Classificação de Alto Nível Baseada em Redes Complexas para Aprendizado Multirrótulo

Vinícius Henrique Resende

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2021

Vinícius Henrique Resende

Classificação de Alto Nível Baseada em Redes Complexas para Aprendizado Multirrótulo

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Murillo Guimarães Carneiro

Uberlândia
2021

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

R433 2021

Resende, Vinicius Henrique, 1996-
Classificação de Alto Nível Baseada em Redes Complexas para Aprendizado Multirrótulo [recurso eletrônico] / Vinicius Henrique Resende. - 2021.

Orientador: Murillo Guimarães Carneiro.
Dissertação (Mestrado) - Universidade Federal de Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.di.2021.121
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Carneiro, Murillo Guimarães,1988-, (Orient.). II. Universidade Federal de Uberlândia. Pós-graduação em Ciência da Computação. III. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:

Gizele Cristine Nunes do Couto - CRB6/2091

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Ciência da Computação
Av. João Naves de Ávila, nº 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Mestrado Acadêmico, 3/2021, PPGCO				
Data:	19 de fevereiro de 2021	Hora de início:	09:00	Hora de encerramento:	11:50
Matrícula do Discente:	11912CCP026				
Nome do Discente	Vinícius Henrique Resende				
Título do Trabalho:	Classificação de Alto Nível Baseada em Redes Complexas para Aprendizado Multirrótulo				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Inteligência Artificial				
Projeto de Pesquisa de vinculação:	-				

Reuniu-se, por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Bruno Augusto Nassif Travençolo - FACOM/UFU; Ricardo Marcondes Marcacini - ICMC/USP e Murillo Guimarães Carneiro - FACOM/UFU, orientador do candidato.

Os examinadores participaram desde as seguintes localidades: Ricardo Marcondes Marcacini - São Carlos/SP; Bruno Augusto Nassif Travençolo e Murillo Guimarães Carneiro - Uberlândia/MG. O discente participou da cidade de Uberlândia/MG.

Iniciando os trabalhos o presidente da mesa, Prof. Dr. Murillo Guimarães Carneiro, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado.

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Murillo Guimarães Carneiro**, **Professor(a) do Magistério Superior**, em 19/02/2021, às 14:41, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Bruno Augusto Nassif Travençolo**, **Professor(a) do Magistério Superior**, em 19/02/2021, às 17:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **RICARDO MARCONDES MARCACINI**, **Usuário Externo**, em 24/02/2021, às 10:48, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2571710** e o código CRC **9A06823A**.

Dedico este trabalho ao leitor.

Agradecimentos

Agradeço a Deus e a todos que me apoiaram direta ou indiretamente durante esta etapa da minha vida. Em específico os meus pais, Silvânia e Fábio, por todo apoio que me foi dado durante toda a minha vida. Nâo poderia deixar de agradecer os meus amigos, Anna Paula, Arthur Ribeiro, Beatriz Moura, Lucas Bissaro, Lusmar Mendes, Matheus Prandini, Tiago Silva e Yuri Cardoso por todos os momentos de descontração que tornaram esta jornada mais leve.

Agradeço também, o meu orientador Prof. Murillo Carneiro, por sua orientação incrível, sempre me motivando e me inspirando a fazer tudo da melhor forma possível.

Deixo meu agradecimento também aos membros da banca de defesa, Prof. Bruno Travençolo e Prof. Ricardo Marcacini, por todos comentários e sugestões pertinentes proferidos durante a arguição.

Por fim, agradeço à CAPES pelo apoio financeiro que foi fundamental para minha permanência no mestrado.

"Descobrir consiste em olhar para o que todo mundo está vendo e pensar uma coisa diferente".

(Roger Von Oech)

Resumo

A classificação de dados é um dos tópicos mais importantes de aprendizado de máquina (AM) e tem como objetivo automatizar problemas de categorização ou predição, atribuindo uma classe (ou rótulo) que caracteriza cada instância do problema abordado. Os algoritmos de classificação tradicionais (ou monorrótulo) assumem que cada instância é associada a uma única classe, entretanto, muitos problemas do mundo real podem ser relacionados a múltiplos rótulos simultaneamente, como por exemplo, a anotação de imagens contendo múltiplos objetos. Por se tratar de uma extensão da classificação monorrótulo, a maioria dos algoritmos de aprendizado multirrótulo (AMR) são baseados em técnicas da classificação tradicional, herdando suas vantagens mas também suas limitações. Em relação às limitações, a maioria das técnicas de classificação monorrótulo possuem o processo de aprendizado guiado apenas por características físicas dos dados (e.g., distância ou distribuição) e ignoram relacionamentos semânticos e estruturais dos dados, como por exemplo, formação de padrão. Recentemente, diversos trabalhos têm utilizado conceitos de redes complexas a fim de capturar relacionamentos estruturais e topológicos dos dados (i.e., características de alto nível) e consequentemente melhorar seus resultados. Inspirado pelo uso emergente de redes complexas no AM, este trabalho investiga um novo método baseado em redes complexas para o AMR, trazendo novas técnicas de modelagem do problema multirrótulo para a forma de rede, além de uma abordagem híbrida capaz de considerar tanto aspectos físicos quanto topológicos dos dados ao combinar redes complexas com técnicas de AMR tradicional. Experimentos realizados em bases de dados artificiais e reais demonstram a capacidade da técnica de alto nível em detectar múltiplos padrões nos dados e em virtude disso aprimorar a habilidade preditiva das técnicas de AMR tradicionais. Mais importante, este trabalho abre caminho para novas pesquisas sobre redes complexas para AMR.

Palavras-chave: Redes complexas, classificação de alto nível, aprendizado multirrótulo, aprendizado de máquina, medidas de redes complexas, classificação de dados.

Abstract

Data classification is one of the most important topics in machine learning (ML) and aims to automate discrete learning tasks by assigning a class (or label) that characterizes each instance of the problem addressed. Traditional classification algorithms (or single-label) assume that each instance is associated with a single class, however, many real-world problems can be related to multiple labels simultaneously, such as the image annotation with multiple objects. As it is an extension of the single-label classification, most of the multi-label learning algorithms (MLL) are based on traditional classification techniques, inheriting their advantages but also their limitations. In relation to the limitations, most single-label classification techniques have a learning process guided only by physical characteristics of the data (e.g., distance or distribution) and ignore semantic and structural relationships of the data, such as pattern formation. Recently, several researches on ML have employed concepts of complex networks in order to capture structural and topological relationships of the data (i.e., high-level characteristics) and consequently improve their results. Inspired by the emerging usage of complex networks in ML, this dissertation investigates new methods based on complex networks for MLL, presenting new techniques for modeling the multi-label problem into a network as well as a new hybrid approach able to consider both physical and topological aspects of the data by combining complex networks with traditional MLL techniques. Experiments performed on artificial and real-world databases demonstrate the ability of the high-level technique to detect multiple patterns in the data and, as a result, improve the predictive performance of traditional MLL techniques. Moreover, this work paves a way to new developments based on complex networks to MLL.

Keywords: Complex networks, high-level classification, multi-label learning, machine learning, complex networks measures, data classification.

Lista de ilustrações

Lista de tabelas

Lista de abreviações

AM Aprendizado de Máquina

BR Binary Relevance

CC Classifier Chain

CD Critical Difference

CXN-MLL CompleX Networks for Multi-Label Learning

ϵ**N** Rede vizinhança de raio ϵ

kNN-G Rede k vizinhos mais próximos

SkNN Rede k vizinhos mais próximos seletiva

D-kNN Rede vizinhos mais próximos de grau k

HL High Level

LL Low Level

NB Naive Bayes

RF Random Forest

SVM Support Vector Machine

Sumário

Capítulo 1

Introdução

O aprendizado supervisionado (AS) é um dos principais paradigmas do aprendizado de máquina (AM) e tem como objetivo resolver de forma automática problemas de predição e categorização de dados (CARNEIRO, 2017). De forma breve, no AS um modelo é treinado para mapear um conjunto de dados de entrada as suas saídas correspondentes (HASTIE; TIBSHIRANI; FRIEDMAN, 2009), em que os valores que as saídas podem assumir são previamente conhecidos e definem o tipo de problema que será tratado – se tais saídas assumem valores contínuos, temos um problema de *regressão*, se discretos chamamos de *classificação* (BISHOP, 2006).

Na classificação tradicional (ou monorrótulo), a saída (também conhecida como classe ou rótulo) pode assumir um único valor para cada objeto de entrada, sendo inviável a sua aplicação em vários problemas do mundo real em que as instâncias podem assumir múltiplas saídas simultaneamente. Para contornar tal problema, faz-se necessário o estudo e desenvolvimento de algoritmos de aprendizado *multirrótulo*.

O aprendizado multirrótulo tem sido um tópico amplamente estudado nos últimos anos devido a sua grande variedade de aplicações. Alguns exemplos são: classificação de imagens (BOUTELL et al., 2004), classificação de texto (KATAKIS; TSOUMAKAS; VLAHAVAS, 2008), classificação funcional de genes (ELISSEEFF; WESTON, 2002), diagnóstico de doenças (PESTIAN et al., 2007), etc. A Fig. 1 mostra um problema que técnicas tradicionais (monorrótulo) não conseguem resolver apropriadamente devido a sua limitação em relação à saída; um objeto de entrada (livro) que pode assumir múltiplas saídas (gêneros). Dentre os principais desafios do aprendizado multirrótulo estão o tamanho do espaço de saída, pois se na classificação monorrótulo existe uma quantidade linear de possibilidades de rótulos, na classificação multirrótulo este número é exponencial; e a dependência de rótulos, em que um rótulo pode estar associado ou ter influência sobre outro rótulo (ZHANG; ZHOU, 2014).

Figura 1 – Exemplo de problema multirrótulo que exibe um livro de múltiplas classificações de gênero.

1.1 Motivação

A grande maioria das técnicas de aprendizado multirrótulo ou adaptam ou usam algoritmos de classificação tradicional como base do processo de aprendizado, trazendo consigo suas vantagens e desvantagens. Uma dessas desvantagens é a dificuldade em considerar relações semânticas muitas vezes escondidas na topologia ou estrutura dos dados. Vários trabalhos da literatura (SILVA; ZHAO, 2012; SILVA; ZHAO, 2015; CUPERTINO et al., 2018; CARNEIRO; ZHAO, 2018; CARNEIRO et al., 2019) mostraram que as técnicas de classificação tradicional possuem essa limitação por considerarem apenas as características físicas dos dados (i.e., distância, similaridade ou distribuição) no processo de classificação. Nesse sentido, o estudo apresentado em (RESENDE; CARNEIRO, 2019) apontou evidências de que algoritmos representativos de classificação multirrótulo sofrem com esse mesmo problema, sendo incapazes de detectar, por exemplo, a formação de padrão nos dados.

Uma das principais ferramentas para modelagem e análise de informações estruturais e topológicas dos dados e que tem sido um tópico de pesquisa importante para vários campos da ciência são as Redes Complexas (NEWMAN, 2018). Entre as suas principais características destaca-se a habilidade de modelar relações e dinâmicas de sistemas complexos e heterogêneos como, por exemplo, a Internet, redes biológicas e redes sociais (NEWMAN, 2003; FORTUNATO, 2010). Também tem sido cada vez mais comum o desenvolvimento de técnicas baseadas em rede para tarefas relacionadas à transformação,

processamento e visualização de dados, as quais costumam envolver aplicações comuns de AM, tais como detecção de comunidades, redução de dimensionalidade, propagação de rótulo e classificação de dados (SCARSELLI et al., 2008; BACKES; CASANOVA; BRUNO, 2009; FORTUNATO, 2010; SILVA; ZHAO, 2012; CARNEIRO et al., 2017; SANTOS et al., 2020; LINHARES et al., 2017; MUSCOLONI et al., 2017; CARNEIRO; ZHAO, 2018; CUPERTINO et al., 2018).

Ainda que o uso de redes complexas em AM seja crescente, alguns cenários ainda carecem de maior atenção por parte dos pesquisadores, especialmente pelas vantagens que a representação em rede pode permitir. Por exemplo, alguns métodos de classificação baseada em redes são capazes de capturar relacionamentos semânticos e estruturais dos dados a partir de associações derivadas de medidas de redes complexas (SILVA; ZHAO, 2012; SILVA; ZHAO, 2015; CUPERTINO et al., 2018; CARNEIRO; ZHAO, 2018; CARNEIRO et al., 2019), as quais muitas vezes são ignoradas pelos algoritmos tradicionais de classificação, essencialmente baseados nas características físicas dos dados. Para se ter um exemplo, a Fig. 2 mostra uma base de dados artificial com 2 classes (triângulos vermelhos e círculos verdes), cada uma com um padrão de formação distinto. Os quadrados pretos são os objetos de teste os quais algoritmos tradicionais têm dificuldade em identificar suas respectivas classes. Tais algoritmos são denominados de baixo nível, uma vez que consideram apenas os atributos físicos dos dados (SILVA; ZHAO, 2012). Na figura, tal dificuldade pode ser explicada pela baixa densidade de objetos da classe verde, fazendo com que o algoritmo não consiga identificar o seu padrão de formação.

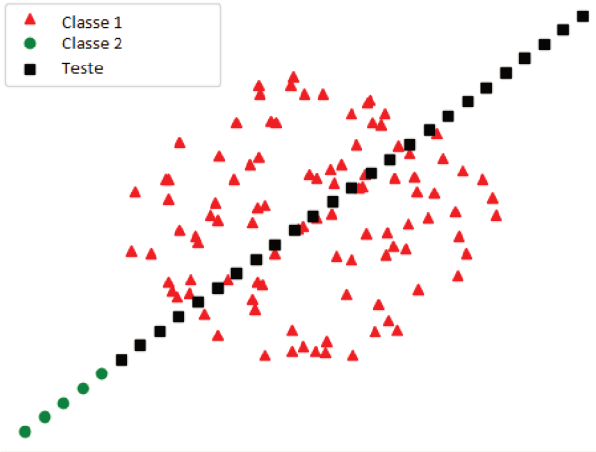


Figura 2 – Conjunto de dados com padrões de formação que algoritmos tradicionais possuem dificuldade em identificar corretamente.

Estudos recentes (SILVA; ZHAO, 2015; CUPERTINO et al., 2018; CARNEIRO; ZHAO, 2018; CARNEIRO et al., 2019), apresentaram técnicas de alto nível para contornar o problema, onde todas são exclusivamente para classificação monorrótulo. Em comum, essas técnicas usam redes complexas para extrair características de alto nível após converter os dados (normalmente apresentados na forma de vetor de atributos) em um

grafo. O termo alto nível se refere à habilidade da técnica de considerar características e relacionamentos estruturais e abstratos que vão além de simples comparações físicas dos dados. Um exemplo de problema que necessita de uma classificação de alto nível é o conjunto de dados apresentado na Fig. 2, em que o mesmo possui múltiplos padrões de formação que dificilmente serão capturados de forma correta por técnicas de baixo nível.

Nesse sentido, essa pesquisa almeja investigar modelos de classificação de alto nível. O tema possui relação com tópicos do estado-da-arte do aprendizado em redes, tais como comitês de classificadores, aprendizado profundo, aprendizado de representação e redes neurais de grafos. Os comitês de classificadores (também denominados *ensemble*), por exemplo, compreendem uma classe de algoritmos e estratégias para combinação de classificadores com o propósito de obter um modelo com melhor performance do que qualquer uma das técnicas combinadas (OPITZ; MACLIN, 1999). A principal diferença em relação à classificação de alto nível é que esta última vai além disso por combinar propriedades físicas e estruturais dos dados, ou seja, não se trata de combinar algoritmos diferentes mas as propriedades distintas que eles são capazes de analisar (CARNEIRO, 2017).

O aprendizado profundo (do inglês, *deep learning*), consiste em uma classe de técnicas que usa múltiplas camadas de processamento e que permite aprender representações de dados com múltiplas camadas de abstração (LECUN; BENGIO; HINTON, 2015). Uma das técnicas mais representativas de aprendizado profundo são as redes neurais de múltiplas camadas, capazes de aprender funções extremamente complexas e capturar relacionamentos abstratos entre os dados. Devido a esse fato, podemos dizer que as técnicas de aprendizado profundo também podem capturar e realizar classificações e predições de alto nível. Diferente dessas técnicas, em que as associações de alto nível são derivadas a partir do processamento dos dados em múltiplas camadas sucessivas, a classificação de alto nível é capaz de derivar associações de alto nível simplesmente por focar na análise de aspectos estruturais e topológicos dos dados representados em rede.

Por sua vez, o aprendizado de representação (do inglês, *representation learning*) é um tópico relativamente recente que compreende o desenvolvimento de estratégias para realizar transformações de forma automática em dados brutos (evitando o trabalho manual), criando novas e melhores representações destes dados de forma a aumentar as chances de sucesso das técnicas de AM (BENGIO; COURVILLE; VINCENT, 2013). Embora um dos principais objetivos deste trabalho esteja relacionado à criação de métodos de representação dos dados na forma de rede, esta pesquisa difere um pouco do contexto principal de aprendizado de representação por focar também no desenvolvimento de algoritmos capazes de capturar e analisar as informações de alto nível presentes na rede.

Diferente das redes neurais tradicionais que recebem dados na forma de vetor de atributos, as redes neurais de grafo (do inglês, *graph neural networks*) foram criadas para explorar problemas onde os dados estão originalmente representados na forma de rede, como por exemplo, as redes sociais (WU et al., 2020). Além das redes neurais de grafo,

existem outras técnicas de AM, inclusive AMR, que recebem os dados já na forma de rede e buscam extrair e explorar informações relevantes de tais redes. Entretando, essas técnicas normalmente tomam o caminho contrário dessa pesquisa, buscando levar os dados na forma de rede para o espaço Euclidiano. Um exemplo de algoritmo com este intuito é o node2vec, proposto em (GROVER; LESKOVEC, 2016).

1.2 Hipótese e Objetivos

A hipótese investigada nesta dissertação afirma que a análise de características estruturais e topológicas de dados multirrótulo contribui para melhorar o desempenho preditivo dos modelos de classificação essencialmente baseados nas características físicas dos dados. Dessa forma, o objetivo geral da pesquisa é o desenvolvimento de uma abordagem híbrida que combina um algoritmo multirrótulo de baixo nível com um de alto nível a fim de realizar uma classificação capaz de considerar tanto aspectos físicos quanto topológicos dos dados. Os objetivos específicos deste trabalho são:

- ❏ Investigar e desenvolver métodos eficientes de construção do grafo para o contexto multirrótulo, afinal é necessário que os dados estejam bem representados na forma de rede para que as características de alto nível sejam devidamente exploradas.

- ❏ Desenvolver uma técnica capaz de combinar as associações de baixo nível produzidas por algoritmos de classificação multirrótulo tradicionais com associações de alto nível obtidas a partir de medidas de redes complexas.

- ❏ Demonstrar o potencial que medidas e propriedades de redes complexas possuem para aprimorar o desempenho preditivo de algoritmos convencionais de classificação multirrótulo.

1.3 Contribuições

Dentre as contribuições geradas pelo presente trabalho, destacam-se:

- ❏ Um novo algoritmo de classificação multirrótulo capaz de considerar tanto aspectos físicos quanto topológicos dos dados.

- ❏ Avaliação e desenvolvimento de diferentes métodos de construção de grafo para o aprendizado multirrótulo.

- ❏ Avaliação do impacto de diferentes combinações de medidas de redes complexas para o algoritmo proposto.

1.4 Organização da Dissertação

Os demais capítulos da dissertação foram organizados da seguinte forma:

- No Capítulo 2 são revisados os conceitos fundamentais para compreensão do trabalho, abordando tópicos e algoritmos de aprendizado supervisionado mono e multirrótulo, com foco especial no problema de classificação. Também são apresentados os principais conceitos de redes complexas usados no trabalho, além dos principais trabalhos relacionados à pesquisa.
- No Capítulo 3 é apresentado o método proposto, denominado CXN-MLL, uma técnica de alto nível que combina associações produzidas por classificadores de baixo e alto nível. No capítulo também são apresentados métodos de construção do grafo projetados para transformar problemas multirrótulos na forma de vetor de atributos para o formato de rede.
- No Capítulo 4 são apresentados os resultados da técnica proposta em experimentos realizados em bases artificiais e reais, considerando diferentes tipos de análise e métricas de desempenho preditivo.
- O Capítulo 5 faz as considerações finais do trabalho, discutindo os principais achados da dissertação além de apresentar os principais tópicos a serem perseguidos nos trabalhos futuros.

Capítulo 2

Fundamentação e Revisão Bibliográfica

Neste capítulo serão apresentados os conceitos fundamentais e trabalhos relacionados as investigações contempladas pela dissertação. Na seção 2.1 e 2.2 são apresentados respectivamente os problemas de aprendizado monorrótulo e multirrótulo, onde são discutidas algumas das principais técnicas da literatura de cada um dos paradigmas. Na seção 2.3 são apresentados seus conceitos fundamentais, formas de explorar a topologia da rede, e técnicas de AS baseadas em rede.

2.1 Aprendizado Supervisionado Monorrótulo

O aprendizado supervisionado (AS) tem como objetivo aprender uma função f capaz de mapear um objeto de entrada $\mathbf{x}$ para uma saída previamente conhecida $\mathbf{y}$, $f : \mathbf{x} \rightarrow \mathbf{y}$. O AS é dividido em 2 classes de problemas, a classificação (quando $\mathbf{y}$ assume valores discretos) e a regressão (quando $\mathbf{y}$ assume valores contínuos). Para construção e avaliação de um modelo, seja de regressão ou classificação, temos 2 fases essenciais chamadas de treino e teste. Na fase de treino, é fornecido ao modelo um conjunto de dados para serem utilizados no seu processo de aprendizagem $X = \{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$, sendo $\mathbf{x}_i = \{a_1, ..., a_d\}$ um vetor de d dimensões que descreve $\mathbf{x}_i$ enquanto $\mathbf{y}_i$ representa sua saída correspondente. No problema de classificação os valores de $\mathbf{y}$ são finitos $1 \leq \mathbf{y} \leq |L|$ onde L é o conjunto de classes do problema. Na fase de teste, $f(\cdot)$ pode ser usada para predizer a saída de novos objetos onde $\mathbf{y}$ é desconhecido, i.e., $(\mathbf{x}, ?)$.

Para exemplificar, um problema de classificação é a detecção de spam em e-mails, onde x é um conjunto de dados descritores do e-mail (ex: assunto, texto do e-mail, etc.) e y é um valor binário que caracteriza o e-mail ($0 =$ não spam, $1 =$ spam). No caso da regressão, podemos utilizá-la para predizer a probabilidade de chuva, onde x é um conjunto de dados representando características do clima (ex: umidade do ar, clima, temperatura, etc) e y é a probabilidade de chover (ex: 80.5%).

Neste trabalho, será abordado em específico o problema de classificação, um dos tópicos mais importantes de IA. Nas próximas seções serão discutidos alguns exemplos populares

de algoritmos de AS monorrótulo.

2.1.1 Classificador Ingênuo de Bayes

Para entender o classificador ingênuo de Bayes, do inglês *Naive Bayes* (NB), primeiro é necessário entender o teorema de Bayes, muito conhecido na estatística e aprendizado de máquina no cálculo de probabilidade de eventos. Imagine dois eventos, A e B, onde a probabilidade de A acontecer é dada por $P(A)$ e a probabilidade de B acontecer é dado por $P(B)$. O teorema de Bayes nos permite calcular a probabilidade de um evento dado que um outro evento aconteceu ($P(A \mid B)$ representa a probabilidade de A acontecer dado que B aconteceu), chamamos isto de probabilidade condicional, que é justamente o que o teorema nos permite calcular.

Para calcular $P(A \mid B)$, temos:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \tag{1}$$

Baseado no teorema apresentado anteriormente, agora podemos entender o classificador Naive Bayes (NB), um dos mais populares de AM devido sua simplicidade e eficiência para resolver problemas de diversos domínios (LEWIS, 1998).

Basicamente, o algoritmo assume que todo o vetor de atributos é independente e utiliza inferência bayesiana para predizer a classe das amostras. Para predizer a classe y de um objeto $\mathbf{x}$ é preciso resolver a seguinte equação:

$$P(y \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid y) \times P(y)}{p(\mathbf{x})} \tag{2}$$

porém, não é possível obter $P(y \mid \mathbf{x})$ diretamente, diferente de $P(\mathbf{x} \mid y)$

$$P(\mathbf{x} \mid y) = \prod_{j=1}^{d} P(x_j \mid y) \tag{3}$$

por fim, assumindo que os valores x_j são independentes, temos:

$$P(y \mid \mathbf{x}) = \frac{P(y) \times \prod_{j=1}^{d} P(x_j \mid y)}{P(\mathbf{x})} \tag{4}$$

a classe de $\mathbf{x}$ é obtida por $argmax\, p(y \mid \mathbf{x})$.

A complexidade do NB é linear $\mathcal{O}(nd)$ na fase de treinamento e $\mathcal{O}(d|L|)$ na fase de teste, o que significa que pode ser usado para dados de alta dimensão.

2.1.2 Máquina de Vetores de Suporte

Uma máquina de vetores de suporte, também conhecidas como *Support Vector Machine* (SVM) é uma técnica poderosa de AM que alcança resultados muito satisfatórios,

em alguns casos superando até mesmo técnicas do estado da arte, como por exemplo as redes neurais artificiais (HEARST et al., 1998).

O objetivo do algoritmo é separar as classes do problema da melhor forma possível, construindo uma margem de separação máxima. Para explicar o algoritmo, consideremos um problema binário onde $y_i = 1$ significa que um objeto $\mathbf{x_i} \in X$ esta associado à classe do problema e $y_i = -1$ não. Para encontrar um hiperplano que separa os dados é preciso satisfazer a seguinte equação:

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = 0 \tag{5}$$

onde $\mathbf{w}$ é o vetor normal ao hiperplano e $\frac{b}{||\mathbf{w}||}$ é a distância do hiperplano à origem (LORENA; CARVALHO, 2007).

Para encontrar a margem máxima, precisamos antes encontrar dois hiperplanos paralelos (vamos chamar aqui de H_1 e H_2) que separam os dados e possuem distância máxima entre si. Podemos definir H_1 como $\mathbf{w} \cdot \mathbf{x} + b = 1$ (onde tudo acima possui $y = 1$) e H_2 como $\mathbf{w} \cdot \mathbf{x} + b = -1$ (onde tudo abaixo possui $y = -1$). A distância d entre H_1 e H_2 pode ser obtida calculando $\frac{2}{||w||}$, que converte o problema de encontrar a margem máxima em um problema de otimização, onde é preciso minimizar o denominador $||w||$ considerando $y_i((\mathbf{w} \cdot \mathbf{x_i}) + b \geq 1)$.

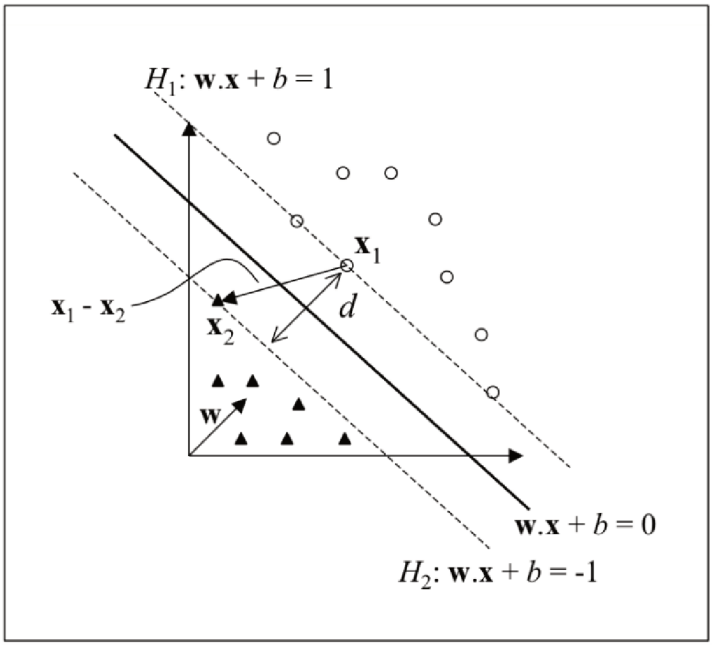


Figura 3 – Ilustração do processo de encontrar a margem máxima da SVM (LORENA; CARVALHO, 2007).

A abordagem discutida nesta seção foi desenvolvida para tratar problemas binários e linearmente separáveis (i.e., um único hiperplano consegue separar todos os dados em $y_i = 1$ e $y_i = -1$), entretando, existem diversas adaptações para resolver problemas com múltiplas classes e não lineares.

2.1.3 Floresta Aleatória

Floresta aleatória ou *Random Forest* (RF), originalmente proposta por Breiman (2001), é uma técnica de AS robusta baseada em árvore de decisão (*decision tree* (DT)) muito popular. O objetivo é construir uma coleção de classificadores estruturados em árvore de forma que cada árvore tem direito a um voto na decisão da classe mais popular. Diferente das DTs, o RF cria as árvores escolhendo aleatoriamente diferentes subconjuntos de características, trazendo mais diversidade e robustez ao modelo.

O algoritmo pode ser resumido em algumas etapas:

1. **Criar um conjunto de dados bootstrap**: nesta etapa, são selecionados amostras de forma aleatória do conjunto de dados original, permitindo repetição de uma mesma amostra.

2. **Criar árvore de decisão**: usando o conjunto de dados gerado anteriormente, é selecionado um subconjunto de características para a árvore de decisão, onde o atributo que melhor separar os dados será a raiz. O mesmo processo irá se repetir para o restante dos nós, nunca repetindo os atributos já usados na árvore.

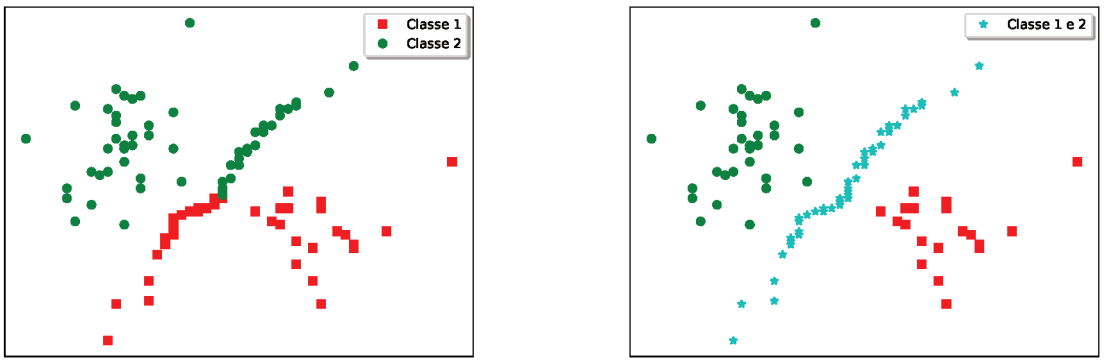
3. **Criar a floresta aleatória**: as etapas 1 e 2 são repetidas várias vezes até formar a quantidade de árvores que foi estabelecida no treinamento.

4. **Classificação**: para classificar uma amostra desconhecida, ela é testada em cada uma das DTs geradas, que farão um esquema de votação. A classe mais votada será atribuída à amostra.

2.2 Aprendizado Supervisionado Multirrótulo

O aprendizado multirrótulo (AMR) é uma derivação do aprendizado supervisionado tradicional (ou monorrótulo) que permite a associação de múltiplas saídas simultâneas a uma mesma entrada (ZHANG; ZHOU, 2014). Representando todas combinações de saída do problema $\alpha = 2^{\mathbf{y}}$, o objetivo é aprender uma função $h : \mathbf{x} \rightarrow \alpha$. Assim como no AS tradicional, o AMR também é dividido em treino e teste. Na fase de treino, é fornecido ao modelo um conjunto de dados de treinamento $\mathcal{X} = \{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$, sendo $\mathbf{x}_i = \{x_1, ..., x_d\}$ um vetor de d dimensões que descreve $\mathbf{x}_i$ e $\mathbf{y}_i = \{y_i^{(1)}, ..., y_i^{(|L|)}\}$ a saída correspondente de $\mathbf{x}_i$. Na classificação multirrótulo, $y_i^{(l)} = 0$ significa que o objeto x_i não está associado à classe l e $y_i^{(l)} = 1$ o contrário. A Figura 4 apresenta um conjunto de dados visto tanto no contexto de aprendizado monorrótulo (a), onde todos objetos são associados à uma única classe (1 **ou** 2), quanto no contexto multirrótulo (b), onde as instâncias podem receber mais de um rótulo ao mesmo tempo (classe 1 **e** 2).

Tsoumakas, Katakis e Vlahavas (2009) definiram os algoritmos de classificação multirrótulo em dois grupos, sendo eles, adaptação de algoritmo e transformação do problema.



(a) Classificação Tradicional (b) Classificação Multirrótulo

Figura 4 – Ilustração da diferença entre classificação monorrótulo (tradicional) e multirrótulo.

As técnicas do primeiro grupo foram desenvolvidas através de adaptações dos algoritmos de classificação monorrótulo e resolvem o problema de forma direta (ZHANG; ZHOU, 2014). Enquanto isso, as técnicas do segundo grupo fazem transformação nos dados, normalmente transformando o problema multirrótulo em uma série de problemas de classificação binária ou multi-classe (TSOUMAKAS; KATAKIS; VLAHAVAS, 2009). Nas seções seguintes serão descritos alguns dos algoritmos mais relevantes de cada uma das categorias citadas.

2.2.1 Adaptação de Algoritmo

Nesta seção serão apresentados os algoritmos MLkNN, BP-MLL e Rank-SVM da categoria de adaptação de algoritmo.

2.2.1.1 Multilabel k-Nearest Neighbor

O *Multilabel k-Nearest Neighbor* (ML-kNN) (ZHANG; ZHOU, 2007) é uma técnica de aprendizagem preguiçosa (*lazy learning*) e uma adaptação de um dos algoritmos mais conhecidos de classificação multi-classe, o kNN. A ideia base consiste em utilizar inferência bayesiana na informação dos rótulos contidos na vizinhança de cada item de dado.

Formalmente, sendo $N(\mathbf{x})$ os vizinhos de $\mathbf{x}$, para cada item de teste serão computados a frequência de cada rótulo C_j de sua vizinhança.

$$C_j = \sum_{i \in N(x)} (y_i^{(j)} = 1) \tag{6}$$

Considere H_j como o evento onde x possui o rótulo y_j e $\mathbb{P}(H_j \mid C_j)$ representa a probabilidade a posteriori considerando que x possui exatamente C_j vizinhos com o rótulo

y_j. Considere também $\mathbb{P}(\neg H_j \mid C_j)$ a probabilidade que x não esteja associado ao rótulo y_j e também possui exatamente C_j vizinhos com y_j. De acordo com o princípio *maximum a posteriori* (MAP), o objeto x vai ser associado a y_j se $\mathbb{P}(H_j \mid C_j) > \mathbb{P}(\neg H_j \mid C_j)$, ou seja, $\frac{\mathbb{P}(H_j|C_j)}{\mathbb{P}(\neg H_j|C_j)} > 1$ (ZHANG; ZHOU, 2014).

2.2.1.2 Rank-SVM

O algoritmo *Rank-SVM* (ELISSEEFF; WESTON, 2002) é uma adaptação do classificador binário *Support Vector Machine* (SVM). A técnica consiste em uma abordagem direta ao problema multirrótulo otimizando um conjunto de classificadores lineares para minimizar uma função de perda. Além disso, a técnica consegue tratar casos não lineares resolvendo um problema dual de programação quadrática através de um truque no kernel (ZHANG; ZHOU, 2014). No processo de atribuição dos rótulos a um item de dado desconhecido, é utilizado um procedimento chamado *stacking-style*, que irá definir uma função de threshold para o algoritmo.

Além do algoritmo discutido anteriormente, recentemente foi proposta uma técnica similar, também baseada em SVM, chamada *Multi-Label Twin Support Vector Machine* (MLTSVM) (CHEN et al., 2016). A ideia base da versão linear do algoritmo é criar $|L|$ hiperplanos na fase de treino, onde o l-ésimo hiperplano fica mais próximo dos dados com a classe l e mais distante do restante, além disso, é utilizada a estratégia *one-against-all* (OAA) para criar múltiplos hiperplanos não paralelos e explorar a informação multirrótulo.

2.2.1.3 BP-MLL

O *Backpropagation for Multi-Label Learning* (BP-MLL) (ZHANG; ZHOU, 2006) é uma adaptação de um dos algoritmos mais populares de treinamento do AS, o backpropagation. No BP-MLL, foi introduzida uma nova função de erro para lidar com o problema multirrótulo assim como foram feitas revisões no algoritmo original. A Fig. 5 mostra a arquitetura geral da rede, onde $a_0, \ldots, a_d$ corresponde a um vetor de atributos de dimensão d, $c_1, \ldots, c_Q$ corresponde a saída da rede, cada uma representando uma das Q classes do problema, V sendo os pesos das M camadas ocultas e W representando os pesos das camadas de saída, todas completamente conectadas.

A função de erro do BP-MLL pode ser definida pela Equação 7,

$$E_i = \sum_{j=1}^{Q} (C_j^i - d_j^i) \tag{7}$$

onde C_j^i é a saída atual do objeto x_i para a classe j e d_j^i é a saída esperada.

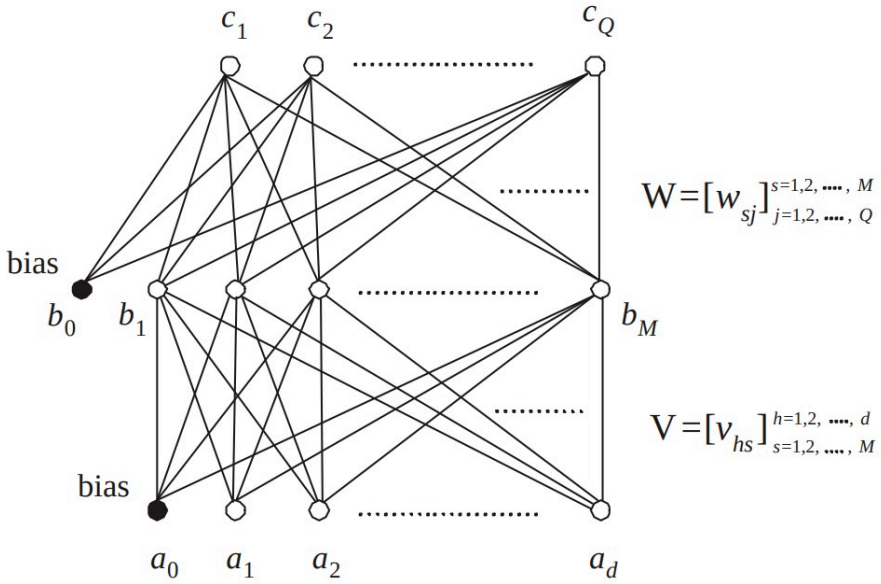


Figura 5 – Arquitetura geral da rede neural BP-MLL (ZHANG; ZHOU, 2006).

2.2.2 Transformação do Problema

Nesta seção, serão apresentados os algoritmos Binary Relevance, Classifier Chain, Label Powerset, ML-Tree e Homer da categoria de transformação do problema.

2.2.2.1 Binary Relevance

O *Binary Relevance* (BR) (TSOUMAKAS; KATAKIS, 2007) é o algoritmo mais clássico da categoria de transformação de problema. A ideia base consiste em dividir a tarefa multirrótulo em $|L|$ problemas independentes de classificação binária. Seja D_j o conjunto de dados para treinamento da classe j e $\mathbf{x}_i$ uma instância do conjunto de treino, a estratégia usada pelo Binary Relevance, chamada de "cross-training" (BOUTELL et al., 2004) irá inserir $(\mathbf{x}_i, y_j)$ em D_j sendo $y_j = +1$ (instância positiva) se $y_j \in Y_i$ e $y_j = -1$ (instância negativa) caso contrário. As Tabelas 1 e 2 exemplificam a transformação feita pelo BR.

Tabela 1 – Conjunto de dados para exemplificação dos métodos de transformação do problema com 3 amostras e suas respectivas classes.

i	$\mathbf{x_i}$	Y_i
1	$[1, 0, 0, 1, 1]$	$[1, 0, 1]$
2	$[0, 0, 1, 1, 0]$	$[0, 0, 1]$
3	$[1, 1, 0, 0, 0]$	$[1, 1, 0]$

Tabela 2 – Exemplo de transformação dos dados feita pelo método Binary Relevance.

	Y
x_1	y_1
x_2	$\neg y_1$
x_3	y_1

	Y
x_1	$\neg y_2$
x_2	$\neg y_2$
x_3	y_2

	Y
x_1	y_3
x_2	y_3
x_3	$\neg y_3$

Em seguida, são treinados $|L|$ classificadores binários, um para cada conjunto gerado; e por fim, na fase de teste, baseados na relevância binária de cada rótulo, são usados cada um dos classificadores treinados para predizer o conjunto de rótulos do objeto de teste.

Por se tratar de um algoritmo que trata cada rótulo com um classificador binário independente, o Binary Relevance pode ser paralelizado, tornando a técnica altamente escalável. Em relação à sua performance, o BR acaba dependendo muito do classificador base utilizado, mas no geral a técnica consegue bons resultados. O principal problema do modelo é a de não considerar a correlação entre os rótulos, assim como todas as técnicas de primeira ordem.

A complexidade do BR é $\mathcal{O}(L \times f(d, N))$ onde $f(d, N)$ é a complexidade do classificador binário escolhido em função do número de atributos (d) e instâncias do conjunto de dados (N).

2.2.2.2 Classifier Chain

O *Classifier Chain* (CC) (READ et al., 2011) é uma variação do BR que considera a correlação entre os rótulos. Assim como o BR, o CC treina $|L|$ classificadores binários, mas dependentes. Na fase de treino, é definida uma sequência aleatória dos classificadores $H = (h_1, \ldots, h_{|L|}) \mid h \in L$, onde cada conjunto D_{h_i} terá o espaço de atributos dos seus dados expandidos com a informação dos rótulos anteriores da sequência $\{h_1, \ldots, h_{i-1}\}$. Na fase de teste, os classificadores são ativados seguindo a mesma ordem definida em H, sempre aumentando o espaço de atributos do teste com o resultado da classificação anterior. A Tabela 3 mostra a diferença dos algoritmos BR e CC.

Tabela 3 – Exemplo da diferença na transformação dos dados feita pelo Binary Relevance e Classifier Chain para o item x_1 da Tabela 1.

(a) Binary Relevance

h :	$x \rightarrow$	y
h_1 :	$[1, 0, 0, 1, 1]$	1
h_2 :	$[1, 0, 0, 1, 1]$	0
h_3 :	$[1, 0, 0, 1, 1]$	1

(b) Classifier Chain

h :	$x' \rightarrow$	y
h_1 :	$[1, 0, 0, 1, 1]$	1
h_2 :	$[1, 0, 0, 1, 1, \mathbf{1}]$	0
h_3 :	$[1, 1, 0, 0, 0, \mathbf{1}, \mathbf{0}]$	1

A principal vantagem do CC em relação ao BR é que ele leva em consideração a

correlação entre os rótulos, entretando, a possibilidade de paralelização do BR é perdida, já que os classificadores são encadeados e seguem a ordem estabelecida em H.

A complexidade do CC é $\mathcal{O}(L \times f(d+L, N))$ onde $f(d+L, N)$ é a complexidade do classificador binário escolhido em função do número de atributos (d), atributos adicionais (rótulos) (L) e instâncias do conjunto de dados (N).

2.2.2.3 Label Powerset

O *Label Powerset* (LP) (TSOUMAKAS; KATAKIS; VLAHAVAS, 2009) transforma o problema multi-rótulo em multi-classe fazendo uma espécie de agrupamento das classes ao transformar cada combinação de rótulos do conjunto de treinamento em uma nova classe única. Feita a transformação, é aplicado então um algoritmo multi-classe que fornecerá as probabilidades para cada classe combinada. Por último, cada rótulo do problema original é considerado por vez, onde são somadas as probabilidade das classes (da transformação) que possuem tal rótulo. A Tabela 4 mostra como é feita a transformação pelo algoritmo LP.

Tabela 4 – Exemplo de transformação feita pelo Label Powerset nas amostras da Tabela 1; $y_{ab}^{(c)}$ representa que a combinação c (que se tornará uma nova classe) possui os rótulos a e b no conjunto de dados original.

i	$\mathbf{x_i}$	Y_i
1	$[1, 0, 0, 1, 1]$	$y_{1,3}^{(1)}$
2	$[0, 0, 1, 1, 0]$	$y_{3}^{(2)}$
3	$[1, 1, 0, 0, 0]$	$y_{1,2}^{(3)}$

A complexidade do LP depende do classificador base escolhido e da quantidade de combinações de rótulos no conjunto de treinamento, que pode ser $min(m, 2^q)$ onde m é o tamanho da base de treino.

2.2.2.4 ML-Tree

O *Multi-Label Tree* (ML-Tree)(WU et al., 2015) é um algoritmo hierárquico baseado em árvore para classificação multirrótulo. A ideia base é representar os dados em forma de árvore, onde o nó raiz guarda todo o conjunto de dados e os demais guardam um subconjunto destes dados, devidamente selecionados por uma SVM e uma medida de pureza. Os nós dos níveis subsequentes da árvore formam um conjunto disjunto que é selecionado através da estratégia *one-against-all* (OAA) nos objetos que ainda não tiveram suas classes definidas naquele nível, i.e., não atingiram pureza máxima $p = 1$ (possui a classe) ou pureza mínima $p = 0$.

A Figura 6 ilustra o processo de criação da árvore, onde p representa a pureza de cada classe e b os rótulos que foram atribuídos ao conjunto de dados de cada nó. Na raiz, nenhum objeto atinge pureza máxima, logo, o nó é particionado pela SVM. No nível seguinte, três novos nós são gerados, onde o primeiro e terceiro nó atingem pureza máxima para as classes y_1 e y_3, enquanto o dados do segundo nó atingem $p = 1$ mas apenas para o nó y_2, rótulo este que será herdado por seus filhos. No último nível da árvore, dois novos nós são gerados pelo OAA (um para cada rótulo onde $p \in (0, 1)$) onde a pureza máxima é atingida em ambos nós.

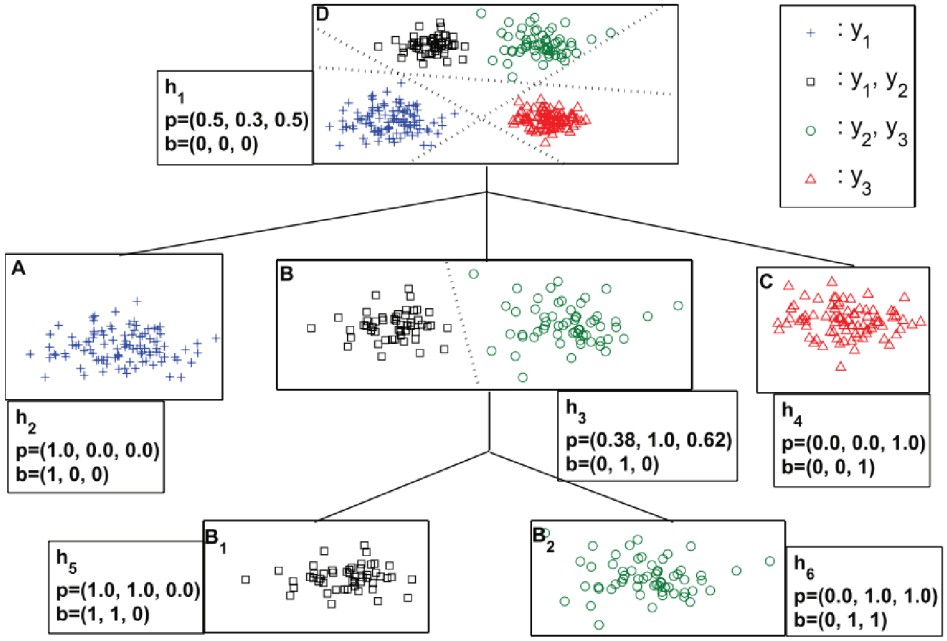


Figura 6 – Ilustração do funcionamento geral do algoritmo ML-Tree (WU et al., 2015).

.

2.2.2.5 HOMER

O HOMER (*Hierarchy Of Multilabel classifiERs*) (TSOUMAKAS; KATAKIS; VLAHAVAS, 2008) é um algoritmo hierárquico de divisão e conquista, desenvolvido especialmente para lidar com problemas que contam com uma grande de quantidade de rótulos em seu domínio. A ideia principal consiste em criar grupos de rótulos similares de modo que são formados k conjuntos disjuntos e de mesmo tamanho.

Inicialmente, o algoritmo constrói a árvore recursivamente fazendo uma busca em profundidade começando pela raiz (onde é armazenado toda a base de dados), onde, para cada nó n são gerados min(k, $|L_n|$) filhos (cada nó representará um conjunto de rótulos), que irão manter apenas os dados que estão associados a pelo menos 1 de seus rótulos. Além disso, um classificador multirrótulo é treinado para cada um dos nós filhos gerados.

A Figura 7 ilustra o processo de criação da árvore. Na raiz, são armazenados todos os dados/rótulos que serão particionados em diferentes conjuntos de mesmo tamanho no

próximo nível. Para tal particionamento, foi utilizado um método chamado de "*balanced k-means*" que fará o agrupamento baseando na similaridade dos dados. Para uma instância não vista, o algoritmo propaga o objeto na árvore usando o classificador treinado para aquele nó, e o repassa para os nós que representam os rótulos que foram preditos.

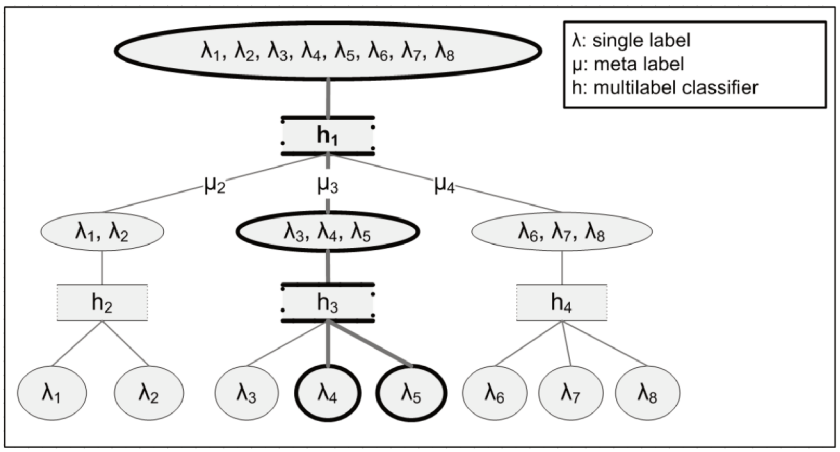


Figura 7 – Ilustração do funcionamento geral do algoritmo HOMER (TSOUMAKAS; KATAKIS; VLAHAVAS, 2008).

2.2.3 Bases de Dados Multirrótulo

Nesta seção serão apresentadas algumas das bases de dados mais conhecidas na literatura de classificação multirrótulo. Tais bases podem ser encontradas em <http://mulan.sourceforge.net/datasets-mlc.html> (TSOUMAKAS et al., 2011).

adicionados mais três conjuntos de dados: Enron, Genbase e Medical.

Tabela 5 – Domínio das bases de dados multirrótulo usadas no trabalho.

Base	Domínio
Birds	Áudio
Emotions	Música
Enron	Texto
Genbase	Biologia
Medical	Texto
Scene	Imagem
Yeast	Biologia

- Na base de dados Birds o problema pode ser resumindo da seguinte forma: dada a gravação de um áudio contendo sons de pássaros, dizer quais espécies de pássaros estavam presentes na gravação (BRIGGS et al., 2012).
- O conjunto de dados Emotions compreende o seguinte problema: dado o timbre e ritmo de uma música, o que sentirá quem a ouve? Os possíveis rótulos (sentimentos) disponíveis no conjunto de dados são: surpresa, felicidade, calma, ficar quieto, tristeza e raiva (TROHIDIS et al., 2008).

- A base Enron (KLIMT; YANG, 2004) consiste de um problema de classificação de texto que busca resolver o seguinte problema: dado um e-mail, qual pasta específica do usuário este e-mail deve ser armazenado ?

- Na base Genbase, o problema pode ser resumido da seguinte forma: dado um conjunto de proteínas, predizer suas classes funcionais.

- A base Medical é uma base de texto que contém relatórios de radiologia e busca associar códigos ICD-9-CM (que funcionam como identificadores para doenças) a tais relatórios.

- O conjunto de dados Scene tem 294 atributos que descrevem uma variedade de imagens. O problema dado por este conjunto de dados é: dada uma imagem, quais conteúdos ela possui ? Exemplos de rótulos (conteúdos) são praia, montanha, campo e pôr do sol (BOUTELL et al., 2004). A Fig. 8 mostra algumas das imagens presentes na base de dados.

(a) Classes: mar, montanhas e campo (b) Classes: árvore e pôr do sol.

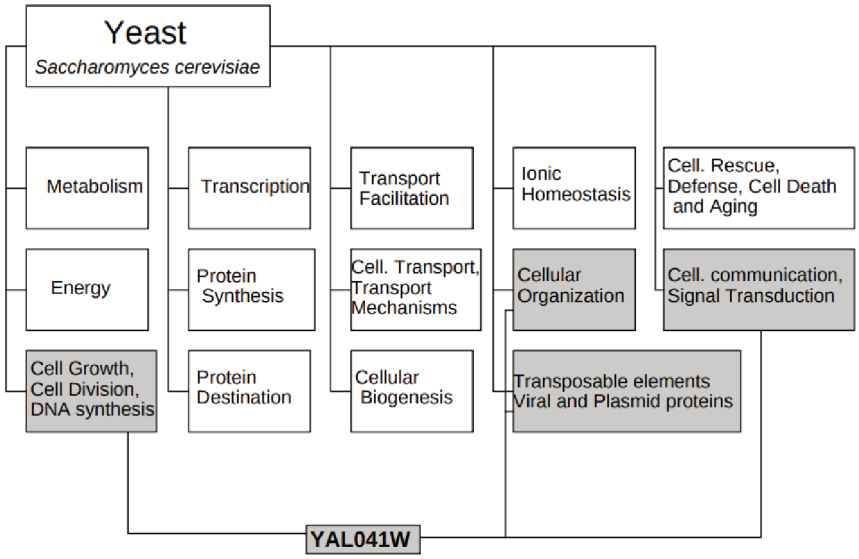
Figura 8 – Exemplos de imagens da base de dados Scene.

- Yeast está relacionado a dados de expressão gênica e perfis filogenéticos produzidos através de microarranjos de genes. O problema a ser abordado neste conjunto de dados é: dada a expressão genética e informação filogenética de uma levedura, quais são suas classes funcionais genéticas? O conjunto de dados tem 14 rótulos para classificação (ELISSEEFF; WESTON, 2002). A Fig. 9 mostra um exemplo de objeto (gene) da base de dados.

2.2.4 Validação, Otimização e Avaliação de Algoritmos

Nesta seção serão apresentados métodos populares na literatura para validação e avaliação de um modelo de classificação.

Figura 9 – Exemplo de amostra da base Yeast, um gene (Y4L041W) que pertence às classes destacadas em cinza (ELISSEEFF; WESTON, 2002).

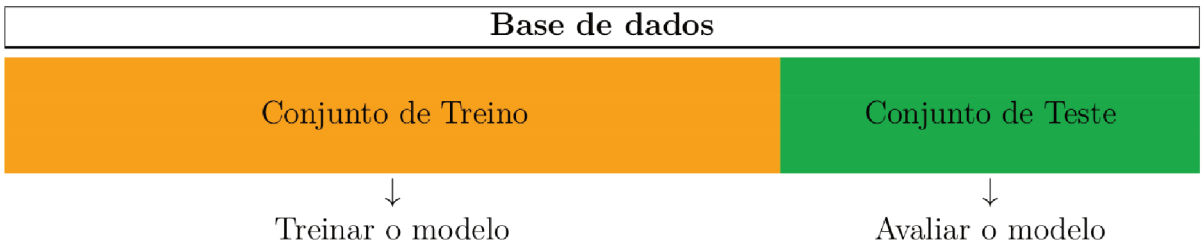


2.2.4.1 Validação Cruzada

❏ **Hold-out**

O hold-out é o método mais simples de validação cruzada. Resumidamente, são separados uma parte dos dados para treinamento e outra parte para teste (avaliação do modelo). Para usar o hold-out, é importante ter uma boa quantidade de dados, além disso, o conjunto de dados de treinamento deve ser sempre maior que o de teste e conter exemplos de todas as classes do problema. A Figura 10 ilustra o método Hold-out.

Figura 10 – Método Hold-out.



❏ **Método k-fold**

No k-fold, os dados são separados em k conjuntos que são usados k vezes, sendo 1 vez para validação e $k-1$ vezes para treinamento. Uma vantagem do k-fold, é a de que todos os dados são utilizados pelo menos uma vez para treino e teste, podemos estimar melhor o modelo. A Tabela 6 exemplifica o método k-fold.

Tabela 6 – Exemplo do método k-Fold para $k = 5$; cada linha na tabela representa o particionamento feito pelo k-fold, onde em cada partição é trocado um conjunto de treino para teste de forma que todos os objetos tenham sido usados para treinar e avaliar o modelo no processo.

Split 1	Treino	Treino	Treino	Treino	Teste
Split 2	Treino	Treino	Treino	Teste	Treino
Split 3	Treino	Treino	Teste	Treino	Treino
Split 4	Treino	Teste	Treino	Treino	Treino
Split 5	Teste	Treino	Treino	Treino	Treino

2.2.4.2 Otimização de Hiperparâmetros

Os hiperparâmetros de uma técnica são os valores que podemos escolher para controlar alguns aspectos específicos do algoritmo. Por exemplo, o hiperparâmetro do kNN é o valor de k, que controla o número de vizinhos mais próximos. Para o RF, podemos escolher o número de árvores da floresta e a profundidade máxima, por exemplo.

Um método que garante encontrar os melhores hiperparâmetros (dentro de um conjunto de valores pré-selecionados) é o Grid Search, um dos métodos mais simples para otimização de hiperparâmetros. O algoritmo consiste basicamente em uma busca exaustiva que treina e testa o modelo com todas as combinações de hiperparâmetros previamente definidos, em seguida, escolhe a combinação que obteve o melhor resultado de acordo com alguma métrica de avaliação. A Tabela 7 mostra um exemplo para 2 hiperparâmetros, cada um com 3 opções, sendo [a, b, c] os valores para o primeiro hiperparâmetro e [x, y, z] para o segundo; cada valor da matriz indica a performance obtida pela combinação dos parâmetros da linha/coluna.

Tabela 7 – Exemplo do método GridSearch para um problema com 2 hiperparâmetros. O algoritmo testará todas combinações e escolherá a combinação com melhor performance (destacado em verde), no caso (a) para o primeiro parâmetro e (z) para o segundo.

a	75%	73%	89%
b	69%	79%	81%
c	73%	83%	67%
	x	y	z

2.2.4.3 Medidas de Avaliação

Nesta seção serão apresentadas algumas das métricas mais populares para avaliação de técnicas multirrótulo. No contexto de classificação multirrótulo, algumas medidas, como

por exemplo, precisão, sensibilidade e F-score, podem ser obtida de diferentes formas:

- Micro-Averaging: Calcula a medida globalmente contando o total de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos[1].

$$B_{micro}(h) = B\left(\sum_{j=1}^{q} TP_j, \sum_{j=1}^{q} FP_j, \sum_{j=1}^{q} TN_j, \sum_{j=1}^{q} FN_j)\right) \tag{8}$$

- Macro-Averaging: Calcula a medida para cada rótulo e considera a média em relação ao número de rótulos (q). Esta medida não considera o desbalanceamento dos rótulos.

$$B_{macro}(h) = \frac{1}{q}\sum_{j=1}^{q} B(TP_j, FP_j, TN_j, FN_j) \tag{9}$$

- Weighted-Averaging: Calcula a medida para cada rótulo e faz uma média ponderada considerando o número de ocorrências do rótulo no conjunto de dados de teste ($TP_j + FP_j$), desta forma, é considerado no cálculo o desbalanceamento das classes.

$$B_{weighted}(h) = \frac{\sum_{j=1}^{q} B(TP_j, FP_j, TN_j, FN_j) \times (TP_j + FP_j)}{\sum_{j=1}^{q}(TP_j + FP_j)} \tag{10}$$

2.2.4.4 Perda de Hamming ↓

Perda de Hamming ou *Hamming Loss* é a fração de rótulos que foram incorretamente preditos (SOKOLOVA; LAPALME, 2009). Formalmente, seja D uma base de dados, Y_i o subconjunto correto de rótulos para o i-ésimo objeto e $Z_i = h(\mathbf{x}_i)$,

$$HammingLoss(h, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_i \oplus Z_i|}{k} \tag{11}$$

2.2.4.5 Acurácia de Subconjunto ↑

Acurácia do subconjunto ou *subset accurary* é a fração de exemplos que tiveram seu conjunto de rótulos perfeitamente classificados (TSOUMAKAS; VLAHAVAS, 2007).

$$SubsetAccuracy(h, D) = \frac{1}{|D|}\sum_{i=1}^{|D|}(Y_i = Z_i) \tag{12}$$

2.2.4.6 Precisão ↑

A medida de precisão ou *precision* mede a habilidade do algoritmo de não classificar como positivo uma instância que é negativa.

$$Precision(TP_j, FP_j, TN_j, FN_j) = \frac{TP_j}{TP_j + FP_j} \tag{13}$$

[1] B pode ser uma das medidas de avaliação (precisão, sensibilidade ou F-score).

2.2.4.7 Sensibilidade ↑

A medida de sensibilidade, também conhecida como *sensibility* ou *recall* nada mais é que a fração de rótulos classificados como positivo dentre os que realmente eram positivos.

$$Sensitivity(TP_j, FP_j, TN_j, FN_j) = \frac{TP_j}{TP_j + FN_j} \tag{14}$$

2.2.4.8 Medida-F ↑

A Medida-F ou F-Score é um balanceamento das medidas de precisão Eq. 13 e sensibilidade Eq. 14. Para o valor de β (fator de balanceamento) é normalmente atribuído 1, o que faz a Medida-F ser uma média harmônica entre a precisão e sensibilidade.

$$F^{\beta}(TP_j, FP_j, TN_j, FN_j) = \frac{(1+\beta^2) \times TP_j}{(1+\beta^2) \times TP_j + \beta^2 \times FN_j + FP_j} \tag{15}$$

2.2.4.9 Dependência de Rótulos

Um dos grandes desafios do aprendizado multirrótulo é explorar a correlação existente entre os rótulos. A dependência de rótulos é um fator importante até mesmo para diferenciação do problema monorrótuo, pois, se não existe nenhuma correlação entre os rótulos, o problema pode ser dividido em várias tarefas de classificação monorrótulo e resolvidos por qualquer classificador comum (BR) (CHERMAN, 2013).

A Tab. 8 mostra um exemplo de correlação entre 2 rótulos (y_1 e y_2). Sempre que y_1 está presente no objeto y_2 também está, e o contrário também se repete.

Tabela 8 – Exemplo de correlação entre os rótulos y_1 e y_3.

	y_1	y_2	y_3	y_4
X_1	1	0	1	1
X_2	0	1	0	1
X_3	1	0	1	0

(ZHANG; ZHANG, 2010) categorizaram os algoritmos de classificação multirrótulo em três grupos diferentes baseados em sua estratégia em relação a correlação de rótulos:

1. Estratégia de primeira ordem: Os algoritmos de primeira ordem costumam ser os mais simples e não consideram nenhuma relação entre os rótulos. Um exemplo é o Binary Relevance, que simplesmente converte o problema multirrótulo em vários problemas independentes de classificação binária.

2. Estratégia de segunda ordem: Os algoritmos de segunda ordem consideram a correlação entre pares de rótulos. Alguns exemplos são Rank-SVM e CML, que classificam os rótulos em termos de relevância ou analisando a interação entre os pares de rótulos.

3. Estratégia de alta ordem: Os algoritmos de alta ordem consideram relações de alta ordem entre todos os rótulos presentes (ZHANG; ZHOU, 2014). Um exemplo é o Classifier Chain que, para determinar se um objeto recebe um determinado rótulo ou não, considera a relação de todos os rótulos previamente testados para tal objeto (READ et al., 2011).

2.3 Redes Complexas para Classificação de Dados

Redes complexas tem sido um tópico amplamente estudado e usado para representar e entender diversos sistemas complexos (ALBERT; BARABÁSI, 2002). Uma rede complexa pode ser entendida como um grafo que não possui topologia nem regular e nem aleatória (CARNEIRO, 2017). Formalmente, uma rede (ou grafo) é uma estrutura $G(V, E)$, onde V representa o conjunto de vértices (ou nós) e E representa as arestas (ou conexões) do grafo. Cada aresta é representada por um par (u, v) que representa uma ligação que indica algum tipo de relacionamento entre os vértices $u, v \in V$. Em (ALBERT; BARABÁSI, 2002) foram investigados alguns modelos de redes complexas com propriedades interessantes, como redes de pequeno mundo, redes livre de escala e redes aleatórias.

Uma vez formada a rede é possível analisar e extrair diversas informações em toda ou parte da rede por meio de medidas de redes complexas. Tais medidas fornecem uma visão do comportamento da rede em algum aspecto, e.g., o grau médio representa a quantidade média de conexões de cada vértice da rede. Na seção 2.3.1 são fornecidos exemplos de medidas de redes complexas.

2.3.1 Medidas de Redes Complexas

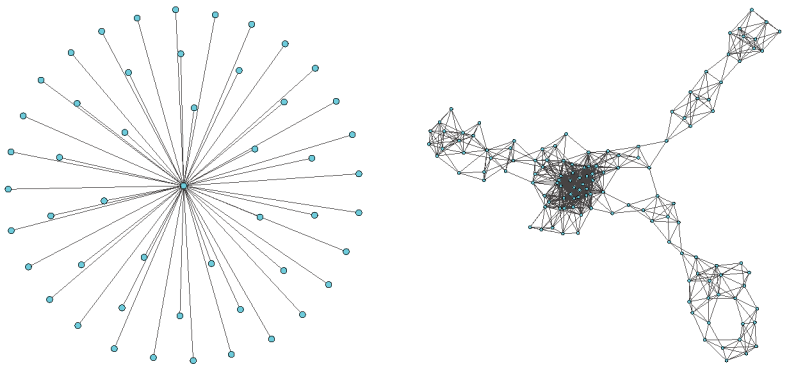
Nesta seção serão discutidas algumas medidas de redes complexas que são usadas para extrair informações da topologia da rede.

2.3.1.1 Assortatividade

A assortatividade quantifica o quanto os vértices tendem a se conectar a outros vértices de grau semelhante. A medida assume valores entre $[-1, 1]$, de modo que os valores positivos indicam que os pares de vértices diretamente conectados têm maior probabilidade de se comportar da mesma maneira, enquanto os valores negativos indicam uma maior probabilidade de terem comportamentos diferentes (CARNEIRO et al., 2016). Seja $E^{(l)}$ o número de arestas no grafo $G^{(l)}$ e $i_u^{(l)}, k_u^{(l)}$ os graus dos vértices i e k que compõem uma aresta u, a assortatividade pode ser calculada por:

$$r^{(l)} = \frac{\frac{1}{E^{(l)}} \sum_u i_u^{(l)} k_u^{(l)} - [\frac{1}{E^{(l)}} \sum_u \frac{1}{2}(i_u^{(l)} + k_u^{(l)})]^2}{\frac{1}{E^{(l)}} \sum_u \frac{1}{2}(i_u^{2(l)} + k_u^{2(l)}) - [\frac{1}{E^{(l)}} \sum_u \frac{1}{2}(i_u^{(l)} + k_u^{(l)})]^2} \tag{16}$$

A Figura 11 mostra exemplos de grafos com diferentes graus de assortatividade. A Fig. 11a exibe um tipo de rede chamada de grafo estrela, que possui $r = -1$ (vários vértices de grau baixo se conectando a um único vértice de grau alto). Já a Fig. 11b exibe uma rede com grau de assortatividade mais alto (construída a partir de uma das bases de dados e algoritmos que serão discutidos nas próximas seções), onde vértices de grau similar estão conectados (i.e., vértices com baixo grau se conectam a vértices de baixo grau e vértices com grau alto se conectam a outros vértices de grau alto).



(a) $r = -1$ (b) $r = 0.76$

Figura 11 – Assortatividade para diferentes tipos de rede.

2.3.1.2 Coeficiente de Agrupamento

O coeficiente de agrupamento (CA) ou *clustering coefficient* quantifica o quanto os vértices tendem a se agrupar. Basicamente, ele mede o quão próximo cada vértice do grafo está para formar um clique. O CA pode ser obtido por:

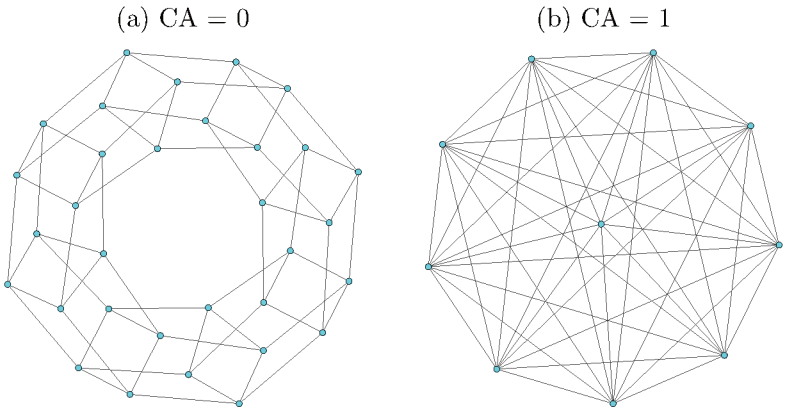
$$CA_i^{(l)} = \frac{|e_{us}^{(l)}|}{k_i^{(l)}(k_i^{(l)} - 1)} , \tag{17}$$

onde $|e_{us}^{(l)}|$ representa o número de conexões compartilhadas por vizinhos adjacentes do vértice i, e k_i o grau do vértice i. Seja $\mathcal{V}^{(l)}$ o número de vértices no grafo $G^{(l)}$, o coeficiente de agrupamento médio da rede pode ser obtido por:

$$CA^{(l)} = \frac{1}{\mathcal{V}^{(l)}} \sum_{i=1}^{\mathcal{V}^{(l)}} CA_i^{(l)} . \tag{18}$$

A Figura 12 apresenta redes com diferentes graus de coeficiente de agrupamento. A Fig. 12a exibe um modelo de grafo conhecido como Lattice, onde no exemplo em questão, nenhum dos vértices formam um clique, o que resulta em um CA = 0, o grafo Lattice apresentado é tridimensional, sendo suas dimensões $[8, 2, 2]$ (32 vértices). Já na Fig. 12b, é apresentando um grafo completo, onde todos os vértices compartilham conexões entre si, resultando em um CA = 1.

Figura 12 – Coeficiente de agrupamento para diferentes tipos de rede.



2.3.1.3 Grau Médio

O grau médio (GM) ou average degree é simplesmente a média do número de conexões dos vértices do grafo. O GM de um grafo $G^{(l)}$ pode ser obtido por:

$$GM^{(l)} = \frac{1}{\mathcal{V}^{(l)}} \sum_{i=1}^{\mathcal{V}^{(l)}} GM_i^{(l)} \tag{19}$$

A Figura 13 mostra duas redes com distintos valores de grau médio. A Fig. 13a é um grafo K-regular (todos os vértices possui o mesmo grau), com k = 2 e portando GM = 2. Já na Fig. 13b é apresentada uma rede Erdos Renyi (grafo aleatório com parâmetros $n = 50$ e $p = 0.15$) com GM 7.72.

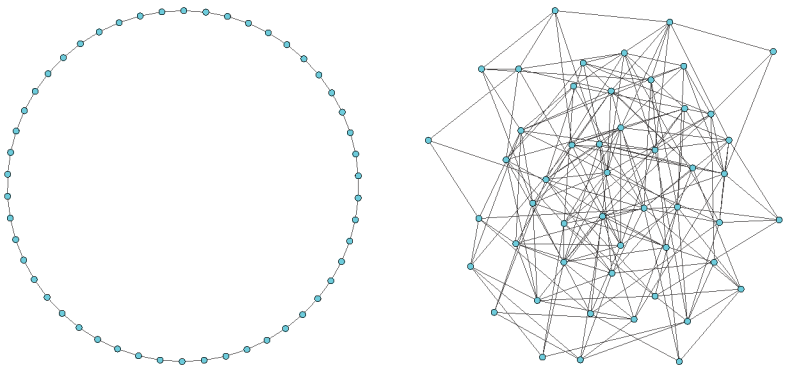


Figura 13 – Grau médio para diferentes tipos de rede.

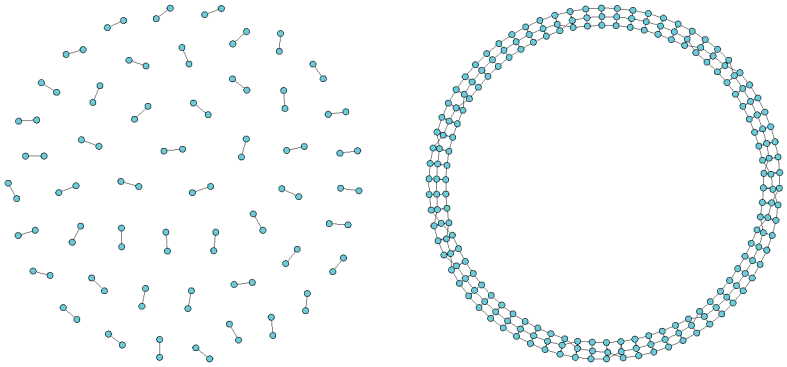
2.3.1.4 Comprimento Médio do Caminho

Comprimento médio do caminho (ALBERT; BARABÁSI, 2002) ou *average path length* (APL) nada mais é do que a média dos menores caminhos entre todos os pares de nós.

Formalmente, seja $d(u, v)$ a menor distância entre o vértice u e v, podemos obter o APL por:

$$APL = \frac{1}{n(n-1)} \sum_{i \neq j} d(v_i, v_j) \tag{20}$$

A Figura 14(a) mostra uma rede k-regular (k = 1) e com APL = 1, onde todos os caminhos possíveis tem o mesmo tamanho, 1. Por outro lado, a Fig. 14(b) mostra uma rede Lattice (dimensões [50, 3]) onde o comprimento médio do caminho é 18.25.



(a) APL = 1 (b) APL = 18.25

Figura 14 – Comprimento Médio do Caminho para diferentes tipos de rede.

2.3.2 Redes Complexas para Classificação Monorrótulo

O aprendizado de alto nível é caracterizado por capturar informações estruturais e topológicas dos dados, não sendo guiado apenas por sua similaridade física. Neste capítulo serão abordados algumas formas de representar os dados na forma de rede no contexto monorrótulo (seção 2.3.2.1) e alguns trabalhos relacionados de técnicas de aprendizado de alto nível (seção 2.3.2.2).

2.3.2.1 Métodos de Construção da Rede

Um dos principais desafios para que se possa capturar de modo efetivo características de alto nível é a construção de uma rede representativa dos dados.

No contexto do AM, um dos métodos mais comuns é o grafo kNN(kNNG). A ideia consiste basicamente em conectar cada item de dado aos seus k vizinhos mais próximos. Formalmente, seja kNN($\mathbf{x}_i$) os k vizinhos mais próximos de $\mathbf{x}_i$, podemos obter a matriz de adjacência do grafo $\mathbf{A}$ como:

$$A_{ij} = \begin{cases} 1 & \text{se } \mathbf{x_i} \in \text{kNN}(\mathbf{x}_j) \text{ e } y_i = y_j, \\ 0 & \text{senão.} \end{cases}$$

A partir desta estratégia surgiram diversas variações, como por exemplo o kNN mútuo (BRITO et al., 1997) e o kNN seletivo (CARNEIRO; ZHAO, 2018).

Outro método bem conhecido é o vizinhança de raio ϵ (eN) (WAN; YI, 2004) que basicamente conecta um item de teste a todos os vizinhos que estão até um raio máximo ϵ, obtendo uma melhor representação de regiões mais densas. Formalmente, seja D a matriz de distância onde D_{ij} é a distância entre $\mathbf{x}_i$ e $\mathbf{x}_j$, $\mathbf{A}$ pode ser obtida por:

$$A_{ij} = \begin{cases} 1 & \text{se } D_{ij} \leq \epsilon \text{ e } y_i = y_j, \\ 0 & \text{caso contrário.} \end{cases}$$

Uma estratégia interessante é combinar o kNNG com o ϵN (SILVA; ZHAO, 2012), desta forma, tanto regiões mais densas quanto regiões com menor densidade ficam bem representadas. Podemos obter $\mathbf{A}$ da seguinte maneira:

$$A_{ij} = \begin{cases} 1 & \text{se } \mathbf{x_i} \in \text{kNN}(\mathbf{x}_j) \text{ e } y_i = y_j, \\ 1 & \text{se } D_{ij} \leq \epsilon \text{ e } y_i = y_j, \\ 0 & \text{caso contrário.} \end{cases}$$

2.3.2.2 Aprendizado Baseado em Rede

O estudo de redes complexas forneceu uma nova forma de analisar e extrair informações de sistemas complexos. Uma gama de problemas indo desde redes de biologia (ALBERT, 2005) a redes de contato sexual (LILJEROS et al., 2001) puderam ser explorados devido à análise topológica fornecida pela teoria das redes complexas.

Recentemente o uso de redes complexas para o aprendizado supervisionado tem sido investigado em diversos trabalhos, sendo a grande maioria para aprendizado monorrótulo (SILVA; ZHAO, 2012; SILVA; ZHAO, 2015; CARNEIRO et al., 2017; CARNEIRO et al., 2019; YU et al., 2020). Acredita-se que ao explorar características dos dados em forma de rede podem ser capturados diferentes tipos de relacionamentos que, normalmente, são ignorados por técnicas tradicionais que usam os dados apenas em sua forma bruta (i.e., vetor de atributos) e ainda são fortemente (ou unicamente) guiados pela similaridade física dos dados.

Nesta seção serão apresentados alguns trabalhos relacionados com propostas de classificadores baseados em rede para aprendizado mono e multirrótulo. O modelo base para esta pesquisa é um framework híbrido para classificação monorrótulo proposto em (SILVA; ZHAO, 2012). Basicamente, é feita uma combinação entre um algoritmo de classificação de baixo nível com um classificador de alto nível baseado em rede. O algoritmo possui algumas etapas importantes:

- Construção da Rede: o termo de alto nível depende da construção da rede. No caso deste algoritmo, a construção do grafo é obtida pela combinação do kNNG com o

ϵN, como descrito na Seção 3.2. É importante notar que utilizando tal estratégia serão geradas diversas componentes, cada uma representando uma classe.

- Variação das Medidas de Rede: São calculadas medidas de redes complexas para cada componente do grafo obtido antes e depois da inserção de um objeto de teste. A probabilidade do classificador de alto nível vai depender da variação causada pelo novo objeto na rede. Se tal objeto causa uma grande variação em uma componente, ele provavelmente não está de acordo com o padrão de formação da classe representada por ela, e receberá uma baixa probabilidade. Caso contrário, ele recebe uma alta probabilidade.

- Combinação dos Classificadores: Na última etapa são combinadas as probabilidades do classificador de alto nível com as probabilidades de um classificador de baixo nível monorrótulo qualquer. A probabilidade final $M_i^{(j)}$ de um objeto de teste x_i pertencer a uma classe j é dada por:

$$M_i^{(j)} = (1 - \lambda)\, T_i^{(j)} + \lambda C_i^{(j)}$$

onde $T_i^{(j)} \in [0, 1]$ é a probabilidade dada por um algoritmo monorrótulo tradicional, $C_i^{(j)} \in [0, 1]$ é a probabilidade dada pelo classificador de alto nível e $\lambda \in [0, 1]$ é um parâmetro que controla o nível de contribuição de cada classificador na probabilidade final, quanto maior o valor de λ mais as características de alto nível são consideradas e quanto menor, mais as de baixo nível são.

A Fig. 15 ilustra o modelo híbrido de alto nível para classificação monorrótulo. Em (a) são apresentadas as componentes formadas por cada uma das classes do problema, que posteriormente terão os valores de cada medida de rede adotada devidamente calculadas. Em (b) um item de teste é temporariamente inserido em seus vizinhos mais próximos para o cálculo da variação, que irá indicar a classe com o padrão mais condizente com teste.

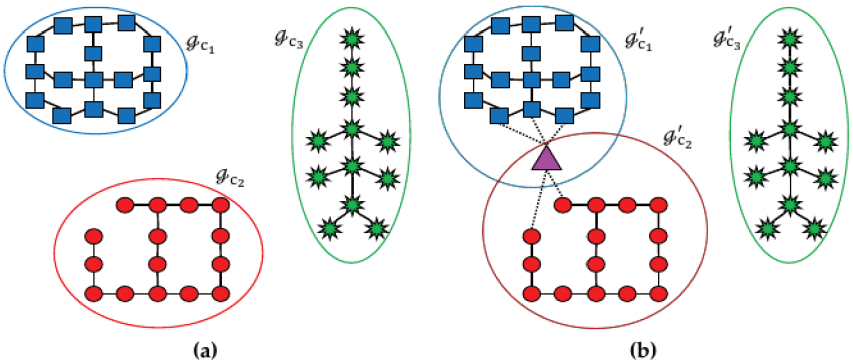


Figura 15 – Ilustração do framework híbrido de alto nível para classificação tradicional (SILVA; ZHAO, 2012).

Em (SILVA; ZHAO, 2015) foi proposto outro algoritmo híbrido baseado neste modelo, mas que utiliza caminhada do turista no processo de identificação da conformidade de padrão.

Além do algoritmo descrito anteriormente, existe outro modelo baseado em rede que faz ranqueamento dos rótulos utilizando um conceito de importância (CARNEIRO et al., 2017). Basicamente, o algoritmo cria uma rede na fase de treinamento e realiza a classificação utilizando uma medida baseada no PageRank do Google, denominada *medida de importância*. Tal medida define em qual classe (cada uma representada por uma componente) o objeto em teste possui maior importância.

Os passos do algoritmo serão descritos a seguir:

- Construção da Rede: Neste trabalho foram utilizados 2 métodos de construção do grafo. kNN e kAOG, o primeiro já foi explicado anteriormente.

  A rede kAOG funciona de maneira similar ao kNN, a diferença é que o valor de k é selecionado automaticamente pelo algoritmo e o processo de criação da rede é guiado por uma medida de pureza. Tal medida define o nível de mistura de uma componente em relação a outras componentes de classes diferentes (CARNEIRO; ZHAO, 2018). O algoritmo aumenta o valor de k até que a pureza não aumente mais.

- Cálculo da eficiência das componentes: cada componente recebe um valor de eficiência baseado nas distâncias físicas dos vértices que a compõem. Tal medida pode ser descrita como:

$$\varepsilon^{\alpha} = \frac{1}{|N^{\alpha}|} \sum_{i \in \alpha} \xi_i^{(\alpha)}$$

  onde ε^{α} é a média da eficiência local ($\xi^{(\alpha)}$) dos vértices de α e $|N^{\alpha}|$ é o tamanho da componente α. A eficiência local pode ser definida como:

$$\xi^{(\alpha)} = \frac{1}{N_i} \sum_{i \to j} D_{i,j}$$

  onde N_i é o número de conexões de i e $D_{i,j}$ é a distância Euclidiana entre i, j obtida pelo vetor de atributos na forma original.

- Cálculo da importância dos vértices: cada nó recebe um valor de importância baseado no número de ligações que recebe e também de ligações de outros vértices com grande importância.

- Seleção de vértices para conectar com o teste: na fase de teste, é verificado se uma conexão do teste com um nó da componente aumenta ou não a eficiência da mesma. Se aumentar, ele recebe conexões temporárias dos vértices que o fizeram aumentar a eficiência da componente. No caso em que o objeto não aumenta a eficiência de nenhuma componente, ele recebe ligações dos vértices que menos o fizeram diminuir a eficiência da componente.

- Classificação baseada em importância: por fim, o teste é inserido temporariamente em cada classe e sua importância é calculada em cada uma. O objeto recebe então o rótulo da classe em que obteve a maior importância.

$$I_y^{(C)} = \sum_{j \in \wedge_y^C} I_j$$

onde $j \in X_{train}$ é um vértice rotulado, A_y^C é o conjunto de nós que pertencem a classe C e estão conectados a y e I_j é a importância do nó j.

2.3.3 Redes Complexas para Aprendizado Multirrótulo

Em relação ao aprendizado multirrótulo, majoritariamente, as técnicas de aprendizado multirrótulo que exploram propriedades estruturais dos dados tratam problemas que já são genuinamente representados em forma de grafos, não tendo relevância para problemas onde os dados são fornecidos em forma de vetor de atributos, como a grande maioria dos problemas enfrentados no AM.

Entretanto, recentemente surgiram alguns algoritmos de AMR que consideram os dados na forma de vetor de atributos e os transformam em uma rede. Em (TAN et al., 2015) foi proposto um método que constrói um grafo via um processo de geração de cliques que utiliza tanto os atributos quanto as classes, visando capturar correlações entre os rótulos. Em seguida, é utilizado uma heurística para definir os cliques mais relevantes, que serão utilizados na fase de teste para predizer o rótulo do objeto usando inferência bayesiana. Outro exemplo foi proposto em (WANG et al., 2020) uma técnica multirrótulo baseada em um grafo de passeio aleatório (*random walk graph*) e na técnica dos k vizinhos mais próximos (kNN), chamada MLRWKNN. O algoritmo funciona da seguinte maneira:

- para cada item de teste x_0 são gerados novos $|L|$ grafos $(G_{x_0}^{(l)}, l = 1, 2, \ldots, |L|)$, um para cada rótulo do problema.

- o conjunto de vértices dos grafos de cada rótulo $V_{x_0}^{(l)}$ é o mesmo, sendo selecionados através dos vizinhos mais próximos do item de teste, $V_{x_0}^{(l)} = kNN(x_0)$.

- são atribuídas arestas bidirecionais entre os vértices dos grafos se eles possuem ao menos uma classe em comum (os grafos são iguais, independente da classe, i.e., $G_{x_0}^{(1)} = G_{x_0}^{(2)} = \ldots = G_{x_0}^{(|L|)}$).

- para cada grafo, o item de teste é conectado aos vértices que possuem a classe representada por tal grafo. A Figura 16 mostra o processo de inserção do item de teste.

- por fim, para obter as probabilidades de pertencimento a cada rótulo, o MLRWKNN utiliza o random walk partindo de x_0 com uma probabilidade $1 - \alpha$ de se mover

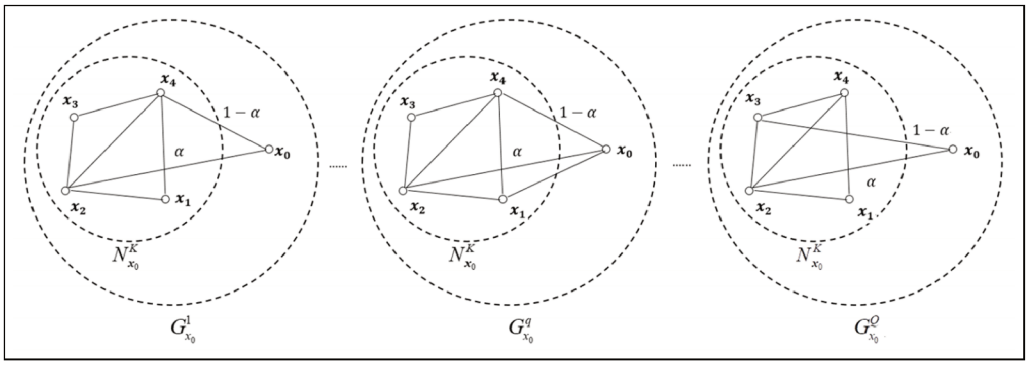


Figura 16 – Ilustração do processo de inserção do item de teste nos grafos gerados no treinamento do algoritmo MLRWKNN (WANG et al., 2020).

para um vértice vizinho e α de ir para qualquer outro vértice do grafo, obtendo $|L|$ vetores de distribuição de probabilidade onde é feita uma média ponderada para obter seus valores finais.

2.4 Considerações Finais

Neste capítulo foram abordados conceitos fundamentais para o desenvolvimento do trabalho, como técnicas de aprendizado monorrótulo e multirrótulo, além de abordagens e conceitos importantes de redes complexas. É importante ressaltar que o uso de redes complexas no aprendizado multirrótulo foi muito pouco explorado, sendo este um dos primeiros trabalhos envolvendo ambos os tópicos. Também foram discutidas formas de avaliar um modelo, em específico, medidas de avaliação multirrótulo e técnicas de otimização de parâmetro. Da mesma forma, foram discutidas algumas das principais bases de dados usadas na literatura para comparação e avaliação de algoritmos de classificação multirrótulo.

Capítulo 3

Classificador de Alto Nível para Aprendizado Multirrótulo

A maioria dos algoritmos de aprendizado multirrótulo são derivados de técnicas usadas para a classificação monorrótulo (ZHANG; ZHOU, 2014), os quais em sua maior parte ignoram correlações entre rótulos ou relações semânticas dos dados (RESENDE; CARNEIRO, 2019). Nesse sentido, em comum ambos os grupos de algoritmos são fortemente caracterizados por realizar a modelagem preditiva dos dados essencialmente baseada nos seus atributos físicos, tais como distância ou distribuição por exemplo, sendo por essa razão chamados de classificadores de baixo nível (SILVA; ZHAO, 2012).

Por outro lado, a literatura tem demonstrado que técnicas baseadas em redes complexas são capazes de considerar, além dos atributos físicos, aspectos estruturais e topológicos dos dados a partir de sua representação em forma de rede (SILVA; ZHAO, 2012; SILVA; ZHAO, 2015; CUPERTINO; ZHAO; CARNEIRO, 2015; CARNEIRO; ZHAO, 2018; CUPERTINO et al., 2018; WANG et al., 2020), sendo por essa razão denominados de classificadores de alto nível. Exemplos dessas técnicas incluem o classificador via conformidade de padrão (SILVA; ZHAO, 2012; SILVA; ZHAO, 2015), via caracterização de importância (CARNEIRO; ZHAO, 2018) e via passeio aleatório (CUPERTINO; ZHAO; CARNEIRO, 2015; WANG et al., 2020), os quais foram discutidos no capítulo anterior. Contudo, apesar dos bons resultados com a classificação de alto nível, o seu desenvolvimento e aplicação ainda estão relacionados apenas ao contexto de aprendizado monorrótulo. Ademais, as poucas técnicas multirrótulos baseadas em grafos existentes na literatura normalmente consideram apenas dados originalmente representados em tal formato.

Diante do exposto, este capítulo apresenta a técnica híbrida de aprendizado multirrótulo para classificação de alto nível desenvolvida nesta dissertação, denominada CXN-MLL, a qual é capaz de considerar estruturas físicas e topológicas dos dados para problemas multirrótulo bem como aprender a partir de dados representados em diferentes formatos, tais como imagem, texto, áudio, vetor de atributos, entre outros.

O restante do capítulo é organizado da seguinte forma. A Seção 3.1 traz uma visão geral sobre o método CXN-MLL; a Seção 3.2 apresenta a etapa de treinamento, caracterizada pela transformação dos dados em forma de rede; a Seção 3.3 define a etapa de teste, caracterizada pela análise de conformidade de padrão via medidas de redes complexas; e a Seção 3.5 traz uma síntese do algoritmo desenvolvido bem como sua análise em termos de complexidade de tempo computacional.

3.1 Visão Geral

A técnica proposta pode ser caracterizada como um modelo híbrido de classificação que combina probabilidades produzidas por um classificador de baixo (*low-level*/LL) e um de alto nível (*high-level*/HL). As probabilidades de baixo nivel são obtidas a partir de uma técnica de classificação convencional que considera essencialmente as características físicas dos dados de entrada, enquanto as de alto nível são obtidas a partir de medidas de redes complexas de modo a considerar também características estruturais e topológicas extraídas da representação dos dados de entrada em forma de rede. A Figura 17 oferece uma visão geral da aplicação do modelo para um problema com 2 classes, com padrões distintos, □/vermelho e ⋄/verde, além de instâncias múltirrótulo apresentadas em azul. Na fase de treino, a técnica de alto nível transforma os dados de entrada em redes (uma para cada rótulo) e extrai as características estruturais das redes obtidas por meio de medidas de redes complexas. Enquanto isso, a técnica de baixo nível (que será um algoritmo tradicional de classificação multirrótulo) é treinada para capturar as características físicas dos dados, separando os dados no espaço. O processo de obtenção das associações de baixo nível depende da técnica escolhida, podendo levar em consideração diferentes técnicas ou estratégias de transformação do problema ou adaptação de algoritmo, bem como correlação entre rótulos, etc. Como a literatura multirrótulo conta com um número considerável de classificadores de baixo nível, alguns deles inclusive discutidos apropriadamente no Capítulo 2, focamos a seguir em descrever a principal contribuição desta dissertação, a técnica de alto nível para classificação multirrótulo.

Durante a fase de teste, o conjunto de rótulos associados a cada novo objeto de teste é determinado a partir dos modelos de baixo e alto nível treinados na etapa anterior. No caso da classificação de alto nível, o novo objeto (representado pela cor magenta) é temporariamente inserido nas redes geradas no treinamento e as medidas de rede são recalculadas. As associações de alto nível são produzidas então a partir das variações que a inserção do item de teste causou nas medidas de rede. Se causa pouca variação, queremos que o objeto tenha uma alta chance de ser associado a classe em questão (pois está em conformidade com o padrão da classe), e se causar muita variação, o contrário. Por fim, são combinadas as associações do modelo de baixo nível com as associações do modelo de alto nível. Na Fig. 17, é possível notar que o o item de teste claramente

completa o padrão de ambas as classes, porém, o classificador de baixo nível só consegue separar bem a classe dos objetos em vermelho, fornecendo uma baixa probabilidade para o padrão formado em verde, que é facilmente detectado pela técnica de alto nível, fazendo com que a combinação de ambas as técnicas consiga classificar e identificar os múltiplos padrões do objeto de teste, contornado o problema da técnica de baixo nível. Vale ressaltar que a análise de conformidade de padrão através de medidas de redes complexas possui inspiração em outras técnicas desenvolvidas para a classificação monorrótulo, tais como (SILVA; ZHAO, 2012; CARNEIRO et al., 2014; SILVA; ZHAO, 2015; CUPERTINO et al., 2018).

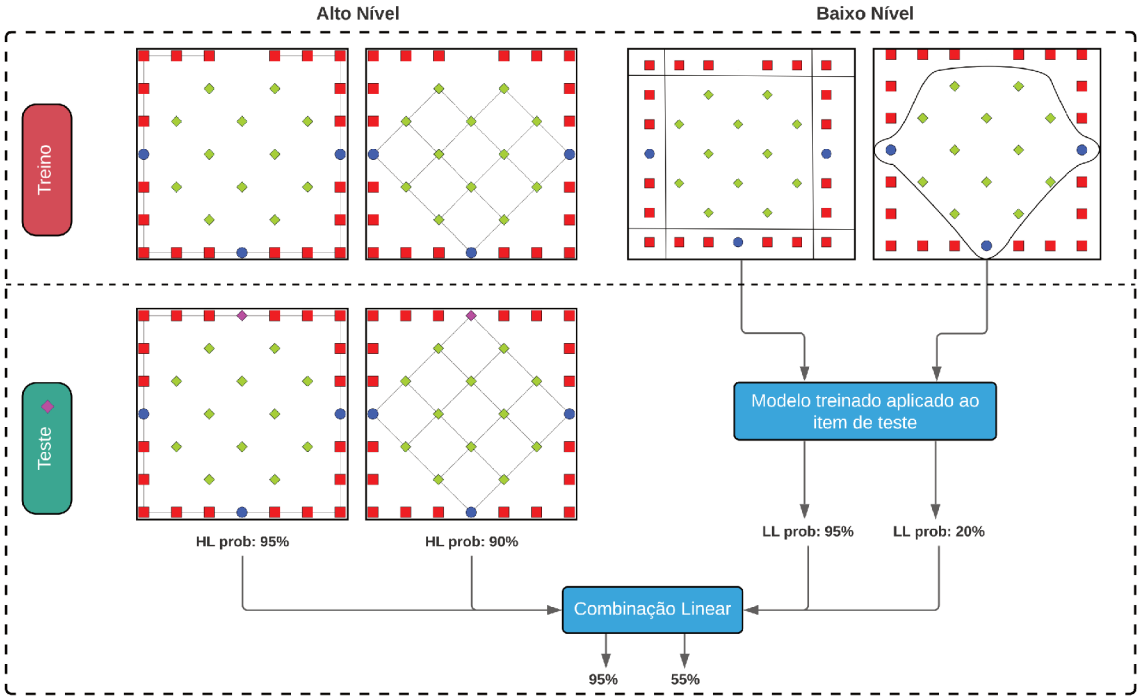


Figura 17 – Visão Geral das fases de treino e teste da técnica desenvolvida para aprendizado multirrótulo via redes complexas (CXN-MLL).

3.2 Construção de Redes para Aprendizado Multirrótulo

A fase de treinamento é composta por duas etapas, a construção do grafo a partir dos dados de treinamento e o cálculo das medidas de rede. O objetivo da primeira etapa é construir um grafo não direcionado $\mathcal{G}^{(l)}$ a partir de $\mathcal{X}$ que represente cada classe $l \in L$, ou seja, $g(\mathcal{X}, l) \rightarrow G^{(l)}$. Na segunda etapa, as medidas de rede são calculadas para cada um dos grafos gerados. Como as quatro medidas de rede consideradas nesse estudo já foram apresentadas no Capítulo 2, a saber *Assortatividade*, *Coeficiente de Agrupamento*, *Grau*

Médio e *Comprimento Médio do Caminho*, a presente seção visa destacar os métodos de construção da rede desenvolvidos para o contexto multirrótulo.

Em classificação de dados, a maioria das bases de dados são disponibilizadas na forma de vetor de atributos (CARNEIRO; ZHAO, 2018). Porém, para explorar relacionamentos espaciais, estruturais e topológicos dos dados de entrada, uma rede representativa destes dados deve ser construída. Nesse sentido, os métodos desenvolvidos aqui foram adaptados a partir de métodos representativos do aprendizado supervisionado monorrótulo (CARNEIRO; ZHAO, 2018), tais como a rede k-vizinhos (kNN), rede k-vizinhos seletiva (S-kNN) e a rede vizinhança de raio ϵ (ϵN). Para tornar a apresentação e implicações dos métodos mais didática, a Figura 18 ilustra uma base de dados que será usada para fins de exemplo. A base de dados é composta de objetos de duas classes representados em um espaço bidimensional, onde a classe 1 é representada em vermelho, a classe 2 é representada em azul e os objetos em pretos representam as instâncias multirrótulo (i.e., possuem a classe 1 e 2). A seguir são descritos os cinco métodos de construção de redes investigados nessa dissertação.

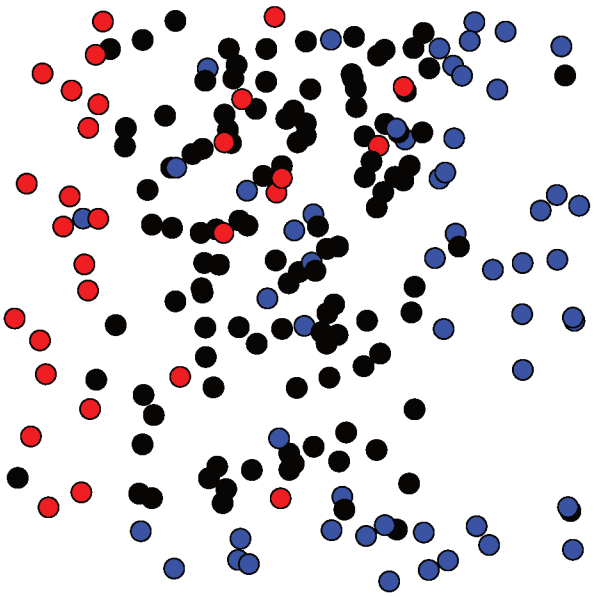


Figura 18 – Conjunto de dados 2D para exemplificação dos métodos de construção da rede adaptados para o contexto multirrótulo.

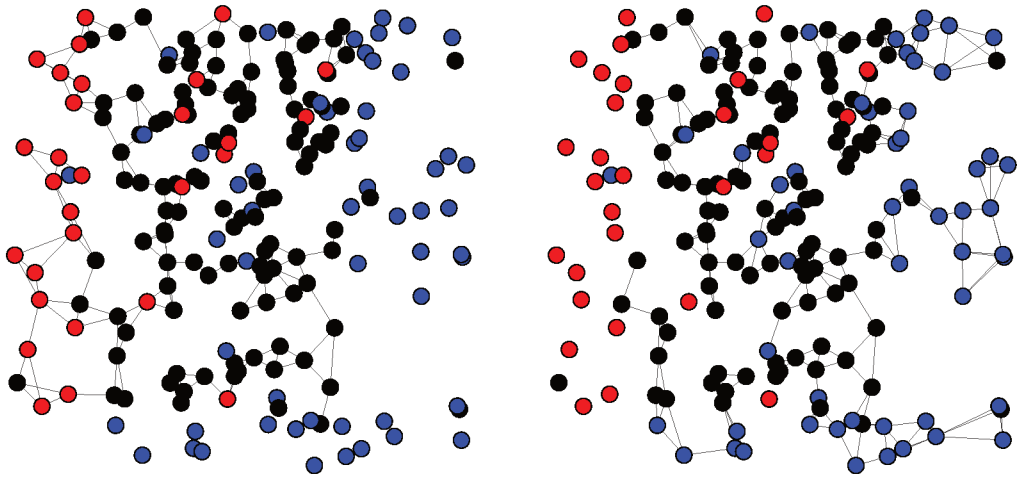
3.2.1 Rede kNN multirrótulo

A rede kNN multirrótulo irá construir $|L|$ grafos não-direcionados onde cada objeto será conectado aos seus k vizinhos mais próximos se possuírem alguma classe em comum. É importante notar que um objeto $x_i \in \mathcal{X}$ estará presente em um grafo $G^{(l)}$ se e somente se $y_i^{(l)} = 1$.

Seja kNN($\mathbf{x}_i$) os k vizinhos de $\mathbf{x}_i$, e $\mathbf{A}^{(l)}$ a matriz de adjacência de $G^{(l)}$, temos:

$$A_{ij}^{(l)} = \begin{cases} 1, & \text{if } \mathbf{x_i} \in \text{kNN}(\mathbf{x}_j) \text{ e } y_i^{(l)} = y_j^{(l)}, \\ 0, & \text{caso contrário.} \end{cases}$$

A Figura 19 mostra como fica a rede kNN para cada uma das classes dos objetos da Fig. 18. Podemos notar que a rede kNN tende a gerar grafos mais esparsos, o que pode ser explicado pelo fato de que nem todo objeto é conectado a seus k vizinhos devido a exigência de compartilharem a mesma classe. Obviamente, a rede kNN pode gerar grafos densos dependendo do valor atribuído a k.



(a) G^1 (b) G^2

Figura 19 – Exemplo de grafo construído usando o método kNN-Graph (k = 3).

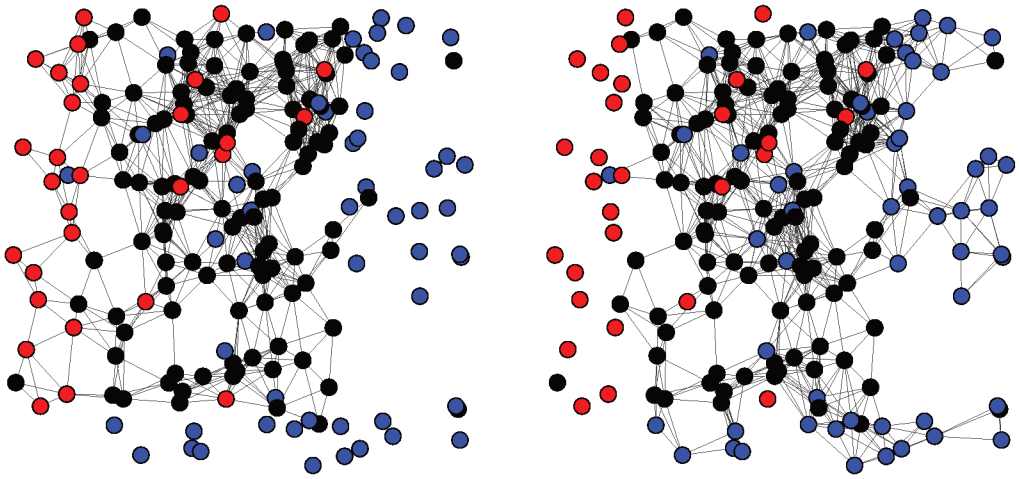
3.2.2 Rede kNN+ϵN multirrótulo

A rede kNN+ϵN é uma variante do método anterior, onde cada objeto se conecta não só em seus k vizinhos, mas também nos objetos que estão a um raio ϵ de distância, sendo mais apropriada para representar regiões densas de dados. Assim como na rede kNN, é exigido que os objetos a serem conectados compartilhem a mesma classe.

Seja D uma matriz de distância e D_{ij} a distância entre $\mathbf{x}_i$ e $\mathbf{x}_j$, $\mathbf{A}^{(l)}$ pode ser obtida por:

$$A_{ij}^{(l)} = \begin{cases} 1, & \text{if } \mathbf{x_i} \in \text{kNN}(\mathbf{x}_j) \text{ e } y_i^{(l)} = y_j^{(l)}, \\ 1, & \text{if } D_{ij} \leq \epsilon \text{ e } y_i^{(l)} = y_j^{(l)}, \\ 0 & \text{caso contrário.} \end{cases}$$

A Figura 20 mostra como fica a rede kNN+ϵN para cada uma das classes dos objetos da Fig. 18. Pode-se observar que os grafos gerados possuem regiões densas, o que não acontece ao usar a rede kNN por si só.



(a) G^1 (b) G^2

Figura 20 – Exemplo de grafo construído usando o método kNN+ϵN-Graph (k = 3, ϵ = 0.3).

3.2.3 Rede S-kNN multirrótulo

A rede S-kNN é muito similar à kNN. A principal diferença é que na rede S-kNN é considerado na vizinhança apenas instâncias que possuem classes em comum.

Seja S-kNN($\mathbf{x}_i$,l) os k vizinhos de $\mathbf{x}_i$ que possuem a classe $l \in L$, $\mathbf{A}^{(l)}$ pode ser obtido por:

$$A_{ij}^{(l)} = \begin{cases} 1, & \text{if } \mathbf{x_i} \in \text{S-kNN}(\mathbf{x}_j, l), \\ 0, & \text{caso contrário.} \end{cases}$$

A Figura 21 mostra como fica a rede kNN para cada uma das classes dos objetos da Fig. 18. Pode-se observar que vértices mais isolados de suas classes ainda conseguem realizar conexões, o que não acontecia na rede kNN.

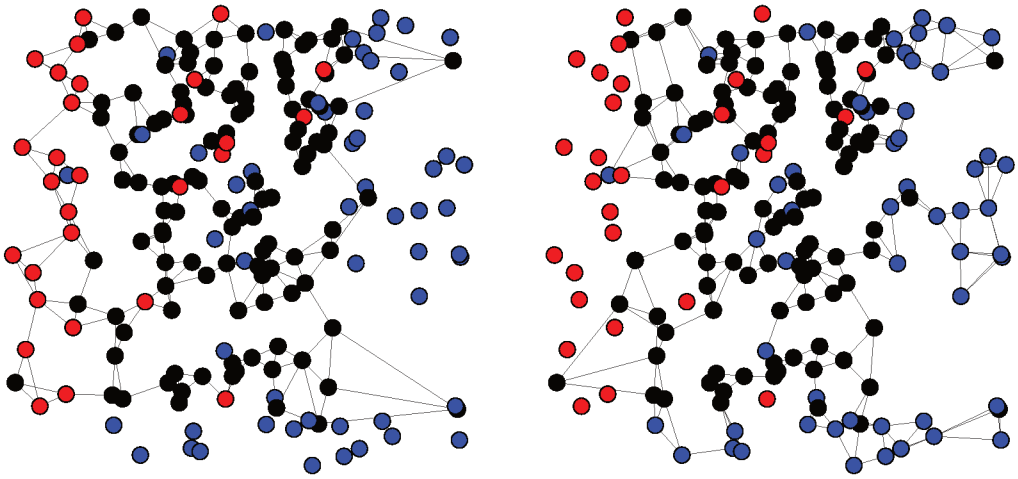
3.2.4 Rede S-kNN+ϵN multirrótulo

Este método é similar ao método da seção 3.2.2, a diferença é que ele tende a realizar mais conexões pela substituição do kNN pelo S-kNN.

Considere a função S-kNN(.) definida anteriormente e a matriz de distância D, $\mathbf{A}^{(l)}$ é definida por:

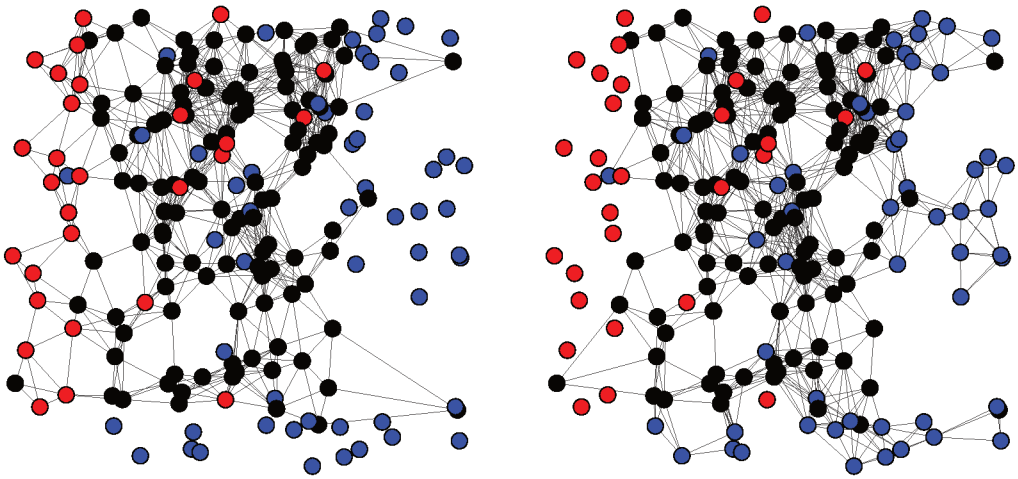
$$A_{ij}^{(l)} = \begin{cases} 1, & \text{if } \mathbf{x_i} \in \text{S-kNN}(\mathbf{x}_j, l), \\ 1, & \text{if } D_{ij} \leq \epsilon \text{ e } y_i^{(l)} = y_j^{(l)}, \\ 0 & \text{caso contrário.} \end{cases}$$

A Figura 22 mostra como fica a rede skNN+ϵN para cada uma das classes dos objetos da Fig. 18.



(a) G^1 (b) G^2

Figura 21 – Exemplo de grafo construído usando o método skNN-Graph (k = 3).



(a) G^1 (b) G^2

Figura 22 – Exemplo de grafo construído usando o método skNN+ϵN-Graph (k = 3, ϵ = 0.3).

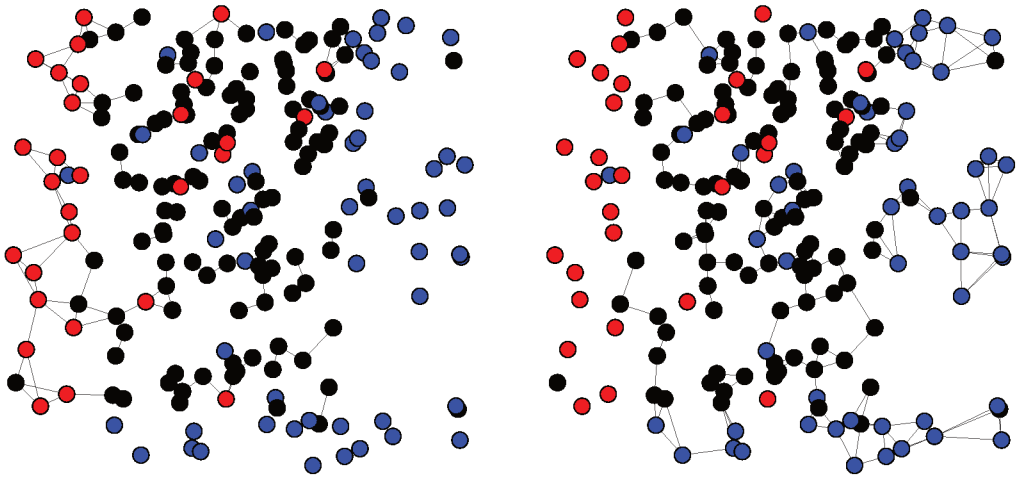
3.2.5 Rede D-kNN multirrótulo

O método D-kNN (*degree* k-Nearest Neighbor), diferente dos outros métodos, usa o valor de k considerando as conexões feitas pelo objeto em todos os grafos (e.g., se um objeto e seu vizinho possuem as classes 1 e 2, é descontado 2 do valor de k, já que ele será conectado em dois grafos distintos), o que faz com que a técnica produza grafos mais esparsos em comparação com as demais.

Seja v_i um nó conectado a $\mathbf{x_i}$ no grafo, a matriz de adjacência é dada por:

$$A_{ij}^{(l)} = \begin{cases} 1, & \text{if } \mathbf{x_i} \in \text{kNN}(\mathbf{x}_j) \text{ e } y_i^{(l)} = y_j^{(l)} \text{ e} \\ & \sum_{l \in L} outdegree(v_i \in V^{(l)}) < k \\ 0, & \text{caso contrário.} \end{cases}$$

A Figura 23 mostra como fica a rede D-kNN para cada uma das classes dos objetos da Fig. 18.



(a) G^1 (b) G^2

Figura 23 – Exemplo de grafo construído usando o método D-kNN-Graph (k = 3).

3.3 Associações de Alto Nível via Conformidade de Padrão

Na fase de teste do classificador de alto nível, o CXN-MLL calcula as variações das medidas de rede para classificar um determinado item de teste nas classes em que sua inserção causa pequenas alterações (ou mesmo nenhuma). Formalmente, seja u uma determinada medida de rede, e $m^{(l)}$ e $m'^{(l)}$ o resultado da aplicação de u para um dado grafo $G^{(l)}$ respectivamente antes e depois da inserção de um item de teste $\mathbf{x}$, a variação da medida de rede u pode ser definida como:

$$\Delta G_{\mathbf{x}}^{(l)}(u) = \frac{|m^{(l)} - m'^{(l)}|}{\sum_{q \in L} |m^{(q)} - m'^{(q)}|} \tag{21}$$

Basicamente, se a inserção do item de teste causa alta variação das medidas de redes complexas naquele grafo, então provavelmente não é compatível com o padrão de classe representado por tal rede (classe). Caso contrário, se tal variação é baixa (ou mesmo não existe), o item de teste provavelmente pertence a essa classe. Observe que u pode ser qualquer uma das medidas de rede definidas anteriormente, com m representando

os resultados obtidos pela medida selecionada. Quanto ao denominador da equação, é apenas um processo de normalização das variações das medidas de rede.

Há também um caso especial no cálculo das medidas de rede que precisa ser tratado individualmente. No caso da medida de assortatividade, por exemplo, pode acontecer que o item de teste não se conecte a nenhum vértice em algum dos grafos multirrótulos. Portanto, sua variação seria zero e, supostamente, estaria em perfeita conformidade com o padrão da classe representada. No entanto, esta deve ser exatamente a situação oposta. Assim, para lidar com casos especiais como este, definimos o valor da variação da medida como o pior possível, ou seja, a diferença máxima entre os intervalos de medida, indicando que o item de teste causou a maior variação. Assim, o valor de $CC^{(l)}$ é definido como 1 e $CC'^{(l)}$ como 0 para o coeficiente de agrupamento; $r^{(l)}$ é definido como 1 e $r'^{(l)}$ como 0 para a variedade; e para o grau médio é definido como $k^{(l)} = \max(k'^{(l)} - \min(k_i'^{(l)}), \max(k_i'^{(l)}) - k'^{(l)})$.

A Fig. 24 ilustra a etapa do cálculo das variações causadas por um item de teste, representado pelo $\diamond$/magenta. Na ilustração são apresentadas 3 classes distintas, cada uma com um padrão de formação diferente. É possível notar que, diferente do padrão da classe 1 (circulos em ciano) o objeto de teste completa o padrão da classe 2 (quadrados vermelhos) e classe 3 (triângulo verde), assim como os objetos em azul (instâncias multirrótulo que possuem as classes 2 e 3). Considerando um cenário onde o objeto se conectasse aos seus 5 vizinhos mais próximos na fase de teste, os grafos de cada classe ficariam como as ilustrações contidas no circulo tracejado ($\mathcal{G}^{(1)}, \mathcal{G}^{(2)}$ e $\mathcal{G}^{(3)}$). Ao calcular as variações causadas na estrutura do grafo pelo objeto de teste, é esperado uma alta variação na classe 1 (onde o objeto não está em conformidade com o padrão da classe) e uma baixa variação nas classes 2 e 3 (onde o objeto completa o padrão e consequentemente está em conformidade com o padrão de tais classes), fornecendo uma análise de alto nível a respeito do item de teste.

Depois que as variações das medidas da rede são calculadas, procedemos com a mesma estratégia proposta em (SILVA; ZHAO, 2012) para lidar com problemas de classes desbalanceadas:

$$f_{\mathbf{x}}^{(l)}(u) = \Delta G_{\mathbf{x}}^{(l)}(u)\, p^{(l)}, \tag{22}$$

onde $p^{(l)}$ é a proporção de itens com rótulo l.

Ao término da classificação de alto nível, o grau de pertinência de um item de teste $\mathbf{x}$ para uma classe l é dada pela soma ponderada das variações das medidas de rede, definida por:

$$\mathcal{H}_{\mathbf{x}}^{(l)} = \sum_{u=1}^{Z} \delta(u)[1 - f_{\mathbf{x}}^{(l)}(u)] \ , \tag{23}$$

com Z sendo o número de medidas de rede adotadas e $\delta \in [0, 1]$ e $\sum_{u=1}^{Z} \delta(u) = 1$ os pesos para os resultados de variação fornecidos por cada medida de rede.

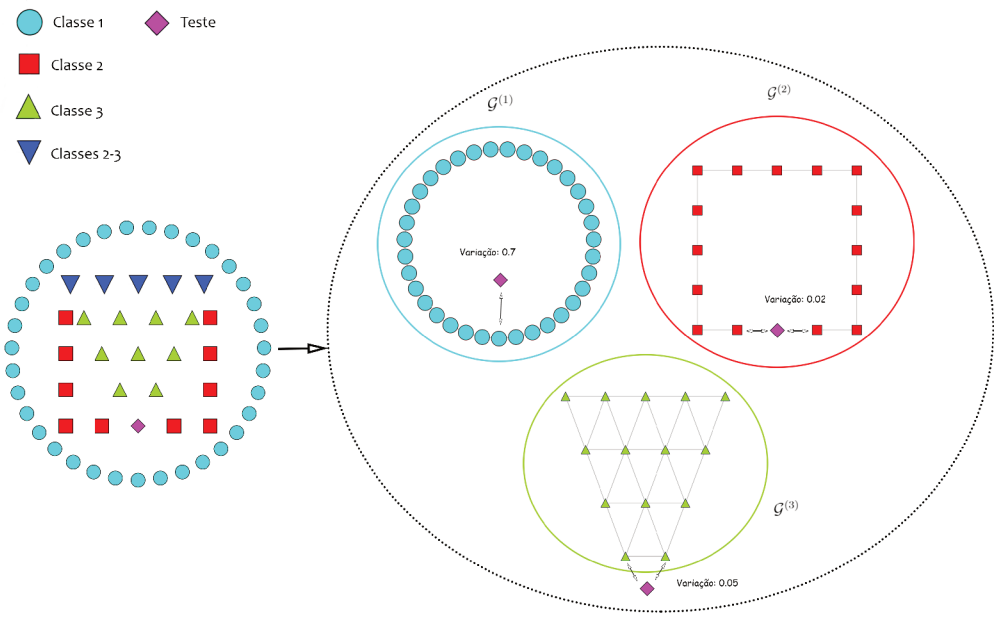


Figura 24 – Ilustração da etapa de cálculo das variações das medidas de rede.

3.4 Combinação de Associações de Alto e Baixo Nível

Após a obtenção do vetor de pertinência por meio das medidas de rede (HL) e o vetor de probabilidades do classificador de baixo nível (LL), tem início a última etapa da técnica CXN-MLL na qual é realizada a combinação de tais vetores. Assim, o grau de pertinência de um objeto a probabilidade de um objeto $\mathbf{x}$ pertencer a uma classe l é definida por:

$$\mathcal{M}_{\mathbf{x}}^{(l)} = \lambda \mathcal{H}_{\mathbf{x}}^{(l)} + (1 - \lambda)\mathcal{C}_{\mathbf{x}}^{(l)} \tag{24}$$

onde $\mathcal{M}_{\mathbf{x}}^{(l)}$ é o valor gerado pela combinação de probabilidades fornecidas por um classificador de baixo nível $\mathcal{C}_{\mathbf{x}}^{(l)}$, e pelo classificador de alto nível $\mathcal{H}_{\mathbf{x}}^{(l)}$. O parâmetro $\lambda \in [0, 1]$ corresponde a uma combinação linear convexa dos classificadores, onde altos valores priorizam características de alto nível, e valores baixos características de baixo nível. Por exemplo, se $\lambda = 0$ apenas as probabilidades fornecidas pelo classificador multirrótulo de baixo nível serão consideradas.

Depois que $\mathcal{M}$ é calculado, a saída da técnica CXN-MLL pode ser obtida por:

$$y_{\mathbf{x}}^{(l)} = \begin{cases} 1 & \text{if } \mathcal{M}_{\mathbf{x}}^{(l)} \geq \tau, \\ 0 & \text{otherwise,} \end{cases}$$

na qual o o item de teste $\mathbf{x}$ receberá o rótulo l se a combinação resultante das probabilidades de baixo e alto nível resultarem em uma probabilidade maior que um dado limiar τ.

3.5 Algoritmo e Complexidade

O Alg. 1 descreve os principais passos da técnica multirrótulo de alto nível apresentada neste capítulo. Uma implementação do algoritmo pode ser encontrada em <https://github.com/resendevinicius/CXN-MLL>.

A seguir, discutiremos a complexidade do tempo inerente a cada uma dessas etapas. Para maior clareza, consideramos o número de nós $\mid \sum_{l=1}^{L} \mathcal{V}^{(l)} \mid = \mathcal{O}(nc)$ e o número de arestas $\mid \sum_{l=1}^{L} \mathcal{E}^{(l)} \mid = \mathcal{O}(nc)$ uma vez que os métodos de construção de rede desenvolvidos fornecem grafos relativamente esparsos, ou seja, $(k \ll n)$. Observe que n denota o tamanho do conjunto de dados em termos de objetos e c a sua cardinalidade em termos de classes por objetos.

Algoritmo 1 CXN-MLL.

Require: k, $\mathcal{X}$, $\mathbf{x}$, λ, δ, τ

1: Construir os grafos a partir de $\mathcal{X}$ usando algum dos métodos de construção da rede para aprendizado multirrótulo (Seção 3.2).
2: Calcular um conjunto de medidas de redes complexas para cada grafo (Eqs. 16-18).
3: Inserir um item de teste $\mathbf{x}$ temporariamente nos grafos dos seus k vizinhos mais próximos.
4: Calcular as probabilidades geradas pelo classificador de alto nível através da variação das medidas de rede (Eqs. 22-23).
5: Combinar as probabilidades de alto nível com aquelas produzidas pelo classificador de baixo nível (Eq. 24).
6: Classificar $\mathbf{x}$ em todas as classes l que satisfazerem $\mathcal{M}_{\mathbf{x}}^{(l)} \geq \tau$

1. A complexidade de tempo para construir o grafo é $\mathcal{O}(n^2)$ já que é preciso calcular a distância euclidana entre todos pares de objetos.

2. A ordem de complexidade das medidas de rede encontra-se em $\mathcal{O}(nca)$ para assortatividade, $\mathcal{O}(nca^2)$ para coeficiente de agrupamento e $\mathcal{O}(na)$ para o grau médio com a denotando o grau médio. Tomando o maior valor, temos $\mathcal{O}(nca^2)$.

3. A complexidade de tempo da etapa de inserção está em $\mathcal{O}(nlog(n))$, pois precisamos encontrar os k vizinhos mais próximos do objeto de teste. Esta etapa tem complexidade linear no caso médio usando o algoritmo *quickselect*.

4. A complexidade de tempo associada ao cálculo da variação das medidas de rede está em $\mathcal{O}(ca^2)$ uma vez que podemos recalcular as medidas apenas para os vizinhos do objeto de teste.

5. Finalmente, a ordem de complexidade do classificador de alto nível é dada por $\mathcal{O}(n^2 + nca^2 + n\ log(n) + ca^2)$. Como normalmente temos $c \ll n$ e $a \ll n$ (grafos

esparsos), obtendo os resultados de termo de ordem mais alta em $\mathcal{O}(n^2)$. A complexidade final do CXN-MLL é $\mathcal{O}(n^2 + h(n, d, |L|))$ sendo h o classificador de baixo nível em função do objetos (n), atributos (d) e classes ($|L|$).

Capítulo 4

Resultados em Bases Artificiais

Neste capítulo são apresentadas simulações em duas bases de dados artificiais com objetivo de destacar algumas vantagens da nossa técnica em relação as outras técnicas de aprendizado multirrótulo tradicionais.

4.1 Base de Dados Toy 1

Considerando a base de dados da Fig. 2 apresentada na introdução, podemos ver que ambas classes de ∘/verde e △/vermelho corresponde a padrões claramente distintos. O primeiro forma o padrão de uma linha reta enquanto o segundo possui um padrão esférico. Além disso, a base de dados possui itens para teste, mostrados pelos □/pretos que continuam o padrão da linha reta e também se encaixam no padrão esférico. Técnicas tradicionais de classificação, como árvores de decisão, redes neurais, kNN e SVM, são muito mais propensas a classificar tais itens de teste na classe de △/vermelhos já que estes algoritmos consideram essencialmente apenas os atributos físicos dos dados (distância ou distribuição) e têm dificuldades em classificar os itens de teste de acordo com seu significado semântico (e.g., formação de padrão).

Mesmo no aprendizado multirrótulo sendo possível atribuir ambas classes ao item de teste, a maioria das técnicas desenvolvidas são fortemente baseadas em algoritmos de classificação monorrótulo, o que significa que elas compartilham não apenas suas vantagens, mas também suas limitações. Com o objetivo de demonstrar isso, foram realizados um experimento simples com o conjunto de dados da Fig. 2 considerando três técnicas de classificação multirrótulo: ML-kNN, CC(SVM) e BR(NB), além da técnica de alto nível proposta que foi combinada com o BR(NB) neste experimento. Em tal experimento, os itens de dados na Fig. 2 são classificados um por um, da esquerda para a direita, sendo o item 1 o □ mais a esquerda. Para manter a distribuição dos dados durante as simulações, sempre que um item de teste é classificado ele é incorporado no conjunto de treinamento com suas classes corretas e as etapas de treinamento e teste começam novamente. Para deixar claro, a Fig. 25 mostra as classes corretas para cada amostra da base toy usada nos

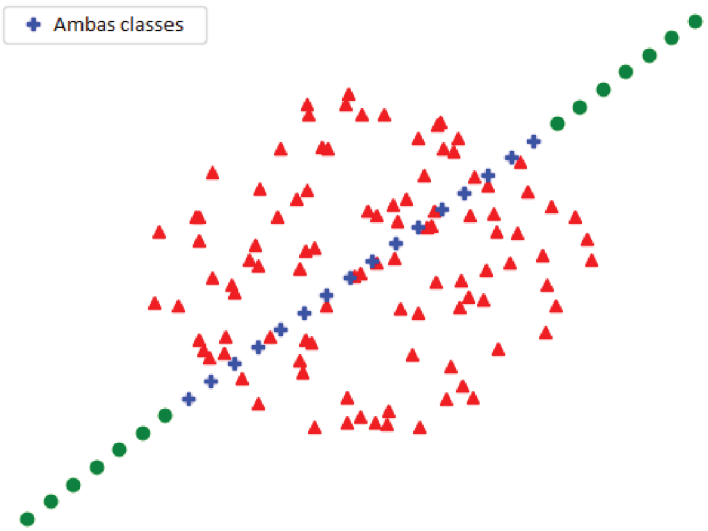
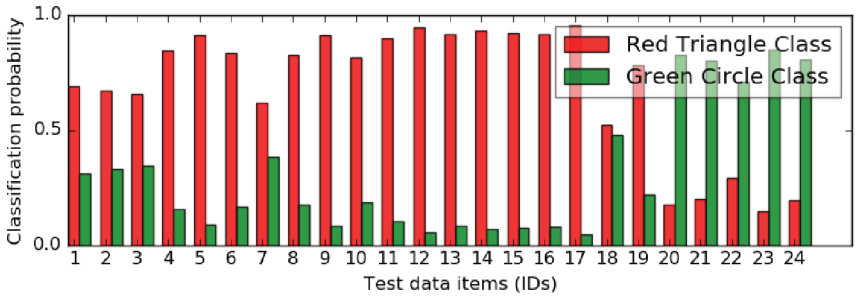


Figura 25 – A classe correta de cada amostra da base de dados da Fig. 2.

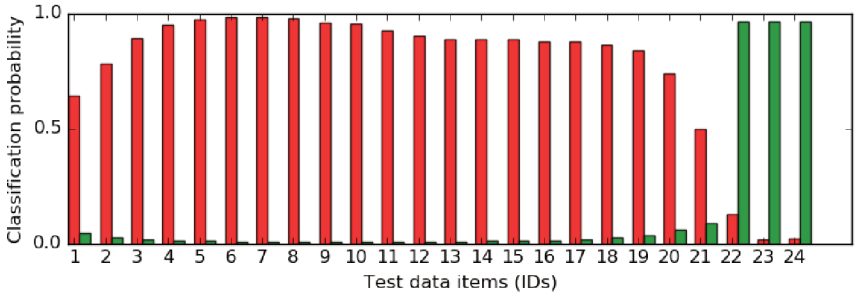
experimentos. Para estas análises foram utilizadas as medidas de rede assortatividade, coeficiente de agrupamento e grau médio, bem como foi definido $\tau = 0.5$ (threhsold) para atribuir uma classe ao objeto.

Os resultados obtidos pelas técnicas são mostrados na Fig. 26 a qual revela a dificuldade dos classificadores tradicionais de detectarem os padrões apresentados. Como mostrado na Fig. 26(a), ML-kNN foi capaz de classificar apenas os últimos cinco itens de teste de acordo com o padrão da linha reta (classe ∘ verde); todos os outros objetos foram atribuídos ao padrão esférico (classe △ vermelho), enquanto nenhum objeto foi associado a ambas classes. De modo similar, Figs. 26(b) e 26(c) mostram que as técnicas CC(SVM) e BR(NB) foram capazes de atribuir apenas os três últimos objetos mais a direita a classe da linha reta, novamente, não atribuindo nenhum objeto a ambas classes. Estes resultados são fáceis de explicar, basicamente, o ML-kNN, CC(SVM) e BR(NB) não consideram relações topológicas entre os itens de dados, apenas suas características físicas.

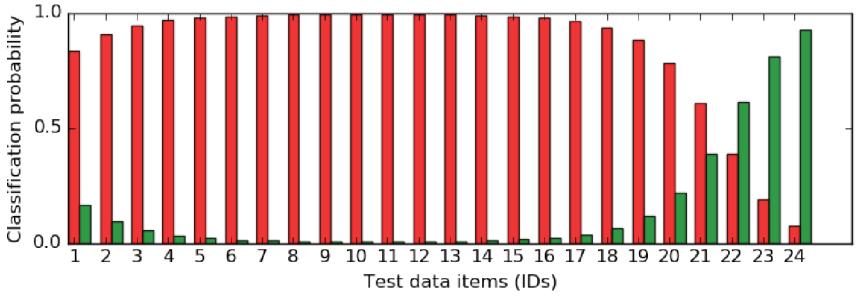
Ao contrário, os resultados da técnica proposta, que são mostrados na Fig. 26(d), identifica os padrões relacionados a ambas classes. Neste experimento, foram adotados $\lambda = 0.8$, o que significa que o algoritmo de alto nível teve uma maior contribuição na probabilidade final (80%). Analisando a figura, é possível notar que todos os objetos foram classificados no padrão da linha reta. Interessante notar também que os itens de teste que estavam em conformidade com o padrão esférico também foram atribuídos a essa classe. Logo, tais resultados fornecem evidências sobre as desvantagens relacionadas às técnicas tradicionais multirrótulo e estabelecem uma importante motivação para o projeto e desenvolvimento de novos algoritmos para o aprendizado multirrótulo, incluindo aqueles baseados na teoria de redes complexas. Além disso, como essas técnicas também consideram apenas as características físicas dos dados, tais combinações nos fornecem resultados semelhantes aos apresentados na figura.



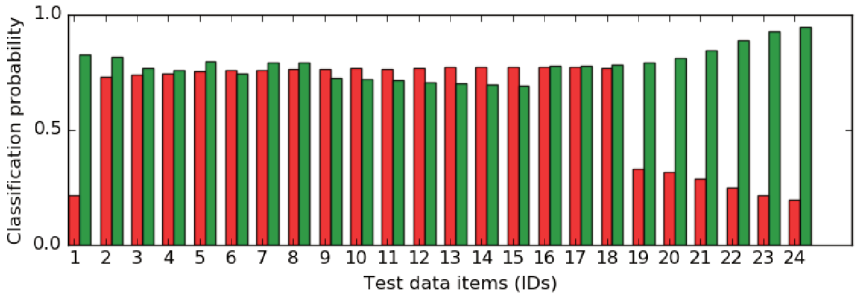
(a) ML-kNN



(b) CC (SVM)



(c) BR (NB)



(d) High-Level + BR(NB)

Figura 26 – Probabilidades obtidas pelos algoritmos de baixo e alto nível para a base de dados artificial apresentada na Fig. 2.

Por fim, é importante deixar claro que o objeto de teste é inserido como uma instância de treinamento após ser classificado apenas nas simulações artificiais, o que implica que a ordem em que os objetos são testados não é relevante para a aplicação prática do nosso algoritmo em bases reais.

4.2 Base de Dados Toy 2

A Fig. 27 denota um conjunto de dados composto por duas classes cujas instâncias são marcadas como Verde/∘ e Vermelho/△. Ambas as classes correspondem a padrões de formação claros. O conjunto de dados também possui itens de teste, denotados por marcadores em Black/□, que são a continuação do padrão Green/∘, embora alguns deles também façam parte do padrão Red/△. Essa base de dados artificial simples foi considerada nos experimentos desta seção. Da mesma forma que o exemplo anterior, os experimentos consistem em classificar cada item de teste, um a um, da esquerda para a direita, sendo que sempre que um item de dado é classificado, ele é adicionado ao conjunto de treinamento com seu(s) rótulo(s) reai(s) correspondente(s), e o treinamento e as fases de teste são reiniciados.

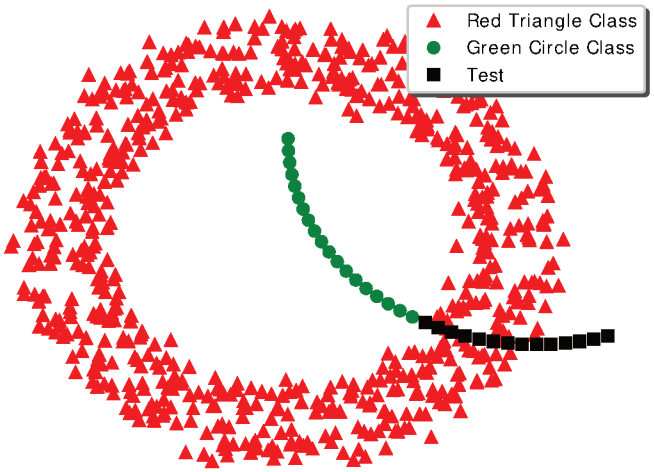


Figura 27 – Uma base de dados artificial que enfatiza a dificuldade dos algoritmos tradicionais de classificação multirrótulo e reforça as características salientes da nossa técnica.

Figs. 28(a)-(d) mostram que técnicas multirrótulo amplamente utilizadas, como MLkNN, CC e BR, mesmo adotando classificadores base de última geração como o SVM, têm dificuldade em detectar corretamente os padrões associados às duas classes. Ao observar apenas as características físicas dos dados (por exemplo, distribuição ou proximidade), tais técnicas têm uma forte tendência de classificar os itens de teste na classe Vermelho/△,

(a) BR (SVM) (b) BR (NB)

(c) CC (SVM) (d) ML-kNN

(e) High-Level + CC(SVM) (f) High-Level + MLkNN

Figura 28 – Probabilidades obtidas pelos algoritmos de baixo e alto nível para a base de dados artificial apresentada na Fig. 27.

conforme revelado pelas figuras, ignorando completamente o forte padrão de formação apresentado pelas instâncias Green/∘ com as quais os itens de teste são muito compatíveis. Na verdade, a maioria das técnicas convencionais multirrótulo apresentam esta mesma limitação relacionada às características físicas, pois são muito baseadas em princípios da aprendizagem monorrótulo.

Agora analisamos a técnica de alto nível no mesmo conjunto de dados. Considerando três medidas de rede, assortatividade, coeficiente de agrupamento e comprimento médio do caminho e os seguintes parâmetros fixos, $k = 3$, $\delta = \frac{1}{3}$, $\lambda = 0.8$ e $\tau = 0.5$, avaliamos a técnica de alto nível em combinação com CC(SVM) e MLkNN, que apresentaram respectivamente o pior e melhor resultado na análise anterior: CC(SVM) não obteve classificação correta em relação a classe Verde/∘, enquanto MLkNN classificou corretamente quatro itens de teste nessa classe. Como mostrado por Figs. 28(e) e 28 (f), houve muitas melhorias em relação à associação correta dos itens de teste com os dois padrões de classe. Tais resultados enfatizam as motivações do nosso trabalho, que é especialmente focado

em contornar as limitações das técnicas multirrótulo atuais, provendo uma ferramenta para análise de padrões estruturais em problemas de aprendizado multirrótulo.

Capítulo 5

Resultados em Bases Reais

Neste capítulo são discutidos resultados de análises exploratórias do classificador de alto nível considerando especialmente a influência de seus principais hiperparâmetros e das medidas de rede. Também são apresentados resultados obtidos pela técnica considerando seleção de parâmetros e diferentes métricas de desempenho.

5.1 Análise Exploratória de CXN-MLL e seus Hiperparâmetros

Nesta seção será detalhado o ambiente experimental em termos de bases de dados, hiperparâmetros, técnicas e medidas utilizadas na análise dos parâmetros do algoritmo. Em relação às bases de dados utilizadas, a seleção foi feita para abranger uma diversidade de domínios de dados. Um resumo numérico sobre os conjuntos de dados em termos de instâncias, características e rótulos é apresentado na Tabela 9. A tabela também apresenta a cardinalidade e densidade de cada conjunto de dados, que são medidas que dizem sobre "quão multi-rótulo é um problema" e que calculam respectivamente o número médio de rótulos por exemplos e o número médio de rótulos em função do número de rótulos possíveis da base. A divisão dos dados em treinamento e conjunto de teste, que segue (Szymański; Kajdanowicz, 2017), também é apresentada na tabela. Os resultados e a influência dos parâmetros foram obtidos usando as bases Birds, Emotions, Scene e Yeast.

Tabela 9 – Breve descrição das bases de dados reais em termos de domínio, número de instâncias, quantidade de rótulos, características, objetos de treino, objetos de teste, cardinalidade e densidade de rótulo.

Base	Domínio	#Inst.	#Atrib.	#Rótulos.	Card.	Dens.	#Treino	#Teste
Birds	Áudio	645	258	19	1.014	0.053	322	323
Emotions	Música	593	72	6	1.869	0.311	391	202
Scene	Imagem	2407	294	6	1.074	0.179	1211	1196
Yeast	Biologia	2417	103	14	4.237	0.303	1500	917

Os classificadores multirrótulo convencionais selecionados para este estudo foram BR, CC e ML-kNN. Com essa seleção, cobrimos técnicas de ambas as categorizações discutidas na seção de revisão, ou seja, transformação de problemas (BR e CC) e adaptação de algoritmo (ML-kNN), e de estratégias de primeira ordem (BR e ML-kNN) e de alta ordem (CC). Como BR e CC transformam o problema multirrótulo em um problema monorrótulo, eles também requerem um classificador base, que virá da literatura de classificação convencional. Nesses experimentos, dois classificadores básicos foram avaliados: Naive Bayes que é uma técnica simples e amplamente utilizada na aprendizagem monorrótulo; e SVM, que tem sido uma técnica de classificação estado da arte para muitos domínios. Em relação aos parâmetros, adotamos os valores recomendados pelos autores em seus artigos. Assim, os parâmetros do ML-kNN foram definidos como $k = 10$ (número de vizinhos) e $s = 1.0$ (parâmetro de suavização). BR não tem nenhum parâmetro além do classificador base, bem como CC, uma vez que definimos a ordem da cadeia como a ordem dos rótulos. Sobre os classificadores básicos, usamos o Naive Bayes Gaussiano; e definimos a função radial como núcleo da SVM. Também avaliamos um pequeno intervalo de valores para os parâmetros kernel (μ) e custo ($\mathcal{C}$) na SVM, mas como os resultados foram próximos, mantivemos os valores padrões de (Szymański; Kajdanowicz, 2017).

Sobre nossa técnica de alto nível, nós consideramos as variações de três parâmetros em nossos experimentos. O primeiro parâmetro derivado de $\mathcal{H}$ é o peso das medidas de rede δ. Como selecionamos três medidas de rede para nossos experimentos, chamadas assortatividade, coeficiente de agrupamento e grau médio, δ foi otimizado no seguinte intervalo $\{(0.1, 0.1, 0.8), (0.1, 0.2, 0.7), \ldots, (0.8, 0.1, 0.1)\}$, que garante que $\sum_{u=1}^{3} \delta(u) = 1$. O segundo parâmetro que é relacionado a combinação convexa do classificador multirrótulo tradicional e do de alto nível, representado pelo λ na Eq. (23), é otimizado no intervalo $\{0, 0.1, \ldots, 1.0\}$, onde $\lambda = 0.8$ significa uma contribuição de 80% do classificador de alto nível na probabilidade final. O último parâmetro chamado τ é o threshold que a probabilidade final deve atingir para que o objeto receba um determinado rótulo. Tal parâmetro é otimizado no intervalo $\{0, 0.1, \ldots, 1.0\}$. Em nossas simulações, a distância euclidiana foi adotada como a medida de dissimilaridade assim como a acurácia foi adotada como medida de avaliação.

A Tabela 10 mostra os resultados dos classificadores multirrótulo convencionais $\mathcal{C}$ e sua respectiva combinação com o classificador de alto nível via medidas de rede complexas, denotado por $\mathcal{M}$. Ao contrário da acurácia de subconjunto, onde apenas os objetos que tiveram seu conjunto de rótulos perfeitamente preditos são levados em consideração, a medida de acurácia usada neste trabalho leva em consideração cada rótulo que foi predito corretamente, o que fornece uma representação mais precisa do desempenho do modelo.

A seguir, analisamos o desempenho das técnicas tradicionais multirrótulo e da técnica de alto nível. Na Tabela 10 é possível notar que os melhores resultados variam de técnica para técnica de acordo com os conjuntos de dados, o que enfatiza a diversidade de nossa

seleção em termos de bases de dados e técnicas. Por exemplo: ML-kNN obteve os melhores resultados para os conjuntos de dados Scene e Yeast, mas o pior resultado para Emotions; BR e CC usando SVM como o classificador base alcançaram o melhor resultado para Birds, seguidas de perto por ML-kNN; e a estratégia Classifier Chain usando Naive Bayes como o classificador base retornou o melhor resultado para Emotions, embora também os piores resultados para o conjunto de dados Yeast.

Os resultados na Tabela 10 mostraram que a técnica multirrótulo de alto nível contribuiu para melhorar o desempenho dos classificadores multirrótulos tradicionais. Analisando cada conjunto de dados separadamente, pode-se notar que a combinação resultou em melhora nos resultados na base para todos os cinco algoritmos; o mesmo aconteceu em Birds (embora com melhora muito pequena), onde um valor alto para o parâmetro τ melhorou os resultados dos algoritmos convencionais; nos conjuntos de dados Scene e Yeast, a técnica proposta não foi capaz de melhorar consideravelmente os resultados do ML-kNN, embora tenha conseguido isso para as outras quatro técnicas avaliadas.

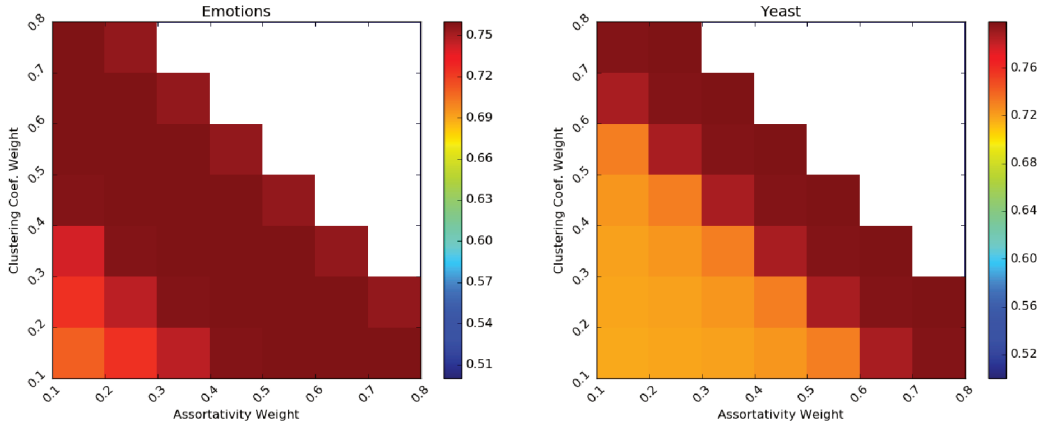
Tabela 10 – Valores de acurácia do classificador baixo nível $\mathcal{C}$ e de sua respectiva combinação com o classificador de alto nível $\mathcal{M}$ para cada base de dados. Os melhores resultados obtidos para cada base de dados estão destacados em negrito.

Alg.	Birds		Emotions		Scene		Yeast	
	$\mathcal{C}$	$\mathcal{M}$	$\mathcal{C}$	$\mathcal{M}$	$\mathcal{C}$	$\mathcal{M}$	$\mathcal{C}$	$\mathcal{M}$
ML-kNN	94.6	**94.9**	69.6	72.1	90.9	**91.0**	79.0	79.1
BR (SVM)	94.8	**94.9**	73.5	74.3	85.9	90.1	76.7	**79.8**
CC (SVM)	94.8	**94.9**	73.5	74.5	86.3	88.8	76.7	79.1
BR (NB)	74.1	**94.9**	72.7	74.2	75.6	85.7	69.9	72.4
CC (NB)	75.3	**94.9**	73.9	**75.9**	79.3	86.1	68.6	70.5

Outro ponto para nossa análise é a complexidade inerente a cada conjunto de dados. Apesar da Emotions ter 6 rótulos (ver Tabela 9), a acurácia obtida em tal conjunto de dados é geralmente menor do que a obtida para as outras bases de dados (com 14 ou 19 rótulos, por exemplo). Isso pode ser parcialmente explicado por duas métricas multirrótulo, cardinalidade e densidade de rótulo, que informam que tal conjunto de dados tem mais ocorrências de itens multirrótulo do que Birds e Scene, por exemplo. O conjunto de dados Yeast ainda tem um valor de cardinalidade maior do que Emotions. Curiosamente, a técnica multirrótulo de alto nível obteve uma melhoria em ambos os conjuntos de dados. Portanto, tal resultado sugere que a classificação por meio de medidas de rede pode ser uma técnica promissora para obter melhor desempenho em um cenário difícil.

Agora, analisamos a influência de cada parâmetro da técnica multirrótulo de alto nível. O primeiro parâmetro que analisamos é o peso das medidas de rede δ. Para tal análise, pegamos o melhor resultado para cada conjunto de dados (ver Tabela 10) e variamos

os pesos das medidas de rede para avaliar o desempenho preditivo da técnica. Observe que nós não alteramos os valores λ e τ na análise. Figura 29 mostra os mapas de calor de cada medida de rede para as bases de dados Emotions e Yeast. Para interpretar os mapas de calor, é necessário considerar o complemento da soma entre os pesos da assortatividade (eixo x) e do coeficiente de agrupamento (eixo y) como o peso do grau médio. Pode-se observar que a assortatividade e o coeficiente de agrupamento têm contribuição equivalente em ambos os conjuntos de dados, com valores maiores de ambas as medidas de rede proporcionando melhores resultados do que valores maiores de grau médio. Na verdade, à medida que o peso do grau médio aumenta, piores são os resultados preditivos. Nossa análise com os conjuntos de dados Birds e Scene não são exibidos devido à pequena diferença em termos de desempenho ao variar os pesos das medidas de rede, ou seja, as três medidas de rede são muito equivalentes em tais conjuntos de dados.

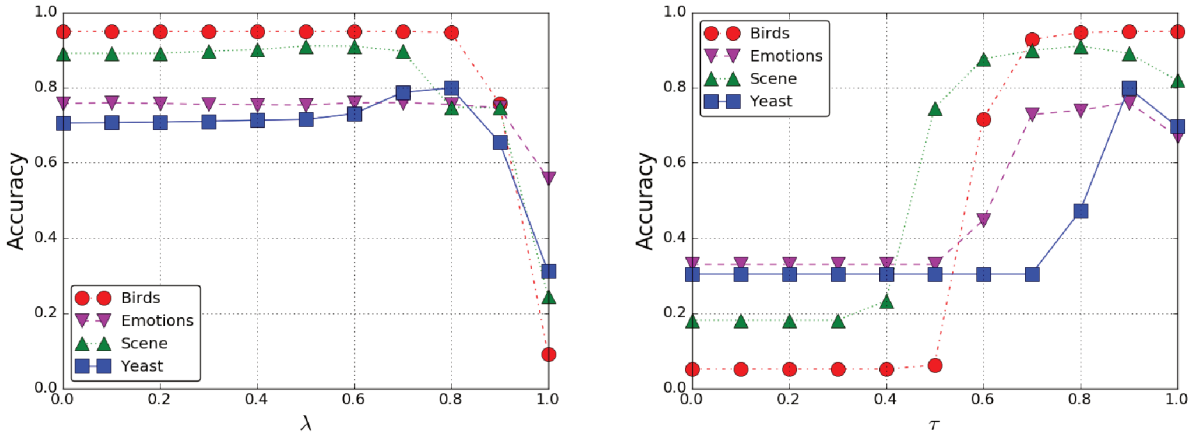


(a) Emotions (b) Yeast

Figura 29 – Análise da influência das medidas de rede através do parâmetro δ .

O segundo parâmetro analisado é o λ, que denota a combinação linear convexa entre as associações produzidas por ambos os classificadores. Novamente, considerando o melhor resultado para cada conjunto de dados na Tabela 10 e variando apenas os valores de λ para avaliar o desempenho preditivo da técnica. Os resultados são mostrados pela Fig. 30(a) que demonstra que λ causou uma melhoria insignificante para Birds e Emotions, pequena melhoria para Scene e melhoria considerável para o conjunto de dados de Yeast. Observe que os resultados de $\mathcal{M}$ com $\lambda = 0$ (ou seja, sem qualquer contribuição do classificador de alto nível) pode ser diferente dos resultados de $\mathcal{C}$ (classificador de baixo nível) como o pós-processamento de $\mathcal{M}$ inclui uma etapa adicional relacionada à aplicação do threshold τ. Outro ponto que se pode observar nesta figura é que $\lambda = 1.0$ (apenas o termo de alto nível é adotado) fornece os piores resultados preditivos, que podem ser facilmente explicados: apesar das medidas de rede poderem detectar propriedades topológicas dos dados, as informações das técnicas tradicionais continuam muito importantes,

pois detectam propriedades físicas dos dados relacionadas à distância ou distribuição, por exemplo.



(a) λ parameter (b) τ parameter

Figura 30 – Análise da (a) combinação linear do parâmetro λ; e (b) do threshold τ.

O threshold τ é o último parâmetro que avaliamos aqui. Para realizar sua análise, executamos o mesmo processo adotado para δ e λ. A Fig. 30(b) apresenta os resultados da variação do parâmetro. É possível notar que valores baixos de τ resultam em um desempenho ruim. A figura sugere claramente que os melhores valores para τ ficaram entre 0.8 e 0.9.

5.2 Análise Exploratória das Medidas de Rede

Sabendo que a técnica de alto nível pode usar qualquer combinação de medidas de rede, nesta seção serão avaliadas a performance do modelo frente a diversas combinações de medidas de redes complexas. A seguir, apresentamos os resultados obtidos pela técnica de alto nível considerando os mesmos conjuntos de dados da seção anterior, bem como os mesmos algoritmos e parâmetros para a classificação de baixo nível.

Em relação à técnica de alto nível, consideramos as variações de três parâmetros em nossos experimentos tomando como referência os resultados e conclusões obtidos na seção anterior. O primeiro parâmetro inerente a $\mathcal{H}$ é o peso das medidas de rede δ. Como consideramos a análise de três medidas de rede por vez na técnica de alto nível, δ foi otimizado no seguinte intervalo $\{(0.1, 0.1, 0.8), (0.1, 0.2, 0.7), \ldots, (0.8, 0.1, 0.1)\}$, o que garante que $\sum_{u=1}^{3} \delta(u) = 1$. O segundo parâmetro que está relacionado com a combinação convexa de ambos os classificadores tradicionais multirrótulo e de alto nível, denotados por λ em (23), é otimizado no intervalo $\{0, 0.1, \ldots, 0.8\}$, onde $\lambda = 0.8$ significa uma contribuição de 80 % do termo de alto nível na previsão final. O último parâmetro denominado τ é um limite inferior que a probabilidade final deve atingir para que o item de teste seja assoaciado à uma classe sob avaliação. Esse parâmetro é otimizado no

intervalo $\{0.5, 0.6, \ldots, 0.9\}$. Em relação às medidas de redes complexas consideradas em nossos experimentos, a Tabela 11 apresenta as quatro combinações derivadas das quatro medidas aqui investigadas: coeficiente de agrupamento, assortatividade, grau médio e comprimento médio do caminho. Mais uma vez, usamos a distância euclidiana como medida de dissimilaridade em todas as simulações.

Tabela 11 – IDs para as diferentes associações entre as medidas de redes complexas.

ID	Medida de Rede Complexa
A	Coeficiente de Agrupamento
B	Assortatividade
C	Grau Médio
D	Comprimento Médio do Caminho
ABC	Coeficiente de Agrupamento, Assortatividade e Grau Médio
ABD	Coeficiente de Agrupamento, Assortatividade e Comprimento Médio do Caminho
ACD	Coeficiente de Agrupamento, Grau Médio e Comprimento Médio do Caminho
BCD	Assortatividade, Grau Médio e Comprimento Médio do Caminho

A Tabela 12 mostra a acurácia obtida por técnicas convencionais e de alto nível nas bases de dados reais apresentadas anteriormente. Na tabela, a coluna ΔG refere-se às medidas de redes complexas levadas em consideração na análise de alto nível, conforme mostrado pela Tabela 11. Também nessa coluna, o símbolo "-" indica que apenas as técnicas de baixo nível foram consideradas, o que significa que esses são os resultados sem o termo de classificação de alto nível. Na tabela, é possível notar que o termo de alto nível foi capaz de melhorar o desempenho preditivo de todos os algoritmos convencionais. De maneira geral, a técnica de alto nível obteve a maior contribuição em termos de desempenho preditivo quando combinada com os métodos BR/CC usando NB como classificador base, o que indica que tal classificador pode ter dificuldade em detectar algumas relações complexas entre os itens de dados; por outro lado, a combinação com MLkNN proporcionou a menor contribuição, o que pode ser explicado pela relação entre as informações de vizinhança consideradas por ambos os métodos.

A Tabela 12 também mostra que as diferentes combinações de medidas de rede geralmente fornecem resultados próximos. Os melhores resultados globais foram fornecidos pela configuração "ABC" das medidas de rede, que equivalem a adotar a técnica de alto nível com as medidas de coeficiente de agrupamento, assortatividade e grau médio. No entanto, esta mesma configuração "ABC" apresentou os piores resultados em comparação com outras configurações de medidas de rede, já que sua combinação com BR/CC (NB) forneceu uma acurácia muito menor na bases de dados Birds.

Agora analisamos a contribuição de cada medida de rede em função do parâmetro δ. Em relação aos demais parâmetros, os experimentos foram conduzidos fixando os valores de λ e τ em 0.8 e 0.85, após considerar a análise do comportamento de tais

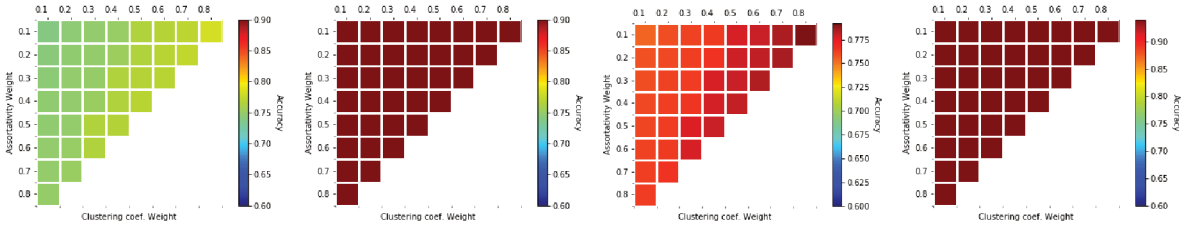
Tabela 12 – Valores de acurácia obtidos para cada algoritmo e combinação das medidas de rede (A: coeficiente de agrupamento, B: assortatividade, C: grau médio, D: comprimento médio do caminho).

Alg.	ΔG	Birds	Emot.	Scene	Yeast
BR(NB)	-	74.1	72.8	75.7	69.9
BR(SVM)	-	94.7	67.5	91.6	81.3
CC(NB)	-	75.4	73.9	79.3	68.7
CC(SVM)	-	94.7	67.5	90.9	76.4
MLkNN	-	94.6	69.6	91.0	79.1
BR(NB)	ABC	81.0	74.9	88.4	72.5
BR(SVM)	ABC	94.8	75.4	**92.2**	**81.4**
CC(NB)	ABC	81.9	**76.2**	87.8	70.6
CC(SVM)	ABC	94.8	75.6	91.6	76.6
MLkNN	ABC	**94.9**	72.2	91.2	79.3
BR(NB)	ABD	94.7	74.9	88.3	76.2
BR(SVM)	ABD	94.8	75.5	**92.2**	81.3
CC(NB)	ABD	94.7	**76.2**	87.8	76.2
CC(SVM)	ABD	94.8	75.6	91.6	79.6
MLkNN	ABD	**94.9**	72.1	91.2	79.3
BR(NB)	ACD	94.7	74.0	87.5	76.2
BR(SVM)	ACD	94.8	75.4	92.1	81.3
CC(NB)	ACD	94.7	75.7	87.0	76.2
CC(SVM)	ACD	94.8	75.5	91.3	79.6
MLkNN	ACD	**94.9**	72.1	91.2	79.3
BR(NB)	BCD	94.7	74.9	88.3	76.2
BR(SVM)	BCD	94.8	75.6	**92.2**	81.3
CC(NB)	BCD	94.7	76.0	87.8	76.2
CC(SVM)	BCD	94.8	75.6	91.6	79.6
MLkNN	BCD	**94.9**	72.0	91.2	79.3

parâmetros nos resultados da Tabela 12. Para facilitar a interpretação visual, adotamos uma representação de mapa de calor, em que a contribuição (δ) de cada medida de rede é mapeada diretamente: valores no eixo a denotam a contribuição da primeira medida de rede, valores no eixo b a contribuição da segunda, e $c_{ij} = 1 - a_i + b_j$ é a contribuição da terceira medida de rede em relação a cada célula do mapa de calor. Por exemplo, quando os pesos (contribuições) da primeira e da segunda medidas de rede são respectivamente $a_1 = 0.1$ e $b_1 = 0.1$, o peso da terceira medida de rede é $c_{11} = 0.8$. A seguir, discutiremos aspectos relevantes de nossos resultados, levando em consideração as bases de dados reais consideradas neste estudo.

Conforme mostrado na Tabela 12, a influência das medidas de rede nos conjuntos de dados Birds foram principalmente relacionadas ao classificador base NB. Tomando os métodos BR/CC (NB), as Figs. 31(a) e 31(c), mostram que a configuração "ABC" fornece

os piores resultados entre as técnicas de alto nível para tais conjuntos de dados. Por outro lado, qualquer configuração que considere a medida do comprimento médio do caminho é capaz de fornecer um resultado muito melhor, como mostrado pelas Figs. 31(b),31(d). Por outro lado, ao considerar BR/CC (SVM) ou MLkNN em ambos os conjuntos de dados, as configurações alcançam resultados semelhantes.

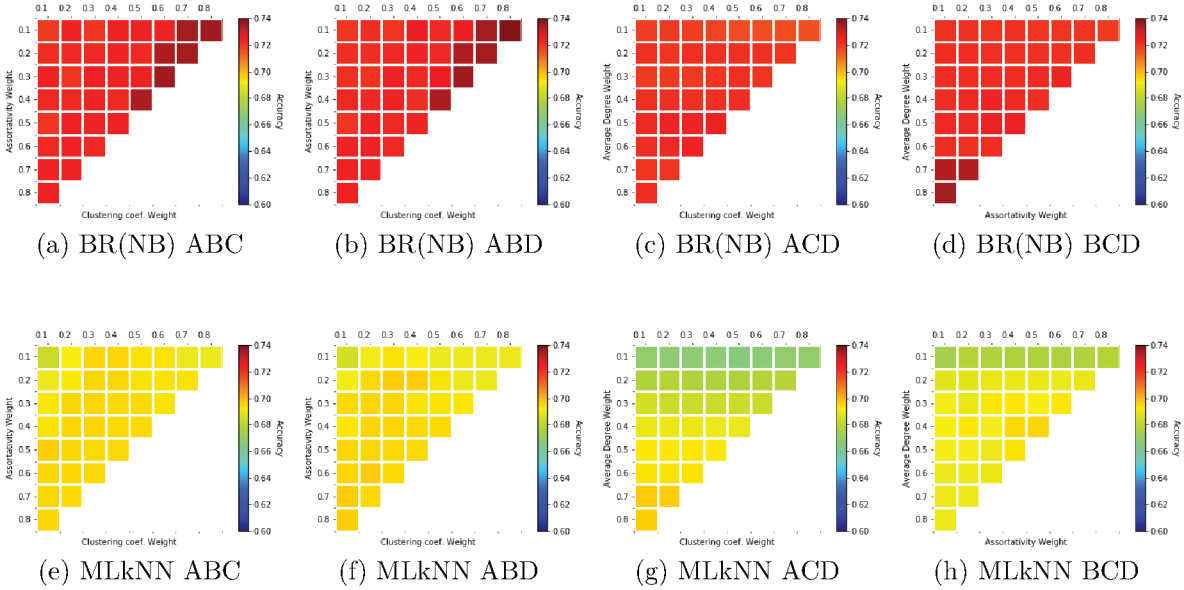


(a) BR(NB) ABC (b) BR(NB) ABD (c) CC(NB) ABC (d) CC(NB) ABD

Figura 31 – Análise das medidas de rede em função do parâmetro δ na base Birds.

As Figuras 32(a)-(d) mostram que a medida de assortatividade contribui para alcançar os melhores resultados para BR(NB) na base Emotions, especialmente ao considerar contribuições baixas (≤ 0.4) para tal medida. Observe que o mesmo comportamento foi observado para outras técnicas avaliadas neste estudo, como os resultados do MLkNN apresentados pelas Figs. 32(e)-(h), que também indicam vantagem para as configurações ABC e ABD, especialmente com altas contribuições da medida de assortatividade (≥ 0.5).

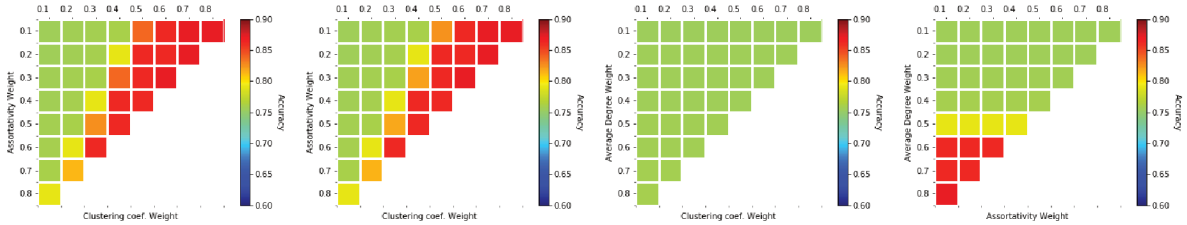
No conjunto de dados Scene, as Figs. 33(a)-(b) indicam que uma maior contribuição



(a) BR(NB) ABC (b) BR(NB) ABD (c) BR(NB) ACD (d) BR(NB) BCD

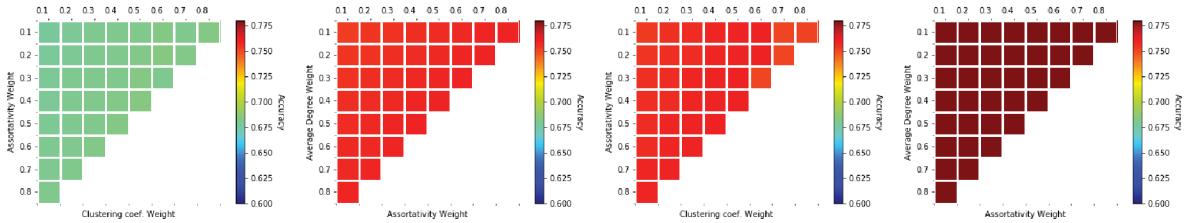
(e) MLkNN ABC (f) MLkNN ABD (g) MLkNN ACD (h) MLkNN BCD

Figura 32 – Análise das medidas de rede em função do parâmetro δ na base Emotions.



(a) BR(NB) ABC (b) BR(NB) ABD (c) BR(NB) ACD (d) BR(NB) BCD

Figura 33 – Análise das medidas de rede em função do parâmetro δ na base Scene.



(a) CC(NB) ABC (b) CC(NB) BCD (c) CC(SVM) ABC (d) CC(SVM) BCD

Figura 34 – Análise das medidas de rede em função do parâmetro δ na base Yeast.

do coeficiente de agrupamento (≥ 0.7) é necessária para obter os melhores resultados de acurácia ao considerar as configurações ABC e ABD para a técnica BR/CC (NB). Além disso, as Figs. 33(c)-(d) mostra que a assortatividade também desempenha um papel relevante aqui, visto que a ausência de tal medida na configuração ACD diminui consideravelmente os resultados de acurácia. Observe que ao considerar BR/CC (SVM) ou MLkNN em tal conjunto de dados, as configurações alcançam resultados semelhantes.

As combinações das medidas de rede foram significativas para BR/CC (NB), CC(SVM) e MLkNN no conjunto de dados Yeast. As Figuras 34(a)-(d) mostram que a medida do comprimento médio do caminho pode melhorar os resultados de acurácia aqui.

Diferente da Tabela 12, em que os resultados apontaram a configuração ABC como a melhor, as análises das medidas de rede complexas aqui apresentadas indicam que a configuração ABD também é capaz de fornecer bons resultados, principalmente em situações onde a configuração ABC falha, como por exemplo nas bases Birds e Emotions. Tais análises também demonstraram que a medida de assortatividade tem notável relevância para alguns conjuntos de dados. Porém, pelos resultados aqui apresentados, temos evidências de que não existe uma medida única que forneça as melhores soluções; i.e., problemas diferentes podem exigir propriedades de rede diferentes para que seus padrões sejam detectados com eficiência. Isso torna nossa investigação ainda mais valiosa, pois a literatura apresenta poucas contribuições relacionadas à análise de medidas de redes complexas na classificação de dados.

5.3 Análise do Desempenho Preditivo de CXN-MLL

Nesta seção são discutidos os principais resultados obtidos pela técnica CXN-MLL considerando sete bases de dados reais e três métricas de desempenho distintas. Além disso, todas as simulações apresentadas consideram uma estratégia adequada para a seleção de hiperparâmetros dos classificadores de baixo e alto nível, bem como são discutidas a partir de análises estatísticas. As bases de dados consideradas nessa seção estão descritas na Tabela 13. Note que em relação à Tabela 9 foram adicionados mais três conjuntos de dados: Enron, Genbase e Medical.

Tabela 13 – Breve descrição das bases de dados reais em termos de domínio, número de instâncias, quantidade de rótulos, características, objetos de treino, objetos de teste, cardinalidade e densidade de rótulo.

Base	Domínio	#Inst.	#Atrib.	#Rótulos	Cardinalidade	Densidade	#Treino	#Teste
Birds	Áudio	645	258	19	1.014	0.053	322	323
Emotions	Música	593	72	6	1.869	0.311	391	202
Enron	Texto	1702	1001	53	3.378	0.064	1123	579
Genbase	Biologia	662	1186	27	1.252	0.046	463	199
Medical	Texto	978	1449	45	1.245	0.028	333	645
Scene	Imagem	2407	294	6	1.074	0.179	1211	1196
Yeast	Biologia	2417	103	14	4.237	0.303	1500	917

Para fins de comparação, avaliamos a técnica proposta com três técnicas multirrótulo amplamente adotadas: BR, CC e MLkNN. Ao transformar um problema multirrótulo em problemas de classificação binária, BR e CC requerem um classificador base para lidar com tais problemas. Nessas simulações, três classificadores básicos foram avaliados: Naive Bayes (NB) que é uma técnica simples e amplamente utilizada no aprendizado monorrótulo; além de Random Forest (RF) e Support Vector Machine (SVM), que são técnicas de classificação do estado-da-arte para muitos domínios.

Para fornecer uma comparação justa, consideramos uma ampla gama de parâmetros para as técnicas. Para BR e CC, o número de árvores do RF é selecionado no conjunto $\{2^4, 2^6, 2^8, 2^{10}\}$; a função do kernel em SVM pode ser {linear, rbf }, com o parâmetro de penalidade sendo selecionado sobre o conjunto $\{2^{-5}, 2^{-3}, \ldots, 2^3\}$; e em NB, a probabilidade é assumida como gaussiana. Para MLkNN, o número de vizinhos é selecionado no conjunto $\{5, 10, \ldots, 30\}$.

Em relação à nossa técnica de alto nível, na fase de construção do grafo, selecionamos o parâmetros k e ϵ do grafo kNN e ϵN respectivamente sobre os valores $\{2, 3, \ldots, 10\}$ e $\in \{0.1, 0.2, \ldots, 0.5\} \cdot \bar{d}$, em que $\bar{d}$ é a distância média entre todos os pares de amostras. Para a fase de classificação, temos λ e τ. O primeiro é selecionado sobre o conjunto $\{0.1, 0.2, \ldots, 1.0\}$, em que $\lambda = 0.7$, por exemplo, significa uma contribuição de 70 % do termo de alto nível na previsão final. O último é selecionado sobre o conjunto $\{0.5, 0.6, \ldots, 0.9\}$ e indica o valor mínimo de associação que deve ser atingido para que o

rótulo seja atribuído a um item de teste. Três medidas de redes complexas foram selecionadas a partir dos resultados obtidos na seção anterior: assortatividade, coeficiente de agrupamento e grau médio. A contribuição de cada uma delas foi fixada respectivamente em 0.4, 0.4 e 0.2.

Vale a pena mencionar que, como os conjuntos de dados já estavam divididos em conjuntos de treinamento e teste, selecionamos todos os parâmetros executando o método *Grid Search* a partir da estratégia de validação cruzada K-Fold, com k = 5, exclusivamente no conjunto de treinamento. Além do mais, um tratamento especial foi feito para que os algoritmos fossem treinados nas bases de dados Enron, Genbase e Medical, pois as mesmas contam com classes pouco representadas, contendo inclusive alguns casos de rótulos que não foram representados por nenhum objeto no treino. Para estas bases, com o objetivo de manter o máximo as classes, foi feito um pré-processamento onde foram removidos todos os rótulos que não estão contidos em pelo menos 1% do conjunto de treino. Além do mais, para essas bases de dados foi utilizada como medida de similaridade do classificador de alto nível o índice de Jaccard, já que o seu espaço de atributos é binário. A seguir, avaliamos o desempenho preditivo das técnicas em três métricas de AMR: acurácia, acurácia de subconjunto e F_1 weighted. Para fins de clareza, o valor da acurácia usada aqui é o complemento da métrica de perda de Hamming (ZHANG; ZHOU, 2014).

5.3.1 Análise Preditiva em Termos de Acurácia

A Tabela 14 apresenta os resultados de acurácia das técnicas multirrótulo em comparação. O termo "LL Alg." na tabela representa os algoritmos de baixo nível considerados (BR, CC e MLkNN); "LL Base" mostra o classificador base equipado com a técnica de transformação do problema; e "HL Graph" indica qual método de construção do grafo foi adotado para realizar a classificação de alto nível. Observe que o símbolo "-" tem significados diferentes na tabela: em "LL Base" significa que MLkNN não requer um classificador base; e no "Grafo HL" significa que a classificação foi realizada exclusivamente por uma técnica de baixo nível (ou seja, nenhum termo de alto nível). Levando em consideração os algoritmos de baixo nível e os classificadores base, os melhores resultados locais na tabela estão sublinhados e os melhores resultados globais estão em negrito. Para o conjunto de dados Birds, os melhores resultados foram alcançados por BR/CC com RF como classificador base combinado com a classificação HL fornecida a partir do grafo S-kNN. Para o conjunto de dados Emotions, CC (SVM) obteve os melhores resultados depois de ser combinado com a classificação HL fornecida pelos grafos kNN e kNN + ϵN. Na base de dados Enron, o melhor resultado foi obtido através da técnica de alto nível combinada ao BR(SVM) utilizando qualquer método de construção da rede. Já na base Genbase, com exceção dos classificadores que usaram NB como base, todas as outras técnicas alcançaram o máximo de acurácia combinadas à tecnica de alto nível, com exceção do BR(SVM) que piorou um pouco seu resultado na combinação. Curiosamente a técnica de alto nível

aumentou bastante a performance do MLkNN neste conjunto de dados. Na base Medical, o melhor resultado foi obtido utilizando o CC(SVM) sem o termo de alto nível. Para o conjunto de dados Scene, o BR(SVM) forneceu os melhores resultados quando combinado com HL (kNN + ϵN) ou HL (S-kNN + ϵN). No conjunto de dados Yeast, o melhor resultado foi alcançado sem usar a técnica de alto nível, apenas usando o CC(SVM).

Tabela 14 – Comparação entre as técnicas de ML de baixo nível e nosso modelo de classificação de alto nível em termos de **acurácia**. "LL. Alg." representa os algoritmos de baixo nível, "LL. Base" os classificadores básicos, "HL. Graph" a construção do grafo do classificador de alto nível. Os melhores resultados locais estão sublinhados e os melhores resultados globais estão em negrito.

LL	LL	HL	Datasets						
Alg.	Base	Graph	Birds	Emotions	Enron	Genbase	Medical	Scene	Yeast
BR	NB	-	95.2	71.3	75.9	99.5	94.7	75.4	70.0
BR	NB	kNN	95.2	77.0	75.9	99.5	94.7	88.2	74.6
BR	NB	kNN+ϵN	95.1	75.7	76.0	99.5	94.7	88.5	74.6
BR	NB	S-kNN	95.1	76.6	75.9	99.5	94.7	87.9	74.3
BR	NB	S-kNN+ϵN	95.1	77.0	75.9	99.5	94.7	87.9	74.3
BR	NB	D-kNN	95.2	76.2	76.0	99.5	94.7	88.2	74.8
CC	NB	-	95.2	72.8	82.6	99.6	95.0	79.0	68.4
CC	NB	kNN	95.7	77.4	82.6	99.6	95.0	87.7	73.5
CC	NB	kNN+ϵN	95.1	76.4	82.6	99.6	95.0	87.8	73.5
CC	NB	S-kNN	95.1	77.1	82.6	99.6	95.0	87.6	73.5
CC	NB	S-kNN+ϵN	95.1	77.3	82.6	99.6	95.0	87.7	73.7
CC	NB	D-kNN	95.2	77.3	82.6	99.6	95.0	87.7	73.4
BR	RF	-	95.7	77.9	93.6	99.9	96.7	90.9	80.7
BR	RF	kNN	95.8	77.4	93.6	**100.0**	97.2	91.7	80.9
BR	RF	kNN+ϵN	96.0	78.5	93.4	**100.0**	97.2	91.5	80.9
BR	RF	S-kNN	96.1	78.2	93.6	**100.0**	97.2	91.6	80.9
BR	RF	S-kNN+ϵN	96.0	78.9	93.6	**100.0**	97.2	91.5	80.7
BR	RF	D-kNN	**96.2**	78.6	93.4	**100.0**	97.2	91.4	81.0
CC	RF	-	95.6	78.0	93.5	**100.0**	96.7	91.1	80.6
CC	RF	kNN	95.8	79.0	93.6	**100.0**	97.2	91.8	80.7
CC	RF	kNN +ϵN	95.8	78.4	93.6	**100.0**	97.2	91.6	81.0
CC	RF	S-kNN	96.1	78.5	93.6	**100.0**	97.2	91.7	80.8
CC	RF	S-kNN+ϵN	96.1	78.5	93.6	**100.0**	97.2	91.6	80.6
CC	RF	D-kNN	96.1	78.1	93.6	**100.0**	97.2	91.2	80.9
BR	SVM	-	95.7	79.5	93.7	**100.0**	97.8	91.9	80.8
BR	SVM	kNN	95.6	80.1	**93.8**	99.9	97.7	91.9	81.3
BR	SVM	kNN+ϵN	95.5	80.4	**93.8**	99.9	97.7	**92.0**	81.1
BR	SVM	S-kNN	95.6	80.0	**93.8**	99.9	97.7	91.9	81.1
BR	SVM	S-kNN+ϵN	95.6	80.0	**93.8**	99.9	97.7	**92.0**	81.1
BR	SVM	D-kNN	95.6	80.0	**93.8**	99.9	97.7	91.9	81.3
CC	SVM	-	95.7	80.4	93.7	**100.0**	**97.9**	91.3	**81.4**
CC	SVM	kNN	95.6	**80.6**	93.7	**100.0**	97.7	91.8	80.8
CC	SVM	kNN+ϵN	95.5	**80.6**	93.7	**100.0**	97.7	91.8	80.8
CC	SVM	S-kNN	95.6	80.4	93.7	**100.0**	97.7	91.8	81.1
CC	SVM	S-kNN+ϵN	95.6	80.4	93.7	**100.0**	97.7	91.8	81.1
CC	SVM	D-kNN	95.6	79.9	93.7	**100.0**	97.7	91.8	81.1
MLkNN	-	-	94.9	78.3	92.6	93.8	96.3	90.4	79.5
MLkNN	-	kNN	95.0	78.0	93.0	**100.0**	96.2	90.7	79.0
MLkNN	-	kNN+ϵN	95.1	78.1	93.1	**100.0**	96.2	91.0	79.1
MLkNN	-	S-kNN	94.9	77.9	93.1	**100.0**	96.3	90.7	78.8
MLkNN	-	S-kNN+ϵN	94.9	77.9	93.1	**100.0**	96.3	90.7	78.9
MLkNN	-	D-kNN	95.0	78.0	93.1	**100.0**	96.2	90.7	78.7

A fim de analisar estatisticamente a eficácia de nossa técnica, selecionamos o teste de Friedman, pois permite comparar várias técnicas frente a várias bases de dados (DEMŠAR, 2006). Considere os resultados de acurácia apresentados na Tabela 14, queremos saber se há alguma evidência de que o desempenho preditivo dos classificadores de baixo nível é diferente quando combinado ou não com os de alto nível. Assim, a hipótese nula diz que eles são estatisticamente equivalentes. Após o cálculo do teste de Friedman sob o nível de significância α de 0.05, a hipótese nula é rejeitada, ou seja, pelo menos um dos métodos difere dos demais. O teste posthoc de Nemenyi é então aplicado considerando novamente o nível de significância α em 0.05. O teste indica que os resultados de precisão obtidos pelas técnicas de baixo nível em combinação com os de alto nível fornecidos pelo grafo kNN supera os resultados de acurácia obtidos exclusivamente pelas técnicas de baixo nível. O diagrama de diferença crítica encontrado pelo teste post-hoc de Nemenyi é mostrado pela Fig. 35. Para interpretar tal diagrama, o primeiro passo é observar o *ranking* médio dos modelos, onde quanto menor o valor (posição) melhor foi o desempenho da técnica. Em seguida, deve ser analisada a diferença crítica (CD), que representa a diferença que deve ser atingida em termos de ranking médio para que dois pares de modelos sejam considerados significativamente diferentes. Por fim, os modelos que são significativamente diferentes (i.e., atingem um valor maior que a diferença crítica) não compartilham do mesmo traço horizontal, como acontece, por exemplo, entre o modelo de alto nível usando kNN e o classificador de baixo nível (LL) no diagrama abaixo.

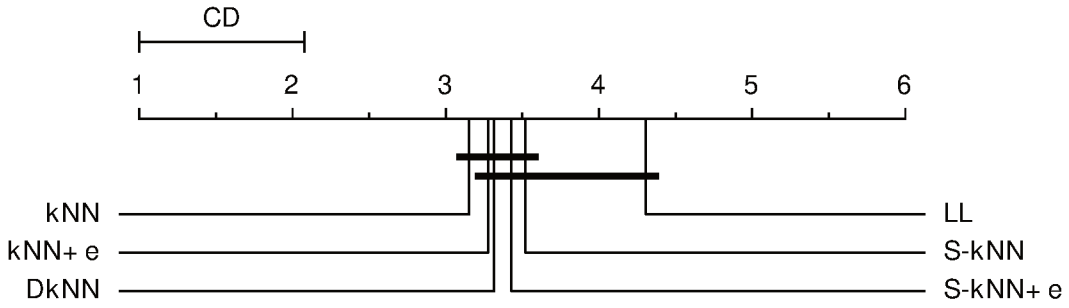


Figura 35 – Diagrama de diferença crítica obtidos pelo teste Nemenyi post-hoc nos resultados de acurácia apresentados na Tabela 14.

5.3.2 Análise Preditiva em Termos de Acurácia de Subconjunto

Nesta seção será analisado o desempenho preditivo das técnicas em termos da métrica de acurácia do subconjunto. A Tabela 15 apresenta os resultados obtidos para as classificações de baixo e alto nível. Pode-se observar que os resultados aqui são muito menores do que os apresentados na Tabela 14, pois esta métrica considera considere em seu cálculo apenas objetos que foram perfeitamente classificados (i.e., o classificador acertou todo o conjunto de rótulos do objeto). Para o conjunto de dados Birds, o melhor resultado foi

fornecido pelo CC(RF) combinado com a classificação HL usando o grafo S-kNN. Para o conjunto de dados Emotions, novamente CC (SVM) obteve os melhores resultados após a combinação com a classificação HL usando os grafos kNN e kNN + ϵN. Na base de dados Enron, o melhor resultado foi obtido através da técnica de alto nível combinada ao CC(SVM) usando qualquer método de construção da rede. Para a base Genbase, a técnica de alto nível aumentou drasticamente os resultados do MLkNN, que obteve os melhores resultados usando os grafos kNN e kNN+ϵN. Na base Medical, o melhor resultado foi obtido pelo CC(SVM), sem usar a técnica de alto nível. Para o conjunto de dados Scene, CC (SVM) forneceu os melhores resultados após ser combinado com HL (kNN + ϵN). No conjunto de dados Yeast, o melhor resultado foi fornecido exclusivamente pela técnica de baixo nível CC(SVM).

Para analisar estatisticamente os resultados apresentados na Tabela 16, adotamos novamente o teste de Friedman. A hipótese nula afirma que os resultados de acurácia de subconjunto obtidos exclusivamente pelas técnicas de baixo nível são equivalentes aos obtidos pela combinação dessas técnicas com o termo de alto nível. Sob o nível de significância α em 0.05, a hipótese nula é rejeitada. O teste posthoc de Nemenyi é então aplicado. O diagrama de diferença crítica é mostrado pela Fig. 36 e indica que os resultados de acurácia de subconjunto obtidos pela técnicas de alto nível usando o grafo $S-kNN$ superam os resultados de acuracia de subconjunto obtidos exclusivamente pelas técnicas de baixo nível.

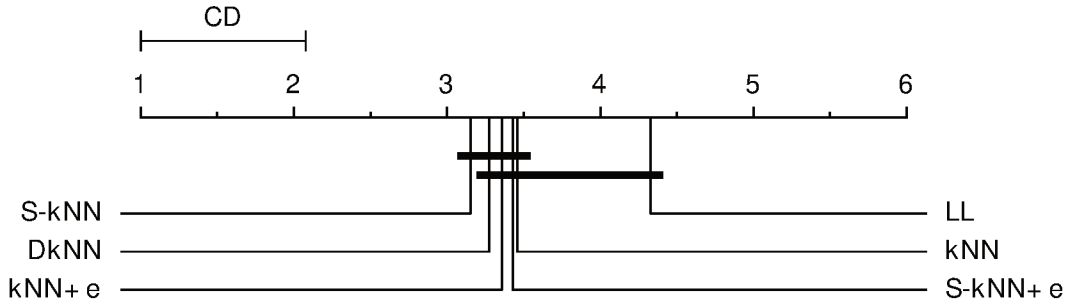


Figura 36 – Diagrama de diferença crítica obtidos pelo teste Nemenyi post-hoc nos resultados de acurácia de subconjunto apresentados na Tabela 15.

5.3.3 Análise Preditiva em Termos de F_1-weighted

A Tabela 16 apresenta os resultados preditivos obtidos pelas técnicas em comparação em termos da medida F_1 weighted. Os resultados aqui não seguem um padrão em relação aos resultados de acurácia ou acurácia de subconjunto. Por exemplo, os resultados de F_1 no conjunto de dados Birds são piores do que aqueles de acurácia de subconjunto, embora os resultados de F_1 nas bases de dados Emotions e Yeast sejam melhores do que aqueles de acurácia de subconjunto. Isso pode ser explicado observando a densidade de

Tabela 15 – Comparação entre as técnicas de ML de baixo nível e nosso modelo de classificação de alto nível em termos de **acurácia de subconjunto**. "LL. Alg." representa os algoritmos de baixo nível, "LL. Base" os classificadores básicos, "HL. Graph " a construção do grafo do classificador de alto nível. Os melhores resultados locais estão sublinhados e os melhores resultados globais estão em negrito.

LL	LL	HL	Datasets						
Alg.	Base	Graph	Birds	Emotions	Enron	Genbase	Medical	Scene	Yeast
BR	NB	-	47.4	18.8	0.0	92.5	19.8	17.1	10.5
BR	NB	kNN	47.7	21.8	0.0	92.5	19.8	50.7	11.2
BR	NB	kNN+ϵN	46.7	18.8	0.0	92.5	19.8	51.8	11.7
BR	NB	S-kNN	47.1	22.8	0.0	92.5	19.8	48.8	11.9
BR	NB	S-kNN+ϵN	47.1	23.3	0.0	92.5	19.8	48.8	11.9
BR	NB	D-kNN	47.7	21.3	0.0	92.5	19.8	50.7	11.2
CC	NB	-	47.4	19.8	0.2	93.0	21.9	28.3	9.2
CC	NB	kNN	47.7	21.8	0.2	93.0	21.9	44.0	11.2
CC	NB	kNN+ϵN	46.7	18.8	0.2	93.0	21.9	43.1	11.0
CC	NB	S-kNN	47.1	22.8	0.2	93.0	21.9	44.6	10.5
CC	NB	S-kNN+ϵN	47.1	23.3	0.2	93.0	21.9	45.1	10.5
CC	NB	D-kNN	47.7	22.8	0.2	93.0	21.9	44.0	11.3
BR	RF	-	50.2	24.8	13.6	99.0	45.0	53.1	16.6
BR	RF	kNN	48.9	24.3	13.0	99.5	56.0	60.5	18.9
BR	RF	kNN+ϵN	52.9	27.7	13.6	99.5	56.0	61.0	18.9
BR	RF	S-kNN	51.7	25.2	13.0	99.5	55.8	60.2	20.1
BR	RF	S-kNN+ϵN	51.4	29.7	13.0	99.5	55.8	60.2	18.9
BR	RF	D-kNN	51.7	28.2	13.6	99.5	55.8	59.3	19.5
CC	RF	-	49.2	27.7	13.1	99.5	46.2	55.1	21.6
CC	RF	kNN	48.3	31.2	13.3	99.5	56.9	62.4	20.2
CC	RF	kNN +ϵN	49.5	28.7	13.3	99.5	56.9	61.5	21.7
CC	RF	S-kNN	**53.3**	28.7	13.3	99.5	57.2	61.7	20.3
CC	RF	S-kNN+ϵN	52.0	26.7	13.3	99.5	57.2	60.5	19.7
CC	RF	D-kNN	51.7	29.7	13.3	99.5	55.8	60.6	20.9
BR	SVM	-	50.8	28.2	14.5	99.5	65.1	62.6	20.0
BR	SVM	kNN	49.5	33.7	14.5	98.5	63.3	63.5	21.9
BR	SVM	kNN+ϵN	51.1	33.2	14.9	98.5	63.3	63.0	20.8
BR	SVM	S-kNN	50.2	33.7	14.9	98.5	64.0	63.4	21.0
BR	SVM	S-kNN+ϵN	50.2	33.7	14.9	98.5	64.0	63.4	21.0
BR	SVM	D-kNN	49.8	33.2	14.9	98.5	64.0	63.5	21.7
CC	SVM	-	50.8	30.7	15.0	99.5	**67.1**	62.9	**23.0**
CC	SVM	kNN	49.5	**34.7**	**15.7**	99.5	63.9	63.7	21.4
CC	SVM	kNN+ϵN	51.1	**34.7**	**15.7**	99.5	63.9	**64.0**	21.4
CC	SVM	S-kNN	50.2	34.2	**15.7**	99.5	64.0	63.7	22.0
CC	SVM	S-kNN+ϵN	50.2	34.2	**15.7**	99.5	63.9	63.9	22.0
CC	SVM	D-kNN	49.8	32.2	**15.7**	99.5	63.9	63.7	21.9
MLkNN	-	-	47.4	24.3	5.9	15.1	41.1	62.0	19.2
MLkNN	-	kNN	46.7	25.7	10.0	**100.0**	44.2	61.5	16.7
MLkNN	-	kNN+ϵN	47.7	26.2	9.8	**100.0**	44.2	60.2	16.6
MLkNN	-	S-kNN	44.9	25.2	10.0	99.5	45.1	61.9	17.0
MLkNN	-	S-kNN+ϵN	44.9	25.2	9.8	99.5	45.1	61.9	16.4
MLkNN	-	D-kNN	46.4	26.2	10.0	99.5	44.2	61.5	16.9

tais conjuntos de dados: a base Birds têm densidade muito baixa, o que significa que há muito poucos rótulos por instância em relação a quantidade de rótulos possíveis. Por outro lado, Emotions e Scene têm densidade mais alta, o que melhora F_1, mas torna muito mais difícil obter uma alta acurácia de subconjunto.

Em relação ao desempenho geral na Tabela 16, o melhor resultado para o conjunto de dados Birds foi obtido por CC (RF) combinado com a classificação HL fornecida a partir do grafo D-kNN. Para o conjunto de dados Emotions, BR (SVM) obteve o melhor desempenho quando combinado com o HL (kNN + ϵN). Para a base Enron, o melhor resultado foi obtido através do BR(SVM) usando o grafo DkNN. Já na base Genbase, os melhores resultados foram obtidos usando o BR(RF) sem o termo de alto nível e o MLkNN, usando os grafos kNN e kNN+ϵN. Para o conjunto de dados Scene, CC (RF) forneceu os melhores resultados quando combinado com HL (kNN). No conjunto de dados de Yeast, o melhor resultado foi obtido pela combinação da classificação de alto nível fornecida por HL (kNN) com a classificação de baixo nível fornecida por BR (RF).

Para analisar estatisticamente os resultados apresentados na Tabela 16, adotamos novamente o teste de Friedman. A hipótese nula afirma que os resultados de F_1 obtidos exclusivamente pelas técnicas de baixo nível são equivalentes aos obtidos pela combinação dessas técnicas com nossas técnicas de alto nível. Sob o nível de significância α em 0.05, a hipótese nula é rejeitada. O teste posthoc de Nemenyi é então aplicado. O diagrama de diferença crítica é mostrado pela Fig. 37 e indica que os resultados de F_1 obtidos pelas técnicas de alto nível superam os resultados de F_1 obtidos exclusivamente pelas técnicas de baixo nível, independentemente de o método de construção de grafos adotado pela técnica de alto nível.

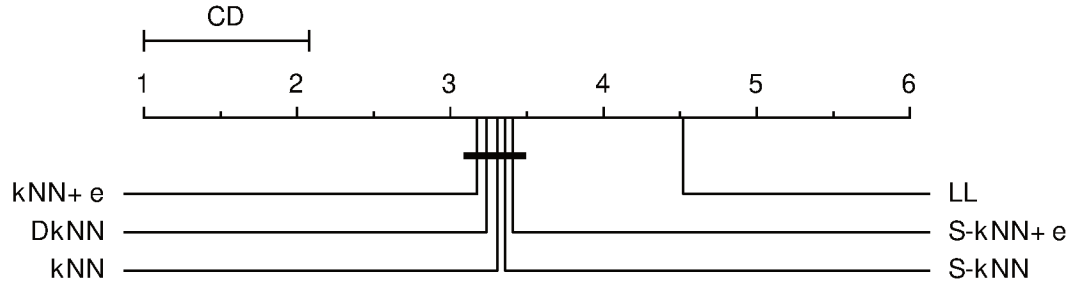


Figura 37 – Diagrama de diferença crítica relacionado aos resultados da medida F_1-weighted apresentados na Tabela 16.

5.3.4 Análise sobre Desbalanceamento de Rótulos

O desbalanceamento de rótulos é um problema natural do AMR, onde as classes não são igualmente representadas, no qual o número de instâncias positivas de cada classe é normalmente muito menor do que o número de instâncias negativas (CHARTE et al., 2013;

Tabela 16 – Comparação entre as técnicas de ML de baixo nível e nosso modelo de classificação de alto nível em termos de **F_1-weighted**. "LL. Alg." representa os algoritmos de baixo nível, "LL. Base" os classificadores básicos, "HL. Graph " a construção do grafo do classificador de alto nível. Os melhores resultados locais estão sublinhados e os melhores resultados globais estão em negrito.

LL	LL	HL	Datasets						
Alg.	Base	Graph	Birds	Emotions	Enron	Genbase	Medical	Scene	Yeast
BR	NB	-	13.0	62.1	27.9	94.8	47.2	56.2	57.8
BR	NB	kNN	27.9	63.6	27.9	94.8	47.2	66.8	56.5
BR	NB	kNN+ϵN	24.6	60.8	27.9	94.8	47.2	66.7	56.5
BR	NB	S-kNN	21.7	63.4	27.9	94.8	47.2	66.6	55.4
BR	NB	S-kNN+ϵN	21.7	64.2	27.9	94.8	47.2	66.6	55.4
BR	NB	D-kNN	27.9	61.9	27.8	94.8	47.2	66.8	56.9
CC	NB	-	13.0	62.5	34.8	95.0	49.5	51.7	56.8
CC	NB	kNN	21.7	64.1	34.8	95.0	49.5	61.3	56.1
CC	NB	kNN+ϵN	24.6	60.0	34.8	95.0	49.5	60.1	54.4
CC	NB	S-kNN	21.7	63.8	34.8	95.0	49.5	61.2	56.1
CC	NB	S-kNN+ϵN	21.7	64.1	34.8	95.0	49.5	61.3	56.1
CC	NB	D-kNN	27.9	63.2	34.7	95.0	49.5	60.8	54.7
BR	RF	-	25.5	59.0	66.1	**100.0**	75.9	67.4	55.4
BR	RF	kNN	49.7	60.6	65.6	99.8	79.9	75.3	73.7
BR	RF	kNN+ϵN	55.2	66.4	66.2	99.8	79.9	74.9	73.4
BR	RF	S-kNN	53.4	65.1	65.5	99.8	79.8	74.8	73.5
BR	RF	S-kNN+ϵN	52.9	66.9	65.5	99.8	79.8	74.0	73.1
BR	RF	D-kNN	55.9	65.7	66.1	99.8	79.8	74.1	**73.8**
CC	RF	-	24.4	59.6	65.7	99.8	77.2	68.3	56.7
CC	RF	kNN	49.7	65.6	65.9	99.8	80.1	**76.3**	73.0
CC	RF	kNN +ϵN	52.5	65.8	65.9	99.8	80.1	75.3	73.6
CC	RF	S-kNN	53.5	64.2	66.0	99.8	80.2	75.6	73.1
CC	RF	S-kNN+ϵN	53.6	64.1	66.1	99.8	80.2	73.8	72.8
CC	RF	D-kNN	**56.1**	65.9	66.2	99.8	80.0	74.4	73.2
BR	SVM	-	35.9	65.4	67.3	99.8	83.6	74.7	60.2
BR	SVM	kNN	45.2	68.3	69.7	99.4	81.8	75.9	73.2
BR	SVM	kNN+ϵN	47.7	**69.5**	69.9	99.4	81.8	75.9	72.8
BR	SVM	S-kNN	47.3	67.6	69.9	99.4	81.9	75.9	72.8
BR	SVM	S-kNN+ϵN	47.3	67.3	69.9	99.4	81.9	76.0	72.8
BR	SVM	D-kNN	46.3	68.1	**70.0**	99.4	81.9	75.9	73.2
CC	SVM	-	35.9	65.6	67.8	99.8	**83.9**	74.0	59.2
CC	SVM	kNN	45.0	69.3	69.5	99.8	81.5	76.0	70.5
CC	SVM	kNN+ϵN	47.7	69.3	69.6	99.8	81.5	76.0	70.5
CC	SVM	S-kNN	47.2	68.7	69.5	99.8	81.6	75.9	71.3
CC	SVM	S-kNN+ϵN	47.2	68.7	69.5	99.8	81.5	75.9	71.3
CC	SVM	D-kNN	46.0	68.2	69.6	99.8	81.5	76.0	71.3
MLkNN	-	-	12.6	60.4	64.4	39.5	71.1	72.7	58.2
MLkNN	-	kNN	20.5	64.2	63.7	**100.0**	68.2	72.1	66.5
MLkNN	-	kNN+ϵN	27.1	64.4	63.7	**100.0**	68.2	72.2	66.3
MLkNN	-	S-kNN	31.6	63.3	63.8	99.8	68.8	72.5	66.0
MLkNN	-	S-kNN+ϵN	31.6	63.4	63.8	99.8	68.8	72.5	65.9
MLkNN	-	D-kNN	27.6	64.1	63.7	99.8	68.2	72.1	66.0

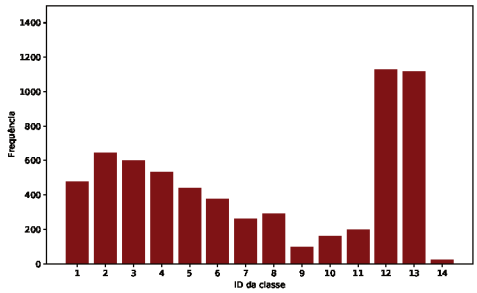
ZHANG et al., 2020). Da mesma forma, existem problemas com classes predominantes que aparecem com muito mais frequência que outras, afetando diretamente na capacidade de generalização dos algoritmos. Para explicar alguns resultados, a Fig. 38 mostra a distribuição dos rótulos do conjunto de treinamento de cada base de dados.

Se olharmos para os valores de acurácia e acurácia de subconjunto, é possível ver por exemplo que na base Enron, o MLkNN conseguiu 92.6% de acurácia mas apenas 5.9% de acurácia de subconjunto, o que pode parecer contraditório, porém, ao observar a distribuição dos rótulos de tal base, percebemos que apenas uma pequena quantidade de rótulos aparece com frequência, logo, se o modelo atribuir poucas instâncias como positivas e muitas como negativas ele vai obter um bom valor de acurácia, já que a grande maioria do conjunto de rótulos será constituída de instâncias negativas. Entretanto, para que ele consiga um bom valor de acurácia de subconjunto, teria que acertar todo conjunto de rótulos, o que é muito difícil em um cenario que possui 53 classes possíveis, e que ainda por cima estão mal distribuídas, possivelmente afetando a capacidade de generalização do algoritmo. Em contrapartida, se olharmos para a base de dados Emotions, vemos que o CC(SVM) mesmo conseguindo 80% de acurácia atingiu 34.7% de acurácia de subconjunto; o que acontece é que na base Emotions, os rótulos estão bem distribuídos, tornando mais difícil obter um bom valor de acurácia, já que o algoritmo teria que realmente identificar e separar o padrão das classes ao invés de apenas classificar a maioria dos rótulos como negativos. Por outro lado, com os rótulos bem distribuídos a capacidade de generalização do modelo é melhor, fazendo com que seja mais fácil estimar o conjunto de rótulos inteiro de uma amostra. É possível observar também, que as maiores contribuições da técnica de alto nível foram nas bases Emotions, Scene e Yeast, cujo os rótulos estão bem distribuídos.

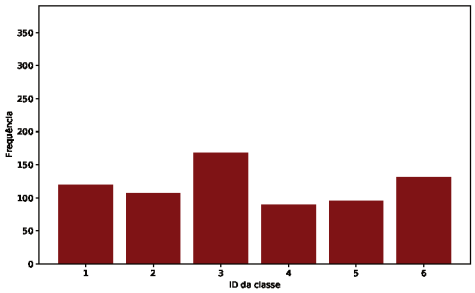
Em razão do desbalanceamento de rótulos, é importante ressaltar que para avaliar um modelo de AMR devemos sempre usar alguma medida de avaliação baseada em exemplo (e.g., acurácia de subconjunto) que irá avaliar o algoritmo na sua habilidade de predizer todo o conjunto de rótulos da amostra ao invés de contabilizar acertos individuais em cada rótulo (como é feito em medidas baseadas em rótulo, como a acurácia).

5.4 Considerações Finais

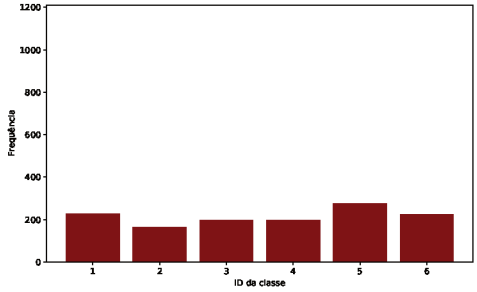
Nesta seção, avaliamos a contribuição de nossa técnica de alto nível em relação às tradicionais de AMR. Os experimentos foram realizados em conjuntos de dados artificiais e reais em termos de três métricas de avaliação multirrótulo: acurácia, acurácia de subconjunto e F_1-weighted. Para cada métrica, um teste estatístico foi aplicado sobre os resultados para apoiar nossa discussão. Tais testes revelaram, com um nível de confiança de 95%, que a classificação fornecida pela combinação entre as técnicas de alto nível e tradicionais supera a fornecida exclusivamente pelas técnicas tradicionais em qualquer uma das medidas. Este é um resultado empolgante, principalmente se considerarmos que



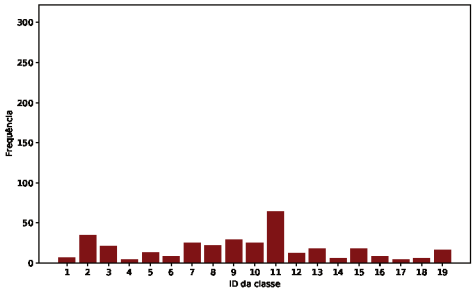
(a) Yeast



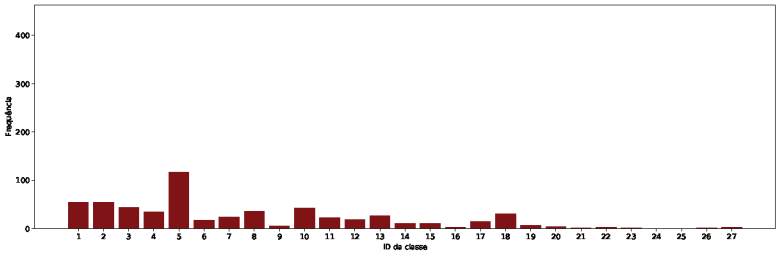
(b) Emotions



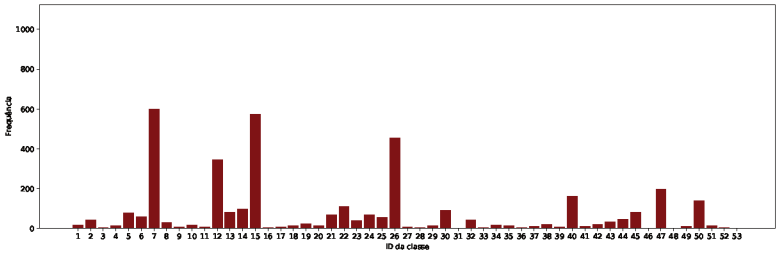
(c) Scene



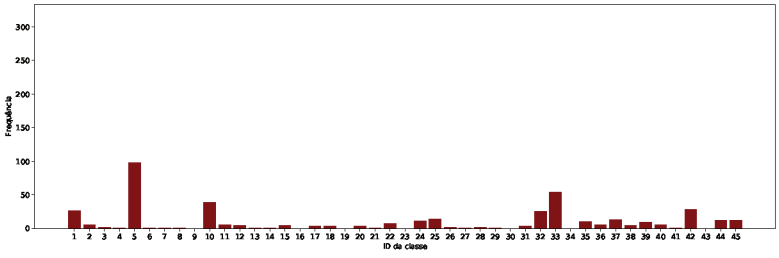
(d) Birds



(e) Genbase



(f) Enron



(g) Medical

Figura 38 – Número de instâncias por rótulo na base de treino.

as técnicas tradicionais tiveram seus parâmetros rigorosamente ajustados, o que torna uma pequena melhoria muito difícil de alcançar. Tais resultados podem ser explicados pela diferença entre nossa técnica e as tradicionais, onde a nossa consegue considerar não apenas os atributos físicos, mas também a estrutura topológica dos dados.

Outro ponto a ser considerado é que o desempenho preditivo do CXN-MLL poderia ser melhorado ainda mais, visto que os pesos das medidas de rede não foram ajustados assim como as técnicas de baixo nível foram equipadas com seus hiperparâmetros padrões.

Capítulo 6

Conclusão

Este trabalho investigou um novo método de classificação híbrido para aprendizado multirrótulo, o qual considera informações de alto nível derivadas de medidas de redes complexas e de baixo nível fornecidas por algoritmos tradicionais de classificação multirrótulo. A técnica desenvolvida foi denominada CXN-MLL e é capaz de combinar eficientemente tais informações de modo a melhorar (com significância estatística) o desempenho preditivo dos algoritmos de baixo nível. Dessa forma, a hipótese levantada nessa pesquisa, a qual afirmava que "a análise de características estruturais e topológicas de dados multirrótulo contribui para melhorar o desempenho preditivo dos modelos de classificação essencialmente baseados nas características físicas dos dados"., foi confirmada. Em síntese, esse é um resultado bastante atrativo que abre caminho para que novas abordagens baseadas em redes complexas sejam desenvolvidas para o aprendizado multirrótulo.

Em relação aos objetivos específicos, destacam-se as seguintes contribuições:

- **Investigar e desenvolver métodos eficientes de construção do grafo para o contexto multirrótulo**

  Ao todo foram propostos 5 métodos de construção da rede para aprendizado multirrótulo, a saber: rede k-vizinhos mais próximo (kNN), rede k-vizinhos mais próximos com vizinhança de raio ϵ (kNN+ϵN), rede k-vizinhos mais próximos seletiva (S-kNN), rede k-vizinhos mais próximos seletivo com vizinhança de raio ϵ (S-kNN+ϵN), e rede vizinhos mais próximos de grau k (D-kNN). Com exceção da rede D-kNN que usa uma heurística baseada no grau para definir o número de conexões, os outros quatro métodos foram adaptados da literatura de classificação monorrótulo para o contexto multirrótulo.

  Os experimentos mostraram que não há um método efetivamente melhor que todos os outros, mas que cada método em particular pode ser mais eficiente dependendo do problema e medida de avaliação utilizadas. Por exemplo, para a medida de acurácia, a rede kNN obteve o melhor desempenho preditivo, para a acurácia de subconjunto foi a rede S-kNN e para a medida F_1 a rede kNN+ϵN.

- **Desenvolver uma técnica capaz de combinar as associações de baixo nível produzidas por algoritmos de classificação multirrótulo tradicionais com associações de alto nível obtidas a partir de medidas de redes complexas**

  O CXN-MLL foi a principal contribuição do trabalho, uma nova técnica de classificação multirrótulo híbrida de alto nível capaz de considerar a estrutura e topologia da rede, sendo capaz de identificar múltiplos padrões nos dados os quais as técnicas tradicionais de baixo nível avaliadas tiveram dificuldades. Para prover associações de alto nível, a técnica desenvolvida adaptou o conceito de conformidade de padrão, anteriormente proposto para a classificação monorrótulo. Nesse sentido, medidas de redes complexas são calculadas antes e após a inserção de cada item de teste nas redes multirrótulo, sendo que aquelas redes que apresentam baixa variação no resultado das medidas são mais propícias a receberem o rótulo correspondente, enquanto aquelas com variação elevada aparentemente não estão em conformidade com o padrão daquela rede (rótulo). Os experimentos conduzidos mostraram que a técnica é promissora, pois apresentou resultados satisfatórios e que trazem motivação para trabalhos futuros.

- **Demonstrar o potencial que medidas e propriedades de redes complexas possuem para aprimorar o desempenho preditivo de algoritmos convencionais de classificação multirrótulo**

  Neste trabalho foram conduzidos experimentos que avaliam as medidas de rede e seus impactos no algoritmo. Além disso, bases de dados artificiais foram usadas para demonstrar a dificuldade dos algoritmos tradicionais em identificar determinados padrões nos dados, os quais a técnica de alto nível proposta consegue detectar com facilidade ao capturar informações estruturais dos dados. Além das bases artificiais, a técnica foi avaliada em bases de dados reais onde os resultados foram posteriormente validados por teste estatístico que demonstrou a capacidade da técnica de alto nível em aprimorar a habilidade preditiva dos algoritmos tradicionais de AMR, confirmando a hipótese apresentada neste trabalho de dissertação.

6.1 Limitações

Em relação às limitações do trabalho, um dos principais problemas está relacionado ao número de hiperparâmetros da técnica. Como a técnica possui vários hiperparâmetros e que podem assumir diversos valores, o processo de otimização e obtenção dos melhores hiperparâmetros do modelo podem tomar algum tempo. Além disso, o desempenho preditivo e a complexidade computacional da técnica é dependente do classificador de baixo nível escolhido. Outra limitação está relacionada ao problema de dependência de rótulos, pois a técnica considera cada classe como um grafo distinto, não levando em consideração

a correlação entre os rótulos, o que definitivamente seria identificado se todas as classes fossem representadas de forma unificada em uma só rede.

6.2 Trabalhos Futuros

Para trabalhos futuros serão incluídos novas investigações nos métodos de construção da rede, como por exemplo, o uso de um multigrafo único que represente todos os dados multirrótulo sem a necessidade de criação de um grafo para cada classe. Além do mais será verificado o uso de grafos direcionados e o uso de redes heterogêneas na representação da rede. Também serão investigadas novas estratégias para exploração da topologia da rede, como por exemplo, passeio aleatório e caminhada do turista.

Ainda no contexto multirrótulo, serão investigadas técnicas que consideram a correlação entre rótulos assim como estratégias para tratar o desbalanceamento de classes, dois grandes desafios do aprendizado multirrótulo.

Em relação ao uso de redes complexas no AMR, serão investigadas novas estratégias de alto nível para classificação de dados em geral, como por exemplo, o uso de componentes fortemente conexas para explorar relações estruturais da rede e também a correlação entre rótulos.

6.3 Contribuições em Produção Bibliográfica

Este trabalho resultou nas seguintes contribuições em termos de produção bibliográfica:

- RESENDE, V. H.; CARNEIRO, M. G. Towards a high-level multi-label classification from complex networks. In: **IEEE. 2019 IEEE 31st International Conference on Tools with Artificial Intelligence**. [S.l.], 2019. p. 1140–1147.
- RESENDE, V. H.; CARNEIRO, M. G. High-level classification for multi-label learning. In: **IEEE. 2020 International Joint Conference on Neural Networks**. [S.l.], 2020. p. 1–8.
- RESENDE, V. H.; CARNEIRO, M. G. Analysis of Complex Network Measures for Multi-Label Classification (aceito). In: **International Journal on Artificial Intelligence Tools.**

Referências

ALBERT, R. Scale-free networks in cell biology. **Journal of cell science**, The Company of Biologists Ltd, v. 118, n. 21, p. 4947–4957, 2005. Disponível em: <https://doi.org/10.1242/jcs.02714>.

ALBERT, R.; BARABÁSI, A.-L. Statistical mechanics of complex networks. **Reviews of modern physics**, APS, v. 74, n. 1, p. 47, 2002. Disponível em: <https://doi.org/10.1103/RevModPhys.74.47>.

BACKES, A. R.; CASANOVA, D.; BRUNO, O. M. A complex network-based approach for boundary shape analysis. **Pattern Recognition**, Elsevier, v. 42, n. 1, p. 54–67, 2009. Disponível em: <https://doi.org/10.1016/j.patcog.2008.07.006>.

BENGIO, Y.; COURVILLE, A.; VINCENT, P. Representation learning: A review and new perspectives. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 35, n. 8, p. 1798–1828, 2013. Disponível em: <https://doi.org/10.1109/TPAMI.2013.50>.

BISHOP, C. M. **Pattern recognition and machine learning**. [S.l.]: springer, 2006.

BOUTELL, M. R. et al. Learning multi-label scene classification. **Pattern recognition**, Elsevier, v. 37, n. 9, p. 1757–1771, 2004. Disponível em: <https://doi.org/10.1016/j.patcog.2004.03.009>.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001. Disponível em: <https://doi.org/10.1023/A:1010933404324>.

BRIGGS, F. et al. Acoustic classification of multiple simultaneous bird species: A multi-instance multi-label approach. **The Journal of the Acoustical Society of America**, ASA, v. 131, n. 6, p. 4640–4650, 2012. Disponível em: <https://doi.org/10.1121/1.4707424>.

BRITO, M. R. et al. Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. **Statistics & Probability Letters**, Elsevier, v. 35, n. 1, p. 33–42, 1997. Disponível em: <https://doi.org/10.1016/S0167-7152(96)00213-1>.

CARNEIRO, M.; ZHAO, L. Analysis of graph construction methods in supervised data classification. In: IEEE. **2018 7th Brazilian Conference on Intelligent Systems (BRACIS)**. 2018. p. 390–395. Disponível em: <https://doi.org/10.1109/BRACIS.2018.00074>.

CARNEIRO, M. G. **Redes complexas para classificação de dados via conformidade de padrão, caracterização de importância e otimização estrutural**. Tese (Doutorado) — Universidade de São Paulo, 2017.

CARNEIRO, M. G. et al. Particle swarm optimization for network-based data classification. **Neural Networks**, Elsevier, v. 110, p. 243–255, 2019.

_____. Nature-inspired graph optimization for dimensionality reduction. In: **IEEE International Conference on Tools with Artificial Intelligence (ICTAI)**. [s.n.], 2017. p. 1113–1119. Disponível em: <https://doi.org/10.1109/ICTAI.2017.00170>.

_____. Network-based data classification: combining k-associated optimal graphs and high-level prediction. **Journal of the Brazilian Computer Society**, v. 20, n. 1, p. 14, 2014. Disponível em: <https://doi.org/10.1186/1678-4804-20-14>.

CARNEIRO, M. G.; ZHAO, L. Organizational data classification based on the importance concept of complex networks. **IEEE Transactions on Neural Networks and Learning Systems**, v. 29, n. 8, p. 3361–3373, 2018. Disponível em: <https://doi.org/10.1109/TNNLS.2017.2726082>.

CARNEIRO, M. G. et al. Network structural optimization based on swarm intelligence for highlevel classification. In: IEEE. **IEEE International Joint Conference on Neural Networks**. 2016. p. 3737–3744. Disponível em: <https://doi.org/10.1109/IJCNN.2016.7727681>.

CHARTE, F. et al. A first approach to deal with imbalance in multi-label datasets. In: SPRINGER. **International Conference on Hybrid Artificial Intelligence Systems**. 2013. p. 150–160. Disponível em: <https://doi.org/10.1007/978-3-642-40846-5_16>.

CHEN, W.-J. et al. Mltsvm: a novel twin support vector machine to multi-label learning. **Pattern Recognition**, Elsevier, v. 52, p. 61–74, 2016. Disponível em: <https://doi.org/10.1016/j.patcog.2015.10.008>.

CHERMAN, E. A. **Aprendizado de máquina multirrótulo: explorando a dependência de rótulos e o aprendizado ativo**. Tese (Doutorado) — Universidade de São Paulo, 2013.

CUPERTINO, T. H. et al. A scheme for high level data classification using random walk and network measures. **Expert Systems with Applications**, Springer, v. 92, p. 289–303, 2018. Disponível em: <https://doi.org/10.1016/j.eswa.2017.09.014>.

CUPERTINO, T. H.; ZHAO, L.; CARNEIRO, M. G. Network-based supervised data classification by using an heuristic of ease of access. **Neurocomputing**, v. 149, p. 86–92, 2015. Disponível em: <https://doi.org/10.1016/j.neucom.2014.03.071>.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. **Journal of Machine learning research**, v. 7, n. Jan, p. 1–30, 2006.

ELISSEEFF, A.; WESTON, J. A kernel method for multi-labelled classification. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2002. p. 681–687.

FORTUNATO, S. Community detection in graphs. **Physics reports**, Elsevier, v. 486, n. 3-5, p. 75–174, 2010. Disponível em: <https://doi.org/10.1016/j.physrep.2009.11.002>.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The elements of statistical learning: data mining, inference, and prediction**. [S.l.]: Springer Science & Business Media, 2009.

HEARST, M. A. et al. Support vector machines. **IEEE Intelligent Systems and their applications**, IEEE, v. 13, n. 4, p. 18–28, 1998. Disponível em: <https://doi.org/10.1109/5254.708428>.

KATAKIS, I.; TSOUMAKAS, G.; VLAHAVAS, I. Multilabel text classification for automated tag suggestion. In: CITESEER. **Proceedings of the ECML/PKDD**. [S.l.], 2008. v. 18, p. 5.

KLIMT, B.; YANG, Y. The enron corpus: A new dataset for email classification research. In: SPRINGER. **European Conference on Machine Learning**. 2004. p. 217–226. Disponível em: <https://doi.org/10.1007/978-3-540-30115-8_22>.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. Disponível em: <https://doi.org/10.1038/nature14539>.

LEWIS, D. D. Naive (bayes) at forty: The independence assumption in information retrieval. In: SPRINGER. **European conference on machine learning**. 1998. p. 4–15. Disponível em: <https://doi.org/10.1007/BFb0026666>.

LILJEROS, F. et al. The web of human sexual contacts. **Nature**, Nature Publishing Group, v. 411, n. 6840, p. 907–908, 2001. Disponível em: <https://doi.org/10.1038/35082140>.

LINHARES, C. D. et al. Dynetvis: a system for visualization of dynamic networks. In: **Proceedings of the Symposium on Applied Computing**. [s.n.], 2017. p. 187–194. Disponível em: <https://doi.org/10.1145/3019612.3019686>.

LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43–67, 2007. Disponível em: <https://doi.org/10.22456/2175-2745.5690>.

MUSCOLONI, A. et al. Machine learning meets complex networks via coalescent embedding in the hyperbolic space. **Nature communications**, Nature Publishing Group, v. 8, n. 1, p. 1–19, 2017. Disponível em: <https://doi.org/10.1038/s41467-017-01825-5>.

NEWMAN, M. **Networks**. Oxford university press, 2018. Disponível em: <https://doi.org/10.1093/oso/9780198805090.001.0001>.

NEWMAN, M. E. The structure and function of complex networks. **SIAM review**, SIAM, v. 45, n. 2, p. 167–256, 2003. Disponível em: <https://doi.org/10.1137/S003614450342480>.

OPITZ, D.; MACLIN, R. Popular ensemble methods: An empirical study. **Journal of artificial intelligence research**, v. 11, p. 169–198, 1999. Disponível em: <https://doi.org/10.1613/jair.614>.

PESTIAN, J. et al. A shared task involving multi-label classification of clinical free text. In: **Biological, translational, and clinical language processing**. [s.n.], 2007. p. 97–104. Disponível em: <https://doi.org/10.3115/1572392.1572411>.

READ, J. et al. Classifier chains for multi-label classification. **Machine learning**, Springer, v. 85, n. 3, p. 333, 2011.

RESENDE, V. H.; CARNEIRO, M. G. Towards a high-level multi-label classification from complex networks. In: IEEE. **2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)**. 2019. p. 1140–1147. Disponível em: <https://doi.org/10.1109/ICTAI.2019.00159>.

_____. High-level classification for multi-label learning. In: IEEE. **2020 International Joint Conference on Neural Networks (IJCNN)**. 2020. p. 1–8. Disponível em: <https://doi.org/10.1109/IJCNN48605.2020.9207177>.

SANTOS, B. N. dos et al. A two-stage regularization framework for heterogeneous event networks. **Pattern Recognition Letters**, Elsevier, v. 138, p. 490–496, 2020. Disponível em: <https://doi.org/10.1016/j.patrec.2020.08.019>.

SCARSELLI, F. et al. The graph neural network model. **IEEE transactions on neural networks**, IEEE, v. 20, n. 1, p. 61–80, 2008. Disponível em: <https://doi.org/10.1109/TNN.2008.2005605>.

SILVA, T. C.; ZHAO, L. Network-based high level data classification. **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, v. 23, n. 6, p. 954–970, 2012. Disponível em: <https://doi.org/10.1109/TNNLS.2012.2195027>.

_____. High-level pattern-based classification via tourist walks in networks. **Information Sciences**, Elsevier, v. 294, p. 109–126, 2015. Disponível em: <https://doi.org/10.1016/j.ins.2014.09.048>.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information Processing & Management**, Elsevier, v. 45, n. 4, p. 427–437, 2009. Disponível em: <https://doi.org/10.1016/j.ipm.2009.03.002>.

Szymański, P.; Kajdanowicz, T. A scikit-based Python environment for performing multi-label classification. **ArXiv e-prints**, fev. 2017.

TAN, M. et al. Learning graph structure for multi-label image classification via clique generation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [s.n.], 2015. p. 4100–4109. Disponível em: <https://doi.org/10.1109/CVPR.2015.7299037>.

TROHIDIS, K. et al. Multi-label classification of music into emotions. In: **ISMIR**. [S.l.: s.n.], 2008. v. 8, p. 325–330.

TSOUMAKAS, G.; KATAKIS, I. Multi-label classification: An overview. **International Journal of Data Warehousing and Mining (IJDWM)**, IGI Global, v. 3, n. 3, p. 1–13, 2007. Disponível em: <https://doi.org/10.4018/jdwm.2007070101>.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Effective and efficient multilabel classification in domains with large number of labels. In: **Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)**. [S.l.: s.n.], 2008. v. 21, p. 53–59.

_____. Mining multi-label data. In: **Data mining and knowledge discovery handbook**. Springer, 2009. p. 667–685. Disponível em: <https://doi.org/10.1007/978-0-387-09823-4_34>.

TSOUMAKAS, G. et al. Mulan: A java library for multi-label learning. **Journal of Machine Learning Research**, v. 12, p. 2411–2414, 2011.

TSOUMAKAS, G.; VLAHAVAS, I. Random k-labelsets: An ensemble method for multi-label classification. In: SPRINGER. **European Conference on Machine Learning**. 2007. p. 406–417. Disponível em: <https://doi.org/10.1007/978-3-540-74958-5_38>.

WAN, P.-J.; YI, C.-W. Asymptotic critical transmission radius and critical neighbor number for k-connectivity in wireless ad hoc networks. In: **Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing**. [s.n.], 2004. p. 1–8. Disponível em: <https://doi.org/10.1145/989459.989461>.

WANG, Z.-W. et al. A novel multi-label classification algorithm based on k-nearest neighbor and random walk. **International Journal of Distributed Sensor Networks**, SAGE Publications Sage UK: London, England, v. 16, n. 3, p. 1550147720911892, 2020. Disponível em: <https://doi.org/10.1177/1550147720911892>.

WU, Q. et al. ML-TREE: A tree-structure-based approach to multilabel learning. **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, v. 26, n. 3, p. 430–443, 2015. Disponível em: <10.1109/TNNLS.2014.2315296>.

WU, Z. et al. A comprehensive survey on graph neural networks. **IEEE transactions on neural networks and learning systems**, IEEE, 2020.

YU, Y. et al. Decision network: a new network-based classifier. In: IEEE. **2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C)**. 2020. p. 390–397. Disponível em: <https://doi.org/10.1109/QRS-C51114.2020.00073>.

ZHANG, M.-L. et al. Towards class-imbalance aware multi-label learning. **IEEE Transactions on Cybernetics**, IEEE, 2020. Disponível em: <https://doi.org/10.1109/TCYB.2020.3027509>.

ZHANG, M.-L.; ZHANG, K. Multi-label learning by exploiting label dependency. In: **Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining**. [s.n.], 2010. p. 999–1008. Disponível em: <https://doi.org/10.1145/1835804.1835930>.

ZHANG, M.-L.; ZHOU, Z.-H. Multilabel neural networks with applications to functional genomics and text categorization. **IEEE transactions on Knowledge and Data Engineering**, IEEE, v. 18, n. 10, p. 1338–1351, 2006. Disponível em: <https://doi.org/10.1109/TKDE.2006.162>.

_____. Ml-knn: A lazy learning approach to multi-label learning. **Pattern recognition**, Elsevier, v. 40, n. 7, p. 2038–2048, 2007. Disponível em: <https://doi.org/10.1016/j.patcog.2006.12.019>.

_____. A review on multi-label learning algorithms. **IEEE transactions on knowledge and data engineering**, IEEE, v. 26, n. 8, p. 1819–1837, 2014. Disponível em: <https://doi.org/10.1109/TKDE.2013.39>.