problema apresenta objetos com relações entre eles, ele pode ser descrito através de redes. Uma rede é um grafo com vértices representando os objetos e arestas representando as relações entre os objetos. Quando um sistema complexo é representado em forma de grafos ele é denominado uma Rede Complexa, essa denominação se dá por suas características não triviais: não há regularidade total e nem completa aleatoriedade (ALBERT; BARABÁSI, 2002). A Figura 2 mostra exemplos de sistemas complexos modelados na forma de Redes Complexas.



(a) Rede social de uma academia de caratê (b) Internet

Figura 2 – Sistemas complexos modelados através de Redes Complexas (CARNEIRO, 2016).

Dada uma rede $G(V, E)$, que representa um conjunto $V = v_1, v_2, ..., v_n$ finito, seus n elementos são chamados vértices, enquanto o conjunto $E = e_1, e_2, ..., e_m$ é composto pelas arestas, portanto uma aresta que conecta os vértices v_i e $v_j \in V$ é descrita por $e_{i,j}$. Uma rede não direcionada em que cada par de vértices está conectado por apenas uma aresta (ou seja, não há arestas paralelas) é dita simples.

Em uma rede ponderada, $G(V, E, W)$ descreve um conjunto de vértices V, um conjunto de arestas E e uma matriz de peso W, com w sendo a função que atribui um peso $w(e_{i,j})$ para cada aresta $e_{i,j} \in E$.

Diz-se que dois vértices são adjacentes quando estão ligados por uma aresta. A matriz de adjacência A de uma rede é dada pela matriz quadrada de ordem n, com $n = |V|$, que tem a aresta entre o vértices v_i e v_j representada por $a_{i,j}$

Uma rede pode ser representada também através de uma lista de adjacência, formada pelo vetor *adj* de n listas, uma para cada vértice de V. A Figura 3 mostra uma rede simples e suas representações por matriz e lista de adjacência.

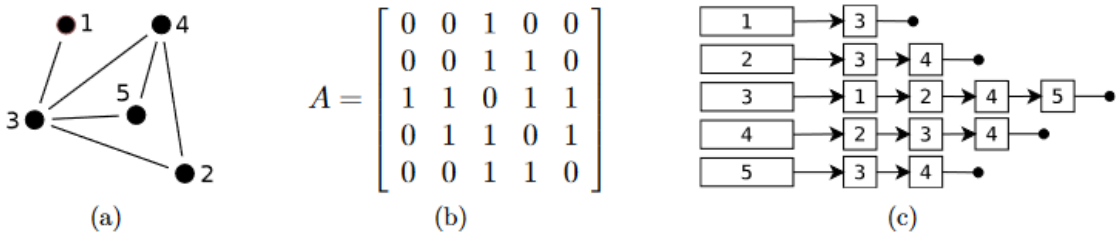


Figura 3 – Representação da rede (a) através de uma Matriz de Adjacência (b) e de uma Lista de Adjacência (c) (VALEJO, 2014).

2.3.1 Algoritmos de Construção de Redes

Alguns sistemas, devido às suas estruturas relacionais, apresentam-se naturalmente modelados em formas de redes. No entanto, em outros casos, os dados podem estar definidos como vetores de características ou até mesmo na forma de atributo-valor. Neste contexto, é possível a utilização de algoritmos capazes de construir redes, o que será abordado nesta Seção.

2.3.1.1 k-NN

A sigla k-NN origina-se do inglês *k-nearest neighbors*, em tradução, k-vizinhos mais próximos. Este algoritmo tem como objetivo a divisão das amostras com base em suas semelhanças (GUO et al., 2003).

Dado um conjunto de amostras $X = \{x_1, \ldots, x_n\}$, o vizinho mais próximo de x_i, considerando-se $i \neq j$, é aquele x_j que apresentar a menor proximidade a x_i, podendo ser aplicados diferentes cálculos de proximidade, tais como Manhattan, Euclidiana ou Cosseno. Vizinhos mais próximos compartilham um maior número de similaridades, com base em tais características, o algoritmo k-NN gera uma rede que conecta, através de uma aresta, os k vizinhos mais próximos.

Sendo assim, o algoritmo k-NN pode ser descrito através dos seguintes passos (WU et al., 2008):

1. É definido o número k de vizinhos.
2. Cada amostra dos dados é construída na forma de um vértice da rede.
3. É calculada a similaridade entre todos os pares de objetos possíveis.
4. Para cada vértice, o algoritmo seleciona os k outros objetos mais próximos e os liga através de arestas.

Um exemplo de redes construídas a partir do algoritmo k-NN com diferentes valores atribuídos para k é mostrado na Figura 4.

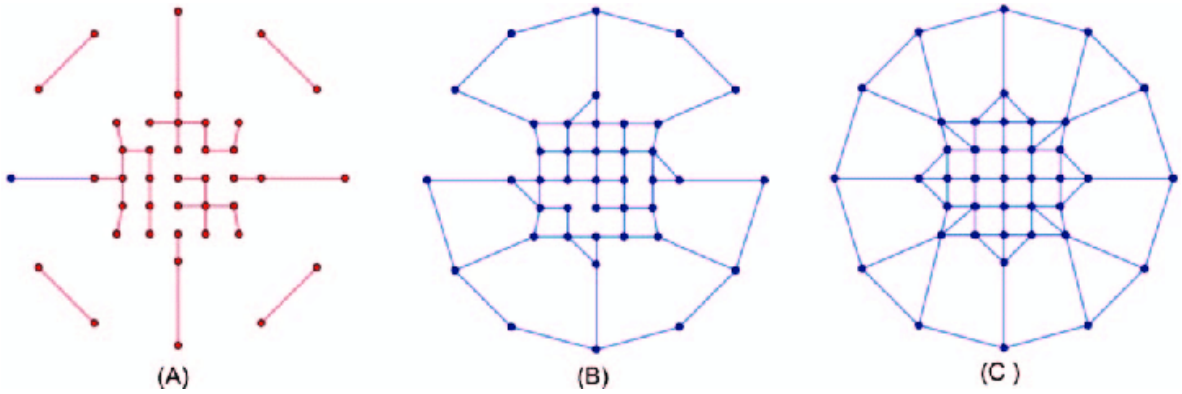


Figura 4 – Redes construídas com o algoritmo k-NN para $k = 1$ (A), $k = 2$ (B) e $k = 3$ (C) (BERTON; LOPES, 2012).

2.3.1.2 Mk-NN

O Mk-NN (*mutual k nearest neighbors*) é uma adaptação do k-NN clássico. Nesta abordagem, dois vértices v_i e v_j só se conectam caso ambos sejam mutuamente vizinhos mais próximos. Apesar de gerar redes mais esparsas, o número de conexões de cada vértice se limita pelo valor k, o que impede que a rede apresente vértices com números muito altos de conexões. Isto reduz a influencia de dados ruidosos dentro da rede, isolando os *outliers* dos

demais vértices, o que em certos cenários gera uma vantagem deste método em relação ao k-NN.

A Figura 5 apresenta a rede construída a partir de um dado conjunto de dados com a utilização do Mk-NN com $k = 11$.

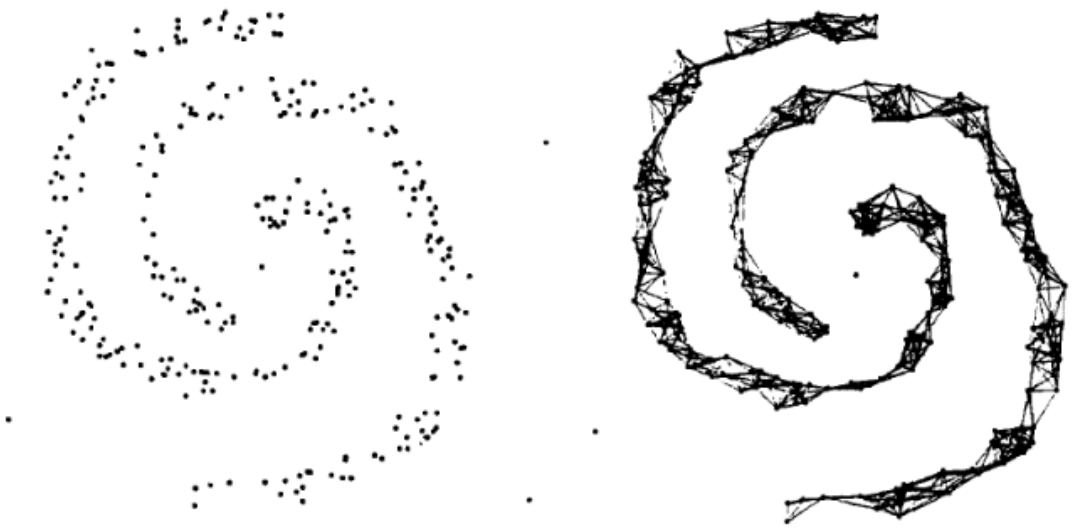


Figura 5 – Rede construída com o algoritmo Mk-NN (BRITO et al., 1997).

2.3.1.3 Sk-NN

O algoritmo Sk-NN (*Sequential k-NN*) consiste na criação de conexões entre os nós de forma incremental, iniciando com $k = 1$ até um definido k_{max}. Após computar os k vizinhos próximos para os vértices e ordená-los através de um critério de relevância, toma-se os vértices do vetor ordenado e tenta-se conectá-los aos k_{max} vizinhos mais próximos que possuem um grau inferior a k, ou seja, com menos de k conexões. Caso não seja possível, o valor de k é incrementado e o processo, repetido. O algoritmo termina quando todos os vértices possuírem um grau maior ou igual a k_{max} (VEGA-OLIVEROS et al., 2014).

O Sk-NN não gera redes perfeitamente regulares, ou seja, com o mesmo número de conexões para todos os vértices, no entanto, tem como objetivo impedir a existência de vértices com graus muito acima da média da rede, os chamados *hubs*. Situação que pode ocorrer em redes geradas pelo algoritmo k-NN e Mk-NN.

Redes altamente regulares podem ser ineficientes na resolução de alguns problemas, como no exemplo de bases textuais que geram redes com conexões entre palavras, visto que coleções de textos geralmente apresentam muitas palavras com baixa frequência de aparição e poucas com alta frequência.

A Figura 6 exibe uma comparação entre redes geradas a partir do algoritmo Sk-NN e k-NN para um mesmo conjunto de dados.

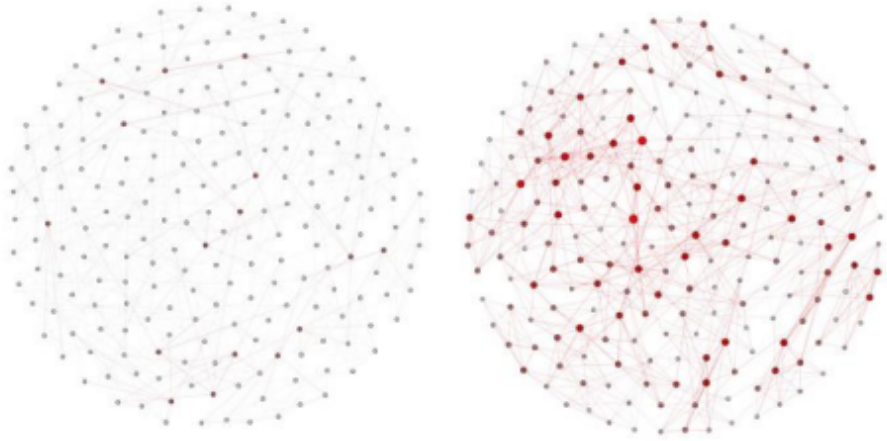


Figura 6 – Rede construída construída através do algoritmo Sk-NN, à esquerda, e rede construída a partir do algoritmo k-NN.(VEGA-OLIVEROS et al., 2014).

2.3.1.4 E-Vizinhança

O algoritmo E-Vizinhança, ou Epsilon, tem como premissa a criação de uma conexão entre os vértices x_i e x_j se $|(x_i, x_j)| \leq \epsilon$, com $|(x_i, x_j)|$ sendo a proximidade entre os objetos e ϵ um limite previamente estabelecido. Ou seja, dado determinado objeto, há uma circunferência de raio ϵ centrada nele e todos os demais objetos localizados dentro desta circunferência serão conectados ao vértice central por meio de arestas. Nota-se que em caso de um valor baixo de ϵ, muitos vértices podem se manter isolados na rede, requerendo um tratamento posterior. Ao mesmo tempo, se ϵ for alto, a rede gerada possivelmente será muito densa. A Figura 7 apresenta exemplos destes comportamentos.

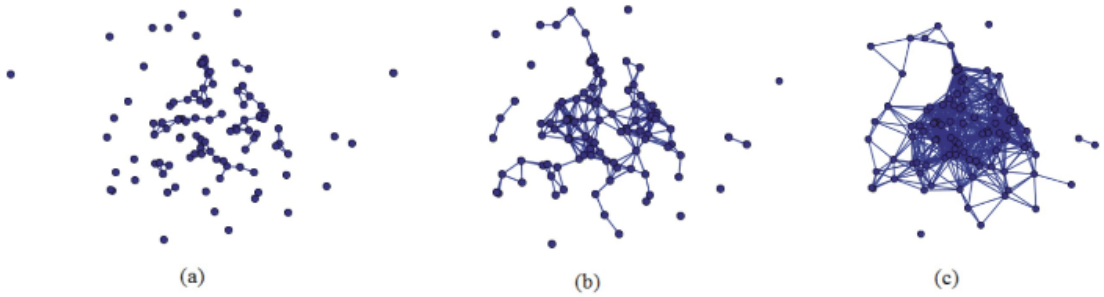


Figura 7 – Exemplo de redes E-Vizinhança para um conjunto de dados com 100 elementos e distribuição gaussiana (a) $E = 0.3$, (b) $E = 0.5$ e (c) $E = 0.9$ (BERTON, 2016).

Portanto, devido a esses resultados, e a dificuldade em se obter o melhor valor o parâmetro ϵ, as redes geradas pelo algoritmo E-Vizinhança são mais complexas de serem utilizadas. Esse algoritmo pode resultar em componentes desconexos, não sendo adequado para muitas situações (BELKIN; NIYOGI, 2003).

2.4 Algoritmos de Agrupamento

Dentro da área de Mineração de Dados há o desafio da extração de informação a partir de conjuntos grandes de dados, objetivando a obtenção de padrões específicos através do uso de técnicas derivadas do Aprendizado de Máquina. Os Algoritmos de Agrupamento fazem parte do ferramental disponível para a realização dessa tarefa.

No agrupamento de dados, o conjunto de amostras disponível não está rotulado, portanto o algoritmo busca agrupá-los através de parâmetros de similaridade, construindo, assim, grupos com características semelhantes referentes ao critério de similaridade definido.

2.4.1 k-Means

O algoritmo mais popular de agrupamento é o k-Means, que divide os objetos de um dado conjunto em k grupos através do seguinte passo a passo (WU et al., 2008):

1. k elementos são selecionados como centroides de cada grupo, podendo ser definidos aleatoriamente ou através de um critério definido;
2. É feito o cálculo da proximidade entre os objetos do conjunto até os centroides;
3. Cada objeto é atribuído ao grupo que possuir o centroide mais próximo;
4. Os centroides são redefinidos pela média das coordenadas dos objetos que compõem o grupo;
5. Os passos 2, 3 e 4 são repetidos até que não haja mais alteração nas coordenadas dos centroides.

2.4.2 Algoritmos de Detecção de Comunidades

No contexto de redes complexas, as comunidades são definidas como a organização de conjuntos de vértices em grupos. Este grupos são identificados pela presença de grande número de arestas conectando seus vértices, o que indica que provavelmente apresentam características comuns ou desempenham funções semelhantes na rede. Detecção de Comunidades é o nome dado ao processo de identificar estruturas de comunidades em uma rede (FORTUNATO, 2010).

Os algoritmos de Detecção de Comunidades diferenciam-se dos métodos tradicionais de agrupamentos pois levam em consideração não apenas um critério de similaridade entre os dados, como também a sua topologia, estrutura ou dinâmica. Esta característica é importante já que permite a identificação de vários padrões pela análise de funcionalidades e processos operando sobre a rede.

Nos agrupamentos tradicionais, o algoritmo recebe os atributos dos dados e calcula as suas similaridades com base em alguma medida de proximidade presente na literatura. Em Redes Complexas, utiliza-se esse conjunto de dados para a formação da rede e, em seguida, aplica-se métodos de Detecção de Comunidades sobre a rede gerada.

2.4.2.1 Fastgreedy

O Algoritmo Fastgreedy se baseia na ideia de Modularidade. A Modularidade calcula um valor para para uma divisão específica da rede, sendo que quanto maior este valor, melhor divididos são os grupos presentes nela. Um valor mais alto de Modularidade corresponde a um melhor particionamento da rede em comunidades, portanto, o algoritmo gera todas as possíveis divisões, calcula a Modularidade de cada uma e encontra a divisão com o maior valor de modularidade (RAGHAVAN; ALBERT; KUMARA, 2007).

No entanto, como as possibilidades de divisão da rede apresentam um crescimento exponencial ao número de vértices, identificar todas as divisões possíveis não é uma tarefa simples (CLAUSET; NEWMAN; MOORE, 2004). Sendo assim, utiliza-se uma heurística a fim de reduzir as divisões geradas. O Fastgreedy faz uso da heurística gulosa, ou seja, procura descarta as possibilidades que não gerarão o melhor resultado.

Inicia-se o algoritmo com cada vértice pertencente a uma comunidade, então, em cada iteração, pares de comunidades são unidos. A seleção dos pares é definida de forma a maximizar a Modularidade. Este processo se repete até que todos os vértices forme uma única comunidade. Armazena-se as uniões obtidas a cada iteração em uma estrutura hierárquica chamada dendograma. Ao final de todas as execuções, o dendograma resultante é analisado e é feita a escolha da divisão que leva ao melhor valor de de Modularidade.

A Figura 8 ilustra a representação de um dendograma e o corte que sugere o valor máximo de Modularidade.

2.4.2.2 Walktrap

O Algoritmo Walktrap se baseia na ideia de que a caminhada aleatória em um grafo tende a se prender em regiões densamente conectadas, que correspondem às comunidades (PONS; LATAPY, 2005). Sedo assim, o Walktrap utiliza métodos estatísticos que estimam a proximidade entre dois grupos, mas define sua medida de distância com base em caminhada aleatória e probabilidade.

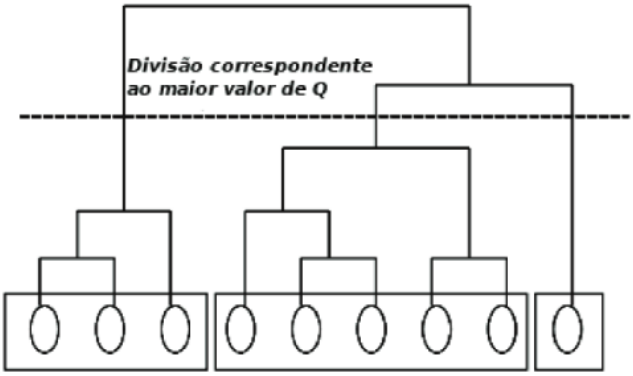


Figura 8 – Ilustração de um dendograma com o corte que representa o valor máximo da modularidade (RAGHAVAN; ALBERT; KUMARA, 2007).

Dada uma matriz de adjacência A, $A_{ij} = 1$ caso os vértices i e j estejam conectados e $A_{ij} = 0$ caso não estejam. O grau, $d(i) = \sum_j A_{ij}$ do vértice i é seu número de vizinhos, incluindo ele mesmo.

Considerando um processo de caminhada aleatória ou de difusão no grafo G. A cada passo, um vértice se move para outro vértice, escolhido aleatoriamente dentre os seus vizinhos. Os vértices são visitados em uma sequência na forma de uma cadeia de Markov, com os vértices do gafo representando os estados. A probabilidade de transição do vértice i para o j é de $P_{ij} = \frac{A_{ij}}{d(i)}$, a cada passo, o que define a matriz de transição P do processo de caminhada aleatória.

A probabilidade de sair de um vértice i para um vértice j através de uma caminhada aleatória de tamanho t é definida por P^t_{ij}. Caso dois vértices i e j estejam na mesma comunidade, a probabilidade P^t_{ij} será alta, no entanto, o contrário nem sempre é verdade, ou seja, um valor de P^t_{ij} alto não implica em i e j estarem na mesma comunidade.

Em (PONS; LATAPY, 2005) a distância entre dois vértices i e j em um grafo é dada por:

$$r_t(i,j) = \sqrt{\sum_{k=1}^{n} \frac{\left(P^t_{ik} - P^t_{jk}\right)^2}{d(k)}} \tag{2.1}$$

A probabilidade e uma caminhada aleatória partindo de um vértice da comunidade C chegar a um vértice j em um número t de passos é definida por:

$$P^t C_j = \frac{1}{|C|} \sum_{i \in C} P^t_{ij} \tag{2.2}$$

Com C_1 e C_2 sendo duas comunidades, a distância entre elas dá-se por:

$$rC_1C_2 = \sqrt{\sum_{k=1}^{n} \frac{\left(P^t_{C_1k} - P^t_{C_2k}\right)^2}{d(k)}} \tag{2.3}$$

No início, tem-se uma partição $P_1 = v, u \in V$ do grafo, em que cada vértice pertence a uma comunidade. Em seguida, computa-se a distância entre todos os vértices adjacentes e os seguintes passos se repetem:

1. Duas comunidades C_1 e C_2 em P_k são escolhidas de acordo com o critério de distância entre elas.

2. Essas comunidades se unem em uma nova $C_3 = C_1 \cup C_2$ e cria-se a nova partição $P_{k+1} = (P_k\{C_1, C_2\}) \cup \{C_3\}$.

3. A distância entre comunidades é alterada.

A escolha das comunidades que se unem possui papel fundamental na qualidade das comunidades resultantes. A cada iteração, as comunidades que se unem são as que minimizam a média das distâncias ao quadrado, σ_k, entre elas. Define-se σ_k como:

$$\sigma_k = \frac{1}{n} \sum_{C \in P_k} \sum_{i \in C} r^2_{iC} \tag{2.4}$$

2.4.2.3 Leading Eigenvector

O algoritmo Leading Eigenvector é de uma classe de algoritmos conhecidos como espectrais. Métodos espectrais se baseiam na análise de autovetores, autovalores e propriedades algébricas através de representações de redes em matrizes (VALEJO, 2014).

Dada uma matriz $M_{n,n}, \lambda \in \mathbb{R}$ é autovalor de M se existe um vetor $\gamma \neq 0$ tal que $M\gamma = \lambda\gamma$. Em decorrência, γ é um autovetor de λ.

Dentro da teoria espectral, usualmente representa-se a rede por uma matriz Laplaciana, no entanto, no caso do algoritmo Leading Eigenvector, utiliza-se como base a matriz de Modularidade B, dada por:

$$B_{v,u} = w_{v,u} - \frac{k_v k_u}{2m}, \tag{2.5}$$

com $w_{v,u}$ sendo o peso da aresta que conecta os vértices v e u e k representando o grau.

No algoritmo Leading Eigenvector, analisá-se o autovetor γ que está associado ao maior autovalor positivo de B. Define-se a separação de dois vértices pelo sinal correspondente a cada vértice de γ. Os vértices com sinais distintos são colocados em comunidades diferentes. Esse processo é realizado recursivamente até se obter o número de comunidades desejado. No

caso de todos os elementos possuírem o mesmo sinal, interpreta-e que a rede não apresenta estrutura de comunidades.

2.5 Medidas de Avaliação

Neste trabalho, algumas medidas de avaliação serão utilizadas para a análise dos resultados. Está Seção foca em introduzi-las.

2.5.1 Modularidade

A Modularidade é uma medida que se relaciona à qualidade de uma determinada divisão da rede (NEWMAN; GIRVAN, 2004). Visto que não espera-se uma estrutura de comunidades em redes aleatórias, a Modularidade compara a densidade de arestas dentro das comunidades com a densidade esperada em uma rede aleatória. Portanto, altos valores de Modularidade representam uma boa divisão da rede (FORTUNATO, 2010).

Em (NEWMAN; GIRVAN, 2004), definiu-se a Modularidade da seguinte forma: dada uma rede com K comunidades, define-se uma matriz simétrica $k \times k$, em que cada elemento e_{ij} é a fração das arestas que conectam vértices da comunidade i a vértices da comunidade j. Em seguida, é feito o traço da matriz, $Tre = \sum_i e_{ii}$, como a fração das arestas que ligam os elementos da mesma comunidade i. Espera-se que boas divisões da rede levem a um alto valor do traço, no entanto, o alto valor do traço não implica em uma boa divisão da rede. Então, a soma das linhas ou colunas que representam a fração de arestas que conectam vértices na comunidade i é definida por $a_i = \sum_j e_{ij}$. Sendo assim, em redes onde as arestas são inseridas de forma a ignorar a qual comunidade elas pertence teria-se $e_{ij} = a_i a_j$. Portanto, a equação de modularidade pode ser dada por:

$$Q = \sum_i (e_{ii} - a_i^2) = Tre = ||e^2||, \tag{2.6}$$

com $||e^2||$ como o somatório dos elementos da matriz. Ou seja, a Modularidade calcula a fração de arestas que conectam vértices de uma mesma comunidade subtraída do valor esperado em uma rede com o mesmo número de elementos, porém com conexões aleatórias.

2.5.2 Silhueta

A Silhueta é um método de interpretação e validação de consistência dentro de agrupamentos de dados. A técnica fornece uma representação gráfica sucinta de quão bem cada objeto foi classificado (ROUSSEEUW, 1987).

O valor da Silhueta é uma medida de quão semelhante um objeto é ao seu próprio grupo em comparação a outros grupos, com resultado variando de -1 a 1. Um valor alto sugere que

a configuração do agrupamento está apropriada, valores próximos a 0 indicam sobreposição de grupos, já valores negativos geralmente informam que amostras foram classificadas em grupos errados, já que outros grupos são mais similares.

Sua fórmula pode ser dada, então, por:

$$S = \frac{b - a}{max(a, b)}, \tag{2.7}$$

com a sendo a proximidade média entre um elemento e todos os demais de um mesmo grupo, e b sendo a proximidade média entre um elemento e todos os demais do grupo mais próximo ao dele.

A Silhueta pode ser calculada com qualquer métrica de proximidade, como a distância Euclidiana, a distância de Manhattan ou a similaridade de cossenos.

2.5.3 ϕ-arestas

A medida ϕ-arestas é utilizada para quantificar a qualidade de algoritmos de construção de redes. Conforme descrito em (OZAKI et al., 2011), a proporção de ϕ-arestas de um grafo representa o número de arestas que ligam objetos de diferentes grupos dividido pelo total de arestas do grafo. Sendo assim, ao analisar a semelhança dos grupos formados com as classes reais dos dados, quanto menor o valor da proporção de ϕ-arestas, melhor. Portanto, de forma resumida, essa medida quantifica a taxa de arestas inseridas na rede que conectam objetos de diferentes grupos, ou seja, é o percentual de arestas criadas erroneamente pelo algoritmo de construção de redes.

2.5.4 NMI

O NMI (*Normalized Mutual Information*) é uma pontuação de informação mútua normalizada entre as partições geradas por um algoritmo de agrupamento e as classes reais dos dados, seu propósito é retornar um resultado que vá de 0 a 1, sendo 0 quando não há nenhuma informação mútua entre os conjuntos e 1 quando há uma correlação perfeita (STREHL; GHOSH, 2002). Sendo assim, essa medida é utilizada quando sabe-se a divisão dos objetos previamente ou quando dados artificiais estão sendo avaliados e há uma partição de referência.

O cálculo para o NMI é feito da seguinte forma: dados dois conjuntos, predito (U) e o previsto (V), calcula-se a entropia (H) para ambos os conjuntos:

$$H(U) = -\sum_{i=1}^{|U|} P(i)log(P(i)) \tag{2.8}$$

$$H(V) = -\sum_{j=1}^{|V|} P(j)log(P(j)) \tag{2.9}$$

$P(i)$ é a probabilidade de um objeto aleatório de U ser atribuído à classe U_i, da mesma forma para $P(j)$. Após o cálculo das entropias, calcula-se o índice de informação mútua (MI) para os conjuntos U e V:

$$MI(U,V) = \sum_{i=1}^{} |U| \sum_{j=1}^{} |V| P(i,j)log(\frac{P(i,j)}{P(i)P'(j)}) \tag{2.10}$$

$P(i,j) = |U \cap V|/N$ sendo o total de objetos. O MI também pode ser escrito da seguinte forma:

$$MI(U,V) = \sum_{i=1}^{} |U| \sum_{j=1}^{} |V| \frac{|Ui \cap Vj|}{N} log(\frac{N|Ui \cap Vj|}{|Ui||Vj|}) \tag{2.11}$$

O índice de informação mútua normalizado, *Normalized Mutual Information* (NMI), define-se, então, por:

$$NMI(U,V) = \frac{MI(U,V)}{media(H(U),H(V))} \tag{2.12}$$

3 Metodologia

Este capítulo discorre sobre o processo de desenvolvimento da pesquisa deste trabalho, apresentando os métodos, técnicas, ferramentas e conjuntos de dados utilizados para a análise dos métodos de construção de rede e construção de comunidades no agrupamento de textos.

Todos os algoritmos usados e implementados utilizaram a linguagem de programação Python[1] em conjunto com as bibliotecas scikit-learn[2] e igraph[3].

3.1 Conjuntos de Dados

Foram utilizados conjuntos de dados retirados do trabalho de (ROSSI et al., 2013). Tratam-se de coleções de textos rotuladas por categoria. Nesta Seção, há uma breve descrição de cada um dos conjuntos de dados textuais utilizados, obtidos a partir da documentação de (ROSSI et al., 2013).

Para cada conjunto, foram selecionadas 300 amostras através de particionamento estratificado, que tem como objetivo manter a mesma proporção de rótulos do conjunto original. A única exceção foi a coleção CSTR, que apresenta apenas 299 amostras em sua totalidade.

3.1.1 Classic4

A coleção Classic4 é composta por 4 coleções distintas: CACM (títulos e resumos do periódico *Communications of the ACM*), CISI (*papers* de recuperação de informação), CRANFIELD (*papers* de sistema aeronáutico) e MEDLINE (periódicos médicos). Os rótulos das amostras é a própria coleção da qual aquela amostra faz parte.

3.1.2 CSTR

A coleção *Computer Science Technical Reports* (CSTR) é formada por resumos e relatórios técnicos publicados entre 1991 e 2007 no Departamento de Ciência da Computação da Universidade de Rochester. Os documentos se dividem entre quatro classes: inteligência artificial, visão/robótica, sistemas e teoria.

[1] <https://www.python.org/>
[2] <https://scikit-learn.org/stable/>
[3] <https://igraph.org/python/>

3.1.3 Irish-Sentiment

Coleção composta por artigos retirados das fontes irlandesas *RTE News*, *The Irish Times* e *The Irish Independent*. Os artigos foram classificados como positivos, negativos ou irrelevantes, sendo essas as três classes entre as quais os documentos se dividem.

3.1.4 La1s

Esta coleção é composta por artigos de notícia do *Los Angeles Times*. As classes se dividem entre: nacional, estrangeiro, entretenimento, financeiro, esportes e metropolitano.

3.1.5 LATimes

A coleção LATimes é a união das coleções La1s e La2s (similar à La1s). As classes se dividem entre as mesmas das suas subcoleções: nacional, estrangeiro, entretenimento, financeiro, esportes e metropolitano.

3.1.6 Oh0

Oh0 é um conjunto de documentos extraídos da coleção OHSUMED, que contém publicações médicas de 1987 a 1991. Os documentos estão divididos entre dez classes cujos temas são: México, ácido úrico, 6-Ketoprostaglandin F1 alpha, laringe, química cerebral, creatina quinase, ética, *fundus oculi*, Inglaterra, e prótese valvular cardíaca.

3.1.7 Opinosis

A coleção Opinosis é composta por opiniões de usuários sobre diferentes tópicos. As classes estão divididas por tópico de avaliação.

Os dados já foram obtidos pré-processados através de técnicas de PLN, estando na forma de vetores de características que contêm a quantidade de aparições de cada palavra nos documentos e rotulados por categorias, gerados através da técnica *bag of words*.

3.2 Escolha dos algoritmos

Os algoritmos escolhidos para realização dos experimentos foram detalhados no Capítulo 2. A seguir estão descritos os parâmetros fixados para implementação de cada um dos algoritmos:

k-Means: Implementação do scikit-learn.

- n_clusters: quantidade de classes da base de dados
- init: k-means++
- n_init: 10
- max_iter: 300

Também foi implementada uma versão de k-Means por similaridade de cossenos com os mesmos parâmetros.

k-NN e Mk-NN: Implementação do scikit-learn: kneighbors_graph

- mode: 'distance'
- metric: 'euclidean' e 'cosine'
- include_self: falso

Sk-NN: Implementação própria.

- mode: 'distance'
- metric: 'euclidean' e 'cosine'

E-Vizinhança: Implementação do igraph: radius_neighbors_graph

- mode: 'distance'
- metric: 'euclidean' e 'cosine'
- include_self: falso

Fastgreedy, Walktrap: Implementação do igraph

- weights: peso da rede ponderada
- n: quantidade de classes da base de dados

Leading Eigenvector: Implementação do igraph

- weights: peso da rede ponderada
- clusters: quantidade de classes da base de dados + 1

3.3 Experimentos

A execução dos experimentos foi estruturada da seguinte forma: em um primeiro nível são executados os algoritmos de construção de redes em conjunto de uma proximidade (Euclidiana ou similaridade de cosseno), tendo o valor de seus parâmetros variados dentro de um intervalo e com um passo específico, como exibido na Tabela 1. Armazena-se a estrutura que apresentar o maior valor de modularidade para todo par formado pelo algoritmo de construção de redes e a proximidade, considerando, para isso, os rótulos reais das amostras.

Tabela 1 – Intervalo de variação para definição de parâmetros.

Algoritmo	**Parâmetro**	**Intervalo**	**Passo**
k-NN	k	1 - 40	1
Mk-NN	k	1 - 40	1
Sk-NN	$kmax$	1 - 40	1
E-Vizinhança	ϵ	0.1 - 25.0	0.1

Para cada uma das estruturas armazenadas, calcula-se a métrica ϕ-arestas, e guarda-se sua Modularidade, juntamente com o valor de parâmetro que levou a maior modularidade. Em seguida, são utilizados três algoritmos de detecção de comunidades em cada rede salva, Fastgreedy, Walktrap e Leading Eigenvector. Calcula-se, então, as métricas NMI, Modularidade em relação aos grupos formados e a Silhueta (respeitando em seu cálculo a métrica de proximidade utilizada para a construção da rede), para as saídas das técnicas de construção de comunidades. Por fim, calcula-se e armazena-se a média dessas métricas e seu desvio padrão. O processo se repete para os sete conjuntos de dados descritos na Seção 3.1.

4 Análise e Discussão dos Resultados

As Seções deste Capítulo visam apresentar os resultados experimentais obtidos na execução dos experimentos descritos no Capítulo 3. A análise dos resultados está dividida em três partes: a primeira com foco nos algoritmos de Construção de Redes, que apresenta os cálculos de ϕ-arestas e modularidade (com base nas classes reais dos objetos) para cada um dos conjuntos selecionados; a segunda parte visa exibir os cálculos de modularidade, desta vez tomando como base os agrupamentos obtidos pelos algoritmos de Detecção de Comunidades; por fim, a terceira parte foca na análise dos agrupamentos, apresentando os resultados de Silhueta e NMI de cada base de dados.

4.1 Algoritmos de Construção de Redes

A Tabela 2 exibe a proporção de ϕ-arestas em cada conjunto de dados. É válido ressaltar que, para este cálculo, foram utilizadas as classes reais dos objetos.

Tabela 2 – Proporção de ϕ-arestas para cada conjunto de dados.

	Classic4	CSTR	IrishEconomic	La1s	LATimes	Oh0	Opinosis	Média
k-NN + Euclidean	0.42	0.51	0.59	0.51	0.65	0.67	0.73	0.58
Mk-NN + Euclidian	0.40	0.52	0.60	0.75	0.74	0.78	0.77	0.65
E-Vizinhança + Euclidean	**0.07**	0.61	0.58	0.80	0.72	0.80	0.86	0.63
Sk-NN + Euclidean	0.57	0.37	0.52	0.42	0.57	0.75	0.86	0.58
k-NN + Cosine	0.17	**0.21**	0.53	0.36	0.46	**0.35**	**0.58**	0.38
Mk-NN + Cosine	0.18	0.23	**0.48**	**0.33**	**0.31**	**0.35**	0.67	**0.36**
E-Vizinhança + Cosine	0.12	0.25	0.51	0.46	0.70	0.41	0.71	0.45
Sk-NN + Cosine	0.47	0.30	0.52	0.43	0.43	0.45	0.69	0.47

Como a proporção de ϕ-arestas indica o percentual de arestas que conectam objetos de classes distintas, os resultados mais próximos de zero são os melhores, pois demonstram uma divisão na rede entre elementos de diferentes classes.

Nota-se que, em média, os algoritmos apresentaram melhores resultados com o uso da similaridade de cossenos como proximidade, o que faz sentido, visto que, no geral, esta métrica apresenta melhores resultados em bases textuais em comparação ao uso da tradicional distância Euclidiana (GUNAWAN; SEMBIRING; BUDIMAN, 2018). Por fim, o algoritmo que apresentou o melhor resultado, considerando a média de todos os conjuntos, foi o Mk-NN em conjunto com a similaridade de cossenos, seguido pelo k-NN, também com similaridade de cossenos.

A Tabela 3 mostra a modularidade de cada conjunto de dados, considerando as redes formadas pelos pares de algoritmo de construção de rede e a medida de proximidade. Novamente,

foram utilizadas as classes reais dos objetos para este cálculo.

Tabela 3 – Modularidade da rede para cada conjunto de dados considerando as classes reais.

	Classic4	CSTR	IrishEconomic	La1s	LATimes	Oh0	Opinosis	Média
k-NN + Euclidean	0.16	0.25	0.12	0.31	0.28	0.27	0.28	0.23
Mk-NN + Euclidian	0.04	0.29	**0.28**	0.39	0.13	0.26	0.24	0.23
E-Vizinhança + Euclidean	0.03	0.01	0.02	0.19	0.04	0.04	0.09	0.06
Sk-NN + Euclidean	0.28	0.43	0.13	0.38	0.39	0.45	0.31	0.33
k-NN + Cosine	0.47	0.43	0.09	0.38	0.33	0.46	**0.35**	0.35
Mk-NN + Cosine	**0.52**	**0.44**	0.16	**0.44**	**0.47**	**0.51**	0.30	**0.40**
E-Vizinhança + Cosine	0.39	0.39	0.15	0.32	0.12	0.32	0.23	0.27
Sk-NN + Cosine	0.18	0.34	0.12	0.35	0.35	0.38	0.26	0.28

Com resultados similares aos da Tabela anterior, vemos que mais uma vez, no geral, os algoritmos aplicados juntamente à similaridade de cossenos levaram a melhores resultados, com exceção do Sk-NN, que se saiu ligeiramente superior ao utilizado com a distância Euclidiana. Novamente, vemos que em média o algoritmo com melhor desempenho foi o Mk-NN com similaridade de cossenos.

4.2 Modularidade Prevista

Na Tabela 4 é indicado as modularidades prevista dos conjuntos, ou seja, desta vez os agrupamentos obtidos pelos algoritmos de detecção de comunidades foram utilizados, substituindo as classes reais dos objetos, utilizadas nos cálculos anteriores. Os cálculos foram feitos para os agrupamentos obtidos pelos três diferentes algoritmos de detecção de redes apresentados na Seção 3.2, Fastgreedy, Walktrap e Leading Eigenvector, o valor adicionado à tabela é a média dos resultados para cada algoritmo e o erro se refere ao seu desvio padrão.

Tabela 4 – Modularidade prevista da rede para cada conjunto de dados.

	Classic4	CSTR	IrishEconomic	La1s	LATimes	Oh0	Opinosis	Média
k-NN + Euclidean	0.33 ± 0.27	0.36 ± 0.26	0.40 ± 0.28	0.46 ± 0.13	0.49 ± 0.20	0.55 ± 0.13	**0.78 ± 0.01**	0.48
Mk-NN + Euclidian	0.42 ± 0.19	0.42 ± 0.21	0.35 ± 0.05	**0.67 ± 0.11**	**0.63 ± 0.05**	0.71 ± 0.07	0.65 ± 0.01	0.55
E-Vizinhança + Euclidean	0.06 ± 0.01	0.07 ± 0.04	0.14 ± 0.03	0.55 ± 0.04	0.34 ± 0.01	0.48 ± 0.01	0.35 ± 0.05	0.28
Sk-NN + Euclidean	0.31 ± 0.27	0.41 ± 0.29	0.36 ± 0.26	0.51 ± 0.06	0.44 ± 0.32	0.45 ± 0.26	0.74 ± 0.04	0.46
k-NN + Cosine	0.55 ± 0.04	0.45 ± 0.17	0.33 ± 0.10	0.47 ± 0.10	0.55 ± 0.12	0.64 ± 0.04	0.70 ± 0.01	0.52
Mk-NN + Cosine	0.46 ± 0.22	**0.61 ± 0.06**	**0.57 ± 0.03**	0.62 ± 0.07	0.60 ± 0.12	**0.75 ± 0.03**	0.69 ± 0.02	**0.61**
E-Vizinhança + Cosine	0.41 ± 0.26	0.49 ± 0.20	0.46 ± 0.11	0.54 ± 0.11	0.31 ± 0.10	0.59 ± 0.04	0.61 ± 0.03	0.48
Sk-NN + Cosine	**0.63 ± 0.05**	0.43 ± 0.21	**0.57 ± 0.08**	0.52 ± 0.07	0.58 ± 0.05	0.73 ± 0.06	0.59 ± 0.02	0.57

Novamente, o padrão identificado para os resultados anteriores se repete: a métrica de similaridade de cossenos levam os algoritmos de construção de comunidades a obter resultados melhores, em comparação a distância Euclidiana. Mais uma vez, o Mk-NN com similaridades de cosseno apresenta uma média superior aos demais algoritmos.

É interessante notar que a Tabela 4 exibe valores de modularidade consideravelmente melhores que os apresentados na Tabela 3, isso nos traz indícios de que os agrupamentos

identificados pelas ferramentas de detecção de comunidades não possuem uma relação direta com as classes reais dos objetos. Esta premissa é esperada, visto que as bases possuem características topológicas que certamente não se relacionam diretamente aos rótulos reais que foram definidos na classificação dos textos de cada base.

4.3 Agrupamentos

Na análise dos agrupamentos, o algoritmo k-Means foi utilizado como um referencial para a comparação dos resultados obtidos no agrupamento baseado em redes com os resultados de um algoritmo tradicional de agrupamento. Outra vez, os cálculos referentes aos algoritmos baseados em redes foram feitos para os agrupamentos obtidos para todas as técnicas de detecção de comunidades escolhidas e selecionou-se o valor da média entre os resultados em conjunto com seu desvio padrão para a taxa de erro.

A Tabela 5 apresenta a silhueta calculada para cada base.

Tabela 5 – Silhueta calculada para cada conjunto de dados.

	Classic4	CSTR	IrishEconomic	La1s	LATimes	Oh0	Opinosis	Média
k-NN + Euclidean	-0.21 ± 0.10	-0.14 ± 0.08	-0.04 ± 0.03	-0.07 ± 0.06	-0.22 ± 0.09	-0.07 ± 0.05	-0.05 ± 0.04	-0.11
Mk-NN + Euclidian	-0.28 ± 0.01	-0.16 ± 0.04	-0.09 ± 0.09	-0.30 ± 0.02	-0.31 ± 0.02	-0.21 ± 0.01	-0.09 ± 0.01	-0.20
E-Vizinhança + Euclidean	-0.07 ± 0.01	0.02 ± 0.09	-0.18 ± 0.02	-0.32 ± 0.01	-0.31 ± 0.02	-0.21 ± 0.01	-0.10 ± 0.01	-0.16
Sk-NN + Euclidean	-0.02 ± 0.15	-0.03 ± 0.04	-0.02 ± 0.05	-0.12 ± 0.05	-0.08 ± 0.12	-0.09 ± 0.06	-0.09 ± 0.03	-0.06
k-NN + Cosine	0.04 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.01 ± 0.01	0.02 ± 0.01	0.01 ± 0.01	0.00 ± 0.05	0.01
Mk-NN + Cosine	0.02 ± 0.01	0.01 ± 0.01	0.00 ± 0.01	0.01 ± 0.01	-0.03 ± 0.03	0.03 ± 0.01	0.02 ± 0.04	0.00
E-Vizinhança + Cosine	0.04 ± 0.01	0.01 ± 0.01	0.00 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.01 ± 0.01	0.03 ± 0.02	0.01
Sk-NN + Cosine	0.00 ± 0.01	0.00 ± 0.01	0.00 ± 0.01	0.00 ± 0.01	0.00 ± 0.01	0.00 ± 0.01	-0.01 ± 0.06	0.00
k-Means + Euclidean	**0.40**	**0.14**	**0.27**	**0.37**	**0.33**	**0.11**	-0.02	**0.22**
k-Means + Cosine	0.07	0.03	0.02	0.04	0.06	0.02	**0.08**	0.04

É possível notar que os resultados dos métodos utilizados com similaridade de cosseno apresentaram-se muito próximos de 0, o que indica que há sobreposição de grupos, não sendo óbvia a divisão dos dados entre eles. Ao analisar os resultados referentes a distância Euclidiana, nota-se que, apesar de próximos a 0, os valores foram negativos, levando a conclusão de que, baseado somente na distância entre os objetos e os grupos definidos, a definição do grupo de cada objeto não foi ótima, havendo objetos classificados em um grupo que se apresentam mais próximo de outros do que de seu próprio. O k-Means com distância Euclidiana foi o único método a apresentar uma silhueta com valor significativamente superior a 0, tal resultado pode justificar-se no fato de que o método k-Means é diretamente baseado em distâncias entre objetos e o centro dos grupos.

A Tabela 6 apresenta o NMI calculado para cada conjunto, ou seja, a taxa de similaridade entre as classes reais dos objetos e os agrupamentos obtidos pelos métodos de detecção de comunidades.

Tabela 6 – NMI calculado para cada conjunto de dados.

	Classic4	CSTR	IrishEconomic	La1s	LATimes	Oh0	Opinosis	Média
k-NN + Euclidean	0.05 ± 0.01	0.04 ± 0.01	0.03 ± 0.01	0.20 ± 0.03	0.10 ± 0.03	0.20 ± 0.02	0.67 ± 0.04	0.18
Mk-NN + Euclidian	0.09 ± 0.03	0.09 ± 0.04	0.01 ± 0.01	0.08 ± 0.02	0.05 ± 0.01	0.09 ± 0.01	0.47 ± 0.06	0.12
E-Vizinhança + Euclidean	0.25 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.05 ± 0.01	0.04 ± 0.01	0.06 ± 0.01	0.34 ± 0.09	0.11
Sk-NN + Euclidean	0.03 ± 0.01	0.08 ± 0.03	0.01 ± 0.01	0.29 ± 0.03	0.07 ± 0.01	0.11 ± 0.01	0.62 ± 0.06	0.17
k-NN + Cosine	0.42 ± 0.09	**0.32 ± 0.06**	0.02 ± 0.01	0.24 ± 0.04	0.22 ± 0.04	0.31 ± 0.03	0.65 ± 0.05	0.31
Mk-NN + Cosine	0.37 ± 0.20	0.27 ± 0.06	**0.06 ± 0.01**	0.24 ± 0.01	0.23 ± 0.06	**0.44 ± 0.03**	0.63 ± 0.05	0.32
E-Vizinhança + Cosine	0.30 ± 0.10	0.16 ± 0.05	0.04 ± 0.01	0.13 ± 0.05	0.13 ± 0.05	0.25 ± 0.01	0.58 ± 0.11	0.22
Sk-NN + Cosine	0.09 ± 0.03	0.13 ± 0.02	0.01 ± 0.01	0.22 ± 0.05	0.21 ± 0.01	0.27 ± 0.02	0.61 ± 0.06	0.22
k-Means + Euclidean	0.04	0.11	0.01	0.06	0.12	0.23	0.67	0.17
k-Means + Cosine	**0.49**	0.22	**0.06**	**0.30**	**0.31**	0.40	**0.70**	**0.35**

Percebe-se que o resultado do NMI está diretamente relacionado a características da base textual utilizada para o seu cálculo, enquanto na IrishEconomic os valores são muito próximos de 0, demonstrando uma total desconexão entre as classes e os agrupamentos, na Opinosis obteve-se até 70% de similaridade. Uma possível explicação para o baixo desempenho da base IrishEconomic neste cálculo com relação as demais é o fato de suas classes serem divididas pelo teor sentimental do texto (positivo, indiferente ou negativo), diferentemente das outras, que possuem suas classes definidas pelo assunto tratado em cada documento. Essa variação entre os resultados de cada base reforça a premissa de que nem sempre as características e padrões identificados pelos algoritmos de agrupamento se relacionam com os rótulos reais nos quais os conjuntos de documentos foram classificados. Isto não indica uma má qualidade dos agrupamentos, informa apenas que as semelhanças utilizadas em suas formações se diferem das caraterísticas analisadas no processo de classificação das bases por categorias.

Também é visível que a utilização de similaridade de cossenos aproxima os agrupamentos das classes reais, tanto para o k-Means (algoritmo tradicional), como para os métodos baseados em redes.

Apesar de o k-Means com similaridade de cossenos ter mostrado, no geral, resultados liminarmente superiores aos dos métodos baseados em redes, os algoritmos Mk-NN e k-NN, também com similaridade de cossenos, tiveram resultados próximos e até superiores em algumas bases de dados, como a Oh0 e a CSTR, o que indica que algoritmos de agrupamento baseados em redes podem gerar soluções tão satisfatórias quanto os algoritmos de agrupamento tradicionais.

É válido acrescentar que, no caso dos algoritmos de agrupamento baseados em redes, foram utilizados três diferentes algoritmos de detecção de comunidades e avaliou-se apenas a média alcançada por eles juntamente ao seu desvio padrão, sendo assim, é possível que um algoritmo específico tenha performado melhor que os demais e tido resultados superiores ao k-Means, mas que isso tenha sido ocultado pelo desempenho dos demais algoritmos.

5 Conclusão

Este trabalho avalia a influência de métodos de construção de redes em conjunto com algoritmos de detecção de comunidades na tarefa de agrupamento de textos.

Foram estudados quatro algoritmos de construção de redes, cada um foi avaliado com duas diferentes medidas de proximidade: Euclidiana e similaridade de cossenos. Ficou claro que o uso da similaridade de cossenos traz vantagem para bases de textos, visto que os resultados derivados de sua utilização foram melhores em todas as métricas avaliadas nos experimentos.

Verificou-se também que a similaridade entre os agrupamentos gerados e as classes originais dos documentos varia de acordo com o conjunto de dados, isto se confirma devido aos resultados do NMI, que apresentou valores que iam de próximo a 0 até 0,7 (sendo que a métrica varia de 0 a 1).

Também é notável um melhor desempenho, dentre os algoritmos de agrupamento baseados em redes, dos algoritmos k-NN e Mk-NN para todas as métricas calculadas, sobressaindo-se em relação a outras técnicas, como E-Vizinhanças e o Sk-NN, e apresentando resultados muito próximos ou melhores que o k-Means, que utiliza-se de um método tradicional de agrupamento.

Por fim, considerando o resultado dos experimentos, é possível concluir que, no geral, os algoritmos de detecção de comunidade são uma alternativa viável e, em determinados conjunto de dados, até melhor para problemas que envolvem agrupamento de dados quando a métrica de interesse é o desempenho do agrupamento, em especial o NMI.

5.1 Trabalhos Futuros

Para a continuação deste trabalho definem-se algumas possibilidades que poderiam ser abordadas com o objetivo de melhorar o estudo e os resultados obtidos:

- Utilizar o conjunto completo de dados e não apenas uma partição dele;
- Analisar outras bases de texto com diferentes critérios para a divisão de suas categorias;
- Avaliar a performance (em termos de tempo de processamento) dos algoritmos utilizados para a realização dos experimentos;
- Utilizar algum método de otimização de hiper-parâmetros para a definição ótima dos parâmetros de cada algoritmo de construção de redes;

- Avaliar os algoritmos de detecção de comunidades isoladamente, para que seja possível identificar aquele com os melhores resultados na tarefa de agrupamento de textos.

Referências

ALBERT, R.; BARABÁSI, A.-L. Statistical mechanics of complex networks. *Reviews of modern physics*, APS, v. 74, n. 1, p. 47, 2002.

ARTHUR, D.; VASSILVITSKII, S. K-means++: The advantages of careful seeding. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. USA: Society for Industrial and Applied Mathematics, 2007. (SODA '07), p. 1027–1035. ISBN 9780898716245.

BELKIN, M.; NIYOGI, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, MIT Press, v. 15, n. 6, p. 1373–1396, 2003.

BERTON, L. *Construção de redes baseadas em vizinhança para o aprendizado semissupervisionado*. Tese (Doutorado) — Universidade de São Paulo, 2016.

BERTON, L.; LOPES, A. Informativity-based graph: Exploring mutual knn and labeled vertices for semi-supervised learning. In: . [S.l.: s.n.], 2012. p. 14–19. ISBN 978-1-4673-4793-8.

BREWKA, G. Artificial intelligence—a modern approach by stuart russell and peter norvig, prentice hall. series in artificial intelligence, englewood cliffs, nj. *The Knowledge Engineering Review*, Cambridge University Press, v. 11, n. 1, p. 78–79, 1996.

BRITO, M. R. et al. Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statistics & Probability Letters*, Elsevier, v. 35, n. 1, p. 33–42, 1997.

CARNEIRO, M. *Redes complexas para classificação de dados via conformidade de padrão, caracterização de importância e otimização estrutural*. Tese (Doutorado) — PhD thesis, Universidade de Sao Paulo, 2016.

CLAUSET, A.; NEWMAN, M. E. J.; MOORE, C. Finding community structure in very large networks. *Phys. Rev. E*, v. 70, p. 066111, 2004.

ERTEL, W. *Introduction to artificial intelligence*. [S.l.]: Springer, 2018.

ESTER, M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. [S.l.]: AAAI Press, 1996. (KDD'96), p. 226–231.

FACELI, K. et al. Inteligência artificial: uma abordagem de aprendizado de máquina. 2011.

FORTUNATO, S. Community detection in graphs. *Physics reports*, Elsevier, v. 486, n. 3-5, p. 75–174, 2010.

GUNAWAN, D.; SEMBIRING, C. A.; BUDIMAN, M. A. The implementation of cosine similarity to calculate text relevance between two documents. *Journal of Physics: Conference Series*, IOP Publishing, v. 978, p. 012120, mar 2018. Disponível em: <https://doi.org/10.1088/1742-6596/978/1/012120>.

GUO, G. et al. Knn model-based approach in classification. In: SPRINGER. *OTM Confederated International Conferences"On the Move to Meaningful Internet Systems"*. [S.l.], 2003. p. 986–996.

HAYKIN, S.; NETWORK, N. A comprehensive foundation. *Neural networks*, v. 2, n. 2004, p. 41, 2004.

KOTSIANTIS, S. B. et al. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, Amsterdam, v. 160, n. 1, p. 3–24, 2007.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: REZENDE, S. O. (Ed.). *Sistemas inteligentes: fundamentos e aplicações*. [S.l.]: Editora Manole Ltda, 2003. cap. 4.

NEWMAN, M. E.; GIRVAN, M. Finding and evaluating community structure in networks. *Physical review E*, APS, v. 69, n. 2, p. 026113, 2004.

OZAKI, K. et al. Using the mutual k-nearest neighbor graphs for semi-supervised classification of natural language data. In: *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*. USA: Association for Computational Linguistics, 2011. (CoNLL '11), p. 154–162. ISBN 9781932432923.

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.

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.

ROSSI, R. G. et al. Benchmarking text collections for classification and clustering tasks. São Carlos, SP, Brasil., 2013.

ROUSSEEUW, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, v. 20, p. 53–65, 1987. ISSN 0377-0427. Disponível em: <https://www.sciencedirect.com/science/article/pii/0377042787901257>.

STREHL, A.; GHOSH, J. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, v. 3, n. Dec, p. 583–617, 2002.

VALEJO, A. D. B. *Refinamento multinível em redes complexas baseado em similaridade de vizinhança*. Tese (Doutorado) — Universidade de São Paulo, 2014.

VEGA-OLIVEROS, D. A. et al. Regular graph construction for semi-supervised learning. In: IOP PUBLISHING. *Journal of physics: Conference series*. [S.l.], 2014. v. 490, n. 1, p. 012022.

WU, X. et al. Top 10 algorithms in data mining. *Knowledge and information systems*, Springer, v. 14, n. 1, p. 1–37, 2008.