UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Aprendizado semissupervisionado baseado em uma única classe por meio de propagação de rótulos em grafos

Julio César Carnevali

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-C²MC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Julio César Carnevali

Aprendizado semissupervisionado baseado em uma única classe por meio de propagação de rótulos em grafos

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Alneu de Andrade Lopes

USP – São Carlos
Novembro de 2020

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

C289a Carnevali, Julio César
Aprendizado semissupervisionado baseado em uma
única classe por meio de propagação de rótulos em
grafos / Julio César Carnevali; orientador Alneu de
Andrade Lopes. -- São Carlos, 2020.
84 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2020.

1. Aprendizado semissupervisionado baseado em
uma única classe. 2. Aprendizado baseado em grafo.
3. Aprendizado transdutivo. 4. Classificação de
textos. 5. Propagação de rótulos. I. de Andrade
Lopes, Alneu, orient. II. Título.

Julio César Carnevali

Positive and unlabeled learning through label propagation in graphs

Master dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Alneu de Andrade Lopes

USP – São Carlos
November 2020

AGRADECIMENTOS

Agradeço primeiramente a Deus pois “até aqui nos ajudou o Senhor” (1 Samuel 7:12) e também meus pais, Moacir e Matilde, pelo apoio em todas as minhas decisões, e por sempre estarem presentes, fazendo parte de todos os passos da minha vida.

Agradeço também minha esposa Lígia, que durante o início do mestrado ainda era minha namorada, por sempre me incentivar a fazer o que gosto, além de me apoiar em continuar os estudos nesta oportunidade que foi vir para o ICMC-USP, cursar o programa de Mestrado em Ciência da Computação e Matemática Computacional na área de Inteligência Artificial (IA).

O apreço pela área de IA, foi cultivado pelo Prof. Dr. Ricardo Marcacini, quando eu ainda cursava Sistemas de Informação na UFMS de Três Lagoas – MS, onde tive o primeiro contato com pesquisa por meio de uma iniciação científica, orientado pelo mesmo. Assim, gostaria de agradecê-lo pelos ensinamentos e incentivo, bem como pelas conversas, que temos até hoje.

Agradeço o meu orientador, Prof. Dr. Alneu, e o meu coorientador (informal) Prof. Dr. Rafael Rossi, que sempre tiveram paciência, e nunca mediram esforços para me orientar. Graças a eles, que pude desenvolver este projeto de mestrado, para que hoje se tornasse uma dissertação com resultados que provam nossas hipóteses. Agradeço também o Prof. Dr. Evangelos, que em suas visitas ao Brasil e até mesmo de Halifax no Canadá, separou parte de seu tempo para participar dessa pesquisa dando valiosas sugestões. Além disso, agradeço o Prof. Alneu pelos conselhos, ensinamentos, e por me abrir portas únicas. Não posso deixar de mencionar que, assim como o Prof. Ricardo, conheço o Prof. Rafael desde a minha graduação, e que nossas conversas sempre foram produtivas e incentivadoras.

Agradeço também ao CNPq (processo 132770/2018-0) pelo apoio financeiro, que possibilitou minha dedicação exclusiva ao mestrado, e ao CALDO-FAPESP (processo 2017/50153-9) que por meio do projeto “Graph-based total recall information retrieval on text document corpora” me permitiu conhecer o Prof. Dr. Evangelos já mencionado acima.

Ainda, gostaria de agradecer aos amigos que fizeram parte dessa jornada, em especial, André Toshio e Luan Martins, que compartilharam muitas experiências comigo, e que contribuíram para minha formação.

Deixo também meus agradecimentos a toda equipe do ICMC-USP, que fizeram parte de cada etapa, do início ao fim do meu mestrado.

RESUMO

CARNEVALI, J. C. **Aprendizado semissupervisionado baseado em uma única classe por meio de propagação de rótulos em grafos**. 2020. 84 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

O aprendizado semissupervisionado baseado em uma única classe, amplamente conhecido pelo termo em inglês *Positive and Unlabeled Learning* (PUL), é um método atrativo para aplicações práticas, pois o usuário só precisa rotular documentos de seu interesse, evitando o esforço de rotular documentos para todas as classes da coleção, tarefa necessária nos métodos multiclasse. Além dos documentos de interesse do usuário, são utilizados também documentos não rotulados durante o processo de aprendizagem, para então classificar os documentos entre classe de interesse ou não (também denominados como documentos positivos e negativos respectivamente). Esse método de aprendizado pode ser utilizado para construir modelos de classificação, recuperação de informação, ou sistemas de recomendação. As abordagens PUL encontradas na literatura raramente fazem uso da representação de dados por meio de grafos. Dado que esse meio de representação de dados é pouco explorado no contexto de PUL, e dado que seu uso em abordagens do aprendizado semissupervisionado para classificação de textos produz resultados tão bons quanto e até melhores que abordagens baseadas no modelo espaço-vetorial, neste projeto de mestrado, é proposto uma abordagem baseada em grafo para PUL denominada, *Label Propagation for Positive and Unlabeled Learning* (LP-PUL). O método proposto consiste de 3 etapas: (i) construção do grafo para representação da coleção textual, (ii) identificação de documentos negativos, e (iii) propagação dos rótulos positivo e negativo para os demais documentos não rotulados. Foi realizada uma extensa avaliação empírica utilizando um grande número de coleções e parâmetros para cada algoritmo utilizado. Durante a avaliação, foi medido o impacto das diferentes escolhas de algoritmos para cada etapa acima mencionadas. Além disso, o método proposto foi comparado com algoritmos PUL baseados em grafos e no modelo espaço-vetorial. Ao final, foi demonstrado que o método proposto obtém melhor performance de classificação que os demais algoritmos PUL.

Palavras-chave: aprendizado semissupervisionado baseado em uma única classe, aprendizado baseado em grafo, aprendizado transdutivo, propagação de rótulos, classificação de textos.

ABSTRACT

CARNEVALI, J. C. **Positive and unlabeled learning through label propagation in graphs**. 2020. 84 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

Positive and Unlabeled Learning (PUL) is an attractive learning approach for real applications since the user has to label only documents of his class of interest. Therefore, the user spends no effort to label documents of all classes in the collection, necessary task in the case of multiclass methods. In addition to the labeled interest documents, the PUL uses unlabeled documents to classify all the documents as documents of interest (positive documents) or not (negative documents). This learning approach can be used to train classification models, to perform information retrieval or recommendation systems. In the literature, the PUL approaches based on graphs are the minority, and the graph-based approaches used in the semi-supervised text classification task have proved to surpass the classification performance of the vector space model-based approaches. Based on these statements, we proposed a graph-based approach for PUL, called Label Propagation for Positive and Unlabeled Learning (LP-PUL). The proposed framework consists of three steps: (i) building a similarity graph, (ii) identifying reliable negative documents, and (iii) performing label propagation to classify the remaining unlabeled documents as positive or negative. We carried out an extensive empirical evaluation using several text collections and parameters for each algorithm. The empirical evaluation measured the impact of the different algorithms for each step of the proposed framework. We also compared the proposed algorithm with vector space-based and graph-based ones, and then was demonstrated that the proposal surpasses the classification performances of other PUL algorithms.

Keywords: positive and unlabeled learning, graph-based learning, transductive learning, label propagation, text classification.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplos de grafos homogêneo e heterogêneo.	29
Figura 2 – Ilustração do processo de classificação automática de textos por meio do aprendizado transdutivo.	33
Figura 3 – Diferença entre SVM e TSVM para o mesmo conjunto de documentos rotulados.	38
Figura 4 – Ilustração da classificação transdutiva via propagação de rótulos. . . .	41
Figura 5 – Comparativo entre o aprendizado baseado em uma única classe (OCL), que utiliza apenas exemplos positivos rotulados em seu treinamento, e o aprendizado semissupervisionado baseado em uma única classe (PUL), que utiliza tanto exemplos positivos rotulados como exemplos não rotulados. Essa figura representa um cenário com poucos exemplos rotulados.	48
Figura 6 – Esquema geral do método proposto: <i>Label Propagation for Positive and Unlabeled Learning</i> (LP-PUL).	54
Figura 7 – Conjunto de objetos apresentados por meio do plano bidimensional e por meio do grafo 3NN construído com base na distância euclidiana. . . .	56
Figura 8 – Comparação da performance de classificação do método proposto, LP-PUL, com dois algoritmos OCL, <i>k-Means</i> e <i>k-NND</i> , e três algoritmos PUL, PE-PUC, PU-LP e RC-SVM. O eixo x representa o número de documentos positivos rotulados e o eixo y representa os valores F_1	63
Figura 9 – Gráficos contendo as melhores performances de classificação referentes ao método proposto quando executado com o LLGC e quando executado com o GFHF na etapa final. O eixo x representa o número de documentos positivos rotulados e o eixo y representa os valores F_1	66
Figura 10 – Melhores performances de classificação referente ao uso de cada algoritmo de construção de grafo, k NN, Mk NN and ϵ N, no método proposto, considerando o uso do LLGC na etapa de propagação de rótulos. O eixo x representa o número de documentos positivos rotulados e o eixo y representa os valores F_1	68
Figura 11 – Melhores performances de classificação referente ao uso de cada algoritmo de construção de grafo, k NN, Mk NN and ϵ N, no método proposto, considerando o uso do GFHF na etapa de propagação de rótulos. O eixo x representa o número de documentos positivos rotulados e o eixo y representa os valores F_1	69

Figura 12 – Diagramas de diferença crítica dos três algoritmos de construção de grafo quando aplicados ao LP-PUL. Os resultados consideram o uso do LLGC na etapa de propagação de rótulos.	70
Figura 13 – Diagramas de diferença estatística dos três algoritmos de construção de grafo quando aplicados ao LP-PUL. Os resultados consideram o uso do GFHF na etapa de propagação de rótulos.	70

LISTA DE ALGORITMOS

Algoritmo 1 – <i>Label Propagation</i>	42
Algoritmo 2 – <i>Learning With Local and Global Consistency</i>	43
Algoritmo 3 – <i>LP-PUL</i>	55
Algoritmo 4 – <i>Extração de documentos negativos</i>	57

LISTA DE TABELAS

- Tabela 1 – Características das coleções de documentos, sendo elas: número de documentos ($|\mathcal{D}|$), número de termos gerados ($|\mathcal{T}|$), número médio de termos por documentos ($|\overline{\mathcal{T}}|$), número de classes ($|\mathcal{C}|$), desvio padrão considerando a porcentagem de distribuição das classes ($\sigma(\mathcal{C})$) e porcentagem de documentos da classe majoritária ($\max(\mathcal{C})$). 60
- Tabela 2 – Esta tabela apresenta o ranqueamento médio (RM), ranqueamento geral (RG), o *p-value* considerando os valores de F_1 para cada quantidade de documentos positivos rotulados (apresentados em negrito quando o LP-PUL supera o referido método com DES), e o valor de p que produz DES. 64

SUMÁRIO

1	INTRODUÇÃO	19
1.1	Principais Resultados	23
1.2	Organização do Texto	23
2	FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS	25
2.1	Representações de Coleções de Textos	25
2.1.1	<i>Limpeza e Padronização de Textos</i>	26
2.1.2	<i>Modelo Espaço-Vetorial</i>	26
2.1.3	<i>Grafos</i>	27
2.2	Aprendizado Semissupervisionado	32
2.2.1	<i>Aprendizado Transdutivo no Modelo Espaço-Vetorial</i>	34
2.2.2	<i>Aprendizado Transdutivo em Grafos</i>	39
2.3	Aprendizado Baseado em Uma Única Classe	43
2.4	Aprendizado Semissupervisionado Baseado em Uma Única Classe	47
3	MÉTODO PROPOSTO: <i>LABEL PROPAGATION FOR POSITIVE AND UNLABELED LEARNING (LP-PUL)</i>	53
3.1	Características Gerais	53
3.2	Construção do Grafo	55
3.3	Extração de Documentos Negativos	55
3.4	Propagação de Rótulos	57
3.5	Análise de Complexidade de Tempo	58
4	AVALIAÇÃO EXPERIMENTAL	59
4.1	Coleções Textuais	59
4.2	Configuração Experimental e Critérios de Avaliação	60
4.3	Resultados	62
5	CONCLUSÕES E TRABALHOS FUTUROS	71
5.1	Principais Resultados e Contribuições	72
5.2	Limitações	73
5.3	Trabalhos Futuros	74
	REFERÊNCIAS	75

INTRODUÇÃO

Uma grande quantidade de dados tem sido disponibilizada na *Internet* por meio de redes sociais, páginas *web* e artigos científicos. Os dados provindos de tais fontes são oriundos de conversas informais, de dados abertos empresariais e governamentais, de pesquisas acadêmicas, entre outros. Parte substancial dessas informações estão em formato textual. Formato esse, que tem sido muito frequente, por exemplo, na propagação de informação por meio de redes sociais (GUPTA; PARVEEN, 2019).

Informações contidas em coleções textuais são úteis para diversos estudos como, a análise do comportamento humano e a análise de opinião pública (ANSTEAD; O'LOUGHLIN, 2014). Empresas também se interessam por informações provindas de tais dados, como informações de mercado. Devido à grande quantidade de informação textual disponível, extrair padrões e informações dessas coleções não é algo praticável de forma manual. Neste sentido, tarefas como a organização, análise e extração de conhecimento dessas coleções têm sido realizadas por meio de mecanismos, que nas últimas décadas têm ganhado importância (TAN *et al.*, 2019b; CAMPOS *et al.*, 2018; BIEMANN; MEHLER, 2014; WANG *et al.*, 2016). Um dos mecanismos mais utilizados, é a classificação automática de textos (SINOARA *et al.*, 2019; KOWSARI *et al.*, 2019; KIM *et al.*, 2020), que consiste em atribuir aos documentos determinados rótulos ou classes de maneira automática. A classificação automática de textos pode ser utilizada em diversas tarefas, como, recuperação e filtragem de documentos (LI *et al.*, 2018), categorização hierárquica de páginas *web* (WANG; GONG, 2008) e, detecção de *spam* (WU *et al.*, 2017).

Um das formas mais utilizadas para a classificação automática de textos é por meio do aprendizado multiclasse (WITTEN *et al.*, 2016). Para que esse modelo seja utilizado, é necessário que a coleção de treinamento seja completamente rotulada, e contenha documentos rotulados para todas as classes existentes na coleção. Essa atividade de rotulação da coleção de treinamento, na maioria das vezes, é realizada pelo usuário. Em

muitas situações, a coleção de treinamento é grande e torna a rotulação manual impraticável, além de exigir do usuário conhecimento a respeito de todas as classes presentes na coleção. Além disso, esse classificador só é capaz de classificar documentos pertencentes a alguma das classes que estejam presentes na coleção de treinamento.

Uma alternativa ao aprendizado multiclasse que visa minimizar as lacunas acima apresentadas é o aprendizado baseado em uma única classe, também conhecido pelo termo em inglês *One Class Learning* (OCL) (TAX, 2001; CHANDOLA; BANERJEE; KUMAR, 2009; AHMED; MAHMOOD; HU, 2016; TAN *et al.*, 2019b). Nesse modelo, o usuário precisa rotular apenas os documentos de seu interesse, inserindo-os na classe de documentos positivos. Então, esse conjunto de documentos positivos rotulados é utilizado para gerar um modelo, que ao receber um novo documento é capaz de distinguir se o mesmo faz parte dos documentos de interesse do usuário ou não. Essa técnica torna o uso do aprendizado de máquina mais viável a aplicações reais, principalmente naquelas envolvendo sensoriamento, como recomendação de notícias e materiais didáticos, ou *websensors* (LIU; DOLAN; PEDERSEN, 2010; MARCACINI *et al.*, 2017). Isso ocorre pois, nesses domínios, o usuário tem um alvo específico de interesse, assim, perderia tempo rotulando documentos para todas as possíveis classes (aprendizado multiclasse), ao invés de apenas rotular documentos de seu interesse, como ocorre no aprendizado baseado em uma única classe.

Ainda assim, rotular vários documentos de uma classe de interesse de forma a representá-la bem (como é necessário no OCL), pode demandar muito tempo e esforço. O cenário ideal, para que a classificação automática de textos se torne ainda mais viável, seria aquele que o usuário faz o menor esforço possível, ou seja, rotule poucos documentos de interesse sem impactar negativamente na performance de classificação do classificador. Tal cenário pode ser contemplado por meio da utilização do aprendizado semissupervisionado baseado em uma única classe (LIU *et al.*, 2002; ZHANG; LEE, 2005; YU; LI, 2007, 2007; LI *et al.*, 2009; MA; ZHANG, 2017; JASKIE; SPANIAS, 2019), também conhecido pelo termo em inglês *Positive and Unlabeled Learning* (PUL), onde são utilizados alguns documentos positivos rotulados (de interesse do usuário), e documentos não rotulados. Assim, o usuário tem um esforço mínimo, uma vez que a coleta de textos é uma tarefa fácil, bem como rotular poucos documentos.

A seguir é apresentada uma listagem de características com o intuito de explicitar as diferenças entre o OCL e o PUL. As informações completas desses dois modelos de aprendizado estão detalhadas na [Seção 2.3](#) e na [Seção 2.4](#).

- Aprendizado baseado em uma única classe (OCL):
 - utiliza em sua fase de treinamento apenas exemplos positivos rotulados;
 - necessita de uma quantidade de exemplos rotulados significativa para representar bem a distribuição dos exemplos da classe de interesse.

- Aprendizado semissupervisionado baseado em uma única classe (PUL):
 - utiliza poucos exemplos positivos rotulados e muitos exemplos não rotulados;
 - busca utilizar os exemplos não rotulados de forma a enriquecer seu conjunto de treinamento para obter melhores performances de classificação (em muitos casos extrai exemplos negativos dos exemplos não rotulados, bem como aumenta o conjunto de exemplos positivos).

Grande parte dos algoritmos para classificação automática de textos, incluindo os algoritmos PUL, fazem uso de representações baseadas no modelo espaço-vetorial para representar as coleções de documentos (AGGARWAL, 2014; SHALEV-SHWARTZ; BENDAVID, 2014; LIU *et al.*, 2002; LI; LIU, 2003; YU; HAN; CHANG, 2004; YU; ZHAI; HAN, 2003; LI *et al.*, 2009). Dessa forma, um documento é representado por um vetor, sendo cada posição do vetor uma dimensão, podendo cada dimensão representar os atributos ou termos dos documentos.

Outra forma de representar textos é por meio de grafos. Esse modelo de representação é utilizado em diversas aplicações como: classificação de textos (ROSSI; LOPES; REZENDE, 2016; WANG *et al.*, 2016), extração de palavras chave (NASAR; JAFFRY; MALIK, 2019), análise de sentimento (RABELO; PRUDÊNCIO; BARROS, 2012) e em processamento de linguagem natural (AMANCIO; JR; COSTA, 2012b; AMANCIO; JR; COSTA, 2012a; JR; LOPES; AMANCIO, 2018). Em algoritmos de aprendizado de máquina baseados em grafos, o grafo de documentos é uma das representações mais utilizadas para representar coleções textuais (SUBRAMANYA; BILMES, 2008; ANGELOVA; WEIKUM, 2006; LU; GETOOR, 2003; CHAKRABARTI *et al.*, 1997). Nesse tipo de grafo, uma aresta entre um par de documentos pode ser dada por arestas explícitas como *hyperlinks*, citações, coautorias, ou implícitas como a similaridade entre os documentos. Uma das formas mais frequentes de gerar as arestas de um grafo de documentos é por meio da similaridade, pois demonstra resultados superiores aos grafos construídos por meio de arestas explícitas, além de poder ser utilizado para qualquer coleção de documentos (ANGELOVA; WEIKUM, 2006). Esse tipo de grafo conecta com maior força pares de documentos cuja similaridade são maiores, já os pares de documentos com baixa similaridade são conectados com menor força, ou não são conectados.

Os grafos de documentos baseados em similaridade podem ser construídos de diferentes formas. Alguns dos algoritmos mais conhecidos para construção de grafos baseados em similaridades são: (i) k -Vizinhos Mais Próximos (k NN), (ii) k -Vizinhos Mais Próximos Mútuos (Mk NN), e (iii) ϵ -Vizinhança (ϵ N) (SOUSA; REZENDE; BATISTA, 2013; BERTON, 2016). Uma vez que a representação da coleção de documentos é realizada por meio de grafos, a classificação dos documentos não rotulados pode ser realizada por meio de algoritmos de propagação de rótulos em grafos. Alguns dos algoritmos mais comuns de

propagação de rótulos para grafos onde existem apenas um tipo de objeto (documentos) são: *Gaussian Fields and Harmonic Functions* (GFHF) (ZHU; GHAMRANI; LAFERTY, 2003) e *Learning With Local and Global Consistency* (LLGC) (ZHOU *et al.*, 2003).

Quando comparados com algoritmos baseados no modelo espaço-vetorial, algoritmos semissupervisionados de classificação automática de textos baseados em grafos têm apresentado performances de classificação superiores (ROSSI *et al.*, 2014; ROSSI, 2015; ROSSI; REZENDE; LOPES, 2015; ROSSI; LOPES; REZENDE, 2016; ROSSI; LOPES; REZENDE, 2017). No entanto, essa forma de representação ainda é pouco explorada em algoritmos PUL (JASKIE; SPANIAS, 2019; MA; ZHANG, 2017; YU; LI, 2007). Além disso, os métodos existentes de PUL abordam apenas um tipo de grafo, bem como apenas um algoritmo de propagação de rótulos.

Entre os métodos PUL existentes, diversas abordagens (YU; LI, 2007; YU; HAN; CHANG, 2004; YU; ZHAI; HAN, 2003; LI; LIU, 2003; LIU *et al.*, 2002) fazem uso de técnicas como *Self-Training* e *Expectation Maximization*, que quando aplicados na classificação transdutiva de textos podem degradar a performance de classificação (ROSSI; LOPES; REZENDE, 2017; ROSSI; LOPES; REZENDE, 2016).

Visando sanar as lacunas acima apresentadas, o objetivo desse projeto de mestrado é a proposta de um método que: (i) represente a coleção de documentos por meio de grafos no contexto PUL, onde essa representação ainda é pouco frequente; (ii) extraia documentos negativos de uma forma simples e intuitiva; (iii) utiliza algoritmos de propagação de rótulos para rotular os demais documentos no grafo ainda não rotulados, assim aproveitando a estrutura do grafo e evitando o uso de técnicas como *Self-Training* e *Expectation Maximization*.

O método proposto foi denominado *Label Propagation for Positive and Unlabeled Learning* (LP-PUL), sendo ele uma abordagem para PUL totalmente baseada em grafo, ou seja, desde a representação da coleção de documentos até a classificação de todos os documentos, o método proposto faz uso de grafos bem como de sua estrutura para extrair conhecimento e então rotular todos os documentos da coleção.

Além de propor um método totalmente baseado em grafo, foram explorados diferentes algoritmos de construção de grafos, a saber, k NN, Mk NN e ϵ N, e diferentes algoritmos de propagação de rótulos, a saber, GFHF e LLGC. Esses diferentes algoritmos foram explorados visando identificar o impacto que cada um, tanto de construção de grafo quanto de propagação de rótulos, causa na performance de classificação do método proposto.

1.1 Principais Resultados

A seguir são apresentados os principais resultados obtidos nesse projeto de mestrado.

- Foi proposto um algoritmo PUL baseado em grafo que efetivamente faz uso de documentos não rotulados para melhorar a performance de classificação. Além disso, durante a avaliação foi comprovado que o método proposto ultrapassa a performance de classificação dos algoritmos baseados no modelo espaço-vetorial, bem como os baseados em grafo.
- Foi conduzida uma avaliação rigorosa utilizando 17 coleções textuais, onde o método proposto foi comparado com algoritmos tradicionais OCL e PUL. Essa avaliação permitiu mostrar os pontos fracos dos algoritmos OCL e PUL existentes e as vantagens do método proposto. Também é discutido o comportamento dos algoritmos para diferentes quantidades de documentos rotulados.
- Foi conduzida uma avaliação referente ao impacto que o uso de diferentes algoritmos de propagação de rótulos, bem como o uso de diferentes algoritmos de construção de grafos causam no método proposto.
- Foi apresentada uma comparação entre os aprendizados supervisionado e semissupervisionado baseado em uma única classe, com o intuito de apresentar os ganhos e as perdas na performance de classificação quando são utilizados documentos não rotulados.
- Foi submetido um artigo para publicação na revista *Information Sciences*, contendo os resultados científicos obtidos nesse projeto de mestrado. No momento da escrita dessa dissertação o referido artigo está em processo de revisão.

1.2 Organização do Texto

O restante do texto está organizado da seguinte forma. No [Capítulo 2](#) é apresentada a fundamentação teórica necessária para o entendimento desse projeto de mestrado, sendo abordado as mais comuns formas de limpeza e padronização da coleção textual, as formas de representação das coleções de documentos, o conceito de grafos, bem como os métodos de construção de grafos que são utilizados nesse projeto, e por fim, são apresentados os trabalhos relacionados. No [Capítulo 3](#) é apresentado o método proposto (LP-PUL), sendo que cada etapa do método recebeu uma seção exclusiva para a explicação da mesma. Ainda, no final desse capítulo, é feita a análise de complexidade de tempo do LP-PUL. A avaliação experimental, composta pelas comparações do LP-PUL com os demais métodos

e pela análise do uso de diferentes algoritmos em cada etapa do LP-PUL, é apresentada no [Capítulo 4](#). Finalmente, no [Capítulo 5](#) são feitas as conclusões e apresentados os trabalhos futuros.

FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

No decorrer deste capítulo, são apresentados os conceitos e os trabalhos relacionados ao tema de pesquisa deste projeto de mestrado. Na [Seção 2.1](#) são abordados os assuntos relacionados a representação de coleções textuais, que se iniciam na [Subseção 2.1.1](#) onde são apresentados os métodos mais comuns para limpeza e padronização de textos, seguidos pelos dois métodos de representação textual considerados nesse projeto, a saber, modelo espaço-vetorial e modelo baseado em grafos, abordados respectivamente na [Subseção 2.1.2](#) e [Subseção 2.1.3](#). Na seção seguinte, [Seção 2.2](#), são abordados assuntos relacionados ao aprendizado de máquina semissupervisionado, envolvendo algoritmos de aprendizado transdutivo baseados no modelo espaço vetorial e em grafos. Além disso, no final desse capítulo são apresentados os dois modelos de aprendizado que fazem parte do tema desse projeto de mestrado, bem como os trabalhos relacionados, a saber, Aprendizado Baseado em Uma Única Classe e Aprendizado Semissupervisionado Baseado em Uma Única Classe, ambos apresentados respectivamente na [Seção 2.3](#) e [Seção 2.4](#).

2.1 Representações de Coleções de Textos

Algoritmos de aprendizado de máquina usualmente necessitam que os textos estejam representados de forma estruturada, para que possam interpretar e processar os mesmos. A qualidade da representação da coleção de texto está diretamente relacionada a qualidade dos resultados do algoritmo. Sendo assim, inicialmente é realizada uma limpeza e padronização dos textos, e em seguida é utilizado algum dos modelos mais frequentes para representação textual, a saber, espaço-vetorial e por meio de grafos ([AGGARWAL, 2014](#); [ROSSI; LOPES; REZENDE, 2017](#)). A seguir são apresentadas a limpeza e padronização de textos, e ambas formas de representação textual mencionadas acima.

2.1.1 Limpeza e Padronização de Textos

Considere $\mathcal{D} = \{d_1, d_2, \dots, d_n\}$ o conjunto de documentos de uma coleção textual com n documentos. Inicialmente cada documento da coleção é pré-processado. O pré-processamento consiste no processo completo de gerar uma representação estruturada para o texto inicial. Dentro do pré-processamento existem as etapas de (i) padronização, onde a caixa dos caracteres é padronizada, e ainda, as palavras são simplificadas (radicalizadas), e (ii) limpeza, também conhecida como remoção de *stopwords*, onde palavras infrequentes ou muito frequentes são removidas, além de caracteres indesejados (ROSSI, 2015). Nesse ponto o texto está pronto para ser representado de forma estruturada. A seguir são apresentadas algumas das formas de representação estruturada de textos.

2.1.2 Modelo Espaço-Vetorial

A representação de documentos por meio do modelo espaço-vetorial é comumente utilizada na área de Aprendizado de Máquina (AGGARWAL, 2014; SHALEV-SHWARTZ; BEN-DAVID, 2014). Nesse modelo, a representação de um documento é dada por um vetor, no qual cada dimensão usualmente representa um termo (SALTON, 1989). Um termo pode ser uma palavra, uma sequência de palavras, ou um conjunto de palavras, geradas por meio da informação textual dos documentos. Considere o conjunto de termos $\mathcal{T} = \{t_1, t_2, \dots, t_m\}$. As dimensões podem também representar atributos como localização geográfica, autores, entre outros (ROSSI, 2015). Neste projeto é considerado que as dimensões dos vetores representam termos.

A geração de termos e atributos influencia diretamente os resultados obtidos na classificação automática de textos. A forma mais simples é a utilização de palavras como atributos. Tal forma de representação recebe o nome de “saco de palavras”, do termo em inglês *bag-of-words* (BoW). Para definir os pesos dos termos e atributos, os quais serão utilizados pelos algoritmos de classificação automática de textos para cálculos como o de similaridade entre os documentos, existem alguns métodos, sendo os mais frequentes apresentados a seguir: (i) binário, onde o valor do termo será 1 caso o mesmo esteja presente no documento, e 0 caso contrário; (ii) frequência do termo, em inglês *term frequency* (*tf*), onde o valor do termo é a frequência do mesmo no documento; e finalmente (iii) frequência do termo ponderada pelo inverso da frequência de documento, em inglês *term frequency - inverse document frequency* (*tf-idf*), onde a frequência do termo é ponderada pelo inverso da quantidade de documentos em que o termo ocorre (MANNING; RAGHAVAN; SCHÜTZE, 2010; AGGARWAL, 2018).

Ao utilizar o modelo espaço-vetorial, considerando que cada dimensão representa um termo (BoW), gera-se uma representação que tende a ter alta esparsidade e dimensionalidade devido ao grande conjunto de palavras encontradas em uma coleção de textos (dimensionalidade), e ao mesmo tempo, o conjunto de palavras de um documento é pe-

queno em relação ao conjunto de palavras de toda a coleção (esparsidade). Além disso, o uso de termos como dimensões faz com que a ordem das palavras sejam perdidas e também, pode apresentar termos semanticamente relacionados porém com grafias diferentes (AGGARWAL, 2018).

Algumas técnicas podem ser utilizadas visando a diminuição da dimensionalidade, assim diminuindo o custo computacional e transformando dois ou mais termos semanticamente relacionados em apenas um atributo. Algumas dessas técnicas são: *Probabilistic Latent Semantic Analysis* (PLSA), *Latent Dirichlet Allocation* (LDA) e *Non-negative Matrix Factorization* (NMF) (XIA *et al.*, 2019). Na maioria das técnicas de redução de dimensionalidade, as dimensões passam a representar tópicos¹, sendo que o valor da dimensão corresponde ao grau de pertinência do respectivo tópico ao documento em questão. Aplicar tais técnicas resultam em um número baixo e fixo de dimensões, e representações espaciais contínuas tanto para documentos como para termos, no entanto são caros computacionalmente.

Modelos de linguagens também são capazes de gerar representações textuais com um número baixo e fixo de dimensões. As representações de termos geradas a partir de modelos de linguagens são amplamente conhecidos como *word embeddings*. Dois métodos frequentemente utilizados para a geração de *word embeddings* são: *word2vec* (MIKOLOV *et al.*, 2013a) e *fastText* (BOJANOWSKI *et al.*, 2017). Ambos métodos são baseados na predição de contexto por meio de abordagens baseadas em redes neurais. Esses métodos apresentam arquitetura simples, treinamento custoso e alta qualidade em muitas aplicações de mineração de textos (FIOK *et al.*, 2020; MIKOLOV *et al.*, 2013a).

Uma vez obtidas as *word embeddings*, algumas formas simples de obter a representação para documentos é por meio de operações com suas respectivas *word embeddings*, como soma, média, soma ponderada, mínimo e máximo (PITA; PAPPA, 2018). No entanto essas operações acarretam na perda da ordem das palavras.

Ainda assim, com todos esses meios sofisticados de representação textual por meio do modelo espaço-vetorial, os resultados apresentados pela BoW são competitivos (AGGARWAL, 2018).

2.1.3 Grafos

A ciência dos grafos² é uma disciplina que abrange outras ciências, como a ciência social, natural, da computação e da engenharia (FORTUNATO; HRIC, 2016). Grafos são utilizados para a representação de sistemas reais, tais como redes sociais, sistemas biológicos, redes de transporte e de comunicação, como a *Internet*. São utilizados tam-

¹ Um tópico representa um conjunto de termos que são relacionados semanticamente.

² Os grafos também são conhecidos como redes. Nesse projeto de mestrado usaremos sempre que possível o termo grafo, salvos alguns momentos em que o termo rede for mais usual.

bém na representação textual em tarefas como classificação de textos (ROSSI; LOPES; REZENDE, 2016; WANG *et al.*, 2016), extração de palavras chave (NASAR; JAFFRY; MALIK, 2019), análise de sentimentos (RABELO; PRUDÊNCIO; BARROS, 2012), processamento de linguagem natural, envolvendo por exemplo desambiguação de palavras e desambiguação de nomes (AMANCIO; JR; COSTA, 2012b; AMANCIO; JR; COSTA, 2012a; JR; LOPES; AMANCIO, 2018). Recentemente, grafos têm recebido muita atenção na área de aprendizado de máquina (NEWMAN, 2016; ROSSI; LOPES; REZENDE, 2014; GUÉDON; VERSHYNIN, 2016; COREL *et al.*, 2018).

Os grafos podem ser formalmente definidos como uma tripla $N = \langle \mathcal{O}, \mathcal{E}, \mathcal{W} \rangle$, contendo um conjunto de objetos $\mathcal{O}$ (também conhecidos como vértices), um conjunto de arestas $\mathcal{E}$ entre os objetos (que representam as relações entre os objetos), e um conjunto de pesos $\mathcal{W}$ das arestas. Dado um par de objetos o_i e $o_j \in \mathcal{O}$, a aresta entre eles, caso exista, é representada por e_{o_i, o_j} (ROSSI, 2015). As arestas de um grafo podem ter ou não direção. Um grafo cujas arestas não possuam direção é denominado grafo não direcionado, por exemplo a rede social, onde uma aresta que representa amizade entre dois indivíduos não necessita de direção, pois se o_i é amigo de o_j , então consequentemente o_j é amigo de o_i . Já quando as arestas possuem direção, o grafo é denominado, grafo direcionado. Em um grafo direcionado, cada aresta possui um ponto de saída e um ponto de chegada, sendo estes definidos com base na ordem com que os objetos aparecem na aresta, ou seja, e_{o_i, o_j} representa uma aresta que sai do objeto o_i e vai para o objeto o_j . A rede de citação é um exemplo de grafo direcionado, uma vez que para um artigo o_i citar outro o_j , o_i necessita ser mais recente, sendo impossível o artigo o_j , no momento em que foi escrito citar o_i que ainda não existia.

Caso uma aresta conecte um objeto a ele mesmo, esta é denominada autoaresta. Uma aresta que conecta mais de dois objetos é denominada hiperaresta e podem ser encontradas em hipergrafos, no entanto, hipergrafos e autoarestas são pouco frequentes na área de aprendizado baseado em grafos.

Os grafos podem ter diferentes escalas de pesos em suas arestas de acordo com a aplicação em que se encontram, e podem ter também um peso fixo, sendo denominados respectivamente grafo ponderado e não ponderado. No grafo não ponderado, a aresta e_{o_i, o_j} tem seu peso $w_{o_i, o_j} = 1$ se $\exists e_{o_i, o_j} \in \mathcal{E}$ e 0 caso contrário. Assim todas as arestas possuem a mesma importância. Redes sociais e redes de citação costumam ser não ponderadas. Caso o grafo seja ponderado, o peso w_{o_i, o_j} poderá ser diferente para cada aresta $e_{o_i, o_j} \in \mathcal{E}$, de acordo com os objetos o_i e o_j que a compõe, e algum critério específico. É frequente o uso da similaridade ou da distância entre os objetos como peso da aresta. Considerando um grafo de documentos, os pesos das arestas podem ser a similaridade entre eles, já em uma rede de computadores, o peso das arestas pode ser dado pela banda de conexão entre os dispositivos. Uma outra possibilidade é ao invés de pesos, as arestas conterem vetores

com informações relacionadas a aresta (ROSSI, 2015).

Se o conjunto $\mathcal{O}$ for composto por apenas um tipo de objeto, o grafo é denominado homogêneo. Caso $\mathcal{O}$ seja composto por mais de um tipo de objeto, o grafo é denominado heterogêneo, e consequentemente suas arestas também podem ser diferentes de acordo com os tipos de objetos que elas conectam. Uma ilustração de grafos homogêneo e heterogêneo é apresentada na Figura 1.

Figura 1 – Exemplos de grafos homogêneo e heterogêneo.

Fonte: Elaborada pelo autor.

Grafos homogêneos, quando utilizados para representação textual, são encontradas com objetos representando sílabas, e com arestas ligando sílabas que fazem parte das mesmas palavras (SOARES; CORSO; LUCENA, 2005); com objetos representando sentenças e arestas ligando sentenças similares (MIHALCEA; TARAU, 2004); com objetos representando termos e arestas ligando termos similares com base na coleção de documentos (MARCACINI; REZENDE, 2010); com objetos representando parágrafos e arestas ligando parágrafos similares com base em seu conteúdo (ARRUDA *et al.*, 2019); com objetos representando documentos e arestas ligando documentos similares (CAI; WANG; HE, 2009). Essa última, onde os objetos são representados por documentos, recebe o nome de grafo de documentos, e é a forma de representação considerada nesse projeto de mestrado.

Grafos heterogêneos também estão presentes na literatura representando textos com diferentes tipos de objetos e arestas. Por exemplo, pode-se citar um grafo de artigos, onde os objetos podem ser artigos, autores, termos e locais de publicação, com arestas do tipo **escrito por** e **escreve** conectando artigos e autores, **publicado em** e **publica** conectando artigos e locais de publicação, **contém** e **contido em** conectando termos e artigos (WANG *et al.*, 2015).

De acordo com o domínio em que está inserido, grafos apresentam características topológicas específicas, determinando sua conectividade e influenciando altamente o dinamismo dos processos executados no grafo (COSTA *et al.*, 2007). Propriedades podem ser extraídas dos grafos por meio de medidas específicas, sendo algumas delas: grau médio, força, caminhos mínimos, grau de intermediação, densidade, clique, coeficiente de agrupamento e grau de proximidade (COSTA *et al.*, 2007; ALMEIDA, 2009; VALEJO, 2014). Uma importante propriedade presente em muitos grafos reais, é a estrutura de comunidades, sendo essa uma concentração elevada de arestas dentro de grupos especiais de objetos, formando assim estruturas modulares locais conhecidas também como sub-grafos ou partições (GIRVAN; NEWMAN, 2002).

Nesse projeto de mestrado, o grafo de documentos é utilizado para representar as coleções textuais, pois é uma das maneiras mais comuns de representação de coleções textuais por meio de grafos (SUBRAMANYA; BILMES, 2008; ANGELOVA; WEIKUM, 2006; LU; GETOOR, 2003; CHAKRABARTI *et al.*, 1997). As arestas do grafo de documentos podem ser dadas de maneira explícita, ou seja, por meio de *hyperlinks*, citações, coautorias, ou podem ser geradas implicitamente por meio da similaridade entre os documentos. Os grafos construídos com base na similaridade entre os documentos podem ser utilizados para representar qualquer coleção de documentos, além de apresentarem melhores resultados que grafos construídos por meio de arestas explícitas (ANGELOVA; WEIKUM, 2006). A seguir, são apresentados os algoritmos mais frequentes para a construção de grafos baseados em similaridade (SOUSA; REZENDE; BATISTA, 2013).

Grafos Baseados em Similaridade

Grafos baseados em similaridade têm a premissa de conectar com maior força objetos que apresentam maior similaridade, enquanto objetos distintos são conectados com menor força, ou ainda não são conectados. Visando a construção de um grafo de documentos baseado em similaridade, a medida comumente utilizada é a similaridade cosseno, apresentada na Equação 2.1 (TAN *et al.*, 2019b).

$$\text{cosseno}(\mathbf{w}_{d_i}, \mathbf{w}_{d_j}) = \frac{\mathbf{w}_{d_i} \cdot \mathbf{w}_{d_j}}{\|\mathbf{w}_{d_i}\| \cdot \|\mathbf{w}_{d_j}\|} = \frac{\sum_{t_k \in \mathcal{T}} w_{d_i, t_k} \cdot w_{d_j, t_k}}{\sqrt{\sum_{t_k \in \mathcal{T}} (w_{d_i, t_k})^2} \cdot \sqrt{\sum_{t_k \in \mathcal{T}} (w_{d_j, t_k})^2}}, \quad (2.1)$$

onde $\mathbf{w}_{d_i}$ representa o vetor de atributos do documento d_i , $\mathcal{T}$ é o conjunto dos termos e w_{d_i, t_k} é o peso do termo t_k no documento d_i .

Dada uma medida de similaridade entre documentos, existem diferentes formas de construir o grafo de documentos baseado em similaridade. A seguir são apresentadas as mais utilizadas na literatura (BERTON; LOPES; VEGA-OLIVEROS, 2018; BERTON, 2016; SILVA; ZHAO, 2016; SOUSA; REZENDE; BATISTA, 2013).

k -Vizinhos Mais Próximos: no algoritmo de geração de grafo k -Vizinhos Mais Próximos, do termo em inglês *k-Nearest Neighbors* (k NN), é inicialmente calculado a semelhança entre cada possível par de objetos por meio de alguma métrica (podendo ser a similaridade cosseno). Então, cada objeto irá se conectar aos seus k objetos mais semelhantes. Ao ser gerada uma aresta, seu respectivo peso pode ser 1, gerando um grafo não ponderado. Pode-se também receber o valor da similaridade entre os documentos como peso da aresta, assim formando um grafo ponderado. Esse algoritmo possui apenas um parâmetro, o k , que deve ser definido previamente, e tem a finalidade de limitar a quantidade de conexões de um objeto.

Existe uma variação do grafo k NN denominada k -Vizinhos Mais Próximos Mútuos, em inglês *Mutual k-Nearest Neighbors* (Mk NN). Esse grafo se diferencia do k NN pois serão conectados apenas os vizinhos mútuos, ou seja, tendo encontrado os k vizinhos mais próximos para cada objeto $o_n \in \mathcal{O}$, o_i e o_j só serão conectados se o_i for um dos vizinhos mais próximos de o_j , e o_j for um dos vizinhos mais próximos de o_i .

O grafo Mk NN é mais restritivo que o k NN, gerando uma menor quantidade de arestas. O algoritmo k NN tem a característica de construir grafos muito densos e com muita variação no grau dos objetos, de forma que ele força a conexão entre um objeto e seus k vizinhos mesmo que eles não sejam tão similares.

ϵ -Vizinhança: na construção de um grafo por meio do algoritmo ϵ -Vizinhança, do termo em inglês *ϵ -Neighborhood* (ϵ N), é necessário definir o parâmetro de entrada ϵ , que representa um limiar (valor mínimo) de similaridade entre dois objetos para esses serem conectados. Dessa forma, o objeto o_i será conectado a todo $o_j \in \mathcal{O}$ se $\text{sim}(o_i, o_j) \geq \epsilon$, onde $\text{sim}()$ representa uma função de similaridade.

A principal desvantagem do grafo ϵ -Vizinhança é a adequação do parâmetro ϵ , que com pequenas variações pode gerar desde grafos com muitos componentes desconexos até grafos com muitas conexões e consequentemente objetos de graus elevados.

A complexidade de tempo para a geração de um grafo de documentos baseado em similaridade, corresponde ao tempo para calcular a similaridade entre todos os pares de documentos, ou seja, $O(|\mathcal{D}|^2)$. No caso dos grafos k NN, Mk NN e ϵ N, um tempo adicional é necessário para manter os objetos mais similares ordenados, que é $O(\log |\mathcal{D}|)$. Nesse caso, a complexidade de tempo total para construir um grafo de documentos baseado em similaridade é $O(|\mathcal{D}|^2 \log |\mathcal{D}|)$.

2.2 Aprendizado Semissupervisionado

A quantidade de dados em formato textual disponível na Internet é gigantesca e cresce a cada dia por meio de páginas *web*, redes sociais, artigos científicos e *e-mails*. (GUPTA; PARVEEN, 2019). Devido a grande quantidade de textos, torna-se inviável a rotulação manual por necessitar de muitas pessoas e tempo. Dessa forma, no aprendizado semissupervisionado, o objetivo é melhorar a performance de classificação em relação ao aprendizado supervisionado por meio do uso de exemplos não rotulados. Com 90% menos exemplos rotulados, em algumas situações, é possível obter uma performance de classificação próxima ao obtido por algoritmos de aprendizado de máquina supervisionado (ROSSI; LOPES; REZENDE, 2014).

O sucesso obtido por técnicas do aprendizado semissupervisionado é atribuído a certas suposições em relação a distribuição dos dados, que devem ser verdadeiras, a saber (Chen; Wang, 2011):

- Premissa de agrupamento: exemplos no mesmo grupo são mais prováveis de pertencer à mesma classe;
- Separação de baixa densidade: a fronteira de decisão deve estar situada em uma região de baixa densidade;
- Premissa *smoothness* do aprendizado semissupervisionado: se dois pontos $\mathbf{x}_1, \mathbf{x}_2$ estão próximos em uma região de alta densidade, logo eles devem ter saídas y_1, y_2 correspondentes;
- Premissa *manifold*: dados de alta dimensionalidade frequentemente podem ser representados por múltiplas representações de baixa dimensionalidade.

O aprendizado de máquina semissupervisionado pode ser dividido em dois tipos, a saber, (i) o aprendizado indutivo (AMINI; USUNIER, 2015; SUBRAMANYA; TALUKDAR, 2014; ZHU; GOLDBERG, 2009; CHAPELLE; SCHOLKOPF; ZIEN, 2006), onde um modelo de classificação é gerado por meio de exemplos rotulados e exemplos não rotulados, para então classificar exemplos nunca vistos, e (ii) o aprendizado transdutivo, descrito a seguir.

No cenário do aprendizado transdutivo, considerando a classificação de textos, $\mathcal{D} = \mathcal{D}^L \cup \mathcal{D}^U$, onde $\mathcal{D}^L$ representa o conjunto dos documentos rotulados, e $\mathcal{D}^U$ representa o conjunto dos documentos não rotulados. Além disso, $\mathcal{Y} = \{y_1, y_2, \dots, y_n\}$ é utilizado para denotar os rótulos de todos os documentos, ou seja, $\mathcal{Y} = \mathcal{Y}^L \cup \mathcal{Y}^U$, onde $\mathcal{Y}^L$ representa os rótulos dos documentos rotulados e $\mathcal{Y}^U$ representa os rótulos atribuídos aos documentos não rotulados durante o processo de classificação.

No aprendizado transdutivo não é necessária a indução de um modelo de classificação, pois os rótulos para os documentos não rotulados são estimados diretamente. O aprendizado transdutivo tem o objetivo de gerar a função $F : \mathcal{D}^L \cup \mathcal{D}^U \rightarrow \mathcal{Y}^L \cup \mathcal{Y}^U$, sendo F gerado a partir de documentos rotulados e não rotulados, visando ser um preditor sobre os documentos não rotulados.

O aprendizado transdutivo tem acesso aos documentos que serão classificados durante o aprendizado, sendo assim uma quantidade finita de documentos. Dessa forma, é possível explorar melhor as características da coleção textual, melhorando assim a performance de classificação, além de ter também um espaço finito de hipóteses (CHAPELLE; SCHOLKOPF; ZIEN, 2006). Um esquema do aprendizado transdutivo, aplicado na classificação automática de textos, é ilustrado na Figura 2.

Figura 2 – Ilustração do processo de classificação automática de textos por meio do aprendizado transdutivo.

Fonte: Adaptada de Rossi (2015)³.

A seguir são apresentados alguns exemplos de aplicações onde o aprendizado transdutivo pode ser aplicado devido toda coleção de documentos ser conhecida (ROSSI, 2015):

- **Realimentação de relevância:** a realimentação de relevância é utilizada na recuperação de informação, sendo que o usuário rotula os documentos que são retornados como documentos relevantes e não relevantes. Feito isso, o aprendizado transdutivo é aplicado, sendo retornado ao usuário os documentos relevantes à consulta (BAEZA-YATES; RIBEIRO-NETO, 1999).
- **Filtro de notícias:** para realizar a filtragem de notícias, o usuário pode informar quais notícias são de interesse. Assim, através do aprendizado transdutivo, é retornado as notícias do conjunto que são relevantes ao usuário (JOACHIMS, 2003).

³ O ícone de engrenagens (*settings*) é de Gregor Cresnar, baixado do site <www.flaticon.com>.

- **Organização de coleção de documentos:** para realizar a organização de uma coleção de documentos por meio do aprendizado transdutivo, basta o usuário rotular alguns documentos para cada uma das categorias definidas (ROSSI; LOPES; REZENDE, 2016).
- **Mineração de opinião/Análise de sentimento:** considerando que comentários a respeito de produtos por exemplo, sejam coletados de páginas *web*, ou redes sociais como *Twitter*, o aprendizado transdutivo poderá ser aplicado, sendo necessário previamente apenas que o usuário rotule alguns comentários como positivos, negativos ou neutros (RABELO; PRUDÊNCIO; BARROS, 2012).

Diversos métodos foram propostos para a classificação transdutiva, sendo os primeiros baseados no modelo espaço-vetorial. Esses não foram capazes, na maioria das vezes, de melhorar a performance de classificação por meio de exemplos não rotulados. Isso ocorre devido à violação das bases de dados à distribuição dos dados (COZMAN; COHEN; CIRELO, 2003; CHAPELLE; SCHOLKOPF; ZIEN, 2006; ZHU; GOLDBERG, 2009). Mais tarde, algoritmos transdutivos baseados em grafos foram propostos. Esses demonstraram melhor performance de classificação quando comparados aos antecessores (algoritmos transdutivos baseados no modelo espaço-vetorial). Abaixo, são apresentados o aprendizado transdutivo no modelo espaço-vetorial e o aprendizado transdutivo em grafos.

2.2.1 *Aprendizado Transdutivo no Modelo Espaço-Vetorial*

Algoritmos como *Self-Training*, *Co-Training*, *Expectation Maximization* e *Transductive Support Vector Machines* são algoritmos tradicionais da classificação transdutiva, além de serem utilizados na classificação automática de textos (ROSSI, 2015; ZHU; GOLDBERG, 2009; CHAPELLE; SCHOLKOPF; ZIEN, 2006). Abaixo, esses algoritmos são descritos:

Self-Training: o *Self-Training* (YAROWSKY, 1995) é um método iterativo de classificação transdutiva. Inicialmente esse método treina um modelo de classificação utilizando apenas os documentos rotulados. Em seguida, o modelo é utilizado para classificar os documentos não rotulados, sendo que aqueles rotulados com maior confiança são atribuídos ao conjunto de documentos rotulados. Esse novo conjunto de documentos rotulados é utilizado então para treinar um novo modelo de classificação, que será utilizado para classificar os documentos ainda não rotulados. Esse procedimento iterativo é realizado até que não existam mais documentos sem rótulos.

O *Self-Training* tem como premissa que os documentos rotulados com alta confiança estão rotulados corretamente. No entanto, isso só é verdade para conjuntos de dados onde os grupos estão bem separados (ZHU; GOLDBERG, 2009). Dessa forma,

se o conjunto não estiver na forma ideal acima apresentada, documentos incorretamente rotulados serão adicionados ao conjunto de documentos rotulados, que serão utilizados no treinamento do modelo, degenerando-o a cada iteração do algoritmo.

Co-Training: o método *Co-Training* foi proposto por [Blum e Mitchell \(1998\)](#). Esse método se assemelha ao *Self-Training*, no entanto, é aplicado para conjuntos de dados com mais de uma visão. Uma página *web*, por exemplo, pode ser descrita em duas visões, sendo a primeira o conteúdo da própria página, e o conteúdo das páginas que apontam para ela como a segunda visão ([ZHU; GOLDBERG, 2009](#)).

Considerando que o conjunto tenha duas visões, serão necessários dois modelos de classificação. O primeiro modelo f^1 será induzido por meio do conjunto de documentos rotulados $\mathcal{D}^{L1}$ da primeira visão, e o segundo modelo f^2 será induzido por meio do conjunto de documentos rotulados $\mathcal{D}^{L2}$ da segunda visão. Os modelos induzidos classificarão os conjuntos de documentos não rotulados, $\mathcal{D}^{U1}$ e $\mathcal{D}^{U2}$, da respectiva visão na qual o modelo foi treinado. Os documentos rotulados com maior confiança em cada modelo, ou seja, f^1 e f^2 , serão adicionados no conjunto de documentos rotulados da outra visão, ou seja, serão adicionados respectivamente à $\mathcal{D}^{L2}$ e $\mathcal{D}^{L1}$. Esse processo se repete até que o conjunto de documentos não rotulados esteja vazio.

A vantagem do *Co-Training* sobre o *Self-Training* está na possibilidade de melhorar a performance de classificação por meio das visões da base de dados. Enquanto no *Self-Training*, documentos podem ser rotulados com baixa confiança devido só ter uma visão, no *Co-Training* uma segunda visão pode ter uma alta confiança na classificação do mesmo documento, sendo assim adicionado ao conjunto de documentos rotulados.

Para utilizar uma base de dados no *Co-Training*, assume-se que a mesma possui duas visões, sendo elas independentes, e suficientes para realizar boas classificações. No entanto, na prática, bases de dados com esses requisitos são difíceis de serem encontradas.

Expectation Maximization: no *Expectation Maximization* (EM) ([DEMPSTER; LAIRD; RUBIN, 1977](#)), diferentemente dos algoritmos anteriores, não é necessário induzir modelos de classificação. Nesse algoritmo, a informação de classe dos documentos não rotulados são ajustadas iterativamente. Considere como “componente” cada posição do vetor de informação de classe dos documentos, sendo que cada componente está associado a uma classe. Além disso, o objetivo é definir as probabilidades de cada termo para os componentes por meio dos documentos rotulados e não rotulados.

Uma abordagem para classificação transdutiva de textos que combina EM com a técnica Multinomial Naïve Bayes é apresentada em [Nigam et al. \(2000a\)](#), onde a

probabilidade de ocorrer uma classe dado um documento ou a informação de classe desse documento, é definida de acordo com a regra de Bayes:

$$P(c_j|d_i; \mathbf{F}(\mathcal{T})) = \frac{P(c_j|\mathbf{F}(\mathcal{T}))P(d_i|c_j; \mathbf{F}(\mathcal{T}))}{P(d_i|\mathbf{F}(\mathcal{T}))}, \quad (2.2)$$

onde $P(c_j|\mathbf{F}(\mathcal{T}))$ é a probabilidade de ocorrer a classe c_j levando em consideração os pesos dos termos para a mesma classe. Por sua vez,

$$P(c_j|\mathbf{F}(\mathcal{T})) = \frac{1 + \sum_{d_i \in \mathcal{D}} P(c_j|d_i; \mathbf{F}(\mathcal{T}))}{|\mathcal{C}| + |\mathcal{D}|}. \quad (2.3)$$

O algoritmo Naïve Bayes tem a premissa de que cada palavra de um documento é gerada de forma independente das outras, o que não é verdade. Entretanto tal suposição simplifica o cálculo, pois não precisamos computar probabilidades condicionais. Assim, a probabilidade de ocorrer um documento d_i dada a classe c_j é dada pela probabilidade de ocorrer seus termos para a mesma classe, ou seja:

$$P(d_i|c_j; \mathbf{F}(\mathcal{T})) = \prod_{t_k \in \mathcal{T}, w_{d_i, t_k} > 0} P(t_k|c_j; \mathbf{F}(\mathcal{T})). \quad (2.4)$$

Já o termo normalizador, $P(d_i|\mathbf{F}(\mathcal{T}))$, é calculado de acordo com a seguinte equação:

$$P(d_i) = \sum_{c_j \in \mathcal{C}} P(c_j)P(d_i|c_j). \quad (2.5)$$

Por fim, a seguinte equação é utilizada para calcular a probabilidade de ocorrer um termo t_i em uma classe c_j com base nas atuais informações de classe dos documentos:

$$P(t_i|c_j; \mathbf{F}(\mathcal{T})) = \frac{1 + \sum_{d_k \in \mathcal{D}} w_{t_i, d_k} P(c_j|d_k; \mathbf{F}(\mathcal{T}))}{|\mathcal{T}| + \sum_{t_l \in \mathcal{T}} \sum_{d_m \in \mathcal{D}} w_{t_l, d_m} P(c_j|d_m; \mathbf{F}(\mathcal{T}))}. \quad (2.6)$$

Nessa abordagem, inicialmente são calculadas as probabilidades dos termos para cada componente com base nas informações de classe dos documentos rotulados. Logo após, inicia-se um laço onde as informações de classe dos documentos não rotulados são calculadas com base nas probabilidades dos termos para cada componente. Em seguida as probabilidades dos termos para cada componente são calculadas novamente, porém considerando as informações de classe dos documentos rotulados e não rotulados. Esse laço é executado até a convergência das informações de classe dos documentos não rotulados.

Ainda em [Nigam et al. \(2000a\)](#), são propostas duas extensões visando melhorar a performance de classificação. A primeira extensão pondera a importância dos

dados não rotulados. Isso foi proposto devido ao fato de que a abordagem EM (sem nenhuma ponderação) se comportaria como um agrupamento não supervisionado em situações que são utilizados poucos exemplos rotulados em relação a quantidade de exemplos não rotulados. Para ponderar a importância dos dados rotulados, ou seja, o seu peso nas definições das probabilidades, foi proposta a utilização da função $\Lambda(d_i)$, dada por:

$$\Lambda(d_i) = \begin{cases} \lambda & \text{se } d_i \in \mathcal{D}^U \\ 1 & \text{se } d_i \in \mathcal{D}^L \end{cases} \quad (2.7)$$

Assim, as Equações 2.3 e 2.6 são respectivamente reescritas nas Equações 2.8 e 2.9.

$$P(c_j | \mathbf{F}(\mathcal{T})) = \frac{1 + \sum_{d_i \in \mathcal{D}} \Lambda(d_i) P(c_j | d_i; \mathbf{F}(\mathcal{T}))}{|\mathcal{C}| + |\mathcal{D}^L| + \lambda |\mathcal{D}^U|}. \quad (2.8)$$

$$P(t_i | c_j; \mathbf{F}(\mathcal{T})) = \frac{1 + \sum_{d_k \in \mathcal{D}} w_{t_i, d_k} P(c_j | d_k; \mathbf{F}(\mathcal{T}))}{|\mathcal{T}| + \sum_{t_l \in \mathcal{T}} \sum_{d_m \in \mathcal{D}} w_{t_l, d_m} P(c_j | d_m; \mathbf{F}(\mathcal{T}))}. \quad (2.9)$$

A segunda extensão propõe o uso de vários componentes por classe, pois em uma classe podem existir diferentes tópicos, assim, utilizar múltiplos componentes possibilita definir as probabilidades dos termos para os tópicos da classe. Considere $\mathcal{S}^{c_j}$ o conjunto dos componentes da classe c_j , assim, utilizando apenas um componente, $P(c_j | d_i; \mathbf{F}(\mathcal{T})) = 1$ se $d_i \in \mathcal{D}^L$ e $y_{d_i, c_j} = 1$. Ao utilizar mais de um componente, é possível atribuir para cada componente de uma classe valores entre 0 e 1. A probabilidade de ocorrer uma classe c_j dado um documento d_i quando são utilizados múltiplos componentes por classe, é dada pela soma das probabilidades de cada componente $s_k \in \mathcal{S}^{c_j}$ dado o documento d_i , como apresentado na equação seguinte:

$$P(c_j | d_i; \mathbf{F}(\mathcal{T})) = \sum_{s_k \in \mathcal{S}} P(c_j | s_k; \mathbf{F}(\mathcal{T})) \frac{\prod_{t_k \in \mathcal{T}, w_{d_i, t_k} > 0} P(t_k | s_k; \mathbf{F}(\mathcal{T}))}{\sum_{s_m \in \mathcal{S}} P(s_m | \mathbf{F}(\mathcal{T})) \prod_{t_k \in \mathcal{T}, w_{d_i, t_k} > 0} P(t_k | s_m; \mathbf{F}(\mathcal{T}))}. \quad (2.10)$$

Transductive Support Vector Machines: o método de classificação *Transductive Support Vector Machines* (TSVM) (VAPNIK, 1998; JOACHIMS, 1999), considera documentos rotulados e não rotulados para a construção de um hiperplano de separação dos documentos, se diferenciando do *Support Vector Machines* (SVM) que só utiliza os documentos rotulados. Uma ilustração do SVM e do TSVM é apresentado na

Figura 3 – Diferença entre SVM e TSVM para o mesmo conjunto de documentos rotulados.

Fonte: (ROSSI, 2015)

Figura 3. Na ilustração é possível notar que os documentos não rotulados levam o hiperplano de separação para regiões não densas.

O hiperplano de separação é definido por $\mathbf{f}^T \cdot \mathbf{w} + b$, onde $\mathbf{f}$ representa os coeficientes lineares (pesos atribuídos aos termos) e b é um valor escalar chamado de *bias*. Formalmente, o TSVM visa encontrar um hiperplano de separação de margem máxima que minimiza a função:

$$\begin{aligned}
 & \min_{\mathbf{f}, b} \frac{1}{2}(\mathbf{f} \cdot \mathbf{f}) \\
 \text{Sujeito a: } & y_{d_i}(\mathbf{f}^T \mathbf{w}_{d_i} + b) \geq 1, \forall d_i \in \mathcal{D}^L \\
 & y_{d_j}(\mathbf{f}^T \mathbf{w}_{d_j} + b) \geq 1, \forall d_j \in \mathcal{D}^U \\
 & y_{d_j} \in \{-1, 1\}, \forall d_j \in \mathcal{D}^U
 \end{aligned} \tag{2.11}$$

Assim, para resolver esse problema é necessário atribuir rótulos aos documentos não rotulados, visando encontrar a margem máxima de separação que divide as diferentes classes. No processo para encontrar o hiperplano ideal, documentos podem ser rotulados erroneamente caso contribuam na maximização da margem. Assim, para lidar com dados não linearmente separáveis são utilizadas as variáveis *slack* (ξ). Com isso, o problema de otimização passa a ser dado por:

$$\begin{aligned}
 & \min_{\mathbf{f}, b} \frac{1}{2}(\mathbf{f} \cdot \mathbf{f}) + C \sum_{d_i \in \mathcal{D}^L} \xi_{d_i} + C' \sum_{d_j \in \mathcal{D}^U} \xi_{d_j} \\
 \text{Sujeito a: } & y_{d_i}(\mathbf{f}^T \mathbf{w}_{d_i} + b) \geq 1 - \xi_{d_i}, \forall d_i \in \mathcal{D}^L \\
 & y_{d_j}(\mathbf{f}^T \mathbf{w}_{d_j} + b) \geq 1 - \xi_{d_j}, \forall d_j \in \mathcal{D}^U \\
 & y_{d_j} \in \{-1, 1\}, \forall d_j \in \mathcal{D}^U \\
 & \xi_{d_i} \geq 0, \forall d_i \in \mathcal{D}^L \\
 & \xi_{d_j} \geq 0, \forall d_j \in \mathcal{D}^U
 \end{aligned} \tag{2.12}$$

Os parâmetros C e C' são responsáveis por balancear o tamanho da margem do hiperplano e o erro de classificação para os documentos rotulados e não rotulados

respectivamente. Além disso, como no SVM, no TSVM também pode ser utilizado *kernels* sem alterar o problema de otimização.

Métodos que buscam uma solução ótima global não são capazes de classificar muitos documentos não rotulados, classificando em tempo hábil uma quantia em torno de 100 documentos. Já métodos que visam uma solução ótima local, como busca local ou relaxamento do problema de otimização, permitem a classificação de uma quantia bem maior de documentos não rotulados (ROSSI, 2015).

Para obter uma boa performance de classificação, as classes devem estar bem separadas para que o hiperplano seja situado em uma região de baixa densidade. Caso contrário, uma baixa performance de classificação será obtida (CHAPELLE; SCHOLKOPF; ZIEN, 2006).

2.2.2 Aprendizado Transdutivo em Grafos

O aprendizado transdutivo baseado em grafos surgiu como uma alternativa ao aprendizado transdutivo baseado no modelo espaço-vetorial, uma vez que os algoritmos baseados no modelo espaço-vetorial possuem fortes premissas quanto a distribuição dos dados, e também deficiências como a alta dimensionalidade e independência das entidades que compõem as coleções textuais.

Visando classificar os documentos não rotulados por meio de informações dos objetos vizinhos, o método baseado em grafos foi capaz de sanar tais deficiências (ROSSI, 2015). As informações providas dos vizinhos podem ser as informações de classes ou os próprios atributos. Utilizar as informações de classe dos vizinhos geram performances de classificação tão boas quanto utilizar os atributos dos vizinhos (LIU *et al.*, 2009; ANGELOVA; WEIKUM, 2006; ZHOU *et al.*, 2003). Assim, pelo motivo acima apresentado e por apresentar menor tempo computacional, nesse projeto de mestrado o foco são algoritmos que utilizam as informações de classe dos documentos vizinhos (ROSSI, 2015).

Como apresentado na Subseção 2.1.3, grafos são utilizados também para representar uma coleção de documentos. Uma vez gerado o grafo a partir da coleção, a estratégia de classificação transdutiva por meio de regularização pode ser aplicada. O objetivo da classificação baseada em regularização é minimizar uma função de custo, abaixo apresentada, que tem duas premissas: (i) objetos vizinhos devem ter informações de classe semelhantes; e (ii) as informações de classe dos objetos rotulados atribuídas durante o processo de classificação devem ser semelhantes as informações de classe reais (ROSSI, 2015). As duas premissas acima citadas podem ser expressadas em um *framework* de regularização composto por dois termos, sendo o primeiro a função de regularização e o segundo termo a função de perda (ZHU, 2005). A seguinte função de custo é capaz de

expressar o *framework* referenciado acima (DELALLEAU; BENGIO; ROUX, 2005):

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{o_i, o_j \in \mathcal{O}} w_{o_i, o_j} \Omega(\mathbf{f}_{o_i}, \mathbf{f}_{o_j}) + \mu \sum_{o_i \in \mathcal{O}^L} \Omega'(\mathbf{f}_{o_i}, \mathbf{y}_{o_i}), \quad (2.13)$$

na qual $\mathbf{f}$ é o vetor que representa as informações de classe de seu respectivo objeto, sendo que cada objeto no grafo possui seu próprio vetor $\mathbf{f}$. O vetor $\mathbf{f}$ armazena as informações de classe que são propagadas no grafo, sendo que cada posição do vetor é responsável por armazenar a pertinência do objeto para a classe que aquela posição representa. Os objetos rotulados possuem um vetor adicional estruturalmente idêntico ao vetor $\mathbf{f}$, o vetor $\mathbf{y}$, que é utilizado para a representação de suas informações reais de classe (rótulos originais). $\Omega(\dots)$ e $\Omega'(\dots)$ são funções de distância, sendo a primeira responsável por medir a proximidade dos vetores $\mathbf{f}$ de cada par de objeto no grafo, já a segunda é responsável por medir a proximidade entre o vetor de informações de classe $\mathbf{f}$ e o vetor de informações reais de classe $\mathbf{y}$ do mesmo objeto, sendo este um objeto rotulado. Logo, o parâmetro que define a importância das informações reais de classe na propagação dos rótulos é dado por μ . Assim, para minimizar a função da Equação 2.13, os vetores de informações de classe $\mathbf{f}$ precisam ser próximos dos vetores de informações reais de classe $\mathbf{y}$ do mesmo objeto, e também, quanto maior o peso w_{o_i, o_j} , menos distantes devem ser os respectivos vetores $\mathbf{f}$. Dessa forma, localmente, objetos altamente relacionados tendem a pertencer a mesma classe, bem como globalmente, objetos pertencentes a uma mesma comunidade tendem a pertencer a mesma classe também (LIU; CHANG, 2009).

Para a classificação por meio de regularização, pode-se resolver a Equação 2.13 por meio de soluções fechadas, ou então utilizar soluções iterativas que propagam as informações de classe pelos objetos do grafo, sendo essa abordagem denominada “propagação de rótulos”, ou *Label Propagation*. Os rótulos são propagados aos objetos vizinhos de forma proporcional aos pesos de suas respectivas arestas, além disso, os objetos rotulados tendem a manter o rótulo original. Na Figura 4 é ilustrada a classificação transdutiva via propagação de rótulos.

As principais diferenças entre os algoritmos baseados em regularização estão nas definições do valor de μ e da função de similaridade (ROSSI, 2015). Abaixo são apresentados dois algoritmos baseados em regularização, sendo eles algoritmos de classificação para grafos homogêneos: *Gaussian Fields and Harmonic Functions* (ZHU; GHAHRAMANI; LAFFERTY, 2003), *Learning With Local and Global Consistency* (ZHOU *et al.*, 2003).

Gaussian Fields and Harmonic Functions: o algoritmo para classificação transdutiva em grafos proposto por Zhu, Ghahramani e Lafferty (2003), é denominado *Gaussian Fields and Harmonic Functions* (GFHF). Para a definição do vetor $\mathbf{f}$ de cada objeto do grafo, é calculado a média harmônica do vetor $\mathbf{f}$ de cada um dos vizinhos

Figura 4 – Ilustração da classificação transdutiva via propagação de rótulos.

Fonte: Adaptada de (ROSSI, 2015).

do objeto corrente, sendo ponderado pelo peso das respectivas arestas, ou seja

$$\mathbf{f}_{o_i} = \frac{\sum_{o_j \in \mathcal{O}} w_{o_i, o_j} \mathbf{f}_{o_j}}{\sum_{o_j \in \mathcal{O}} w_{o_i, o_j}}. \quad (2.14)$$

O GFHF, bem como o algoritmo apresentado anteriormente, pode ser visto como uma função de regularização minimizando a seguinte função:

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{o_i, o_j \in \mathcal{O}} w_{o_i, o_j} (\mathbf{f}_{o_i} - \mathbf{f}_{o_j})^2 + \lim_{\mu \rightarrow \infty} \mu \sum_{o_i \in \mathcal{O}^L} (\mathbf{f}_{o_i} - \mathbf{y}_{o_i})^2. \quad (2.15)$$

$\lim_{\mu \rightarrow \infty} \mu$ é utilizado para que os objetos rotulados mantenham suas informações de classe reais, pois no GFHF a função harmônica é aplicada somente a objetos não rotulados. No GFHF, os valores das informações de classe podem ser reais, o que implica na existência de uma solução única globalmente ótima para $\mathbf{F}$ (ZHU; GOLDBERG, 2009).

Para minimizar a Equação 2.15, pode-se utilizar o algoritmo *Label Propagation* (ZHU; LAFFERTY; ROSENFELD, 2005). No Algoritmo 1 é apresentado o pseudocódigo do *Label Propagation*. As entradas do algoritmo são: o conjunto $\mathcal{O} = \mathcal{O}^U \cup \mathcal{O}^L$, que possui os objetos rotulados e não rotulados do grafo; a matriz $\mathbf{Y}$, que possui os vetores $\mathbf{y}$ de cada objeto rotulado; a matriz dos pesos das arestas entre os objetos $\mathbf{W}$ e finalmente $\mathbf{D}$, uma matriz diagonal de tamanho $|\mathcal{O}| \times |\mathcal{O}|$. As células de $\mathbf{D}$

possuem o valor 0, exceto as células diagonais, que armazenam o grau do objeto, sendo o grau $d_{o_i, o_i} = \sum_{o_j \in \mathcal{O}} w_{o_i, o_j}$.

Algoritmo 1 – *Label Propagation*

Entrada: $\mathcal{O}, \mathbf{W}, \mathbf{D}, \mathbf{Y}$

Saída: $\mathbf{F}(\mathcal{O}^U)$

- 1: $\mathbf{P} \leftarrow \mathbf{D}^{-1} \cdot \mathbf{W}$
 - 2: **repita**
 - 3: $\mathbf{F} \leftarrow \mathbf{P} \cdot \mathbf{F}$
 - 4: $\mathbf{F}(\mathcal{O}^L) \leftarrow \mathbf{Y}(\mathcal{O}^L)$
 - 5: **até** convergência ou número fixo de iterações
 - 6: **retorna** $\mathbf{F}(\mathcal{O}^U)$
-

Na linha 1 do [Algoritmo 1](#), é calculada a matriz $\mathbf{P}$ que é dada por

$$\mathbf{P} = \mathbf{D}^{-1} \mathbf{W}, \quad (2.16)$$

ou seja, $\mathbf{P}$ possui tamanho $|\mathcal{O}| \times |\mathcal{O}|$, e cada célula p_{o_i, o_j} é preenchida com a probabilidade da existência da aresta entre os objetos o_i e o_j . A operação matricial $\mathbf{F} \leftarrow \mathbf{P} \cdot \mathbf{F}$, localizada na linha 3, corresponde à [Equação 2.14](#) aplicada a todos os objetos $o_i \in \mathcal{O}$ e é responsável por minimizar iterativamente a [Equação 2.15](#). Na linha 4, os objetos rotulados têm suas respectivas informações de classe, contidas na matriz $\mathbf{F}$, sobrescritas pelos seus valores reais de classe, assim preservando seu rótulo original.

Os passos 3 e 4 se repetem até a convergência ou até que um número fixo de iterações seja atingido. Ao final, a matriz $\mathbf{F}$ é retornada e cada objeto não rotulado é atribuído a classe referente a posição do vetor $\mathbf{f}$ que possuir maior valor em relação as outras posições do mesmo vetor.

Learning With Local and Global Consistency: no *Learning With Local and Global Consistency* (LLGC), proposto por [Zhou et al. \(2003\)](#), considera-se que os objetos rotulados podem ter sido rotulados incorretamente, o que causaria a deteriorização da performance de classificação. Por esse motivo é permitido a alteração das informações de classe dos objetos rotulados. Outra característica do LLGC é a diminuição da influência dos objetos de grau elevado sobre a definição da informação de classe de seus vizinhos, para que não exerçam excessiva influência na classificação.

Por meio da minimização da função abaixo apresentada, o LLGC pode ser visto

como uma função de regularização.

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{o_i, o_j \in \mathcal{O}} w_{o_i, o_j} \left\| \frac{\mathbf{f}_{o_i}}{\sqrt{\sum_{o_k \in \mathcal{O}} w_{o_i, o_k}}} - \frac{\mathbf{f}_{o_j}}{\sqrt{\sum_{o_k \in \mathcal{O}} w_{o_j, o_k}}} \right\|^2 + \mu \sum_{o_i \in \mathcal{O}^L} \|\mathbf{f}_{o_i} - \mathbf{y}_{o_i}\|^2. \quad (2.17)$$

No [Algoritmo 2](#) é apresentado o pseudocódigo do LLGC. Na linha 3 é realizado a propagação de rótulos, sendo a operação matricial

$$\mathbf{F} = \alpha \mathbf{S} \mathbf{F} + (1 - \alpha) \mathbf{Y}, \quad (2.18)$$

onde

$$\mathbf{S} = \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}, \quad (2.19)$$

executada iterativamente até que a matriz $\mathbf{F}$ venha convergir ou até que seja atingido um número pré-definido de iterações. Ao final, os objetos não rotulados recebem o rótulo cuja posição do vetor $\mathbf{f}$ possui maior valor.

Algoritmo 2 – *Learning With Local and Global Consistency*

Entrada: $\mathcal{O}, \mathbf{W}, \mathbf{D}, \mathbf{Y}, \alpha$

Saída: $\mathbf{F}(\mathcal{O}^U)$

- 1: $\mathbf{S} = \mathbf{D}^{-\frac{1}{2}} \cdot \mathbf{W} \cdot \mathbf{D}^{-\frac{1}{2}}$
 - 2: **repita**
 - 3: $\mathbf{F} \leftarrow \alpha \cdot \mathbf{S} \cdot \mathbf{F} + (1 - \alpha) \cdot \mathbf{Y}$
 - 4: **até** convergência ou número fixo de iterações
 - 5: **retorna** $\mathbf{F}(\mathcal{O}^U)$
-

As soluções baseadas em propagação de rótulos apresentadas anteriormente possuem complexidade de tempo $O(|\mathcal{E}| * num_it)$, na qual *num_it* é o número de iterações realizadas até que os rótulos fossem propagados.

Os métodos de aprendizado de máquina até aqui apresentados não restringem a quantidade de classes utilizadas no processo de aprendizado, o que pode dificultar a rotulação de conjuntos de treino em que se encontram diversas classes, sendo essas de interesse ou não. A seguir é apresentado o aprendizado baseado em uma única classe, onde o usuário deve classificar apenas documentos de seu interesse, para que o classificador possa diferenciar documentos positivos dos documentos negativos.

2.3 Aprendizado Baseado em Uma Única Classe

Na tarefa de classificação por meio de algoritmos de aprendizado de máquina é frequente o uso do aprendizado multiclasse ([SEBASTIANI, 2002](#); [WITTEN et al., 2016](#); [LI et](#)

[al., 2019](#)). Para a classificação automática de textos por meio do aprendizado multiclasse, é necessário que o usuário tenha documentos rotulados de todas as classes existentes em sua base de dados. Dessa forma o algoritmo pode ser treinado, e assim classificar como uma das classes conhecidas os novos documentos apresentados.

O aprendizado multiclasse apresenta limitações em aplicações práticas, sendo algumas apresentadas a seguir: (i) todas as classes existentes no conjunto de dados devem ter exemplos rotulados, (ii) deve haver uma grande quantidade de exemplos rotulados, (iii) classificação incorreta de exemplos pertencentes a classes que não estão presentes na coleção de treinamento e, (iv) em geral é importante que o número de exemplos por classe seja balanceado. Abaixo, são descritas as limitações acima mencionadas.

A não existência de exemplos rotulados de determinada classe presente no conjunto de dados resultará em classificações incorretas no aprendizado multiclasse. Isso ocorre porque esse classificador não é capaz de classificar exemplos pertencentes a classes nunca vistas. *Outliers* são um exemplo em que o classificador multiclasse erra, pois os *outliers* não fazem parte de nenhuma das classes conhecidas ([KHAN; MADDEN, 2010](#)).

Outra limitação é a necessidade de uma grande quantidade de exemplos rotulados de cada classe do conjunto de dados para compor o conjunto de treinamento. Na classificação de textos, rotular documentos para o conjunto de treinamento é uma tarefa que demanda muito tempo e esforço, além de ser necessário amplo conhecimento dos domínios de cada classe. Uma boa distribuição no conjunto de dados também é requerida, sendo os exemplos representativos à classe ao qual pertencem. Caso a quantidade de exemplos não seja grande o suficiente e não representem bem sua classe, o método de classificação não conseguirá ser acurado, o que fará com que ele aprenda os exemplos rotulados, mas não seja capaz de classificar exemplos nunca vistos ([MUNROE; MADDEN, 2005](#)).

Visando minimizar as limitações e problemas do método multiclasse, uma alternativa é a utilização do aprendizado baseado em uma única classe, também conhecido pelo termo em inglês *One Class Learning* (OCL) ([TAX, 2001](#); [CHANDOLA; BANERJEE; KUMAR, 2009](#); [TAN et al., 2019b](#)). O OCL recebe como entrada apenas exemplos rotulados da classe de interesse do usuário. Assim, com base nos exemplos da classe de interesse o classificador irá classificar os novos exemplos apresentados como positivos ou negativos.

Devido o usuário ter que rotular apenas exemplos da sua classe de interesse, o OCL torna o uso de aprendizado de máquina mais viável em aplicações reais. Um exemplo de ferramenta que se tornaria mais prática é o *Websensors* ([MARCACINI et al., 2017](#)), que realiza o sensoriamento de notícias com intuito de monitorar tópicos por exemplo. Para que um sensor faça bem seu trabalho, é preciso que ele consiga classificar os documentos coletados por um *crawler*. Como já foi mencionado, o aprendizado multiclasse necessita de exemplos rotulados de todas as classes existentes na base de dados. Ao aplicar o OCL

nesse contexto, esse problema deixaria de existir, assim sendo necessário apenas que o usuário informe quais são os documentos de seu interesse. Dessa forma, o treinamento de um sensor se torna mais viável em comparação ao método multiclasse de aprendizado.

Um algoritmo de aprendizado baseado em uma única classe induzirá um modelo de classificação $f(x)$, que por sua vez gerará um *score* para cada novo exemplo apresentado. Esse *score* representa o quão pertinente à classe positivo (de interesse) o novo exemplo é, podendo ser uma medida de distância ou uma probabilidade de o exemplo pertencer à classe positivo.

$$classe = \begin{cases} f(d_i) \geq \text{limiar} \rightarrow \text{positivo} \\ f(d_i) < \text{limiar} \rightarrow \text{negativo} \end{cases} \quad (2.20)$$

Como apresentado na [Equação 2.20](#), se o *score* gerado por $f(d_i)$, sendo d_i o i -ésimo documento, for maior ou igual a um limiar, o exemplo será rotulado como um exemplo positivo, caso contrário será rotulado como negativo. Os diferentes métodos de classificação baseado em uma única classe se diferem pela definição de suas funções de distância (ou probabilidade), pela otimização da referida função e pelo limiar. Na maioria dos classificadores o foco está na otimização do modelo que gerará o *score*, sendo a otimização do limiar realizada depois. Poucos métodos otimizam sua função com base em um limiar previamente definido ([TAX, 2001](#)).

Existem muitas técnicas de aprendizado baseado em uma única classe, sendo elas divididas de diferentes formas por diferentes autores. Por exemplo, [Tan et al. \(2019b\)](#) faz a seguinte divisão: (i) Técnicas Baseadas em Modelos, (ii) Técnicas Baseadas em Similaridade, e (iii) Técnicas Baseadas em Densidade. Uma divisão mais detalhada é feita por [Chandola, Banerjee e Kumar \(2009\)](#), sendo essa divisão apresentada a seguir:

Técnicas Baseadas em Classificação: as técnicas baseadas em classificação funcionam de forma semelhante aos modelos de classificação tradicionais, ou seja, na fase de treinamento um modelo é treinado com base em exemplos rotulados, e então, na fase de testes, novos exemplos são classificados pelo modelo como positivo ou negativo.

Técnicas Baseadas nos Vizinhos Mais Próximos: essas técnicas seguem a simples suposição de que: exemplos da classe positiva estão localizados em vizinhanças de alta densidade, enquanto que exemplos da classe negativa estão distantes de seus mais próximos vizinhos. Para utilizar esse tipo de técnica é necessário antes definir uma medida de distância ou similaridade, que será utilizada para medir a distância (ou similaridade) entre dois exemplos. As técnicas baseadas nos vizinhos mais próximos podem ser divididas em dois grupos: (i) técnicas que utilizam a distância de uma instância de dados para seus k vizinhos mais próximos como sua função $f(d_i)$,

e (ii) técnicas que utilizam a densidade relativa de cada exemplo como sua função $f(d_i)$.

Técnicas Baseadas em Agrupamento: existem três técnicas baseadas em agrupamento.

A primeira, agrupa os exemplos utilizando algum algoritmo de agrupamento, e logo, define como negativo aqueles exemplos que não pertencem a nenhum grupo. A segunda técnica, inicialmente aplica algum algoritmo de agrupamento e em seguida, para cada exemplo, utiliza sua distância para o mais próximo dos centroides como seu *score*. Já a terceira técnica considera que, os exemplos que fazem parte de grupos com densidade ou tamanho menor que um limiar, são da classe negativa.

Técnicas Estatísticas: as técnicas estatísticas são baseadas na seguinte suposição: os exemplos da classe positiva ocorrem em regiões de alta probabilidade de um modelo estocástico, enquanto os exemplos da classe negativa ocorrem em regiões de baixa probabilidade. As técnicas estatísticas ajustam o modelo estatístico a um dado conjunto de dados, para então aplicar um teste de inferência estatística visando determinar se um certo exemplo nunca visto pertence ao modelo, ou tem baixa probabilidade de pertencer, sendo então considerado um exemplo da classe negativa.

Técnicas de Teoria da Informação: técnicas de teoria da informação analisam o conteúdo informativo de um conjunto de dados por meio de medidas de teoria da informação como: complexidade de Kolomogorov, entropia e entropia relativa. Esse tipo de técnica segue a seguinte suposição: Os exemplos da classe negativa induzem irregularidades no conteúdo informacional do conjunto de dados. Considere $C(\mathcal{S})$ a complexidade de um certo conjunto de dados $\mathcal{S}$, e $\mathcal{I}$ um subconjunto de exemplos de $\mathcal{S}$. Assim, essas técnicas buscam o menor $\mathcal{I}$ que gere a máxima diferença $C(\mathcal{S}) - C(\mathcal{S} - \mathcal{I})$. Quando encontrado, todas os exemplos em $\mathcal{I}$ são considerados como sendo da classe negativa.

Técnicas Espectrais: as técnicas espectrais buscam uma aproximação dos dados utilizando uma combinação de atributos que consigam representar a maior parte da variabilidade contida nos dados. Essas técnicas são baseadas na seguinte suposição: Os dados podem ser contidos em um subespaço com menor quantidade de dimensões, no qual os exemplos da classe negativa sejam significantemente diferentes dos exemplos da classe positiva. Logo, as técnicas espectrais visam encontrar um subespaço que torna fácil a identificação dos exemplos da classe negativa.

Dois *baselines* do aprendizado supervisionado baseado em uma única classe, o *k-Nearest Neighbor Density-based* (*k*-NND) e o *k-Means*, foram utilizados na comparação com algoritmos PUL com intuito de verificar se o uso de documentos não rotulados por algoritmos PUL realmente contribuem na performance de classificação. O *k*-NND se encaixa

na descrição referente às técnicas estatísticas não paramétricas bem como às técnicas baseadas nos vizinhos mais próximos. O *k-Means* também se encaixa às técnicas estatísticas não paramétricas bem como às técnicas baseadas em agrupamento. Ambos algoritmos foram escolhidos por obterem melhores resultados que outras técnicas quando aplicados na classificação de textos por meio do aprendizado baseado em uma única classe (TAN *et al.*, 2019a).

No *k*-NND, $f(d_i)$ é dado pela Eq. 2.21:

$$f(d_i) = \sum_{d_j \in \mathcal{N}_{(d_i, k)}} \text{cosseno}(\mathbf{w}_{d_i}, \mathbf{w}_{d_j}) / |\mathcal{N}_{(d_i, k)}|, \quad (2.21)$$

na qual $\mathcal{N}_{(d_i, k)}$ é o conjunto dos k vizinhos mais próximos de d_i .

No *k-Means*, os documentos positivos são separados em k grupos, ou seja, $\mathcal{D}^+ = \mathcal{G}_1 \cup \mathcal{G}_2 \cup \dots \cup \mathcal{G}_k$ na qual $\mathcal{G}_i \subset \mathcal{D}^+$ é representado por um centróide $\mathbf{g}_i$. O centróide $\mathbf{g}_i$ é a média dos vetores dos documentos no grupo $\mathcal{G}_i$. Uma vez que os centróides são calculados, cada novo documento recebe seu *score* de acordo com sua similaridade ao centróide mais próximo, ou seja:

$$f(d_i) = \max_{\mathcal{G}_j \subset \mathcal{D}^+} \text{cosseno}(\mathbf{w}_{d_i}, \mathbf{g}_j). \quad (2.22)$$

Nas técnicas acima apresentadas, considera-se que a quantidade de documentos positivos é grande o suficiente de forma a representar bem a classe de interesse. Existem casos onde a quantidade de documentos positivos são poucos, no entanto, existem também documentos não rotulados. Abordagens semissupervisionadas são capazes de lidar com tais situações. A seguir é apresentado a abordagem semissupervisionada do aprendizado baseado em uma única classe.

2.4 Aprendizado Semissupervisionado Baseado em Uma Única Classe

Recentemente o aprendizado semissupervisionado baseado em uma única classe, também conhecido pelo termo em inglês *Positive and Unlabeled Learning* (PUL), tem recebido crescente atenção (LIU *et al.*, 2002; LI; LIU, 2003; LI *et al.*, 2009; MA; ZHANG, 2017; JASKIE; SPANIAS, 2019). Diferentemente do OCL, que faz uso apenas de exemplos positivos rotulados, sendo ainda necessário uma quantidade de exemplos que represente bem a classe de interesse, a abordagem PUL pode ser aplicada quando existem poucos exemplos positivos rotulados e exemplos não rotulados. Esse cenário viabiliza ainda mais a classificação automática de textos em relação ao OCL, uma vez que classificar vários documentos, mesmo que apenas da classe de interesse, ainda pode ser uma tarefa custosa. Já a coleta de documentos, para serem utilizados como documentos não rotulados, é fácil. Assim, por meio dos documentos positivos rotulados e dos não rotulados, os documentos

não rotulados podem ser classificados. Além disso, os documentos não rotulados podem influenciar positivamente o processo de classificação, conforme apresentado na [Figura 5](#), onde é exemplificado o aprendizado supervisionado e o semissupervisionado baseado em uma única classe em um cenário onde existem poucos exemplos rotulados e exemplos não rotulados.

Figura 5 – Comparativo entre o aprendizado baseado em uma única classe (OCL), que utiliza apenas exemplos positivos rotulados em seu treinamento, e o aprendizado semissupervisionado baseado em uma única classe (PUL), que utiliza tanto exemplos positivos rotulados como exemplos não rotulados. Essa figura representa um cenário com poucos exemplos rotulados.

Fonte: Elaborada pelo autor.

Formalmente e especificamente para o PUL quando aplicado à classificação de textos, o conjunto $\mathcal{D}^L$ é composto por documentos somente da classe positiva (de interesse), ou seja, $\mathcal{D}^L = \mathcal{D}^+$, onde $\mathcal{D}^+$ é o conjunto de documentos rotulados como positivos. Além disso, o objetivo do PUL, é obter a função $F : \mathcal{D}^+ \cup \mathcal{D}^U \rightarrow \mathcal{Y}^+ \cup \mathcal{Y}^-$. As abordagens mais comuns do PUL são divididas em dois estágios ([JASKIE; SPANIAS, 2019](#)). No primeiro estágio, o conjunto de documentos positivos $\mathcal{D}^+$ é aumentado, e além disso, um conjunto de documentos negativos $\mathcal{D}^-$ é criado, tudo isso por meio dos documentos não rotulados $\mathcal{D}^U$. No segundo estágio, usando os documentos positivos, negativos e os documentos não rotulados, um algoritmo transdutivo é utilizado para rotular os documentos não rotulados. Exemplos de abordagens divididas em dois estágios são: *Spy Expectation-Maximization*

(S-EM) (LIU *et al.*, 2002), RC-SVM (LI; LIU, 2003), *Mapping-Convergence* (MC) (YU; HAN; CHANG, 2004; YU; ZHAI; HAN, 2003), PUL *by Extracting Likely Positive and Negative Micro-Clusters* (PUL-LELC) (LI *et al.*, 2009), *Positive and Unlabeled Learning by Label Propagation* (PU-LP) (MA; ZHANG, 2017) e *Positive and Unlabeled Documents Enlarging PU Classifier* (PE-PUC) (YU; LI, 2007). A seguir, são apresentadas algumas técnicas PUL.

O *Spy Expectation Maximization* (S-EM) (LIU *et al.*, 2002), em seu primeiro estágio, classifica exemplos não rotulados como negativos em casos que se tenha alta confiança. Nesse estágio, uma porcentagem de documentos positivos, denominados de “espiões” ($\mathcal{D}^{Spy} \subset \mathcal{D}^+$), é unida ao conjunto dos documentos não rotulados $\mathcal{D}^U$, gerando o conjunto $\mathcal{D}^M = \mathcal{D}^{Spy} \cup \mathcal{D}^U$. Os conjuntos $\mathcal{D}^M$ e $\mathcal{D}^+$, são dados como entrada no *Initial Expectation Maximization* (I-EM), sendo esse uma variação do EM (DEMPSTER; LAIRD; RUBIN, 1977). Assim que o I-EM finaliza, as probabilidades dos documentos positivos inseridos como espiões são utilizadas para definir um limiar, que será utilizado para classificar como negativo os documentos com probabilidade de ser positivos menor que o limiar. No segundo estágio, o algoritmo EM é utilizado para classificar os exemplos não rotulados $\mathcal{D}^U$ a partir dos exemplos positivos $\mathcal{D}^+$ e negativos $\mathcal{D}^-$, esse último obtido por meio do primeiro estágio.

As técnicas RC-SVM (LI; LIU, 2003) e *Mapping-Convergence* (MC) (YU; HAN; CHANG, 2004; YU; ZHAI; HAN, 2003) se diferenciam em seu primeiro estágio durante a seleção dos documentos negativos $\mathcal{D}^-$. O RC-SVM, em seu primeiro estágio, gera protótipos baseados no algoritmo Rocchio (ROCCHIO, 1971) para os documentos positivos e os documentos não rotulados. Então, os documentos não rotulados que são mais similares aos protótipos dos documentos não rotulados do que dos documentos positivos são escolhidos como negativos. Já no MC, um conjunto de características positivas, ou seja, características frequentes nos documentos positivos e pouco frequentes nos documentos não rotulados, são extraídas. Então, os documentos não rotulados são filtrados de acordo com as características positivas, sendo os documentos restantes considerados como negativos. No segundo estágio ambas técnicas utilizam a abordagem *Self-Training* combinada ao algoritmo SVM supervisionado para classificar todos os documentos não rotulados restantes. Para isso, o SVM é treinado com $\mathcal{D}^+$ e $\mathcal{D}^-$ (extraído no primeiro estágio), sendo utilizado para classificar os documentos não rotulados $\mathcal{D}^U$. Os documentos rotulados pelo SVM como negativos são removidos de $\mathcal{D}^U$ e inseridos no conjunto $\mathcal{D}^-$. Assim, o SVM é treinado novamente como descrito acima, realizando tal processo iterativamente até que nenhum documento de $\mathcal{D}^U$ seja rotulado como negativo. Ainda, no RC-SVM, o último SVM treinado é utilizado para classificar os documentos positivos $\mathcal{D}^+$, e caso mais de 5% sejam classificados como negativos, é escolhido como classificador final o primeiro SVM treinado ao invés do último.

Na técnica PUL-LELC (*PU Learning by Extracting Likely Positive and Negative micro-Clusters*) (LI *et al.*, 2009), inicialmente são extraídos dos documentos não rotulados $\mathcal{D}^U$ os documentos negativos $\mathcal{D}^-$ mais fortes por meio da estratégia de *ensemble* utilizando o S-EM e o RC-SVM, ambos acima apresentados. Então, notou-se que em $\mathcal{D}^+$ os tópicos existentes são relacionados, enquanto que $\mathcal{D}^-$ é composto por muitos documentos de diversos tópicos, de forma que seria bom agrupar os documentos em $\mathcal{D}^-$. Logo, são realizados agrupamentos em $\mathcal{D}^-$, sendo os grupos e o conjunto $\mathcal{D}^+$ utilizados para a geração de protótipos para representação das classes positiva e negativa. Para classificar os documentos restantes em $\mathcal{D}^U$, os mesmos também são agrupados, e então, os documentos de cada grupo são classificados, recebendo o rótulo de positivo caso estejam mais próximos de um dos protótipos positivos, e negativo caso estejam mais próximos de um dos protótipos negativos. Esse rótulo que acabara de ser atribuído é denominado rótulo temporário, pois, os documentos de um mesmo grupo, receberão o mesmo rótulo, sendo ele o majoritário do grupo. Os documentos dos grupos positivamente rotulados passam a fazer parte de $\mathcal{D}^{+'}$ e os documentos dos grupos negativamente rotulados são atribuídos a $\mathcal{D}^{-'}$. Finalmente, é construído um modelo de classificação por meio do SVM, sendo utilizados os conjuntos $\mathcal{D}^+ \cup \mathcal{D}^{+'}$ e $\mathcal{D}^- \cup \mathcal{D}^{-'}$.

O *Positive and Unlabeled Learning by Label Propagation* (PU-LP) (MA; ZHANG, 2017) em seu primeiro estágio aumenta a quantidade de exemplos positivos e extrai exemplos negativos. Para isso, ele constrói um grafo k NN (apresentado na Seção 2.1.3), que é utilizado para calcular a matriz de similaridade $\mathbf{W}$ por meio do índice de Katz⁴. Utilizando a matriz $\mathbf{W}$, os exemplos não rotulados $\mathcal{D}^U$ são ordenados de acordo com sua similaridade média aos exemplos positivos $\mathcal{D}^+$. Os exemplos com maior média de similaridade são extraídos e atribuídos ao conjunto dos positivos confiáveis $\mathcal{D}^{+'}$, que por sua vez são unidos a $\mathcal{D}^+ = \mathcal{D}^+ \cup \mathcal{D}^{+'}$ e removidos de $\mathcal{D}^U = \mathcal{D}^U - \mathcal{D}^{+'}$. Após aumentar o conjunto de exemplos positivos, inicia-se o processo para extrair os exemplos negativos confiáveis. Para isso, com base em $\mathbf{W}$ os exemplos não rotulados $\mathcal{D}^U$ são novamente ordenados em relação a sua similaridade aos exemplos positivos $\mathcal{D}^+$. Então, os exemplos cuja similaridade média são as menores são extraídos de $\mathcal{D}^U$ criando o conjunto de documentos negativos $\mathcal{D}^-$. Finalmente, no segundo estágio os rótulos são propagados de $\mathcal{D}^+$ e $\mathcal{D}^-$ para $\mathcal{D}^U$ por meio do algoritmo de propagação de rótulos GFHF (ZHU; GHAMRANI; LAFFERTY, 2003) (apresentado na Subseção 2.2.2).

O PE-PUC (YU; LI, 2007) utiliza uma abordagem baseada em grafo, mas somente na tarefa de aumentar o conjunto de documentos positivos. Para realizar essa tarefa, documentos negativos confiáveis, segundo os autores, são extraídos por meio do classificador *Naïve Bayes*, o qual é construído considerando os documentos não rotulados $\mathcal{D}^U$ como documentos negativos. Os documentos em $\mathcal{D}^U$ são então classificados, e aqueles que re-

⁴ O índice de Katz mede a similaridade entre os objetos o_i e o_j com base nos caminhos entre eles de todos os possíveis tamanhos, de forma que quanto maior o caminho, menor a similaridade

ceberam o rótulo negativo passam a fazer parte do conjunto $\mathcal{D}^-$. Então, o conjunto de documentos positivos confiáveis $\mathcal{D}^{+'}$ é extraído por meio de uma abordagem baseada em grafo, onde inicialmente, um grafo baseado em similaridade e totalmente conectado é gerado usando $\mathcal{D}^U - \mathcal{D}^-$, e alguns documentos positivos, que são selecionados aleatoriamente de $\mathcal{D}^+$ como objetos (vértices) rotulados. Então, uma abordagem de propagação de rótulos, similar ao *Learning with Local and Global Consistency* (ZHOU *et al.*, 2003) é aplicado para atribuir um *score* para cada objeto não rotulado. Uma vez que o algoritmo converge, os objetos não rotulados são ranqueados em ordem decrescente em relação aos *scores*. Os top λ/m objetos são atribuídos ao conjunto de positivos confiáveis $\mathcal{D}^{+'}$, o qual é adicionado ao conjunto de documentos positivos $\mathcal{D}^+ = \mathcal{D}^+ \cup \mathcal{D}^{+'}$, e removido dos documentos não rotulados $\mathcal{D}^U = \mathcal{D}^U - \mathcal{D}^{+'}$. Todo esse processo descrito acima é repetido m vezes. No fim, documentos negativos $\mathcal{D}^-$ são novamente extraídos através do classificador Naïve Bayes considerando o conjunto dos positivos $\mathcal{D}^+$ como a classe positivo e os documentos não rotulados $\mathcal{D}^U$ como a classe negativo. Então, os documentos não rotulados restantes são classificados por um Naïve Bayes construído com $\mathcal{D}^+$ e $\mathcal{D}^-$, ou uma abordagem transdutiva, por meio do Expectation-Maximization é aplicado considerando $\mathcal{D}^+$, $\mathcal{D}^-$ e $\mathcal{D}^U$ (NIGAM *et al.*, 2000b).

No decorrer da revisão bibliográfica, as seguintes lacunas se destacaram entre os métodos PUL: diversas abordagens fazem uso de *Self-Training* (aplicação iterativa do algoritmo *SVM*) ou *Expectation Maximization* para realizar a classificação transdutiva de textos. Além desses, também foi notado a utilização do algoritmo *Rocchio*. Esses algoritmos têm demonstrado performances de classificação inferior em relação a outras abordagens utilizadas para a classificação multiclasse de textos (ROSSI *et al.*, 2014; ROSSI; LOPES; REZENDE, 2016). No caso do aprendizado semissupervisionado, a utilização de documentos não rotulados por meio desses métodos, as vezes degrada a performance de classificação (ROSSI; LOPES; REZENDE, 2017; CHAPELLE; SCHOLKOPF; ZIEN, 2006).

Outra característica presente na maioria das abordagens do aprendizado semissupervisionado baseado em uma única classe, é o modelo de representação espaço-vetorial (apresentado na Subseção 2.1.2) (ZHANG; LEE, 2005; LIU *et al.*, 2002; LI; LIU, 2003; YU; ZHAI; HAN, 2003; YU; HAN; CHANG, 2004; LI *et al.*, 2009). Na classificação automática de textos, o modelo de representação baseado em redes tem obtido performance de classificação superior ao modelo espaço-vetorial (ROSSI *et al.*, 2014; ROSSI, 2015; ROSSI; REZENDE; LOPES, 2015; ROSSI; LOPES; REZENDE, 2016; ROSSI; LOPES; REZENDE, 2017). Dessa forma, é esperado que as abordagens baseadas em redes, também superem os métodos baseados no modelo espaço-vetorial. No entanto, somente alguns métodos utilizam redes como meio de representação. Ainda, alguns utilizam redes em apenas uma parte de sua abordagem, como a extração de documentos positivos (YU; LI, 2007).

Em algumas abordagens também foi notada a prática de aumentar a quantidade de documentos positivos (YU; LI, 2007; MA; ZHANG, 2017). No entanto tal estratégia pode inserir no conjunto de documentos positivos alguns documentos negativos, classificados como positivos incorretamente, o que causa a degradação da performance de classificação, principalmente por essa prática acontecer antes da extração de documentos negativos. E finalmente, não foi notado em nenhum trabalho a exploração de diferentes algoritmos de geração de redes e nem de algoritmos de propagação de rótulos. Visando explorar as lacunas mencionadas anteriormente, um novo método totalmente baseado em grafos é proposto e apresentado no próximo capítulo.

MÉTODO PROPOSTO: *LABEL PROPAGATION FOR POSITIVE AND UNLABELED LEARNING* (LP-PUL)

No decorrer deste capítulo é apresentado o método proposto neste projeto de mestrado. Inicialmente são apresentadas as características gerais do método proposto, e em seguida cada uma de suas três etapas é detalhada, sendo que cada etapa recebeu uma seção dedicada. Além disso, na última seção é apresentada a análise da complexidade de tempo.

3.1 Características Gerais

O método proposto neste projeto de mestrado, denominado *Label Propagation for Positive and Unlabeled Learning* (LP-PUL), é um método totalmente baseado em grafos, que faz uso do grafo de documentos e suas propriedades em todo seu processo de execução. Além disso, o LP-PUL permite a utilização de diferentes algoritmos de construção de grafos e de propagação de rótulos. Em nenhum momento são utilizados métodos como *Self-Training* e *Expectation Maximization*, que como já mencionado, podem degradar a performance de classificação de textos. Ainda, o método proposto não extrai documentos positivos dos documentos não rotulados, visando evitar a inserção de documentos classificados incorretamente no conjunto de documentos positivos.

O LP-PUL é dividido em três etapas: (i) construção do grafo, (ii) extração de documentos negativos e (iii) propagação de rótulos. As iterações entre as etapas são ilustradas na [Figura 6](#) e descritas a seguir. Inicialmente tem-se, como entrada, uma coleção de documentos, composta por alguns documentos rotulados positivamente e documentos não rotulados. A primeira etapa, corresponde à construção do grafo de documentos para

a representação da coleção textual. Esse grafo será utilizado pelas próximas duas etapas até a classificação de todos os documentos. Na segunda etapa (extração de documentos negativos) são rotulados alguns documentos como negativos. Para isso, o grafo de documentos é utilizado para encontrar os documentos mais distantes dos documentos positivos que são classificados como documentos negativos. Por fim, na terceira e última etapa, os rótulos dos documentos positivos e negativos são propagados por meio do grafo para os demais documentos não rotulados e os documentos positivos são retornados.

Figura 6 – Esquema geral do método proposto: *Label Propagation for Positive and Unlabeled Learning* (LP-PUL).

Fonte: Elaborada pelo autor.

O pseudocódigo do LP-PUL é apresentado no [Algoritmo 3](#). Como entrada, o algoritmo recebe a matriz de similaridade $\mathbf{W}$, o conjunto de documentos positivos $\mathcal{D}^+$, o conjunto de documentos não rotulados $\mathcal{D}^U$, o tipo do grafo que será gerado *tipo_grafo*, o algoritmo que será utilizado para propagar os rótulos *alg_lp* e o número de documentos negativos que será extraído *num_neg*. Como saída, são retornados os documentos rotulados com a classe positivo $\mathcal{D}^{+'}$. Na linha 1 é construído o grafo para representação da coleção textual. Em seguida, são extraídos os documentos negativos na linha 2. Finalmente, na linha 3 um dos algoritmos de propagação de rótulos é utilizado, e então após sua convergência são retornados todos os documentos divididos em dois grupos, os novos positivos $\mathcal{D}^{+'}$ e os novos negativos $\mathcal{D}^{-'}$. Nas próximas seções é detalhado o funcionamento de cada função apresentada no [Algoritmo 3](#).

Algoritmo 3 – LP-PUL

Entrada: $\mathbf{W}, \mathcal{D}^+, \mathcal{D}^U, \text{tipo_grafo}, \text{alg_lp}, \text{num_neg}$
Saída: $\mathcal{D}^{+'}$

- 1: $G \leftarrow \text{construtorGrafo}(\mathbf{W}, \text{tipo_grafo})$
- 2: $\mathcal{D}^- \leftarrow \text{extracaoNegativos}(G, \mathcal{D}^+, \mathcal{D}^U, \text{num_neg})$
- 3: $\mathcal{D}^{+'}, \mathcal{D}^{-'} \leftarrow \text{propagacaoRotulos}(G, \mathcal{D}^+, \mathcal{D}^-, \text{alg_lp})$

3.2 Construção do Grafo

A primeira etapa, construção do grafo, inicia-se pelo pré-processamento da coleção textual, que passa a ser representada por uma *bag-of-words* (BoW) (apresentada na [Subseção 2.1.2](#)). Em seguida um grafo de documentos é gerado. Os algoritmos considerados para gerar esse grafo nesta investigação são: (i) k NN, (ii) Mk NN e (iii) ϵ N. Esses algoritmos de construção de grafo, apresentados na [Subseção 2.1.3](#), produzem grafos esparsos e que conectam com maior peso os documentos mais similares. Além disso, esses grafos permitem que a propagação de rótulos seja mais rápida e possibilitam uma melhor acurácia de classificação ([ROSSI; LOPES; REZENDE, 2016](#); [SOUSA; REZENDE; BATISTA, 2013](#)).

Outra maneira de construir grafos baseados em similaridade é através de conectar um objeto o_i com todos os outros objetos e dar pesos as arestas por meio de uma função Gaussiana, ou seja, $w_{o_i, o_j} = \exp(-(1 - \text{sim}(o_i, o_j))/\sigma^2)$. Entretanto, essa estratégia de construção de grafos geralmente resulta em uma diminuição na qualidade da classificação e aumenta o tempo de processamento em comparação aos grafos k NN e ϵ N ([ROSSI; LOPES; REZENDE, 2016](#); [SOUSA; REZENDE; BATISTA, 2013](#)). Por isso, foi descartado o uso dessa abordagem nesse projeto.

3.3 Extração de Documentos Negativos

Como os grafos considerados no LP-PUL são grafos baseados na similaridade entre documentos, ou seja, estão conectados com maior peso os documentos que são mais similares, é suposto que documentos vizinhos no grafo tendem a fazer parte da mesma classe. Assim, aqueles que estão distantes tendem a pertencer a classes diferentes. Considerando essa suposição, a estrutura do grafo é utilizada para extrair os documentos negativos, sendo esses os mais distantes dos documentos positivos no grafo.

A utilização do grafo na extração de documentos negativos é uma escolha promissora, pois se sai melhor do que o uso de medidas de distância como a euclidiana, em cenários como o apresentado na [Figura 7](#). Nessa figura, considere a distância euclidiana para mensurar as distâncias no plano bidimensional, e considere no grafo a distância entre dois objetos como a soma das arestas do menor caminho entre os objetos. Considere

inicialmente a classe verde como a classe de documentos positivos, e que o objeto rotulado é o objeto 1. O objeto mais distante do 1 no plano bidimensional é o 2, já no grafo é o 4. Assim, a distância euclidiana iria extrair como negativo um objeto da classe positiva, já o grafo, extrairia como negativo um objeto da classe azul, ou seja, de uma classe distinta a classe positiva. O mesmo acontece quando a classe azul é considerada positiva, e o objeto 3 é considerado rotulado. A distância euclidiana erra, extraindo o objeto 4 como negativo, e o grafo acerta extraindo o objeto 2 como negativo.

Figura 7 – Conjunto de objetos apresentados por meio do plano bidimensional e por meio do grafo 3NN construído com base na distância euclidiana.

Fonte: Elaborada pelo autor.

Para extrair os documentos negativos por meio do grafo, os documentos não rotulados são ranqueados de acordo com sua distância média para os documentos positivos. Para exemplificar o cálculo da distância média, considere que existem 5 documentos positivos, logo a distância média do documento $d_i \in \mathcal{D}^U$ é dada pela média das distâncias de d_i para todos os documentos positivos. É importante ressaltar que são utilizados na prática poucos documentos positivos, assim, a quantidade de documentos positivos não implica num aumento de complexidade de tempo significativo no cálculo das distâncias de um documento não rotulado para todos os documentos positivos. Para encontrar a menor distância entre dois documentos, o algoritmo de Dijkstra (DIJKSTRA, 1959) é aplicado. Esse algoritmo calcula a distância entre dois documentos somando os pesos das arestas do menor caminho entre eles. Assim, seguindo a suposição de que os documentos negativos são os mais distantes dos positivos no grafo, o peso da aresta entre dois documentos deve representar a distância entre eles, para que então documentos próximos tenham o peso da aresta menor do que documentos distantes. Sendo assim, como o peso das arestas dos grafos utilizados correspondem à similaridade cosseno entre os documentos, ou seja, $w_{o_i, o_j} = \text{cosseno}(o_i, o_j)$, deve ser realizada a inversão desse peso. Dessa forma, para o

algoritmo de Dijkstra, o peso de cada conexão passa a ser $w'_{o_i, o_j} = 1/\text{cosseno}(o_i, o_j)$. A complexidade de tempo do algoritmo de Dijkstra é $O(|\mathcal{D}| \log |\mathcal{D}| + |\mathcal{E}|)$.

No [Algoritmo 4](#) é apresentado o pseudocódigo da extração de documentos negativos. Como entrada o algoritmo recebe o grafo G , o conjunto de documentos positivos $\mathcal{D}^+$, o conjunto de documentos não rotulados $\mathcal{D}^U$, e o número de documentos negativos a serem extraídos num_neg . Na linha 1, o vetor $\mathbf{a}$, onde cada posição corresponde a um documento não rotulado sendo então seu tamanho $|\mathcal{D}^U|$, é inicializado com zeros por meio da função *inicializa_com_zero(size)* que recebe como parâmetro o tamanho do vetor a ser inicializado. Na linha 3, por meio do algoritmo de Dijkstra, o vetor $\mathbf{d}$ recebe a menor distância de cada documento não rotulado para um determinado documento positivo. Em seguida, na linha 5, o vetor $\mathbf{a}$ é utilizado para armazenar na posição correspondente a um documento não rotulado, a soma de suas distâncias para cada documento positivo. Então, na linha 9, cada posição do vetor $\mathbf{a}$ é dividida pela quantidade de documentos positivos, sendo assim obtida a distância média de cada documento não rotulado para todos os documentos positivos. Finalmente, na linha 11, a função *ordenar()* ranqueia as distâncias médias em $\mathbf{a}$ e retorna os num_neg documentos não rotulados mais distantes dos documentos positivos, que passam a compor o conjunto dos documentos negativos.

Algoritmo 4 – *Extração de documentos negativos*

Entrada: $G, \mathcal{D}^+, \mathcal{D}^U, \text{num_neg}$

Saída: os num_neg documentos mais distantes em $\mathbf{a}$

- 1: $\mathbf{a} \leftarrow \text{inicializa_com_zero}(|\mathcal{D}^U|)$ /* cada posição do vetor $\mathbf{a}$ corresponde a um objeto u , e é inicializado com o valor 0 */
 - 2: **para todo** $p \in \mathcal{D}^+$ **faça**
 - 3: $\mathbf{d} \leftarrow \text{Dijkstra}(G, p)$ /* o vetor $\mathbf{d}$ recebe a distância de p para cada objeto não rotulado */
 - 4: **para todo** $u \in \mathcal{D}^U$ **faça**
 - 5: $\mathbf{a}_u \leftarrow \mathbf{a}_u + \mathbf{d}_u$ /* é salvo em cada célula de $\mathbf{a}$ a soma das distâncias de u para cada p */
 - 6: **fim para**
 - 7: **fim para**
 - 8: **para todo** $u \in \mathcal{D}^U$ **faça**
 - 9: $\mathbf{a}_u \leftarrow \mathbf{a}_u / |\mathcal{D}^+|$ /* cada posição do vetor $\mathbf{a}$ passa a conter a média das distâncias de u para todos os objetos positivos p */
 - 10: **fim para**
 - 11: $\text{neg_mais_distantes} \leftarrow \text{ordenar}(\mathbf{a}, \text{num_neg})$
-

3.4 Propagação de Rótulos

Neste momento, o grafo está composto por três conjuntos de objetos: objetos da classe positiva ($\mathcal{D}^+$), objetos da classe negativa ($\mathcal{D}^-$) e objetos não rotulados ($\mathcal{D}^U$). Com isso, a estratégia de propagação de rótulos pode ser aplicada para que os rótulos “positivo”

e “negativo” sejam propagados para os demais documentos não rotulados no grafo. Como os grafos utilizados nesse trabalho são todos homogêneos, os algoritmos de propagação de rótulos considerados para essa etapa são dois dos mais comuns encontrados na literatura para esse tipo de grafo: (i) GFHF e (ii) LLGC, ambos apresentados na [Subseção 2.2.2](#). Após os rótulos serem propagados pelo grafo, os documentos com rótulo positivo são retornados ao usuário como documentos de seu interesse, ou podem ser utilizados, juntamente aos documentos que foram classificados como negativos, para induzir um modelo de classificação.

3.5 Análise de Complexidade de Tempo

A complexidade de tempo do LP-PUL é composta pela complexidade de tempo de cada uma das suas etapas, ou seja:

- i. Construção do grafo: $O(|\mathcal{D}|^2 \log |\mathcal{D}|)$;
- ii. Extração de documentos negativos:
 - a. O algoritmo de Dijkstra é executado uma vez para cada documento positivo, sendo esse documento positivo o ponto de partida: $O(|\mathcal{D}^+| * (|\mathcal{D}| \log |\mathcal{D}| + |\mathcal{E}|))$.
 - b. A cada execução do algoritmo de Dijkstra, o vetor de saída é somado ao vetor **a**: $O(|\mathcal{D}^+| * |\mathcal{D}^U|)$;
 - c. Todas as posições do vetor **a** são visitadas para serem divididas pela quantidade de positivos para então encontrar a média: $O(|\mathcal{D}^U|)$;
 - d. O vetor **a** é ordenado: $O(|\mathcal{D}^U| \log |\mathcal{D}^U|)$;
- iii. Propagação dos rótulos: $O(|\mathcal{E}| * num_it)$.

A etapa (ii) soma uma complexidade de tempo igual a $O((|\mathcal{D}^+| * (|\mathcal{D}| \log |\mathcal{D}| + |\mathcal{E}|)) + (|\mathcal{D}^+| * |\mathcal{D}^U|) + |\mathcal{D}^U| + (|\mathcal{D}^U| \log |\mathcal{D}^U|))$, que resulta em aproximadamente $O(|\mathcal{D}^+| * (|\mathcal{D}| \log |\mathcal{D}| + |\mathcal{E}|))$. Assim, unindo as três etapas a complexidade total é dada por: $O(|\mathcal{D}|^2 \log |\mathcal{D}|) + O(|\mathcal{D}^+| * (|\mathcal{D}| \log |\mathcal{D}| + |\mathcal{E}|)) + O(|\mathcal{E}| * num_it) \sim O(|\mathcal{D}|^2 \log |\mathcal{D}|)$.

AVALIAÇÃO EXPERIMENTAL

Nesta seção, o método proposto (LP-PUL) é comparado com os algoritmos PUL, *Positive and Unlabeled Documents Enlarging PU Classifier* (PE-PUC), *Positive and Unlabeled Learning by Label Propagation* (PU-LP) e RC-SVM (Seção 2.4), e com os algoritmos OCL, *k-Nearest Neighbor Density-based* (k -NND) e *k-Means* (Seção 2.3), permitindo assim demonstrar o quão documentos não rotulados podem influenciar na performance de classificação. Nas próximas seções são apresentadas as coleções textuais utilizadas, as configurações experimentais, os critérios de avaliação, os resultados e as discussões.

4.1 Coleções Textuais

Os experimentos foram conduzidos utilizando 17 coleções textuais de diferentes domínios, como páginas web (PW), documentos médicos (DM), notícias (NO), documentos científicos (DC) e documentos da TREC (DT). Na Tabela 1 são apresentadas as características de cada coleção textual. Entre as coleções utilizadas, a que possui menos documentos é a TR23 com 204 documentos, e a que possui mais documentos é a Fbis com 2463 documentos. As coleções foram coletadas de Rossi, Marcacini e Rezende (2013), onde são disponibilizadas por meio da representação BoW, com esquema de pesos *tf-idf* e com *stop-words* removidas.

É importante mencionar que devido a baixa disponibilidade de coleções textuais para PUL, foram simulados cenários para PUL utilizando as coleções multiclasse já mencionadas. Para simular um cenário PUL, iterativamente uma das classes é considerada como classe positiva, e as restantes como classe negativa.

Tabela 1 – Características das coleções de documentos, sendo elas: número de documentos ($|\mathcal{D}|$), número de termos gerados ($|\mathcal{T}|$), número médio de termos por documentos ($|\overline{\mathcal{T}}|$), número de classes ($|\mathcal{C}|$), desvio padrão considerando a porcentagem de distribuição das classes ($\sigma(\mathcal{C})$) e porcentagem de documentos da classe majoritária ($\max(\mathcal{C})$).

Text Collection	$ \mathcal{D} $	$ \mathcal{T} $	$ \overline{\mathcal{T}} $	$ \mathcal{C} $	$\sigma(\mathcal{C})$	$\max(\mathcal{C})$
CSTR (DC)	299	1726	54,27	4	18,89	42,81
Fbis (NO)	2463	2001	159,24	17	5,66	26,54
Oh0 (DM)	1003	3183	52,50	10	5,33	19,34
Oh10 (DM)	1050	3239	55,64	10	4,25	15,71
Oh15 (DM)	913	3101	59,30	10	4,27	17,20
Oh5 (DM)	918	3013	54,43	10	3,72	16,23
Re0 (NO)	1504	2887	51,73	13	11,56	40,43
Re1 (NO)	1657	3759	52,70	25	5,54	22,39
SyskillWebbert (PW)	334	4340	93,16	4	10,75	41,02
TR11 (DT)	414	6430	281,66	9	9,80	31,88
TR12 (DT)	313	5805	273,60	8	7,98	29,71
TR21 (DT)	336	7903	469,86	6	25,88	68,75
TR23 (DT)	204	5833	385,29	6	15,58	44,61
TR31 (DT)	927	10129	268,50	7	13,37	37,97
TR41 (DT)	878	7455	195,33	10	9,13	27,68
TR45 (DT)	690	8262	280,58	10	6,69	23,19
Wap (PW)	1560	8461	141,33	20	5,20	21,86

Fonte: (ROSSI; MARCACINI; REZENDE, 2013; ROSSI, 2015).

4.2 Configuração Experimental e Critérios de Avaliação

Nesta seção são apresentados os algoritmos utilizados, seus respectivos parâmetros e o esquema de avaliação utilizado na avaliação experimental. Foram utilizados cinco métodos¹, previamente apresentados nas Subseções 2.3 e 2.4, para serem comparados com o método proposto LP-PUL, a saber: (i) k -NND, (ii) k -Means, (iii) PE-PUC, (iv) PU-LP, e (v) RC-SVM. As duas primeiras técnicas são técnicas OCL e as últimas três são PUL. Para ambas as técnicas OCL foram considerados os seguintes limiares, utilizados para definir se um novo documento é classificado como positivo ou negativo:

- Manualmente definido: sendo $limiar = 0.05 \times z$, na qual $z \in \mathbb{N} : 1 \leq z \leq 19$;
- Automaticamente definido: limiares definidos por meio da técnica 6σ (MUIR, 2006). Para isso, geram-se os *scores* $f(\mathbf{d}_i)$ para todos os documentos positivos, calcula-se a média (μ) e o desvio padrão (σ), e define-se os limiares como $limiar \in \{\mu - 3\sigma, \mu - 2\sigma, \mu - 1\sigma, \mu + 1\sigma, \mu + 2\sigma, \mu + 3\sigma\}$. Foram considerados adicionalmente também, a média e o valor mínimo de $f(\mathbf{d}_i)$, ou seja $limiar \in \{\mu - 3\sigma, \mu - 2\sigma, \mu - 1\sigma, \mu + 1\sigma, \mu + 2\sigma, \mu + 3\sigma, \mu, \min(f(\mathbf{d}_i))\}$

Os parâmetros utilizados e apresentados a seguir, foram coletados de seus respectivos artigos, e também foram baseados em testes empíricos. Além dos parâmetros, são

¹ Os algoritmos foram implementados em Java OpenJDK versão 1.8.0_181 e Python 3.7.3.

informados a quantidade de possíveis combinações entre eles, resultando na quantidade total de configurações de cada técnica.

- i. **k-NND**: $k \in \{1, 3, 5, 7, 9, 11, 13, 15, 17, 19\}$, a medida de similaridade cosseno e os limiares apresentados nos parágrafos anteriores. Total de configurações: 270.
- ii. **k-Means**: $k \in \{1, 3, 5, 7, 9, 11, 13, 15, 17, 19\}$, 100 como número máximo de iterações, a medida de similaridade cosseno e os limiares apresentados nos parágrafos anteriores. Além disso, foi executado 10 vezes para então ser escolhido o agrupamento com maior coesão. Total de configurações: 270.
- iii. **PE-PUC**: $\alpha = 0.99$, $\lambda \in \{0.6, 0.7, 0.8\}$, e $m \in \{1, 2, 3\}$ (YU; LI, 2007). Como peso para os documentos não rotulados no *Expectation Maximization* (EM), foi considerado $\beta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. O Naive Bayes também foi utilizado como opção ao EM, porém não possui parâmetros. Total de configurações: 54.
- iv. **PU-LP**: $\alpha \in \{0.02, 0.04, 0.06, 0.08, 0.1\}$, $\lambda \in \{0.2, 0.6, 0.8, 1.0, 1.2, 1.4, 2.0\}$, e $m \in \{1, 2, 3\}$ (MA; ZHANG, 2017). Para o grafo k NN foi considerado $k = 3 + 4 * p$, na qual $p \in \mathbb{N} : 0 \leq p \leq 7$. Total de configurações: 840.
- v. **RC-SVM**: para o algoritmo Rocchio, foi utilizado $\alpha \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$, e $\beta = 1 - \alpha$. Para o algoritmo SVM foi considerado $C \in \{0.1, 1.0, 10, 100, 1000\}$ e um núcleo linear uma vez que esse tipo de núcleo apresenta os melhores resultados para classificação de textos (ROSSI *et al.*, 2014; CARUANA; NICULESCU-MIZIL, 2006). Total de configurações: 25.
- vi. **LP-PUL**: o parâmetro k para gerar os grafos k NN e Mk NN foi utilizado com os valores $k = 3 + 4 * p$, e o parâmetro ϵ para o grafo ϵ N foi utilizado com os valores $\epsilon = 0.1 + p * 0.05$, na qual $p \in \mathbb{N} : 0 \leq p \leq 7$. O número de documentos negativos a serem extraídos foi $num_neg \in \{1 \times |\mathcal{D}^+|, 2 \times |\mathcal{D}^+|, 5 \times |\mathcal{D}^+|, 10 \times |\mathcal{D}^+|\}$, no qual $|\mathcal{D}^+|$ é o número de documentos positivos. Finalmente, os valores para o parâmetro α do algoritmo LLGC foram $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. O método de propagação de rótulos GFHF também foi utilizado, porém este é um método livre de parâmetros. Total de configurações: 576.

Para simular o cenário PUL utilizando as coleções textuais multiclasse, foi utilizado um procedimento iterativo que durante cada iteração seleciona uma classe para representar a classe positiva, sendo as demais classes utilizadas para representar a classe negativa. Assim, uma coleção de documentos com 5 classes por exemplo, possui 5 representações para o cenário PUL, onde cada cenário PUL tem uma das classes da coleção multiclasse como a classe positiva. Após este procedimento, para cada um dos cenários PUL, alguns documentos da classe positiva foram selecionados aleatoriamente para serem

os documentos rotulados, sendo os demais documentos, positivos e negativos, utilizados como documentos não rotulados. Como os documentos rotulados são escolhidos aleatoriamente, eles podem causar variação na performance de classificação e, para lidar com isso, foram extraídos 10 conjuntos de documentos rotulados para cada cenário PUL. Assim, o resultado para cada coleção multiclasse é a média dos resultados de cada um dos 10 conjuntos de documentos rotulados de cada classe. Nos experimentos, foram considerados os seguintes números de documentos positivos rotulados (interesse do usuário) $|\mathcal{D}^+| = \{1, 5, 10, 20, 30\}$. Para uma comparação justa de todos os métodos, os mesmos conjuntos de documentos rotulados foram fornecidos como entrada.

Para analisar a performance de classificação dos métodos foi utilizada a medida F_1 (MA; ZHANG, 2017; LI *et al.*, 2009; YU; ZHAI; HAN, 2003; LIU *et al.*, 2002). F_1 é a média harmônica da medida de Precisão (*Precision*) e Revocação (*Recall*), sendo todas respectivamente dadas por:

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}, \quad (4.1)$$

$$Precision = \frac{TP}{TP + FP}, \quad (4.2)$$

$$Recall = \frac{TP}{TP + FN}, \quad (4.3)$$

na qual TP (*True Positives*) é o número de documentos positivos corretamente classificados como positivo, FN (*False Negatives*) é o número de documentos positivos classificados como negativo, e FP (*False Positives*) é o número de documentos negativos classificados como positivos. É importante mencionar que para a obtenção da quantidade de TP , é considerado apenas o conjunto de teste, ou seja, os documentos positivos rotulados dados para o algoritmo como entrada não são contabilizados.

Os resultados apresentados na próxima seção correspondem às configurações que obtiveram os melhores resultados, ou seja, os resultados são os melhores, podendo cada resultado ter uma configuração específica. Em <https://juliocarnevali.github.io/lp-pul/> estão os resultados obtidos de todos os algoritmos considerando todas as combinações de seus parâmetros. Além disso, nesse mesmo endereço estão disponíveis as implementações, tanto do método proposto LP-PUL, como dos demais métodos utilizados na comparação. O método proposto, na próxima seção, é comparado com algoritmos PUL e OCL. Também é analisada a performance de classificação do LP-PUL quando empregado diferentes algoritmos de construção de grafo e de propagação de rótulos.

4.3 Resultados

Figura 8 – Comparação da performance de classificação do método proposto, LP-PUL, com dois algoritmos OCL, k -Means e k -NND, e três algoritmos PUL, PE-PUC, PU-LP e RC-SVM. O eixo x representa o número de documentos positivos rotulados e o eixo y representa os valores F_1 .

Na [Figura 8](#) são apresentados os valores de F_1 obtidos para cada quantidade de documentos positivos rotulados e para cada algoritmo. Cada gráfico possui os resultados de uma coleção de documentos, onde o eixo x representa a quantidade de documentos positivos rotulados e o eixo y o valor de F_1 . Pode-se observar que, quase sempre, o LP-PUL supera os outros métodos independentemente da quantidade de documentos positivos rotulados. Pode-se observar também que o PE-PUC, e RC-SVM, ambos algoritmos PUL, geralmente obtém menor performance de classificação que os outros algoritmos PUL e até mesmo os algoritmos OCL, que não utilizam documentos não rotulados. Portanto, além do método proposto obter maior valor de F_1 , ele efetivamente faz uso dos documentos não rotulados para melhorar a performance de classificação.

Para reforçar a afirmação de que o método proposto é melhor que os demais utilizados na comparação, os resultados da [Figura 8](#) foram submetidos ao teste estatístico de Friedman e pós-teste de Li com 95% de nível de confiança para avaliar se há diferença estatisticamente significativa (DES) entre os algoritmos ([TRAWIŃSKI et al., 2012](#)). O pós-teste de Li é utilizado em comparações $1 \times N$, ou seja, em casos onde um algoritmo de controle é comparado com os demais algoritmos. Nesse caso, o pós-teste de Li foi aplicado considerando o LP-PUL como algoritmo de controle, assim, o LP-PUL é comparado com cada um dos demais algoritmos. Cada par de algoritmos, sendo sempre um deles o LP-PUL, possui uma hipótese nula, que afirma não haver diferença estatisticamente significativa (DES) entre os referidos algoritmos. O pós-teste de Li então, gera um p -value para cada hipótese nula, sendo ela rejeitada quando o seu p -value é igual ou menor que o valor de p definido pelo pós-teste de Li. Uma vez que a hipótese nula é rejeitada, pode-se afirmar que o método proposto é melhor que o outro método com diferença estatisticamente significativa (DES).

Tabela 2 – Esta tabela apresenta o ranqueamento médio (RM), ranqueamento geral (RG), o p -value considerando os valores de F_1 para cada quantidade de documentos positivos rotulados (apresentados em negrito quando o LP-PUL supera o referido método com DES), e o valor de p que produz DES.

Alg.	1 doc. positivo			5 docs. positivos			10 docs. positivos			20 docs. positivos			30 docs. positivos		
	RM	RG	p -value	RM	RG	p -value	RM	RG	p -value	RM	RG	p -value	RM	RG	p -value
LP-PUL	1.53	1º	-	1.5	1º	-	1.65	1º	-	1.29	1º	-	1.18	1º	-
RC-SVM	5.59	6º	0	5.35	5º	0	5.18	5º	0	5.24	6º	0	4.59	5º	0
PU-LP	3.59	4º	0.001335	3.41	4º	0.002889	3.35	4º	0.007851	3.18	3º	0.003352	3.59	4º	0.000171
PE-PUC	5.41	5º	0	5.65	6º	0	5.59	6º	0	5.00	5º	0	4.88	6º	0
k -NND	2.44	2º	0.155351	2.09	2º	0.359301	2.41	2º	0.233376	2.82	2º	0.017153	3.24	2º	0.001335
k -Means	2.44	3º	0.155351	3.00	3º	0.019409	2.82	3º	0.066744	3.47	4º	0.000694	3.53	3º	0.000246
	DES $p \leq 0.044455$			DES $p \leq 0.033721$			DES $p \leq 0.040349$			DES $p \leq 0.05$			DES $p \leq 0.05$		

A [Tabela 2](#) apresenta os resultados do teste estatístico de Friedman e pós-teste de Li. Essa tabela contém o ranqueamento médio (RM) calculado pelo teste de Friedman, ranqueamento geral (RG), o *p-value* de cada hipótese nula vinculada a um par de algoritmos, sendo que o *p-value* é apresentado em negrito quando o LP-PUL supera o referido método com DES, e o valor de *p* que produz DES.

De acordo com a [Tabela 2](#), pode-se observar que o método proposto é o melhor com DES quando o número de documentos positivos rotulados é maior ou igual a 20. No cenário com 1 e 10 documentos positivos rotulados, o LP-PUL é o primeiro no ranqueamento geral, mas não tem DES em relação aos métodos OCL: *k*-NND e *k*-Means. Entretanto, há DES em comparação aos algoritmos PUL: RC-SVM, PU-LP, e PE-PUC. Considerando o cenário com 5 documentos positivos rotulados, o LP-PUL alcança também o primeiro no ranqueamento geral, e só não apresenta DES em relação ao algoritmo *k*-NND. Também foi observado que, além do LP-PUL, o PU-LP foi capaz de obter ranqueamento médio melhor que os demais algoritmos PUL em todos os cenários. Esses resultados também confirmam que os algoritmos baseados em grafos, PU-LP e LP-PUL, provêm melhor classificação que algoritmos PUL baseados no modelo espaço-vetorial.

Na [Figura 9](#) são apresentadas as performances de classificação do LP-PUL quando utilizados em sua fase final os algoritmos GFHF e LLGC, com a finalidade de analisar e comparar o impacto desses algoritmos de propagação de rótulos no método proposto. É importante esclarecer que os resultados apresentados referentes ao uso do GFHF e do LLGC são as combinações dos melhores resultados dos três algoritmos de construção de grafo *k*NN, *Mk*NN, and ϵ N. Pode-se notar que os resultados obtidos com o LLGC são sempre melhores que os resultados obtidos com o GFHF quando utilizado 1 documento positivo rotulado e em aproximadamente metade das coleções são melhores quando utilizados 5 documentos positivos rotulados. Já o GFHF apresentou resultados melhores que o LLGC na grande maioria das coleções textuais quando há 10 ou mais documentos positivos rotulados.

Para confirmar estatisticamente as afirmações feitas sobre o desempenho dos algoritmos de propagação de rótulos LLGC e GFHF no LP-PUL, os respectivos resultados foram submetidos ao teste dos postos sinalizados de Wilcoxon ([TRAWIŃSKI et al., 2012](#)) com 95% de nível de confiança. O teste de Wilcoxon é utilizado para comparações não paramétricas entre dois algoritmos. Os resultados demonstraram que no método proposto, o uso do LLGC é superior ao GFHF com DES no cenário em que é utilizado apenas 1 exemplo positivo rotulado. Para 5 e 10 exemplos positivos rotulados, não existe DES entre os algoritmos de propagação de rótulos. Já quando são utilizados 20 ou mais documentos positivos rotulados, a utilização do GFHF é superior ao LLGC com DES, fator esse interessante em situações práticas pois o GFHF não possui parâmetros.

Figura 9 – Gráficos contendo as melhores performances de classificação referentes ao método proposto quando executado com o LLGC e quando executado com o GFHF na etapa final. O eixo x representa o número de documentos positivos rotulados e o eixo y representa os valores F_1 .

Também foi analisado o impacto dos algoritmos de construção de grafo no método proposto. A [Figura 10](#) apresenta os resultados para os diferentes tipos de grafo considerando o algoritmo LLGC na etapa de propagação de rótulos do LP-PUL. Já na [Figura 11](#) os resultados para cada grafo leva em consideração o uso do GFHF na etapa de propagação de rótulos. Os gráficos apresentados nas Figuras 10 e 11 mostram que há pouca diferença entre os resultados referentes à aplicação de cada algoritmo construtor de grafo no LP-PUL, tanto para o uso do LLGC quanto para o GFHF na etapa de propagação de rótulos. Então, para evidenciar possíveis diferenças entre os grafos, os resultados foram submetidos ao teste estatístico de Friedman e ao pós-teste de Nemenyi, frequentemente utilizado para comparar resultados de múltiplos algoritmos em múltiplos conjuntos de dados ([TRAWIŃSKI et al., 2012](#)).

As Figuras 12 e 13 apresentam os resultados do teste estatístico para os métodos de construção de grafos aplicados ao LP-PUL, considerando respectivamente os algoritmos de propagação de rótulos, LLGC e GFHF, na etapa final do LP-PUL. Os resultados são apresentados através dos diagramas de diferença crítica, nos quais os métodos são ordenados de acordo com seu ranqueamento médio, e são conectados por uma linha aqueles métodos que não apresentam DES entre si ([DEMSAR, 2006](#)). Considerando que na etapa de propagação de rótulos do LP-PUL foi utilizado o LLGC, o grafo k NN obteve os melhores ranqueamentos médios em todos os cenários. Além disso, o grafo k NN obteve DES em relação ao grafo Mk NN em todos os cenários, e somente com 30 documentos positivos rotulados o grafo k NN obteve DES em relação aos grafos Mk NN e ϵ N. Agora, considerando o uso do GFHF, o grafo k NN sempre obteve o melhor ranqueamento médio e em todos os cenários obteve DES em relação ao grafo ϵ N. Para 1 e 5 documentos positivos rotulados, os grafos k NN e Mk NN foram melhores que o grafo ϵ N com DES, e para 30, o grafo k NN foi melhor que os grafos Mk NN e ϵ N com DES. Assim, entre os métodos usados, o k NN demonstrou ser a melhor escolha para a etapa de construção de grafos.

Figura 10 – Melhores performances de classificação referente ao uso de cada algoritmo de construção de grafo, k NN, Mk NN and ϵ N, no método proposto, considerando o uso do LLGC na etapa de propagação de rótulos. O eixo x representa o número de documentos positivos rotulados e o eixo y representa os valores F_1 .

Figura 11 – Melhores performances de classificação referente ao uso de cada algoritmo de construção de grafo, k NN, Mk NN and ϵ N, no método proposto, considerando o uso do GFHF na etapa de propagação de rótulos. O eixo x representa o número de documentos positivos rotulados e o eixo y representa os valores F_1 .

Figura 12 – Diagramas de diferença crítica dos três algoritmos de construção de grafo quando aplicados ao LP-PUL. Os resultados consideram o uso do LLGC na etapa de propagação de rótulos.

Figura 13 – Diagramas de diferença estatística dos três algoritmos de construção de grafo quando aplicados ao LP-PUL. Os resultados consideram o uso do GFHF na etapa de propagação de rótulos.

CONCLUSÕES E TRABALHOS FUTUROS

A quantidade de dados no formato textual disponibilizada por meio da Internet é gigantesca, o que torna a identificação e extração de informações de forma manual praticamente impossível. Para isso, ferramentas de aprendizado de máquina como a classificação automática de textos são empregadas. Uma das técnicas mais utilizadas é a classificação por meio do aprendizado multiclasse, a qual, por necessitar de exemplos rotulados para todas as classes da coleção de treinamento, torna-se difícil sua aplicação na prática. Além disso, o aprendizado multiclasse não é capaz de classificar exemplos de uma classe que não está presente na coleção de treinamento.

Uma alternativa ao aprendizado multiclasse é o aprendizado baseado em uma única classe (OCL). Nesse modelo, o aprendizado é realizado utilizando apenas uma classe, geralmente a classe de interesse do usuário. Após aprendido o modelo, esse é capaz de classificar os exemplos entre exemplos positivos e negativos ao interesse do usuário. Para utilizar essa ferramenta de classificação, o usuário deve entregar ao sistema apenas os documentos de seu interesse, tornando assim o uso do sistema mais prático em aplicações reais. Devido a necessidade de informar uma quantidade de exemplos que represente bem a classe de interesse do usuário, esse modelo de aprendizado pode ainda não ser muito atrativo em relação ao esforço do usuário na classificação de exemplos. No entanto, ao utilizar sua versão semissupervisionada, o usuário então pode fornecer uma pequena quantidade de exemplos de seu interesse uma vez que o modelo fará uso de exemplos não rotulados visando alcançar performances de classificação tão boas e até superiores a versão supervisionada.

A maioria das técnicas de aprendizado semissupervisionado baseado em uma única classe (*Positive and Unlabeled Learning* (PUL)), utilizam para a representação de dados o modelo espaço-vetorial, sendo pouco explorado o uso de grafos, que na classificação de textos tem apresentado performances de classificação tão boas e até superiores ao modelo

espaço-vetorial. Além disso, muitas técnicas fazem uso da abordagem *Self-Training* aliada ao SVM supervisionado para a classificação dos exemplos não rotulados, sendo que o *Self-Training* na classificação de textos geralmente demonstra resultados inferiores a outras abordagens.

Visando as lacunas acima mencionadas, neste projeto de mestrado foi proposto uma abordagem baseada em grafos para o *Positive and Unlabeled Learning*. A abordagem proposta, nomeada *Label Propagation for Positive and Unlabeled Learning* (LP-PUL), consiste de três etapas: (i) construção do grafo, (ii) extração de documentos negativos, e (iii) propagação dos rótulos. Nas etapas (i) e (iii) foram considerados respectivamente diferentes métodos de construção de grafos e de propagação de rótulos. Além disso, o método proposto possui as seguintes características que o diferenciam das demais técnicas existentes:

- utiliza a representação de dados por meio de grafos, além de utilizar a estrutura do grafo gerado na etapa (i) em todas as demais etapas, até a classificação de todos os documentos;
- não aumenta o conjunto de documentos positivos, evitando a inserção de documentos incorretamente rotulados no conjunto;
- não utiliza métodos como *Self-Training* e *Expectation Maximization*, os quais podem degradar a performance de classificação, considerando a classificação semissupervisionada de textos.

5.1 Principais Resultados e Contribuições

A principal contribuição deste projeto de mestrado foi propor um novo método PUL inteiramente baseado em grafos capaz de superar os demais métodos utilizados na avaliação experimental, assim confirmando a positiva contribuição de utilizar documentos não rotulados na classificação, além de confirmar que o uso de grafos geram resultados tão bons e até superiores a representação baseada no modelo espaço-vetorial.

Foi realizada uma avaliação experimental extensa, onde os métodos OCL e PUL foram comparados levando em consideração cinco quantidades de documentos rotulados, a saber, $|\mathcal{D}^+| = \{1, 5, 10, 20, 30\}$. Com isso, foi possível analisar o impacto que a quantidade de exemplos rotulados causa na performance de classificação de cada método. Além disso, foi observado o comportamento do LP-PUL em relação aos demais algoritmos. O método proposto obteve melhor performance de classificação com diferença estatisticamente significativa (DES) em relação a todos os outros métodos quando utilizados 20 e 30 documentos positivos rotulados. Quando foi utilizado 1, 5 e 10 documentos positivos rotulados, o método proposto não foi melhor com DES que os métodos OCL (salvo com

5 documentos positivos rotulados, que o método proposto não obteve DES somente em relação ao k -NND), mas contra os demais métodos PUL, os resultados foram superiores com DES. Além disso, em todas as comparações, o LP-PUL obteve um ranqueamento médio superior a todos os demais métodos.

Por meio dos gráficos utilizados para a comparação dos métodos, também foi apresentado o impacto que a quantidade de documentos rotulados causa na performance de classificação, tanto nos métodos OCL quanto nos métodos PUL.

Também foram conduzidas avaliações para analisar diferentes métodos em cada etapa do LP-PUL. Na etapa de construção do grafo, foram utilizados três algoritmos bem conhecidos na literatura: k -Vizinhos Mais Próximos (k NN), k -Vizinhos Mais Próximos Mútuos (Mk NN) e ϵ -Vizinhança (ϵ N). Foi identificado por meio da análise dos resultados e de testes estatísticos que o grafo k NN geralmente provê as melhores performances de classificação, mas não apresenta resultados estatisticamente superiores a todos os demais algoritmos de construção de grafos. No entanto, apresenta DES em relação ao grafo ϵ N em todos os cenários quando considerado o GFHF como algoritmo de propagação de rótulos, bem como apresenta DES em relação ao grafo Mk NN em todos os cenários quando considerado o LLGC como algoritmo de propagação de rótulos.

Considerando a etapa de propagação de rótulos, foram utilizados dois algoritmos para grafos homogêneos, o *Gaussian Fields and Harmonic Functions* (GFHF) e o *Learning With Local and Global Consistency* (LLGC). Durante a análise dos resultados, foi comprovado que o uso do LLGC gerou melhores resultados que o uso do GFHF com DES quando utilizado 1 documento positivo rotulado, enquanto o GFHF foi melhor com DES quando utilizados 20 ou mais documentos positivos rotulados. Resultados interessantes uma vez que o GFHF não utiliza parâmetros, ao contrário do LLGC que pode ser ajustado em relação a importância dada aos rótulos originais dos objetos rotulados.

Os resultados obtidos nesse projeto de mestrado foram utilizados para submeter um artigo que está em processo de revisão para publicação na revista *Information Sciences*.

5.2 Limitações

Dada a complexidade da execução dos experimentos realizados, este trabalho apresentou algumas limitações quanto à complexidade da avaliação experimental, uma vez que foram utilizadas várias coleções de textos, e para cada coleção de textos, iterativamente cada classe foi considerada como classe de interesse além de serem realizadas diferentes inicializações de exemplos positivos para cada classe. Esse tipo de avaliação foi executado não só no método proposto, mas em todos os métodos utilizados na comparação.

Dado isso, foi necessário limitar o número de algoritmos de propagação de rótulos e o número de algoritmos de construção de grafos, de forma que se pudessem obter os resultados em tempo viável. Optou-se por escolher dois algoritmos tradicionais de propagação de rótulos (LLGC e GFHF) e, que além disso, possuem uma pequena quantidade de parâmetros a ser definido, e três algoritmos de construção de grafos tradicionais (k NN, Mk NN e ϵ N).

Também não foram implementados a tempo algoritmos de OCL ou PUL baseados em aprendizado profundo, uma vez que a implementação dos algoritmos propostos e utilizados, bem como a avaliação dos mesmos tomaram todo o tempo disponível. Além disso, como os algoritmos de OCL e PUL baseados em aprendizado profundo encontrados na literatura não apresentaram comparativos de resultados com os métodos tradicionais, não foram obtidos subsídios para encorajar a implementação de tais métodos (XU; TAN, 2019).

5.3 Trabalhos Futuros

Como trabalhos futuros, pretende-se considerar outros tipos de grafos além dos baseados em similaridade, tais como, rede de termos (ROSSI; REZENDE; LOPES, 2015) ou então redes heterogêneas (JI *et al.*, 2010; ROSSI; LOPES; REZENDE, 2016; LIU; WANG, 2020). Além disso, pretende-se utilizar modelos de linguagens para representar os documentos antes da construção de grafos (PETERS *et al.*, 2018; MIKOLOV *et al.*, 2013b).

REFERÊNCIAS

AGGARWAL, C. C. **Data Classification: Algorithms and Applications**. 1st. ed. [S.l.]: Chapman Hall/CRC, 2014. ISBN 1466586745. Citado nas páginas 21, 25 e 26.

_____. **Machine learning for text**. [S.l.]: Springer, 2018. ISBN 978-3319735306. Citado nas páginas 26 e 27.

AHMED, M.; MAHMOOD, A. N.; HU, J. A survey of network anomaly detection techniques. **Journal of Network and Computer Applications**, Academic Press Ltd., v. 60, n. C, p. 19–31, 2016. ISSN 1084-8045. Citado na página 20.

ALMEIDA, L. J. **Detecção de comunidades em redes complexas utilizando estratégia multinível**. Tese (Mestrado em Ciências de Computação e Matemática Computacional) — Ciências de Computação e Matemática Computacional, Universidade de São Paulo, São Carlos, 2009. Disponível em: <www.teses.usp.br/teses/disponiveis/55/55134/tde-29032010-161550/publico/Dissertacao.pdf>. Acesso em: 04 jan. 2019. Citado na página 30.

AMANCIO, D. R.; JR, O. N. O.; COSTA, L. d. F. On the use of topological features and hierarchical characterization for disambiguating names in collaborative networks. **EPL (Europhysics Letters)**, IOP Publishing, v. 99, n. 4, p. 48002, 2012. Citado nas páginas 21 e 28.

_____. Unveiling the relationship between complex networks metrics and word senses. **EPL (Europhysics Letters)**, IOP Publishing, v. 98, n. 1, p. 18002, 2012. Citado nas páginas 21 e 28.

AMINI, M.-R.; USUNIER, N. **Learning with Partially Labeled and Interdependent Data**. [S.l.]: Springer, 2015. ISBN 978-3319157269. Citado na página 32.

ANGELOVA, R.; WEIKUM, G. Graph-based text classification: Learn from your neighbors. In: **Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. [S.l.]: Association for Computing Machinery, 2006. p. 485–492. ISBN 1595933697. Citado nas páginas 21, 30 e 39.

ANSTEAD, N.; O'LOUGHLIN, B. Social Media Analysis and Public Opinion: The 2010 Uk General Election. **Journal of Computer-Mediated Communication**, v. 20, n. 2, p. 204–220, 2014. ISSN 1083-6101. Citado na página 19.

ARRUDA, H. F. de; MARINHO, V. Q.; COSTA, L. d. F.; AMANCIO, D. R. Paragraph-based representation of texts: A complex networks approach. **Information Processing & Management**, Elsevier, v. 56, n. 3, p. 479–494, 2019. Citado na página 29.

BAEZA-YATES, R. A.; RIBEIRO-NETO, B. **Modern Information Retrieval**. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1999. ISBN 020139829X. Citado na página 33.

- BERTON, L. **Construção de redes baseadas em vizinhança para o aprendizado semissupervisionado**. Tese (Doutorado em Ciências de Computação e Matemática Computacional) — Ciências de Computação e Matemática Computacional, Universidade de São Paulo, São Carlos, 2016. Disponível em: <www.teses.usp.br/teses/disponiveis/55/55134/tde-29072016-100548/publico/LilianBerton_revisada.pdf>. Acesso em: 03 jan. 2019. Citado nas páginas 21 e 30.
- BERTON, L.; LOPES, A. de A.; VEGA-OLIVEROS, D. A. A comparison of graph construction methods for semi-supervised learning. In: **Proceedings of the International Joint Conference on Neural Networks**. [S.l.]: IEEE, 2018. p. 1–8. Citado na página 30.
- BIEMANN, C.; MEHLER, A. **Text Mining: From Ontology Learning to Automated Text Processing Applications**. [S.l.]: Springer, 2014. ISBN 978-3319126548. Citado na página 19.
- BLUM, A.; MITCHELL, T. Combining labeled and unlabeled data with co-training. In: **Proceedings of the Annual Conference on Computational Learning Theory**. [S.l.]: Association for Computing Machinery, 1998. p. 92–100. ISBN 1581130570. Citado na página 35.
- BOJANOWSKI, P.; GRAVE, E.; JOULIN, A.; MIKOLOV, T. Enriching word vectors with subword information. **Transactions of the Association for Computational Linguistics**, MIT Press, v. 5, p. 135–146, 2017. Citado na página 27.
- CAI, D.; WANG, X.; HE, X. Probabilistic dyadic data analysis with local and global consistency. In: **Proceedings of the Annual International Conference on Machine Learning**. [S.l.]: Association for Computing Machinery, 2009. p. 105–112. ISBN 9781605585161. Citado na página 29.
- CAMPOS, R.; MANGARAVITE, V.; PASQUALI, A.; JORGE, A. M.; NUNES, C.; JATOWT, A. A text feature based automatic keyword extraction method for single documents. In: PASI, G.; PIWOWARSKI, B.; AZZOPARDI, L.; HANBURY, A. (Ed.). **Advances in Information Retrieval**. [S.l.]: Springer International Publishing, 2018. p. 684–691. ISBN 978-3-319-76941-7. Citado na página 19.
- CARUANA, R.; NICULESCU-MIZIL, A. An empirical comparison of supervised learning algorithms. In: **Proceedings of the International Conference on Machine Learning**. [S.l.]: Association for Computing Machinery, 2006. p. 161–168. ISBN 1595933832. Citado na página 61.
- CHAKRABARTI, S.; DOM, B. E.; AGRAWAL, R.; RAGHAVAN, P. Using taxonomy, discriminants, and signatures for navigating in text databases. In: **Proceedings of the International Conference on Very Large Data Bases**. [S.l.]: Morgan Kaufmann Publishers, 1997. p. 446–455. Citado nas páginas 21 e 30.
- CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM Computing Surveys**, Association for Computing Machinery, v. 41, n. 3, 2009. ISSN 0360-0300. Citado nas páginas 20, 44 e 45.
- CHAPELLE, O.; SCHOLKOPF, B.; ZIEN, A. **Semi-Supervised Learning**. [S.l.]: The MIT Press, 2006. (Adaptive Computation and Machine Learning). ISBN 0262033585, 978-0262033589, 978-1429414081. Citado nas páginas 32, 33, 34, 39 e 51.

Chen, K.; Wang, S. Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 33, n. 1, p. 129–143, 2011. Citado na página 32.

COREL, E.; MÉHEUST, R.; WATSON, A. K.; MCINERNEY, J. O.; LOPEZ, P.; BAPTESTE, E. Bipartite network analysis of gene sharings in the microbial world. **Molecular biology and evolution**, Oxford University Press, v. 35, n. 4, p. 899–913, 2018. Citado na página 28.

COSTA, L. d. F.; RODRIGUES, F. A.; TRAVIESO, G.; BOAS, P. R. V. Characterization of complex networks: A survey of measurements. **Advances in physics**, Taylor & Francis, v. 56, n. 1, p. 167–242, 2007. Citado na página 30.

COZMAN, F. G.; COHEN, I.; CIRELO, M. C. Semi-supervised learning of mixture models. In: **Proceedings of the International Conference on Machine Learning**. [S.l.: s.n.], 2003. p. 99–106. Citado na página 34.

DELALLEAU, O.; BENGIO, Y.; ROUX, N. L. Efficient non-parametric function induction in semi-supervised learning. In: **Proceedings of the International Workshop on Artificial Intelligence and Statistics**. [S.l.: s.n.], 2005. p. 96–103. Citado na página 40.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. **Journal of the Royal Statistical Society: Series B (Methodological)**, Wiley Online Library, v. 39, n. 1, p. 1–22, 1977. Citado nas páginas 35 e 49.

DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. **Journal of Machine Learning Research**, JMLR.org, v. 7, p. 1–30, 2006. ISSN 1532-4435. Citado na página 67.

DIJKSTRA, E. W. A note on two problems in connexion with graphs. **Numerische mathematik**, Springer, v. 1, n. 1, p. 269–271, 1959. Citado na página 56.

FIOK, K.; KARWOWSKI, W.; GUTIERREZ, E.; REZA-DAVAHLI, M. Comparing the quality and speed of sentence classification with modern language models. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 10, n. 10, p. 3386, 2020. Citado na página 27.

FORTUNATO, S.; HRIC, D. Community detection in networks: A user guide. **Physics Reports**, Elsevier, v. 659, p. 1–44, 2016. ISSN 0370-1573. Citado na página 27.

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 30.

GUÉDON, O.; VERSHYNIN, R. Community detection in sparse networks via grothendieck’s inequality. **Probability Theory and Related Fields**, Springer, v. 165, n. 3-4, p. 1025–1049, 2016. Citado na página 28.

GUPTA, H. K.; PARVEEN, R. Comparative study of big data frameworks. In: **Proceedings of the International Conference on Issues and Challenges in Intelligent Computing Techniques**. [S.l.]: IEEE, 2019. v. 1, p. 1–4. Citado nas páginas 19 e 32.

JASKIE, K.; SPANIAS, A. Positive and unlabeled learning algorithms and applications: A survey. In: **Proceedings of the International Conference on Information, Intelligence, Systems and Applications**. [S.l.]: IEEE, 2019. p. 1–8. Citado nas páginas 20, 22, 47 e 48.

JI, M.; SUN, Y.; DANILEVSKY, M.; HAN, J.; GAO, J. Graph regularized transductive classification on heterogeneous information networks. In: **Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases**. [S.l.]: Springer-Verlag, 2010. p. 570–586. ISBN 3-642-15879-X, 978-3-642-15879-7. Citado na página 74.

JOACHIMS, T. Transductive inference for text classification using support vector machines. In: **Proceedings of the International Conference on Machine Learning**. [S.l.]: Morgan Kaufmann Publishers Inc., 1999. v. 99, p. 200–209. ISBN 1558606122. Citado na página 37.

_____. Transductive learning via spectral graph partitioning. In: **Proceedings of the International Conference on Machine Learning**. [S.l.]: AAAI Press, 2003. p. 290–297. ISBN 1577351894. Citado na página 33.

JR, E. A. C.; LOPES, A. A.; AMANCIO, D. R. Word sense disambiguation: A complex network approach. **Information Sciences**, Elsevier, v. 442, p. 103–113, 2018. Citado nas páginas 21 e 28.

KHAN, S. S.; MADDEN, M. G. A survey of recent trends in one class classification. In: **Artificial Intelligence and Cognitive Science**. [S.l.]: Springer Berlin Heidelberg, 2010. p. 188–197. ISBN 978-3642170805. Citado na página 44.

KIM, J.; JANG, S.; PARK, E.; CHOI, S. Text classification using capsules. **Neurocomputing**, Elsevier, v. 376, p. 214–221, 2020. ISSN 0925-2312. Citado na página 19.

KOWSARI, K.; MEIMANDI, K. J.; HEIDARYSAFA, M.; MENDU, S.; BARNES, L.; BROWN, D. Text classification algorithms: A survey. **Information**, Multidisciplinary Digital Publishing Institute, v. 10, n. 4, p. 150, 2019. Citado na página 19.

LI, C.; CHEN, S.; XING, J.; SUN, A.; MA, Z. Seed-guided topic model for document filtering and classification. **ACM Transactions on Information Systems**, Association for Computing Machinery, v. 37, n. 1, p. 1–37, 2018. ISSN 1046-8188. Citado na página 19.

LI, J.; LIU, Y.; YIN, R.; WANG, W. Multi-class learning using unlabeled samples: Theory and algorithm. In: **Proceedings of the International Joint Conference on Artificial Intelligence**. [S.l.]: International Joint Conferences on Artificial Intelligence Organization, 2019. p. 2880–2886. Citado nas páginas 43 e 44.

LI, X.; LIU, B. Learning to classify texts using positive and unlabeled data. In: **Proceedings of the International Joint Conference on Artificial Intelligence**. [S.l.]: Morgan Kaufmann Publishers Inc., 2003. v. 3, p. 587–592. Citado nas páginas 21, 22, 47, 49 e 51.

LI, X.-L.; YU, P. S.; LIU, B.; NG, S.-K. Positive unlabeled learning for data stream classification. In: **Proceedings of the SIAM International Conference on Data**

Mining. [S.l.]: Society for Industrial and Applied Mathematics, 2009. p. 259–270. Citado nas páginas [20](#), [21](#), [47](#), [49](#), [50](#), [51](#) e [62](#).

LIU, B.; LEE, W. S.; YU, P. S.; LI, X. Partially supervised classification of text documents. In: **Proceedings of the International Conference on Machine Learning**. [S.l.]: Morgan Kaufmann Publishers Inc., 2002. p. 387–394. ISBN 1558608737. Citado nas páginas [20](#), [21](#), [22](#), [47](#), [49](#), [51](#) e [62](#).

LIU, J.; CHEN, Y.; XIE, J.; GAO, X.; GAO, W. Semi-supervised learning of caricature pattern from manifold regularization. In: **Advances in Multimedia Modeling**. [S.l.]: Springer Berlin Heidelberg, 2009. p. 413–424. ISBN 978-3540928928. Citado na página [39](#).

LIU, J.; DOLAN, P.; PEDERSEN, E. R. Personalized news recommendation based on click behavior. In: **Proceedings of the International Conference on Intelligent User Interfaces**. [S.l.]: Association for Computing Machinery, 2010. p. 31–40. ISBN 9781605585154. Citado na página [20](#).

LIU, L.; WANG, S. Meta-path-based outlier detection in heterogeneous information network. **Frontiers of Computer Science**, Springer, v. 14, n. 2, p. 388–403, 2020. Citado na página [74](#).

LIU, W.; CHANG, S.-F. Robust multi-class transductive learning with graphs. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.]: IEEE, 2009. p. 381–388. Citado na página [40](#).

LU, Q.; GETOOR, L. Link-based classification. In: **Proceedings of the International Conference on Machine Learning**. [S.l.]: AAAI Press, 2003. p. 496–503. Citado nas páginas [21](#) e [30](#).

MA, S.; ZHANG, R. Pu-lp: A novel approach for positive and unlabeled learning by label propagation. In: **Proceedings of the IEEE International Conference on Multimedia Expo Workshops**. [S.l.]: IEEE, 2017. p. 537–542. Citado nas páginas [20](#), [22](#), [47](#), [49](#), [50](#), [52](#), [61](#) e [62](#).

MANNING, C.; RAGHAVAN, P.; SCHÜTZE, H. Introduction to information retrieval. **Natural Language Engineering**, Cambridge university press, v. 16, n. 1, p. 100–103, 2010. Citado na página [26](#).

MARCACINI, R. M.; REZENDE, S. O. Incremental construction of topic hierarchies using hierarchical term clustering. In: **Proceedings of the International Conference on Software Engineering and Knowledge Engineering**. [S.l.]: Knowledge Systems Institute Graduate School, 2010. p. 553. Citado na página [29](#).

MARCACINI, R. M.; ROSSI, R. G.; NOGUEIRA, B. M.; MARTINS, L. V.; CHERMAN, E. A.; REZENDE, S. O. Websensors analytics: Learning to sense the real world using web news events. In: **Anais Estendidos do XXIII Simpósio Brasileiro de Sistemas Multimídia e Web**. [S.l.]: SBC, 2017. p. 169–173. ISSN 2596-1683. Citado nas páginas [20](#) e [44](#).

MIHALCEA, R.; TARAU, P. Textrank: Bringing order into text. In: **Proceedings of the Conference on Empirical Methods in Natural Language Processing**. [S.l.]: Association for Computational Linguistics, 2004. p. 404–411. Citado na página [29](#).

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. **arXiv preprint arXiv:1301.3781**, 2013. Citado na página 27.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G.; DEAN, J. Distributed representations of words and phrases and their compositionality. In: **Proceedings of the International Conference on Neural Information Processing Systems**. [S.l.]: Curran Associates Inc., 2013. p. 3111–3119. Citado na página 74.

MUIR, A. K. S. **Lean Six Sigma statistics: Calculating process efficiencies in transactional projects**. [S.l.]: McGraw-Hill New York, 2006. ISBN 978-0071445856. Citado na página 60.

MUNROE, D. T.; MADDEN, M. G. Multi-class and single-class classification approaches to vehicle model recognition from images. In: **Proceedings of the Irish Conference on Artificial Intelligence and Cognitive Science**. [S.l.]: AICS, 2005. p. 1–11. Citado na página 44.

NASAR, Z.; JAFFRY, S. W.; MALIK, M. K. Textual keyword extraction and summarization: State-of-the-art. **Information Processing & Management**, Elsevier, v. 56, n. 6, p. 102088, 2019. Citado nas páginas 21 e 28.

NEWMAN, M. Community detection in networks: Modularity optimization and maximum likelihood are equivalent. **arXiv preprint arXiv:1606.02319**, 2016. Citado na página 28.

NIGAM, K.; MCCALLUM, A. K.; THRUN, S.; MITCHELL, T. Text classification from labeled and unlabeled documents using em. **Machine learning**, Springer, v. 39, n. 2-3, p. 103–134, 2000. Citado nas páginas 35 e 36.

_____. Text classification from labeled and unlabeled documents using EM. **Machine Learning**, Kluwer Academic Publishers, v. 39, n. 2-3, p. 103–134, 2000. ISSN 0885-6125. Citado na página 51.

PETERS, M.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; ZETTLEMOYER, L. Deep contextualized word representations. In: **Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. [S.l.]: Association for Computational Linguistics, 2018. p. 2227–2237. Citado na página 74.

PITA, M.; PAPPA, G. L. Strategies for short text representation in the word vector space. In: **Proceedings of the Brazilian Conference on Intelligent Systems**. [S.l.]: IEEE, 2018. p. 266–271. Citado na página 27.

RABELO, J.; PRUDÊNCIO, R. B.; BARROS, F. Collective classification for sentiment analysis in social networks. In: **Proceedings of the IEEE International Conference on Tools with Artificial Intelligence**. [S.l.]: IEEE, 2012. v. 1, p. 958–963. Citado nas páginas 21, 28 e 34.

ROCCHIO, J. J. Relevance feedback in information retrieval. **The SMART retrieval system: experiments in automatic document processing**, Prentice-Hall Inc., p. 313–323, 1971. Citado na página 49.

ROSSI, R. G. **Classificação automática de textos por meio de aprendizado de máquina baseado em redes**. Tese (Doutorado em Ciências de Computação e Matemática Computacional) — Ciências de Computação e Matemática Computacional, Universidade de São Paulo, São Carlos, 2015. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-05042016-105648/publico/VersaoRevisada_RafaelGeraldelliRossi.pdf>. Acesso em: 28 dez. 2018. Citado nas páginas 22, 26, 28, 29, 33, 34, 38, 39, 40, 41, 51 e 60.

ROSSI, R. G.; LOPES, A. A.; REZENDE, S. O. A parameter-free label propagation algorithm using bipartite heterogeneous networks for text classification. In: **Proceedings of the Annual ACM Symposium on Applied Computing**. [S.l.]: Association for Computing Machinery, 2014. p. 79–84. ISBN 9781450324694. Citado nas páginas 28 e 32.

ROSSI, R. G.; LOPES, A. de A.; FALEIROS, T. de P.; REZENDE, S. O. Inductive model generation for text classification using a bipartite heterogeneous network. **Journal of Computer Science and Technology**, Springer, v. 29, n. 3, p. 361–375, 2014. Citado nas páginas 22, 51 e 61.

ROSSI, R. G.; LOPES, A. de A.; REZENDE, S. O. Optimization and label propagation in bipartite heterogeneous networks to improve transductive classification of texts. **Information Processing & Management**, Elsevier, v. 52, n. 2, p. 217–257, 2016. Citado nas páginas 21, 22, 28, 34, 51, 55 e 74.

_____. Using bipartite heterogeneous networks to speed up inductive semi-supervised learning and improve automatic text categorization. **Knowledge-Based Systems**, Elsevier, v. 132, p. 94–118, 2017. Citado nas páginas 22, 25 e 51.

ROSSI, R. G.; MARCACINI, R. M.; REZENDE, S. O. Benchmarking text collections for classification and clustering tasks. **Relatório Técnico, Institute of Mathematics and Computer Sciences, University of Sao Paulo**, 2013. Citado nas páginas 59 e 60.

ROSSI, R. G.; REZENDE, S. O.; LOPES, A. de A. Term network approach for transductive classification. In: **Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics**. [S.l.]: Springer International Publishing, 2015. p. 497–515. ISBN 978-3319181172. Citado nas páginas 22, 51 e 74.

SALTON, G. **Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer**. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201122278. Citado na página 26.

SEBASTIANI, F. Machine learning in automated text categorization. **ACM Computing Surveys**, Association for Computing Machinery, v. 34, n. 1, p. 1–47, 2002. ISSN 0360-0300. Citado nas páginas 43 e 44.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding machine learning: From theory to algorithms**. [S.l.]: Cambridge university press, 2014. ISBN 978-1107057135. Citado nas páginas 21 e 26.

SILVA, T. C.; ZHAO, L. **Machine Learning in Complex Networks**. [S.l.]: Springer, 2016. ISBN 978-3-31-917289-7, 978-3-31-917290-3. Citado na página 30.

SINOARA, R. A.; CAMACHO-COLLADOS, J.; ROSSI, R. G.; NAVIGLI, R.; REZENDE, S. O. Knowledge-enhanced document embeddings for text classification. **Knowledge-Based Systems**, Elsevier, v. 163, p. 955–971, 2019. Citado na página 19.

SOARES, M. M.; CORSO, G.; LUCENA, L. The network of syllables in portuguese. **Physica A: Statistical Mechanics and its Applications**, Elsevier, v. 355, n. 2-4, p. 678–684, 2005. Citado na página 29.

SOUSA, C. A. R. de; REZENDE, S. O.; BATISTA, G. E. A. P. A. Influence of graph construction on semi-supervised learning. In: **Machine Learning and Knowledge Discovery in Databases**. [S.l.]: Springer Berlin Heidelberg, 2013. p. 160–175. ISBN 978-3-642-40994-3. Citado nas páginas 21, 30 e 55.

SUBRAMANYA, A.; BILMES, J. Soft-supervised learning for text classification. In: **Proceedings of the Conference on Empirical Methods in Natural Language Processing**. [S.l.]: Association for Computational Linguistics, 2008. p. 1090–1099. Citado nas páginas 21 e 30.

SUBRAMANYA, A.; TALUKDAR, P. P. Graph-based semi-supervised learning. **Synthesis Lectures on Artificial Intelligence and Machine Learning**, Morgan & Claypool Publishers, v. 8, n. 4, p. 1–125, 2014. Citado na página 32.

TAN, P.; STEINBACH, M.; KARPATNE, A.; KUMAR, V. Anomaly detection. In: **Introduction to Data Mining**. Oxford: Pearson, 2019. cap. 10, p. 651–684. Citado na página 47.

_____. **Introduction to Data Mining**. [S.l.]: Pearson, 2019. (What's New in Computer Science Series). ISBN 978-0133128901. Citado nas páginas 19, 20, 30, 44 e 45.

TAX, D. M. J. One-class classification: concept-learning in the absence of counter-examples [ph. d. thesis]. **Delft University of Technology**, 2001. Citado nas páginas 20, 44 e 45.

TRAWIŃSKI, B.; SMETEK, M.; TELEC, Z.; LASOTA, T. Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. **International Journal of Applied Mathematics and Computer Science**, Versita, v. 22, n. 4, p. 867–881, 2012. Citado nas páginas 64, 65 e 67.

VALEJO, A. D. B. **Refinamento multinível em redes complexas baseado em similaridade de vizinhança**. Tese (Mestrado em Ciências de Computação e Matemática Computacional) — Ciências de Computação e Matemática Computacional, Universidade de São Paulo, São Carlos, 2014. Disponível em: <www.teses.usp.br/teses/disponiveis/55/55134/tde-14042015-142526/publico/AlanDemetriusBarianValejo_dissertacaoRevisada.pdf>. Acesso em: 04 jan. 2019. Citado na página 30.

VAPNIK, V. **Statistical learning theory**. [S.l.]: Wiley, New York, 1998. v. 3. ISBN 978-0471030034. Citado na página 37.

WANG, C.; LIU, J.; DESAI, N.; DANILEVSKY, M.; HAN, J. Constructing topical hierarchies in heterogeneous information networks. **Knowledge and Information Systems**, Springer, v. 44, n. 3, p. 529–558, 2015. Citado na página 29.

WANG, C.; SONG, Y.; LI, H.; ZHANG, M.; HAN, J. Text classification with heterogeneous information network kernels. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.]: AAAI Press, 2016. p. 2130–2136. Citado nas páginas 19, 21 e 28.

WANG, Y.; GONG, Z. Hierarchical classification of web pages using support vector machine. In: **Proceedings of the International Conference on Asian Digital Libraries: Universal and Ubiquitous Access to Information**. [S.l.]: Springer-Verlag, 2008. p. 12–21. ISBN 9783540895329. Citado na página 19.

WITTEN, I. H.; FRANK, E.; HALL, M. A.; PAL, C. J. **Data Mining: Practical machine learning tools and techniques**. [S.l.]: Morgan Kaufmann, 2016. ISBN 978-0128042915. Citado nas páginas 19, 43 e 44.

WU, T.; LIU, S.; ZHANG, J.; XIANG, Y. Twitter spam detection based on deep learning. In: **Proceedings of the Australasian Computer Science Week Multiconference**. [S.l.]: Association for Computing Machinery, 2017. ISBN 9781450347686. Citado na página 19.

XIA, L.; LUO, D.; ZHANG, C.; WU, Z. A survey of topic models in text classification. In: **Proceedings of the International Conference on Artificial Intelligence and Big Data**. [S.l.]: IEEE, 2019. p. 244–250. Citado na página 27.

XU, W.; TAN, Y. Semisupervised text classification by variational autoencoder. **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, v. 31, n. 1, p. 295–308, 2019. Citado na página 74.

YAROWSKY, D. Unsupervised word sense disambiguation rivaling supervised methods. In: **Proceedings of the Annual Meeting of the Association for Computational Linguistics**. [S.l.]: Association for Computational Linguistics, 1995. p. 189–196. Citado na página 34.

YU, H.; HAN, J.; CHANG, K. C.-C. Pebl: Web page classification without negative examples. **IEEE Transactions on Knowledge and Data Engineering**, IEEE Educational Activities Department, v. 16, n. 1, p. 70–81, 2004. ISSN 1041-4347. Citado nas páginas 21, 22, 49 e 51.

YU, H.; ZHAI, C.; HAN, J. Text classification from positive and unlabeled documents. In: **Proceedings of the International Conference on Information and Knowledge Management**. [S.l.]: Association for Computing Machinery, 2003. p. 232–239. ISBN 1581137230. Citado nas páginas 21, 22, 49, 51 e 62.

YU, S.; LI, C. PE-PUC: A graph based PU-learning approach for text classification. In: **Machine Learning and Data Mining in Pattern Recognition**. [S.l.]: Springer Berlin Heidelberg, 2007. p. 574–584. ISBN 978-3-540-73499-4. Citado nas páginas 20, 22, 49, 50, 51, 52 e 61.

ZHANG, D.; LEE, W. S. A simple probabilistic approach to learning from positive and unlabeled examples. In: **Proceedings of the Annual UK Workshop on Computational Intelligence**. [S.l.: s.n.], 2005. p. 83–87. Citado nas páginas 20 e 51.

ZHOU, D.; BOUSQUET, O.; LAL, T. N.; WESTON, J.; SCHÖLKOPF, B. Learning with local and global consistency. In: **Proceedings of the International Conference on Neural Information Processing Systems**. [S.l.]: MIT Press, 2003. p. 321–328. Citado nas páginas [22](#), [39](#), [40](#), [42](#) e [51](#).

ZHU, X. Semi-supervised learning literature survey. **Relatório Técnico, Computer Sciences, University of Wisconsin-Madison**, n. 1530, 2005. Citado na página [39](#).

ZHU, X.; GHAHRAMANI, Z.; LAFFERTY, J. Semi-supervised learning using gaussian fields and harmonic functions. In: **Proceedings of the International Conference on International Conference on Machine Learning**. [S.l.]: AAAI Press, 2003. p. 912–919. ISBN 1577351894. Citado nas páginas [22](#), [40](#) e [50](#).

ZHU, X.; GOLDBERG, A. B. **Introduction to Semi-Supervised Learning**. [S.l.]: Morgan and Claypool Publishers, 2009. ISBN 1598295470. Citado nas páginas [32](#), [34](#), [35](#) e [41](#).

ZHU, X.; LAFFERTY, J.; ROSENFELD, R. **Semi-supervised learning with graphs**. Tese (Doutorado) — Carnegie Mellon University, 2005. Citado na página [41](#).