Construção de redes baseadas em vizinhança para o aprendizado semissupervisionado

Lilian Berton

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura:_____

Lilian Berton

Construção de redes baseadas em vizinhança para o aprendizado semissupervisionado

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

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Alneu de Andrade Lopes

USP – São Carlos
Março de 2016

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

B547c

Berton, Lilian
Construção de redes baseadas em vizinhança para o aprendizado semissupervisionado / Lilian Berton; orientador Alneu de Andrade Lopes. -- São Carlos, 216.
135 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 216.

1. construção de redes. 2. redes baseadas em vizinhança. 3. métodos baseados em redes para a classificação. 4. aprendizado semissupervsionado. I. Lopes, Alneu de Andrade, orient. II. Título.

Lilian Berton

Graph construction based on neighborhood for semi-supervised learning

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação - ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Alneu de Andrade Lopes

USP – São Carlos
Março 2016

Agradecimentos

Este doutorado não trouxe apenas novos aprendizados e amadurecimento acadêmico, mas também amizades, colaborações, experiências, contato com outras culturas e novas visões para minha vida profissional e pessoal. Agradeço a todos que me proporcionaram essa oportunidade.

Sou grata a Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP) pelo suporte financeiro, o qual permitiu dedicação exclusiva e cobriu onerações relacionadas a pesquisa.

Agradeço ao meu orientador, prof. Alneu, por acompanhar o desenvolvimento desse trabalho e pelas sugestões para melhoria do mesmo.

Agradeço aos colegas do grupo de pesquisa: Thiago, Jorge, Alan e Nils pelas colaborações realizadas e por suas amizades.

Agradeço a todos os colegas e professores do ICMC/USP que diretamente ou indiretamente colaboraram nesse trabalho.

Agradeço especialmente aos professores Zeza, Marlon, Fábio e Zhao que me orientaram em ICs, TCC e mestrado, me deram bases e motivação para chegar ao doutorado.

Agradeço também ao meu esposo, Didier, por todo apoio, paciência e amor ao longo desse tempo. Obrigada querido!

Resumo

Com o aumento da capacidade de armazenamento, as bases de dados são cada vez maiores e, em muitas situações, apenas um pequeno subconjunto de itens de dados pode ser rotulado. Isto acontece devido ao processo de rotulagem ser frequentemente caro, demorado e necessitar do envolvimento de especialistas humanos. Com isso, diversos algoritmos semissupervisionados foram propostos, mostrando que é possível obter bons resultados empregando conhecimento prévio, relativo à pequena fração de dados rotulados. Dentre esses algoritmos, os que têm ganhado bastante destaque na área têm sido aqueles baseados em redes. Tal interesse, justifica-se pelas vantagens oferecidas pela representação via redes, tais como, a possibilidade de capturar a estrutura topológica dos dados, representar estruturas hierárquicas, bem como modelar *manifolds* no espaço multi-dimensional. No entanto, existe uma grande quantidade de dados representados em tabelas atributo-valor, nos quais não se poderia aplicar os algoritmos baseados em redes sem antes construir uma rede a partir desses dados. Como a geração das redes, assim como sua relação com o desempenho dos algoritmos têm sido pouco estudadas, esta tese investigou esses aspectos e propôs novos métodos para construção de redes, considerando características ainda não exploradas na literatura. Foram propostos três métodos para construção de redes com diferentes topologias: 1) S-kNN (*Sequential k Nearest Neighbors*), que gera redes regulares; 2) GBILI (*Graph Based on the Informativeness of Labeled Instances*) e RGCLI (*Robust Graph that Considers Labeled Instances*), que exploram os rótulos disponíveis gerando redes com distribuição de grau lei de potência; 3) GBLP (*Graph Based on Link Prediction*), que se baseia em medidas de predição de *links* gerando redes com propriedades mundo-pequeno. As estratégias de construção de redes propostas foram analisadas por meio de medidas de teoria dos grafos e redes complexas e validadas por meio da classificação semissupervisionada. Os métodos foram aplicados em *benchmarks* da área e também na classificação de gêneros musicais e segmentação de imagens. Os resultados mostram que a topologia da rede influencia diretamente os algoritmos de classificação e as estratégias propostas alcançam boa acurácia.

Palavras-chave: construção de redes; redes baseadas em vizinhança; métodos baseados em redes para classificação; aprendizado semissupervisionado

Abstract

With the increase capacity of storage, databases are getting larger and, in many situations, only a small subset of data items can be labeled. This happens because the labeling process is often expensive, time consuming and requires the involvement of human experts. Hence, several semi-supervised algorithms have been proposed, showing that it is possible to achieve good results by using prior knowledge. Among these algorithms, those based on graphs have gained prominence in the area. Such interest is justified by the benefits provided by the representation via graphs, such as the ability to capture the topological structure of the data, represent hierarchical structures, as well as model manifold in high dimensional spaces. Nevertheless, most of available data is represented by attribute-value tables, making necessary the study of graph construction techniques in order to convert these tabular data into graphs for applying such algorithms. As the generation of the weight matrix and the sparse graph, and their relation to the performance of the algorithms have been little studied, this thesis investigated these aspects and proposed new methods for graph construction with characteristics litle explored in the literature yet. We have proposed three methods for graph construction with different topologies: 1) S-kNN (*Sequential k Nearest Neighbors*) that generates regular graphs; 2) GBILI (*Graph Based on the informativeness of Labeled Instances*) and RGCLI (*Robust Graph that Considers Labeled Instances*), which exploit the labels available generating *power-law* graphs; 3) GBLP (*Graph Based on Link Prediction*), which are based on link prediction measures and generates *small-world* graphs. The strategies proposed were analyzed by graph theory and complex networks measures and validated in semi-supervised classification tasks. The methods were applied in benchmarks of the area and also in the music genre classification and image segmentation. The results show that the topology of the graph directly affects the classification algorithms and the proposed strategies achieve good accuracy.

Keywords: graph construction; neighborhood graphs; graph-based methods for classification; semi-supervised learning

Sumário

Agradecimentos i
Resumo iii
Abstract v
Lista de Figuras xi
Lista de Acrônimos xiii
Lista de Símbolos xvii

1 Introdução **1**
1.1 Motivação 2
1.2 Identificação do problema 5
1.3 Objetivos e metodologia 6
1.4 Contribuições do trabalho 9
1.5 Estrutura do documento 10

2 Fundamentação teórica e trabalhos relacionados **13**
2.1 Teoria dos grafos 13
2.2 Redes complexas 16
2.2.1 Modelo de redes regulares 16
2.2.2 Modelo de redes aleatórias 17
2.2.3 Modelo de redes mundo-pequeno 18
2.2.4 Modelo de redes livre de escala 19
2.2.5 Medidas de redes 20
2.3 Dados proposicionais e relacionais 23
2.3.1 Representação atributo-valor 23
2.3.2 Representação relacional 24
2.3.3 Diferenças entre dados relacionais e proposicionais 25
2.4 Aprendizado semissupervisionado 26
2.4.1 Aprendizado semissupervisionado e *manifold* 27
2.4.2 Algoritmos semissupervisionados baseados em redes 29
2.4.2.1 Caminhada aleatória de Markov 30
2.4.2.2 Propagação de rótulos 31
2.4.2.3 Agrupamento 34
2.5 Métodos para construção de redes 35
2.5.1 Rede *minimum/maximum spanning tree* (Min/MaxST) 35
2.5.2 Rede ϵ-vizinhança 36
2.5.3 Rede k-vizinhos mais próximos (kNN) 37

2.5.4 Rede *b-matching* 39
2.6 Considerações finais 41

3 Proposta: métodos para construção de redes para o aprendizado semissupervisionado 43
3.1 Construção de redes regulares 45
3.1.1 *Sequential* kNN (S-kNN) 45
3.1.2 Análise de complexidade 47
3.1.3 Visualização da rede S-kNN 47
3.2 Construção de redes a partir das instâncias rotuladas *a priori* em SSL 49
3.2.1 *Graph Based on the Informativeness of Labeled Instances* (GBILI) 49
3.2.2 Análise de complexidade 51
3.2.3 Visualização da rede GBILI 52
3.2.4 *Robust Graph that Considers Labeled Instances* (RGCLI) . . 54
3.2.5 Análise de complexidade 55
3.2.6 Análise dos pressupostos de SSL nas redes GBILI e RGCLI . 58
3.3 Construção de redes via predição de *links* 61
3.3.1 Predição de *links* 61
3.3.2 *Graph Based on Link Prediction* (GBLP) 63
3.3.3 Análise de complexidade 64
3.4 Considerações finais 65

4 Avaliação dos métodos para construção de redes propostos 67
4.1 Análise e aplicação das redes regulares 67
4.1.1 Classificação semissupervisionada 68
4.1.1.1 Configuração dos experimentos 69
4.1.1.2 Experimentos em bases artificiais 69
4.1.1.3 Experimentos em bases reais 72
4.1.1.4 Distribuição da acurácia, dos parâmetros e comparação do tempo de execução 74
4.1.1.5 Discussão 74
4.1.2 Caracterização das redes 77
4.1.2.1 Medidas de redes 77
4.1.2.2 Correlação entre acurácia e medidas de redes 82
4.1.2.3 Discussão 82
4.1.3 Aplicação do método S-kNN na classificação de gêneros musicais 83
4.1.3.1 Características musicais 85
4.1.3.2 Classificadores relacionais 85
4.1.3.3 Base de dados 86
4.1.3.4 Configuração dos experimentos 86
4.1.3.5 Resultados 87
4.1.3.6 Discussão 90
4.2 Análise e aplicação das redes baseadas nas instâncias rotuladas . 91
4.2.1 Classificação semissupervisionada 92
4.2.1.1 Bases de dados 93
4.2.1.2 Configuração dos experimentos 93

4.2.1.3 Resultado da classificação 94
4.2.1.4 Análise de sensibilidade dos parâmetros 95
4.2.1.5 Análise de densidade das redes 95
4.2.2 Medidas de centralidade 99
4.2.3 Aplicação de RGCLI na segmentação de imagens 99
4.2.3.1 Base de dados 101
4.2.3.2 Configuração dos experimentos 101
4.2.3.3 Resultado da segmentação 102
4.2.3.4 Discussão 105
4.3 Análise das redes baseadas em predição de *links* 105
4.3.1 Classificação semissupervisionada 106
4.3.1.1 Bases de dados 106
4.3.1.2 Configuração dos experimentos 106
4.3.1.3 Resultado da classificação 106
4.3.2 Classificação supervisionada 108
4.3.2.1 Bases de dados 108
4.3.2.2 Configuração dos experimentos 109
4.3.2.3 Resultado da classificação 109
4.3.3 Análise das redes 110
4.3.4 Discussão 113
4.4 Considerações finais 114

5 Conclusões **117**
5.1 Resultados e Contribuições 117
5.1.1 Contribuições 119
5.1.2 Publicações 119
5.2 Limitações 121
5.2.1 Alta dimensionalidade 121
5.2.2 Escalabilidade para bases de dados gigantes 121
5.2.3 Rótulos contaminados e desbalanceamento de rótulos 122
5.2.4 Redes heterogêneas 122
5.2.5 Evolução de redes 122
5.3 Trabalhos Futuros 123
5.3.1 Aplicações 123
5.3.2 *Active learning* e SSL 123
5.3.3 Estudos teóricos para construção de redes 123
5.3.4 Rotulagem de dados representativos 124

Referências **135**

Lista de Figuras

1.1 Rede de interligações dos clientes da Tabela 1.1. Elaborado por Osmar Zaiane. 5

2.1 Exemplo de uma rede. 14
2.2 Exemplo de rede regular (a) *lattice* e (b) anel. 17
2.3 Modelo de Watts-Strogatz para redes mundo-pequeno. Adaptada de Watts e Strogatz (1998). 18
2.4 Exemplo de diagrama entidade-relacionamento de um banco de dados de publicações científicas. 24
2.5 Rede de co-autoria em artigos científicos. 25
2.6 Passos para a aplicação de técnicas de aprendizado semissupervisionado baseada em redes. 27
2.7 (a) Duas classes contidas em uma curva. (b) Exemplos rotulados sendo "?" um ponto a ser classificado. (c) pontos não rotulados. (d) Representação ideal da curva. (e) Posição dos dados não rotulados e "?" após aplicar funções Laplacianas. (f) Posição de todos os exemplos. Adaptada de Belkin (2003) 29
2.8 Exemplo de uma rede MinST (a) e de uma rede MaxST (b) para um conjunto de dados com 100 elementos e distribuição gaussiana. . . 36
2.9 Exemplo de redes ϵ-vizinhança para um conjunto de dados com 100 elementos e distribuição gaussiana (a) $\epsilon = 0.3$, (b) $\epsilon = 0.5$ e (c) $\epsilon = 0.9$. 37
2.10 Limites de decisão gerados pelo algoritmo k-vizinhos mais próximos, (a) $k = 1$ e (b) $k = 5$. 38
2.11 Exemplo de redes kNN para um conjunto de dados com 100 elementos e distribuição gaussiana (a) $k = 1$, (b) $k = 3$ e (c) $k = 7$. 38
2.12 Exemplo de rede b-matching para a base de dados Iris, $b = 3$. 40

3.1 Redes S-kNN em um lattice: (a) $k = 1$ (b) $k = 2$ (c) $k = 3$. 46
3.2 Distribuição de grau para as redes geradas a partir dos métodos kNN (a) e S-kNN (b) para a base de dados Breast-Cancer, $k = 7$. Obtida em (Vega-Oliveros et al., 2014). 48
3.3 Redes kNN (a) e S-kNN (b) geradas para a base de dados Glass, $k = 5$. O tamanho dos vértices representa a magnitude do grau. Obtida em (Vega-Oliveros et al., 2014). 48

3.4 Resultado da classificação utilizando: (a) o método proposto; (b) e (c) kNN para a construção das redes. (a) $k = 2$, (b) $k = 2$, (c) $k = 3$. Obtida em (Berton e de Andrade Lopes, 2014). 50
3.5 Construção da rede GBILI (a) $k = 3$, (b) $k = 7$, (c) $k = 10$. Os vértices centrais são rotulados. Obtida em (Berton e de Andrade Lopes, 2012). 50
3.6 Grau médio em uma rede kNN, M-kNN e GBILI construídas com a base USPS. Obtida em (Berton e de Andrade Lopes, 2014). 53
3.7 Redes kNN (a) e GBILI (b) geradas para a base USPS utilizando 10 exemplos rotulados. Obtida em (Berton e de Andrade Lopes, 2014). 53
3.8 Distribuição de grau para a rede GBILI na base de dados USPS. . . 54
3.9 Passos para construção do RGCLI: (a) O método kNN é calculado para $k_e = 15$. (b) A soma das distâncias entre v_i e seus vizinhos mútuos e dos vizinhos mútuos até um vértice rotulado v_l é calculada. (c) São estabelecidas $k_i = 3$ conexões para o vértice v_i. 55
3.10 Comparação de tempo entre GBILI e RGCLI. 57
3.11 (a) Base de dado espiral; (b) rede kNN com $k = 7$; (c) rede M-kNN com $k = 7$; (d) rede RGCLI com $k_e = 7$ e $k_i = 2$. 58
3.12 Passos para a construção da rede GBLP: (a) Base de dados Gaussiana. (b) Rede MinST. (c) Rede MinST+LP (Katz) considerando 30% dos *links* preditos. Obtida em (Berton et al., 2015) 64

4.1 Bases de dados artificiais. (a),(b) e (c) *Gaussian*; (d),(e) e (f) *Spherical*; (g),(h) e (i) *Banana*; (j),(k) e (l) *Difficult*; (m),(n) e (o) *Lithuanian*. . 70
4.2 Comparação dos métodos de construção de redes pelo teste de Nemenyi considerando as bases de dados artificiais. 72
4.3 Comparação dos métodos de construção de redes pelo teste de Nemenyi considerando as bases de dados artificiais. 74
4.4 Histograma de distribuição das taxas de acurácia com os melhores valores de k ou b e diferentes números de vértices rotulados (subescrito) para as bases (a) e (b) Iris, (c) e (d) Glass, (e) e (f) Breast Cancer. 75
4.5 Distribuição dos parâmetros k ou b. 76
4.6 Comparação de tempo de execução entre os métodos de construção de redes. 76
4.7 Visualização das redes para a base Iris, considerando $k = b = 5$. . . 77
4.8 Comparação do número de *hubs* para as bases (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli. 78
4.9 Comparação do grau médio para as bases (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli. 78
4.10 Comparação do diâmetro da rede para as bases (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli. 79
4.11 Comparação da entropia do grau para as bases: (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli. 80
4.12 Comparação do segundo momento para as bases: (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli. 80
4.13 Comparação da complexidade para as bases: (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli. 81

4.14 Comparação do coeficiente de *clustering* para as bases: (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli. 81
4.22 Comparação de todos os classificadores com o teste de Nemenyi. Obtida em (Berton e de Andrade Lopes, 2014). 95
4.23 Comparação da porcentagem da acurácia e desvio padrão nos métodos kNN e GBILI para as bases: (a) digit$_1$, (b) g241c, (c) g241n, (d) COIL$_2$, (e) COIL$_6$ e (f) USPS utilizando-se 10 exemplos rotulados. . . 96
4.24 Comparação da porcentagem da acurácia e desvio padrão nos métodos kNN e GBILI para as bases: (a) digit$_1$, (b) g241c, (c) g241n, (d) COIL$_2$, (e) COIL$_6$ e (f) USPS utilizando-se 100 exemplos rotulados. Obtida em (Berton e de Andrade Lopes, 2014). 97
4.25 Comparação dos valores de ϕ-*edge* entre os métodos kNN e GBILI para as bases (a)digit$_1$, (b)g241c, (c)g241n, (d)COIL$_2$, (e)COIL$_6$ e USPS, utilizando 10 exemplos rotulados. Quanto menor o valor de ϕ-*edge ratio*, melhor a rede. Adaptada de (Berton e de Andrade Lopes, 2014). 98
4.26 (a), (b) e (c) imagens originais. (d), (e) e (f) imagens com os objetos marcados pelo usuário. 102
4.27 Resultado da segmentação: a primeira linha representa o resultado de RGCLI+CLP, a segunda linha representa kNN+CLP e a terceira linha representa M-kNN+CLP. Figuras (a), (b), (c), (g), (h), (i), (m), (n) e (o) $k = 10$. Figuras (d), (e), (f), (j), (k), (l), (p), (q) e (r) $k = 50$. 103
4.28 Segmentação feita por humanos com o contorno detectado: (a), (c) e (e). Versão colorida das segmentações humanas com os segmentos detectados: (b), (d) e (f). Segmentação feita pelo método *Live-wire*: (g-h) 71 cliques do mouse, (i-j) 17 cliques do mouse e (k-l) 5 cliques do mouse. 104
4.29 Teste post-hoc de Nemenyi para a classificação semissupervisionada. Obtida em (Berton et al., 2015). 108
4.30 Teste post-hoc de Nemenyi para a classificação supervisionada. Obtida em (Berton et al., 2015). 111
4.31 Distribuição dos parâmetros k e $percentageL$ utilizados na construção das redes para a classificação supervisionada. Obtida em (Berton et al., 2015). 111
4.32 Grau médio para as redes kNN, M-kNN, MST e GBLP: kNN+LP, M-kNN+LP, MST+LP aplicadas a base Gaussians3. As redes GBLP usaram $k = 3$ e o índice *Common Neighbors*. Obtida em (Berton et al., 2015). 112
4.33 Coeficiente de *clustering* para as redes kNN, M-kNN, MST e GBLP: kNN+LP, M-kNN+LP, MST+LP, aplicadas a base Gaussians3. As redes GBLP usaram $k = 3$ e o índice *Common Neighbors*. Obtida em (Berton et al., 2015) 112
4.34 Média dos caminhos mínimos para as redes kNN, M-kNN, MST e GBLP: kNN+LP, M-kNN+LP, MST+LP, aplicadas a base Gaussians3. As redes GBLP usaram $k = 3$ e o índice *Common Neighbors*. 113

Lista de Acrônimos

AM Aprendizado de Máquina.

BA modelo de rede livre de escala Barabasi e Albert (1999).

CLP *Community Label Propagation*.

ER modelo de rede Erdös e Rényi (1959).

GBILI *Graph Based on Informativeness of Labeled Instances*.

GBLP *Graph Based on Link Prediction*.

IA Inteligência Artificial.

IID Independente e Identicamente Distribuído.

LGC *Local and Global Consistency*.

MaxST *Maximum Spanning Tree*.

MinST *Minimum Spanning Tree*.

RGCLI *Robust Graph that Considers Labeled Instances*.

S-kNN *Sequential k-Nearest Neighbors*.

SSL aprendizado semissupervisionado, em inglês *Semi-Supervised Learning*.

WS modelo de rede Watts e Strogatz (1998).

Lista de Símbolos

γ	potência de uma rede livre de escala
λ	autovalor
$\langle g \rangle$	grau médio
$\langle g^2 \rangle$	segundo momento do grau
$\mathcal{M}$	*manifold*
$\mho$	complexidade da rede
ρ_p	coeficiente de Pearson
ρ_s	coeficiente de Spearman
$\tilde{H}$	entropia
A	matriz de adjacência
B_i	*betweenness* do vértice i
C_c	coeficiente de *clustering* da rede
C_i	coeficiente de *clustering* do vértice i
C_l	closeness
D	matriz diagonal
E	conjunto de arestas
$F(Y)$	função de classificação
g_i	grau do vértice i
I	matriz identidade
L	matriz laplaciana
$l + u$	quantidade de dados rotulados e não rotulados
M	número de arestas

N	número de elementos ou vértices
$N(i)$	conjunto de vizinhos de um vértice i
Sp	espectro da rede
t_i	ciclos de ordem três formados a partir de v_i
V	conjunto de vértices
W	matriz de peso
X	cojunto de dados
X_L	conjunto de dados rotulados
X_U	conjunto de dados não rotulados
Y	atributo classe
z	autovetor

CAPÍTULO

1

Introdução

De acordo com Gantz e Reinsel (2010), em 2009, a quantidade de dados armazenados chegou a aproximadamente 800 mil petabytes. A previsão desses autores para 2020 é que a quantidade de dados criada e replicada no mundo deve chegar a 35 milhões de petabytes. Consequentemente, classificar, organizar ou explorar todos esses dados sem o auxílio de técnicas automáticas é humanamente impossível.

O Aprendizado de Máquina (AM) é um campo de Inteligência Artificial (IA) cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento de forma automática (Micthell, 1997). Além disso, essa área busca o desenvolvimento de métodos computacionais que adquiram novos conhecimentos, habilidades e meios de organizar o conhecimento existente.

As principais técnicas de Aprendizado de Máquina (AM) são divididas em: *aprendizado supervisionado*, *aprendizado não-supervisionado* e *aprendizado semissupervisionado*. No aprendizado supervisionado, o objetivo é induzir conceitos a partir de exemplos previamente rotulados. A tarefa do indutor é encontrar uma função que mapeia os dados em seus respectivos rótulos, a qual é chamada de hipótese ou modelo. Após o processo de treinamento, dado um novo exemplo (não visto no treinamento), o modelo ou hipótese induzida deve ser capaz de classificá-lo.

No aprendizado não-supervisionado, na ausência da informação do rótulo dos exemplos, o objetivo é agrupar os dados segundo algum critério de similaridade estabelecido, ou seja, procura-se determinar agrupamentos de objetos

semelhantes, padrões de co-ocorrências, ou outras regularidades nos dados. O primeiro caso é chamado de agrupamento de dados, no qual o conjunto de dados é dividido em grupos (*clusters*), nos quais os objetos compartilham propriedades comuns. Os casos seguintes são representados por técnicas como regras de associação e sumarização, entre outras.

O aprendizado semissupervisionado, em inglês *Semi-Supervised Learning* (SSL), busca aprender a partir de dados rotulados e não-rotulados. Esse modo de aprendizado tem se destacado e ganhado mais atenção na comunidade científica, já que é bastante útil quando poucos exemplos rotulados encontram-se disponíveis (Chapelle et al., 2010), fato que ocorre na maioria dos casos, principalmente em dados armazenados nos meios digitais.

1.1 Motivação

Instâncias rotuladas são caras e consomem muito tempo para serem obtidas, pois requerem, em geral, que um especialista humano as classifique. Por outro lado, dados não-rotulados são mais fáceis de serem coletados, porém, sua utilização pode ser mais restrita. Desse modo, como o SSL exige um esforço humano menor e produz resultados com acurácia alta, ele vem sendo uma área de grande interesse e estudo, aplicado em várias tarefas, dentre as quais destacam-se algumas comentadas a seguir (Chapelle et al., 2010), (Zhu, 2005). No reconhecimento de fala, registrar grandes quantidades de locuções é pouco custoso, porém, rotulá-las requer um grande trabalho manual. Sequências de proteínas são adquiridas facilmente, mas resolver uma estrutura tri-dimensional ou determinar as funções de uma proteína pode requerer anos de trabalho. Grandes quantidades de páginas *web* estão disponíveis para processamento autônomo, mas classificá-las também exige muito trabalho manual.

Zhu (2005) e Chapelle et al. (2010) destacam que a área de pesquisa mais ativa em SSL tem sido a de métodos baseados em redes. Esses métodos exigem que o conjunto de dados seja representado por vértices de uma rede, no qual os vértices rotulados são utilizados para propagar informações de rótulos para os demais. Desse modo, os métodos baseados em redes, em geral, utilizam uma abordagem de aprendizado transdutiva na qual, diferentemente da abordagem indutiva, todos os exemplos (rotulados e não rotulados) são apresentados ao algoritmo simultaneamente na fase de aprendizagem. Nessa fase, o algoritmo "aprende" a classificar os exemplos não rotulados apresentados, e não se supõe a apresentação de novos dados. A abordagem indutiva, por sua vez, gera um

modelo ou hipótese que será usado para classificação de novos exemplos não vistos pelo indutor durante o treino.

Este crescente interesse pelo uso de redes pode ser justificado pelas vantagens dessa representação, pois métodos baseados em redes permitem: (i) capturação da estrutura topológica dos dados; (ii) representações hierárquicas (uma rede pode ser particionada em subredes, que por sua vez, pode ser dividida em outras subredes, e assim por diante); (iii) detecção de agrupamentos ou classes arbitrárias (Schaeffer, 2007); (iv) combinação de estruturas locais e estatísticas globais (Hein et al., 2007). Mais especificamente, dados representados em redes possibilitam o uso de inferência coletiva (como vértices podem afetar uns aos outros), propagação de rótulos (auto-correlação entre vizinhos) e utilização de características da vizinhança de um vértice (Macskassy e Provost, 2007a). Além de possibilitar o uso de todo formalismo de estudos em teoria dos grafos e, atualmente, em redes complexas (Newman, 2003).

A área de redes complexas estuda redes com milhares ou até bilhões de vértices e topologias não triviais. Existêm diversos modelos de redes, dentre eles destaca-se o modelo de rede mundo-pequeno (Watts e Strogatz, 1998) no qual a média dos menores caminhos na rede é relativamente baixa, isso significa que partindo-se de qualquer vértice da rede pode-se facilmente atingir qualquer outro. As características desse modelo de rede são bastante comuns em redes sociais, nas quais a distância média entre as pessoas é pequena. Outro modelo de rede cujas características são comuns em redes do mundo real é o livre de escala (Barabasi e Albert, 1999), no qual a distribuição de grau segue uma lei de potência, isso significa que poucos vértices da rede possuem muitas conexões e são denominados *hubs*, enquanto a maioria dos vértices possuem poucas conexões. Além dos modelos de redes, diversas medidas têm sido propostas, entre elas medidas de centralidade da rede e medidas globais, relacionadas a distância, modularidade, conectividade.

A representação em redes pode auxiliar na obtenção de diferentes informações comparada à análise de dados no formato atributo-valor. Considere o exemplo hipotético de uma companhia telefônica cujos clientes desejam renovar seus planos de dado/telefone, deseja-se saber para quais clientes a companhia deve dar um incentivo para mantê-los. Ordenando os clientes pelo valor médio semestral gasto em ligações obtém-se a Tabela 1.1. A partir dessa tabela, poderia-se pensar em descartar os clientes que gastam menos de $\$50$, nesse caso seriam 6 clientes menos rentáveis.

Entretanto, construindo-se uma rede a partir das ligações realizadas pelos clientes (ver Figura 1.1) é possível analisar os dados sob essa nova representação. Por meio de medidas de centralidade de redes, por exemplo, observa-se

Tabela 1.1: Dados hipotéticos de uma companhia telefônica. Elaborado por Osmar Zaiane.

ID	Nome	Telefone	Cidade	Gasto semestral
33	**Natalie May**	**403 409 6223**	**Calgary**	**$0,96**
22	**Wilma Renton**	**780 118 2388**	**Edmonton**	**$8,00**
1	**John Smith**	**647 225 8085**	**Toronto**	**$12,00**
24	**Ben Rikon**	**403 262 3134**	**Calgary**	**$26,23**
21	**Patrick Klum**	**403 337 9291**	**Calgary**	**$33,79**
12	**Kent Wafegert**	**647 631 0348**	**Toronto**	**$38,78**
18	Patty Klien	780 550 1819	Edmonton	$50,18
34	Aly Huffington	403 255 0304	Calgary	$55,03
29	Iris Cristle	403 644 1423	Calgary	$64,14
32	Fred Couros	416 773 2234	Toronto	$73,22
23	Ryan Waters	403 715 7550	Calgary	$75,50
4	Randy Regal	705 234 6767	Toronto	$77,10
30	Gunther Twallaby	403 778 6040	Calgary	$78,31
26	Maggie Wong	226 882 0911	Toronto	$89,11
25	Jun Liu	226 690 4241	Toronto	$90,42
9	Wanda Rhymes	403 441 2534	Calgary	$92,00
28	Karen Pollonts	403 750 9201	Calgary	$92,75
7	Susan Willcox	780 291 6063	Edmonton	$131,00
3	John Simon	780 886 5053	Edmonton	$189,45
17	Wayne Jones	780 236 3006	Edmonton	$236,06
15	Brent Mavka	403 566 7372	Calgary	$299,29
6	Mary Smith	780 334 3434	Edmonton	$369,00
16	Brian Olso	403 939 7574	Calgary	$430,78
8	Martha Witherby	780 322 9768	Edmonton	$459,37
14	Kim Cho	780 434 2399	Edmonton	$542,00
20	Morris Slevchuk	780 434 6280	Edmonton	$628,01
5	Jane Smith	780 233 5645	Edmonton	$673,38
2	Joe Burns	416 345 6060	Toronto	$724,00
19	Greg Aderan	403 332 7468	Calgary	$746,82
13	Megan Potink	780 432 5623	Edmonton	$802,00
11	Kurt Locke	780 654 1121	Edmonton	$830,00
10	Julie Austinshaur	403 223 7654	Calgary	$983,12
31	Monica Kwalshuck	403 210 4448	Calgary	$1044,48
27	Joe Garther	416 224 1109	Toronto	$1100,00

que os clientes 33 e 1 são importantes e descartá-los poderia trazer um prejuízo para a companhia. Desse modo, seria mais indicado dar um incentivo para os clients 1 (John Smith - $\$12$) e 33 (Natalie May - $\$0,96$) para mantê-los na companhia. Nesse exemplo, a exploração de diferentes representações dos dados pôde fornecer uma perspectiva diferente obtendo-se *insights* inesperados e levando a vantagem competitiva da companhia.

A representação em rede também é útil para a classificação semissupervisionada, pois os rótulos de alguns vértices são utilizados para predizer os rótulos dos demais vértices. Como as conexões não são estabelecidas de maneira arbitrária, mas representam algum grau de similaridade entre vértices conectados, pode-se inferir que vértices vizinhos possuem características semelhantes e, consequentemente, os mesmos rótulos.

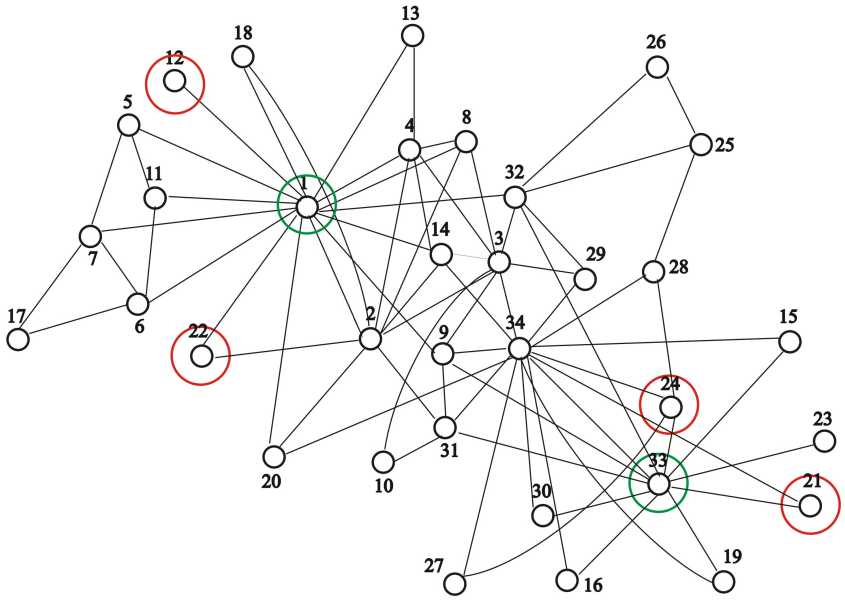


Figura 1.1: Rede de interligações dos clientes da Tabela 1.1. Elaborado por Osmar Zaiane.

1.2 Identificação do problema

Em muitos domínios do mundo real, como por exemplo, Internet, páginas *web*, redes sociais, redes de citações, entre outros, os dados são relacionais por natureza. Tais domínios podem ser "naturalmente" representados por uma rede, na qual vértices representam partes do sistema e arestas as relações entre essas partes. Tal representação relacional baseada em redes, difere da representação proposicional, na qual os dados são representados por tabela atributo-valor. Khosravi e Bina (2010a) destacam as seguintes diferenças: (i) uma base de dados relacional armazena os dados em diversas tabelas, as quais representam diferentes tipos de entidades e relações entre elas. Enquanto os dados proposicionais são armazenados em uma única tabela; (ii) dados proposicionais consistem de entidades com estruturas idênticas, sobre as quais assume-se que são Independente e Identicamente Distribuído (IID), enquanto dados relacionais consistem em entidades de tipos diferentes que podem estar relacionadas entre si; (iii) dados relacionais apresentam autocorrelação, ou seja, os dados possuem ligações e semelhanças entre si.

Observa-se, entretanto, que uma única tabela também pode conter informações relacionais. Seria um caso, por exemplo, de uma tabela não normalizada, com informações redundantes, onde existêm várias instâncias ou tuplas da tabela com relações entre si. Por exemplo, uma tabela que tem a relação de filhos de funcionários (juntamente com a informação do funcionário). Embora neste trabalho, quando dados tabulares são referidos, considera-se neste caso dados

proposicionais, isto é, não se pressupõe relações entre as instâncias da tabela. Ou seja, quando se fala em criar uma rede a partir de uma tabela, as instâncias (linhas da tabela) serão representadas por vértices, e alguma relação entre essas instâncias será criada, em geral baseada na similaridade entre as instâncias.

A maioria das técnicas de aprendizado não baseadas em redes, assume que as instâncias são IID. Desse modo, para utilizar as técnicas baseadas em redes, principalmente no caso do aprendizado semissupervisionado, deve-se criar uma rede a partir das instâncias independentes. No entanto, pouca atenção tem sido dada à construção da rede. A maioria das pesquisas foca apenas nos algoritmos para difusão do rótulo. A importância da construção da rede passou a ser levantada por autores como Maier et al. (2009), Wang e Zhang (2008), Jebara et al. (2009).

Entre os métodos mais populares para construção de redes estão o método ϵ-vizinhança, onde um elemento x_i se conecta com um elemento x_j se $|(x_i, x_j)| < \epsilon$, em que $|.|$ representa a distância entre os pontos, e o método k-vizinhos mais próximos (kNN), o qual conecta cada elemento x_i aos seus k vizinhos mais próximos com base em uma medida de similaridade. Ozaki et al. (2011) argumentam que a presença de *hubs* no espaço dos dados levam a geração de *hubs* na rede kNN degenerando a acurácia da classificação. Em um contexto diferente, Fouss e Parijs (2014) e Knees et al. (2014) melhoram a acurácia de sistemas de recomendação baseados em vizinhança, diminuindo a importância de *hubs*. Jebara et al. (2009) argumentam que redes regulares, na qual todos os vértices possuem o mesmo grau, são mais indicadas para a classificação semissupervisionada, pois além de não produzirem *hubs* são balanceadas. Entretanto, a geração de uma rede regular pode ser custosa computacionalmente.

Apesar de muitos métodos para construção de redes terem sido propostos, essa área de pesquisa ainda continua com muitas questões abertas e merece investigação. Não existêm diretrizes que suportam a escolha de uma técnica em detrimento de outra, nem análises que indiquem os efeitos do grau dos vértices ou da topologia da rede sobre o resultado da classificação. Além disso, a maioria dos métodos não consideram propriedades globais ou os rótulos disponíveis em SSL para a construção das redes.

1.3 Objetivos e metodologia

O objetivo geral desta tese foi analisar métodos para a construção de redes a partir de dados tabulares, presentes na literatura, e desenvolver novos métodos considerando propriedades ainda não exploradas, tempo de processamento

e características topológicas desejáveis, principalmente para a tarefa de classificação. As redes propostas foram aplicadas principalmente em SSL, por meio do uso de algoritmos de classificação baseados em redes.

A hipótese desta investigação é que explorando propriedades estruturais, informações disponíveis *a priori* e esparsidade na distribuição do grau é possível gerar redes mais apropriadas para a classificação semissupervisionada baseada em redes.

Ressalta-se que as redes propostas neste projeto são do tipo homogênea, ou seja, assumem que os objetos da rede são de um mesmo tipo. Diferentemente das redes heterogêneas (Ji et al., 2010), nas quais os objetos da rede e as relações entre eles são de mais de um tipo.

O objetivo geral pode ser subdividido nos seguintes objetivos específicos:

- estudar estratégias de construção de redes presentes na literatura;
- explorar estratégias ainda não empregadas na construção de redes para SSL, como medidas de predição de *links* e uso dos rótulos disponíveis *a priori*;
- investigar propriedades topológicas das redes, como regularidade do grau, redes livre de escala e redes mundo-pequeno, e sua relação com o SSL;
- melhorar a acurácia dos algoritmos semissupervisionados baseados em redes, a partir da utilização das redes propostas;
- desenvolver métodos de construção de redes com tempo de processamento quadrático ou menor, para que possam ser executados em grandes bases de dados;
- avaliar os métodos propostos em bases de dados reais com um grande número de instâncias.

Para o desenvolvimento deste trabalho considerou-se como entrada dados no formato atributo-valor. Assume-se que esses dados são IID e alguns elementos são rotulados. A partir das características desses dados aplica-se uma medida de similaridade obtendo-se uma matriz $N \times N$. Essas similaridades são utilizadas por um algoritmo de construção de redes para estabelecer ligações entre pares de dados. Obtém-se assim uma rede com a estrutura $G(V, E, W)$, onde V é um conjunto de N vértices, correspondentes a cada elemento x_i da tabela. E é um conjunto de arestas, correspondente as ligações estabelecidas entre dois elementos x_i e x_j. W representa o peso/similaridade dessas ligações. Considerando-se o

conjunto de rótulos disponíveis, algoritmos de classificação podem ser utilizados para fornecer uma rotulagem para os vértices não-rotulados.

A partir de conjuntos de dados diferentes, obtêm-se redes com semântica diferente para os vértices e rótulos. Entretanto, a topologia da rede mantém-se constante de acordo com o método de construção de redes utilizado, já que esse não faz distinção sobre a natureza dos dados. Por exemplo, o método kNN estabelece uma ligação entre dois pares de vértices x_i e x_j, se x_i pertence aos k vizinhos mais próximos de x_j ou vice-versa.

As técnicas propostas para construção de redes foram aplicadas em dados no formato atributo-valor obtidos em *benchmarks* da área. Utilizaram-se bases propostas por Chapelle et al. (2010) especialmente para SSL, bases do repositório UCI (Bache e Lichman, 2013) para classificação, e bases do *Berkeley segmentation dataset* para a aplicação em segmentação de imagens. Observa-se que as técnicas de construção de redes propostas não foram desenvolvidas especificamente para esta aplicação, porém, segmentação de imagens é um problema real e desafiador, pois a qualidade da rede é fundamental para a qualidade da segmentação.

Com relação aos algoritmos para classificação, para não ter o viés de muitos algoritmos e focar na variação dos métodos de construção de redes, utilizou-se principalmente o método LGC (*Local and Global Consistency*) (Zhou et al., 2004), o qual é amplamente conhecido e utilizado na área de SSL (Jebara et al., 2009; Ozaki et al., 2011; Li e Fu, 2013). Em alguns casos também realizou-se a classificação supervisionada baseada em redes, onde utilizaram-se os algoritmos de Macskassy e Provost (2003, 2007a); Lu e Getoor (2003). Na aplicação envolvendo segmentação de imagens considerou-se o algoritmo de propagação baseado em comunidades *Community Label Propagation* (CLP) (Raghavan et al., 2007).

Foram propostos três métodos para construção de redes com diferentes topologias: i) S-kNN (*Sequential kNN*) que explora a regularidade do grau e pode ser aplicada em contextos gerais; ii) GBILI (*Graph Based on the Informativeness of Labeled Instances*) e sua versão otimizada RGCLI (*Robust Graph that Considers Labeled Instances*), que exploram os rótulos disponíveis gerando redes livre de escala, as quais podem ser aplicadas em tarefas interativas onde se tem confiabilidade nos rótulos; iii) redes GBLP (*Graph Based on Link Prediction*), que se baseiam em medidas de predição de *links* gerando redes com propriedades mundo-pequeno e podem ser utilizadas para melhorar uma rede já existente.

1.4 Contribuições do trabalho

As contribuições desta tese são apresentadas a seguir.

- Desenvolvimento do método S-kNN (*Sequential kNN*) (Vega-Oliveros et al., 2014) para geração de redes regulares, como alternativa ao método b-matching (Jebara et al., 2009). O método S-kNN apresenta como vantagens: menor tempo de processamento, facilidade na implementação e acurácia na classificação comparável ao b-matching.

- Aplicação do método proposto S-kNN na classificacação de gêneros musicais (Valverde-Rebaza et al., 2014). Neste trabalho propôs-se o uso de abordagens relacionais na classificação de gêneros musicais e mostrou-se que essa abordagem atinge melhores resultados comparados aos de algoritmos tradicionais, como árvores de decisão, *naive Bayes*, redes neurais e *support vector machine*. A partir de características extraídas das músicas, aplicou-se o método proposto S-kNN e o método tradicional kNN para construção das redes, mostrou-se que as redes regulares obtiveram melhores resultados na classificação.

- Análise da influência do grau nos métodos de construção de rede, na qual comparou-se métodos que geram redes com distribuição de grau homogêneas, como S-kNN e b-matching, e métodos que geram redes com distribuição de grau heterogêneas, como kNN. A partir dos experimentos realizados observou-se que redes regulares atingem melhores resultados na classificação e a distribuição da acurácia apresentou diferentes variações, indicando que a estrutura topológica da rede pode influenciar o resultado da classificação. Além disso, medidas de teoria dos grafos e redes complexas foram calculadas. A correlação de Pearson e de Spearman apresentaram uma correlação moderada negativa entre as medidas relacionadas ao grau e a acurácia, indicando que quanto menor o valor das medidas maior a acurácia.

- Desenvolvimento do método GBILI (*Graph Based on the Informativeness of Labeled Instances*) (Berton e de Andrade Lopes, 2012, 2014), o qual explora as informações *a priori*, neste caso os rótulos disponíveis no aprendizado semissupervisionado, para a construção da rede. As principais vantagens do método proposto são: tempo de processamento quadrático, geração de redes esparsas, em que redes com grau médio igual a 2 obtiveram boa acurácia na classificação, pouca sensibilidade ao parametro k especialmente

para valores maiores que dez, formação de *hubs* a partir dos dados rotulados o que facilita a propagação dos rótulos.

- Melhoria do método GBILI e proposta do método RGCLI (*Robust Graph that Considers Labeled Instances*), com tempo de processamento $O(nk \log n)$ e demonstração teórica que o método cumpre com os pressupostos do aprendizado semissupervisionado (ver Seção 2.4). Além disso, aplicou-se o método RGCLI em segmentação de imagens. Nessa aplicação, o usuário seleciona os objetos para reconhecimento na imagem marcando alguns pixels. Estes pixels representam dados rotulados que são utilizados pelo método RGCLI para a construção da rede. Um algoritmo de propagação de rótulos pode ser aplicado, segmentando as imagens. Os resultados mostram que o método RGCLI faz uma melhor segmentação que o método kNN, indicando que o uso de dados *a priori* e sua estrutura topológica favorecem a propagação dos rótulos.

- Proposta do método GBLP (*Graph Based on Link Prediction*) (Berton et al., 2015), o qual faz uso de medidas de predição de *links* para a construção de redes. Nesta abordagem considera-se uma rede esparsa inicial, construída a partir de qualquer método da literatura, tal como kNN ou *minimum spanning tree*. A partir dessa estrutura inicial, aplica-se uma medida para predição de *links* para estimar novas arestas na rede, como por exemplo *common neighbors* ou *Katz* (Valverde-Rebaza e Lopes, 2013). Obtém-se assim uma rede com propriedade mundo-pequeno, com alto coeficiente de *clustering* e a média dos caminhos mínimos relativamente baixa. O método tem tempo de processamento $O(nk \log n)$ considerando kNN e a medida *common neighbors* para predição de *links*.

1.5 Estrutura do documento

O restante deste trabalho está organizado da seguinte maneira:

- no Capítulo 2 são apresentados os conceitos relacionados ao trabalho, como notações e medidas de grafos e de redes complexas utilizadas, as diferenças entre a representação de dados proposicional e relacional, o paradigma semissupervisionado e os principais algoritmos baseados em redes. Além disso, são introduzidas as principais técnicas de construção de redes utilizadas em SSL;

- no Capítulo 3 são apresentadas as propostas para construção de redes explorando diferentes topologias: i) o método S-kNN (*Sequential kNN*) para

construção de redes regulares; ii) os métodos GBILI (*Graph Based on the Informativeness of Labeled Instances*) e RGCLI (*Robust Graph that Considers Labeled Instances*), os quais fazem uso dos vértices rotulados para construção da rede e geram redes livre de escala; iii) as redes GBLP (*Graph Based on Link Prediction*), que fazem uso de medidas de predição de *links* e geram redes com propriedades mundo-pequeno;

- no Capítulo 4 são apresentadas avaliações para as redes propostas, englobando: i) análises entre métodos que geram redes com distribuição de grau homogêneas versus métodos que geram redes com distribuição de grau heterogêneas; ii) aplicação da rede S-kNN em classificação de genêros musicais; iii) caracterização das redes GBILI; iv) aplicação da rede RGCLI em segmentação de imagens; v) aplicação das redes GBLP na classificação baseada em redes;

- no Capítulo 5 são apresentadas as contribuições e publicações geradas, as limitações e trabalhos futuros.

CAPÍTULO

2

Fundamentação teórica e trabalhos relacionados

Neste capítulo são introduzidos conceitos e medidas de redes, do aprendizado semissupervisionado e de construção de grafos, que foram empregados no trabalho. Nas Seções 2.1 e 2.2 são apresentadas algumas definições e medidas de grafos e redes complexas respectivamente. Na Seção 2.3 são diferenciados dados proposicionais e relacionais. Na Seção 2.4 é apresentado o paradigma semissupervisionado, seus pressupostos e os algoritmos baseados em redes para classificação semissupervisionada. Por fim, na Seção 2.5 são apresentados os principais algoritmos para construção de redes para o aprendizado semissupervisionado.

2.1 Teoria dos grafos

Uma rede $G(V, E)$ representa um conjunto finito $V = \{v_1, v_2, \ldots, v_N\}$, cujos N elementos são denominados *vértices*, juntamente com um conjunto $E = \{e_1, e_2, \ldots, e_M\}$ composto por M pares não-ordenados de V, cujos elementos são denominados *arestas*, tal que, uma aresta e_k ligando os vértices v_i e $v_j \in V$ é denotada por $e_{i,j}$. Do mesmo modo, uma rede dirigida é definida por um conjunto finito V e E, porém os elementos em E são direcionados. Uma rede $G'(V', E')$ é uma *subrede* de $G(V, E)$, se $V' \subseteq V$ e $E' \subseteq E$.

Uma rede ponderada $G(V, E, W)$ representa um conjunto de vértices V, um conjunto de arestas E e uma matriz de pesos W, sendo w uma função $w : E \to \Re$ que atribui para cada aresta $e_{ij} \in E$ um peso $w(e_{ij})$, ou simplesmente w_{ij}.

Dois vértices são ditos *adjacentes* ou *vizinhos*, se eles são conectados por uma aresta. A *matriz de adjacência* A de uma rede cujo conjunto de vértices é dado por $v_1, v_2, \ldots, v_N$ é uma matriz quadrada de ordem N, cuja entrada a_{ij} representa uma aresta iniciando no vértice v_i e terminando no vértice v_j.

Por exemplo, a matriz de adjacência da rede mostrada na Figura 2.1 é dada por:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Figura 2.1: Exemplo de uma rede.

Se cada vértice em G for adjacente a todos os demais, então G é uma rede *completa*. Uma subrede completa de G é chamada *clique*.

A vizinhança de um vértice v_i refere-se ao seu conjunto de vizinhos, sendo denotada por $N(i)$. O número de vizinhos ou arestas incidentes em um vértice não-direcionado é chamado de *grau* (ou valência) do vértice, com cada aresta contribuindo para o grau dos dois vértices que a limita. Nas redes dirigidas distingue-se entre os graus de entrada do vértice e os graus de saída, indicando respectivamente quantas arestas entram e saem do vértice. O grau g_i é definido em termos de uma matriz de adjacência conforme a Equação 2.1.

$$g_i = \sum_{j=1}^{N} A_{ij} \tag{2.1}$$

O grau médio $\langle g \rangle$ representa a média do grau de todos os vértices da rede, é obtido conforme a Equação 2.2.

$$\langle g \rangle = \frac{1}{N} \sum_{i=1}^{N} g_i \tag{2.2}$$

Uma rede pode ser caracterizada por sua distribuição de grau, $P(g)$, definida como a fração de vértices na rede com grau g. Pode-se fazer um gráfico de $P(g)$ para se obter informações de como o grau está distribuído entre os vértices.

Através do grau dos vértices é possível encontrar vértices altamente conectados, denominados *hubs*, os quais podem ter uma função importante na rede.

Percurso é uma sequência finita de k vértices e $k-1$ arestas na qual v_i e v_{i+1} são conectados por uma aresta para $i = 1, \ldots, k-1$. *Cadeia* é um percurso sem repetição de arestas. *Caminho* é uma cadeia sem repetição de vértices. O tamanho de um caminho é o número de arestas contido nele. Um *ciclo* é uma cadeia fechada.

Uma rede é *conexa* se existe um caminho entre todos os vértices, e *desconexa* caso contrário. Uma rede desconexa consiste de duas ou mais partes chamadas *componentes*. Dois vértices estão em componentes diferentes se eles não podem ser conectados por um caminho.

Partindo-se de um vértice v_i, é possível determinar por quantas arestas é necessário passar para se chegar a qualquer vértice v_j de uma rede. Quando se encontra o menor número de arestas necessário, este caminho percorrido é chamado de *menor caminho* entre v_i e v_j, ou *caminho geodésico*. A *excentricidade* de um vértice v_i é a maior distância geodésica entre v_i e qualquer outro vértice. O *raio* de um grafo é a excentricidade mínima de qualquer vértice do grafo. O *diâmetro* de um grafo é a excentricidade máxima de qualquer vértice do grafo.

O símbolo I representa uma matriz identidade. Se A é uma matriz $N \times N$ simétrica então um vetor não-nulo $z \in \Re^N$ é chamado um *autovetor* de A se Az é um múltiplo escalar de z, ou seja, $Az = \lambda z$ para algum escalar $\lambda \in \Re$. O escalar λ é chamado um *autovalor* de A e dizemos que z é um autovetor associado a λ.

O polinômio característico $|\lambda I - A|$ de uma matriz de adjacência A de uma rede G é chamado de polinômio característico de G e denotado por $P_G(\lambda)$. Os autovalores de A (i.e. os zeros de $|\lambda I - A|$) e o espectro de A (que consistêm dos autovalores) são chamados também de autovalores de G, respectivamente. Se $\lambda_1, \ldots, \lambda_N$ são os autovalores de G, o espectro é denotado por $Sp(G) = [\lambda_1, \ldots, \lambda_N]$.

Os autovalores de A também podem ser definidos como os números λ que satisfazem $Az = \lambda z$ para um vetor não-zero z. Para a rede da Figura 2.1, tem-se:

$$P_G(\lambda) = \begin{pmatrix} \lambda & -1 & 0 & 0 \\ -1 & \lambda & -1 & 0 \\ 0 & -1 & \lambda & -1 \\ 0 & 0 & -1 & \lambda \end{pmatrix} = \lambda^4 - 3\lambda^2 + 1$$

$$Sp(G) = \frac{1+\sqrt{5}}{2}, \frac{-1+\sqrt{5}}{2}, \frac{1-\sqrt{5}}{2}, \frac{-1-\sqrt{5}}{2}$$

Dada uma amostra sobre um *submanifold* no espaço euclidiano, pode-se construir uma rede baseada em vizinhança, a qual pode ser vista como uma aproximação do *submanifold*. A matriz Laplaciana desta rede é usada em vários métodos de Aprendizado de Máquina (AM) como em aprendizado semissupervisionado (Belkin e Niyogi, 2004), (Zhou et al., 2004), (Zhu e Ghahramani, 2002), agrupamento espectral (Luxburg, 2007) e redução de dimensionalidade (Belkin e Niyogi, 2003).

A matriz *Laplaciana* de uma rede ponderada é a matriz $L = D - W$, na qual D é uma matriz diagonal $D(i,i) = \sum_j W(i,j)$. Já a matriz laplaciana normalizada é dada por $L = I - D^{1/2} W D^{1/2}$.

Se os dados residem no $\Re^d$, a rede baseada em vizinhança, construída a partir de uma amostra aleatória, pode ser vista como uma aproximação da estrutura contínua. Em particular, se os dados estão em um *submanifold* de baixa dimensionalidade, a rede baseada em vizinhança é uma aproximação discreta do *submanifold*. Em AM se está interessado nas propriedades intrínsecas e objetos deste *submanifold*. Diversos autores têm feito análises teóricas sobre a matriz Laplaciana, para mais detalhes ver: Hein et al. (2005), Belkin e Niyogi (2005), Hein (2006) e Hein et al. (2007).

2.2 Redes complexas

A partir da metade da década de 90, as redes de *Internet* e a *World Wide Web* (WWW) começaram a destacar-se. Essas redes são grandes, dinâmicas e possuem arquiteturas complexas, surgindo assim a necessidade de entendimento de seu funcionamento e organização global (Dorogovtsev e Mendes, 2003). Alguns pesquisadores começaram a estudar diversas redes naturais, como por exemplo, redes de citações científicas (Redner, 1998), redes de reações metabólicas (Jeong et al., 2000), redes de interação de proteínas (Jeong et al., 2001), entre outras, e perceberam que essas redes apresentam propriedades topológicas, como padrão de conexão não trivial, presença de *hubs*, grande número de vértices e arestas, que não são encontradas em redes mais simples. Essas redes passaram a ser denominadas redes complexas. Alguns modelos dessas redes são descritos a seguir.

2.2.1 Modelo de redes regulares

Em uma rede regular, todos os vértices possuem o mesmo grau. Uma rede regular direcionada deve satisfazer a condição de que o grau de entrada e de

saída são iguais. Uma rede regular com vértices de grau r é chamada r-regular. Para um rede r-regular de ordem N existir, $N \geq r + 1$ e Nr deve ser par.

Uma rede regular pode ser representada por um *lattice* (Figura 2.2 (a) onde cada vértice cinza tem o mesmo grau, conectando-se os vértices das extremidades inferior e superior todos os vértices teriam o mesmo grau). Uma rede regular também pode ser representada por um anel (Figura 2.2 (b)).

Existem algoritmos para a geração de redes regulares, conforme o trabalho de Meringer (1999), entretanto essa tarefa pode ser muito custosa computacionalmente.

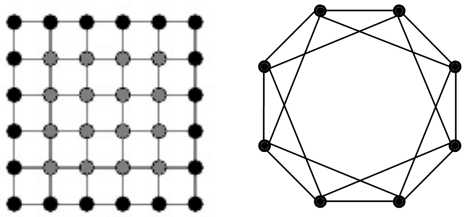


Figura 2.2: Exemplo de rede regular (a) *lattice* e (b) anel.

2.2.2 Modelo de redes aleatórias

Erdös e Rényi (1959) propuseram as redes aleatórias (modelo de rede Erdös e Rényi (1959) (ER)). Nestas redes: (i) o número total de vértices, N, é fixo; (ii) a probabilidade de dois vértices arbitrários se conectarem é igual a $\hat{p}$. Em média, a rede contém $\hat{p}N(N-1)/2$ arestas. A distribuição do grau é binomial conforme a Equação 2.3.

$$P(g) = \begin{pmatrix} N-1 \\ g \end{pmatrix} \hat{p}^g (1-\hat{p})^{N-1-g} \tag{2.3}$$

assim, o grau médio é $\langle g \rangle = \hat{p}(N-1)$. Para valores de N muito grande, a distribuição do grau (Equação 2.3) assume a forma de Poisson, dada pela Equação 2.4.

$$P(g) = e^{\langle g \rangle} g^{\langle g \rangle} / g! \tag{2.4}$$

Isso significa que a maioria dos vértices terá grau próximo de $\langle g \rangle$ e não existe a presença de *hubs* na rede.

2.2.3 Modelo de redes mundo-pequeno

Muitas redes reais, apesar de serem muito grandes, possuem um caminho relativamente curto entre quaisquer pares de vértices. Essa característica é conhecida como propriedade mundo-pequeno em inglês *small-world* e foi inicialmente observada por Milgram (1967). Em um experimento enviando cartas nos Estados Unidos, Milgram descobriu que a distância média entre as pessoas em uma rede social é muito pequena, em média, existem seis graus de separação entre duas pessoas quaisquer.

Do mesmo modo, Watts e Strogatz (1998) notaram que as redes reais possuíam essa característica, além de uma grande presença de ciclos de ordem três, i.e, ciclos que contém três arestas, muito maior em comparação com redes aleatórias semelhantes (com valores iguais de N e M). Desse modo, foi inicialmente evidenciado que as redes reais não são totalmente aleatórias, e que existem algumas regras que governam sua formação. Foi proposto um método de construção de redes conhecido como modelo de rede Watts e Strogatz (1998) (WS), no qual é possível variar a proporção de ciclos de ordem três na rede. O modelo consiste em um processo de re-ligação de arestas dada uma probabilidade $\hat{p}$.

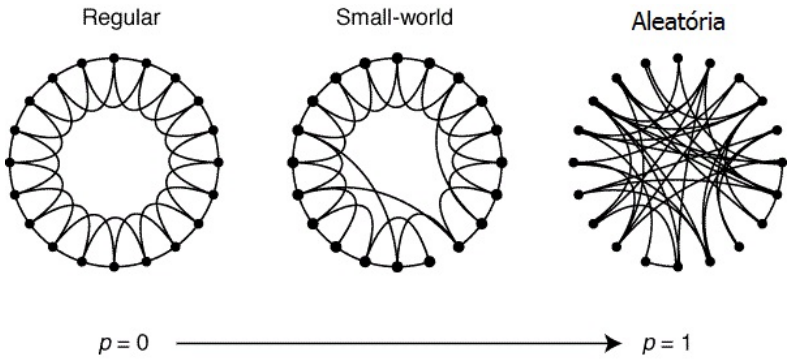


Figura 2.3: Modelo de Watts-Strogatz para redes mundo-pequeno. Adaptada de Watts e Strogatz (1998).

O processo começa com um *lattice* unidimensional regular de N vértices, no qual cada vértice é simetricamente ligado aos seus $2g$ vizinhos mais próximos para um total de $m = gN$ arestas. Para cada vértice, cada aresta ligada a um vizinho no sentido horário é re-ligada a outro vértice escolhido aleatoriamente com uma probabilidade $\hat{p}$, ou preservada com uma probabilidade $1 - \hat{p}$. Nota-se que para $\hat{p} = 1$ o modelo gera uma rede aleatória, com baixa quantidade de ciclos de ordem três, com caminho médio pequeno mas com a restrição de que cada vértice tem conectividade mínima $g_{min} = g$. A rede aleatória é diferente do modelo ER, pois o número mínimo de arestas é mantido mas os graus dos vértices são

diferentes. Quando $\hat{p} = 0$, o modelo gera redes com alta presença de triângulos e regular.

No caso em que $\hat{p} \to 1$ temos que $P(g)$ segue uma distribuição de Poisson, com $g' = g - m$ e $\langle g' \rangle = m$, isto é dado pela Equação 2.5.

$$P(g) = \frac{(m)^{g-m}}{(g-m)!} e^{-m} \,. \tag{2.5}$$

Considerando $\hat{p} = 0$, o número esperado de ciclos de ordem três ou conexões entre vizinhos para cada vértice é $C_i = 3m(m-1)/2$ e o número total esperado de conexões seria $C_c = 2m(2m-1)/2$. Assim, pode-se definir o valor máximo do coeficiente de agrupamento fazendo C_i/C_c, obtendo como resultado $C_{\max} = \frac{3(m-1)}{2(2m-1)}$, que para fins práticos e valores grandes de m, temos que $C_{\max} \leq 3/4$.

2.2.4 Modelo de redes livre de escala

Nos modelos anteriores, a probabilidade de dois vértices se conectarem está definida por uma probabilidade constante $\hat{p}$, seguindo uma distribuição Poisson para o grau dos vértices. Em contraste, Barabasi e Albert (1999) encontraram que em muitas redes reais existe um grande número de vértices com poucas conexões e uma pequena minoria de vértices com muitas conexões. Esse tipo de rede segue uma distribuição lei de potência em inglês *power-law* (Albert e Barabasi, 2002; Newman, 2003), do tipo $P(g) \sim g^{-\gamma}$. A distribuição de grau é altamente heterogênea, com alguns poucos *hubs* ligados a muitos vértices que possuem graus baixos.

As redes livre de escala são um subconjunto das redes com distribuição lei de potência, dado que seu γ varia entre $[2, 3]$. São denomindadas livre de escala porque a média do grau $\langle g \rangle$ não representa um valor significativo de escala na rede.

Barabasi e Albert (1999) propõem um modelo que gera redes livre de escala com $\gamma = 3$, conhecido como modelo de rede livre de escala Barabási e Albert. O processo inicia com m_0 vértices totalmente conectados. A cada passo de tempo $t = 1, 2, 3...n - m_0$ um novo vértice j com m arestas é adicionado à rede. A probabilidade que o novo vértice j se conecte com um vértice da rede é dada pela Equação 2.6.

$$\prod(g_i) \;=\; \frac{g_i}{\sum_j \; g_j} \,, \tag{2.6}$$

onde o denominador é a soma do grau para todos os vértices. Esta abordagem faz com que os vértices com maior número de conexões tenham maior probabi-

lidade de se conectarem com os novos vértices. Essa característica, denomidada *preferential attachment*, é a principal diferença dos modelos livre de escala com os anteriores. Pela regra de conexão preferencial, as conexões entre os vértices não acontecem de modo aleatório e sim de modo preferencial, onde os "ricos ficam cada vez mais ricos". Todos os vértices novos com suas M arestas são conectados seguindo esse procedimento.

2.2.5 Medidas de redes

Existem diversas medidas que calculam a importância de um vértice em uma rede, a seguir, algumas medidas empregadas neste trabalho são apresentadas. Para uma lista mais completa ver Newman (2003).

A partir da distribuição de grau, pode-se calcular o grau de entropia e o grau médio de heterogeneidade da rede. Considerou-se a versão da *entropia de Shannon* ($\tilde{H}$) dada pela Equação 2.7.

$$\tilde{H} = -\frac{\sum_{g=0}^{\infty} P(g)\log(P(g))}{\log(N)}, \quad 0 \leq \tilde{H} \leq 1 \tag{2.7}$$

O valor máximo da entropia é alcançado quando a distribuição do grau é uniforme. O valor mínimo ocorre em redes cujos vértices possuem o mesmo grau. Além disso, a entropia está relacionada ao grau de resiliência da rede (Wang et al., 2006).

A *assortatividade* mede a correlação entre o grau dos vértices. Em uma rede *assortativa*, vértices com grau semelhante são conectados entre si. Em uma rede *disassortativa*, vértices de grau baixo contectam-se com vértices de grau alto (*hubs*). Redes *não-assortativas* não apresentam relações entre o grau dos vértices. Calculou-se a assortatividade por meio do coeficiente de Pearson (ρ_p) (Newman, 2002):

$$\rho_p = \frac{(1/M)\sum_{j>i} g_i g_j A_{ij} - \left[(1/M)\sum_{j>i} \frac{(g_i+g_j)A_{ij}}{2}\right]^2}{(1/M)\sum_{j>i} \frac{({g_i}^2+{g_j}^2)A_{ij}}{2} - \left[(1/M)\sum_{j>i} \frac{(g_i+g_j)A_{ij}}{2}\right]^2} \tag{2.8}$$

Se $\rho_p > 0$ a rede é assortativa, se $\rho_p < 0$ é dissortativa e se $\rho_p = 0$, não existe correlação entre o grau dos vértices.

Transitividade ou *coeficiente de clustering* mede o grau com que os nós de uma rede tendem a agrupar-se. É uma propriedade comum em redes sociais referindo-se aos círculos de amigos, i.e., se A é amigo de B e C, existe uma forte tendência de B e C também serem amigos. O coeficiente de *clustering* (C_i)

proposto por Watts e Strogatz (1998) é definido localmente para cada vértice como:

$$C_i = \frac{2t_i}{g_i(g_i - 1)} \tag{2.9}$$

onde $g_i(g_i - 1)/2$ é o número máximo de conexões possíveis e t_i é o número de ciclos de ordem três formados a partir do vértice v_i. Se $g_i \leq 1$, então $C_i = 0$. C_i possui valor 1 se todos os vizinhos de i são conectados. O coeficiente de clustering (C_c) na rede é caracterizado por:

$$C_c = \frac{1}{n}\sum_{i=1}^{n} C_i. \tag{2.10}$$

Barrat et al. (2008) argumentam que observando a distribuição do grau considerando uma escala simples, onde $g \to \delta g$ e δ representa o fator de escala, a distribuição ainda teria a mesma forma. Isso implica que também é possível obter uma caracterização heurística do nível de heterogeneidade de uma rede por $\mho = \frac{\langle g^2 \rangle}{\langle g \rangle}$. Desse modo, redes livre de escala são caracterizadas por $\langle g^2 \rangle \to \infty$, ou seja, o segundo momento da distribuição de grau tende a infinito para N grande. Redes regulares ou com distribuição de grau homogêneas possuem $g_i \approx \langle g \rangle$. Denominou-se essa medida como *complexidade* da rede.

A medida *closeness* (Sabidussi, 1996) quantifica quão próximo um vértice v_i está em relação aos outros vértices da rede sendo definida como:

$$C_l = \frac{1}{\sum_{x_j \in V} |x_i, x_j|} \tag{2.11}$$

onde $|.|$ refere-se a uma medida de distância. Neste trabalho, inspirado nessa medida, definiu-se um medida análoga para instâncias em um espaço multidimensional, referenciada como *closeness'* C_l'. O *closeness'* de uma instância x_i é definido como $C_l' = \frac{N}{\sum_{j \neq i} |x_i, x_j|}$. Instâncias com alto *closeness'* são instâncias mais próximas de outras considerando, por exemplo, uma medida de distância euclidiana.

Neste trabalho também explorou-se medidas de centralidade em alguns experimentos, as medidas exploradas foram *grau do vértice*, *betweenness* (Freeman, 1977), *eigenvector* (Bonacich, 1972) e *pageRank* (Brin e Page, 1998). Medidas de centralidade indicam quanto um vértice é importante dentro de um contexto. Por exemplo, o grau do vértice indica a popularidade do mesmo na rede, betweenness considera quão importante é um vértice na transmissão de informação na rede, eigenvector e pageRank consideram a importância de um vértice dada a importância de suas conexões.

Caminhos mínimos desempenham um papel importante no transporte e comunicação em uma rede. Supondo que um pacote de dados precisa ser enviado de um computador a outro pela Internet, o caminho mínimo pode indicar o caminho ótimo para transferência do mesmo, considerando velocidade e economia de recursos. A comunicação entre dois vértices não adjacentes, j e k, depende dos vértices no caminho que conecta j a k. Desse modo, uma medida de relevância que um dado vértice pode ter faz a contagem da fração de caminhos mínimos que passam por este vértice. Essa medida é denominada betweenness (B_i) (Freeman, 1977) e pode ser calculada por:

$$B_i = \sum_{j,k \in N,\, j \neq k} \frac{d_{jk}(i)}{d_{jk}}, \tag{2.12}$$

onde d_{jk} é o número de caminhos mínimos conectando todos os pares j e k da rede, enquanto $d_{jk}(i)$ é o número de caminhos mínimos conectando j e k e passando por i.

Em muitos casos, um vértice é importante se possuir muitas conexões ou se estiver conectado a outros vértices importantes. Bonacich (1972) propôs uma medida que calcula a importância de um vértice considerando a importância de seus vizinhos: $Az = \lambda z$. Nesta medida $z = \{z_1, z_2, \ldots, z_N\}$ é o vetor de centralidade para cada vértice i e λ é um valor constante que representa o autovalor de A com z sendo seu autovetor correspondente. Computacionalmente, pode-se definir o valor de cada z_i como um valor calculado a partir de um número de interações no tempo, dado pela Equação 2.13.

$$z_i^t = \sum_j A_{ij} z_j^{t-1}, \tag{2.13}$$

onde t é o tempo de interação e para $t = 0$ assume-se $z^0 = \{1, \ldots, 1\}$.

PageRank foi proposto por Brin e Page (1998) e considera um vértice i importante se ele recebe mais conexões que um dado j, e se recebe conexões de outros vértices importante. Essa medida é usada pela *Google*[1] para ranquear páginas *webs* e simular o comportamento de um usuário na Internet. Um usuário pode visitar páginas diferentes, clicando em *hyperlinks* ou escrevendo uma nova URL no *browser*, por exemplo. PageRank pode ser calculada pela Equação 2.14.

$$\prod_i^t = \alpha \sum_j \frac{A_{ij} \prod_j^{t-1}}{g_j^{out}}, \tag{2.14}$$

[1] `http://www.google.com`

onde g_j^{out} é o grau de saída do vértice j, o qual diminui a importância de i. Pode-se utilizar um ajuste estocástico, fazendo com que j tenha a mesma probabilidade de se conectar com qualquer outro vértice na rede, dado por $\widetilde{g}_j^{out} = g_j^{out} + \delta(g_j^{out}, 0)(1/N)$. Além disso, um ajuste pode ser considerado fazendo com que i possa pular para j dada uma probabilidade $(1 - \alpha)$, tal que α indica a probabilidade de navegação sequencial.

2.3 Dados proposicionais e relacionais

Nessa tese, investigou-se a construção de redes a partir de dados tabulares. Raedt (2008) propõe uma hierarquia das principais representações dos dados, incluindo booleana, atributo-valor, multi-instância e relacional, em termos de expressividade. A seguir as representações atributo-valor e relacional são apresentadas com mais detalhes.

2.3.1 Representação atributo-valor

Na representação atributo-valor, os dados são caracterizados por um conjunto de K atributos $T_1, T_2, \ldots, T_K$, sendo que cada exemplo pode assumir valores específicos para cada atributo. Desse modo, os atributos definem um espaço de exemplos $T = T_1 \times T_2 \times \ldots \times T_K$, e um exemplo $\vec{t} \in T$ é representado como um vetor $(t_1, t_2, \ldots, t_K)$ no qual t_j refere-se ao valor do atributo T_j em $\vec{t}$. Os valores dos atributos podem ser contínuos, discretos ou booleanos. Na obtenção de classificadores em AM, um atributo que tem um tratamento diferenciado é a *classe*, o qual sempre assume valores discretos.

Na Tabela 2.1 é apresentado o formato atributo-valor para N exemplos rotulados $X_1, \ldots, X_N$. Cada exemplo rotulado X_i assume a forma $(\vec{t_i}, y_i)$, sendo que $\vec{t_i} = (t_{i1}, t_{i2}, \ldots, t_{iK}) \in T$ e y_i é o valor do atributo classe Y referente a t_i.

Tabela 2.1: Exemplos rotulados no formato atributo-valor

	T_1	T_2	$\ldots$	T_K	Y
X_1	t_{11}	t_{12}	$\ldots$	t_{1K}	y_1
X_2	t_{21}	t_{22}	$\ldots$	t_{2K}	y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_N	t_{N1}	t_{N2}	$\ldots$	t_{NK}	y_N

Essa representação tem a mesma expressividade da representação proposicional, pois os objetos representados na tabela podem ser descritos por "proposições" a respeito dos atributos e valores desses atributos de acordo com o objeto em particular.

2.3.2 Representação relacional

No caso de um esquema relacional derivar de um modelo entidade-relacionamento (*entity-relationship*) (Ullman, 1982), assume-se que as tabelas se dividem em *tabelas entidades* e *tabelas relacionamentos*.

Na Figura 2.4 é mostrado um diagrama entidade-relacionamento de publicações científicas, demonstrando as relações existentes entre os dados. Nas Tabelas 2.2, 2.3 e 2.4 são representadas as informações dos artigos, dos autores e das co-autorias respectivamente.

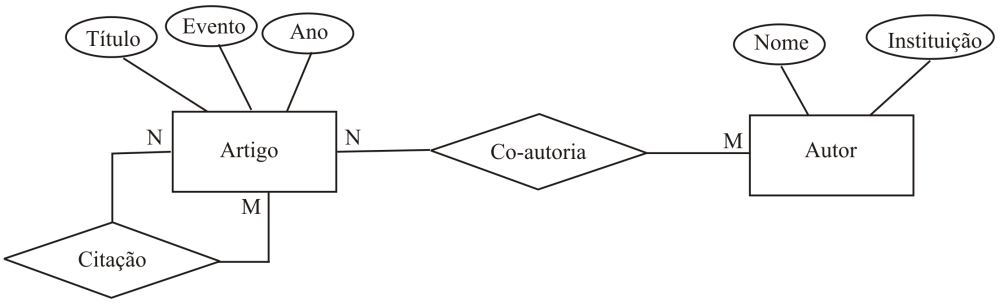


Figura 2.4: Exemplo de diagrama entidade-relacionamento de um banco de dados de publicações científicas.

Tabela 2.2: Artigo

título do artigo	evento	ano
Finding community structure in very large networks	*Physical Review E*	2004
Community structure in social and biological networks	*PNAS*	2002
The Power of Choice in Network Growth	*European Journal of Physics B*	2007
Explosive Percolation in Random Networks	*Science*	2009

Tabela 2.3: Autor

autor	instituição
Aaron Clauset	*University of New Mexico*
Mark Newman	*University of Michigan*
Cristotpher Moore	*University of New Mexico*
Michelle Girvan	*University of Michigan*
Raissa D´Souza	*University of California*
Paul Krapivsky	*Boston University*
Dimitrius Achlioptas	*University of California*
Joel Spencer	*New York University*

Segundo Raedt (2008) a representação relacional apresenta uma descrição mais rica dos dados, pois além das informações dos objetos, representa também a relação existente entre eles. Um conjunto de objetos do mesmo tipo, como autores ou artigos, com alguma relação específica entre eles, relação de co-autoria,

Tabela 2.4: Co-autoria

título do artigo	autor
Finding community structure in very large networks	Aaron Clauset
Finding community structure in very large networks	Mark Newman
Finding community structure in very large networks	Cristotpher Moore
Community structure in social and biological networks	Michelle Girvan
Community structure in social and biological networks	Mark Newman
The power of Choice in Newtork Growth	Raissa D´Souza
The power of Choice in Newtork Growth	Paul Krapivsky
The power of Choice in Newtork Growth	Cristopher Moore
Explosive Percolation in Random Networks	Dimitrius Achlioptas
Explosive Percolation in Random Networks	Raissa D´Souza
Explosive Percolation in Random Networks	Joel Spencer

por exemplo, pode ser representada em uma rede. Essa rede, é portanto um caso específico de representação relacional que consegue com naturalidade uma expressividade maior que a representação atributo-valor.

Considerando o conjunto de dados apresentado na Tabela 2.4 é possível modelar em uma rede as relações de co-autoria (Figura 2.5), de modo que os vértices representem os autores e as arestas as participações conjuntas em artigos.

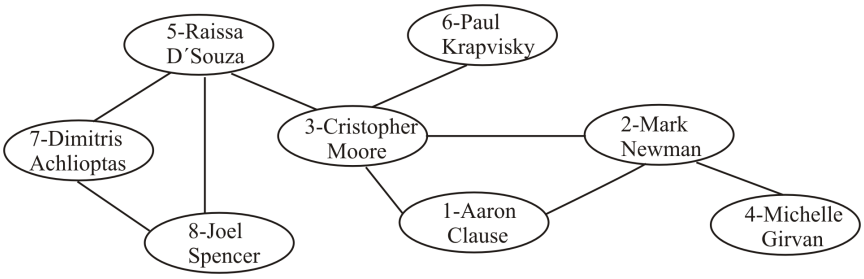


Figura 2.5: Rede de co-autoria em artigos científicos.

Diversos domínios relacionais podem ser modelados por redes, por exemplo, páginas *web*, Internet, redes de proteínas, redes sociais, entre outras. Como os dados relacionais possuem mais informações sobre um objeto por meio dos seus *links*, usando-se dados relacionais, pode-se obter resultados com melhores acurácias em classificação e predição, bem como em tarefas como classificação coletiva (Jensen et al., 2004), agrupamento baseado em *link* (Wang e Kitsuregawa, 2001) e predição de *links* (Taskar et al., 2004).

2.3.3 Diferenças entre dados relacionais e proposicionais

Dados relacionais apresentam diversas diferenças com relação aos dados proposicionais. Khosravi e Bina (2010b) destacam os seguintes.

1. Quanto ao método de representação: uma base de dados relacional armazena os dados em diversas tabelas, as quais representam diferentes tipos de entidades e relações entre elas. Enquanto os dados proposicionais são armazenados em uma única tabela.

2. Dados proposicionais consistem de entidades com estruturas idênticas, sobre as quais assume-se que são independentes e identicamente distribuídas (IID), enquanto dados relacionais consistem em entidades de tipos diferentes que podem estar relacionadas entre si. Por exemplo, em dados proposicionais, todas as tuplas são do tipo estudantes e assume-se que todos os estudantes são independentes. Já em dados relacionais, as tuplas podem ser estudantes e cursos, as quais podem ser dependentes uma das outras.

3. Na modelagem de dados proposicionais, o número de estados diferentes é exponencial ao número de atributos, por exemplo, com N atributos binários, o número de estados é $O(2^N)$. Na modelagem de dados relacionais, o número de diferentes estados é exponencial ao produto de atributos e objetos, por exemplo, com M objetos e N atributos binários, o número de estados é $O(2^{NM})$.

4. Dados relacionais apresentam autocorrelação, o que aumenta a complexidade do aprendizado relacional. Por exemplo, *amigos*(s_1, s_2) = *verdadeiro* é uma relação onde os estudantes s_1 e s_2 são amigos. Deve haver, possivelmente, uma relação entre os gostos dos estudantes para que se tornem amigos.

5. Relações um para muitos e muitos para muitos, nas quais um objeto relaciona-se com um conjunto de objetos é uma característica de dados relacionais. Por exemplo, um estudante pode ser registrado em um grupo de cursos nos quais o *ranking* do estudante está relacionado com a *dificuldade* dos cursos.

2.4 Aprendizado semissupervisionado

A classificação semissupervisionada é uma abordagem de AM utilizada quando poucos exemplos rotulados e muitos exemplos não rotulados estão disponíveis. Pode ser considerada uma extensão da classificação supervisionada, pois o conjunto de treinamento além de conter informações dos dados rotulados X_L da forma $X_L = \{(x_1, y_1), \ldots (x_l, y_l)\}$, utiliza-se também dos dados não rotulados X_U , sendo que estes podem ser representados da forma $X_U = \{x_{l+1}, \ldots x_{l+u}\}$. Nota-se que a quantidade total de dados de treinamento é $l + u$.

O processo de classificação semissupervisionada baseada em redes divide-se em três etapas (Jebara et al., 2009): 1) inicialmente deve-se escolher uma função de similaridade que estima a afinidade entre pares de exemplos, ou seja, o peso de uma possível conexão entre dois vértices; 2) posteriormente seleciona-se um algoritmo para gerar uma rede esparsa; 3) dado a rede construída, cuja estrutura geométrica é representada pela matriz de peso, a tarefa de inferência de rótulos é difundir os rótulos conhecidos para todos os vértices não-rotulados na rede. Estes passos são ilustrados na Figura 2.6.

Figura 2.6: Passos para a aplicação de técnicas de aprendizado semissupervisionado baseada em redes.

Dependendo do problema a tratar, o aprendizado semissupervisionado, em inglês *Semi-Supervised Learning* (SSL) pode-se dividir em duas estratégias: *aprendizado indutivo* e *aprendizado transdutivo*. No aprendizado transdutivo todos os exemplos (rotulados e não rotulados) são apresentados ao algoritmo simultaneamente na fase de aprendizagem. Nessa fase o algoritmo "aprende" a classificar os exemplos não rotulados apresentados, não se supõe a apresentação de novos dados, como no aprendizado indutivo.

Formalmente, o aprendizado indutivo treina o classificador utilizando o conjunto de treinamento que pode conter tanto dados rotulados X_L, quanto dados não rotulados X_U. Este tipo de aprendizado visa aprender uma função da forma: $f : X \rightarrow Y$, de modo que f obtenha um bom desempenho na predição de rótulos em exemplos novos fora de X_U. Já o aprendizado transdutivo, tendo como entrada tanto os dados rotulados X_L, quanto os dados não rotulados X_U, visa gerar uma função da forma $f : X^{l+u} \rightarrow Y^{l+u}$, de modo que f obtenha um bom desempenho de predição de rótulos de X_U.

Ressalta-se que o foco de estudo desta tese são os algoritmos baseados em redes aplicados em SSL. Os métodos baseados em redes utilizam, em geral, uma abordagem de aprendizado transdutiva.

2.4.1 Aprendizado semissupervisionado e *manifold*

Muitos métodos em AM operam sobre dados. Esses dados encontram-se entorno de um *manifold* ($\mathcal{M}$) de baixa dimensão contido em um espaço de alta

dimensão. Por exemplo, considere a categorização de textos, os documentos são representados por vetores contendo a frequência dos termos que aparecem no mesmo. Entretanto ele ocupa apenas uma pequena porção do espaço de todas as palavras. Esse *submanifold* no qual os dados estão contidos é desconhecido, exceto por um número finito de pontos amostrados a partir de uma distribuição de probabilidade. Podemos reconstruir este *submanifold* por meio da geração de uma rede de adjacência e da aplicação de funções da matriz Laplaciana (Belkin, 2003).

No contexto de SSL, os dados não rotulados podem ser utilizados para estimar um *manifold* enquanto os dados rotulados são utilizados por um classificador sobre esse *manifold*. Considere o exemplo da Figura 2.7 (Belkin, 2003), no qual têm-se dados de duas classes sendo duas partes de uma curva (Figura 2.7(a)), alguns dados são rotulados e outros são não rotulados. Não se pode classificar o ponto "?" com confiabilidade usando apenas os pontos rotulados (Figura 2.7(b)). Considerando os pontos não rotulados percebe-se um *manifold* subjacente (Figura 2.7(c)) no qual seria preferível definir um classificador na própria curva que no plano ($\Re^2$). Existem diversas representações para o *manifold*, uma representação ideal seria uma curva fechada (Figura 2.7(d)), pois as coordenadas variam o menos possível quando se percorre a curva. Considerando uma representação bi-dimensional dos dados a partir de autovalores laplacianos, obtém-se uma incorporação da curva próxima a representação ideal (Figura 2.7(e)). Com isso o ponto "?" localiza-se no meio dos pontos "+" e pode ser facilmente classificado como tal.

Os métodos semissupervisionados também podem considerar conhecimento a priori no processo de aprendizado. Geralmente os pressupostos considerados são de agrupamento (*cluster*) e suavidade (*smoothness*). No *pressuposto de agrupamento* assume-se que os pontos que formam um mesmo grupo tendem a pertencer à mesma classe. Enquanto no *pressuposto de suavidade* define-se que os rótulos dos exemplos devem variar de uma forma suave em relação ao *manifold*.

Para reconstruir o *manifold* a partir de um conjunto de dados X e modelar o pressuposto de suavidade pode-se construir uma rede de adjacência $G(V, E, W)$. V representa um conjunto de N vertices, cada vértice $v_i \in V$ corresponde a um ponto $x_i \in X$. E representa um conjunto de M arestas, as quais representam similaridade entre vértices. W é gerada a partir de uma função de peso $w : E \rightarrow \mathbb{R}_+$, a qual associa um valor real w_{ij} para cada aresta $e_{ij} = (v_i, v_j) \in E$.

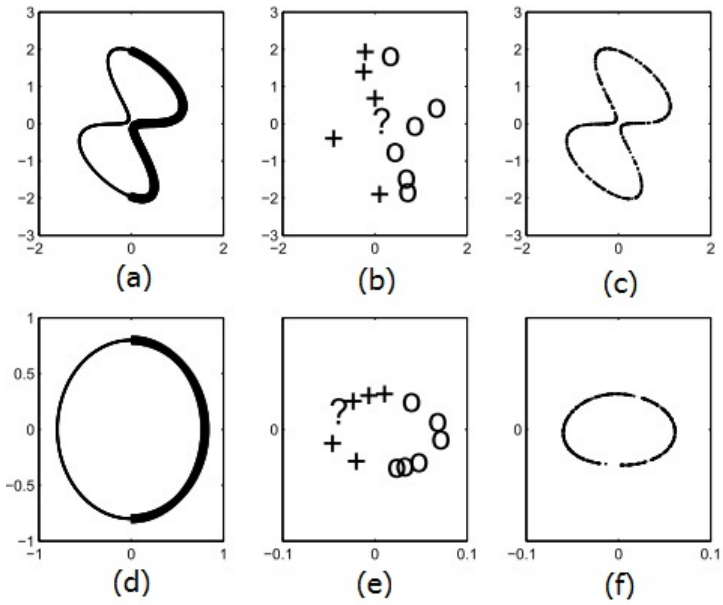


Figura 2.7: (a) Duas classes contidas em uma curva. (b) Exemplos rotulados sendo "?" um ponto a ser classificado. (c) pontos não rotulados. (d) Representação ideal da curva. (e) Posição dos dados não rotulados e "?" após aplicar funções Laplacianas. (f) Posição de todos os exemplos. Adaptada de Belkin (2003)

A suavização de uma função de classificação Y será alta se $F(Y)$ for baixa:

$$F(Y) = \sum_{ij} (y_i - y_j)^2 W_{ij} \tag{2.15}$$

Essa minimização garante que se x_i e x_j estão próximos, então y_i e y_j estão próximos também.

Para qualquer Y temos que,

$$\frac{1}{2}\sum_{ij}(y_i - y_j)^2 W_{ij} = \sum_{ij}(y_i^2 + y_j^2 - 2y_i y_j)W_{ij} = \sum_i y_i^2 D_{ii} + \sum_j y_j^2 D_{jj} - 2\sum_{ij} y_i y_j W_{ij} = 2Y^T LY \tag{2.16}$$

onde $L = D - W$ é o grafo Laplaciano e $D_{ii} = \sum_j W_{ij}$ é a matriz diagonal.

Desse modo, o problema de minimização se reduz a

$$argmin_Y Y^T LY \tag{2.17}$$

2.4.2 Algoritmos semissupervisionados baseados em redes

Esta seção apresenta as principais abordagens para classificação baseada em redes para o aprendizado semissupervisionado.

2.4.2.1 Caminhada aleatória de Markov

Os métodos baseados em caminhada aleatória consideram que a probabilidade de rotular um vértice $v_i \in V$ com rótulo $c \in Y$ é a probabilidade total que uma partícula aleatória inicie uma caminhada no vértice v_i e termine em um vértice com rótulo c.

Uma caminhada aleatória é definida por uma matriz de transição P, de modo que a partícula saia de um vértice v_i para um vértice v_j com probabiliade p_{ij}, $0 \leq p_{ij} \leq 1$ e $\sum_j p_{ij} = 1$.

Seja p_{ij}^t a probabilidade de uma partícula saindo do vértice v_i alcançar o vértice v_j após t passos. Seja P^t a matriz correspondente no tempo t. Quando $t \to \infty$, a entrada p_{ij} da matriz P^∞ representa a probabilidade da partícula iniciar a caminhada em v_i e estar no vértice v_j quando o tamanho da caminhada tende ao infinito. Isso significa que se a distribuição inicial é e_i (um vetor com valor 1 na i-ésima posição e 0 nas demais), uma partícula aleatória em G com matriz de transição P converge para uma distribuição estacionária, que é a i-ésima linha da matriz P^∞.

A probabilidade de um vértice v_i obter um rótulo $c \in Y$ é calculada como a probabilidade total de sair do vértice v_i chegar em um vértice com rótulo c, conforme a Equação 2.18.

$$\tilde{y}_i[c] = \sum_{j|v_j \in V_l} p_{ij}^\infty y_j[c] \tag{2.18}$$

Se $t \to \infty$ a probabilidade de atingir um vértice rotulado é 1. Assim, o rótulo de saída $\tilde{y}_i$ para o vértice $v_i \in V$ tem a distribuição de probabilidade dada pela Equação 2.19.

$$\sum_{c \in Y} \sum_{j|v_j \in V_l} p_{ij}^\infty y_j[c] = 1 \tag{2.19}$$

Como deve ser atribuído um único rótulo para cada vértice, o rótulo mais provável é dado pela Equação 2.20.

$$\tilde{y}_i = argmax_{c \in Y} \sum_{j;y_j=c} p_{ij}^\infty \tag{2.20}$$

A equação matricial é dada pela Equação 2.21.

$$\tilde{Y} = \mathcal{D} P^\infty Y \tag{2.21}$$

onde $\mathcal{D}$ é a matriz diagonal normalizada, definida como $\mathcal{D}_{ii} = \sum_{j=1}^n (P^\infty Y)_{ij}$.

Uma partícula aleatória em um vértice v_i seleciona uma aresta com probabilidade proporcional ao peso da aresta. Se v_i for rotulado a partícula faz um

loop nesse vértice, para que seu rótulo não mude. Ordenando os vértices de modo que os rotulados são indexados primeiro que os não rotulados, a matriz de transição pode ser escrita como:

$$P = \begin{pmatrix} P_L & P_{lu} \\ P_{ul} & P_{uu} \end{pmatrix} = \begin{pmatrix} I & 0 \\ P_{ul} & P_{uu} \end{pmatrix}$$

onde P_L corresponde as probabilidades de transição de um vértice rotulado a outro rotulado, e assim por diante. Quando uma partícula atinge um vértice rotulado para nele, assim P_L é igual a matriz identidade e P_{lu} é a matriz de zeros. Quando $lim_{t\to\infty} P^t$ temos:

$$P = \begin{pmatrix} I & 0 \\ (I - P_{uu})^{-1} P_{ul} & P_{uu}^{\infty} \end{pmatrix}$$

Os rótulos podem ser computados conforme a Equação 2.22.

$$\tilde{Y_u} = (I - P_{uu})^{-1} P_{ul} Y_l \tag{2.22}$$

Se os vértices rotulados não se tornassem *loops* a partícula aleatória convergeria para uma distribuição estacionária independente do ponto inicial. Szummer e Jaakkola (2002) consideraram uma variação dessa abordagem onde os vértices rotulados não são *loops*. Nesse caso, a caminhada aleatória é executada t passos. Desse modo, a eficiência do algoritmo depende estritamente desse parâmetro. Os autores usaram $t = 8$ nos experimentos.

2.4.2.2 Propagação de rótulos

Os algoritmos do *framework* denominado propagação de rótulos são bastante similares e propagam os rótulos de cada vértice rotulado para seus vizinhos. Os algoritmos mais conhecidos são propagação de rótulos (Zhu e Ghahramani, 2002), funções harmônicas (Zhu et al., 2003) e consistência local e global (Zhou et al., 2004).

Propagação de rótulos

O algoritmo de propagação de rótulos foi proposto por Zhu e Ghahramani (2002) e trata-se de um algoritmo iterativo, ou seja, a estrutura da rede é utilizada para gradualmente espalhar os rótulos para o restante dos vértices. A cada iteração, cada vértice espalha as informações de rótulos que possui para seus vizinhos, até que haja convergência.

A ideia geral do algoritmo é iterar a função f, fazendo $f^{(m)} = P\, f^{(m+1)}$ até que $f^{(m)} \approx f^{(m+1)}$. O Algoritmo 1 sumariza o processo de propagação de rótulos.

Algorithm 1 Propagação de rótulos

Entradas:
A matriz de rótulos, Y_l;
A matriz de pesos, W;
Saída: A matriz com todos os exemplos rotulados, Λ;

Gerar a matriz de transição de probabilidades, P;
Gerar a matriz $f^{(0)}$ inicial;
Repita
$f^{(m+1)} \leftarrow P\, f^{(m)}$;
$f_L^{(m+1)} \leftarrow Y_l$;
Até que f convirja

Funções harmônicas

O algoritmo de funções harmônicas foi proposto por Zhu et al. (2003). Os autores apresentam a seguinte definição de função harmônica:

Definição: se h é uma função harmônica, então $h(i) = \frac{1}{d_{ii}} \sum_{j \approx i} W_{ij} h(j)$.

Essa definição indica que a saída da função harmônica no vértice i é baseada na ponderação das saídas nos vértices j pelos seus respectivos pesos, sendo j um vértice vizinho de i.

Reescrevendo a definição de função harmônica, obtém-se a Equação 2.23.

$$d_{ii}\, h(i) = \sum_{i \approx j} W_{ij} h(j) \tag{2.23}$$

Escrevendo a Equação 2.23 em notação matricial, se obtém a Equação 2.24.

$$D h^{(m+1)} = W h^{(m)} \tag{2.24}$$

A partir da Equação 2.24, se pode escrever o algoritmo de funções harmônicas (ver Algoritmo 2), o qual apresenta o mesmo princípio proposto no algoritmo de propagação de rótulos.

Quando f estiver associada a uma matriz, o *framework* de regularização do algoritmo de funções harmônicas pode ser escrito conforme a Equação 2.25.

$$Q(f) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} \left\| f_i - f_j \right\|^2 \tag{2.25}$$

Algorithm 2 Funções harmônicas

Entradas:
A matriz de rótulos, Y_l;
A matriz de pesos, W;
Saída:A matriz com todos os exemplos rotulados, Λ;

Gerar a matriz diagonal, D;
Gerar a matriz $h^{(0)}$;
Repita
$Dh^{(m+1)} \leftarrow Wh^{(m)}$;
$h_L^{(m+1)} \leftarrow Y_l$;
Até que h convirja

Consistência local e global

O método *Local and Global Consistency* (LGC) introduzido por Zhou et al. (2004) representa os rótulos iniciais por uma matriz binária Y, tal que $v_i = 1$ se o vértice v_i possui o rótulo $c \in Y$.

O método pode ser visto como uma caminhada aleatória iniciando-se no vértice v_i. A partícula caminha para um vértice vizinho com probabilidade α e volta para o vértice v_i com probabilidade $1 - \alpha$. Na formulação matricial, a matriz de probabilidade de transição P^t pode ser escrita conforme a Equação 2.26.

$$P^t = \alpha T P^{t-1} + (1 - \alpha) I. \tag{2.26}$$

Esta caminhada aleatória converge para a distribuição estacionária da Equação 2.27.

$$P^\infty = (1 - \alpha)(I - \alpha T)^{-1}. \tag{2.27}$$

E a distribuição dos rótulos pode ser calculada pela Equação 2.28.

$$\tilde{Y} = P^\infty Y = (1 - \alpha)(I - \alpha T)^{-1} Y. \tag{2.28}$$

O método também pode ser implementado como um processo iterativo. A cada passo, a distribuição do rótulo de um vértice v_i é calculada como uma fração α da soma das distribuições dos rótulos de seus vizinhos, adicionada a uma fração $1 - \alpha$ de sua distribuição de rótulo inicial. Esse processo é sumarizado no Algoritmo 3. Essa formulação leva a mesma solução da caminhada aleatória.

Regularização em redes

Zhou et al. (2004) propuseram uma função que satisfaz e minimiza duas condições: i) a diferença entre os rótulos iniciais e finais deve ser pequena; e ii) a diferença entre os rótulos de vértices vizinhos deve ser pequena. Esse processo

Algorithm 3 LGC

Entradas:
A matriz de rótulos, Y_l;
A matriz de pesos, W;
Saída: A matriz com todos os exemplos rotulados, Λ;

Gerar a matriz diagonal, D;
Gerar a matriz $L = D^{1/2}WD^{1/2}$;
$f(0) \leftarrow Y$;
Repita
$f(t+1) \leftarrow \alpha Lf(t) + (1-\alpha)Y$, com $\alpha \in (0,1)$;
Até que f convirja
Rotular os exemplos x_i com o rótulo $c = argmax_{1 \leq j \leq c} f_{ij}$, sendo f_{ij} a solução do algoritmo.

de otimização é conhecido como regularização, definido pela Equação 2.29.

$$\tilde{f} = argmin_f \frac{\mu}{2} \|f - Y\|^2 + f^T L f \tag{2.29}$$

O primeiro termo mede a diferença entre o rótulo dado pela função de rotulagem e o rótulo original Y. O segundo termo usa a matriz laplaciana L para medir a suavidade da função de rotulagem. O parâmetro μ controla o balanço entre esses dois termos. Reescrevendo essa minimização em termos da norma Euclidiana obtém-se a Equação 2.30.

$$min_f \frac{1}{2} \sum_{i,j=1}^{n} W_{ij} \left\| \frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right\|^2 + \frac{\mu}{2} \|f - Y\|^2 \tag{2.30}$$

Diferenciando e igualando a zero obtém-se a Equação 2.31.

$$\tilde{f} - D^{-1/2}WD^{-1/2}\tilde{f} + \mu(\tilde{f} - Y) = 0(1+\mu)\tilde{f} - P\tilde{f} = \mu Y \tag{2.31}$$

Resolvendo $\tilde{f}$ e considerando $\alpha = \frac{1}{1+\mu}$, tem-se a Equação 2.32, a qual converge para a Equação 2.28.

$$\tilde{f} = (1-\alpha)(I - \alpha P)^{-1} Y \tag{2.32}$$

2.4.2.3 Agrupamento

Algumas abordagens para classificação em redes particionam a rede em grupos e atribuem o mesmo rótulo a todos os vértices de um dado grupo. Blum e Chawla (2001) propuseram o uso de corte mínimo para classificação binária, onde a rede é particionada de modo a minimizar o número de arestas cruzando um corte. Assume-se que vértices altamente conectados pertencem a mesma classe. O Algoritmo 4 é baseado em fluxo máximo e pode ser resolvido em tempo

polinomial.

Algorithm 4 Corte mínimo

Entradas:
O conjunto de rótulos, (v_+, v_-);
A matriz de pesos, W;
Saída: A matriz com todos os exemplos rotulados, Λ;

Atribuir $w(v, v_+) = \infty \forall v \in V_{l+}$ e $w(v, v_-) = \infty \forall v \in V_{l-}$;
Determinar corte mínimo entre (v_+, v_-), usando algoritmo de fluxo máximo, no qual v_+ é a fonte e v_- é o depósito e os pesos são as capacidades;
Atribuir rótulo positivo aos vértices no conjunto V_+ e rótulo negativo aos vértices no conjunto V_-;

2.5 Métodos para construção de redes

Diversas áreas de aprendizado de máquina (AM) utilizam redes para modelar relações locais entre dados. Aplicações incluem *clustering* espectral, redução de dimensionalidade, aprendizado semissupervisionado, entre outras. Entretanto, Maier et al. (2007a) enfatizam que uma das principais questões relacionadas com o aprendizado baseado em redes trata de qual técnica de construção de redes utilizar para cada aplicação e como escolher seus parâmetros. Zhu (2005) também destacam que muitos algoritmos em SSL são bastante parecidos, sendo mais importante a escolha do método de construção de redes do que do algoritmo de aprendizado.

Dependendo do paradigma de AM, alguns modelos de redes podem ser mais adequados que outros. Para o aprendizado supervisionado, as redes construídas, em geral, têm como restrição não estabelecer conexões entre elementos da mesma classe. Para o aprendizado não-supervisionado deseja-se que o número de conexões intra-grupos seja maior que o número de conexões entre-grupos. Nesta seção são apresentados os principais métodos para construção de redes considerados no aprendizado semissupervisionado.

2.5.1 Rede *minimum/maximum spanning tree* (Min/MaxST)

Uma rede acíclica conexa que contém todos os vértices de G é denominada *spanning tree* de G. Uma *spanning tree* possui exatamente $n - 1$ arestas. Se as arestas possuem peso, a *spanning tree* com menor custo total é chamada *Minimum Spanning Tree* (MinST) e com maior custo total é uma *Maximum Spanning Tree* (MaxST).

Para encontrar uma Min/MaxST pode-se utilizar os algoritmos de Kruskal ou de Prim, os quais consideram uma abordagem gulosa e têm custo de execução $O(M \log N)$ (Cormen et al., 2009). Para a construção de uma rede Min/MaxST em SSL, a Min/MaxST é encontrada a partir de uma rede ponderada completa. Neste caso a complexidade é $O(N^2 \log N)$. Um exemplo de redes MinST e MaxST é apresentado na Figura 2.8. Observa-se que as redes são bastante esparsas e podem ser utilizadas combinadas a outros métodos de construção de redes, conforme o trabalho de Ozaki et al. (2011).

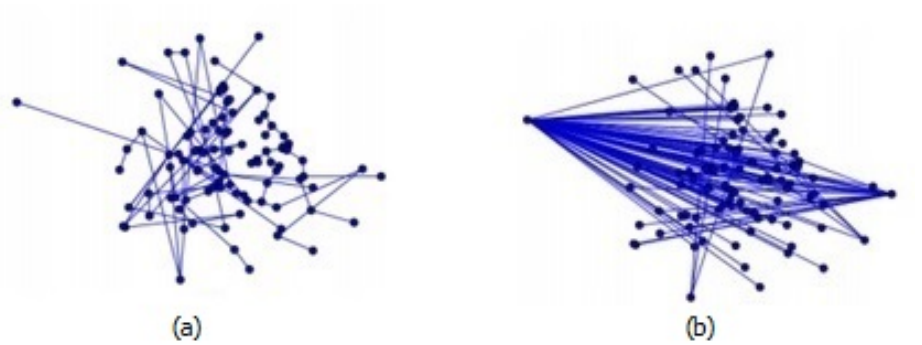


Figura 2.8: Exemplo de uma rede MinST (a) e de uma rede MaxST (b) para um conjunto de dados com 100 elementos e distribuição gaussiana.

2.5.2 Rede ϵ-vizinhança

Na rede ϵ-vizinhança cria-se uma aresta entre os elementos x_i e x_j se $|(x_i, x_j)| \leq \epsilon$, sendo $|.|$ uma função de distância e ϵ um limiar pré-definido. Ao se utilizar esta técnica, se ϵ for um valor muito baixo, deve-se fazer um pós-processamento para que não haja vértices isolados na rede, já se ϵ for um valor muito alto, a rede gerada pode ser muito densa.

A fim de exemplificação, considere a Figura 2.9, que mostra redes geradas para diferentes valores de ϵ. Observa-se que ao mesmo tempo em que a rede da Figura 2.9(b) é desconexa e possui vértices isolados, alguns vértices possuem grau bastante alto. Se aumentarmos mais o valor de ϵ para que a rede torne-se conexa, conforme a Figura 2.9(c) estes vértices terão um grau ainda maior e a rede torna-se muito densa.

Devido a essa desvantagem, as redes ϵ-vizinhança têm sido pouco utilizadas. Belkin e Niyogi (2003) destacam que as redes ϵ-vizinhança resultam facilmente em componentes desconexos, não sendo adequadas em várias situações. Jebara et al. (2009) constroem redes sintéticas e mostram que as redes ϵ-vizinhança tendem a gerar vários componentes desconexos, principalmente se os dados apresentam variações de densidade.

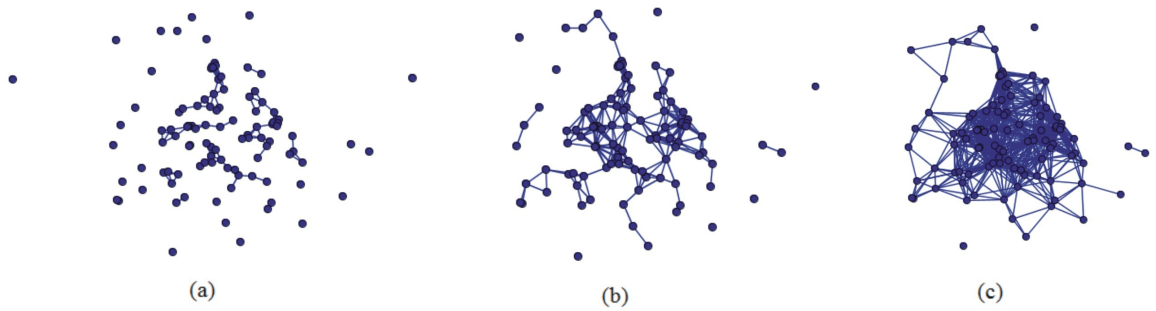


Figura 2.9: Exemplo de redes ϵ-vizinhança para um conjunto de dados com 100 elementos e distribuição gaussiana (a) $\epsilon = 0.3$, (b) $\epsilon = 0.5$ e (c) $\epsilon = 0.9$.

2.5.3 Rede k-vizinhos mais próximos (kNN)

O algoritmo k-vizinhos mais próximos (kNN) foi proposto para o aprendizado supervisionado. Neste contexto, dado um novo exemplo x_i que deve ser rotulado, obtêm-se os k vizinhos mais similares a x_i, representados por $k\text{NN}(x_i) = \{(x_1, y_1), \ldots, (x_k, y_k)\}$ atribui-se a x_i o rótulo majoritário em $\{y_i, \ldots, y_k\}$ do conjunto $k\text{NN}(x_i)$.

Observa-se que os limites de decisão do algoritmo podem mudar com o valor de k, sendo que quanto menor o valor de k, mais detalhada é a fronteria de decisão, e quanto maior é o valor de k, mais suave e uniforme se torna a fronteira, com tendência a gerar tantos subgrupos quantas classes existirem. Esta característica pode ser observada na Figura 2.10. Na Figura 2.10(a) $k = 1$ e diversos subgrupos isolados são gerados. Na Figura 2.10(b) $k = 5$ e a superfície de decisão torna-se mais uniforme, não importando se alguns exemplos do conjunto de treinamento são incorretamente considerados como pertencentes a outra classe. Os quais passam a ser considerados como *outliers* ou ruído nas classes.

Já para a geração de redes, o algoritmo kNN escolhe inicialmente k vizinhos para cada instância, com base na distância ou na afinidade entre instâncias, e gera uma conexão entre os vértices v_i e v_j, se a distância entre v_i e v_j for menor ou igual a k-ésima distância a partir de v_i e os demais objetos em V. Essa abordagem é conhecida por redes *kNN simétricas*. Outra possibilidade são as redes *kNN mútuas* (M-kNN), na qual um vértice estabelece uma ligação somente se a regra de vizinhança se cumprir mutuamente, ou seja, existirá uma aresta entre v_i e v_j, se e somente se, $v_i \in N(v_j) \wedge v_j \in N(v_i)$. Por isso, as redes kNN mútuas são consideradas mais restritivas, sendo utilizadas tradicionalmente no aprendizado não supervisionado (Gowda e Krishna, 1978), (Brito et al., 1997). Atualmente, a nomenclatura redes kNN se referem as redes kNN simétricas, pois são as mais utilizadas no processo de geração de redes por similaridade. Eppstein et al. (1997) apresentam mais detalhes matemático-geométrico da formação das redes

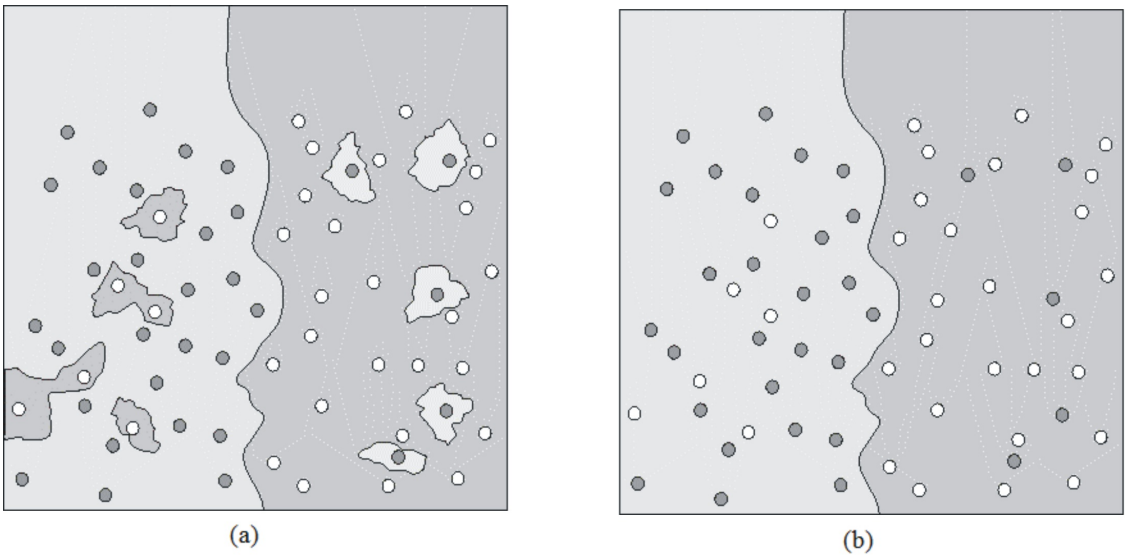


Figura 2.10: Limites de decisão gerados pelo algoritmo k-vizinhos mais próximos, (a) $k = 1$ e (b) $k = 5$.

kNN. O tempo computacional para gerar uma rede kNN é $O(dN^2)$, onde d é o número de dimensões e N o número de elementos.

A Figura 2.11 ilustra o processo de formação das redes kNN simétricas. Nota-se que neste caso, um valor baixo de k gera uma rede desconexa e conforme o valor de k é incrementado, a densidade de arestas na rede aumenta. Além disso, a rede kNN força um vértice a se conectar com seus k vizinhos mais próximos, mesmo se eles estiverem distantes. Dependendo do caso, a vizinhança desse vértice pode conter exemplos dissimilares.

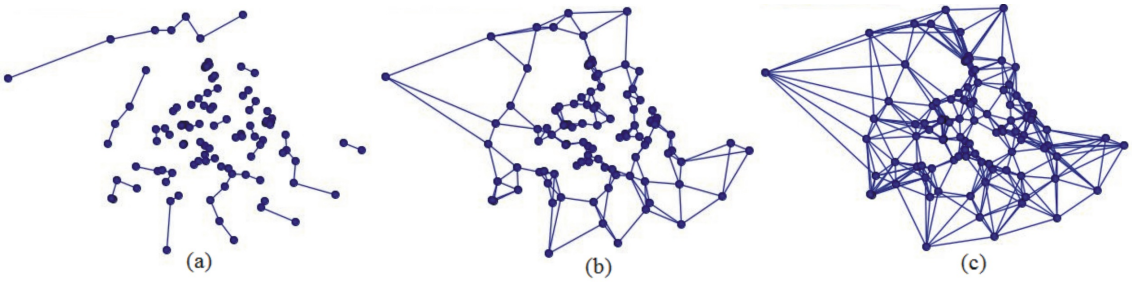


Figura 2.11: Exemplo de redes kNN para um conjunto de dados com 100 elementos e distribuição gaussiana (a) $k = 1$, (b) $k = 3$ e (c) $k = 7$.

Maier et al. (2007a) estudaram e compararam as redes kNN mútuas e simétricas aplicadas no aprendizado não-supervisionado. Os autores provam que para diferentes valores de k, grupos (*clusters*) podem ser identificados. Além disso, observam que as redes kNN mútuas favorecem a identificação de grupos, em particular, se os grupos possuem diferentes densidades e pesos. Já as redes kNN simétricas são mais apropriadas para identificar vários grupos simultâneamente, por terem melhores propriedades de conectividade.

2.5.4 Rede *b-matching*

O algoritmo kNN inicia com uma matriz $\hat{A}$ com todos os valores iguais a 0. Para cada instância busca-se os k vizinhos mais próximos (sem considerar o próprio ponto). Se um exemplo v_j for um dos k vizinhos mais próximos de v_i, então $\hat{A}_{ij} = 1$. Neste caso, como uma instância v_i possui vizinhos diferentes de uma instância v_j, a seguinte situação pode ocorrer com frequência: v_i inclui v_j como seu k vizinho, mas v_j não inclui v_i em seus k vizinhos. Com isso, uma rede kNN pode conter mais do que k arestas por vértice e sua representação matricial não fornece uma matriz simétrica.

A matriz pode ser simetrizada por $A_{ij} = max(\hat{A}_{ij}, \hat{A}_{ji})$. Neste caso, o algoritmo dos k-vizinhos mais próximos resolve o problema de otimização dado pela Equação 2.33. Depois produz a solução $A = max(\hat{A}, \hat{A^T})$.

$$min_{\hat{A} \in R} \sum_{i,j} \hat{A}_{ij} \, F_{ij} \tag{2.33}$$

$$s.a. \sum_{j} \hat{A}_{ij} = k, \;\; \hat{A}_{ii} = 0, \;\; \forall i, j \in 1, \cdots, n$$

onde $F_{ij} = \sqrt{A_{ii} + A_{jj} - 2A_{ij}}$.

Jebara et al. (2009) destacam que este algoritmo guloso não está resolvendo um problema bem definido de otimização de matrizes binárias. Como ele produz uma matriz simétrica apenas com uma otimização *ad hoc* sobre $\hat{A}$ e sua transposta, a solução para A não satisfaz a igualdade $\sum_k A_{ij} = k$, apenas satisfaz a desigualdade $\sum_j A_{ij} \geq k$. Como o algoritmo kNN produz uma rede não-direcionada com mais de k vizinhos para cada vértice, os autores propuseram o algoritmo *b-matching*, com uma versão gulosa, denominada *Greedy*, que produz uma matriz simétrica. O algoritmo encontra exatamente b vizinhos para cada vértice, em que b é um parâmetro inicial, pois o objetivo dessa técnica é gerar redes regulares, onde todos os vértices possuam o mesmo grau.

A técnica b-matching busca encontrar a matriz binária que minimiza o problema de otimização dado pela Equação 2.34, a qual encontra a matriz simétrica diretamente, sem a necessidade de pós-processamento.

$$min_{A \in R} \sum_{i,j} A_{ij} \, F_{ij} \tag{2.34}$$

$$s.a. \sum_{j} A_{ij} = b, \;\; A_{ii} = 0, \;\; A_{ij} = A_{ji}, \;\; \forall i, j \in 1, \cdots, n$$

A matriz simétrica é encontrada devido a restrição adicional $A_{ij} = A_{ji}$. Assim, a matriz satisfaz estritamente a igualdade $\sum_j A_{ij} = \sum_i A_{ji} = b$. Uma implemen-

tação desta técnica foi desenvolvida utilizando-se um algoritmo de propagação que garante a convergência para uma solução global em tempo $O(bN^3)$, sendo N o número de elementos (Huang e Jebara, 2007).

Na Figura 2.12, observa-se que b-matching é regular, ou seja, cada vértice na rede possui apenas b arestas conectando-o aos demais vértices. Os autores declaram que isso é uma vantagem na propagação de rótulos.

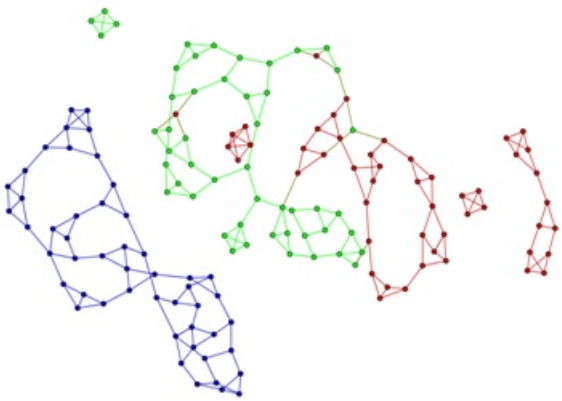


Figura 2.12: Exemplo de rede b-matching para a base de dados Iris, $b = 3$.

Os autores aplicam a técnica b-matching para esparsificação e para o aprendizado semissupervisionado. Nos resultados relatados a técnica apresenta melhor desempenho em comparação com o algoritmo kNN para a construção de redes. Esse estudo confirma a importância do estudo de métodos de construção de redes no aprendizado semissupervisionado.

M. Bayati (2005) formulam o problema de 1-matching como uma distribuição de probabilidade discreta e o resolvem por meio de *belief propagation* em $O(n^3)$. Huang e Jebara (2007) estendem o problema para b-matching e fornecem uma prova de convergência em $O(N^3)$.

Os autores apresentam o problema para uma rede bipartida. b-matching retorna um conjunto de vizinhos $N(v_i)$ e $N(v_j)$ para os vértices v_i e v_j respectivamente. O problema combinatorial pode ser escrito conforme a Equação 2.35.

$$max_N W(N) = max_N \sum_{i=1}^{|V|} \sum_{v_k \in N(v_i)} A_{ik} + \sum_{j=1}^{|V|} \sum_{v_l \in N(v_j)} A_{lj} \tag{2.35}$$

Para cada vértice, definem-se duas variáveis aleatórias $z_i \in Z$ e $s_j \in S$, $z_i = N(v_i)$ e $s_j = N(v_j)$. Obtendo-se as seguintes funções dadas pela Equação 2.36.

$$\Phi(z_i) = exp(\sum_{v_j \in z_i} A_{ij});\ \Phi(s_j) = exp(\sum_{v_i \in s_j} A_{ij});\ \Psi(z_i, s_j) = \neg(v_j \in z_i \otimes v_i \in s_j) \tag{2.36}$$

Multiplicando as funções acima, a função objetivo para o problema de b-matching pode ser formulada como uma distribuição de probabilidades via $p(Z, S) \propto exp(W(N))$, onde a distribuição conjunta pode ser expressada conforme a Equação 2.37.

$$p(Z, S) = \frac{1}{Z} \prod_{i=1}^{|V|} \prod_{j=1}^{|V|} \Psi(z_i, s_j) \prod_{k=1}^{|V|} \Phi(z_i)\Phi(s_j) \tag{2.37}$$

2.6 Considerações finais

Muitos trabalhos em AM focam em dados proposicionais, que são IID. Entretanto, a maioria dos dados reais são relacionais e transformar dados proposicionais para relacionais pode melhorar a acurácia na classificação. A estrutura dos dados relacionais permite aos objetos levar informações adicionais por meio de suas conexões e possibilita também que o modelo mostre correlações entre os objetos e seus relacionamentos. Além disso, é possível modelar o *manifold* em que os dados estão contidos por meio de uma rede baseada em vizinhança.

Apesar dos métodos desenvolvidos para dados relacionais terem bons resultados, ainda há a necessidade de mais pesquisas nesta área, como por exemplo, analisar e propor métodos para a construção de redes. Alguns autores apontam para o fato da estrutura da rede ser tão importante quanto as variações dos algoritmos para classificação. Os métodos para construção de redes frequentemente utilizados são o kNN, que pode gerar redes muito densas e com muita variação no grau dos vértices, e o método ϵ-vizinhança, que possui como principal desvantagem a adequação do parâmetro ϵ. Novos métodos para construção de redes têm sido propostos, porém, esta área continua com muitas questões abertas.

Esta tese foca no estudo de técnicas para construção de redes para o aprendizado semissupervisionado e propõe novos métodos explorando diferentes topologias. Essas propostas são apresentadas em detalhes no próximo capítulo.

CAPÍTULO

3

Proposta: métodos para construção de redes para o aprendizado semissupervisionado

Muitos dados reais possuem uma estrutura natural em formato de rede, a qual descreve relações de similaridades entre os elementos, por exemplo redes sociais, redes de informações, redes biológicas e redes tecnológicas. Entretanto, também existem dados IID, no formato de tabelas atributo-valor. Para aplicar algoritmos baseados em redes nesses dados, faz-se necessário a construção de uma rede.

A representação em rede tem diversas vantagens, como evidenciar correlações entre os objetos e seus relacionamentos, representar estruturas hierárquicas, modelar *manifolds* no espaço multi-dimensional e facilitar a propagação de poucos rótulos iniciais para os vizinhos por meio das arestas. Existem diversas aplicações considerando métodos baseados em grafos como em classificação de proteínas (Tsuda et al., 2005), agrupamento de dados (Maier et al., 2007a), recomendação de vídeos (Baluja et al., 2008), além disso, os métodos baseados em grafos possuem forte embasamento teórico (Belkin et al., 2006; Hein et al., 2005).

Entre os principais trabalhos baseados em redes que utilizam aprendizado semissupervisionado destacam-se os que empregam corte mínimo (Blum e Chawla, 2001); os que fazem a propagação de rótulos (Zhu e Ghahramani, 2002),

(Bengio et al., 2006); os baseados na consistência local e global (Zhou et al., 2004); os baseados em caminhada aleatória de Markov (Szummer e Jaakkola, 2002) e os que utilizam funções harmônicas (Zhu et al., 2003). Entretanto, a maioria dos métodos da literatura foca no desenvolvimento de algoritmos para a inferência de rótulos sem se preocupar com a construção da rede e seu efeito no desempenho do algoritmo.

O método mais comum para a construção de redes é o k-vizinhos mais próximos (kNN). Nesse método os vértices terão grau maior ou igual a k e a seleção do parâmetro k influencia a geração da rede e consequentemente a tarefa de aprendizado. A importância da construção da rede passou a ser levantada por autores como Maier et al. (2009), Wang e Zhang (2008), Jebara et al. (2009). Porém, essa área de pesquisa ainda continua com muitas questões abertas. Quais propriedades das redes são atraentes para o aprendizado de máquina? Os algoritmos de classificação podem se comportar de maneira diferente em diferentes tipos de redes? Os algoritmos de construção de redes realmente modelam a estrutura subjacente dos dados (*submanifold*)?

Destaca-se ainda a dificuldade de guardar em memória e processar uma matriz $N \times N$, no caso de aplicações considerando grandes bases de dados, como por exemplo, no reconhecimento de objetos, processamento de linguagem natural e recuperação de informação multimídia. Assim, desenvolver métodos otimizados para a construção e o aprendizado baseado em redes ainda é um desafio.

Nesta tese são propostas três abordagens para a construção de redes a partir de dados em formato atributo-valor. As propostas investigam diferentes topologias: redes regulares, redes livre de escala e redes mundo pequeno. Buscou-se verificar se a topologia das redes e propriedades relacionadas ao grau dos vértices influenciam a classificação semissupervisionada.

A Seção 3.1 apresenta o método *Sequential* kNN (S-kNN) (Vega-Oliveros et al., 2014) que visa a construção de redes regulares. Essa proposta é uma alternativa ao método b-matching (Jebara et al., 2009) apresentando menor tempo computacional para a construção das redes.

A Seção 3.2 apresenta os métodos *Graph Based on the Informativeness of Labeled Instances* (GBILI) (Berton e de Andrade Lopes, 2012, 2014) e *Robust Graph that Considers Labeled Instances* (RGCLI), que fazem uso dos rótulos disponíveis no SSL para construção das redes, tornam os vértices rotulados em *hubs* e geram redes com distribuição de grau livre de escala.

A Seção 3.3 apresenta o método *Graph Based on Link Prediction* (GBLP) (Berton et al., 2015), que faz uso de medidas de predição de *links* para adicionar arestas em uma rede esparsa inicial. Obtém-se assim uma rede com proprie-

dade mundo-pequeno, com alto coeficiente de *clustering* e a média dos menores caminhos relativamente baixa.

3.1 Construção de redes regulares

O método k-vizinhos mais próximos (kNN), conforme já mencionado, pode retornar redes nas quais os vértices possuem mais de k vizinhos. No contexto supervisionado, a classificação baseada em vizinhos mais próximos não funciona adequadamente em espaços multi-dimensionais. Radovanovic et al. (2010) argumentam que isso ocorre devido à existência de pontos próximos a muitos outros pontos no espaço, esses pontos são denominados *hubs*. Ozaki et al. (2011) estendem esse argumento e observam que um *hub* no espaço dos dados resultará em um *hub* na rede kNN, o qual pode degenerar a acurácia da classificação. Os autores propõem a utilização do método kNN mútuo (M-kNN) combinado com *maximum spanning tree* (MaxST), gerando assim redes com menos *hubs* para aplicar na classificação de linguagem natural. Em outro contexto, Fouss e Parijs (2014) e Knees et al. (2014) melhoram a acurácia de sistemas de recomendação baseados em vizinhos mais próximos reduzindo a importância dos *hubs*.

A fim de gerar redes regulares, Jebara et al. (2009) propuseram a utilização de uma estratégia denominada b-matching, que garante que todos os vértices possuam b vizinhos. Outra versão do método b-matching chama-se Greedy e utiliza uma abordagem gulosa para estabelecer conexões. Li e Fu (2013) também aplicaram b-matching e *low-rank* para a classificação semissupervisionada. Os autores destacam que uma rede regular obtém melhores resultados. Entretanto, a construção de uma rede b-matching é custosa computacionalmente, sendo inviável em grandes bases de dados (Ozaki et al., 2011).

Como uma alternativa ao método b-matching, propôs-se um novo método para a construção de redes quase regulares, isto é, a rede não é totalmente regular mas aproxima-se de uma. O método denominado *Sequential* kNN (S-kNN) (Vega-Oliveros et al., 2014) apresenta tempo computacional menor que o método b-matching.

3.1.1 *Sequential* kNN (S-kNN)

O método *Sequential* k *Nearest Neighbors* (S-kNN) (Vega-Oliveros et al., 2014) consiste na inclusão de arestas que conectam vizinhos mais próximos, iniciando em $k = 1$ até um valor máximo pré-definido k_{max}. Neste processo, os vértices são selecionados para estabelecerem uma conexão por um critério de relevância. O critério escolhido ordena os vértices de acordo com a medida *closeness'*

C'_l. No algoritmo proposto, o critério para estabelecer as conexões considera o inverso de C_l, desse modo, o vértice com menor valor de *closeness'* estabelece uma conexão em primeiro lugar. Esse vértice se conecta inicialmente com seu vizinho mais próximo. Em seguida o vértice com segundo menor valor de *closeness* estabelece uma conexão, e assim por diante. As conexões são estabelecidas sequencialmente, considerando os vizinhos disponíveis. Um vizinho disponível é aquele entre os k_{max} vizinhos de v_i cujo grau é menor que o valor corrente de k. Se não existir nenhum vizinho disponível o vértice fica sem ligação nesta rodada, k é incrementado e o processo é repetido.

Para o critério de conexões usa-se o inverso da medida *closeness'*, assim os elementos mais distantes estabelecem conexões antes dos elementos mais centrais. Com isso, facilita-se a escolha de vizinhos, pois os elementos centrais têm muitos vizinhos mais próximos, ao contrário dos elementos distantes. Por exemplo no *lattice* da Figura 3.1, os vértices nas esquinas estabelecem conexões primeiro que os vértices centrais, conforme o valor de k aumenta.

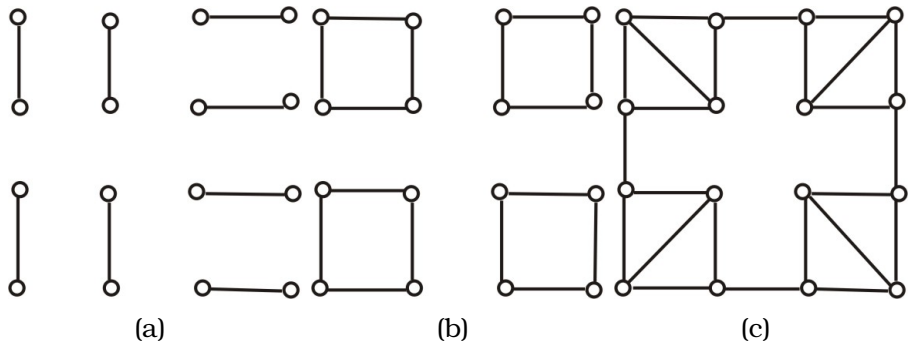


Figura 3.1: Redes S-kNN em um lattice: (a) $k = 1$ (b) $k = 2$ (c) $k = 3$.

O Algoritmo 5 descreve os passos para a construção da rede S-kNN. Na linha 1 são calculados os k_{max} vizinhos mais próximos de cada elemento e na linha 2 os elementos são ordenados pelo critério de relevância (*closeness'*). Nas linhas 4 a 14 as conexões são estabelecidas do seguinte modo: na linha 6, seleciona-se um vértice v_i do vetor ordenado segundo a medida *closeness'* e na linha 7 seleciona-se um vizinho v_j dentre os kNN de v_i que possua grau menor que k. Se existir um vizinho disponível, na linha 9 gera-se uma conexão entre v_i e v_j. Após tentar se estabelecer conexões para todos os elementos considerando $k = 1$, incrementa-se k na linha 11 e repete-se o processo. O algoritmo termina quando k for igual a k_{max}. Se após essa etapa, ainda existirem vértices com grau menor a k_{max}, a condição é relaxada e esses vértices se conectam a um k vizinho mais próximo com menor grau.

Algorithm 5 S-kNN

entrada: base de dados X, parâmetro k_{max} **saída:** G

1: kNN $\leftarrow$ get$k$-NearestNeighbor(X, k_{max});
2: $C_l \leftarrow$ getCloseness(X);
3: v_i, v_j, $k = 1$;
4: **repeat**
5: **for**($i = 1, i < N, i++$) {
6: $v_i \leftarrow$ getNextNode(C_l);
7: $v_j \leftarrow$ getNearestPossibleNeighbor(v_i, k, k_{max}, kNN);
8: **if** $v_j \neq$ NULL
9: $G \leftarrow$ conect(v_i, v_j);
10: }
11: $k \leftarrow k + 1$;
12: **until** $k == k_{max}$;
13: **if** ($\exists\ v_i \mid g_{v_i} < k_{max}$)
14: $G \leftarrow$ conect(v_i, kNN);

3.1.2 Análise de complexidade

Primeiro encontra-se os k vizinhos de cada vértice. Para calcular o kNN para um elemento o algoritmo gasta $kn - \frac{k(k-1)}{2}$ passos, onde n é o número de elementos e k é o número de vizinhos considerado. Para encontrar o kNN para todos os elementos, leva-se $kn^2 - \frac{kn(k-1)}{2}$ passos.

Segundo calcula-se a medida *closeness*, a qual é baseada em uma medida de distância ou similaridade entre pares de vértices. Considerando-se a distância Euclidiana leva-se $\frac{n(n-1)}{2}$ passos, onde n é o número de elementos.

Por fim, seleciona-se os elementos por ordem da medida *closeness* com custo $n \log n$ e cria-se uma conexão sequencialmente com seus k vizinhos mais proximos, até que todos os elementos tenham k_{max} conexões. Com isso leva-se nk passos.

A complexidade final é $O(kn^2 - \frac{kn(k-1)}{2} + \frac{n(n-1)}{2} + n \log n + nk)$, resultando em quadrática.

3.1.3 Visualização da rede S-kNN

O gráfico 3.2 mostra a distribuição do grau para as redes S-kNN e kNN construídas para a base de dados Breast-Cancer do repositório UCI (Bache e Lichman, 2013), considerando $k = 7$. Nota-se que o método S-kNN apresenta quase todos os vértices com grau igual a 7 e menos de 250 vértices possuem grau 8, 9 ou 10. No total são geradas 2597 arestas e o grau médio da rede é 7.6. Por outro lado, o método kNN apresenta vértices com diferentes graus, onde menos de 200 vértices possuem grau igual a 7. O restante dos vértices possuem grau variando de 8 até 36. No total são geradas 3548 arestas e o grau médio da rede é 10.4.

A Figura 3.3 mostra uma rede construída para a base de dados Glass do repositório UCI considerando $k = 5$. Quanto maior os pontos, maior o grau do vértice. Neste caso, a rede kNN apresenta pontos maiores que a rede S-kNN, confirmando que o método proposto gera redes mais regulares.

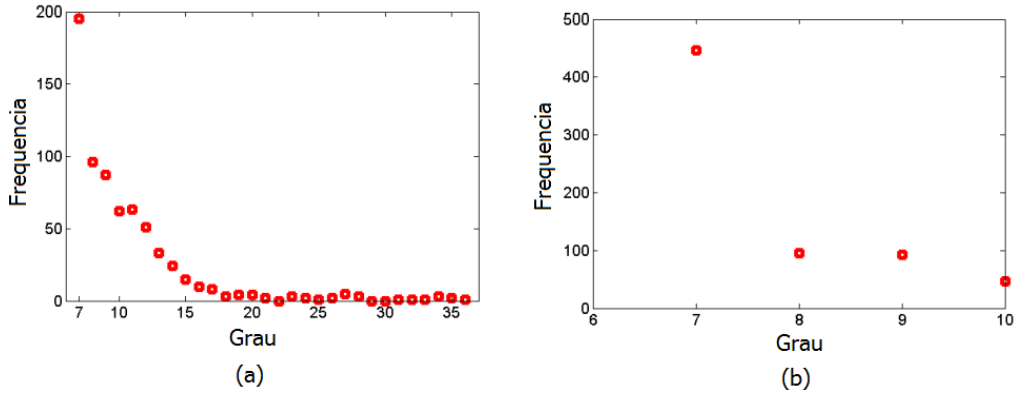


Figura 3.2: Distribuição de grau para as redes geradas a partir dos métodos kNN (a) e S-kNN (b) para a base de dados Breast-Cancer, $k = 7$. Obtida em (Vega-Oliveros et al., 2014).

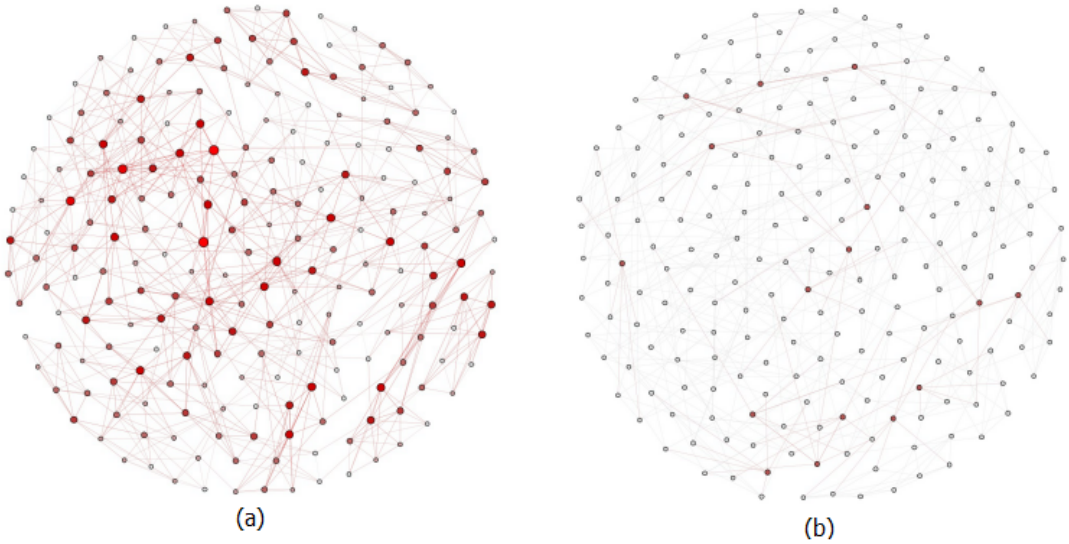


Figura 3.3: Redes kNN (a) e S-kNN (b) geradas para a base de dados Glass, $k = 5$. O tamanho dos vértices representa a magnitude do grau. Obtida em (Vega-Oliveros et al., 2014).

3.2 Construção de redes a partir das instâncias rotuladas *a priori* em SSL

A maioria dos métodos para construção de redes em SSL são não supervisionados, isto é, eles não utilizam a informação dos rótulos disponíveis no processo de construção da rede. Dhillon et al. (2010) abordam esta questão e propuseram uma medida de distância para computar similaridade entre pares de instâncias considerando os rótulos. Os dados rotulados podem ser considerados como uma informação *a priori*, que também pode ser útil para melhorar a construção da rede para a tarefa de classificação. Com base nisso, propôs-se um método para construção de redes que faz uso dos dados rotulados disponíveis no aprendizado semissupervisionado (Berton e de Andrade Lopes, 2012) (Berton e de Andrade Lopes, 2014). Este método é denominado *Graph Based on the Informativeness of Labeled Instances* (GBILI).

3.2.1 *Graph Based on the Informativeness of Labeled Instances* (GBILI)

Normalmente, os pontos de um conjunto de dados não estão isolados no espaço de atributos, mas tendem a formar certos padrões. Por exemplo, a Figura 3.4 (a) ilustra um exemplo artificial contendo dois grupos de pontos que formam dois quadrados e outro grupo de pontos que forma uma linha. Considerando-se os padrões desse exemplo visualmente, um humano observaria três grupos. Entretanto, métodos de aprendizado baseados em distância geralmente apresentam dificuldades neste reconhecimento de padrões. O método proposto para construção de redes busca reconhecer certos padrões especificamente relacionados com variações na densidade dos dados. Para isso, faz uso do método kNN mútuo, já que este método evita conectar subredes de grupos diferentes. Na Figura 3.4 (a) empregou-se o método proposto e ele identifica três grupos de pontos, por outro lado, na Figura 3.4 (b e c) empregou-se o método kNN e este não identifica três grupos de pontos.

O método também explora informações disponíveis *a priori* (neste caso os vértices rotulados) para a construção da rede, priorizando conexões entre vértices que estão mais próximos de um vértice rotulado. Isso torna os vértices rotulados em *hubs*, especialmente quando o valor de k aumenta (Figura 3.5).

Como este método considera informação da vizinhança de um vértice e dos vértices rotulados para estabelecer conexões, ele é denominado *Graph Based on the Informativeness of Labeled Instances* (GBILI). O método, para cada vértice,

escolhe o ponto "mais informativo" para estabelecer uma conexão otimizando a Equação 3.1.

$$min \sum_i \sum_j \left(\mathcal{D}_{i,j} + \sum_l \mathcal{D}_{j,l} \right) \tag{3.1}$$

$$s.a.\ \mathcal{D}_{ij} \geq 0,\ i \in \{1 \ldots n\},\ j \in k\text{NN mútuo de } i, l \in L$$

onde kNN mútuo é o conjunto de vizinhos mútuos de um vértice, L é o conjunto de vértices rotulados *a priori* e $\mathcal{D}$ é a matriz de distância.

No algoritmo 6 são apresentados os passos para construir a rede GL. Inicialmente, é necessário gerar a matriz de distância $\mathcal{D}$, para tal a distância Euclidiana pode ser empregada. Sobre esta matriz pode-se encontrar os k vizinhos mais próximos dos elementos. É necessário também inicializar o parâmetro K com um número natural e gerar uma lista dos pontos rotulados L.

No algoritmo 6, as linhas 3 a 7 encontram os k vizinhos mais próximos para um vértice v_i, depois, os kNN vizinhos mútuos para v_i e a distância de cada elemento para um vértice rotulado mais próximo. As linhas 8 a 11 calculam a soma das distâncias do vértice v_i para cada elemento da lista de vizinhos mútuos, e a partir destes elementos para um vértice rotulado. É estabelecida uma conexão entre os vértices v_i e v_j que minimiza esta soma. As linhas 12 a 19 executam um pós-processamento na rede conectando componentes isolados. Este passo é importante pois o kNN mútuo, especialmente para valores baixos de k, gera muitos

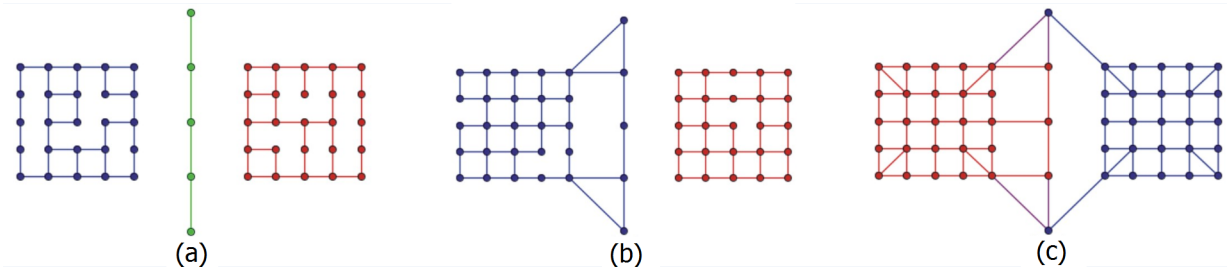


Figura 3.4: Resultado da classificação utilizando: (a) o método proposto; (b) e (c) kNN para a construção das redes. (a) $k = 2$, (b) $k = 2$, (c) $k = 3$. Obtida em (Berton e de Andrade Lopes, 2014).

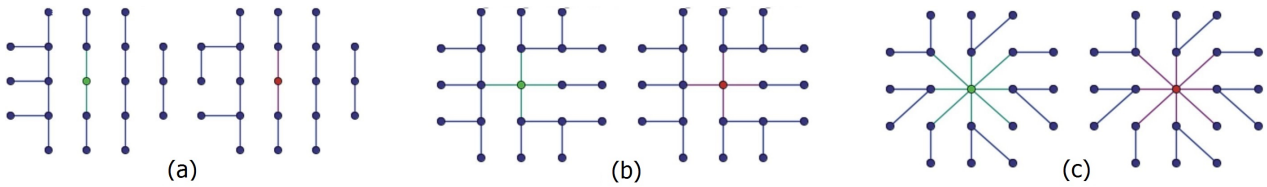


Figura 3.5: Construção da rede GBILI (a) $k = 3$, (b) $k = 7$, (c) $k = 10$. Os vértices centrais são rotulados. Obtida em (Berton e de Andrade Lopes, 2012).

componentes desconexos. Estes passos realizam uma busca em profundidade (*Breadth-First Search - BFS*) buscando por componentes na rede. Componentes sem nenhum vértice rotulado são conectados a um componente vizinho que possua um ou mais vértices rotulados. Limitamos estas novas conexões para evitar que a rede torne-se densa.

Algorithm 6 GBILI

Entrada: base de dados X
número de vizinhos para o kNN K
lista dos vértices rotulados L
Saída: GL

```
 1: V ← conjunto de vértices a partir de X
 2: GL ← (V, E, W)
 3: for i = 1; i < |V|; i++ {
 4:         kNN[v_i] ← calcula kNN (K,V);
 5:         M-kNN[v_i] ← calcula M-kNN (kNN,V);
 6:         L[v_i] ← encontra vértice rotulado mais próximo de v_i;
 7: }
 8: for i = 1; i < M-kNN[v_i]; i++ {
 9:         M[i] ← D(v_i, v_j) + D(v_j, L[v_j]);
10:         E ← min ∈ M[i];
11: }
12: Componente ← BFS(GL);
13: for i = 1; i < |V|; i++ {
14:         if Componente(v_i) ∉ L {
15:                 For k = 1; k < kNN(v_i); k++ {
16:                         E ← v_k;
17:                 }
18:         }
19: }
20: Retornar GL
```

3.2.2 Análise de complexidade

Com base na matriz de similaridade encontram-se os k vizinhos mais próximos para cada elemento. Para calcular o k vizinho mais próximo para um elemento o algoritmo gasta $kn - \frac{k(k-1)}{2}$ passos, onde n é o número de elementos e k é o número de vizinhos considerado. Para encontrar os k vizinhos mais próximos para todos os elementos, a complexidade é $O(kn^2 - \frac{kn(k-1)}{2})$. Como é necessário calcular o kNN mútuo, neste caso, para cada j-ésimo vizinho de um vértice v_i precisamos verificar se $v_i \in k\text{NN}(v_j)$. Desse modo, leva-se k^2 passos a mais. A complexidade final para calcular os k vizinhos mútuos é $O(kn^2 - \frac{kn(k-1)}{2} + k^2)$.

Também é necessário calcular para todos os elementos, a distância mínima para um ponto rotulado. Se existir l pontos rotulados, é necessário acessar l

vezes a matriz de similaridade para encontrar o ponto rotulado mais próximo de cada elemento. Isso leva hn passos.

O método proposto estabelece uma conexão entre os vértices v_i e v_j que minimiza a soma das distâncias de v_i até um de seus k vizinhos mútuos e deste até um vértice rotulado v_l. É necessário acessar a matriz de similaridade kl vezes, onde k são todos os vizinhos mútuos de v_i e l são todos os vértices rotulados. Este custo é adicionado ao custo de encontrar os k vizinhos mútuos.

No passo de pós-processamento é necessário encontrar todos os componentes da rede, isso pode ser feito por meio da busca em profundidade *Breadth-First Search* (BFS). Este algoritmo inicia em um vértice raíz e inspeciona todos os vértices vizinhos. Para cada vértice ele inspeciona os vizinhos que não foram visitados, e assim por diante. Sua complexidade de tempo é $O(N + M)$ (Cormen et al., 2009), onde N é o número de vértices e M é o número de arestas. Como a rede construída pelo método GBILI possui grau médio aproximadamente de 2 (conforme mostrado no Capítulo 5), pela equação $\langle k \rangle = \frac{2N}{M}$, o número de arestas é próximo ao número de vértices. Quando um componente é identificado sinalizamos se ele possui ou não um vértice rotulado. Se um vértice v_i pertence a um componente que não possui vértices rotulados, ele deve se conectar a um vértice vizinho que pertençe a outro componente com vértices rotulados. No pior caso é necessário verificar entre os k vizinhos de n exemplos, resultando em uma complexidade $O(kn)$.

Ao final a complexidade do algoritmo proposto resulta em $O(kn^2 - \frac{kn(k-1)}{2} + k^2 + kl)$ + $O(2n + kn)$. Ou seja, é uma complexidade quadrática, próxima a do método kNN.

3.2.3 Visualização da rede GBILI

Conforme o valor de k aumenta, as redes kNN tornam-se mais densas. Observa-se na Figura 3.6 a comparação do grau médio entre os métodos kNN, M-kNN e GBILI. Apenas GBILI não é afetado pelo valor de k e permanece com um número constante de arestas. Observa-se que GBILI possui um grau médio próximo a 2, independente do valor de k, indicando que o método proposto gera redes mais esparsas que o kNN.

As redes geradas para a base USPS (Chapelle et al., 2010) com 10 exemplos rotulados podem ser visualizadas na Figura 3.7. A rede kNN foi construída com k igual a 3 gerando uma rede com grau médio em torno de 4.5. A rede GBILI foi construída com k igual a 30 gerando uma rede com grau médio em torno de 2. A quantidade de arestas em uma rede está relacionada com o custo de processamento e eficiência dos algoritmos de propagação de rótulo, portanto redes mais esparsas são mais apropriadas.

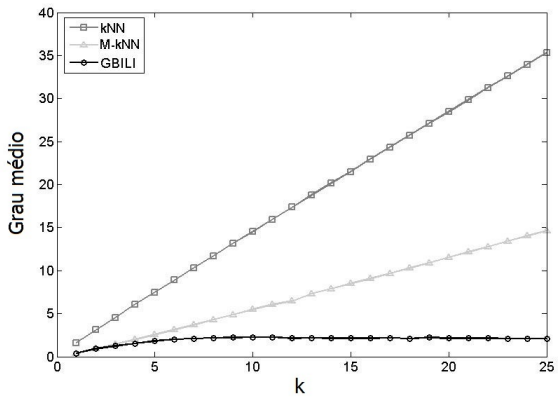


Figura 3.6: Grau médio em uma rede kNN, M-kNN e GBILI construídas com a base USPS. Obtida em (Berton e de Andrade Lopes, 2014).

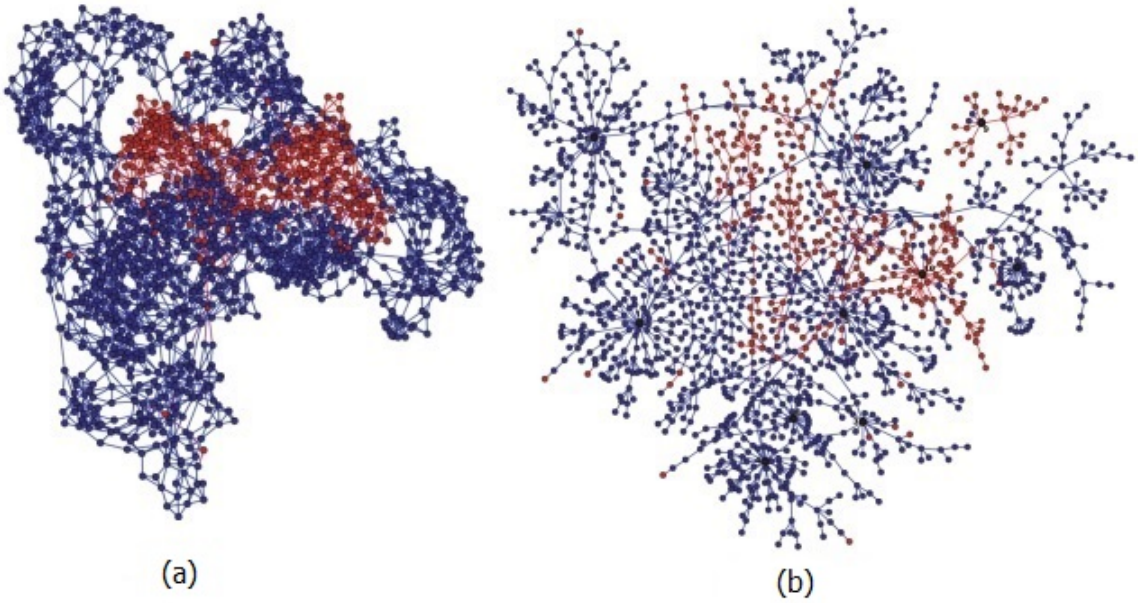


Figura 3.7: Redes kNN (a) e GBILI (b) geradas para a base USPS utilizando 10 exemplos rotulados. Obtida em (Berton e de Andrade Lopes, 2014).

O método GBILI forma *hubs* a partir dos vértices rotulados (pontos pretos na Figura 3.7). Desse modo, a rede GBILI se aproxima de uma "rede livre de escala", pois a probabilidade de se encontrar um vértice com conectividade (ou grau) k diferente da conectividade média da rede, decai exponencialmente para valores altos de k (Figura 3.8).

Para saber se uma rede apresenta distribuição de grau lei de potência pode ser utilizada a técnica de máxima verossimilhança. Com essa técnica busca-se encontrar o melhor valor correspondente para γ. O valor mínimo de corte pode ser calculado usando o método de Kolmogorov-Smirnow. Para isso, utilizou-se

o código disponibilizado por Clauset et al. (2009)[1] e confirmou-se que as redes GBILI são livre de escala, pois $\gamma = 3.82$ e $\gamma > 3$.

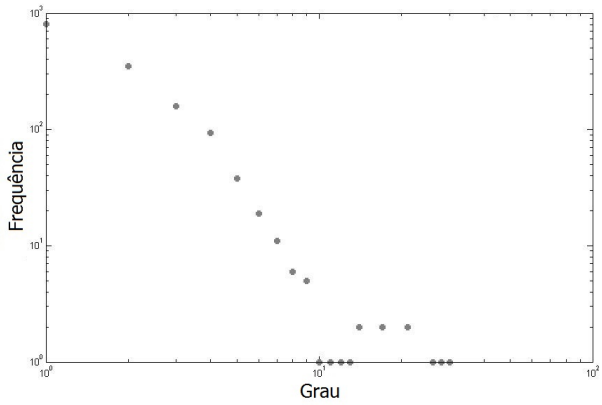


Figura 3.8: Distribuição de grau para a rede GBILI na base de dados USPS.

3.2.4 *Robust Graph that Considers Labeled Instances* (RGCLI)

Apesar do método GBILI ser eficiente para a classificação SSL ele possui limitação quanto a complexidade de tempo, o que pode restringir seu uso em grandes bases de dados. Por exemplo, uma imagem com tamanho 320×480 possui 153600 pixels. Se cada pixel representa um vértice e uma rede 10NN é criada, tem-se uma rede com mais de $1,5 \times 10^6$ arestas. Com base nisso, desenvolveu-se uma versão otimizada do algoritmo GBILI, denominado *Robust Graph that Considers Labeled Instances* (RGCLI). Além disso, provou-se matematicamente que os métodos GBILI e RGCLI cumprem com os pressupostos de SSL.

O método GBILI possui o parâmetro K o qual controla a vizinhança do kNN, porém apenas uma conexão para cada vértice é estabelecida, gerando-se assim uma rede com grau médio próximo a 2. No método RGCLI considerou-se dois parâmetros: k_e, que controla o tamanho da vizinhança do kNN, e k_i, que permite estabelecer mais de uma conexão para cada vértice. Neste caso não se faz necessário o passo de pós-processamento, pois em geral, a rede é conexa.

As entradas para o método RGCLI são um conjunto de pontos X, alguns vértices rotulados ($\mathcal{L} = \{v_{l1}, v_{l2}, v_{l3}\}$) e dois parâmetros ($k_e$ e k_i, onde k_e é o número de vizinhos para o kNN e k_i é o número de vizinhos para o RGCLI). Inicialmente, k_e vizinhos mais próximos de um vértice v_i são encontrados (ver Figura 3.9 (a), neste exemplo $k_e = 15$). Em seguida um score $\mathcal{E}$ é calculado, o qual é dado pela soma das distâncias de um vértice v_i a um vizinho mútuo v_j, e de v_j até um vértice rotulado $v_l \in \mathcal{L}$ (ver Figura 3.9 (b)). Nota-se que o número de vizinhos

[1] http://tuvalu.santafe.edu/ãaronc/powerlaws/

mútuos é menor que k_e encontrado pelo kNN. Por fim, v_i estabelece k_i conexões e escolhe os vizinhos mútuos que minimizam a soma anterior (ver Figura 3.9 (c), neste exemplo $k_i = 3$), priorizando conexões com vértices que são seus vizinhos mútuos, mas que estejam mais próximos de vértices rotulados.

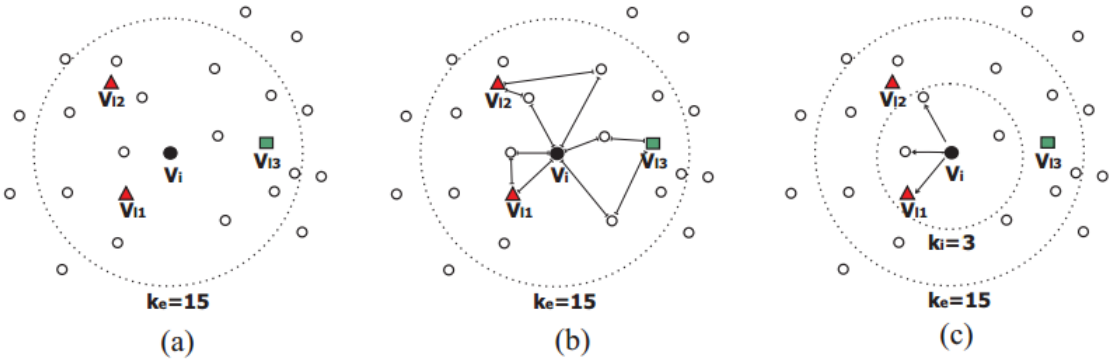


Figura 3.9: Passos para construção do RGCLI: (a) O método kNN é calculado para $k_e = 15$. (b) A soma das distâncias entre v_i e seus vizinhos mútuos e dos vizinhos mútuos até um vértice rotulado v_l é calculada. (c) São estabelecidas $k_i = 3$ conexões para o vértice v_i.

O algoritmo 7 sumariza os passos para a construção da rede RGCLI. Os parâmetros de entrada são k_e e k_i. Para melhores resultados, k_e deve ser maior que k_i, pois k_e representa uma vizinhança maior que RGCLI irá utilizar para selecionar os melhores k_i para se conectar. Foram utilizadas *threads* para tornar a execução mais rápida.

As linhas 1 a 8 incializam os parâmetros e estruturas. As linhas 9 a 12 usam *threads* e chamam a *procedure SearchKNN* que computa os k_e vizinhos mais próximos para cada vértice usando Kd-tree. Além disso, os vértices rotulados mais próximos também são computados. As linhas 13 a 16 usam *threads* e chamam a *procedure SearchRGCLI*. Esta procedure calcula os vizinhos mútuos e o score $\mathcal{E}$, o qual computa a soma da distância de um vértice até seu vizinho mais próximo e desse vizinho até um vértice rotulado. Finalmente o algoritmo estabele k_i conexões com os vizinhos que minimizarem o score $\mathcal{E}$. O qual também é utilizado para representar o peso das arestas.

3.2.5 Análise de complexidade

Inicialmente, uma Kd-tree estática deve ser construída a partir das n instâncias gerando um custo de $O(n \log n)$ se um algoritmo linear baseado em médias for utilizado (Maneewongvatana e Mount, 2001) (Algoritmo 7: linha 4). A partir da Kd-tree precisa-se encontrar k vizinhos mais próximos para cada instância.

Algorithm 7 RGCLI

Entrada:
número de vizinhos para o kNN k_e,
número de vizinhos para RGCLI k_i,
lista de exemplos rotulados L,
base de dados X,
número de threads nt
Saída: GL

1: $V \leftarrow$ cria um conjunto de vértices a partir de X,
2: $E, W \leftarrow \emptyset$,
3: $GL \leftarrow (V, E, W)$,
4: $kdtree \leftarrow$ cria uma Kd-tree a partir de X,
5: kNN $\leftarrow map$,
6: $\mathcal{F} \leftarrow map$,
7: $\mathcal{L} \leftarrow map$,
8: $\mathcal{T} \leftarrow \{T_i : \bigcup_{i=1}^{nt} T_i = V, \bigcap_{i=1}^{nd} T_i = \emptyset\}$,
9: **for** $T_i \in \mathcal{T}$ {
10: $t \leftarrow$ Thread(**SearchKNN**$(T_i, k_e, kdtree, k\text{NN}, \mathcal{L})$)
11: $t.start()$
12: }
13: **for** $T_i \in \mathcal{T}$ {
14: $t \leftarrow$ Thread(**SearchRGCLI**$(GL, T_i, k_i, k\text{NN}, \mathcal{L})$)
15: $t.start()$
16: }
17: **SearchKNN**(T, k_e, $kdtree$, kNN, $\mathcal{L}$) {
18: **for** vértice $v_i \in T$
19: $k\text{NN}[v_i] \leftarrow kdtree.query(v_i, k_e)$
20: $\mathcal{L}[v_i] \leftarrow$ encontra $\mathcal{L}$ vértices rotulados mais próximos de v_i
21: $\mathcal{F}[v_i] \leftarrow$ encontra kth vizinho mais distante de v_i
22: }
23: }
24: **SearchRGCLI**(GL, T, k_i, kNN, $\mathcal{L}$) {
25: **for** vertex $v_i \in T$ {
26: //seja $\mathcal{E}$ um map que atribui um score para uma aresta
27: $\mathcal{E} \leftarrow map$
28: **for** vértice $v_j \in k\text{NN}[v_i]$
29: //condição kNN mútuo
30: **If** $dist(v_i, v_j) \leq dist(v_j, \mathcal{F}[v_j])$ {
31: $e \leftarrow (v_j, v_i)$
32: $\mathcal{E}[e] \leftarrow dist(v_i, v_j) + dist(v_j, \mathcal{L}[v_j])$
33: }
34: }
35: $E^* \leftarrow$ recebe k_i arestas com min score de $\mathcal{E}$
36: $E \leftarrow E^* \cup E$
37: $w(e) \leftarrow \frac{1}{dist(v_i, v_j)} \; \forall \; e = (v_i, v_j) \in E^*$
38: }
39: }

Para encontrar um vizinho mais próximo a partir de pontos distribuídos aleatoriamente, gasta-se $O(\log n)$. Logo, para procurar k_e vizinhos mais próximos

gasta-se $O(k_e \log n)$. Totalizando um tempo de $O(nk_e \log n)$ para completar a busca sobre as n instâncias (Algoritmo 7: linhas 17 - 23).

Posteriormente, o kNN mútuo precisa ser calculado. Neste caso, para cada j-ésimo vizinho de um elemento v_i, verifica-se se $v_i \in k\text{NN}(v_j)$. Para melhorar a complexidade de tempo para o cálculo do kNN mútuo, uma lista de distâncias $\mathcal{F}$ é mantida, a qual armazena a distância máxima de cada vértice v_j para seu vizinho mais distante em $k\text{NN}(v_j)$. Isso permite encontrar o kNN mútuo em k_e passos. Caso v_j esteja na vizinhança mútua de v_i, é calculado a soma das distâncias entre v_i e v_j e entre v_j e um vértice rotulado v_l e armazenado numa variável $\mathcal{E}$. Uma lista de tamanho k_e deve ser ordenada com os valores de $\mathcal{E}$ e apenas os k_i vizinhos mútuos que minimizam $\mathcal{E}$ são selecionados para se conectarem. A complexidade final para encontrar os k_i vizinhos mútuos é $O(nk_i \log n)$ (Algoritmo 7: linhas 24 - 39).

Finalmente, a complexidade do algoritmo RGCLI resulta em $O(n \log n)+O(nk_e \log n)+O(nk_e + nk_i \log n)$, próximo a $O(nk \log n)$. O uso de *threads* foi explorado e os dados foram divididos em $|\mathcal{T}|$ subconjuntos temporários iguais, onde "$|.|$" indica a cardinalidade do conjunto. Utilizando $n \equiv \frac{n}{|\mathcal{T}|}$ reduziu ainda mais o tempo de processamento do algoritmo com o uso de mais processadores (Algoritmo 7: linhas 8 - 16).

Executou-se ambos os métodos, GBILI e a nova versão RGCLI em imagens artificiais de diferentes tamanhos, 50×50, 100×100 e 150×150. Conforme o tamanho da imagem aumenta GBILI aumenta o tempo de execução enquanto RGCLI permanece quase constante (Figura 3.10).

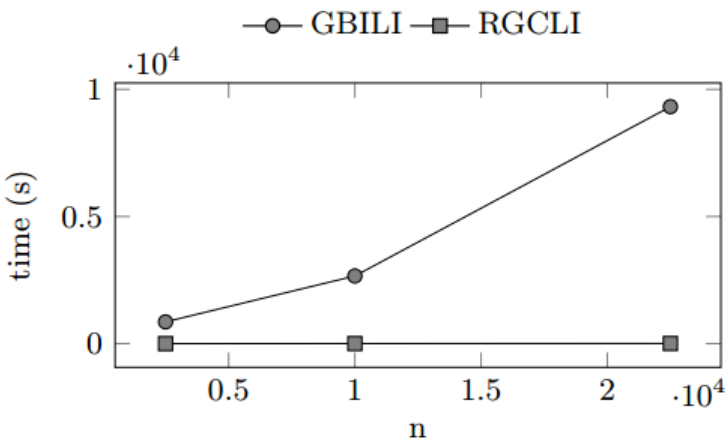


Figura 3.10: Comparação de tempo entre GBILI e RGCLI.

3.2.6 Análise dos pressupostos de SSL nas redes GBILI e RGCLI

Métodos baseados no aprendizado semissupervisionado consideram especialmente dois pressupostos (Chapelle et al., 2010): (i) pontos próximos (vizinhos) tendem a ter o mesmo rótulo e (ii) pontos localizados na mesma estrutura (*cluster* ou *manifold*) tendem a ter o mesmo rótulo. O primeiro pressuposto é local enquanto o segundo é global. Para ilustrar estes pressupostos, considere os dados apresentados na Figura 3.11 (a), gerados a partir de três espirais entrelaçadas. O rótulo (cor) de cada ponto deve ser similar aos pontos em sua vizinhança e os rótulos dos pontos em uma espiral deveriam ser similares entre si e não com outras espirais.

Ao se construir uma rede com o método kNN observa-se que a ligação entre pontos respeita apenas o primeiro pressuposto. Um valor alto para o parâmetro k viola o segundo pressuposto, pois estabelece ligações entre espirais distintas (Figura 3.11 (b)). Por outro lado, uma rede construída com o método M-kNN não viola nenhum pressuposto (Figura 3.11 (c)). Já uma rede construída com o método GBILI ou RGCLI está contida na rede M-kNN e como apenas alguns elementos próximos a um vértice rotulado são selecionados para se estabelecer conexão, a rede é mais esparsa que M-kNN e também não viola nenhum pressuposto (Figura 3.11 (d)).

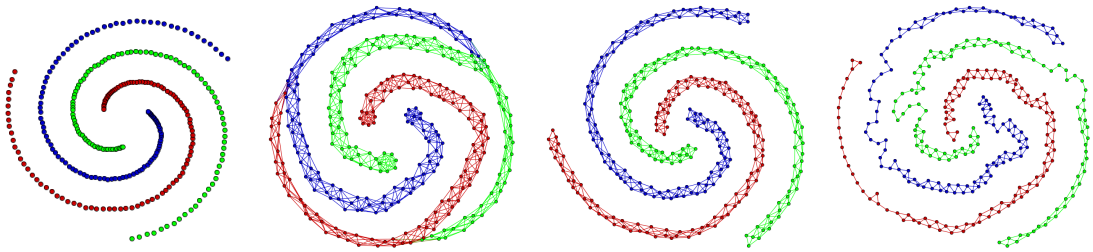


Figura 3.11: (a) Base de dado espiral; (b) rede kNN com $k = 7$; (c) rede M-kNN com $k = 7$; (d) rede RGCLI com $k_e = 7$ e $k_i = 2$.

Uma rede ótima, portanto, deveria respeitar os pressupostos do aprendizado semissupervisionado. Rohban e Rabiee (2012) propuseram uma função objetivo linear para a construção de redes. Os autores provaram que uma rede baseada em vizinhança ótima é uma subrede da rede kNN para um k suficientemente grande, o qual é ainda muito menor que o número total de pontos. Nesta seção é mostrado como os métodos propostos GBILI (Berton e de Andrade Lopes, 2014) e RGCLI respeitam os pressupostos acima para construção de redes.

De acordo com o pressuposto de *manifold*, uma rede baseada em vizinhança deve ser uma rede sobre a qual a função de classificação seja suave. Para tal, os autores (Rohban e Rabiee, 2012) propuseram um problema de otimização linear que maximiza a suavidade dos rótulos com respeito aos pesos das arestas da rede:

$$\min_{W} \sum_{ij} P(y_i \neq y_j | x_i, x_j, y, \{x_i \in N_k(x_j) \vee x_j \in N_k(x_i)\}) \tag{3.2}$$

$$\forall i, j : x_j \notin N_k(x_i) \wedge x_i \notin N_k(x_j) : W_{ij} = 0; \ \forall i \sum_{x_j \in N_k(x_i) \vee x_i \in N_k(x_j)} W_{ij} = k$$

onde $N_k(x_i)$ é o conjunto dos kNNs de x_i. Note que $\sum_{ij} W_{ij} P(y_i \neq y_j | x_i, x_j, y) \propto \sum_{ij} W_{ij}(y_i - y_j)^2$.

Seja $N_\epsilon^{\mathcal{M}}(x_i) = \{x_j : d_{\mathcal{M}}(x_i, x_j) \leq \epsilon\}$ uma ϵ-vizinhança de x_i no *manifold* $\mathcal{M}$, onde $d_{\mathcal{M}}$ é a distância geodésica sobre $\mathcal{M}$. A rede de vizinhança ótima que minimiza o problema de otimização (3.2.6) pode ser definida sobre um *manifold* $\mathcal{M}$, com base em $N_\epsilon^{\mathcal{M}}(x_i)$, ao invés de uma ϵ-vizinhança em um espaço Euclidiano $\mathbb{R}^m$, com base em $N_\epsilon^{\mathbb{R}^m}(x_i) = \{x_j : ||x_i - x_j||_2 \leq \epsilon\}$.

Rohban e Rabiee (2012) mostram que a rede de vizinhança ótima é uma subrede de uma rede $\epsilon + \epsilon'$-vizinhança baseado em distância Euclideana, com grande probabilidade. Dado $\epsilon, \epsilon', \epsilon'', \mu > 0$, suponha que os dados sejam aleatóriamente distribuídos, dentro de uma função de densidade α, existe uma taxa de amostragem grande o suficiente sobre o *manifold* $\mathcal{M}$ tal que $P(\forall x_i : N_\epsilon^{\mathcal{M}}(x_i) \subseteq N_{\epsilon+\epsilon'}^{\mathbb{R}^m}(x_i)) \geq 1 - \mu$. Aqui, mostraremos que as redes GBILI e RGCLI são uma subrede da rede $\epsilon + \epsilon''$-vizinhança, onde $\epsilon'' \leq \epsilon'$.

Proposição 1: Dado $\epsilon, \epsilon', \epsilon'', \mu > 0$, $\epsilon'' \leq \epsilon'$, suponha um conjunto de dados aleatoriamente escolhidos, com uma certa função de densidade α, existe uma taxa de amostragem grande o suficiente sobre o *manifold* $\mathcal{M}$ tal que $P(\forall x_i : N_\epsilon^{\mathcal{M}}(x_i) \subseteq N_{\epsilon+\epsilon''}^{\mathbb{R}^m}(x_i) \subseteq N_{\epsilon+\epsilon'}^{\mathbb{R}^m}(x_i)) \geq 1 - \mu$.

Demonstração: Seja $\lambda = 1 + 4\delta/\varepsilon$, onde $d_{\mathcal{M}}(x_i, x_j) \leq \varepsilon$ para todas as arestas (x_i, x_j), existe um ponto x_i para o qual $d_{\mathcal{M}}(m, x_i) \leq \delta$ para cada ponto m. Em Bernstein et al. (2000) mostra-se que para $\mu > 0$, existe uma amostragem sobre $\mathcal{M}$ tal que

$$P(|d_G(x_i, x_j) - d_{\mathcal{M}}(x_i, x_j)| \leq \lambda d_{\mathcal{M}}(x_i, x_j)) \geq 1 - \mu \tag{3.3}$$

onde $d_G(x_i, x_j)$ é o tamanho do caminho mínimo entre x_i e x_j na rede kNN .

Pela Equação (3.3), Rohban e Rabiee (2012) definem um limite superior $M = \sup_{x_i, x_j \in \mathcal{M}} d_{\mathcal{M}}(x_i, x_j)$. Além disso, $P(|d_G(x_i, x_j) - d_{\mathcal{M}}(x_i, x_j)| \leq \lambda M) \geq 1 - \mu$. Seja $\epsilon' = \lambda M$. Desse modo, com probabilidade de pelo menos $1 - \mu$, tem-se $|d_G(x_i, x_j) - d_{\mathcal{M}}(x_i, x_j)| \leq \epsilon'$. Considerando ainda que, $d_G(x_i, x_j) \leq d_{\mathcal{M}}(x_i, x_j) + \epsilon' \leq \epsilon + \epsilon'$. E além disso, $||x_i - x_j||_2 \leq d_G(x_i, x_j)$. Daí, $||x_i - x_j||_2 \leq \epsilon' + \epsilon$. Isso prova que a rede

de vizinhança ótima, definida no *manifold*, é uma subrede de uma rede $\epsilon + \epsilon'$-vizinhança incorporado em um espaço Euclidiano.

Agora, considerando as redes GBILI ou RGCLI, busca-se diminuir o limite superior $\epsilon + \epsilon'$ para $\epsilon + \epsilon''$, onde $\epsilon'' \leq \epsilon'$. Seja $\mathcal{L}$ o conjunto de rótulos disponível *a priori* para construir as redes GBILI e RGCLI GL. Pela construção das redes GBILI ou RGCLI, temos

$$d_{GL}(x_i, x_j) \leq d_{GL}(x_i, x_l), \forall\ x_l \in \mathcal{L}, \tag{3.4}$$

onde d_{GL} é o tamanho do caminho mínimo entre dois vértice nas redes GBILI ou RGCLI. Do mesmo modo, $d_{\mathcal{M}}(x_i, x_j) \leq d_{\mathcal{M}}(x_i, x_l)$ para todas as arestas (x_i, x_j) em GL. Desse modo, pode-se definir a constante $M' = \sup_{x_i, x_l \in \mathcal{M}\ \wedge\ x_l \in \mathcal{L}} d_{GL}(x_i, x_l)$; note que $M' \leq M$. Assim,

$$1 - \mu\ \leq P(|d_{GL}(x_i, x_j) - d_{\mathcal{M}}(x_i, x_j)| \leq \lambda M') \leq\ P(|d_G(x_i, x_j) - d_{\mathcal{M}}(x_i, x_j)| \leq \lambda M) \tag{3.5}$$

Agora, define-se $\epsilon'' = \lambda M'$. Pela Equação (3.5), tem-se $|d_{GL}(x_i, x_j) - d_{\mathcal{M}}(x_i, x_j)| \leq \epsilon'' \leq \epsilon'$. Desse modo, conclui-se que $||x_i - x_j||_2 \leq \epsilon'' + \epsilon \leq \epsilon' + \epsilon$. Ou seja, existe uma amostragem $\mathcal{M}$ com taxa α, com probabilidade de pelo menos $1 - \mu$, a ϵ-vizinhança para cada ponto $x_i \in \mathcal{M}$ reside na rede $\epsilon'' + \epsilon$-vizinhança criada pelos métodos GBILI e RGCLI.

Pela Proposição 1, definiu-se um novo limite superior para a rede baseada em vizinhança considerando propriedades definidas pela rede GBILI. Isso é importante para a construção de redes pois ajuda a limitar o espaço de possíveis estruturas de redes no problema de otimização (3.2.6).

Além disso, a Proposição 1 pode ser generalizada considerando as redes kNN e GBILI. Para tal, note que para uma rede ϵ-vizinhança $G_\epsilon = (V_\epsilon, E_\epsilon)$, existêm k e k' tal que G_ϵ são uma super-rede e uma sub-rede das redes kNN respectivamente (Rohban e Rabiee, 2012). Do mesmo modo, existem k_i e k_e, e k'_i e k'_e tal que G_ϵ é uma super-rede e sub-rede das redes GBILI ou RGCLI, considerando $k_e = k_i = k = \min_{v_i \in V_\epsilon} deg(v_i)$ e $k'_e = k'_i = k' = \max_{v_i \in V_\epsilon} deg(v_i)$, onde $deg(v_i)$ é uma função que retorna o grau do vértice v_i.

Outra propriedade importante das redes GBILI e RGCLI é obtida pelo conjunto de rótulos $\mathcal{L}$. Como ϵ'' é proporcional ao valor de $M' = \sup_{x_i, x_l \in \mathcal{M}\ \wedge\ x_l \in \mathcal{L}} d_{GL}(x_i, x_l)$, observa-se na Proposição 1 a influência de $\mathcal{L}$ nas redes GBILI ou RGCLI. Bons vértices rotulados, os quais possam discriminar exemplos de diferentes classes, terão menores valores de M', consequentemente fornecerão um limite inferior menor.

3.3 Construção de redes via predição de *links*

A predição de *links* têm sido utilizada em diversos domínios relacionais para predizer quais elementos estão relacionados entre si e qual pode ser o tipo dessas relações. A estrutura dos *links* em uma rede tem sido utilizada, por exemplo, para classificar a importância de documentos em publicações científicas (Tague-Sutcliffe, 1992), para predizer futuras amizades em redes sociais (Wasserman e Pattison, 1996) ou para melhorar a classificação de documentos (Getoor et al., 2001).

Poucos estudos utilizam a predição de *links* para a construção de redes no contexto de classificação. Greenfield e Campbell (2013) propõem diversos métodos para gerar redes que superam o kNN, por meio de medidas de rede e um processo de predição de *links*. Rohban e Rabiee (2012) modificam a rede kNN por meio de uma seleção de *links* e removem aqueles que conectam classes diferentes.

Nesta proposta considerou-se a predição de *links* como um mecanismo para evoluir uma rede inicial e propôs-se o método *Graph Based on Link Prediction* (GBLP) (Berton et al., 2015). Inicialmente, uma estrutura básica e esparsa de rede é construída a partir de um método tradicional. A partir dessa rede, índices de predição de *links* são calculados estimando novas arestas na rede. O raciocínio por trás da proposta é que se uma rede é muito esparsa faltam informações estruturais para os algoritmos de inferência. Por outro lado, se uma rede é muito densa, o excesso de arestas torna-se ruído degradando os algoritmos de inferência. Assim, buscou-se obter uma rede com número de arestas balanceado e topologia próxima a redes mundo-pequeno, com alto coeficiente de *clustering* e a média dos caminhos mínimos baixa. A seguir são apresentadas as medidas de predição de *links* consideradas e a proposta de construção de redes.

3.3.1 Predição de *links*

A predição de *links* (*link prediction* - LP) utiliza a similaridade entre vértices para calcular quais arestas podem ser incluídas em uma rede, seja gerando todas as arestas ou adicionado arestas em uma rede já existente (Liben-Nowell e Kleinberg, 2007; Lü e Zhou, 2011). Dado uma rede G, seja U o conjunto contendo todas as $\frac{|V|(|V|-1)}{2}$ possíveis arestas, onde $|V|$ representa o número de elementos no conjunto V. Desse modo, o conjunto de *links* não existente é $U - E$. A tarefa da predição de *links* é encontrar os *links* não existentes no conjunto $U - E$.

As medidas de predição de *links* podem ser classificadas em diferentes tipos. As medidas locais incluem *common neighbors*, *Jaccard coefficient*, *Adamic*

Adar, *resource allocation* e *preferential attachment*. As medidas globais mais utilizada são *Katz*, *rooted pageRank* e *simRank* (Valverde-Rebaza e Lopes, 2013). Além disso, algumas medidas baseadas no peso das arestas foram propostas como variantes de medidas locais (Murata e Moriyasu, 2007; Lü e Zhou, 2010). Considerou-se neste trabalho apenas uma medida de cada tipo: *common neighbors* e sua variação com peso *weighted common neighbors* e a medida *Katz*.

A definição formal das medidas de predição de *links* consideradas são apresentadas a seguir.

***Common neighbors* (c)**: para um vértice v_i, considere um conjunto de vizinhos $\Gamma(v_i)$. Dois vértices, v_i e v_j, têm maior probabilidade de terem um *link* se eles tiverem muitos vizinhos em comum. O cálculo dessa sobreposição de vizinhança é a contagem direta dada pela Equação 3.6.

$$s^c_{v_i,v_j} = |\Gamma(v_i) \cap \Gamma(v_j)| \tag{3.6}$$

***Weighted common neighbors* (w)**: a versão com peso do índice *common neighbors* considera a adição de todos os pesos dos vizinhos em comum, resultantes do score $s^c_{v_i,v_j}$, sendo definida pela Equação 3.7 (Lü e Zhou, 2010).

$$s^w_{v_i,v_j} = \sum_{v_k \in \Gamma(v_i) \cap \Gamma(v_j)} w(v_i, v_k) + w(v_k, v_j) \tag{3.7}$$

***Katz* (k)**: este índice é baseado no conjunto de todos os caminhos, que são amortecidos pelo comprimento dos mesmos, para dar assim um peso maior aos menores caminhos de acordo com a Equação 3.8 (Katz, 1953).

$$s^k_{v_i,v_j} = \sum_{l=1}^{\infty} \beta^l \cdot |paths^{\langle l \rangle}_{v_i,v_j}| = \beta A_{v_i,v_j} + \beta^2 (A^2)_{v_i,v_j} + \dots \tag{3.8}$$

onde $paths^{\langle l \rangle}_{v_i,v_j}$ é o conjunto de todos os caminhos com tamanho l conectando v_i e v_j, A é a matriz de adjacência e β é um parâmetro livre (i.e. o fator de amortecimento) controlando o peso dos caminhos. Um β muito pequeno leva a uma medida próxima a c, já que os caminhos mais longos contribuem pouco.

Uma versão matricial dessa medida é dada pela Equação 3.9.

$$S^k_{v_i,v_j} = (I - \beta A)^{-1} - I \tag{3.9}$$

β deve ser menor que o maior autovalor da matriz A para a convergência da Equação 3.8. Nos experimentos foram atribuídos os valores 0.05 e 5 para β e l respectivamente.

3.3.2 *Graph Based on Link Prediction* (GBLP)

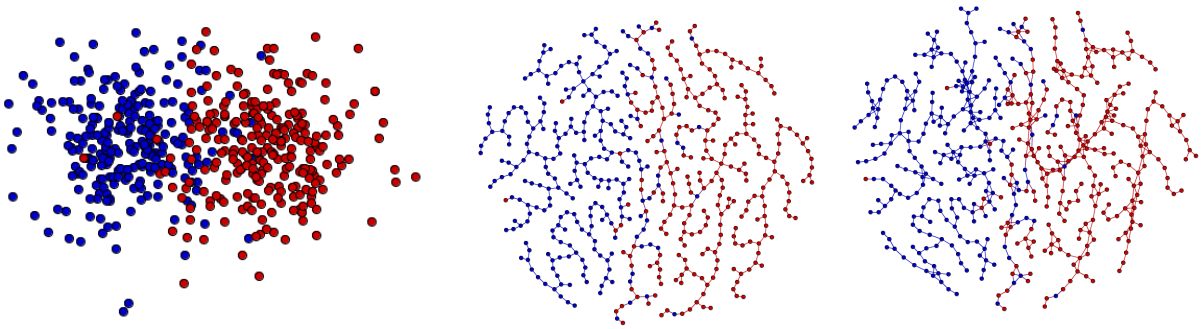
Como se deseja predizer novas arestas em uma rede, é necessário considerar arestas/*links* já existentes. Desse modo, uma estrutura básica de uma rede é necessária. Essa rede inicial deve ser esparsa, minimizando assim o custo dos métodos de predição de *links* e mantendo a esparsidade mesmo com a adição de novas arestas.

As técnicas *minimum/maximum spanning tree* (MinST/MaxST) ou k-vizinhos mais próximos (kNN), com valores baixos de k, foram utilizadas para obter essa rede básica inicial. Em seguida, um conjunto de potenciais *links* entre os vértices desconectados da rede inicial são preditos. Para predizê-los, três índices baseados em similaridade são utilizados: um índice local (*common neighbors*), um índice com peso (*weighted common neighbors*) e um índice global (*Katz*). Entretanto, qualquer índice para predição de *links* pode ser utilizado.

Para a predição dos *links*, associa-se um *score* para cada par de vértices desconectados v_i e v_j, o qual indica a similaridade entre v_i e v_j. Todos os *links* não observados são ranqueados segundo esse *score*, e aqueles que conectam vértices mais similares supõe-se que deveriam existir. Pode-se considerar uma porcentagem desses *links* preditos, por exemplo 30%, para ser adicionada na rede.

Um exemplo do esquema de construção de redes proposto, denominadas *Graph Based on Link Prediction* (GBLP), é mostrado na Figura 3.12. A partir de dados em formato atributo-valor um método de construção de redes conhecido é aplicado (MinST). Em seguida, calcula-se uma medida de predição de *links* (*Katz*) e uma porcentagem dos *links* preditos (nesse caso 30%) é considerada. Obtém-se assim uma rede final que contém as arestas da rede MinST e as arestas preditas pelo índice *Katz*. Essa rede final é então utilizada para a classificação.

O Algoritmo 8 mostra os passos para construir as redes GBLP. A entrada é um conjunto de dados X e a porcentagem de *links* a ser considerada $percentageL$. A saída é a rede G. As linhas 1 a 3 inicializam as estruturas. As linhas 6 a 11 criam uma rede inicial. Como já comentado pode-se aplicar qualquer método para construção de redes esparsa, como kNN, MinST, etc. Por fim, as linhas 12 a 18 aplicam uma medida de predição de *links* sobre a rede inicial para computar novas arestas. Pode-se aplicar qualquer índice para predição de *links*, como *common neighbors*, *weighted common neighbors*, *Katz*, etc. Uma porcentagem dessas novas arestas são consideradas para a rede final.



(a) (b) (c)

Figura 3.12: Passos para a construção da rede GBLP: (a) Base de dados Gaussiana. (b) Rede MinST. (c) Rede MinST+LP (Katz) considerando 30% dos *links* preditos. Obtida em (Berton et al., 2015)

Algorithm 8 Rede GBLP

Entrada: X, $percentageL$
Saída: G

1: $V \leftarrow$ cria um conjunto de vértices a partir de X,
2: $E, W \leftarrow \emptyset$,
3: $G \leftarrow (V, E, W)$,
4: *SearchGraph*(G)
5: *SearchLP*$(G, percentageL)$
6: **SearchGraph**(G) {
7: **for** vértice $v_i \in G$ {
8: $N \leftarrow$ getNeighbors(v_i);
9: $E \leftarrow$ getEdges(N);
10: }
11: }
12: **SearchLP**$(G, percentageL)$ {
13: **for** vértice $v_i \in G$ {
14: $LP \leftarrow$ getPredictedLinks(G);
15: }
16: $E^* \leftarrow$ getTopLinks$(percentageL, LP)$;
17: $E \leftarrow E^* \cup E$;
18: }

3.3.3 Análise de complexidade

Nos experimentos considerou-se os métodos kNN, M-kNN, MinST e MaxST para construção da rede inicial. A seguir é apresentado a complexidade de tempo para calcular as redes kNN e MinST. As redes M-kNN e MaxST possuem um tempo similar as redes kNN e MinST, respectivamente.

Para computar o kNN, uma Kd-tree estática pode ser construída sobre as n instâncias, o que resulta em um tempo $O(n \log n)$ se um algoritmo baseado em médias for utilizado (Maneewongvatana e Mount, 2001). Para encontrar um

vizinho mais próximo em uma Kd-tree balanceada a partir de pontos distribuídos aleatoriamente leva-se $O(\log n)$. Para encontrar k vizinhos mais próximos leva-se $O(k \log n)$. Resultando em um tempo final de $O(nk \log n)$ para iterar sobre as n instâncias.

Para computar o MinST, uma rede completa precisa ser construída inicialmente. Como esta é uma operação básica o tempo de construção é constante (por exemplo, criar uma matriz $A^{n \times n}$ com todos os elementos iguais a 1). Após isso, o algoritmo de Kruskal, ou Prim, pode ser aplicado para encontrar uma MinST. Ambos os algoritmos exploram a rede dada a partir de um vértice árbitrário v_i, iterando sobre os vizinhos desses vértices. Cada vizinho não visitado é adicionado a uma estrutura para ser explorado posteriormente. Uma MinST pode ser formada conectando-se pares de vértices que minimizam a soma dos pesos de suas arestas. Esses algoritmos levam um tempo igual a $O(|E| \log |E|)$ (Cormen et al., 2009), onde $|E|$ é o número de arestas da rede. Como o número de arestas em uma rede completa é igual a $\frac{n(n-1)}{2}$, a complexidade final é $O(n^2 \log n)$.

Subsequentemente, um índice de predição de *links* precisa ser calculado. Neste caso, índices diferentes resultam em tempos diferentes. Considerando o índice *Common Neighbor*, para cada j-ésimo vizinho de um elemento v_i é necessário verificar quais vizinhos são comuns. Se já tivermos uma lista dos k vizinhos de cada vértice, será necessário acessar esses k vizinhos k vezes, resultando em um tempo $O(nk^2)$.

A complexidade final do método proposto resultará na soma das complexidades da construção da rede com a predição de *links*, o que depende dos métodos escolhidos. Por exemplo, será $O(nk \log n) + O(nk^2)$ se a rede kNN e o índice *Common Neighbor* forem utilizados.

3.4 Considerações finais

Neste capítulo foram apresentados três métodos propostos para a construção de redes especialmente para o SSL. O método S-kNN gera redes regulares e foi proposto como uma alternativa ao método b-matching, porém, com um tempo computacional quadrático. O método b-matching é cúbico e seu uso é inviável em grandes bases de dados.

Os métodos GBILI e RGCLI foram propostos para se utilizar os rótulos disponíveis *a priori* em SSL na construção das redes. Esses rótulos se tornam *hubs*, gerando redes livres de escala. Além disso, mostrou-se que tais redes respeitam os pressupostos de SSL. Para a aplicação desses métodos, destaca-se a impor-

tância do uso de bons rótulos para se obter uma rede mais apropriada para a classificação.

O método GBLP foi proposto para melhorar a conectividade de uma rede esparsa pré-existente, gerando redes com características mundo-pequeno. Observa-se que diferentes medidas para predição de novos *links* podem ser usadas de acordo com a natureza do problema a ser modelado pela rede.

O próximo capítulo apresenta a aplicação dos métodos propostos em *benchmarks* da área além de aplicações específicas, a fim de evidenciar as características intrínsecas de cada método.

CAPÍTULO

4

Avaliação dos métodos para construção de redes propostos

As redes propostas foram avaliadas considerando-se o cenário de classificação semissupervisionado, além disso, foram analisadas por meio de medidas de redes, principalmente relacionadas ao grau, e foram aplicadas em cenários reais. A Seção 4.1 apresenta avaliações das redes S-kNN, a Seção 4.2 avalianções das redes GBILI e RGCLI e a Seção 4.3 avaliações das redes GBLP.

4.1 Análise e aplicação das redes regulares

A fim de estudar se métodos que geram redes regulares são mais adequados para a classificação semissupervisionada, analisou-se alguns métodos para construção de redes regulares (b-matching, Greedy, S-kNN) versus não regulares (kNN, M-kNN), buscando por uma relação entre a topologia das redes e a acurácia na classificação. As principais contribuições dessa análise são apresentadas a seguir.

- Os resultados da classificação considerando bases artificiais e reais, dos repositórios UCI (Bache e Lichman, 2013) e Chapelle (Chapelle et al., 2010), mostram que as redes regulares obtém melhor acurácia.

- A distribuição da acurácia da classificação apresenta três variações: uma normal, bimodal e concentrada em um intervalo. Essas variações indicam que a acurácia pode ser influenciada, por exemplo, pela topologia da rede.

- A comparação de tempo de processamento mostra que os métodos kNN, M-kNN e S-kNN são mais rápidos, construindo uma rede em menos de 1 segundo para uma base com 1500 instâncias. Os métodos b-matching e Greedy levam mais de duas horas nas mesmas condições.

- Todos os métodos podem gerar redes desconexas, especialmente para valores baixos de k e b. Apenas considerando *maximum* e *minimum spanning tree* a rede sempre é conexa.

- Medidas de redes baseadas no grau, tais como grau médio, maior grau (*hub*), diâmetro, entropia do grau, segundo momento do grau, complexidade e coeficiente de *clustering*, indicam que o método kNN gera redes mais densas e heterogêneas, comparadas com as redes geradas pelos métodos b-matching, Greedy e S-kNN.

- As correlações de Pearson e Spearman indicam que os métodos b-matching e S-kNN, possuem correlação negativa moderada entre acurácia e as medidas de redes baseadas no grau indicando que quanto menor o valor dessas medidas maior a acurácia. Além disso, todos os métodos apresentam correlação moderada com a média dos caminhos mínimos, indicando que redes esparsas obtém melhores resultados na classificação.

A seguir são apresentadas análise das redes regulares e não regulares na classificação semissupervisionada, na caracterização topológica e por fim em uma aplicação real considerando classificação de gêneros musicais.

4.1.1 Classificação semissupervisionada

Nesta seção são apresentados os resultados da classificação semissupervisionada comparando métodos regulares e não regulares para a construção de redes. Na Seção 4.1.1.1 são descritas as configurações para a realização dos experimentos; na Seção 4.1.1.2 são apresentados os resultados em bases artificiais e na Seção 4.1.1.3 os resultados em bases reais, na Seção 4.1.1.4 são apresentadas as distribuições da acurácia, são apresentadas as distribuições dos parâmetros e são comparados os tempos de execução dos métodos para construção das redes consideradas.

4.1.1.1 Configuração dos experimentos

Foram empregados os métodos kNN, S-kNN (Vega-Oliveros et al., 2014), b-matching via *belief propagation* (BP ou b-mathing) e b-matching via *greedy approximation* (Greedy) (Jebara et al., 2009) e M-kNN combinado com *minimum spanning tree* para construção das redes. O método para inferência dos rótulos considerado foi o algoritmo *Local and Global Consistency* (LGC) (Zhou et al., 2004), o qual é frequentemente reportado em trabalhos da literatura que comparam construção de redes (Jebara et al., 2009; Ozaki et al., 2011). Este algoritmo faz uso da rede G para difundir os rótulos conhecidos para os vértices não rotulados.

A matriz de distância $D \in \Re^{n \times n}$, obtida por meio da distância Euclidiana entre os pontos, foi utilizada para construir as redes. A matriz de adjacência A para os métodos kNN, M-kNN e b-matching é transformada em simétrica, $\widehat{A}_{ij} = max(A_{ij}, A_{ji})$. Para a geração da matriz de pesos W, considerou-se a abordagem de peso binária, tal que $W = A$. Alguns autores consideram a função *kernel gaussiano* para ponderar a rede, entretanto não existe um acordo sobre qual parâmetro aplicar em σ.

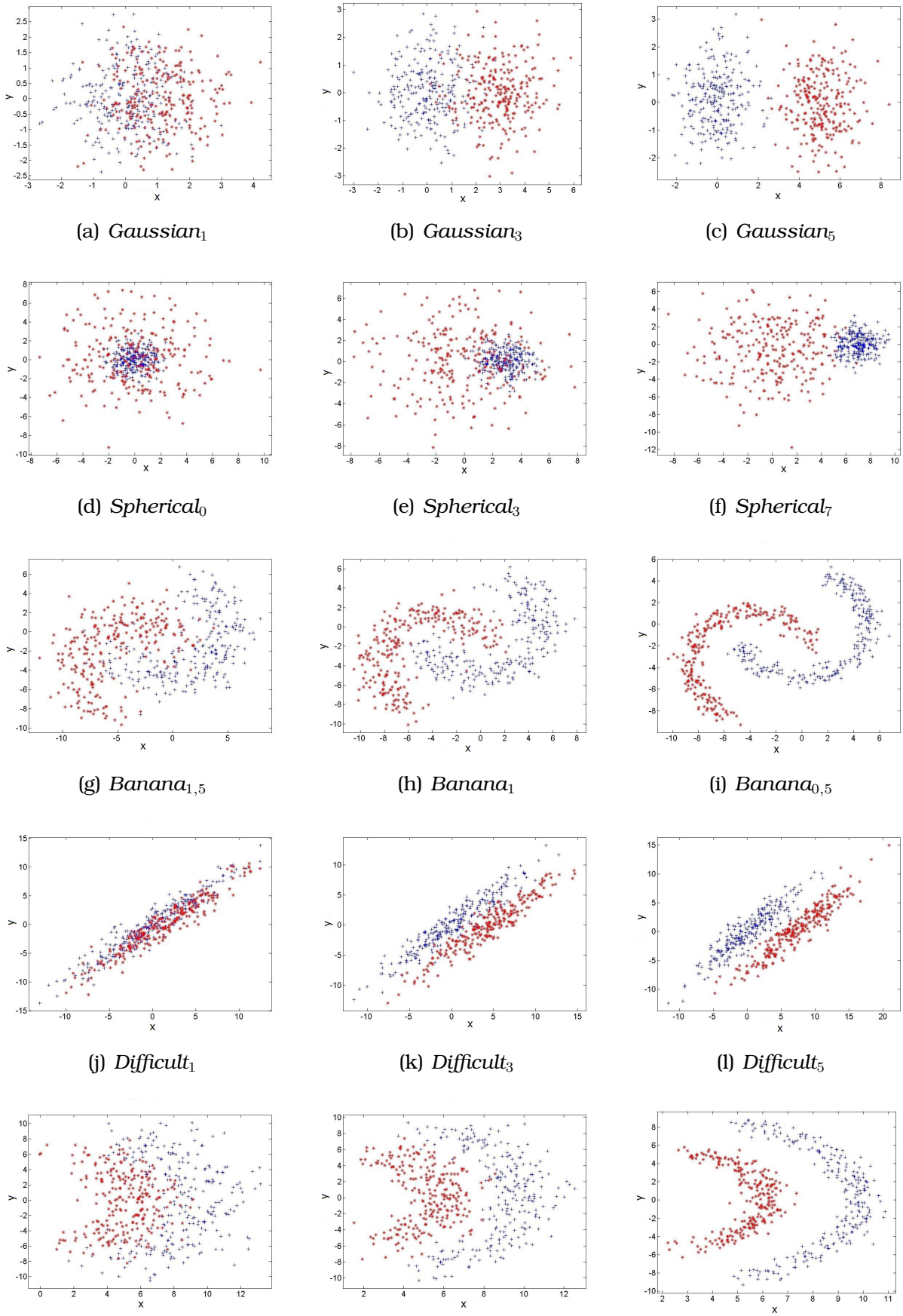
A média da acurácia da classificação sobre 30 execuções, considerando seleção aleatória dos pontos rotulados, foi utilizada como medida de avaliação. A mesma seleção dos pontos rotulados foi aplicada para todos os métodos de construção de redes. Os parâmetros k e b variaram de 1 até 20.

4.1.1.2 Experimentos em bases artificiais

Nesta seção são descritos experimentos aplicando os métodos de construção de redes em diferentes bases de dados artificiais.

Descrição das bases artificiais

Nos experimentos, considerou-se bases artificiais com diferentes distribuições e sobreposições de classes. Na Figura 4.1 mostra-se as bases consideradas. Todas as bases possuem duas dimensões e duas classes contendo 250 elementos em cada uma. Considerou-se também as bases de dados *g241c* e *g241n* com distribuição Gaussiana (Chapelle et al., 2010), ambas geradas sem estrutura de *manifold*, sendo que a base *g241c* possui estrutura de cluster e *g241n* não possui. O método *Principal Component Analysis* (PCA) foi aplicado apenas nas bases *g241c* e *g241n* reduzindo as dimensões iniciais de 241 para 50, pois em dados com muitas dimensões, a distância do vizinho mais próximo se aproxima a distância do vizinho mais distante, degenerando a qualidade das redes geradas.



(a) *Gaussian*$_1$ (b) *Gaussian*$_3$ (c) *Gaussian*$_5$

(d) *Spherical*$_0$ (e) *Spherical*$_3$ (f) *Spherical*$_7$

(g) *Banana*$_{1,5}$ (h) *Banana*$_1$ (i) *Banana*$_{0,5}$

(j) *Difficult*$_1$ (k) *Difficult*$_3$ (l) *Difficult*$_5$

(m) *Lithuanian*$_{1,5}$ (n) *Lithuanian*$_1$ (o) *Lithuanian*$_{0,5}$

Figura 4.1: Bases de dados artificiais. (a),(b) e (c) *Gaussian*; (d),(e) e (f) *Spherical*; (g),(h) e (i) *Banana*; (j),(k) e (l) *Difficult*; (m),(n) e (o) *Lithuanian*.

Tabela 4.1: Média da acurácia (%) e desvio padrão considerando a classificação das bases artificiais. Os valores mais altos de acurácia para cada base de dados estão em negrito.

Data set(# pontos rotulados)	b-matching	Greedy	kNN	S-kNN	M-kNN
Gaussian$_1$(2)	**61.86 ± 10.35**	61.43 ± 10.72	58.55 ± 8.92	58.50 ± 11.63	57.91 ± 10.15
Gaussian$_3$(2)	**88.15 ± 6.01**	86.98 ± 11.00	83.88 ± 8.17	83.57 ± 8.81	81.94 ± 5.11
Gaussian$_5$(2)	**98.69 ± 0.61**	98.59 ± 0.39	91.47 ± 5.08	91.24 ± 5.04	90.72 ± 4.75
Spherical$_0$(2)	69.01 ± 10.14	**75.53 ± 8.23**	63.02 ± 7.69	64.79 ± 7.87	64.34 ± 9.41
Spherical$_3$(2)	78.36 ± 8.07	**82.90 ± 4.93**	74.06 ± 7.20	74.48 ± 7.10	75.34 ± 8.47
Spherical$_7$(2)	93.99 ± 3.48	**95.20 ± 2.03**	85.94 ± 8.47	88.03 ± 5.96	88.70 ± 8.25
Banana$_{0,5}$(2)	89.23 ± 7.06	85.42 ± 10.48	**95.66 ± 5.05**	95.21 ± 6.34	87.27 ± 9.24
Banana$_1$(2)	80.09 ± 9.44	78.36 ± 9.10	78.92 ± 15.43	80.66 ± 11.89	**86.14 ± 7.85**
Banana$_{1,5}$(2)	76.07 ± 9.99	76.43 ± 9.89	72.34 ± 21.47	71.25 ± 11.33	**81.08 ± 12.22**
Difficult$_1$(2)	52.08 ± 3.07	**52.40 ± 10.87**	51.72 ± 4.77	51.61 ± 5.38	52.04 ± 40.38
Difficult$_3$(2)	62.78 ± 11.20	**64.42 ± 16.64**	62.11 ± 19.25	62.52 ± 21.87	59.53 ± 23.51
Difficult$_5$(2)	83.24 ± 13.90	80.18 ± 18.29	**89.72 ± 22.97**	89.45 ± 25.40	81.91 ± 30.83
Lithuanian$_{0,5}$(2)	**93.92 ± 18.26**	83.97 ± 18.05	93.36 ± 24.12	93.46 ± 25.73	86.90 ± 27.22
Lithuanian$_1$(2)	**71.52 ± 19.02**	69.16 ± 16.48	64.42 ± 28.92	68.80 ± 31.95	61.90 ± 26.41
Lithuanian$_{1,5}$(2)	**71.82 ± 12.27**	69.46 ± 15.68	62.02 ± 20.17	63.64 ± 22.04	60.35 ± 21.95
g241c(2)	**60.50 ± 5.76**	55.39 ± 4.32	53.63 ± 5.55	57.24 ± 6.48	50.96 ± 1.85
g241n(2)	**54.47 ± 5.73**	53.48 ± 4.21	52.57 ± 4.35	53.09 ± 3.45	50.83 ± 1.79

Resultados da classificação considerando as bases artificiais

A partir das bases de dados artificiais, obtêm-se os resultados da classificação exibidos na Tabela 4.1. A primeira coluna da Tabela 4.1 apresenta o nome da base de dados juntamente com o número de vértices rotulados considerados, entre parênteses. A segunda coluna e as seguintes, até a sexta, apresentam a acurácia da classificação e o desvio padrão obtidos para cada método de construção de redes considerado. O maior valor de acurácia para cada base está em negrito.

Considerou-se os valores mais baixos possíveis de vértices rotulados (apenas um para cada classe). Se esse número é aumentado, a acurácia também aumenta e o desvio padrão diminui. Nesses experimentos, os métodos b-matching e Greedy obtiveram em geral maior valor de acurácia. Os métodos kNN e M-kNN tiveram valores mais altos apenas nas bases *Bananas*.

Aplicou-se o teste de Freidman e *post-hoc* de Nemenyi (Demšar, 2006) para verificar se existe diferença estatística significativa entre os algoritmos de construção de redes avaliados. Considerando os resultados da classificação apresentados na Tabela 4.1, o valor crítico para comparar o *ranking* médio de dois algoritmos diferentes no percentil 95 é 1.48. O resultado é apresentado na Figura 4.2. No topo do diagrama encontra-se a diferença crítica (*critical difference* - CD) e no eixo encontram-se as médias dos *ranks* das medidas, onde os menores *ranks* (melhores) estão à esquerda. Os métodos que não possuem diferença significativa são conectados por uma linha preta no diagrama. Observa-se que os métodos que geram redes regulares estão melhor ranqueados e o método b-matching possui diferença estatística significativa comparado aos métodos kNN e M-kNN.

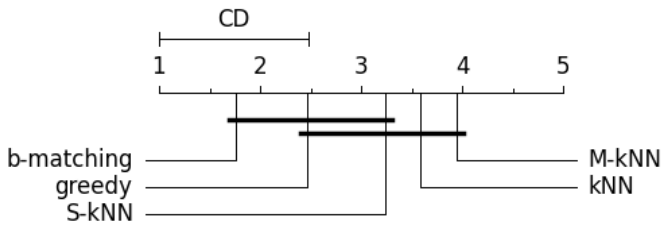


Figura 4.2: Comparação dos métodos de construção de redes pelo teste de Nemenyi considerando as bases de dados artificiais.

4.1.1.3 Experimentos em bases reais

Nesta seção são descritos experimentos aplicando os métodos de construção de redes em diferentes bases de dados reais.

Descrição das bases reais

Considerou-se bases de dados reais com diferentes números de instâncias, atributos e classes. Em geral as bases são pequenas pois o método b-matching e sua variação Greedy tomam muito tempo de processamento e os experimentos foram executados 30 vezes para cada variação de b entre 1 e 20, obtendo-se 600 redes para cada base. Do repositório *UCI Machine Learning Repository* (Bache e Lichman, 2013) considerou-se as bases: Zoo, Iris, Glass, Ecoli, Breast Cancer, Blood Transfusion e Yeast. Para as bases Zoo, Glass, Ecoli, Yeast e Breast Cancer removeu-se o primeiro atributo relacionado ao nome ou ID. Para a base Breast Cancer também removeu-se as instâncias com valores ausentes.

Considerou-se também três bases propostas por Chapelle et al. (2010). USPS é derivada de uma base de imagens de dígitos escritos a mão, na qual os dígitos "2" e "5" são atribuídos à classe $+1$ e os demais dígitos à classe -1. Digit_1 consiste em imagens geradas artificialmente do dígito "1". COIL_2 é derivada do *Columbia Object Image Library* a qual se constitui de um conjunto de imagens coloridas de 100 objetos tomados a partir de diferentes ângulos. o método PCA foi aplicado nas bases USPS, Digit_1 e COIL_2 para reduzir as dimensões de 241 para 50. Mais detalhes das bases consideradas são apresentados na Tabela 4.2.

Resultados da classificação considerando as bases reais

A partir das bases de dados reais obteve-se os resultados da classificação exibidos na Tabela 4.3. A primeira coluna da Tabela 4.3 apresenta o nome da base de dados juntamente com o número de vértices rotulados considerado, entre parênteses. A segunda coluna e as seguintes, até a sexta, apresentam a acurácia

Tabela 4.2: Descrição das bases de dados reais.

Data set	# Instâncias	# Atributos	# Classes	Características dos atributos
Zoo	101	17	7	Categórico e inteiro, características de animais
Iris	150	4	3	Real, planta Iris
Glass	214	10	6	Real, identificação de vidros
Ecoli	336	8	8	Real, localização de proteínas
Breast Cancer	699	10	2	Inteiro, tipos de cancer
Blood transfusion	748	5	2	Real, doação de sangue
Yeast	1484	8	10	Real, localização celular de proteínas
USPS	1500	241	2	Real, dígitos escritos a mão: 2 e 5 vs restante
Digit_1	1500	241	2	Real, imagens artificiais
COIL_2	1500	241	2	Real, imagens

Tabela 4.3: Média da acurácia (%) e desvio padrão considerando a classificação das bases reais. Os valores mais altos de acurácia para cada base de dados estão em negrito.

Data set (# pontos rotulados)	b-matching	Greedy	kNN	S-kNN	M-kNN
Zoo(7)	**70.88 ± 13.98**	65.17 ± 11.05	67.55 ± 16.68	65.92 ± 14.59	69.32 ± 13.66
Zoo(21)	58.41 ± 11.34	62.62 ± 7.17	61.75 ± 10.78	**64.29 ± 10.15**	62.5 ± 12.11
Iris(3)	90.36 ± 6.95	88.29 ± 7.04	90.34 ± 6.45	**93.15 ± 4.32**	87.23 ± 8.78
Iris(9)	86.87 ± 6.30	84.49 ± 8.28	85.13 ± 8.36	**88.29 ± 7.55**	81.18 ± 11.75
Glass(6)	**41.42 ± 8.99**	38.89 ± 7.86	39.69 ± 16.30	38.52 ± 7.60	41.28 ± 8.33
Glass(18)	54.60 ± 9.46	48.70 ± 7.03	51.78 ± 9.76	**55.10 ± 8.97**	54.21 ± 7.07
Ecoli(8)	**68.41 ± 9.68**	64.28 ± 10.79	64.69 ± 10.00	66.55 ± 8.04	62.64 ± 11.46
Ecoli(24)	74.64 ± 7.01	69.54 ± 6.98	75.35 ± 7.20	**76.76 ± 6.39**	76.73 ± 5.39
Breast Cancer(2)	93.72 ± 9.55	94.35 ± 1.39	96.04 ± 3.06	**96.66 ± 0.75**	95.50 ± 2.07
Breast Cancer(6)	83.99 ± 12.72	91.75 ± 7.69	92.05 ± 9.74	93.86 ± 7.84	**95.76 ± 2.49**
Blood transfusion(2)	25.47 ± 9.43	55.68 ± 17.74	56.34 ± 13.60	**59.27 ± 12.81**	57.19 ± 11.99
Blood transfusion(6)	70.94 ± 15.75	**72.12 ± 7.02**	67.42 ± 7.11	68.01 ± 6.86	66.67 ± 8.77
Yeast(10)	**30.77 ± 5.55**	28.91 ± 6.57	25.90 ± 6.75	26.29 ± 5.91	25.31 ± 6.21
Yeast(30)	**40.62 ± 4.78**	39.02 ± 5.31	39.11 ± 5.30	39.37 ± 5.021	39.88 ± 4.30
USPS(10)	81.88 ± 2.49	82.36 ± 3.26	82.80 ± 5.07	82.26 ± 5.78	**84.12 ± 5.86**
USPS(100)	85.78 ± 3.11	83.52 ± 2.42	86.13 ± 3.77	87.30 ± 1.85	**92.37 ± 1.60**
Digit_1(10)	86.05 ± 9.63	81.29 ± 11.36	84.98 ± 8.83	84.29 ± 9.54	**87.69 ± 6.37**
Digit_1(100)	**96.62 ± 0.80**	95.29 ± 1.84	94.23 ± 1.01	94.09 ± 1.13	92.91 ± 1.73
COIL_2(10)	49.93 ± 0.045	**65.79 ± 6.77**	64.97 ± 4.06	64.66 ± 3.81	63.19 ± 6.31
COIL_2(100)	49.99 ± 0.34	89.68 ± 2.49	**92.46 ± 1.82**	91.70 ± 2.54	88.79 ± 2.80

da classificação e o desvio padrão obtidos por cada método de construção de redes considerado. O maior valor de acurácia para cada base está em negrito.

Para as bases do UCI, a quantidade de vértices rotulados considerados foram igual ao número de classes e três vezes esse número. Para as bases de Chapelle et al. (2010), considerou-se 10 e 100 vértices rotulados. Neste caso também considerou-se um número baixo de vértices rotulados.

Aplicou-se o teste de Freidman e *post-hoc* de Nemenyi (Demšar, 2006) para verificar se existe diferença estatística significativa entre os algoritmos de construção de redes avaliados. Considerando os resultados da classificação apresentados na Tabela 4.3, o valor crítico para comparar o *ranking* médio de dois algoritmos diferentes no percentil 95 é 1.36. O resultado é mostrado na Figura 4.3. No topo do diagrama encontra-se a diferença crítica (*critical difference* - CD) e no eixo encontram-se as médias dos *ranks* das medidas, onde os menores *ranks* (melhores) estão a esquerda. Os métodos que não possuem diferença significativa são conectados por uma linha preta no diagrama. Observa-se na Figura 4.3 que apenas o método S-kNN, que gera redes regulares, ficou melhor

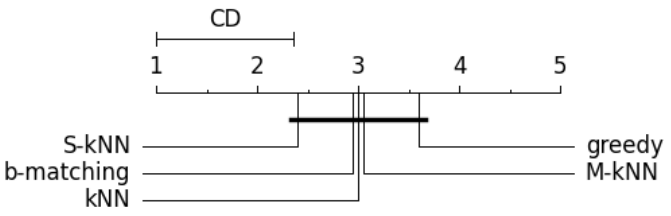


Figura 4.3: Comparação dos métodos de construção de redes pelo teste de Nemenyi considerando as bases de dados artificiais.

ranqueado. Os métodos b-matching, kNN e M-kNN ficaram em posições próximas e Greedy em última posição.

4.1.1.4 Distribuição da acurácia, dos parâmetros e comparação do tempo de execução

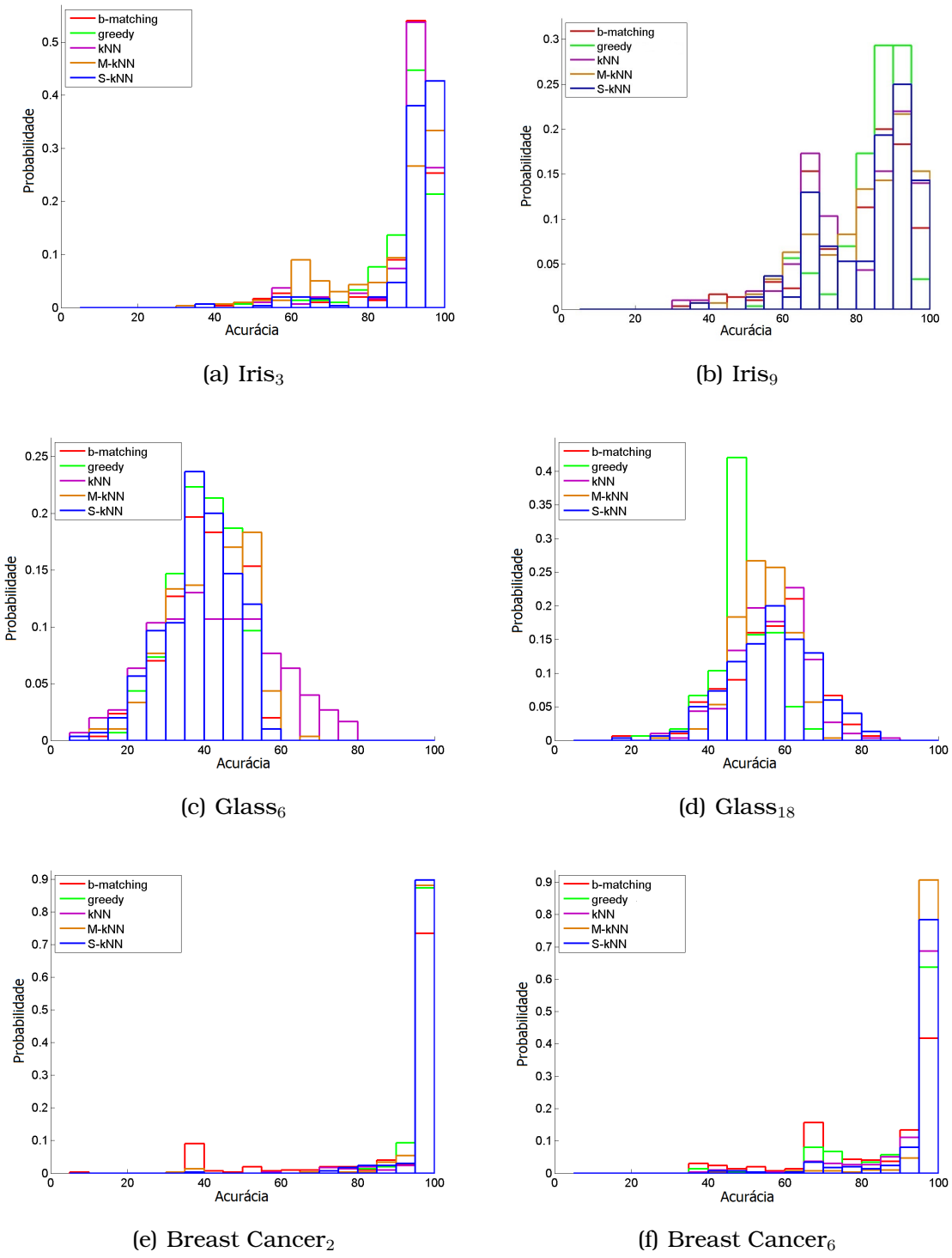
A distribuição da acurácia para as bases de dados foi analisada e três principais varições se destacaram (Figura 4.4). Na base Iris, especialmente para 9 rótulos, observa-se a ocorrência de uma bimodal. Neste caso, existe uma distribuição de rótulos que degrada o valor da acurácia. Na base Glass, observa-se a ocorrência de uma distribuição Gaussiana, e a média pode ser considerada o balanço das frequências. Na base Breast Cancer a distribuição se concentra em um único intervalo. Neste caso, a acurácia é sempre constante independente da configuração da rede ou da seleção dos vértices rotulados.

A fim de analisar quão sensitivo os métodos de geração de redes são em relação aos parâmetros de entrada, plotou-se a frequência dos mesmos considerando a melhor acurácia alcançada em cada parâmetro. Os parâmetros são k (para kNN, M-kNN e S-kNN) e b (para b-matching e Greedy). Na Figura 4.5 estão plotados a distribuição dos mesmos e nota-se que os valores com maior distribuição são menores que 6.

O tempo de execução entre os métodos foi comparado na base USPS com $k = 9$. A Figura 4.6 mostra que os métodos b-matching e Greedy levam um tempo considerável para gerar as redes, enquanto os métodos kNN e S-kNN são mais rápidos, levando menos de 1 segundo para construir as redes. Os algoritmos foram executados em um computador 16 X Intel Xeon E5-2690 de 2.90 GHz e 64GB RAM. O código de b-matching e Greedy estão em C++, enquanto kNN, S-kNN e M-kNN estão em Java.

4.1.1.5 Discussão

A partir dos resultados da classificação, observa-se que nem sempre existe diferença estatística significativa entre os métodos considerados para construção



(a) Iris_3

(b) Iris_9

(c) Glass_6

(d) Glass_{18}

(e) Breast Cancer_2

(f) Breast Cancer_6

Figura 4.4: Histograma de distribuição das taxas de acurácia com os melhores valores de k ou b e diferentes números de vértices rotulados (subescrito) para as bases (a) e (b) Iris, (c) e (d) Glass, (e) e (f) Breast Cancer.

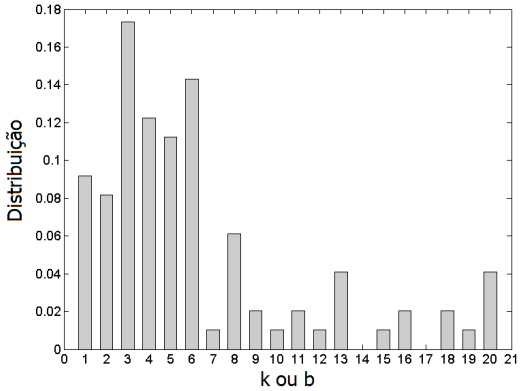


Figura 4.5: Distribuição dos parâmetros k ou b.

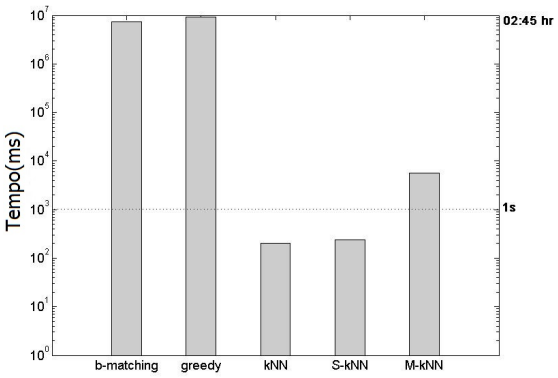


Figura 4.6: Comparação de tempo de execução entre os métodos de construção de redes.

de redes. Entretanto, os métodos para construção de redes regulares ficam melhor ranqueados: b-matching foi melhor nas bases de dados artificiais (com duas classes e duas dimensões), enquanto S-kNN foi melhor nas bases reais (com número de classes e dimensões variando). Entretanto o modo como o método gera a rede regular também é importante, pois Greedy realiza conexões de maneira gulosa e nem sempre obtém bons resultados.

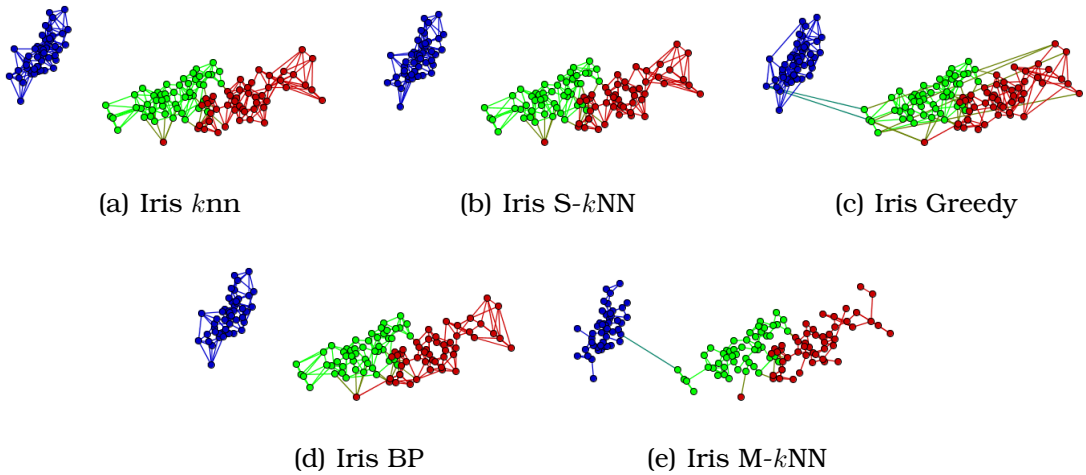
Com relação a distribuição da acurácia, diversos fatores podem influenciá-la, como por exemplo, a topologia da rede, a distribuição/quantidade dos vértices rotulados e os parâmetros k e b. Muitos usuários podem encontrar dificuldades em considerar todos estes fatores nos experimentos. Isso indica a importância de realizar mais pesquisas acerca destas questões, a fim de fornecer um suporte maior na escolha dos métodos e obter melhores resultados na classificação.

Considerando o tempo de execução, b-matching e Greedy podem ser muito custosos quando executados em grandes bases de dados. Essa limitação deve ser considerada mesmo em aplicações *offline*. No trabalho de Ozaki et al. (2011), que trata de processamento de textos, os autores reportam que após uma se-

mana de processamento não foi possível obter resultados utilizando o método b-matching.

4.1.2 Caracterização das redes

Nesta seção é apresentada a caracterização das redes construídas pelos métodos kNN, S-kNN, b-matching via *belief propagation* (BP ou b-matching), b-matching via *greedy approximation* (Greedy) e M-kNN. Algumas medidas de teoria dos grafos e redes complexas foram calculadas a fim de estudar a topologia das redes e sua relação com a acurácia da classificação. Conforme observa-se na Figura 4.7, existem diferenças visuais entre as redes, as quais são confirmadas pelas medidas calculadas.



(a) Iris $k$nn (b) Iris S-kNN (c) Iris Greedy

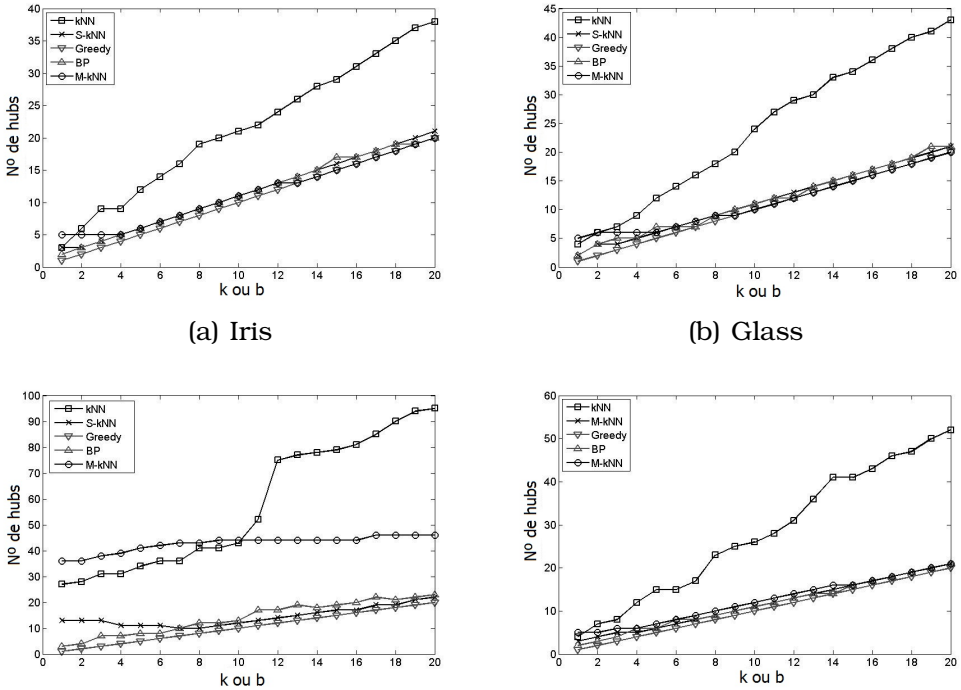
(d) Iris BP (e) Iris M-kNN

Figura 4.7: Visualização das redes para a base Iris, considerando $k = b = 5$.

4.1.2.1 Medidas de redes

Computou-se algumas medidas de teoria dos grafos e redes complexas para caracterizar as redes geradas. Os resultados para quatro bases de dados do UCI (Iris, Glass, Breast Cancer e Ecoli) são apresentados a seguir. Utilizou-se quatro bases a fim de demonstrar que os padrões são similares.

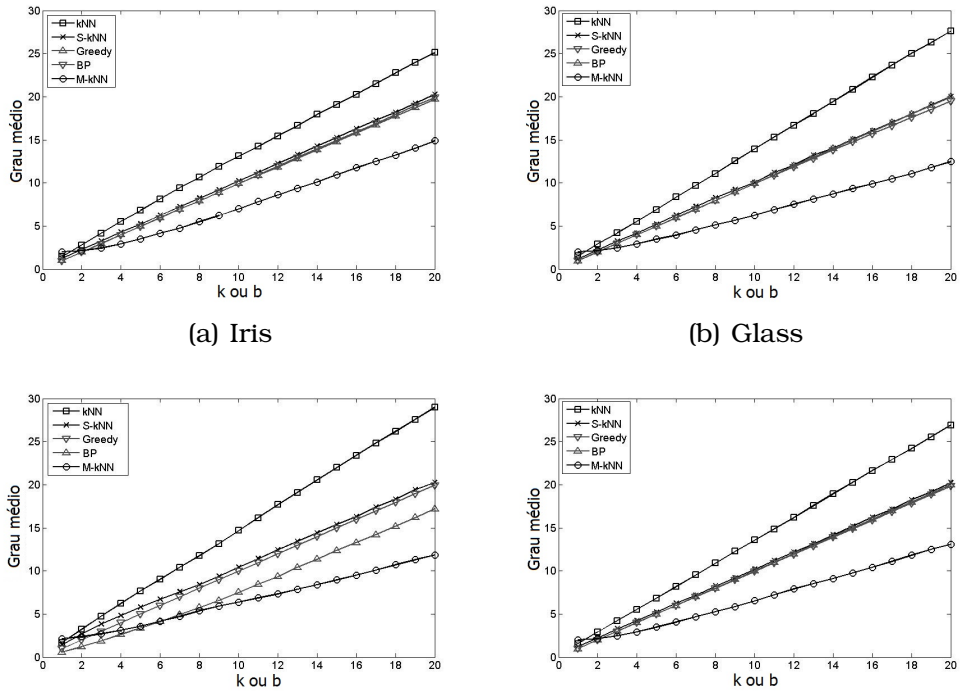
Ozaki et al. (2011) argumentam que *hubs* em uma rede degeneram a acurácia na classificação. A fim de testar quais métodos geram *hubs*, calculou-se o maior grau gerado em cada rede para diferentes valores de k. A Figura 4.8 mostra que o método kNN sempre gera *hubs*. Em alguns casos, M-kNN também gera *hubs* (ver base Breast Cancer). Nenhum método regular gera *hubs*, entretanto a presença de *hubs* não necessariamente influencia a acurácia. Já que em alguns casos o



(a) Iris (b) Glass

(c) Breast Cancer (d) Ecoli

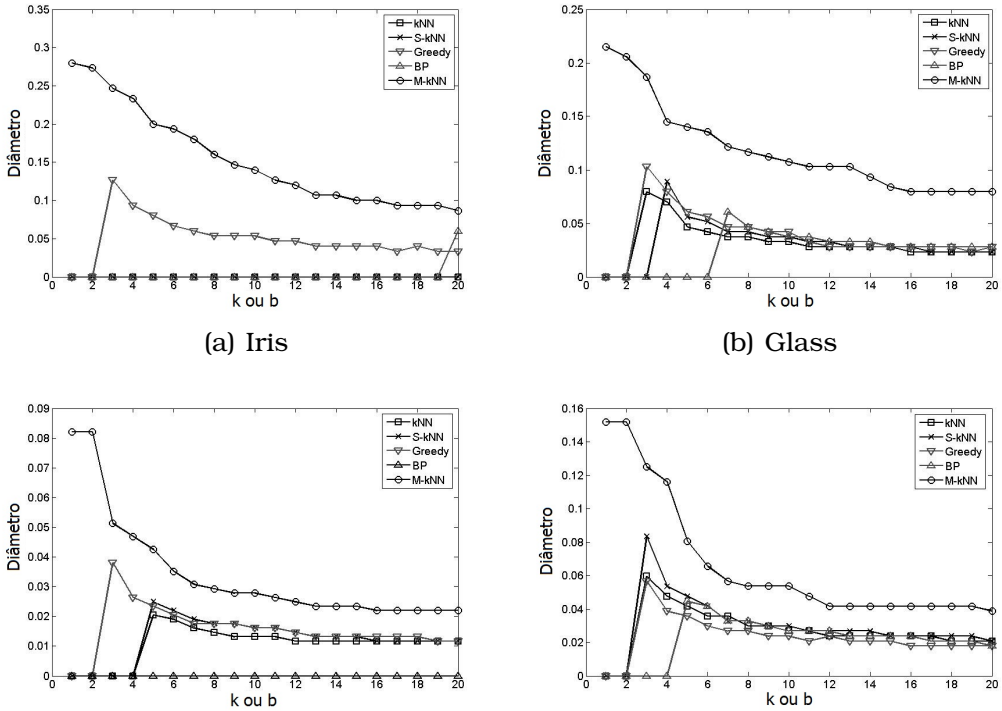
Figura 4.8: Comparação do número de *hubs* para as bases (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli.



(a) Iris (b) Glass

(c) Breast Cancer (d) Ecoli

Figura 4.9: Comparação do grau médio para as bases (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli.



(a) Iris (b) Glass

(c) Breast Cancer (d) Ecoli

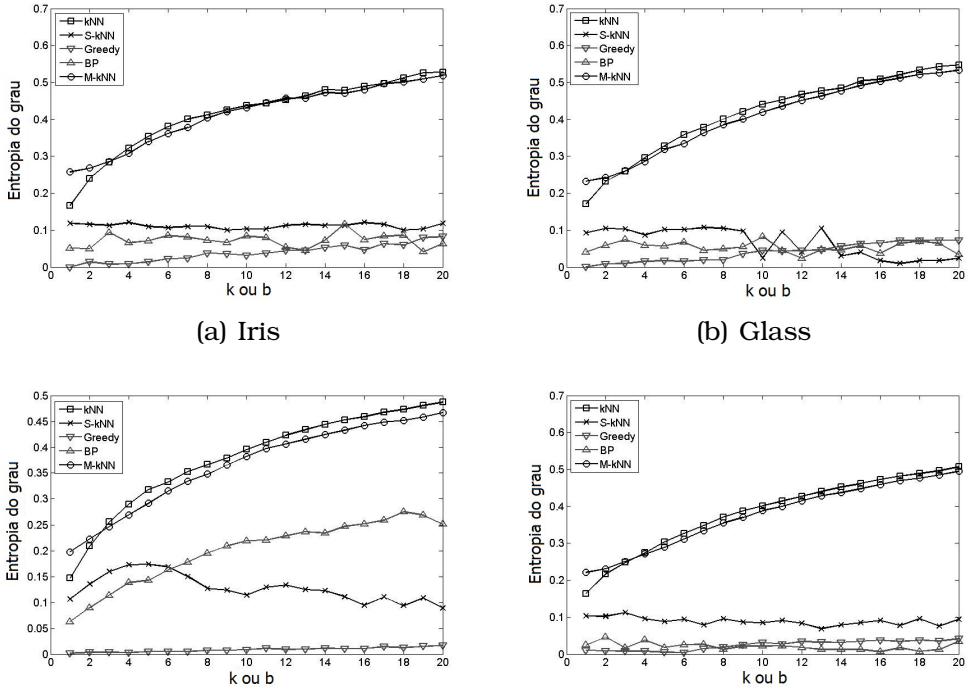
Figura 4.10: Comparação do diâmetro da rede para as bases (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli.

método kNN alcança bons resultados, enquanto o método Greedy não gera *hubs* e não atinge bons resultados nas bases reais.

Os métodos para construção de redes deveriam gerar redes esparsas, permitindo assim melhorar a eficiência, acurácia e robustez a ruído na propagação dos rótulos (Jebara et al., 2009). Devido a isso, outra medida calculada foi *grau médio* (Figura 4.9). Observa-se que o método kNN gera redes com o maior número de arestas e o método M-kNN com o menor. Todos os métodos que geram redes regulares possuem um grau médio próximo a k e b.

As redes deveriam ser conexas, pois os algoritmos de classificação propagam o rótulo através das arestas. Se um componente não contém um vértice rotulado, o algoritmo não pode predizer os rótulos dos vértices contidos neste componente. Calculou-se o diâmetro das redes e os resultados são apresentados na Figura 4.10. Um valor igual a zero indica componentes desconexos. Como o método M-kNN é combinado com o método *minimum spanning tree*, este sempre gera redes conexas e com maior diâmetro. Os demais métodos, especialmente para valores baixos de k e b, geram redes desconexas.

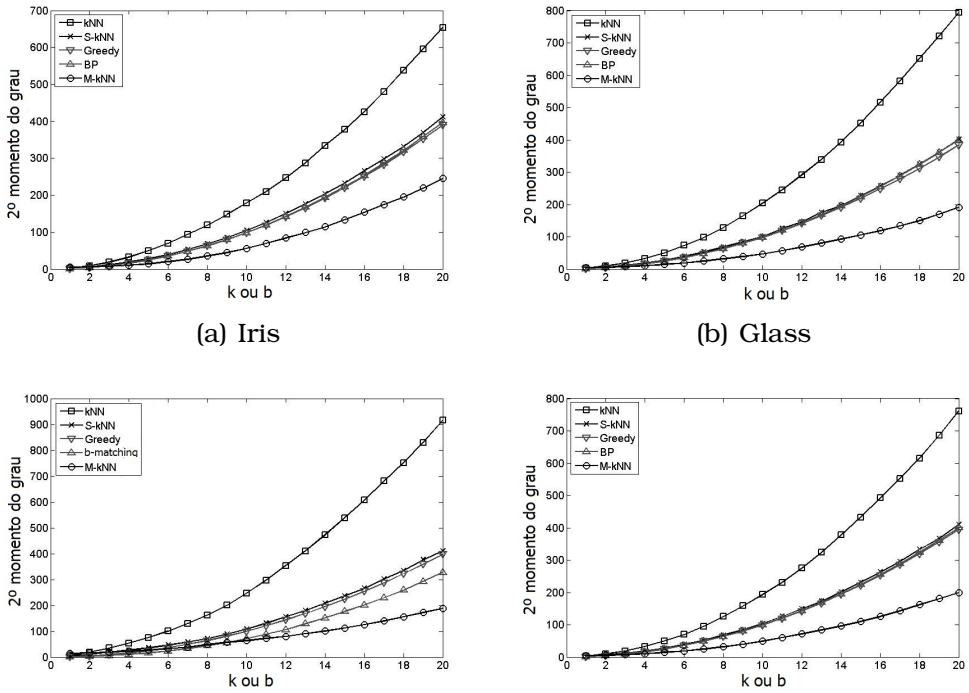
Calculou-se, também, medidas relacionadas a conectividade da rede, como entropia do grau, o segundo momento do grau e complexidade. Conforme ilus-



(a) Iris (b) Glass

(c) Breast Cancer (d) Ecoli

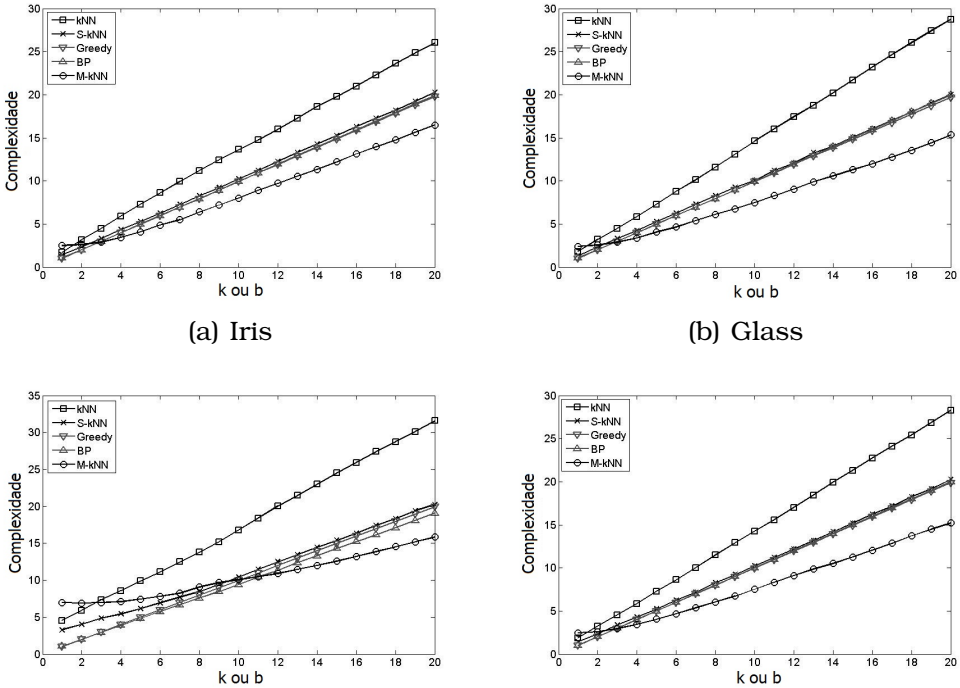
Figura 4.11: Comparação da entropia do grau para as bases: (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli.



(a) Iris (b) Glass

(c) Breast Cancer (d) Ecoli

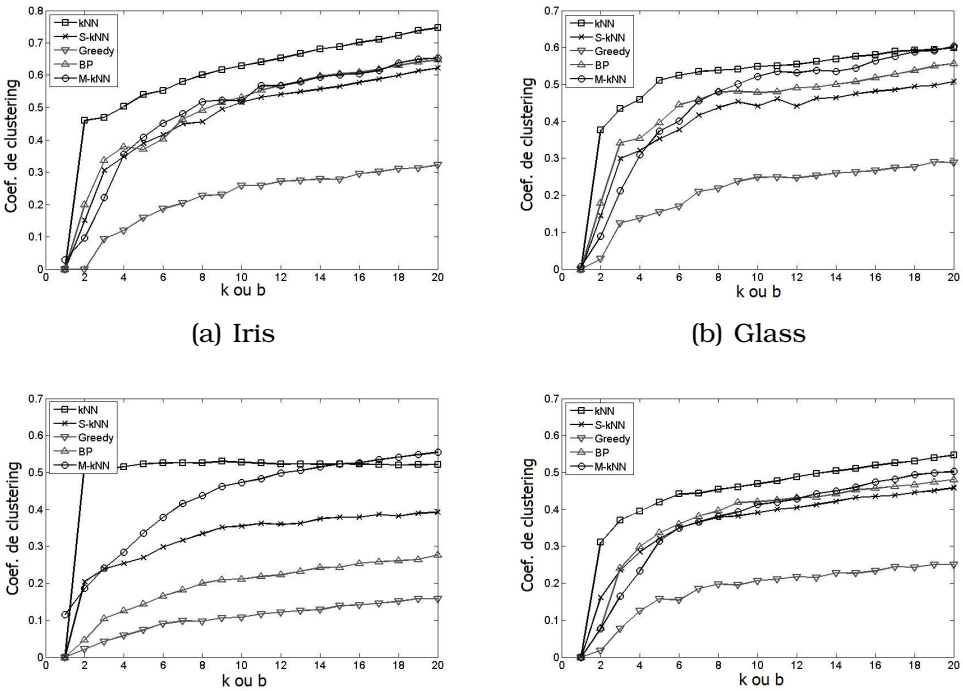
Figura 4.12: Comparação do segundo momento para as bases: (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli.



(a) Iris (b) Glass

(c) Breast Cancer (d) Ecoli

Figura 4.13: Comparação da complexidade para as bases: (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli.



(a) Iris (b) Glass

(c) Breast Cancer (d) Ecoli

Figura 4.14: Comparação do coeficiente de *clustering* para as bases: (a) Iris, (b) Glass, (c) Breast Cancer e (d) Ecoli.

trado na Figura 4.11, as redes regulares possuem valores baixos de entropia, ao contrário das redes kNN e M-kNN. Entre as redes regulares, as redes Greedy apresentam os menores valores, indicando que os vértices são ainda mais similares que os outros métodos que geram redes regulares.

O segundo momento do grau para as redes regulares apresenta uma variância menor (Figura 4.12). Pois todos os elementos possuem grau próximo da média, ou seja, todos possuem graus similares. Ao contrário das redes kNN e M-kNN.

A complexidade da rede indica se a maioria dos vértices são similares ou não. Conforme ilustrado na Figura 4.13, as redes kNN apresentam complexidade maior que as redes regulares. Isso porque as redes kNN possuem distribuição de grau exponencial. E as redes regulares possuem o segundo momento próximo a média do grau.

A medida de coeficiente de *clustering* indica a existência de estruturas locais em uma rede, calculando a existência de triplas de vértices conectados. As redes kNN apresentam os maiores valores para o coeficiente de *clustering*, enquanto as redes Greedy tem os menores, conforme a Figura 4.14. Uma proporção alta de triângulos pode representar ruídos na rede enquanto um valor muito baixo pode indicar pouca estrutura na rede para a propagação de rótulos.

4.1.2.2 Correlação entre acurácia e medidas de redes

A fim de analisar se existe relação entre as medidas calculadas para as redes e a acurácia na classificação, calculou-se as correlações de Pearson e Spearman entre a acurácia e as seguintes medidas: grau médio, presença de *hubs*, complexidade, segundo momento do grau, entropia do grau, assortatividade, coeficiente de *clustering*, média dos menores caminho e raio.

A Tabela 4.4 contém os resultados das correlações de Spearman (ρ_s) e Pearson (ρ_p). Em geral, existe uma correlação moderada (Cohen, 1988), i.e., maior que 0.30 e menor que 0.49 (em negrito). Os métodos b-matching e S-kNN apresentam mais correlações com as medidas calculadas, enquanto os métodos kNN e M-kNN apresentam pouca correlação e o Greedy não apresenta nenhuma correlação. Isso pode acontecer pois o Greedy faz conexões de maneira gulosa gerando características aleatórias na rede, como acontece com modelos para geração de redes aleatórias.

4.1.2.3 Discussão

Os métodos para construção de redes geram redes com diferentes topologias, conforme indicado pelas medidas analisadas. O método kNN gera redes mais

Tabela 4.4: Correlação de Spearman e Pearson para b-matching, Greedy, kNN, S-kNN e M-kNN.

Medidas	b-matching		Greedy		kNN		M-kNN		S-kNN	
	ρ_s	ρ_p	ρ_s	ρ_p	ρ_s	ρ_p	ρ_s	ρ_p	ρ_s	ρ_p
Grau médio	**-0.315**	-0.231	-0.033	-0.228	0.097	0.181	-0.255	-0.083	**-0.403**	-0.209
Hubs	**-0.392**	**-0.389**	-0.025	-0.229	0.059	0.251	-0.210	0.006	**-0.335**	-0.174
Complexidade	**-0.386**	**-0.389**	-0.033	-0.228	0.065	0.189	-0.273	-0.057	**-0.403**	-0.206
2° momento do grau	**-0.393**	-0.261	-0.033	-0.179	0.083	0.218	-0.282	0.054	**-0.403**	-0.111
Entropia do grau	0.263	0.198	0.019	0.283	0.133	0.166	-0.282	-0.204	0.076	0.171
Assortatividade	0.243	0.052	-0.058	-0.040	0.134	0.226	-0.220	**-0.341**	**0.387**	**0.494**
Coef. de *clustering*	**-0.351**	**-0.418**	-0.173	-0.196	0.162	0.248	**-0.383**	**-0.328**	-0.263	-0.240
Média dos caminhos mín.	**0.497**	**0.493**	0.279	0.127	**0.443**	**0.430**	**0.340**	**0.390**	**0.421**	**0.410**
Raio	0.171	**0.350**	0.289	0.271	**0.570**	**0.481**	0.210	**0.336**	**0.375**	**0.342**

densas e com distribuição do grau heterogêneas, conforme evidenciado pelas medidas entropia e coeficiente de *clustering*. Os métodos regulares geram redes com distribuição do grau homogêneas e com topologia similares. O método M-kNN gera redes mais esparsas e com heterogeneidade na distribuição do grau. Apesar de duas redes apresentarem a mesma topologia, a maneira que as conexões foram geradas também pode influenciar o desempenho da classificação.

Nas correlações entre medidas da rede e acurácia, nenhuma correlação alta emergiu. Alguns métodos, especialmente o b-matching e o S-kNN apresentaram correlação negativa moderada entre a acurácia e medidas da rede relacionadas com o grau, tais como grau médio, presença de *hub*, complexidade e segundo momento do grau. Isso indica que quanto menor o valor dessas medidas mais alta é a acurácia. Além disso, todos os métodos tiveram correlação moderada com a média dos caminhos mínimos e o grau, indicando que uma melhor acurácia é atingida quando a rede é esparsa. Observou-se que medidas relacionadas ao grau são relacionadas com a acurácia em certo nível, desse modo redes com baixo grau médio e com distribuição de grau homogênea mostraram-se mais indicadas para a classificação semissupervisionada baseada em redes.

4.1.3 Aplicação do método S-kNN na classificação de gêneros musicais

A Internet tornou-se um grande repositório de músicas. Os usuários que compram ou baixam músicas de coleções *online* normalmente formulam suas preferências em termos de gênero. Porém, nem todas as músicas estão classificadas por gênero. Perrot e Gjerdigen (1999) dizem que uma pessoa leiga no assunto pode detectar o gênero ouvindo três segundos de uma música, com uma acúrácia de 72%. Porém, se considerarmos uma grande coleção, levaria muito tempo ouvindo e a pessoa ficaria cansada, diminuindo a percepção. Este

processo é muito árduo para ser feito manualmente e depende da experiência e tempo disponível do usuário. Por isso, é necessário o desenvolvimento de algoritmos computacionais para a classificação automática, o que ajuda na organização e classificação das músicas (Poria et al., 2013).

A classificação automática envolve duas tarefas: extração das características e aplicação de classificadores sobre estas características. A primeira tarefa relaciona-se com a taxonomia que será usada na classificação, por exemplo: instrumentos, duração, gênero, entre outras. O gênero é a taxonomia mais comum. Na literatura existem alguns trabalhos que exploram esta tarefa: Shao et al. (2004) usam *clustering* hierárquico aglomerativo para determinar o gênero musical, Pampalk et al. (2005) usam um classificador kNN, Mandel e Ellis (2005), Lidy e Rauber (2005), e Scaringella e Mlynek (2005) usam classificadores SVM, Xu et al. (2005) usam *co-training* baseado em características "multi-visão", Yaslan e Cataltepe (2009) exploram um método de subespaço aleatório para o co-training, Poria et al. (2013) usam SVM em uma abordagem semissupervisionada.

Como os trabalhos da literatura utilizam em geral algoritmos tradicionais, propôs-se utilizar algoritmos relacionais para a classificação de gêneros musicais (Valverde-Rebaza et al., 2014). Esse trabalho contou com a colaboração de pesquisadores do laboratório VICG-ICMC, especificamente a aluna Aurea Soriano Vargas e sua orientadora, que possuem experiência com exploração de características musicais. Mostrou-se que esta metodologia obtém melhores resultados que outras abordagens supervisionadas, tais como *Decision Trees*, *Naive Bayes*, *Neural Networks* e *Support Vector Machines*. A representação relacional pode explorar as relações entre os elementos, além da informação dos atributos. Explorou-se ainda, três características musicais: momentos, medido pela distância Euclidiana, histograma e estrutura (Soriano, 2013), medidos pela função DTW (Niels, 2004). Aplicou-se três métodos de construção de redes para os algoritmos relacionais: kNN, M-kNN e S-kNN (Vega-Oliveros et al., 2014). Mostrou-se, a partir disso, qual combinação de característica, técnica de construção de redes e classificador relacional obtém o melhor desempenho na tarefa de classificação de gêneros musicais.

Considerou-se descritores musicais no formato MIDI (*Musical Instrument Digital Interface*) (Casabona e Frederick, 1988), o qual permite reconstruir uma representação da música como um *score* simbólico. Em geral o formato MIDI é preferido por usuários músicos ou que buscam aplicações interativas, por ser mais conveniente para extrair informações de alto nível de uma música (Mckay, 2004).

A seguir são apresentados as características musicais exploradas, os algoritmos relacionais utilizados e os resultados experimentais em uma base de dados com 919 trilhas de áudio, além da acurácia obtida e teste estatístico de significância.

4.1.3.1 Características musicais

Uma das característica exploradas foi o *histograma das notas*, que é útil para representar sinais musicais tanto no formato simbólico como acústico (Tzanetakis et al., 2002). Um histograma contém 12 possíveis notas musicais. Os arquivos musicais foram explorados no formato MIDI, que permite 128 notas (entre 0 e 127), o que diferencia duas notas iguais em diferentes oitavas. No trabalho, foi utilizada uma escala de 12 diferentes notas obtidas por meio do operador módulo (%12 do total de notas).

Momentos estatísticos também podem ser úteis para capturar características globais de uma música (Lartillot et al., 2007). Uma abordagem consiste em quantificar a velocidade do conteúdo e as notas musicais. Esta abordagem provê uma aproximação da percepção humana e permite diferenciar ritmo e melodia. Neste trabalho, utilizou-se a média (primeiro momento), o desvio padrão (segundo momento), a uniformidade (terceiro momento) e a entropia (quarto momento) como momentos estatísticos. Os quais foram aplicados no histograma de notas e no histograma de velocidades.

Para identificar tonalidade e acorde, *características estruturais* foram extraídas dos arquivos MIDI segundo a abordagem proposta por Soriano (2013). Esta abordagem segmenta a música em compassos, após isso, identifica os tons considerando a progressão da música e, depois, a harmonia é usada para obter acordes. Para identificar padrões, o algoritmo *Horspool* é aplicado na sequência de acordes, gerando subconjuntos de acordes de tamanho $\frac{N}{2}$, onde N é o número total de acordes. Este processo iterativo vai decrementando em 1 o conjunto de acordes, gerando uma lista de padrões. A partir desta lista, um vetor de características é gerado, com números que representam o tamanho de cada padrão.

4.1.3.2 Classificadores relacionais

No estudo realizado considerou-se quatro classificadores relacionais que utilizam inferência coletiva: *probabilistic relational neighbor* (prn), *weighted vote relational neighbor* (wvrn), *network-only Bayes* (no-Bayes) e *network-only link-based* (no-lb). O classificador prn estima a probabilidade da classe assumindo

Tabela 4.5: Distribuição dos gêneros musicais para a base de dados utilizada. Adaptada de Valverde-Rebaza et al. (2014).

Gênero	N^o trilhas
clássica	31
sertanejo	243
poprock	550
jazz	95
Total	919

que o rótulo de um vértice depende apenas de seus vizinhos imediatos, e vértices conectados têm uma propensão de pertencer a mesma classe (Macskassy e Provost, 2003). Do mesmo modo, os classificador wvrn estima a probabilidade de uma classe, assumindo que vértices conectados possuem uma propensão de pertencer a mesma classe, e usam a média do peso da probabilidade da classe com base na vizinhança de cada vértice analisado (Macskassy e Provost, 2007b). O classificador no-Bayes usa classificação *naive Bayes multinomial* baseada na classe da vizinhança de cada vértice (Macskassy e Provost, 2007b).

Já o classificador no-lb cria um vetor de características para um vértice agregando os rótulos de seus vizinhos, e depois usa regressão logística para construir um modelo discriminativo baseado no vetor de características (Lu e Getoor, 2003). Utilizou-se três métodos de agregação: *binary-link* (no-lb-binary), *mode-link* (no-lb-mode) e *count-link* (no-lb-count), proposto por Lu e Getoor (2003). Outro método de agregação é o *class-distribution-link* (no-lb-distrib), proposto por Macskassy e Provost (2007b).

4.1.3.3 Base de dados

A coleção de músicas utilizadas foi proposta por Soriano (2013) . Ela consiste em 919 trilhas de áudio, em formato MIDI e classificadas em quatro gêneros: clássica, sertanejo, poprock e jazz. As classes são desbalanceadas, como pode ser visto na Tabela 4.5.

4.1.3.4 Configuração dos experimentos

Para execução dos experimentos, primeiramente foi realizado o pré-processamento das músicas. Para extração das características musicais utilizou-se: histograma das notas, momentos estatísticos e características estruturais. A partir do histograma de notas obteve-se uma base formada por vetores de características com 12 atributos cada. A partir dos momentos estatísticos, obteve-se uma base formada por vetores de características com 8 atributos cada. A partir das carac-

terísticas estruturais, obteve-se uma base formada por vetores de características estruturais com número variado de atributos (em geral 250).

Depois disso, para cada base, gerou-se três tipos de redes utilizando as técnicas kNN, M-kNN e S-kNN. Para cada técnica variou-se o parâmetro k entre 1 e 15. Para as bases geradas por momentos estatísticos, o peso das arestas considerado foi a distância Euclidiana entre os vértices. Para as bases geradas por histograma de notas e características estruturais considerou-se a distância DTW, a qual é bastante utilizada para medir dissimilaridade em séries temporais, pois contabiliza deslocamentos de tempo e série de diferentes tamanhos (Sakoe e Chiba, 1978).

Para o processo de classificação, aplicou-se abordagens tradicionais e relacionais. Os algoritmos tradicionais utilizados foram: *decision tree* (J48), *naive Bayes* (NB), *multilayer perceptron with backpropagation* (MLP) e *support vector machine* (SMO) disponíveis no Weka[1] com a configuração padrão. Os algoritmos relacionais utilizados foram: *weighted vote relational neighbor* (wvrn), *network-only Bayes* (no-Bayes), *probabilistic relational neighbor* (prn) e *network-only link-based* (no-lb) disponíveis em Netkit-SRL[2] com configuração padrão. Para os classificadores *network-only link-based* utilizou-se: *mode-link* (no-lb-mode), *count-link* (no-lb-count), *binary-link* (no-lb-binary) e *class-distribution-link* (no-lb-distrib). Em todos os casos utilizou-se *10-fold cross validation*.

4.1.3.5 Resultados

Para avaliação dos resultados considerou-se a AUC (*area under the receiver operating characteristic curve*). A Tabela 4.6 contém a média da AUC para os classificadores tradicionais e as Tabelas 4.7, 4.8 e 4.9 contém a média da AUC para os classificadores relacionais considerando as redes kNN, M-kNN e S-kNN respectivamente. Para os classificadores relacionais, apenas o maior valor de AUC obtido é apresentado, considerando as diferentes redes construídas para cada base de dados utilizando diferentes valores de k. O valor de k aparece entre parênteses. Em todas as tabelas os melhores resultados para cada classificador aparece em negrito e os melhores resultados para cada característica musical aparece com a célula da tabela em cinza.

A partir destes resultados observa-se o impacto das características musicais, dos classificadores e dos métodos de construção de redes, na acurácia da classificação.

[1] http://www.cs.waikato.ac.nz/ml/weka/
[2] http://netkit-srl.sourceforge.net/index.html

Tabela 4.6: AUC para os classificadores tradicionais. Adaptada de Valverde-Rebaza et al. (2014).

	J48	NB	MLP	SMO
histograma	0.619	0.607	0.665	0.506
momentos	0.706	0.750	0.771	0.585
estrutura	**0.738**	**0.920**	**0.816**	**0.724**

Tabela 4.7: AUC para os classificadores relacionais nas redes kNN. Adaptada de Valverde-Rebaza et al. (2014).

	no-lb-mode	no-lb-count	no-lb-binary	no-lb-distrib	wvrn	no-Bayes	prn
histograma	0.575 (k=11)	0.723 (k=9)	0.622 (k=2)	0.71 (k=8)	0.712 (k=9)	0.515 (k=1)	0.537 (k=1)
momentos	0.547 (k=5)	0.635 (k=13)	0.575 (k=8)	0.644 (k=7)	0.644 (k=9)	0.563 (k=2)	0.571 (k=3)
estrutura	**0.834 (k=7)**	**0.939 (k=14)**	**0.851 (k=4)**	**0.945 (k=14)**	**0.931 (k=15)**	**0.922 (k=15)**	**0.903 (k=9)**

Tabela 4.8: AUC para os classificadores relacionais nas redes M-kNN. Adaptada de Valverde-Rebaza et al. (2014).

	no-lb-mode	no-lb-count	no-lb-binary	no-lb-distrib	wvrn	no-Bayes	prn
histograma	0.621 (k=1)	0.712 (k=10)	0.626 (k=2)	0.735 (k=12)	0.727 (k=13)	0.555 (k=1)	0.571 (k=1)
momentos	0.570 (k=1)	0.657 (k=14)	0.588 (k=2)	0.633 (k=15)	0.630 (k=14)	0.578 (k=1)	0.574 (k=1)
estrutura	**0.864 (k=1)**	**0.955 (k=14)**	**0.818 (k=2)**	**0.963 (k=15)**	**0.964 (k=14)**	**0.913 (k=6)**	**0.902 (k=2)**

Tabela 4.9: AUC para os classificadores relacionais nas redes S-kNN. Adaptada de Valverde-Rebaza et al. (2014).

	no-lb-mode	no-lb-count	no-lb-binary	no-lb-distrib	wvrn	no-Bayes	prn
histograma	0.608 (k=1)	0.724 (k=8)	0.611 (k=2)	0.737 (k=12)	0.730 (k=12)	0.544 (k=1)	0.553 (k=1)
momentos	0.569 (k=1)	0.652 (k=13)	0.571 (k=1)	0.620 (k=8)	0.625 (k=6)	0.560 (k=1)	0.565 (k=1)
estrutura	**0.904 (k=1)**	**0.948 (k=11)**	**0.82 (k=1)**	**0.967 (k=15)**	**0.966 (k=15)**	**0.923 (k=3)**	**0.904 (k=2)**

Com relação as características musicais, nos classificadores tradicionais (Tabela 4.6) nota-se que todos obtiveram melhores resultados para a característica estrutural. Para a característica histograma e momentos, o classificador MLP obteve maior acurácia na classificação se comparado aos demais. Para os classificadores relacionais (Tabelas 4.7, 4.8 e 4.9) todos obtiveram melhores resultados usando a característica estrutural. A característica histograma gerou melhor acurácia no classificador no-lb-count, para as redes kNN, e no classificador no-lb-distrib, para as redes M-kNN e S-kNN. Já a característica momentos gerou melhor acurácia no classificador no-lb-distrib, para as redes kNN, e no classificador no-lb-count, para as redes M-kNN e S-kNN.

Para determinar o melhor classificador aplicou-se o teste de Freidman e *post-hoc* de Nemenyi (Demšar, 2006). A Figura 4.15 mostra o resultado para os classificadores tradicionais. A diferença crítica (*critical difference* - CD) no percentil 95 é 2.71. No eixo é mostrado os *ranks* médios dos classificadores. Neste eixo os menores (melhores) *ranks* estão no lado esquerdo. Os classificadores analisados que não apresentam diferença significativa estão conectados por uma linha preta. Observa-se que nesse caso não houve diferença significativa e os classificadores MLP e NB ficaram em primeiro e segundo lugar no *rank*.

As Figuras 4.16, 4.17 e 4.18 contém os resultados para os classificadores relacionais nas redes kNN, M-kNN e S-kNN, respectivamente. A CD para todos os testes *post-hoc* no percentil 95 é 5.20. Não houve diferença estatística significativa entre os classificadores e os métodos no-lb-distrib, no-lb-count e wvrn obtiveram os três melhores *ranks*.

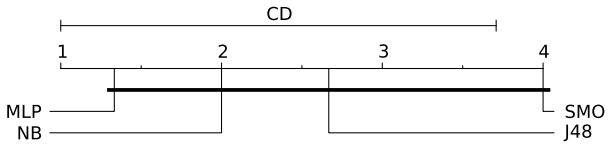


Figura 4.15: Resultado do teste de Nemenyi para os classificadores tradicionais. Obtida em Valverde-Rebaza et al. (2014).

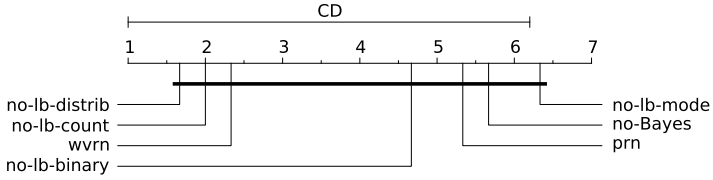


Figura 4.16: Resultado do teste de Nemenyi para os classificadores relacionais, considerando as redes kNN. Obtida em Valverde-Rebaza et al. (2014).

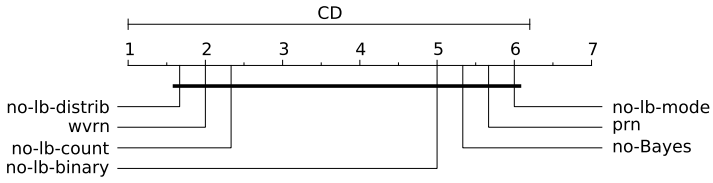


Figura 4.17: Resultado do teste de Nemenyi para os classificadores relacionais, considerando as redes M-kNN. Obtida em Valverde-Rebaza et al. (2014).

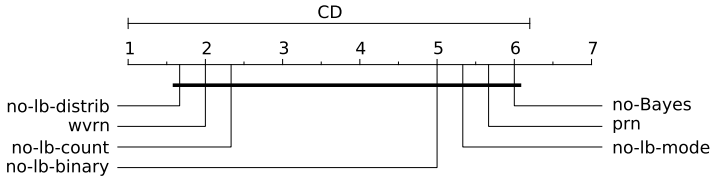


Figura 4.18: Resultado do teste de Nemenyi para os classificadores relacionais, considerando as redes S-kNN. Obtida em Valverde-Rebaza et al. (2014).

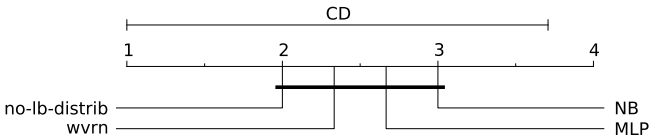


Figura 4.19: Resultado do teste de Nemenyi para os melhores classificadores tradicionais e relacionais (considerando as redes S-kNN). Obtida em Valverde-Rebaza et al. (2014).

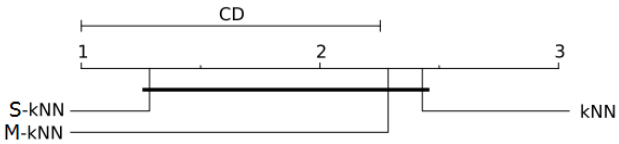


Figura 4.20: Resultado do teste de Nemenyi para identificar o melhor método de construção de redes. Obtida em Valverde-Rebaza et al. (2014).

Compararam-se os dois melhores classificadores tradicionais (MLP e NB) com os dois melhores classificadores relacionais (no-lb-distrib e wrn) considerando as redes S-kNN. A Figura 4.19 mostra o resultado do teste *post-hoc* de Nemenyi. O valor de CD no percentil 95 é 2.71. Não se observa diferença significativa, porém os classificadores relacionais ficam melhor ranqueados.

Finalmente, comparou-se os métodos para construção de redes. Para isso, calculou-se a média dos melhores resultados (valores em negrito das Tabelas 4.7, 4.8 e 4.9) e analisou-se o resultado estatisticamente. A Figura 4.20 mostra o resultado do teste *post-hoc* de Nemenyi. O valor de CD para o percentil 95 é 1.25. Não houve diferença estatística, entretanto o método S-kNN ficou melhor ranqueado.

Analisou-se a variação da AUC para diferentes valores de k nas redes S-kNN (Figura 4.21). Em geral, todos os classificadores mantêm um valor constante da acurácia para $k > 3$. Apenas os classificadores no-lb-mode, no-lb-binary e prn têm o desempenho reduzido significatimente na característica estrutura conforme o valor de k aumenta.

4.1.3.6 Discussão

Nesta seção foi apresentada uma aplicação do método S-kNN em classificação de gêneros musicais (Valverde-Rebaza et al., 2014). Utilizaram-se três medidas de extração de características de arquivos músicais no formato MIDI: histograma, momentos e estrutura gerando três bases de dados. A partir dessas bases analisou-se o impacto dessas características na classificação de gêneros músicais. Além disso, analisou-se o desempenho de classificadores tradicionais

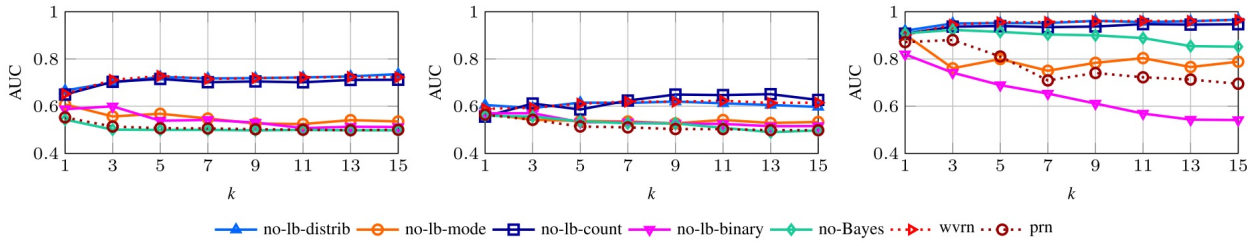


Figura 4.21: Desempenho dos classificadores relacionais para as redes S-kNN e diferentes valores de k. A primeira figura representa a característica histograma, a segunda momentos e a terceira estrutura. Obtida em Valverde-Rebaza et al. (2014).

e relacionais, neste caso, utilizando redes construídas a partir de três técnicas diferentes: kNN, M-kNN e S-kNN, com diferentes valores de k.

Os resultados indicam que a melhor característica foi estrutura, a qual proporcionou melhores resultados tanto para classificadores tradicionais como relacionais. Com relação ao desempenho dos classificadores não houve diferença estatística entre eles, porém os classificadores relacionais ficaram melhor ranqueados. Em geral, o classificador no-lb-distrib foi o melhor entre os relacionais e o classificador MLP foi o melhor entre os tradicionais.

Com relação aos métodos de construção de redes, o S-kNN obteve bons resultados. Conclui-se que dependendo das características das bases de dados, um método de construção de redes pode ser melhor que outro. O que corrobora a necessidade de estudo de técnicas de construção de redes para algoritmos relacionais.

4.2 Análise e aplicação das redes baseadas nas instâncias rotuladas

A fim de estudar se os métodos propostos GBILI e RGCLI, que consideram instâncias rotuladas para a construção da rede, são adequados para a classificação semissupervisionada, realizaram-se diversas análises. As principais contribuições do método proposto são destacadas a seguir.

- O método GBILI apresenta boa acurácia na classificação, alcançando resultados melhores que o método kNN. Pelo teste estatístico de Nemenyi (Demšar, 2006), o algoritmo GBILI é melhor ranqueado que o kNN. Os resultados da acurácia na classificação foram medidos em bases de dados propostas por Chapelle et al. (2010) para o aprendizado semissupervisionado.

- Uma análise de sensibilidade do parâmetro k, variando os valores de 1 a 50, mostra que o GBILI, para $k > 10$, apresenta uma estabilidade na acurácia da classificação. Como a seleção de parâmetros é um problema para muitos métodos, o algoritmo proposto apresenta vantagem neste ponto.

- Mostrou-se pela medida ϕ-*Edge Ratio*, que quando o parâmetro k aumenta, o número de arestas conectando vértices com rótulos diferentes também aumenta nas redes kNN, resultando na propagação errada de informações de rótulo, o que não ocorre no GBILI, já que conforme o valor de k aumenta GBILI permanece com um número de arestas constante.

- O método GBILI torna os vértices rotulados *a priori* em *hubs*. Isso é indicado calculando-se medidas de centralidade de redes complexas como: grau do vértice, *betweenness*, *eigenvector* e *pageRank*. Estas medidas são relacionadas ao processo de difusão em redes, como difusão de informação. Como os pontos rotulados tornam-se *hubs* na rede GBILI, isso facilita a propagação dos rótulos.

- Um exemplo de aplicação do método RGCLI foi em segmentação interativa de imagens. Observa-se que o RGCLI é especialmente adequado para cenários interativos, onde a disponibilização dos rótulos pelo usuário pode facilitar o processo de propagação.

A seguir são apresentados os experimentos para a classificação semissupervisionada, análises das redes geradas e a aplicação de RGCLI em segmentação interativa de imagens.

4.2.1 Classificação semissupervisionada

Nesta seção são apresentados diversos resultados explorando o método GBILI. Na Seção 4.2.1.1 é apresentada a descrição das bases utilizadas para a classificação; na Seção 4.2.1.2 são apresentados detalhes do pré-processamento dos experimentos; na Seção 4.2.1.3 são mostrados os resultados da classificação e do teste estatístico; na Seção 4.2.1.4 são mostrados o comportamento da acurácia e desvio padrão para k variando de 1 a 50; na Seção 4.2.1.5 é comparada a medida ϕ-*edge ratio* entre as redes kNN e GBILI; na Seção 4.2.2 são apresentadas algumas medidas de centralidade de redes complexas usadas para carecterizar as redes geradas pelo método GBILI.

4.2.1.1 Bases de dados

Os experimentos foram realizados com seis bases de dados descritas na Tabela 4.10. Estas bases são frequentemente utilizadas na literatura no aprendizado semissupervisionado e foram propostas por Chapelle et al. (2010). As três primeiras foram criadas artificialmente e as demais foram derivadas de dados reais.

Tabela 4.10: Descrição das bases de dados.

Base de dados	# Instâncias	# Atributos	# Classes	Comentário
Digit_1	1500	241	2	artificial e balanceada
g241c	1500	241	2	artificial e balanceada
g241n	1500	241	2	artificial e balanceada
COIL_2	1500	241	2	real e balanceada
COIL_6	1500	241	6	real e balanceada
USPS	1500	241	2	real e desbalanceada

4.2.1.2 Configuração dos experimentos

Inicialmente, aplicou-se *Principal Component Analysis* (PCA) em todas as bases de dados reduzindo as dimensões para 50. Pois no espaço multi-dimensional dos dados, a distância ao vizinho mais próximo se aproxima a distância do vizinho mais distante, degenerando a qualidade da rede. Depois, executou-se os experimentos utilizando 10 e 100 vértices rotulados aleatoriamente selecionados a partir de todo o conjunto de dados. Isto representa $0,66$ e $6,66$ por cento de vértices rotulados, respectivamente.

Para a construção das redes utilizaram-se os métodos kNN e GBILI. Ambos os métodos operam sobre a matriz de distância $D \in \Re^{n \times n}$, obtida da distância Euclidiana entre os pontos. Este processo resulta em uma matriz $\widehat{A}$ onde $\widehat{A}_{ij} = 1$ se um ponto j possuir uma conexão com o ponto i e $\widehat{A}_{ij} = 0$ caso contrário. Finalmente, esta matriz é simetrizada como segue $A_{ij} = max(\widehat{A}_{ij}, \widehat{A}_{ji})$. Para gerar a matriz de pesos W considerou-se a abordagem binária, onde $W = A$.

O algoritmo utilizado para o processo de inferência dos rótulos foi o *Local and Global Consistency* (LGC) (Zhou et al., 2004), o qual é frequentemente utilizado na literatura (Jebara et al., 2009), (Ozaki et al., 2011). Este algoritmo emprega a rede G e a matriz W para difundir os rótulos conhecidos Y_l a todos os vértices não rotulados. O resultado da média da acurácia na classificação sobre 30 execuções é utilizado como medida de avaliação.

Comparou-se o resultado da classificação com os resultados apresentados por Chapelle et al. (2010) para os seguinte algoritmos: 1-NN, Discrete Reg., Transductive SVM, Cluster Kernel, Low-density separation (LDS), Laplacian RLS.

Tabela 4.11: Média da acurácia com 10 pontos rotulados. Obtida em (Berton e de Andrade Lopes, 2014).

	g241c	**g241n**	**digit$_1$**	**COIL$_6$**	**USPS**
1-NN	56	56.78	76.53	34.09	80.18
Discrete Reg.	50.41	50.95	87.36	36.62	83.93
TSVM	75.29	49.92	82.23	32.5	74.5
Cluster-Kernel	51.72	57.95	81.27	35.68	80.59
LDS	71.15	49.37	84.37	38.1	82.43
Laplacian RLS	56.05	54.32	94.56	45.46	81.01
kNN+LGC	55.16	51.69	**89.61**	41.59	83.54
GBILI+LGC	**56.96**	**56.81**	84.37	39.95	**85.07**

Tabela 4.12: Média da acurácia com 100 pontos rotulados. Obtida em (Berton e de Andrade Lopes, 2014).

	g241c	**g241n**	**digit$_1$**	**COIL$_6$**	**USPS**
1-NN	59.72	62.51	93.88	76.73	92.36
Discrete Reg.	56.35	58.35	97.23	90.39	95.32
TSVM	81.54	77.58	93.85	74.2	90.23
Cluster-Kernel	86.51	95.05	96.21	78.01	90.32
LDS	81.96	76.26	96.54	86.28	95.04
Laplacian RLS	75.64	73.54	97.08	88.08	95.32
kNN+LGC	59.29	57.81	**95.6**	**82.9**	85.07
GBILI+LGC	**61.77**	**66.95**	**95.6**	**82.9**	**94.1**

Destaca-se que foram utilizados 10 e 100 dados rotulados, aleatoriamente selecionados a partir de todo o conjunto de dados, enquanto Chapelle et al. (2010) divide o conjunto em 12 partições e seleciona 10 e 100 pontos rotulados destas partições.

4.2.1.3 Resultado da classificação

Nas Tabelas 4.11 e 4.12 estão a acurácia dos diferentes métodos para 10 e 100 dados rotulados para as bases g241c, g241, digit$_1$, COIL$_6$ e USPS.

Executou-se o teste estatístico de Freidman e post-hoc de Nemenyi (Demšar, 2006) para verificar se existe diferença estatística significativa entre os algoritmos, a partir dos resultados apresentados nas Tabelas 4.11 e 4.12. De acordo com o teste de Nemenyi, o valor crítico para comparar o *rank* médio de dois algoritmos diferentes no percentil 95 é 3.32. A análise é mostrada na Figura 4.22. Observa-se a diferença crítica (*critical difference* - CD) no topo do diagrama. No eixo do diagrama indicam-se os *ranks* médios das medidas. Neste eixo, os menores (melhores) *ranks* encontram-se no lado esquerdo. Os métodos analisados não apresentam diferença significativa, por isso são conectados por uma linha preta no diagrama. Observa-se, no entanto, que GBILI está melhor ranqueado que o método kNN.

4.2.1.4 Análise de sensibilidade dos parâmetros

Como não se conhece *a priori* qual o melhor valor para o parâmetro k para construir a rede, testou-se valores de 1 a 50 para analisar quão sensível a este parâmetro de entrada são os métodos de geração de redes kNN e GBILI.

A Figura 4.23 mostra a acurácia dos métodos kNN e GBILI aplicados as bases: digit$_1$, g241c, g241n, COIL$_2$, COIL$_6$ e USPS, utilizando 10 exemplos rotulados. A Figura 4.24 mostra a acurácia dos métodos aplicadas nas mesmas bases mas utilizando 100 exemplos rotulados.

A partir desses experimentos observa-se que o método kNN normalmente alcança bons resultados para valores de k menores que 10, pois valores altos de k tornam a rede mais densa. Já o método GBILI alcança melhores resultados para valores de k maiores que 10, pois faz uso do kNN mútuo, o qual se conecta com menos vizinhos que o kNN. Desse modo, é necessário um valor maior de k para alcançar mais vizinhos. Nas redes GBILI a acurácia estabiliza e pode-se utilizar um valor de k fixo maior que 10, o que é uma grande vantagem do método.

Em geral, o GBILI apresenta menor desvio padrão que kNN. Quanto maior o seu valor, maior é o grau de heterogeneidade dos casos vis-à-vis o valor da média. Quanto menor, mais homogênea é a distribuição dos casos em torno do termo médio.

4.2.1.5 Análise de densidade das redes

A densidade da rede também pode afetar a acurácia da classificação, já que muitos algoritmos baseados em redes propagam o rótulo na rede. Arestas conectando vértices com rótulos diferentes podem levar a erros na classificação. Utilizou-se a medida ϕ-*edge ratio* como Ozaki et al. (2011) para avaliar a qualidade da rede. Os autores definem ϕ-*edge* de uma rede rotulada (G, y) como qualquer aresta (v_i, v_j) na qual $y_i \neq y_j$, e ϕ-*edge ratio* de uma rede como ϕ-*edge* dividido pelo número total de arestas da rede. Segundo essa medida, quanto menor o valor de ϕ-*edge ratio* melhor a rede.

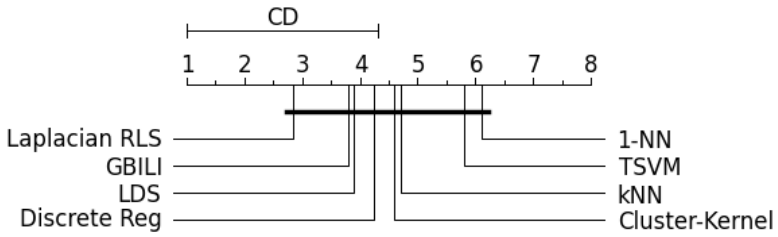
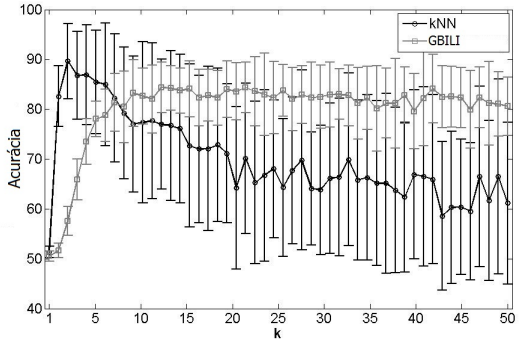
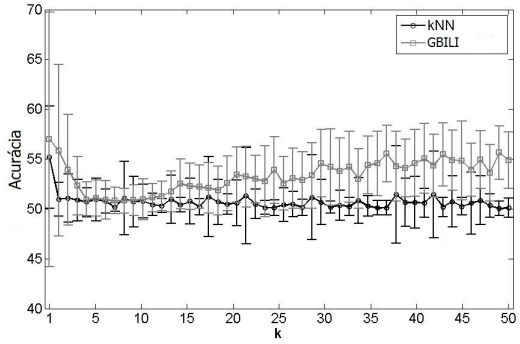


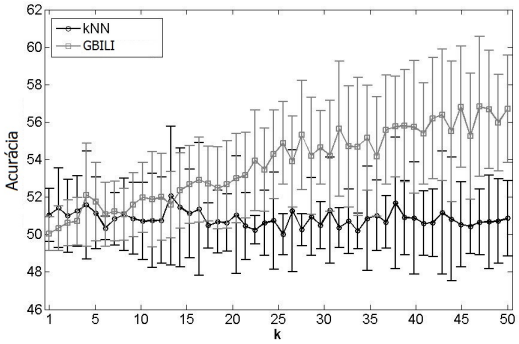
Figura 4.22: Comparação de todos os classificadores com o teste de Nemenyi. Obtida em (Berton e de Andrade Lopes, 2014).



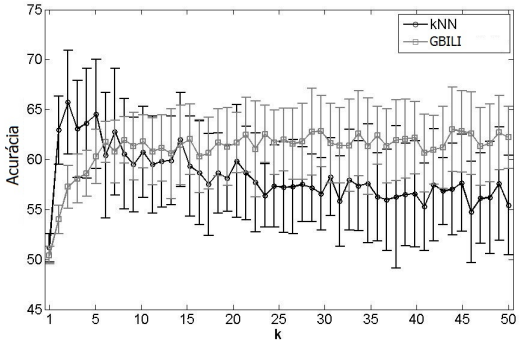
(a) digit$_1$



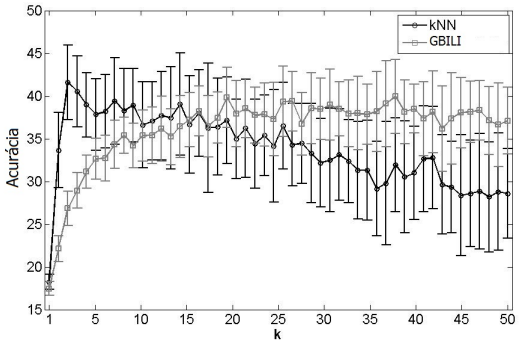
(b) g241c



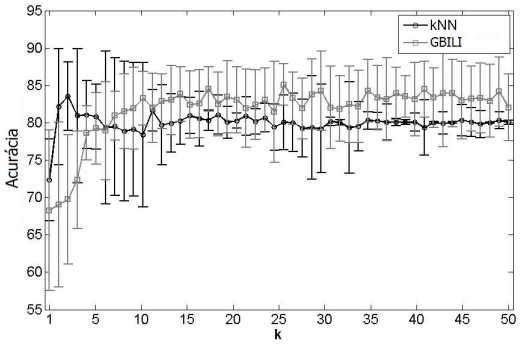
(c) g241n



(d) COIL$_2$



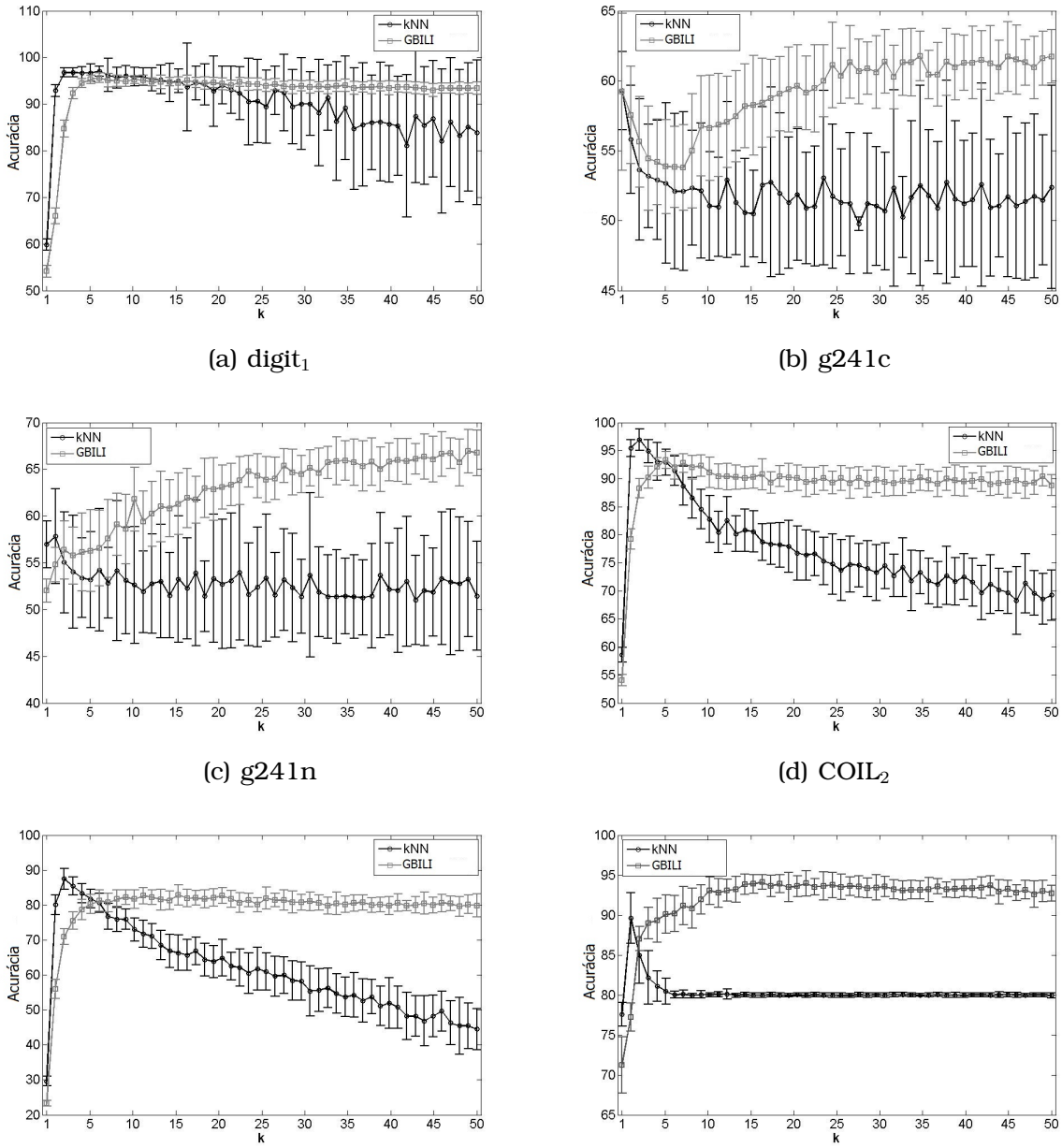
(e) COIL$_6$



(f) USPS

Figura 4.23: Comparação da porcentagem da acurácia e desvio padrão nos métodos kNN e GBILI para as bases: (a) digit$_1$, (b) g241c, (c) g241n, (d) COIL$_2$, (e) COIL$_6$ e (f) USPS utilizando-se 10 exemplos rotulados.

A Figura 4.25 mostra a medida ϕ-*edge ratio* das redes GBILI e kNN para as bases: digit$_1$, g241c, g241n, COIL$_2$, COIL$_6$ e USPS. O eixo y denota a medida ϕ-*edge ratio* das redes construídas e o eixo x denota os valores do parâmetro k utilizados. Resalta-se que o número de arestas entre as redes kNN e GBILI é diferente. Conforme o valor de k aumenta, o número de arestas nas redes kNN aumenta, mas nas redes GBILI o número de arestas permanece constante.



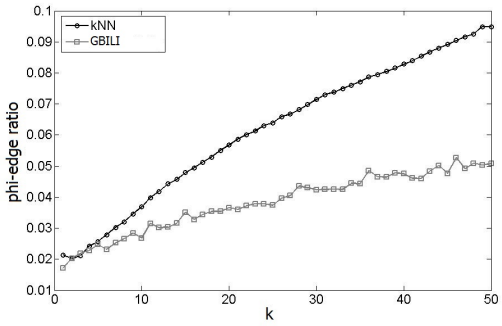
(a) digit$_1$ (b) g241c

(c) g241n (d) COIL$_2$

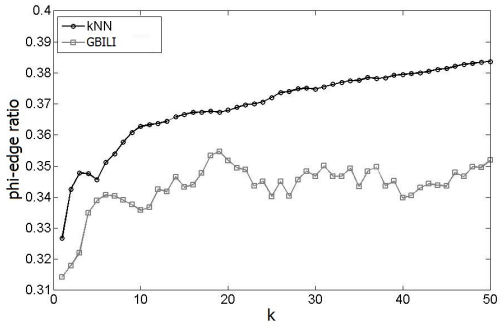
(e) COIL$_6$ (f) USPS

Figura 4.24: Comparação da porcentagem da acurácia e desvio padrão nos métodos kNN e GBILI para as bases: (a) digit$_1$, (b) g241c, (c) g241n, (d) COIL$_2$, (e) COIL$_6$ e (f) USPS utilizando-se 100 exemplos rotulados. Obtida em (Berton e de Andrade Lopes, 2014).

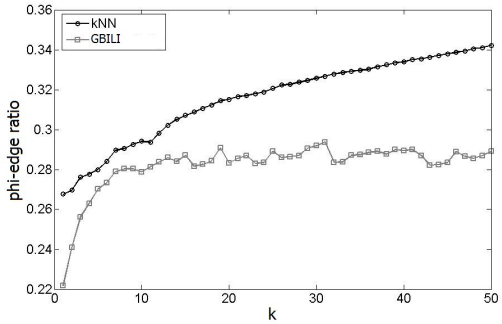
Desse modo, o método GBILI alcança valores menores para o ϕ-*edge ratio*. Já o método kNN tem valores maiores, degradando a acurácia na classificação.



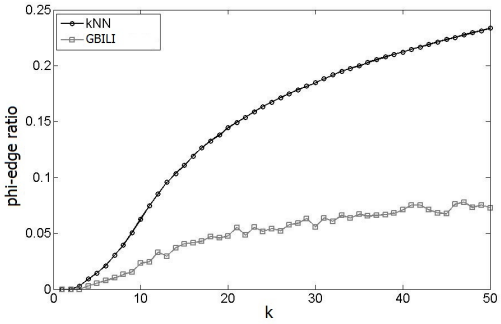
(a) digit$_1$.



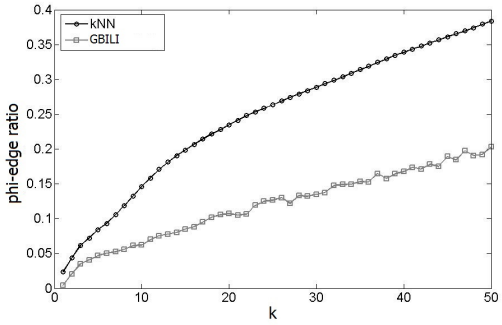
(b) g241c



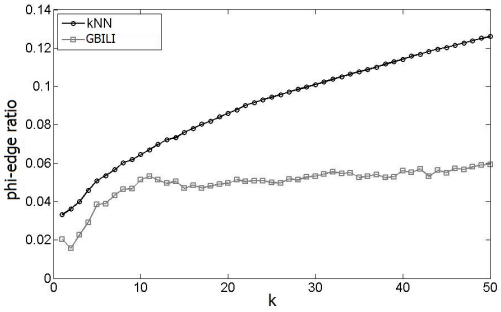
(c) g241n



(d) COIL$_2$



(e) COIL$_6$



(f) USPS

Figura 4.25: Comparação dos valores de ϕ-*edge* entre os métodos kNN e GBILI para as bases (a)digit$_1$, (b)g241c, (c)g241n, (d)COIL$_2$, (e)COIL$_6$ e USPS, utilizando 10 exemplos rotulados. Quanto menor o valor de ϕ-*edge ratio*, melhor a rede. Adaptada de (Berton e de Andrade Lopes, 2014).

4.2.2 Medidas de centralidade

Algumas medidas de centralidade nas redes GBILI foram calculadas e mostra-se que os dados previamente rotulados são importantes na topologia da rede. As medidas exploradas são o grau do vértice, *betweenness* (Freeman, 1977), *eigenvector centrality* (Bonacich, 1972) e *pageRank* (Brin e Page, 1998).

Estas medidas são apresentadas nas Tabelas: 4.13 (grau do vértice), 4.14 (*betweenness*), 4.15 (*eigenvetor*) e 4.16 (*pageRank*) para as bases: digit_1, g241c, g241n, COIL_2, COIL_6 e USPS com 10 exemplos rotulados. Os 15 maiores valores das redes GBILI são apresentados. Os vértices enumerados de 1 a 10 são rotulados e seus valores são mostrados em negrito nas tabelas. A identificação dos vértices está subescrita nos valores das medidas.

A presença de vértices com um número alto de conexões (*hubs*) facilita o espalhamento de informações ou epidemias, especialmente se os *hubs* forem infectados. As redes GBILI geram *hubs* a partir dos pontos rotulados, como pode ser observado na Tabela 4.13, na qual a maior parte dos vértices rotulados apresentam o maior grau. Na Tabela 4.14 muitos vértices rotulados apresentam um valor alto de *betweenness*, isso indica que muitos caminhos mínimos passam por eles. Esses vértices influenciam muitos outros vértices e são importantes na transmissão de informação na rede. Além de um vértice i ser importante por receber muitas conexões, i pode ser importante por receber conexões de outros vértices importantes. Isso é calculado pelas medidas *Eingenvector centrality* e *PageRank* e o resultado pode ser visto nas Tabelas 4.15 e 4.16, as quais também indicam que os vértices rotulados são importantes na rede.

O método GBILI constrói redes explorando os pontos préviamente rotulados os quais tornam-se *hubs*. Estes pontos são importantes na topologia da rede conforme indicam as medidas de centralidade calculadas. Em muitas aplicações tecnológicas e comerciais é desejável espalhar a "informação" de maneira rápida e eficiente. Exemplos dessas aplicações são protocolos baseados em rumor para disseminação de dados, campanhas de *marketing* usando rumor como estratégia, entre outras. Pode-se inferir, que a rede GBILI, dada sua topologia e importância dos vértices rotulados, facilitaria o espalhamento dos rótulos na rede.

4.2.3 Aplicação de RGCLI na segmentação de imagens

Os métodos baseados em redes têm sido utilizados em diversos domínios, entre eles na segmentação de imagens. O objetivo dos algoritmos de segmentação é dividir uma imagem em regiões importantes associadas aos objetos contidos na

Tabela 4.13: Grau do vértice. Em negrito vértices rotulados. Adaptada de (Berton e de Andrade Lopes, 2014).

digit$_1$	**g241c**	**g241n**	**COIL**$_2$	**COIL**$_6$	**USPS**
30$_2$	**32**$_{10}$	**31**$_6$	94$_{17}$	**30**$_{10}$	**28**$_1$
29$_6$	**31**$_9$	**28**$_8$	**86**$_5$	**28**$_1$	**26**$_6$
27$_7$	25$_{77}$	24$_{1491}$	**56**$_9$	**25**$_5$	**25**$_4$
26$_3$	**25**$_3$	24$_{1348}$	**50**$_1$	**23**$_9$	**25**$_7$
23$_5$	**22**$_4$	**21**$_2$	**45**$_{10}$	**21**$_8$	**23**$_5$
22$_1$	19$_{1060}$	19$_{20}$	**43**$_4$	21$_{402}$	**21**$_2$
18$_4$	19$_{1205}$	**19**$_1$	40$_{50}$	**17**$_3$	**19**$_8$
17$_{10}$	18$_{468}$	17$_{1456}$	**28**$_{11}$	17$_{257}$	**19**$_{10}$
16$_9$	17$_{32}$	16$_{192}$	26$_{65}$	**16**$_4$	16$_{936}$
12$_{612}$	**16**$_8$	16$_{769}$	25$_{38}$	12$_{682}$	14$_{398}$
12$_8$	16$_{1450}$	15$_{1098}$	**25**$_2$	11$_{809}$	**13**$_3$
12$_{1002}$	**16**$_6$	**15**$_{10}$	**24**$_8$	10$_{16}$	12$_{946}$
11$_{285}$	15$_{96}$	15$_{266}$	**22**$_6$	10$_{423}$	12$_{347}$
10$_{1275}$	14$_{1063}$	14$_{100}$	20$_{112}$	10$_{538}$	10$_{165}$
10$_{1238}$	14$_{994}$	14$_{14}$	19$_{49}$	9$_{726}$	10$_{184}$

Tabela 4.14: *Betweenness* do vértice. Em negrito vértices rotulados. Adaptada de (Berton e de Andrade Lopes, 2014).

digit$_1$	**g241c**	**g241n**	**COIL**$_2$	**COIL**$_6$	**USPS**
320972$_2$	**374096**$_9$	**318244**$_6$	**235375**$_5$	**129459**$_8$	**290077**$_1$
231152$_6$	**363743**$_{10}$	**264527**$_8$	156545$_{337}$	76138$_{130}$	**257168**$_5$
201053$_5$	**245290**$_3$	226145$_{20}$	151665$_{36}$	**71748**$_5$	186240$_{581}$
184264$_3$	**201235**$_4$	175395$_{1348}$	140075$_{17}$	71437$_{1142}$	**180672**$_7$
179110$_9$	**140562**$_5$	167572$_{63}$	**132353**$_1$	59952$_{286}$	**179414**$_4$
167644$_{133}$	135788$_{32}$	**166780**$_1$	128723$_{92}$	58196$_{379}$	**160360**$_6$
154881$_1$	127991$_{77}$	162047$_{631}$	**102044**$_8$	50977$_{658}$	**154759**$_{10}$
141761$_4$	97753$_{23}$	149359$_{33}$	65243$_{91}$	50804$_{1069}$	149967$_{398}$
137102$_{1040}$	94256$_{113}$	140287$_{1491}$	56913$_{144}$	49324$_{189}$	**149408**$_8$
127247$_{114}$	88480$_{1205}$	137298$_{410}$	56233$_{70}$	49314$_{24}$	138086$_{983}$
123675$_{112}$	86400$_{461}$	134365$_{100}$	**50513**$_9$	48424$_{682}$	128377$_{407}$
122236$_{1331}$	83555$_{1359}$	128183$_{969}$	48409$_{46}$	45947$_{534}$	125788$_{18}$
121595$_{76}$	80632$_{1180}$	114791$_{999}$	**36544**$_6$	44552$_{227}$	123331$_{544}$
112803$_{746}$	74063$_{1121}$	**107599**$_2$	33454$_{112}$	**44030**$_{10}$	123235$_{120}$
111766$_{34}$	72950$_{468}$	103861$_{18}$	32696$_{55}$	42575$_{342}$	120593$_{823}$

mesma (Kurz e Benteftifa, 2006). Muitas vezes uma estratégia de segmentação precisa ser desenvolvida de tal modo que permita aos usuários especificarem o que eles desejam. Neste caso, tem-se uma segmentação de imagem semi-automática e interativa, na qual alguns pixels de cada objeto são marcados pelo usuário e um algoritmo classifica os demais pixels. Alguns trabalhos abordando a segmentação de imagens semi-automática e interativa foram desenvolvidos (Boykov e Kolmogorov, 2001; Boykov e Jolly, 2001; Guan e Qiu, 2006; Rother et al., 2004; Wang e Cohen, 2005; Yu e Shi, 2004; Protiere e Sapiro, 2007; Ta et al., 2009; Houhou et al., 2009; Camps-Valls et al., 2007), porém, poucos deles são baseados em redes.

Devido a estas características inerentes das imagens, otimizou-se o método GBILI (Berton e de Andrade Lopes, 2014) e propôs-se o método Robust Graph that Considers Labeled Instances (RGCLI) para operar em grandes bases de dados. Aplicou-se o RGCLI na segmentação interativa de imagens e verificou-se

Tabela 4.15: *Eingenvector centrality*. Em negrito vértices rotulados. Adaptada de (Berton e de Andrade Lopes, 2014).

digit$_1$	**g241c**	**g241n**	**COIL**$_2$	**COIL**$_6$	**USPS**
1.000$_2$	**1.000**$_9$	**1.000**$_6$	1.000$_{17}$	1.000$_{402}$	**1.000**$_1$
0.837$_6$	**0.862**$_{10}$	**0.870**$_8$	**0.863**$_5$	0.919$_{257}$	**0.699**$_6$
0.497$_3$	**0.584**$_3$	0.393$_{1348}$	**0.470**$_9$	0.512$_{682}$	**0.687**$_4$
0.494$_7$	0.452$_{77}$	0.350$_{234}$	0.412$_5$0	**0.487**$_9$	**0.667**$_5$
0.388$_5$	0.399$_{32}$	0.343$_{66}$	0.214$_3$8	**0.443**$_{10}$	**0.609**$_7$
0.346$_1$	**0.391**$_4$	0.328$_{14}$	0.187$_{158}$	**0.387**$_5$	**0.307**$_2$
0.246$_{1275}$	0.343$_{468}$	0.293$_{969}$	0.168$_{218}$	0.381$_{403}$	**0.301**$_{10}$
0.235$_{1219}$	0.305$_{461}$	0.291$_{46}$	0.167$_{15}$	0.381$_{495}$	**0.293**$_8$
0.233$_{34}$	**0.281**$_5$	0.267$_{232}$	0.167$_{929}$	0.372$_{219}$	0.282$_{398}$
0.222$_{1158}$	0.279$_{1063}$	0.266$_{1456}$	0.167$_{90}$	0.372$_{36}$	0.251$_{823}$
0.212$_{708}$	0.263$_{812}$	0.265$_{63}$	0.167$_{312}$	0.372$_{205}$	0.238$_{1441}$
0.208$_{94}$	0.252$_{994}$	0.241$_{20}$	0.167$_{453}$	0.372$_{412}$	0.229$_{1180}$
0.208$_{797}$	0.248$_{266}$	0.239$_{1491}$	0.167$_{983}$	0.371$_{94}$	0.225$_{1151}$
0.207$_{170}$	0.238$_{1450}$	0.235$_{1141}$	0.166$_{715}$	0.371$_{453}$	0.220$_{394}$
0.204$_{23}$	0.221$_{1335}$	0.235$_{899}$	0.166$_{399}$	**0.342**$_1$	0.219$_{789}$

Tabela 4.16: *PageRank*. Em negrito vértices rotulados. Adaptada de (Berton e de Andrade Lopes, 2014).

digit$_1$	**g241c**	**g241n**	**COIL**$_2$	**COIL**$_6$	**USPS**
0.008$_2$	**0.008**$_{10}$	**0.007**$_6$	0.016$_{17}$	**0.008**$_{10}$	**0.007**$_1$
0.007$_6$	**0.007**$_9$	**0.007**$_8$	**0.015**$_5$	**0.008**$_1$	**0.007**$_6$
0.007$_7$	**0.006**$_3$	0.007$_{1491}$	**0.011**$_1$	**0.006**$_5$	**0.007**$_4$
0.007$_3$	0.006$_{77}$	**0.006**$_2$	**0.010**$_{10}$	**0.005**$_8$	**0.007**$_7$
0.006$_5$	0.006$_{1060}$	0.006$_{1348}$	**0.010**$_9$	**0.005**$_9$	**0.006**$_2$
0.006$_1$	**0.005**$_4$	**0.005**$_1$	**0.008**$_4$	**0.005**$_3$	**0.006**$_5$
0.005$_{10}$	0.005$_{1205}$	0.005$_{1456}$	0.007$_{50}$	0.004$_{402}$	**0.005**$_8$
0.005$_4$	0.005$_{1450}$	0.005$_{769}$	**0.006**$_8$	**0.004**$_4$	**0.005**$_{10}$
0.004$_9$	0.005$_{468}$	0.004$_{192}$	**0.005**$_2$	0.003$_{257}$	0.004$_{936}$
0.004$_{1002}$	**0.004**$_8$	0.004$_{20}$	**0.005**$_{11}$	0.002$_{809}$	0.004$_{398}$
0.003$_{612}$	**0.004**$_6$	**0.004**$_{10}$	**0.005**$_6$	0.002$_{682}$	**0.004**$_3$
0.003$_{285}$	0.004$_{126}$	0.004$_{266}$	0.005$_{65}$	**0.002**$_6$	0.004$_{347}$
0.003$_{1275}$	0.004$_{994}$	0.004$_{1098}$	0.004$_{38}$	0.002$_{423}$	0.003$_{946}$
0.003$_8$	0.004$_{32}$	0.004$_{1340}$	0.004$_{112}$	0.002$_{16}$	0.003$_{1375}$
0.003$_{1238}$	0.004$_{96}$	0.004$_{1365}$	0.004$_{49}$	0.002$_{538}$	0.003$_{184}$

que o método obtém melhores resultados na segmentação comparado a outros métodos que não utilizam os rótulos na construção das redes, como o kNN.

4.2.3.1 Base de dados

As bases de dados utilizadas são imagens do *Berkeley Segmentation Dataset* (Martin et al., 2001) cujo tamanho é 320×480, o que gera uma rede com 153600 vértices. As imagens selecionadas são mostradas na Figura 4.26 (a), (b) e (c). A partir das imagens, o usuário fornece alguns vértices rotulados selecionando os objetos que deseja reconhecer na imagem (veja Figura 4.26 (d),(e) e (f)).

4.2.3.2 Configuração dos experimentos

Os métodos RGCLI, kNN e M-kNN foram utilizados para construir uma rede a partir de uma imagem. Cada pixel foi representado por um ponto $(\text{x,y}, r, g, b)$ no

(a) (b) (c) (d) (e) (f)

Figura 4.26: (a), (b) e (c) imagens originais. (d), (e) e (f) imagens com os objetos marcados pelo usuário.

espaço, onde (x,y) representa a localização do pixel na imagem e (r, g, b) representa o canal *Red-Green-Blue* do pixel. A função de peso é dada pelo inverso da distância Euclidiana, i.e., o peso de uma aresta $e_{i,j} = (v_i, v_j)$ e seus pixels correspondentes $p_i = (\mathbf{x}_i, \mathbf{y}_i, r_i, g_i, b_i)$ e $p_j = (\mathbf{x}_j, \mathbf{y}_j, r_j, g_j, b_j)$ é calculado como $w_{i,j} = \frac{1}{||p_i - p_j||}$.

Para a construção da rede RGCLI, $k_e = 500$ foi utilizado, pois como já comentado, valores grandes favorecem uma escolha melhor de vizinhos. Variou-se o parâmetro k_i de 5 até 50, acrescendo-se k_i de 5 em 5. Neste caso não foi necessário testar valores altos para k_i pois é possível conseguir bons resultados com uma rede RGCLI esparsa. Para a construção da rede kNN variou-se o parâmetro k de 10 até 500, acrescendo-se k de 10 em 10.

Após a rede ser construída, o algoritmo de propagação baseado em comunidades *Community Label Propagation* (CLP) Raghavan et al. (2007) (implementado no *igraph* (Csardi e Nepusz, 2006)) foi utilizado para espalhar os rótulos e segmentar as imagens.

4.2.3.3 Resultado da segmentação

Primeiro, comparou-se o resultado da segmentação usando RGCLI+CLP (Figuras 4.27 (a), (b), (c), (d), (e) e (f)), kNN+CLP (Figuras 4.27 (g), (h), (i), (j), (k) e (l)) e M-kNN+CLP (Figuras 4.27 (m), (n), (o), (p), (q) e (r)). O RGCLI obtém uma boa delineação da imagem para qualquer valor de k_i considerado. Este parâmetro influencia as conexões que existem na rede. Mesmo uma rede esparsa ($k_i = 10$) obtém bons resultados. Já o parâmetro k_e indica uma vizinhança maior na qual serão selecionados os k_i vizinhos. Por isso mantivemos um valor alto de $k_e = 500$. Entretanto, independente do valor testado no parâmetro k para os métodos kNN e M-kNN, não é possível separar os objetos do fundo da imagem. Estes resultados indicam que a rede exerce um papel fundamental para a classificação semissupervisionada.

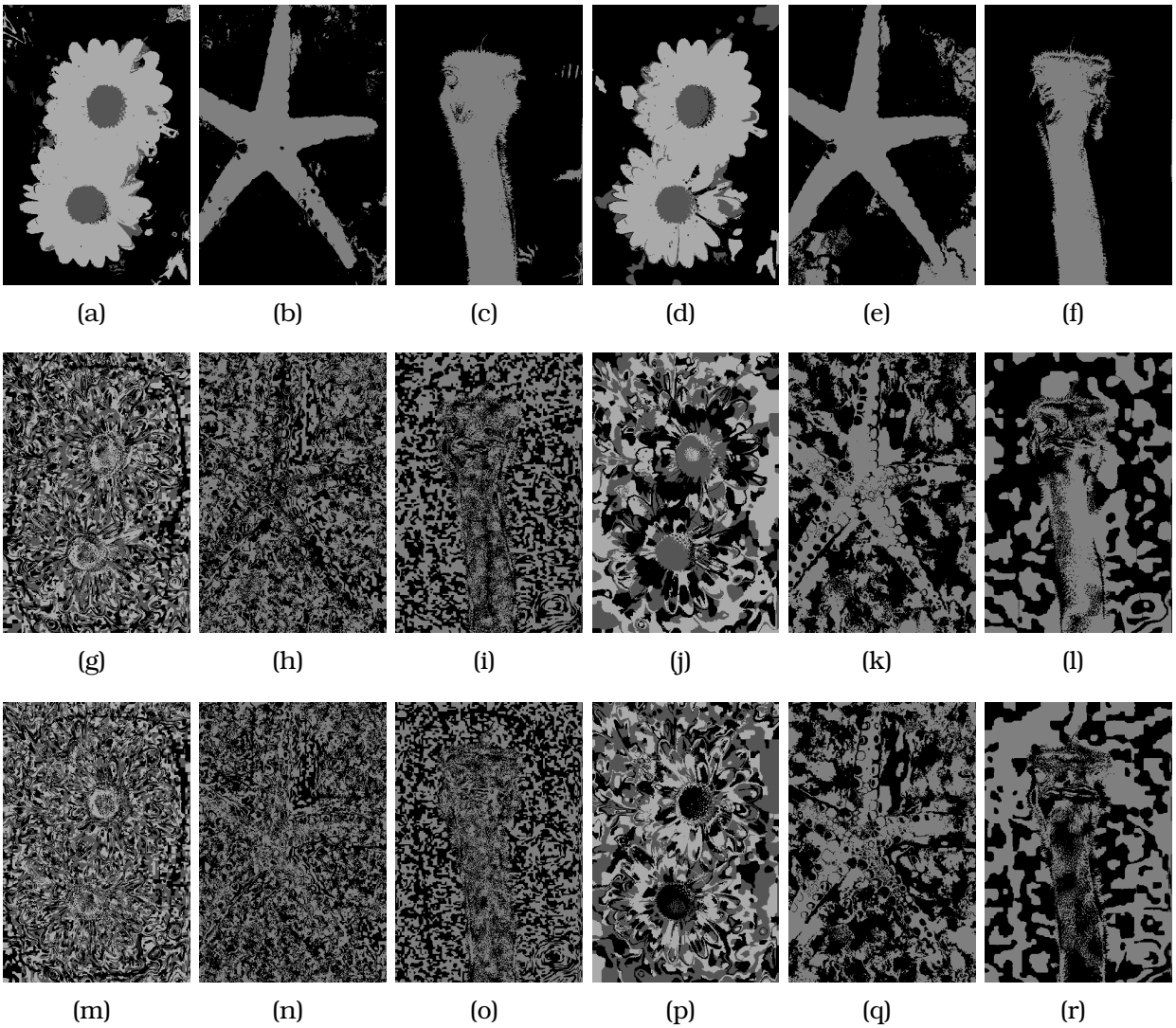


Figura 4.27: Resultado da segmentação: a primeira linha representa o resultado de RGCLI+CLP, a segunda linha representa kNN+CLP e a terceira linha representa M-kNN+CLP. Figuras (a), (b), (c), (g), (h), (i), (m), (n) e (o) $k = 10$. Figuras (d), (e), (f), (j), (k), (l), (p), (q) e (r) $k = 50$.

Segundo, validou-se a segmentação gerada pelos métodos RGCLI+CLP, kNN+CLP e M-kNN+CLP. Considerou-se como segmentação ideal a segmentação feita por humanos disponível no *Berkeley Segmentation Dataset* (Martin et al., 2001) (Figuras 4.28 (a), (c) e (e)). Como essas figuras apresentam detecção da borda, coloriu-se as imagens para obter os segmentos. Executou-se também o método *Live-wire* (Mortensen e Barrett, 1995), que permite ao usuário selecionar regiões de interesse em uma imagem por meio de cliques do mouse (Figuras 4.28 (g), (i) e (k)). O número de cliques no mouse necessários para obter essa segmentação foram respectivamente 71, 17 e 5. O resultado da segmentação é bom e o método

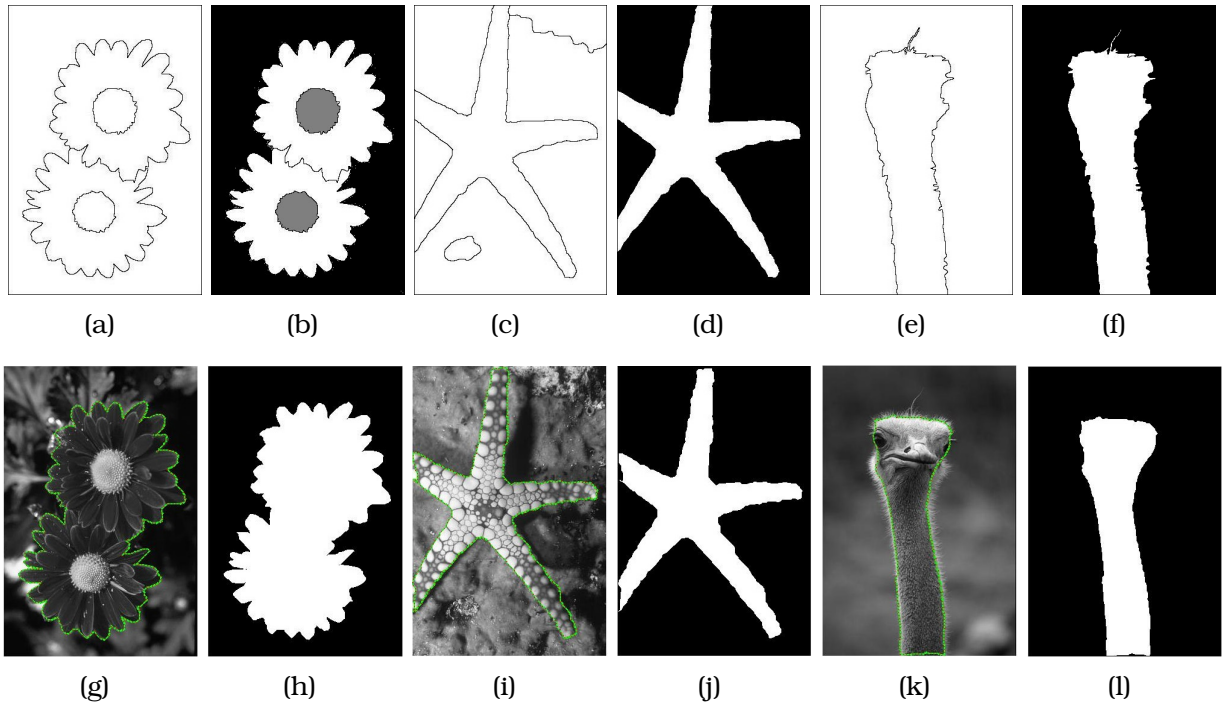


Figura 4.28: Segmentação feita por humanos com o contorno detectado: (a), (c) e (e). Versão colorida das segmentações humanas com os segmentos detectados: (b), (d) e (f). Segmentação feita pelo método *Live-wire*: (g-h) 71 cliques do mouse, (i-j) 17 cliques do mouse e (k-l) 5 cliques do mouse.

é rápido, entretanto requer bastante interação do usuário caso a imagem tenha muitos contornos.

Para comparar o resultado da segmentação dos experimentos realizados com aqueles fornecidos por humanos e pelo método *Live-wire*, empregou-se medidas de entropia como *Homogeneity*, *Completeness* e *V-measure* (Rosenberg e Hirschberg, 2007); e medidas de acurácia como *Adjusted Rand Index* (Rand, 1971). O valor da *Homogeneity* é máximo se cada segmento contém apenas pixels do segmento correto. O valor de *Completeness* é máximo se todos os pixels com o mesmo rótulo estão contidos em um mesmo segmento. E o valor de *V-measure* é a média harmônica de *Homogeneity* e *Completeness*. *Adjusted Rand Index* é uma medida de similaridade entre os segmentos reais e os preditos. Para todas as medidas valores próximos a 1 indicam maior similaridade/acurácia. Conforme já esperado, o método *Live-wire* alcança melhores resultados, já que a segmentação é guiada por um humano. Observa-se, no entanto, que os resultados do RGCLI se aproximam dos resultados do *Live-wire* indicando similaridade entre as segmentações e na imagem Avestruz o RGCLI saiu-se melhor (Tabela 4.17).

Tabela 4.17: Avaliação da segmentação

Base de dados	Método de segmentação	Homogeneity	Completeness	V-measure	Adj. Rand Index
Flor	RGCLI+CLP	0.713	0.548	0.620	0.869
	kNN+CLP	0.035	0.022	0.027	0.364
	M-kNN+CLP	0.015	0.009	0.012	0.351
	Live-wire	**0.836**	**0.843**	**0.839**	**0.949**
Estrela do Mar	RGCLI+CLP	0.708	0.702	0.705	0.935
	kNN+CLP	0.070	0.058	0.063	0.544
	M-kNN+CLP	0.024	0.020	0.022	0.522
	Live-wire	**0.793**	**0.826**	**0.809**	**0.959**
Avestruz	RGCLI+CLP	**0.695**	**0.711**	**0.703**	**0.934**
	kNN+CLP	0.027	0.023	0.025	0.527
	M-kNN+CLP	0.023	0.019	0.021	0.520
	Live-wire	0.602	0.699	0.647	0.899

4.2.3.4 Discussão

Os experimentos em segmentação de imagem evidenciam o papel dos métodos de construção de redes na classificação. Como RGCLI faz uso dos rótulos disponíveis no aprendizado semissupervisionado, os quais se tornam *hubs*, gera-se uma rede mais adequada para a tarefa seguinte (propagação de rótulos). Além disso, o RGCLI gera redes mais esparsas. Nos experimentos, $k_i = 10$ forneceu bons resultados na segmentação. Enquanto os métodos kNN e M-kNN geram redes mais densas, com arestas conectando vértices de diferentes classes e, com isso, degradando a segmentação.

4.3 Análise das redes baseadas em predição de *links*

A predição de *links* tem sido aplicada em diversas áreas, principalmente em redes sociais onlines para recomendar potenciais amizades. Neste trabalho, aplicou-se predição de *links* para evoluir uma rede esparsa inicial a ser utilizada na classificação semissupervisionada e supervisionada. O método proposto é denominado *Graph Based on Link Prediction* (GBLP). As principais contribuições do trabalho são destacadas a seguir.

- Utilização de medidas de predição de *links* para predizer novas arestas em uma rede e geração de uma rede com topologia próxima a redes mundo-pequeno, com alto coeficiente de *clustering* e a média dos caminhos mínimos baixa.

- Aplicação das redes na classificação supervisionada e semissupervisonada, os resultados apontam que essas redes aumentaram a acurácia de classificação quando comparadas com o uso das redes construídas apenas com os métodos inicias.

4.3.1 Classificação semissupervisionada

Nesta seção são apresentados os experimentos aplicando o método GBLP na classificação semissupervisionada.

4.3.1.1 Bases de dados

Os experimentos semissupervisionados foram executados em cinco bases de dados (Tabela 4.18) frequentemente utilizadas em SSL e disponíveis em (Chapelle et al., 2010).

Tabela 4.18: Base de dados para a classificação semissupervisionada

Data set	# Instâncias	# Atributos	# Classes
g241c	1500	241	2
g241n	1500	241	2
Digit_1	1500	241	2
USPS	1500	241	2
COIL_2	1500	241	2

4.3.1.2 Configuração dos experimentos

O método *principal component anaysis* (PCA) foi aplicado nas bases reduzindo as dimensões para 50 já que no espaço multi-dimensional, conforme já comentado, a distância do primeiro vizinho se aproxima da distância do último, degenerando a qualidade da rede (Beyer et al., 1999). Os experimentos foram executados considerando-se 10 e 100 vértices rotulados, aleatoriamente selecionados. Na construção das redes considerou-se os métodos MinST, MaxST, kNN e M-kNN, com $1 \leq k \leq 20$. E as redes GBLP considerando os mesmos métodos combinados com índices de predição de *links*: MinST+LP, MaxST+LP, kNN+LP e M-kNN+LP, com $1 \leq k \leq 3$. Para geração da matriz de peso W considerou-se a abordagem binária. O algoritmo para propagação dos rótulos foi o *Local and Global Consistency* (LGC) (Zhou et al., 2004). A média da acurácia sobre 30 execuções foi utilizada como medida de avaliação.

4.3.1.3 Resultado da classificação

O resultado da classificação semissupervisionada é mostrado na Tabela 4.19. Na primeira coluna estão as bases consideradas, na segunda coluna estão os métodos para construção de redes e na terceira e quarta colunas, estão as porcentagens da acurácia usando 10 e 100 rótulos respectivamente, além de um parâmetro (entre parênteses). Para kNN e M-kNN este parâmetro é o número de

vizinhos k. Para as redes GBLP este parâmetro é o valor de k, i.e., $(1, \ldots, 5)$, o método de predição de *links* utilizado (c, w ou k) e a porcentagem de *links* considerada ($10 \ldots 100$). O maior valor de acurácia para cada configuração de rótulos em cada base de dados está em negrito. As redes GBLP melhoram a acurácia especialmente quando poucos rótulos são considerados, neste caso menos de 1%.

Tabela 4.19: Acurácia (%) da classificação semissupervisionada.

BD	Method	LGC(10)	LGC(100)
g241c	kNN	0.544 ± 0.066 (1)	0.581 ± 0.026 (1)
	MkNN	0.515 ± 0.028 (4)	0.607 ± 0.082 (2)
	MinST	0.499 ± 0.011	0.50 ± 0.035
	MaxST	0.499 ± 0.009	0.501 ± 0.028
	kNN+LP	**0.581 ± 0.075 (1, c-10)**	0.597 ± 0.022 (1, c-40)
	MkNN+LP	0.574 ± 0.156 (2, k-50)	**0.622 ± 0.084 (2, c-40)**
	MinST+LP	0.535 ± 0.039 (c-40)	0.588 ± 0.021 (c-40)
	MaxST+LP	0.51 ± 0.016 (c-60)	0.536 ± 0.017 (c-70)
g241n	kNN	0.52 ± 0.03 (4)	0.573 ± 0.065 (4)
	MkNN	0.515 ± 0.024 (11)	0.571 ± 0.052 (12)
	MinST	0.499 ± 0.014	0.502 ± 0.03
	MaxST	0.499 ± 0.019	0.50 ± 0.043
	kNN+LP	**0.524 ± 0.03 (4, c-10)**	0.57 ± 0.023 (1, c-20)
	MkNN+LP	0.508 ± 0.131 (5, c-10)	0.538 ± 0.014 (5, c-30)
	MinST+LP	0.51 ± 0.161 (k-70)	**0.574 ± 0.02 (c-10)**
	MaxST+LP	0.502 ± 0.061 (c-60)	0.504 ± 0.01 (c-90)
Digit_1	kNN	0.894 ± 0.048 (3)	0.971 ± 0.097 (4)
	MkNN	0.89 ± 0.097 (7)	**0.972 ± 0.074 (10)**
	MinST	0.50 ± 0.01	0.501 ± 0.011
	MaxST	0.503 ± 0.099	0.499 ± 0.01
	kNN+LP	0.899 ± 0.06 (3, k-10)	0.966 ± 0.013 (5, w-30)
	MkNN+LP	0.905 ± 0.046 (5, c-40)	0.948 ± 0.01 (5, w-60)
	MinST+LP	**0.917 ± 0.042 (w-40)**	0.94 ± 0.018 (w-50)
	MaxST+LP	0.592 ± 0.077 (c-90)	0.716 ± 0.031 (w-80)
USPS	kNN	0.838 ± 0.03 (3)	0.892 ± 0.027 (2)
	MkNN	0.841 ± 0.063 (12)	0.913 ± 0.027 (9)
	MinST	0.709 ± 0.021	0.74 ± 0.018
	MaxST	0.709 ± 0.021	0.65 ± 0.026
	kNN+LP	0.843 (3, k-10) ± 0.034	0.89 ± 0.032 (2, c-20)
	MkNN+LP	0.799 ± 0.049 (5, w-80)	0.90 ± 0.014 (5, w-60)
	MinST+LP	**0.845 ± 0.063 (k-20)**	**0.933 ± 0.014 (w-50)**
	MaxST+LP	0.792 ± 0.015 (c-80)	0.798 ± 0.057 (c-70)
COIL_2	kNN	0.653 ± 0.045 (3)	**0.971 ± 0.017 (3)**
	MkNN	0.656 ± 0.039 (7)	0.965 ± 0.015 (7)
	MinST	0.499 ± 0.01	0.499 ± 0.031
	MaxST	0.499 ± 0.01	0.498 ± 0.029
	kNN+LP	**0.682 ± 0.062 (5, k-60)**	0.956 ± 0.019 (3, w-80)
	MkNN+LP	0.65 ± 0.038 (5, k-90)	0.953 ± 0.019 (5, c-30)
	MinST+LP	0.647 ± 0.042 (w-80)	0.898 ± 0.028 (w-60)
	MaxST+LP	0.526 ± 0.039 (c-90)	0.567 ± 0.038 (c-100)

A partir da Tabela 4.19, o teste de Freidman e post-hoc de Nemenyi (Demšar, 2006) foi executado para verificar se existe diferença estatística significativa entre os métodos de construção de redes considerados. Os resulados são mostrados na Figura 4.29. No topo do diagrama está a diferença crítica (*Critical difference* - CD) e no eixo x estão as médias dos *ranks* dos métodos avaliados,

onde os melhores *ranks* estão a esquerda. Quando os métodos analisados não possuem diferença significativa, eles são conectados por uma linha preta no diagrama. De acordo com o teste de Nemenyi, o valor crítico para comparar o *rank* médio de dois métodos no percentil 95 é $3,32$. Observa-se que todos os métodos melhoraram a acurácia quando combinados com os métodos de predição de *links*, com exceção do método M-kNN.

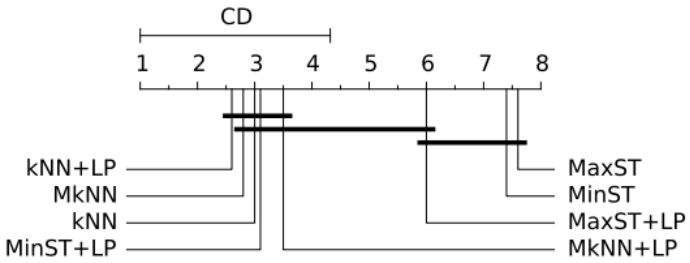


Figura 4.29: Teste post-hoc de Nemenyi para a classificação semissupervisionada. Obtida em (Berton et al., 2015).

Existe diferença estatística significativa entre todos os métodos contra MaxST e MinST, exceto MaxST+LP. Isso mostra que redes muito esparsas não são adequadas para a classificação semissupervisionada. Quando MinST e MaxST são combinados com LP a acurácia aumenta. Com relação as redes kNN, os melhores resultados foram obtidos com valores baixos de k e mesmo assim, quando kNN é combinado com LP a acurácia aumenta um pouco. Considerando o método M-kNN apenas valores de k menores que 5 foram testados e as redes são muito esparsas. Mesmo aplicando índices de LP não foi possível melhorar a acurácia. Desse modo, recomenda-se utilizar valores maiores de k para o método M-kNN+LP.

4.3.2 Classificação supervisionada

Nesta seção são apresentados os experimentos aplicando o método GBLP na classificação supervisionada.

4.3.2.1 Bases de dados

Os experimentos supervisionados foram aplicados em sete base de dados (Tabela 4.20) onde bases artificiais (Gaussians3 e Gaussians5) e reais (Wine, Ecoli, Customers, Cancer e Blood) do *UCI Machine Learning Repository* (Bache e Lichman, 2013) foram consideradas.

Tabela 4.20: Base de dados para a classificação supervisionada.

Data set	# Instances	# Attributes	# Classes
Wine	178	13	3
Ecoli	336	8	8
Customers	440	8	2
Cancer	699	10	2
Blood	748	5	2
Gaussians3	500	2	2
Gaussians5	500	2	2

4.3.2.2 Configuração dos experimentos

Para as bases Ecoli e Cancer, o primeiro atributo relacionado com o nome ou ID foram removidos. Para a base Cancer os atributos com valores ausentes foram removidos. Na construção das redes considerou-se os métodos MinST, MaxST, kNN e M-kNN, com $1 \leq k \leq 20$. E as redes GBLP considerando os mesmos métodos combinados com índices de predição de *links*: MinST+LP, MaxST+LP, kNN+LP e M-kNN+LP, com $1 \leq k \leq 3$. Para geração da matriz de peso W o oposto da Distância Euclidiana (DE) foi considerado (menor DE maior a similaridade entre os pontos). Os algoritmos utilizados para a classificação foram: nobayes, nolb-lr-binary, nolb-lr-count, nolb-lr-mode, prn (Macskassy e Provost, 2003, 2007b; Lu e Getoor, 2003). A acurácia sobre *10-fold-cross-validation* foi utilizada como medida de avaliação.

4.3.2.3 Resultado da classificação

O resultado da classificação supervisionada é sumarizado na Tabela 4.21. Na primeira coluna estão as bases consideradas, na segunda coluna estão os métodos para construção de redes e nas seguintes colunas, estão a porcentagem da acurácia usando os diferentes classificadores, além de um parâmetro (entre parênteses). Para kNN e M-kNN este parâmetro é o número de vizinhos k. Para os métodos propostos este parâmetro é o valor de k, i.e., $(1, 2$ ou $3)$, o método de predição de *links* utilizado (c, w ou k) e a porcentagem de *links* considerada $(10 \ldots 100)$. O maior valor de acurácia para cada classificador em cada base de dados está em negrito. Na maior parte dos casos as redes GBLP obtém melhor acurácia.

O teste de Freidman e post-hoc de Nemenyi (Demšar, 2006) também foi executado nos resultados da Tabela 4.21 para verificar a existência de diferença estatística significativa entre os métodos de construção de redes. Considerou-se o melhor resultado obtido por cada método entre todos os classificadores. A Figura 4.30 mostra o resultado. De acordo com o teste de Nemenyi, o valor crítico para comparar o *rank* médio de dois métodos no percentil 95 é $3,97$. Os resulta-

Tabela 4.21: Acurácia (%) da classificação supervisionada

Data set	Method	*no-Bayes*	*nolb-lr-binary*	*nolb-lr-count*	*nolb-lr-mode*	*prn*
Wine	kNN	0.353 + 0.071 (11)	0.663 + 0.101 (1)	0.719 + 0.144 (18)	0.731 + 0.130 (2)	0.647 + 0.108 (1)
	M-kNN	0.310 + 0.108 (15)	0.618 + 0.103 (2)	0.713 + 0.068 (18)	0.686 + 0.140 (9)	0.579 + 0.133 (2)
	MinST	0.237 + 0.088	0.545 + 0.116	0.715 + 0.101	0.724 + 0.114	0.623 + 0.123
	MaxST	**0.620 + 0.119**	0.613 + 0.085	0.600 + 0.155	0.398 + 0.160	0.307 + 0.050
	kNN+LP	0.410 + 0.062 (1, c-50)	0.675 + 0.081 (1, c-10)	**0.749 + 0.131 (1, c-50)**	**0.760 + 0.010 (2, c-40)**	**0.714 + 0.101 (1, k-100)**
	M-kNN+LP	0.427 + 0.120 (1, w-100)	0.664 + 0.139 (1, k -50)	0.670 + 0.103 (2, k-50)	0.698 + 0.144 (2, c-60)	0.646 + 134 (2, c-50)
	MinST+LP	0.348 + 0.154 (w-80)	0.647 + 0.115 (c-10)	0.742 + 0.078 (k-90)	0.749 + 0.148 (k-70)	0.670 + 1.159 (c-30)
	MaxST+LP	0.433 + 0.091 (c-10)	**0.761 + 0.203 (c-10)**	0.695 + 0.115 (w-70)	0.668 + 0.120 (w-80)	0.350 + 0.112 (w-50)
Ecoli	kNN	0.125 ± 0.060 (1)	0.684 ± 0.080 (1)	**0.837 ± 0.070 (17)**	**0.848 ± 0.063 (7)**	0.651 ± 0.105 (1)
	M-kNN	0.289 ± 0.055 (1)	0.667 ± 0.089 (4)	0.822 ± 0.064 (12)	0.839 ± 0.036 (11)	0.524 ± 0.080 (4)
	MinST	0.098 ± 0.032	0.670 ± 0.081	0.726 ± 0.064	0.732 ± 0.064	0.565 ± 0.103
	MaxST	0.155 ± 0.090	0.545 ± 0.128	0.538 ± 0.038	0.560 ± 0.123	0.392 ± 0.056
	kNN+LP	0.143 ± 0.044 (1, k-60)	0.753 ± 0.079 (1, w-40)	0.801 ± 0.061 (2, w-40)	0.825 ± 0.044 (3, k-80)	**0.667 ± 0.087 (1, c-20)**
	M-kNN+LP	**0.331 ± 0.090 (1, k-30)**	0.634 ± 0.066 (2, w-90)	0.714 ± 0.072 (3, k-10)	0.712 ± 0.090 (3, k-60)	0.530 ± 0.067 (3, k-40)
	MinST+LP	0.143 ± 0.070 (c-60)	**0.690 ± 0.112 (w-50)**	0.806 ± 0.055 (w-100)	0.810 ± 0.066 (w-70)	0.589 ± 0.093 (c-80)
	MaxST+LP	0.152 ± 0.128 (c-70)	0.607 ± 0.185 (k-40)	0.661 ± 0.077 (w-20)	0.664 ± 0.072 (w-100)	0.416 ± 0.052 (w-60)
Customers	kNN	0.543 ± 0.068 (13)	0.782 ± 0.042 (1)	0.870 ± 0.043 (3)	0.866 ± 0.035 (3)	0.789 ± 0.072 (1)
	M-kNN	0.614 ± 0.070 (1)	0.807 ± 0.056 (2)	0.884 ± 0.046 (17)	0.884 ± 0.038 (17)	0.748 ± 0.045 (4)
	MinST	0.470 ± 0.089	0.745 ± 0.037	0.859 ± 0.035	0.875 ± 0.036	0.780 ± 0.048
	MaxST	0.509 ± 0.086	0.675 ± 0.042	0.677 ± 0.099	0.677 ± 0.063	0.411 ± 0.121
	kNN+LP	0.555 ± 0.056 (1, c-20)	**0.811 ± 0.057 (1, w-50)**	0.873 ± 0.022 (2, c-50)	0.875 ± 0.053 (3, c-30)	0.807 ± 0.048 (1, c-70)
	M-kNN+LP	**0.639 ± 0.048 (1, c-90)**	**0.811 ± 0.056 (2, c-20)**	0.852 ± 0.050 (3, c-40)	0.843 ± 0.045 (3, c-70)	0.791 ± 0.067 (2, k-30)
	MinST+LP	0.541 ± 0.063 (c-80)	0.768 ± 0.084 (k-10)	**0.889 ± 0.020 (w-90)**	**0.886 ± 0.034 (w-90)**	**0.810 ± 0.069 (k-90)**
	MaxST+LP	0.580 ± 0.089 (k-10)	0.709 ± 0.095 (c-30)	0.730 ± 0.072 (w-60)	0.677 ± 0.090 (w-80)	0.523 ± 0.064 (w-30)
Cancer	kNN	0.526 ± 0.070 (17)	0.856 ± 0.067 (1)	0.972 ± 0.019 (5)	**0.974 ± 0.020 (8)**	0.884 ± 0.043 (1)
	M-kNN	0.654 ± 0.064 (2)	0.858 ± 0.060 (4)	0.962 ± 0.027 (12)	0.950 ± 0.036 (20)	0.755 ± 0.041 (5)
	MinST	0.475 ± 0.045	0.877 ± 0.058	0.953 ± 0.021	0.957 ± 0.030	0.880 ± 0.031
	MaxST	0.562 ± 0.094	0.789 ± 0.160	0.837 ± 0.058	0.831 ± 0.056	0.396 ± 0.088
	kNN+LP	0.556 ± 0.047 (1, w-70)	**0.893 ± 0.069 (1, w-70)**	**0.974 ± 0.018 (3, w-70)**	0.972 ± 0.028 (3, w-90)	**0.896 ± 0.039 (1, k-50)**
	M-kNN+LP	**0.666 ± 0.049 (2, w-50)**	0.856 ± 0.047 (3, k-40)	0.908 ± 0.040 (3, c-90)	0.843 ± 0.056 (3, k-10)	0.746 ± 0.051 (3, c-20)
	MinST+LP	0.591 ± 0.070 (w-20)	0.839 ± 0.115 (k-10)	0.969 ± 0.019 (w-100)	0.971 ± 0.026 (w-100)	0.894 ± 0.052 (k-50)
	MaxST+LP	0.613 ± 0.102 (c-10)	0.792 ± 0.033 (k-60)	0.930 ± 0.032 (w-80)	0.943 ± 0.021 (w-40)	0.531 ± 0.081 (c-100)
Blood	kNN	0.533 ± 0.056 (17)	0.765 ± 0.064 (4)	**0.777 ± 0.049 (4)**	**0.774 ± 0.061 (16)**	0.660 ± 0.051 (2)
	M-kNN	0.656 ± 0.052 (1)	0.762 ± 0.059 (4)	0.773 ± 0.056 (5)	0.762 ± 0.058 (1)	0.614 ± 0.044 (4)
	MinST	0.509 ± 0.071	0.762 ± 0.043	0.763 ± 0.031	0.762 ± 0.047	0.603 ± 0.070
	MaxST	0.535 ± 0.093	0.714 ± 0.174	0.769 ± 0.046	0.767 ± 0.051	0.368 ± 0.103
	kNN+LP	0.548 ± 0.072 (1, c-50)	0.769 ± 0.053 (3, k-60)	**0.777 ± 0.038 (3, c-90)**	0.762 ± 0.081 (3, w-10)	**0.666 ± 0.045 (2, c-60)**
	M-kNN+LP	**0.680 ± 0.063 (1, w-60)**	0.762 ± 0.051 (3, k-80)	0.775 ± 0.037 (3, c-60)	0.762 ± 0.061 (1, w-30)	0.641 ± 0.073 (2, w-40)
	MinST+LP	0.524 ± 0.063 (k-10)	0.766 ± 0.045 (w-100)	0.770 ± 0.065 (c-40)	0.762 ± 0.068 (k-90)	0.636 ± 0.063 (w-60)
	MaxST+LP	0.549 ± 0.114 (k-40)	**0.774 ± 0.040 (c-50)**	0.769 ± 0.043 (k-80)	0.769 ± 0.045 (k-70)	0.535 ± 0.048 (w-90)
Gaussians3	kNN	0.546 ± 0.061 (9)	0.834 ± 0.057 (1)	**0.916 ± 0.023 (18)**	**0.914 ± 0.032 (6)**	0.828 ± 0.057 (1)
	M-kNN	0.548 ± 0.066 (17)	0.774 ± 0.088 (3)	**0.916 ± 0.044 (19)**	**0.914 ± 0.028 (15)**	0.788 ± 0.058 (3)
	MinST	0.488 ± 0.078	0.816 ± 0.066	0.894 ± 0.034	0.880 ± 0.028	0.824 ± 0.029
	MaxST	0.522 ± 0.092	0.818 ± 0.153	0.902 ± 0.030	0.902 ± 0.036	0.440 ± 0.113
	kNN+LP	0.552 ± 0.085 (1, c-100)	0.848 ± 0.069 (1, c-50)	**0.916 ± 0.036 (3, k-50)**	0.912 ± 0.030 (3, w-70)	**0.844 ± 0.042 (1, k-80)**
	M-kNN+LP	0.542 ± 0.089 (2, k-30)	0.804 ± 0.064 (3, w-20)	0.870 ± 0.030 (3, w-100)	0.890 ± 0.040 (3, w-40)	0.810 ± 0.066 (3, c-50)
	MinST+LP	**0.554 ± 0.092 (k-80)**	0.840 ± 0.041 (c-30)	0.904 ± 0.045 (k-60)	0.908 ± 0.039 (c-70)	0.832 ± 0.030 (c-20)
	MaxST+LP	0.538 ± 0.111 (c-30)	**0.884 ± 0.077 (c-10)**	0.902 ± 0.042 (c-20)	0.902 ± 0.404 (c-40)	0.538 ± 0.086 (w-70)
Gaussians5	kNN	0.518 ± 0.090 (11)	0.916 ± 0.048 (1)	**0.988 ± 0.010 (7)**	**0.988 ± 0.017 (7)**	0.934 ± 0.035 (1)
	M-kNN	0.560 ± 0.075 (12)	0.866 ± 0.084 (3)	0.982 ± 0.026 (19)	0.986 ± 0.016 (19)	0.852 ± 0.046 (3)
	MinST	0.520 ± 0.076	0.878 ± 0.090	0.972 ± 0.019	0.978 ± 0.024	0.868 ± 0.067
	MaxST	0.534 ± 0.140	**0.938 ± 0.106**	0.986 ± 0.021	0.986 ± 0.016	0.358 ± 0.090
	kNN+LP	0.558 ± 0.080 (1, w-100)	0.922 ± 0.048 (1, c-40)	**0.988 ± 0.014 (3, c-80)**	**0.988 ± 0.014 (3, c-10)**	**0.944 ± 0.030 (1, c-20)**
	M-kNN+LP	0.532 ± 0.072 (2, k-70)	0.894 ± 0.055 (3, c-20)	0.966 ± 0.031 (3, k-20)	0.966 ± 0.021 (3, k-10)	0.902 ± 0.037 (3, c-60)
	MinST+LP	0.548 ± 0.084 (c-70)	0.898 ± 0.067 (c-20)	0.984 ± 0.023 (w-70)	0.984 ± 0.013 (c-20)	0.914 ± 0.039 (c-40)
	MaxST+LP	**0.570 ± 0.124 (k-100)**	0.926 ± 0.172 (c-10)	0.986 ± 0.017 (c-10)	0.986 ± 0.013 (c-10)	0.548 ± 0.082 (w-40)

dos tiveram um comportamento similar aos semissupervisionados: i) MinST+LP e MaxST+LP aumentaram a acurácia de MinST e MaxST; ii) M-kNN+LP não aumentou a acurácia de M-kNN pois utilizou-se valores muito baixo de k para M-kNN+LP; iii)kNN+LP aumentou um pouco a acurácia de kNN. Isso confirma que as redes LP possuem *links* adicionais estratégicos que podem melhorar a estrutura da rede e consequentemente a classificação.

4.3.3 Análise das redes

A fim de avaliar a influência dos parâmetros k (para as redes kNN e M-kNN) e $percentageL$ (para as redes GBLP) plotou-se a frequência dos mesmos. A Figura

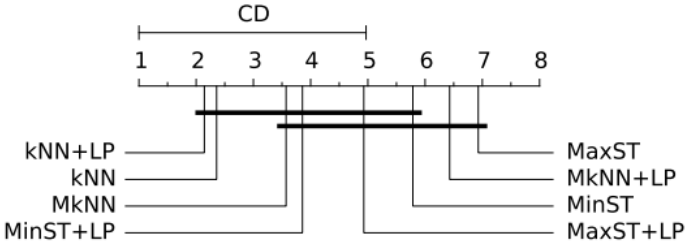


Figura 4.30: Teste post-hoc de Nemenyi para a classificação supervisionada. Obtida em (Berton et al., 2015).

4.31 mostra a quantidade de redes cujo melhor resultado de classificação foi obtido variando-se k, tal que $1 \leq k \leq 10$ (na décima posição consideramos $k \geq 10$). A figura também mostra a quantidade de redes GBLP cujo melhor resultado de classificação foi obtido variando-se $percentageL$, tal que $10 \leq percentageL \leq 100$. Observa-se na Figura 4.31 que a maioria das redes com melhor acurácia tem $k = 1$ para kNN e $k \geq 10$ para M-kNN. Já para as redes GBLP a porcentagem de *links* está distribuída igualmente, não tem um valor que se sobrepõe aos demais.

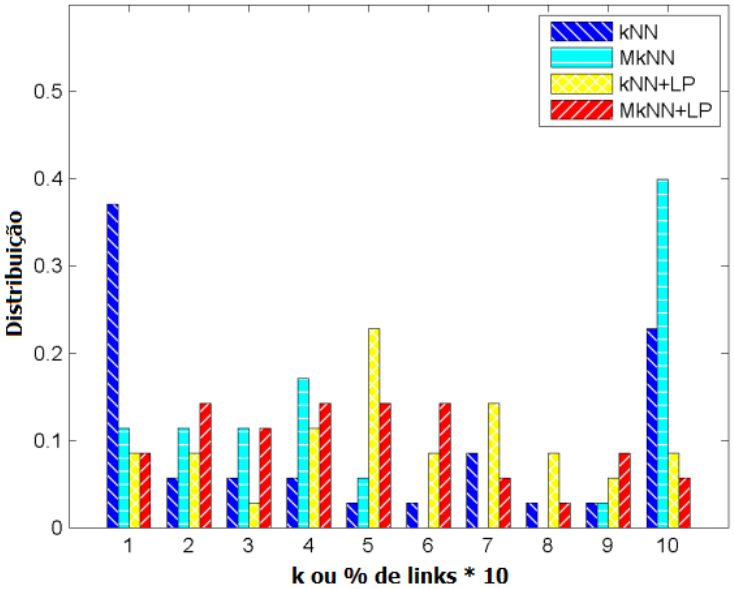


Figura 4.31: Distribuição dos parâmetros k e $percentageL$ utilizados na construção das redes para a classificação supervisionada. Obtida em (Berton et al., 2015).

As Figuras 4.32, 4.33 e 4.34 mostram o grau médio das redes, o coeficiente de *clustering* e a média dos caminhos mínimos para as redes geradas a partir da base Gaussians3. Como os valores dessas medidas é igual para as redes MinST e MaxST, elas são referidas como MST nas imagens.

Conforme o valor de k aumenta, observa-se na Figura 4.32 que o grau médio das redes kNN aumenta quase linearmente. Como as redes tornam-se mais

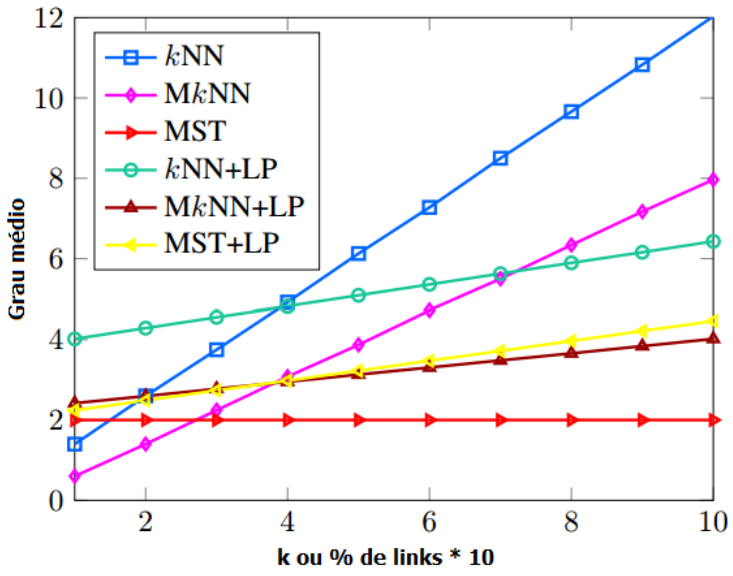


Figura 4.32: Grau médio para as redes kNN, M-kNN, MST e GBLP: kNN+LP, M-kNN+LP, MST+LP aplicadas a base Gaussians3. As redes GBLP usaram $k = 3$ e o índice *Common Neighbors*. Obtida em (Berton et al., 2015).

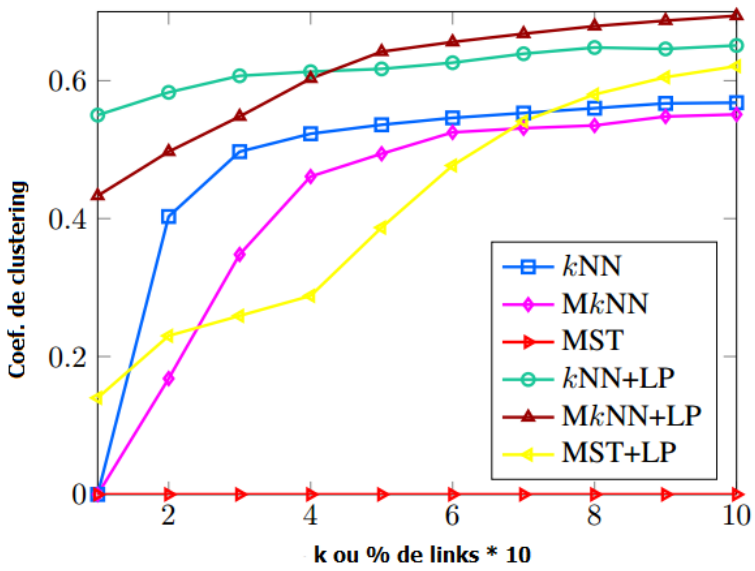


Figura 4.33: Coeficiente de *clustering* para as redes kNN, M-kNN, MST e GBLP: kNN+LP, M-kNN+LP, MST+LP, aplicadas a base Gaussians3. As redes GBLP usaram $k = 3$ e o índice *Common Neighbors*. Obtida em (Berton et al., 2015)

densas, valores baixos de k proporcionam melhores resultados. Por outro lado as redes M-kNN são muito esparsas para valores baixos de k, gerando inclusive redes desconexas. Por isso M-kNN com valores mais altos de k proporcionam

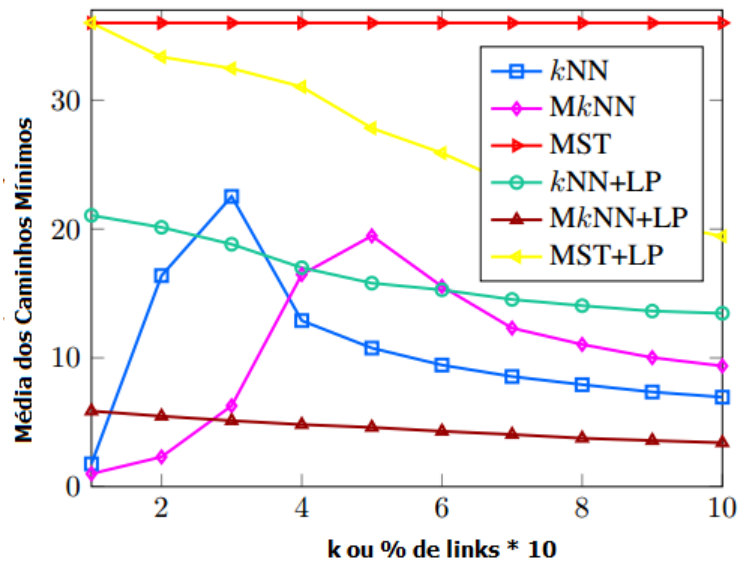


Figura 4.34: Média dos caminhos mínimos para as redes kNN, M-kNN, MST e GBLP: kNN+LP, M-kNN+LP, MST+LP, aplicadas a base Gaussians3. As redes GBLP usaram $k = 3$ e o índice *Common Neighbors*.

melhores resultados. Já as redes GBLP (kNN+LP, M-kNN+LP e MST+LP) possuem um grau médio quase constante independente do parâmetro $percentageL$.

A medida de coeficiente de *clustering* (C_c) quantifica a quantidade de ciclos de ordem três na rede. A Figura 4.33 mostra que as redes MST tem $C_c = 0$ o que é esperado, uma vez que não existem ciclos de ordem três nas redes MST. Quando combinamos MST com LP, seu C_c aumenta. As redes kNN possuem grau médio maior que as redes GBLP, entretanto o C_c é menor. Isso indica que redes com maior C_c alcançam um resultado melhor na classificação.

As redes MST possuem a maior média dos caminhos mínimos conforme se observa na Figura 4.34. Quando aplicamos uma medida de predição de *links*, a média dos caminhos mínimos tende a diminuir. O mesmo acontece com as redes M-kNN. Apenas as redes kNN não apresentam menor média de caminhos mínimos quando uma medida de predição de *links* é calculada, pois redes com $k \leq 3$ foram consideradas. Observa-se que as redes GBLP tem topologia semelhante a redes mundo-pequeno, com alto coeficiente de *clustering* e baixa média dos menores caminhos. Esse tipo de rede é comum em redes sociais reais.

4.3.4 Discussão

A predição de *links* têm sido utilizada em diversos domínios, como em sistemas de recomendação, em redes sociais, entre outros. As medidas de predição de *links* buscam encontrar arestas com alta probabilidade de existência em uma rede real. Neste trabalho, considerou-se a predição de *links* na construção de

redes para a classificação. Neste caso, redes com um grau médio muito baixo indicam falta de informações estruturais e redes com um grau médio muito alto indicam a presença de muitas arestas que se tornam ruído. As redes GBLP propostas, combinam métodos de construção de redes esparsas com medidas de predição de *links*, obtendo redes com topologia próxima a redes mundo-pequeno. Ou seja, com alto coeficiente de *clustering* e a média dos menores caminhos relativamente baixa. Estas redes se aproximam de redes sociais e mostraram-se adequadas para a classificação semissupervisionada e supervisionada.

4.4 Considerações finais

Neste capítulo foram mostrados os resultados da aplicação dos três métodos propostos para construção de redes em diversos domínios, além da caracterização das redes geradas. Observa-se que os métodos propostos (S-kNN, GBILI, RGCLI e GBLP (kNN+CN)) resultaram em boas acurácias na classificação semissupervisionada e supervisionada. Ressalta-se também que os vértices rotulados (para GBILI, RGCLI) foram aleatoriamente selecionados e os métodos obtiveram bons resultados, mas poderiam ser melhorados através de escolhas de vértices rotulados bem posicionados.

As medidas de centralidade e conectividade aplicadas nas redes geradas mostram que as redes apresentam diferentes topologias, que podem influenciar na propagação dos rótulos. As redes regulares são mais esparsas e apresentam uma distribuição de grau mais homogênea e com isso menos ruído na rede. As redes S-kNN são indicadas para aplicações gerais em que não se tem muito conhecimento sobre os dados. As redes GBILI e RGCLI fazem uso dos rótulos disponíveis *a priori* no SSL os quais se tornam *hubs* formando redes com topologia livre de escala. Como os *hubs* são importantes na difusão de informação eles podem facilitar a propagação dos rótulos. Além disso, os rótulos devem ser confiáveis, por isso esse método é mais indicado para aplicações interativas, nas quais o usuário fornece alguns poucos rótulos para o algoritmo. As redes GBLP são indicadas para melhorar a conectividade de uma rede pré-existente que é muito esparsa. Por meio da predição de *links*, novas arestas são adicionadas na rede tornando-a com propriedades de redes mundo-pequeno. A Tabela 4.22 sumariza as principais caraterísticas dos métodos propostos.

Os resultados indicam que a rede exerce um papel fundamental na classificação de dados relacionais e os algoritmos de classificação podem se comportar de maneira diferente em diferentes tipos de redes. Destaca-se a necessidade de escolha de um método adequado para a construção de redes a partir de dados em

Tabela 4.22: Sumarização dos métodos propostos.

Método	Complexidade	Parâmetros	Topologia	Aplicações
S-kNN	kn^2	k	Regular	Aplicações gerais, não se conhece os dados
GBILI	kn^2	$k > 10$	Livre de Escala	Aplicações interativas, rótulos confiáveis
RGCLI	$nk \log n$	k_e e k_i	Livre de Escala	Aplicações interativas, rótulos confiáveis
GBLP	$nk \log n$	k	Mundo-pequeno	Melhoria de uma rede pré-existente

formato atributo-valor. Além disso, ainda existe espaço para mais estudos nesta área, algumas das limitações existentes e trabalhos futuros são apresentados no próximo capítulo.

CAPÍTULO

5

Conclusões

5.1 Resultados e Contribuições

As redes têm sido amplamente utilizadas para modelar problemas reais. Diversos domínios como redes sociais, redes de citações, páginas *web*, entre outros, possuem uma estrutura inerente de rede, onde vértices representam os elementos e arestas representam relações entre esses elementos. Entretanto, muitos dados são independentes e identicamente distribuídos, sem uma estrutura de rede. Para aplicar um método baseado em redes nestes dados, primeiro é necessário construir uma rede apropriada.

Muitas técnicas baseadas em redes para o SSL têm sido propostas, porém estudos que envolvem a influência da rede em tais algoritmos, bem como novas técnicas para construção de redes, ainda têm recebido pouca atenção. Desse modo, esta tese buscou investigar tais aspectos e propor novas técnicas de construção de redes para o aprendizado semissupervisionado, explorando diferentes topologias.

Desenvolveu-se um método para geração de redes regulares com tempo de processamento quadrático, inferior a outros métodos da literatura que geram redes regulares. Este método intitulado *Sequential* k*-Nearest Neighbor* (S-kNN), constrói uma rede com base nos k vizinhos mais próximos, porém estabelece as conexões de maneira sequencial buscando manter todos os vértices com apenas k conexões. Realizou-se um comparativo entre os principais métodos de construção de redes da literatura, incluindo métodos regulares e não regulares e

observou-se que métodos regulares obtêm melhores resultados na classificação semissupervisionada. Além disso, explorou-se características das redes relacionadas com o grau e observou-se que redes com distribuição de grau homogênea e esparsas obtiveram correlação moderada com a acurácia. Além disso, aplicou-se o método S-kNN na construção de redes para a classificação de gêneros musicais. Comparou-se os resultados entre métodos relacionais e tradicionais e observou-se que os algoritmos relacionais obtém melhor acurácia na classificação. Os resultados também indicam que o método de construção de redes regulares proposto atinge melhor acurácia comparado com métodos tradicionais, como o kNN.

Os rótulos disponíveis *a priori* em SSL são informações que podem ser utilizadas para melhorar a construção do grafo. Assim, foram propostos os métodos *Graph Based on Labeled Instances* (GBILI) e *Robust Graph that Considers Labeled Instances* (RGCLI), que fazem uso dos rótulos para a construção das redes. As principais vantagens desses métodos são a geração de redes esparsas, formação de *hubs* a partir dos dados rotulados gerando redes com distribuição de grau lei de potência, acurácia na classificação melhor que o método kNN e tempo de processamento quadrático para GBILI e $O(nk \log n)$ para RGCLI. Além disso, mostrou-se matematicamente que os métodos são adequados para modelar a restrição de suavidade do SSL. Uma aplicação de RGCLI na tarefa de segmentação interativa de imagens também foi apresentada, ilustrando a capacidade do RGCLI lidar com redes grandes, bem como a importância do uso dos rótulos disponíveis no SSL.

Em outra abordagem proposta considerou-se a construção de redes por meio de medidas de predição de *links*, adicionando-se potenciais arestas em uma rede esparsa já existente. Os resultados na classificação supervisionada e semissupervisonada mostraram que as redes *minimum/maximum spanning tree* e kNN, $1 \leq k \leq 3$, obtiveram maior acurácia quando combinadas com *link prediction*. Além disso, essa abordagem gera redes com propriedades mundo-pequeno, apresentando alto coeficiente de *clustering* e a média dos caminhos mínimos relativamente baixa. Esse tipo de rede é bastante comum em redes sociais.

Conclui-se que é possível construir redes mais apropriadas para o aprendizado semissupervisionado, explorando propriedades estruturais ou informações disponíveis *a priori*. Além disso, a topologia da rede influencia o resultado dos métodos de aprendizado baseados em redes. Por exemplo, redes esparsas são mais adequadas que redes densas, se os vértices rotulados tornam-se *hubs* podem facilitar a propagação dos rótulos, redes com distribuição do grau homogêneas podem ser mais adequadas em alguns domínios que redes com distribuição do grau heterogêneas.

5.1.1 Contribuições

As principais contribuições deste trabalho de doutorado são sumarizadas nos pontos a seguir:

- Três novos métodos para a construção de redes, a partir de dados em formato atributo-valor, que fazem uso de propriedades ainda não exploradas na literatura, foram propostos. Buscou-se investigar diferentes topologias, tais como redes regulares, redes com distribuição de grau lei de potência e redes com características mundo-pequeno, comuns em redes reais.
- Melhoria da acurácia dos algoritmos semissupervisonados baseados em redes, utilizando os métodos de construção de redes propostos.
- Aplicação dos métodos para construção de redes e dos algoritmos semissupervisionados baseados em redes em bases de dados com muitas instâncias. Como aplicações reais relevante tem-se o uso das técnicas propostas em classificação de gêneros musicais e na segmentação de imagens.
- Desenvolvimento de métodos com complexidade de tempo quadrática ou menor.

5.1.2 Publicações

Publicações e submissões diretamente relacionadas com o tema de pesquisa:

1. Lilian Berton, Didier Vega-Oliveros, Francisco A. Rodrigues e Alneu de Andrade Lopes. *Robustness analysis of graph construction methods for semi-supervised learning*. ACM TKDD. (under review).
2. Lilian Berton, Thiago de Paulo Faleiros, Alan Valejo, Jorge Valverde-Rebaza e Alneu de Andrade Lopes. *RGCLI: Robust Graph that Considers Labeled Instances*. Neurocomputing. (under review).
3. Lilian Berton e Alneu de Andrade Lopes. *Graph construction for semi-supervised learning*. In Doctoral Consortium. International Joint Conference on Artificial Intelligence (IJCAI 2015), pp. 4343-4344. Buenos Aires, Argentina 2015.
4. Lilian Berton, Jorge Valverde-Rebaza e Alneu de Andrade Lopes. *Link prediction in graph construction for supervised and semi-supervised learning*. In Proceedings of The International Joint Conference on Neural Networks (IJCNN 2015), pp. 1-8. Killarney, Ireland 2015.

5. Lilian Berton e Alneu de Andrade Lopes. *Graph Construction Based on Labeled Instances for Semi-Supervised Learning*. In Proceedings of International Conference on Pattern Recognition (ICPR 2015), pp. 2477-2482, Stockholm, Sweden 2014.

6. Jorge Valverde-Rebaza, Aurea Soriano, Lilian Berton, Maria Cristina Ferreira de Oliveira e Alneu de Andrade Lopes. *Music genre classification using traditional and relational approaches*. In Proceedings of 2014 Brazillian Conference on Intelligent Systems (BRACIS 2014), pp. 259-264. Sao Carlos, Brazil 2014.

7. Didier Vega-Oliveros, Lilian Berton, Andre Mantini Eberle, Alneu de Andrade Lopes e Liang Zhao. *Regular Graph Construction for Semi-supervised Learning*. Journal of Physics: Conference Series (Online), 490, pp. 012022-1-012022-4, 2014.

8. Lilian Berton e Alneu de Andrade Lopes. *Informativity-based Graph: Exploring Mutual kNN and Labeled Vertices for Semi-supervised Learning*. In Proceedings of Fourth International Conference on Computational Aspects of Social Networks (CASoN 2012), pp. 14-19. Sao Carlos, Brazil 2012.

Publicações resultantes de colaborações da aluna com outros colegas e grupos de pesquisa do ICMC/USP em áreas co-relacionadas ao tema de pesquisa:

1. Jorge Valverde-Rebaza, Alan Valejo, Lilian Berton, Thiago Faleiros e Alneu Lopes. *A naïve bayes model based on overlapping groups for link prediction in online social networks*. In Proceedings of The 30th ACM/SIGAPP Symposium On Applied Computing (SAC 2015), pp. 1136-1141. Salamanca, Spain, 2015.

2. Didier Vega-Oliveros, Lilian Berton, Alneu de Andrade Lopes e Francisco Rodrigues. *Influence maximization based on the least influential spreaders*. In Workshop on Social Influence Analysis (SocInf 2015). International Joint Conference on Artificial Intelligence (IJCAI 2015), pp. 03-08. Buenos Aires, Argentina 2015.

3. Didier Vega-Oliveros e Lilian Berton. *Spreader Selection by Community to Maximize Information Diffusion in Social Networks*. In Track on Web and Text Intelligence (WTI 2015). International Symposium on Information Management and Big Data (SIMBig 2015), pp. 73-82. Cusco, Peru 2015.

4. Lucas Assirati, Núbia Rosa da Silva, Lilian Berton, Alneu de Andrade Lopes e Odemir Martinez Bruno. *Performing edge detection by Difference of*

Gaussians using q-Gaussian kernels. Journal of Physics. Conference Series (Online), v. 490, pp. 012020-1-012020-4, 2014.

5. Nils Ever Murrugarra Llerena, Lilian Berton e Alneu de Andrade Lopes. *Graph-based Cross-validated Committees Ensembles*. In Proceedings of Fourth International Conference on Computational Aspects of Social Networks (CASoN 2012), pp. 75-80, Sao Carlos, Brazil 2012.

6. Thiago Faleiros, Lilian Berton e Alneu de Andrade Lopes. *Exploring Data Classification with K-associated Network*. In: 4th International Workshop on Web and Text Intelligence (WTI 2012), Curitiba, Brazil, 2012.

5.2 Limitações

Apesar de métodos baseados em redes obterem bons resultados e serem foco de diversas pesquisas e aplicações, ainda existem limitações. Algumas das principais limitações na área, que também são limitações das propostas desse trabalho, são apresentadas a seguir.

5.2.1 Alta dimensionalidade

A construção de redes baseadas em vizinhança é um passo necessário em diversas aplicações como *data mining* (Brito et al., 1997), *manifold learning* (Belkin e Niyogi, 2003; Saul e Roweis, 2003), movimentação de robôs (Choset et al., 2005) e computação gráfica (Sankaranarayanan et al., 2007). Entretanto, conforme a dimensionalidade dos dados aumenta, a distância do primeiro vizinho se aproxima a distância do último vizinho. Ou seja, o contraste entre distâncias de diferentes exemplos deixa de existir (Beyer et al., 1999). Desse modo deve-se levar em conta essa limitação na construção de redes baseadas em vizinhança.

5.2.2 Escalabilidade para bases de dados gigantes

Na era da divulgação de dados, existe um amplo interesse em explorar uma enorme quantidade de dados disponíveis em fontes abertas como a *web*, para ajudar a resolver problemas como reconhecimento de padrões, detecção de tópicos, recuperação de informação, etc. Entretanto surge um questionamento: como desenvolver métodos para processar e armazenar milhões ou bilhões de exemplos. Muitos métodos baseados em redes têm tempo de processamento quadrático, isso pode ser uma restrição em aplicações dessa escala. Para construir um grafo baseado em vizinhança também são necessários métodos ágeis, a

fim de acelerar o processo de encontrar os vizinhos mais próximos de cada vértice. Técnicas usando *hashing*, por exemplo, ajudam a alcançar um resultado mais rápido. Entretanto deve-se buscar o desenvolvimento de novas abordagens avançadas para construção de redes que obtenham bom desempenho na execução, armazenação e acurácia.

5.2.3 Rótulos contaminados e desbalanceamento de rótulos

Os dados da *web* podem ser rotulados, por exemplo considerando-se *tags* em imagens e vídeos compartilhados por usuários do Flickr ou Youtube. Porém, esses rótulos podem conter *outliers* ou baixa acurácia. Kennedy et al. (2006) destacam que a acurácia de *tags* em imagens da *web* podem ser menor que 50%. Como lidar com rótulos contaminados? Alguns métodos de construção de redes, como o GBILI proposto, devem ser aplicados considerando apenas rótulos confiáveis. Outra questão trata do desbalanceamento de rótulos no SSL. Suponha que sejam disponibilizados 3 rótulos negativos e apenas 1 rótulo positivo, neste caso a propagação será bastante enviesada para o rótulo negativo.

5.2.4 Redes heterogêneas

Grande parte dos trabalhos em SSL consideram redes homogêneas, entre eles os métodos estudados e propostos nesta tese. Nesse caso os vértices correspondem a um único tipo de entidade e as arestas correspondem a um único tipo de relação. Porém em muitas bases reais existe heterogeneidade nas informações. Por exemplo, dados do Flickr possuem diversas entidades: usuários, imagens, *tags*. E diversos tipos de relações: usuário-usuário (amizade), usuário-conteúdo (propriedade, visualização, etc), conteúdo-*tags*. O desenvolvimento de novos modelos que capturem dados heterogêneos irá aumentar a flexibilidade e o poder de representação das técnicas SSL.

5.2.5 Evolução de redes

Muitas redes reais evoluem, i.e., alguns vértices desaparecem e outros surgem. Porém os métodos de classificação transdutiva consideram apenas redes estáticas. Existem poucos métodos baseados em redes que consideram classificação indutiva ou redes dinâmicas.

5.3 Trabalhos Futuros

Técnicas de construção de redes é uma área de pesquisa em expansão com muitas questões a serem investigadas. Além de trabalhos que lidem com as limitações descritas na seção anterior, a seguir são apresentadas algumas questões que podem ser pesquisadas futuramente.

5.3.1 Aplicações

Os métodos de construção de redes propostos nessa tese podem ser aplicados em divesos contextos. Por exemplo, Greenfield e Campbell (2013) utilizaram redes baseadas em vizinhança para reconhecimento de fala humana. Em uma rede de conteúdo de fala, os vértices representam sinais de voz e as arestas representam similaridade entre os locutores. Deseja-se obter redes onde as arestas conectem apenas vértices do mesmo locutor. Muitas aplicações para redes de conteúdo tem sido desenvolvidas, porém pouca atenção tem sido dada na criação de tais redes.

5.3.2 *Active learning* e SSL

Tanto *active learning* (Tong e Koller, 2002) quanto SSL buscam lidar com a disponibilidade de poucos rótulos. A maioria dos métodos de *active learning* são realizados em um esquema de aprendizagem supervisionada, como *SVM active learning*, onde os vértices não rotulados mais informativos são selecionados para minimizar a fronteira de decisão atual. A integração de *active learning* com métodos baseados em redes e SSL ainda foi pouco explorada.

5.3.3 Estudos teóricos para construção de redes

Existem poucos estudos teóricos relacionados com a construção de redes. Maier et al. (2009, 2007b) analisam matematicamente a influência da construção da rede, porém com foco na rede kNN no contexto de agrupamento. Rohban e Rabiee (2012) provam que uma rede ótima é um subgrafo do kNN e analisam matematicamente métodos de construção de redes para classificação supervisionada. Mais estudos matemáticos envolvendo a construção de redes podem ser desenvolidos e assim auxiliar na busca de uma rede ideal para determinada aplicação.

5.3.4 Rotulagem de dados representativos

Um tópico de pesquisa correlacionado com o tema desta tese, diz respeito a rotulagem de vértices mais representativos para o aprendizado semissupervisionado. Alguns trabalhos da literatura exploram esse assunto. Araujo e Zhao (2013) por exemplo, analisaram medidas de redes complexas para escolha de vértices representativos em redes com estrutura de comunidades homogêneas (Danon et al., 2005) e heterogêneas (Lancichinetti et al., 2008). Poderia ser feito análises semelhantes, considerando medidas estruturais nos modelos de redes propostos, além de modelos de dinâmica em redes.

Referências Bibliográficas

ALBERT, R.; BARABASI, A.-L. *Reviews of Modern Physics*, v. 74, n. 1, páginas 47–97, 2002.

ARAUJO, B.; ZHAO, L. Detecting and labeling representative nodes for network-based semi-supervised learning. Em: *The International Joint Conference on Neural Networks*, 2013, páginas 1–8.

BACHE, K.; LICHMAN, M. UCI machine learning repository. 2013. Disponível em: `http://archive.ics.uci.edu/ml`

BALUJA, S.; SETH, R.; SIVAKUMAR, D.; JING, Y.; YAGNIK, J.; KUMAR, S.; RAVICHANDRAN, D.; ALY, M. Video suggestion and discovery for youtube: Taking random walks through the view graph. Em: *Proceedings of the 17th International Conference on World Wide Web*, ACM, 2008, páginas 895–904.

BARABASI, A.-L.; ALBERT, R. Emergence of scaling in random networks. *Science*, v. 286, páginas 509–512, 1999.

BARRAT, A.; BARTHÉLEMY, M.; VESPIGNANI, A. *Dynamical processes on complex networks*. Cambridge University Press, 2008.

BELKIN, M. *Problems of learning on manifolds*. Tese de Doutorado, 2003.

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

BELKIN, M.; NIYOGI, P. Semi-supervised learning on riemannian manifolds. *Machine Learning*, v. 56, páginas 209–239, 2004.

BELKIN, M.; NIYOGI, P. Towards a Theoretical Foundation for Laplacian-based Manifold Methods. Em: *COLT'05: Proceedings of the 18th Annual Conference on Learning Theory*, 2005, páginas 486–500.

BELKIN, M.; NIYOGI, P.; SINDHWANI, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, v. 7, páginas 2399–2434, 2006.

BENGIO, Y.; DELALLEAU, O.; ROUX, N. L. *Semi-Supervised Learning - chapter Label Propagation and Quadratic Criterion*. MIT Press, Cambridge, 2006.

BERNSTEIN, M.; SILVA, V.; LANGFORD, J.; TENENBAUM, J. *Graph approximations to geodesics on embedded manifolds*. Relatório Técnico, Stanford University, Stanford, 2000.

BERTON, L.; ANDRADE LOPES, A. Informativity-based graph: Exploring mutual knn and labeled vertices for semi-supervised learning. Em: *Proceedings of Fourth International Conference on Computational Aspects of Social Networks*, IEEE, 2012, páginas 14–19.

BERTON, L.; ANDRADE LOPES, A. Graph construction based on labeled instances for semi-supervised learning. Em: *22nd International Conference on Pattern Recognition*, 2014, páginas 2477–2482.

BERTON, L.; VALVERDE-REBAZA, J.; ANDRADE LOPES, A. Link prediction in graph construction for supervised and semi-supervised learning. Em: *International Joint Conference on Neural Networks*, 2015, páginas 1–8.

BEYER, K.; GOLDSTEIN, J.; RAMAKRISHNAN, R.; SHAFT, U. When is "nearest neighbor"meaningful? Em: *International Conference on Database Theory*, 1999, páginas 217–235.

BLUM, A.; CHAWLA, S. Learning from Labeled and Unlabeled Data using Graph Mincuts. Em: *ICML'01: Proceedings of the Eighteenth International Conference on Machine Learning*, 2001, páginas 19–26.

BONACICH, P. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, v. 2, n. 1, páginas 113–120, 1972.

BOYKOV, Y.; JOLLY, M.-P. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. Em: *International Conference on Computer Vision*, 2001, páginas 105 – 112.

BOYKOV, Y.; KOLMOGOROV, V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, páginas 359–374, 2001.

BRIN, S.; PAGE, L. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, v. 30, n. 1-7, páginas 107–117, 1998.

BRITO, M. R.; CHAVEZ, E. L.; QUIROZ, A. J.; YUKICH, J. E. Connectivity of the mutual k-nearest neighbor graph in clustering and outlier detection. *Statistics and Probability Letters*, v. 35, n. 1, páginas 33–42, 1997.

CAMPS-VALLS, G.; MARSHEVA, T. V. B.; ZHOU, D. Semi-supervised graph-based hyperspectral image classification. *IEEE Transactions Geoscience and Remote Sensing*, v. 45, n. 10, páginas 3044–3054, 2007.

CASABONA, H.; FREDERICK, D. *What is midi?* Alfred Music Publishing, 1988.

CHAPELLE, O.; SCHLKOPF, B.; ZIEN, A. *Semi-supervised learning*. 1st ed. The MIT Press, 2010.

CHOSET, H.; LYNCH, K. M.; HUTCHINSON, S.; KANTOR, G. A.; BURGARD, W.; KAVRAKI, L. E.; THRUN, S. *Principles of robot motion: Theory, algorithms, and implementations*. Cambridge, MA: MIT Press, 2005.

CLAUSET, A.; SHALIZI, C. R.; NEWMAN, M. E. J. Power-law distributions in empirical data. *SIAM Review*, v. 51, n. 4, páginas 661–703, 2009.

COHEN, J. *Statistical Power Analysis for the Behavioral Sciences*. 2 ed. Routledge, 1988.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. *Introduction to algorithms*, cáp. 10 The MIT Press, 2009.

CSARDI, G.; NEPUSZ, T. The igraph software package for complex network research. *InterJournal*, v. Complex Systems, p. 1695, 2006.

DANON, L.; DÃAZ-GUILERA, A.; DUCH, J. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, páginas 09008–1–09008–10, 2005.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, v. 7, páginas 1–30, 2006.

DHILLON, P. S.; TALUKDAR, P. P.; CRAMMER, K. Inference driven metric learning for graph construction. 2010.

DOROGOVTSEV, S. N.; MENDES, J. F. F. *Evolution of Networks: From Biological Nets to Internet and WWW*. Oxford, 2003.

EPPSTEIN, D.; PATERSON, M. S.; YAO, F. F. On nearest-neighbor graphs. *Discrete and Computational Geometry*, 1997.

ERDÖS, P.; RÉNYI, A. On random graphs. *Publicationes Mathematicae*, v. 6, n. 1, páginas 290–297, 1959.

FOUSS, F.; PARIJS, C. V. Improving accuracy by reducing the importance of hubs in nearest-neighbor recommendations. Em: *22th European Symposium on Artificial Neural Networks*, 2014, páginas 59–64.

FREEMAN, L. C. A set of measures of centrality based on betweenness. *Sociometry*, páginas 35–41, 1977.

GANTZ, J. F.; REINSEL, D. The Digital Universe Decade Are You Ready? IDC iView, 2010.

GETOOR, L.; SEGAL, E.; TASKAR, B.; KOLLER, D. Probabilistic models of text and link structure for hypertext classification. Em: *IJCAI Workshop on Text Learning: Beyond Supervision*, 2001.

GOWDA, K.; KRISHNA, G. Agglomerative clustering using the concept of mutual nearest neighborhood. *Pattern Recognition*, v. 10, n. 2, páginas 105–112, 1978.

GREENFIELD, K.; CAMPBELL, W. Link prediction methods for generating speaker content graphs. Em: *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, páginas 7721–7725.

GUAN, J.; QIU, G. *Interactive image matting using optimization*. Relatório Técnico, The University of Nottingham - School of Computer Science and Information Technology, 2006.

HEIN, M. Uniform Convergence of Adaptive Graph-based Regularization. Em: *COLT'06: Proceedings of the 19th Conference on Learning Theory*, 2006, páginas 50–64.

HEIN, M.; AUDIBERT, J.-Y.; LUXBURG, U. From Graphs to Manifolds - Weak and Strong Pointwise Consistency of Graph Laplacians. Em: *COLT'05: Proceedings of the 18th Conference on Learning Theory*, 2005, páginas 470–485.

HEIN, M.; AUDIBERT, J.-Y.; LUXBURG, U. Graph laplacians and their convergence on random neighborhood graphs. *Journal of Machine Learning Research*, v. 8, páginas 1325–1368, 2007.

HOUHOU, N.; BRESSON, X.; SZLAM, A.; CHAN, T. F.; THIRAN, J.-P. Semi-supervised segmentation based on non-local continuous min-cut. Em: TAI, X.-C.; MORKEN, K.; LYSAKER, M.; LIE, K.-A., eds. *International Conference on Scale Space and Variational Methods in Computer Vision*, Springer, 2009, páginas 112–123 (*Lecture Notes in Computer Science*, v.5567).

HUANG, B.; JEBARA, T. Loopy belief propagation for bipartite maximum weight b-matching. *Journal of Machine Learning Research - Proceedings Track*, páginas 195–202, 2007.

JEBARA, T.; WANG, J.; CHANG, S.-F. Graph construction and b-matching for semi-supervised learning. Em: *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, 2009, páginas 441–448.

JENSEN, D.; NEVILLE, J.; GALLAGHER, B. Why collective inference improves relational classification. Em: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2004, páginas 593–598.

JEONG, H.; MASON, S. P.; BARABASI, A. L.; OLTVAI, Z. N. Lethality e centrality in protein networks. *Nature*, v. 411, páginas 41–42, 2001.

JEONG, H.; TOMBOR, B.; ALBERT, R.; OLTVAI, Z. N.; BARABASI, A. L. The large-scale organization of metabolic networks. *Nature*, v. 407, páginas 651–655, 2000.

JI, M.; SUN, Y.; DANILEVSKY, M.; HAN, J.; GAO, J. Graph Regularized Transductive Classification on Heterogeneous Information Networks. Em: *ECML PKDD'10: Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part I*, 2010, páginas 570–586.

KATZ, L. A new status index derived from sociometric analysis. *Psychometrika*, v. 18, n. 1, páginas 39–43, 1953.

KENNEDY, L. S.; CHANG, S.-F.; KOZINTSEV, I. To search or to label?: predicting the performance of search-based automatic image classifiers. Em: WANG, J. Z.; BOUJEMAA, N.; CHEN, Y., eds. *Multimedia Information Retrieval*, ACM, 2006, páginas 249–258.

KHOSRAVI, H.; BINA, B. A Survey on Statistical Relational Learning. Em: *AI'10: Proceedings of the 23rd Canadian Conference on Advancese in Artificial Intelligence*, 2010a, páginas 256–268.

KHOSRAVI, H.; BINA, B. A survey on statistical relational learning. Em: *Proceedings of the 23rd Canadian Conference on Advances in Artificial Intelligence*, AI'10, Berlin, Heidelberg: Springer-Verlag, 2010b, páginas 256–268 (*AI'10*,).

KNEES, P.; SCHNITZER, D.; FLEXER, A. Improving neighborhood-based collaborative filtering by reducing hubness. Em: *Proceedings of International Conference on Multimedia Retrieval*, ACM, 2014, páginas 161–168.

KURZ, L.; BENTEFTIFA, M. H. *Analysis of variance in statistical image processing*. Cambridge University Press, 2006.

LANCICHINETTI, A.; FORTUNATO, S.; RADICCHI, F. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, v. 78, n. 4, p. 046110, 2008.

LARTILLOT, O.; TOIVIAINEN, P.; EEROLA, T. A matlab toolbox for music information retrieval. Em: PREISACH, C.; BURKHARDT, H.; SCHMIDT-THIEME, L.; DECKER, R.,

eds. *Studies in Classification, Data Analysis, and Knowledge Organization*, Springer, 2007, páginas 261–268.

LI, S.; FU, Y. Low-rank coding with b-matching constraint for semi-supervised classification. Em: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, AAAI Press, 2013, páginas 1472–1478.

LIBEN-NOWELL, D.; KLEINBERG, J. The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology*, v. 58, n. 7, páginas 1019–1031, 2007.

LIDY, T.; RAUBER, A. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. Em: *International Conference on Music Information Retrieval*, 2005, páginas 34–41.

LÜ, L.; ZHOU, T. Link prediction in weighted networks: The role of weak ties. *Europhysics Letters*, v. 89, n. 1, p. 18001, 2010.

LÜ, L.; ZHOU, T. Link prediction in complex networks: A survey. *Physica A*, v. 390, n. 6, páginas 1150 – 1170, 2011.

LU, Q.; GETOOR, L. Link-based vev. Em: *ICML*, 2003, páginas 496–503.

LUXBURG, U. A tutorial on spectral clustering. *Statistics and Computing*, v. 17, n. 4, páginas 395–416, 2007.

M. BAYATI, D. SHAH, M. S. Maximum weight matching via max-product belief propagation. *Proceedings of International Symposium on Information Theory*, páginas 1763–1767, 2005.

MACSKASSY, S. A.; PROVOST, F. J. A simple relational classifier. Em: *2nd Workshop on Multi-Relational Data Mining*, 2003.

MACSKASSY, S. A.; PROVOST, F. J. Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research*, v. 8, páginas 935–983, 2007a.

MACSKASSY, S. A.; PROVOST, F. J. Classification in networked data: A toolkit and a univariate case study. *JMLR*, v. 8, páginas 935–983, 2007b.

MAIER, M.; HEIN, M.; LUXBURG, U. Cluster identification in nearest-neighbor graphs. Em: *ALT'07: Proceedings of the 18th International Conference on Algorithmic Learning Theory*, 2007a, páginas 196–210.

MAIER, M.; HEIN, M.; LUXBURG, U. Cluster identification in nearest-neighbor graphs. Em: *Algorithmic Learning Theory*, Springer, 2007b, páginas 196–210 (*Lecture Notes in Computer Science*, v.4754).

MAIER, M.; LUXBURG, U.; HEIN, M. Influence of graph construction on graph-based clustering measures. *Advances in Neural Information Processing Systems*, v. 21, páginas 1025–1032, 2009.

MANDEL, M. I.; ELLIS, D. P. Song-level features and support vector machines for music classification. Em: *International Conference on Music Information Retrieval*, 2005, páginas 594–599.

MANEEWONGVATANA, S.; MOUNT, D. M. On the efficiency of nearest neighbor searching with data clustered in lower dimensions. Em: ALEXANDROV, V. N.; DONGARRA, J.; JULIANO, B. A.; RENNER, R. S.; TAN, C. J. K., eds. *International Conference on Computational Science*, Springer, 2001, páginas 842–851 (*Lecture Notes in Computer Science*, v.2073).

MARTIN, D.; FOWLKES, C.; TAL, D.; MALIK, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Em: *IEEE International Conference on Computer Vision*, 2001, páginas 416–423 vol.2.

MCKAY, C. *Automatic Genre Classification of MIDI Recordings*. Dissertação de Mestrado, McGill University, 2004.

MERINGER, M. Fast generation of regular graphs and construction of cages. *Journal of Graph Theory*, v. 30, páginas 137–146, 1999.

MICTHELL, T. M. *Machine Learning*. McGraw-Hill Education, 1997.

MILGRAM, S. The Small World Problem. *Psychology Today*, v. 1, n. 1, páginas 61–67, 1967.

MORTENSEN, E. N.; BARRETT, W. A. Intelligent scissors for image composition. Em: *Proceedings of the 22nd annual conference on computer graphics and interactive techniques*, 1995, páginas 191–198.

MURATA, T.; MORIYASU, S. Link prediction of social networks based on weighted proximity measures. Em: *International Conference on Web Intelligence*, 2007, páginas 85–88.

NEWMAN, M. Assortative mixing in networks. *Physical Review Letters*, v. 208701, n. 20, 2002.

NEWMAN, M. E. J. The structure and function of complex networks. *SIAM Review*, v. 45, n. 2, páginas 167–256, 2003.

NIELS, R. *Dynamic time warping*. Dissertação de Mestrado, University of Nijmegen, 2004.

OZAKI, K.; SHIMBO, M.; KOMACHI, M.; MATSUMOTO, Y. Using the mutual k-nearest neighbor graphs for semi-supervised classification of natural language data. Em: *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, Association for Computational Linguistics, 2011, páginas 154–162.

PAMPALK, E.; FLEXER, A.; WIDMER, G.; ET AL. Improvements of audio-based music similarity and genre classificaton. Em: *International Conference on Music Information Retrieval*, 2005, páginas 634–637.

PERROT, D.; GJERDIGEN, R. Scanning the dial: An exploration of factors in the identification of musical style. Em: *Proceedings of the 199 Society for Music Perception and Cognition*, 1999, p. 88.

PORIA, S.; GELBUKH, A.; HUSSAIN, A.; BANDYOPADHYAY, S.; HOWARD, N. Music genre classification: A semi-supervised approach. Em: *Pattern Recognition*, páginas 254–263, 2013.

PROTIERE, A.; SAPIRO, G. Interactive image segmentation via adaptive weighted distances. *IEEE Transactions on Image Processing*, v. 16, n. 4, páginas 1046–1057, 2007.

RADOVANOVIC, M.; NANOPOULOS, A.; IVANOVIC, M. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, v. 11, páginas 2487–2531, 2010.

RAEDT, L. D. *Logical and Relational Learning*. Cognitive Technologies - Springer, 2008.

RAGHAVAN, U. N.; ALBERT, R.; KUMARA, S. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, v. 76, p. 036106, 2007.

RAND, W. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, v. 66, n. 336, páginas 846–850, 1971.

REDNER, S. How popular is your paper? an empirical study of the citation distribution. *European Physical Journal B - Condensed Matter and Complex Systems*, v. 4, n. 2, páginas 131–134, 1998.

ROHBAN, M. H.; RABIEE, H. R. Supervised neighborhood graph construction for semi-supervised classification. *Pattern Recognition*, v. 45, n. 4, páginas 1363–1372, 2012.

ROSENBERG, A.; HIRSCHBERG, J. V-measure: A conditional entropy-based external cluster evaluation measure. Em: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007, páginas 410–420.

ROTHER, C.; KOLMOGOROV, V.; BLAKE, A. Grabcut-interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 2004.

SABIDUSSI, G. The centrality index of a graph. *Psychometrika*, v. 31, páginas 581–603, 1996.

SAKOE, H.; CHIBA, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, v. 26, n. 1, páginas 43–49, 1978.

SANKARANARAYANAN, J.; SAMET, H.; VARSHNEY, A. A fast all nearest neighbor algorithm for applications involving large point-clouds. *Computers & Graphics*, v. 31, n. 2, páginas 157–174, 2007.

SAUL, L. K.; ROWEIS, S. T. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, v. 4, páginas 119–155, 2003.

SCARINGELLA, N.; MLYNEK, D. A mixture of support vector machines for audio classification. *Music information retrieval evaluation exchange*, 2005.

SCHAEFFER, S. Graph clustering. *Computer Science Review*, v. 1, n. 1, páginas 27–64, 2007.

SHAO, X.; XU, C.; KANKANHALLI, M. S. Unsupervised classification of music genre using hidden markov model. Em: *IEEE International Conference on Multimedia and Expo - ICME'04.*, 2004, páginas 2023–2026.

SORIANO, A. *Metáforas visuais alternativas para layouts gerados por projeções multidimensionais: um estudo de caso na visualização de músicas*. Dissertação de Mestrado, University of São Paulo, 2013.

SZUMMER, M.; JAAKKOLA, T. Partially labeled classification with markov random walks. *Advances in Neural Information Processing Systems*, v. 14, páginas 945–952, 2002.

TA, V.-T.; LÉZORAY, O.; ELMOATAZ, A.; SCHÜPP, S. Graph-based tools for microscopic cellular image segmentation. *Pattern Recognition*, v. 42, n. 6, páginas 1113–1125, 2009.

TAGUE-SUTCLIFFE, J. An introduction to informetrics. *Information Processing and Management*, v. 28, n. 1, páginas 1–3, 1992.

TASKAR, B.; WONG, M.; ABBEEL, P.; KOLLER, D. Link prediction in relational data. Em: THRUN, S.; SAUL, L.; SCHÖLKOPF, B., eds. *Advances in Neural Information Processing Systems (NIPS) 16*, Cambridge, MA: MIT Press, 2004.

TONG, S.; KOLLER, D. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, v. 2, páginas 45–66, 2002.

TSUDA, K.; SHIN, H.; SCHÖLKOPF, B. Fast protein classification with multiple networks. *Bioinformatics*, v. 21, n. 2, páginas 59–65, 2005.

TZANETAKIS, G.; ERMOLINSKYI, A.; COOK, P. Pitch histograms in audio and symbolic music information retrieval. Em: *International Conference on Music Information Retrieval*, 2002, páginas 31–38.

ULLMAN, J. D. *Principles of Database Systems*. Computer Science Press, Rockville, 1982.

VALVERDE-REBAZA, J.; LOPES, A. Exploiting behaviors of communities of Twitter users for link prediction. *SNAM*, v. 3, n. 4, páginas 1063–1074, 2013.

VALVERDE-REBAZA, J. C.; SORIANO, A.; BERTON, L.; OLIVEIRA, M. C. F.; ANDRADE LOPES, A. Music genre classification using traditional and relational approaches. Em: *2014 Brazilian Conference on Intelligent Systems, BRACIS*, 2014, páginas 259–264.

VEGA-OLIVEROS, D. A.; BERTON, L.; EBERLE, A. M.; ANDRADE LOPES, A.; ZHAO, L. Regular graph construction for semi-supervised learning. *Journal of Physics: Conference Series*, v. 490, n. 1, páginas 012022–1–012022–4, 2014.

WANG, B.; TANG, H.; GUO, C.; XIU, Z. Entropy optimization of scale-free networks' robustness to random failures. *Physica A: Statistical Mechanics and its Applications*, v. 363, n. 2, páginas 591–596, 2006.

WANG, F.; ZHANG, C. Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, v. 20, n. 1, páginas 55–67, 2008.

WANG, J.; COHEN, M. F. An iterative optimization approach for unified image segmentation and matting. Em: *International Conference on Computer Vision*, IEEE Computer Society, 2005, páginas 936–943.

WANG, Y.; KITSUREGAWA, M. Link based clustering of web search results. Em: WANG, X.; YU, G.; LU, H., eds. *Advances in Web-Age Information Management*, v. 2118 de *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, páginas 225–236, 2001.

WASSERMAN, S.; PATTISON, P. Logit models and logistic regressions for social networks: I. an introduction to markov graphs and p*. *Psychometrika*, v. 61, n. 3, páginas 401–425, 1996.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of 'small-world' networks. *Nature*, v. 393, páginas 440–442, 1998.

XU, Y.; ZHANG, C.; YANG, J. Semi-supervised classification of musical genre using multi-view features. Em: *International Computer Music Association*, 2005, páginas 5–9.

YASLAN, Y.; CATALTEPE, Z. Audio genre classification with semi-supervised feature ensemble learning. Em: *IWMLM*, 2009.

YU, S. X.; SHI, J. Segmentation given partial grouping constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 2, páginas 173–183, 2004.

ZHOU, D.; BOUSQUET, O.; LAL, T. N.; WESTON, J.; SCHLKOPF, B. Learning with local and global consistency. *Advances in Neural Information Processing Systems*, v. 16, páginas 321–328, 2004.

ZHU, X. *Semi-supervised learning literature survey*. Relatório Técnico 1530, University of Wisconsin-Madison, computer Sciences, 2005.

ZHU, X.; GHAHRAMANI, Z. *Learning from labeled and unlabeled data with label propagation*. Relatório Técnico, Carnegie Mellon University, Pittsburgh, 2002.

ZHU, X.; GHAHRAMANI, Z.; LAFFERTY, J. Semi-Supervised Learning using Gaussian Fields and Harmonic Functions. Em: *ICML'03: Proceedings of the Twentieth International Conference on Machine Learning*, 2003, páginas 912–919.