UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Processamento de sinal em grafos: Teoria de amostragem e sua aplicação no aprendizado semi-supervisionado ativo

Evaristo Calisto Nhassengo

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

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Evaristo Calisto Nhassengo

Processamento de sinal em grafos: Teoria de amostragem e sua aplicação no aprendizado semi-supervisionado ativo

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. José Alberto Cuminato

USP – São Carlos
Novembro de 2019

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

N576p Nhassengo, Evaristo Calisto
Processamento de sinal em grafos: Teoria de amostragem e sua aplicação no aprendizado semi-supervisionado ativo / Evaristo Calisto Nhassengo; orientador José Alberto Cuminato. -- São Carlos, 2020.
89 p.

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

1. Processamento de sinais em grafos. 2. Teoria de amostragem. 3. Aprendizado Semi-supervisionado ativo. I. Cuminato, José Alberto, orient. II. Título.

Evaristo Calisto Nhassengo

**Signal Processing on Graphs: Sampling Theory and its
Application to Active Semi-supervised Learning**

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. José Alberto Cuminato

USP – São Carlos
November 2019

À minha família e amigos.

AGRADECIMENTOS

A realização deste trabalho contou com a colaboração de diversas pessoas as quais gostaria de expressar sinceros agradecimentos.

- Em primeiro lugar agradeço a Deus por ter me ajudado a superar vários obstáculos que surgiram neste período.
- Aos meus pais, Calisto Nhassengo e Eva Nhassengo, às minhas irmãs Nilza, Quinaula e Ercilia pelo incentivo e apoio incondicional.
- Aos meus supervisores José Alberto Cuminato e Luís Gustavo Nonato, pela paciência que tiveram comigo nos ensinamentos, e pelas sugestões que me deram ao longo da realização deste trabalho. A dedicação ao trabalho e a atenção com que se dirigiram a mim em todos os momentos, posso dizer com extrema sinceridade, servem-me de modelo.
- Aos meus familiares, amigos e colegas: Juniomar Organista, Laison Furlan, Vinícius Hagemeyer, Anna Caroline, Ana Lemmes, Gustavo Miziara, Hugo França e Aquisson.
- Ao Instituto de Ciências Matemáticas e de Computação (ICMC) e seus funcionários dedicados, pelo apoio técnico-administrativo, pela competência na realização de seu trabalho, pela paciência, e pelas orientações.
- Ao Instituto Superior Politécnico De Songo (ISPS), pelo financiamento ao longo do meu mestrado e pelo apoio moral durante este período.

“No meio da dificuldade encontra-se a oportunidade.”
(Albert Einstein)

RESUMO

NHASSENCO, E.C. **Processamento de sinal em grafos: Teoria de amostragem e sua aplicação no aprendizado semi-supervisionado ativo**. 2019. 86 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, 2019.

No uso dos algoritmos de aprendizado de máquina para as tarefas de classificação, admite-se a existência de um conjunto de exemplos rotulados conhecido como conjunto de treinamento. Este conjunto é utilizado para treinar um classificador. Porém em vários casos de aplicação real o conjunto de treinamento pode não ser suficiente para treinar um bom classificador.

Existe uma variação dos algoritmos de aprendizado de máquina supervisionado, conhecida como algoritmos semi-supervisionados. Os algoritmos semi-supervisionados, assumem que, juntamente com o conjunto de treinamento, existe um segundo conjunto de exemplos não rotulados, também disponível durante o treinamento. Um dos objetivos dos algoritmos semi-supervisionados é o treinamento de classificadores quando uma grande quantidade de exemplos não rotulados está disponível e o conjunto de exemplos rotulados é escasso. Para tratar o problema da escassez de dados rotulados propõe-se uma escolha estratégica de exemplos a rotular para uma classificação eficaz.

Neste trabalho é generalizado o problema clássico da amostragem de sinais, considera-se o problema da amostragem de sinais definidos em grafos. A teoria de amostragem de sinais em grafos estuda o problema da escolha do melhor subconjunto de vértices para a reconstrução perfeita do sinal através de sua amostra. No estudo de sinais através de grafos a representação do domínio de frequência é dada através dos autovalores e autovetores do Laplaciano.

Aplica-se o algoritmo de amostragem de sinais em grafos ([ANIS; GADDE; ORTEGA, 2016](#)) para selecionar o melhor conjunto de treinamento em um conjunto de dados. Estuda-se o efeito que a amostragem de sinais em grafos tem nos algoritmos de aprendizado semi-supervisionado.

Propõe-se um algoritmo semi-supervisionado ativo baseado no algoritmo proposto por [Anis, Gadde e Ortega \(2016\)](#) e no algoritmo de propagação de rótulos **LLGC** ([ZHOU *et al.*, 2004](#)). A proposta é modificar a matriz de similaridade dos dados, para considerar amostras da região de classificação incerta a cada iteração.

Mostra-se através de experimentos numéricos que a proposta supera outras técnicas de aprendizado ativo, quando os dados rotulados são escassos. Nos experimentos são usados conjuntos de dados reais e artificiais.

Palavras-chave: Processamento de sinais em grafos, Teoria de amostragem, Transformada de Fourier do grafo, Aprendizado Semi-supervisionado.

ABSTRACT

NHASSENGO, E.C. **Signal Processing on Graphs: Sampling Theory and its Application to Active Semi-supervised Learning**. 2019. 86 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, 2019.

In the use of machine learning algorithms for classification tasks, it is assumed that there is a set of labeled examples, known as the training set. This set is used to train a classifier. However in many cases of actual application the training set may not be sufficient to train a good classifier.

There is a class of supervised machine learning algorithms, known as semi-supervised algorithms. Semi-supervised algorithms assume that, along with the training set, there is a second set of unlabeled examples also available during training. One of the goals of semi-supervised algorithms is to train classifiers when a large number of unlabeled examples are available and the set of labeled examples is scarce. To address the problem of labeled data scarcity, a strategic choice of data to label for effective classification is proposed.

In this work it is generalized the classical problem of sampling, it is considered the problem of sampling signals defined on graphs. The theory of sampling signals on graphs studies the problem of choosing the best subset of vertices for the perfect reconstruction of the original signal through its sample. In the study of signals through graphs the representation of the frequency domain is given by the eigenvalues and eigenvectors of the Laplacian.

The graph signal sampling algorithm [Anis, Gadde e Ortega \(2016\)](#) is applied to select the best training set from a data set. We study the effect that signal sampling on graphs can have on semi-supervised learning algorithms.

An active semi-supervised algorithm based on the algorithm proposed by ([ANIS; GADDE; ORTEGA, 2016](#)) and the label propagation algorithm **LLGC** ([ZHOU *et al.*, 2004](#)) is proposed. The proposal is to modify the data similarity matrix to consider samples from the uncertain classification region at each iteration.

It is shown through numerical experiments that the proposal outperform other active learning techniques, when the labeled data is scarce. In the experiments real and artificial data sets are used.

Keywords: Signal processing on graphs, Sampling Theory, Graph Fourier Transform, Semi-supervised Learning .

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de um grafo de uma rede social	27
Figura 2 – Diagrama de aprendizado semi-supervisionado considerado neste trabalho. . .	30
Figura 3 – Métodos de aprendizado de máquina.	32
Figura 4 – Métodos de aprendizado semi-supervisionado.	35
Figura 5 – Ciclo do aprendizado ativo	38
Figura 6 – Curvas de aprendizado da classificação de textos: <i>baseball</i> vs <i>hockey</i> . O eixo horizontal indica o número de instâncias consideradas para o treino e o eixo vertical indica a acurácia obtida. Cada curva indica uma estratégia de seleção. É notável que a estratégia usando o aprendizado ativo obtém uma classificação melhor que a amostragem aleatória.	39
Figura 7 – Amostragem seguida de interpolação.	40
Figura 8 – (a) Amostra ótima $ \mathcal{S} = 25$ (vértices de cor vermelha) de um grafo com 200 vértices. (b) Amostra ótima $ \mathcal{S} = 20$ (vértices de cor vermelha) de um grafo circular com 200 vértices e arestas adicionais.	42
Figura 9 – Comparação dos métodos de aprendizado semi-supervisionado com o método proposto no trabalho (GADDE; ANIS; ORTEGA, 2014) usando conjunto de dados reais.	43
Figura 10 – Grafo de Erdős-Renyi. Todos os pesos das arestas são iguais a 1. O grafo abaixo contém 64 nós, e 188 arestas e foi gerado considerando $p = 0.1$. . .	46
Figura 11 – Exemplo de sinais. Valores dos sinais são representados por cores. (a) medições de temperatura em diferentes regiões dos Estados Unidos, o grafo representa uma rede de sensores meteorológicos. (b) um sinal aleatório definido num grafo do tipo cometa. O grafo do cometa é um grafo de caminho com uma estrela de grau k no seu final	47
Figura 12 – Comportamento dos autovetores $u_0, u_5, u_{10}, u_{15}, u_{20}$ de um grafo de grelha com 100 vértices. O eixo vertical indica o índice do autovetor e o eixo horizontal indica o valor do autovetor.	50
Figura 13 – Representações equivalentes de três sinais em grafos, em domínios diferentes: domínio dos vértices (a) e domínio espectral (b). Os sinais $\mathbf{f}$ foram definidos no grafo de duas luas. O grafo tem 2000 vértices construído usando vizinhos mais próximos.	51
Figura 14 – Exemplo de um grafo de caminho com 100 vértices.	51
Figura 15 – Exemplo de um grafo do qual se pretende extrair uma amostra.	60

Figura 16 – Frequências de corte para diferentes valores de k . $k = 2, k = 8, k = 16$ e $k = 32$.	63
Figura 17 – Exemplo de amostras extraídas de grafos. Os vértices circundados pelas bolas pretas são os vértices amostrados. (a) representação visual do grafo circular e da amostra $ \mathcal{S}_{opt} = 10$. (b) representação visual do grafo de comunidades e da amostra $ \mathcal{S}_{opt} = 8$. (b) representação visual do grafo de ruas de Minnesota e da amostra $ \mathcal{S}_{opt} = 10$	63
Figura 18 – Exemplo de amostras de tamanho $ \mathcal{S}_{opt} = 10$ retiradas de grafos de sensores aleatórios, mostrando a qualidade visual da amostra.	64
Figura 19 – Gráficos de dispersão do conjunto de dados <i>two moons</i> com 1200 pontos. Em (a) tem-se o conjunto de dados que se pretende classificar, o conjunto tem duas classes, representadas pelas cores azul e roxa. Em (b) tem-se a classificação dos dados pelo algoritmo de propagação de rótulos LLGC . Em (c) a representação do vetor $\mathbf{p}$ no conjunto de dados. Os pontos marcados por x são os que foram inicialmente classificados.	68
Figura 20 – Gráfico de dispersão dos dados sintéticos usados na avaliação numérica do aprendizado ativo semi-supervisionado.	73
Figura 21 – Resultado da comparação da proposta de aprendizado ativo com as amostragens aleatória, e a baseada em <i>proxies</i>	73
Figura 22 – Comparação de técnicas de aprendizado semi-supervisionado ativo para o conjunto de dados MNIST. O eixo vertical indica a porcentagem de dados rotulados por iteração na propagação de rótulos e o eixo horizontal indica a respectiva acurácia obtida por iteração.	74
Figura 23 – Acurácia média da classificação do conjunto de dados MNIST, pelas técnicas de aprendizado ativo semi-supervisionado.	75
Figura 24 – Comparação de técnicas de aprendizado semi-supervisionado ativo para o conjunto de dados 20 news group.	76
Figura 25 – Efeito do parâmetro k na acurácia, para a classificação do algoritmo LLGC. O gráfico mostra a acurácia média de diferentes porcentagens dos dados (MNIST).	77
Figura 26 – Efeito do parâmetro k na acurácia, para a classificação do algoritmo LLGC. O gráfico mostra a acurácia média de diferentes porcentagens dos dados 20 news.	77
Figura 27 – Resumo da performance do aprendizado ativo proposto, para o conjunto de dados de câncer de mama. O eixo horizontal indica a porcentagem que foi escolhida durante determinada iteração, e o eixo vertical indica a acurácia da classificação.	79

LISTA DE ALGORITMOS

Algoritmo – Self-Training	35
Algoritmo 2 – Algoritmo heurístico guloso para estimar $\mathcal{S}_{est}$	60
Algoritmo – Algoritmo de propagação de rótulos de $\mathcal{S}$	66
Algoritmo – Algoritmo ativo para melhorar a da classificação semi-supervisionada	69

LISTA DE TABELAS

Tabela 1 – Estimativa da frequência de corte para uma determinada amostra.	62
Tabela 2 – Descrição dos conjuntos de dados usados	72
Tabela 3 – Resumo dos parâmetros utilizados durante os experimentos numéricos . . .	72

LISTA DE SÍMBOLOS

G — Grafo

$\mathcal{V}$ — Conjunto de vértices

n — Numero de Vértices

v, τ — Vértices do Grafo

E — Conjunto de Arestas

A — Matriz de adjacência

L — Laplaciano do Grafo

$\mathcal{L}$ — Laplaciano Normalizado

D — Matriz de Grau do Grafo

$(.)^T$ — Matriz transposta

$(.)^H$ — Matriz conjugada da transposta

$||.||_2$ — Norma ℓ_2 de um vetor

$\lambda_{max}(.)$ — Máximo autovalor de uma matriz

$\mathcal{O}(.)$ — Notação Big O

u — Autovetor

λ — Autovalor

$\sigma(\mathcal{L})$ — Espetro do laplaciano

f, g, s, ϕ — Sinal

$\hat{f}$ — Transformada de Fourier de f

U — Matriz que as suas colunas são autovetores de L

$|E|$ — Número de arestas do Grafo

$\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$ — Domínio espectral

$\mathcal{S}$ — Subconjunto de vértices (Amostra)

$\omega(s)$ — Limite de banda do sinal s

PW_w — Espaço de Paley-Wiener

$\omega_c(\mathcal{S})$ — Frequência de corte do conjunto

$L_2(\mathcal{S}^c)$ — Espaço de todos sinais nulos em todos os vértices excepto possivelmente em alguns pontos de $\mathcal{S}^c$

Ψ — Operador de amostragem

Φ — Operador de interpolação

SUMÁRIO

1	INTRODUÇÃO	25
1.1	Motivação	28
1.2	Objetivos e resultados esperados	28
1.2.1	<i>Objetivo Geral</i>	28
1.2.2	<i>Objetivos Específicos</i>	28
1.3	Contribuições e organização do texto	29
2	REVISÃO BIBLIOGRÁFICA	31
2.1	Aprendizado supervisionado e não supervisionado	31
2.1.1	<i>Aprendizado não supervisionado</i>	32
2.1.2	<i>Aprendizado supervisionado</i>	32
2.1.2.1	<i>Problemas de Classificação</i>	32
2.1.2.2	<i>Problemas de Regressão</i>	33
2.1.3	<i>Medidas de Distância</i>	33
2.2	Aprendizado semi-supervisionado	33
2.2.1	<i>Suavidade do aprendizado semi-supervisionado</i>	34
2.2.2	<i>A suposição de cluster</i>	34
2.2.3	<i>Aprendizado transdutivo</i>	34
2.2.4	<i>Self-Training</i>	34
2.2.5	<i>Co-Training</i>	35
2.2.6	<i>Máquinas de vetor de suporte transdutivo</i>	35
2.2.7	<i>Modelos de mistura EM</i>	36
2.2.8	<i>Métodos SSL baseados em grafos</i>	36
2.3	Aprendizado Ativo	37
2.3.1	<i>Exemplo de aprendizado ativo</i>	38
2.4	Descrição do problema da amostragem de sinais em grafos e trabalhos relacionados	40
3	TEORIA ESPECTRAL E AMOSTRAGEM EM GRAFOS	45
3.1	Grafos e Sinais em Grafos	45
3.2	Transformada de Fourier em grafos	48
3.3	Filtros, Translações e Convoluções	52
3.4	Amostragem em grafos	54

3.5	Amostragem e interpolação	55
3.5.1	<i>Frequência de corte</i>	56
3.6	Amostra ótima	59
3.7	Experimentos preliminares de amostragem	61
4	ALGORITMO PARA APRENDIZADO SEMI-SUPERVISIONADO	65
4.1	Estrutura do aprendizado semi-supervisionado ativo	65
4.1.1	<i>Algoritmo de propagação de rótulos da amostra $\mathcal{S}$</i>	65
4.1.2	<i>Como definir os pontos de incerteza</i>	67
4.1.3	<i>Modificação da matriz de pesos</i>	68
4.1.4	<i>Como selecionar pontos das regiões de incerteza para amostra</i>	68
4.2	O algoritmo proposto	69
5	RESULTADOS E TRABALHOS FUTUROS	71
5.1	Conjunto sintético	72
5.2	Conjunto de dados MNIST	73
5.3	Classificação de textos	75
5.4	Efeito do parâmetro k do algoritmo de amostragem	76
5.5	Conjunto de dados de câncer Mama (<i>Breast Cancer Data</i>)	78
5.6	Análise da complexidade	78
5.7	Conclusão e Trabalhos Futuros	79
	REFERÊNCIAS	81

INTRODUÇÃO

A inteligência artificial tem influenciado o surgimento de novas áreas desde o seu surgimento. Áreas tais como por exemplo, as redes neurais artificiais, especificamente as redes profundas. Estas técnicas tem se mostrado muito eficientes na resolução de problemas complexos, tais como reconhecimento de discurso ([HINTON; DENG; YU, 2012](#)), reconhecimento de padrões em imagens ([KRIZHEVSKY; SUTSKEVER; HINTON, 2012](#)) e ([LECUN; BENGIO; HINTON, 2015](#)) veículos autônomos ([GOODFELLOW; BENGIO; COURVILLE, 2016](#)). Porém o sucesso destas técnicas de Inteligência artificial (**IA**) vem com um custo, elas requerem uma grande quantidade de dados rotulados para apresentarem uma boa performance. Daí surgiu a necessidade de introduzir algoritmos que utilizem o mínimo de dados rotulados. A classe de tais algoritmos designa-se **Algoritmos semi-supervisionados**.

Os algoritmos semi-supervisionados consideram o problemas de classificação de dados, quando somente um subconjunto dos dados é rotulado. Estes problemas são de interesse vasto devido às suas inúmeras aplicações em vários ramos, incluído pesquisa de imagens ([WEISS; TORRALBA; FERGUS, 2009](#)) e reconhecimento de discurso ([HINTON; DENG; YU, 2012](#); [WEI et al., 2013](#)), onde os dados não rotulados são abundantes mas obter a classe dos dados rotulados é uma tarefa dispendiosa. A questão que foi levada em consideração neste trabalho é: **como escolher o menor conjunto de treinamento de forma que tenhamos a melhor performance possível?** A resposta para essa pergunta é obtida usando ferramentas de processamento de sinais em grafos.

Na última década o processamento de sinais foi estendido para domínios irregulares representados na forma de grafos. Essa extensão é denominada processamento de sinais em grafos (**GSP**). Processamento de sinais em grafos pode ser abordado usando a teoria espectral de grafos ou usando a matriz Laplaciana do grafo.

A abordagem pela teoria espectral de grafos emprega a matriz de adjacência ponderada como elemento base da representação do sinal ([CHEN ROHAN VARMA, ; SANDRYHAILA,](#)

2013). Esta abordagem referida como processamento digital de sinais em grafos ou simplesmente DSP_G , adotada no trabalho de [Chen et al. \(2015\)](#) permite lidar com sinais definidos em grafos direcionais e não-direcionais, com pesos reais ou complexos. No entanto esta abordagem pode ser impraticável, em termos de armazenamento e custo computacional, quando empregada em grafos muito grades, apresentando ainda questões relacionadas à estabilidade numérica.

A segunda abordagem baseia-se na matriz Laplaciana de grafos não direcionados com pesos reais não negativos associados às arestas do grafo para construir uma base para o espaço de sinais ([SHUMAN SUNIL K. NARANG; VANDERGHEYNST, 2013](#)).

Existe um grande número de aplicações que podem beneficiar-se da teoria de processamento de sinais em grafos, tais como a análise de redes sociais ([HAND, 2010](#)), rede de sensores ([ZHU; RABBAT, 2012](#)), processamento de imagens ([NARANG; CHAO; ORTEGA, 2012](#)), sistemas de recomendação ([NARANG; GADDE; ORTEGA, 2013](#)), detecção de comunidades (agrupamento em grafos) ([TREMBLAY; BORGNAT, 2014](#); [DONG et al., 2014](#)), aprendizado semi-supervisionado ([GADDE; ANIS; ORTEGA, 2014](#)), visualização de grafos dinâmicos ([VALDIVIA et al., 2015](#); [COL et al., 2018](#)) e reconstrução do sinal de grafo ([CHEN et al., 2014](#); [WANG; LIU; GU, 2015](#)).

Neste trabalho, estuda-se um problema do processamento clássico de sinais, porém desenvolve-se a teoria em sinais definidos em grafos. O estudo utiliza a abordagem da matriz Laplaciana. Rigorosamente, amostragem em grafos é o estudo da seleção de nós em grafos, selecionados de tal forma que o sinal original pode ser reconstruído com alta precisão. O estudo da amostragem de sinais em grafos fornece as condições em que um sinal tem a reconstrução única e estável, dada uma amostra. Amostragem de sinais tem inúmeras aplicações em vários campos tais como na atribuição de rótulos em exames médicos e na reconstrução de dados de sensores ([DANZIGER et al., 2009](#); [SHEN et al., 2015](#)).

Considere um exemplo motivador para o estudo da amostragem de sinais definidos em grafos: a previsão de opinião em redes sociais, figura 1. Considere um grafo que representa as relações de conectividade de usuários do *Instagram*, sendo que cada usuário é representado por um vértice. A relação entre os usuários é representado por uma aresta. Modela-se como sinal do grafo a opinião que cada usuário tem em relação a algum assunto, opinião positiva representada por $+1$ e a opinião negativa por -1 . Podemos questionar uma fração (amostra) dos usuários do *Instagram* sobre a sua opinião e posteriormente recuperar (classificar) a opinião do restante dos usuários. Aplicações como estas são importantes para situações em que o acesso aos rótulos é dispendioso.

Apesar do problema da amostragem de sinais em grafos ter sido muito estudado na última década, tal tarefa continua apresentando vários desafios, pois no domínio de grafos não temos a possibilidade de amostragem regular (extrair uma amostra a cada tantos vértices), enquanto que na amostragem de sinais no sentido clássico temos a noção de amostragem regular. Contudo várias técnicas de amostragem de sinais em grafos foram propostas para sinais de banda

Figura 1 – Exemplo de um grafo de uma rede social

limitada. Porém na prática essa condição nunca pode ser garantida, pois os sinais em estudo estão sujeitos a ruídos e portanto estes sinais não serão de banda limitada ou qualquer outra classe pré-definida. Na tentativa de resolver este problema, foram propostos alguns operadores de amostragem que identificam vértices que minimizam uma certa medida de erro de reconstrução (GADDE; ORTEGA, 2015; CHEN *et al.*, 2015; ANIS; GADDE; ORTEGA, 2016). Outro problema desafiador da amostragem é a complexidade, pois algumas técnicas de amostragem requerem o cálculo dos primeiros K autovetores da transformada de Fourier (GFT) (CHEN *et al.*, 2015), o que pode ser impraticável para grafos muito grandes. (ANIS; GADDE; ORTEGA, 2016) usa frequências aproximadas (*proxies*) ao invés de frequências exatas, reduzindo a complexidade.

A redução da complexidade do problema de amostragem pode resultar em várias aplicações interessantes visto que a escolha da melhor amostra de um grafo G é equivalente a escolher os vértices mais significativos do grafo, ou seja os vértices definidos a partir dos pontos que mais dimensões capturam no conjunto de dados. Os métodos de aprendizado semi-supervisionado (ZHOU *et al.*, 2004; ZHANG *et al.*, 2011; GU; HAN, 2012) são muito beneficiados com a teoria de amostragem em grafos, pois esses métodos no estado da arte em geral não dão as condições nas quais um sinal pode ser única e perfeitamente interpolado de suas amostras. A amostragem de sinais em grafos tem também possíveis aplicações na redução de dimensionalidade de um conjunto de dados (BELKIN; NIYOGI, 2004).

Nesta dissertação, considera-se o problema do aprendizado semi-supervisionado ativo para o caso onde têm-se um oráculo (ou usuário). O objetivo do oráculo é fornecer uma supervisão iterativa de uma amostra de um conjunto de dados a um algoritmo de propagação de rótulos. O objetivo do algoritmo é atingir a melhor classificação dos dados em poucas iterações. Para atingir a melhor classificação possível, define-se as regiões de certeza e incerteza da classificação do conjunto de dados e classifica-se as amostras que estão na região de incerteza. Propõe-se um operador que aumente a chance dos vértices da região de classificação incerta serem escolhidos durante o processo de amostragem iterativa.

Para tal, estuda-se inicialmente a teoria de processamento de sinais em grafos, utilizada como suporte para a teoria de amostragem apresentada. Estuda-se também os algoritmos de aprendizado semi-supervisionado. A proposta apresentada consiste em escolher a amostra pelo

algoritmo de (ANIS; GADDE; ORTEGA, 2016), considerando dados da região de incerteza e classificar todos os dados pelo algoritmo LLGC (ZHOU *et al.*, 2004), em cada iteração.

Para o entendimento dos principais conceitos da **GSP** pode-se usar as ferramentas de *Graph Signal Processing Toolbox (GSP-BOX)*, que está disponível nas linguagens de programação MATLAB e Python (PERRAUDIN *et al.*, 2014). A ferramenta implementa, grafos, operações sobre grafos, filtros, Polinômios de Chebyshev para acelerar a operação de filtros e outras funcionalidades.

1.1 Motivação

Como mencionado o aprendizado supervisionado trabalha com conjuntos de dados cujos exemplos estão pré-classificados, i.e., estão rotulados com atributo classe. O objetivo do aprendizado supervisionado é construir um modelo de predição que consiste em rotular novos exemplos que não possuem o atributo classe associado. Ou seja a partir de exemplos rotulados, denominados exemplos de treinamento, um algoritmo de aprendizado é usado para induzir padrões presentes em cada classe, gerando uma descrição de cada classe, essas descrições são usadas para prever a classe de novos exemplos.

Em muitos problemas reais geralmente o número de exemplos rotulados é muito pequeno, e classificadores induzidos a partir de um número pequeno de exemplos rotulados, apresentam geralmente baixa precisão. Tal fator inviabiliza o uso dos algoritmos supervisionados. Uma alternativa seria rotular manualmente cada exemplo. Porém este processo é extremamente caro ou impossível, dada a imensa quantidade de dados. Todavia torna-se interessante a existência de mecanismos que possam: escolher automaticamente os exemplos a serem rotulados, rotular os exemplos automaticamente, com o objetivo de viabilizar o uso do aprendizado supervisionado nas situações em que o número de exemplos rotulados é pequeno e o número de exemplos não rotulados é grande. Essa é uma das metas do aprendizado semi-supervisionado ativo, objeto de estudo desta dissertação.

1.2 Objetivos e resultados esperados

1.2.1 *Objetivo Geral*

Estudar a teoria de amostragem em grafos e a relação com o aprendizado de máquina.

1.2.2 *Objetivos Específicos*

- Aplicar a teoria de amostragem proposta para problemas de aprendizado semi-supervisionado ativo;

- Mostrar que as técnicas de amostragem em grafos no domínio espectral melhoram a performance dos algoritmos de aprendizado semi-supervisionado;
- Propor um algoritmo semi-supervisionado ativo.

1.3 Contribuições e organização do texto

Esta dissertação está organizada em 5 capítulos, divididos em três áreas a saber: Processamento de sinais em grafos e amostragem de sinais em grafos, aprendizado de máquina semi-supervisionado ativo, e a terceira parte que é composta pela proposta de aprendizado ativo semi-supervisionado, os experimentos numéricos, a conclusão e trabalhos futuros.

A proposta de aprendizado semi-supervisionado ativo consiste em: primeiramente escolher uma amostra inicial do conjunto de dados usando o algoritmo proposto no trabalho (ANIS; GADDE; ORTEGA, 2016) e supervisionar (atribuir rótulos) esta amostra. Em seguida o usuário recorre a um algoritmo de propagação de rótulos (ZHOU *et al.*, 2004) para propagar as amostras supervisionadas. Este processo deve continuar até que os recursos de supervisão sejam esgotados. Porém propõe-se que a partir da segunda iteração a matriz de pesos do conjunto de dados seja modificada para que o algoritmo de amostragem (ANIS; GADDE; ORTEGA, 2016) considere amostras de classificação incerta na iteração anterior.

- **Capítulo 1 (Introdução):** Este capítulo introduz a dissertação, vai desde o processamento de sinal clássico, processamento de sinais definidos em grafos e aplicações até o caso específico que é o estudo da amostragem em grafos.
- **Capítulo 2 (Revisão bibliográfica):** Neste capítulo revisa-se a literatura focando no Aprendizado de máquina, teoria de amostragem em grafos, aplicações da amostragem no aprendizado semi-supervisionado. Mostra-se algumas propostas anteriormente apresentadas sobre o aprendizado semi-supervisionado ativo.
- **Capítulo 3 (Teoria espectral e amostragem em grafos):** Descreve-se as ferramentas teóricas e computacionais da teoria espectral de grafos. Descreve-se detalhadamente as ferramentas mais importantes para o estudo do processamento de sinais em grafos, juntamente com as definições básicas e a notação usada ao longo desta dissertação. Apresenta-se a técnica de amostragem proposta nos trabalhos (PESENSEN, 2008; ANIS; GADDE; ORTEGA, 2014; ANIS; GADDE; ORTEGA, 2016), que serve como ponto de partida para os desenvolvimentos da dissertação. Apresenta-se também o estado da arte das técnicas de amostragem, usando a abordagem espectral.
- **Capítulo 4 (Proposta):** Neste Capítulo apresenta-se detalhadamente a proposta de aprendizado semi-supervisionado ativo. Na figura 2 apresenta-se o diagrama da proposta.

Figura 2 – Diagrama de aprendizado semi-supervisionado considerado neste trabalho.

Fonte: Elaborada pelo autor.

- **Capítulo 5 (Resultados e trabalhos futuros):** Neste Capítulo são apresentados os resultados obtidos nesta dissertação, as conclusões da pesquisa, resume-se as principais contribuições do trabalho, apresenta-se as direções da pesquisa que seriam interessantes seguir para trabalhos futuros.

REVISÃO BIBLIOGRÁFICA

No uso dos algoritmos de aprendizado de máquina para as tarefas de classificação, admite-se a existência de um conjunto de exemplos rotulados conhecido como conjunto de treinamento. Este conjunto é utilizado para treinar um classificador. Porém em vários casos de aplicação real o conjunto de treinamento pode não ser representativo ou suficiente para treinar um bom classificador. O problema da insuficiência do conjunto de treinamento é contornado mediante o uso de algoritmos semi-supervisionados (BASU; BANERJEE; MOONEY, 2004; LI *et al.*, 2012). O problema da representatividade do conjunto de treinamento pode ser contornado usando algoritmos de amostragem (HASAN; ZAKI, 2009; STUTZBACH *et al.*, 2006; SHOMORONY; AVESTIMEHR, 2014; ANIS; GADDE; ORTEGA, 2016).

Neste capítulo é apresentada uma visão geral do aprendizado de máquina, dos algoritmos de amostragem pelo domínio espectral e alguns resultados relativos à aplicação da amostragem no aprendizado semi-supervisionado.

2.1 Aprendizado supervisionado e não supervisionado

O Aprendizado de Máquina (AM) é um campo da inteligência artificial (IA) que propõe algoritmos para resolver problemas automaticamente, sem serem explicitamente programados para tal. Estes algoritmos têm características e premissas que os permitem ser aplicados em diversas áreas de conhecimento, tornando um campo interdisciplinar (MITCHELL, 2004). Devido à natureza dinâmica dos problemas, e à disponibilidade de recursos, os algoritmos de AM são tradicionalmente divididos em dois tipos: aprendizado supervisionado e aprendizado não supervisionado (LUXBURG; SCHOLKOPF, 2008). O esquema da figura 3 mostra os diferentes tipos de aprendizado de máquina.

Figura 3 – Métodos de aprendizado de máquina.

Fonte: Elaborada pelo autor.

2.1.1 *Aprendizado não supervisionado*

Seja $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, um conjunto de pontos. Estes pontos são considerados **i.i.d** (Independentes e identicamente distribuídos) de uma distribuição conjunta em $\mathcal{P}$ com probabilidade $P(\mathbf{x})$, onde $\mathbf{x} \in \mathcal{P}$. Define-se frequentemente uma matriz $n \times d$, $\mathbf{X} = (\mathbf{x}_i^T)_{i \in [n]}^T$ com $[n] := \{1, 2, \dots, n\}$, que contém os pontos do conjunto de dados como suas linhas. O aprendizado não supervisionado tem o objetivo de encontrar estruturas interessantes do conjunto de dados. As formas comuns de aprendizado não supervisionado são: estimação de quantil, clusterização, detecção de *outliers* e a redução de dimensionalidade.

2.1.2 *Aprendizado supervisionado*

No aprendizado supervisionado, os algoritmos são fornecidos um conjunto de dados rotulados, a premissa deste aprendizado é que existe uma relação entre os dados e seus rótulos. O objetivo é aprender uma função que relaciona os dados de entrada e seus rótulos, após um treinamento suficiente. Os problemas de aprendizado supervisionado são particionados em classificação e regressão.

2.1.2.1 *Problemas de Classificação*

Nos problemas de classificação treina-se um algoritmo (BISHOP, 2006; PANG; LEE; VAITHYANATHAN, 2002; TRAFALIS; GILBERT, 2006) para que preveja valores discretos. Identifica-se os dados de entrada como membros de um grupo ou classe. Em um problema de classificação de imagens de gato ou cão por exemplo, cada imagem é pré-rotulada como gato ou cão. E o algoritmo é avaliado pela precisão com que pode classificar novas imagens de gato ou cão.

2.1.2.2 Problemas de Regressão

Nos problemas de regressão os algoritmos (BENNETT; CAMPBELL, 2000; BISHOP, 2006) são treinados para prever valores contínuos. Ou seja os algoritmos devem estimar o mapeamento $f: \mathcal{X} \rightarrow \mathcal{Y}$, das variáveis de entrada $\mathcal{X}$, para as saídas reais y . Neste caso, como y é uma saída real, os algoritmos de regressão geralmente preveem quantidades ou tamanhos. Exemplo de um problema de regressão: dado um conjunto de dados sobre casas, prever o preço delas. Visto que a saída (preço de casa) é um valor real então este é um problema de regressão.

2.1.3 Medidas de Distância

Um dos tópicos mais importantes no momento de se realizar uma análise sobre um conjunto de dados é em relação à medida de distância adotada.

Considerando os atributos dos exemplos como dimensões de um espaço de alta dimensão, a descrição de cada exemplo corresponde a um ponto no espaço. Assim, pode-se definir a **similaridade** entre dois exemplos como sendo a distância entre eles. Isto é, caso dois exemplos sejam similares, então a distância entre eles é pequena. Conforme a similaridade diminui, a distância aumenta.

Na obra (WITTEN *et al.*, 2016) detalham várias formas de calcular a distância entre dois pontos. Exemplos de medidas de distancia são: a medida de distância Minkowsky, a medida de Manhattan, a distância Euclideana, a distância do cosseno e a distância de Chebychev. Contudo, nesta dissertação considera-se a distância do cosseno para problemas envolvendo a classificação de textos e a distância Euclidiana para o restante dos problemas considerados.

2.2 Aprendizado semi-supervisionado

O aprendizado semi-supervisionado é uma combinação do aprendizado supervisionado e não supervisionado. Ou seja os algoritmos semi-supervisionados usam dados rotulados e não rotulados. De forma geral os algoritmos de aprendizado semi-supervisionado apresentam maneiras de aprender a estrutura dos dados através dos exemplos não rotulados.

Os algoritmos de classificação semi-supervisionada são mais utilizados nas situações em que os dados rotulados não são representativos o suficiente para usar um classificador indutivo. No trabalho (FACELI *et al.*, 2011) mostra-se a importância destes algoritmos para problemas de *overfitting* e *underfitting*.

Neste caso o conjunto de dados $\mathcal{X}$ é transformado em duas matrizes $\mathbf{X}_\ell = (\mathbf{x}_1, \dots, \mathbf{x}_\ell)$ e $\mathbf{X}_u = (\mathbf{x}_{\ell+1}, \dots, \mathbf{x}_{\ell+u})$ com rótulos disponíveis $\mathbf{Y}_\ell = (y_1, \dots, y_\ell)$ e com rótulos desconhecidos. Neste trabalho considera-se o problema de aprendizado semi-supervisionado **transdutivo**. O SSL transdutivo foi introduzido por Vapnik Bennett e Demiriz (1999). No SSL transdutivo o objetivo é aprender uma função que classifique somente os dados de teste (não rotulados), ou

seja, encontrar uma função $f : \mathbf{X}_u \rightarrow \mathcal{Y}$. E no aprendizado **indutivo** o objetivo é aprender uma função para a predição em todo o espaço $\mathcal{P}$, ou seja, encontrar uma função $f : \mathcal{P} \rightarrow \mathcal{Y}$, que será usada para classificar cada $\mathbf{x}_i$ do conjunto de teste.

Apesar do aprendizado semi-supervisionado ter se mostrado muito bom para a resolução de certos problemas, resultados de testes mostram que este pode ser ruim na resolução de outros problemas. Para que o aprendizado semi-supervisionado tenha sucesso é necessário impor condições sobre a distribuição e topologia dos dados, ou seja o conhecimento de $P(\mathbf{x})$ dos dados não rotulados deve conter informações importantes para a inferência $P(y|\mathbf{x})$. Se estas condições não forem satisfeitas o **SSL** não será melhor que **SL**.

2.2.1 Suavidade do aprendizado semi-supervisionado

Hipótese: se duas amostras $\mathbf{x}_i$ e $\mathbf{x}_j$ estão próximas, então os rótulos correspondentes f_i e f_j são próximos.

2.2.2 A suposição de cluster

As primeiras técnicas de aprendizado semi-supervisionado supunham que pontos na mesma classe tendem a formar um *cluster*. Com esta suposição pode-se usar um algoritmo de clusterização usando informações dos dados rotulados e não rotulados, e finalmente rotular os pontos de cada cluster formado usando os dados rotulados.

Suposição de cluster: Se pontos estão no mesmo *cluster* então são provavelmente da mesma classe.

2.2.3 Aprendizado transdutivo

No processo de aprendizado indutivo tem-se como entrada $(\mathbf{X}_\ell, \mathbf{Y}_\ell)$ e $\mathbf{X}_u$, treina-se uma função $f : \mathbb{R}^d \rightarrow L$, onde $L = \{1, 2, \dots, c\}$ para um problema com c classes. Os parâmetros de f são aprendidos a partir dos conjuntos $\mathbf{X}_\ell \cup \mathbf{X}_u$ e $\mathbf{Y}_\ell$, com o objetivo de efetuar a predição de $\mathbf{X}_u$, respeitando os rótulos previamente conhecidos (YAROWSKY, 1995), (ZHANG *et al.*, 2011).

2.2.4 Self-Training

Esta é uma adaptação semi-supervisionada de um algoritmo supervisionado. A ideia geral da técnica é treinar um classificador supervisionado f_{st} com os dados rotulados disponíveis $\mathbf{X}_\ell$. Depois do treino é escolhido um subconjunto $\mathbf{X}_s \subset \mathbf{X}_u$ e é classificado usando f_{st} , e em seguida f_{st} é treinado usando o conjunto rotulado $\mathbf{X}_\ell \cup \mathbf{X}_s$. Escolhe-se um novo conjunto não rotulado de $\mathbf{X}_u - \mathbf{X}_s$ que é classificado usando o novo f_{st} . Este processo é repetido até que não haja mais dados sem rótulos (BASU; BANERJEE; MOONEY, 2004).

Figura 4 – Métodos de aprendizado semi-supervisionado.

Fonte: Elaborada pelo autor.

Algoritmo. – Self-Training

```

1: repeat
2:    $f_{st} \leftarrow \text{treinamodelo}(\mathbf{X}_\ell)$ 
3:   for  $\mathbf{x} \in \mathbf{X}_u$  do
4:     if  $\max[f_{st}(\mathbf{x})] > \varepsilon$  (limiar) then
5:        $\mathbf{X}_\ell = \mathbf{X}_\ell \cup \{(\mathbf{x}, p(\mathbf{x}))\}$ 
6: until  $\mathbf{X}_u = \emptyset$ 
  
```

2.2.5 Co-Training

Co-Training (BLUM; MITCHELL, 1998) é um método clássico de treinamento que faz suposições fortes sobre os dados. Este método requer que o conjunto $\mathbf{X}_\ell$ possa ser representado usando dois conjuntos independentes $\mathbf{X}_\ell^1$ e $\mathbf{X}_\ell^2$ e que cada conjunto seja suficientemente bom para treinar um bom classificador.

Teoricamente *Co-training* se baseia na cooperação de dois algoritmos supervisionados. A ideia geral consiste em um classificador rotular exemplos para outro classificador, e vice-versa. Mais detalhes sobre esta estratégia podem ser encontrados no trabalho (BLUM; MITCHELL, 1998)

2.2.6 Máquinas de vetor de suporte transdutivo

Estas técnicas são a variação semi-supervisionada dos algoritmos supervisionados de máquinas de vetor de suporte. Comumente designados como Máquinas de vetor de suporte

semi-supervisionadas (**S3VMs**). **S3VMs** usam dados rotulados e não rotulados para maximizar a margem geométrica não apenas usando os dados rotulados $\mathbf{X}_\ell$ mas também com o uso dos dados não rotulados $\mathbf{X}_u$. Os **S3VMs** encontram um hiperplano que melhor separa as classes $\mathbf{X}_\ell$ e ao mesmo tempo melhor divide $\mathbf{X}_u$, que não é necessariamente o mesmo da versão supervisionada (CHAPELLE; SINDHWANI; KEERTHI, 2008).

2.2.7 Modelos de mistura EM

Muitas vezes ao resolver um problema fazemos suposições sobre os dados, como por exemplo assumir que cada observação segue a distribuição normal. E prosseguimos com a resolução desse problema, estimando parâmetros tais como a média e variância. No entanto em muitos casos supor que todas as observações seguem a mesma distribuição de probabilidades pode ser uma suposição sem fundamento prático, pois os dados podem ser complexos. Por exemplo os dados podem ser multimodais contendo muitas regiões com alta probabilidade. Para contornar essas dificuldades estuda-se os modelos de mistura.

Um modelo de mistura é um modelo probabilístico projetado para considerar a presença de subpopulações dentro de uma população total, sem impor que um conjunto de dados observados identifique a subpopulação à qual pertence uma observação individual.

Modelos de mistura EM são uma classe de técnicas de aprendizado semi-supervisionado. Estas assumem um modelo generativo $p(\mathbf{x}, y) = p(y)p(\mathbf{x}|y)$, onde $\mathbf{x}$ é um ponto, y o seu rótulo e $p(\mathbf{x}, y)$ é uma distribuição mista de probabilidade inidentificável. Com uma grande quantidade de dados não rotulados fica fácil de identificar as componentes da distribuição mista (COZMAN; COHEN; CIRELO, 2003).

A suposição dos modelos de mistura semi-supervisionada é que os dados provêm da distribuição do modelo de mistura, onde as probabilidades $p(y)$, a probabilidade condicional $p(\mathbf{x}|y)$ e o número de componentes são reais.

2.2.8 Métodos SSL baseados em grafos

A literatura apresenta várias técnicas de aprendizado semi-supervisionado (UEFFING; HAFFARI; SARKAR, 2007; ZHOU; BORÉN; AKYÜREK, 2007). Neste trabalho propõe-se um esquema de aprendizado semi-supervisionado que utiliza uma técnica baseada em grafos. Portanto a seguir apresenta-se a ideia geral dos métodos baseados em grafos.

Os métodos de **SSL** baseados em grafos definem um grafo considerando os vértices como o conjunto de dados $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, e as arestas representando a similaridade entre os pontos. Estes métodos frequentemente assumem a suavidade dos rótulos sobre o grafo e são de natureza transdutiva. A maioria dos métodos **SSL** baseado em grafos adotam o uso do Laplaciano do grafo.

Seja $G = (\mathcal{E}, \mathcal{V})$ um grafo com pesos de arestas dados pela função $w : \mathcal{E} \rightarrow \mathbb{R}$. O peso $w(e)$ de uma aresta e corresponde à similaridade dos nós incidentes. A matriz de pesos $\mathbf{W}$ do grafo G é definida como:

$$\mathbf{W}_{ij} = \begin{cases} w(e), & \text{se } e = (i, j) \in \mathcal{E} \\ 0, & \text{se } e = (i, j) \notin \mathcal{E}. \end{cases} \quad (2.1)$$

Define-se a matriz diagonal $\mathbf{D}_{ii}$, como $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$, para definir a matriz Laplaciana:

- O Laplaciano não normalizado $\mathbf{L} = \mathbf{D} - \mathbf{W}$
- O Laplaciano normalizado do Grafo $\mathcal{L} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}$

A matriz de pesos $\mathbf{W}$ pode ser definida também como a matriz dos K -vizinhos mais próximos: $\mathbf{W}_{ij} = 1$ se e somente se $\mathbf{x}_i$ esta entre os vizinhos mais próximos de $\mathbf{x}_j$ e 0 caso contrário. Outra matriz de similaridade típica é a matriz do núcleo Gaussiano:

$$\mathbf{W}_{ij} = \begin{cases} \exp\left(-\frac{\text{dist}(\mathbf{x}_i, \mathbf{x}_j)^2}{2\sigma^2}\right), & \text{se } i \neq j; \\ 0, & \text{se } i = j. \end{cases} \quad (2.2)$$

Para o desenvolvimento dos experimentos deste trabalho foi considerada a matriz do núcleo Gaussiano e a matriz $\mathcal{W} = \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}$ na propagação de rótulos, usando um algoritmo simples e eficiente baseado em grafos, para a propagação de rótulos **LLGC**, proposto no trabalho (ZHOU *et al.*, 2004). O algoritmo **LLGC** constrói uma função suficientemente suave, com respeito à estrutura intrínseca dos dados rotulados e não rotulados.

A principal característica do **LLGC** é deixar todos os pontos distribuir iterativamente as informações de seus rótulos para seus vizinhos até que se atinja o estado de convergência. Este algoritmo é apresentado com mais detalhes no capítulo 4

2.3 Aprendizado Ativo

A hipótese central do aprendizado ativo é de que um algoritmo de aprendizado de máquina pode obter um nível de sucesso maior se for permitido a escolha de alguns (poucos) exemplos de treinamento no conjunto que se pretende aprender.

O aprendizado ativo é de interesse do aprendizado de máquina, principalmente para casos em que os dados rotulados são caros e os dados não rotulados são abundantes. Em alguns casos os dados rotulados são obtidos de forma simples, tal como o exemplo de *ranking* dos filmes dados pelos usuários da *Nextfix*. Porém em outros casos os rótulos são de difícil acesso. Alguns

dos exemplos incluem reconhecimento de discurso (ZHOU; CHEN; JIANG, 2004), extração de informação (SETTLES; CRAVEN; FRIEDLAND, 2008), classificação de imagens médicas, etc.

Desta forma o aprendizado ativo tem como objetivo melhorar a acurácia usando o mínimo de dados rotulados possível.

2.3.1 Exemplo de aprendizado ativo

Figura 5 – Ciclo do aprendizado ativo

Adaptada de: Settles (2009)

A figura 5 ilustra um aprendizado cíclico em que um ser humano tem a prerrogativa de escolher os dados de treinamento, o algoritmo começa com uma quantidade pequena de dados rotulados L , e requisita mais dados rotulados, escolhidos usando um determinado critério. Uma vez escolhidos os dados de treinamento, estes são adicionados a L e o algoritmo supervisionado ou semi-supervisionado é novamente aplicado.

A diferença entre os algoritmos semi-supervisionados e ativos é que os algoritmos ativos assumem que uma supervisão externa não está disponível, mas é construída com o auxílio de um oráculo (usuário). Para (SETTLES, 2009) a diferença está no fato de que nos algoritmos de aprendizado semi-supervisionado procura-se estabelecer uma supervisão externa partindo do que se supõe saber sobre o problema enquanto que no aprendizado ativo a supervisão externa é estabelecida buscando explorar os aspectos desconhecidos e promissores do problema. Resumindo: os algoritmos ativos buscam minimizar as dificuldades que surgem durante o ato de prover as supervisões externas. Para tal, estabelecem como resultado um conjunto mínimo $\mathbf{X}_\ell$.

Existem várias estratégias de aprendizado ativo na literatura (DASGUPTA, 2009; LI *et al.*, 2014). Estas estratégias são divididas em três categorias: Amostragem informativa (HASSENJÄGER; RITTER, 1996), amostragem representativa (BASU; BANERJEE; MOONEY, 2004; MALLAPRAGADA; JIN; JAIN, 2008; ZHANG *et al.*, 2011) e amostragem baseada em agrupamento (XU *et al.*, 2003; HU; NAMEE; DELANY, 2010).

A amostragem informativa está relacionada com a busca de observações explorando suas incertezas ou potencial de influência na solução do problema. A amostragem representativa está relacionada com a exploração da informação que se mostra representativa no problema. A amostragem baseada em agrupamento é relacionada ao pré-processamento dos dados.

Quando um algoritmo ativo é combinado com um algoritmo semi-supervisionado, denomina-se **algoritmo semi-supervisionado ativo**.

Figura 6 – Curvas de aprendizado da classificação de textos: *baseball vs hockey*. O eixo horizontal indica o número de instâncias consideradas para o treino e o eixo vertical indica a acurácia obtida. Cada curva indica uma estratégia de seleção. É notável que a estratégia usando o aprendizado ativo obtém uma classificação melhor que a amostragem aleatória.

Fonte: Settles (2009)

No trabalho (SETTLES, 2009), mostra-se por meio de um experimento simples que a escolha dos dados de treinamento por um algoritmo ativo é mais vantajoso que a escolha aleatória.

Os resultados da figura 6 foram obtidos usando o conjunto de textos *20 Newsgroups* restrito às classes *baseball* e *hockey*, gerando 2000 documentos. O modelo de classificação considerado foi a regressão logística, utilizando a estratégia de validação cruzada ($k = 10$). O experimento mostra que os algoritmos de aprendizado ativo melhoram a performance de um classificador. Neste trabalho propõe-se uma técnica de aprendizado semi-supervisionado ativo, baseado na estratégia de amostragem desenvolvida nos trabalhos (GADDE; ANIS; ORTEGA, 2014; ANIS; GADDE; ORTEGA, 2016) e o algoritmo de propagação de rótulos LLGC. Uma

descrição da proposta é apresentada no capítulo 4.

2.4 Descrição do problema da amostragem de sinais em grafos e trabalhos relacionados

As técnicas para a resolução de problemas envolvendo grafos podem se tornar ineficientes se os dados resultantes desses problemas forem massivos e de alta dimensão, (HAND, 2010; ZHU; RABAT, 2012; NARANG; CHAO; ORTEGA, 2012; NARANG; GADDE; ORTEGA, 2013; GADDE; ANIS; ORTEGA, 2014; SANDRYHAILA; MOURA, 2014; CHEN *et al.*, 2015). Todavia a redução de dimensionalidade pode tornar a mineração destes dados efetiva. Desta forma introduzimos o seguinte problema.

Considere o problema da extração de um conjunto $\mathcal{S} = \{\mathcal{S}_0, \dots, \mathcal{S}_{m-1}\}$ de coeficientes do sinal $\mathbf{x} \in \mathbb{R}^n$ para produzir um sinal amostrado $\mathbf{x}_{\mathcal{S}} \in \mathbb{R}^m$ ($n > m$), onde $\mathcal{S}$ denota a sequência de coeficientes amostrados e então interpolar $\mathbf{x}_{\mathcal{S}}$ para obter $\mathbf{x}' \in \mathbb{R}^n$. O operador de amostragem é um mapeamento linear de $\mathbb{R}^n$ para $\mathbb{R}^m$, definido como:

$$\Psi_{ij} = \begin{cases} 1, & \text{se } j = \mathcal{S}_i; \\ 0, & \text{caso contrário} \end{cases} \quad (2.3)$$

e o operador de interpolação Φ é um mapeamento linear de $\mathbb{R}^m$ para $\mathbb{R}^n$. A amostragem é efetuada pela operação $\mathbf{x}_{\mathcal{S}} = \Psi\mathbf{x} \in \mathbb{R}^m$ e a interpolação pela operação $\mathbf{x}' = \Phi\mathbf{x}_{\mathcal{S}} = \Phi\Psi\mathbf{x} \in \mathbb{R}^n$, onde $\mathbf{x}' \in \mathbb{R}^n$ recupera $\mathbf{x} \in \mathbb{R}^n$ exata ou aproximadamente.

Figura 7 – Amostragem seguida de interpolação.

Fonte: Chen *et al.* (2015).

É interessante saber quando $\mathbf{x}' \in \mathbb{R}^n$ recupera $\mathbf{x} \in \mathbb{R}^n$ exata ou aproximadamente. Tentamos responder estas questões estabelecendo a relação com a teoria de amostragem clássica.

Os métodos de amostragem em grafos adotados atualmente na literatura podem ser divididos em dois tipos: A abordagem pelo domínio de vértices e a abordagem pelo domínio espectral.

As **abordagens pelo domínio de vértices** são aquelas abordagens usadas para extrair amostras de vértices sem considerar informações espectrais do grafo. Narang e Ortega (2010) e

Nguyen e Do (2015) selecionam amostras (conjunto de vértices) baseando-se em cortes máximos e árvores de dispersão, respectivamente. Os métodos baseados no domínio de vértices não consideram o problema da otimização da amostra em termos de qualidade da reconstrução. Contudo, os trabalhos de Narang e Ortega (2012) e Narang e Ortega (2013) mostram que estes métodos são adequados para construir operadores de decimação (NGUYEN; DO, 2014).

A maioria das **abordagens no domínio espectral** assumem que uma porção da base de Fourier é conhecida (SHUMAN SUNIL K. NARANG; VANDERGHEYNST, 2013). As técnicas de amostragem espectrais baseiam-se no teorema clássico de Nyquist-Shannon, que de todas as funções ou sinais do espaço de Paley-Wiener de largura de banda fixa no espaço Euclidiano, é possível extrair um conjunto de amostras que podem ser usadas para representar as funções relevantes. Especificamente, dado um sinal de largura de banda f e uma taxa de amostragem f_s , o teorema de amostragem de Nyquist Shannon impõe uma condição para a recuperação única do sinal de suas amostras exigindo que $f < f_s/2$. Este teorema estabelece um limitante superior da largura de banda de sinais que podem ser unicamente reconstituídos numa dada taxa de amostragem.

A teoria de amostragem de sinais em grafos tenta generalizar o teorema de amostragem clássico de Nyquist-Shannon. A teoria de amostragem em grafos foi primeiramente apresentada no trabalho (PESENSEN, 2008), onde foi detalhada uma condição suficiente para a reconstrução única de um sinal.

Em trabalhos anteriores tentou-se formular um teorema de amostragem para o domínio de grafos, respondendo às seguintes questões: dado um conjunto de vértices, qual é a maior largura de banda (frequência de corte) que um sinal em grafo pode ter para que este possa ser reconstruído? E a questão inversa, qual é a menor amostra possível que permite a reconstrução correta de todos os sinais dada uma largura de banda?

No trabalho (NARANG; ORTEGA, 2011) foi estabelecida a frequência de corte para grafos bipartidos. Enquanto que no trabalho de Pesenson (2008) foram dadas condições suficientes para a reconstrução única de um sinal definido em um grafo arbitrário e usada para derivar um limite inferior da frequência de corte.

Anis, Gadde e Ortega (2014) fazem um progresso muito significativo, estabelecendo uma teoria de amostragem de sinais definidos em grafos. Apresentam condições necessárias e suficientes para que uma amostra reconstrua um sinal perfeitamente, dada uma frequência de corte. Estas condições necessárias e suficientes são usadas para obter uma sequência crescente de frequências de corte que aproximam da frequência verdadeira. Com esta aproximação das frequências de corte os autores propõem um algoritmo guloso para estimar a amostra ótima de um sinal. Porém não fica claro se de fato esse método fornece aproximações arbitrariamente próximas da frequência de corte. E se de fato essa sequência se aproxima arbitrariamente da frequência de corte, até que ponto ela precisa se aproximar para obter uma boa amostra. Em trabalhos anteriores (AVRON; BOUTSIDIS, 2013; BOUTSIDIS; DRINEAS; MAGDON-ISMAIL, 2014),

resolveram um problema de otimização similar para a aproximação de matrizes e mostraram que o algoritmo guloso fornece uma boa aproximação da solução.

Shomorony e Avestimehr (2014) mostram que as condições dadas no trabalho (ANIS; GADDE; ORTEGA, 2014) podem ser usadas para calcular a frequência de corte exata. E que com a frequência de corte exata pode-se obter a amostra ótima. Com esta abordagem é possível extrair uma amostra muito boa de um conjunto de dados. Veja na figura 8 a ilustração de uma amostra extraída usando o algoritmo proposto no trabalho Shomorony e Avestimehr (2014).

Figura 8 – (a) Amostra ótima $|\mathcal{S}| = 25$ (vértices de cor vermelha) de um grafo com 200 vértices. (b) Amostra ótima $|\mathcal{S}| = 20$ (vértices de cor vermelha) de um grafo circular com 200 vértices e arestas adicionais.

Fonte: Shomorony e Avestimehr (2014).

Chen *et al.* (2015) propõem um algoritmo de amostragem de sinais definidos em grafos arbitrários, mostram que a reconstrução de um sinal de banda limitada sobre a transformada de Fourier é possível. Propõem um operador de amostragem obtido experimentalmente que garante a reconstrução perfeita e robusta a ruídos. Porém os algoritmos de amostragem propostos nos trabalhos de Chen *et al.* (2015), Tsitsvero, Barbarossa e Lorenzo (2016), Chamon e Ribeiro (2018) requerem a decomposição em valores singulares (SVD) em cada passo, o que torna estas técnicas computacionalmente caras.

Anis et al. (ANIS; GADDE; ORTEGA, 2016) apresentam uma técnica de amostragem baseada em quantidades denominadas *proxies*. Esta técnica é mais consistente e com um custo computacional menor em relação às outras técnicas, pois ela não requer o cálculo e o armazenamento dos autovetores do Laplaciano. Neste trabalho foi adotada a técnica de Anis, Gadde e Ortega (2016).

No trabalho (GADDE; ANIS; ORTEGA, 2014) os autores propõem uma técnica de reconstrução de um sinal através de suas amostras. Baseando-se na ideia de projeção em conjuntos convexos (POCS) (NARANG *et al.*, 2013). Os autores mostram por meio de experimentos numéricos que a amostragem e a reconstrução pelo método POCS pode superar a acurácia dos algoritmos de aprendizado supervisionado LLR (ZHANG *et al.*, 2011), LLGC (GU; HAN, 2012)

$\psi - \max$ (GUILLORY; BILMES, 2012).

Nesta dissertação propõe-se usar o algoritmo de amostragem do trabalho (GADDE; ANIS; ORTEGA, 2014) para escolher as amostras considerando as regiões de classificação incerta e propagar o rótulo destas amostras usando o algoritmo de propagação de rótulos LLGC (ZHOU *et al.*, 2004).

Figura 9 – Comparação dos métodos de aprendizado semi-supervisionado com o método proposto no trabalho (GADDE; ANIS; ORTEGA, 2014) usando conjunto de dados reais.

Fonte: Gadde, Anis e Ortega (2014).

TEORIA ESPECTRAL E AMOSTRAGEM EM GRAFOS

Neste capítulo, revisa-se de forma breve os conceitos relevantes para o estudo de processamento de sinal discreto em grafos. Apresenta-se uma abordagem teórica que generaliza a teoria clássica de processamento de sinal para domínios não cartesianos representados na forma de grafos.

3.1 Grafos e Sinais em Grafos

Formalmente defini-se um **grafo** $G = (\mathcal{V}, \mathcal{E})$ como sendo um objeto abstrato formado pelo conjunto de vértices $\mathcal{V}$ e o conjunto de arestas $\mathcal{E}$, onde $\mathcal{V} = \{v_0, v_1, \dots, v_{n-1}\}$ é um conjunto finito de n vértices ou nós e $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ é o conjunto de arestas definidas como pares ordenados (i, j) .

A **matriz de adjacência** $\mathbf{W} \in \mathbb{R}^{n \times n}$ de um grafo é definida como:

$$\mathbf{W}_{ij} = \begin{cases} w_{ij}, & \text{se } (i, j) \in \mathcal{E}; \\ 0, & \text{caso contrário.} \end{cases} \quad (3.1)$$

Se a entrada $\mathbf{w}_{ij} \neq 0$, então o vértice $v_j \in \mathcal{V}$ influencia o vértice $v_i \in \mathcal{V}$ (eles são ditos adjacentes) e diz-se que há uma aresta de v_i para v_j com peso $\mathbf{w}_{ij}$. A soma dos pesos das arestas incidentes em um vértice v_i será chamada grau de entrada de v_i denotado por d_i^- , enquanto se denotará por grau de saída d_i^+ a soma dos pesos das arestas que partem de v_i .

Para grafos não ponderados, $\mathbf{w}_{ij}$ é 1 se os vértices i e j estão conectados e 0 caso contrário. Para grafos não direcionados a matriz de adjacência é simétrica. Na figura 10 temos o exemplo de um grafo aleatório não ponderado, conhecido como grafo de Erdős-Renyi. O grafo

de Erdős-Renyi é construído conectando aleatoriamente os nós. Cada aresta é incluída no grafo com probabilidade p , independentemente de qualquer outra aresta.

Figura 10 – Grafo de Erdős-Renyi. Todos os pesos das arestas são iguais a 1. O grafo abaixo contém 64 nós, e 188 arestas e foi gerado considerando $p = 0.1$.

Fonte: Elaborada pelo autor.

Definição 1. Dada a representação do grafo $G = (\mathcal{V}, \mathcal{E})$, um **signal definido no grafo** é o mapeamento $f : \mathcal{V} \rightarrow \mathbb{R}$ dos vértices do grafo para o conjunto dos números reais.

Por extensão, pode-se definir sinais multidimensionais como um vetor de escalares em vez de escalares nos vértices.

Observação 1. A definição do sinal em um grafo, apresentada acima, pode ser generalizada para casos em que os vértices não são mapeados apenas para valores reais, mas alguma característica que pode ser associada a um número (conjunto enumerável). Por exemplo cada vértice pode ser associado a uma cor.

Exemplos de sinais definidos em grafos são ilustrados na figura 11. Em (a) tem-se um conjunto de dados colectados por um sensor: cada vértice do grafo é um sensor e as arestas conectam sensores próximos. Em (b) tem-se um sinal aleatório definido em um grafo do tipo cometa.

Uma vez fixada a ordem dos vértices, um sinal pode ser representado na forma vetorial:

$$\mathbf{f} = [f_0 \ f_1 \ \dots \ f_{n-1}]^T \in \mathbb{R}^n \quad (3.2)$$

onde o m -ésimo coeficiente do sinal $\mathbf{f}$ corresponde ao vértice v_m .

Figura 11 – Exemplo de sinais. Valores dos sinais são representados por cores. (a) medições de temperatura em diferentes regiões dos Estados Unidos, o grafo representa uma rede de sensores meteorológicos. (b) um sinal aleatório definido num grafo do tipo cometa. O grafo do cometa é um grafo de caminho com uma estrela de grau k no seu final

Fonte: Elaborada pelo autor.

Definição 2. O **Laplaciano** ou **matriz Laplaciana** de um grafo é definido como $\mathbf{L} = \mathbf{D} - \mathbf{W}$, onde $\mathbf{W}$ é a matriz de adjacência e $\mathbf{D} = (d_{ii})$ é uma matriz diagonal com entradas $d_{ii} = \sum_k \mathbf{W}_{ik}$ que é a soma dos pesos de todas as arestas que incidem no vértice v_i .

Para um grafo genérico as matrizes de adjacência Laplaciana são matrizes gerais (sem propriedades específicas). Por exemplo se o grafo for não direcionado e com pesos positivos a matriz de adjacência e o Laplaciano são simétricas e positivas semi-definidas.

Tratamos neste trabalho grafos não direcionados e com pesos positivos e portanto o Laplaciano $\mathbf{L}$ é uma matriz simétrica e positiva semi-definida, sendo diagonalizável e seus autovetores $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n$ são ortonormais. Assumimos que os autovalores estão ordenados de

forma não decrescente $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$.

Observação 2. A multiplicidade algébrica do autovalor nulo é igual ao número de componentes conexos do grafo G . Portanto se o grafo for conexo, zero é um autovalor simples do Laplaciano, e teremos $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_n$.

Outra versão da matriz Laplaciana é a versão **normalizada**, definida por

$$\mathcal{L} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2}$$

e como

$$\mathbf{L} = \mathbf{D} - \mathbf{W},$$

então

$$\mathcal{L} = \mathbf{D}^{-1/2} (\mathbf{D} - \mathbf{W}) \mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2},$$

com $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_n \leq 2$, $\lambda_n = 2$ se e somente se o grafo G é bipartido i.e o conjunto de vértices $\mathcal{V}$ pode ser particionado em dois subconjuntos $\mathcal{V}_1$ e $\mathcal{V}_2$ tal que cada aresta $e \in \mathcal{E}$ conecta um vértice de $\mathcal{V}_1$ a um de $\mathcal{V}_2$.

3.2 Transformada de Fourier em grafos

A transformada de Fourier de um sinal corresponde à sua projeção em uma base de funções invariantes à filtragem linear no tempo (VETTERLI; KOVAČEVIĆ; GOYAL, 2014). Por analogia estendemos esta ideia para o domínio dos grafos: a transformada de Fourier em grafos, é a decomposição de um sinal em termos de uma base de autovetores da matriz Laplaciana.

Na presente seção apresenta-se primeiramente a **transformada de Fourier** do processamento clássico de sinais. No contexto clássico a transformada de Fourier de um sinal é a expansão do sinal em termos das autofunções $e^{i\lambda t}$, do operador Laplaciano unidimensional, ou seja, se $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}$ é uma função ou sinal definido num domínio regular, a transformada de Fourier do sinal é dada por:

$$\hat{\mathbf{f}}(\lambda) = \mathcal{F}(\mathbf{f}(t)) = \int_{-\infty}^{+\infty} \mathbf{f}(t) e^{-i\lambda t} dt. \quad (3.3)$$

Para recuperar o sinal inicial $\mathbf{f}$ usamos a **transformada inversa de Fourier** que é dada por:

$$\mathbf{f}(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{\mathbf{f}}(\lambda) e^{i\lambda t} d\lambda. \quad (3.4)$$

As definições da transformada de Fourier e a sua inversa podem ser estendidas para o domínio de grafos. As auto funções do Laplaciano clássico são substituídas pelos autovetores da matriz Laplaciana do grafo.

Seja $G = (\mathcal{V}, \mathcal{E})$ um grafo ponderado, sendo $\mathcal{L}$ a matriz Laplaciana normalizada, e $\mathbf{f} : \mathcal{V} \rightarrow \mathbb{R}$ uma função definida nos vértices de G . A **transformada de Fourier do sinal $\mathbf{f}$** é definida como a projecção em cada autovetor da base,

$$\hat{\mathbf{f}}(\lambda_\ell) = \langle \mathbf{f}, \mathbf{u}_\ell \rangle = \sum_{i=1}^n \mathbf{f}(v_i) \mathbf{u}_\ell(v_i), \quad (3.5)$$

onde $\mathbf{u}_\ell$ são os autovetores e λ_ℓ são os autovalores de $\mathcal{L}$.

A transformada inversa de Fourier (**iGFT**) é a reconstrução do sinal usando os coeficientes da **GFT**

$$\mathbf{f}(i) = \sum_{\ell=1}^n \hat{\mathbf{f}}(\lambda_\ell) \mathbf{u}_\ell. \quad (3.6)$$

Os autovalores de $\mathcal{L}$ representam as frequências do sinal $\mathbf{f}$, e os autovetores a base de Fourier. A transformada de Fourier de um sinal no grafo é análoga à transformada de Fourier de um sinal no sentido clássico, ou seja ela revela as frequências presentes no sinal $\mathbf{f}$. A presença de frequências altas na transformada de Fourier revela que o sinal tem uma variação alta em algumas regiões do grafo, a presença de frequências baixas indica que o sinal tem variação baixa em outras regiões do grafo. Um autovalor grande implica maior variação no seu autovetor correspondente e um autovalor baixo indica menor variação na direção do seu autovetor correspondente. Para grafos conexos, o autovetor do Laplaciano, $\mathbf{u}_1$ correspondente ao autovalor $\lambda_1 = 0$ é constante e igual a $\frac{1}{\sqrt{n}}$ em cada vértice de G .

No exemplo da figura 12 nota-se que os autovetores associados à autovalores grandes mudam mais rapidamente nos vértices do grafo, são os autovetores mais prováveis de ter valores dissimilares nos vértices conectados por arestas de peso alto. Para mostrar o comportamento dos autovetores, foi considerando um grafo de caminho com 100 vértices.

A transformada de Fourier e a sua inversa permitem-nos representar o mesmo sinal em dois domínios diferentes: o domínio dos vértices e o domínio espectral do grafo.

A Figura 13 mostra a importância que a transformada de Fourier tem. Nos fornece a possibilidade de analisar o sinal no domínio da frequência. A Figura 13 mostra também a suavidade de cada sinal, a suavidade de um sinal pode também depender da geometria do grafo no qual é definido. A suavidade de um sinal $\mathbf{f}$ pode ser computada pela forma quadrática:

$$\mathbf{f}^T \mathbf{L} \mathbf{f} = \sum_{i \sim j} \mathbf{W}_{ij} (\mathbf{f}_i - \mathbf{f}_j)^2. \quad (3.7)$$

Quanto menor a quantidade dada pela equação 3.7 mais suave será o sinal $\mathbf{f}$ definido no grafo G e quanto maior esta quantidade o sinal será mesmo suave.

Figura 12 – Comportamento dos autovetores $u_0, u_5, u_{10}, u_{15}, u_{20}$ de um grafo de grelha com 100 vértices. O eixo vertical indica o índice do autovetor e o eixo horizontal indica o valor do autovetor.

Fonte: Elaborada pelo autor.

Figura 13 – Representações equivalentes de três sinais em grafos, em domínios diferentes: domínio dos vértices (a) e domínio espectral (b). Os sinais $\mathbf{f}$ foram definidos no grafo de duas luas. O grafo tem 2000 vértices construído usando vizinhos mais próximos.

Fonte: Elaborada pelo autor.

Figura 14 – Exemplo de um grafo de caminho com 100 vértices.

Fonte: Elaborada pelo autor.

Analogamente ao processamento clássico de sinais, os coeficientes de um sinal suave como o sinal mostrado na figura 13 ($\mathbf{f}^T \mathbf{L} \mathbf{f} = 19.92$) decaem muito rapidamente. Para além da análise espectral a transformada de Fourier **GFT** é também usada para generalizar conceitos do processamento clássico de sinais, tais como convolução, translação, ou modulação (HAMMOND; VANDERGHEYNST; GRIBONVAL, 2011; SHUMAN; FARAJI; VANDERGHEYNST, 2016).

3.3 Filtros, Translações e Convoluções

Definição 3. Dados dois sinais $\mathbf{f}$ e $\mathbf{h}$, a **convolução** destes dois sinais no vértice m é definida como (SHUMAN; FARAJI; VANDERGHEYNST, 2016; HAMMOND; VANDERGHEYNST; GRIBONVAL, 2011):

$$(\mathbf{f} * \mathbf{h})(m) = \sum_{\ell=0}^{n-1} \hat{\mathbf{f}}(\lambda_{\ell}) \hat{\mathbf{h}}(\lambda_{\ell}) \mathbf{u}_{\ell}(m). \quad (3.8)$$

Tal como no domínio Euclidiano temos que a convolução no domínio de vértices é equivalente à multiplicação no domínio espectral do grafo.

A convolução definida em 3.8 satisfaz as seguintes propriedades (SHUMAN; FARAJI; VANDERGHEYNST, 2016):

1. A convolução no domínio de vértices é equivalente à multiplicação no domínio espectral do grafo:

$$\widehat{\mathbf{f} * \mathbf{h}} = \hat{\mathbf{f}} \hat{\mathbf{h}}. \quad (3.9)$$

- 2.

$$\alpha(\mathbf{f} * \mathbf{h}) = (\alpha \mathbf{f}) * \mathbf{h} = \mathbf{f} * (\alpha \mathbf{h}), \quad \forall \alpha \in \mathbb{R} \quad (3.10)$$

3. Comutativa:

$$\mathbf{f} * \mathbf{h} = \mathbf{h} * \mathbf{f}. \quad (3.11)$$

4. Distributiva:

$$\mathbf{f} * (\mathbf{g} + \mathbf{h}) = \mathbf{f} * \mathbf{g} + \mathbf{f} * \mathbf{h}. \quad (3.12)$$

5. Associativa:

$$(\mathbf{f} * \mathbf{g}) * \mathbf{h} = \mathbf{f} * (\mathbf{g} * \mathbf{h}). \quad (3.13)$$

6. Definindo a função $\mathbf{h}_0$ em $\mathbb{R}^n$, por $\mathbf{h}_0(i) = \sum_{\ell=0}^{n-1} \mathbf{u}_{\ell}(i)$ então $\mathbf{h}_0$ é o elemento identidade da convolução

$$\mathbf{f} * \mathbf{h}_0 = \mathbf{f}. \quad (3.14)$$

7. Invariância com respeito ao Laplaciano normalizado:

$$\mathcal{L}(\mathbf{f} * \mathbf{h}) = (\mathcal{L}\mathbf{f}) * \mathbf{h} = \mathbf{f} * (\mathcal{L}\mathbf{h}). \quad (3.15)$$

8. A soma da convolução de dois sinais é igual a uma constante vezes o produto das somas dos dois sinais:

$$\sum_{i=1}^n (\mathbf{f} * \mathbf{h})(i) = \sqrt{n} \hat{\mathbf{f}}(0) \hat{\mathbf{h}}(0) = \frac{1}{\sqrt{n}} \left[\sum_{m=1}^n \mathbf{f}(i) \right] \left[\sum_{m=1}^n \mathbf{h}(i) \right]. \quad (3.16)$$

A operação clássica de **translação** é definida através da mudança de variável $(T_v \mathbf{f})(t) = \mathbf{f}(t - v)$. Contudo não pode ser generalizada para o domínio dos grafos. Os trabalhos de (HAMMOND; VANDERGHEYNST; GRIBONVAL, 2011; SHUMAN; RICAUD; VANDERGHEYNST, 2012) definem um operador generalizado de translação T_v de um sinal $\mathbf{f}$ do grafo G , como sendo a convolução com uma função de Kronecker δ centrada no vértice v :

$$T_v f(m) = \sqrt{n}(\mathbf{f} * \delta)(m) = \sqrt{n} \sum_{\ell=0}^{n-1} \hat{\mathbf{f}}(\lambda_\ell) \mathbf{u}_\ell^T(v) \mathbf{u}_\ell(m) \quad (3.17)$$

onde

$$\delta_v(m) = \begin{cases} 1, & \text{se } m = v; \\ 0, & \text{caso contrário.} \end{cases} \quad (3.18)$$

A função de Kronecker $\delta_v(m)$ é um sinal n -dimensional que é nulo em todos os vértices, exceto no vértice v , onde ele toma o valor 1.

Definição 4. Filtro de um grafo é um sistema ou operador $\mathbf{H} \in \mathbb{R}^{n \times n}$ que recebe o sinal $\mathbf{f}$ de um grafo como entrada e produz outro sinal como saída $\mathbf{y} = \mathbf{H}\mathbf{f}$.

O filtro de um grafo é um operador linear, visto que: Para quaisquer dois sinais $\mathbf{f}_1, \mathbf{f}_2$ e $\alpha_1, \alpha_2 \in \mathbb{R}$

$$\mathbf{H}(\alpha_1 \mathbf{f}_1 + \alpha_2 \mathbf{f}_2) = \alpha_1 \mathbf{H}(\mathbf{f}_1) + \alpha_2 \mathbf{H}(\mathbf{f}_2). \quad (3.19)$$

O filtro mais elementar de um grafo é a matriz de adjacência, este filtro substitui o valor de um sinal no vértice pela combinação linear dos sinais da vizinhança do vértice, esta operação é escrita como

$$\mathbf{y} = \mathbf{W}\mathbf{f}.$$

De forma análoga ao processamento de sinais clássico, a saída $\mathbf{f}_{out}$ da filtragem do sinal $\mathbf{f}$ com respeito ao filtro do grafo $\mathbf{h}$ é definido como sendo a multiplicação dos coeficientes da **GFT** $\hat{\mathbf{f}}(\lambda_\ell)$ com a função de transferência $\hat{\mathbf{h}}(\lambda_\ell)$ tal que

$$\hat{\mathbf{f}}_{out}(\lambda_\ell) = \hat{\mathbf{f}}(\lambda_\ell) \hat{\mathbf{h}}(\lambda_\ell), \quad \forall \lambda_\ell \in \sigma(\mathcal{L}), \quad (3.20)$$

onde $\sigma(\mathcal{L})$ é o espectro do Laplaciano normalizado. O sinal filtrado $\mathbf{f}_{out}$ no vértice m é calculado encontrando a transformada inversa de Fourier (**iGFT**) da equação 3.20:

$$\mathbf{f}_{out}(m) = \sum_{\ell=0}^{n-1} \hat{\mathbf{f}}(\lambda_\ell) \hat{\mathbf{h}}(\lambda_\ell) \mathbf{u}_\ell(m). \quad (3.21)$$

A equação 3.21 pode ser expressa na forma matricial (HIGHAM, 2008) como:

$$\mathbf{f}_{out}(m) = \hat{\mathbf{h}}(\mathcal{L})\mathbf{f}, \quad (3.22)$$

onde

$$\hat{\mathbf{h}}(\mathcal{L}) = \mathbf{U} \begin{bmatrix} \hat{\mathbf{h}}(\lambda_0) & & 0 \\ & \ddots & \\ 0 & & \hat{\mathbf{h}}(\lambda_{n-1}) \end{bmatrix} \mathbf{U}^H$$

é o filtro do grafo ou núcleo definido no domínio espectral do grafo.

Proposição 1. Se o filtro do grafo for um polinômio de grau K com coeficientes $\{a_k\}_{k=0}^K$ tais que

$$\hat{\mathbf{h}}(\lambda_\ell) = \sum_{k=0}^K a_k \lambda_\ell^k, \quad (3.23)$$

o sinal filtrado em cada vértice m é a combinação linear no sinal original na vizinhança de m .

Demonstração. Combinando as equações 3.22 e 3.23 temos:

$$\mathbf{f}_{out}(m) = \sum_{\ell=0}^{n-1} \hat{\mathbf{f}}(\lambda_\ell) \hat{\mathbf{h}}(\lambda_\ell) \mathbf{u}_\ell(m) \quad (3.24)$$

$$= \sum_{p=1}^n \mathbf{f}(m) \sum_{k=0}^K a_k \sum_{\ell=0}^{n-1} \lambda_\ell^k \mathbf{u}_\ell^T(p) \mathbf{u}_\ell(m) \quad (3.25)$$

$$= \sum_{p=1}^n \mathbf{f}(m) \sum_{k=1}^K a_k (\mathcal{L}^k)_{m,p}, \quad (3.26)$$

onde $(\mathcal{L}^k)_{m,p} = 0$ se o caminho mais curto entre os vértices m e p é maior que k (HAMMOND; VANDERGHEYNST; GRIBONVAL, 2011).

□

Esta propriedade pode ser útil para projetar sinais localizados no domínio de vértices do grafo. Uma abordagem mais detalhada dessas operações básicas pode ser encontrada no trabalho (SHUMAN SUNIL K. NARANG; VANDERGHEYNST, 2013).

3.4 Amostragem em grafos

Neste capítulo são discutidas questões relacionadas ao teorema de amostragem. Condições necessárias e suficientes para a reconstrução única de um sinal de banda limitada a partir de suas amostras e um algoritmo guloso para a extração de uma amostra ótima.

A estratégia de amostragem apresentada neste capítulo foi desenvolvida e aprimorada nos trabalhos (PESENSE, 2008; ANIS; GADDE; ORTEGA, 2014; ANIS; GADDE; ORTEGA, 2016). Esta estratégia foi motivada por filtros espectrais localizados no domínio de vértices de um grafo. Para tal define-se quantidades denominadas *proxies espectrais do grafo* baseadas em potências do Laplaciano. Usando estes *proxies* pode-se aproximar da máxima frequência de corte um sinal que garante a reconstrução perfeita. Mostra-se também que esta aproximação nos fornece uma noção da medida de estabilidade da reconstrução para uma dada amostra. Por último introduz-se um algoritmo guloso que tem como objetivo minimizar o erro de aproximação.

3.5 Amostragem e interpolação

Seja o grafo $G = (\mathcal{V}, \mathcal{E})$. Um subconjunto de vértices do grafo será denotado por um subconjunto de índices $\mathcal{S} \subset \mathcal{V}$, e $\mathcal{S}^c = \mathcal{V} \setminus \mathcal{S}$ será o conjunto complementar.

Denota-se por $\mathbf{W}_{\mathcal{S}_1, \mathcal{S}_2}$ a submatriz de qualquer matriz $\mathbf{W}$ com linhas em $\mathcal{S}_1$ e colunas em $\mathcal{S}_2$, sendo $\mathcal{S}_1$ e $\mathcal{S}_2$ subconjunto de índices. Para abreviar $\mathbf{W}_{\mathcal{S}, \mathcal{S}} = \mathbf{W}_{\mathcal{S}}$.

Considera-se o problema da extração de um conjunto $\mathcal{S} = \{\mathcal{S}_0, \dots, \mathcal{S}_{m-1}\}$ de índices de coordenadas do sinal $\mathbf{x} \in \mathbb{R}^n$ para produzir um sinal amostrado $\mathbf{x}_{\mathcal{S}} \in \mathbb{R}^m$ ($n > m$), onde $\mathcal{S}$ denota a sequência de coeficientes amostrados e então interpolar $\mathbf{x}_{\mathcal{S}}$ para obter $\mathbf{x}' \in \mathbb{R}^n$. O operador de amostragem Ψ é um mapeamento linear de $\mathbb{R}^n$ para $\mathbb{R}^m$, definido como:

$$\Psi_{ij} = \begin{cases} 1, & \text{se } j = \mathcal{S}_i; \\ 0, & \text{caso contrário} \end{cases} \quad (3.27)$$

e o operador de interpolação Φ é um mapeamento linear de $\mathbb{R}^m$ para $\mathbb{R}^n$. A amostragem é efetuada pela operação $\mathbf{x}_{\mathcal{S}} = \Psi \mathbf{x} \in \mathbb{R}^m$ e a interpolação pela operação $\mathbf{x}' = \Phi \mathbf{x}_{\mathcal{S}} = \Phi \Psi \mathbf{x} \in \mathbb{R}^n$, onde $\mathbf{x}' \in \mathbb{R}^n$ recupera $\mathbf{x} \in \mathbb{R}^n$ exata ou aproximadamente.

Definição 5. Um sinal $\mathbf{f}$ é de banda ω se os coeficientes da transformada de Fourier (GFT) de $\mathbf{f}$ são limitados por ω , i.e. seu suporte espectral é restrito ao conjunto de frequências $[0, \omega]$ (PESENSE, 2008).

Observação 3. O espaço de todos os sinais de banda ω é conhecido como o espaço de **Paley-Wiener** e é denotado por $PW_{\omega}(G)$ (PESENSE, 2008).

$$PW_{\omega}(G) = \{\mathbf{f} : \hat{\mathbf{f}}(\lambda) = 0 \text{ se } \lambda \geq \omega\}. \quad (3.28)$$

Note que $PW_{\omega}(G)$ é um subespaço do $\mathbb{R}^n$. Pela linearidade da equação 3.5 pode-se mostrar que $PW_{\omega}(G)$ é um subespaço de $\mathbb{R}^n$.

A **GFT** introduz a noção de frequência e portanto podemos recorrer à teoria de amostragem. Neste contexto as seguintes questões surgem (PESENSON, 2008; ANIS; GADDE; ORTEGA, 2014; ANIS; GADDE; ORTEGA, 2016).

1. **Frequência de corte:** Para um dado subconjunto de nós $\mathcal{S}$, encontre a frequência de corte ω , de modo que qualquer $\mathbf{f} \in PW_\omega(G)$ possa ser reconstituído com exatidão a partir de suas amostras $\mathbf{f}(\mathcal{S})$.
2. **Amostra ótima:** Para uma determinada frequência de corte ω , encontre o menor subconjunto de nós $\mathcal{S}_{opt}$ (ou seja, com a mínima cardinalidade possível), de tal forma que todos os sinais $\mathbf{f} \in PW_\omega(G)$ possam ser reconstruídos através de suas amostras $\mathbf{f}(\mathcal{S}_{opt})$.

Para amostragem regular no processamento de sinal clássico, as questões 1 e 2 são recíprocas, isto é, o conhecimento de um leva automaticamente à solução do outro. No entanto, isso não vale para o caso de sinais em grafos.

3.5.1 Frequência de corte

Definição 6. (Conjunto de exclusividade ou unicidade (WANG; WANG; GU, 2015)) Um subconjunto $\mathcal{S} \subset \mathcal{V}$ é de unicidade no espaço $PW_\omega(G)$ se, para quaisquer dois sinais de $PW_\omega(G)$, o fato deles coincidirem em $\mathcal{S}$ implica que coincidem em $\mathcal{V}$, ou seja $\forall \mathbf{f}, \mathbf{g} \in PW_\omega(G)$, $\mathbf{f}(\mathcal{S}) = \mathbf{g}(\mathcal{S}) \implies \mathbf{f} = \mathbf{g}$.

A definição acima implica que a amostragem para $\mathcal{S}$ é uma operação injetora para todo sinal limitado de banda ω , portanto é suficiente conhecer seus valores no conjunto de unicidade $\mathcal{S}$.

Teorema 1. Para um grafo G com Laplaciano normalizado $\mathcal{L}$, um conjunto $\mathcal{S}$ é de **unicidade** para todo sinal de $PW_\omega(G)$ se $\omega < \omega_{\mathcal{S}}^* = \sigma_{min}$, onde $\sigma_{min}^2 = \lambda_{min}(\mathcal{L}^2)_{\mathcal{S}^c}$ (ANIS; GADDE; ORTEGA, 2014).

Este teorema nos fornece apenas uma condição suficiente e não garante que a frequência de corte estimada seja a melhor possível, dado $\mathcal{S}$. Todavia, seria importante dar uma condição necessária e suficiente para $\mathcal{S}$ ser um conjunto de unicidade e fornecer uma maneira prática de melhorar a estimativa da frequência de corte.

Definição 7. Define-se $L_2(\mathcal{S}^c)$ como sendo o espaço de todos os sinais que são nulos em todos os vértices de $\mathcal{S}$, isto é, $\forall \phi \in L_2(\mathcal{S}^c)$, $\phi(\mathcal{S}) = 0$.

Lema 1. $\mathcal{S}$ é um conjunto de unicidade para sinais em $PW_\omega(G)$ se e somente se $PW_\omega(G) \cap L_2(\mathcal{S}^c) = \{\mathbf{0}\}$.

Demonstração. Dado que $PW_\omega(G) \cap L_2(\mathcal{S}^c) = \{\mathbf{0}\}$, suponha que $\mathcal{S}$ não é conjunto de unicidade. Então

$$\exists \mathbf{f}, \mathbf{g} \in PW_\omega(G), \mathbf{f} \neq \mathbf{g} : \mathbf{f}(\mathcal{S}) = \mathbf{g}(\mathcal{S}). \quad (3.29)$$

Consequentemente teremos $\mathbf{f} - \mathbf{g} \in L_2(\mathcal{S}^c)$, $\mathbf{f} - \mathbf{g} \neq \mathbf{0}$. Temos também que $\mathbf{f} - \mathbf{g} \in PW_\omega(G)$. Mas isto é uma contradição sendo que $PW_\omega(G) \cap L_2(\mathcal{S}^c) = \{\mathbf{0}\}$. Portanto $\mathcal{S}$ deve ser conjunto de unicidade.

Por outro lado assumindo que $\mathcal{S}$ é um conjunto de unicidade. Seja $\mathbf{s}$ qualquer sinal em $PW_\omega(G) \cap L_2(\mathcal{S}^c)$. Então, para qualquer sinal $\mathbf{f} \in PW_\omega(G)$, temos $\mathbf{g} = \mathbf{f} + \mathbf{s} \in PW_\omega(G)$ e $\mathbf{f}(\mathcal{S}) = \mathbf{g}(\mathcal{S})$. Mas visto que $\mathcal{S}$ é um conjunto de unicidade, devemos ter $\mathbf{f} = \mathbf{g}$, portanto $\mathbf{s} = \mathbf{0}$, implicando em $PW_\omega(G) \cap L_2(\mathcal{S}^c) = \{\mathbf{0}\}$.

□

Definição 8. A **frequência de corte** é definida como sendo a maior frequência ω para a qual $\mathcal{S}$ é conjunto de unicidade para $PW_\omega(G)$ (ANIS; GADDE; ORTEGA, 2014).

Do lema 1: temos, para que $\mathcal{S}$ seja conjunto de unicidade em $PW_\omega(G)$, ω precisa ser menor que o mínimo limite de banda que um sinal pode ter em $L_2(\mathcal{S}^c)$. Contudo a frequência de corte de um conjunto de amostragem $\mathcal{S}$, pode ser expressa da seguinte forma:

$$\omega_c(\mathcal{S}) = \inf_{\mathbf{0} \neq \phi \in L_2(\mathcal{S}^c)} \omega(\phi) \quad (3.30)$$

onde, $\omega_c(\mathcal{S})$ é a frequência de corte de $\mathcal{S}$ e $\omega(\phi)$ é a largura de banda do sinal ϕ .

Teorema 2. Para um grafo G , com o Laplaciano normalizado $\mathcal{L}$, todo sinal $\phi \in PW_\omega(G)$ pode ser perfeitamente recuperado de sua amostra $\mathcal{S} \subset \mathcal{V}$ se e somente se

$$\omega < \omega_c(\mathcal{S}) = \inf_{\phi \in L_2(\mathcal{S}^c)} \omega(\phi) \quad (3.31)$$

onde $\omega_c(\mathcal{S})$ é a frequência de corte (ANIS; GADDE; ORTEGA, 2014). (Prova: ver referência)

Para aplicar o teorema de amostragem, primeiro precisamos do limite da banda $\omega(\phi)$ de qualquer sinal ϕ . Usando a transformada de Fourier isto é simples, mas implicaria calcular a **GFT** para todos os sinais em $L_2(\mathcal{S}^c)$ e procurar pelo sinal mais suave ϕ^* . Precisa-se também minimizar a largura da banda do sinal, para todos os sinais de $L_2(\mathcal{S}^c)$. Este problema pode ser resolvido de forma mais eficiente, definindo a quantidade que se segue:

Definição 9. Para qualquer sinal $\mathbf{f} \neq \mathbf{0}$, definimos o seu **k-ésimo momento espectral** (ANIS; GADDE; ORTEGA, 2014)

$$\omega_k(\mathbf{f}) = \left(\frac{\mathbf{f}^T \mathcal{L}^k \mathbf{f}}{\mathbf{f}^T \mathbf{f}} \right)^{1/k} = \left(\sum_{i=1}^n \lambda_i^k \frac{\hat{\mathbf{f}}(i)^2}{\sum_{j=1}^n \hat{\mathbf{f}}(j)^2} \right)^{1/k} \quad (3.32)$$

Essas quantidades são denominadas de “*momentos espectrais*”. Pode ser mostrado (ANIS; GADDE; ORTEGA, 2016) que $\omega_k(\mathbf{f})$ cresce monotonicamente em relação a k :

$$\forall \mathbf{f}, k_1 > k_2 \Rightarrow \omega_{k_1}(\mathbf{f}) \geq \omega_{k_2}(\mathbf{f}). \quad (3.33)$$

Portanto definimos a **frequência de corte de ordem k** de $\mathcal{S}$ como:

$$\Omega_k(\mathcal{S}) = \inf_{\phi \in L_2(\mathcal{S}^c)} \omega_k(\phi) = \inf_{\phi \in L_2(\mathcal{S}^c)} \left(\frac{\phi^T \mathcal{L}^k \phi}{\phi^T \phi} \right)^{1/k}. \quad (3.34)$$

Usando as definições de $\Omega_k(\mathcal{S})$ e $\omega_c(\mathcal{S})$ e a equação 3.33 concluímos que para quaisquer $k_1 > k_2$:

$$\omega_c(\mathcal{S}) \geq \lim_{k \rightarrow \infty} \Omega_k(\mathcal{S}) \geq \Omega_{k_1}(\mathcal{S}) \geq \Omega_{k_2}(\mathcal{S}). \quad (3.35)$$

O resultado acima tem implicações significativas: primeiro, à medida que aumentamos o valor de k , Ω_k tende a fornecer uma estimativa melhor da frequência de corte. Em segundo lugar, Ω_k é sempre menor que o corte real ω_c , garantindo assim a reconstrução perfeita. Isso ocorre porque se $\mathcal{S}$ é um conjunto de unicidade para $PW_{\omega_1}(G)$ e $\omega_1 > \omega_2$, então $PW_{\omega_2}(G) \subseteq PW_{\omega_1}(G)$ e, portanto, $\mathcal{S}$ também é um conjunto de unicidade para $PW_{\omega_2}(G)$. Assim usando a equação 3.35 e o teorema 2 enuncia-se a proposição:

Proposição 2. Para todo k , $\mathcal{S} \subset \mathcal{V}$ é um conjunto de unicidade em $PW_{\omega}(G)$ se $\omega < \Omega_k(\mathcal{S})$ pode ser computado por 3.34 da seguinte forma:

$$\Omega_k(\mathcal{S}) = \left[\inf_{\psi} \left(\frac{\psi^T (\mathcal{L}^k)_{\mathcal{S}^c} \psi}{\psi^T \psi} \right) \right]^{1/k} = (\sigma_1, k)^{1/k}. \quad (3.36)$$

Numericamente Ω_k e ψ_* podem ser determinados do menor par (autovalor, autovetor) $(\sigma_{1,k}, \psi_{1,k})$ da matriz $(\mathcal{L}^k)_{\mathcal{S}^c}$:

$$\Omega_k(\mathcal{S}) = \sigma_{1,k}, \quad (3.37)$$

$$\phi_k^*(\mathcal{S}^c) = \psi_{1,k}, \phi_k^*(\mathcal{S}) = \mathbf{0}. \quad (3.38)$$

Notamos de 3.35 que precisamos simplesmente de um k maior para obter uma estimativa melhor da verdadeira frequência de corte. Portanto, há um “*trade-off*” entre a precisão da estimativa,

por um lado a complexidade e estabilidade numérica. Por outro lado uma vez que temos uma estimativa do ponto de corte, a interpolação pode ser realizada usando técnicas propostas nos trabalhos (GU; HAN, 2012) (GUILLORY; BILMES, 2012) (GADDE; ANIS; ORTEGA, 2014) (WANG; LIU; GU, 2015) e (ZHANG *et al.*, 2011).

3.6 Amostra ótima

Nesta secção apresenta-se o algoritmo de amostragem guloso. A questão estudada é: dado um sinal $\mathbf{f}$, como determinar a melhor amostra $\mathcal{S}_{opt}$ de tamanho M . Para determinar uma estimativa do conjunto $\mathcal{S}_{opt}$ utiliza-se um algoritmo eurístico guloso.

De forma geral o algoritmo guloso inicia com um conjunto vazio ($\Omega_k(\mathcal{S}) = 0$) e continua adicionando vértices de $\mathcal{S}^c$ um por um tentando atingir o máximo crescimento de $\Omega_k(\mathcal{S})$ a cada passo. Espera-se que $\Omega_k(\mathcal{S})$ alcance a frequência de corte ω_c com o mínimo possível de vértices adicionados no conjunto $\mathcal{S}$.

Para decidir quais vértices devem ser adicionados a $\mathcal{S}$, introduzimos uma relaxação binária da frequência de corte e definimos a quantidade:

$$\lambda_k^\alpha(\mathbf{I}_{\mathcal{S}}) = \inf_{\mathbf{x}} \left(\frac{\mathbf{x}^T \mathcal{L}^k \mathbf{x}^T}{\mathbf{x}^T \mathbf{x}} + \alpha \frac{\mathbf{x}^T \text{diag}(\mathbf{I}_{\mathcal{S}}) \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right) = \inf_{\mathbf{x}} \left(\omega_k(\mathbf{x}) + \alpha \frac{\mathbf{x}^T \text{diag}(\mathbf{I}_{\mathcal{S}}) \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right). \quad (3.39)$$

onde $\mathbf{I}_{\mathcal{S}} : \mathcal{V} \rightarrow \{0, 1\}$ (i.e. $\mathbf{I}(\mathcal{S}) = 1$ e $\mathbf{I}(\mathcal{S}^c) = 0$) é a função característica de $\mathcal{S}$.

Note que o lado direito da equação 3.39 é uma simples regularização do problema de minimização 3.34. Quando $\alpha \gg 1$, as componentes de $\mathbf{x}(\mathcal{S})$ são altamente penalizadas durante a minimização. Contudo se $\mathbf{x}_k^\alpha(\mathbf{I}_{\mathcal{S}})$ for a solução de 3.39, então $[\mathbf{x}_k^\alpha(\mathbf{I}_{\mathcal{S}})](\mathcal{S}) \rightarrow \mathbf{0}$ ou seja os valores de $\mathcal{S}$ tendem a ser muito pequenos. Portanto, para $\alpha \gg 1$ temos que:

$$\phi_k^* \approx \mathbf{x}_k^\alpha(\mathbf{I}_{\mathcal{S}}), \quad (3.40)$$

$$\Omega_k(\mathcal{S}) \approx (\lambda_k^\alpha(\mathcal{S}))^{1/k}. \quad (3.41)$$

Das equações 3.40 e 3.41, observa-se que o problema de maximizar $\Omega_k(\mathcal{S})$ é equivalente à maximização de $\lambda_k^\alpha(\mathbf{I}_{\mathcal{S}})$, e portanto podemos simplesmente estudar a variação de $\lambda_k^\alpha(\mathbf{t})$. Esta relaxação contorna a natureza combinatória do problema.

$$\lambda_k^\alpha(\mathbf{t}) = \inf_{\mathbf{x}} \left(\omega_k(\mathbf{x}) + \alpha \frac{\mathbf{x}^T \text{diag}(\mathbf{t}) \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \right) = \inf_{\mathbf{x}} \omega_k^\alpha(\mathbf{x}, \mathbf{t}), \quad \mathbf{t} \in \mathbb{R}^n. \quad (3.42)$$

Note que

$$\frac{d\omega_k^\alpha(\mathbf{x}, \mathbf{t})}{d\mathbf{t}(i)} = \frac{d\left(\omega_k(\mathbf{x}) + \alpha \frac{\mathbf{x}^T \text{diag}(\mathbf{t})\mathbf{x}}{\mathbf{x}^T \mathbf{x}}\right)}{d\mathbf{t}(i)} = \alpha \left(\frac{\mathbf{x}(i)}{\|\mathbf{x}\|^2}\right)^2 \quad \forall \mathbf{x}, \mathbf{t}. \quad (3.43)$$

Da equação 3.40 sabemos que quando $\mathbf{t} = I_{\mathcal{S}}$ então ϕ_k^* minimiza a equação 3.42. Consequentemente,

$$\left. \frac{d\lambda_k^\alpha(\mathbf{t})}{d\mathbf{t}(i)} \right|_{\mathbf{t}=I_{\mathcal{S}}} = \left. \frac{d\omega_k^\alpha(\phi_k^*, \mathbf{t})}{d\mathbf{t}(i)} \right|_{\mathbf{t}=I_{\mathcal{S}}} = \alpha \left(\frac{\phi_k^*(i)}{\|\phi_k^*\|^2}\right)^2. \quad (3.44)$$

A equação 3.44 nos diz exatamente qual vértice devemos incluir a cada passo do algoritmo guloso, sendo este o vértice em que o sinal mais suave ϕ_k^* tem máxima energia, ou seja $I_{\mathcal{S}}(i) \leftarrow 1, i = \text{argmax}_j (\phi_k^*(j))^2$. O método para estimar $\mathcal{S}_k^{opt}$ esta resumido no algoritmo heurístico 1.

Algoritmo 2 – Algoritmo heurístico guloso para estimar $\mathcal{S}_{est}$

- 1: **ENTRADA:** $G = (\mathcal{V}, \mathcal{E})$, $\mathcal{L}$, tamanho da amostra $M, k \in \mathbb{Z}^+$
 - 2: **INICIALIZAR:** $\mathcal{S} = \emptyset$.
 - 3: **enquanto** $|\mathcal{S}| < M$ **faça**
 - 4: Para $\mathcal{S}$, compute a estimativa $\Omega_k(\mathcal{S})$ e o sinal mais suave $\phi_k^* \in L_2(\mathcal{S}^c)$ usando a proposição 2.
 - 5: $v \leftarrow \text{argmax}_i (\phi_k^*(i))^2$.
 - 6: $\mathcal{S} \leftarrow \mathcal{S} \cup v$.
 - 7: **fim enquanto**
 - 8: $\mathcal{S}_{est} \leftarrow \mathcal{S}$.
-

Definição 10. Definimos o operador $\Psi(\mathbf{W}, k, M)$, como sendo o operador de amostragem que aplica o algoritmo 2 para extrair uma amostra de tamanho M , de um grafo cuja matriz de pesos é $\mathbf{W}$.

Exemplo 1. Considere o grafo G da figura 15:

Figura 15 – Exemplo de um grafo do qual se pretende extrair uma amostra.

Fonte: Elaborada pelo autor.

- A matriz de pesos é:

$$\mathbf{W} = \begin{bmatrix} 0 & 0.6 & 0 & 0 & 0 & 0 \\ 0.6 & 0 & 0.4 & 0 & 0.1 & 0 \\ 0 & 0.4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7 & 0 \\ 0 & 0.1 & 0 & 0.7 & 0 & 0.9 \\ 0 & 0 & 0 & 0 & 0.9 & 0 \end{bmatrix}$$

- A aplicação do operador $\Psi(\mathbf{W}, k, |\mathcal{S}|)$, com $|\mathcal{S}| = 2$ e $k = 8$ devolve um conjunto de dois índices, sendo $\Psi(\mathbf{W}, 8, 2) = \{1, 4\}$. O que, significa que aplicando a técnica de amostragem no grafo G para obter uma amostra de tamanho 2, o algoritmo escolhe os vértices v_1 e v_4 . A escolha destes dois vértices para a mostra não é surpreendente porque estes tem grau mais alto que os outros. Portanto um vértice com grau alto tem chances altas de ser considerado durante o processo de amostragem. Isso acontece porque estes vértices contêm mais informações e podem ser úteis durante a propagação de informação.

3.7 Experimentos preliminares de amostragem

Nesta seção apresenta-se os resultados visuais e numéricos da amostragem usada para a classificação semi-supervisionada. Mostra-se através de resultados numéricos que a frequência de corte aumenta com k . Para os experimentos, primeiro foram gerados quatro tipos de grafos conexos e não direcionados, a saber:

G_1 : Grafo circular com $n = 90$ vértices.

G_2 : Grafo de comunidades, com 8 comunidades e um total de 256 vértices.

G_3 : Grafo de sensores aleatórios com 360 vértices gerados aleatoriamente no conjunto $[0, 1] \times [0, 1]$ do plano cartesiano. Os pesos são atribuídos com uma função de ponderação de Kernel Gaussiano com base na distância Euclidiana entre os nós:

$$\mathbf{W}_{ij} = \begin{cases} \exp\left(-\frac{[dist(i, j)^2]}{2\sigma_1^2}\right), & \text{se } d(i, j) \leq \sigma_2; \\ 0, & \text{caso contrário,} \end{cases} \quad (3.45)$$

com $\sigma_1 = 0.075$ e $\sigma_2 = 0.076$.

G_4 : Grafo Minnesota (representa as ruas de Minnesota).

A figura 17 mostra o quão eficaz é a técnica de amostragem que se pretende usar para uma pós classificação dos dados. Em (a) temos a ilustração de um grafo circular. Pode ser

notado que na escolha de uma amostra de tamanho dez a técnica de amostragem tenta escolher pontos que maximizem o espaço entre os elementos da amostra. O grafo (b) é um grafo de oito comunidades, nota-se que ao extrair a amostra, a técnica de amostragem escolhe pelo menos um ponto de cada comunidade, o que mostra que a técnica extrai uma amostra que pode levar a um erro pequeno na classificação dos pontos restantes. O mesmo pode ser dito para o grafo das ruas de Minnesota (c).

Na figura 18 ilustra-se a amostra obtida pelo algoritmo para diferentes valores k (2, 8, 16 e 32) de grafos de sensores aleatórios. Nota-se que para k pequeno o algoritmo escolhe pontos próximos, e quanto maior for k o algoritmo maximiza a distância entre as amostras. Para um espaço de dimensão maior ($d > 2$) o algoritmo deve escolher pontos que capturem mais dimensões em termos de informação (GADDE; ANIS; ORTEGA, 2014).

Mesmo que aumentar o valor de k possa aumentar a frequência de corte, o que pode resultar na melhoria dos algoritmos de propagação de rótulos, aumentar este valor de forma demasiada pode criar instabilidades numéricas durante o processo de amostragem e também pode tornar o processo mais lento, já que durante a amostragem devemos calcular $(\mathcal{L}^k)_{\mathcal{S}^c}$.

A figura 16 mostra o comportamento instável da frequência de corte quando considera-se k muito grande. No experimento da figura 16 foi extraída uma amostra de tamanho 200 em um grafo de sensores aleatórios de 1000 vértices, usando os diferentes valores de k . O objetivo foi mostrar o comportamento instável da frequência de corte durante a amostragem.

A figura 18 mostra visualmente como o algoritmo escolhe os vértices. É possível notar que para frequências de corte menores ($k = 2$) o algoritmo tende a escolher vértices que estão próximos, e para valores grandes de k o algoritmo escolhe vértices que estão mais distantes. Os algoritmos de aprendizado semi-supervisionado funcionam bem quando supõe-se a formação de *clusters*, ou seja pontos próximos tendem a ter o mesmo rótulo. Portanto neste caso escolher k muito pequeno não é viável já que o algoritmo tende a escolher amostras próximas. No capítulo 4 faz-se um estudo mais detalhado de como escolher k para diferentes tipos de problema.

Tabela 1 – Estimativa da frequência de corte para uma determinada amostra.

Grafo	$ \mathcal{S} $	$k = 2$	$k = 8$	$k = 16$	$k = 25$
G_1 (Grafo circular)	10	0.0881	0.5480	0.7381	0.8234
G_2 (Grafo de comunidades)	16	0.1462	0.6562	0.7851	0.8788
G_3 (Grafo de Tráfego de Minnesota)	14	0.007	0.2741	0.5351	0.6785

Figura 16 – Frequências de corte para diferentes valores de k . $k = 2, k = 8, k = 16$ e $k = 32$.

Fonte: Elaborada pelo autor.

Figura 17 – Exemplo de amostras extraídas de grafos. Os vértices circundados pelas bolas pretas são os vértices amostrados. (a) representação visual do grafo circular e da amostra $|\mathcal{S}_{opt}| = 10$. (b) representação visual do grafo de comunidades e da amostra $|\mathcal{S}_{opt}| = 8$. (c) representação visual do grafo de ruas de Minnesota e da amostra $|\mathcal{S}_{opt}| = 10$

Fonte: Elaborada pelo autor.

Figura 18 – Exemplo de amostras de tamanho $|\mathcal{S}_{opt}| = 10$ retiradas de grafos de sensores aleatórios, mostrando a qualidade visual da amostra.

(a) Amostras obtidas pelo algoritmo para $k = 2$

(b) Amostras obtidas pelo algoritmo para $k = 4$

(c) Amostras obtidas pelo algoritmo para $k = 8$

(d) Amostras obtidas pelo algoritmo para $k = 16$

Fonte: Elaborada pelo autor.

ALGORITMO PARA APRENDIZADO SEMI-SUPERVISIONADO

4.1 Estrutura do aprendizado semi-supervisionado ativo

Neste capítulo sumariza-se os passos necessários para a implementação da técnica de aprendizado semi-supervisionado ativo proposta. O algoritmo de amostragem considerado é aquele proposto por (ANIS; GADDE; ORTEGA, 2016), adaptado para considerar não apenas os vértices que maximizam a frequência de corte, mas também os vértices com classificação incerta. A hipótese é que considerar pontos de classificação incerta pode ajudar a maximizar a acurácia da classificação de um algoritmo semi-supervisionado. Apresenta-se o algoritmo de propagação de rótulos que foi usado, define-se regiões de incerteza, apresenta-se uma estratégia de seleção de vértices da região de incerteza através da modificação da matriz de similaridades do grafo, e por fim a proposta do algoritmo ativo semi-supervisionado. A estrutura da amostragem iterativa se baseia no modelo de amostragem de (ANIS; GADDE; ORTEGA, 2016) e no modelo da consistência local e global (ZHOU *et al.*, 2004).

A figura 2 representa o diagrama do aprendizado proposto.

4.1.1 Algoritmo de propagação de rótulos da amostra $\mathcal{S}$

Considere o conjunto de n pontos $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_\ell, \mathbf{x}_{\ell+1}, \dots, \mathbf{x}_n\} \subset \mathbb{R}^d$ e o conjunto de classes $L = \{1, \dots, c\}$. Os pontos $(\mathbf{x}_{i \leq \ell})$ são rotulados com $f_i \in L$, sendo que o restante dos pontos $\mathbf{x}_u (\ell + 1 \leq u \leq n)$ não são rotulados. O objetivo é classificar os pontos não rotulados a partir dos pontos rotulados.

Seja $\mathcal{F}$ o conjunto de matrizes $n \times c$ com entradas não negativas. A matriz $\mathbf{F} = [\mathbf{F}_1^T, \dots, \mathbf{F}_n^T]^T \in \mathcal{F}$ corresponde à classificação no conjunto de dados $\mathcal{X}$, rotulando cada ponto $\mathbf{x}_i$ com, $f_i = \operatorname{argmax}_{j \leq c} \mathbf{F}_{ij}$. Nota-se que $\mathbf{F} : \mathbf{X} \rightarrow \mathbb{R}^c$ é um operador que associa cada vetor $\mathbf{F}_i$

a cada ponto $\mathbf{x}_i$. Define-se uma matriz $n \times c$, $\mathbf{Y} \in \mathbf{F}$ por $\mathbf{Y}_{ij} = 1$ se $\mathbf{x}_i$ foi rotulado com $f_i = j$ e $\mathbf{Y}_{ij} = 0$ caso contrário. Visto que a proposta é apresentada para cenários reais em que existe uma escassez de pontos rótulados, assume-se que os rótulos fornecidos pelo usuário são considerados consistentes. Portanto não devem mudar de classificação. Contudo $\mathbf{Y}$ é consistente com a rotulação inicial. O algoritmo de propagação de rótulos é descrito abaixo:

LLGC. – Algoritmo de propagação de rótulos de $\mathcal{S}$

1: A partir da matriz de similaridades, definida por:

$$\mathbf{W}_{ij} = \begin{cases} \exp\left(-\frac{\text{dist}(\mathbf{x}_i, \mathbf{x}_j)^2}{2\sigma^2}\right), & \text{se } i \neq j; \\ 0, & \text{se } i = j \end{cases} \quad (4.1)$$

2: Constrói-se a matriz: $\mathcal{W} = \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}$.

3: Itera-se $\mathbf{F}(t+1) = \alpha \mathcal{W} \mathbf{F}(t) + (1 - \alpha) \mathbf{Y}$ até convergência, $\alpha \in (0, 1)$.

4: Seja $\mathbf{F}^*$ o limite da sequência $\{\mathbf{F}(t)\}$. Rotule cada $\mathbf{x}_i$ como $f_i = \arg \max_{1 \leq j \leq c} \mathbf{F}_{ij}^*$.

1. No primeiro passo definimos o grafo G formado por todos os n pontos, e construímos a sua matriz de adjacência $\mathbf{W}$ através do núcleo Gaussiano. A distância entre dois pontos $\mathbf{x}_i$ e $\mathbf{x}_j$, $\text{dist}(\mathbf{x}_i, \mathbf{x}_j)$ é comumente considerada a distância Euclidiana, e a métrica é a do cosseno para problemas envolvendo classificação de textos. O parâmetro σ é um parâmetro livre no intervalo $[0, 1]$.
2. No segundo passo constrói-se a matriz $\mathcal{W} = \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2}$, onde $\mathbf{D} = \text{diag}(\mathbf{d})$ é uma matriz diagonal com entradas $\mathbf{d}_i = \sum_j w_{ij}$, e $\mathbf{W} = (w_{ij})$ é a matriz de pesos.
3. No terceiro passo itera-se $\mathbf{F}(t+1) = \alpha \mathcal{W} \mathbf{F}(t) + (1 - \alpha) \mathbf{Y}$ até a convergência, onde $\alpha \in (0, 1)$.
4. Finalmente no quarto passo se $\mathbf{F}^*$ é o limite da sequência $\{\mathbf{F}(t)\}$, rotula-se cada $\mathbf{x}_i$ com $f_i = \arg \max_{1 \leq j \leq c} \mathbf{F}_{ij}^*$.

Pode-se provar (ZHOU *et al.*, 2004) que a sequência $\{\mathbf{F}(t)\}$ converge para $\mathbf{F}^* = (\mathbf{I} - \alpha \mathcal{W})^{-1} \mathbf{Y}$. O algoritmo acima é conhecido abreviadamente por **LLGC**.

Vale ressaltar que o auto-reforço é evitado já que os elementos da diagonal de $\mathcal{W}$ são ajustados para zero no primeiro passo. Além disso, a informação é espalhada simetricamente desde que $\mathcal{W}$ seja uma matriz simétrica. Finalmente, o rótulo de cada ponto não rótulado é definido para ser a classe da qual ele recebeu mais informações durante o processo de iterações.

A estrutura de regularização do algoritmo iterativo acima é:

$$\mathbf{F}^* = \arg \min_{\mathbf{F}} \frac{1}{2} \left\{ \sum_{i,j}^n w_{ij} \left\| \frac{\mathbf{F}_i}{\sqrt{\mathbf{D}_{ii}}} - \frac{\mathbf{F}_j}{\sqrt{\mathbf{D}_{jj}}} \right\|^2 + \mu \sum_{i=1}^n \|\mathbf{F}_i - \mathbf{Y}_i\|^2 \right\}, \quad (4.2)$$

onde μ é o parâmetro de regularização que controla o equilíbrio entre a perda e a suavidade dos rótulos.

4.1.2 Como definir os pontos de incerteza

Seja $\mathcal{S}$ o conjunto de amostragem para o qual o algoritmo de propagação (**LLGC**) converge para $\mathbf{F}^*$. Visto que $\mathbf{F}_{ij}^*$ representa o nível de informação que o ponto $\mathbf{x}_i$ recebe da classe j , definimos o vetor que representa a probabilidade de cada ponto $\mathbf{x}_i$ pertencer à classe na qual foi classificado, como:

$$\mathbf{p} = \frac{1}{\|\tilde{\mathbf{p}}\|_{\infty}} \tilde{\mathbf{p}}, \quad (4.3)$$

onde

$$\tilde{\mathbf{p}} = \left[\sum_{j=1}^c \mathbf{F}_{1j}^*, \sum_{j=1}^c \mathbf{F}_{2j}^*, \dots, \sum_{j=1}^c \mathbf{F}_{nj}^* \right]^T. \quad (4.4)$$

A partir da definição anterior podemos inferir que os pontos com uma classificação muito incerta são aqueles cujas coordenadas correspondentes do vetor $\mathbf{p}$ estão muito próximas de 0. Por outro lado, se a coordenada i do vetor $\mathbf{p}$ estiver muito próxima de 1 então o nível de certeza da classificação do ponto $\mathbf{x}_i$ é alto.

Neste trabalho propõe-se uma estrutura de aprendizado semi-supervisionado ativo e iterativo, que tem como hipótese que pontos que são classificados com uma certeza menor podem ajudar a melhorar a performance do algoritmo semi-supervisionado, quando considerados na amostra $\mathcal{S}$ durante o processo iterativo.

Definição 11. O **percentil de ordem** K (onde K é qualquer valor entre 0 e 100), denotado por P_k , é o valor tal que $K\%$ dos valores do conjunto de dados são menores ou iguais a ele.

Definição 12. Define-se a região de incerteza de classificação dos pontos como sendo o conjunto de todos os pontos com nível de certeza não maior que o percentil 30 do vetor $\mathbf{p}$, ou seja:

$$\mathbf{I} = \{i : \mathbf{p}(i) \leq P_{30}\}$$

onde P_{30} é o percentil de ordem 30 do vetor $\mathbf{p}$ e $\mathbf{I}$ é a região de incerteza.

4.1.3 Modificação da matriz de pesos

O algoritmo de amostragem proposto em (ANIS; GADDE; ORTEGA, 2016) adiciona ao conjunto $\mathcal{S}$ (amostra), os vértices que maximizam a frequência de corte a cada iteração.

Seja $\mathbf{I} = \{i_1, i_2, \dots, i_m\}$ um conjunto de m índices da região de incerteza de um grafo $G = (\mathcal{V}, \mathcal{E})$. Define-se uma matriz $\mathbf{W}^{\mathbf{I}}$, como:

$$\mathbf{W}_{ij}^{\mathbf{I}} = \begin{cases} w_{ij} + \|\mathbf{W}\|_{\infty}, & \text{se } i \in \mathbf{I}, \forall j \in \mathbf{I}, i \neq j; \\ w_{ij}, & \text{caso contrário.} \end{cases} \quad (4.5)$$

Este operador é simplesmente a adição de cada elemento de $\mathbf{W}$ da linha e coluna i (índice para o qual se pretende aumentar a chance de ser escolhido na amostra) com o maior elemento da matriz $\mathbf{W}$.

O operador $\mathbf{W}_{ij}^{\mathbf{I}}$ é definido com o objetivo de aumentar as chances dos vértices com índices em $\mathbf{I}$ serem considerados na amostra $\mathcal{S}$, ao mesmo tempo conservando a relação simétrica entre outros vértices.

4.1.4 Como selecionar pontos das regiões de incerteza para amostra

Figura 19 – Gráficos de dispersão do conjunto de dados *two moons* com 1200 pontos. Em (a) tem-se o conjunto de dados que se pretende classificar, o conjunto tem duas classes, representadas pelas cores azul e roxa. Em (b) tem-se a classificação dos dados pelo algoritmo de propagação de rótulos LLGC. Em (c) a representação do vetor $\mathbf{p}$ no conjunto de dados. Os pontos marcados por x são os que foram inicialmente classificados.

Fonte: Elaborada pelo autor.

A figura 19 apresenta um gráfico de dispersão de um conjunto de pontos de $\mathbb{R}^2$ que é conhecido como *two moons*. Foi representado também gráfico de dispersão da classificação dos pontos e o vetor (ou sinal) $\mathbf{p}$, que representa as regiões de incerteza. As regiões com classificação mais incerta tem a cor azul segundo a escala de cor na figura 19 (c).

Na figura 19 (c) nota-se que o nível de incerteza aumenta quando afasta-se de uma amostra classificada. A proposta é maximizar as regiões um com bom nível de classificação.

Para maximizar estas regiões precisa-se escolher amostras não próximas da região de boa classificação, ou seja escolher amostras das regiões de cor azul na figura 19 (c). Para definir os pontos a serem escolhidos nas regiões de incerteza nos auxiliamos nas medidas de posição estatística, os percentis.

Propõe-se a escolha dos vértices da região de incerteza pelo operador de amostragem Ψ (definição 10). Porém deve-se modificar os pesos da matriz de pesos $\mathbf{W}$, pelo operador da equação 4.5.

4.2 O algoritmo proposto

Definição 13. Seja $\mathbf{p}$ o vetor que representa o nível de incerteza de cada ponto na iteração i , representamos este vetor por $\mathbf{p}_{(i)}$.

Proposta. – Algoritmo ativo para melhorar a da classificação semi-supervisionada

- 1: **Entrada:** Matriz de similaridades dos dados $\mathbf{W}$, iter (número de iterações)
 - 2: **Saída:** Classificação do conjunto de dados.
 - 3: Calcule $\mathcal{S}_0 = \Psi(\mathbf{W}, k, M_0)$
 - 4: Classifique os dados de $\mathcal{S}_0$
 - 5: Propague os rótulos de $\mathcal{S}_0$
 - 6: **enquanto** $i < iter$
 1. Calcule o conjunto de índices $\mathbf{I}_i$ de região de classificação incerta que deseja adicionar a $\mathcal{S}_i$ (30% dos índices mais incertos de $\mathbf{p}_{(i-1)}$)
 2. Calcule $\mathbf{W}_{ij}^{\mathbf{I}_i}$
 3. Calcule $\mathcal{S}_i = \Psi(\mathbf{W}_{ij}^{\mathbf{I}_i}, k, M_i)$
 4. $\mathcal{S} = \mathcal{S}_{i-1} \cup \mathcal{S}_i$
 5. Avalie a classificação obtida com a amostra $\mathcal{S}$.
-

Observação 4. A cada iteração do algoritmo descrito acima adicionamos novos vértices ao conjunto de amostragem. Observe que $\mathcal{S}_i = \mathcal{S}_{i-1} \cup \mathcal{S}_i$, sendo que uma porcentagem de vértices de $\mathcal{S}$ são vértices que apresentam muita incerteza na classificação da iteração $i - 1$. A porcentagem dos vértices de maior incerteza na amostra $\mathcal{S}$ não deve ser muito grande, pois pode prejudicar a performance de classificação na iteração i .

A hipótese central do nosso aprendizado semi-supervisionado é de que a rotulagem de alguns pontos, que apresentam muita incerteza na iteração i , pode melhorar de forma significativa a classificação na iteração $i + 1$.

Para testar a nossa hipótese, foram efetuados experimentos numéricos, com dados sintéticos e reais, são apresentados no próximo capítulo.

RESULTADOS E TRABALHOS FUTUROS

Neste capítulo são apresentados resultados numéricos da técnica de aprendizado semi-supervisionado proposta. A performance da técnica proposta foi avaliada usando dados sintéticos e reais: *Breast Cancer Wisconsin*, *MNIST* e *20 newsgroup*.

Compara-se a proposta da seleção de amostras contra três estratégias de amostragem, nomeadamente, o algoritmo de amostragem proposto por [Anis, Gadde e Ortega \(2016\)](#) sem a modificação da matriz de pesos, a amostragem aleatória e considera-se também a escolha de amostras com classificação mais incerta a cada iteração.

A performance de cada técnica de amostragem será representada por uma curva em um gráfico. A curva de cor azul escuro representa a performance do algoritmo proposto. A curva de cor vermelha representa a performance do algoritmo proposto por [Anis, Gadde e Ortega \(2016\)](#). A curva de cor azul céu representa a performance da amostragem aleatória e a curva de cor cinza representa a performance da amostragem que considera as regiões mais incertas em cada iteração.

Os parâmetros dos algoritmos de amostragem e de propagação considerados, estão resumidos na tabela 3.

Tabela 2 – Descrição dos conjuntos de dados usados

Conjunto de dados	Tamanho	Número de classes	Dimensão
Sintéticos (<i>Moom data set</i>)	1000	2	2
MNIST	46000	10	784
20 Newsgroups	20000	20	63760
Breast Cancer Wisconsin	569	2	32

Tabela 3 – Resumo dos parâmetros utilizados durante os experimentos numéricos

Conjunto de dados	k	α	σ	Similaridade
Sintéticos	10	0.1	0.1	Euclidiana
MNIST	10	0.1	0.1	Euclidiana
20 Newsgroups	16	0.1	**	Cosseno
Breast Cancer Wisconsin	16	0.1	0.1	Euclidiana

5.1 Conjunto sintético

Nesta secção, a proposta de aprendizado semi-supervisionado ativo é avaliada usando um conjunto de pontos sintéticos. A avaliação é feita usando um conjunto de pontos $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{1000}\} \subset [-2, 3] \times [-1, 2]$ sintéticos, com duas classes. Um gráfico de dispersão do conjunto de dados é apresentado na figura 20.

A matriz de pesos foi obtida usando o núcleo Gaussiano, com a distância Euclidiana. Os parâmetros α , σ e k , foram tomados como 0.1, 0.1 e 10 respectivamente.

Foram realizados experimentos envolvendo o parâmetro k para estudar a estabilidade numérica do processo de amostragem. A conclusão é que valores muito grandes de k causam instabilidade numérica no processo de amostragem.

A figura 21 mostra que escolher as amostras usando o aprendizado ativo proposto é mais vantajoso do que escolher as amostras aleatoriamente em cada iteração, ou escolher usando apenas o algoritmo de seleção proposto por (ANIS; GADDE; ORTEGA, 2016). No trabalho (GADDE; ANIS; ORTEGA, 2014) foi mostrado que esta técnica supera outras técnicas de seleção, nomeadamente *LLGC error bound minimization* (GU; HAN, 2012), LLR (ZHANG *et al.*, 2011) e METIS (KARYPIS; KUMAR, 1998). Neste trabalho modifica-se a técnica de seleção proposta por (ANIS; GADDE; ORTEGA, 2016), para que considere as amostras que foram classificadas com incerteza durante a classificação. Mostra-se que usando esta estratégia para a classificação semi-supervisionada é melhor que usar o algoritmo de amostragem proposto por (ANIS; GADDE; ORTEGA, 2016) ou escolher a amostra aleatória por iteração.

Figura 20 – Gráfico de dispersão dos dados sintéticos usados na avaliação numérica do aprendizado ativo semi-supervisionado.

Fonte: Elaborada pelo autor.

Figura 21 – Resultado da comparação da proposta de aprendizado ativo com as amostragens aleatória, e a baseada em *proxies*.

Fonte: Elaborada pelo autor.

5.2 Conjunto de dados MNIST

Neste experimento avalia-se a performance do aprendizado ativo proposto para a classificação do conjunto de dígitos manuscritos MNIST¹. Os dígitos manuscritos contêm 10 classes, os dígitos de 0 – 9. O MNIST contém aproximadamente 60000 amostras. Para este experimento

¹ <https://www.kaggle.com/oddrational/mnist-in-csv>

cada imagem digital foi normalizada para conter $28 \times 28 = 784$ pixels. O grafo é construído usando o núcleo Gaussiano

$$\mathbf{W}_{ij} = \begin{cases} \exp\left(-\frac{\text{dist}(\mathbf{x}_i, \mathbf{x}_j)^2}{2\sigma^2}\right), & \text{se } i \neq j; \\ 0, & \text{se } i = j \end{cases} \quad (5.1)$$

com a distância euclidiana. Os parâmetros σ e k foram fixados como 0.1 e 10 respectivamente.

O resultado de 5.1 é um grafo totalmente conexo, com uma matriz de adjacência simétrica. Usando o grafo construído anteriormente, selecionamos ativamente os nós pelo algoritmo proposto e avaliamos a sua performance em comparação com a seleção aleatória, com a amostragem proposta no trabalho (ANIS; GADDE; ORTEGA, 2016) e com a amostragem dos pontos mais incertos por iteração. Os resultados são ilustrados na figura 22. Observa-se que a nossa proposta supera as técnicas supracitadas, principalmente quando se tem poucas amostras rotuladas.

Figura 22 – Comparação de técnicas de aprendizado semi-supervisionado ativo para o conjunto de dados MNIST. O eixo vertical indica a porcentagem de dados rotulados por iteração na propagação de rótulos e o eixo horizontal indica a respectiva acurácia obtida por iteração.

Fonte: Elaborada pelo autor.

A acurácia média deste teste foi calculada em 10 tentativas. Os resultados da média da acurácia estão resumidos na figura 23. Foram escolhidas 2000 amostras aleatórias em cada tentativa.

Figura 23 – Acurácia média da classificação do conjunto de dados MNIST, pelas técnicas de aprendizado ativo semi-supervisionado.

Fonte: Elaborada pelo autor.

5.3 Classificação de textos

Para a classificação de textos foi usado o conjunto de dados 20 *newsgroup*². Este conjunto de dados contém cerca de 20.000 textos, particionados em 20 grupos diferentes. Para os experimentos envolvendo este conjunto de dados foram usados 4 grupos diferentes, nomeadamente, '*comp.os.ms-windows.misc*', '*alt.atheism*', '*comp.graphics*' e '*soc.religion.christian*'. Totalizando um corpo de 2255 textos.

Os dados foram pré-processados removendo palavras ou artigos muito usados em cada documento, como “a”, “de”, “o”, “da” entre outras. Para formar os vetores de característica de cada documento foi usada a estatística (TF-IDF). A estatística TF-IDF captura a importância relativa de uma palavra em um documento, de um corpo de documentos:

$$TF - IDF = (1 + \log(TF)) \times \log\left(\frac{n}{IDF}\right), \quad (5.2)$$

onde TF é a frequência de uma palavra em um documento, IDF é o número de documentos em que a palavra aparece e n é o número total de documentos. Nestas condições cada documento é representado por um vetor de dimensão 63760. A similaridade ($dist(\mathbf{x}_i, \mathbf{x}_j)$) entre dois textos será calculada como:

$$dist_{ij} = \begin{cases} \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2}, & \text{se } i \neq j; \\ 0, & \text{se } i = j. \end{cases} \quad (5.3)$$

² <http://qwone.com/~jason/20Newsgroups/>

Os resultados do aprendizado semi-supervisionado ativo estão resumidos na figura 24. Estes resultados mostram que a proposta pode melhorar a performance do algoritmo semi-supervisionado de forma significativa, principalmente quando os recursos de rotulação são limitados. É possível notar que a proposta supera significativamente as técnicas em comparação, mostrando desta forma que os vértices considerados pela proposta podem ser considerados na amostra $\mathcal{S}$.

Figura 24 – Comparação de técnicas de aprendizado semi-supervisionado ativo para o conjunto de dados 20 news group.

Fonte: Elaborada pelo autor.

5.4 Efeito do parâmetro k do algoritmo de amostragem

No trabalho (GADDE; ANIS; ORTEGA, 2014) é feito o estudo do efeito que o parâmetro k tem na acurácia do método semi-supervisionado apresentado por Gadde, Anis e Ortega (2014). Nesta seção estuda-se o efeito que este parâmetro tem na proposta apresentada anteriormente. Para a escolha do melhor k foi reproduzido o experimento da figura 22, com diferentes valores de k . Os resultados resumidos na figura 25 mostram que o aumento excessivo do parâmetro k é prejudicial durante a propagação de rótulos. O motivo de um k muito grande prejudicar a classificação é que este cria uma instabilidade no cálculo dos autovetores na maximização da frequência de corte durante a computação da amostra ótima. Porém as figuras 25 e 26 confirmam a hipótese do trabalho (ANIS; GADDE; ORTEGA, 2014) de que na classificação de imagens da base de dados MNIST um k relativamente pequeno (entre 1 e 10) produz uma classificação melhor que k muito grandes. Devido à dimensão muito alta de dados dos problemas de classificação de textos, valores grandes deste parâmetro são benéficos à classificação de textos. Esta hipótese mantém-se válida para a nossa proposta de aprendizado ativo.

Figura 25 – Efeito do parâmetro k na acurácia, para a classificação do algoritmo LLGC. O gráfico mostra a acurácia média de diferentes porcentagens dos dados (MNIST).

Fonte: Elaborada pelo autor.

Figura 26 – Efeito do parâmetro k na acurácia, para a classificação do algoritmo LLGC. O gráfico mostra a acurácia média de diferentes porcentagens dos dados *20 news*.

Fonte: Elaborada pelo autor.

A figura 26 mostra a instabilidade numérica criada por k muito grande na computação da frequência de corte no conjunto de dados MNIST.

5.5 Conjunto de dados de câncer Mama (*Breast Cancer Data*)

O câncer de mama é uma doença bastante comum entre as mulheres de todo o mundo, os casos de câncer de mama têm aumentado significativamente nos últimos anos de acordo com as estatísticas globais. O diagnóstico desta doença pode melhorar o prognóstico e a chance de sobrevivência da vítima, no caso do resultado ser verdadeiro positivo (paciente com câncer), pois o paciente pode ser submetido a um tratamento clínico. Uma classificação bem precisa dos tumores benignos pode prevenir que pacientes sejam submetidos a tratamentos desnecessários. Assim, o diagnóstico correto do câncer de mama é de interesse de todos. Os algoritmos de aprendizado de máquina podem ser úteis na classificação de diagnósticos médicos. Em particular nesta seção avalia-se a performance do algoritmo ativo semi-supervisionado proposto neste trabalho, no diagnóstico do câncer de mama.

O conjunto de dados utilizado foi criado pelo Dr William H. Wolberg, médico do hospital de Wiscosin nos EUA. O conjunto de dados Brest cancer data³ contém 596 instâncias com 32 atributos, e duas classes que são câncer benigno (B), câncer maligno (M).

Neste experimento, a distância entre os pontos é a distância euclidiana, e $\sigma = 0.99$. Os parâmetros k , α foram considerados 16 e 0.1 respectivamente. Os resultados deste experimento são mostrados na figura 27, que confirmam que a técnica proposta, supera a estratégia aleatória e a estratégia de amostragem proposta por (GADDE; ANIS; ORTEGA, 2014). Em termos quantitativos temos que a partir da decima primeira iteração o algoritmo proposto acerta 12 diagnósticos a mais comparado com a amostragem proposta por Gadde, Anis e Ortega (2014) (Proxies espectrais) e pelo menos 29 diagnósticos certos a mais que a amostragem aleatória.

Pode-se verificar que a estratégia proposta é mais robusta a rótulos inconsistentes comparativamente as outras estratégias visitadas nesta dissertação.

5.6 Análise da complexidade

Nesta seção analisa-se a complexidade do algoritmo de seleção considerado. O passo mais complexo do algoritmo guloso proposto no trabalho (ANIS; GADDE; ORTEGA, 2016) é a computação do menor auto-par de $(\mathcal{L}^k)_{\mathcal{S}^c}$ durante a maximização da frequência de corte. Existe muitos métodos para determinar o menor auto-par de uma matriz, entre estes se destaca a minimização do quociente de Rayleigh (KNYAZEV, 2001). Sendo que para todos estes métodos o passo mais complexo é a computação do produto de matriz por vetor. No caso do algoritmo de amostragem, este passo consiste em determinar $(\mathcal{L}^k \mathbf{x})_{\mathcal{S}^c}$. Porém os grafos de aplicação de aprendizado de máquina tipicamente tem Laplacianos esparsos, o que resulta em implementações eficientes de $\mathcal{L}^k \mathbf{x}$. Se $|\mathcal{L}^k|$ denota o número de zeros de $\mathcal{L}^k$, então a

³ <https://www.kaggle.com/uciml/breast-cancer-wisconsin-data>

Figura 27 – Resumo da performance do aprendizado ativo proposto, para o conjunto de dados de câncer de mama. O eixo horizontal indica a porcentagem que foi escolhida durante determinada iteração, e o eixo vertical indica a acurácia da classificação.

Fonte: Elaborada pelo autor.

complexidade do produto matriz-vetor é $\mathcal{O}(|\mathcal{L}^k|)$. Portanto a complexidade para computar cada auto-par de $\mathcal{L}^k \mathbf{x}$ é $\mathcal{O}(k|\mathcal{L}^k|r)$, onde r é o número médio de iterações para o algoritmo LOPCG (KNYAZEVA, 2001). Contudo a complexidade do algoritmo de seleção passa a ser $\mathcal{O}(k|\mathcal{L}^k|mr)$, onde m é o tamanho da amostra.

O algoritmo de propagação de rótulos merece uma atenção especial no passo 3. Neste passo a tarefa mais complexa é a multiplicação de duas matrizes $\mathcal{W}$ e $\mathbf{F}$, o que tem custo computacional $\mathcal{O}(n^2c)$, onde n é a dimensão da matriz quadrada $\mathcal{W}$, e c o número de classes. No mesmo passo temos a multiplicação da matriz $\mathcal{W}$ pelo escalar α com complexidade $\mathcal{O}(n^2)$ e a multiplicação do escalar $1 - \alpha$ por $\mathbf{Y}$, que é $\mathcal{O}(nc)$. Portanto a complexidade da propagação de rótulos por iteração é $\mathcal{O}(n^2c + n^2 + nc)$.

5.7 Conclusão e Trabalhos Futuros

Neste trabalho foi proposto um algoritmo semi-supervisionado ativo, a ser usado para a rotulação e classificação de um conjunto de dados. As amostras a serem supervisionadas são escolhidas a partir do algoritmo de amostragem do trabalho (ANIS; GADDE; ORTEGA, 2016). A classificação é feita pelo algoritmo de propagação de rótulos LLGC (ZHOU *et al.*, 2004). A proposta consiste em modificar a matriz de similaridades do conjunto de dados para que a cada iteração da escolha dos pontos a supervisionar sejam considerados pontos de classificação incerta na amostra.

Para propor o algoritmo semi-supervisionado ativo desta dissertação foram conduzidos

os seguintes estudos:

- Primeiramente generaliza-se conceitos da teoria de processamento de sinal clássico, a transformada de Fourier (**GFT**), filtros, translações e convoluções. Sempre que possível são apresentados exemplos e ilustrações, para uma melhor compreensão destes conceitos.
- Estuda-se a amostragem de sinais no domínio de grafos, considera-se grafos não direcionados e conexos. Faz-se uma análise das técnicas de amostragem propostas anteriormente. Estuda-se a técnica de amostragem proposta por [Anis, Gadde e Ortega \(2016\)](#).
- Estuda-se também a teoria de aprendizado de máquina semi-supervisionado, estabelecendo uma ligação entre a amostragem e os algoritmos semi-supervisionados.
- Definimos as regiões de classificação incerta e apresentamos uma técnica de seleção de instâncias que utiliza o algoritmo baseado em *proxies* e considera pontos de classificação incerta através da modificação da matriz de similaridade dos dados. Para além de considerar os pontos de incerteza na amostra, propõe-se a diminuição da influência destes pontos durante a propagação de rótulos.
- Apresenta-se resultados experimentais que mostram que a nossa proposta melhora significativamente a acurácia do algoritmo de propagação de rótulos **LLGC** quando comparado com as técnicas de seleção comuns.

Foi concluído que a proposta de aprendizado semi-supervisionado ativo é mais eficaz porque apresenta uma acurácia que supera os outros métodos de seleção de amostras. Mostrou-se, através de experimentos com dados reais e sintéticos que a proposta apresenta uma performance muito boa em comparação com os métodos do estado da arte. Durante os experimentos realizados foi possível concluir que a proposta pode superar as técnicas do estado da arte por até 20% de acurácia. As figuras [23](#) e [24](#) mostram o resumo da performance da proposta e os métodos do estado da arte quando aplicados aos conjuntos *MNIST* e *20 News*.

Como trabalho futuro espera-se apresentar uma estimativa mínima do tamanho de amostras necessárias para a obtenção da acurácia desejada em algoritmos semi-supervisionados. Seria útil também estudar a região de incerteza, ou seja estudar até que percentil deve ser considerada a região de incerteza para o vetor **p**. Na dissertação considera-se até percentil 30 de **p** como região de incerteza. Contudo durante os experimentos foi possível notar que existe um *trade-off* entre a porcentagem escolhida e a propagação do erro de classificação. Portanto seria interessante estudar uma forma de considerar uma porcentagem diferente de pontos de classificação incerta a cada iteração do algoritmo semi-supervisionado proposto.

REFERÊNCIAS

ANIS, A.; GADDE, A.; ORTEGA, A. Towards a sampling theorem for signals on arbitrary graphs. p. 3864–3868, 2014. Citado nas páginas [29](#), [41](#), [42](#), [55](#), [56](#), [57](#), [58](#) e [76](#).

_____. Efficient sampling set selection for bandlimited graph signals using graph spectral proxies. **IEEE Transactions on Signal Processing**, IEEE, v. 64, n. 14, p. 3775–3789, 2016. Citado nas páginas [11](#), [13](#), [27](#), [28](#), [29](#), [31](#), [39](#), [42](#), [55](#), [56](#), [58](#), [65](#), [68](#), [71](#), [72](#), [74](#), [78](#), [79](#) e [80](#).

AVRON, H.; BOUTSIDIS, C. Faster subset selection for matrices and applications. **SIAM Journal on Matrix Analysis and Applications**, SIAM, v. 34, n. 4, p. 1464–1499, 2013. Citado na página [41](#).

BASU, S.; BANERJEE, A.; MOONEY, R. J. Active semi-supervision for pairwise constrained clustering. In: SIAM. **Proceedings of the 2004 SIAM international conference on data mining**. [S.l.], 2004. p. 333–344. Citado nas páginas [31](#), [34](#) e [39](#).

BELKIN, M.; NIYOGI, P. Semi-supervised learning on riemannian manifolds. **Machine learning**, Springer, v. 56, n. 1-3, p. 209–239, 2004. Citado na página [27](#).

BENNETT, K. P.; CAMPBELL, C. Support vector machines: hype or hallelujah? **Acm Sigkdd Explorations Newsletter**, Acm, v. 2, n. 2, p. 1–13, 2000. Citado na página [33](#).

BENNETT, K. P.; DEMIRIZ, A. Semi-supervised support vector machines. In: **Advances in Neural Information processing systems**. [S.l.: s.n.], 1999. p. 368–374. Citado na página [33](#).

BISHOP, C. M. **Pattern recognition and machine learning**. [S.l.]: springer, 2006. Citado nas páginas [32](#) e [33](#).

BLUM, A.; MITCHELL, T. Combining labeled and unlabeled data with co-training. In: CITESEER. **Proceedings of the eleventh annual conference on Computational learning theory**. [S.l.], 1998. p. 92–100. Citado na página [35](#).

BOUTSIDIS, C.; DRINEAS, P.; MAGDON-ISMAIL, M. Near-optimal column-based matrix reconstruction. **SIAM Journal on Computing**, SIAM, v. 43, n. 2, p. 687–717, 2014. Citado na página [41](#).

CHAMON, L. F.; RIBEIRO, A. Greedy sampling of graph signals. **IEEE Trans. Signal Process.**, v. 66, n. 1, p. 34–47, 2018. Citado na página [42](#).

CHAPELLE, O.; SINDHWANI, V.; KEERTHI, S. S. Optimization techniques for semi-supervised support vector machines. **Journal of Machine Learning Research**, v. 9, n. Feb, p. 203–233, 2008. Citado na página [36](#).

CHEN ROHAN VARMA, A. S. J. K. S. Discrete signal processing on graphs: Sampling theory. **IEEE Transactions on Signal Processing**, v. 63. Citado nas páginas [25](#) e [26](#).

CHEN, S.; SANDRYHAILA, A.; MOURA, J. M.; KOVACEVIC, J. Signal recovery on graphs. **IEEE Trans. Signal Process**, 2014. Citado na página [26](#).

- CHEN, S.; VARMA, R.; SANDRYHAILA, A.; KOVAČEVIĆ, J. Discrete signal processing on graphs: Sampling theory. *IEEE transactions on signal processing*, IEEE, v. 63, n. 24, p. 6510–6523, 2015. Citado nas páginas 26, 27, 40 e 42.
- COL, A. D.; VALDIVIA, P.; PETRONETTO, F.; DIAS, F.; SILVA, C. T.; NONATO, L. G. Wavelet-based visual analysis of dynamic networks. *IEEE transactions on visualization and computer graphics*, IEEE, v. 24, n. 8, p. 2456–2469, 2018. Citado na página 26.
- COZMAN, F. G.; COHEN, I.; CIRELO, M. C. Semi-supervised learning of mixture models. p. 99–106, 2003. Citado na página 36.
- DANZIGER, S. A.; BARONIO, R.; HO, L.; HALL, L.; SALMON, K.; HATFIELD, G. W.; KAISER, P.; LATHROP, R. H. Predicting positive p53 cancer rescue regions using most informative positive (mip) active learning. *PLoS computational biology*, Public Library of Science, v. 5, n. 9, p. e1000498, 2009. Citado na página 26.
- DASGUPTA, S. The two faces of active learning. In: **Discovery science**. [S.l.: s.n.], 2009. p. 35. Citado na página 39.
- DONG, X.; FROSSARD, P.; VANDERGHEYNST, P.; NEFEDOV, N. Clustering on multi-layer graphs via subspace analysis on grassmann manifolds. *IEEE Transactions on signal processing*, IEEE, v. 62, n. 4, p. 905–918, 2014. Citado na página 26.
- FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. d. L. *et al.* Inteligência artificial: Uma abordagem de aprendizado de máquina. 2011. Citado na página 33.
- GADDE, A.; ANIS, A.; ORTEGA, A. Active semi-supervised learning using sampling theory for graph signals. p. 492–501, 2014. Citado nas páginas 15, 26, 39, 40, 42, 43, 59, 62, 72, 76 e 78.
- GADDE, A.; ORTEGA, A. A probabilistic interpretation of sampling theory of graph signals. In: IEEE. **2015 IEEE international conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2015. p. 3257–3261. Citado na página 27.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [S.l.]: MIT press, 2016. Citado na página 25.
- GU, Q.; HAN, J. Towards active learning on graphs: An error bound minimization approach. In: IEEE. **Data Mining (ICDM), 2012 IEEE 12th International Conference on**. [S.l.], 2012. p. 882–887. Citado nas páginas 27, 42, 59 e 72.
- GUILLORY, A.; BILMES, J. A. Active semi-supervised learning using submodular functions. **arXiv preprint arXiv:1202.3726**, 2012. Citado nas páginas 43 e 59.
- HAMMOND, D. K.; VANDERGHEYNST, P.; GRIBONVAL, R. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, Elsevier, v. 30, n. 2, p. 129–150, 2011. Citado nas páginas 51, 52, 53 e 54.
- HAND, D. J. Statistical analysis of network data: Methods and models by eric d. kolaczyk. **International Statistical Review**, Wiley Online Library, v. 78, n. 1, p. 135–135, 2010. Citado nas páginas 26 e 40.
- HASAN, M. A.; ZAKI, M. J. Output space sampling for graph patterns. **Proceedings of the VLDB Endowment**, VLDB Endowment, v. 2, n. 1, p. 730–741, 2009. Citado na página 31.

HASENJÄGER, M.; RITTER, H. Active learning of the generalized high-low-game. In: SPRINGER. **International Conference on Artificial Neural Networks**. [S.l.], 1996. p. 501–506. Citado na página 39.

HIGHAM, N. J. **Functions of matrices: theory and computation**. [S.l.]: Siam, 2008. v. 104. Citado na página 54.

HINTON, G.; DENG, L.; YU, D. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, *ieee signal processing magazine*, pp. 82-97. 18 10. 2012. Citado na página 25.

HU, R.; NAMEE, B. M.; DELANY, S. J. Off to a good start: Using clustering to select the initial training set in active learning. In: **Twenty-Third International FLAIRS Conference**. [S.l.: s.n.], 2010. Citado na página 39.

KARYPIS, G.; KUMAR, V. A fast and high quality multilevel scheme for partitioning irregular graphs. **SIAM Journal on scientific Computing**, SIAM, v. 20, n. 1, p. 359–392, 1998. Citado na página 72.

KNYAZEY, A. V. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. **SIAM journal on scientific computing**, SIAM, v. 23, n. 2, p. 517–541, 2001. Citado nas páginas 78 e 79.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 1097–1105. Citado na página 25.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015. Citado na página 25.

LI, J.; SANDER, J.; CAMPELLO, R.; ZIMEK, A. Active learning strategies for semi-supervised dbscan. In: SPRINGER. **Canadian Conference on Artificial Intelligence**. [S.l.], 2014. p. 179–190. Citado na página 39.

LI, M.; ZHANG, H.; WU, R.; ZHOU, Z.-H. Sample-based software defect prediction with active and semi-supervised learning. **Automated Software Engineering**, Springer, v. 19, n. 2, p. 201–230, 2012. Citado na página 31.

LUXBURG, U. v.; SCHOLKOPF, B. Statistical learning theory: Models. **Concepts, and Results**, 2008. Citado na página 31.

MALLAPRAGADA, P. K.; JIN, R.; JAIN, A. K. Active query selection for semi-supervised clustering. In: IEEE. **2008 19th International Conference on Pattern Recognition**. [S.l.], 2008. p. 1–4. Citado na página 39.

MITCHELL, T. M. The role of unlabeled data in supervised learning. In: **Language, Knowledge, and Representation**. [S.l.]: Springer, 2004. p. 103–111. Citado na página 31.

NARANG, S. K.; CHAO, Y. H.; ORTEGA, A. Graph-wavelet filterbanks for edge-aware image processing. In: IEEE. **Statistical Signal Processing Workshop (SSP), 2012 IEEE**. [S.l.], 2012. p. 141–144. Citado nas páginas 26 e 40.

NARANG, S. K.; GADDE, A.; ORTEGA, A. Signal processing techniques for interpolation in graph structured data. In: IEEE. **Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on**. [S.l.], 2013. p. 5445–5449. Citado nas páginas 26 e 40.

NARANG, S. K.; GADDE, A.; SANOU, E.; ORTEGA, A. Localized iterative methods for interpolation in graph structured data. In: IEEE. **Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE**. [S.l.], 2013. p. 491–494. Citado na página 42.

NARANG, S. K.; ORTEGA, A. Local two-channel critically sampled filter-banks on graphs. In: IEEE. **Image Processing (ICIP), 2010 17th IEEE International Conference on**. [S.l.], 2010. p. 333–336. Citado na página 40.

_____. Downsampling graphs using spectral theory. In: IEEE. **Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on**. [S.l.], 2011. p. 4208–4211. Citado na página 41.

_____. Perfect reconstruction two-channel wavelet filter banks for graph structured data. **IEEE Transactions on Signal Processing**, IEEE, v. 60, n. 6, p. 2786–2799, 2012. Citado na página 41.

_____. Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs. **IEEE transactions on signal processing**, IEEE, v. 61, n. 19, p. 4673–4685, 2013. Citado na página 41.

NGUYEN, H. Q.; DO, M. N. Downsampling of signals on graphs via maximum spanning trees. **IEEE Transactions on Signal Processing**, IEEE, v. 63, n. 1, p. 182–191, 2014. Citado na página 41.

_____. Downsampling of signals on graphs via maximum spanning trees. **IEEE Trans. Signal Processing**, v. 63, n. 1, p. 182–191, 2015. Citado na página 41.

OPENSHAW, S. **Artificial intelligence in geography**. [S.l.: s.n.], 1997. Nenhuma citação no texto.

PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs up?: sentiment classification using machine learning techniques. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. **Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10**. [S.l.], 2002. p. 79–86. Citado na página 32.

PERRAUDIN, N.; PARATTE, J.; SHUMAN, D.; MARTIN, L.; KALOFOLIAS, V.; VANDERGHEYNST, P.; HAMMOND, D. K. Gspbox: A toolbox for signal processing on graphs. **arXiv preprint arXiv:1408.5781**, 2014. Citado na página 28.

PESENSON, I. Sampling in paley-wiener spaces on combinatorial graphs. **Transactions of the American Mathematical Society**, v. 360, n. 10, p. 5603–5627, 2008. Citado nas páginas 29, 41, 55 e 56.

SANDRYHAILA, A.; MOURA, J. M. Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. **IEEE Signal Processing Magazine**, IEEE, v. 31, n. 5, p. 80–90, 2014. Citado na página 40.

SANDRYHAILA, J. M. F. M. A. Discrete signal processing on graphs. **IEEE Transactions on Signal Processing**, v. 61, n. 7, p. 1644–1656, 2013. Citado nas páginas 25 e 26.

- SETTLES, B. **Active learning literature survey**. [S.l.], 2009. Citado nas páginas 38 e 39.
- SETTLES, B.; CRAVEN, M.; FRIEDLAND, L. Active learning with real annotation costs. In: VANCOUVER, CA. **Proceedings of the NIPS workshop on cost-sensitive learning**. [S.l.], 2008. p. 1–10. Citado na página 38.
- SHEN, H.; LI, X.; CHENG, Q.; ZENG, C.; YANG, G.; LI, H.; ZHANG, L. Missing information reconstruction of remote sensing data: A technical review. **IEEE Geoscience and Remote Sensing Magazine**, IEEE, v. 3, n. 3, p. 61–85, 2015. Citado na página 26.
- SHOMORONY, I.; AVESTIMEHR, A. S. Sampling large data on graphs. **arXiv preprint arXiv:1411.3017**, 2014. Citado nas páginas 31 e 42.
- SHUMAN, D. I.; FARAJI, M. J.; VANDERGHEYNST, P. A multiscale pyramid transform for graph signals. **IEEE Transactions on Signal Processing**, IEEE, v. 64, n. 8, p. 2119–2134, 2016. Citado nas páginas 51 e 52.
- SHUMAN, D. I.; RICAUD, B.; VANDERGHEYNST, P. A windowed graph fourier transform. In: IEEE. **Statistical Signal Processing Workshop (SSP), 2012 IEEE**. [S.l.], 2012. p. 133–136. Citado na página 53.
- SHUMAN SUNIL K. NARANG, P. F. A. O. D. I.; VANDERGHEYNST, P. The emerging field of signal processing on graphs: Extending highdimensional data analysis to networks and other irregular domains. **IEEE Signal Processing Magazine**, Institute of Electrical and Electronics Engineers, v. 30, n. 3, p. 83–98, 2013. Citado nas páginas 26, 41 e 54.
- STUTZBACH, D.; REJAIE, R.; DUFFIELD, N.; SEN, S.; WILLINGER, W. Sampling techniques for large, dynamic graphs. In: IEEE. **Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications**. [S.l.], 2006. p. 1–6. Citado na página 31.
- TRAFALIS, T. B.; GILBERT, R. C. Robust classification and regression using support vector machines. **European Journal of Operational Research**, Elsevier, v. 173, n. 3, p. 893–909, 2006. Citado na página 32.
- TREMBLAY, N.; BORGNAT, P. Graph wavelets for multiscale community mining. **IEEE Transactions on Signal Processing**, IEEE, v. 62, n. 20, p. 5227–5239, 2014. Citado na página 26.
- TSITSVERO, M.; BARBAROSSA, S.; LORENZO, P. D. Signals on graphs: Uncertainty principle and sampling. **IEEE Transactions on Signal Processing**, IEEE, v. 64, n. 18, p. 4845–4860, 2016. Citado na página 42.
- UEFFING, N.; HAFFARI, G.; SARKAR, A. Transductive learning for statistical machine translation. In: **Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics**. [S.l.: s.n.], 2007. p. 25–32. Citado na página 36.
- VALDIVIA, P.; DIAS, F.; PETRONETTO, F.; SILVA, C. T.; NONATO, L. G. Wavelet-based visualization of time-varying data on graphs. In: IEEE. **Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on**. [S.l.], 2015. p. 1–8. Citado na página 26.
- VETTERLI, M.; KOVAČEVIĆ, J.; GOYAL, V. K. Foundations of signal processing. In: . [S.l.]: Cambridge University Press, 2014. Citado na página 48.

WANG, X.; LIU, P.; GU, Y. Local-set-based graph signal reconstruction. **IEEE Transactions on Signal Processing**, IEEE, v. 63, n. 9, p. 2432–2444, 2015. Citado nas páginas 26 e 59.

WANG, X.; WANG, M.; GU, Y. A distributed tracking algorithm for reconstruction of graph signals. **IEEE Journal of Selected Topics in Signal Processing**, IEEE, v. 9, n. 4, p. 728–740, 2015. Citado na página 56.

WEI, K.; LIU, Y.; KIRCHHOFF, K.; BILMES, J. Using document summarization techniques for speech data subset selection. In: **Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. [S.l.: s.n.], 2013. p. 721–726. Citado na página 25.

WEISS, Y.; TORRALBA, A.; FERGUS, R. Spectral hashing. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2009. p. 1753–1760. Citado na página 25.

WITTEN, I. H.; FRANK, E.; HALL, M. A.; PAL, C. J. **Data Mining: Practical machine learning tools and techniques**. [S.l.]: Morgan Kaufmann, 2016. Citado na página 33.

XU, Z.; YU, K.; TRESP, V.; XU, X.; WANG, J. Representative sampling for text classification using support vector machines. In: SPRINGER. **European conference on information retrieval**. [S.l.], 2003. p. 393–407. Citado na página 39.

YAROWSKY, D. Unsupervised word sense disambiguation rivaling supervised methods. In: **33rd annual meeting of the association for computational linguistics**. [S.l.: s.n.], 1995. p. 189–196. Citado na página 34.

ZHANG, L.; CHEN, C.; BU, J.; CAI, D.; HE, X.; HUANG, T. S. Active learning based on locally linear reconstruction. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 33, n. 10, p. 2026–2038, 2011. Citado nas páginas 27, 34, 39, 42, 59 e 72.

ZHOU, D.; BOUSQUET, O.; LAL, T. N.; WESTON, J.; SCHÖLKOPF, B. Learning with local and global consistency. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2004. p. 321–328. Citado nas páginas 11, 13, 27, 28, 29, 37, 43, 65, 66 e 79.

ZHOU, X.; BORÉN, J.; AKYÜREK, L. M. Filamins in cardiovascular development. **Trends in cardiovascular medicine**, Elsevier, v. 17, n. 7, p. 222–229, 2007. Citado na página 36.

ZHOU, Z.-H.; CHEN, K.-J.; JIANG, Y. Exploiting unlabeled data in content-based image retrieval. In: SPRINGER. **European Conference on Machine Learning**. [S.l.], 2004. p. 525–536. Citado na página 38.

ZHU, X.; RABBAT, M. Graph spectral compressed sensing for sensor networks. In: IEEE. **Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on**. [S.l.], 2012. p. 2865–2868. Citado nas páginas 26 e 40.